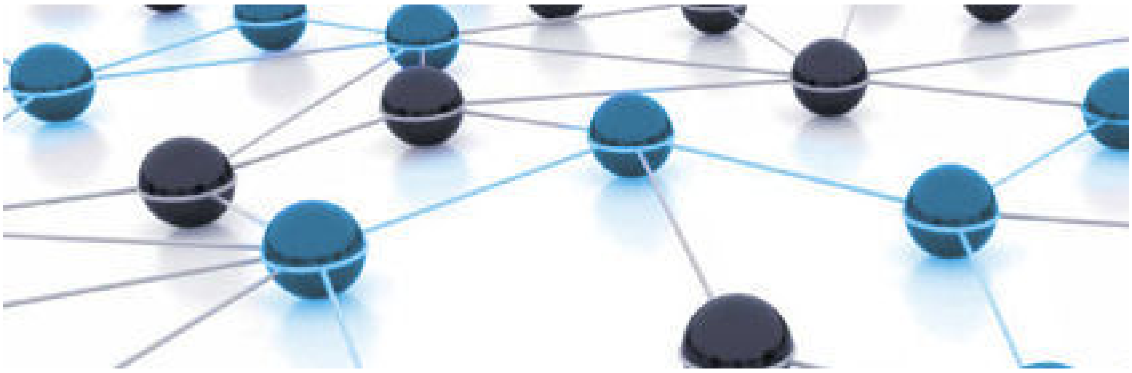

Multivariate Data Methods for ASD Screening

Sofia de Sousa Fernandes Filipe Belém

Thesis to obtain the Master of Science Degree in

Biological Engineering

Supervisors: Dr. João Alberto Pacheco Marques de Vasconcelos e Sá
Prof. Dr. Maria Raquel Murias dos Santos Aires Barros

Examination Committee

Chairperson: Prof. Dr. Miguel Nobre Parreira Cacho Teixeira
Supervisor: Dr. João Alberto Pacheco Marques de Vasconcelos e Sá
Member of the Committee: Prof. Dr. José Monteiro Cardoso de Menezes

January 2021

Preface

The work presented in this thesis was performed at the company Hovione FarmaCiencia (Lisbon, Portugal), during the period February-August 2020, under the supervision of Dr. João Sá. The thesis was co-supervised at Instituto Superior Técnico by Prof. Maria Raquel Aires Barros.

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

After five intense years in this Institution, many people have been present from the beginning and so many others became relevant along the way. There aren't enough words to thank the people who have made possible the grand finale of this course, but nevertheless a few paragraphs are in order.

First and foremost, I would like to thank my family: to my parents, Anabela and António, and to my sister Beatriz, a huge thank you, because none of this would have been possible without you. Thank you for putting up with my stressful exams periods, with the intense studying sessions and for always supporting me even in the hardest times.

To my boyfriend, Gonçalo, I want to thank for all the hours spent motivating me with hugs and delicious meals, for all the love and support, for never letting me give up and for always, always believing in me and my abilities.

To my friends, who have been one of the best parts of university, I want to give my sincerest gratitude. You have been there through thick and thin, in the most stressful projects, in some of the most intense periods of my academic path, and I truly don't think I could have made it without you. Beatriz G., Clara, Francisca, Irina and Madalena, you are without a doubt an amazing group to be a part of, and I honestly couldn't have asked for better company during these years. You were truly indispensable, and I thank you so much for that. Hélder, you are one of the people that makes me laugh the most, and because of that you always made me feel better during hard times even without knowing; you became a truly great friend, so thank you. Rita G., you were the first friend I made in college and you have stayed a good friend ever since, and therefore I would also like to thank you for having been there.

Finally, I would like to give a big thank you to the people who made this thesis possible at Hovione. There were a lot of people who contributed to this project with various inputs and valuable advice. However, I would like to give a very special thank you to my supervisor, João Sá, for all the indispensable guidance, for being so invested in my success and for being available every single time I needed advice – I couldn't have asked for a better supervisor to develop this project with. I would also like to thank João Henriques for all the expert recommendations and for always being willing to help me.

Abstract

Amorphous Solid Dispersions (ASDs) – solid-solid dispersions of a drug within a polymer matrix to enhance solubility properties – are a prominent way to formulate low solubility drugs, but the variables that influence its success are still poorly understood . Two important characteristics in the ASD formulation process are the drug loading (drug/polymer ratio) and the spring and parachute (SP) effect. The objective of this work was to develop computational tools to predict these outputs to simplify the initial steps of ASD screening. To predict the maximum drug loading, the successful models developed were a partial least squares (PLS) model with drug, polymer and interaction variables as inputs and a PLS models with drug-only inputs. Defining a threshold of 10% difference to the real value, accuracies of 71% and 57% were obtained for a set of commercial ASDs, and 81% and 75% for a set of internal ASDs (respectively) . It was shown that these models performed much better than the currently used methodology (Flory-Huggins theory). To predict the SP output, two models were developed: a random forest and a neural network (accuracies on external dataset of 67% and 56%, respectively). These models can be used to explore a "best-polymer" output in a preliminary phase. The new workflow would involve running the new drug through "drug-only-PLS" model to exclude non-promising drugs, run SP model to exclude non-promising polymers, run drug combined with remaining polymers through "all-features-PLS" to obtain drug loading for those formulations, and experimentally formulate ASDs with predicted drug loading.

Keywords

Amorphous Solid Dispersions; Multivariate Data Analysis; Machine Learning; Partial Least Squares; Artificial Neural Network; Random Forest

Resumo

Dispersões sólidas amorfas (ASDs) – dispersões sólido-sólido de um fármaco numa matriz polimérica para melhoramento da solubilização – são formas proeminentes de formular fármacos com baixa solubilidade, mas as variáveis que influenciam o seu sucesso ainda não são totalmente compreendidas. Duas características importantes na sua formulação são a carga de fármaco (rácio fármaco/polímero) e o efeito *spring and parachute* (SP). Este trabalho teve como objetivo desenvolver ferramentas computacionais para previsão destes outputs e simplificar os passos iniciais da triagem de ASDs. Para prever a carga, os modelos com sucesso desenvolvidos foram um modelo mínimos quadrados parciais (PLS) com variáveis do fármaco, polímero e interação, e um modelo PLS apenas com variáveis do fármaco. Definindo como limite 10% de diferença face ao valor real, foram obtidas precisões de 71% e 57% para um conjunto de ASDs comerciais, e 81% e 75% para um conjunto de ASDs internos, respetivamente. Estes modelos tiveram um desempenho muito superior face à metodologia atualmente utilizada (teoria de Flory-Huggins). Para prever o output SP, desenvolveram-se dois modelos: uma floresta aleatória e uma rede neural artificial (precisões de 67% e 56% num conjunto de observações externas, respetivamente). Estes podem ser utilizados para explorar um output "melhor-polímero" numa fase preliminar. A nova metodologia envolveria correr o novo fármaco no "PLS-fármaco-apenas" para excluir fármacos não promissores, correr o modelo SP para excluir polímeros inadequados, correr o fármaco combinado com os restantes polímeros no "PLS-todas-variáveis" para obter a carga para essas formulações, e formular experimentalmente os ASDs com a carga prevista.

Palavras Chave

Dispersões Sólidas Amorfas; Análise de Dados Multivariada; Machine Learning; Mínimos Quadrados Parciais; Rede Neural Artificial; Floresta Aleatória

Contents

List of Figures

List of Tables

Abbreviations

ANN Artificial Neural Network

API Active Pharmaceutical Ingredient

ASD Amorphous Solid Dispersion

AUC Area Under the Curve

BCS Biopharmaceutics Classification System

DSC Differential Scanning Calorimetry

F-H Flory-Huggins

FNR False Negative Rate

FPR False Positive Rate

GI Gastrointestinal

GMP Good Manufacturing Practices

GPN Gelling Polymer Network

HME Hot Melt Extrusion

HPC HydroxyPropyl Cellulose

HPMCAS HydroxylPropyl MethylCelluloseAcetate Succinate

HSP Hansen Solubility Parameters

HSPiP Hansen Solubility Parameters in Practice

LogP Partition Coefficient

MW Molecular Weight

PCA Principal Component Analysis

pKa Logarithmic Acid Dissociation Constant

PLM Polarized Light Microscopy

PLS Partial Least Squares

PVP Poly (Vinyl Pyrrolidone)

PVPVA Poly (1-VinylPyrrolidone-co-Vinyl Acetate)

RF Random Forest

ROC Receiver Operating Characteristic

SCG Scaled Conjugate Gradient

SD Spray-Drying

SDD Spray-Dried Dispersions

T_g Glass Transition Temperature

T_m Melting Temperature

TNR True Negative Rate

TPP Target Product Profile

TPR True Positive Rate

VIP Variable Importance in Projection

1

Introduction

Contents

1.1 Background

Oral administration, in particular solid oral doses, is the preferred and most common route for drug delivery, given that it's cost effective, convenient for the patient, easily handled, carries less sterility constraints and implies lower manufacturing costs [1–3]. For the active pharmaceutical ingredient (API) to be transported to its physiological target, it must be released and absorbed in the gastrointestinal (GI) tract, where it will enter the circulatory system. This implies that the bioavailability of a given drug is dependent on its ability to dissolve in the GI fluid (solubility) and to pass through the intestinal membrane (permeability) [3]. Based on this two concepts, a regulatory mechanism was created: the Biopharmaceutics Classification System (BCS). This system divides the compounds into four classes (Figure 1.1 [4]) and can be used, for example, to determine which drugs should not be clinically tested unless submitted to adequate formulation techniques.

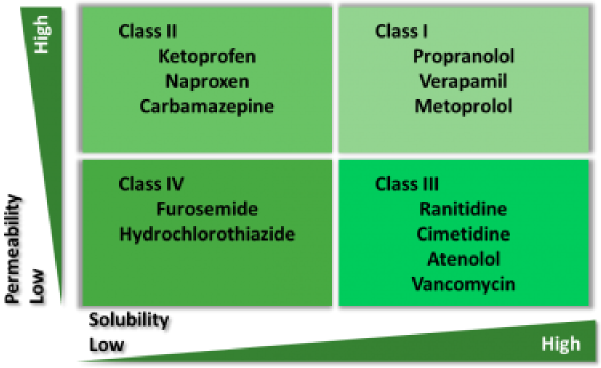

Figure 1.1: Class division of compounds by BCS: Class I – High Solubility, High Permeability; Class II – Low Solubility, High Permeability; Class III – High Solubility, Low Permeability; Class IV – Low Solubility, Low Permeability. [4]

Since the decade of 1990, the use of high-throughput *in vitro* screening assays and combinatorial chemistry, x-ray diffraction of proteins and computational tools has increased the understanding of how small molecules bind to targets, leading to the rapid discovery of a vast array of molecules with high potency, binding affinity, and selectivity that could come to fulfill unmet medication needs [3, 5]. This approach brought new challenges to the pharmaceutics industry, since it's biased toward hydrophobic and crystalline molecules that are usually inserted in the classes II and IV of the BCS and do not obey Lipinski's rule of five (a rule formulated by Christopher A. Lipinski in 1997 which states that generally, an orally active drug obeys at least three of following criteria: has no more than 5 hydrogen bond donors; has no more than 10 hydrogen bond acceptors; has a molecular mass less than 500 Dalton; has an octanol-water partition coefficient (log P) that does not exceed 5 [6]). These drugs present low solubility in aqueous media (below 100 $\mu g/mL$) [5], resulting in poor *in vivo* dissolution and poor or variable bioavailability, and thus are not viable for oral administration [7].

Poor aqueous solubility results in a low dissolution rate, which is specially problematic for drugs with a restrict absorption window as they might dissolve after passing their absorptive sites [8]. The relation between solubility and dissolution rate is given by the Noyes–Whitney equation [9], where dM/dt is the dissolution rate, A is the specific surface area of the drug particle, D is the diffusion coefficient, h is the diffusion layer thickness, C_s is the saturation solubility and C_t is the drug concentration at time t (Equation 1.1).

$$\frac{\mathrm{d}M}{\mathrm{d}t} = \frac{AD\left(C_s - C_t\right)}{h} \tag{1.1}$$

Because many of these compounds have the potential to be safe and efficacious, it is essential that these solubility and bioavailability challenges are overcome. The diffusion coefficient is dependent on the molecular weight of the compound and on the viscosity of the GI fluids, which is highly subject to intra- and inter-subject variability. The diffusion layer thickness, being dependent on the hydrodynamics during GI transit, is also highly variable. It is, then, accepted that these parameters are less adequate targets for bioavailability optimization. A more suitable approach is the manipulation of the saturation solubility and specific surface area, which has given rise to a large variety of formulation strategies, such as via particle size reduction, improved wetting, chemically modifying the compound or by changing the physical state of the drug in the formulation, such as amorphous solids and solid dispersions [8].

1.2 Amorphous Solid Dispersions: A New Hope

A successful and effective approach to this challenge is the use of an amorphous solid dispersion (ASD). These formulations consist of a solid-solid blend of the API within a polymer excipient (the API molecules are uniformly dispersed within a polymer matrix); the mixture is vitrified so that the crystalline drug transforms into meta-stable amorphous glass [5]. Amorphous pharmaceutic products are characterized by its solid-state nature and lack of distinct intermolecular arrangement without crystalline structure and, consequently, with poor thermodynamic stability, meaning they have associated a higher energy state (entropy, enthalpy and free Gibbs energy), providing enhanced solubility properties: in a crystalline structure, the dissolution process begins with the breaking of the crystal structure in order to occur molecular dissolution, a step that is abbreviated in amorphous structures [7, 10]. ASDs are preferred instead of pure amorphous APIs because the physical stability of the latter is usually not sufficient to avoid rapid API crystallization [11]. In addition to the increased bioavailability, the final product can be delivered through a tablet or capsule, providing greater chemical stability and patient compliance than solutions or semisolid dosage forms [12]. ASDs can also be manufactured using scalable processes, such as spray drying (SD) and hot melt extrusion (HME), causing it to receive ever more attention from the scientific community [5].

The principal development goals for ASD formulation are *in vivo* performance (increase solubility, supersaturation conditions and extended drug exposure); physical stability (amorphous stability, polymer miscibility and inhibition of recrystallization); chemical stability (impurity profile, chemical compatibility and toxicological effects); and manufacturability (process selection, processability and scale-up) [13].

1.2.1 Main Production Methodologies

The number of good manufacturing practices (GMP) compliant processes to produce ASDs is reduced, because the majority of laboratorial processes are hard to scale-up and do not fulfil GMP requirements (such as contact materials, reproducibility and sanitation). The most common industrial scale processes for ASD manufacturing are solvent evaporation and melting [7].

1.2.1.A Melting: Hot Melt Extrusion

Hot melt extrusion (HME) has been explored as a promising procedure for industrial production of solid dispersions. It consists in the high rotation speed extrusion of the previously mixed drug and carriers at melting temperature, for a small period. The resulting product is cooled at room temperature, collected and then milled into a powder or granule form. This technique is solvent free, easily scaled up and allows continuous processing, which is highly advantageous because it allows huge versatility [7,14].

It is necessary to choose adequate composition and process parameters for the successful development of a solid dispersion by HME. The most important variable is the screw design, but other crucial parameters in the definition of the properties of the final product are feed rate, temperature and rotation speed. The screw speed and feed rate are related to shear stress and mean residence time, which can affect the dissolution rate and stability of the final products. The process requires a minimum temperature in order to reduce the torque needed to rotate the screw and allow an efficient process; HME requires high energy input due to these temperatures used and to the shear forces, which constitutes an important drawback, in addition to the inadequacy to process thermolabile compounds [7].

1.2.1.B Solvent Evaporation: Spray-Drying

Spray-drying (SD) is a well-established process used in the transformation of solutions, emulsions or suspensions into dry powder. The feed is pumped to an atomizer inside a drying chamber, where it’s broken into a plume of small droplets. These are mixed with a hot stream of drying gas (typically nitrogen) in the drying chamber, resulting in the transference of heat from the gas stream to the droplets. This will provide the latent heat of vaporization necessary for the rapid evaporation of the solvent from the droplets. The particle morphology and size and the density of the spray-dried dispersion (SDD) can be controlled by manipulating the inlet and outlet temperature and both the feed rates [15, 16].

Using the adequate SD conditions (sufficiently high drying gas inlet temperature and sufficiently high ratio of gas to liquid flow) results in the rapid removal of solvent from the droplets. This is essential to induce the rapid solidification of the droplets, a requirement to prevent phase separation of the drug from the polymer. Additionally, the spray-drying technology can be applied to a wide range of scales, including full-scale commercial production. This process can be manipulated to produce particles with properties favorable to the process of formulation of solid oral dosage forms, such as tablets and capsules [7,15,17].

Amorphous solid dispersions manufactured via spray-drying processes are entitled ASD SDs, and will be the main focus of the present work.

1.2.2 Important Characteristics, Advantages and Challenges

Spray-dried amorphous solid dispersions present a number of advantages for low-solubility API delivery. They rapidly dissolve due to their high free energy, enhance the oral absorption of poorly soluble compounds by sustaining supersaturated concentrations of the drug in the GI fluid, provide a physically stable drug form avoiding crystallization or phase separation, provide a solid drug form that can be manufactured in a reproducible, controllable and scalable process, and provide a technology that is applicable to diverse insoluble components across a vast range of physicochemical properties. Being a kinetically stabilized amorphous form, ASD SDs also overcome some challenges that other amorphous solid formulations present, such as physical instability and poor manufacturability at large scale. However, ADS also present some challenges, such as the requirement of complex manufacturing processes, the high thermodynamic instability due to the high free energy and the fact that they are not adequate for all APIs [11, 15] .

The formation of amorphous drug/polymer nanostructures and small aggregates of these structures is crucial to the performance of ASD SDs, as these are rapidly formed when introduced in an aqueous media, produce an enhanced free-drug concentration relative to the crystalline form of the drug, sustain a high free-drug concentration by replacing it as it is absorbed over a biologically relevant time frame and are stable in aqueous suspension, inhibiting compound crystallization. Above the solubility of the amorphous drug, drug/polymer colloids begin to form; these are not dissolved, but its small size and high free energy causes rapid dissolution and replacing of the absorbed drug [15].

The dissolution behaviour of ASDs is often described by the ”spring and parachute” model (Figure 1.2). The ”spring” represents the initial phase where the drug is propelled into a solution as the polymer matrix dissolves, resulting in a supersaturated solution. In order to maintain the drug in the supersaturated state long enough for it to be absorbed, the polymer must also inhibit precipitation of the drug - the ”parachute”: precipitation inhibitors interact with the API molecules in solution, slowing down critical steps in the process of drug crystallization [18].

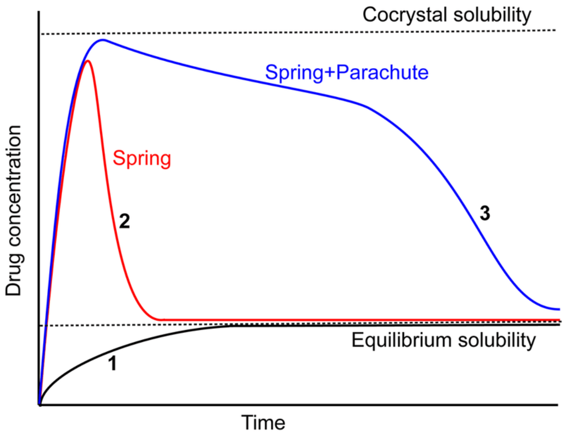

Figure 1.2: Schematic representation of the drug concentration–time profiles, illustrating the spring and parachute effect of supersaturating drug delivery systems. Profile 1: dissolution of the drug in the crystalline form; profile 2: dissolution of a higher energy cocrystal with supersaturation/ precipitation process; profile 3: dissolution of a cocrystal with precipitation inhibitors that acts as a parachute [18].

A very important aspect to be taken into account when manufacturing ASD SDs is the crystallization tendency of a given compound; two good parameters to evaluate this propensity are the T_m/T_g ratio and the partition coefficient, $logP$. T_g, the glass transition temperature, represents the temperature at which amorphous materials transition from a hard, glassy state into a viscous state, while T_m, the melting temperature, represents the temperature at which a given substance changes from solid state to liquid. A high T_m implies a high crystallization tendency due to the high thermodynamic driving force; a low T_g poses a small kinetic barrier to molecular diffusion and, therefore, allows higher mobility, implying high crystallization tendency. Therefore, the higher the T_m/T_g ratio, the bigger the crystallization tendency. $LogP$, the partition coefficient, represents the ratio of concentrations of a compound in a mixture of two immiscible solvents at equilibrium. A high $logP$ value means the compound is highly hydrophobic, and therefore poorly soluble in water.

A useful visual representation based on the melting temperature T_m and the $LogP$ is the one shown in Figure 1.3 [11]. This technology map presents the applicability of ASDs compared to other technologies, based on the assessment of the dose/solubility ratio required to achieve high oral bioavailability for the compound. The top-most solid diagonal line in the map represents the maximal solubility (S_{max}) of the lowest-energy, neutral form of the compound, calculated through Equation 1.2 (which assumes that the compound is a liquid at ambient temperature) [11].

$$S_{\max}(\mathrm{mg/mL}) = 1000 \times 10^{(-\log \mathrm{P})} \tag{1.2}$$

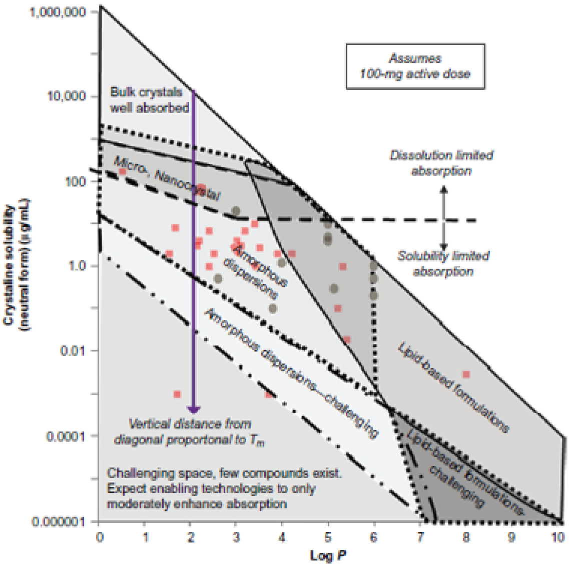

Figure 1.3: Crystalline solubility versus $LogP$ of compounds showing applicability of ASDs in comparison with other technologies. This map was traced by Vig, B., and M. Morgen based on: solubility of the neutral bulk crystalline form at pH 6.5, assuming the fraction of dose ionized at pH 7 is less than 50%, the molecular weight of the compound is less than approximately 700 Da (so permeability is not adversely affected by molecular size), and the human dose is 100 mg of active dose [11].

At a constant $logP$, decreasing aqueous solubility is primarily driven by an increase in the solid state interactions, which are directly proportional to T_m – the furthest down a compound is, the higher its T_m. In the upper region of the map, crystalline solubility is high enough to achieve acceptable bioavailability using the crystalline form of the drug. When solubility falls below 1mg/mL, particle-reduction technologies (such as micro or nanocrystals) provide an acceptable solution in terms of bioavailability, by increasing the surface area in order to overcome the slow dissolution rate of the crystal. However, when solubility decreases even more, such technologies become obsolete due to inadequate absorption even with high dissolution rates. In this case it becomes necessary that the concentration of the drug is increased over its solubility limit, and ASDs are used for this purpose. To be noted that for compounds with high lipophilicity, the addition of lipidic excipients can help solubilize and enhance transport of the compound through the aqueous boundary layer [11].

1.2.3 ASD SD's Key Performance Parameters

ASD SDs performance is dependent on the formulation composition (API and polymer characteristics, API/polymer ratio, and other excipients), and on the manufacturing process (choice of spray solvents and process parameters). These two groups are interdependent, and it is essential to understand the relation between formulation composition and manufacturing process to ensure consistent production of the desired product. In the present work, emphasis will be given to the formulation composition.

1.2.3.A API Properties

The physicochemical and biopharmaceutical properties of the API are crucial to the ASD SD performance, and a fundamental understanding of the impact of API properties can guide the development of a robust SDD formulation. Key API properties include the previously explained parameters (T_m, T_g and $logP$); Kauzmann temperature (T_k), the temperature at which the difference between the liquid and solid phase entropies becomes zero; crystalline form; logarithmic acid dissociation constant (pka); molecular weight; miscibility in polymers; hydrogen bond donors and acceptors; epithelial membrane permeability; solubility in aqueous media; solubility in spray solvents; and chemical stability [1, 11, 15, 17, 19, 20]. It has been demonstrated that physicochemical properties of the compounds that had fast crystal growth rates included lower molecular weights, high T_m values, lower T_g values, fewer rotatable bonds, lower melt entropy, lower melt viscosity and higher crystal densities [20, 21].

As explained before, the T_m/T_g ratio and $logP$ are two very important parameters in ASD SD formulation. An API's tendency to crystallize and its lipophilicity can be used to guide the selection of the polymer and excipients, and to determine the maximum drug loading. Generally, compounds that have higher tendency to crystallize require a higher polymer/drug ratio to resist crystallization. Compounds with high $logP$ values may also require a larger polymer/drug ratio and excipients in order to reduce drug-drug interactions, improve wetting, and achieve the desired dissolution rate [11, 15]. Wetting and dissolution can also be improved by adding a surfactant to the formulation [22].

The graphical representation of the T_m/T_g ratio versus $logP$ divides the substances into four groups, and provides a gross estimate of the adequate drug load for each group (Figure 1.4 [15]).

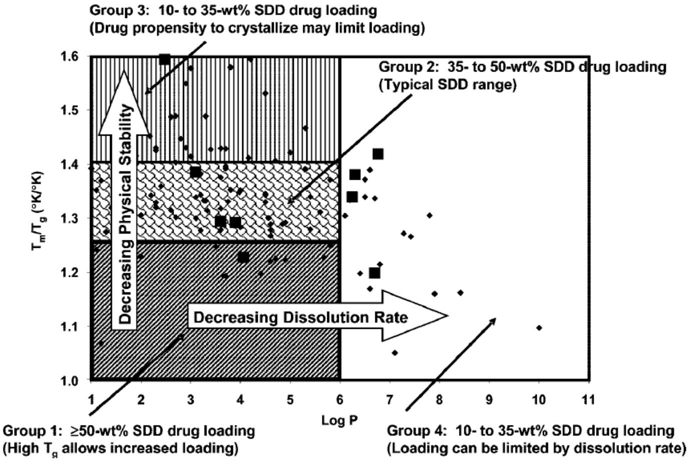

Figure 1.4: Graphical representation of the T_m/T_g versus $logP$ for 139 low-solubility compounds, from a study by Friesen, Dwayne T., *et al.* The big squares correspond to compounds studied by the authors [15].

Different drug forms can have different solubilities in spray solvents, which can affect processing ease, so this parameter needs close evaluation. One should also examine the API's tendency to form a solvate with the spray solvent, because these will have lower solubility. Care must also be taken when dealing with acidic or basic API's, as acid-base reactions may take place in the amorphous state, leading to changes in various properties. [11]. When a salt of the crystalline form is used to form an SDD, it is necessary to pay special attention to the choice of counterion as it can affect properties of amorphous materials. It has been reported that the use of counterions with high electrophilicity indices produced dispersions with higher T_g, and that lower pka values also lead to higher T_g [23]. It has also been reported that the choice of counterion affected not only T_g but also fragility (temperature dependence of molecular mobility), crystallization tendency, and chemical stability of the drug [24].

1.2.3.B Polymer Choice

In ASD SDs, polymers are mixed with APIs to increase their T_g and alter their interaction with water, consequently improving their physical stability and performance. The nature and type of polymer can be a crucial factor in the determination of the physical properties of ASDs, specifically in relation to inhibition of API crystallization as well as improvement of drug release and dissolution [25]. Key polymer properties that influence the dispersion are miscibility with the drug (within the target range of compositions), functional groups that can result in drug-polymer interactions (such as acidic, basic and hydrogen bond donors or acceptors), T_g (it is preferred a polymer with higher T_g), low hygroscopicity, aqueous solubility, precipitation inhibition characteristics and solubility in volatile organic solvents used in spray-drying [11]. The most commonly used polymers in the pharmaceutical industry include the polyvinyl based synthetic polymers (such as poly (vinyl pyrrolidone) (PVP) and poly (1-vinylpyrrolidone-co-vinyl acetate)(PVPVA)), as well as the derivatives of cellulose (hydroxypropyl cellulose (HPC) and hydroxylpropyl methylcelluloseacetate succinate (HPMCAS)) [25].

Polymers reduce the molecular mobility of the drug by forming intermolecular interactions between drug and polymer and reduce the chemical potential of the drug (minimizing the crystallization driving force), resulting in a stabilization of the SDD. An amorphous drug is usually most stable when drug and polymer are mixed homogeneously at molecular level. The strong interactions between an API and a polymer via ionic interactions, hydrogen bonding, halogen bonding, van der Waals forces, and hydrophobic interactions are expected to facilitate miscibility of the drug in the polymer and may increase physical stability. The major complexity related to the miscibility of a small molecule drug in an SDD is that the amorphous drug is usually meta-stable relative to the crystalline state, inclining to crystallize when reaching an equilibrium. The miscibility in the case of these dispersions is therefore associated with a meta-stable equilibrium and requires that the drug stay below T_m and above T_g without crystallizing [26]. The miscibility behaviour of these dispersions is typically described by the Flory-Huggins theory [27,28], a lattice-based statistical mechanics model where the free energy of mixing is broken into an entropy part (that always favors mixing) and an enthalpy part (that can facilitate or prevent mixing, depending on the nature and intensity of the interaction between the components) [26]. The expression for the Gibbs energy of mixing is shown in Equation 1.3, where x_1 and x_2 are the molecular fractions of solvent and polymer (respectively), φ_1 and φ_2 are the volume fraction of solvent and polymer (respectively), χ is the Huggins interaction parameter and V_1 and V_2 are the molar volumes of the solvent and polymer (respectively) [29]. This equation is furtherly explained in section 1.2.4 of the present chapter.

$$\frac{\Delta G}{RT} = x_1 \ln \varphi_1 + x_2 \ln \varphi_2 + \chi \varphi_1 \varphi_2 \left(x_1 + x_2 \frac{V_2}{V_1} \right) \tag{1.3}$$

An advantage of spray-drying over other techniques is that the rapid drying kinetics trap the drug and excipients in a well-mixed state in the polymer matrix, so the drug does not have to be miscible in the polymer to the level at which is loaded. However, the miscibility of the drug in the polymer can be essential in the physical stability of the dispersion (specifically in respect to phase separation) [11].

Reducing the drug mobility in the polymer matrix is also directly related to the parameter T_g: high T_g limits drug mobility and, thus, phase separation. In an SDD, the amorphous API is optimally homogeneously dispersed in the polymer matrix, so the dispersion exhibits a single T_g value, between the polymer T_g and the drug T_g.The experimental T_g may be different from the theoretical prevision, and that can be due to the fact that drug-polymer interactions are different than drug-drug or polymer-polymer interactions. Therefore, a strong drug-polymer interaction will lead to a more stable SDD [11].

Another crucial role of polymers in these formulations is their improvement of the ability of the SDD to achieve and sustain supersaturation in solution due to the previously explained ”spring and parachute effect”. Most of the polymers used in ASD SDs are hydrophilic or amphiphilic. Neutral polymers can be useful when rapid dissolution of the drug in the stomach is needed and sustainment of supersaturated drug is not made difficult by a drug's strong tendency to crystallize. Enteric polymers can be used when rapid gastric dissolution is not needed or desired (for example, a weakly basic drug in the low pH gastric environment can result in high supersaturated drug-levels and subsequent neutralization in the high pH of the GI, leading to rapid crystallization; if an enteric polymer is used, the material dissolves in the small intestine and the supersaturation is more moderate). Enteric polymers are usually the more amphiphilic ones used in SDDs and, therefore, form colloids more easily. These provide a high-activity, high-surface area source of drug that can rapidly replace free drug as it is absorbed, and may help shuttling the drug across the unstirred boundary layer [11].

To sum up, the choice of the most adequate polymer for the formulation of an ASD SD depends mainly on solubility, ioniziability, lipophilicity, propension to crystallize and necessary drug loading.

1.2.3.C Additional Excipients

Additional excipients are frequently included in the formulation, with a wide range of purposes. These can include, but aren't limited to, antioxidants (to improve oxidative stability of the drug), pH modifiers (to improve stability or mitigate a pH effect), superdisintegrants (to improve disintegration and dissolution), glidants (to improve bulk material properties), complexing agents (to bind and solubilize drug), or surfactants (to improve wettability and maintain supersaturation). The addition of excipients can alter interactions and can affect the spray-drying process, which in turn influences particle properties and performance. It is, then, important to understand the interactions at molecular level and their impact in the ASD processability, performance and stability [11, 30–36]. In the present work, only formulations with no need for additional excipients were considered.

1.2.3.D Drug Loading

A target product profile (TPP) defines the desired characteristics of a target product that is aimed at a particular disease [37]. As dose and bioavailability are interrelated, target drug exposure is a critical factor in the drug loading requirements for the ASD. The TPP is typically based on the size and maximum number of dosage units to deliver the target dose; generally, a higher drug loading (drug/polymer ratio) is preferable to minimize the final product size and units. The maximum drug loading in the ASD SD depends on the physical and chemical stability, dissolution performance, and powder properties as a function of drug loading. The maximum achievable loading is often limited for drugs with high T_m and low $logP$ values that have a strong tendency to crystallize from the amorphous state, as shown previously in Figure 1.4. Polymers in which the drug is more miscible or that offer a lower mobility environment can help stabilize the drug against crystallization or phase separation. In cases where the T_g is lower in the drug than in the polymer (most cases), increasing the drug load will also increase the tendency for the drug to crystallize. High drug loadings can also result in poor dissolution properties, especially for highly lipophilic drugs (with poor wettability in aqueous media) [11].

1.2.4 Prediction of ASD Stability Through Propensity for Phase Separation: Lattice Models and the Flory-Huggins Theory

Related to the increased attention dispensed to the development of ASDs has been the increase in the application of various solution theories or computational methods to predict the propensity for phase separation of the components in these mixtures. Several thermodynamic and kinetic factors influence ASD stability, but it is generally assumed that the maximum physical stability relative to inhibition of API crystallization requires that the drug and polymer remain closely mixed [38].

In ideal solutions, the enthalpy of mixing, ΔH_{mix} is always zero so that the molar Gibbs free energy of mixing is determined entirely by the entropy of mixing, ΔS_{mix}, where R is the universal gas constant and x_i is the mole fraction of component i in the solution (equation 1.4. The change in the Gibbs free energy of mixing two components A and B at a given temperature T is, thus, given by equation 1.5. For an ideal solution, this value is always negative (favorable mixing) [38].

$$\Delta S_{\text{mix}} = -R \sum_i x_i \ln x_i \tag{1.4}$$

$$\Delta G_{\text{mix}} = \text{RT} \left(x_A \ln x_A + x_B \ln x_B \right) \tag{1.5}$$

The entropy of mixing can be derived from lattice models, in which the components are assumed to be identical in size, and each molecule is randomly placed into one site in the lattice until the lattice is completely filled. Regular solutions deviate from ideality due to the fact that the enthalpy of mixing, ΔH_{mix} is not zero. However, the entropy of mixing is assumed to be the same as for ideal solutions as the interactions between the components are weak enough so that the random mixing found in ideal solutions is preserved. The change in enthalpy of mixing two small molecules A and B derived from lattice models of regular solutions is given by equation 1.6, where χ_{AB} is the interaction parameter [38].

$$\Delta \text{H}_{\text{mix}} = \text{RTx}_\text{A}\text{x}_\text{B}\chi_{\text{AB}} \tag{1.6}$$

The overall result for the Gibbs free energy of mixing from the lattice model for regular solutions is given by equation 1.7 [39].

$$\frac{\Delta G_{\text{mix}}}{RT} = x_A \ln x_A + x_B \ln x_B + x_{AB} x_A x_B \tag{1.7}$$

The Flory-Huggins (F-H) theory (equation 1.3) represents an extension of the lattice models for regular solutions to adjust for the size disparity between molecules in solutions containing one or more polymers. The lattice site in F-H theory typically represents a polymer segment, and the probability that a given lattice site is occupied by a polymer segment takes into account the polymer chain connectivity. The change of free energy by mixing a drug and a polymer (per mole of lattice sites) in F-H theory is, therefore, expressed in terms of the volume fractions of the drug and polymer (φ_1 and φ_2), rather than mole fractions [38].

To determine the χ parameter, one of the most popular approaches is the solubility parameter method, because it only requires knowledge of the properties of the pure components [38]. The original Hildebrand solubility parameter (δ) is defined as the square root of the cohesive energy density [40] (equation 1.8, where ΔE_v is the energy of vaporization and V_m is the molar volume of the pure component in its condensed form). The interaction parameter χ would be determined through equation 1.9, where v is the volume of a lattice site and δ_{A} and δ_{B} are the pure component solubility parameters.

$$\delta = \sqrt{\frac{\Delta \mathrm{E_v}}{\mathrm{V_m}}} \tag{1.8}$$

$$\chi_{\mathrm{AB}} = \frac{v}{\mathrm{RT}} \left(\delta_{\mathrm{A}} - \delta_{\mathrm{B}}\right)^2 \tag{1.9}$$

1.2.4.A The Hansen Solubility Parameters

An important limitation of the Hildebrand solubility parameter derives from the fact that only positive values of χ are possible as a result of the squared term; therefore, it can only account for positive deviations from the ideal free energy of mixing (it is the mathematical equivalent of ”like dissolves like”). Negative deviations that derive from more favorable interactions between the drug and excipient, such as the formation of hydrogen-bonded complexes, are not accounted for. Therefore, this geometric mean assumption limits the utility of Hildebrand solubility parameters to systems in which relatively weak, non-polar interactions dominate [38].

The Hansen solubility parameters (HSP) [41,42] are an attempt to extend solubility parameter theory to include polar and hydrogen-bonding interactions. The solubility parameter is divided into three partial solubility parameters: δD, descriptive of dispersion interactions, δP, relative to polar interactions, and δH, corresponding to hydrogen bonding interactions.

The HSP can be used to predict how miscible two molecules are through the HSP distance (Ra, equation 1.10): the smaller the Ra, the more likely they are to be compatible [43].

$$\mathrm{Ra}^2 = 4\left(\delta \mathrm{D}_1 - 8\mathrm{D}_2\right)^2 + \left(\delta \mathrm{P}_1 - \delta \mathrm{P}_2\right)^2 + \left(\delta \mathrm{H}_1 - \delta \mathrm{H}_2\right)^2 \tag{1.10}$$

The 4 in front of the dispersion parameter has been controversial, but Dr. Hansen has defended its use and its validity. The Hansen solubility parameters includes the cohesive energy derived from dipolar and hydrogen bonds, as well as the dispersive bonds. Through a series of systematic steps, the HSP methodology divides this total cohesive energy into separate energies for these three major effects. It is the square root of these separate cohesive energy densities that gives the three HSP. Dipoles and hydrogen bonds are directional and involve reasonably stable bonds between two molecules or sections of molecules (that may or may not be of the same kind). However, the dispersive forces are not directional and change position rapidly, but yield a net cohesive energy when averaged over time. A logical assumption is that the directional nature of the interactions of the molecules having dipolar and hydrogen bonds may effectively prevent about one half of their “expected” interactions with surrounding molecules or pairs of molecules: the cohesive energy is based on the latent heat for evaporation of single molecules, but the majority of the single molecules are not present in the liquid, effectively preventing half of the functional groups from relatively rapid motion and potential interaction with surrounding molecules or pairs of molecules of forming a solution. If all this is assumed, then the effective cohesive energy the dipolar and hydrogen bonding parameters is reduced by a factor of $\frac{1}{2}$. Since the solubility parameter is the square root of the cohesive energy density, the factor $\frac{1}{4}$ would appear for the dipolar and hydrogen bonding terms. For practical terms, it is instead chosen to put a 4 factor in the dispersion term [43].

1.3 ASDs Screening Process

At Hovione, the development of an ASD formulation comprises four main steps. The first one is an *in-silico* screening, in which different models are used to compute API properties in order to identify promising stabilizing polymers, propensity for micellization, protonation profiles, and solvent systems; the main goal of this stage is to reduce experimentation through first principles and mechanistic models. Then, an *in vitro* screening step takes place. In this stage, the best candidate formulations from the previous step are evaluated using miniaturized high-throughput tools in order to rank prototype formulations using material sparing methods. The final prototypes are selected based on physical stability, manufacturability, and performance based on bio-relevant dissolution. The third stage is the manufacturing of the first spray drying prototypes for characterization, considering material attributes that are representative of potential clinical supplies in larger scales. Lastly, one must develop scalable tablet formulation that maintains or improves ASD performance. This screening processes allows for the development of seamless ASD formulations using a data and science-based approach that enables a reduction of lab-scale experimentation and maximizes performance [13]. The focus of this work will be on the *in-silico* screening stage.

1.3.1 In-silico screening

There are several relevant API characteristics with significant impact in ASD formulation development; protonation profile, surfactant micellization propensity, solvent system and polymer/drug miscibility are some of the most important in guiding the initial *in-silico* development strategy [13].

Protonation profiles can be generated by informatics tools, and used to assess how the ionic species of a given compound is favored in the various physiological pH values. Stomachal pH typically varies between 1.4-2.1 in the fasted state, and 3.0-7.0 immediately after a meal, depending on its composition. On its turn, small intestine pH rises between the duodenum and the ileum, varying between 5.5 and 8.3 (depending on the location and on the fed or fasted state) [44,45]. This has significant impact in the API interaction with the various GI conditions and its bioavailability [13]. For example, some class II (high permeability, low solubility) weakly basic drugs have favoured ionization and adequate solubilization in the low pH gastric environment, resulting in high supersaturation; however, when transitioning to the higher pH small intestine environment drug precipitation may occur due to its low solubility. On the contrary, weakly acidic drugs are poorly soluble in the stomach and highly soluble in the small intestine. This shows that understanding the API protonation profile is crucial, so that different measures can be used (for example, polymers with different characteristics) [11, 45].

The inclusion of surfactants in an ASD formulation in order to enhance or extend the supersaturation of the API in the GI tract may also be assessed *a priori*. Surfactants were shown to enhance wetting [46], improve dispersibility [22], inhibit crystallization in certain temperatures and conditions [47], stabilize the amorphous state [48], and enhance dissolution and supersaturation [49]. On the other hand, their presence may also lower the dissolution rates [22], promote undesired crystallization and leaching of API from the drug-rich particles into the medium [47], and influence particulate species formation [48]. Usually surfactants are used when high dose numbers are obtained; the potential of a surfactant to increase drug solubility is evaluated based on the correlation between molar solubilization capacity and micelle-water partition propensity with a drug's physicochemical characteristics [13].

The most relevant variables are usually the polymer used and the drug load. This is also the most complex part of the screening, as there can be hundreds of combinations of different polymers and drug loads. Hence, to reduce API usage, development time and ensure an optimal formulation, it is crucial to narrow down the number of prototypes for testing through computational screening tools that identify the most promising polymer and API load combinations [13].

Finally, the solvent system can also be critical in the manufacturing and performance of an ASD. The polymer, API and excipients all need to be solubilized in a solvent system with spray drying compatible characteristics. *In-silico* models can be used to narrow down the solvent systems based on API solubility parameters and a UNIFAC model.

All in all, this *in-silico* screening stage leads to a massive reduction of early stage API requirements, as only a reduced number of adequate ASD formulations and solvent systems move through to the next step.

1.3.2 *In-vitro* screening

The following step consists of experimental screening using high throughput miniaturized methods comprising two main dimensions: physical stability and dissolution performance.

Solvent casting experiments and supersaturation studies by solvent shift spiking in bio-relevant media are carried at this stage. Solvent casting consists of ”forming thermoplastic polymer samples by dipping a mould into a solution of the polymer and drawing off the solvent to leave a polymer film adhering to the mould” [50], and it's a worst-case estimation of the physical stability of the polymer since spray drying provides much faster drying kinetics, but it's an efficient method to differentiate ASD formulations based on their ability to avoid phase separation and crystallization. Differential scanning calorimetry (DSC) is used to evaluate the presence of melting peaks (due to crystallization) and the presence of a single and high T_g characteristic of a stable, homogeneous ASD. Polarized light microscopy (PLM) provides a qualitative assessment of the conditions that inhibit API crystallization [13].

As to dissolution performance, the API's solubility in the amorphous state is initially characterized in order to allow a better understanding of the mechanisms behind the drug's supersaturated state and the colloidal equilibrium that from there derives. The supersaturation performance is evaluated by a solvent shift method (where a drug is dissolved in a solvent at a high concentration and then added to an aqueous media containing a given polymer to study its ability to inhibit drug precipitation [51]) in order to select the conditions that extend the drug's window of supersaturation and improve its exposure in the GI tract. The lab-scale experiments are then ranked, and a set of lead formulation conditions are selected to move forward to the next screening stage [13].

1.3.3 SD Prototypes

The best candidates from the previous stage are then subjected to this screening phase. By modulating the SD parameters, various material attributes with major impact on processability and performance can be adjusted. This is done via modified scaled-down spray dryers that replicate the drying conditions of larger units (allowing for the production of SDD with attributes representative of those in commercial scales) combined with a comprehensive SD model and is essential for the success of the development program. Not accounting for scale-up requirements (such as using excipients and process solvents that are not suitable for large-scale processing or generating lab-scale material with properties that aren't easily reproducible at larger scales) often leads to the need of reformulation or further development work, possibly delaying time-to-market [13].

1.3.4 ASDs Tablet Formulation

When formulating tablet drugs from an ASD, it's necessary to ensure that the performance gains obtained in the SDD formulation are kept or improved, and that supersaturation is not only promoted and maximized, but also maintained through the drug's absorption window [13].

Other parameters are also taken into account in this step. Drug load usually varies between 20-80%, and needs to be maximized in order to reduce size and number of doses through the use of an excipient database coupled with mixture models for virtual formulations) [13]. Disintegration is also a requirement for the release of the API. The stabilizing polymers tend to form a gelling polymer network (GPN) that slows down disintegration. Water penetration into the tablet is reduced, and slow erosion of the gel determines API release [52]. It is crucial to understand the impact of the GPN formation in tablet disintegration to allow the definition of strategies to improve performance without compromising patient compliance, processability and stability. Due to the complexity of the problem, this is usually done through an empirical model coupled with a formulation database. Dry granulation is another important parameter, since it overcomes the low flowability problem of the low density, hollow spheres obtained through spray drying; wet granulation is avoiding due to the enhanced risk of crystallization. Finally, the use of compaction simulation presents significant advantages in ASD formulations, not only because it reduces the material consumption during early compression trials, but also because the plastic-deforming polymers used in ASDs usually present a behaviour that depends on residence time and strain rate.

1.4 Motivation and Computational Modelling Application

The current ASD screening methodology employed requires a considerable amount of time and material resources. Presently, the initial *in-silico* analysis addresses only the computation of API properties and the assessment of the API/polymer system miscibility based on thermodynamic properties. Furthermore, this miscibility assessment, from which derives a preliminary prediction of the maximum API loading (process described in subsection 3.3.2), is based on the Flory-Huggins theory – this methodology is limited, since it was originally developed for a mixture of two polymers, and not a mixture of one polymer and an API, which leads to considerable deviations from the reality. All supersaturation studies (for various API/polymer combinations and drug loads) are carried out *in-vitro*, usually leading to a few promising conditions amongst many failed combinations. The existence and maintenance of supersaturation is a very important condition to be analysed in the early stages of ASD screening and development, given that the attainment and sustainment of supersaturation is crucial in the process of GI tract absorption of poorly soluble APIs, often with a restrict absorption window. It would naturally be of interest to perform this study in a less material-consuming and more prompt manner to accelerate the whole screening process, predicting *a priori* ASD stability and performance. Thus, it is predicted that the development of a workflow that begins with an *in-silico* analysis of the suitability of a given API to be formulated into an ASD with each polymer available (and prediction of which combinations will likely lead to success not only in terms of miscibility but also relating to supersaturation assays), returning a list of best candidate mixtures, would be valuable.

Statistical models, such as Principal Component Analysis (PCA) and Partial Least Squares (PLS), and machine learning approaches can be highly useful in the resolution of this challenge. Machine learning algorithms improve automatically through experience, by building a mathematical model based on a set of training data and subsequently evaluating how good the model is through a cost function in a sequence of iterations. There are three approaches to machine learning: supervised, unsupervised and reinforcement learning - the current project will use solely supervised learning approaches.

In supervised learning, the inputs include the training examples coupled with the respective desired outputs; the algorithm develops a learning function that maps an input to an output based on the input/output correspondence in training examples. Ideally, the function should be able to predict the correct output for a novel input data example. This requires the learning algorithm to generalize from the training data to new situations without a high level of bias or variance. If the function shows a high variance, it means that the model is overfitting the data - it is extremely well adjusted to the training examples provided, predicting very accurately the outputs for this specific data set, but performs poorly in new examples that weren't used in the learning process. If the opposite happens, the function is biased - this means that the model is underfitting the data, failing to predict the correct output solution both for training examples and for new examples. A learning algorithm should present an equilibrium in flexibility so that it is able to present low bias, but also so that the variance is not so high that it fits each data set differently. Supervised learning has been previously used in the pharmaceutical industry for the prediction of the behaviour of diverse formulations; for example, artificial neural networks (ANN) have been used with success to predict the disintegrating time of oral disintegrating tablets [53], while there are examples of random forest (RF) algorithms being very successful at predicting the stability of ASDs [54].

The goal of this project was to design an alternative workflow for the initial part of the ASD screening process that would save time and material resources. This workflow should allow scientists, when receiving a new API to formulate, to assess ASD stability and performance *a priori* through statistical and machine learning models that would predict two distinct outputs: the maximum API load and the existence (or absence thereof) of a spring and parachute behaviour. In the next chapters I will present the work done during the development of this project, and the arising results.

2

Methodology

Contents

2.1 Database construction

The first step in the development of the intended models was to build a database that comprised all the data that would be needed for the project; naturally, this included the inputs and the target outputs. The variables were divided into ”API descriptors”, ”Polymer descriptors” and ”ASD variables / interaction parameters”, described in Table 2.1.

Table 2.1: Summarization of the different variables (including inputs and target outputs) harvested for the different data observations, including variables related to the polymer, variables related to the API and variables inherent to the ASD or related to the interaction between API and polymer.

Polymer	Molecular Weight polymer (g/mol)
	Molecular Weight monomer (g/mol)
	Number of monomers
	LogP polymer
	Tg polymer (ºC)
	Tm polymer(ºC)
	Tm/Tg polymer
	Number of H-bond acceptors polymer
	Number of H-bond donors polymer
	Number of rotatable bonds polymer
	Polar Surface Area polymer (Å^2)
	Hansen's Solubility Parameters polymer(δD, δP, δH)
	pKa polymer
API	Molecular Weight API (g/mol)
	LogP API
	Tg API (ºC)
	Tm API (ºC)
	Tm/Tg API
	Number of H-bond acceptors API
	Number of H-bond donors API
	Number of rotatable bonds API
	Polar Surface Area API (Å^2)
	Aqueous Solubility API $(\mu g/L)$
	Hansen's Solubility Parameters API (δD, δP, δH)
	pKa API
ASD / interaction	API loading (%)
	Spring and Parachute (categorial variable)
	Ra^2 (HSP distance between two molecules) (Å^2)

In total, 136 observations referring to ASD SD formulations were harvested, comprised by combinations of 37 different APIs and 25 different polymers. To be noted that each individual model wasn't trained on this number of observations: the dataset includes two outputs, ”API loading” and ”Spring and parachute effect”; therefore, some API/polymer combinations can exist more than once with different API loadings, as that can have an impact in the existence of a spring and parachute behaviour – these observations will all exist in the ”Spring and parachute effect” models, but only the one with the highest

API loading will exist in the ”maximum API loading” models. On the other hand, for some of the harvested observations the author didn’t provide information relative to the spring and parachute behaviour of the formulation upon dissolution – those observations only exist in the ”maximum API loading” models. Furthermore, observations with missing values were not used in the MATLAB models. The number of observations used in the training of each individual model is stated in the respective subsection describing the development of that specific model. To be noted that, in the models developed in MATLAB, the variables ”LogP polymer” and ”pKa polymer” were excluded to minimize the number of observations excluded, since they had many missing values.

The observations were harvested from two primary sources: company’s internal reports derived from research and development campaigns, and external scientific papers from diverse sources ([15,26,55–75]). All the observations used are referent to *in vitro* dissolution studies of ASD formulations – it was chosen not to include *in-vivo* studies due to the high variability to those attached. The features mentioned in Table 2.1 were, when possible, obtained from the original source of the data observation (internal report or external paper); however, many times it was necessary to obtain API or polymer features from external sources. The software ChemDraw by PerkinElmer was used to estimate the variables ”number of H-bond acceptors”, ”number of H-bond donors”, ”number of rotatable bonds” and ”polar surface area” both for polymers and APIs, and also for some values of pKa and LogP; the software ”Hansen Solubility Parameters in Practice” (HSPiP) was used to estimate the Hansen solubility parameters (δD, δP and δH) both for polymers and APIs; the remaining values, when unavailable at the original source, were harvested from the following databases: PharmaCircle, PubChem, DrugBank, Hazardous Substances Data Bank (HSDB), ChemSpider and Chem-Space. An additional set of observations was harvested from external literature papers to serve as an external validation set ([76–85]).

2.2 Statistical Models Overview

2.2.1 Partial Least Squares

It is often very useful to use controllable or easily measurable variables (factors) to explain, regulate or predict the behaviour of other variables (responses). The least squares model is a method of estimating the unknown parameters in a linear regression model by minimizing the sum of the squares of the differences between the observed dependent variable in the dataset and the values predicted by the function [86]. These models yield poor results when data has a small sample size, missing values and multicollinearity between predictors, challenges that the partial least squares (PLS) model was designed to overcome [87,88]. This method combines regression and dimension reduction techniques, as well as modeling tools. Partial least squares methods assume that the observed data is generated by a process driven by a small number of latent variables (not directly measured variables) – this is called indirect modeling (Figure 2.1) [88]. A PLS model measures covariation between two or more blocks of variables, and creates a new set of variables that is optimized for maximum covariance (and not correlation) using the fewest dimensions [89].

The concept of PLS is to try to extract these latent factors, accounting for as much variation as possible while modeling the responses well. A PLS model creates orthogonal score vectors (also called components) by maximising the covariance between different sets of variables. The predictor and predicted variables are each considered as a block of variables. The PLS model extracts the score vectors that will then serve as a new predictor representation, and regresses the response variables on these new predictors [90].

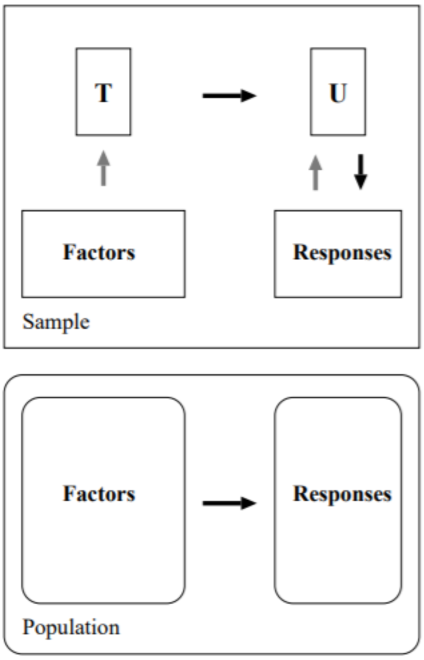

Figure 2.1: Schematic representation of an indirect modeling process. The goal is to use the factors to predict the responses; this is achieved indirectly, by extracting latent variables T (X-scores) and U (Y-scores) from sampled factors and responses, respectively. The X-scores are used to predict the Y-scores, that in turn are used to construct predictions for the responses [86].

2.2.1.A How does PLS work?

Lets take into consideration the following notation:

- $\mathcal{X} \subset \mathcal{R}^N$: N-dimensional space of variables representing the first block;
- $\mathcal{Y} \subset \mathcal{R}^M$: M-dimensional space of variables representing the second block;

The PLS models the relation between these two blocks through score vectors: after observing n data samples from each block of variables, the model decomposes the (n × N) matrix of zero-mean variables

X and the (n $\times$ M) matrix of zero-mean variables Y through the following formulas:

$$\begin{array}{l}\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \\ \mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F}\end{array} \tag{2.1}$$

Where:

- $\mathbf{T}, \mathbf{U}$ are $n \times p$ matrices of the p extracted score vectors (components);
- the ($N \times p$) matrix $\mathbf{P}$ and the ($M \times p$) matrix $\mathbf{Q}$ represent loadings matrices;
- the ($n \times N$) matrix $\mathbf{E}$ and the ($n \times M$) matrix $\mathbf{F}$ are the residuals matrices.

The PLS will find weight vectors $\mathbf{w}$, $\mathbf{c}$ so that:

$$[\mathrm{cov}(\mathbf{t}, \mathbf{u})]^2 = [\mathrm{cov}(\mathbf{Xw}, \mathbf{Yc})]^2 = \max_{|\mathbf{r}|=|\mathbf{s}|=1} [\mathrm{cov}(\mathbf{Xr}, \mathbf{Ys})]^2 \tag{2.2}$$

Where $\mathrm{cov}(\mathbf{t}, \mathbf{u}) = \mathbf{t}^T\mathbf{u}/n$ represents the sample covariance between the score vectors $\mathbf{t}$ and $\mathbf{u}$. The score vectors $\mathbf{t}$ and $\mathbf{u}$ are then given as

$$\mathbf{t} = \mathbf{Xw} \quad \text{and } \mathbf{u} = \mathbf{Yc} \tag{2.3}$$

where weights $\mathbf{c}$ and $\mathbf{w}$ were computed through an iterative algorithm [90].

To sum up, each dimension expresses a linear relation between an X-score vector $\mathbf{t}$ and Y-score vector $\mathbf{u}$. The weight vectors of each model dimension express how the X-variables are combined to form $\mathbf{t}$, and the Y-variables are combined to form $\mathbf{u}$. In this way the data are modeled as a set of ”factors” in X and Y and their relationships. The weights for the X-variables,$\mathbf{w}$, represent the importance of said variables and how much they contribute to the modeling of Y. The weights for the Y-variables, $\mathbf{c}$, represent which Y-variables are modeled in the respective model dimensions. When these coefficients are ploted in a $\mathbf{w} \times \mathbf{c}$ plot, the result is a visual representation of the relationships between X and Y, the important X-variables, which Y-variables are related to each X-variables, and so on [91].

2.2.1.B Advantages and limitations of PLS models

The partial least squares methodology offers several advantages in comparison with methods such as multiple regression. To begin with, PLS is able to produce robust equations even when the number of X-variables (or independent variables) is greater than the number of data points, and PLS-based predictions tend to be more accurate than those from multiple regression models. In addition, PLS models are much more stable when the independent variables are correlated rather than orthogonal, and PLS models can simultaneously derive models for more than one dependent variables [92].

The differences between PLS and multiple regression result from their different strategies to identify a linear relationship: multiple regression treats independent variables as independent entities, offsetting each variable separately to obtain the best overall relationship with the dependent variable. On the other hand, PLS considers all the independent variables together as a block, and in the iterative process, the model repeatedly transforms both blocks (dependent and independent variables) so that their commonality is maximal [92].

Both for multiple regression and for PLS models, there are two opposite kinds of errors: omitting structural factors which in fact are related to response (type I errors) and reporting structural factors which in fact are not related to response (type II errors). These last types of errors are called 'chance correlation' and are very common in multiple regression models: if enough combinations of independent factors are independently compared to a few responses, sooner or later the numbers will agree by chance. On the other hand, PLS models are more prone to type I errors (overlooking 'true' correlations): PLS models can fail to discover good correlations involving only a small fraction of the independent variables under consideration, while a multiple regression algorithm will always find small numbers of highly correlated columns [92].

From this, it comes naturally that the signal to noise ratio in the data set (meaning the ratio between (i) the variance in all the independent variables together which correlates with variance in the dependent variables, and (ii) the variance in all the independent variables together which does not correlate with variance in the dependent variables) is an essential factor to determine the success of the PLS model. Given the tendency of PLS models to type I errors, a PLS will only be successful if the signal to noise ratio is high enough. This leads to the conclusion that the PLS results are highly susceptible to scaling: if one, for example, multiplies all the values in a column by 10^7, the magnitude of the resulting values (whether signal or noise) will overwhelm any possible influence from any other column. It is, then, essential to scale the data (for example, into a [0,1] interval) before applying a PLS model [92].

2.2.1.C Evaluating the PLS model performance

The most widely used method to evaluate the performance of a regression method is the R^2, defined by equation 2.4,

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}} \tag{2.4}$$

where *sum squared regression* is the sum of the residuals squared, and the *total sum of squares* is the sum of the distance the data is away from the mean, all squared. The value of R^2 is a value between 0 and 1 (or 0% and 100%), and it represents the goodnes of the fit to the model, or the percentage of explained variance by the model. However, this performance measure does not disclose information about the causation relationship between the independent and dependent variables. It's not always true that the higher the R^2, the better; sometimes, a model with a high R^2 may not be generalizable and have poor predictive capability for a new set of data.

Another very important factor to take into consideration is the Q^2 parameter, that represents the R^2 applied to the cross validation data (the term cross validation is explained furtherly). Therefore, while the R^2 parameter represents the descriptive capability of the model and approaches 100% as the model complexity increases, the Q^2 parameter represents the predictive capability of the model, and at a certain degree of complexity will not improve any further and then degrade.

2.3 Machine Learning Models Overview

Machine Learning is a data analysis method based on the idea that a computational system can learn from data, recognize patterns and facilitate decision making with a certain confidence interval. Machine learning algorithms are fed a certain input dataset (training data) and analyze it; in supervised learning (the main focus of this project), a ”correct answer” is also fed to the algorithm. The program will then try to find patterns that correlate the input with the correct output, with the objective of gaining the ability to predict the correct output for new input data. During the training, when corrections are identified, the algorithm learns from that information to improve its future decision making.

2.3.1 Artificial Neural Network

Artificial neural networks (ANN) are brain-inspired systems, and have a high level of popularity in the machine learning environment. In the present project, only supervised learning neural networks will be covered and used.

These models consist of at least an input and an output layer of neurons (or nodes), and usually one or more hidden layers. The connections between the nodes are called weights, and each node has associated a ”bias” term: the weights represent the strength of a particular node, and the bias term shifts the activation function (furtherly explained) up or down. In Figure 2.2 [93] is represented a single layer neural network, where x_0, x_1, ..., x_n represent various inputs (independent variables), which are multiplied by the weights w_0, w_1, ..., w_n.To move forward through the network, each neuron in the next layer is iteratively calculated (in this simplified case, the only further layer is the output layer): these products are summed and fed to an activation function ϕ (equation 2.5) which generates an output, the activation a [93, 94].

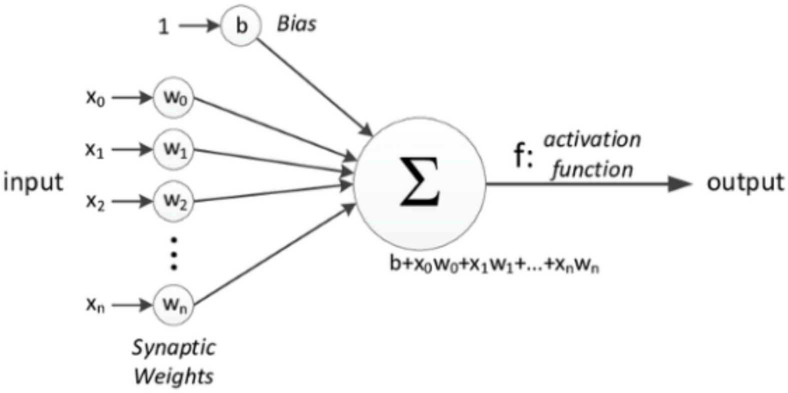

Figure 2.2: Representation of a neural network with a single layer, and the respective parameters [91].

$$a = \phi(\sum_{i=1}^{n} x_i w_i + b) \tag{2.5}$$

To sum up, each neuron has an activation a and each neuron that is connected to a new neuron has a weight w. Taking now into account a less simplified example of a neural network with one or more hidden layers, we denote each activation as $a_{neuron}^{l(layer)}$ and each weight as $w_{j,k}^{l}$, where k is the origin neuron, j the destiny neuron and l the origin layer. To calculate each activation in the next layer, we need all the activations from the previous layer, and all the weights connected to each neuron in the next layer; combining these two and adding a bias function, we wrap the whole equation in the sigmoid function and get the final result (equation 2.6) [94]:

$$a^{(1)} = \phi\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,k} \\ w_{1,0} & w_{1,1} & \dots & w_{1,k} \\ \dots & \dots & \dots & \dots \\ w_{j,0} & w_{j,1} & \dots & w_{j,k} \end{bmatrix} \begin{bmatrix} a_0^0 \\ a_1^0 \\ \dots \\ a_n^0 \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \dots \\ b_n \end{bmatrix}\right) \iff a^{(1)} = \phi\left(Wa^0 + b\right) \tag{2.6}$$

2.3.1.A Activation Function

The activation function serves the purpose of inserting non-linearity into the model: by calculating the weighted sum and further adding bias to it, it converts an input signal of a node to an output signal, which is used as input to the next layer. If no activation function is applied, the output would be merely linear, and while linear functions are easy to solve, they have very limited modeling power. There are several types of activation functions, but the present work used only two of them: the sigmoid activation function, and the hyperbolic tangent function.

The sigmoid function (Figure 2.3) ranges between 0 and 1, making it appropriate for models where the output is a probability. This function has the drawback of getting stuck in training if strong negative inputs are fed to the model. The hyperbolic tangent function (Figure 2.4) is similar to the sigmoid function, but performs better. It ranges between -1 and 1, so strong negative inputs will be mapped to negative output, and only near-zero inputs will be mapped to zero [93].

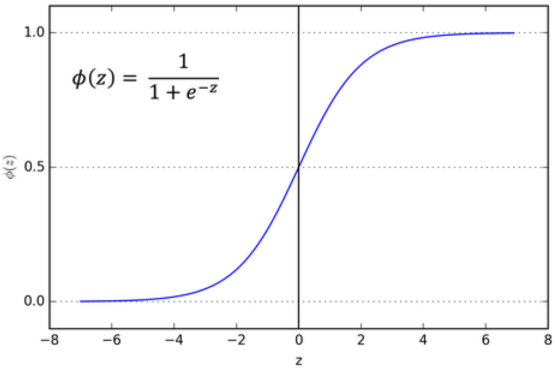

Figure 2.3: Representation of the sigmoid activation function.

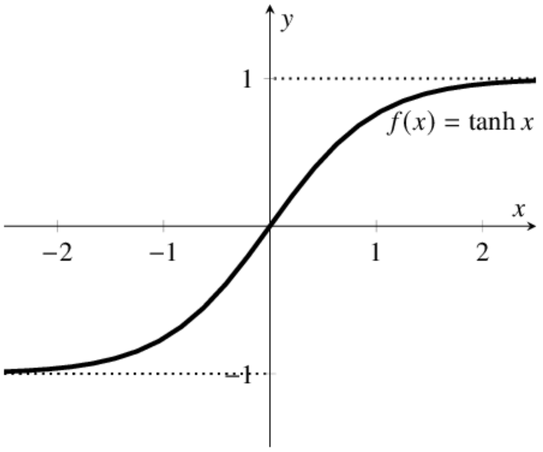

Figure 2.4: Representation of a the hyperbolic tangent function.

2.3.1.B How does a neural network learn?

Neural networks learn in a similar way to humans: humans perform an action that is either accepted or corrected; similarly, neural networks predict a value and compare it to the value provided by the training data, and based on the actual value and the predicted value, an error value (the cost function) is computed and sent back through the system. There are several different cost functions, but one of the simplest and most common ones is the sum of the squared differences, where y is the target output and a is, once again, the output of the neuron (equation 2.7).

$$C = \frac{1}{n}\sum_{i=1}^{n}(y_i - a_i)^2 \tag{2.7}$$

The neuron learns by changing the weight and bias at a rate determined by the partial derivatives of the cost function. However, due to the inherent shape of the sigmoid function, when the neuron's output is close to one, the derivatives become very small - this is called learning slowdown. To correct this, other cost functions can be used: one of the most common ones in classification problems is the cross entropy cost function (equation 2.8). This function is always positive and tends to zero as the predicted value approximates from the target value [95].

$$C = -\frac{1}{n}\sum_x [y \ln a + (1-y)\ln(1-a)] \tag{2.8}$$

For each layer, the cost function is used to adjust the weights for the next input with the aim to minimize said cost function. The error keeps becoming smaller in each run as the network learns how to analyze values. The resulting data is fed back through the entire neural network. This process, called back propagation, is repeated until the cost function has reached a minimum.

For this, the partial derivatives of the cost function from the weights and bias are calculated and saved in a gradient vector ∇, that has as many dimensions as the number of weights and biases (equation 2.9):

$$-\nabla C\left(w_1, b_1, \ldots, w_n, b_n\right) = \begin{bmatrix} \frac{\partial C}{\partial w_n} \\ \frac{\partial C}{\partial b_1} \\ \vdots \\ \frac{\partial C}{\partial w_n} \\ \frac{\partial C}{\partial b_n} \end{bmatrix} \tag{2.9}$$

The gradient is then calculated and updated according to the chosen algorithm (for example, gradient descend, mini-batch gradient descend, stochastic gradient descend or scaled conjugate gradient). As we move more layers back through the network, there would be more partial derivatives to compute each weight, bias and activation.

To sum up, the learning process of a neural network starts with randomly initializing the weights and feeding the first observation of the training data to the input layer. Then, the algorithm performs forward propagation, where from left to right the neurons are activated with an impact limited by its weights, and the activations are propagated until the predicted output is obtained. The generated cost function is analysed, and a step of backpropagation follows, when from right to left, the error is backpropagated and the weights are updated according to how much they are responsible for the error (a parameter α called learning rate decides how much the weights are updated). These steps are repeated and the weights are updated after each observation; an epoch is completed after the whole training set has passed through the neural network, and then more epochs are performed [93].

2.3.2 Random Forest

The random forest (RF) classifier, that consists of an assemble of decision trees, is one of the most used supervised learning models to approach classification problems. A decision tree is a classifier that, at each node, asks what feature of the data allows the division of the observations in a way that the resulting groups are as different as possible, and that the members of each group are as similar as possible. In a random forest, each decision tree makes an individual class prediction; in the end, the class that was predicted more often is the class predicted by the random forest (Figure 2.5 [96]). The idea behind these models is that a large number of uncorrelated deciders will outperform any individual one. The key for the good performance of these models is the low correlation between the individual trees; that way, the trees protect each other from their individual errors, as long as they don't constantly err in the same direction [97].

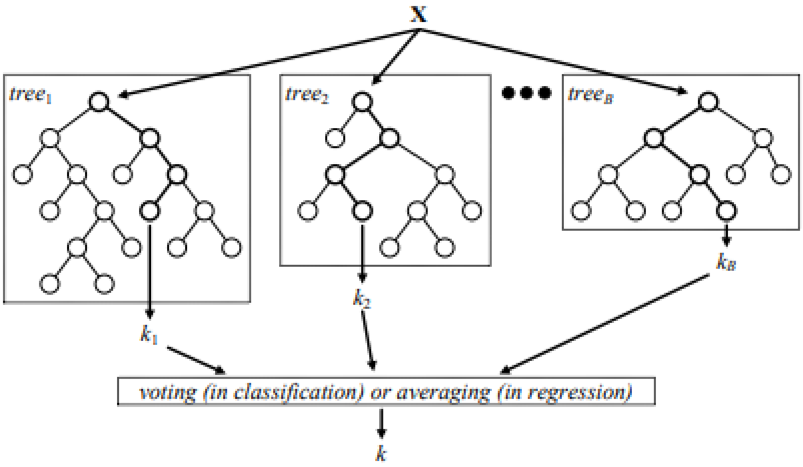

Figure 2.5: Schematic representation of a random forest construction [94].

An important feature that makes random forests a very useful algorithm is its ability to rank the predictors according to its internal measure of variable importance [98], unlike, for example, the neural network model, that predicts an output but its internal process is a black box. Random forests are also very easy to develop, since they require no feature scaling and no to very little hyperparameter tuning [99].

2.3.2.A How does a random forest ensure the uncorrelation between individual trees?

The model uses two methods to ensure the behaviour of the independent trees is uncorrelated enough: bootstrap aggregation and feature randomness.

The concept of bootstrap aggregation, or tree bagging, is based on the fact that decision trees are very sensitive to the data they are trained on and, therefore, small changes to the training set can result in significantly different tree structures. The model takes advantage of this by training individual trees on randomly sampled subsets of the training data with replacement. The training set isn't being divided into smaller chunks to be fed to different trees; instead, the whole dataset of size N is fed to every tree, and each tree takes a random sample of size N with replacement. Individual trees are very prone to overfitting and are sensitive to noise in the data, so bagging trees while making sure they are not correlated will make them more robust without increasing the bias [97, 100].

Another key aspect of random forests is feature randomness: in a single decision tree, when it's time to split a node, every possible feature is considered, and the algorithm chooses the one that produces the most separation between the observations in the two resulting nodes. On the other hand, in a random forest each tree only has available a random subset of features to choose from, forcing even more variation between the several trees in the model, and resulting in lower correlation and more diversification.

2.4 Model Development

2.4.1 When complexity strikes back: the overfitting challenge

When training a machine learning model, there are two phenomenons one has to look out for: bias and variance. Bias (or underfitting) is an algorithm's tendency to pick a model that is not structurally correct for the data, by making incorrect assumptions about the dataset. On the other hand, variance (or overfitting) arises from sensitivity to small fluctuations in the training set, because the model learned every quantitative detail of the training data, inevitably including random noise and missing the broader regularities in the data. The critical variable modulating these two phenomenons is complexity: more complex models will fit the training data more closely, but may be less generalizable to new data (overfitting). A model with high variance may have a very good performance in the dataset it was trained on, leading one to believe the model is very accurate, but perform poorly when it is tested in an independent test set [101–103].

Avoiding overfitting is key when developing a model for various reasons. First of all, models that include irrelevant predictors require the user to waste more resources harvesting the needed data, eventually making the effort needed surpass the benefit of the model. Furthermore, these models will have a poorer performance in reality, since they will not be used to make predictions in relation to the training set [103].

2.4.1.A How to prevent overfitting?

Validation

Validation is very useful in detecting and preventing overfitting. It consists on testing the model on data the model hasn't yet been exposed to, which allows the user to analyze the performances on both sets and decide if there is probably a problem of overfitting. Two of the most common validation techniques are hold-out validation and cross validation [104].

Hold-out validation consists of splitting the dataset into a ”training set” and into a ”validation set”; the training set is what the model is trained on, and the validation set is used to see how well that model performs on unseen data. Usually the splitting is around 70% or 80% for training, and 30% or 20% for validation. Cross-validation, or k-fold cross validation, consists on splitting the dataset randomly into ’k’ sub-groups, or folds; one of the groups works as the validation set (hold-out fold), while the rest k-1 folds work as a training set. The model is trained on the training set and scored on the validation set, and the process is repeated until each unique group as been used for validation [104].

Adding more data

Adding more data is a simple tool to prevent overfitting - naturally, if the training set is too small, even a simple model will adjust almost perfectly to it. However, this is more a pre-requisite to train a model rather than a solution for overfitting; the dataset should be large enough, clean and relevant [105].

Removing features

As it has been mentioned before, having irrelevant features is not only expensive computationally and in the sense that it's necessary to harvest more data, but it also causes overfitting by introducing unnecessary noise and complexity into the model [103]. Some models have built-in tools to help identify irrelevant features, while in other models the developer might have to select the features by hand [105].

Early Stopping

When an algorithm is being trained iteratively, it is possible to measure the performance of the algorithm at each iteration. Until a certain number of iterations, the performance goes up (the error diminishes); reached a certain number, the error in the validation set starts increasing, as the model is starting to overfit the data. Early stopping consists on stopping the training when the model reaches that point (Figure 2.6 [105]).

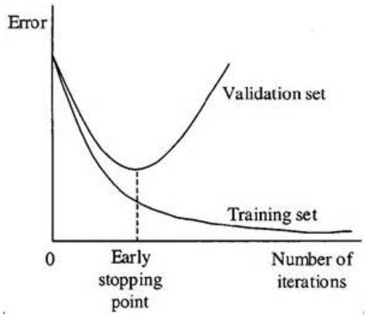

Figure 2.6: Representation of the early stopping method in an iterative training process [103].

2.4.2 Partial Least Squares (PLS)

The PLS models were developed in the software SIMCA by Umetrics ®. Two separate PLS models were developed: one that takes into account API, polymer and interaction variables, and one that takes makes the predictions solely based on the API features. Both PLS models have as target output the prediction of the maximum API loading. For the PLS models, the dataset was scaled through mean normalization (equation 2.10) so that all the values would fall into a [-1, 1] interval.

$$x' = \frac{x - \text{ average } (x)}{\max(x) - \min(x)} \tag{2.10}$$

2.4.2.A PLS model - API + Polymer + Interaction Variables

The first PLS model developed is meant to be used to predict the maximum API load for a given API/polymer combination, and takes into account variables related do the API individually, to the polymer individually, and to the ASD (or interaction between API and polymer). Prior to model optimization, the model had as inputs 84 observations (comprised by combinations of 37 different APIs and 23 different polymers) and 30 features (or variables) descriptive of the observations (the graphical representation of the data distribution across API loads can be found on Figure 2.7).

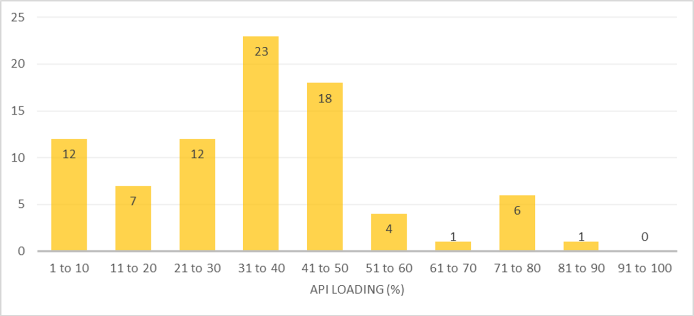

Figure 2.7: Data distribution across API loadings for the PLS model taking into account API and polymer features.

The model was then optimized: variables which mainly produced noise were excluded according to the VIP (Variable Importance in Projection) value (equation 2.11, where b is the regression coefficient, w_j is the weight vector, w_{kj} is the kth element of the weight vector j and t_j is the score vector), and outliers were excluded based on the prediction plot (predicted output by the model versus actual output) and the normal probability plot of residuals, referred to as n-plot.

$$VIP_K = \sqrt{n \frac{\sum_{j=1}^{a} b_j^2 t_j^T t_j \left(\frac{w_{kj}}{\|w_j\|}\right)^2}{\sum_{j=1}^{a} b_j^2 t_j^T t_j}} \tag{2.11}$$

One of the assumptions for regression analysis is that the residuals (error terms, or the differences between the observed value of the dependent variable and the predicted value) are normally distributed, and this plot is a method of learning whether this is a valid assumption, and therefore to identify possible outliers. If the data follows a normal distribution with mean μ and variance σ^2, then a plot of the theoretical percentiles of the normal distribution versus the observed sample percentiles should be approximately linear. The theoretical p-th percentile of any normal distribution is the value such that p% of the measurements fall below the value. For example, the median (the 50th-percentile) is the value so that 50%, of the measurements fall below the value. The normal probability plot of the residuals is a scatter plot with the theoretical percentiles of the normal distribution on the x-axis and the sample percentiles of the residuals on the y-axis [106]. In Figure 2.8, the four types of possible n-plots are represented, as well as the corresponding interpretations [107].

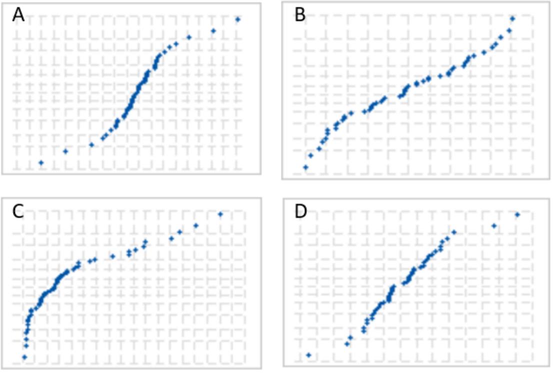

Figure 2.8: The four types of possible n-plots. (A): S-curve implies a distribution with long tails; (B): Inverted S-curve implies a distribution with short tails; (C): A curve implies a skewed distribution (downward curve – right-skewed and upward curve – left-skewed); (D): A few points lying away from the line implies a distribution with outliers [105].

The validation was performed through cross-validation; to diminish overfitting and enhance the model's performance on new unseen data, each cross validation group consisted of the observations of a single API, so that when the Q^2 valued was calculated, for each cross-validation step the model had never been exposed to that API before. Therefore, the value of k folds was the number of APIs in the model, so this valued changed during model optimization due to the removal of outliers.

2.4.2.B PLS model - API variables only

The second PLS model developed is meant to predict the maximum API loading in a given ASD for a given API, without taking into account the polymer (its inputs are, therefore, the API characteristics); this would allow scientists to have an *a priori* idea of the maximal loading for a given drug before expending the resources to consider the many possible polymer options, therefore having the possibility of immediately excluding the ASD formulation for that substance, or beginning the experimental studies closer to the real value of API loading, wasting less time and money.

Prior to model optimization, the model had as inputs 37 observations (and, therefore, the same number of APIs) and 12 features (or variables) descriptive of the said APIs (the graphical representation of the data distribution across API loads can be found on Figure 2.9).

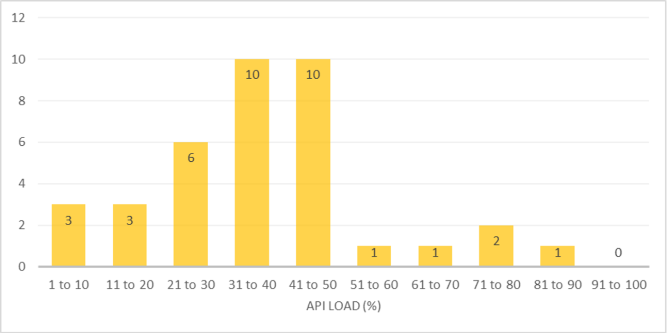

Figure 2.9: Data distribution across API loadings for the PLS model taking into account API features only.

The model was then optimized in the same manner as before: variables which mainly produced noise were excluded according to the VIP value, and outliers were excluded based on the prediction plot and the normal probability plot of residuals.

Cross validation was performed; since in this model there are no two observations with the same API, the number of folds was simply defined as $k = 7$.

2.4.3 Random Forest (RF)

The RF classifier model was developed in the software MATLAB by MathWorks®. Two models were developed: the first one had the objective of predicting the output ”maximum API loading”, and the second one had the objective of predicting the output ”spring and parachute” effect. The random forest models do not require data scaling [108].

2.4.3.A Random Forest classifier model - maximum API loading

The first random forest model developed (bagged trees ensemble, through the function ’fitcensemble’) is meant to be used to predict the maximum API load for a given API/polymer combination, and takes into account variables related do the API individually, to the polymer individually, and to the ASD (or interaction between API and polymer). Initially, a random forest regression was tried; however, it performed very poorly on the data, so the random forest classifier was tested instead. The model had as inputs 66 observations (comprised by combinations of 29 different APIs and 15 different polymers) and 28 features (or variables) descriptive of the observations (the graphical representation of the data distribution across API loads can be found on Figure 2.10). The validation was performed through cross-validation using 20 folds. The maximum number of splits was defined by default as 65. The number of trees in the model was defined by default as 30. This value was kept, since adding more trees provided no additional benefit for the performance of the model. The API loadings were divided into labels, or classes, presented in Table 2.2.

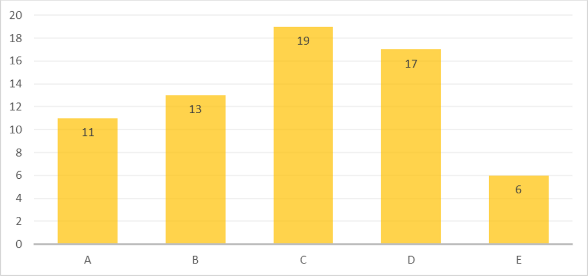

Figure 2.10: Data distribution across API loadings for the RF model taking into account API and polymer features. The correspondence between classes and API loadings is represented in Table 2.2.

Table 2.2: Correspondence between drug load ranges and the labels used in the random forest models developed for the prediction of the maximum drug loading.

Label	Drug Load Range
A	1-15 %
B	16-30 %
C	31-45 %
D	46-70 %
E	>70%

A second model, taking into account only the API features, was attempted. This model took into account 29 observations (or APIs), and 12 descriptors or variables (the graphical representation of the data distribution across API loads can be found on Figure 2.11). The maximum number of splits was defined as 20, and the number of trees as 30. The validation was performed through k-fold cross validation, with k=5.

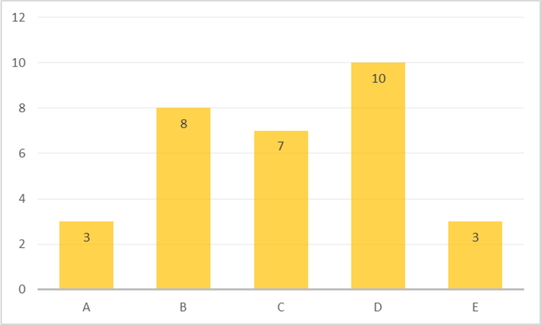

Figure 2.11: Data distribution across API loadings for the RF model taking into account API features only.

2.4.3.B Random Forest classifier model - Spring and Parachute effect

The second random forest model developed (bagged trees ensemble, through the function 'fitcensemble') is meant to be used to predict the existence or lack thereof of a spring and parachute behaviour, based on the variables descriptive of the API, of the polymer and of the ASD / interaction variables. This model should predict a class of 1 if the ASD is predicted to have good spring and parachute effect, and 0 otherwise. The model had as inputs 92 observations (comprised by combinations of 23 different APIs and 19 different polymers) and 29 features (or variables) descriptive of the observations (the graphical representation of the data distribution for both classes can be found on Figure 2.12). The validation was performed through cross-validation using 20 folds. The maximum number of splits was defined by default as 91, and the number of trees in the model was defined by default as 30.

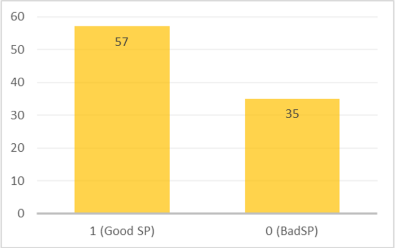

Figure 2.12: Data distribution for the spring and parachute effect output for the models developed for this variable, taking into account API and polymer features.

A second model, taking into account solely the features descriptive of the API, was attempted. This model took into account 23 observations (or APIs), and 12 descriptors or variables (the graphical representation of the data distribution for both classes can be found on Figure 2.13).

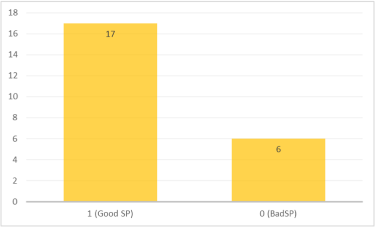

Figure 2.13: Data distribution for the spring and parachute effect output for the RF model developed for this variable, taking into account API features only..

2.4.4 Artificial Neural Network (ANN)

The ANN model was developed in the software MATLAB by MathWorks®. For this model, the data was not normalized manually; instead, the function 'mapminmax' was applied. This model had the objective of predicting if the ASD would yield spring and parachute behaviour. The model had as inputs 92 observations (comprised by combinations of 23 different APIs and 19 different polymers) and 29 features (or variables) descriptive of the observations (the graphical representation of the data distribution for both classes can be found on Figure 2.12) . In the ANN model, the dataset is divided in three: the training set, where the model is trained and the weights are updated; the validation set (hold-back validation), used to prevent overfitting through early-stopping (training continues as long as the network's performance continues improving on the validation set), and the test set, that works as an independent test set without any role in the model training itself, but with the purpose of analyzing the performance of the model in new, unseen data. The division was made in the following manner: 65% to training (60 samples), 15% to validation (14 samples) and 20% to testing (18 samples).

The neural network architecture was defined as follows: one input layer comprised of 29 nodes (the number of features the model is trained on); one output layer comprised of 1 node, that outputs either a value of 1 or a value of 0; and one hidden layer, with a number of nodes with approximately a medium value between the input and output layers (15 nodes). The backpropagation algorithm used was scaled conjugate gradient (SCG). For the hidden layer, the hyperbolic tangent sigmoid transfer function (returns values between -1 and 1) was used; for the output layer, the sigmoid transfer function was used (maps values between 0 and 1). The cost function used was the cross-entropy cost function. The ANN is represented in Figure 2.14.

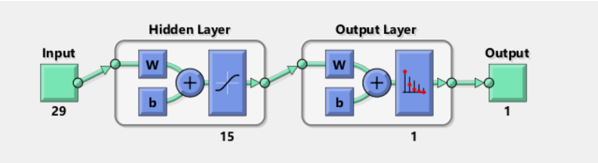

Figure 2.14: Representation of the architecture of the ANN developed for the prediction of the spring and parachute output.

3

Results and Discussion

Contents

3.1 Development of models for the prediction of the maximum drug loading

For the prediction of this output, three successful models were developed: two PLS models and one RF classifier model.

3.1.1 Partial Least Squares models

3.1.1.A PLS model - API + Polymer + Interaction Variables

The first step to assess the performance of the PLS regression is to obtain the R^2 and Q^2 values, as explained before. A higher R^2 implies a better fit to the training data, and a higher Q^2 corresponds to a better fit in the cross-validation sets. It is mentioned in the software user guide [91] that, for a model to have good predictive power, the Q^2 value should be at least equal to 0.5; however, this is merely a guideline and not a bounding rule, as the performance is also dependent on the intended use of the model (that may allow more or less margin of error in the predictions to be considered a good model). For the first version of the model, prior to any optimization, the R^2 and Q^2 obtained were, respectively, 40% and 20%, and only one principal component was obtained for optimal performance (Figure 3.1).

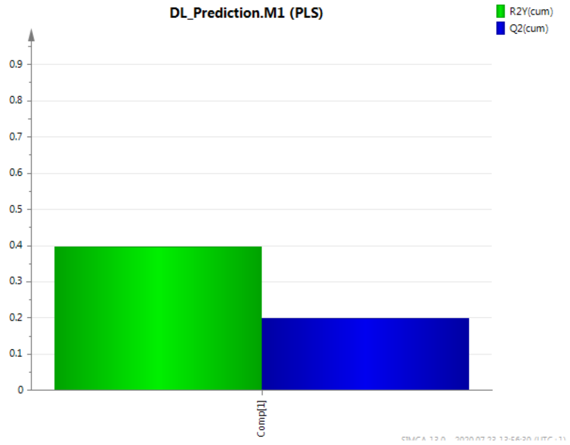

Figure 3.1: Representation of the values of R^2 (green) and Q^2 (blue) for the first and only principal component obtained for the PLS model taking into account all the features, prior to optimization.

Due to the disparity between the two values (the Q^2 value is half the R^2 value) there is a high possibility that, despite the use of cross validation where each fold has a single API, the model is overfitting the data. Since one of the possible reasons for this is the excessive number of descriptive features, it is logical to assess which variables are contributing more or less to modulate the dependant variable of the model. In a PLS, this is done through the previously mentioned parameter VIP (Variable Importance in Projection); the SIMCA software outputs a graphical representation of all the input variables, arranged from higher VIP to lower VIP. Usually, it is stated that variables with a VIP value equal to or greater than one are important for the prediction of the model's output [109–111]. However, in the software's user guide [91] it is stated that the value of 1 isn't necessarily the cutoff value to exclude unimportant variables; they suggest that variables with a VIP equal to or higher than 1 should generally be considered important, and that variables with a VIP value below 0.5 should be excluded. In this user guide, it is also stated that variables with a VIP between 0.5 and 1 belong to a ”grey area”, that may or may not be excluded, varying from case to case. In the present model, it was chosen to initially exclude all the 16 variables with a VIP value lower than 0.5, represented in red in Figure 3.2. An analysis of the variables considered most and least important, as well as their physical significance, can be read in section A.1.1.C, in the Confidential Appendix.

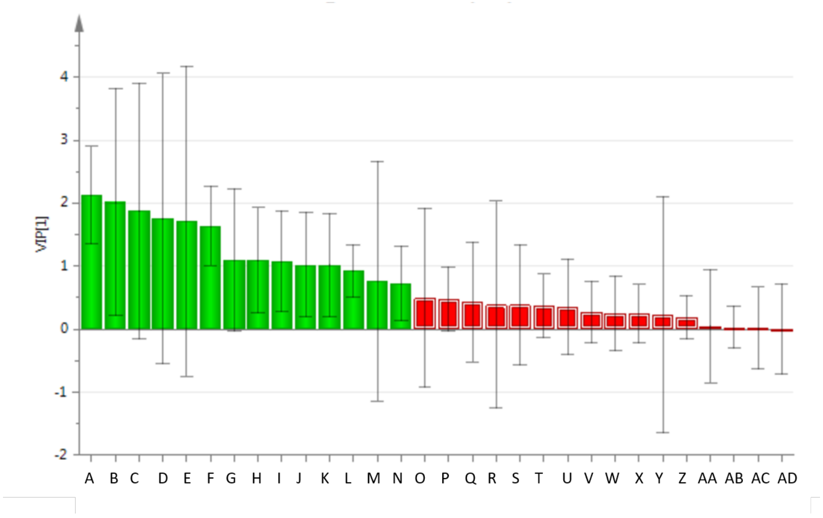

Figure 3.2: VIP plot of the unoptimized model. The variables in red were followingly excluded from the model. The correspondence between the letters and the represented variables are present in Table A.1 (Confidential Appendix).

Upon this optimization step, the value of R^2 was maintained at 40% and the value of Q^2 changed to 26% . While the R^2 value was approximately maintained, the Q^2 improved significantly, supporting the previously stated thesis that the initial model was overfitting the training data. To furtherly improve these values, the next step taken was to analyze the outliers, and choose which outliers to remove. The first way to assess which observations are outliers is to analyze the scores plot (the mapping of the observations according to the first, and only, principal component of the model) - Figure 3.3. The second way to decide if an observation may be worsening the model's predictability is to analyze the normal probability plot of residuals (n-plot) - Figure 3.4.

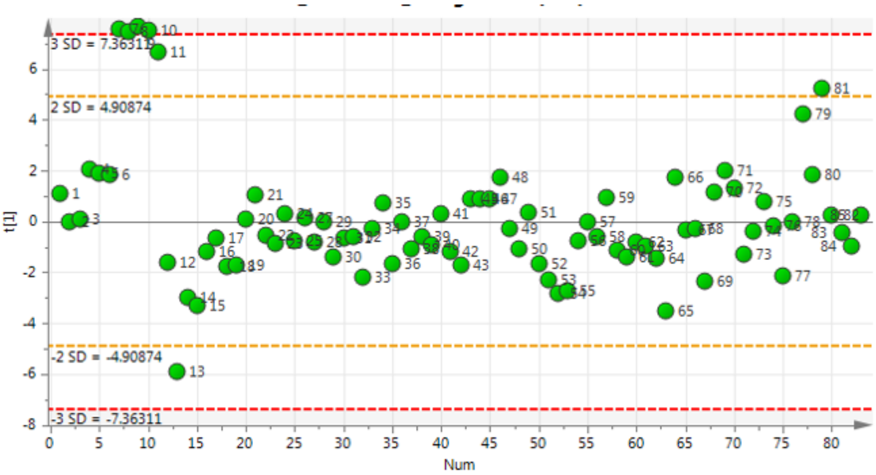

Figure 3.3: Scores plot for the present step of optimization of the model - distribution of the different observations according to the first, and only, principal component obtained.

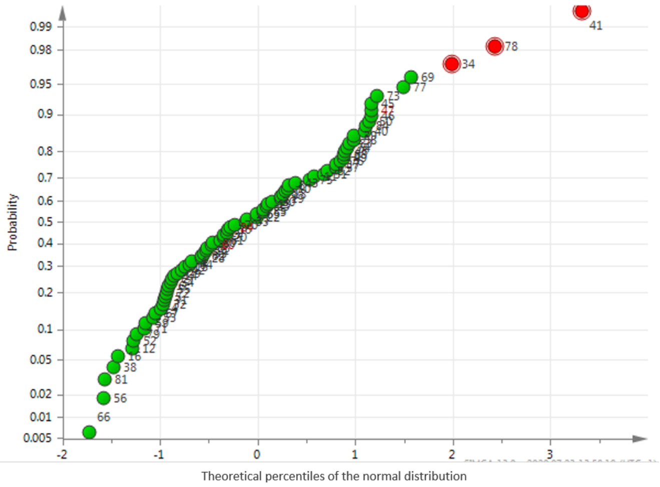

Figure 3.4: Normal probability plot of residuals, which displays the residuals standardized - raw residual divided by the residual standard deviation (values in a double log scale). A dataset with no outliers would have an approximately linear distribution, as mentioned in the methodology section. Observations with a standard deviation of more than 2 or less than -2 standard deviations were excluded (observations 34, 78 and 41).

The exclusion of the outliers based on the scores plot (observations 7-11, 13 and 81) resulted in a model with a much poorer performance (R^2 of 17% and Q^2 of 3.2%); these observations, despite being outside the confidence interval of 95%, are aiding the predictions and not the contrary. In fact, if we represent the dependent variable versus the predicted output (Figure 3.5 A), and the dependent variable versus the predicted output through cross-validation (Figure 3.5 B), it is visible that the outliers detected in the scores plot are not too deviated from the trend line. On the contrary, if the outliers are identified based on the n-plot through threshold of more than 2 or less than -2 standard deviations, three outliers are detected (observations 34, 41 and 78, represented in red in Figure 3.4). It is visible in Figure 3.5 that these outliers are more deviated from the trend line than the ones from the score plots; therefore, the n-plot was chosen as the preferred way to detect outliers for this dataset.

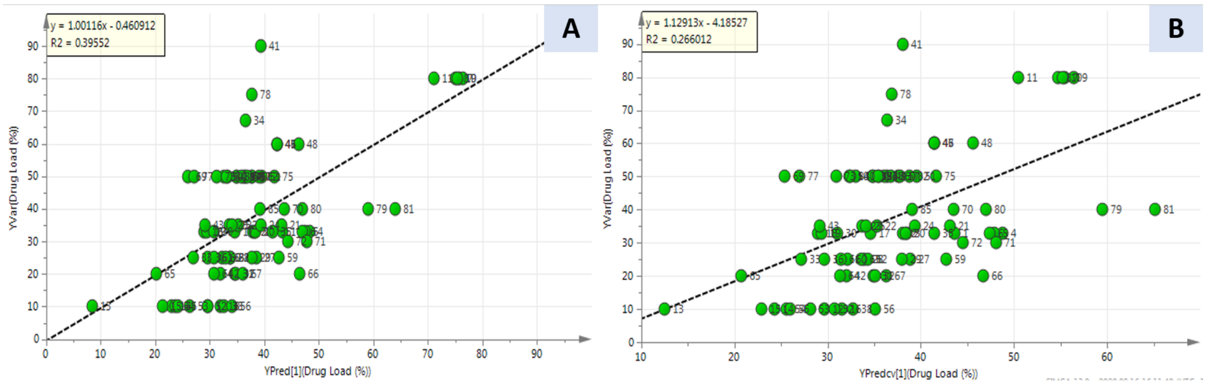

Figure 3.5: Prediction plots. A – actual values of the dependent variable (drug loading) versus the predicted values for this variable; B – actual values of the dependent variable (drug loading) versus the predicted values through cross validation for this variable.

Upon removing the outliers identified in the n-plot, the R^2 and Q^2 values changed to, respectively, 49% and 33%. In order to furtherly optimize the model, three more steps were taken: from the obtained model, one variable had a VIP below 0.5, and was removed (Figure 5.1, supplementary material 1); then, the n-plot was analyzed again and the outliers were removed this time with a threshold of 1.5 standard deviations (Figure 5.2, supplementary material 1); finally, another variable that now had VIP below 0.5 was removed (Figure 5.3, supplementary material 1).

By the end of this optimization process, the final R^2 value was 58%, and the final Q^2 value was 48% (Figure 5.4, supplementary material 1). The final results regarding this model are represented in Figures 3.6 to 3.8.

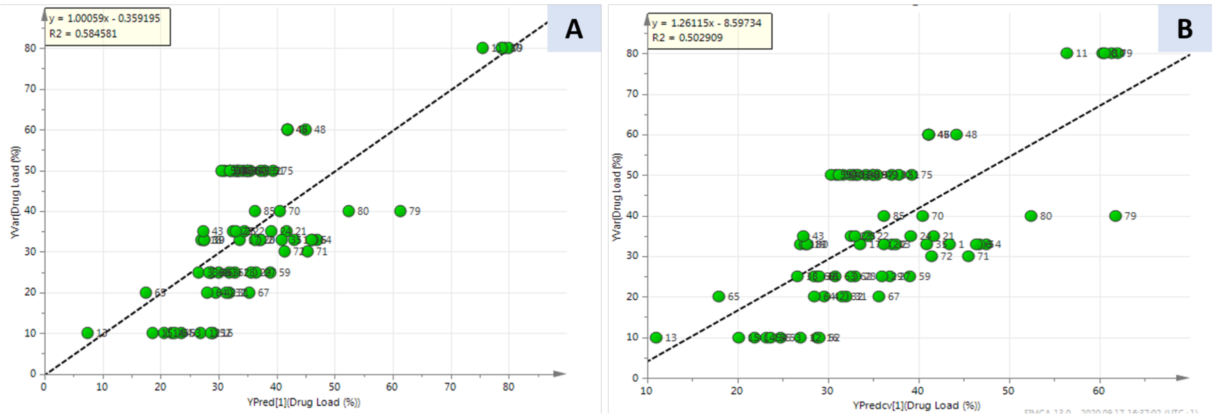

Figure 3.6: Prediction plots for the final model. A – actual values of the dependent variable (drug loading) versus the predicted values for this variable; B – actual values of the dependent variable (drug loading) versus the predicted values through cross validation for this variable.

It is visible in the prediction plots (Figure 3.6) that, even when recurring to cross-validation (plot B), the data points do not fall far from the regression line. In fact, even for the observations with the highest drug loading (observations 7-11, with a real drug loading of 80%), the model using cross-validation (meaning that the model hasn't been exposed to any observations with a drug loading as high) does an acceptable job at extrapolating, predicting values between 55% and 65%. Taking into account that, except those observations, there are only two observations with a drug loading of 60%, and all the other ones have smaller values (as visible in the graphical representation of the data distribution for this model, Figure 2.7), this extrapolation is indeed promising.

It is also worth noting that in both prediction plots, there are more data points falling below the trend line rather than on top; this means that it is more common for the model to predict a drug loading with a higher value comparing to its real value, than to predict a drug loading value that is inferior to the real drug loading. When harvesting the data, when possible the observations were taken from internal reports or literature papers where the authors tested several drug/polymer ratios in order to choose the highest possible API loading that conferred stability to the ASD. However, in many cases the authors did not perform these experiments; instead, often a ”typical” API:Polymer ratio was chosen (for example, 1:2 or 1:3). It is, then, logical to conclude that for many observations the drug loading that was used in the referred paper isn't, indeed, the maximum API loading allowed for that API/polymer combination. Therefore, the fact that the model is predicting more ”higher than real” values rather than ”lower than real” may mean that, at least in some cases, the value input as real may simply not be the maximum possible API loading.

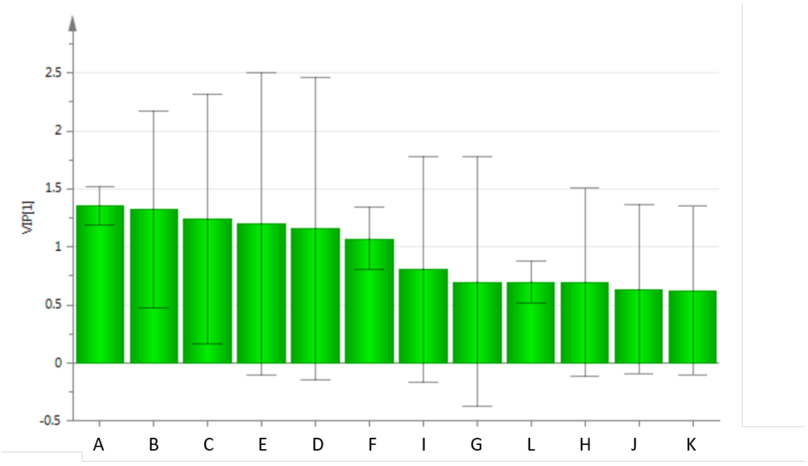

Figure 3.7: VIP plot of the final model, comprising the 12 variables that were retained in the model. The correspondence between the letters and the represented variables are present in Table A.1 (Confidential Appendix).

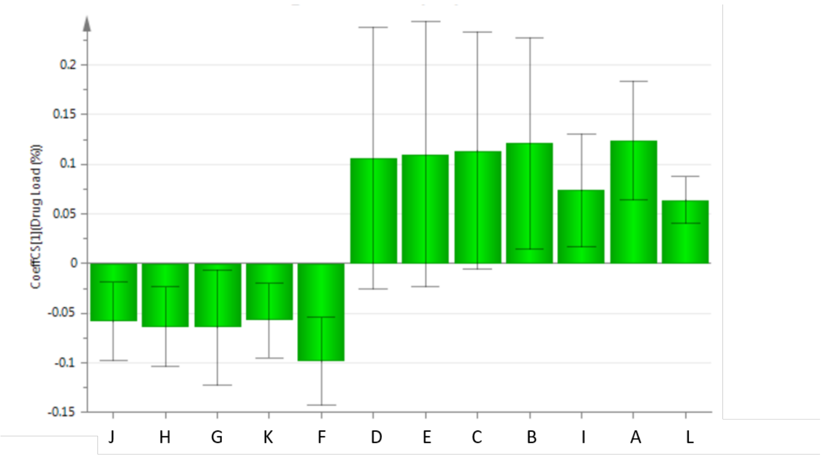

Figure 3.8: Coefficients plot of the final model, representing the positive or negative correlation of each individual variable to the dependant variable. The correspondence between the letters and the represented variables are present in Table A.1 (Confidential Appendix).

The final model is taking as inputs 12 variables: 4 related to the polymer, 7 related to the API, and 1 interaction variable. In the coefficients plot (Figure 3.8) it is possible to see the positive or negative correlations between each individual variable and the dependant variable. An analysis of the variables considered most and least important (Figure 3.7), as well as their correlation with the dependant variable (Figure 3.8) and their physical significance can be found in section A.1.1.C, in the Confidential Appendix.

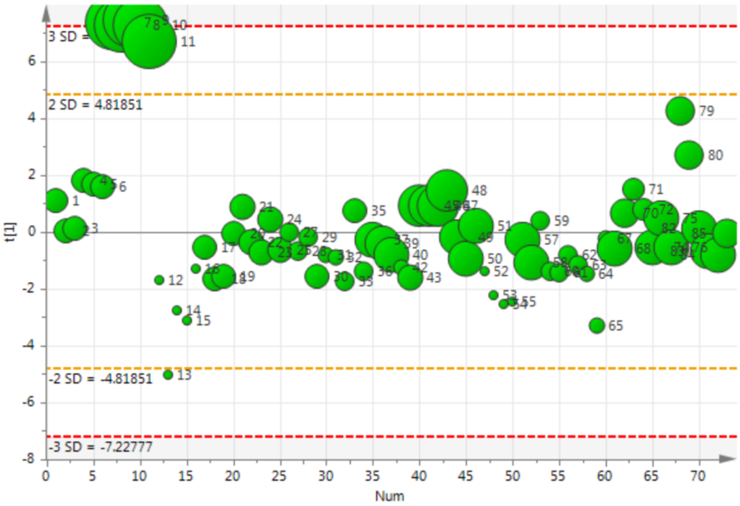

Figure 3.9: Scores plot for the final model - distribution of the different observations according to the first, and only, principal component obtained (t1). The sizing is proportional to the real drug loading of the data points - observations represented by a bigger circle have a greater drug loading.

In the scores plot (Figure 3.9), it is visible that only observations 7-11 are on the line of 3 standard deviations, and observation 13 is on the line of 2 standard deviations. Observations 7-11 are all referent to the same API, only with different polymers. While the API doesn't differ very much from the others, it is the API in the dataset with the highest value for variable C and the lowest value for variable F (correspondence to the names of the variables in Table A.1, Confidential Appendix); since both these variables have a VIP higher than 1, it is possible that this is the cause for the deviation. Another obvious characteristic of these observations is the fact that they have the highest drug loading of the data set. As to the observation 13, the API does not seem to have any distinct characteristics (to be noted that this API is common to observations 12 to 18, but only observation 13 is deviated). This could be due to the fact that the polymer in this ASD is unique in the whole dataset, and it is the polymer with the highest value for variables T and X, and the lowest value for variable O (correspondence to the names of the variables in Table A.1, Confidential Appendix).

All in all, this model seems very promising for the prediction of the maximum API loading of a given ASD, given both the API and the polymer. However, these results need further validation from an external dataset to allow the conclusion that the model, indeed, succeeds at predicting this output. This validation will be presented in subsection 3.1.3, and additional tests in other external datasets will be presented in section 3.3.

3.1.1.B PLS model - API Variables Only

It is very useful to be able to predict the maximum API loading for a given API/polymer combination. However, since the API characteristics are more determining for this output than the polymer descriptors, it is also very interesting (and possibly even more useful) to be able to predict the maximum API loading for a given API prior to any decisions about the polymer. Therefore, a new model was developed: all the polymer variables were excluded, and for each API, only the observation with the highest drug loading parameter was kept. Prior to optimization, the R^2 and Q^2 values obtained were 20% and -4%, respectively (Figure 3.10).

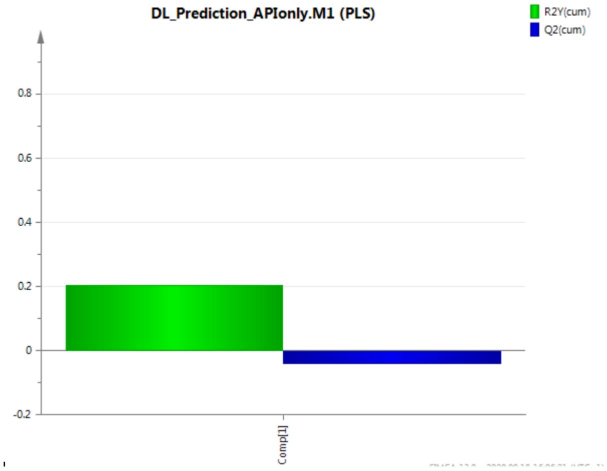

Figure 3.10: Representation of the values of R^2 (green) and Q^2 (blue) for the first and only principal component obtained for the PLS model taking into account only the API features, prior to optimization.

A model with a negative Q^2 is, naturally, not at all successful at predicting the dependant variable. However, from the previous model, it is already known that an excess of variables was diminishing the model's performance (probably due to overfitting), and that the outliers detected in the n-plot were harming the predictive power of the model. Therefore, despite the initial poor performance of the non optimized model, it was chosen to still try to optimize the model. Six steps were taken: removing six variables with VIP lower than 0.5 (Figure 5.5, supplementary material 1), then removing three outliers with more than 1.5 or less than -1.5 standard deviations (Figure 5.6, supplementary material 1), then removing one variable with VIP lower than 0.5 (Figure 5.7, supplementary material 1), then removing three outliers using the same rule as previously (Figure 5.8, supplementary material 1), in another step removing two outliers that were causing ”tails” in the distribution described in Figure 2.8 C (Figure 5.9, supplementary material 1), and finally one last variable with VIP lower than 0.5 was removed (Figure 5.10, supplementary material 1).

By the end of this optimization process, the final R^2 value was 39%, and the final Q^2 value was 31% (Figure 5.11, supplementary material). The final results regarding this model are represented in Figures 3.11 to 3.14.

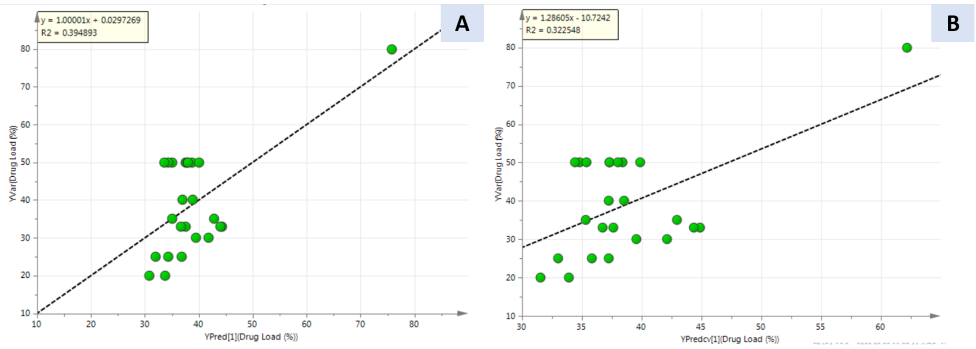

Figure 3.11: Prediction plots for the final API only model. A – actual values of the dependent variable (drug loading) versus the predicted values for this variable; B – actual values of the dependent variable (drug loading) versus the predicted values through cross validation for this variable.

It is visible in Figure 3.11 that, according to what was expected due to the lower number of predictors and observations, for the API only model the predictions aren't as close to the trend line, compared to the model with all the features. However, even using cross validation (plot B), the maximum deviation between the real y value and the predicted y value is approximately 15-20%; this means that, even if the model doesn't predict exactly the maximum API loading for a new formulation, it certainly allows to greatly reduce the spectrum of API loadings to experimentally test. The model is also performing good extrapolations: the API with the highest y value has a drug loading of 80%, and the API with the second highest has a value of 50%; this means that, when using cross-validation (graphic B), the model predicts the value for the first observations based on a dataset that only goes as far as 50%. However, the model is predicting an API loading of over 60% for this API, which falls outside the range in which the model was trained. It is also worth noting that, similarly to the previous model, there are more APIs falling below the trend line rather than on top: the model is more commonly predicting a superior drug loading comparing to its real value rather than the contrary. As previously explained, many observations come from literature papers where the authors did not study the maximum API loading, but instead chose an average API loading, which means some observations may have as a ”real value” a drug loading that is, in fact, not the maximum possible for that given API.

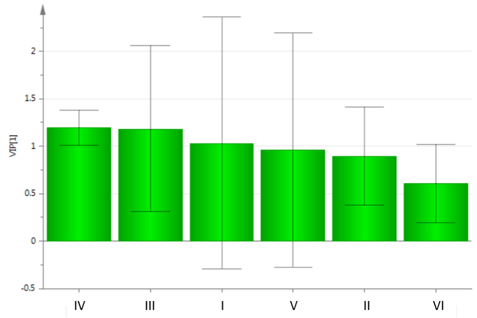

Figure 3.12: VIP plot of the final API only model, comprising the 6 variables that were retained in the model. The correspondence between the letters and the represented variables are present in Table A.2 (Confidential Appendix).

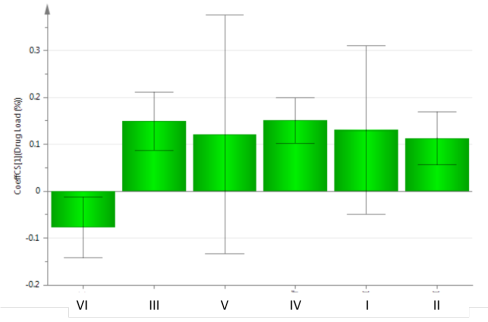

Figure 3.13: Coefficients plot of the final API only model, representing the positive or negative correlation of each individual variable to the dependant variable. The correspondence between the letters and the represented variables are present in Table A.2 (Confidential Appendix).

The final model is taking as inputs 6 variables descriptive of the API. In the coefficients plot (Figure 3.13) it is possible to see the positive or negative correlations between each individual variable and the dependant variable. An analysis of the variables considered most and least important (Figure 3.12), as well as their correlation with the dependant variable (Figure 3.13) and their physical significance can be found in section A.1.1.C, in the Confidential Appendix.

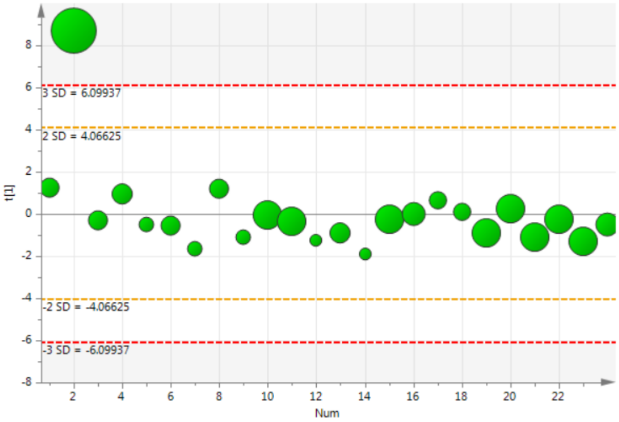

Figure 3.14: Scores plot for the final API only model - distribution of the different observations according to the first, and only, principal component obtained (t1). The sizing is proportional to the real drug loading of the data points – APIs represented by a bigger circle have a greater drug loading.

In the scores plot (Figure 3.14), only one observation is considered an outlier. This API has the highest value for variable C and the lowest value for variable F (correspondence to the names of the variables in Table A.1, Confidential Appendix) in the dataset, and was already considered an outlier in the scores plot of the previous model (Figure 3.9); also contributing to the difference between this observation and the remaining APIs may be the fact that it is the observation with the highest drug loading.

Despite being, as expected, less accurate than the previous model, this model making use of uniquely 6 API descriptors is very promising in aiding the reduction of experimental testing: since the deviation from the real value is approximately 15-20% using cross validation and the scientist only needs to harvest 6 descriptors, the utilization of this model may highly contribute to the minimization of time and resources consumed in the process of ASD screening. However, these results need further validation from an external dataset, which will be represented in subsection 3.1.3, and additional tests in other external datasets will be presented in section 3.3.

3.1.2 Random Forest model

The random forest classifier, dividing the drug loadings into classes, was the chosen machine learning model to test in the prediction of this output. In this model, it is possible to assess its performance through the general accuracy of the model, and through four other parameters: the true positive rate (TPR), or sensitivity (equation 3.1), the true negative rate (TNR), or specificity (equation 3.2), the false negative rate (FNR), or miss rate (equation 3.3), and the false positive rate (FPR), or fall-out (equation 3.4). In the present model, the evaluation parameters will be the model's general accuracy, as well as the TPR and FNR through the confusion matrix – an example confusion matrix can be found in Figure 3.15.

$$TPR = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{3.1}$$

$$TNR = \frac{TrueNegatives}{FalsePositives + TrueNegatives} \tag{3.2}$$

$$FNR = \frac{FalseNegatives}{TruePositives + FalseNegatives} \tag{3.3}$$

$$FPR = \frac{FalsePositives}{FalsePositives + TrueNegatives} \tag{3.4}$$

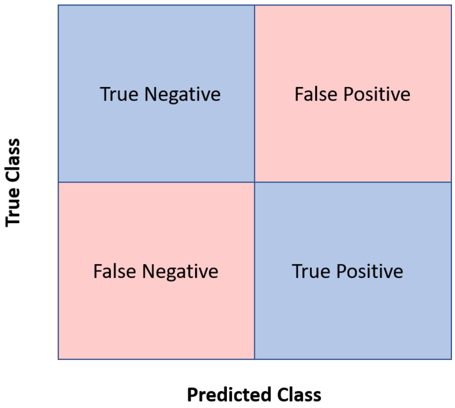

Figure 3.15: Generic example of confusion matrix, for a binary prediction. The number of rows and columns is equal to the number of classes in the model.

The model was, then, trained on all the observations and features, as described in the methodology section. The overall accuracy for the model indicated by MATLAB was 78.8%. The confusion matrix referent to this model is presented in Figure 3.16

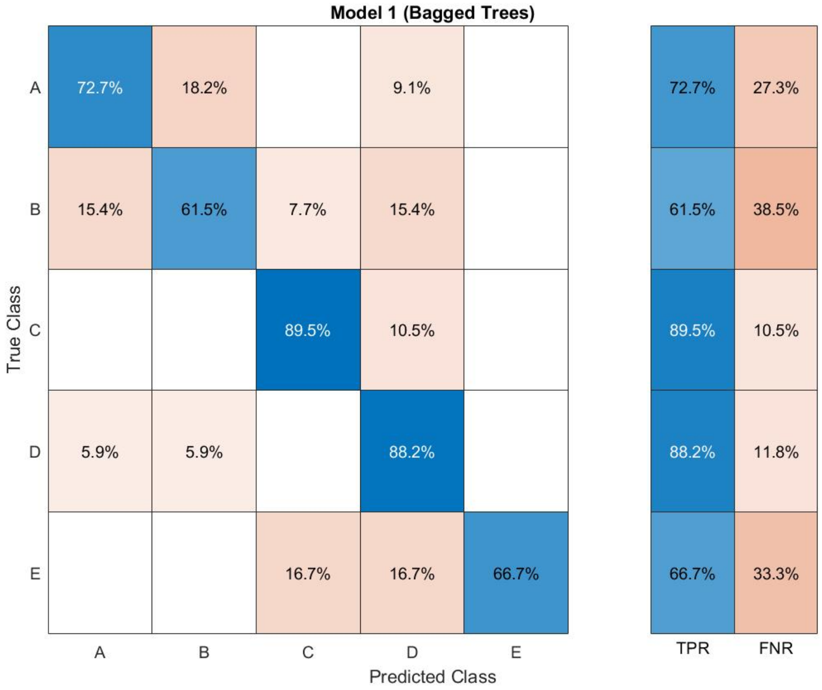

Figure 3.16: Confusion matrix referent to the random forest classifier model developed for the prediction of the maximum API loading. In the x axis is the predicted class, and in the y axis the true class. The diagonal (in blue) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class. The correspondence between class letters and API loading ranges can be found on Table 2.2.

It is visible that, with exception of classes B and E, all the classes have a very high TPR (all above 70%). This, by itself, is a good indicator of the performance of the model, as well as the 78.8% overall accuracy. However, the class B's low TPR is somewhat worrying: not only is this value low, but in addition, one of the most commonly predicted classes for the observations that belong in class B is actually class D, meaning they have a disparity of over 15%. This could, partially, be explained by the fact that, as previously explained, some observations are likely to have API loadings that don't correspond to the true maximum. However, it is also visible that, in class D, all the misclassified observations were classified into class B and class A; this means that half of the misclassified observations have a disparity of more than 30%, and all the misclassified observations have a disparity of more than 15%. This is particularly delicate because the model is predicting wrongfully a lower value, and not a higher value – this means that those predictions are necessarily wrong, since a higher API loading was formulated and, therefore, is naturally doable. In class E, the same happens: half of the wrongly predicted observations are two classes below the correct value. However, overall the confusion matrix results and the accuracy do indicate a promising model, but additional tests would need to be performed to assess this.

The random forest model allows one to evaluate which variables are contributing more or less to the separation into classes through parallel coordinates plots. This plots show, for each variable (x axis), the standard deviation (y axis) represented in different colours for different output classes: this way, if for a given variable the colours are distinctly separated, or at least somewhat differentiated, that variable is important for the prediction capacity of the model. If, on the contrary, the colours are indistinguishable or overlapping, the feature is not providing good separation characteristics. In Supplementary Material 2 are presented these plots for the present model (Figures 5.12 to 5.15). The correspondence between the plots' labels and the variables is presented in Table A.5 (Confidential Appendix).

This process of selecting the variables that allow more and less separation is somewhat biased, since it requires manual analysis and, therefore, might differ from user to user, especially if the separation isn't very clear. The variables considered to allow the most separation according to Figures 5.12 to 5.15 were V, A, B, I and D. The variables that were considered to provide less separation were T, X, W, N, Q, F and Z.

For a random forest classifier with five possible classes, the number of observations in this dataset (67) is little, especially since this is a non-linear algorithm (which usually require a much larger dataset [112]); despite the fact that cross-validation was used, it is likely that overfitting is occurring, due to the low number of observations and high number of features, especially since the accuracy for the training set is quite high (77.3%). Therefore, a possible next step would be to remove features that do not seem to be adding any benefits to the model. Initially, these variables were removed individually and the resulting accuracy was collected, being presented in Table 3.1.

Table 3.1: Accuracy resulting from removing each individual variable of the ones identified as providing less separation between classes, for the random forest model developed for prediction of the ”maximum API loading” output. The correspondence between the plots' labels and the variables is presented in Table A.5 (Confidential Appendix).

Variable Removed	Resulting Accuracy
T	81.8%
X	78.8%
W	77.3%
N	80.3%
Q	78.8%
F	78.8%
Z	74.2%

It is visible that removing variables T and N improves the model's performance, and removing variables X, Q or F maintains it; on the contrary, removing variables W and Z lowers the model's accuracy. However, if all the variables are removed at once, the accuracy remains at 78.8%; therefore, since the dataset is small, all the seven features were removed. The resulting confusion matrix is presented in Figure 3.17. It is visible that the TPR for class B diminished with this step of optimization, with now a value of 53.8%. For class A, less values higher than real were predicted, and for class C and D both the TPR and FNR remained the same. For class E, there are now observations from class B that were predicted into class A, which implies a disparity of over 40%. However, this model may be preferable in terms of overfitting the training data, and therefore both models will be tested in the external validation group, and in the other tests performed.

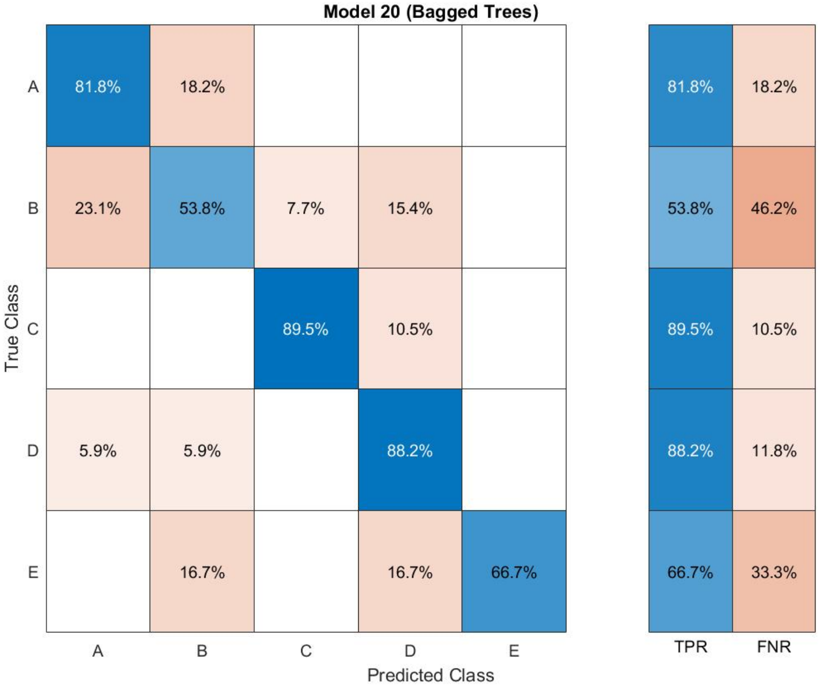

Figure 3.17: Confusion matrix referent to the random forest classifier model developed for the prediction of the maximum API loading, after removing the variables mentioned in Table 3.1. In the x axis is the predicted class, and in the y axis the true class. The diagonal (in blue) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class. The correspondence between class letters and API loading ranges can be found on Table 2.2.

Despite the fact that, as mentioned, non-linear algorithms need an acceptable amount of data to be correctly trained on, a RF model based on uniquely the API variables was attempted. This model yielded a 35.5% accuracy, and the confusion matrix referent to it is presented on Figure 3.18. As expected, this model performed poorly: the overall accuracy is very low, and there are two classes with a FNR of 100%. Once again, class C is the one with the best performance: it has a TPR of 71.4% (it is the only class with a TPR of over 50%). Therefore, this model was not furtherly pursued.

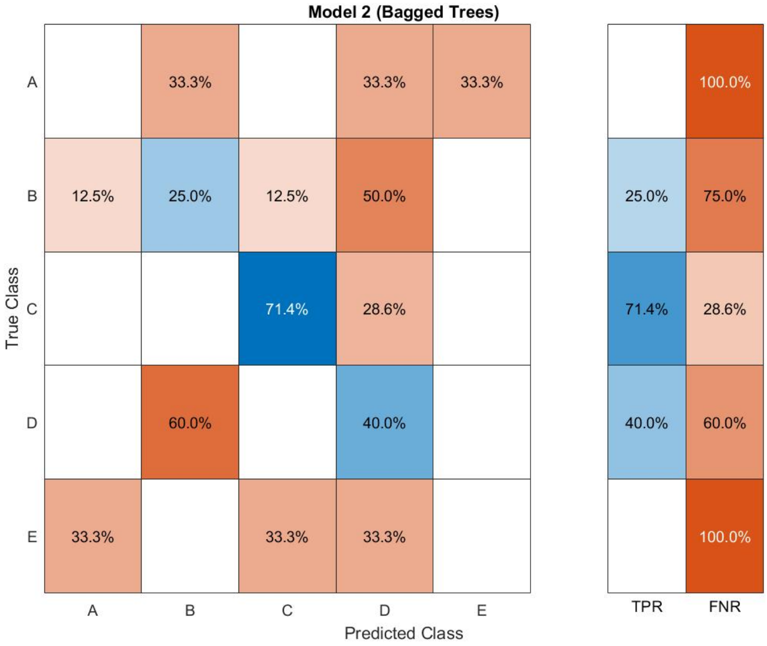

Figure 3.18: Confusion matrix referent to the random forest classifier model developed for the prediction of the maximum API loading, with the API features only. In the x axis is the predicted class, and in the y axis the true class. The diagonal (in blue) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class. The correspondence between class letters and API loading ranges can be found on Table 2.2.

3.1.3 External validation of results

To further validate the results obtained, an external, completely independent dataset was harvested to be used as an external validation set ([76–85]). The results obtained are presented on Table 3.2.

Table 3.2: API loading predictions by the developed models for the output "API loading" for an external validation dataset (observations harvested from literature). Yellow: observations that, for the PLS, have a difference between prediction and real value of over 10%.

API	Polymer	API load (%)	Prediction PLS All Features	Prediction PLS API Only	Prediction Random Forest	Prediction Random Forest Optimized
Rebamipide	PVP K30	**33**	44	36	C	C
Taranabant	HPMCAS L	**10**	27	23	D	B
Raloxifene	PVP K30	**20**	38	30	C	C
Itraconazole	HPMCP HP55	**33**	33	38	C	C
Sirolimus	Eudragit E	**33**	33	28	C	C
Sirolimus	HPMC	**10**	25		C	C
Andrographolide	PVP K30	**33**	32	23	A	E
Piroxicam	PVP K25	**20**	45	38	C	B
Tadalafil	PVP/VA 64	**50**	38	31	E	C
Rivaroxaban	Eudragit 100L	**43**	36	29	C	C
Ciprofloxacin	HPMC E3	**50**	43	36	B	B

Beginning with the PLS, if we calculate a percentage of correct predictions taking into account the cutoff of 10% difference, values of 45% and 50% are obtained for the model with all features and the model with only the API features, respectively. These values are merely representative, since this validation set has only 11 observations (10 for the model with API variables only). Moreover, the 10% value is an arbitrary cutoff; if this cutoff was altered to 15%, these accuracies go up to 73% and 70%, respectively. This cutoff of 15% would be perfectly acceptable: even if the model does not yield the exact, final API loading, it allows a great reduction of the interval of drug loadings to experiment.

As for the RF mode, the results on the external set weren't as promising. In the unoptimized model, all the observations assigned to the correct class, or to a near class, were assigned to class C; this shows that the model is being susceptible to the skew present in the data (Figure 2.10). In the model with 21 instead of 28 features, the correct or nearly correct predictions were belonging to classes B and C; this still reflects the sken on the data. Observations with lower API loadings are not being as correctly predicted by this model. To be noted that this validation set has more observations belonging to class C than any other class, and only two belonging to classes A, B and D, and none belonging to class E. The unoptimized model isn't predicting correctly for observations with lower API loadings, but the second model performs somewhat better: out of the 4 observations belonging to classes A and B, the first model has two wrong predictions and two nearly-correct predictions (both classified in class C), and the second model has only one wrong prediction, one correct prediction (class B) and two nearly-correct predictions (one for class B and one for class C). This model is, then, less likely to only predict correctly observations in class C than the first model. If we calculate an accuracy value as was done for the PLS model, values of 36% and 45% were obtained for the unoptimized and optimized RF models, respectively. If the nearly-correct predictions are also counted, then these values go up to 55% and 73%, respectively. However, if nearly-correct predictions are considered as correct for the purpose of evaluating the model's performance, if a scientist used the model to predict an API loading and it yields, for example, class C, then the model would be yielding an API loading between 31% and 45%, but in reality, the scientist would have to assume this value could go from 16% (beginning of class B) to 70% (ending of class C). This is the big disadvantage of using a classification model to predict this output through ”labels” – if an observations has a true API loading of 40% (class C), the penalty is the same if the model predicts 46% (beginning of class D) or 70% (ending of class D). Therefore, despite being a good exercise to analyze the important variables and to assess its feasibility, this model isn't ideal for the present purpose – the PLS model was chosen as the final model to predict the API loading in a real day-to-day setting.

3.2 Development of models for the prediction of Spring and Parachute effect

For the analysis of this output, two models were explored: an ANN model, and a RF classifier model.

3.2.1 Random Forest model

The random forest classifier was also explored in the prediction of the spring and parachute effect, through the assignment of the observations to a label ”0” (no spring and parachute effect) or ”1” (good spring and parachute effect). This model was trained as described in the methodology section. The overall accuracy for the model indicated by MATLAB was 78.3%. The confusion matrix referent to this model is presented in Figure 3.19.

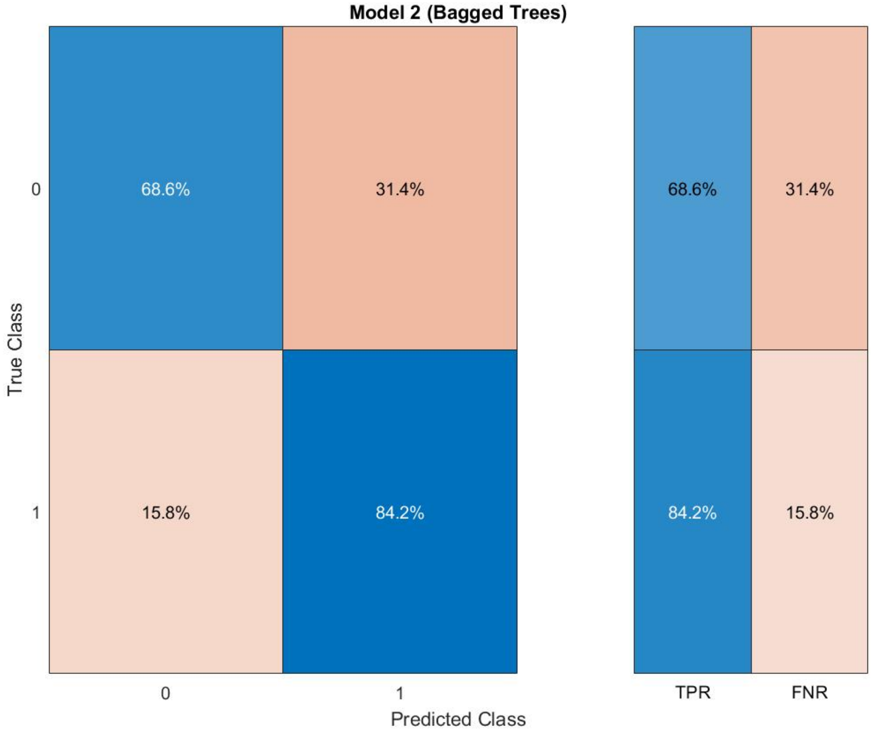

Figure 3.19: Confusion matrix referent to the random forest classifier model developed for the prediction of the spring and parachute effect. In the x axis is the predicted class, and in the y axis the true class. The diagonal (in blue) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class.

For class ”1”, the performance seems to be very good – the model has a TPR of 84.2%. For the class ”0”, the TPR is lower, but still promising – 68.6%. This numbers may indicate that the model is more frequently predicting a positive output rather than a negative output – in practical terms, this means that it is more probable for the model to predict a bad ASD to be good, rather than to predict a good ASD to be bad. For the present purpose, it is actually an advantage for the classifier to more frequently predict false positives rather than false negatives – the prediction of false negatives may lead to the scientist not testing experimentally an API/polymer combination that may actually be successful. This may be a reflection of the data skew presented in Figure 2.12.

Once again, the most and least relevant variables were evaluated through the parallel coordinates plots, that show for each variable (x axis) the standard deviation (y axis) represented in different colours for different output classes. In Supplementary Material 3 are presented these plots for the present model (Figures 5.16 to 5.19). The correspondence between the plots' labels and the variables is presented in Table A.7 (Confidential Appendix).

Once again, the variables that allow more and less separation were manually picked through the presented plots. The variables considered to provide the best separation were AD, T, X, S, AC, C and A. The variables that appeared to allow no separation between the two classes were W, H, N, G, K, AB, V, U, Q, F, M, D and Z.

A non-linear algorithm such as the random forest classifier requires a large enough dataset, as it has been mentioned; therefore, despite the use of 20-fold cross validation, overfitting may be occurring. A possible resolution is to remove variables that are only adding noise to the model. Initially, these variable were removed individually and the resulting accuracy is presented in Table 3.3.

Table 3.3: Accuracy resulting from removing each individual variable of the ones identified as providing less separation between the two classes for the random forest model developed to predict the "spring and parachute effect" output.

Variable Removed	Resulting Accuracy
W	78.3%
H	80.4%
N	77.2%
G	80.4%
K	76.1%
AB	79.3%
V	78.3%
U	79.3%
Q	77.2%
F	78.3%
M	78.3%
D	76.1%
Z	76.1%

The removal of the variables H, G, AB and U enhanced the model's accuracy, while removing the variables N, K, Q, D and Z diminished it. The individual removal of the remaining variables maintained the model's accuracy. When removing all the variables mentioned in the table, the model's accuracy was lowered to 73.9%; when removing only the variables which the removal enhanced the accuracy and the ones which maintained it, the model's accuracy was maintained at 78.3%. Therefore, the variables that worsened the performance of the model upon their removal (N, K, Q, D and Z) were kept. After this optimization step, the confusion matrix presented in Figure 3.20 was obtained.

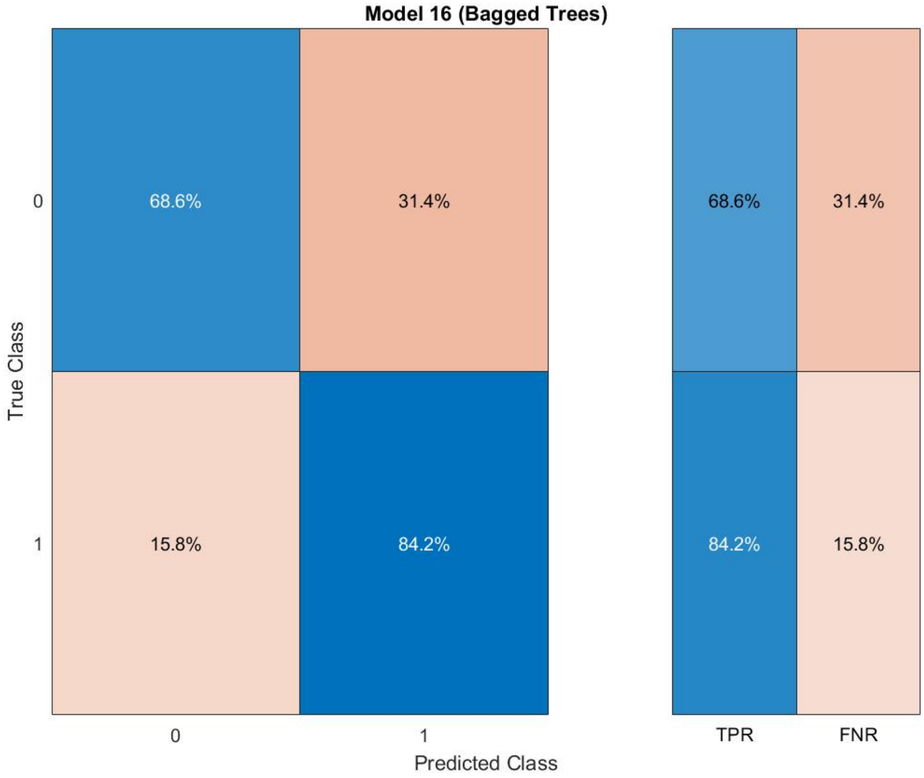

Figure 3.20: Confusion matrix referent to the random forest classifier model developed for the prediction of the spring and parachute effect, after the optimization step. In the x axis is the predicted class, and in the y axis the true class. The diagonal (in blue) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class.

If the Figures 3.19 and 3.20 are compared, one will notice that they are exactly similar. This isn't surprising, since the accuracy of the model was maintained; this, therefore, means that the removed variables were merely adding noise and, while removing them did not improve the performance of the model on the training data, it may have reduced the overfitting (this model has less eight variables than before). Therefore, the external validation performed on subsection 3.2.3 was performed on both models (before and after optimization).

Despite this highly diminishing the number of observations, a model taking into account only the features relating to the API was attempted, and an accuracy of 60.9% was obtained. The confusion matrix referent to this model is presented on Figure 3.21.

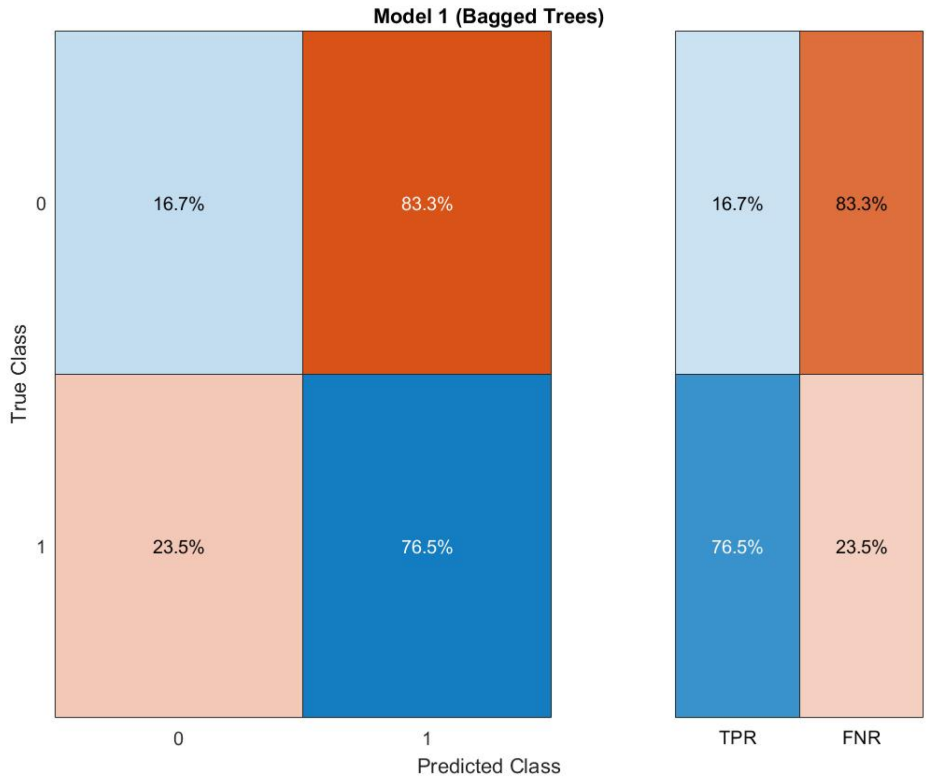

Figure 3.21: Confusion matrix referent to the random forest classifier model developed for the prediction of the spring and parachute effect, based solely on the characteristics of the API. In the x axis is the predicted class, and in the y axis the true class. The diagonal (in blue) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class.

Despite the apparently acceptable accuracy, the ”0” class has 83.3% of FNR; since the TPR for class ”1” is 76.5%, the model is predicting ”1” much more frequently than it is predicting a negative output. This may be due to the fact that the data is highly skewed (Figure 2.13). Therefore, there was no follow through for the development of this model.

3.2.2 Artificial Neural Network model

Despite being a ”black-box” model where the most and least important variables aren’t known, an ANN model was attempted at the prediction of the spring and parachute effect. The performance was assessed through the confusion matrix of each dataset (training, validation and test) and for the overall model, through the accuracy for each dataset set and for the overall model, and through the receiver operating characteristic (ROC) curve for each dataset and for the overall model. A ROC curve is a graphical representation of the TPR (y axis) versus FPR (x axis) relationship at different classification thresholds. One can evaluate the model's accuracy in two different ways through these graphics: (1) the closer the graphic is to the top and left, the better the performance (and, in opposition, the closer to the diagonal the less accurate); and (2) through the area under the curve (AUC). The AUC represents the probability that the model ranks a random positive example more highly than a random negative example (assuming 'positive' ranks higher than 'negative') [113].

The confusion matrices for the different subsets of data used in the training of the ANN are represented on Figure 3.22 (from left to right, up to down: training set, validation set, test set, overall dataset). In Figure 3.23 are represented the ROC curves, in the same order.

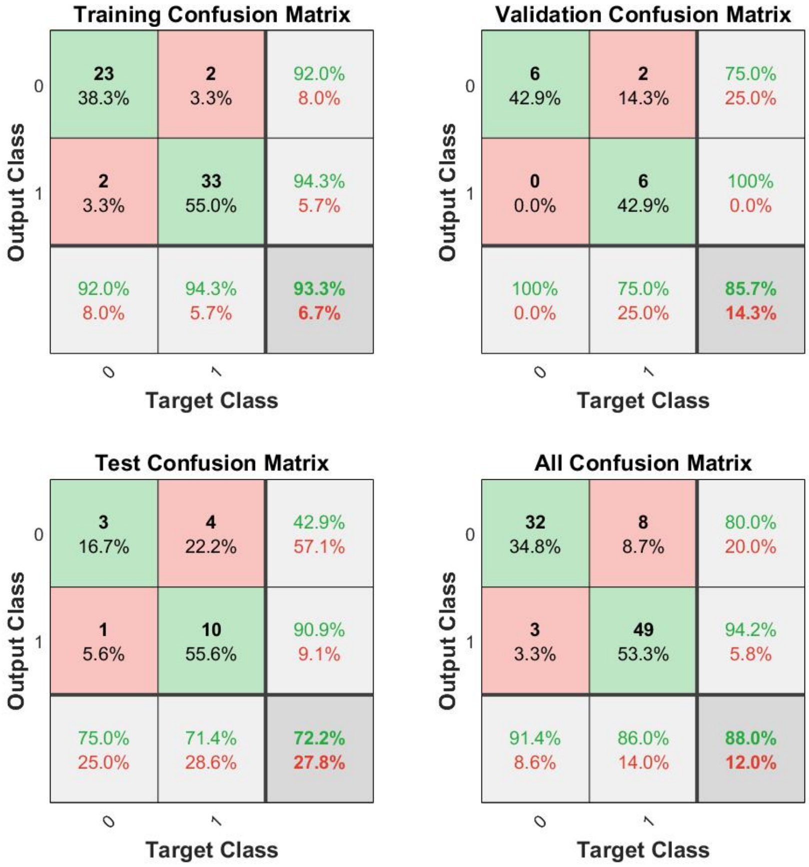

Figure 3.22: Confusion matrices referent to the several subsets of data used in the training of the ANN model for the prediction of the spring and parachute effect. From left to right, up to down: training set, with 60 observations; validation set, with 14 observations; test set, with 18 observations; overall dataset, with 92 observations. The diagonal (in green) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class.

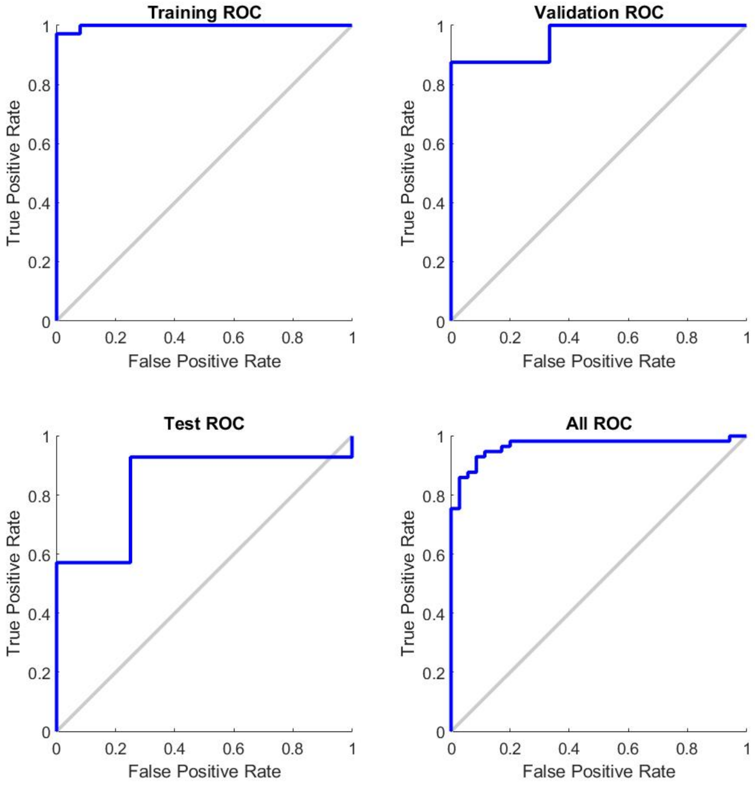

Figure 3.23: ROC curves referent to the several subsets of data used in the training of the ANN model for the prediction of the spring and parachute effect. From left to right, up to down: training set, with 60 observations; validation set, with 14 observations; test set, with 18 observations; overall dataset, with 92 observations.

For the spring and parachute effect models, a theoretical goal of 80% accuracy was established as a guideline – if the intended formulations are tested in an 80% accuracy model, of 5 formulations predicted to be good, 4 will actually be (or in 10 tested formulations predicted to be good, 8 will actually be). The overall accuracy of the model obtained was 88% (Figure 3.22, lower right corner), highly surpassing this objective. However, this accuracy takes into account training, validation and test sets accuracy. All these accuracies are very promising: 93.3% for the training set (Figure 3.22, upper left corner), being naturally the highest; 85.7% for the validation set (Figure 3.22, upper right corner); and 72.2% for the test set (Figure 3.22, lower left corner). The most relevant value belongs to the test set (the model's performance on a completely unseen and independent dataset). Despite not reaching the 80% guideline threshold, it is nonetheless very promising: for 10 observations predicted to be good, 7 will actually be. In practical terms, this can be a very helpful model, since its main objective would be to alleviate the experimental testing and all the costs that come with it in the preliminary ASD screening steps, and not to predict an actual final result.

In relation to the ROC curves, as it has been mentioned, the closer to the upper left corner, the better. The diagonal line $x = y$ represents the randomly guessing of the class – a random classifier will produce a ROC curve that switches between both sides of the diagonal based on the frequency with which it guesses the positive class. Any classifier that appears below the diagonal line does worse than random guessing – it works as if the classes were switched. Both the training and the overall ROC curves (Figure 3.23 upper left corner and lower right corner, respectively) appear to be very good; the validation ROC curve is also very near the upper left corner. For the test set (as well as for the validation set), the dataset size is very small, and therefore, the conclusions are merely tentative observations. In the test set, the classifier appears to perform better in the left size of the graphic (it is better at identifying likely positives than at identifying likely negatives), since at the end of the plot, the curve crosses the diagonal. The optimal threshold for FPR seems to be around 0.25 – for a balanced dataset, this value should be around 0.5. This reflects the data skew presented in Figure 2.12. As mentioned before, for the present purpose it is actually an advantage for the classifier to more frequently predict false positives rather than false negatives – the prediction of false negatives may lead to the scientist not testing experimentally an API/polymer combination that may actually be successful. However, if the FPR is too high, the model becomes superfluous.

Since the accuracy for the training data (93.3%) is approximately 20% higher than the test data accuracy (72.2%), it is probable that some overfitting of the data is happening. Nonetheless, as discussed previously, this value is more than adequate for the proposed objective, and very promising. To be noted that the division between training, validation and test data was done randomly – this was done to ensure that the subsets of data were more homogeneous, since the initial observations of the dataset were taken from internal reports, and the final ones were from external literature papers. This may bring the additional challenge mentioned in the discussion of the PLS model – since there are several observations with the same API, the model may be performing the validation and the testing in some observations with an already seen API. However, it would also be biased to manually choose which observations would be assigned into the test and validation groups, so random division was the chosen method.

Despite the fact that the model's training includes performing a test on an independent set, due to these limitations this model will also be subject to external validation in subsection 3.2.3, and to other additional tests presented in section 3.3.

3.2.3 External validation of results

To further validate the results obtained, an external, completely independent dataset was harvested to be used as an external validation set ([76–85]). The results obtained are presented on Table 3.4 (to be noted that this external dataset is not exactly equal to the validation set used for the API loading output).

Table 3.4: Spring and parachute effect predictions by the developed models (artificial neural network, random forest, and random forest post-optimization) for this output for an external validation dataset (observations harvested from literature). Red: observations that are classified in the wrong class.

API	Polymer	Spring and Parachute Effect (real)	Prediction ANN	Prediction RF	Prediction RF Optimized
Rebamipide	PVP K30	**1**	1	1	1
Raloxifene	PVP K30	**1**	0	1	1
Sirolimus	Eudragit E	**1**	1	0	0
Sirolimus	HPMC	**0**	0	0	0
Tadalafil	PVP/VA 64	**1**	1	1	1
Rivaroxaban	Eudragit 100 L	**1**	0	1	1
Ciprofloxacin	HPMC E3	**0**	1	1	1
Sorafenib	PVP/VA 64	**1**	0	0	0
Sorafenib	PVP K30	**0**	0	0	0

First of all, it is noticeable that both random forest classifiers have predicted exactly equal outputs for this dataset. Since they have the same accuracy and exactly the same confusion matrix, this isn't unexpected; the removed variables were likely to merely provide noise, but since random forest models aren't prone to overfitting due to tree bagging, removing the features likely did not change the way the model predicted the output at all. However, it is still advantageous to have an algorithm run on as little features as possible in order to diminish the time and resources spent on calculating, computing or experimentally obtaining the necessary variables for each observation to be tested. Therefore, from now on, only the optimized random forest classifier will be considered.

Comparing the ANN and the RF models, for this external validation set one can obtain accuracies of 56% and 67%, respectively. Naturally, due to the small size of this validation set, this accuracies are merely representative. However, an accuracy of 56% is not significant for a model with binary outputs – it is very close to 50%, which would mean that the predictions are merely arbitrary. However, an accuracy of 67% is high enough to be considered non-arbitrary. Despite being far from the 80% guideline used, taking into account the fact that, as mentioned, this accuracy is representative due to the limited size of the external validation set used, this model seems quite promising for a preliminary screening application. Moreover, the difference between the training accuracy and the external validation accuracy is approximately 11%, which indicates that if any overfitting is occurring, it is not to an extend that it would impede the utilization of the model. As for the ANN model, the accuracy obtained for this dataset is very far from the test-set accuracy obtained during the training of the model (72.2%); therefore, this model will not be promptly discarded, but the random forest model was chosen as the preferential one.

3.3 Additional tests performed to the models

3.3.1 Predicting the maximum API loading for a set of commercial ASDs

In order to make sure that the model was accurate enough to provide confidence for future scientists and clients to use it in the day to day ASD formulation process, some tests were performed in a set of real commercialised ASD formulations. If the model provided good predictions for APIs that were previously and successfully formulated as ASDs and commercialized as such, then the confidence in the model would increase greatly.

To begin with, the observations used to train the PLS model and the commercial observations were joined in a single dataset, and after removing the variables and observations removed in the original PLS model, a PCA with three principal components was performed (with an R^2 of 46% in the first component, 73% in the second component and 81% in the third component). The scores map obtained for the two first principal components is represented in Figure 3.24, where the observations used in the process of model training are represented in red and the commercial observations are represented in blue.

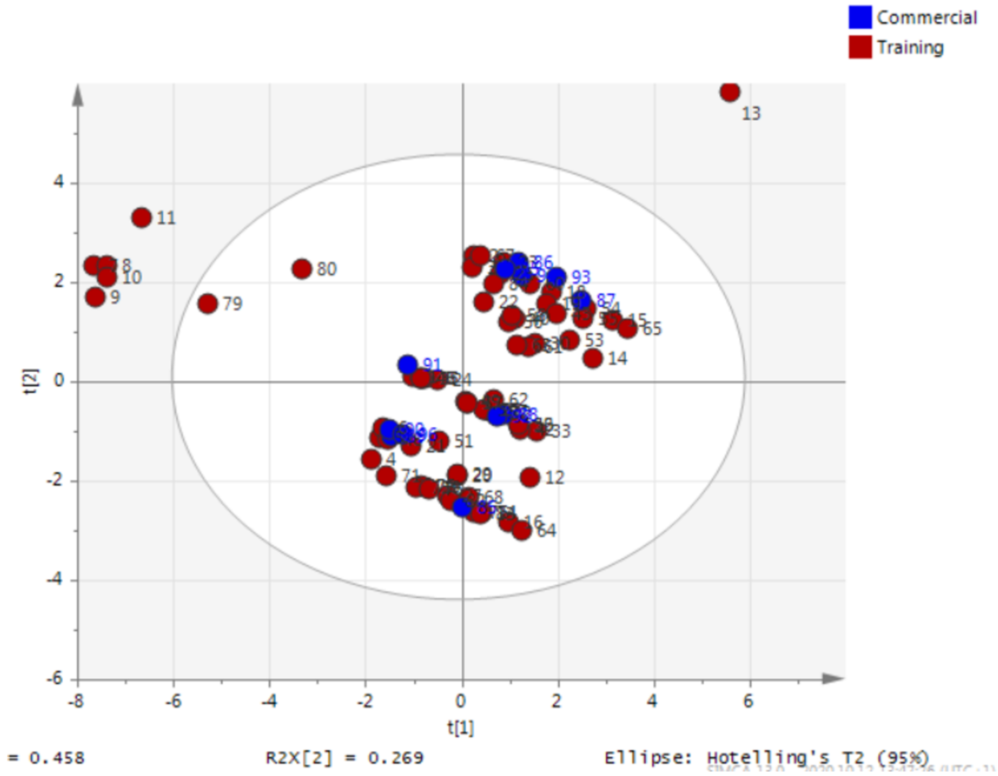

Figure 3.24: PCA map: distribution of the observations used in model training (red) and of the commercial ASDs (blue) across the first two principal components from the PCA analysis performed.

By analyzing Figure 3.24, one can see that the commercial ASDs perfectly adjusted to the PCA map built: there is not a single outlier in the commercial ASDs set, and they are homogeneously distributed amongst the training ASD observations. This gives extra confidence in the results to reluctant clients: nowadays, there is still a lot of reluctance in formulating APIs as amorphous solid dispersions due to the unpredictability of whether the formulation will be a success; however, if the new API falls into the confidence zone of this PCA map, it means that it is similar both to APIs used in the training of the model, and to APIs that were successfully formulated as ASDs – therefore, the new API is likely to be a good fit to be formulated as an ASD, and the predictions made by the model if the API is run by it are likely to be accurate.

Followingly, the set of commercial ASDs was run through the models developed for the prediction of the API loading (PLS for API and polymer features, PLS for the API only, random forest classifier with all the features and random forest classifier optimized). The results obtained, as well as the real API load for these observations, are presented on Table 3.5.

Table 3.5: API loading predictions by the developed models for the output ”API loading” for set of commercial ASDs. Yellow: observations that, for the PLS, have a difference between prediction and real value of over 10%. Green: observations that, for the RF models, are classified in the wrong class, but in a class adjacent to the real class (from now on, referred to as ”nearly-correct predictions”. Red: observations that, for the RF models, are classified two or more classes away from the real class. Purple: observations with unknown real API loading values.

API	Polymer	Prediction PLS API + Polymer (%)	Prediction PLS API only8 (%)	Prediction Random Forest Classifier (%)	Prediction Random Forest Classifier Optimized (%)	Real Formulated API Load (%)
Elbasvir	HPMC	31	43	D	A	**25**
Evacetrapib	HPMC	24	36	D	A	**50**
Torcetrapib	HPMCAS L	33	36	B	B	**unknown**
Voxilaprevir	PVP/VA 64	45	45	B	B	**50**
Velpatasvir	PVP/VA 64	44	46	C	C	**50**
Telaprevir	HPMCAS L	42	44	B	D	**50**
Ivacaftor	HPMCAS H	31	37	D	D	**80**
Everolimus	HPMC	25	41	C	C	**unknown**
Etravirine	HPMC	31	41	C	C	**unknown**
Rosuvastatin	HPMC	33	43	D	D	**unknown**
Ledipasvir	PVP/VA 64	43	45	C	C	**50**

Beginning with the analysis of the PLS results, if we take into account the 10% threshold, accuracies of 71% and 57% are obtained for the API+Polymer PLS and the API only PLS, respectively. However, if the threshold is changed to 15% instead of 10%, the accuracy of the PLS taking into account API and polymer is maintained, while the accuracy of the API only PLS also goes up to 71%. These accuracies are merely illustrative, since they are being calculated based on only 7 observations. However, for most of these observations, both PLS models do predict outputs very similar to the real API loading formulated, which is very promising.

As for the random forest classifiers, they are once again failing to predict the output correctly: even for the optimized model, only two observations were correctly predicted. Despite the fact that only two were predicted two or more classes away from the real class, this can't be taken into account as a correct prediction, since the classes have very broad drug loading intervals.

3.3.2 Predicting the maximum API loading for a set of internal projects - comparison with Flory-Huggins theory

The Flory-Huggins (F-H) theory, succinctly described before in subsection 1.2.4, has been used to predict the maximum API loading for a given API/polymer combination: first, the Hansen solubility parameters are calculated, and then the Gibbs free energy of mixing. Then, the combination of the API being tested with different polymers are represented in a graphic, where the x-axis contains the API loading and the y-axis contains the parameter $\Delta G/KT$. The maximum API loading for a given API/polymer combination as predicted by the F-H theory is the first inflexion point in the respective curve. It is, then, possible to create a rank of best polymers for the formulation of that API. However, this methodology has some limitations, since it was originally created for a mixture of two polymers [114]. The PLS models developed would have some advantages over this approach: to begin with, the models were developed specifically to predict the success of amorphous solid dispersions, based on observations composed of APIs and polymers. Second of all, the F-H theory is assessing only the miscibility of the two components, while the PLS models are taking into account more variables and, therefore, are probably evaluating more phenomenons than the F-H theory. Finally, the PLS model taking into account both API and polymer also allows the user to create a rank of best polymers; however, unlike the F-H theory, the PLS model taking into account only the API features allows the prediction of the maximum API loading before any assessment of suitable polymers, which would allow the user to exclude beforehand APIs that would not yield a good ASD, independently of the polymer.

To evaluate if the models developed are actually advantageous over the F-H approach, it was necessary to check if the predictions of the referred models were better than (or at least as good as) the F-H predictions. For that purpose, 31 API/Polymer combinations from 8 different internal projects (and, therefore, 8 different APIs) were run through both PLS models to obtain an API loading prediction, and these values were compared with the predictions obtained by the F-H theory, present in said reports. The results obtained are shown in Figure 3.25.

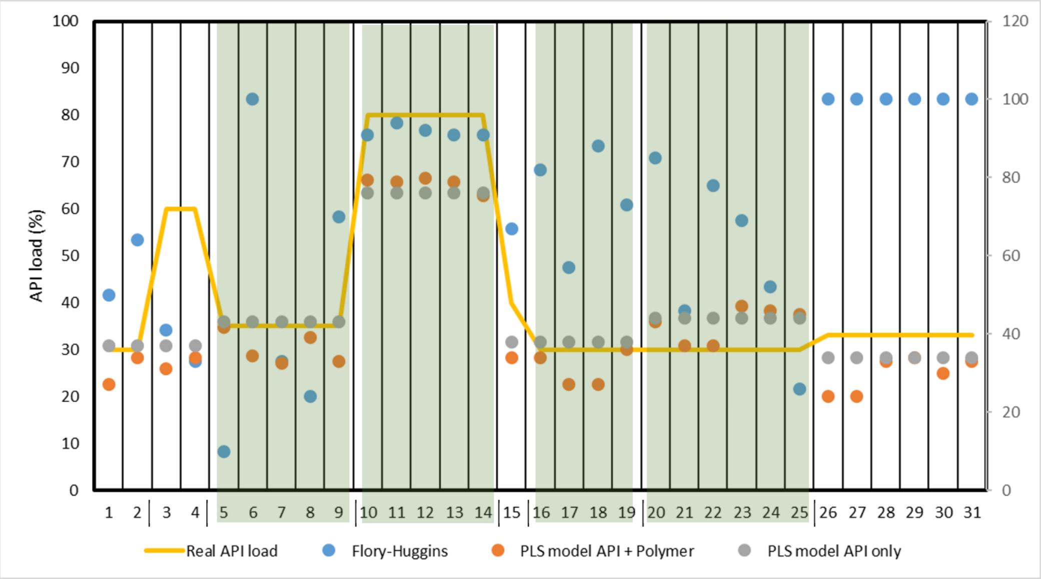

Figure 3.25: Graphical representation of the API load prediction for 31 observations from 8 different internal reports. Blue: F-H theory prediction; orange: PLS API and polymer features prediction; grey: PLS API features prediction; yellow: real API loading formulated. Vertical lines in x-axis: division between different projects. Observations shaded in green: reports that had been used in the training of the PLS models that, therefore, may be yielding biased predictions for these models.

As it is visible in Figure 3.25, most of the predictions made by the PLS models were more accurate than the predictions made by the F-H theory, both for reports that were used in the training set and for reports that weren't (and, therefore, these APIs were completely unknown for the PLS models). The only observations where the predictions yielded by the F-H theory were closer to the real API loading were observation 3 and observation 7. For all the other observations, the PLS models were better at predicting the real API loading. For the PLS model containing API and polymer variables, only observations 3, 4, 20, 23, 24 and 25 have a $\Delta RealAPILoading/Prediction$ larger than 10% (which translates into an accuracy of 81%). For the four projects that weren't part of the training set, for the first one (observations 1 and 2) both predictions were extremely close to the real API loading (less than 5% difference), and for the last one (observations 26-31) two of the predictions (observations 29 and 30) were extremely close to the real API loading (less than 5% difference) and two of the predictions (28 and 31) were exactly equal to the real API loading. As for the PLS containing uniquely API features, only observations 3-4 (project 2) and 20-25 (project 7) have a $\Delta RealAPILoading/Prediction$ larger than 10% (meaning 2 APIs amongst 8 – accuracy of 75%). For the four projects that weren't part of the training set, for project 5 (observation 15) the value predicted is extremely close to the real API value (2% difference), being even closer than the prediction made by the PLS model taking into account all features; for the last project (observations 26-31), the prediction is also extremely close to the real value (1% difference). These results provide a very high level of confidence to the model: not only are the predictions made by the PLS models much more accurate than the previously used computational tool, but also they are extremely accurate for the observations tested. Moreover, since these observations are from internal reports, the workflow to obtain them is known, and therefore it is known that the API loading formulated was indeed thoroughly studied and extended to the maximum, and therefore, this validation is internally more valued than a validation obtained from using external observations.

3.3.3 Using Spring and Parachute models to explore ”best polymer” output

One of the possible applications of the models developed to predict if a formulation will yield a good spring and parachute effect would be to use them to predict a ”best polymer” output: a scientist would create a dataset containing a given number of combinations of the API to be formulated with different polymers. The scientist would, then, run that dataset through the random forest or artificial neural network model and obtain the results. If the neural network model is used, the scientist will be able to construct a rank of polymers, since the model yields continuous values between 0 and 1 with probabilistic meaning. If the random forest classifier is used, it is not possible to create a quantitative or qualitative rank, but it is possible to predict beforehand which polymers would allow that API to yield a good spring and parachute effect with an acceptable accuracy. Since the model is more frequently predicting a positive output rather than a negative output (in practical terms, this means that it is more probable for the model to predict a bad ASD to be good, rather than to predict a good ASD to be bad), this means that it would be possible to exclude beforehand polymers that wouldn't be a good fit for that formulation, with a lower probability of excluding polymers that would, indeed, be advantageous for that ASD. Therefore, less polymers would need to be laboratorially experimented, allowing the scientist to wast less materials and less time. To be noted that, despite the fact that the ANN model provides a ranking of the polymers, it has been analyzed before in subsection 3.2.3 that the random forest classifier model seems to have a more adequate accuracy and, therefore, this should be the preferred model.

4

Conlusion and Future Prospects

Contents

4.1 Summary and final remarks

The starting point of this work was the assumption that currently, because the mechanisms behind an ASD formulation's success or lack thereof are still not completely understood, there was no widely accepted and generalizable method to predict computationally the maximum API loading for a given combination API/polymer. One used method, in literature and in industry, is the Flory-Huggins theory; however, this theory was developed for a mixture of two polymers and, therefore, erroneous results were expected for other mixtures. This was shown in section 3.3.2 – values obtained from this approach are often very different from the real values that end up being formulated. Therefore, the development of models to predict with enough accuracy the maximum API loading for a given ASD formulation would be a big competitive advantage, and to understand which variables are contributing to that factor is very appealing, industrially and academically. Another relevant aspect in the behaviour of ASD formulations is the spring and parachute effect, as explained before. Since it is usually key for the formulation to present this behaviour upon dissolution so that the supersaturation is maintained long enough for the drug to be absorbed, being able to predict if a given API/polymer combination will yield a spring and parachute effect is also very useful industrially, and it is very interesting to understand which variables are contributing to this. With this in mind, the initial objective was to develop models that could predict the maximum API loading and the spring and parachute effect for a given API/polymer combination.

To sum up, and in order to reach these objectives, the first step was to develop a curated database containing all the information needed to train the models. This database was constructed based on information from internal reports and from external reports. This was essential to organize the information in order to analyze and use it, and will be useful for future projects, since many API and polymer parameters were calculated or harvested from online databases and, therefore, the effort required to obtain them in the future will be much smaller.

Then, three models were developed to predict the maximum API loading. The first one was a PLS model taking into account API and polymer features (as well as interaction variables), with an R^2 of 49% and a Q^2 of 33%. When a cutoff threshold of 10% difference from the real API load was defined, this model yielded an accuracy of 45% in an external validation set, an accuracy of 71% for a set of commercialized ASDs and an accuracy of 81% for a set of internally developed ASDs. These results are very good – these accuracies allow a scientist to begin the experimental testing very near to the final, real API loading, saving a lot of time, resources and money. To be noted that the external validation set, which is the dataset with lower accuracy, is also the dataset with lower confidence in the observations – many authors decide upon a given API loading without effectively studying which would be the maximum. In the set of commercial ASDs, with the second highest accuracy, this is less of a problem – since the ASD was commercialized, it was probably thoroughly studied and tested. In the set of internal projects (with the highest accuracy), since the reports are internally available, it is known that the API loading was actually extended to the maximum. These accuracies with this cutoff provide a very high level of confidence to the models, and could even allow them to be used to predict the API loading in a more automated way, without the need of being experimentally validated by a scientist.

A second PLS model was developed to predict this output, taking into account solely the API features – this model yielded an R^2 of 39% and a Q^2 of 31%. When a cutoff threshold of 10% difference from the real API loading was defined, this model yielded an accuracy of 50% in an external validation set, an accuracy of 57% for a set of commercialized ASDs and an accuracy of 75% for a set of internally developed ASDs. However, when the cutoff is changed to 15% instead of 10%, the accuracies in the external validation set and commercial ASDs set go up to, respectively, 71% and 70%. This model would be used in a preliminary environment – when a scientist receives an API to formulate, they would run the API through the model previously to thinking about which polymers would be suitable, and they could begin the experimental testing in a range close to the real maximum for that formulation. Therefore, a cutoff of 15% is perfectly acceptable, since the value would be experimentally modulated.

Using observations from internal reports, both PLS models were compared with the F-H theory, which is the current computational method used for the preliminary prediction of the API loading for a given API/polymer combination. Both models performed much better than the F-H theory, and most of the time they yielded values very close to the real formulated API loading. Therefore, it would be a major advantage to resort to these models instead of the F-H theory.

In all datasets, the best performance of the PLS models is for more moderate values of API loadings, probably due to the fact that the data isn't equally distributed, being more concentrated in the 21%-50% range. However, most ASDs are formulated with API loadings similar to this and not with extreme values, so this particularity isn't too limiting.

Still for this output, a random forest classifier model was developed, with an accuracy of 78.8%. However, this model performed poorly in all the other datasets and was, therefore, excluded. This may have been due to lack of data – a non-linear algorithm such as a random forest usually require a large dataset [112], and for this output only 66 observations were available (while for the spring and parachute model, 92 observations were available, which may be enough to make a difference in terms of predictive power). Furthermore, this model is designed to assign the data into five classes, while the spring and parachute model has a binary output – this, coupled to the fact that the API loading model has a significantly smaller dataset compared to the spring and parachute effect model, results in a much lower observations-to-classes ratio in the first model, which may explain why the random forest classifier did not work for this output.

The next step was to explore the spring and parachute output, for which two models were developed. The random forest classifier developed for this output yielded an accuracy of 78.3%. For the external validation datasets, the accuracy obtained was 67%. Then, an artificial neural network model was developed. This model had an overall accuracy of 88%, composed by an accuracy of 93.3% for the training set, an accuracy of 85.7% for the validation set and an accuracy of 72.2% for the test set. However, in the external validation set, the accuracy was 57% – since a binary output is being predicted, an accuracy so close to 50% (and, therefore, to random prediction) is not significant. Furthermore, since the random forest model can be easily optimized through feature selection and, therefore, is using less variables than the neural network model, it is less time consuming to harvest the necessary data for this model. Therefore, the random forest model was chosen as the preferential one for the prediction of the spring and parachute effect. The accuracies for these models aren't ideal and, therefore, they should simply be used in a preliminary screening. However, since they are more often predicting positive than negative values, it is less likely for the model to predict a formulation that would be good as "bad" and, therefore, to lead the scientist to miss a good formulation, so they are still promising. The poor performance in the external validation set by the ANN model may be due to various factors. First of all, as explained in section 3.2.2, the division between training, testing and validation data was done randomly, and therefore there may be repeated APIs in these three datasets. Furthermore, as explained before, the tree bagging in the random forest classifier highly reduces its proneness to overfitting – the ANN algorithm does not have a similar phenomenon and, therefore, may be subject to data overfitting. Lastly, the ANN model does not allow one to know which variables are contributing more or less to the assignment of the different observations into the classes, so all the features were kept; as mentioned before, keeping variables that are mere noise highly contributes to overfitting. Taking all this into account, it is not surprising that the random forest classifier performed better overall.

4.2 How can these models be used? Development of new workflow for scientists

When a scientist receives a new API to formulate, they may want to assess if the API is suitable to be formulated as an ASD, and if so, what conditions would yield the best performance. The developed models can facilitate and accelerate this process greatly.

The models developed were incorporated into a new workflow developed for formulation scientists to assess the feasibility and probably success of a new ASD, presented in the Confidential Appendix, section A.3.

These models provide a new promising method to greatly accelerate the initial process of ASD screening by predicting *in silico* two very important aspects of an ASD: if it will behave according to the spring and parachute effect in order to maintain the supersaturated condition long enough to be absorbed, and what is the maximum possible ratio of API to polymer (API loading). The experimental testing can be highly reduced, since ASDs unlikely to succeed can be excluded *a priori*, and formulations that go on to the next step of the screening can be formulated with an API loading close to the real maximum loading from the beginning, requiring less adjustments and rectifications.

4.3 Future work and applications

There are several possibilities to ameliorate the models' performances. To begin with, a simple step would be to add more observations to the datasets – this would largely help in the prevention of overfitting and, since these are complex algorithms, would overall aid the process of establishing relationships between observations and outputs. Related to the harvesting of more data, it would also be advisable to try and balance the dataset to prevent data skew – this step was already attempted, but both in internal reports and in external literature papers, the API loadings found had a tendency to fall in the same, more moderate range. It would also be possible to develop new models based on these ones that explored the addition of other excipients, such as surfactants, and mixtures of polymers instead of simply ASDs with a single polymer. Developing models taking into account *in vivo* data would also be a possibility; while this would have the advantage of taking into account factors such as permeability and absorption, *in vivo* data has a much higher variability and, therefore, it could make the development of the models a much harder task.

It would also be possible to include this approach to ASD screening in an automatized, high throughput screening strategy. If this were to be implemented, the polymers predicted to be good could automatically move on to the next step of the screening and the resulting prototypes could be formulated in a high-throughput manner with API loadings similar to the one obtained by the PLS model for those API/polymer combinations (this model presented an accuracy of approximately 70-80% for a confidence interval of 10%, so the API loadings formulated could be in the 10% range, and a broader range would be explored if the result was unsatisfying). This approach would allow to use the models without need of human experimentation, highly accelerating the process of ASD screening, while maintaining or ameliorating the confidence in the *in-silico* results obtained.

5

Supplementary Material

Contents

5.1 Supplementary Material 1 – Partial Least Squares Models

5.1.1 PLS model - API + Polymer + Interaction Variables

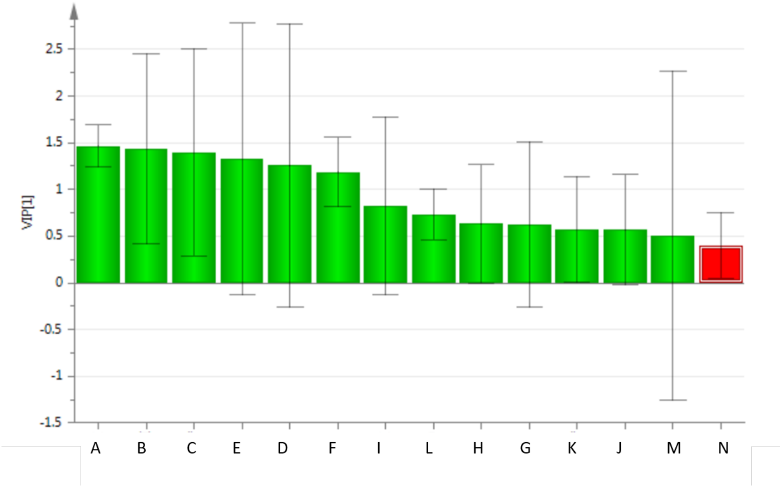

Figure 5.1: VIP plot of the next step of the model's optimization. The variable in red was followingly excluded from the model. The correspondence between the letters and the represented variables are present in Table A.1 (Confidential Appendix).

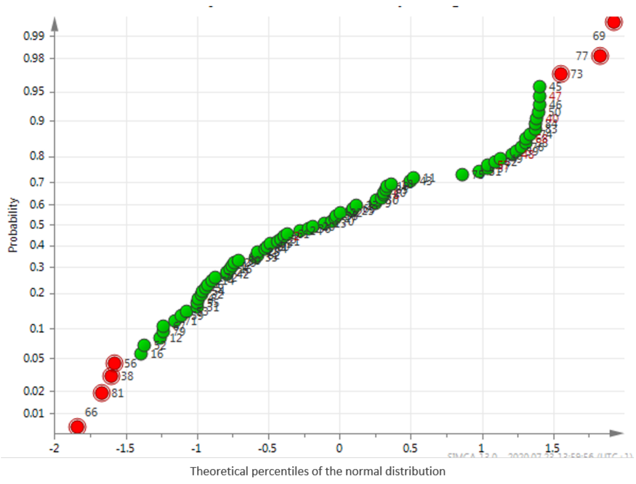

Figure 5.2: Normal probability plot of residuals, which displays the residuals standardized - raw residual divided by the residual standard deviation (values in a double log scale). A dataset with no outliers would have an approximately linear distribution, as mentioned in the methodology section. Observations with more than 1.5 or less than -1.5 standard deviations were excluded (observations 38, 56, 66, 69. 73, 77 and 81).

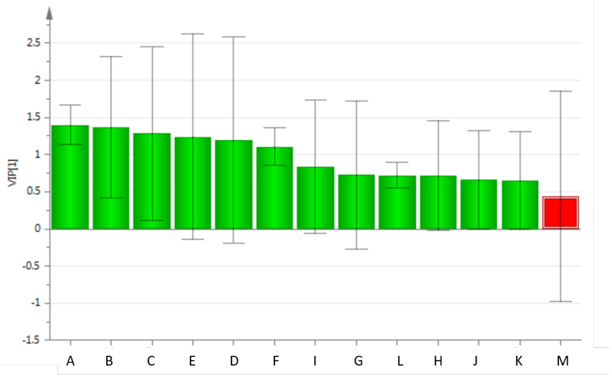

Figure 5.3: VIP plot of the last step of the model's optimization. The variable in red was followingly excluded from the model. The correspondence between the letters and the represented variables are present in Table A.1 (Confidential Appendix).

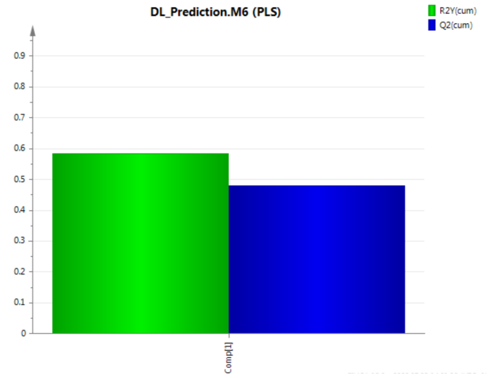

Figure 5.4: Representation of the values of R^2 (green) and Q^2 (blue) for the first and only principal component obtained for the final version of the PLS model taking into account all the features.

5.1.2 PLS model - API Variables Only

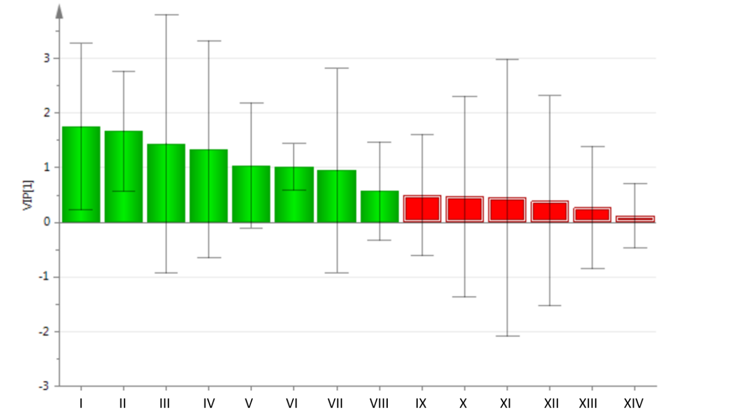

Figure 5.5: VIP plot of the unoptimized model. The variables in red were followingly excluded from the model. The correspondence between the letters and the represented variables are present in Table A.2 (Confidential Appendix).

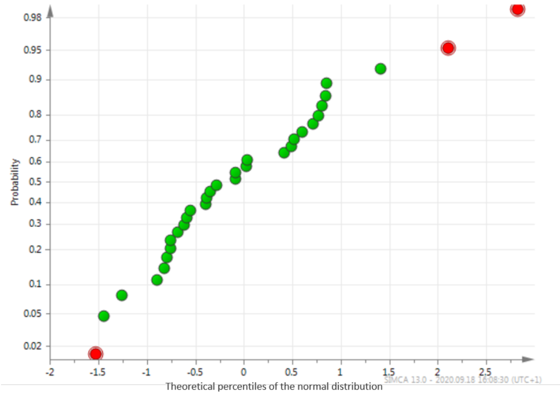

Figure 5.6: Normal probability plot of residuals, which displays the residuals standardized - raw residual divided by the residual standard deviation (values in a double log scale). A dataset with no outliers would have an approximately linear distribution, as mentioned in the methodology section. Observations with a standard deviation of more than 1.5 or less than -1.5 standard deviations were excluded.

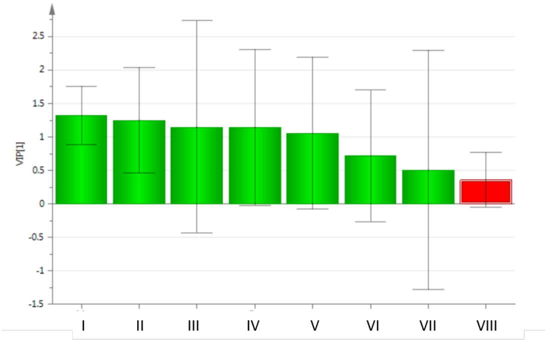

Figure 5.7: VIP plot of the model being optimized. The variable in red were followingly excluded from the model. The correspondence between the letters and the represented variables are present in Table A.2 (Confidential Appendix).

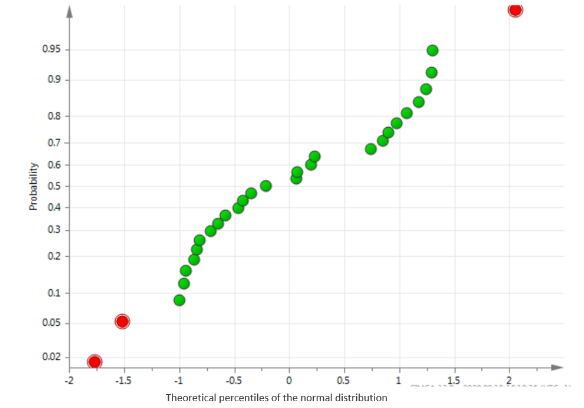

Figure 5.8: Normal probability plot of residuals, which displays the residuals standardized - raw residual divided by the residual standard deviation (values in a double log scale). A dataset with no outliers would have an approximately linear distribution, as mentioned in the methodology section. Observations with a standard deviation of more than 1.5 or less than -1.5 standard deviations were excluded.

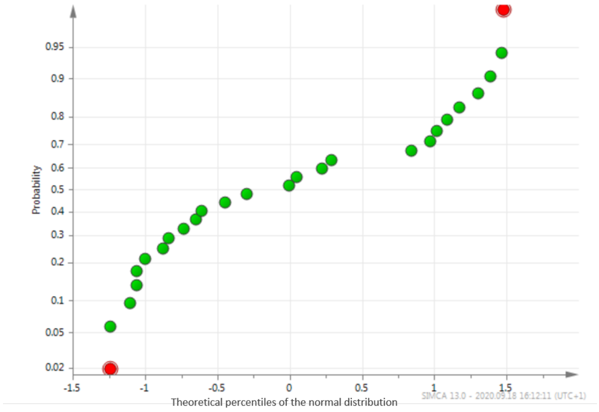

Figure 5.9: Normal probability plot of residuals, which displays the residuals standardized - raw residual divided by the residual standard deviation (values in a double log scale). A dataset with no outliers would have an approximately linear distribution, as mentioned in the methodology section. Observations causing tails in the distribution were excluded.

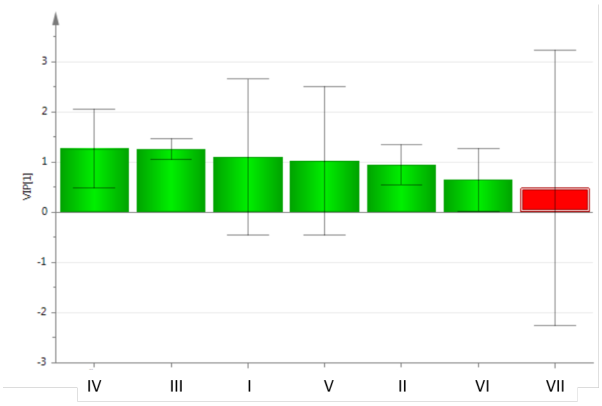

Figure 5.10: VIP plot of the final step of the model's optimization. The variable in red were followingly excluded from the model. The correspondence between the letters and the represented variables are present in Table A.2 (Confidential Appendix).

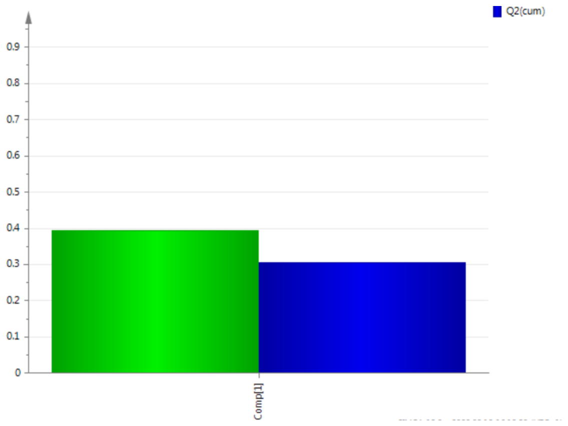

Figure 5.11: Representation of the values of R^2 (green) and Q^2 (blue) for the first and only principal component obtained for the final version of the PLS model taking into account the API features only.

5.2 Supplementary Material 2 – Random Forest Model (API Loading Output)

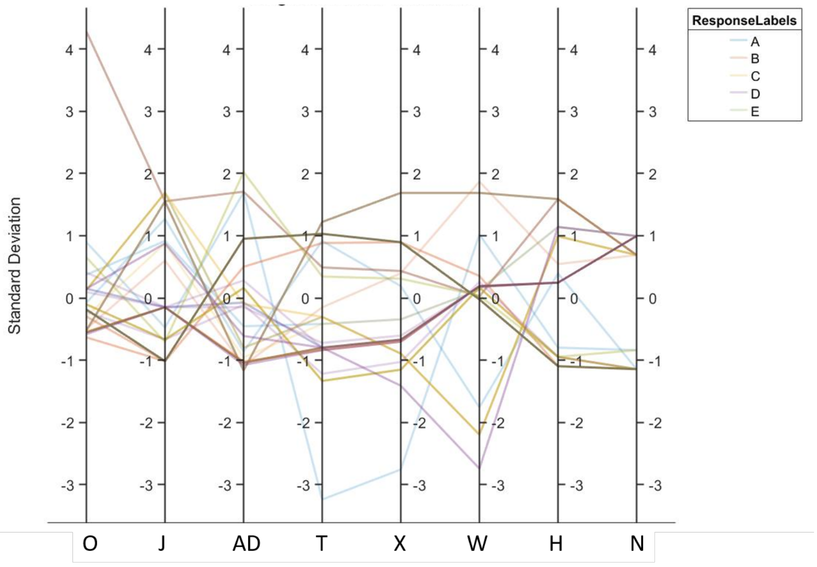

Figure 5.12: Random forest classifier model: parallel coordinates plot for the first eight features. The color code is represented in the graphic (correspondence between class letters and API loading ranges can be found on Table 2.2). The correspondence between the plots' labels and the variables is presented in Table A.5 (Confidential Appendix).

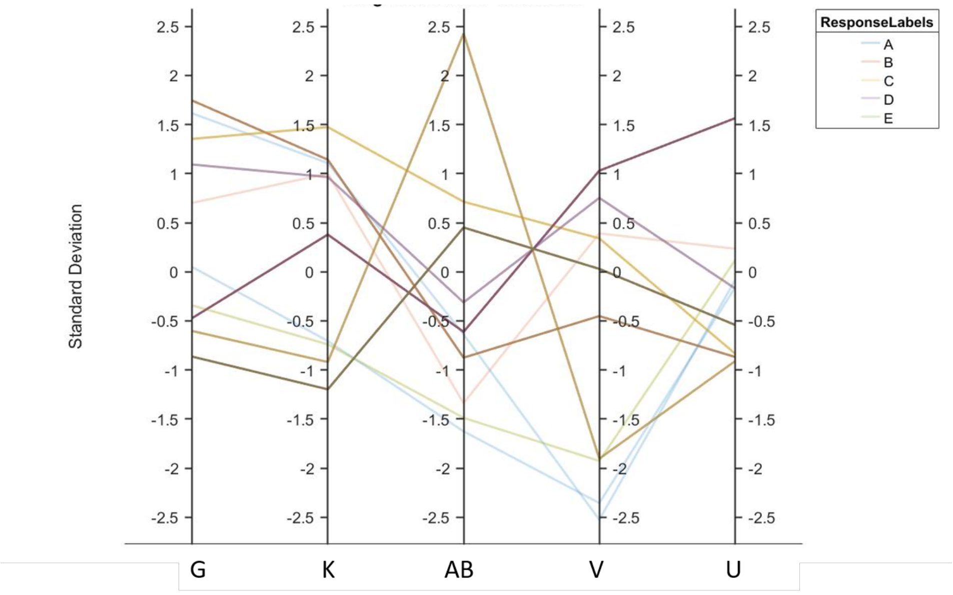

Figure 5.13: Random forest classifier model: parallel coordinates plot for the next five features. The color code is represented in the graphic (correspondence between class letters and API loading ranges can be found on Table 2.2). The correspondence between the plots' labels and the variables is presented in Table A.5 (Confidential Appendix).

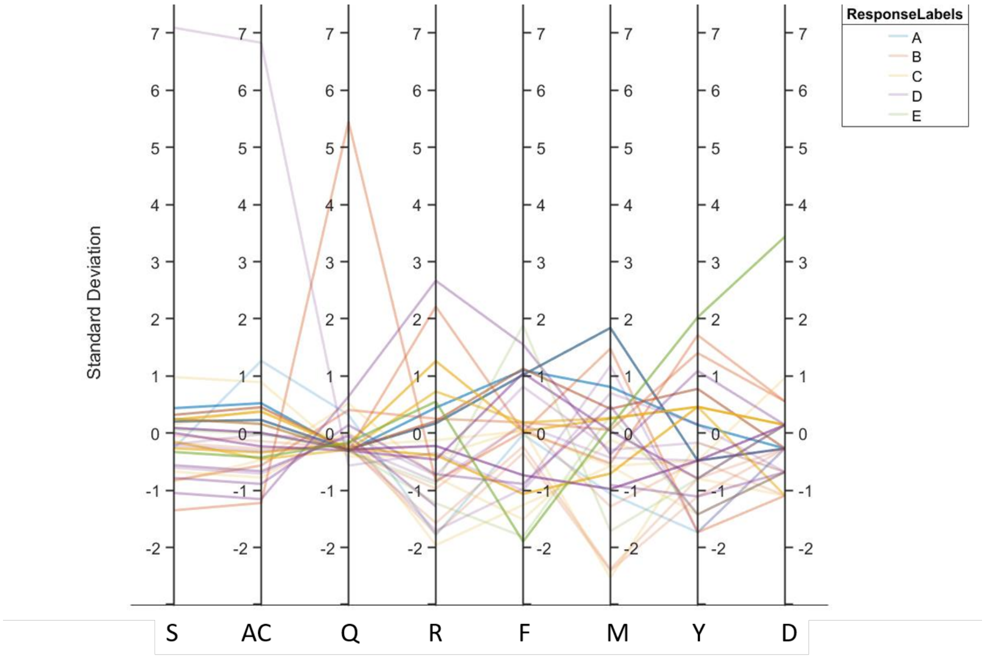

Figure 5.14: Random forest classifier model: parallel coordinates plot for the next eight features. The color code is represented in the graphic (correspondence between class letters and API loading ranges can be found on Table 2.2). The correspondence between the plots' labels and the variables is presented in Table A.5 (Confidential Appendix).

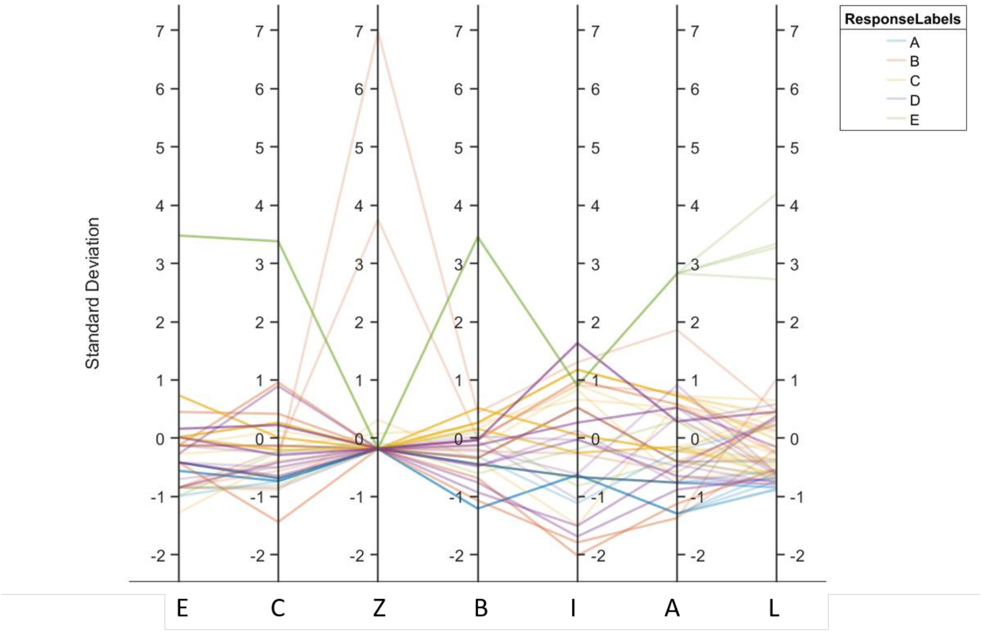

Figure 5.15: Random forest classifier model: parallel coordinates plot for the last seven features. The color code is represented in the graphic (correspondence between class letters and API loading ranges can be found on Table 2.2). The correspondence between the plots' labels and the variables is presented in Table A.5 (Confidential Appendix).

5.3 Supplementary Material 3 – Random Forest Model (Spring and Parachute Effect Output)

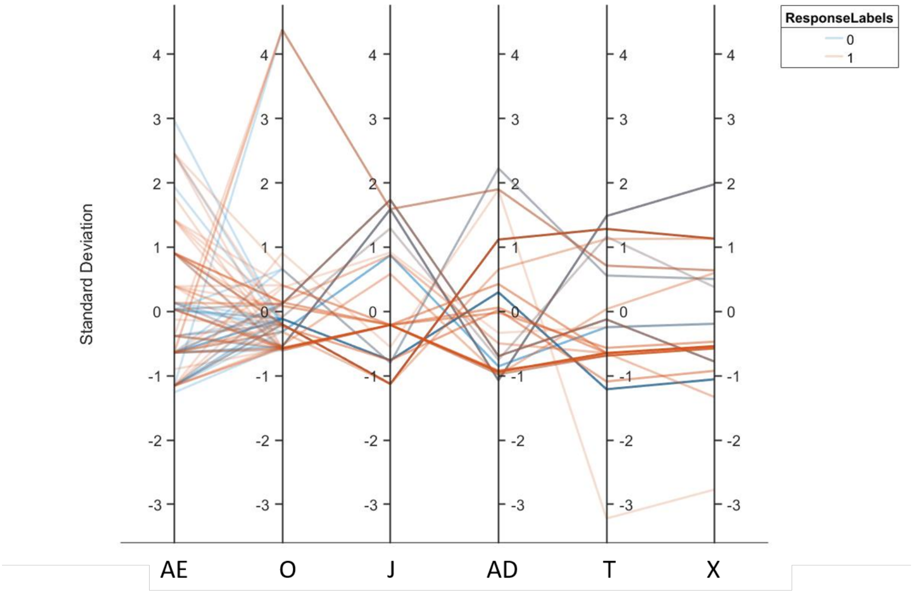

Figure 5.16: Random forest classifier model: parallel coordinates plot for the first six features. The color code is represented in the graphic (0 – bad spring and parachute effect; 1 – good spring and parachute effect). The correspondence between the plots' labels and the variables is presented in Table A.7 (Confidential Appendix).

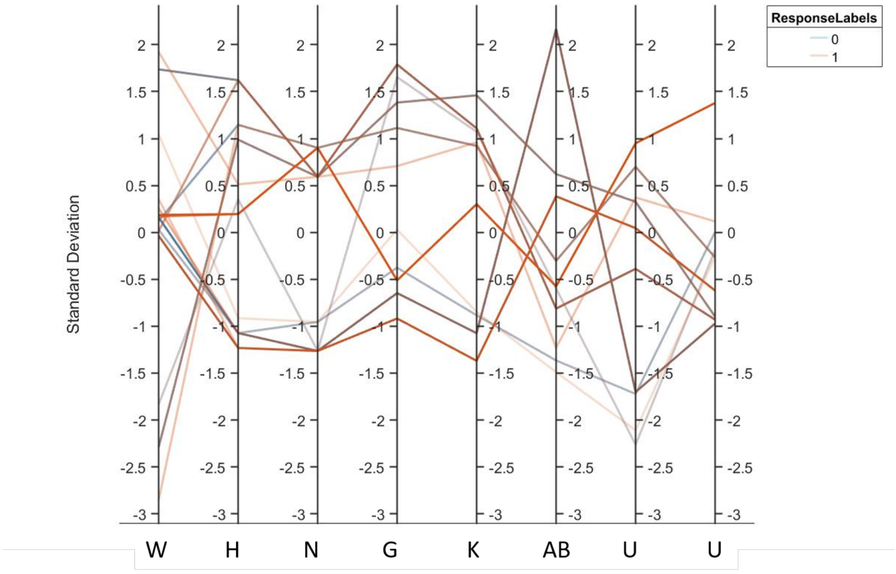

Figure 5.17: Random forest classifier model: parallel coordinates plot for the next eight features. The color code is represented in the graphic (0 – bad spring and parachute effect; 1 – good spring and parachute effect). The correspondence between the plots' labels and the variables is presented in Table A.7 (Confidential Appendix).

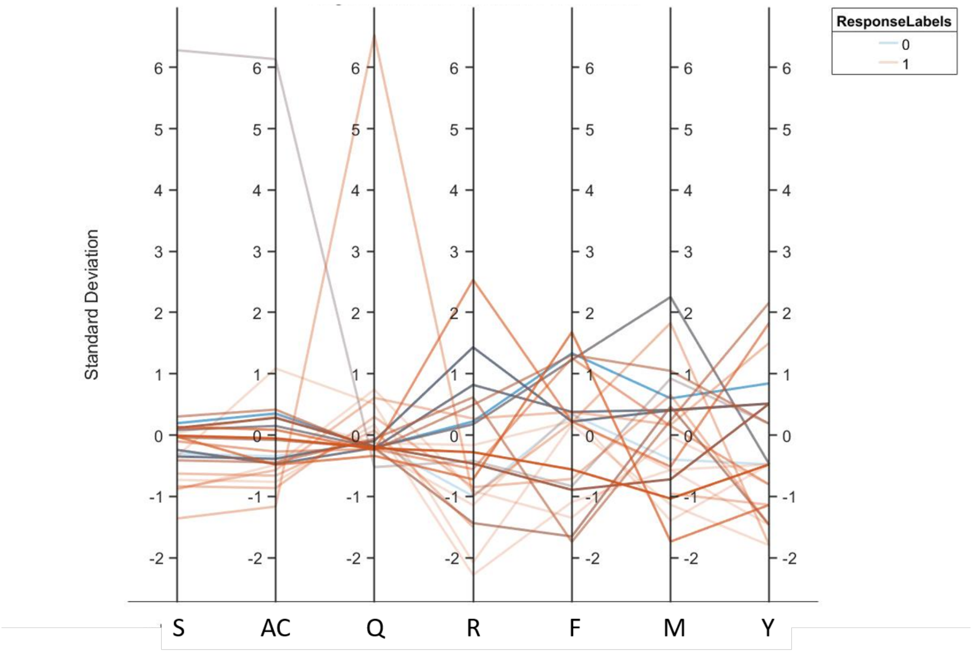

Figure 5.18: Random forest classifier model: parallel coordinates plot for the next seven features. The color code is represented in the graphic (0 – bad spring and parachute effect; 1 – good spring and parachute effect). The correspondence between the plots' labels and the variables is presented in Table A.7 (Confidential Appendix).

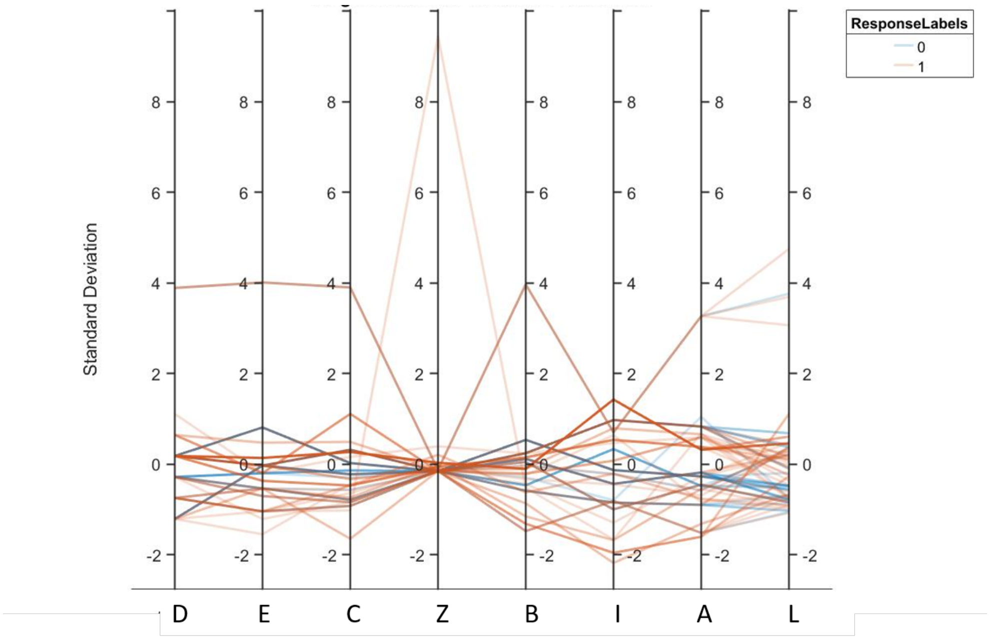

Figure 5.19: Random forest classifier model: parallel coordinates plot for the last eight features. The color code is represented in the graphic (0 – bad spring and parachute effect; 1 – good spring and parachute effect). The correspondence between the plots' labels and the variables is presented in Table A.7 (Confidential Appendix).

Bibliography

[1] J. A. Baird, B. Van Eerdenbrugh, and L. S. Taylor, “A classification system to assess the crystallization tendency of organic molecules from undercooled melts,” *Journal of pharmaceutical sciences*, vol. 99, no. 9, pp. 3787–3806, 2010.

[2] Y. S. Krishnaiah, “Pharmaceutical technologies for enhancing oral bioavailability of poorly soluble drugs,” *J Bioequiv Availab*, vol. 2, no. 2, pp. 28–36, 2010.

[3] D. J. Price, F. Ditzinger, N. J. Koehl, S. Jankovic, G. Tsakiridou, A. Nair, R. Holm, M. Kuentz, J. B. Dressman, and C. Saal, “Approaches to increase mechanistic understanding and aid in the selection of precipitation inhibitors for supersaturating formulations–a peer review,” *Journal of Pharmacy and Pharmacology*, vol. 71, no. 4, pp. 483–509, 2019.

[4] A. Systems, “Biopharmaceutical classification system,” accessed 20-Feb-2020. [Online]. Available: https://www.absorption.com/kc/biopharmaceutics-classification-system-bcs//

[5] R. G. Ricarte, N. J. Van Zee, Z. Li, L. M. Johnson, T. P. Lodge, and M. A. Hillmyer, “Recent advances in understanding the micro-and nanoscale phenomena of amorphous solid dispersions,” *Molecular pharmaceutics*, vol. 16, no. 10, pp. 4089–4103, 2019.

[6] C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney, “Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings,” *Advanced drug delivery reviews*, vol. 23, no. 1-3, pp. 3–25, 1997.

[7] T. Vasconcelos, S. Marques, J. das Neves, and B. Sarmento, “Amorphous solid dispersions: Rational selection of a manufacturing process,” *Advanced drug delivery reviews*, vol. 100, pp. 85–101, 2016.

[8] S. Janssens and G. Van den Mooter, “Physical chemistry of solid dispersions,” *Journal of Pharmacy and Pharmacology*, vol. 61, no. 12, pp. 1571–1586, 2009.

[9] A. A. Noyes and W. R. Whitney, “The rate of solution of solid substances in their own solutions.” *Journal of the American Chemical Society*, vol. 19, no. 12, pp. 930–934, 1897.

[10] S. Greco, J.-R. Authelin, C. Leveder, and A. Segalini, “A practical method to predict physical stability of amorphous solid dispersions,” *Pharmaceutical research*, vol. 29, no. 10, pp. 2792–2805, 2012.

[11] B. Vig and M. Morgen, “Formulation, process development, and scale-up: Spray-drying amorphous solid dispersions for insoluble drugs,” pp. 793–820, 2017.

[12] D. Zhou, E. A. Schmitt, D. Law, P. J. Brackemeyer, and G. G. Zhang, “Assessing physical stability risk using the amorphous classification system (acs) based on simple thermal analysis,” *Molecular pharmaceutics*, vol. 16, no. 6, pp. 2742–2754, 2019.

[13] P. Lino and J. Henriques, “Amorphous solid dispersions - increasing solubility from api to tablets,” pp. 32–37, April 2019, accessed 26-Feb-2020. [Online]. Available: https://drug-dev.com/amorphous-solid-dispersions-increasing-solubility-from-api-to-tablets/

[14] A. M. Agrawal, M. S. Dudhedia, and E. Zimny, “Hot melt extrusion: development of an amorphous solid dispersion for an insoluble drug from mini-scale to clinical scale,” *AAPS PharmSciTech*, vol. 17, no. 1, pp. 133–147, 2016.

[15] D. T. Friesen, R. Shanker, M. Crew, D. T. Smithey, W. Curatolo, and J. Nightingale, “Hydroxypropyl methylcellulose acetate succinate-based spray-dried dispersions: an overview,” *Molecular pharmaceutics*, vol. 5, no. 6, pp. 1003–1019, 2008.

[16] A. Singh and G. Van den Mooter, “Spray drying formulation of amorphous solid dispersions,” *Advanced drug delivery reviews*, vol. 100, pp. 27–50, 2016.

[17] A. Paudel, Z. A. Worku, J. Meeus, S. Guns, and G. Van den Mooter, “Manufacturing of solid dispersions of poorly water soluble drugs by spray drying: formulation and process considerations,” *International journal of pharmaceutics*, vol. 453, no. 1, pp. 253–284, 2013.

[18] S. Emami, M. Siahi-Shadbad, K. Adibkia, and M. Barzegar-Jalali, “Recent advances in improving oral drug bioavailability by cocrystals,” *BioImpacts: BI*, vol. 8, no. 4, p. 305, 2018.

[19] X. C. Tang, M. J. Pikal, and L. S. Taylor, “The effect of temperature on hydrogen bonding in crystalline and amorphous phases in dihydropyrine calcium channel blockers,” *Pharmaceutical research*, vol. 19, no. 4, pp. 484–490, 2002.

[20] N. S. Trasi, J. A. Baird, U. S. Kestur, and L. S. Taylor, “Factors influencing crystal growth rates from undercooled liquids of pharmaceutical compounds,” *The Journal of Physical Chemistry B*, vol. 118, no. 33, pp. 9974–9982, 2014.

[21] B. Van Eerdenbrugh, J. A. Baird, and L. S. Taylor, “Crystallization tendency of active pharmaceutical ingredients following rapid solvent evaporation—classification and comparison with crystallization tendency from under cooled melts,” *Journal of pharmaceutical sciences*, vol. 99, no. 9, pp. 3826–3838, 2010.

[22] A. N. Ghebremeskel, C. Vemavarapu, and M. Lodaya, “Use of surfactants as plasticizers in preparing solid dispersions of poorly soluble api: selection of polymer–surfactant combinations using solubility parameters and testing the processability,” *International journal of pharmaceutics*, vol. 328, no. 2, pp. 119–129, 2007.

[23] C. S. Towler, T. Li, H. Wikstro¨m, D. M. Remick, M. V. Sanchez-Felix, and L. S. Taylor, “An investigation into the influence of counterion on the properties of some amorphous organic salts,” *Molecular pharmaceutics*, vol. 5, no. 6, pp. 946–955, 2008.

[24] V. M. Sonje, L. Kumar, V. Puri, G. Kohli, A. M. Kaushal, and A. K. Bansal, “Effect of counterions on the properties of amorphous atorvastatin salts,” *European journal of pharmaceutical sciences*, vol. 44, no. 4, pp. 462–470, 2011.

[25] J. Li, J. Zhao, L. Tao, J. Wang, V. Waknis, D. Pan, M. Hubert, K. Raghavan, and J. Patel, “The effect of polymeric excipients on the physical properties and performance of amorphous dispersions: part i, free volume and glass transition,” *Pharmaceutical research*, vol. 32, no. 2, pp. 500–515, 2015.

[26] F. Qian, J. Huang, and M. A. Hussain, “Drug–polymer solubility and miscibility: stability consideration and practical challenges in amorphous solid dispersion development,” *Journal of pharmaceutical sciences*, vol. 99, no. 7, pp. 2941–2947, 2010.

[27] P. J. Flory, “Thermodynamics of high polymer solutions,” *The Journal of Chemical Physics*, vol. 9, no. 8, pp. 660–660, 1941.

[28] M. L. Huggins, “Thermodynamic properties of solutions of long-chain compounds,” *Annals of the New York Academy of Sciences*, vol. 43, no. 1, pp. 1–32, 1942.

[29] P. Gurikov, I. Lebedev, A. Kolnoochenko, and N. Menshutina, “Prediction of the solubility in supercritical carbon dioxide: a hybrid thermodynamic/qspr approach,” in *Computer Aided Chemical Engineering.* Elsevier, 2016, vol. 38, pp. 1587–1592.

[30] P. Gupta and A. Bansal, “Ternary amorphous composites of celecoxib, poly (vinyl pyrrolidone) and meglumine with enhanced solubility,” *Die Pharmazie-An International Journal of Pharmaceutical Sciences*, vol. 60, no. 11, pp. 830–836, 2005.

[31] P. Gupta and A. K. Bansal, "Spray drying for generation of a ternary amorphous system of celecoxib, pvp, and meglumine," *Pharmaceutical development and technology*, vol. 10, no. 2, pp. 273–281, 2005.

[32] M. M. Leane, W. Sinclair, F. Qian, R. Haddadin, A. Brown, M. Tobyn, and A. B. Dennis, "Formulation and process design for a solid dosage form containing a spray-dried amorphous dispersion of ibipinabant," *Pharmaceutical development and technology*, vol. 18, no. 2, pp. 359–366, 2013.

[33] L. S. Koester, P. Mayorga, and V. L. Bassani, "Carbamazepine/βcd/hpmc solid dispersions. i. influence of the spray-drying process and βcd/hpmc on the drug dissolution profile," *Drug development and industrial pharmacy*, vol. 29, no. 2, pp. 139–144, 2003.

[34] V. B. Pokharkar, L. P. Mandpe, M. N. Padamwar, A. A. Ambike, K. R. Mahadik, and A. Paradkar, "Development, characterization and stabilization of amorphous form of a low tg drug," *Powder technology*, vol. 167, no. 1, pp. 20–25, 2006.

[35] H. Takeuchi, S. Nagira, H. Yamamoto, and Y. Kawashima, "Solid dispersion particles of amorphous indomethacin with fine porous silica particles by using spray-drying method," *International journal of pharmaceutics*, vol. 293, no. 1-2, pp. 155–164, 2005.

[36] Y.-D. Yan, J. H. Sung, K. K. Kim, D. W. Kim, J. O. Kim, B.-J. Lee, C. S. Yong, and H.-G. Choi, "Novel valsartan-loaded solid dispersion with enhanced bioavailability and no crystalline changes," *International journal of pharmaceutics*, vol. 422, no. 1-2, pp. 202–210, 2012.

[37] W. H. Organization, "Target product profile," accessed 24-Feb-2020. [Online]. Available: https://www.who.int/research-observatory/analyses/tpp/en/

[38] B. D. Anderson, "Predicting solubility/miscibility in amorphous dispersions: it is time to move beyond regular solution theories," *Journal of Pharmaceutical Sciences*, vol. 107, no. 1, pp. 24–33, 2018.

[39] J. H. Hildebrand and R. L. Scott, *Regular solutions*. Prentice-Hall, 1962.

[40] A. F. Barton, *CRC handbook of solubility parameters and other cohesion parameters*. CRC press, 1991.

[41] C. Hansen, "Three dimensional solubility parameter and solvent diffusion coefficient. importance in surface coating formulation," *Doctoral Dissertation*, 1967.

[42] C. M. Hansen, "The universality of the solubility parameter," *Industrial & engineering chemistry product research and development*, vol. 8, no. 1, pp. 2–11, 1969.

[43] T. official site of Hansen Solubility Parameters and H. software, “The famous factor of 4 - dr hansen’s view,” accessed 15-Sept-2020. [Online]. Available: https://www.hansen-solubility.com/HSP-science/4factor.php

[44] J. B. Dressman, G. L. Amidon, C. Reppas, and V. P. Shah, “Dissolution testing as a prognostic tool for oral drug absorption: immediate release dosage forms,” *Pharmaceutical research*, vol. 15, no. 1, pp. 11–22, 1998.

[45] R. Hamed, A. Awadallah, S. Sunoqrot, O. Tarawneh, S. Nazzal, T. AlBaraghthi, J. Al Sayyad, and A. Abbas, “ph-dependent solubility and dissolution behavior of carvedilol—case example of a weakly basic bcs class ii drug,” *AAPS PharmSciTech*, vol. 17, no. 2, pp. 418–426, 2016.

[46] J. P. Lakshman, Y. Cao, J. Kowalski, and A. T. Serajuddin, “Application of melt extrusion in the development of a physically and chemically stable high-energy amorphous solid dispersion of a poorly water-soluble drug,” *Molecular pharmaceutics*, vol. 5, no. 6, pp. 994–1002, 2008.

[47] L. I. Mosquera-Giraldo, N. S. Trasi, and L. S. Taylor, “Impact of surfactants on the crystal growth of amorphous celecoxib,” *International journal of pharmaceutics*, vol. 461, no. 1-2, pp. 251–257, 2014.

[48] K. J. Frank, U. Westedt, K. M. Rosenblatt, P. Hölig, J. Rosenberg, M. Mägerlein, G. Fricker, and M. Brandl, “The amorphous solid dispersion of the poorly soluble abt-102 forms nano/microparticulate structures in aqueous medium: impact on solubility,” *International journal of nanomedicine*, vol. 7, p. 5757, 2012.

[49] N. G. Solanki, K. Lam, M. Tahsin, S. G. Gumaste, A. V. Shah, and A. T. Serajuddin, “Effects of surfactants on itraconazole-hpmcas solid dispersion prepared by hot-melt extrusion i: miscibility and drug release,” *Journal of pharmaceutical sciences*, vol. 108, no. 4, pp. 1453–1465, 2019.

[50] R. S. of Chemistry, “Solvent casting method,” accessed 26-Feb-2020. [Online]. Available: https://www.rsc.org/publishing/journals/prospect/ontology.asp?id=CMO:0002204&MSID=C0

[51] N. Shah, H. Sandhu, D. S. Choi, H. Chokshi, and A. W. Malick, “Amorphous solid dispersions,” *Theory and Practice; Springer: Berlin, Germany*, p. 180.

[52] B. Démuth, Z. K. Nagy, A. Balogh, T. Vigh, G. Marosi, G. Verreck, I. Van Assche, and M. Brewster, “Downstream processing of polymer-based amorphous solid dispersions to generate tablet formulations,” *International journal of pharmaceutics*, vol. 486, no. 1-2, pp. 268–286, 2015.

[53] R. Han, Y. Yang, X. Li, and D. Ouyang, “Predicting oral disintegrating tablet formulations by neural network techniques,” *Asian journal of pharmaceutical sciences*, vol. 13, no. 4, pp. 336–342, 2018.

[54] R. Han, H. Xiong, Z. Ye, Y. Yang, T. Huang, Q. Jing, J. Lu, H. Pan, F. Ren, and D. Ouyang, "Predicting physical stability of solid dispersions by machine learning techniques," *Journal of Controlled Release*, vol. 311, pp. 16–25, 2019.

[55] I.-H. Beak and M.-S. Kim, "Improved supersaturation and oral absorption of dutasteride by amorphous solid dispersions," *Chemical and Pharmaceutical Bulletin*, vol. 60, no. 11, pp. 1468–1473, 2012.

[56] T. Kai, Y. AKIYAMA, S. NOMURA, and M. SATo, "Oral absorption improvement of poorly soluble drug using solid dispersion technique," *Chemical and pharmaceutical bulletin*, vol. 44, no. 3, pp. 568–571, 1996.

[57] M. Kennedy, J. Hu, P. Gao, L. Li, A. Ali-Reynolds, B. Chal, V. Gupta, C. Ma, N. Mahajan, A. Akrami *et al.*, "Enhanced bioavailability of a poorly soluble vr1 antagonist using an amorphous solid dispersion approach: a case study," *Molecular pharmaceutics*, vol. 5, no. 6, pp. 981–993, 2008.

[58] G. F. Palmieri, F. Cantalamessa, P. Di Martino, C. Nasuti, and S. Martelli, "Lonidamine solid dispersions: in vitro and in vivo evaluation," *Drug development and industrial pharmacy*, vol. 28, no. 10, pp. 1241–1250, 2002.

[59] P.-C. Chiang, Y. Cui, Y. Ran, J. Lubach, K.-J. Chou, L. Bao, W. Jia, H. La, J. Hau, A. Sambrone *et al.*, "In vitro and in vivo evaluation of amorphous solid dispersions generated by different bench-scale processes, using griseofulvin as a model compound," *The AAPS journal*, vol. 15, no. 2, pp. 608–617, 2013.

[60] S. Lohani, H. Cooper, X. Jin, B. P. Nissley, K. Manser, L. H. Rakes, J. J. Cummings, S. E. Fauty, and A. Bak, "Physicochemical properties, form, and formulation selection strategy for a biopharmaceutical classification system class ii preclinical drug candidate," *Journal of Pharmaceutical Sciences*, vol. 103, no. 10, pp. 3007–3021, 2014.

[61] Z. Lu, Y. Yang, R.-A. Covington, Y. V. Bi, T. Dürig, M. A. Ilies, and R. Fassihi, "Supersaturated controlled release matrix using amorphous dispersions of glipizide," *International journal of pharmaceutics*, vol. 511, no. 2, pp. 957–968, 2016.

[62] W. Curatolo, J. A. Nightingale, and S. M. Herbig, "Utility of hydroxypropylmethylcellulose acetate succinate (hpmcas) for initiation and maintenance of drug supersaturation in the gi milieu," *Pharmaceutical research*, vol. 26, no. 6, pp. 1419–1431, 2009.

[63] J. M. Pereira, R. Mejia-Ariza, G. A. Ilevbare, H. E. McGettigan, N. Sriranganathan, L. S. Taylor, R. M. Davis, and K. J. Edgar, "Interplay of degradation, dissolution and stabilization of clar-

ithromycin and its amorphous solid dispersions," *Molecular pharmaceutics*, vol. 10, no. 12, pp. 4640–4653, 2013.

[64] S. Baghel, H. Cathcart, and N. J. O'Reilly, "Investigation into the solid-state properties and dissolution profile of spray-dried ternary amorphous solid dispersions: a rational step toward the design and development of a multicomponent amorphous system," *Molecular pharmaceutics*, vol. 15, no. 9, pp. 3796–3812, 2018.

[65] O. Mahmah, R. Tabbakh, A. Kelly, and A. Paradkar, "A comparative study of the effect of spray drying and hot-melt extrusion on the properties of amorphous solid dispersions containing felodipine," *Journal of Pharmacy and Pharmacology*, vol. 66, no. 2, pp. 275–284, 2014.

[66] A. Paradkar, A. A. Ambike, B. K. Jadhav, and K. Mahadik, "Characterization of curcumin–pvp solid dispersion obtained by spray drying," *International journal of pharmaceutics*, vol. 271, no. 1-2, pp. 281–286, 2004.

[67] Y. Tian, V. Caron, D. S. Jones, A.-M. Healy, and G. P. Andrews, "Using f lory–h uggins phase diagrams as a pre-formulation tool for the production of amorphous solid dispersions: a comparison between hot-melt extrusion and spray drying," *Journal of Pharmacy and Pharmacology*, vol. 66, no. 2, pp. 256–274, 2014.

[68] A. A. Ambike, K. Mahadik, and A. Paradkar, "Stability study of amorphous valdecoxib," *International Journal of Pharmaceutics*, vol. 282, no. 1-2, pp. 151–162, 2004.

[69] R. Dontireddy and A. M. Crean, "A comparative study of spray-dried and freeze-dried hydrocortisone/polyvinyl pyrrolidone solid dispersions," *Drug development and industrial pharmacy*, vol. 37, no. 10, pp. 1141–1149, 2011.

[70] A. K. Mann, L. Schenck, A. Koynov, A. C. Rumondor, X. Jin, M. Marota, and C. Dalton, "Producing amorphous solid dispersions via co-precipitation and spray drying: impact to physicochemical and biopharmaceutical properties," *Journal of Pharmaceutical Sciences*, vol. 107, no. 1, pp. 183–191, 2018.

[71] P. Thybo, J. Kristensen, and L. Hovgaard, "Characterization and physical stability of tolfenamic acid-pvp k30 solid dispersions," *Pharmaceutical development and technology*, vol. 12, no. 1, pp. 43–53, 2007.

[72] J. H. Lee, M. J. Kim, H. Yoon, C. R. Shim, H. A. Ko, S. A. Cho, D. Lee, and G. Khang, "Enhanced dissolution rate of celecoxib using pvp and/or hpmc-based solid dispersions prepared by spray drying method," *Journal of Pharmaceutical Investigation*, vol. 43, no. 3, pp. 205–213, 2013.

[73] A. Paudel, Y. Loyson, and G. Van den Mooter, “An investigation into the effect of spray drying temperature and atomizing conditions on miscibility, physical stability, and performance of naproxen–pvp k 25 solid dispersions,” *Journal of pharmaceutical sciences*, vol. 102, no. 4, pp. 1249–1267, 2013.

[74] K. Yamashita, T. Nakate, K. Okimoto, A. Ohike, Y. Tokunaga, R. Ibuki, K. Higaki, and T. Kimura, “Establishment of new preparation method for solid dispersion formulation of tacrolimus,” *International journal of pharmaceutics*, vol. 267, no. 1-2, pp. 79–91, 2003.

[75] M. G. Fakes, B. J. Vakkalagadda, F. Qian, S. Desikan, R. B. Gandhi, C. Lai, A. Hsieh, M. K. Franchini, H. Toale, and J. Brown, “Enhancement of oral bioavailability of an hiv-attachment inhibitor by nanosizing and amorphous formulation approaches,” *International journal of pharmaceutics*, vol. 370, no. 1-2, pp. 167–174, 2009.

[76] X. Xiong, K. Xu, S. Li, P. Tang, Y. Xiao, and H. Li, “Solid-state amorphization of rebamipide and investigation on solubility and stability of the amorphous form,” *Drug Development and Industrial Pharmacy*, vol. 43, no. 2, pp. 283–292, 2017.

[77] S. Sotthivirat, C. McKelvey, J. Moser, B. Rege, W. Xu, and D. Zhang, “Development of amorphous solid dispersion formulations of a poorly water-soluble drug, mk-0364,” *International journal of pharmaceutics*, vol. 452, no. 1-2, pp. 73–81, 2013.

[78] T. H. Tran, B. K. Poudel, N. Marasini, J. S. Woo, H.-G. Choi, C. S. Yong, and J. O. Kim, “Development of raloxifene-solid dispersion with improved oral bioavailability via spray-drying technique,” *Archives of pharmacal research*, vol. 36, no. 1, pp. 86–93, 2013.

[79] D. Engers, J. Teng, J. Jimenez-Novoa, P. Gent, S. Hossack, C. Campbell, J. Thomson, I. Ivanisevic, A. Templeton, S. Byrn *et al.*, “A solid-state approach to enable early development compounds: Selection and animsal bioavailability studies of an itraconazole amorphous solid dispersion,” *Journal of pharmaceutical sciences*, vol. 99, no. 9, pp. 3901–3922, 2010.

[80] Y. Cho, E.-S. Ha, I.-H. Baek, M.-S. Kim, C.-W. Cho, and S.-J. Hwang, “Enhanced supersaturation and oral absorption of sirolimus using an amorphous solid dispersion based on eudragit® e,” *Molecules*, vol. 20, no. 6, pp. 9496–9509, 2015.

[81] C. Bothiraja, M. B. Shinde, S. Rajalakshmi, and A. P. Pawar, “Evaluation of molecular pharmaceutical and in-vivo properties of spray-dried isolated andrographolide—pvp,” *Journal of Pharmacy and Pharmacology*, vol. 61, no. 11, pp. 1465–1472, 2009.

[82] K. Wu, J. Li, W. Wang, and D. A. Winstead, “Formation and characterization of solid dispersions of piroxicam and polyvinylpyrrolidone using spray drying and precipitation with compressed anti-solvent,” *Journal of pharmaceutical sciences*, vol. 98, no. 7, pp. 2422–2431, 2009.

[83] K. Wlodarski, L. Tajber, and W. Sawicki, “Physicochemical properties of direct compression tablets with spray dried and ball milled solid dispersions of tadalafil in pvp-va,” *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 109, pp. 14–23, 2016.

[84] H. Yu and K. Hadinoto, “Mitigating the adverse effect of spray drying on the supersaturation generation capability of amorphous nanopharmaceutical powders,” *Powder Technology*, vol. 277, pp. 97–104, 2015.

[85] C. Liu, Z. Chen, Y. Chen, J. Lu, Y. Li, S. Wang, G. Wu, and F. Qian, “Improving oral bioavailability of sorafenib by optimizing the “spring” and “parachute” based on molecular interaction mechanisms,” *Molecular pharmaceutics*, vol. 13, no. 2, pp. 599–608, 2016.

[86] Encyclopedia.com, “Ordinary least squares regression,” accessed 24-Apr-2020. [Online]. Available: https://www.encyclopedia.com/social-sciences/applied-and-social-sciences-magazines/ordinary-least-squares-regression

[87] D. M. Pirouz, “An overview of partial least squares,” *Available at SSRN 1631359*, 2006.

[88] R. D. Tobias *et al.*, “An introduction to partial least squares regression,” in *Proceedings of the twentieth annual SAS users group international conference*, vol. 20. SAS Institute Inc Cary, 1995.

[89] A. McIntosh, W. Chau, and A. Protzner, “Spatiotemporal analysis of event-related fmri data using partial least squares,” *Neuroimage*, vol. 23, no. 2, pp. 764–775, 2004.

[90] R. Rosipal and N. Krämer, “Overview and recent advances in partial least squares,” in *International Statistical and Optimization Perspectives Workshop” Subspace, Latent Structure and Feature Selection”*. Springer, 2005, pp. 34–51.

[91] Umetrics, “User guide to simca,” version 13.

[92] R. D. Cramer, “Partial least squares (pls): its strengths and limitations,” *Perspectives in Drug Discovery and Design*, vol. 1, no. 2, pp. 269–278, 1993.

[93] T. D. Science, “Introduction to artificial neural networks(ann),” accessed 19-Apr-2020. [Online]. Available: https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9

[94] M. L. F. Scratch, "Neural networks: Feedforward and backpropagation explained optimization," accessed 19-Apr-2020. [Online]. Available: https://mlfromscratch.com/neural-networks-explained/#/

[95] M. Nielsen, "Neural networks and deep learning: improving the way neural networks learn." accessed 20-Apr-2020. [Online]. Available: http://neuralnetworksanddeeplearning.com/chap3.html

[96] A. Verikas, E. Vaiciukynas, A. Gelzinis, J. Parker, and M. C. Olsson, "Electromyographic patterns during golf swing: Activation sequence profiling and prediction of shot effectiveness," *Sensors*, vol. 16, no. 4, p. 592, 2016.

[97] T. Yiu, "Understanding random forest: How the algorithm works and why it is so effective," accessed 20-Apr-2020. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[98] W. Zong, J. Zhang, and Y. Jiang, "Life-oriented household energy consumption research," in *Transport and Energy Research*. Elsevier, 2020, pp. 373–391.

[99] H. Deng, "An introduction to random forest: Illustration, interpretation, biases, and usage for outlier detection and clustering," accessed 24-Apr-2020. [Online]. Available: https://towardsdatascience.com/random-forest-3a55c3aca46d

[100] W. Koehrsen, "An implementation and explanation of the random forest in python: a guide for using and understanding the random forest by building up from a single decision tree." accessed 20-Apr-2020. [Online]. Available: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76

[101] A. Bilogur, "Bias variance tradeoff," accessed 27-Apr-2020. [Online]. Available: https://www.kaggle.com/residentmario/bias-variance-tradeoff

[102] M. in Data Science, "What is the difference between bias and variance?" accessed 27-Apr-2020. [Online]. Available: https://www.mastersindatascience.org/resources/difference-between-bias-and-variance/

[103] E. Briscoe and J. Feldman, "Conceptual complexity and the bias/variance tradeoff," *Cognition*, vol. 118, no. 1, pp. 2–16, 2011.

[104] E. Allibhai, "Hold-out vs. cross-validation in machine learning," accessed 27-Apr-2020. [Online]. Available: https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f

[105] E. D. Science, “Overfitting in machine learning: What it is and how to prevent it,” accessed 28-Apr-2020. [Online]. Available: https://elitedatascience.com/overfitting-in-machine-learning#overfitting-vs-underfitting

[106] D. o. S. PennState Eberly College of Science, “Normal probability plot of residuals,” accessed 09-Sept-2020. [Online]. Available: https://online.stat.psu.edu/stat501/lesson/4/4.6

[107] Minitab, “Graphs for partial least squares regression,” accessed 09-Sept-2020. [Online]. Available: https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/regression/how-to/partial-least-squares/interpret-the-results/all-statistics-and-graphs/graphs/

[108] S. Misra and H. Li, “Noninvasive fracture characterization based on the classification of sonic wave travel times,” *Machine Learning for Subsurface Characterization*, p. 243, 2019.

[109] I.-G. Chong and C.-H. Jun, “Performance of some variable selection methods when multicollinearity is present,” *Chemometrics and intelligent laboratory systems*, vol. 78, no. 1-2, pp. 103–112, 2005.

[110] A. Lazraq, R. Cleroux, and J.-P. Gauchi, “Selecting both latent and explanatory variables in the pls1 regression model,” *Chemometrics and Intelligent Laboratory Systems*, vol. 66, no. 2, pp. 117–126, 2003.

[111] X.-M. Sun, X.-P. Yu, Y. Liu, L. Xu, and D.-L. Di, “Combining bootstrap and uninformative variable elimination: Chemometric identification of metabonomic biomarkers by nonparametric analysis of discriminant partial least squares,” *Chemometrics and Intelligent Laboratory Systems*, vol. 115, pp. 37–43, 2012.

[112] J. Brownlee, “How much training data is required for machine learning?” accessed 25-Sept-2020. [Online]. Available: https://machinelearningmastery.com/much-training-data-required-machine-learning/

[113] G. Developers, “Classification: Roc curve and auc,” accessed 28-Sept-2020. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[114] S. Hossain, A. Kabedev, A. Parrow, C. A. Bergström, and P. Larsson, “Molecular simulation as a computational pharmaceutics tool to predict drug solubility, solubilization processes and partitioning,” *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 137, pp. 46–55, 2019.