Ensemble of Machine Learning Algorithms for Economic Recession Detection

Rodrigo Lopes do Ó Barbosa

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves

Examination Committee

Chairperson: Prof. António Manuel Raminhos Cordeiro Grilo

Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves

Members of the Committee: Prof. Alexandre Paulo Lourenço Francisco

November 2018
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.
Abstract

This work intends to solve one of the most desired questions in economics, predicting when an economic recession will happen. The study uses the United States of America economy to develop its propositions, since it remains the most powerful economy in the world. It intends to solve this problem by using several macroeconomic indicators and applying Machine Learning algorithms to them. Three algorithms were used, the linear Logistic Regression and two nonlinear algorithms, the Random Forest Classifier and the XGBoost. Also used was an equally weighted average of these three algorithms to improve on their results. This work proved that with these models, with some transformations to the macroeconomic and recession signals, it is possible to predict recessions up to eighteen months in advance and with high accuracy. The best predictions were obtained using the Models’ Average, where above 50% of probability the results show residual amounts of false positives. These results were tested using the NBER recession dates and metrics such as the F1-Score, and the ROC curve with its AUC.

Keywords
Recession, United States of America, Macroeconomy, Indicators, Machine Learning, Classification, Binary, Logistic Regression, Random Forest, XGBoost, NBER, F1-Score, ROC curve, AUC.
Resumo

Este trabalho pretende resolver uma das questões mais ambicionadas em economia, prever quando ocorrerá uma recessão econômica. O estudo usa a economia dos Estados Unidos da América para desenvolver suas propostas, uma vez que esta continua a ser economia mais poderosa do mundo. Pretende-se resolver este problema usando vários indicadores macroeconômicos e aplicando algoritmos de Aprendizagem Automática. Foram utilizados três algoritmos, a Regressão Logística, um algoritmo linear, e dois algoritmos não-lineares, o Random Forest Classifier e o XGBoost. Também foi utilizada uma média ponderada destes três algoritmos para melhorar os seus resultados. Este trabalho provou que com estes modelos, e com algumas transformações nos sinais macroeconômicos e de recessão, é possível prever recessões com até dezoito meses de antecedência e com alta precisão. As melhores previsões foram obtidas usando a Média dos Modelos, onde acima de 50% de probabilidade os resultados mostram quantidades residuais de falsos positivos. Estes resultados foram testados usando as datas de recessão do NBER e com métricas, como o F1-Score, e a curva COR com a ASC.

Palavras-Chave

Recessão, Estados Unidos da América, Macroeconômico, Indicadores, Aprendizagem Automática, Classificação, Binário, Regressão Logística, Random Forest, XGBoost, NBER, F1-Score, Curva COR, ASC.
Acknowledgements

I would like to thank every person that crossed paths with me during college, from my course mates to the friends and co-workers I’ve made in the Students’ Association (AEIST). But I would like to leave a special thanks to my close friends from my first house, in Av. Rovisco País, for believing in me and supporting me through this journey: Pascal Esteves, Pedro Vieira, Gustavo Carita, Paulo Figueira, João Morgado, Rui Alves, Luis Minas, and Paulo Rente.

To my professor Rui Neves, some thanks for helping me develop a thesis that I’m proud of and that is relevant for our society. Another special thanks to Gustavo Carita for his unceasing help, guidance, and discussion that elevated the quality of my work.

To Lia Barbieri a kiss for all the patience and care that she gave me, not only during this work but since we met. To my mother and father, a very grateful thank you is in order, for all the support they gave since day one and for all the support the still give me. To my little sister the biggest of hugs and kisses, even though she is still young her spirit guides me to be good and exceed myself at every turn, thanks Sis. To all of my family a great hug, for believing in me during this ride and in every day of my life.

To everyone else whom I might have forgotten but were part of my life, thank you.

Dedication

I dedicate this work to my late great-grandmother for raising me to be a good man, that cares for those around me and helps them with work, love, and compassion. I hope that this work may help others, as she taught me to do, and in some way inspire them to do the same.

Thanks, Vó Luisa.
# Table of Contents

Declaration .............................................................................................................................. i
Abstract ................................................................................................................................. iii
Keywords ............................................................................................................................... iii
Resumo ................................................................................................................................... iv
Palavras-Chave ....................................................................................................................... iv
Acknowledgements .............................................................................................................. v
Dedication .............................................................................................................................. v
Table of Contents ................................................................................................................ vi
List of Figures ....................................................................................................................... ix
List of Tables. ......................................................................................................................... xi
List of Acronyms ................................................................................................................... xii
List of Symbols .................................................................................................................... xiii

1 Introduction ....................................................................................................................... 15
   1.1 Motivation ..................................................................................................................... 17
   1.2 Objectives ................................................................................................................... 18
   1.3 Contributions .............................................................................................................. 18
   1.4 Thesis Structure .......................................................................................................... 19

2 Background ....................................................................................................................... 21
   2.1 Economy and Economics .......................................................................................... 23
      2.1.1 Macroeconomics .................................................................................................. 23
      2.1.2 Economic Cycles .................................................................................................. 24
      2.1.3 Turning Point Prediction ...................................................................................... 24
   2.2 Time Series Analysis and Forecasting ....................................................................... 25
      2.2.1 Time Series Components ...................................................................................... 26
   2.3 Binary Classification .................................................................................................. 27
      2.3.1 Linear ................................................................................................................... 27
         2.3.1.1 Logistic Regression ....................................................................................... 28
      2.3.2 Nonlinear ............................................................................................................. 29
         2.3.2.1 Bootstrapping Aggregating (Random Forest) ............................................... 29
         2.3.2.2 Gradient Boosting (XGBoost) ..................................................................... 32
2.4 State of the Art .................................................................................................................. 35
  2.4.1 Data ............................................................................................................................ 35
  2.4.2 Models ......................................................................................................................... 36
  2.4.3 Results & Metrics ......................................................................................................... 36

3 Proposed Architecture and Implementation .................................................................. 39
  3.1 Introduction .................................................................................................................... 41

3.2 System Architecture ....................................................................................................... 41
  3.2.1 Database Layer .......................................................................................................... 42
  3.2.2 Data Transformation Layer ....................................................................................... 42
  3.2.3 Classification Layer ................................................................................................... 43
  3.2.4 Validation Layer ......................................................................................................... 43

3.3 System Implementation .................................................................................................. 44
  3.3.1 Database Layer Implementation .................................................................................. 44
  3.3.1.1 Macroeconomic Indicator Database ...................................................................... 44
  3.3.1.2 Recession Database ............................................................................................... 45
  3.3.1.3 Data Analyst .......................................................................................................... 46
  3.3.1.3.1 Individual Signal Analysis ................................................................................ 46
  3.3.1.3.2 General Signal Analysis .................................................................................... 51
  3.3.2 Data Transformation Layer Implementation ............................................................. 52
  3.3.2.1 Indicator Selection & Transformation .................................................................. 52
  3.3.2.1.1 Selection Phase ................................................................................................... 53
  3.3.2.1.2 Transformation Phase ........................................................................................ 54
  3.3.2.2 Lag Adder ............................................................................................................... 55

3.3.3 Classification Layer Implementation ............................................................................ 57
  3.3.3.1 Classification Algorithm ........................................................................................ 58
  3.3.3.1.1 Cross-Validation ................................................................................................. 58
  3.3.3.1.2 Linear Approach (Logistic Regression) .............................................................. 59
  3.3.3.1.3 Nonlinear Approach (Random Forest Classifier & XGBoost) ....................... 60
  3.3.3.2 Model Prediction Database ..................................................................................... 61
  3.3.3.3 Models’ Average ................................................................................................... 62

3.3.4 Validation Layer Implementation ................................................................................. 62
4 Results & System Validation ........................................................................................................ 63
  4.1 Metrics ................................................................................................................................ 65
    4.1.1 Confusion Matrix ............................................................................................................. 65
    4.1.2 Precision ......................................................................................................................... 65
    4.1.3 Recall .............................................................................................................................. 66
    4.1.4 F1-Score ......................................................................................................................... 66
    4.1.5 ROC Curve .................................................................................................................... 66
      4.1.5.1 Area Under the Curve (AUC) .................................................................................. 67
  4.2 Case Studies .......................................................................................................................... 68
    4.2.1 Methodology .................................................................................................................. 68
    4.2.2 Case Study A – Linear Approach, the Logistic Regression .......................................... 69
    4.2.3 Case Study B – Nonlinear Approach, the Random Forest Classifier ......................... 72
    4.2.4 Case Study C – Nonlinear Approach, the XGBoost .................................................... 75
    4.2.5 Case Study D – Model Averaging Approach ................................................................. 78
    4.2.6 Final Review .................................................................................................................. 81
5 Conclusion .................................................................................................................................. 83
  5.1 Conclusion ............................................................................................................................ 85
  5.2 Future Work ......................................................................................................................... 86
References ...................................................................................................................................... 87
Annexes ......................................................................................................................................... 92
  ANNEX A ..................................................................................................................................... 92
  ANNEX B ..................................................................................................................................... 93
  ANNEX C ..................................................................................................................................... 94
  ANNEX D ..................................................................................................................................... 95
  ANNEX E ..................................................................................................................................... 96
List of Figures

Figure 1. An example of a simple decision tree. ................................................................. 30
Figure 2. Example of a feature importance graphic [33]. .................................................... 32
Figure 3. Gradient Boosting algorithm proposed by Friedman [35]. .................................. 33
Figure 4. Weiling Liu and Emanuel Moench recession prediction plot and ROC curve plot in their work [17]. ........................................................................................................... 37
Figure 5. System Architecture ............................................................................................ 41
Figure 6. Partial graphic representation of the US recession signal. ..................................... 46
Figure 7. Information returned by the methods in the pandas_summary library .................. 47
Figure 8. Moving average plot of the Industrial Production of the US (cropped). ................ 48
Figure 9. Annual and monthly variation of the Industrial Production of the US (cropped). .... 48
Figure 10. Frequency distribution of the Industrial Production of the US. ............................ 49
Figure 11. Prophet forecast of the Industrial Production of the US (cropped). ...................... 50
Figure 12. Prophet trend and seasonality analysis of the Industrial Production of the US. ...... 50
Figure 13. Heatmap of the correlations between the macroeconomic indicators signals. ......... 51
Figure 14. Indicator Selection & Transformation internal architecture. ............................... 52
Figure 15. Dataframe of the selected and resampled macroeconomic indicators with the mean, maximum, minimum and standard deviation values (cropped). .................................... 54
Figure 16. Memory adder conceptual diagram. ....................................................................... 54
Figure 17. Lag Adder internal architecture. ............................................................................ 55
Figure 18. US recession signal shifted by eighteen months (cropped). .............................. 56
Figure 19. Logical operations to retrieve the first point of a signal. ........................................ 56
Figure 20. US recession signal with six, twelve and eighteen months lag (cropped). .......... 57
Figure 21. Classification Algorithm internal architecture. .................................................... 57
Figure 22. Forward chaining cross-validation. ...................................................................... 59
Figure 23. Model Prediction Database for the US recession signal with no lag .................. 61
Figure 24. Dataset produced by the averaging of the predictions of the three models. .......... 62
Figure 25. ROC curve of a tested model not used in the thesis. .......................................... 67
Figure 26. ROC curve with AUC of a tested model not used in the thesis. ......................... 68
Figure 27. The probability of a US recession happening with eighteen months of lag and six months of memory calculated by a Logistic Regression algorithm. ......................................................... 71
Figure 28. The probability of a US recession happening with twelve months of lag and six months of memory calculated by a Logistic Regression algorithm. .............................................. 71
Figure 29. The probability of a US recession happening with eighteen months of lag and three months of memory calculated by a Random Forest Classifier algorithm. ........................................... 74
Figure 30. The probability of a US recession happening with twelve months of lag and six months of memory calculated by a Random Forest Classifier algorithm. ......................................................... 74
Figure 31. Feature importance of the Random Forest Classifier with eighteen months of lag and no memory.............................................................................................................. 75
Figure 32. The probability of a US recession happening with six months of lag and no memory calculated by an XGBoost algorithm................................................................. 77
Figure 33. The probability of a US recession happening with twelve months of lag and six months of memory calculated by an XGBoost algorithm. ................................................................. 77
Figure 34. Feature importance of the XGBoost with six months of lag and no memory.................... 78
Figure 35. The probability of a US recession happening with six months of lag and no memory calculated by an average of all the models’ algorithms. .............................................. 80
Figure 36. The probability of a US recession happening with twelve months of lag and six months of memory calculated by an average of all the models’ algorithms.............................................. 80
Figure 37. Example of possible transformations to the recession signal. ........................................ 86
Figure 38. NBER list of Business cycle dates. ................................................................. 92
Figure 39. Weiling Liu and Emanuel Moench metric results table [17]. ........................................ 93
Figure 40. Some of the ROC curve results presented in the Wells Fargo work [41]. ................. 94
Figure 41. Estrella and Mishkin some metric results [3]............................................. 95
Figure 42. One of Estrella and Mishkin plot results [3]............................................. 95
Figure 43. Baba and Kişinbay some metric results [18]. ............................................. 96
List of Tables.

Table 1. Metadata from the macroeconomic indicators in the *Macroeconomic Indicator Database.* ... 44
Table 2. Selected macroeconomic indicators............................................................... 53
Table 3. Confusion matrix table for the US recession prediction. .................................. 65
Table 4. Case study A parameter configuration. ................................................................ 69
Table 5. *Classification Report* results for the *Logistic Regression* Model. .................. 70
Table 6. AUC results from the ROC curves of the *Logistic Regression* Model. .............. 70
Table 7. Case study B parameter configuration. ............................................................... 72
Table 8. *Classification Report* results for the *Random Forest Classifier* Model. .......... 73
Table 9. AUC results from the ROC curves of the *Random Forest Classifier* Model. .... 73
Table 10. *Classification Report* results for the *XGBoost* Model. ............................... 75
Table 11. AUC results from the ROC curves of the *XGBoost* Model. ........................... 76
Table 12. *Classification Report* results for the Models’ Average.................................. 79
Table 13. AUC results from the ROC curves of the Model’s Average.............................. 79
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Autoregressive</td>
</tr>
<tr>
<td>ARIMA</td>
<td>Autoregressive Integrated Moving Average</td>
</tr>
<tr>
<td>ARMA</td>
<td>Autoregressive Moving Average</td>
</tr>
<tr>
<td>AUC</td>
<td>Area Under the Curve</td>
</tr>
<tr>
<td>CFNAI</td>
<td>Chicago Fed National Activity Index</td>
</tr>
<tr>
<td>DM</td>
<td>Data Mining</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>FRED</td>
<td>Federal Reserve Bank of St. Louis</td>
</tr>
<tr>
<td>GDP</td>
<td>Gross Domestic Product</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IMF</td>
<td>International Monetary Fund</td>
</tr>
<tr>
<td>MA</td>
<td>Moving Average</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>NAFTA</td>
<td>North American Free Trade Agreement</td>
</tr>
<tr>
<td>NBER</td>
<td>National Bureau of Economic Research</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Networks</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>SARIMA</td>
<td>Seasonal Autoregressive Integrated Moving Average</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td>XGBoost</td>
<td>Extreme Gradient Boost</td>
</tr>
</tbody>
</table>
List of Symbols

\( Y(X) \) Linear Result

\( \beta_0 \) Constant

\( \beta_i \) Parameters

\( X \) Predictor Variable

\( l(x) \) Logistic Function

\( o \) Odds

\( b \) Base

\( p \) Probability of a class happening

\( Y \) Total production

\( L \) Labor

\( K \) Capital Input

\( \alpha \) Total Factor productivity

\( \beta \) Output elasticity of Capital

\( \gamma \) Output elasticity of Labor

\( D_t \) Distribution

\( h_t \) Weak Hypothesis

\( \epsilon_t \) Error with respect to the distribution \( D_t \)

\( L(y_t, \rho) \) Loss Function

\( a_m \) Parameter

\( \rho_m \) Multiplier

\( h(x, a_m) \) Base Learner

\( J_m \) Number of leaves in a Three

\( R \) Regions

\( F_m(x) \) Prediction

\( b_{jm} \) Coefficient
1

Introduction
1.1 Motivation

A common perception of working modern economies shows that they work around a trend rate of growth, with expansion and recession phases. The expansion periods bring economic growth, and usually increasing standards of living, such as purchasing power, increased salaries and the ability to afford good healthcare and education. As for the recession periods, the opposite tends to occur. As the economies halt their growth, salaries tend to stabilize, and with it, a reduction in purchasing power that sometimes can lead to difficulties in accessing basic healthcare and education. These fluctuations in growth affect not only individuals but also businesses, with reduced demands, reduction of profits and profitable economic opportunities. Clearly these shifts in economic growth also affect governments, as macroeconomic agents, with an increased pressure from people and business for help in reversing these downward trends.

In the light of these statements, it becomes obvious the importance that people, businesses, and governments give to discovering when these shifts are bound to happen. Therefore, the primary concern of this thesis is to find a way to predict these events with enough of a lead time so that these agents can take the right precautions. Since the analyses of all global economies is impractical to perform due to structural differences between them, the United States of America (US) economy must be the one to focus on, since its economy is still the strongest of the advanced economies, per the latest International Monetary Fund (IMF) report [1], meaning that a recession in the US is more likely to affect all other global economies.

The use and classification of economic variables to infer economic downturns has been used for a long time in economic research, at least since Burns & Mitchell [2]. Throughout time several distinct variables have been used, or in this case, economic indicators, that are recognized as useful indicators of an economic turning point, but the most acknowledged remains the yield curve [3]–[5]. There are also several academic researchers who focus on producing analyses of the state of the economy, which ended up being used for the implementation of the Chicago Fed National Activity Index (CFNAI) at the Federal Bank of Chicago [6], [7]. Nevertheless, most economists follow a broad variety of indicators of which their usability cannot be determined to assess the future state of the economy, since until now, none of them has proved completely reliable in the past.

So, this thesis intends to develop machine learning models that analyze diverse economic indicators and signal the possibility of an inversion point of the economic growth, in this case, the beginning of a recession phase through a variety of different time horizons.
1.2 Objectives

Given the motivation previously described, a set of objectives are defined to achieve what was proposed. These objectives are based on a set of assumptions required to make this problem solvable.

- Only the US economy is used for the study.
- The origin or cause of each recession is not explored.
- The discovery of the end date of a recession is not a goal.
- The strength of a future recession will not be obtained.

Having these assumptions defined for this work leads to the setting of the followings objectives.

- Retrieval of several macroeconomic indicators of the US economy.
- Analyze and simulate the evolution of these indicators.
- Use the well-defined recession dates to establish a target for the models.
- Construct several models with these indicators to discover the economic turning points.
- Analyze the difference in performance of the tested models.

By completing these objectives, it is expected to do an exploratory study of the matter and develop a method that can predict, with a reasonable time advance, a shift towards an economic recession and if possible, beating the current state of the art.

1.3 Contributions

This thesis adds some new contributions and developments to the current state of the art.

- **Input data transformation:**
  The macroeconomic indicators are transformed by a process of memory addition, increasing the amount of data in each point by adding previous information.

- **Early recession detection by lagging:**
  Instead of using the prediction results of a model and forecasting them into the future, this thesis adds lag to the economic recession signal.

- **Model Average of linear and nonlinear models:**
  To improve the results of each mode by decreasing the effects of their particular weaknesses, an average between the linear and nonlinear models is used.
1.4 Thesis Structure

This work is composed of five chapters.

- Chapter 1 – Introduction
  Grants an overview of the work, providing its motivation, objectives and contributions.

- Chapter 2 – Background
  Establishes the theoretical basis used in this thesis, on the economic level but also on the models used to retrieve results. Presents the state of the art regarding this subject.

- Chapter 3 – Proposed Architecture and Implementation
  Displays the adopted methodologies and software architecture used to accomplish this work. Also presents computational improvements to boost the algorithm.

- Chapter 4 – System validation
  Defines the validation procedure to analyze the results achieved by the models.

- Chapter 5 – Conclusion
  Summarizes the work development, and settles the best results and strategies established by this work. Additionally, describes future work on this area.
Background
2.1 Economy and Economics

To tackle the problem exposed in this work, some basic concepts are needed to understand not only the problematic but also the proposed solutions.

To begin with a definition of economy and economics is needed. Economy represents the consumption and production of scarce resources in a certain area, relating to goods and services, which exist to satisfy its participant's needs [8]. There are several types of economies that are mainly distinguished by private ownership or social ownership of the means of production, and by the market or planning decision of the allocation of resources. For the purpose of this study, it is important to know that the US economy is market capitalistic [9], meaning it is based on private ownership, with a market decision method for allocating resources. In other terms, producers and consumers determine what is produced and sold. Producers own the products and define their price, and consumers own what they buy, and decide how much they want to pay.

These economies and their affecting factors can be analyzed and studied, by a social science called economics which can be broken into two main subdivisions, microeconomics, and macroeconomics. Microeconomics studies the behavior of households and businesses, and their decision on what to buy and produce, respectively, and the quantities bought and produced. On the other end, macroeconomics deals with the economy as a whole, examining broader factors such as national consumption, national output, general price level or even unemployment [10]. For an analysis of global economic trends, it becomes clear that a macroeconomic study is needed, and is therefore an object of study of this thesis.

Having established these facts, this subchapter will further explore the concepts of macroeconomic studies, economic cycles, economic recessions and predictions of the economic turning points, since they are fundamental for understanding the solutions proposed in this work.

2.1.1 Macroeconomics

As noted before macroeconomics is the preferred study for understanding the structure and performance of an economy. It uses the aggregate expenditures and consumptions of a nation or region, the amount saved and spent by all households, or even the productiveness of a country’s labor as a basis for its analysis. For instance, most economists use the real Gross Domestic Product (GDP)\(^1\), a measure of total output, to determine if the economy of a nation is growing or shrinking [11]. But not only the GDP is used for the assessment of the state of the economy. Other indicators like industrial growth, manufacturing of new orders, and retail sales volume growth [12] can also be used for this purpose. This provides us with the first basic postulation that current macroeconomic studies use several models and indicators, related to production, labor or business, to analyze their object of study.

Macroeconomics also introduces the concept of business cycles or economic cycles, to describe the fluctuations in economic growth through time, based on expansions and contractions (recessions) of the

\(^1\)Real GDP - The total value of all final goods and services produced during a particular year or period, adjusted to eliminate the effects of changes in prices (Inflation).
economy, as described further ahead. This way it becomes clear that the use of macroeconomic indicators is a requirement to analyze the economy and its variations, making them indispensable for the resolution of the proposed work of this thesis.

2.1.2 Economic Cycles

As previously stated, an economic cycle describes the trend, and its variation, of the economy. A more specific approach for defining an economic cycle was proposed by Burns & Mitchell back in 1946 which is still being used today.

“Business cycles are a type of fluctuation found in the aggregate economic activity of nations that organize their work mainly in business enterprises: a cycle consists of expansions occurring at about the same time in many economic activities, followed by similarly general recessions, contractions, and revivals which merge into the expansion phase of the next cycle; this sequence of changes is recurrent but not periodic; in duration business cycles vary from more than one year to ten or twelve years; they are not divisible into shorter cycles of similar character with amplitudes approximating their own”. [2]

Even though it’s a broad definition and most economists tend to use more specific analysis, usually by measuring the trend of the GDP, the National Bureau of Economic Research (NBER)\textsuperscript{2} uses a more extensive range of parameters to define each cycle [13]. This information is important because the NBER has been widely accepted as the leading authority on both the definition of the business cycle dates, and the economic research behind it, providing a standard in this area of expertise. (ANNEX A).

Considering this explanation, the economy can only fluctuate between two distinct states, expansion or recession, simplified as economic growth or economic shrinkage respectively. This leads to the conclusion that during an expansion, macroeconomic indicators such as GDP, employment or production, tend to grow and during a recession, they tend to decrease. Therefore, it is safe to assume that the evolution of macroeconomic indicators, during a period of time, can describe not only the current state of the economy but also in which phase of the cycle it is currently positioned.

In sum, these facts demonstrate that for a conclusive analysis of the economic cycles the use of an evolutionary examination of macroeconomic indicators is required and that the NBER cycle dates are the benchmark to beat.

2.1.3 Turning Point Prediction

The cyclical behavior of the economy has been studied for several years by various economists and institutions, but like all cyclical behavior studies, their interests focus mainly on the inversion points, not the stable phases. It is after these shifting periods that critical correctional measures and different economic methods must be applied, resulting occasionally in structural changes in the economy, usually

\textsuperscript{2} The NBER is a private, non-profit, non-partisan organization dedicated to conducting economic research and to disseminating research findings among academics, public policy makers, and business professionals. [74]
after a recession. Therefore, the prediction of an inversion point leading to a recession phase is one of the most coveted goals in economics.

These changes to the economy have a propensity to occur after a recession, due to their detrimental effects on society. Recessions weigh very heavily on families, mainly due to unemployment, loss of purchasing power, the decline in welfare and education conditions, which ultimately tend to lead to earlier deaths, suicides, depressions, lower school achievement, child poverty and decreased birth rates [14]. The combination of these negative events has a tendency to change family habits, making them more economically literate on the causes of the previous recession, and able to deal with income and asset losses in the future [15]. This is also true for businesses and governments, who also must adapt to these changes in the fabric of the economy.

To detect these turning points, and following previous evidence, economists use a set of macroeconomic indicators and evaluation methods. Accordingly, it’s safe to assume that the variation of these indicators, related to consumption, production or employment, can be used to describe and identify the current phase of the economic cycle and a possible inflection point. But the set of indicators related to these areas is vast and presents several challenges, such as contradicting signals, survey sampling errors, and the attributed weight of each indicator [16]. It is clear that no one indicator is viable, and that a combination of them is required to assess the state of the economy. However, there are some indicators that surface in numerous literature as very strong predictors, like the yield curve, initial claims of unemployment insurance, new housing permits, monthly employment gains, or industrial production [3]–[5], [17]–[19]. In order to detect these turning points, current economist and researchers mainly use probabilistic models, as they are currently the best predictive tools available. Probit models, a specific type of binary classification models, are the most commonly used probabilistic models to estimate the probability that an observation fits into one of the two categories available, in this case, recession or non-recession. These models will be discussed in further chapters.

In conclusion, the detection of inflection points in business cycles is made using several macroeconomic indicators, mainly in the areas of consumption, production and employment, and the variation of these signals. The methods used to achieve these predictions are based on probabilistic models with an emphasis on binary classification models. Most of the economic community uses these assessments of inflection points to detect recessions due to their effects on the economy. These statements and those from previous subchapters set the most fundamental economic background used in this thesis.

2.2 Time Series Analysis and Forecasting

Time Series analysis, modeling, and forecasting have been applied to several problems in the last decades and has been the object of several studies and research. Its objective is to create and develop a model that can accurately describe a series throughout time and into the future using predictions or forecasts. This forecasting is usually described as the act of predicting the future by looking into the past and understanding it [20]. As such, time series analysis has become very important in the fields of engineering, economics, and finances. These last two are also considered to be the most challenging
to work with, due to their noisy, non-stationary and deterministically chaotic characteristics, as mentioned in [20].

Given the importance of a time series analyses to comprehend the behavior of financial and economic signals\(^3\), it is crucial to understand their components and how their models work in forecasting their future behaviors. This chapter will only examine the time series components, mainly used in univariate models\(^4\), leaving to further chapters the analyses of multivariate\(^5\) models. This distinction is needed to differentiate the evolution and forecasting of each economic indicator from the use of several economic signals to try to ascertain a different one, in this case, a recession signal.

### 2.2.1 Time Series Components

Time series are non-deterministic by nature, meaning that they cannot predict the future with certainty. It is presumed that they follow a probability model, which describes the distribution of a random time-dependent variable. The mathematical expression that describes this is referred to as a stochastic process [21], which can be designated as [22]:

> “Given an index set \(I\), a stochastic process indexed by \(I\) is a collection of random variables \(\{x_\lambda : \lambda \in I\}\) on a probability space \((\Omega, \mathcal{F}, P)\) taking values in a set \(S\). The set \(S\) is called the state space of the process.”

A simplification of this definition can be understood as a representation of the numerical values of a system, that randomly change over time. This description and the definition of a stochastic process, allows for the conclusion that a time series is a stochastic process. Therefore, the time series can be stationary, i.e. when all its random variables are identically distributed, and they typically have a normal distribution. But that is not always the case, especially for longer time span observations, where the time series components affect the stationarity nature of the series.

Since so many time series are nonstationary it is required to understand that they can be decomposed into four main components [23]:

- **Trend**: the monotonic change in the average level of the time series.
- **Trade Cycle**: A long wave in the time series. The object of study of this thesis.
- **Seasonality**: Fluctuation in the time series that recur during specific periods of less than a year and usually caused by factors such as weather, vacations or national holidays.
- **Residues**: Represents all the influences on the time series that are not explained by the previous components.

---

\(^3\) For the purposes of this thesis an economic or financial signal can be understood as a macroeconomic indicator.

\(^4\) Univariate analysis is the simplest form of data analysis where the data being analyzed contains only one variable.

\(^5\) Multivariate analysis is the analysis of three or more variables.
In general practice, the data is first analyzed and examined, usually by performing correlations\textsuperscript{6}, autocorrelation\textsuperscript{7} or even basic plot analyses, to discover these four components. These processes are important because they provide information that is useful in the choosing of which models to apply to achieve the proposed goals. They also demonstrate which transformations are required to perform on the data so that it can be used by the different forecasters. Some transformations are usually made for univariate models, like forcing stationarity by removing the trend and seasonality of the signal. Others are more typical of multivariate models, like changing the granularity\textsuperscript{8} of the signals, adding lag or normalizing the data.

### 2.3 Binary Classification

Converting the problem of economic recession detection to a mathematical and probabilistic one requires an examination of the possible and favorable outcomes. By reviewing the previous chapters, it is easy to conclude that there are only two possible outcomes available, there is either an economic recession or there isn’t. This leads to the postulation that this problem is not only a matter of classification, i.e. the outcome belongs to a specific category, but it is also binary since there are only two possible categories to which the results can be attributed. This deduction allows for a reduction of possible models to apply to the available indicators.

The first step is to understand what information is available and what is missing. From the economic analyses, it can be established that the input and the output data are easily available, meaning that the input, economic indicators, and the output, economic recession dates, have well-defined values and do not have to be inferred. However, the financial chapters also provide the first obstacle, the transformation or correlation between the input and the output. This knowledge is important when building forecasting models since the output is an unknown variable and the only available information is the input. To infer on the process of transformation of an input to an output, one of the leading areas of study is considered to be Machine Learning (ML) [24]. Mainly machine learning techniques use the available data to infer on some other information, which in the study of this thesis presents itself as a good solution for discovering recession dates based on economic signals.

Machine learning binary classification models tend to devise a rule, from the data available, mainly using a multivariate analysis, that establishes when a certain piece of information would result in a zero or one response. In this thesis, the result could represent: 1 – recession, 0 – no recession. And to achieve this goal two main multivariate approaches are used, the linear approach and the nonlinear one.

#### 2.3.1 Linear

The linear approach follows the premise that the developed model will use a linear regression. First, it is necessary to understand the concept behind a linear regression so that the study of linear models can be comprehended.

\textsuperscript{6} Correlation is the statistical relationship between two signals.
\textsuperscript{7} Autocorrelation is the correlation of a signal with a delayed copy of itself as a function of delay.
\textsuperscript{8} Data granularity refers to the size in which data fields are sub-divided, e.g. months, weeks, days.
A model is defined as linear when each of its terms is either a constant, or the product of a parameter with a predictor variable [25]. These models tend to follow an equation of the type,

\[ Y(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k \]  

(1)

where \( \beta_0 \) is a constant, \( \beta \) is a parameter and \( X \) is a predictor variable. Following this concept, it may appear that the linear models are incapable of fitting curves but that is not the case, since the linearity is set in the parameters and not on the predictor variables, so the model could be,

\[ Y(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2^2 \]  

(2)

and still be linear [25]. This demonstrates that the linear models can be applied to a binary classification problem even with some modifications. One of these models that is interesting to study is the Logistic Regression.

### 2.3.1.1 Logistic Regression

The Logistic Regression model is one of most used models in statistical models, in that it uses the logistic function to model binary dependent variables. This model appears suited to make a binary classification analysis since it uses several continuous values to determine an outcome that can only be one or zero [26].

To begin understanding the logistic regression model, an example following the basis of [26] and [27] can be presented. First, considering a model with a certain number of predictors, for instance, macroeconomic indicators, \( x_1, x_2, \ldots, x_p \), that take continuous values can be displayed in a logistic function like,

\[ l(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \]  

(3)

very much like the equation (1), where \( \beta \) are the model’s parameters, confirming that this is a linear model. The corresponding odds can be given by,

\[ o = b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p} \]  

(4)

where \( b \) is the base of the logarithm and exponent. Now to predict the probability of finding a recession, meaning, \( o = o : 1 \), it can be given by,

\[ p = \frac{b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p}}{1 + b^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}} \]  

(5)

where \( \beta_0 \) is the y-intercept\(^9\), and can be interpreted as log-odds, if all the predictors are zero. Also, the \( \beta_1, \beta_2 \) to \( \beta_p \) can be interpreted as the effects of \( x_1, x_2 \) to \( x_p \) since they increase the odds by \( b^{x_i} \). This model can be used for several purposes, and with numerous predictors, which for this thesis is advantageous, since there are various macroeconomic indicators that can be used to discover the binary US recession signal. In conclusion, the Logistic Regression model can be described as [27],

---

\(^9\) The point that intercepts the function in the Y axis.
\[ \text{logit } p = \frac{e^{l(x)}}{e^{l(x)} + 1} \] (6)

substituting \( b \), from equation (5), with the exponential \( e \) and using \( l(x) \) from equation (3).

2.3.2 Nonlinear

The nonlinear approach follows a very different methodology than the linear one. These models use nonlinear regressions to fit signals, and are usually preferred for fitting curved ones.

The nonlinear models can be divided into two categories, one is nonlinear in the variables and linear in the parameters, and the other is nonlinear in the parameters [28]. One example of a nonlinear model would be the Cobb-Douglas production function,

\[ Y = \alpha L^\beta K^\gamma \] (7)

Using the logarithms yields in (7),

\[ \ln(Y) = \delta + \beta \ln(L) + \gamma \ln(K) \] (8)

where \( \delta = \ln(\alpha) \). This function is nonlinear in the \( Y \), \( L \) and \( K \) variables and is linear on the parameters \( \delta \), \( \beta \) and \( \gamma \). This shows that the nonlinear models can take different forms and are therefore able to fit very different kinds of signals, proving to be very adaptable, and probably a good solution for a binary classification system. There are many methods for implementing nonlinear models, but two of the most used and interesting are the Bootstrapping Aggregating and the Gradient Boosting.

2.3.2.1 Bootstrapping Aggregating (Random Forest)

Bootstrapping aggregating, also called bagging, is a meta-algorithm created to improve the stability and accuracy of ML algorithms, being considered a special case of the model averaging approach\(^\text{10}\). To begin this study first it is required to understand what decision trees are, as they are the main subject of the application of bagging.

\(^{10}\) Model Averaging or Ensemble Learning, uses multiple learning algorithms to obtain better predictive performance than only using one single algorithm.
A decision tree is a decision support tool that uses tree-like graphs to make decisions. These graphs consist of three types of nodes [29],

- **Decision nodes:**
  A decision node is a decision making procedure that chooses which action to take, i.e. selects to which of its “children” should it continue the process according to the information it has received. These nodes are usually represented as squares.

- **Chance nodes:**
  A chance node does not make decisions but differentiates the connections to the “children” nodes by probabilities. These nodes are usually represented as circles.

- **End nodes:**
  The end nodes, also called leaves, end the sequence of actions and/or reactions, in the decision problem. They are usually represented by triangles.

![Figure 1. An example of a simple decision tree.](image)

As can be seen in Figure 1, a decision tree receives a question and provides answers according to a set of decisions that it makes depending on the situational factors. Having understood what decision trees are, the next step is to understand the predictive model, Decision Tree Learning.

The Decision Tree Learning, uses two types of decision trees, the regression tree, where the targeted variable can be a continuous value, or classifications trees, where the targeted variable belongs to a class, like in Figure 1. In this thesis, only the classification trees are of interest. The classification trees are used to classify an object or instance into a specific class, based on their attributed values [30]. They are usually used in fields like finance, engineering, and medicine and are very good exploratory methods [30].

There are two methods of the Decision Tree Learning that use several classification trees at a time in order to improve their results. These are, the **Boosted Trees** and the **Bootstrapped Aggregated** decision trees. The former will be studied in the next chapter. Having established the connection between the classification decision trees and the bootstrapping aggregating algorithm it is then required to understand their function.
The bagging was proposed by Leo Breiman in 1994, where he stated that [31],

"Bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor."

Basically, Breiman proposed that using several bootstrap replicates of the datasets, running them in classification trees and averaging their results would improve the accuracy of the model instead of using only one dataset. Afterward, in 1995, Tin Ho developed the concept of random decision forests. They were devised because decision trees could not grow in complexity, and therefore had a loss in accuracy for unseen data, and suboptimal accuracy in training data. This problem was solved using the stochastic principle of modeling, and the accuracy for both the training and unseen data was increased by building multiple trees in randomly selected subspaces of the feature space [32]. Ho concluded that trees in different subspaces generalize their classification in complementary ways and their combined classification would therefore be improved. Breiman then used the concept devised by Ho and joined his bagging proposition and developed what now is one of the most used models in ML, the Random Forests.

Breiman defined random forests as [33],

“A random forest is a classifier consisting of a collection of tree structured classifiers \( \{ h(x, \theta_k), k = 1, \ldots \} \) where the \( \{ \theta_k \} \) are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input \( x \).”

Essentially, Breiman constructs several trees that have random vectors or datasets. Each tree gets several results that it groups by class, and then votes on the most occurring class for a specific input. This strategy works as an effective prediction tool because it doesn’t overfit due to the Law of Large Numbers. This model also provides an interesting mechanism for detecting which variables are providing the predictive accuracy for the model, usually called feature importance.

The feature importance mechanism uses M input variables, and after each tree is constructed the out-of-bag\(^\text{11}\) data is run through the corresponding trees and their classification is saved. This process is repeated for all the M variables, and after it ends all the out-of-bag classes vote [33]. Afterward, these classifications are compared with the true label of the classes to construct a variable importance.

\(^{11}\) Out-of-bag data is a set of bootstrap datasets which does not contain a particular record from the original dataset.
In Figure 2 the second variable is the most important, followed by the eighth and the sixth ones. This example was used by Breiman where he proves that the second variable is the most important, but with the addition of all variables, the global error diminishes [33].

This concludes that the Random Forests are good methods for generating classification predictions using a multivariate analysis because they don’t tend to overfit, and they also provide an a feature importance mechanism that can be useful in an economic analysis of a recession.

2.3.2.2 Gradient Boosting (XGBoost)

As with bootstrapped aggregating, gradient boosting is an ML technique for regression and classification problems. It typically uses decision trees, and therefore Decision Tree Learning. Consequently, it can use two types of trees, regression and classification ones. Once again only the classification trees will be studied in this thesis. Also, as stated in the previous subchapter, there are two methods of the Decision Tree Learning that use several classification trees at a time, the Boosted Trees and the Bootstrapped Aggregated decision trees. For the gradient boosting approach, only the Boosted Trees are used. But before explaining what Boosted Trees and Gradient Boosting are, it is necessary to understand Boosting.

Boosting is an ML algorithm designed to reduce bias and variance. Schapire and Freund describe Boosting as [34],

“…converting a ‘weak’ PAC\textsuperscript{12} learning algorithm that performs just slightly better than random guessing into one with arbitrarily high accuracy.”

\textsuperscript{12} Probably Approximately Correct is a framework for mathematical analysis that receives samples and selects a generalization function from a certain class of possible functions [75].
They also go on to describe it as, “Boosting refers to this general problem of producing a very accurate prediction rule by combining rough and moderately inaccurate rules-of-thumb”.

Basically, Boosting is the process of using predictors that are known to be imperfect, combining them to generate an improved predictor capable of producing better predictions on the target. This procedure uses a set of labeled training examples, \((x_i, y_i)^N\), where \(y_i\) is the label associated with \(x_i\). Each round \(t = 1, \ldots, T\) the booster devises a distribution \(D_t\) over the set of examples and request for a weak hypothesis, or a rule-of-thumb, \(h_t\), with low error, \(\varepsilon_t\), with respect to \(D_t\). This way, distribution \(D_t\) specifies the relative importance of each example for each round. After the \(T\) rounds, the procedure combines the weak hypothesis into a single prediction rule [34]. This describes the basic functioning of the Boosting algorithm.

In 1999 Jerome Friedman officially proposed the first gradient boosting algorithms and Tree Boosting. He considered that the gradient boosting of regression and classification trees produced competitive and highly robust procedures for regressions and classifications [35]. The idea behind gradient boosting was to follow the concept of the Boosting, combining the ‘weak’ learners into a strong one, and finding an approximation to the function that minimized the expected value of some specific loss function. Therefore, to achieve this he used a Gradient Descent\(^{13}\) that gave the name to the algorithm. Friedman proposed an algorithm as follows,

<table>
<thead>
<tr>
<th>Algorithm 1: Gradient Boost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

Figure 3. Gradient Boosting algorithm proposed by Friedman [35].

With a training set of \(\{(x_i, y_i)\}_{i=1}^{N}\), a loss function \(L(y_i, \rho)\), and a number of iterations \(M\). The algorithm from Figure 3 is initialized with a constant value in line 2. It then enters a loop until it completes all the iterations. In this loop, it starts by calculating the pseudo residuals in line 3, then fitting a base learner\(^{14}\), with the parameter \(a_m\), to the pseudo residuals in line 4 and computes a multiplier \(\rho_m\) by solving a one-

\(^{13}\) Gradient Descent is a first-order iterative optimization algorithm that finds a minimum of a function.

\(^{14}\) Weak learner.
dimensional optimization problem in line 5. Finally, it ends in line 6 by updating the model using the multiplier $\rho_m$ and the base learner $h(x, a_m)$ [35]. Basically, it iteratively will add the weighted base learners at each point to create a final one that is the sum of all the base learners.

With the gradient boosting established, the next step is to comprehend the Tree Boosting. Friedman also developed a special modification to use with trees from the Decision Tree Learning, so that they could better fit each base learner. If the algorithm in Figure 3 were to be applied to a tree, the first thing was to partition the input space by the number of leaves in the tree, $J_m$, into disjointed regions $R_1, ..., R_{J_m}$, and then predict a constant value in each region. Therefore, the base learner for an input of $x$ would be,

$$h_m(x) = \sum_{j=1}^{J_m} b_{jm} I_{R_{jm}}(x) \tag{9}$$

where $b_{jm}$ would be the value/class predicted in the region $R_{jm}$. Then, using the lines 5 and 6 of Figure 3, it would return something like,

$$F_m(x) = F_{m-1} + \rho_m h_m(x) \tag{10}$$

where all the coefficients $b_{jm}$ would be multiplied by some value $\rho_m$. Friedman then proposes to modify its algorithm to choose a different $\rho_m$ for each of the tree’s regions, instead of a single $\rho_m$ for the entire tree, calling it Tree Boosting [35]. The Tree Boosting simply discards the coefficients $b_{jm}$ from the tree fitting procedure,

$$\rho_{jm} = \arg \min_\rho \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \rho) \tag{11}$$

Using equation 11 to reach a similar result of equation 9,

$$F_m(x) = F_{m-1} + \sum_{m=1}^{J_m} \rho_{jm} I_{R_{jm}}(x) \tag{12}$$

In conclusion, this solution would provide better results for each tree. These notions present the basis of the Gradient Boosting and Tree Boosting, and show that they provide good results for classification problems as the ones faced in this thesis.

But the work on gradient boosting trees has not stopped since Friedman. One of the latest models that uses this concept is the Extreme Gradient Boost (XGBoost), proposed by Tianqi Chen and Carlos Guestrin [36]. This model follows the previous notions of Tree Boosting and Gradient Boosting but uses different weight distribution procedures for sparse datasets. This model was specifically designed for computational use, with improvements on the former models in terms of computational time and memory allocation, but still using the same principles. Therefore, this model presents itself as a very good option when trying to implement a gradient boosting classifier as the ones seen in this chapter.
2.4 State of the Art

As stated before the prediction of an economic recession is a long-desired goal by many economists, but also by many other areas of study, including engineering and computer science. With the development of the computational power of Personal Computers (PC) some old mathematical models became increasingly easy to apply and use on great amounts of information. This development led to the booming of areas of study like ML, Data Mining (DM), and Neural Networks (NN), that began providing answers to long unanswered questions in our societies, as seen in the works of Rui Neves and Nuno Horta [37]–[40]. The combinations of these two ideas guided many scientists and economists to the computer realm in order to predict economic recessions.

Even though this is recognized as a hugely competitive market, these studies are still surrounded in some mystery, where not all literature methodologies, datasets or even graphic results are presented. These omissions present themselves as one of the main reasons for this thesis exploration, and the objective of improving or replicating their results with fewer, and possibly weaker, indicators and less computational power. So, to establish a field of comparison this chapter uses the most complete literature available and is divided into three areas of relevance, the data, the models, and the results & metrics.

2.4.1 Data

The data used to feed the different models is the first, and probably one of the most important parts in designing a process to discover when a recession is going to happen. Here is where some of the literature begins to hide their datasets or even where the information was acquired. This last point is very important because macroeconomic information is generally subjected to changes and updates by the structures that produce it. In this way, it is not only a matter of having the right amount of information, but also having the right information.

Each author uses several different indicators, but as stated in the economic background, some areas appear to be of bigger relevance than others. For Instance Arturo Estrella and Frederic Mishkin, some of the first to explore this theme, use variables more connected with interest rates and money availability, rather than production and working conditions [3]. Indicators like the yield curve, the stock prices of the Dow Jones and S&P500, and the money stock in all its variances, represent their investment in this area, but nonetheless, they still used other information from different areas like housing permits or the consumer price index. Other authors like Travis Berge, use more macroeconomic indicators like the industrial production, the average weekly hours of the manufacturing industry or the initial claims of unemployment [19]. These approaches follow a smaller set of indicators, usually between ten and twenty different ones, but not all authors use this low information method. Chikako Baba and Turgut Kışınbay of the IMF use up to 166 macroeconomic indicators divided into five categories: income and output, employment, construction, interest rates, and finally nominal prices and wages [18]. Also, the Wells Fargo Securities Economics Group uses a high-level number of indicators approach. They start by retrieving 500,000 indicators from the Federal Reserve Bank of St. Louis (FRED), then by only using
the start date of 1972 reduce the dataset to 5,889 variables and eventually, by using further procedures, they reduce the dataset to a final count of 192 different indicators [41].

These approaches show two big differences, one with a low number of economic indicators, where the variables are chosen accordingly to an economic analysis and differ from each authors’ point of view. The other uses high numbers of economic indicators and does not make any real assumptions on the data but requires an arduous process of data retrieval and preparation.

2.4.2 Models

Having established the data to use in the forecasting process there are several mathematical models proposed to solve this type of binary classification problem. Most of the literature follows the same basic principles on this matter, adding some alterations in data handling and forecasting methods, but almost always using multivariate models.

The most commonly used methods are the probit and logit models. These models are considered good solutions for binary classification problems, and therefore many authors used them, if not to develop the final forecast then as part of their system [17], [18]. But even with the positive results shown by these models, the community kept on trying different approaches to the problem and lately, companies like Wells Fargo also tried Random Forest approaches and Gradient Boosting to deal with their data [41]. These models are the current state of the art, not only in the recession detection area but in almost every problem that deals with binary classification.

But apart from the models themselves, the authors have used several other strategies to improve their results by meddling with the data or even with the models. One of the strategies for forecasting recessions with some time advance was to make a recession prediction, using some specific model, and then using its results and forecasting them with a new univariate model [4]. This technique allowed them to understand the predictions would lead. They also presented a more complicated alternative to the direct route, an iterative mode of forecasting several quarters ahead and trying to decipher if there was a recession between these times [4]. But not only upon the future prediction methods are these changes applied, Travis Berge for instance, also applies several model averages on his forecasting models to try and improve their power. He uses a normal weighted average, a different weighted average called Bayesian Model Average, and also an alternative solution of choosing the better model through a boosting algorithm [19].

All of the cases presented are not only the state of the art solutions for detecting an economic recession, but they also set some of the tests and transformations made on the data and models to improve their results.

2.4.3 Results & Metrics

Regarding the results and metrics of the state of the art economic recession detection, no benchmark is used alternatively to the US recession dates provided by the NBER. These dates provide a term of comparison in all literature and are widely accepted as the signal to beat. As for the metrics used, they tend to vary from author to author, with different explanations being used.
The mainly used metrics tend to be the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). These metrics are better explained in chapter 4 but suffice it to say that if the ROC curve is near the top left corner and the AUC is almost 1, then the model is considered accurate. Authors like Weiling Liu and Emanuel Moench, use these metrics to evaluate their work [17], retrieving results for their several tests with AUC values around 0.8 (ANNEX B).

![ROC Curve Plot]

Figure 4. Weiling Liu and Emanuel Moench recession prediction plot and ROC curve plot in their work [17].

Figure 4 represents the type of results expected to be unveiled at the end of a recession detection prediction. There is a visual representation of the prediction against the actual recessions (grey vertical bars in the first plot), and the ROC curve to validate the results. This metric is not only used in this work, it is also used in the recent work of Wells Fargo, with AUC results around the 0.9 values, even though these values don't seem to fit the graphics they represent (ANNEX C) [41].

Older works like the Estrella and Mishkin (ANNEX D) and the Baba and Kışınıbay (ANNEX E), used other types of metrics to evaluate their works. Estella uses the Pseudo R² and Baba the Quadratic Probability Score (QPS) and the Log Probability Score (LPS) [3], [18]. These metrics are not used very often in the more recent works in the area.

This review establishes the state of the art of the US recession detection, setting some benchmarks for this thesis, and supplying some ideas for the system implementation and validation.
3

Proposed Architecture and Implementation
3.1 Introduction

This chapter addresses the architecture and implementation of a solution to the previously stated problem of predicting an economic recession in the United States of America. This solution is conceptualized in the System Architecture subchapter and its practical application in System Implementation. The goal is to describe the methods used to reach the results presented in chapter 4, and to provide insight into the process of the decision and design of these methods. The choices made in this section follow the conclusions retrieved from the Background chapter.

3.2 System Architecture

The design of the solution is based on a layered scheme architecture, where each layer of the system provides a logical division between the storage, the transformation and the forecasting of the data. This segmentation has a very straightforward approach, but it is abstract enough so that future changes in the system can be applied with no major complications.

![Figure 5. System Architecture](image)

There are four main layers in this system architecture, the Database Layer, Data Transformation Layer, Classification Layer, and the Validation Layer, and each have several components, as shown in Figure 5. It can clearly be noted that the CSV Files do not enter any of these layers, mainly because they were retrieved directly from the internet with no resource to any tool designed for this thesis.

The process begins by retrieving the CSV Files from the FRED website [42], and sending them to the Database Layer. This layer stores the information sent by the CSV Files where it can be analyzed, for instance, by its granularity and correlation, and is then directed to the next step, the Data Transformation Layer. Here it receives these signals and reshapes them accordingly to the users’ desires, changing their memory, granularity, or even adding lag, so that this information is processed by the Classification Layer in the desired manner. Then the Classification Layer takes these signals and
runs them through the chosen models to produce the final results that are sent to the **Verification Layer** to be observed, evaluated, and compared to the benchmark performance.

The following sub-chapters will provide a deeper explanation of the architecture of each segment and their according goals.

### 3.2.1 Database Layer

This Layer, as seen in Figure 5, has two main components, the **Macroeconomic Indicator Database** and the **Recession Database**, and each has its own branch that extends to the **Classification Layer**. There is also an analytic component, the **Database Analyst**, that can receive data from both databases and provide simple analyses.

The **Macroeconomic Indicator Database** stores and handles the macroeconomic indicators retrieved from the FRED and provides access to the **Data Analyst** and the **Indicator Selection & Transformation** components. The data stored in this database has very different granularities, starting and ending dates, units, and seasonalties. This leads not only to difficulties in analyzing the data, but also to a requirement to transformed them, so they can be used in the **Classification Layer**, as it will be demonstrated in the System Implementation chapter.

The **Recession Database** has the information about US recession dates with a monthly granularity, and its instances can be used in the **Data Analyst** component like the **Macroeconomic Indicator Database** and in the **Lag Adder**. This database may seem unnecessary, but it is separated from the **Macroeconomic Indicator Database** for abstraction purposes, guaranteeing that in the future other economic recession signals can be added and, analyzed between each other and against indicators from different countries.

The last component of the **Database Layer** is the **Data Analyst**, which is responsible for the analysis on each individual indicator or recession, the combination of data from within each database and the combination of data between both databases. The **Data Analyst** performs evaluations on the granularity, number count, verification of missing data, histogram studies. It also creates a primary forecasting for each signal and analyzes the correlation between signals from the same and different databases. This allows for a more comprehensive study of the available signals in order to help the user choose which signals to use. This reveals importance because the signals can be highly correlated, which would lead to overfitting, or even to reduce the size of the dataset if computational power is not available. In conclusion, this layer has to be able to compartmentalize the data and grant the user the ability to analyze it.

### 3.2.2 Data Transformation Layer

The **Data Transformation Layer** like the **Database Layer** has two main branches, mainly differentiated by the content of the data. One deals with the macroeconomic indicators and the other uses the economic recessions. This is required since the data has different specialties, different methods to change it and two different goals to achieve.
The *Indicator Selection & Transformation* component’s main goal is to supply the user with the ability to choose which indicator to use, according to its analyses made in the *Data Analyst*, and transformations to use in the *Classification Algorithm*. This component also guarantees that all the indicators have the same granularity, end and start dates, and can be processed by the *Classification Algorithm*. For the architecture design phase, this component was simplified, not showing the entirety of the processes within it, but it will be further explored in the Data Transformation Layer Implementation subchapter.

As with the *Indicator Selection & Transformation* the *Lag Adder* was also simplified for this stage and will be developed in further chapters. But its main goal is to add a lag to the economic recession signal in order to “trick” the *Classification Algorithm* into “thinking” that a recession begins before it actually does. It is expected that with this technique the chosen models would be able to detect a recession with some advance and therefore accomplish the purpose of this thesis.

### 3.2.3 Classification Layer

This layer only has one component, the *Classification Algorithm*, that receives two datasets, one from the *Indicator Selection & Transformation* and the other from *Lag Adder*, with the chosen macroeconomic indicators and the lagged economic recession signals, respectively. The purpose of this layer is to produce forecasts of US economic recessions using the macroeconomic indicators, with the most precision and recall possible. These concepts will be explored in further chapters.

The *Classification Algorithm* component uses several models that receive the selected macroeconomic signals and their transformations, in order to produce a binary response that assimilates the US recession signals with their several lags. The component also returns other validation responses that are fundamental for the user to assess the confidence that can be handed to the model and its responses. This segment of the system may require more computational power and therefore, can and must be worked by the user in order to make it viable for running in the testing machine. The explanation of the working process of the component was simplified and will be described in detail in the subchapter Classification Layer Implementation.

### 3.2.4 Validation Layer

This layer has a very different behavior from the previous ones, since it does not have any component, and only shows the results from the *Classification Layer* to the user so that they can be analyzed and concluded upon. This way the *Verification Layer* can be considered an abstract layer or a ghost layer, since it doesn’t really exist but also makes no logical sense to join it with the *Classification Layer*.

The *Verification Layer* presents to the user, the plots returned from the used models for a graphic evaluation and the ROC curve, the precision, recall, f1-score, and confusion matrix for an analytic examination of the results. This layer can work as a powerful iterative tool to understand possible problems with the chosen models, their definitions or even with the indicators themselves, enabling the user to adjust the system for better results. The explanation and evaluation of these results will be presented in the chapter Results & Validation.
3.3 System Implementation

Having established the four layers of this thesis’ system architecture, presented in Figure 5, and their logical connection, conceptual processes, and objectives, this chapter will now focus on the implementation of said architecture.

All the programs used in this work were developed with the use of Python programming language (version 3.7) [43] and Jupyter Notebook (version 5.7) [44] as an Integrated Development Environment (IDE) for Python. These technologies were chosen for their simplicity in data management, and intuitive and user-friendly IDE. All the databases used in this system use CSV Files [45].

3.3.1 Database Layer Implementation

As stated in previous chapters, the Database Layer has three components, the Macroeconomic Indicator Database, the Recession Database, and the Database Analyst. Both databases are very primary and are only composed of CSV Files retrieved directly from the FRED, and are not worked upon directly. They are loaded in the Database Analyst, the Indicator Selection & Transformation and the Lag Adder components with the use of the Pandas library [46] to convert the files into Dataframes for Python routines. The Dataframes have been used in the state of the art, in favor of other types of data structures, like Lists or Dictionaries, for their improved methods on dealing with large amounts of data.

3.3.1.1 Macroeconomic Indicator Database

The Macroeconomic Indicator Database uses several indicators with very different parameters, so to cope with these differences it was necessary to create a table with its metadata15

Table 1. Metadata from the macroeconomic indicators in the Macroeconomic Indicator Database.

<table>
<thead>
<tr>
<th>Indicator Name</th>
<th>Code</th>
<th>Granularity</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Hours Earning</td>
<td>AHE1%</td>
<td>Monthly</td>
<td>Percent, Seasonally Adjusted</td>
</tr>
<tr>
<td>Average Weekly Hours Manufacturing</td>
<td>AW1HM</td>
<td>Monthly</td>
<td>Hours, Seasonally Adjusted</td>
</tr>
<tr>
<td>Consumer Sentiment OECD</td>
<td>CSICP020USM665S</td>
<td>Monthly</td>
<td>Normalised (Normal=100), Seasonally Adjusted</td>
</tr>
<tr>
<td>Industrial Production</td>
<td>INDRPR</td>
<td>Monthly</td>
<td>Index 2012=100, Seasonally Adjusted</td>
</tr>
<tr>
<td>Initial Claims Unemployment Average</td>
<td>IC40SA</td>
<td>Weekly</td>
<td>Number, Seasonally Adjusted</td>
</tr>
<tr>
<td>Money Stock</td>
<td>M2</td>
<td>Weekly</td>
<td>Billions of Dollars, Seasonally Adjusted</td>
</tr>
<tr>
<td>New Private Housing Building Permit</td>
<td>PERMFT</td>
<td>Monthly</td>
<td>Thousands of Units, Seasonally Adjusted Annual Rate</td>
</tr>
<tr>
<td>Production Manufacor Index</td>
<td>PMI</td>
<td>Monthly</td>
<td>Percent, above 50% expected growth and vice-versa</td>
</tr>
<tr>
<td>Real Disposable Income Value</td>
<td>DSPC096</td>
<td>Monthly</td>
<td>Billions of Chained 2009 Dollars, Seasonally Adjusted Annual Rate</td>
</tr>
<tr>
<td>Real Median Household Income</td>
<td>MEH01NSIA672N</td>
<td>Annual</td>
<td>2016 CPI-U-RS Adjusted Dollars, Not Seasonally Adjusted</td>
</tr>
<tr>
<td>Real Median Person Income</td>
<td>MPANUSGA672N</td>
<td>Annual</td>
<td>2016 CPI-U-RS Adjusted Dollars, Not Seasonally Adjusted</td>
</tr>
<tr>
<td>Real Money Stock</td>
<td>NREALE</td>
<td>Monthly</td>
<td>Billions of 1982 84 Dollars, Seasonally Adjusted</td>
</tr>
<tr>
<td>TED Spread</td>
<td>TEDRATE</td>
<td>Daily</td>
<td>Percent, Not Seasonally Adjusted</td>
</tr>
<tr>
<td>Trade Weighted Dollar</td>
<td>DTWEXM</td>
<td>Daily</td>
<td>Index Mar 1973=100, Not Seasonally Adjusted</td>
</tr>
<tr>
<td>Unemployment Rate</td>
<td>UNRATE</td>
<td>Monthly</td>
<td>Percent, Seasonally Adjusted</td>
</tr>
<tr>
<td>Volatility Index</td>
<td>VIXCLS</td>
<td>Daily</td>
<td>Index, Not Seasonally Adjusted</td>
</tr>
<tr>
<td>Yield Curve</td>
<td>T10Y2Y</td>
<td>Daily</td>
<td>Percent, Not Seasonally Adjusted</td>
</tr>
<tr>
<td>Yield Spread AAA</td>
<td>AAA10Y</td>
<td>Daily</td>
<td>Percent, Not Seasonally Adjusted</td>
</tr>
<tr>
<td>Yield Spread BAA</td>
<td>BAA10Y</td>
<td>Daily</td>
<td>Percent, Not Seasonally Adjusted</td>
</tr>
</tbody>
</table>

As it can be stated from Table 1, the retrieved data has very different information, specifically in terms of granularity, units, and seasonality. The process to deal with variances in the data will be explained further ahead.

15 Metadata is data that provides information on other data.
The choosing of the indicators to compose this database was achieved by analyzing the data used by the state of the art and the economic background study. These indicators describe the segment of the economy destined to comprehend labor, money availability, purchasing power, production, and trust. For instance:

- **Labor:**
  - Average Hourly Earnings
  - Unemployment Rate

- **Money Availability:**
  - Real Money Stock
  - Yield Curve
  - TED spread

- **Purchasing Power:**
  - Real Median Personal Income
  - New Private Housing Building Permit

- **Production:**
  - Industrial Production
  - Average Weekly Hours of Manufacturing

- **Trust:**
  - Consumer Sentiment OECD
  - Production Manufacturing Index

This is an example of how different indicators affect different areas of the economy, even though some of them affect more than one area at a time. It was possible to add more indicators to this database, but since the objective of this thesis was to conduct an exploratory study on the matter, and prove that the concept is attainable, only the most important ones, according to the state of the art and economic study, were chosen, and not even all of them went to the final solution as can be perceived in the following chapters.

### 3.3.1.2 Recession Database

The *Recession Database* is far simpler than the *Macroeconomic Indicator Database* since it only includes the US economic recession data, provided by one CSV File. Although the system is ready for more economic recession signals, for this study only the US economy was relevant.
As seen in Figure 6, the signal can only take the values one and zero, which as previously stated, turns this problem into a binary classification one. The graphic shows that in recent years the data has been unbalanced, meaning that there are far more zero values than ones. This could turn into a problem as it can be determined in the following chapters. Thought it cannot be ascertained by a visual analysis of Figure 6, the data has a monthly granularity and begins in the year 1845, providing more than enough information to test the models.

3.3.1.3 Data Analyst

The Data Analyst is far more complex than the previous components of the layer. It contains several Jupyter Notebooks, divided into two main areas, the Individual Signal Analyses and the General Signal Analyses. Each of these areas provides a different set of methods for the comprehension of the data available in both databases.

3.3.1.3.1 Individual Signal Analysis

The Individual Signal Analysis is a composition of several Notebooks, each for every available signal. This allows for a more particular study of each individual signal by trying different transformations and manipulations. But even though each signal may have particular alterations, the tests performed on them are all the same.

First, every Notebook imports the desired signal and converts it from a CSV File to a Dataframe, as stated in previous chapters. Afterward, it uses the library pandas_summary [47], to retrieve basic information about the signal it is analyzing.
The results can give the user very useful information, such as the value of the mean, the standard deviation, or even if there is any missing data as shown in Figure 7. This is the first step in understanding if there are any changes that might be necessary to apply in the data.

Then come the first visual representations of the signal, that provide an easier and comprehensive analysis, using the Python libraries Matplotlib [48] and Seaborn [49]. The initial process applies a moving average\textsuperscript{16} to the signal in order to smooth some of its variations so that any trends underneath it may become visible.

\textsuperscript{16} A moving average is a time series constructed by taking averages of several sequential values of another time series [76].
Figure 8. Moving average plot of the Industrial Production of the US (cropped).

The vertical blue bars in Figure 8 represent the US recessions, this characteristic is presented in all Notebooks so that the user can cross check the evolution of the signal with the recessions and understand how the signal tends to react during and before these times. The blue line represents the signal itself, the yellow line represents the moving average of six months, the green of twelve months and the red eighteen months. The moving averages have a clear lag in comparison with the original signal but allow the user to understand its trend and behavior in a clearer manner.

The next plot supplied by the program is the annual and monthly variation of the signal, also cross-checked with the economic recessions.

Figure 9. Annual and monthly variation of the Industrial Production of the US (cropped).
As with the Figure 8, the blue vertical bars in Figure 9 are also the representation of the US recessions, and the yellow and blue lines are, the annual and monthly variations of the signal, respectively. This analysis permits the user to comprehend how the signal varies before and during the recessions. These last two processes were developed in the system to grant the user tools to perform a very immediate and preliminary examination of the indicator and decide if the signal is relevant for the study.

The next process of the Individual Signal Analysis is a graphic representation of the signal’s histogram, to discover the underlying distributions of the data and let the user infer on its distribution, outliers or skewness\textsuperscript{17}. This histogram is applied to the monthly variation and the yearly variation because the signal is not normalized. Using the absolute values of the signal would not lead to any useful conclusions since no real frequency could be ascertained.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{Industrial_Production_Histogram}
\caption{Frequency distribution of the Industrial Production of the US.}
\end{figure}

Figure 10 allows the user to comprehend where the means of the signal’s variations and their standard deviation lie, in a visual manner contrary to the analytic view of the pandas\textunderscore summary. In this graphic, the blue signal represents the monthly variation and the yellow the annual variation.

To conclude this process, a forecasting tool called Prophet\textsuperscript{[50]}, developed by Facebook Open Source, is applied. This tool does not work very well on data with more than weekly granularity, like months or years, but it supplies the user with a primary forecast of the signal in a chosen time period, and is a useful trend and seasonality graphic analyst\textsuperscript{[51]}.  

\textsuperscript{17} Skewness is the measure of asymmetry of the probability distribution.
Figure 11. Prophet forecast of the Industrial Production of the US (cropped).

Figure 11 characterizes the forecast of the Industrial Production of the US, where the black dots represent the signal, and the blue line the forecast produced by the tool. As it can be seen in the figure, the model doesn’t cope well with the monthly data granularity but allows the user to have a small glimpse of a possible forecast of the signal, even if it is not validated by any metric system.

Figure 12. Prophet trend and seasonality analysis of the Industrial Production of the US.

As in Figure 11, Figure 12 doesn’t have a title in its graphics since it is not a characteristic supplied by this tool. The first graphic of Figure 12 represents the trend of the entirety of the signal, and the second graphic represents the seasonality of the signal. These allow for an understanding of how the signal tends to behave through time and if it has any kind of seasonal repetition.
With all these capacities, the Individual Signal Analysis of the Data Analyst provides the user with the ability to understand what the signals obtained from the Macroeconomic Indicator Database represent and what their characteristics are. It also provides knowledge to the user for a possible choice of what indicators to use in the Classification Algorithm.

3.3.1.3.2 General Signal Analysis

The General Signal Analysis, like the Individual Signal Analysis, performs an evaluation of the signals provided by the databases from the Database Layer. But contrary to the later this program is destined to a cross-analysis between signals and is executed in a single Notebook.

Like the Individual Signal Analysis, it imports the signals from the CSV Files to a Dataframe and uses the pandas_summary library for a primary analytic examination. Afterward, the program normalizes the data in order to be able to make comparisons between the signals from Table 1, since they have very different units. To execute this normalization, the Standard Scaler from the scikit/sklearn (version 0.19.2) library [52], [53] is used, which removes the mean and scaling of the unit variance.

The main feature of this program is its correlation analysis, performed with the use of the missigno library [54].

![Heatmap of the correlations between the macroeconomic indicators signals.](image)

The values represented in Figure 13 are the level of correlation between the signal where:

- **Values:**
  - 1 – Highly correlated.
  - 0 – No correlation.
  - -1 – Negative correlation.

This tool is very useful in understanding which signals have high correlation and may overfit the models of the Classification Algorithm, which signals present contrary behaviors, and which are almost
completely uncorrelated. The choice of indicators presented in the following chapters mainly followed the results provided by this tool, with the assistance of the Individual Signal Analysis processes. The General Signal Analysis also offers more plotting capabilities for graphic representation of several signals at a time with the usage of the Matplotlib and Seaborn libraries.

3.3.2 Data Transformation Layer Implementation

The Data Transformation Layer receives the data from the Database Layer, and executes transformations and selections on the data. Like its predecessor, this layer has two branches, one that deals with the macroeconomic indicators’ data and the other that deals with the recession’s data. Each performs the transformations that are required to supply the necessary information to the Classification Layer so that it can complete the objectives of this thesis. This layer has only two major components, but they can be deconstructed in smaller parts as it is described in the following subchapters.

3.3.2.1 Indicator Selection & Transformation

The Indicator Selection & Transformation oversees the means to select which indicators will be used in the Classification Algorithm and transforms them in order to extract more information or make it usable by different models. So, to achieve this goal the Indicator Selection & Transformation can be separated into two phases, the Selection Phase and the Transformation Phase, each composed of two different components, one database, and one program.

![Diagram of Indicator Selection & Transformation](image)

Figure 14. Indicator Selection & Transformation internal architecture.

Figure 14 exemplifies the internal structure of the Indicator Selection & Transformation. The data from the Macroeconomic Indicator Database is delivered to the Selection Phase into the Indicator Selector program where the user can select which indicators to use in the Classification Algorithm, and automatically set them to the same granularity. Afterward, this data is sent to the Selected Indicator Database that stores the previous information and makes it available for use. The Transformation
Phase receives the information in the Indicator Transformation program and uses it to make the desired transformations to the data and finally sends the transformed information to the Selected & Transformed Database for storage. This modular behavior was implemented for expansion reasons, making the system able to create several Selected Indicator Databases and Selected & Transformed Databases to test different combinations of indicators and transformations.

### 3.3.2.1.1 Selection Phase

In this phase the Indicator Selector receives the data from the Macroeconomic Indicator Database and as in the previous phase, begins to convert it from CSV Files to a Dataframe. It applies the pandas_summary in case any information about the indicators is required. For the next step, the user chooses which indicator it wants to save and which granularity to use.

In this work two granularities were tested, the monthly and weekly. However, the weekly granularity was not properly validated, for different transformations were required, and to simplify the process of the exploratory study, the efforts were centered on the monthly granularity. Even so, both Notebooks containing the program for each granularity were created. Having chosen the monthly granularity, used by the majority of the indicators available, the next step was to choose which indicators to use. Using Figure 13, and removing the indicators with the biggest correlation, a shorter set of indicators was chosen while still trying to retain some indicators of each of the areas of interest such as labor, money availability, purchasing power, production, and trust.

<table>
<thead>
<tr>
<th>Selected Indicator Name</th>
<th>Code</th>
<th>Granularity</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Hourly Earnings</td>
<td>AHETPI</td>
<td>Monthly</td>
<td>Percent, Seasonally Adjusted</td>
</tr>
<tr>
<td>Industrial Production</td>
<td>INDPRO</td>
<td>Monthly</td>
<td>Index 2012=100, Seasonally Adjusted</td>
</tr>
<tr>
<td>Initial Claims Unemployment</td>
<td>IC4WSA</td>
<td>Weekly</td>
<td>Number, Seasonally Adjusted</td>
</tr>
<tr>
<td>New Private Housing Building</td>
<td>PERMIT</td>
<td>Monthly</td>
<td>Thousands of Units, Seasonally Adjusted Annual Rate</td>
</tr>
<tr>
<td>Production Manufacter Index</td>
<td>PMI</td>
<td>Monthly</td>
<td>Percent: above 50% expected growth and vice-versa</td>
</tr>
<tr>
<td>Real Money Stock</td>
<td>M2REAL</td>
<td>Monthly</td>
<td>Billions of 1882-84 Dollars, Seasonally Adjusted</td>
</tr>
<tr>
<td>Yield Curve</td>
<td>T10Y2Y</td>
<td>Daily</td>
<td>Percent, Not Seasonally Adjusted</td>
</tr>
</tbody>
</table>

Two simple conclusions can be ascertained from the chosen dataset of Table 2. The first conclusion is that not all the indicators have the same granularity, which will still be solved in this program. The second conclusion is that the dataset became far more reduced. For this second problem, the Transformation Phase presents itself as a possible way to increase the information of the dataset.

Even though the quantity of information was quite reduced for the purpose of an exploratory search, if the results presented in the Validation Layer are favorable, then the postulation that if more information is added the better the results will be, strengthens this work validity.

To solve the problem of different granularities, a method of Pandas is called, the resample [55]. This method changes the granularity of all indicators to a newly selected frequency, in this case, monthly. To achieve its goal, the method has several processes available, but for this study, only the mean(), max(), min() and std() were used. Since there are several weeks and days in a month, the resample uses these methods to choose the mean of every week/day of each month, the maximum value, the minimum value, or the standard deviation of the mean values, respectively.
Figure 15. Dataframe of the selected and resampled macroeconomic indicators with the mean, maximum, minimum and standard deviation values (cropped).

The program finally sends to the Selected Indicator Database five different datasets, each with a particular method of the resample, and a final one, Figure 15, with all the methods.

3.3.2.1.2 Transformation Phase

The Transformation Phase begins by retrieving the data from the Selected Indicator Database and sending it to the Indicator Transformation component. Here the data will be changed according to the users’ desires. Several Notebooks can be created here, each for one kind of transformation, guaranteeing that the transformations don’t overlap. For this thesis, only a memory adder with normalization was created, as a possible transformation, but the system was designed so that more transformation can be added.

To surpass the problem of a shorter dataset, some strategies can be applied to extend the amount of data available. This work used a memory adder that adds past values at the current point of time.

Figure 16. Memory adder conceptual diagram.
As can be seen in Figure 16, the *Indicator Transformation* retrieves the data from the previous months and adds them to the current month, this way multiplying the amount of data available. The program starts by shorting the data available so that it starts and ends at the same dates, using a method specifically created for this thesis called *shorting_dataframe*. This is required so that the following actions can work. After the dataset is shortened the *memory_adder* function is called to execute the concept explored in Figure 16. This method was also created purposefully for this work.

In this thesis only two datasets with memory were created, the Three Month Memory and the Six Month Memory, each of them stored into the *Selected & Transformed Database* together with the dataset with no memory. Additionally, these datasets were also normalized so that one of the models used in the *Classification Algorithm* attained better results. Consequently, the *Selected & Transformed Database* received six datasets, three normalized and three with absolute values.

### 3.3.2.2 Lag Adder

The *Lag Adder*, like the *Indicator Selection & Transformation*, can also be deconstructed into different components, but since only one recession can be used at a time it does not require a *Selection Phase*.

![Figure 17. Lag Adder internal architecture.](image)

Figure 17 demonstrates that the *Lag Adder* has a very simple composition, with one program, the *Lag Adder* and one database, the *Lagged Recession Database*. The program lets the user choose which recession to use, and the amount of lag necessary for the analysis. For this thesis, the only recession used was the US recession and the lags added were of six, twelve and eighteen months.

The idea behind adding lag to the signal, was to “trick” the ML algorithms into “thinking” that a recession started before it actually happen, by extending the signal into the past, as will be seen ahead. This way, instead of making out of the bag predictions with the models, they are trained to look for early signals of a recession by extending the recession signal a certain amount of months and making the predictions accordingly.

The *Lag Adder* has only two major functions apart from choosing the recession, adding the lag and resampling the data so that it can be used in the *Classification Algorithm* with data from different granularities stored in the *Selected & Transformed Database*. The program starts by adding lag to the signal. At first, it may sound like an easy mission to accomplish, but for high levels of lag, a simple shift method may not suffice since the original signal and the shifted one may not become continuous.
As can be seen in Figure 18, if the original signal and the shifted one were to be united, they wouldn’t form a continuous lagged signal as it should happen. Therefore, a more ingenious solution had to be developed. The main goal was to discover when the start date of every recession was and then choose the previous month according to the desired lag and extend the signal from that point until the end date of the original recession.

The solution was to use the logical operations of Figure 19, and convert the original signal to a signal with only the first dates of the recession. The next objective was finding the previous points, according to the desired lags, and forward fill the points until the last dates of the recession’s original signal. This
process was developed in the method shift_fill(), created specifically for the purpose, which used the `monthdelta` library [56], and various methods of the NumPy library [57] to achieve its results, like `logical_and`, `logical_xor`, and `logical_or`.

Figure 20. US recession signal with six, twelve and eighteen months lag (cropped).

The result presented in Figure 20 demonstrates the lagged signal ready for use in the Classification Algorithm, but before it is stored in the Lagged Recession Database the program still performs the required resample of the data as executed in the Indicator Selector, but only by using the mean value. All the signals with different granularities are then stored in the Lagged Recession Database.

3.3.3 Classification Layer Implementation

The Classification Layer can also be deconstructed in several components, the Classification Algorithm, the most important part of this work solution, the Model Prediction Database and the Models’ Average.

Figure 21. Classification Algorithm internal architecture.
As can be seen, in Figure 21 the Classification Algorithm receives the data from the Selected & Transformed Database and Lagged Recession Database, and starts its process, running the ML models and returning the models’ predictions to the Model Prediction Database and to the Validation Layer. The data stored in the Model Prediction Database is used by the Models’ Average, which does an average of the predicted results of all models and sends the results also to the Validation Layer.

3.3.3.1 Classification Algorithm

Following the background study of the binary classification models, two main approaches were taken in this component, a linear and a nonlinear. For the linear approach the Logistic Regression model was chosen, and for the nonlinear approach two models were chosen, the Random Forest Classifier and the XGBoost. These were selected for their reliable binary classification forecasting, and their different characteristics supported by most literature.

In all three models, the macroeconomic indicator signals and the US recession signal with lag are used. As stated before, to fulfill the objective of this thesis, i.e. accomplishing an exploratory study of economic recessions and also being able to predict them, these transformations to the signals are very important. The macroeconomic signal transformations try to diminish the problems of using a short dataset, but also provide some economic relevant information on the importance of the previous data points for economic recession detection. As for the lagged signal of the US recessions, not only does it allow for early recession predictions, but it also provides some information on the probability and reliability of those actions. For each model, every combination of indicators’ memories and recession lags, there is a different Notebook, resulting in this component having the greatest number of Notebooks of the entire system, one for each situation.

With the implementation of these ML algorithm solutions, the problem of poor performance due to overfitting or underfitting usually arises. Since the dataset has very few information, it is likely for it to happen, even with the usage of Figure 13 to remove the already correlated signals that would probably produce overfitted analyses. To counteract this probability a cross-validation system was implemented in the Classification Algorithm.

3.3.3.1.1 Cross-Validation

The basic principle behind a cross-validation system is to break the data into training sets and test sets. By doing this, the models don’t fit the entirety of the data in one single turn, preventing them from over or under fitting. There are several methods for achieving this, like k-fold cross-validation or leave-one-out cross-validation. These methods break the data into randomly defined parts and cross-check them with other sets of data. In the leave-one-out procedure, every possible combination of train and test datasets are tested, and in the k-fold the data is broken in k parts, or folds, and then trained in k-1 sets and tested on the last until all sets were tested. There is no point in exploring this further because these

---

18 Overfitting is the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably [77].

19 Underfitting is the production of an analysis that does not corresponds closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably [78].
systems either don’t work or can be prejudicial for a time series analysis since they train and test the data in a random fashion, and in a time series analysis, the order of the data cannot be discarded.

Therefore, a different type of cross-validation must be used. Following the principle of breaking the data into test and training sets and using the concept of the k-fold cross-validation for breaking the set in a specific number of folds, the best way to solve the problem becomes the usage of forward chaining.

Figure 22. Forward chaining cross-validation.

Figure 22 presents the concept behind the forward chaining cross-validation, where the dataset is split into a training set and a testing one. The algorithm starts by attributing an amount of the first available data to the training set and then a continuous specific amount of data to the testing set. Afterward, the algorithm uses the training data to fit the model and predict over the test data, even permitting, if the user desires it, a validation phase. After the prediction is concluded, the training data increases its size until it reaches the sum of its original size and the testing size. The new test set is continuously added to the new training data, and this is done indefinitely until all the data has been passed by the training set. To accomplish this algorithm all models use the TimeSeriesSplit from scikit-learn library [52], [58], which then splits the time series dataframe into divisions like the ones showed in Figure 22.

3.3.3.1.2 Linear Approach (Logistic Regression)

The linear approach present in the Classification Algorithm uses the Logistic Regression model to make its forecast. Contrary to its nonlinear counterparts, the Logistic Regression model has some different approaches to its implementation.

First, it imports the datasets from both the Selected & Transformed Database and the Lagged Recession Database and begins by matching their end dates with a specifically produced method, the equal_dataframe(). This method equals the last dates of both datasets so that they can be used in the model. Afterward, both dataframes are sent to the method that joins both the model and the cross-validation algorithm, the time_series_cross_validation_logistic_regression(). This method receives not only the dataframes, but also the numbers of splits or folds to use in the cross-validation algorithm. The process then rearranges the data, to have the same size, and uses the cross-validation to fit it and predict new values. This new predicted data is derived from the predictions made over the testing period in each cross-validation iteration, and are then stored into a new dataset. After all the iterations of the cross-validation are completed, the new dataset will be returned by the function. To use the Logistic
Regression model in this function, it is imported the LogisticRegression library [59], and for the forecasting of the signal, it is used the predict_proba() method instead of the predict(). The predict_proba() method returns the probability that one of the classifications is true, in this case, it returns the probability that the recession is true. More importantly than the knowledge of the existence of the recession, it is important to know how “likely” it is for the recession to happen. Also, the LogisticRegression model creator received an attribute to deal with the imbalance of the recession signal (class_weight='balanced'), that analyzes the percentage of each class in the signal and balances their weights.

Upon receiving the results, the program prepares the returned data to be later used in the validation process. Once the data is prepared, a new method is called for plotting the results of the model's prediction, the show_forecast(). This functions was also specifically prepared for this thesis with the usage of the Plotly library [60].

Once the plots become available to the user for a first visual analysis of the results, the Notebook then executes its last method, also created for this work, the evaluation_model(). This method prints on the screen three elements required for the validation of the model, the confusion matrix, the classification report, and the ROC curve. To retrieve this information it uses the previously prepared data and the classification_report, confusion_matrix, roc_auc_score, roc_curve from the scikit/learn library [52], [61]–[64] and the Matplotlib library [48] for the plots. The program then finally sends the resulting predictions to the Model Prediction Database for storage.

Contrary to the nonlinear models, the Logistical Regression benefited from the normalized dataset instead of the dataset with absolute values and was therefore implemented with them. Also, the decision on the number of folds used in the cross-validation was chosen through an engineering process of trial and error. The final choice of folds considered the establishment of a first training set that would have two recessions and three recessions in the data afterward.

3.3.3.1.3 Nonlinear Approach (Random Forest Classifier & XGBoost)

The nonlinear approach presents some similarities with the linear approach, but some parts of its implementation vary. The nonlinear approach has two different models to test, the Random Forest Classifier and the XGBoost, a nonlinear bootstrapping and boosting algorithm, respectively. Both these models present similar Notebooks, differing only in the creation of the model, so for simplicity reasons, the explanation of the implementation for both will be the same.

The program starts by importing the datasets from both the Selected & Transformed Database and the Lagged Recession Database, and then matching their end dates, using the equal_dataframe() method, exactly like in the linear approach. It then, as in the Logistic Regression, calls the respective models with a method that combines the cross-validation algorithm and each algorithm, also receiving the number of splits and estimators. The time_series_cross_validation_random_forest() and the time_series_cross_validation_xgboost() are the respective methods of the Random Forest Classifier and the XGBoost models. The only difference from the previous method is that these use two different libraries, the RandomForestClassifier and the XGBClassifier [65], [66], and the model creator requires
an estimator argument. Also, as with the Logistic Regression, the values of the folds and the estimators were defined by a trial and error engineering process. The number of folds remained the same as in the linear approach, and the estimators were set at one hundred for a stronger prediction without questioning the computational power of the machine. Afterward, the programs also prepare the data for validation purposes and use the `show_forecast()` method to plot the results.

At this point, both approaches start to differ since the nonlinear approach presents a feature that is somehow relevant for an economic analysis but is not present in the Logistic Regression. The nonlinear models can call the attribute `feature_importances_` that has the importance of each signal in the production of its results. The results are presented in a percentage form and are later plotted with the use of the `Matplotlib` and `Seaborn` libraries [48], [49]. The low percentage indicators can be removed with the usage of the purposely made function `indicator_removal()`, that removes the indicators below a certain percentage value. Afterward, the model function can be called and plotted again with fewer indicators. This may not present value to increase the predictability power of the model but may present some economically interesting information to the user.

Finally, and like with the linear approach, the resulting predictions are sent to the `Model Prediction Database`, and the method `evaluation_model()` is called and returns the metrics' results to be used in the `Validation Layer`.

### 3.3.3.2 Model Prediction Database

The `Model Prediction Database` receives the resulting prediction probabilities from all the models and stores them accordingly with the lag of the US recession signal so that it can be easily used by the `Models' Average`.

Figure 23 represents one of the tables of the database, where all the predictions of the models are stored. The acronyms LR stand for `Logistic Regression`, RFC for `Random Forest Classifier` and XGB for `XGBoost`. This database can also be used to perform other tests or transformations that are not noted in this thesis.
3.3.3.3 Models’ Average

The Models’ Average component is composed of four Notebooks, each corresponding to a different level of lag of the US recession signal.

The programs all work in the same way. First, they retrieve the information from the Model Prediction Database and join them in a temporary dataset. This dataset is then deconstructed into three different ones, one for each memory transformation, i.e. a dataset for the three month transformations, the six month transformations and for the data with no transformation.

After this division, each dataset runs an average of the of its columns, each representing a model.

![Table]

<table>
<thead>
<tr>
<th>DATE</th>
<th>LR_Six_Memory</th>
<th>RFC_Six_Memory</th>
<th>XGB_Six_Memory</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1985-02-01</td>
<td>0.002075</td>
<td>0.01</td>
<td>0.005889</td>
<td>0.005938</td>
</tr>
<tr>
<td>1985-03-01</td>
<td>0.002093</td>
<td>0.00</td>
<td>0.005889</td>
<td>0.002601</td>
</tr>
<tr>
<td>1985-04-01</td>
<td>0.002587</td>
<td>0.01</td>
<td>0.010254</td>
<td>0.007614</td>
</tr>
<tr>
<td>1985-05-01</td>
<td>0.003602</td>
<td>0.00</td>
<td>0.010991</td>
<td>0.004864</td>
</tr>
<tr>
<td>1985-06-01</td>
<td>0.003849</td>
<td>0.02</td>
<td>0.013035</td>
<td>0.012295</td>
</tr>
<tr>
<td>1985-07-01</td>
<td>0.003657</td>
<td>0.04</td>
<td>0.010704</td>
<td>0.018120</td>
</tr>
<tr>
<td>1985-08-01</td>
<td>0.002704</td>
<td>0.03</td>
<td>0.008772</td>
<td>0.014159</td>
</tr>
<tr>
<td>1985-09-01</td>
<td>0.001660</td>
<td>0.03</td>
<td>0.010704</td>
<td>0.014121</td>
</tr>
</tbody>
</table>

Figure 24. Dataset produced by the averaging of the predictions of the three models.

Figure 24 presents the result of the averaging of the models, in this case, the ones with the six month transformation. Afterward, the program, like in the Classification Algorithm, prepares the data for the performance metrics calculations and for plotting. It also uses the functions show_forecast() and evaluation_model() to produce the results to be sent to the Validation Layer, where the user can now analyze not only the predictions of each model but also the average of all of them.

3.3.4 Validation Layer Implementation

As stated before, the Validation Layer has no real components and uses the results from the show_forecast() and evaluation_model() methods from the previous subchapters to give information to the user so that if necessary, some changes can be made to the models, or to use the result for their own purposes. In the next chapter, the results sent to the Validation Layer will be discussed and analyzed in order to understand if the system proposed in this thesis has any validity.
4

Results & System Validation
4.1 Metrics

To evaluate the quality of the different approaches, some global metrics must be applied. These metrics measure the performance of the several models and their variations, following the state of the art for a time series binary classification analysis. For the following examination, the resulting signal provided by the models’ probability prediction is averaged and classified as a recession or not, i.e. the result value is one (recession), for recession probabilities above 50%, and zero (no recession) otherwise.

4.1.1 Confusion Matrix

The Confusion Matrix is a table composed of two rows and two columns that cross the results of the prediction model and the targeted values.

Table 3. Confusion matrix table for the US recession prediction.

<table>
<thead>
<tr>
<th>Predicted Recession</th>
<th>Actual Recession</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True Positive</td>
</tr>
<tr>
<td></td>
<td>False Positive</td>
</tr>
<tr>
<td>False</td>
<td>False Negative</td>
</tr>
<tr>
<td></td>
<td>True Negative</td>
</tr>
</tbody>
</table>

Table 3 represents the confusion matrix of this thesis and presents the relation between the predicted results and targeted results. From this crossing, four concepts arise, the True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN).

- **True Positive**: Is an outcome where the model correctly predicts the positive class.
- **False Positive**: Is an outcome where the model incorrectly predicts the positive class.
- **False Negative**: Is an outcome where the model incorrectly predicts the negative class.
- **True Negative**: Is an outcome where the model correctly predicts the negative class.

In this thesis, the positive class, represents the case of a recession really happening, and the negative class, the opposite. The matrix of Table 3, returns the amount of each of these concepts present in the results of each model.

4.1.2 Precision

The Precision metric represents the ratio between the correctly predicted positive observations and the total of positive observations. This metric answers the question, “of all recessions labeled as true, how many actually happened?”.

\[
\text{Precision} = \frac{TP}{TP + FP}
\]  

(13)
Equation 13 presents the calculus formula to assess this ratio where a return of 1 represents a good quality model and 0 a bad quality model.

4.1.3 Recall

The Recall, or sometimes called Sensitivity, metric represents the ratio between the correctly predicted positive observations and all the observations of that class. This metric answers the question “of all the recessions that truly happened, how many did we label?”, where a return of 1 represents a good quality model and 0 a bad quality model.

\[
\text{Recall} = \frac{TP}{TP + FN}
\] (14)

Equation 14 presents the calculus formula to assess this ratio where a return of 1 represents a good quality model and 0 a bad quality model.

4.1.4 F1-Score

The F1-Score metric represents the weighted average of the Precision and Recall metrics. This metric takes both the False Positives and the False Negatives into account. It may not be easy to understand, but this metric acts as a kind of accuracy\textsuperscript{20}, but is better suited for unbalanced datasets. In view of these assertions, this metric presents itself as the best for verifying the quality of this thesis models’ predictions.

\[
F1 - Score = \frac{2TP}{2TP + FP + FN}
\] (15)

Equation 15 presents the calculus formula to assess this ratio, where a return of 1 represents a good quality model and 0 a bad quality model.

4.1.5 ROC Curve

The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the power of a binary classification system with different rates of TP and FP. The ROC curve is created by plotting the Recall against the False Alarm Rate\textsuperscript{21} and as the curve approaches the left and the top side of the graphic the more accurate the test is. By contrary, if the curve approaches the 45° degree mark the less accurate it is.

\textsuperscript{20} Accuracy is the ratio between the correctly predicted observations and the total number of observations. \[\frac{TP + TN}{TP + FP + FN + TN}\]

\textsuperscript{21} The False Alarm Rate is rate of false positives among all the cases that should be negative. \[\frac{FP}{FP + TN}\]
Figure 25. ROC curve of a tested model not used in the thesis.

Figure 25 presents an example of an ROC curve that appears to have good accuracy, since the curve is somehow near to the top left of the plot. But just by analyzing the graphic no conclusion can be reached if compared with another model. To solve this issue another metric can be used, the Area Under the Curve (AUC).

4.1.5.1 Area Under the Curve (AUC)

The AUC measures the area beneath the ROC curve by making an integral of the signal. If the AUC returns near the value 1, then it represents a perfect test, but if it is near 0.5, it is a worthless one. The following ranges are a crude way of assessing the strength of the model:

- 1 - 0.9 = Excellent
- 0.9 – 0.8 = Good
- 0.8 – 0.7 = Fair
- 0.7 – 0.6 = Poor
- 0.6 – 0.5 = Fail
So, applying this method to Figure 25 we have:

![ROC curve](image)

Figure 26. ROC curve with AUC of a tested model not used in the thesis.

Now it becomes clear that even though the curve is near the top left of the plot, it can only be considered a fair model. This metric system allows for a more analytic comparison between models, and since it tests for all probability thresholds, i.e. the threshold defined by the user to differentiate above which probability value it is considered a recession or not, it can improve on the previous metrics.

### 4.2 Case Studies

#### 4.2.1 Methodology

The results of this thesis will be benchmarked with the dates of the US recessions provided by the NBER, the respective lags, and validated with the previously stated metrics. The proposed solutions expect to fit the signal for US recessions, with and without lags, in a clear manner, so that it can be possible to visually analyze and comprehend the results. The outcomes are also expected to produce good metric values to follow the visual representations and produce intuitive and easy to analyze responses. Intuitively the values of these metrics are expected to improve with the addition of memory and deteriorate with the increase of the US recession Lag. The most positive results, in a best-case scenario, would be to beat the benchmark and the most recent state of the art results, providing a clear and reliable US recession forecaster.

This work is divided into four case studies to compare their different solutions on achieving the goal of detecting and predicting a US recession. In all case studies, four different US recession lags are used, and three different input signals, two of which are transformed.
• **Case Study A:**
A linear approach is taken, using the *Logistic Regression* model, normalized data, and monthly granularity.

• **Case Study B:**
A nonlinear approach is taken, using the *Random Forest Classifier* model, absolute value data, and monthly granularity.

• **Case Study C:**
A nonlinear approach is taken, using the *XGBoost* model, absolute value data, and monthly granularity.

• **Case Study D:**
A model averaging approach is taken, where the models’ prediction results used in the previous case studies are averaged by a simple, same weighted average.

Even though the thesis’ system provides a *Confusion Matrix* for the user, in this analysis it will not be used since it doesn’t add any relevant information for this study.

### 4.2.2 Case Study A – Linear Approach, the Logistic Regression

This case study follows a linear approach to the modeling and uses normalized data to achieve it. For this study several US recession lags were used, the six months lag, twelve months lag, eighteen months lag and the signal with no lag. This allows forecasting in four different time spans and at maximum, an attempt to detect a recession one year and a half before of time. Also, in this case study, three types of input signals were used, the three month memory transformation, the six month transformation and the original signal with no transformation.

To perform this study some parameters had to be set in the models’ construction and preparation.

Table 4. Case study A parameter configuration.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study Period</td>
<td>1975/08/01-2018/05/01</td>
</tr>
<tr>
<td>Train Period</td>
<td>Begins at 1975/08/01</td>
</tr>
<tr>
<td>Test Period</td>
<td>Begins at 1995/02/01</td>
</tr>
<tr>
<td>Nº of Splits</td>
<td>100</td>
</tr>
<tr>
<td>Class Weight</td>
<td>Balanced</td>
</tr>
</tbody>
</table>

Table 4 shows the main parameters set in this case study. The *Train Period* and the *Test Period* were automatically chosen by the *TimeSeriesSplit* [44], using the *Nº of Splits*. Using Figure 22 it can be comprehended that the *Train Period* increases its size for each iteration, starting with an initial size of approximately ten years and finishing four months before the *Study Period*. Also, as a result of the *TimeSeriesSplit* and the *Nº of Splits*, the *Test Period* has a four-month size in each iteration and finishes in the same date as the *Study Period*. The value for the *Nº of Splits* was chosen so that the initial training set included two recessions. The Logistic Regression model is not automatically ready for every type of
data, so it improves its results with normalized data and must receive an argument defining how the targeted information is balanced. The *Class Weight* argument deals with this setup and inserting the ‘Balanced’ specification defines that the model will search for the balance in the targeted signal and define new weights for the predictions.

Having defined the configuration of the case study, the next step is to review the previously defined metrics.

Table 5. *Classification Report* results for the *Logistic Regression* Model.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Month Lag</td>
<td>Six Month Lag</td>
</tr>
<tr>
<td></td>
<td>precision</td>
<td>recall</td>
</tr>
<tr>
<td>No Memory</td>
<td>0</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>0.74</td>
</tr>
<tr>
<td>Three Month Memory</td>
<td>0</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>0.95</td>
</tr>
<tr>
<td>Six Month Memory</td>
<td>0</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 5 represents the results of all the iterations of the Logistic Regression model. The blue column represents the memory transformations, and the green columns and rows indicate the values and transformations of the recession signal, respectively. The Y green column represents the classes tested in the model, in this case, the class 0 (no recession) and the class 1 (recession), and also the average of both classes. Table 5 is also separated by the lags of the US recession signal, and each of these separations is composed of four different metrics. The *Precision*, *Recall*, and *F1-Score* were already reviewed in the previous subchapter, the *Support* metric defines the number of occurrences of each class.

By performing an analysis of the *F1-Score* of class 1 in Table 5, it can be noted that mainly the model improves its quality with the increase of memory and loses quality by adding lag, except for the eighteen month lag, which surprisingly improves on the results of the no month lag. Since the results present some doubt, that can be possibly explained by the chosen probability threshold to consider a recession (threshold = 50%), then the analysis of the AUC values are required to draw conclusions.

Table 6. AUC results from the ROC curves of the *Logistic Regression* Model.

<table>
<thead>
<tr>
<th>X</th>
<th>Logistic Regression</th>
<th>Area Under the Curve (AUC)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Month Lag</td>
<td>Six Month Lag</td>
</tr>
<tr>
<td>No Memory</td>
<td>0.89</td>
<td>0.81</td>
</tr>
<tr>
<td>Three Month Memory</td>
<td>0.9</td>
<td>0.82</td>
</tr>
<tr>
<td>Six Month Memory</td>
<td>0.89</td>
<td>0.83</td>
</tr>
</tbody>
</table>
Now using Table 6, it becomes clear that the model behaves exactly as expected, and that the probability thresholds must be modified for some of the instances to improve their results. From the analysis of Table 6, the most interesting result is provided by the six-month memory model with eighteen months of anticipation.

Figure 27. The probability of a US recession happening with eighteen months of lag and six months of memory calculated by a Logistic Regression algorithm.

Figure 27 displays graphically the probability of a recession happening with an eighteen month antecedence, where the green lines represent the recession signal lagged by eighteen months, the red the actual recessions and the blue, the probability of a recession occurring calculated by the model.

For visual comparison between case studies, another model result can be plotted. Following an average lag with full memory approach, the models that will be compared in all case studies will have twelve months of lag and six months of memory.

Figure 28. The probability of a US recession happening with twelve months of lag and six months of memory calculated by a Logistic Regression algorithm.
Figure 28 presents very similar results to those in Figure 27 as was expected, since their AUC values presented in Table 6 are very similar. This plot will be of further assistance in a visual comparison between models.

The Logistic Regression model presents good results against the NBER benchmark but also against the state of the art, validating it as a good option for this thesis.

4.2.3 Case Study B – Nonlinear Approach, the Random Forest Classifier

This case study follows a nonlinear approach and uses the Random Forest Classifier model to achieve it. The data, contrary to Case Study A, uses absolute values, but like the previous case study, it uses all the US recession lags and all the memory transformations provided by the system.

The Random Forest Classifier model has a very similar construction and parameter semblance with the Case Study A.

Table 7. Case study B parameter configuration.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study Period</td>
<td>1976/08/01-2018/05/01</td>
</tr>
<tr>
<td>Train Period</td>
<td>Begins at 1976/08/01</td>
</tr>
<tr>
<td>Test Period</td>
<td>Begins at 1985/02/01</td>
</tr>
<tr>
<td>Nº of Splits</td>
<td>100</td>
</tr>
<tr>
<td>Nº of Estimators</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 7 shows that Case Study A and B share the same Study Period, Train Period, Test Period and Nº of Splits, but it does not have a Class Weight parameter. This is due to the ability of the Random Forest Classifier to deal with unbalanced datasets. But on the other hand, it has a new parameter, the Nº of Estimators, that defines the number of trees in each forest. The value 100 was chosen because, even though it is the defaulted value, it presents a good amount of information with reduced computational power requirements.

Concluding the configuration analysis, the next step is to review the Classification Report results.
Like in Table 5, Table 8 also represents the results of all the iterations of the model, in this case, the Random Forest Classifier. The parameters of Table 8 are the same as the ones from Table 5, and allow for an analytic analysis of the model.

From Table 8 some conclusions can be ascertained. First, the model didn’t improve with the increase of memory as was expected, decreasing in many cases. With the increase of the lag, the model behaved exactly the opposite of what was expected, increasing its power with the lag increment. Like in the previous case, many of these results may have happened due to the chosen threshold, so another analysis is required.

Table 9, in this case, solved the memory problem improvement for the case with no lag, but didn’t for the other lags. Even though the values didn’t behave as expected, they are very similar, and little conclusion can be derived by it if not that the increase in memory does not help this model. As for the lag, Table 9 didn’t improve the results from Table 8 and kept increasing with the lag. This can maybe be explained by the increasing size of the targeted outcome.

Even though the results don’t behave as expected, all of them present good results. The model with three months of memory and eighteen months of lag presents the best results, and so is therefore one of the most interesting to observe.
Figure 29. The probability of a US recession happening with eighteen months of lag and three months of memory calculated by a Random Forest Classifier algorithm.

Figure 29 graphically represents the probability of a recession happening with an eighteen month antecedence and three months of memory and follows the same principles as Figure 27. It fits very well with the recessions even though it appears more volatile than its Case Study A counterparts. For a more valid comparison between them, it is required to analyze the results of the twelve-month lag model with six months of memory.

Figure 30. The probability of a US recession happening with twelve months of lag and six months of memory calculated by a Random Forest Classifier algorithm.

Figure 30, like Figure 29, appears to have more volatile results than in Case Study A, but on the other hand, it fits the recessions better and doesn’t raise such higher FP probabilities. Both case studies provide very good and reliable results, but clearly show that they work in a very different manner providing distinct results.

This case study also adds a feature not presented in Case Study A, that is the feature importance in each model. Since they are non-essential for the results verification only the feature importance of the eighteen month lag with no memory model will be displayed, for its simple analysis.
Figure 31. Feature importance of the Random Forest Classifier with eighteen months of lag and no memory.

Figure 31 displays the feature importance of the Random Forest Classifier model with no memory and twelve months of lag, where each macroeconomic indicator is displayed by order of importance in the construction of the predictions that resulted from the model. In conclusion Case Study B also presents good results against the NBER dates and the state of the art.

4.2.4 Case Study C – Nonlinear Approach, the XGBoost

This case study, like the Case Study B, presents a nonlinear approach, but instead of using the Random Forest Classifier model, it uses the XGBoost model. Also, like the Case Study B, it also uses absolute values and all the US recession lags, and all the memory transformations provided by the system.

The XGBoost model works in a very similar way as the Random Forest Classifier model, and therefore uses the same parameter configuration as described in Table 7. Both these models have a very similar constitution and processes, but provide very different results.

Table 10. Classification Report results for the XGBoost Model.
Table 10, like its predecessors, Table 5 and Table 8, also describes the results returned by the Classification Report of the model, in this case, the XGBoost. Unfortunately, unlike in its predecessor case studies, Table 10 does not allow for a comprehensive understanding of the quality of the several iterations. It is only possible to understand that after the six month lag all the models lose their predictive power with the increase in the lag. In all the other cases the variations don’t appear to follow a specific rule. The zero-month lag presents the most disparities between the rest of the lags, offering the worst results. This, once again, may be the result of having the same probability thresholds. Then the analysis of the AUC is still required.

Table 11. AUC results from the ROC curves of the XGBoost Model.

<table>
<thead>
<tr>
<th>X</th>
<th>XGBoost - Area Under the Curve (AUC)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Month Lag</td>
</tr>
<tr>
<td>No Memory</td>
<td>0.77</td>
</tr>
<tr>
<td>Three Month Memory</td>
<td>0.82</td>
</tr>
<tr>
<td>Six Month Memory</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Unlike the results from Table 10, but also from Table 9, the results of Table 11 show that most of the iterations decrease their predictive power with the increase in the lag. But on the other hand, they tend to decrease their predictive power with the increase in the memory, raising the point that the XGBoost model does not deal well with the increase in the memory of the signal.

The results of this case study are not as good as the ones from Case Study B but follow a more expected path. As for Case Study A, it loses in most of the iterations, but at the six month lag presents better results. Therefore, the six-month lag with no memory XGBoost model appears to be the most interesting to analyze, since it beat Case Study A, and almost does the same for Case Study B.
Figure 32. The probability of a US recession happening with six months of lag and no memory calculated by an XGBoost algorithm.

Figure 32 displays the graphical probability of a recession happening with a six month antecedence and six months of memory. The results are very well fitted for the economic recession, but contrary to its predecessors of Case Study A and B, its FP present a very “spiky” shape, sharply raising the probability of a recession in a very short amount of time. But, as before, for a better comparison it is necessary required to plot the twelve-month lag model with six months of memory.

Figure 33. The probability of a US recession happening with twelve months of lag and six months of memory calculated by an XGBoost algorithm.

Figure 33 presents much more volatile results than the ones from the previous case studies, as expected, since the results from Table 11 aren’t as strong as their previous counterparts. Even so, the results are also very well fitted to the US recessions, and they follow the same dates as Case Study A and B for its FP.
Also, like in Case Study B, the XGBoost also provides the property to display the feature importance of its models’ results.

Figure 34. Feature importance of the XGBoost with six months of lag and no memory

Figure 34 presents a very different order of priorities than Figure 31. This places the possibility that for different lags, memories and algorithms, the importance of the macroeconomic indicators changes. According to the economic background, this idea makes sense since some indicators affect different areas of the economy, and different areas of the economy tend to react first to recessions. In conclusion, the XGBoost model provides very good results but they have to be analyzed in a cautionary manner due to their very rapid growths and declines in probability.

4.2.5 Case Study D – Model Averaging Approach

The final case study of this thesis is by far the most different of all since it isn’t based on a single model, but the average of the three previous ones. So, it uses the joint power of the linear and nonlinear approach to try and compensate for the flaws of each model.

Case Study D uses the results from the previous models, and in this specific case study, the other models follow the configurations of Table 4 and Table 7, for the Logistic Regression and the Random Forest Classifier and XGBoost respectively. Having settled the configurations of this study the next step is to analyze the results of the Classification Report.
Table 12. Classification Report results for the Models’ Average.

<table>
<thead>
<tr>
<th>X</th>
<th>No Month Lag</th>
<th>Six Month Lag</th>
<th>Twelve Month Lag</th>
<th>Eighteen Month Lag</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>precision</td>
<td>recall</td>
<td>F1 score</td>
<td>support</td>
</tr>
<tr>
<td>No Memory</td>
<td>precision</td>
<td>recall</td>
<td>F1 score</td>
<td>support</td>
</tr>
<tr>
<td>0</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>366</td>
</tr>
<tr>
<td>1</td>
<td>0.76</td>
<td>0.65</td>
<td>0.7</td>
<td>34</td>
</tr>
<tr>
<td>Avg</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>400</td>
</tr>
<tr>
<td>Three Month Memory</td>
<td>precision</td>
<td>recall</td>
<td>F1 score</td>
<td>support</td>
</tr>
<tr>
<td>0</td>
<td>0.96</td>
<td>0.98</td>
<td>0.98</td>
<td>366</td>
</tr>
<tr>
<td>1</td>
<td>0.76</td>
<td>0.74</td>
<td>0.75</td>
<td>34</td>
</tr>
<tr>
<td>Avg</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>400</td>
</tr>
<tr>
<td>Six Month Memory</td>
<td>precision</td>
<td>recall</td>
<td>F1 score</td>
<td>support</td>
</tr>
<tr>
<td>0</td>
<td>0.96</td>
<td>0.98</td>
<td>0.98</td>
<td>366</td>
</tr>
<tr>
<td>1</td>
<td>0.76</td>
<td>0.76</td>
<td>0.76</td>
<td>34</td>
</tr>
<tr>
<td>Avg</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>400</td>
</tr>
</tbody>
</table>

Table 12 supplies the same information as its predecessors, but automatically shows that this model behaves as expected for the increase of memory, increasing its predictive power. Still the same can’t be said of the evolution of its predictive power, since it increases with the lag instead of reducing. This can possibly be explained by the inability of the equally weighted average to change the trend in these results. But even though these results appear to have some strength, it still bears value to analyze the ROC curve AUC.

Table 13. AUC results from the ROC curves of the Model’s Average.

<table>
<thead>
<tr>
<th>X</th>
<th>No Month Lag</th>
<th>Six Month Lag</th>
<th>Twelve Month Lag</th>
<th>Eighteen Month Lag</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Memory</td>
<td>0.61</td>
<td>0.96</td>
<td>0.86</td>
<td>0.97</td>
</tr>
<tr>
<td>Three Month Memory</td>
<td>0.86</td>
<td>0.95</td>
<td>0.87</td>
<td>0.9</td>
</tr>
<tr>
<td>Six Month Memory</td>
<td>0.87</td>
<td>0.95</td>
<td>0.89</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table 13 doesn’t correct the problems of Table 12, but it demonstrates that even though the predictive power increases with the lag they have very similar values and very high results. That is expected when averaging the three models since it covers the weak points of each model providing a more stable solution. One interesting model to analyze would be the average of the models with six months of lag and no months of memory, for its high value of AUC.
Figure 35. The probability of a US recession happening with six months of lag and no memory calculated by an average of all the models’ algorithms.

Figure 35 presents the visual outcome of the averaging of all the models presented in Case Studies A, B and C. As expected, the results appear to counteract the volatility of Case Study B and C, and improve on the results of A. But, yet again, for a more valid comparison, the results of the models’ average with twelve months of lag with six months of memory are required.

Figure 36. The probability of a US recession happening with twelve months of lag and six months of memory calculated by an average of all the models’ algorithms.

Figure 36 presents very stable and well-fitted results, demonstrating that the average of these models is very important in the construction of a reliable solution for this thesis’ problem. The Models’ Average contrary to Case Study B and C, does not possess a feature importance property.

With all the case studies reviewed and analyzed, it is easy to conclude that all of them have strengths and weaknesses. But, even with some failures, almost all of them present very high levels of AUC’s and F1-Scores, and their visual representations are clear enough to retrieve some conclusions. So, in a primary analysis, the system presented by this thesis appears to have fulfilled its purpose.
4.2.6 Final Review

By the analysis of the graphics, some correlations can be noted. The clearest is that the FP or peaks in the graphs are not randomly spread in time, but concentrate especially around the years 1995, 1998, 2014-15 and 2016. After a brief investigation it was noted that during these periods the US economy was fragile, and many economists feared an economic recession.

In 1995, the Mexican Republic suffered from a massive devaluation of the peso that led to a catastrophic banking crisis. This affected the US since in the late 80’s Mexico undertook large scale reforms and deregulations of its economy to enter the global market, and in the beginning of the 90’s entered in the North American Free Trade Agreement (NAFTA) [67]. The NAFTA agreement then linked both the US and Mexican economies. In 1995, Robert Rubin, treasury secretary of the US said [67],

“further collapse of the peso and of the Mexican economy could bring down economies around the world”

This led the US to issue a bailout to the Mexican Republic and in turn, destabilized the US economy, increasing the fear of a recession. In 1998, the world’s economies nearly fell into a recession as the Asian economy plundered. The Tokyo stock market and the yen had a big crash, and the U.S Treasury Department interceded together with the IMF to stop the full scale crisis [68]. In 2014-15 some economists feared that the commodity markets were developing a bubble. Adding the financial instability in Europe, Russia and China, and the previously striking recession of 2008-10 fear spread through the US economy [69], [70]. Finally, in 2016, a cocktail of significant financial and political events elevated the instability. Primary among those was the financial crisis of the Chinese stock market and the slowdown of their economy. Furthermore, political events such as the BREXIT referendum, the election of the 45th president of the US, Donald J. Trump, or the ending of the Iran sanctions created ripples in the world’s economies [71]–[73].

In conclusion the results of the previous case studies provide not only good values on their metrics, but they also signal considerably well the economic instability in the US economy.
5

Conclusion
5.1 Conclusion

The main objective of this thesis was to develop an exploratory study on the detection of the US economic turning points, more specifically the turning point to recessions. It was also an objective to try and use several macroeconomic indicators and models, and to somehow understand how all of them fell into place in a recession detection.

By reviewing the previous chapter, it can be concluded that the main objectives of this thesis were accomplished bearing good results. Three distinct models were used, with distinct capabilities and formulations, and all bore good results, each with its strengths and weaknesses. For instance, the Logistic Regression model was best at predicting recessions with no lag. The Random Forest Classifier presented the best results for the six, twelve and eighteen months of anticipation, and the XGBoost model was the most all rounded model maintaining the quality of its predictions no matter the changes. This can be confirmed by the usage of Case Study D, where all the previous models’ results were averaged, and produced very high-quality predictions for all the transformations and lags. Even the false positives granted by the models had an economic explanation, marking time spans where the US economy presented sings of fragility. In conclusion, the best model to use is the Models’ Average, where for a probability above 50%, the user might clearly be alerted for a possibility of a recession.

The results were very positive, but other conclusions were derived by some of their failures. For instance, the $F_1$-Scores and the ROC curves of each model demonstrated that using a probability threshold of 50% for every model is a mistake, even though it bears good results for the Models’ Average. Each model should be analyzed according to a specific threshold, so that no false conclusions can be derived from it. The results also showed that for some models the memory transformation didn’t improve their results, raising the question if some other transformations could improve the results.

In a final analysis of the models’ outcomes, the Random Forest Classifier and the XGBoost provided the ability to see which indicators were influential in devising their solutions. Even though the results of the feature importance characteristics of the models weren’t fully analyzed in this thesis, some preliminary conclusions could be ascertained. Mainly that they present different orders of importance according to the model being used and more importantly, according to the lag of the US recession signal. This tool will grant the possible users of this work the ability to better understand these recessions.

On an end note, this work devised several solutions, all thoroughly evaluated using state of the art metrics, and provided good results. It proved that it is possible to detect US recessions with a certain confidence, with the usage of several macroeconomic indicators assigned to specific areas of the economy. It also proves that this problem can be solved with available public information, and using a personal computer to produce the necessary computations for each model.
5.2 Future Work

Since this thesis was built on an exploratory context there are many future developments that could be produced following this work. They can be divided into two main separations, economic and technical.

As for economic developments,

- Use different macroeconomic indicators for the same areas of the economy used in this work.
- Use macroeconomic indicators from completely different areas of the economy.
- Use the maximum number of macroeconomic indicators at a time, with obvious care for different timespans and correlations between them.
- Test different economies and their recessions, e.g. European recession, Chinese recessions.
- Use indicators from different economies to detect recessions, e.g. use US macroeconomic indicators for European recession detection.

As for technical developments,

- Use different classifier models, e.g. Neural Networks, Naïve Bayes Classifier.
- Use univariate forecasting of each macroeconomic indicator, with anomaly detection. Compare the resulting anomalies between the forecast and the true signal with the other signals’ anomalies and the recession signals.
- Transform the recession signals so that they could start subtly instead of abruptly like in Figure 37. This would “tell” the model that the recession doesn’t begin in that month without notice but starts by an accumulation of a series of events.
- Use different transformations on the macroeconomic indicators, e.g. logarithmic transformations.
- Use different model selection and averaging, e.g. Bayesian Model Averaging

These are some of the possible and advised future developments that any individual following this work could take for improving the presented results and deriving further conclusions for the economic recession detection thematic.
References


### US Business Cycle Expansions and Contractions

Contractions (recessions) start at the peak of a business cycle and end at the trough.

Latest announcement from the NBER’s Business Cycle Dating Committee, dated 9/20/10.

Download Excel file with machine-readable chronology

Press citations on NBER Business Cycles

#### BUSINESS CYCLE DURATION IN MONTHS

<table>
<thead>
<tr>
<th>Peak to Trough</th>
<th>Previous Trough to this peak</th>
<th>Trough from Previous Trough</th>
<th>Peak from Previous Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 1854</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>June 1857(I)</td>
<td>December 1858 (IV)</td>
<td>18</td>
<td>30</td>
</tr>
<tr>
<td>October 1860(III)</td>
<td>June 1861 (III)</td>
<td>8</td>
<td>22</td>
</tr>
<tr>
<td>April 1862(II)</td>
<td>December 1867 (II)</td>
<td>32</td>
<td>46</td>
</tr>
<tr>
<td>June 1868(II)</td>
<td>December 1870 (IV)</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>October 1873(III)</td>
<td>March 1879 (I)</td>
<td>65</td>
<td>34</td>
</tr>
<tr>
<td>March 1882(I)</td>
<td>May 1885 (II)</td>
<td>38</td>
<td>36</td>
</tr>
<tr>
<td>March 1887(II)</td>
<td>April 1888 (I)</td>
<td>13</td>
<td>22</td>
</tr>
<tr>
<td>July 1893(II)</td>
<td>May 1891 (I)</td>
<td>10</td>
<td>27</td>
</tr>
<tr>
<td>January 1899(II)</td>
<td>June 1894 (III)</td>
<td>17</td>
<td>29</td>
</tr>
<tr>
<td>December 1895(IV)</td>
<td>June 1897 (III)</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>June 1899(III)</td>
<td>December 1900 (IV)</td>
<td>18</td>
<td>24</td>
</tr>
<tr>
<td>September 1902(IV)</td>
<td>August 1904 (III)</td>
<td>23</td>
<td>21</td>
</tr>
<tr>
<td>May 1907(III)</td>
<td>June 1908 (II)</td>
<td>13</td>
<td>33</td>
</tr>
<tr>
<td>January 1910(I)</td>
<td>January 1912 (IV)</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>January 1913(II)</td>
<td>July 1915 (II)</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>January 1913(III)</td>
<td>December 1914 (IV)</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>August 1918(III)</td>
<td>March 1919 (I)</td>
<td>7</td>
<td>44</td>
</tr>
<tr>
<td>January 1920(I)</td>
<td>July 1921 (II)</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>February 1924(II)</td>
<td>July 1924 (III)</td>
<td>14</td>
<td>22</td>
</tr>
<tr>
<td>October 1926(III)</td>
<td>November 1927 (IV)</td>
<td>13</td>
<td>27</td>
</tr>
<tr>
<td>August 1929(III)</td>
<td>March 1933 (I)</td>
<td>43</td>
<td>21</td>
</tr>
<tr>
<td>May 1937(II)</td>
<td>June 1938 (II)</td>
<td>13</td>
<td>50</td>
</tr>
<tr>
<td>February 1945(II)</td>
<td>October 1945 (IV)</td>
<td>8</td>
<td>80</td>
</tr>
<tr>
<td>November 1946(IV)</td>
<td>October 1949 (IV)</td>
<td>11</td>
<td>37</td>
</tr>
<tr>
<td>July 1953(II)</td>
<td>May 1954 (II)</td>
<td>10</td>
<td>45</td>
</tr>
<tr>
<td>August 1957(II)</td>
<td>April 1958 (I)</td>
<td>8</td>
<td>39</td>
</tr>
<tr>
<td>April 1960(I)</td>
<td>February 1961 (I)</td>
<td>10</td>
<td>24</td>
</tr>
<tr>
<td>December 1969(IV)</td>
<td>November 1970 (IV)</td>
<td>11</td>
<td>106</td>
</tr>
<tr>
<td>November 1970(IV)</td>
<td>March 1975 (II)</td>
<td>16</td>
<td>36</td>
</tr>
<tr>
<td>January 1980(I)</td>
<td>July 1980 (III)</td>
<td>6</td>
<td>58</td>
</tr>
<tr>
<td>July 1981(II)</td>
<td>November 1982 (IV)</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>July 1990(II)</td>
<td>March 1991 (II)</td>
<td>8</td>
<td>92</td>
</tr>
<tr>
<td>March 2001(I)</td>
<td>November 2001 (IV)</td>
<td>8</td>
<td>120</td>
</tr>
<tr>
<td>December 2007 (IV)</td>
<td>June 2009 (III)</td>
<td>18</td>
<td>73</td>
</tr>
</tbody>
</table>

Average, all cycles:
- 1854-2009 (33 cycles): 17.5, 38.7, 56.2, 56.4*
- 1854-1919 (16 cycles): 23.6, 26.6, 48.2, 48.9**
- 1919-1945 (6 cycles): 18.2, 35.0, 53.2, 53.0

* 32 cycles
** 15 cycles

---

Figure 38. NBER list of Business cycle dates.
## ANNEX B

<table>
<thead>
<tr>
<th>Panel</th>
<th>Model</th>
<th>AUROC</th>
<th>T-test 1</th>
<th>T-test 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Panel A: 3 months ahead</strong></td>
<td><strong>Spread(t) only</strong></td>
<td>0.562</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>Spread(t)+spread(t-6)</strong></td>
<td>0.765</td>
<td>4.341***</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>S&amp;P 500, 1y % chg</strong></td>
<td>0.963</td>
<td>12.830***</td>
<td>7.743***</td>
</tr>
<tr>
<td></td>
<td><strong>Michigan consumer survey</strong></td>
<td>0.941</td>
<td>12.706***</td>
<td>7.320***</td>
</tr>
<tr>
<td></td>
<td><strong>Debit margins (BD)</strong></td>
<td>0.933</td>
<td>11.920***</td>
<td>6.458***</td>
</tr>
<tr>
<td><strong>Panel B: 6 months ahead</strong></td>
<td><strong>Spread(t) only</strong></td>
<td>0.674</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>Spread(t)+spread(t-6)</strong></td>
<td>0.794</td>
<td>3.219***</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>S&amp;P 500, 1y % chg</strong></td>
<td>0.900</td>
<td>8.060***</td>
<td>4.658***</td>
</tr>
<tr>
<td></td>
<td><strong>Debit margins (BD)</strong></td>
<td>0.862</td>
<td>6.826***</td>
<td>3.407***</td>
</tr>
<tr>
<td></td>
<td><strong>5 yr-FF spread</strong></td>
<td>0.859</td>
<td>6.168***</td>
<td>2.667***</td>
</tr>
<tr>
<td><strong>Panel C: 12 months ahead</strong></td>
<td><strong>Spread(t) only</strong></td>
<td>0.858</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>Spread(t)+spread(t-6)</strong></td>
<td>0.883</td>
<td>1.152</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>5 yr-FF spread</strong></td>
<td>0.902</td>
<td>2.785***</td>
<td>1.365</td>
</tr>
<tr>
<td></td>
<td><strong>1 yr-FF spread</strong></td>
<td>0.897</td>
<td>2.669***</td>
<td>1.102</td>
</tr>
<tr>
<td></td>
<td><strong>NAPM com price</strong></td>
<td>0.897</td>
<td>1.555</td>
<td>0.610</td>
</tr>
<tr>
<td><strong>Panel D: 18 months ahead</strong></td>
<td><strong>Spread(t) only</strong></td>
<td>0.906</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>Spread(t)+spread(t-6)</strong></td>
<td>0.881</td>
<td>-1.014</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>Debit margins (BD)</strong></td>
<td>0.934</td>
<td>1.423</td>
<td>2.615***</td>
</tr>
<tr>
<td></td>
<td><strong>2 yr-FF spread</strong></td>
<td>0.897</td>
<td>-0.661</td>
<td>1.188</td>
</tr>
<tr>
<td></td>
<td><strong>30 yr-FF spread</strong></td>
<td>0.894</td>
<td>-1.022</td>
<td>0.907</td>
</tr>
<tr>
<td><strong>Panel E: 24 months ahead</strong></td>
<td><strong>Spread(t) only</strong></td>
<td>0.853</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>Spread(t)+spread(t-6)</strong></td>
<td>0.808</td>
<td>-1.214</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td><strong>1 yr-FF spread</strong></td>
<td>0.846</td>
<td>-0.613</td>
<td>1.762*</td>
</tr>
<tr>
<td></td>
<td><strong>2 yr-FF spread</strong></td>
<td>0.827</td>
<td>-0.995</td>
<td>1.227</td>
</tr>
<tr>
<td></td>
<td><strong>Baa-Aaa spread</strong></td>
<td>0.801</td>
<td>-1.960</td>
<td>-0.298</td>
</tr>
</tbody>
</table>

Figure 39. Weiling Liu and Emanuel Moench metric results table [17].
Figure 40. Some of the ROC curve results presented in the Wells Fargo work [41].
ANNEX D

PREDICTING U.S. RECESSIONS FROM FINANCIAL VARIABLES

Table 3.—Measures of Fit and t-Statistics for Probit Model Variables with Spread, In Sample

\[ P(R_{t+k} = 1) = \Phi(\alpha + \beta X + \gamma \text{SPREAD}_t) \]

<table>
<thead>
<tr>
<th>x Variables</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPREAD</td>
<td>0.071</td>
<td>0.211</td>
<td>0.271</td>
<td>0.296</td>
<td>0.256</td>
<td>0.149</td>
<td>0.078</td>
<td>0.031</td>
</tr>
<tr>
<td>t-stat</td>
<td>-2.71c</td>
<td>-4.21c</td>
<td>-4.71c</td>
<td>-4.57c</td>
<td>-3.87c</td>
<td>-4.13c</td>
<td>-3.02c</td>
<td>-1.63</td>
</tr>
<tr>
<td>RMT</td>
<td>0.142</td>
<td>0.233</td>
<td>0.272</td>
<td>0.307</td>
<td>0.295</td>
<td>0.165</td>
<td>0.102</td>
<td>0.051</td>
</tr>
<tr>
<td>t-stat</td>
<td>1.96</td>
<td>3.06</td>
<td>3.32</td>
<td>1.14</td>
<td>-2.28d</td>
<td>-2.55e</td>
<td>-2.27d</td>
<td>-1.09d</td>
</tr>
<tr>
<td>NYSE</td>
<td>0.154</td>
<td>0.213</td>
<td>0.283</td>
<td>0.309</td>
<td>0.258</td>
<td>0.151</td>
<td>0.081</td>
<td>0.033</td>
</tr>
<tr>
<td>t-stat</td>
<td>-3.17c</td>
<td>-0.03</td>
<td>-0.90</td>
<td>-1.03</td>
<td>-0.43</td>
<td>-0.54</td>
<td>-0.53</td>
<td>-0.12</td>
</tr>
<tr>
<td>LEP</td>
<td>0.223</td>
<td>0.32</td>
<td>0.321</td>
<td>0.314</td>
<td>0.261</td>
<td>0.196</td>
<td>0.096</td>
<td>0.083</td>
</tr>
<tr>
<td>t-stat</td>
<td>-3.54c</td>
<td>-4.81c</td>
<td>-3.22c</td>
<td>-1.57</td>
<td>0.02</td>
<td>1.17</td>
<td>1.71</td>
<td>3.69c</td>
</tr>
<tr>
<td>LAW</td>
<td>0.256</td>
<td>0.283</td>
<td>0.331</td>
<td>0.296</td>
<td>0.265</td>
<td>0.16</td>
<td>0.106</td>
<td>0.054</td>
</tr>
<tr>
<td>t-stat</td>
<td>-3.11c</td>
<td>-4.29c</td>
<td>-2.07c</td>
<td>-0.08</td>
<td>1.44</td>
<td>1.03</td>
<td>1.48</td>
<td>1.22</td>
</tr>
<tr>
<td>S&amp;P</td>
<td>0.223</td>
<td>0.218</td>
<td>0.275</td>
<td>0.296</td>
<td>0.264</td>
<td>0.100</td>
<td>0.013</td>
<td>0.037</td>
</tr>
<tr>
<td>t-stat</td>
<td>-3.54c</td>
<td>-3.74c</td>
<td>-0.69</td>
<td>-0.07</td>
<td>-0.62</td>
<td>0.60</td>
<td>0.70</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Notes: *t* denotes t-statistic for SPREAD variable.
*Significance at the 1% level.
*Significance at the 5% level.

Figure 41. Estrella and Mishkin some metric results [3].

Figure 42. One of Estrella and Mishkin plot results [3].

95
ANNEX E

Table 1. Forecast Loss at Different Significance Levels

<table>
<thead>
<tr>
<th>Significance Level</th>
<th>AVE</th>
<th>0.01</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
<th>0.20</th>
<th>0.25</th>
<th>0.30</th>
<th>0.35</th>
<th>0.40</th>
<th>0.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>1M ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPS</td>
<td>0.17</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
</tr>
<tr>
<td>LPS</td>
<td>0.29</td>
<td>0.23</td>
<td>0.25</td>
<td>0.24</td>
<td>0.25</td>
<td>0.25</td>
<td>0.26</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.26</td>
</tr>
<tr>
<td>3M ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPS</td>
<td>0.19</td>
<td>0.18</td>
<td>0.18</td>
<td>0.17</td>
<td>0.17</td>
<td>0.16</td>
<td>0.17</td>
<td>0.17</td>
<td>0.17</td>
<td>0.17</td>
<td>0.18</td>
</tr>
<tr>
<td>LPS</td>
<td>0.33</td>
<td>0.60</td>
<td>0.34</td>
<td>0.31</td>
<td>0.31</td>
<td>0.29</td>
<td>0.28</td>
<td>0.28</td>
<td>0.29</td>
<td>0.29</td>
<td>0.30</td>
</tr>
<tr>
<td>6M ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPS</td>
<td>0.21</td>
<td>0.22</td>
<td>0.20</td>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.19</td>
<td>0.19</td>
<td>0.18</td>
<td>0.19</td>
<td>0.19</td>
</tr>
<tr>
<td>LPS</td>
<td>0.36</td>
<td>0.36</td>
<td>0.35</td>
<td>0.33</td>
<td>0.33</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.33</td>
</tr>
<tr>
<td>12M ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPS</td>
<td>0.23</td>
<td>0.24</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
</tr>
<tr>
<td>LPS</td>
<td>0.38</td>
<td>0.41</td>
<td>0.40</td>
<td>0.38</td>
<td>0.38</td>
<td>0.38</td>
<td>0.37</td>
<td>0.37</td>
<td>0.36</td>
<td>0.36</td>
<td>0.37</td>
</tr>
</tbody>
</table>

Notes: QPS and LPS of the EAL forecasts are reported for all horizons. The performance is evaluated between 1973M1 and 2008M12.

Table 2. Performance of the Algorithm Relative to Simple Forecast Averaging

<table>
<thead>
<tr>
<th>Significance Level</th>
<th>0.01</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
<th>0.20</th>
<th>0.25</th>
<th>0.30</th>
<th>0.35</th>
<th>0.40</th>
<th>0.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>1M ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPS</td>
<td>0.75</td>
<td>0.79</td>
<td>0.79</td>
<td>0.82</td>
<td>0.82</td>
<td>0.81</td>
<td>0.80</td>
<td>0.81</td>
<td>0.83</td>
<td>0.85</td>
</tr>
<tr>
<td>LPS</td>
<td>0.77</td>
<td>0.84</td>
<td>0.82</td>
<td>0.85</td>
<td>0.84</td>
<td>0.83</td>
<td>0.82</td>
<td>0.83</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>3M ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPS</td>
<td>0.96</td>
<td>0.92</td>
<td>0.89</td>
<td>0.89</td>
<td>0.86</td>
<td>0.86</td>
<td>0.84</td>
<td>0.88</td>
<td>0.89</td>
<td>0.91</td>
</tr>
<tr>
<td>LPS</td>
<td>1.83</td>
<td>1.04</td>
<td>0.94</td>
<td>0.94</td>
<td>0.88</td>
<td>0.83</td>
<td>0.85</td>
<td>0.87</td>
<td>0.89</td>
<td>0.91</td>
</tr>
<tr>
<td>6M ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPS</td>
<td>1.02</td>
<td>0.96</td>
<td>0.86</td>
<td>0.85</td>
<td>0.84</td>
<td>0.88</td>
<td>0.88</td>
<td>0.86</td>
<td>0.87</td>
<td>0.87</td>
</tr>
<tr>
<td>LPS</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.83</td>
<td>0.85</td>
<td>0.87</td>
<td>0.87</td>
<td>0.85</td>
<td>0.87</td>
<td>0.87</td>
</tr>
<tr>
<td>12M ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPS</td>
<td>1.06</td>
<td>1.02</td>
<td>1.02</td>
<td>0.97</td>
<td>1.01</td>
<td>1.01</td>
<td>1.00</td>
<td>0.98</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>LPS</td>
<td>1.06</td>
<td>1.04</td>
<td>0.96</td>
<td>0.94</td>
<td>0.99</td>
<td>0.96</td>
<td>0.97</td>
<td>0.90</td>
<td>0.95</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Notes: For each significance level, the table reports the ratio of QPS and LPS of algorithm forecasts to that of the simple averaging. A ratio of less than one indicates better performance by the competing model relative to AVE.

Figure 43. Baba and Kışınbay some metric results [18].