Study of Market Influence on Tender Performance

Bruno Miguel Repolho Pires
brunomrpires@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

July 2021

Abstract

Companies in construction and industrial fields participate in reverse tenders in order to carry out contracts with clients. In these tenders, a client requires a certain service to be performed and then a pool of companies propose offers for the given project from which the client must choose the best offer.

To improve performance, two different lines of improvement can be identified. Namely, how to extract additional useful information from the previous auctions and how to integrate the external factors into the consideration. This project has two main objectives: to study the influence of the state of the market on the company's performance; to create a decision support tool for the company managers to optimize the performance in tenders.

In order to study the market influence, we consider two approaches: one explicit, where we create an economic index specific to the market; and one implicit, where we use those index values and, using Hidden Markov Models (HMM), we compute the hidden states of the market.

After considering various approaches for predicting the success probability of future tenders, we consider a Logistic Regression approach where we develop Frequentist and Bayesian models. These models are then incorporated in a decision tool that the company managers can use.

After comparing the performance of all the models, we conclude that the Bayesian models with the market states from the implicit study of the market influence yield the best results and are the best at capturing the relationship between the market and tender performance.

Keywords: Hidden Markov Models, Logistic Regression, Bayesian Statistics, Client Categorization

1. Introduction

EQS Global is a service provider for the highly demanding industries that operates on construction and industrial markets. On these markets it is common for companies to participate in reverse auctions/tenders or RFQ (Request for Quotation). First, the client sends the set of services for which the price is asked, then the suppliers answer the request with a price proposal, and finally the client provides a feedback. In the present paper we look only at one phase auctions where the clients chooses one of the bids from the first set of offers presented by the suppliers.

Although the price of the bid is the main criteria, it is not the unique criteria for the selection of the buyers. As the data shows, the relationship between the buyer and seller plays an important role in the outcome.

We are considering the scenario that the partner company, (further denominated by company) is participating in such tenders. Each tender is characterized by multiple internal variables: a global margin K, the date at which the auction took place, an indicator of which client the company engaged in business with, ClientID, and an indicator of whether or not the auction was successful. The global margin corresponds to the percentage increase of the proposal over the base cost of the service being provided and corresponds to the sum of the following factors: the profit margin of the proposal (P), overheads (O), which corresponds to the cost associated with managing the project and the human resources contribution, and the risk factor (R), which accounts for the risk associated with the proposal. All of these factors are given as a percentage of the baseline cost of the service/product.

When constructing a proposal the managers typically focus on the industry standard variable, the global margin K. Other important factors to consider are the market conditions at the time of the proposal and the relationship the company has with the client.

This project's two main objectives are: to create a decision support tool for the company managers to optimize the performance in tenders; to study the influence of the state of the market on the company's performance in tenders.

2. Market Index

In the explicit approach to the study of the market influence on tender performance we construct a Market Index (MI) and then define models to predict its future values.

2.1. Index Construction

EQS Global operates is a very specific market which can be considered as a combination of several fields. This means that it is essential to construct a market index that can reflect the environment in which the company operates. After several discussions with our industry partners it was decided to use four indexes from the EuroStat public platform, that gathers economic and social indexes regarding the European region. Namely, the Producer Price in Industry Index (Pr. Price), Producer Price in Construction Index (Co. Price), Turnover Index (Turn) and Labour Input in Construction Index (Lab. Inp.), that can be found in [1] and are updated every quarter. The index is then constructed by using Principal Component Analysis (PCA), as in [14].

Table 1: Market Index coefficients

Index	Pr. Price	Co. Price	Turn	Lab. Inp.
Coef.	0.263	0.449	0.537	0.664

Considering the first principal component, it explains over

86% of the variation in the data and thus, it is a good representative of the indexes considered. The market index (MI) is then defined as the first principal component obtained and has the coefficients shown in Table 1.

2.2. Index Forecast

We then have MI values until the second quarter of 2020. Once we need to make predictions on tenders in the first and second quarters of 2021, it is necessary to develop a method to estimate future values for the MI. In the literature, estimation and prediction of such market indexes that involves the construction industry are very closely related to estimation of Tender Price Index (TPI), as in [18]. By definition, Tender Price Index measures the movement of prices in tenders for building contracts in the public sector in a respective region. It doesn't, however, include contracts for housing, engineering and maintenance works. In the literature various methods to predict Tender Price Index values are applied, from Regression Analysis(RA) [18], Time Series(TS) [17] and Neural Networks(NN) [25]. The general consensus among the researchers is that an integrated model, as presented in [18], is the best approach and the most reliable alternative, which we decide to follow. Then, taking a closer look at the example in [18] one can see that the final presented model corresponds to a combination of a time series Autoregressive Integrated Moving Average (ARIMA) [5] model with a Regression model, where macroeconomic and other construction based variables are used.

2.2.1 Regression Model

For the Regression Analysis we gather several other macroeconomic variables from the EuroStat platform. As the impact of certain variables on the MI might be delayed for a few quarters we also need to consider, in the pool of possibly relevant variables, the one, two and three quarters lagged variants of the variables already gathered. We then adopt an automated stepwise procedure in order to eliminate those variables with negligible impact on the MI. The selected variables are chosen based on the p-values of each feature.

Table 2: Summary table of the stepwise procedure of multivariate regression

Var	R^2	R^2_{adj}	BIC	p-value
PSI	0.9604	0.9597	266.68	7.647e-39
PPSI	0.9669	0.9656	260.86	2.407e-3
LII3	0.9788	0.9776	240.31	2.046e-6
IR3	0.9869	0.9858	218.06	1.157e-6
PCI	0.9892	0.9880	211.49	2.254e-3
GDPI	0.9909	0.9897	205.92	4.040e-3

Table 2 summarizes the stepwise procedure of the multivariate Regression Analysis. Variables are added or removed from the regression model step by step. The variables selected to incorporate the model are the following: Production in Service Index (PSI), Producer Price in Service Index (PPSI), Labour Input in Industry Index with 3 quarters lag (LII3), Interest Rates with 3 quarters lag (IR3), Production in Construction Index (PCI) and GDP Index (GDPI).

Thus, the resulting model is given by the following expression:

$$\begin{aligned}\widehat{Y}_{MI} = &-243.4089 + 0.6425X_{PSI} + 3.3905X_{PPSI} \\ &- 0.3808X_{LII3} + 1.6982X_{IR3} - 0.2423X_{PCI} \\ &+ 0.9369X_{GDPI}\end{aligned} \tag{1}$$

with X_{PSI} denoting the values for the PSI and similarly for the other variables and $\widehat{Y}_{MI}$ denoting the estimate of the MI.

Then, considering the task of determining the future values of the explanatory variables of the model in equation (1), taking into consideration the problems identified by Yuu in [26], we improve on the method in [18]. We improve on this method by using a stochastic time series modelling technique known as Auto Regressive Integrated Moving Average (ARIMA) [5].

A stationary autoregressive moving average (ARMA) process, $(X_t)_{t\in\mathbf{N}_0}$, of order p and q, abbreviated as ARMA(p,q), is a combination of an Auto-Regressive (AR) model hat expresses the present value of a process as a linear combination of past values plus a random stochastic term representing uncorrelated forces acting on the system, with a Moving Average (MA) model that expresses the present value of a process as a linear combination of white noise variables. It can then be defined as follows:

$$X_t = \psi_1 X_{t-1} + ... + \psi_p X_{t-p} + Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-p} \tag{2}$$

which can be rewritten using the backward shift notation, $B^d X_t = X_{t-d}$ as:

$$\begin{aligned}&\Psi(B)X_t = \Theta(B)Z_t \\ &\text{with } \Psi(B) = 1 - \sum_{i=1}^{p}\psi_i B^i,\ \Theta(B) = 1 + \sum_{j=1}^{q}\theta_j B^j\end{aligned} \tag{3}$$

where $\psi_p \neq 0, \theta_q \neq 0$, $\psi_1, ..., \psi_p$ and $\theta_1, ..., \theta_q$ are constants, and $(Z_t)_{t\in\mathbf{N}_0} \sim N(0, \sigma^2)$, is usually called white noise process.

If the process in question is non stationary then we can attempt to remove the trend component by differencing the series until the transformed observations resemble a realization of some stationary time series. Then, $(Y_t)_{t\in\mathbf{N}_0}$ is an ARIMA(p,d,q) process if

$$X_t = (1-B)^d Y_t \tag{4}$$

is an ARMA(p,q) process (with d a non-negative integer that indicates the number of differencing steps).

In order to obtain the best fit ARIMA model for each feature we first use the Augmented Dickey-Fuller test to check if the given series is stationary. When stationarity is not satisfied we perform differencing to the series. In cases where differencing is not enough, we apply a logarithm transformation to the series and then apply differencing to achieve stationarity. Then, having the new stationary series we use a GridSearch approach to find the best fit ARMA model for each series using as selection criteria the Akaike Information Criterion (AIC) [2]. The methods applied to each variable, as well as the p-value from the Augmented Dickey-Fuller test for the transformed series can be seen in Table 3. At 10% significance level all the transformed series can be considered stationary.

In Table 3 the variables are named after the nomenclature used for the regression model equation, in equation (1). The

transformations applied to the series range from none, to differencing in one quarter indicated by "Differencing(1)" or to differencing in one quarter after being applied the logarithm transformation to the series, indicated by "Log + Differencing(1)".

Table 3: Transformations to explanatory variables and p-value of AD-Fuller test of stationarity

Variable	Transformation	p-value
PSI	None	0.047410
PPSI	Differencing(1)	0.000043
LII3	Log + Differencing(1)	0.070523
IR3	None	0
PCI	Differencing(1)	0
GDPI	None	0.002998

Afterwards, considering now the stationary transformed series, we can find the best fit ARMA model to each series using a GridSearch approach. The implemented algorithm cycles through multiple combinations of the model parameters, i.e., it computes ARMA(p,q) models for the series with p and q varying from 0 to 4. In each iteration it computes the AIC score for the model. The final model corresponds to the one with the lowest AIC score.

The obtained estimated ARMA models' p and q parameters can be seen in Table 4. After obtaining the models we still need to do some diagnostic checking to see if the models are indeed a good fit for the data. The Ljung-Box test is used to assess if the residuals have no autocorrelation. The Jarque-Bera test is used to check if the residuals resemble white noise. In Table 4 we have the p-values for both the Ljung-Box and the Jarque-Berta test for each models' residuals.

Table 4: ARMA models and Ljung-Box and Jarque-Bera test results on the residuals of the explanatory variables

Var	ARMA	Ljung - Box				Jarq.
	(p,q)	Lag 1	Lag 2	Lag 3	Lag 4	Bera
PSI	(2,1)	0.765	0.931	0.967	0.989	0
PPSI	(3,1)	0.991	0.997	0.877	0.915	0
LII3	(1,3)	0.698	0.894	0.952	0.968	0.58
IR3	(1,2)	0.675	0.865	0.899	0.887	0
PCI	(0,0)	0.817	0.954	0.991	0.997	0
GDPI	(2,2)	0.923	0.988	0.998	0.999	0

From the Ljung-Box test and Jarque-Bera tests results we conclude that the residuals are not autocorrelated and that only the residuals of the Labour Index with 3 quarters lag show evidence of being normally distributed.

2.2.2 Time Series Model

For the time series model we perform a similar process to the one described for the prediction of future values of the explanatory variables of the regression model. First, we perform Augmented Dickey-Fuller test to check if the series is stationary. When the stationary is not satisfied we perform differencing and logarithm transformation to the data in order to achieve the condition. The final transformed series is obtained after differencing 2 quarters on the logarithm of the initial series values. For this transformed series the Ad-Fuller test yields the p-value 0.001502 and thus we accept the stationarity of this series.

Afterwards, we find the best fit ARMA model for the transformed series through the same GridSearch approach on the p and q parameters of the ARMA model with the decision criterion being the AIC score. The best fit model obtained is the ARMA(2,1) model. With the Ljung-Box test we assess that the residuals have no autocorrelation and with the Jarque-Bera test we cannot assess normality in the residuals distribution. These results can also be seen in Table 5

Table 5: ARMA models and Ljung-Box and Jarque-Bera test results on the residuals of the transformed MI time seires

Var	ARMA	Ljung-Box				Jarque
	(p,q)	Lag 1	Lag 2	Lag 3	Lag 4	Bera
MI	p=1,q=2	0.856	0.973	0.934	0.950	0

The ARMA(2,1) model obtained is defined in the following equation (using back shift notation)

$$(1 - 1.5873B + 0.7688B^2)Y_t = 0.0013 + (1 - 0.9631B)Z_t \quad (5)$$

where Y_t is the value of the transformed MI series in current time period t, B is the back shift operator, and $Z_t \sim N(0, 0.0003)$. In Table 6 we have the summary of the model's coefficients. On the one hand, all the p-values indicate that all the coefficients are statistically different from zero, on the other hand the Jarque-Bera test yields a p-value of 0 and thus we cannot assume that the residuals are normally distributed. As in the regression model, the latter does not have an impact on the remaining of the analysis.

Table 6: Summary of the coefficients of the ARMA(2,1) model for the MI time series

Coef	Value	Std Err	z	P>|z|
intercept	0.0013	0.001	2.306	0.021
ar.L1	1.5873	0.107	14.777	0.000
ar.L2	-0.7688	0100	-7.717	0.000
ma.L1	-0.9631	0.260	-3.700	0.000
sigma2	0.0003	8.73e-5	3.295	0.001

2.2.3 Integrated Model

The procedure for determining the coefficient of the linear combination between the Regression (RA), and Time Series, (TS), is the same as the algorithm presented in [18]. The coefficient for the linear combination is obtained such that it minimizes the Root Mean Square Error (RMSE) of the forecasts of the integrated model.

The RMSE for the Regression model and the Time Series model are defined, respectively as:

$$\mathcal{R} = \sqrt{\frac{\sum_{i=1}^{19}(R_i)^2}{19}} \quad \text{and} \quad \mathcal{T} = \sqrt{\frac{\sum_{i=1}^{19}(T_i)^2}{19}} \quad (6)$$

where R_i corresponds to the error of the i^{th} prediction of the Regression model and T_i corresponds to the error of the i^{th} prediction of the Time Series model. The RMSE of the Integrated model, denoted by $\mathcal{IM}$ is given by $\mathcal{IM} = \beta \times \mathcal{T} + (1 - \beta) \times \mathcal{R}$ with $\beta \in [0, 1]$.

The algorithm for determining the coefficient β that minimizes $\mathcal{IM}$ is constructed by successive decimal approximations:

- Compute the RMSE of the Integrated model, $\mathcal{IM}$ for the following values of β: $\{0, 0.1, 0.2, ..., 0.8, 0.9, 1\}$ and define β_m as the minimizer within that subset of β values;
- Compute the RMSE of the Integrated model, $\mathcal{IM}$ for the following values of β: $\{\beta_m - 0.1, \beta_m - 0.09, \beta_m - 0.08, ..., \beta_m + 0.09, \beta_m + 0.1\}$ and update β_m with the minimizer within that subset of β values;
- Compute the RMSE of the Integrated model, $\mathcal{IM}$ for the following values of β: $\{\beta_m - 0.01, \beta_m - 0.009, \beta_m - 0.008, ..., \beta_m + 0.009, \beta_m + 0.01\}$ and update β_m with the minimizer within that subset of β values;

After performing the algorithm, the obtained coefficient is $\beta_m = 0.545$ which means that the predictions of the integrated model have a slightly larger influence of the Time Series model (0.545) than the Regression model (0.455).

In Figure 1, it is possible to compare the real values for the market index (dashed line) with the predictions for the integrated model (dotted line), since 2015Q3 until 2020Q2.

We see that the values obtained with the integrated model closely resemble the actual values, especially if we consider the period from 2015Q3 to 2019Q4. Obviously the last three quarters compared, 2019Q4 to 2020Q2, are directly impacted by the COVID-19 pandemic. Therefore it is difficult to predict the substantial drop in the market index during these three quarters. Nevertheless, one can see that a portion of the drop is, indeed, captured by the prediction model.

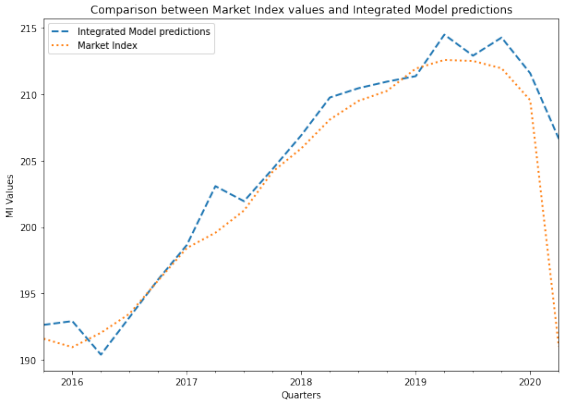


Figure 1: Market Index predictions

3. Data Analysis

Although the main objective is to study the impact of the state of the market and to develop a strategy that improves future performance, the given data not only provides some insights to each proposal's characteristics but also about the clients.

The provided data includes: Time at which the auctions took place (between 2018 and the last quarter of 2020); The global variable K; The three variables that form the global margin (profit P, overheads O and risk R); The identification of each individual client (ClientID); A binary variable, indicating if the auctions was successful (Adjudicated).

By plotting the number of successful and unsuccessful tenders in each quarter alongside an evolution of the market index values, Figure 2, we see that in most quarters there are more unsuccessful tenders than successful ones, except for the first quarter of 2019 and the last two quarters of 2020. Additionally, we also note that there are considerably fewer observations from 2020 compared to the previous 2 years, with 23 observations for 2020, 53 for 2019 and 41 for 2018. The lack of observations from 2020 is mainly due to the fact that the data was gathered in December of 2020 and thus not only wasn't the quarter over but also some of the information from completed tenders wasn't yet available for this study. Additionally, we see that the MI rises steadily until the last quarter of 2019 when it suddenly collapses, due to the COVID-19 pandemic. The index starts to recover on the third quarter of 2020, according to the model's predictions.

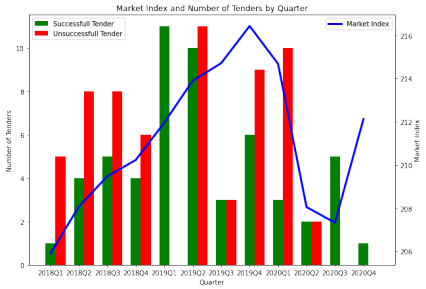


Figure 2: Market Index and Number of Tenders by Quarter

Then, by plotting the global margin as a function of time, including also, information about the outcome of each tender, Figure 3, we see that many tenders have a K value in the range $30-35\%$. Furthermore, we can see that the successful tenders tend to have lower global margins and also that a majority of the tenders with global margin above 50% are unsuccessful. However, there are still a considerable amount of unsuccessful tenders with low levels of K and also some successful tenders with high levels of K.

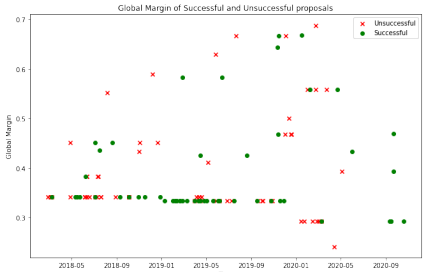


Figure 3: Global Margin of Successful and Unsuccessful tenders

3.1. Client Categorization

The remaining variable of interest is the ClientID variable that has a unique value for each client with which the company has engaged in business since 2018. By analysing the number of observations per client available we see that: each client appears in the dataset an irregular number of times; a large percentage of the clients only appear once in the dataset.

These remarks justify considering clusters in the data. In the experiments, various clustering techniques are performed with the objective of partitioning the clients into two, three or four categories. Partitioning the clients using the k-means method [10], not only does it not improve the performance of the model but also, the partitions created are not clear or intuitive. Therefore, we decide to create a new variable, called *Sympathy*, that describes the level of sympathy that the clients have with the company in the following way: Category 0 - Unfriendly, corresponds to those clients with which the company tried to engage in business more than once and was unsuccessful every time; Category 1 - Friendly, corresponds to those clients with which the company tried to engage in business more than once and was successful every time; Category 2 - New, corresponds to those clients from which there isn't enough information about

the state of their relationship with the company, because they only engaged with each other once (independently of the fact that they won or lost the respective auction with); Category 3 - Regular, corresponds to those clients with which the company tried to engage in business multiple times and the overall outcome is uncertain, i.e. there are some cases when the company wins the auction, and in others, it loses;

3.2. Correlation Analysis

After defining the Sympathy variable we can then proceed with the analysis of correlation between variables. Interpreting such correlations may provide some insights into the company's behaviour in some situations.

In our case, it is important to study the relations between the continuous variables MI and K, and the binary variable indicating the success/failure of the tenders. When computing the correlation between MI and Adjudicated or K and Adjudicated we are dealing with one continuous and one binary variable. Thus we have to calculate the point biseral correlation coefficient [13]. Considering the entire data we get the following results:

Table 7: Point Biseral Correlation between MI and Adjudicated, and K and Adjudicated

	Score	P-Value
K	-0.0404	0.6655
MI	-0.0199	0.8315

We see that both correlations are negative but non-significant. Although there does not seem to exist any significant correlation between these variables at first glance we can compute the correlations but now conditioning the set of observations according to the type of client. After conditioning on the set of observations by client categories we also obtain correlation values that are non significant at 10% level.

The absence of significant correlation between variables as we will see indicates the existence of non-linear relations between variables.

4. Models

After analysing all the variables we can now proceed to the creation of models that predict the success probability of future tenders. We have, at our disposal to include in the model, several explanatory variables such as the global margin K, the MI variable, the Sympathy variable and the binary indicators for each client category.

For this situation, several types of models are considered, such as Neural Networks [6], Random Forests [27], XGBoost algorithm [27] and Logistic Regression [4]. The method we choose to model the probability of success of the tenders is Logistic Regression. When fitting the different models to the data, Neural Network, Random Forests and XGBoost models have poor performances. Additionally, all of these models return success probabilities between 45% and 55% for New and Regular clients while the Logistic Regression models yield probability curves that span across a wider range of values. With such a narrow difference between proposals, the former models prove to be not suitable for our case because they do not provide enough distinction between the proposals in their predictions. Therefore, we focus on the Logistic Regression approach.

4.1. Logistic Regression

For the Logistic Regression model construction it is important to consider two different approaches. On one hand, a frequentist model, similar to [4] with the additional component of robustness considered [20]. On the other hand, a Bayesian approach, where prior knowledge of the field and the company can be taken into consideration in the model. In both approaches, the base model we first consider consists of a logistic regression model given by the following expression:

$$P(Success) = \pi(X) = \frac{exp(\beta_0 + \beta X)}{1 + exp(\beta_0 + \beta X)} \tag{7}$$

where $\beta = (\beta_K, \beta_{MI}, \beta_N, \beta_R, \beta_F, \beta_U)$ and $X = (X_K, X_{MI}, X_N, X_R, X_F, X_U)$, where the β_i's are the coefficients associated with the explanatory variables X_i's which, in this case, correspond to the global margin variable, K, the market index variable, MI, and the sympathy indicators for each client category, New, N, Regular, R, Friendly, F and Unfriendly, U.

After the definition of the model, we then consider two different approaches in the remaining steps of the model conception: the frequentist and Bayesian.

4.1.1 Frequentist Approach

In the frequentist case we perform 1000 iterations of the train/test split of the data, where, for each iteration we consider a Logistic Regression model with the formula from equation (7) and then, use the Logistic Regression Python routine to estimate the parameters of the model, via maximum likelihood. This maximization, for each one of the train partitions can be adjusted and calibrated with several parameters. The best parameters are obtained through a GridSearch approach. When looking at the AUC scores for the 1000 different dataset partitions, we realise that there is a large variance in the overall AUC score, indicating that the dataset partition has a huge impact on the final score. In view of this, we propose the use of a robust version of the previous algorithm. For that purpose, we use the redescending M-estimators method [15]. Using this technique, we can compute a new, robust, logistic regression model. After computing each model for 20 different train/test partitions, the AUC scores distributions for both cases can be seen in Figure 4.

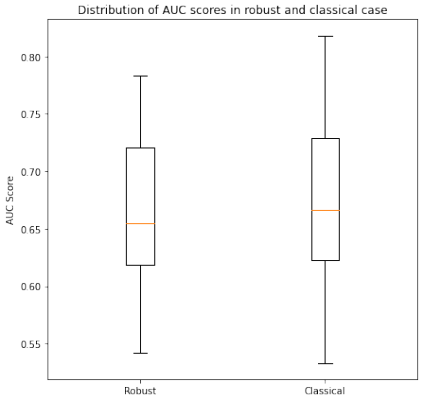


Figure 4: Boxplot for the AUC results of the robust and classical logistic regression models for the same 20 train/test partitions

In some of the partitions, there is no significant difference between the classical and robust model, suggesting that in

those cases either there are no outliers present in the partition or their impact is not relevant. But overall the performance of the robust methods is better and hence from this point on, whenever we present results from the frequentist approach, we use their robust version. In Table 8 we can see the estimated values for each one of the β_i coefficients that correspond to the mean estimated values. In Table 8 we also have the p-values from the Likelihood Ratio Test (LRT) [23]. The LRT tests the hypothesis $H_0 : \beta_i = 0$ against the hypothesis $H_1 : \beta_i \neq 0$. The test statistics is given by $LR = -2(l(\widehat{\beta}|H_0) - l(\widehat{\beta}|H_1)$, where l denotes the log-likelihood. When we test an hypothesis for just one coefficient, $LR \sim \chi^2_1$. At a 10% significance level we reject the null hypothesis in all the cases. At a 1% significance level, all except β_U are statistically significant.

Table 8: Mean coefficients and LRT p-values for the robust model

Coefficient	Value	LRT p-value
β_0	10.563	0
β_K	-5.140	0
β_{MI}	-0.049	0
β_N	1.468	0.0001
β_R	2.674	0.0081
β_F	19.208	0
β_U	-15.031	0.0953

4.1.2 Bayesian Approach

The other approach for model creation consists on taking advantage of Bayesian statistics theory. For estimation of this model we use the Python package, PyMC3 [22]. In this package, one can construct Bayesian logistic regression models and define specific prior distributions for the coefficients. This Python package has a unique modeling process that generally follows the following steps:

1. Encode a probability model by defining the following:
   (a) The prior distributions that quantify knowledge and uncertainty about the β parameters.
   (b) The likelihood function that combines the parameters with the data according to the specification of the logistic regression.
2. Analyze the posterior by sampling from the posterior using Markov Chain Monte Carlo (MCMC) methods [3].
3. Check the model using various diagnostic tools.
4. Generate predictions.

The resulting model can be used for inference to gain detailed insights into parameter values as well as to predict outcomes for new data points.

For logistic regression, the likelihood contribution from the i^{th} observation is binomial and given by:

$$L(y_i|x_i, \beta) = \pi(x_i)^{yi}(1 - \pi(x_i))^{1-y_i} \tag{8}$$

with $\pi(x_i) = P(y_i = 1|x_i, \beta)$ given by equation (7). Then the likelihood function over the data set of n observations is:

$$p(y|\beta, X) = \prod_{i=1}^{n} \left[\left(\frac{e^{\beta_0+\beta x_i}}{1+e^{\beta_0+\beta x_i}} \right)^{y_i} \left(1 - \frac{e^{\beta_0+\beta x_i}}{1+e^{\beta_0+\beta x_i}} \right)^{1-y_i} \right] \tag{9}$$

where $\beta = (\beta_K, \beta_{MI}, \beta_N, \beta_R, \beta_F, \beta_U)$, and $x_i = (x_{K_i}, x_{MI_i}, x_{N_i}, x_{R_i}, x_{F_i}, x_{U_i})$ corresponds to the values of the i^{th} observation.

Although there are many possible options for the prior distributions of the unknown parameters β_j, for simplification purposes let's consider the normal distribution:

$$\beta_j \sim N(\mu, \sigma_j^2) \tag{10}$$

where $\mu = 0$ and σ is usually chosen to be large enough to be considered as non-informative. Common choices being in the range from $\sigma = 10$ to $\sigma = 100$.

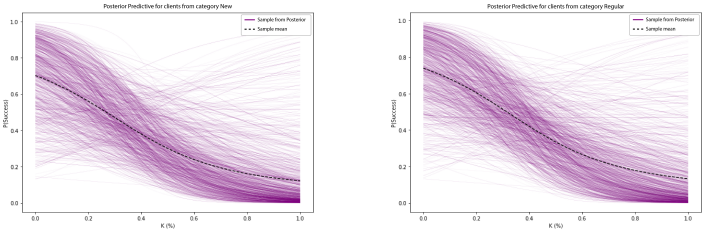
Then the posterior distribution is derived by multiplying the prior distribution over all the parameters by the full likelihood function, so that:

$$p(y|\beta, X) = \prod_{i=1}^{n} \left[\left(\frac{e^{\beta_0+\beta x_i}}{1+e^{\beta_0+\beta x_i}} \right)^{y_i} \times \left(1 - \frac{e^{\beta_0+\beta x_i}}{1+e^{\beta_0+\beta x_i}} \right)^{1-y_i} \right] \times \prod_{j=1}^{7} \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(\beta_j-\mu_j)^2}{2\sigma_j^2}} \tag{11}$$

The above expression is computationally expensive and multiple integrations have to be performed to obtain the marginal distribution for each coefficient.To avoid this MCMC (Markov Chain Monte Carlo) [3] methods are used.

MCMC simulation [3] is a general method based on drawing values of β from approximate distributions and then correcting those draws to better approximate the posterior distribution, $p(\beta|y, x)$. The algorithm used in the PyMC3 package [22] is denoted NUTS (No-U-Turn Sample) [11]. The algorithm can still be, very roughly, described as an improvement over the Hamilton Monte Carlo (HMC) [9] algorithm, by removing the need to set a number of steps parameter L. In turn, the HMC algorithm can be described as an improvement over the simpler Gibbs Sampler [8] and Metropolis-Hastings [16] algorithms, by avoiding the random walk behaviour present in the former.

Then, we can sample from the posterior predictive distribution and create density plots similar to the ones from Figure 5. Here, in Figure 5, the plots represent the posterior predictive distributions for the New and Regular clients' categories using generalized t-Student distributions as priors for each of the β coefficients.



(a) Posterior Predictive for New clients

(b) Posterior Predictive for Regular clients

Figure 5: Posterior Predictive plots for the New and Regular clients on 2021Q1

Figure 5(a) represents 2000 samples (given by the purple lines) from the posterior predictive distribution for the New clients in 2021Q1 and the mean curve of those 2000 sample curves (given by the black dashed line) while Figure 5(b) represents 2000 samples for Regular clients in the same time frame. The means of the two categories appear to be relatively similar and by analysing the density of the 2000 curves

in each case one can say that there is quite some volatility and uncertainty in both cases.

5. Market States

After constructing two different models that estimate the success probability of proposals, we address next the study of the market influence through an implicit approach. According to the experts from the company, the state of the market should influence the decision about the success/failure of the bids. This idea is vastly explored in the economical theories. For example, in the research dedicated to the real estate market, it is long established the importance of not only the global economy/business cycles but also property cycles. The economy/business cycles are described as intervals of either expansion or recession of economical activity. The property cycles on the other hand are divided into Boom, Slump and Recovery. The relationship between these cycles is still a matter of debate between scholars. The seminal work on property cycles is attributed to Homer Hoyt in his doctoral dissertation [12]. Here, we will try to create the similar notion for our particular market by using Hidden Markov Models.

5.1. Hidden Markov Model

We consider a time discretization, with the time instances denoted by t, and the state of the system denoted by Q_t, with $q_t \in \{S_1, S_2, ..., S_N\}$. We say that $Q = (Q_t, t \in \mathbb{N})$ is a Markov chain if

$$\begin{aligned} &P(Q_t = S_j | Q_{t-1} = S_i, Q_{t-2} = S_k, ...) \\ &= P(Q_t = S_j | Q_{t-1} = S_i), \quad \forall t, i, j, k \end{aligned} \tag{12}$$

If, in addition, this probability does not depend on time t, then Q is an homogeneous Markov chain and we define:

$$a_{ij} = P(Q_t = S_j | Q_{t-1} = S_i) \quad \forall i, j, t \tag{13}$$

The matrix $A = \{a_{ij}\}_{1 \le i,j \le N}$ is called transition probability matrix.

In case the states sequence (Q_t) is not observable but can only be observed through another stochastic process $(\theta_t, t \in \mathbb{N})$, taking values in $\{V_1, V_2, ..., V_M\}$, then we say that (Q_t) is an Hidden Markov process. In that case, besides the transition probability matrix A, we also need the following probability matrix of observations:

$$\begin{aligned} &B = (b_{ju})_{j \in \{1,2,...,N\}, u \in \{1,2,...,M\}} \\ &\text{with} \quad b_{ju} = P(\theta_t = V_u | Q_t) \end{aligned} \tag{14}$$

Finally, as for all Markov processes, we also need to specify the distribution of the initial state of Q : $\pi = (\pi_i)_{i \in \{1,2,...,N\}}$, where $\pi_i = P(Q_0 = S_i)$. Hence a HMM is characterized by the tuple (A, B, π).

An inference regarding HMM regards the estimation of A, B and π, such that

$$(\widehat{A}, \widehat{B}, \widehat{\pi}) = \text{arg max } \{P(\theta | A, B, \pi)\} \tag{15}$$

where $\theta = \theta_1 \theta_2 ... \theta_T$ is the sequence of T observations of the stochastic process θ.

This optimization problem is usually solved using the Braum-Welch algorithm [7]. In addition, once the parameters A, B and π are estimated, we can then obtain the sequence of the most likely states of the Markov Chain, Q. For this, the Viterbi algorithm [24] is used.

Returning to our problem in hands, we consider that the HMM has two states, with time discretization corresponding to one quarter. One of the states corresponds to "Stable Market", whereas the other corresponds to "Turbulent Market" (in the sense that it is highly volatile). The observable process is the 1-quarter percentage change in the MI, denoted Fluctuation.

5.2. Python Implemmentation

We use the `hmmlearn` package in Python for the estimation of the HMM parameters (A, B, π). For the model it is sufficient to indicate the number of hidden states to be estimated and then, by calling the `fit()` function the HMM model can be trained. The inferred optimal hidden states can be obtained by calling the `predict()` function.

We decide to train the HMM with the new variable, Fluctuation, that corresponds to the percentage change between the MI from consecutive quarters. We test several possibilities for the number of states: two, three and four. The four and three states models end up with artificial data separations without a clear economic meaning.

For the HMM with two hidden states we obtain the following results for the states and the estimated transition matrix:

Table 9: Market States and estimated transition matrix obtained from the HMM

States		
	0	1
Mean	-0.189	0.68
Variance	7.81	0.353
Name	Turbulent	Stable
Transition matrix A		
States	Turbulent	Stable
Turbulent	0.871	0.129
Stable	0.049	0.951

In Figure 6 we present the values of the Fluctuation variable and the most likely states of the HMM, colored by green (red) for the Stable (Turbulent) states. The identified states clearly divide the market into two categories, a more optimistic, stable and healthy state (Stable Market) and a more pessimistic, volatile one (Turbulent Market). We can clearly see that the market fluctuations, when the market in Stable, are smaller in percentage and usually do not vary drastically between positive and negative changes. On the other hand, in the Turbulent state we see that the percentage market changes are much higher in absolute value and it is common to have a positive MI change followed by a negative change indicating volatility and instability in the market at the time.

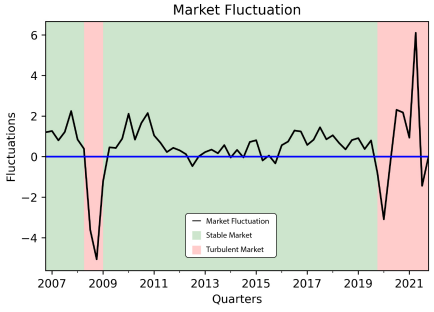
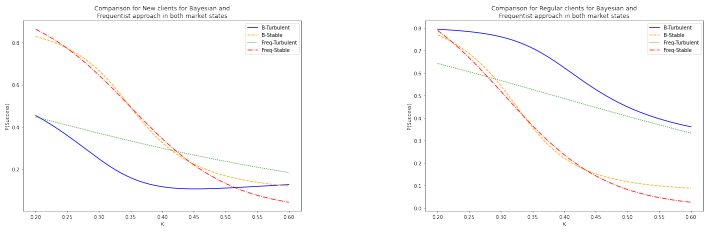


Figure 6: Market Fluctuation values and hidden Market States

We can then construct new frequentist and Bayesian models conditioned on the two market states from the HMM model. In Figure 7 we have the probability curves for the

frequentist model and the mean curve obtained from 2000 samples of the posterior predictive distributions from the Bayesian model with generalized t-Student prior distributions.



(a) (b)

Figure 7: Probability curves for the frequentist and Bayesian models in Stable and Turbulent Market states for New and Regular clients

For the clients in the New category the relation between Stable and Turbulent Market curves seem to be as expected, in the sense that the higher probability curves correspond to the Stable Market state while the lower probability curves correspond to the Turbulent Market state. On the other hand, for clients in the Regular category, the Stable/Turbulent Market relationship is the opposite, i.e., for both the frequentist model and the Bayesian model the company is expected to perform better in Turbulent Market situations than in Stable Market situations with clients.

Upon further consideration about this situation it is possible to deduce a reasoning for this relationship. In dire times the Regular clients tend to give preferential treatment to the company due to their prior relationship. On the other hand, when the market is Stable the Regular clients do not have too much pressure to choose a safer, well known company and can afford to take some risks by considering other companies. This translates into the company having better performance in Turbulent states rather than in Stable states with this set of clients affected by their prior relationship and the fact that they constitute a safe harbour in more volatile times for this clients.

Considering the New clients, the prior rationale cannot be applied. When the market is in Stable states the company is expected to perform better with these clients whereas when the market is volatile these clients are more apprehensive in engaging in business with companies with whom they do not have an already established relationship.

These important market relationships are captured by Market States models and ignored by the MI models. The market states of the HMM transmit a relation much closer to the property cycles also described in [12]. Albeit, the property cycles mentioned have three phases, Boom, Slump and Recovery, here we limit the number of states to two, Stable and Turbulent. This classification provides the models with a different relationship when compared with the MI variable. We can look at the HMM states as a filter for MI. It removes the excess noise that is not relevant for our problem, exposing this useful relationship.

6. Conclusions

At last, we can then develop models for all the different approaches mentioned previously. In order to compare all the models we need first to standardize the testing process. In each testing iteration the full dataset is divided into 4 different partitions in every turn corresponding to the following: $Train\_S$ containing the training observations in the Stable market category; $Test\_S$ containing the testing observations in the Stable market category; $Train\_T$ containing the training observations in the Turbulent market category; $Test\_T$ containing the testing observations in the Turbulent market category;

The partition is done in such a way that the number of observations in each partition remains constant. The models with the MI or Fluctuation variables are trained using the combined dataset $(Train\_S + Train\_T)$ and for testing $(Test\_S + Test\_T)$, while the models that take advantage of the market state classification are trained separately, i.e. two models are created, one for the Stable observations and other for the Turbulent observations and the prediction results are gathered and combined in the end in order to compare those results with the remaining models.

The frequentist model (Freq) results correspond to the results obtained with the model whose coefficients are estimated using the redescending M-estimators method described previously. Then, the (Freq F) corresponds to the frequentist model similar to the one just described but now considering the variable with the fluctuation values instead of the absolute MI values. At last, the remaining frequentist model (Freq-States) results correspond to the aggregate of the results from two frequentist models constructed each considering the Stable market observations or the Turbulent market observations, also with the robust algorithm.

Additionally, we have several Bayesian models with different configurations. In the Uniform case, the prior distributions are all Uniform distributions centered on the coefficient values used for the frequentist model and range 10, i.e. taking the example of the prior distribution for β_{MI}, it is $Uniform(\widehat{\beta}_{MI} - 10, \widehat{\beta}_{MI} + 10)$, where $\widehat{\beta}_{MI}$ corresponds to the value of β_{MI} in the frequentist model. The same methodology is applied to the other coefficients. In the t-Student case, the prior distributions are all generalized t-Student distributions with location parameter $\mu = \widehat{\beta}_i$, where $\widehat{\beta}_i$ is the value used in the frequentist model for the coefficient β_i, scale parameter $\sigma = 10$ and degrees of freedom $\nu = n - 1$, where n is the size of the training set. For the Mixed case, the prior distribution for the coefficients correspond to a mixture of generalized t-Student, Chi-Squared and Uniform distributions, namely, β_K, β_F and β_U have Uniform priors identical to the ones used in the Uniform case, β_{MI}, β_N and β_R have generalized t-Student priors such as in the t-Student case, and β_0 has a $\chi^2_{(5)}$ distribution. Finally, we have the results from the Bayesian model without the MI or the Fluctuation variables, denoted B-States, where two Bayesian models are constructed, one for each partition of the data into the Stable and Turbulent market states. These two Bayesian models have, as priors, the same distributions for $\beta_0, \beta_K, \beta_N, \beta_R, \beta_F, \beta_U$ as the ones considered in the Bayesian model with generalized t-Student prior. The AUC scores correspond to the scores obtained after grouping the predictions of both models for the respective test partition and then computing the AUC score with all the predictions.

Analysing Figure 8, there are several conclusions that can be drawn. When looking at the frequentist models, we see that the first frequentist model is the worst performing model of all the models tested. Then, we immediately see a clear improvement in the results with the model where the MI variable is replaced by the Fluctuation variable. Furthermore, the frequentist model with the market states is the best performing model out of the three frequentist models tested. While

it has a larger variance than the frequentist model with fluctuation, the $(25\%, 75\%)$ quantiles are considerably higher in the former model and it also reaches higher AUC values in general.

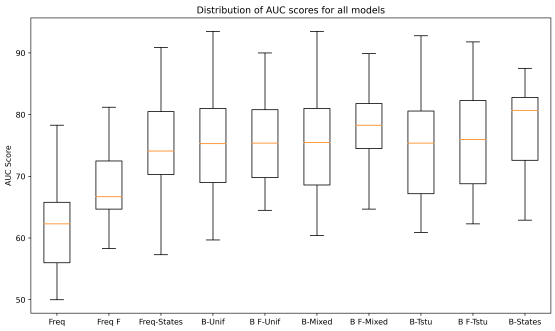


Figure 8: Boxplot for AUC scores of each model constructed

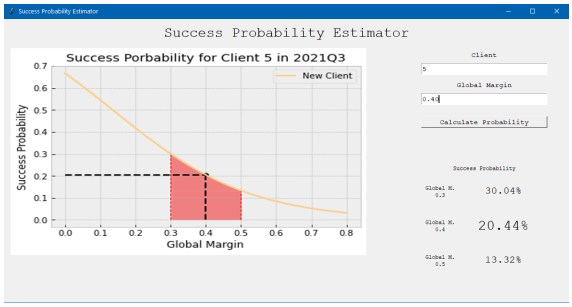
Considering the Bayesian models, the first major conclusion to take is that the overall performance of the models is relatively invariant to the distributions chosen for the priors. When considering all three different sets of prior distributions, the performance of those models is very similar. Additionally, contrary to the frequentist case, when considering the Fluctuation values instead of the MI values there is not a significant, but only slight, increase in the performance of the models. Furthermore, the Bayesian model with the market states has the best performance of all the other Bayesian models tested. Although it does not reach the highest levels of AUC scores, its variance is one of the lowest registered, similar to the Bayesian model with the Fluctuation variable and with a mixed configuration of prior distributions, and the $(25\%, 75\%)$ quantiles are the highest of all the models. It has the highest median at 80% AUC score, the highest 75% quantile, and the 25% is the second highest.

When comparing the frequentist and the Bayesian models we see that the Bayesian models have better performance. The frequentist model with the market states is the best performing frequentist model and even then its performance is on par with the worst performing Bayesian counterpart. This result had already been corroborated by the company managers when they mentioned that the more conservative approach of the Bayesian models more closely resemble the day-to-day situations.

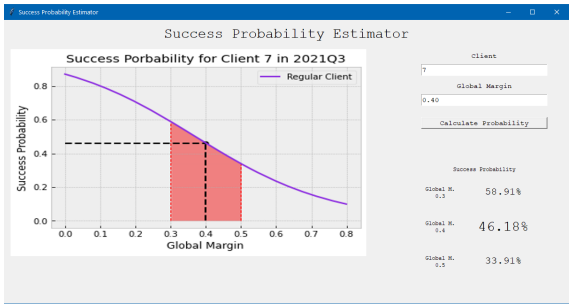
We can conclude that we successfully achieved the objective of creating a market classification procedure that divided the market into Stable and Turbulent states using HMM [21] and then, the models created with that market classification and without any additional information from the market index offer better performance than the models with information from the market index. This means that the classification procedure is able to encapsulate the influence of the market while, at the same time, by not having to include additional information in the models.

Regarding the other objective of this project, developing a decision support tool for the company managers, we can take advantage of several Python routines to construct a Graphical Interface that incorporates the obtained models in an easy to use and manipulate tool.

In Figure 9 we have two different stages of the Graphical User Interface (GUI) created. These images reflect what EQS managers observe when using the decision tool. We show the example where the manager is considering a proposal with a global margin value of 0.40 (40%) for the client with ID 5 in the second quarter of 2021. After clicking "Calculate Probability" the decision tool shows what is presented in Figure 9(a). In the lower right side, the manager can see the estimated success probability for the K value inserted in the previous text field, as well as, the success probability values for proposals with global margins of $K + 0.10$ and $K - 0.10$. In the left side, the decision tool shows a plot of the evolution of the success probability as a function of the global margin K for the client category of client 5, which, in this case, corresponds to a New client. The black dot and dashed black line show the K value defined by the manager and its corresponding success probability for this client. The red region constitutes the probability region between $K-0.10$ and $K+0.10$ which, according to the managers, corresponds to the region the managers will focus on when calibrating the proposal.



(a) Output for New Clients



(b) Output for Regular Clients

Figure 9: Decision Tool Graphical Interface

In Figure 9(b) we have the output for a different client, client 7, that is in the Regular category. We see that, we have a different probability curve and different success probability values for the same levels of K. This tool allows then the company managers to better adjust their proposals in future tenders and also have more confidence in their proposals knowing that there is a data driven model backing their decisions.

This tool was presented to the managers and board members of EQS and id expected to be incorporated in the day-to-day operations. The Python code for this project can be accessed on the following Github page [19].

Overall, in this project we are given a real life problem that a Portuguese company is facing where there is a lack of information on the impact of the state of the market on the performance of its proposals in tenders. In order to address this issue and to properly study the influence of the market on the performance of the company we develop a procedure with multiple steps. In each step, a different problem needs to be addressed and a different area needs to be explored. In the end, not only are we able to construct several methods to

study the market influence through different approaches, but we also create models that predict the success probability of future tenders and incorporate those models into a business decision tool that can be used by the company managers to improve their performance in future cases. Thus this project's goals were achieved both at an academic and business level with satisfactory results in both ends.

References

[1] Eurostat: Statistical office of the european union.

[2] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[3] C. Andrieu and J. Thoms. A tutorial on adaptive mcmc. *Statistics and computing*, 18(4):343–373, 2008.

[4] C. Á. Benítez. Analytical methods for logistic tenders. Master's thesis, Universitat Oberta de Catalunya (UOC), 2019.

[5] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[6] S. O. Cheung, P. S. P. Wong, A. S. Fung, and W. Coffey. Predicting project performance through neural networks. *International Journal of Project Management*, 24(3):207–215, 2006.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[9] M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[10] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[11] M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[12] H. Hoyt. *One hundred years of land values in Chicago: the relationship of the growth of Chicago to the rise of its land values, 1830-1933*. Beard Books, 2000.

[13] D. Kornbrot. Point biserial correlation. *Encyclopedia of statistics in behavioral science*, 2005.

[14] V. Krishnan. Constructing an area-based socioeconomic index: A principal components analysis approach. *Edmonton, Alberta: Early Child Development Mapping Project*, 2010.

[15] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera. *Robust statistics: theory and methods (with R)*. John Wiley & Sons, 2019.

[16] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[17] S. T. Ng, S. O. Cheung, M. Skitmore, K. C. Lam, and L. Y. Wong. Prediction of tender price index directional changes. *Construction Management and Economics*, 18(7):843–852, 2000.

[18] S. T. Ng, S. O. Cheung, M. Skitmore, and T. C. Wong. An integrated regression analysis and time series model for construction tender price index forecasting. *Construction Management and Economics*, 22(5):483–493, 2004.

[19] B. Pires. Market tender performance. https://github.com/brunomrpires/market$_{t}ender_{p}erformance, 2021$.

[20] D. Pregibon. Resistant fits for some commonly used logistic models with medical applications. *Biometrics*, pages 485–498, 1982.

[21] L. Rabiner and B. Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[22] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

[23] P. Sur, Y. Chen, and E. J. Candès. The likelihood ratio test in high-dimensional logistic regression is asymptotically a rescaled chi-square. *Probability theory and related fields*, 175(1):487–558, 2019.

[24] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

[25] T. P. Williams. Predicting changes in construction cost indexes using neural networks. *Journal of construction engineering and management*, 120(2):306–320, 1994.

[26] K. Yu. The economics of construction price inflation: Measurement. *Output and Productivity (Doctoral Dissertation, UCL (University College London)*, 2014.

[27] M. Zhang, G. Johnson, and J. Wang. Predicting takeover success using machine learning techniques. *Journal of Business & Economics Research (JBER)*, 10(10):547–552, 2012.