Imperial College London

Department of Mathematics

# Interest Rate Prediction with Twitter Sentiment

*Author:* Francesco Sciacovelli (CID: 01790720)

A thesis submitted for the degree of

*MSc in Mathematics and Finance, 2019-2020*

## Abstract

The growing popularity of social media produced, coincidentally, a vast database of people's sentiment throughout time. The aim of this paper is to use this sentiment organically as a new parameter in the prediction models. We will analyse how the prediction of a country's interest rate can be enhanced when taking into consideration people's sentiment. We will focus on the sentiment sparked from specific events in history. First analysing the events together in order to create a sentiment variable spanning through a longer period of time; then we will analyse four events independently to study how the choice of events influences predictions. We will see that non-linear models are able to incorporate the information contained in the sentiments in a much better way than the linear model; moreover, it will be clear how the choice of tweets to analyse has to comprehend tweets closely related to the country studied.

**Acknowledgements**

I would like to thank my supervisor, Dr Damiano Brigo, for suggesting this topic and providing guidance and supported whenever needed. Writing this paper has been an enjoyable and enriching experience.

I would like to thank the Mathematics and Finance section at Imperial College London for the great opportunity to undertake this MSc programme. It has been an incredible journey that I will cherish in the years to come.

Finally, I would like to thank my family for their love and support; and my friends, the old and the new ones, for making this year one to remember.

# Contents

# List of Figures

# List of Tables

# Introduction

The forecasting of time series data has always been a crucial factor in different fields of study. Economics, and finance in particular, represents one of those fields where the correct forecast of a variable could lead to life-changing effects. Among the variables that economists attempt to predict, we find inflation, exchange rates and clearly stock market indexes. In this context, we found the risk-free interest rate to be one of those variable that heavily impact the whole financial market. Indeed, this is used in the valuation of derivatives, e.g. even the easiest version of the Black-Scholes formula uses the risk-free rate as one of its parameters, as well as for the calculation of the Sharpe ratio, which measures the performance of an investment, and in the capital asset pricing model (CAPM), where the return of an asset is expressed in terms of excessive return over the risk-free rate. Due to its great importance in the financial market, there exists a numerous quantity of models used to predict its future value. This paper wants to further analyse and extend the existent model outlined in the article *An Efficient Deep Learning Based Model to Predict Interest Rate Using Twitter Sentiment* by Yasir M., et al ([1]). In this case, as a proxy for the risk-free rate, we decided to use the 1-Month treasury bond. The aim of this paper will then be to forecast the change in the 1-Month treasury bond for four different countries: the United Kingdom, Japan, Mexico and Switzerland, using two different sets of predictors: in the first case we will use the change of the exchange rate of each country's currency against the US dollar; in the second case we will add, to this first predictor, people's sentiments expressed on the famous platform Twitter. We will analyse two different influences of Twitter sentiment; first of all we will use a dataset containing sentiments for different events that spanned between 2015 and 2016 to see whether this will help our models. Later on, we will analyse whether the choice of the events taken into consideration is relevant in terms of accuracy improvement. To do so, we will forecast the 1-Month treasury bond for each country using the exchange rate against the US Dollar and the sentiment from just one event, to see if there are events that improve the prediction more than others or not. Core element of the whole analysis is assigning sentiments to each tweet. This process is divided into different steps. First of all we proceed to 'clean' each tweet, meaning that we remove all words that do not convey any sentiment; we also transform each tweet in a simpler version of itself in order to facilitate sentiment assignment. We then use a pre-trained neural network to assign sentiment to each tweet; 0 will be assigned to the *negative* tweets and 1 to the *positive* ones. The whole procedure is explained more in depth in the *Data and Methodology* section ( Section 1), which also explain the functioning of each prediction model as well as introducing the evaluation metrics used throughout the paper. Section 2 shows how the data has been transformed before using it as an input, as well as highlighting important characteristics of it. We are then ready to proceed to the actual forecasting of the 1-Month treasury bond; in Section 3 we show the errors made by each prediction. We first analyse whether the Twitter sentiment is a useful input in the prediction models and then we examine if the choice of the events is somehow relevant when predicting the 1-Month treasury bond. Following this, we find the last section which collects the conclusion.

# Literature review

The importance of the risk-free rate in finance, urged economists to propose new and different forecasting models for it. Recently, the focus has been concentrated on those models that are able to capture complex interactions among data. Model that are able to capture nonlinear relationships are becoming more and more popular in recent research. For example, a model that assumes the possibility of changes in regime (Markov switching model) for the prediction of the 3-months Treasury bonds between 1962 and 1987 has been proven to be far more accurate than a simple linear model ([2]). These can be explained by the fact that non-linearity is present in the interest rates because of the stochastic swings that characterise it ([3]). Furthermore, different researches show how the use of artificial neural networks (ANN) models provide robust results even when the sample size is low ([4]). We can summarise recent literature about this topic, dividing it into four different streams. The first one use both linear and non-linear models to link the change in the yield curve to price levels and GDP. The spread among short-term and long-term bond rates has been associated to different macroeconomics factor, such as economic growth, recession indices and industrial production ([5] and [6]). The second stream is characterised by 'data-driven' models; in this case the focus is on using mathematical models to interpolate the yield curve, for example splines models ([7]) and parsimonious models ([8]) are used. The third stream uses the so-called 'dynamic models'; in this case arbitrage-free models ([9]) and equilibrium models ([10]) are used. The fourth stream makes use of data-driven models as well but, in addition to the ones used in the second stream, these models are able to manage complexity in the data, such as non-linearity and seasonality issues. In this context, we find different studies using neural networks and cased-base reasoning ([11] and [12]), where it is shown how including structural changes in the economy (due to government economy policy) leads to more accurate results.

Next to the literature regarding interest rates forecasting, it is important to analyse the one concerning sentiment analysis as well. Indeed, this approach has become more and more prominent in recent literature. Mining of public opinions and emotions revealed to be useful in the most different fields. From politics ([13]) to disaster management ([14]), traffic management ([15]) and clearly, finance ([16]). Each paper that uses sentiment analysis approaches the problem differently. Indeed, there exists numerous different methods to gather and process public opinions. A comparative study on sentiment analysis approach ([17]) shows how the three most popular models used for sentiment analysis using deep learning are: Deep Neural Network (DNN, [18]), Convolutional Neural Network (CNN, [19]), and hybrid methods ([20]). Furthermore, it can be noticed that even though the methods used are different, most studies share a common feature. Indeed, in most cases, the text features are transferred into word embedding using the Word2Vec tool ([16]) before being passed to the chosen deep learning method.

# Chapter 1

# Data and Methodology

In this section, we will thoroughly examine the methodology used to clean the data, fundamental issue when dealing with raw tweets, and to analyse it. The building blocks of our analysis are three sets of data: the 1-month treasury bond, which is the variable we aim to predict, the exchange rate against the USD, which will be our independent variable, and finally a dataset of tweets concerning five major events. The first step is to gather the treasury bond and exchange rate data for the four different countries taken into consideration: UK, Japan, Mexico and Switzerland in the years between 2016 and 2018. We will then use three different prediction models; linear regression, support vector regression and deep learning to predict the treasury bond. The following step is then to use the twitter sentiment as an additional parameter to investigate whether the accuracy of the prediction is enhanced or not.

## 1.1 Tweets cleaning

The dataset provided by Dr. Zubiaga ([21]) contains 30 different data sets, each one concerning one specific event between the years 2012 and 2016; the Superbowl of 2012 characterises the first dataset and the earthquake in Ecaudor of 2016 characterises the last one. For our analysis we will analyse a small subset of these datasets.

For each event, the author of the data set fixed a temporal frame; which could vary from just a couple of days up to a month, and some characterising hashtags, so that every tweet posted during the fixed timeframe, containing at least one of the characterising hashtags, has been saved in the data set. This process results in the creation of exhausting datasets which contain from hundreds of thousands of tweets up to millions of tweets.

Once we have all these tweets, the natural step is to 'clean' them. This procedure takes into consideration different problems related to how people write when on Twitter and also related to how we are going to assign a sentiment to each of these tweets.

The first issue generates the presence of misspelled words as well as words where a letter is repeated more than necessary in order to stress the meaning of the word itself (e.g *'Their is a bad thunderstorm outside'* instead of *'There is a bad thunderstorm outside'* or *'I looove ice cream'* instead of *'I love ice cream'*)

The second issue cause the need of more drastic changes in the structure of the sentence. First of all, we have to deal with negations ([22]), meaning that when we find a verb or an adjective preceded by a negation, for example the classic *'not'*, we then need to actively replace the construct *negation + verb(/adjective)* with the antonym of the verb(/adjective). In order to do so, we previously have to deconstruct all contractions, which are very frequent in an informal setting as Twitter. Furthermore, we need to eliminate all the elements that do not convey any sentiment, such as numbers, punctuation, websites, tags etc. The exact code used to clean all data sets is provided in the Appendix A. Table 1.1 shows the exact events taken into consideration, as well as how many tweets we have for each event and the timeframe used to gather the tweets. The number of tweets represent the number of significant tweets; meaning that we already eliminated all those tweets containing only an hashtags, a link, a tag etc.

| Event | Dates | Number of tweets |
|---|---|---|
| Charlie Hebdo shooting | 07/01/2015 - 14/01/2015 | 1,493,917 |
| Germanwings plane crash | 24/03/2015 - 30/03/2015 | 890,853 |
| Nepal earthquake | 25/04/2015 - 18/05/2015 | 7,174,962 |
| Hurricane Patricia | 24/10/2015 - 08/12/2015 | 732,559 |
| Irish election | 03/02/2016 - 06/03/2016 | 215,564 |
| Brexit | 24/02/2016 - 03/05/2016 | 657,276 |
| Brussels Airport explosion | 22/05/2016 - 30/05/2016 | 2,413,014 |
| Lahore protest blast | 27/03/2016 - 30/03/2016 | 582,846 |
| Cyprus hijacked plane | 29/03/2016 - 30/03/2016 | 278,495 |
| Panama papers | 03/04/2016 - 03/05/2016 | 2,774,399 |
| Ecuador earthquake | 17/04/2016 - 28/04/2016 | 148,414 |

Table 1.1: Details of tweets

## 1.2 Prediction models

The three models used in our analysis are very different from each other and it is therefore beneficial to investigate each one of them more in depth.

### 1.2.1 Linear Regression

This is the simplest model out of the three we use. Our variables can be written as $(x_i, y_i)_{i=1,\dots,n}$ where $x_i$ represents the exchange rate at time $i$ and $y_i$ represents the treasury bond at the same time $i$.

In this case we assume that our dependent variable has a linear dependence on the independent variable; meaning that

$$y_i = f(x_i) + \epsilon_i \tag{1.2.1}$$

where $f(\cdot)$ is a linear function: $f(x) = \alpha + \beta x$, and the sequence of $\epsilon_i$ are centered independent random noises with constant variance. We can express 1.2.1 in a more compact form as:

$$\mathbf{Y} = \alpha \mathbb{1} + \beta \mathbf{X} + \boldsymbol{\epsilon} \tag{1.2.2}$$

where clearly $\mathbf{Y} = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$, $\mathbf{X} = (x_1, \dots, x_n)^\top \in \mathbb{R}^n$, $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)^\top \in \mathbb{R}^n$, and $\mathbb{1} = (1, \dots, 1)^\top \in \mathbb{R}^n$.

### 1.2.2 Support Vector Regression (SVR)

The Support Vector Regression can be seen as an extension of the Support Vector Network (SVN). Both are based on the idea of mapping the input, through a non linear map chosen a priori, into a higher dimensional space; then a linear decision surface is constructed in this space. Meaning that we construct a hyperplane which distinctly classifies the data points, clearly in two dimensions this hyperplane is simply a line (Fig. 1.1 [23]).

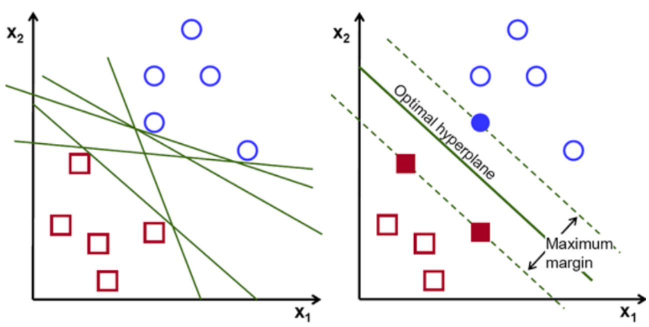



Figure 1.1: Example of SVR in 2 dimensions

The support vectors are those data points closer to the separating hyperplane, influencing its position and orientation. It is important to notice that through the mapping into a higher dimensional space, this methodology is able to perform non-linear classification. To give more structure to the SVR, we can assume the data to be written as $(\mathbf{x}_i, \mathbf{y}_i)_{i=1,\dots,n}$; then, we take a function:

$$\phi : \mathbb{R}^n \to \mathbb{R}^N \quad \text{where } N > n$$

which will be our map into a higher dimensional space.

Then what we have for our prediction is:

$$f(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) - b$$

Finally, we fix a tolerance $\epsilon$, and define as $\xi$ the distance from the tolerance interval of points on one side of the hyperplane and $\xi^*$ the same distance, but of points on the other side; ending up with the situation shown in Fig 1.2 ([24]); these are called *slack variables*. Thanks to the introduction of these further variables, the algorithm allows us to choose how tolerant we are of errors, both through an acceptable error margin ($\epsilon$) and through tuning our tolerance of falling outside that acceptable error rate. In the end then we have to solve the minimisation problem expressed by

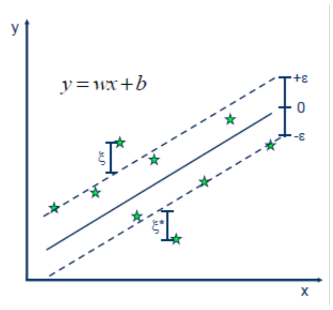



Figure 1.2: SVR in $\mathbb{R}^2$ with $\epsilon$ as the confidence interval

equation 1.2.3

$$\min \left( \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=0}^{N}(\xi_i + \xi_i^*) \right) \tag{1.2.3}$$

subject to the constraints:

$$\begin{aligned} \mathbf{y}_i - \mathbf{w}\phi(\mathbf{x}_i) - b &\leq \epsilon + \xi_i \\ \mathbf{w}\phi(\mathbf{x}_i) + b - \mathbf{y}_i &\leq \epsilon + \xi_i^* \\ \xi_i \ , \ \xi_i^* &\geq 0 \end{aligned}$$

The parameter $c$ acts as a trade-off between simplicity and generalisability; indeed, lower the value of $c$ is, easier the model will be since we are not penalising too harshly falling outside the tolerance interval, on the other hand higher the value of $c$ more complicated the model will be but less data points will fall outside the tolerance interval.

It can be proved that the vector $\mathbf{w}$ obtained from this minimisation problem can be written as a linear combinations of the support vectors ([25]), meaning that it can be written as in Eq 1.2.4 if we assume to have $l$ support vectors.

$$\mathbf{w} = \sum_{i=1}^{l} y_i \alpha_i \phi(\mathbf{x}_i) \tag{1.2.4}$$

So we can rewrite our prediction as in Eq. 1.2.5

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \tag{1.2.5}$$

Furthermore, it is important to notice that, in our context, we will use the kernel mapping approach, meaning that the dot product in Eq. 1.2.5 is expressed by a kernel function which in our case will be the so-called *Gaussian Radial Function* expressed in Eq. 1.2.6

$$K(\mathbf{x}, \mathbf{x}') = \exp\left( - \frac{\|\mathbf{x} - \mathbf{x}'\|}{2\sigma^2} \right) \tag{1.2.6}$$

### 1.2.3 Deep Learning

Deep learning is a prediction method that uses Neural Networks (NNs) as its main tool. In our case we will use a Feed-forward Neural Network (FNN), which differs from a Recurrent Neural Network (RNN) where the output is fed back into the neural network. Following [26], we can define a feed-forward neural network as below:

**Definition 1.2.1.** Let $I, O, r \in \mathbb{N}$. Let $d_i \in \mathbb{N}$ represents the units in the $i$-th hidden layer, where $i = 1, \dots, r-1$, and $\boldsymbol{\sigma}_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_i}$, where $i = 1, \dots, r$, and $d_r := O$. Then a FNN is a function $f : \mathbb{R}^I \to \mathbb{R}^O$, with $r-1$ hidden layers if we can write:

$$\boldsymbol{f} = \boldsymbol{\sigma}_r \circ \boldsymbol{L}_r \circ \cdots \circ \boldsymbol{\sigma}_1 \circ \boldsymbol{L}_1$$

where $\boldsymbol{L}_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$, for any $i$, is an affine function

$$\boldsymbol{L}_i(\boldsymbol{x}) := W^i \boldsymbol{x} + b^i, \quad \boldsymbol{x} \in \mathbb{R}^{d_{i-1}}$$

parameterised by a weight matrix $W^i = [W^i_{j,k}]_{j=1,\dots,d_i, k=1,\dots,d_{i-1}} \in \mathbb{R}^{d_i \times d_{i-1}}$, and bias vector $\boldsymbol{b}^i = (b^i_1, \dots, b^i_{d_i})$ with $d_0 := I$.

Figure 1.3 ([26]) is a good way to visualise this complex definition; where each line represents the use of an affine function, while the dots are the places where the activation functions are used.

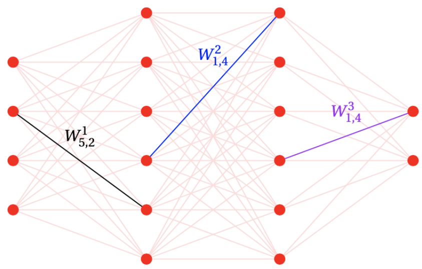



Figure 1.3: Example of FNN with $I = 4$, $O = 2$, and $r = 2$ where $d_1 = d_2 = 6$.

This method requires then to choose the value of $r$, as well of $d_1, \dots, d_{r-1}$; moreover, we have to decide what kinds of functions to use as activation functions.

Furthermore, another important element when training a neural network is to decide the loss function. This will be the function that our neural network tries to minimise. There exists a multitude of loss functions depending on the distribution of the data as well as the aim of the neural network itself (e.g. there exists loss functions specifically for label classification problems and other ones for predictions problems).

## 1.3 Assigning sentiment

Core step of our analysis is to assign a sentiment to the tweets. In order to do so we will use a technique ([27]) which uses the *Word2Vec* algorithm and then a neural network, both trained on a dataset of over 1 billion tweets already classified in Positive and Negative.

The Word2Vec algorithm is used to map words into a multidimensional space, which assign to each word its own vector, in such a way that words that have similar meaning are indeed close to each other. This algorithm also creates a vocabulary containing all the words 'seen' during the training period which could be investigated to see how general our model is, we could simply link a larger vocabulary to a more general model which is then more suitable to be used on new tweets where the likelihood to find a new world is low. The main purpose of this algorithm remains its ability to assign to each word a specific multidimensional vector. This vector structure will then be used by our neural network.

In this process we will use an evolved variant of a Recurrent Neural Network (RNN), which could be easily explained as a FNN where we allow cyclical connections ([28]). The basic version of a RNN presents a major problem: *short-term memory*; meaning that when dealing with a large amount of data (i.e. a large paragraph of text) it is not able to carry information from the beginning of this data resulting in an output which is actually influenced by a smaller portion of this data (i.e. the ending lines of the paragraph). To solve this problem we use a GRU network which takes care of the short-term memory of the RNN ([29]). To core concept of GRU are the cell state, one is illustrated well in Figure 1.4, and its two gates. The update gate decides how much information of the past to keep and how much new information to add; on the other hand the reset gate has the only function of deciding the amount of information to forget before going to the next gate. Thanks to these two gates, the neural network is able to take into account even past information when producing an output. Furthermore the neural network will also contain an

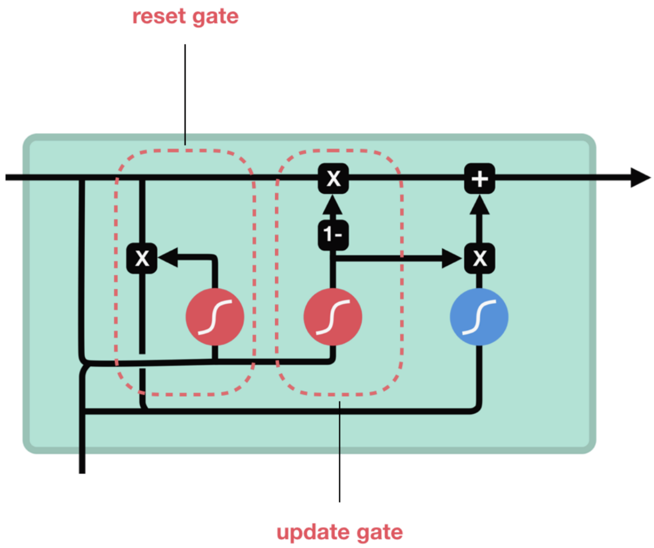



Figure 1.4

additional layer characterised by the sigmoid activation function, defined in Eq.1.3.1, which will produce the one-dimensional output needed from the neural network.

$$f(x) = \frac{1}{1+e^{-x}} \tag{1.3.1}$$

In this case, the loss function used is the *binary cross-entropy* (Eq. 1.3.2). This is the loss function which is usually used when we need to assign to our input one of two classes; like in this case we have to decide for each tweet whether this is negative (assigning in this case 0) or positive (assigning 1). The neural network will actually output numbers ranging from 0 to 1 and not just 0 and 1; so we could look at the output as the probability of that specific tweet to be positive, so that the closer this number is to 1 the more likely this tweet is positive, while the close is to 0 the more likely this tweet is negative (since these are the two only possible alternatives).

$$\ell(y, \hat{y}) = -y \log \hat{y} - (1-y)\log(1-\hat{y}) \quad \text{where} \quad \hat{y} \in (0,1),\, y \in \{0,1\} \tag{1.3.2}$$

To train our neural network we need to divide our training data in two different sets; one that will actually be used to train the model and another one which is used to see how the model perform outside of the training set. The latter one will be called validation data or validation set. Furthermore, we will train the model not on the whole set of data at once but we will use the so-called *minibatches*; these are randomly drawn subsets of the whole training set. This procedure is used for two main reasons: first of all, it lowers significantly the computational time needed to train the model; and most importantly, it helps to avoid overfitting.

At the end of our training period the model arrived at an accuracy of around 78% on the validation set.

### 1.3.1 Using sentiment for prediction

For this last step we will use the same models used in the case without the sentiment. The only difference will be the dimension of the independent data. Indeed, if earlier we had $\mathbf{X} \in R^n$ we now have $\mathbf{X} \in R^{2n}$; since we will have a column representing the exchange rate and another column representing the sentiment. The only problem with this is the fact that while for the exchange rate we have a data point for each day, this is not true for the sentiment. Indeed, for each event we have thousands of tweets for each day; we therefore use the mean of the daily sentiment as our only data point for that day.

### 1.3.2 Evaluation metrics

To evaluate the accuracy of our predictions we use two different measures: the mean absolute error (MAE) and the root mean squared error (RMSE). The first one is simply the mean of the absolute difference between the forecasted value and the actual value; as shown in Eq. 1.3.3 . The second one, on the other hand, is the square root of the mean of the squared difference between the forecasted value and the actual one, as shown in Eq. 1.3.4

$$MAE = \frac{\sum_{t=1}^{n} |\tilde{y}(t) - y(t)|}{n} \tag{1.3.3}$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (\tilde{y}(t) - y(t))^2}{n}} \tag{1.3.4}$$

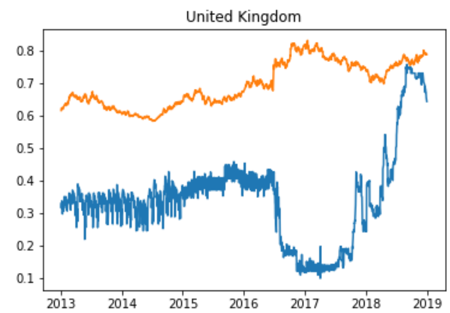
# Chapter 2

# Analysis and predictions

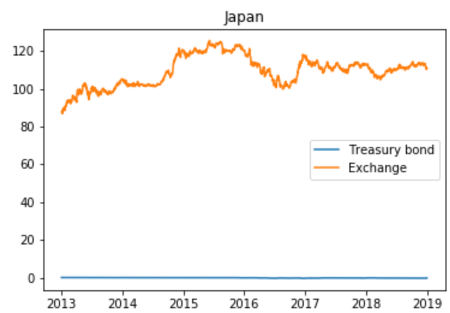
## 2.1 Data analysis

### 2.1.1 Main statistics

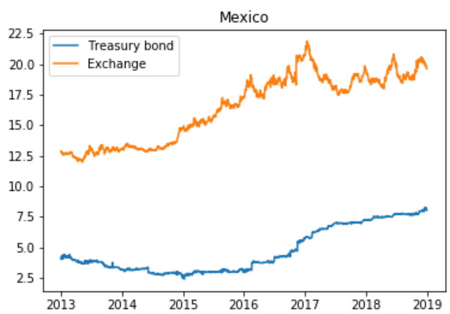
To build our prediction models we first need to have a clear picture of the data we are handling. We need to understand some of its characteristics as well as seeing whether it needs to be somehow scaled or transformed in order to make our prediction easier. In order to do so, we plot the 1-month Treasury bond and the exchange rate for the four different countries in order to visualise this data (Fig. 2.1); at the same time we draw a table containing the most significant statistics of the data.



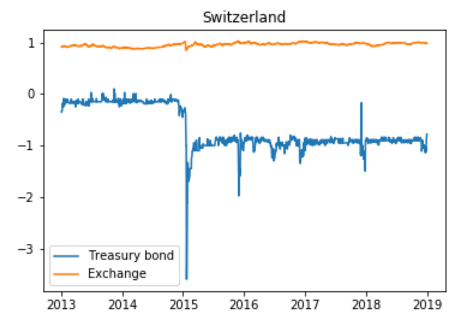

(a) UK




(b) Japan




(c) Mexico




(d) Switzerland

Figure 2.1: Data without scaling

The graphs shown in Fig. 2.1 show that our independent and dependent variable sometimes have very different values. In the extreme case of Japan, for example, the Exchange Rate values are more than 100 times the ones of the Treasury bond. This is also confirmed by Table 2.1 where we can clearly see how different the ranges for the two variables are for each country. Furthermore

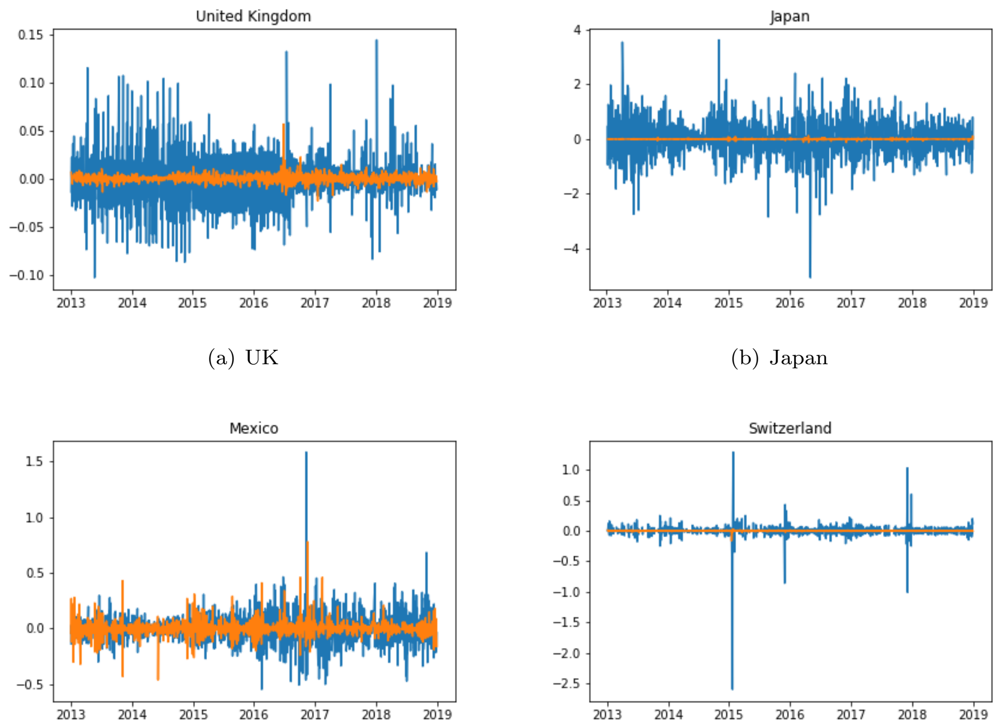
[1] In this case we actually mean excess kurtosis, meaning that this value would be zero for a normal distribution.

| | UK TB | UK ER | Japan TB | Japan ER | Mexico TB | Mexico ER | Switzerland TB | Switzerland ER |
|---|---|---|---|---|---|---|---|---|
| Range | [-0.26,0.40] | [0.63,0.83] | [-0.20,0.19] | [99.91,125.63] | [-0.96,1.01] | [14.49,21.90] | [-3.6,0.1] | [0.854,1.030] |
| Mean | 2.25 | 0.73 | 1.64 | 113.01 | 5.21 | 18.11 | -0.693 | 0.959 |
| Variance | 0.03 | 2.99 e-3 | 0.01 | 33.14 | 0.42 | 2.61 | 0.168 | 0.001 |
| Skewness [1] | 0.54 | -0.24 | -0.19 | 0.08 | 0.03 | -0.45 | -0.055 | -0.501 |
| Kurtosis | -0.18 | -1.23 | -0.65 | -0.48 | -0.65 | -0.47 | 1.693 | -0.623 |

Table 2.1: Statistics for the Treasury Bond (TB) and Exchange Rate (ER)

all the variables have very different means and variances which, again, is not a desirable property of the data. Furthermore, it is not of great interest to compare the actual value of the two quantities but, what is far more interesting, is to find how the change of one variable influences the other. In order to study this we decided to actually look at the first difference of these two time series.

Figure 2.2 shows the first difference for both the treasury bond and the exchange rate of the four countries. It is clear, from the graphs, how the data is now more comparable.




(a) UK

(b) Japan

(c) Mexico

(d) Switzerland

Figure 2.2: First difference of data

Furthermore, we can also look at Table 2.2 to see how the values are now of the same magnitude. Therefore, these will be the values on which we will run our models.

| | UK TB | UK ER | Japan TB | Japan ER | Mexico TB | Mexico ER | Switzerland TB | Switzerland ER |
|---|---|---|---|---|---|---|---|---|
| Range | [-0.10,0.14] | [-0.02,0.06] | [-0.13,0.13] | [-5.06,3.63] | [-0.46,0.78] | [-0.54,1.58] | [-2.60,1.29] | [-0.16,0.02] |
| Mean | 2.0e-4 | 1.1e-4 | -1.5e-4 | 2.3e-4 | 2.7e-3 | 4.5e-3 | -2.8e4 | 4.2e-5 |
| Variance | 6.0e-3 | 1.7e-5 | 2.4e-4 | 0.45 | 0.005 | 0.02 | 0.0.01 | 4.2e-5 |
| Skewness [2] | 0.49 | 1.69 | -0.93 | -0.21 | 1.49 | 1.10 | -8.92 | -10.67 |
| Kurtosis | 4.18 | 25.70 | 21.05 | 4.23 | 21.39 | 14.03 | 286.35 | 266.29 |

Table 2.2: Statistics for first difference of the Treasury Bond (TB) and Exchange Rate (ER)

[2] Again, we actually mean excess kurtosis, meaning that this value would be zero for a normal distribution.

### 2.1.2 ADF test

An important feature of a time series is whether this is stationary or not.

**Definition 2.1.1** (Stationary)**.** A time series $(X_t)_{t\in\mathbb{Z}}$, is said to be *(strictly) stationary* if

$$(X_{t_1}, \dots, X_{t_n}) \stackrel{d}{=} (X_{t_1+k}, \dots, X_{t_n+k}) \tag{2.1.1}$$

where $\stackrel{d}{=}$ represents the equality of joint distributions. Consequently, mean and variance do not change over time.

Therefore, if our time series is stationary, then its statistical features do not change over time, making it easier to predict.

There exists different ways to check for stationarity, but the most commonly used one is the *Augmented Dicky Fuller test.* This test assesses whether we can reject a null hypothesis against an alternative hypothesis. In this case the null hypothesis is that a unit root is present in the sample, meaning it is not stationary, while the alternative hypothesis is that the time series is stationary. Table 2.3 shows the ADF statistic, as well as the 1% critical value and the $p$-value. To reject the null hypothesis we can look at both the statistic or at the $p$-value. For the first one, we need to check whether this is more negative of the 1% critical value or not. In our context, for each data series, the ADF statistic is well below the critical value, meaning that we can reject the null hypothesis with a significance level of less than 1%. For the $p$-value, on the other hand, we need to check whether this is close to zero or not; again in all our cases the $p$-values is extremely small.

| **Country** | **Data** | **ADF statistic** | **1% critical value** | **$p$-value** |
|---|---|---|---|---|
| UK | Treasury bond | -10.78 | -3.43 | 2.22e-19 |
| | Exchange rate | -39.13 | -3.43 | 0.0 |
| Japan | Treasury bond | -23.80 | -3.43 | 0.0 |
| | Exchange rate | -39.93 | -3.43 | 0.0 |
| Mexico | Treasury bond | -29.60 | -3.43 | 0.0 |
| | Exchange rate | -16.06 | -3.43 | 5.64e-29 |
| Switzerland | Treasury bond | -18.86 | -3.43 | 0.0 |
| | Exchange rate | -15.36 | -3.43 | 3.64e-28 |

Table 2.3: ADF test results

## 2.2 Preparing data

Before handling our data to the prediction models we need to divide it into training data and testing data. In our analysis we make this division in two different ways in order to study two different impacts of twitter sentiment as an extra parameter of prediction.

In the first case we take into consideration only the days spanned by the event in Table 1.1, using the first 70% of it as training data and 30% as testing data.

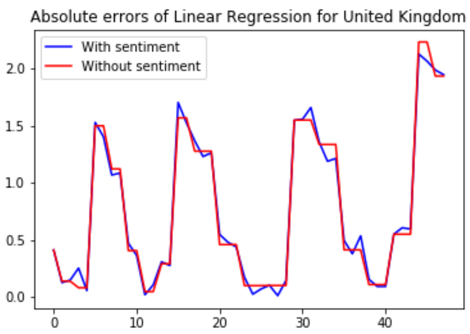
In the second case, we choose the four events which span the most days; these are: Hurricane Patricia, Irish election, Brexit, and Panama Papers. We now study how the twitter sentiment of each event enhances, or not, the accuracy of the prediction for each country. Meaning that, for each country, we will study the time period spanned by each event, where 80% of the days will represent the training set and 20% of it will represent the testing set.

# Chapter 3

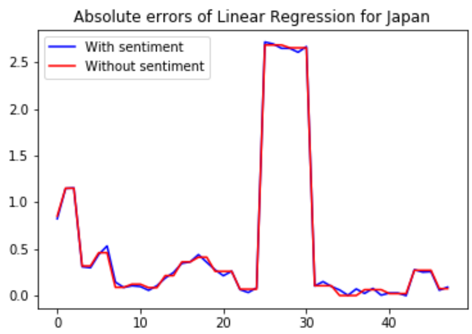
# Results

## 3.1 Analysis by country

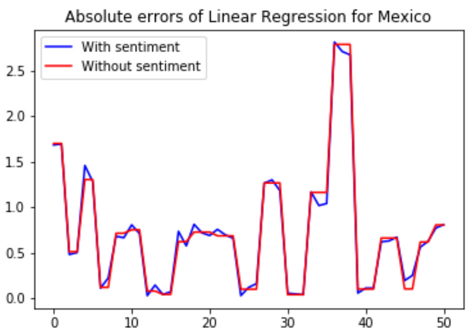
In this section we outline the results we have for the three different prediction models in the case where we take into consideration all the events in Table 1.1. We will show different graphs displaying the absolute errors for each prediction model but the graphs for the squared errors will not be explicitly displayed here since they are very similar, except for the scale. Fig. 3.1 shows the absolute errors of the linear regression, in the case we use only the exchange rate and in the case where sentiment is used as a parameter. We notice from the graphs how including twitter sentiment in the linear regression model does not help our prediction in a sensible way. In some instances, for example, this data seems even to be misleading as the prediction worsen.



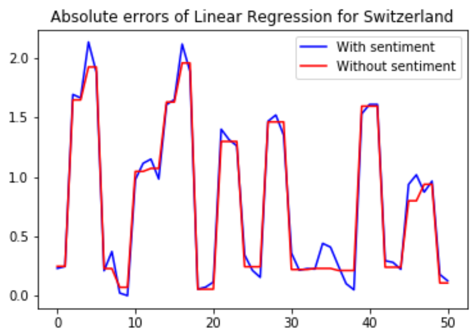

(a) UK




(b) Japan




(c) Mexico




(d) Switzerland

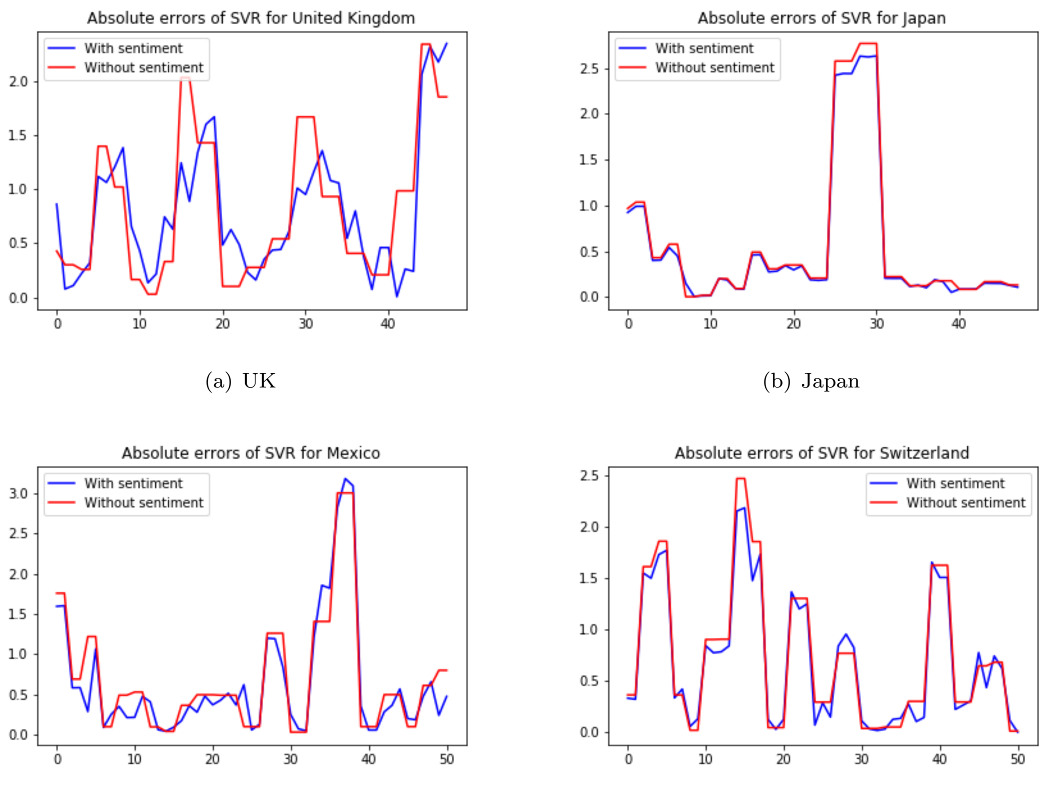
Figure 3.1: Linear regression absolute errors

This is confirmed by the values reported in Table 3.1 where the values for MAE and RMSE are listed. From here we see that, while most of the times the errors are smaller when sentiment is taken into account, this is not always true, as shown in the case of Switzerland. Furthermore, we notice that the difference between errors with and without sentiment is very small compared to the size of the errors themselves.

| **Country** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| UK | 1.011 | 1.017 | 0.769 | 0.766 |
| Japan | 0.997 | 0.999 | 0.536 | 0.538 |
| Mexico | 0.990 | 1.003 | 0.729 | 0.732 |
| Switzerland | 1.044 | 1.023 | 0.806 | 0.785 |

Table 3.1: Evaluation metrics for Linear Regression

Figure 3.2 displays the absolute errors from the SVR method. In this case, we notice how the information form the sentiment is more relevant in the prediciton of the treasury bond. Indeed, the graphs of the errors with and without sentiment are quite dissimilar, showing how the sentiment gathered from tweets is clearly impacting the prediction. In this case is more obvious how, in most cases, having sentiment as an extra parameter improves the prediction. This is further confirmed




(a) UK

(b) Japan

(c) Mexico

(d) Switzerland

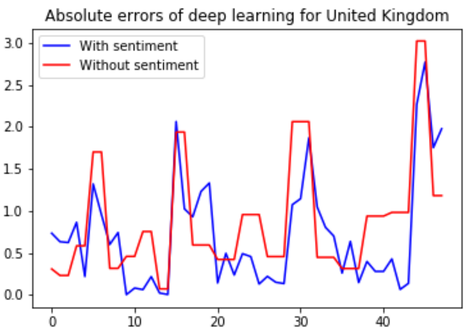
Figure 3.2: SVR absolute errors

by Table 3.2 where we notice how, besides the RMSE values for Mexico, the error is smaller when sentiment is taken into consideration. Also, the predictions using sentiment improved slightly more than in the linear regression case.

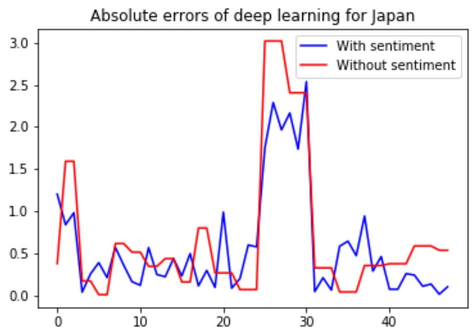
| Country | RMSE | | MAE | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| UK | 1.007 | 1.068 | 0.802 | 0.824 |
| Japan | 1.004 | 1.008 | 0.568 | 0.573 |
| Mexico | 1.047 | 1.024 | 0.686 | 0.702 |
| Switzerland | 0.993 | 1.015 | 0.730 | 0.739 |

Table 3.2: Evaluation metrics for SVR

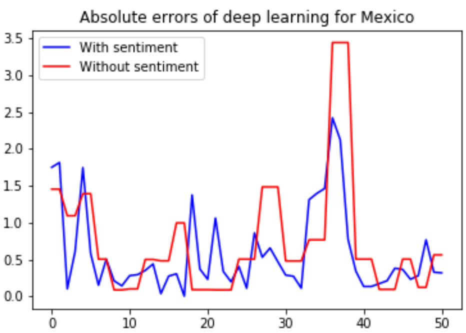
Finally, we look at the performance of the deep learning method. Figure 3.3 shows the absolute errors, in this case, for the four countries. In this graphs we see how twitter sentiment becomes a relevant parameter in the predicton. The error's graphs are very different depending on whether we use the sentiment in our independent variable or not; furthermore, we notice how the errors are smaller in the first instance. Indeed, in this case, all our prediction are enhanced when using sentiment and every error is sensibly smaller than its without-sentiment counterpart. To see the improvement more clearly we can look at Table 3.3.



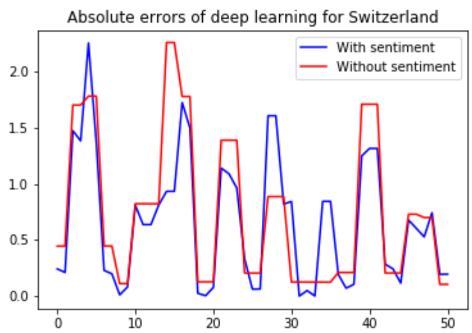

(a) UK




(b) Japan




(c) Mexico




(d) Switzerland

Figure 3.3: Deep learning absolute errors

Table 3.3 shows how the deep learning method is the best one, out of the three, to use the information given by twitter sentiment. Improving the prediction in every instances from a 12% decrease of error in the worst case (the MAE for Switzerland) up to a 25% decrease in the best case (the RMSE for Mexico).

| **Country** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| UK | 0.968 | 1.128 | 0.711 | 0.881 |
| Japan | 0.857 | 1.081 | 0.570 | 0.688 |
| Mexico | 0.825 | 1.097 | 0.588 | 0.745 |
| Switzerland | 0.873 | 1.007 | 0.660 | 0.753 |

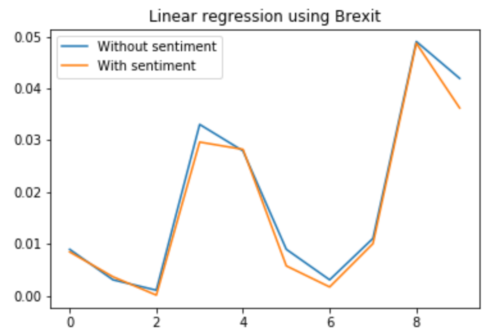
Table 3.3: Evaluation metrics for Deep Learning

## 3.2 Analysis by event

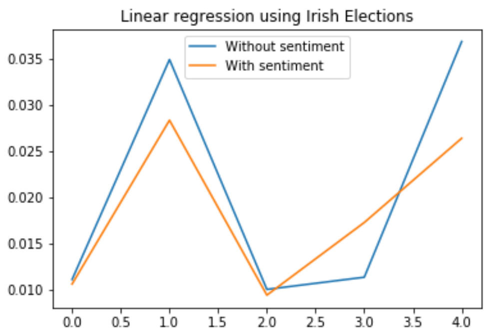
We now take into consideration each country by itself choosing four different events: Brexit, Irish elections, the hurricane Patricia and the Panama papers. The purpose of this section is to notice whether some events are more useful as predictor; meaning that we want to find those events, if they exist, that make our prediction more accurate.

### 3.2.1 United Kingdom

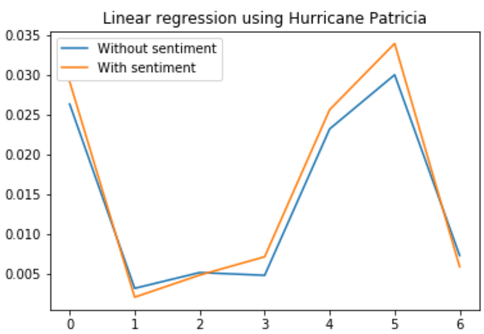
We will look at the three different prediction models separately. Figure 3.4 shows the absolute error made by the linear prediction for the four different events. As already pointed out in Section 3.1 we notice that the linear model does not really benefit from the extra input of twitter sentiment. We see that the model is not very responsive to the additional input and in some cases it even worsen due to it. It is still worth noticing that, out of the four event, *Brexit* seems to be the one which improves the prediction consistently throughout the whole testing period.



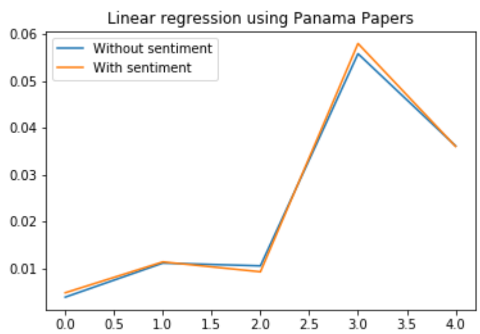

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

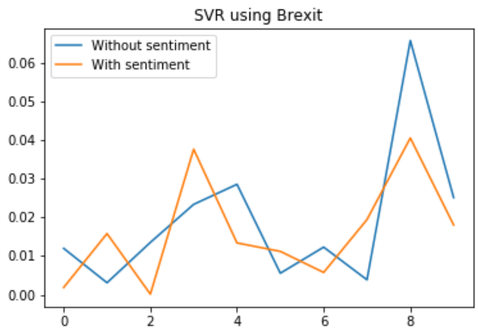
Figure 3.4: Absolute errors for Linear Regression: UK

Analysing Table 3.4 we notice how the use of twitter sentiment enhances the linear prediction only in the cases of Brexit and Irish Elections. Therefore, the table confirms what we could see from the graphs in the instance that the event is Brexit; but at the same time tells us that even in the case of Irish Elections the prediction is overall improved even if it is not improved in every testing point.

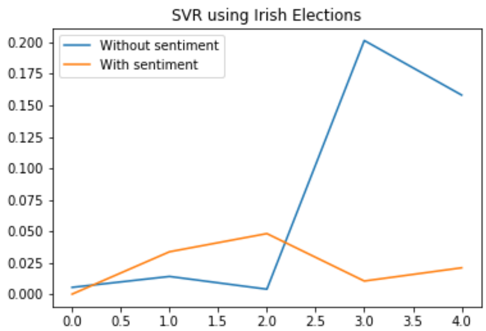
| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0236 | 0.0252 | 0.0173 | 0.0188 |
| Irish Elections | 0.0200 | 0.0242 | 0.0184 | 0.0208 |
| Hurricane Patricia | 0.0199 | 0.0179 | 0.0156 | 0.0143 |
| Panama Papers | 0.0313 | 0.0306 | 0.0239 | 0.0235 |

Table 3.4: Evaluation metrics for Linear Regression: UK

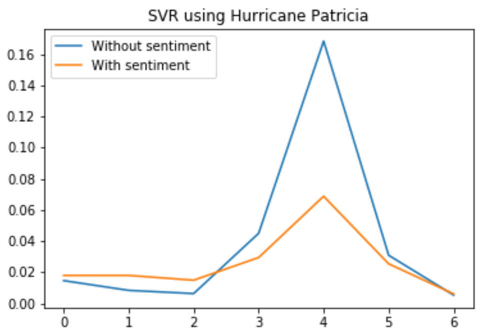
The second prediction method, SVR, once again shows to be more responsive to the new input. Figure 3.5 shows the absolute errors for the four different events. We clearly see how the model is really able to use the information contained in the twitter sentiment, changing its prediction accordingly. In this case, none of the events shows to be able to improve the prediction for every testing point; but it is clear how, on average, the absolute error is lower in the case where sentiment is used for all the events. Table 3.5 confirms the intuition we get from the graphs. Indeed, we can



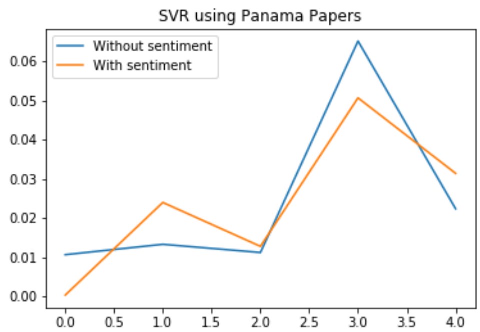

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

Figure 3.5: Absolute errors for SVR: UK

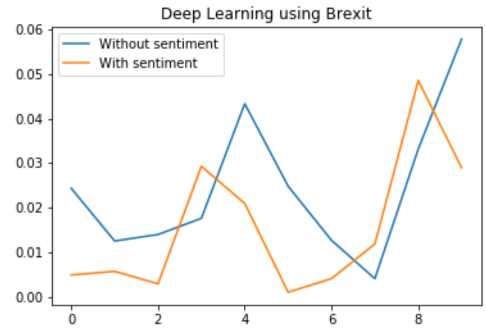
see how both the RMSE and the MAE are sensibly smaller when sentiment is taken into account, for all the events. We see the greater improvement in the case of the Irish Elections where the RMSE decreases by 75% and the MAE by 70%.

The last prediction method analysed, deep learning, shows to be very responsive to the twitter sentiment as well as SVR. In this case the graphs do not show a clear improvement of the prediction across the four different events. We can clearly see a clear refinement in the prediction in the case of Brexit, while for the other three events it is not clear, at a first sight, whether the prediction improved or not.

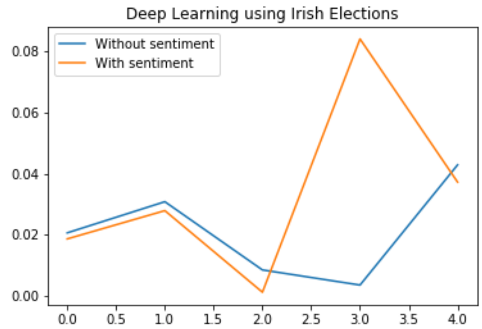
Table 3.6 gives us a clearer idea of the effect of twitter sentiment on our prediction. We can read from the table that actually the prediction is enhanced for three out of the four events. The

| Event | RMSE | | MAE | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0208 | 0.0261 | 0.0163 | 0.0193 |
| Irish Elections | 0.0283 | 0.115 | 0.0227 | 0.0766 |
| Hurricane Patricia | 0.0384 | 0.0673 | 0.0309 | 0.0399 |
| Panama Papers | 0.0293 | 0.0321 | 0.0238 | 0.0245 |

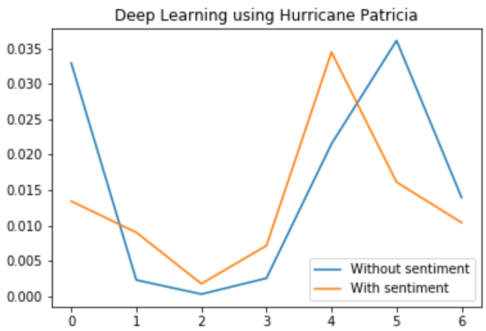
Table 3.5: Evaluation metrics for SVR: UK



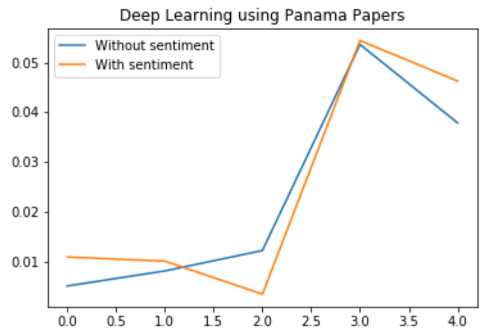

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

Figure 3.6: Absolute errors for Deep Learning: UK

only event for which using the sentiment generates a worst prediciton is the Irish Election. For this case though, we see from the graph in Figure 3.6 (b), how this is due to the presence of a single testing point which increases the two evaluation metrics dramatically.

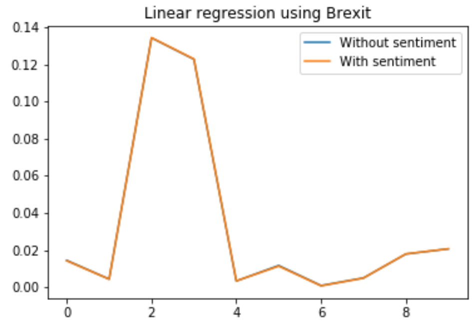
| Event | RMSE | | MAE | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0217 | 0.0289 | 0.0159 | 0.0245 |
| Irish Elections | 0.0438 | 0.0256 | 0.0338 | 0.0212 |
| Hurricane Patricia | 0.0179 | 0.0209 | 0.0145 | 0.0157 |
| Panama Papers | 0.0294 | 0.0302 | 0.0226 | 0.0234 |

Table 3.6: Evaluation metrics for Deep Learning: UK

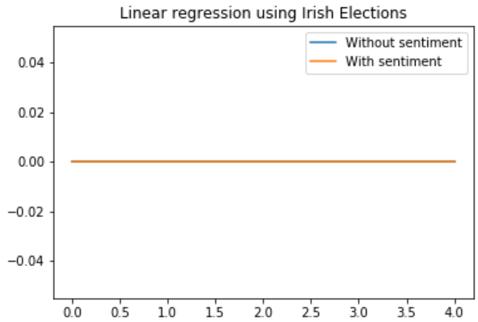
### 3.2.2 Japan

Among the countries we decided to study, Japan is the one which shows the steadies 1-month Treasury bond, resulting in long periods where it does not move at all. Consequently, since we are looking at the first difference of this data, we have timeframes where our dataset in always equal to zero. This is the case for the testing points of the Irish elections, resulting in misleading results for this specific event.

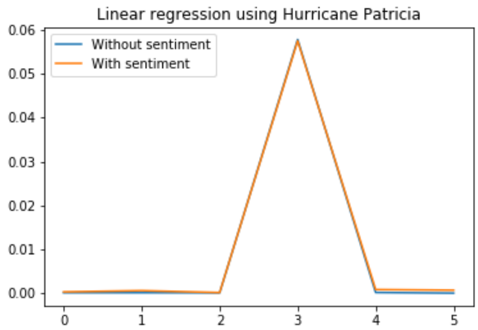
The first method of prediction used, shows to be almost completely indifferent to the presence of twitter sentiment in all events beside the Panama papers. In this case, our prediction though worsen due to the presence of the extra independent variable.



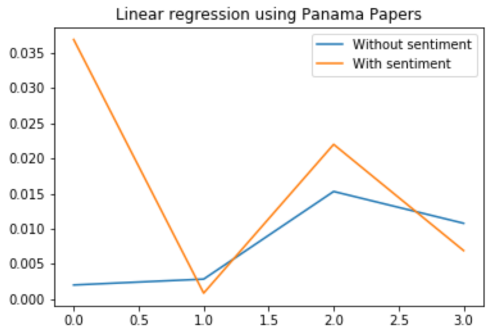

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

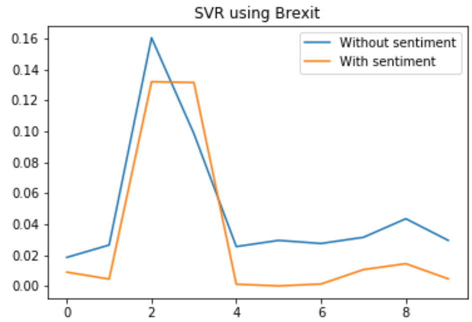
Figure 3.7: Absolute errors for Linear regression: Japan

What is already clear from the graphs, is confirmed in Table 3.7 where we see that the evaluation metrics for most events do not change sensibly when sentiment is taken into account. Again, the only exception is the event Panama paper which shows a significant increase of RMSE and MAE values when using sentiment.

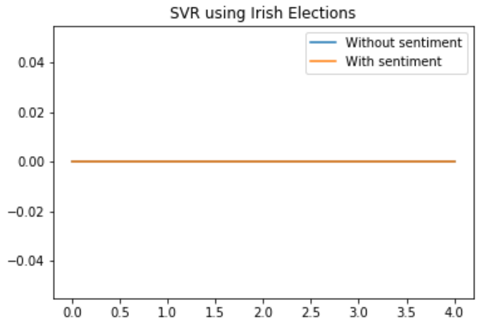
| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0585 | 0.0585 | 0.0334 | 0.0335 |
| Irish Elections | 0.0 | 0.0 | 0.0 | 0.0 |
| Hurricane Patricia | 0.0235 | 0.0236 | 0.0100 | 0.009 |
| Panama Papers | 0.0217 | 0.0093 | 0.0167 | 0.0086 |

Table 3.7: Evaluation metrics for Linear regression: Japan

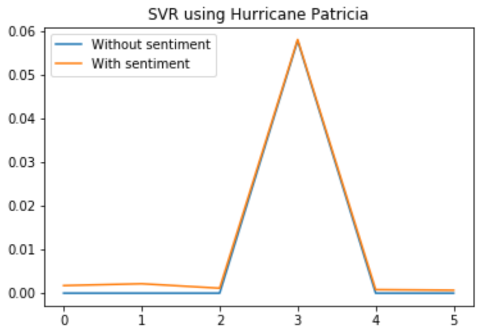
The SVR method is able to use the information from the twitter sentiment in a better way respect to the linear regression. Indeed, in this case we notice an improvement in most predictions. Besides the case of the Irish elections, where both predictions do not move from the value zero, we see how none of the other ones is able to outperform the case in which sentiment is not used, in every testing point.At the same time, it is still quite evident how, on average, the prediction improves



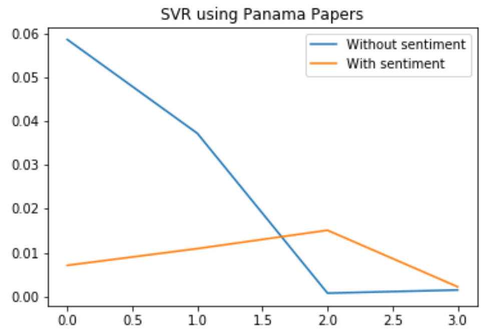

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

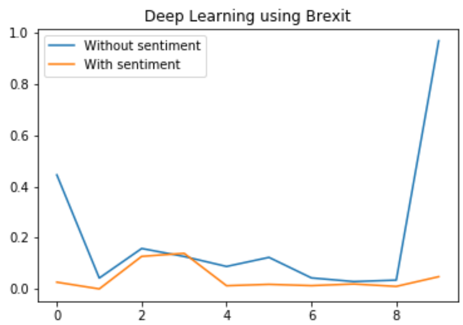
Figure 3.8: Absolute errors for SVR: Japan

Table 3.8 shows how, in the case of the Irish elections and the Panama papers, the sentiment has been a valid helper in the prediction. In the case of the Hurricane Patricia, on the other hand, the prediction is slightly worse when sentiment is used.

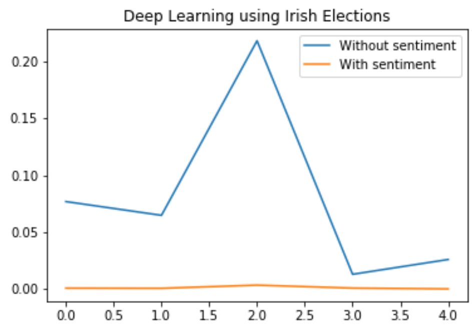
| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0932 | 0.0652 | 0.0309 | 0.0491 |
| Irish Elections | 0.0 | 0.0 | 0.0 | 0.0 |
| Hurricane Patricia | 0.0238 | 0.0237 | 0.0108 | 0.009 |
| Panama Papers | 0.0100 | 0.0347 | 0.0089 | 0.0245 |

Table 3.8: Evaluation metrics for SVR: Japan

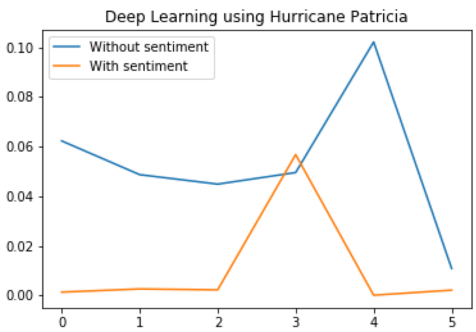
The Deep Learning approach is the one which shows the highest improvement when using twitter sentiment. Indeed, for the first three events shown in Figure 3.9 we notice an improvement in the prediction in every, or almost every, testing point; sometimes lowering the absolute error enormously. The only event which shows a worse prediction is the Panama paper one where the absolute error, when including sentiment, is significantly larger than the one obtained from the simpler model. As said earlier though, it is worth noticing that the graphs for the Irish Elections should not be regarded when judging our model since, in this case, we are actually trying to predict a constant data. Due to the high number of parameters contained in the Deep Learning method and the fact that the Japanese treasury bond was not constant in the training period of the Irish Elections event causes the model to not obtain the best training for this kind of prediction. All of this, yields a far worse prediction for the simpler model.



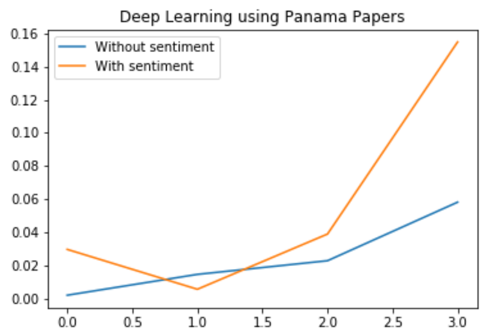

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

Figure 3.9: Absolute errors for Deep Learning: Japan

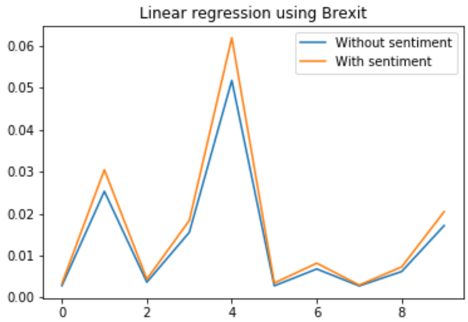
Table 3.9 is able to give us a sense of how much better the prediction is when using the sentiment. Indeed, we see that in the best case scenario, when Brexit's sentiment is used as a parameter, the root mean squared error decreases by over 80% and the absolute mean error decreases by 78%.

| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0630 | 0.3472 | 0.0416 | 0.206 |
| Irish Elections | 0.0019 | 0.1084 | 0.0015 | 0.0799 |
| Hurricane Patricia | 0.0232 | 0.0596 | 0.0108 | 0.0530 |
| Panama Papers | 0.0813 | 0.0321 | 0.0573 | 0.0244 |

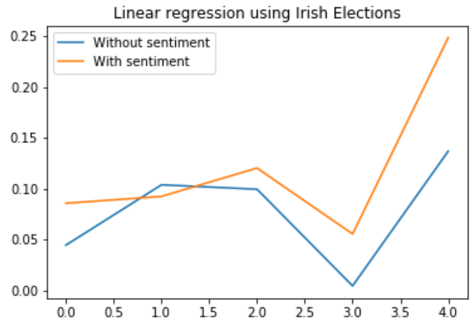
Table 3.9: Evaluation metrics for Deep Learning: Japan

### 3.2.3 Mexico

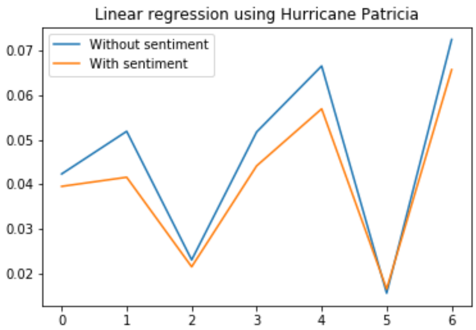
The linear regression for Mexico shows to be worsen by the presence of twitter sentiment in all cases, beside when the event is the hurricane Patricia. While it is pretty obvious by the graphs in Figure 3.10 that the average of the absolute values increases when using sentiment, it is still worth noticing how in the case of the hurricane Patricia, the prediction actually improves in almost every testing point. It is also worth noticing, that out of the four events, this is indeed the one that impacted Mexico the most.



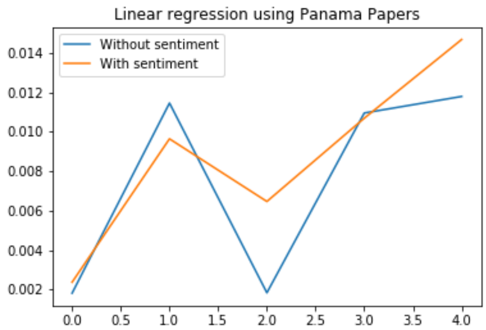

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

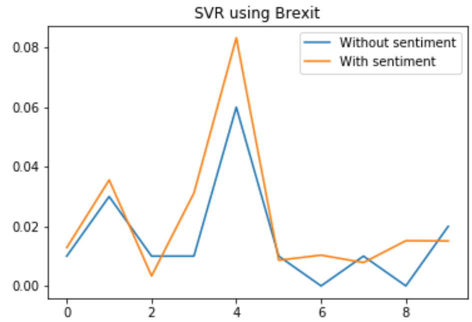
Figure 3.10: Absolute errors for Linear regression: Mexico

Table 3.10 shows clearly the worsen of the prediction when using sentiment. Also, we can see that, while it is true that the two evaluation metrics increase, this does not happen by a extremely large amount meaning the two predictions (the one with and without sentiment) are actually fairly close to each other. This holds true even for the only case where the prediction improves, indeed for hurricane Patricia the RMSE decreases by only 13 % and the MAE by 12%.

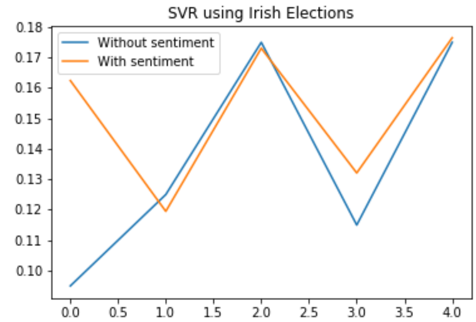
| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0239 | 0.0199 | 0.0161 | 0.0134 |
| Irish Elections | 0.0138 | 0.0910 | 0.1206 | 0.0779 |
| Hurricane Patricia | 0.0439 | 0.0502 | 0.0408 | 0.0462 |
| Panama Papers | 0.0097 | 0.0089 | 0.0088 | 0.0076 |

Table 3.10: Evaluation metrics for Linear regression: Mexico

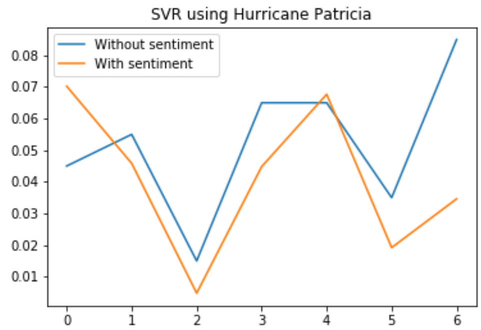
The SVR method shows (in Fig. 3.11) an actual improve in the predictions for two events: hurricane Patricia and Panama papers. For the first one we see how the prediction with sentiment is usually close to the one without sentiment outperforming it most of the times; instead, in the second case, the two predictions are significantly far from each other. For the other two events, the use of sentiment does not help the prediction leading in some cases to underperform the simpler prediction by a significant quantity.



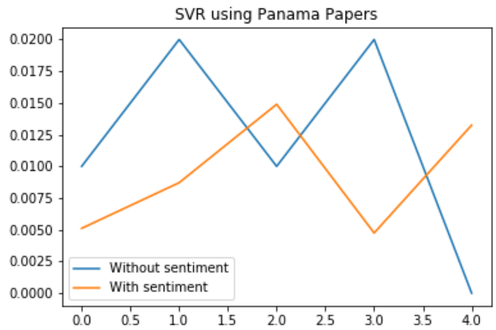

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

Figure 3.11: Absolute errors for SVR: Mexico

Table 3.11 shows how the improvement in the prediction for the hurricane Patricia and the Panama papers is very similar in terms of MAE, indeed for both events this quantity decreases by about 21%, while in terms of RMSE the prediction which uses the Panama papers shows a decrease of this quantity of 27% against a decrease of 17% for when the Hurrican Patricia is used. For the other two events, the evaluation metrics simply confirm the intuition we had looking at the graphs; indeed both these predictions had smaller RMSE and MAE when no sentiment is involved.

| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0317 | 0.0232 | 0.0223 | 0.0160 |
| Irish Elections | 0.1544 | 0.1408 | 0.1527 | 0.1370 |
| Hurricane Patricia | 0.0466 | 0.0563 | 0.0410 | 0.0521 |
| Panama Papers | 0.0102 | 0.0141 | 0.0093 | 0.0119 |

Table 3.11: Evaluation metrics for SVR: Mexico

Figure 3.12 shows the absolute errors for the deep learning technique. We notice how, in this case, the information from twitter sentiment is better used in most predictions; indeed, all different events show an improvement in the prediction in several testing points. In the case of the hurricane Patricia, we see how the prediction improves in all testing points beside just one; furthermore, the difference between the absolute values is fairly significant in almost every instance. For the other events, it is clear how the two different predictions tend to be quite far, outperforming and underperforming each other by a significant amount throughout the entirety of the testing data.


(a) Brexit

(b) Irish Elections

(c) Hurricane Patricia

(d) Panama Papers

Figure 3.12: Absolute errors for Deep Learning: Mexico

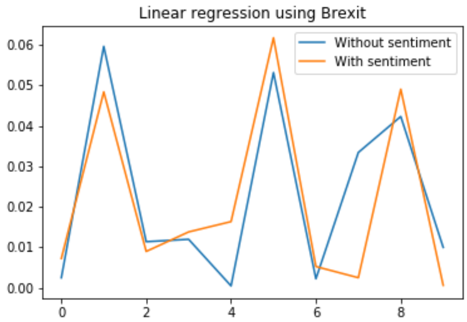
Table 3.12 shows the two evaluation metrics for the four different events. It is clear how the use of twitter sentiment proves to be beneficial in the first three events. Indeed, both the RMSE and MAE decrease in these cases. It is worth noticing that the prediction which improves the most is the one where the sentiment from hurricane Patricia is used. In fact, in this case, the RMSE decreases by 37% and the MAE by 49% showing how the prediction in this case is significantly more accurate than when we simply look at the exchange rate. The only event which is not useful in the prediction is the Panama papers in which we see a small worsening of the two evaluation metrics when sentiment is taken into account.

| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0267 | 0.0372 | 0.0208 | 0.0319 |
| Irish Elections | 0.0614 | 0.0901 | 0.0569 | 0.0841 |
| Hurricane Patricia | 0.0438 | 0.0695 | 0.0319 | 0.0628 |
| Panama Papers | 0.1638 | 0.1634 | 0.1637 | 0.1632 |

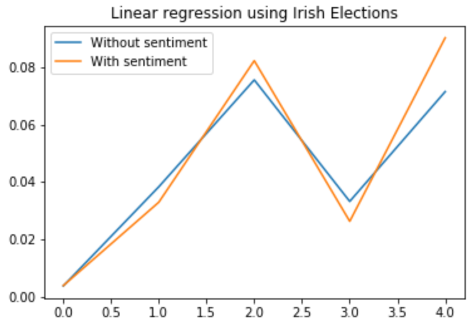
Table 3.12: Evaluation metrics for Deep Learning: Mexico

### 3.2.4 Switzerland

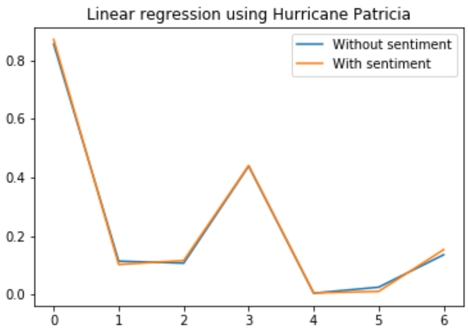
The linear regression absolute errors for Switzerland are shown in Figure 3.13. We notice how none of the events is able to enhance the prediction substantially. Indeed, for all the events, the absolute errors for the models, with and without sentiment, appear to be fairly close to each other. This makes difficult to understand, simply from the graphs, whether the use of twitter sentiment is in any way helpful.



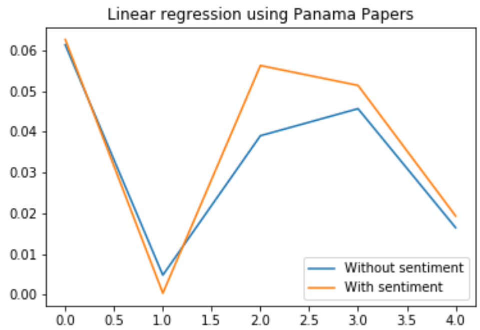

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

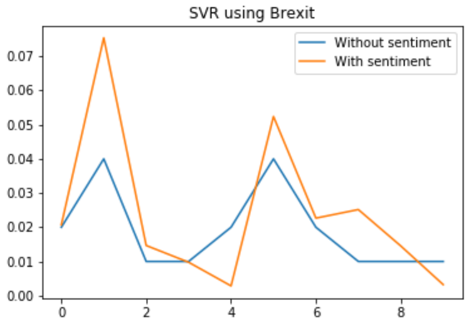
Figure 3.13: Absolute errors for Linear regression: Switzerland

Table 3.13 shows in a clearer way the performance of the different predictions. We notice how the prediction is actually improved in three out of the four cases. Nevertheless, the improvement is not significant in all three cases, indeed, while for Brexit both evaluation metrics decrease by roughly 30%, in the other two cases (hurricane Patricia and Panama papers) they decrease by only 8-9%. Furthermore, in the case of the Irish elections, the increase of the two evaluation metrics is also not very significant since the RMSE increases by 11% and the MAE by simply 5%.

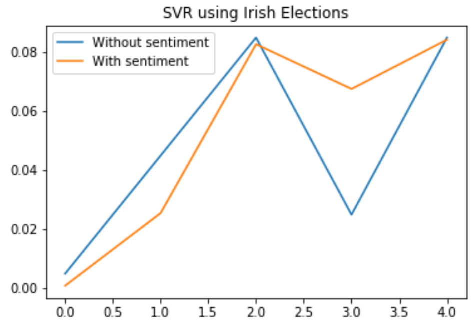
| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0212 | 0.0311 | 0.0149 | 0.0227 |
| Irish Elections | 0.0578 | 0.0518 | 0.0471 | 0.0445 |
| Hurricane Patricia | 0.3406 | 0.3723 | 0.2184 | 0.2401 |
| Panama Papers | 0.0359 | 0.0392 | 0.0304 | 0.0334 |

Table 3.13: Evaluation metrics for Linear regression: Switzerland

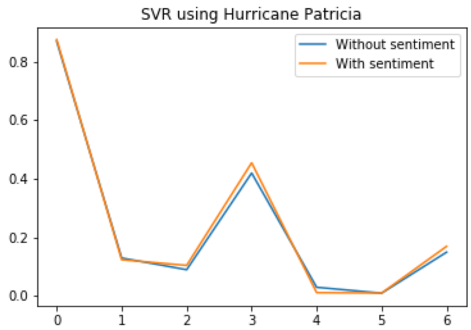
The SVR method is far more responsive to the introduction of twitter sentiment but it is not able to use it efficiently. Indeed, looking at the graphs in Figure 3.14, it is clear how all the predictions worsen when sentiment is used. Meaning that, for Japan, using sentiment in the SVR method for these four events seems to be misleading with respect to the treasury bond. Nevertheless, we notice how some events are more impactful than others; for example Brexit's sentiments is sensibly misleading while the hurricane Patricia is influences the prediction far less heavily.



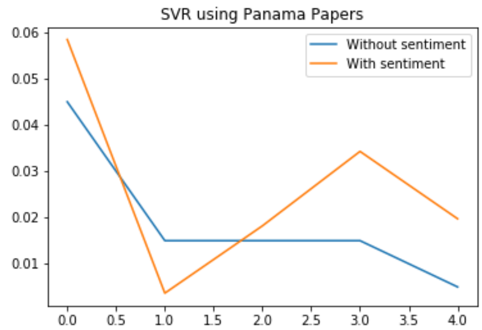

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

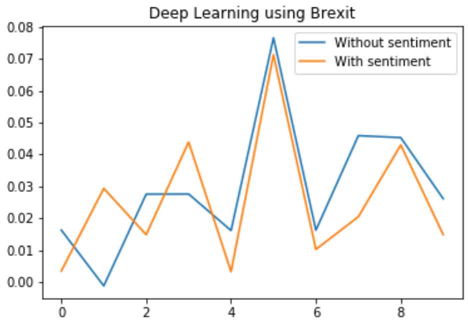
Figure 3.14: Absolute errors for SVR: Switzerland

The evaluation metrics shown in Table 3.14 confirms what we already said about the graphs. Neither the RMSE nor the MAE improve when sentiment is taken into account in any case. Indeed, the RMSE increases between 2.4% (hurricane Patricia) and 47% (Brexit), while the MAE increase between 3% (hurricane Patricia) and 40% (Panama papers). Then, it is confirmed how, in the case of the hurricane Patricia, the forecasting is not very influenced by the sentiment; but, at the same time, it is the event for which the magnitudes of the errors are the largest among the four events taken into consideration.

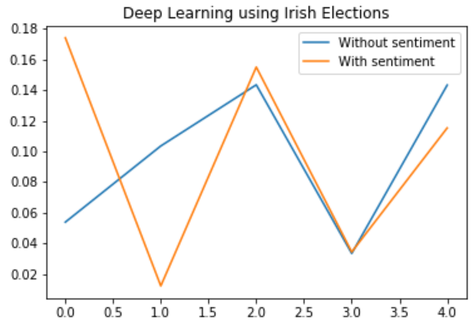
| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0325 | 0.0221 | 0.0241 | 0.0190 |
| Irish Elections | 0.0619 | 0.0585 | 0.0522 | 0.0489 |
| Hurricane Patricia | 0.3835 | 0.3745 | 0.2503 | 0.2429 |
| Panama Papers | 0.0326 | 0.0233 | 0.0267 | 0.0190 |

Table 3.14: Evaluation metrics for SVR: Switzerland

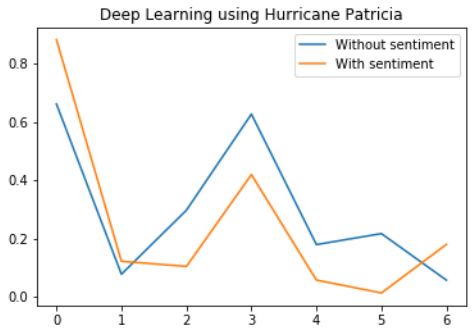
The deep learning method is the one that incorporates the information of the twitter sentiment in the best way out of the three methods. Indeed, Figure 3.15 shows clearly that, with respect to the other models, we now have far more testing points in which the prediction is enhanced when using sentiment. Again, this improvement does not seem to be significant with respect to the higher error made in other testing points, suggesting that the use of these four events for the prediction of the Swiss Treasury bond is not a sensible choice.



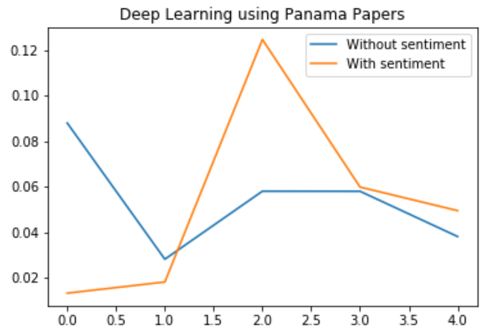

(a) Brexit




(b) Irish Elections




(c) Hurricane Patricia




(d) Panama Papers

Figure 3.15: Absolute errors for Deep Learning: Switzerland

What has been said about the graphs is confirmed by Table 3.15, here we see that the root mean squared error and the mean absolute error tend to be larger when the model uses sentiment. The only case in which the forecast is closer to the real value of the Treasury bond is when Brexit's sentiment is used. Nevertheless, even in this case, the two metrics do not improve substantially; indeed, the RMSE decreases by only 9% and the MAE by 14%.

| **Event** | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | With sentiment | Without sentiment | With sentiment | Without sentiment |
| Brexit | 0.0327 | 0.0360 | 0.0255 | 0.0297 |
| Irish Elections | 0.1174 | 0.1057 | 0.0982 | 0.0955 |
| Hurricane Patricia | 0.3816 | 0.3801 | 0.2544 | 0.3029 |
| Panama Papers | 0.0665 | 0.0578 | 0.0531 | 0.0541 |

Table 3.15: Evaluation metrics for Deep Learning: Switzerland

# Conclusion

This paper analyses the effect of twitter sentiment in three different predictions models. This analyses is split in two different approaches: the first one does not differentiate between events taking into account simply how people's sentiment, in that specific time frame, influences the prediction; the second one analyses four specific events in order to see whether the choice of events is somehow linked to the improvement of the prediction model.

The results of the first approach are shown in Section 3.1. In this context, it is noticeable how the linear regression technique is not able to take advantage of the extra information given by twitter sentiment. We see, by the absolute error's graphs, how, when using twitter sentiment, the forecast follows closely the forecast obtained using the exchange rate by itself. Furthermore, the values of the root mean squared error and the mean absolute error are significantly high. On the other hand, the SVR method is slightly better at incorporating the information contained in the twitter sentiment. We notice how the forecast obtained by the introduction of the extra parameter distance itself by the one obtained simply by the exchange rate. Furthermore, also the evaluation metrics analysed show a slight improvement. The deep learning method is the one which better uses the sentiment parameter, lowering the root mean squared error and mean absolute error for every country when using Twitter sentiment. It is clear how the forecast is then significantly improved also when looking at the absolute errors' graphs (3.3).

Adding Twitter sentiment as an extra parameter in the prediction models, then, reveals to be a sensible choice as it is able to yield better predictions. The method which is more able to use the information contained in the sentiment is deep learning.

The second approach shows how the selection of events should be taken seriously. Even if we already showed how the SVR and the deep learning method are able to use the sentiment to improve their predictions, this is very dependent on the kind of events that make up the training and test data. Section 3.2 analyses four specific events: Brexit, Irish elections, hurricane Patricia and Panama papers; showing how the forecast of the treasury bond is influenced when these events are taken into account. It is noticeable how not all events have the same effect on the prediction. It is clear how Brexit and Irish elections are able to improve the forecast, of all three models, in the case of UK but not for the other countries. Also, it can be seen how the same effect is caused by the sentiment towards hurricane Patricia when predicting the Mexican treasury bond. For the other two countries, Japan and Switzerland, none of the four events stands out as more useful than the others. This analysis lead us to believe that, when choosing events to study the sentiment of, we need to pick those events which are closely linked to the country we want to study. For example, an event like the hurricane Patricia is not closely related to Switzerland and, indeed,the prediction of the Swiss treasury bond is not significantly improved when Twitter sentiment towards this event is taken into consideration.

The analysis carried out in this project leads then to believe that using twitter sentiment as an extra parameter of prediction is useful mostly when using deep learning techniques. At the same time, it shows us how this sentiment should not be randomly chosen among all the sentences tweeted by people, but it should be cautiously selected among those tweets that refers to the specific country, or to an event closely linked to the country, we are interested in.

# Appendix A

# Code

Listing A.1: Source code for cleaning tweets

```
import re, csv
import enchant
import nltk
from nltk.corpus import wordnet
from nltk.corpus import stopwords
from nltk.metrics import edit_distance
from nltk.tokenize import word_tokenize
from nltk.tokenize.treebank import TreebankWordDetokenizer


hashtags = ['charliehebdo', 'jesuischarlie', 'charlie hebdo', 'paris',
           'germanwings', 'flughafen duesseldorf', 'a', 'u', 'absturz','nepal', 'earthquake',
           'nepalearthquake', 'refugeeswelcome', 'hurricane', 'patricia', 'hurricanepatricia', '**
                **huracanpatricia', 'parisattacks', 'bataclan', 'paris', 'ge', 'eureferendum', '**
                **brexit', 'euref', 'brussels', 'airport', 'zaventem', 'lahoreblast', 'lahore', '**
                **pakistan',
            'egyptair', 'hijacked', 'plane', 'cyprus', 'airport', 'panamapapers', 'sismoecuador', '**
                 **terremotoecuador', 'terremoto', 'ecuador']

replacement_patterns = [
        (r'won\'t', 'will not'),
        (r'can\'t', 'cannot'),
        (r'i\'m', 'i am'),
        (r'ain\'t', 'is not'),
        (r'(\w+)\'ll', '\g<1> will'),
        (r'(\w+)n\'t', '\g<1> not'),
        (r'(\w+)\'ve', '\g<1> have'),
        (r'(\w+)\'s', '\g<1> is'),
        (r'(\w+)\'re', '\g<1> are'),
        (r'(\w+)\'d', '\g<1> would'),
        (r'(\w+)\-', '\g<1>'),
        (r'(\w+)}]', '\g<1>'),
        (r'\"(\w+)', '\g<1>')

]

# The RegexpReplacer class can take any list of replacement patterns


class RegexpReplacer(object):
        """ Replaces regular expression in a text.
        >>> replacer = RegexpReplacer()
```

```
        >>> replacer.replace("can't is a contraction")
        'cannot is a contraction'
        >>> replacer.replace("I should've done that thing I didn't do")
        'I should have done that thing I did not do'
        """
        def __init__(self, patterns=replacement_patterns):
                self.patterns = [(re.compile(regex), repl) for (regex, repl) in patterns]

        def replace(self, text):
                s = text

                for (pattern, repl) in self.patterns:
                        s = re.sub(pattern, repl, s)

                return s

contraction_replacer = RegexpReplacer()


class RepeatReplacer(object):
        """ Removes repeating characters until a valid word is found.
        >>> replacer = RepeatReplacer()
        >>> replacer.replace('looooove')
        'love'
        >>> replacer.replace('oooooh')
        'ooh'
        >>> replacer.replace('goose')
        'goose'
        """
        def __init__(self):
                self.repeat_regexp = re.compile(r'(\w*)(\w)\2(\w*)')
                self.repl = r'\1\2\3'

        def replace(self, word):
        # This first two lines ensure to keep words that are correctly spelled
        # but have two repeating characters (e.g. goose is NOT gose)
                if wordnet.synsets(word):
                        return word

                repl_word = self.repeat_regexp.sub(self.repl, word)

                if repl_word != word:
                        return self.replace(repl_word)
                else:
                        return repl_word

repetition_replacer = RepeatReplacer()


from nltk.corpus import wordnet
class AntonymReplacer(object):
    def replace(self, word, pos=None):
        antonyms = []
        for syn in wordnet.synsets(word, pos=pos):
            for lemma in syn.lemmas():
                for antonym in lemma.antonyms():
                    antonyms.append(antonym.name())
        if len(antonyms) > 1:
```

```
            return antonyms[0]
        else:
            return None
# This function returns the first antonym found,
# even when there are more than one;
# while returning None if no antomys are found

    def replace_negations(self, sent):
        i, l = 0, len(sent)
        words = []
        while i < l:
            word = sent[i]
            if word == 'not' and i+1 < l:
                ant = self.replace(sent[i+1])
                if ant:
                    words.append(ant)
                    i += 2
                    continue
            words.append(word)
            i += 1
        return words

antonym_replacer = AntonymReplacer()

class SpellingReplacer(object):
    def __init__(self, dict_name='en', max_dist=2):
        self.spell_dict = enchant.Dict(dict_name)
        self.max_dist = max_dist
    def replace(self, word):
        if self.spell_dict.check(word):
            return word
        suggestions = self.spell_dict.suggest(word)
        if suggestions and edit_distance(word, suggestions[0]) <= self.max_dist:
            return suggestions[0]
        else:
            return word
spelling_replacer = SpellingReplacer()


def cleanDataset(dataset,hashtags):
    count = 0
    for i in range(len(dataset)):

        temporary_text = dataset['text'][i]
        temporary_text = temporary_text.lower()
        temporary_text = ''.join([element for element in temporary_text if element.isalpha() or element**
            ** == ' ' or element in string.punctuation])
        temporary_text = temporary_text.replace('\\',' ')
        temporary_text = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', '', temporary_text)
        temporary_text = re.sub('@[^\s]+', '', temporary_text)
        temporary_text = contraction_replacer.replace(temporary_text)
        temporary_text = repetition_replacer.replace(temporary_text)
        tokenization = word_tokenize(temporary_text)
        tokenization = antonym_replacer.replace_negations(tokenization)
        for j in range(len(tokenization)):
            tokenization[j] = spelling_replacer.replace(tokenization[j])
        temporary_text = TreebankWordDetokenizer().detokenize(tokenization)
        temporary_text = temporary_text.lower()
```

```
        if temporary_text in hashtags:
            temporary_text = np.nan
        dataset['text'][i] = temporary_text

    return dataset.dropna()
```

# Bibliography


[1] Muhammad Yasir, Sitara Afzal, Khalid Latif, Ghulam Chaudhary, Nazish Malik, Farhan Shahzad, and Oh-Young Song. An efficient deep learning based model to predict interest rate using twitter sentiment. *Sustainability*, 12:1660, 02 2020.

[2] J.D. Hamilton. Rational-expectations econometric analysis of changes in regime: An investigation of the term structure of interest rates. *Journal of Economic Dynamics and Control*, 12:385–424, 06 1988.

[3] S. Das. Mean rate shifts and alternative models of the interest rate: Theory and evidence. 1993.

[4] Suh Young Kang. *An Investigation of the Use of Feedforward Neural Networks for Forecasting.* PhD thesis, USA, 1992. UMI Order No. GAX92-01899.

[5] Arturo Estrella and Frederic S. Mishkin. The predictive power of the term structure of interest rates in europe and the united states: Implications for the european central bank. *European Economic Review*, 41:1375–1401, 07 1997.

[6] Francis X. Diebold and C. Li. Forecasting the term structure of government bond yields. *Journal of Economics*, 130:337–364, 02 2006.

[7] J. Huston McCulloch. Measuring the term structure of interest rates. *The Journal of Business*, 44(1):19–31, 1971.

[8] Charles R. Nelson and Andrew F. Siegel. Parsimonious modeling of yield curves. *The Journal of Business*, 60(4):473–489, 1987.

[9] Thomas S. Y. Ho and Sang-Bin Lee. Term structure movements and pricing interest rate contingent claims. *The Journal of Finance*, 41(5):1011–1029, 1986.

[10] Oldrich Vasicek. An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5:177–188, 11 1977.

[11] H.Kim Steven and Ju Noh Hyun. Predictability of interest rates using data mining tools: A comparative analysis of korea and the us. *Experts System with Applications*, 13:85–95, 08 1997.

[12] Kyong Jo Oh and Ingoo Han. Using change-point detection to support artificial neural networks for interest rates forecasting. *Experts System with Applications*, 19:105–115, 08 2000.

[13] Josemar A. Caetano, Helder S. Lima, Mateus F. Santos, and Humberto T. Marques-Neto. Using sentiment analysis to define twitter political users' classes and their homophily during the 2016 american presidential election. *Journal of Internet Services and Applications*, 9:18, 09 2018.

[14] Sounak Sadhukhan, Soumya Banerjee, Prasun Das, and Arun Kumar Sangaiah. *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications.* Elsevier, 2018.

[15] Zafeiris Kokkinogenis, João Filguieras, Sara Carvalho, Luís Sarmento, and Rosaldo J.F.Rossetti. *Advances in Artificial Transportation Systems and Simulation.* Elsevier, 2015.

[16] Ming Li Chew, Sahil Puri, Arsh Sood, and Adam Wearne. Blackrock applied finance project: Using natural language processing techniques for stock return predictions. Preprint available at ssrn:2940564, 2017.

[17] Nhan Cach Dang, María N. Moreno-García, and Fernando De la Prieta. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 01 2020.

[18] Adyan Marendra Ramadhani and H. Goo. Twitter sentiment analysis using deep learning methods. *2017 7th International Annual Engineering Seminar (InAES)*, pages 1–4, 2017.

[19] Fazeel Abid, Muhammad Alam, Muhammad Yasir, and Chen Li. Sentiment analysis through recurrent variants latterly on convolutional neural network of twitter. *Future Generation Computer Systems*, 95:292–308, 2019.

[20] Avinash Chandra Pandey, Dharmveer Singh Rajpoot, and Mukesh Saraswat. Twitter sentiment analysis using hybrid cuckoo search method. *Information Processing Management*, 53:764–779, 2017.

[21] A Zubiaga. Twitter event datasets (2012-2016). 10 2018.

[22] U Farooq, H Mansoor, A Nongaillard, Y Ouzrout, and M. A. Qadir. Negation handling in sentiment analysis at sentence level. *Journal of Computers*, 05 2016.

[23] G Drakos. Support vector machine vs logistic regression. 06 2019. Available at `https://gdcoder.com/support-vector-machine-vs-logistic-regression/`.

[24] S Sayad. Support vector machine - regression (svr). Available at `https://www.saedsayad.com/support_vector_machine_reg.htm`.

[25] C. Corinna and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[26] M Pakkanen. Lecture notes in deep learning, December 2019.

[27] J Nabi. Machine learning — word embedding sentiment classification using keras. Available at `https://towardsdatascience.com/machine-learning-word-embedding-sentiment-classification-using-keras-b83c28087456`.

[28] Graves A. Supervised sequence labelling with recurrent neural network. *Studies in Computational Intelligence*, 385, 02 2012.

[29] M. Phi. Illustrated guide to lstm's and gru's: A step by step explanation. Available at `https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21`.