**DEPARTMENT OF ECONOMICS**

UNIVERSITY OF STELLENBOSCH

Correlating Factors of U.S. Presidential Speeches with Stock Market Movements – a Deep Learning Approach

Appendix B

By Pablo Rees

[19119461]

Appendix B formally presented with intention of fulfilment of the requirements for completion of the MCom Economics Programme (full thesis only) at Stellenbosch University.

With supervision from Dawie van Lill

September 01 2022

# Appendix B: Machine Learning Model Specifications and Hyper-Parameters

## B.1. Regression Algorithm Hyper-Parameters

The Stochastic Gradient Descent model uses SciKit Learn’s [SGDRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html#sklearn.linear_model.SGDRegressor) class. I specified the following hyper-parameters:

* loss =' squared\_error'
* max\_iter = 10000
* shuffle = True
* random\_state = 42
* fit\_intercept = False

The Neural Network model uses SciKit Learn’s [MLPRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html) class. I specified the following hyper-parameters:

* hidden\_layer\_sizes = (L\_1, 8, 1)
* activation = 'relu'
* solver = 'sgd'
* max\_iter = 10000
* random\_state = 42

If the number of input variables is above 40 then L\_1 is set to the number of input variables divided by 8 (rounded up to the nearest whole number). If the number of input variables is smaller than 40 then L\_1 is set to 8.

The Multiple Linear Regression model uses SciKit Learn’s [LinearRegression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html) class. I specified the following hyper-parameters:

* fit\_intercept = False
* n\_jobs = -1

The Gradient Boosting model uses SciKit Learn’s [GradientBoostingRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html) class. I specified the following hyper-parameters:

* random\_state = 42
* loss =' squared\_error'
* learning\_rate = 0.1
* min\_samples\_split = 0.05
* min\_samples\_leaf = 0.02
* max\_depth = 3

## B.2. Classification Algorithm Hyper-Parameters

The Stochastic Gradient Descent model uses SciKit Learn’s [SGDClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html) class. I specified the following hyper-parameters:

* loss =' log’
* random\_state = 42
* shuffle = True
* max\_iter = 10000

The Neural Network model uses SciKit Learn’s [MLPClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html) class. I specified the following hyper-parameters:

* hidden\_layer\_sizes = (L\_1, 8, L\_3)
* activation = 'relu'
* solver = 'sgd'
* max\_iter = 1000
* random\_state = 42

If the number of input variables is above 40 then L\_1 is set to the number of input variables divided by 8 (rounded up to the nearest whole number). If the number of input variables is smaller than 40 then L\_1 is set to 8.

If Binary is True then L\_3 is set to 1 while if Binary is False then L\_3 is set to 7

The Logistic Regression model uses SciKit Learn’s [LogisticRegression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html) class. I specified the following hyper-parameters:

* solver = ‘lbfgs’
* penalty = ‘l2’
* max\_iter = 10000
* random\_state = 42

The Gradient Boosting model uses SciKit Learn’s [GradientBoostingClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html) class. I specified the following hyper-parameters:

* random\_state = 42
* loss =' squared\_error'
* learning\_rate = 0.1
* min\_samples\_split = 0.05
* min\_samples\_leaf = 0.02
* max\_depth = 5

## B.3. Datasets and variables

The full X\_control dataset contains 13 autoregressive variables from the S&P 500 daily closing price. These are:

* DlogDif – the demeaned log difference of the S&P 500 daily closing price (this should be used as a dependent variable only)
* DlogDif\_1 – the first lag of DlogDif
* DlogDif\_2 – the second lag of DlogDif
* absDlogDif – the absolute of DlogDif (this should be used as a dependent variable only)
* absDlogDif\_1 – the first lag of absDlogDif
* logDif – the log difference of the S&P 500 daily closing price (dependent variable only)
* logDif\_date\_resid – the residual of the regression of logDif on the Date variable
* logDif\_date\_resid\_1 – the first lag of logDif\_date\_resid
* blackSwan\_SD3\_1 – a dummy variable that is set to 1 if the first lag of the logDif\_date\_resid variable lies outside 3 standard deviations from its mean and 0 otherwise
* blackSwan\_SD4\_1 – the same as blackSwan\_SD3\_1 but for 4 standard deviations
* blackSwan\_SD5\_1 – the same as blackSwan\_SD3\_1 but for 5 standard deviations
* stdVol\_1DateResid – the first lag of the residual of standardized volume regressed on the Date variable
* pos\_neg\_transform – the first lag of a pseudo-dummy variable which is set to 1 if DlogDif is positive and set to the ratio of the mean of all negative DlogDifs to the mean of all positive DlogDifs

The full X\_meta dataset contains 12 adjacent financial variables that may have correlation with S&P 500 movements. These are:

* BTC\_ld\_1 – the first lag of the log difference of the daily Bitcoin closing price.
* BTC\_dr\_1 – the first lag of the residual of the regression of the Bitcoin closing price on the date.
* Nasdaq\_ld\_1 – the first lag of the log difference of the daily Nasdaq Composite Index closing price.
* Nasdaq\_dr\_1 – the first lag of the residual of the regression of the Nasdaq Composite Index closing price on the date.
* Oil\_ld\_1 – the first lag of the log difference of the daily crude oil closing price.
* Oil\_dr\_1 – the first lag of the residual of the regression of the crude oil closing price on the date.
* SSE\_ld\_1 – the first lag of the log difference of the daily Shanghai Stock Exchange Composite Index closing price.
* SSE\_dr\_1 – the first lag of the residual of the regression of the Shanghai Stock Exchange Composite Index closing price on the date.
* USDX\_ld\_1 – the first lag of the log difference of the daily US Dollar Index closing price.
* USDX\_dr\_1 – the first lag of the residual of the regression of the US Dollar Index closing price on the date.
* VIX\_ld\_1 – the first lag of the log difference of the daily Chicago Board Options Exchange's Volatility Index closing price.
* VIX\_dr\_1 – the first lag of the residual of the regression of the Chicago Board Options Exchange's Volatility Index closing price on the date.

The full X\_test dataset contains 426 NLP variables extracted from the US Presidential Speeches. 6 of these are generated by the VADER and TextBlob sentiment analysis tools. They are:

* VaderNeg – The negativity score generated using the VADER tool.
* VaderNeu – The neutrality score generated using the VADER tool.
* VaderPos – The positivity score generated using the VADER tool.
* VaderComp – The compound score generated using the VADER tool. This is a combination of the negativity, neutrality and positivity scores.
* blobPol – The polarity score generated using the TextBlob tool.
* blobSubj – The subjectivity score generated using the TextBlob tool. The higher the score the more subjective a text is.

200 of these are generated using the Word2Vec text vectorization tool. These variables do not contain specified meaning but are rather the average of final parameters from hidden layers in neural networks performing the pseudo-task of word prediction. Their purpose is to relate words to each other depending on the context of each word within the broader corpus. These are annotated as:

* WV\_0
* WV\_1
* …
* WV\_198
* WV\_199

The final 220 variables are generated using the Doc2Vec text vectorization tool. The Doc2Vec tool works similarly to the Word2Vec tool except that during training a second hidden layer is incorporated. This second layer is persisted across each word in a text in the training corpus and its hyper parameters adjusted so as to create a single vector describing the text as a whole. This tool was used to create 2 sets of variables (each one a hidden layer from a pseudo-task). These are a 200 variable set, annotated as:

* DV\_200\_0
* DV\_200\_1
* …
* DV\_200\_198
* DV\_200\_199

and a 20 variable set, annotated as:

* DV\_20\_0
* DV\_20\_1
* …
* DV\_20\_18
* DV\_20\_19