January 2016

Can You Really Predict Markets With Twitter?

Christopher Plenzick

CAN YOU *REALLY* PREDICT MARKETS WITH TWITTER?

by

Christopher Plenzick

Bachelor of Business Administration, University of Georgia, 2010

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Applied Economics

Grand Forks, North Dakota

December

2016

This thesis, submitted by Christopher Plenzick in partial fulfillment of the requirements for the Degree of Master of Science Applied Economics from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

Xiao Wang

Cullen Geonner

Prodosh Simlai

This thesis is being submitted by the appointed advisory committee as having met all of the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

Wayne Swisher

Dean of the School of Graduate Studies

Date

PERMISSION

Title	Can You *Really* Predict Markets with Twitter?
Department	Applied Economics
Degree	Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in her absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

Christopher Plenzick

November 15, 2016

Contents

List of Figures

List of Tables

Abstract

In this paper, I attempt to apply an emotional proxy derived by applying the Affective Norms for English Words (ANEW) to messages posted to the Twitter social networking service in order to forecast the movement two stock market indices: the Dow Jones Industrial Average (DJIA) and the CBOE Volatility Index (VIX). In contrast to previous works, I have compared the results of various forecast models employing different sentiment variables, as well as comparing the neural network approach to more standard logistic regression. Additionally, several of the models used employ an as-yet unique sentiment proxy, focusing on the average of expressed emotion rather than the volume of expressed emotion. The results indicate that while there is a distinct possibility that sentiment variables can assist in accurately forecasting market movement, the differences in choice of sentiment proxy and forecast method are less important than anticipated.

1 Introduction

Being able to predict the future is a fascinating possibility, even when the predictions are restricted to a single event. Recent works such as Bollen et al. (2010) and Zhang et al. (2011) claim to have such a power, by way of using online microblogs from Twitter as a sentiment proxy to predict the daily changes in the stock market and related indices. In this paper, I attempt to use a similar method to predict the direction of stock market movements and compare my results with the results of previous works. Due both to limitations in my own capabilities as a single researcher with limited computing ability, and some "black box"-ish methods used by previous work, the methods used herein are modified quite a bit, with my own focus placed on streamlining the process to make it relatively simple and fast to reproduce and modify. Given the binary nature of movements (either up or down) I selected two methods that excel in categorization tasks: logistic regression and neural networks. My findings using these methods suggest that, overall, the sentiment proxies perform better than random guessing but slightly less well than expected given the optimism of existing research. However, the differences between the various sentiment proxies are minimal.

1.1 Background

The role of the emotion in economic choice has, within the neoclassical context, gotten little attention from economists, being folded into the general concept of utility. Beginning with papers such as Elster (1998) and Loewenstein (2000), the role of emotion in economic decision theory has recently come under greater scrutiny. Both authors propose theoretical frameworks that move beyond the utility model of emotion. These new frameworks describe emotional states as integral to the decision-making process. Certain, more visceral states act as a survival mechanism, shifting the perception of

both the choices available and their outcomes. Additionally, emotions also exist on the other side of the decision-making equation, as the outcomes of a choice can lead to the individual obtaining a different emotional state. However, since we can't use future data as a predictor, the focus remains on the current and past emotional states and their effects on decision-making.

So, knowing that sentiment has a strong effect on decision-making by the individuals that make up markets, it stands to reason that we should find some correlation between measures of sentiment and some measurable market. The most easily measurable market is, of course, that for equities such as stocks (and also likely the easiest market to profit from foreknowledge of). Much research has been done to determine if, as Elster and Lowenstein predict, the sentiments of individuals does indeed affect their decision making and thus correlate with stock market activity.

In Neal and Wheatley (1998), they discovered that 2 of the suggested measures of investor sentiment did predict returns, but that others added no additional predictability. Hirschleifer and Shumway (2003) finds that a measure of general sentiment, the weather, is highly correlated to the daily returns of stock markets for 26 countries. The difference in the results between these papers, the first finding only mixed results using strictly investors' sentiment, while the second having a very strong conclusion using a more public sentiment, suggests that a more general measure of overall public sentiment may be more useful than one that attempts to directly capture only investors.

With the advent of Internet media, a new potential to examine the voices and sentiments of both investors and the general public became widely available. Antweiler and Frank (2004) examined the contents of stock message boards. They found that the messages posted to these boards have predictive value for returns, volume, and volatility. Again, because message posting is not limited to those who actually do invest, or even those who are located near a trading floor, this sample captures a

wider range of the public than just institutional investors. So the trend appears to exist that more general measures of public sentiment capture more of the predictive value for stocks.

As social networking sites have grown into the omnipresent entity that they now represent in the lives of many, a new breed of sentiment data has become available. With so many people freely sharing their intimate thoughts and feelings, this gives researchers a natural opportunity to compare public sentiment with world events. In particular, the Twitter social network has been extremely attractive to researchers. Twitter offers easy-to-use direct access to streams of publicly shared messages, and the length of each message is limited to 140 characters, making it very easy to compile and analyze a large number of messages. The initial volley of research, such as Bollen et al. (2010) and Zhang et al. (2011), found a striking correlation between Twitter sentiment variables and the movements of stock indices. Bollen et al. (2010) find that the calm emotion, as determined by their proprietary GPOMS algorithm, is the best predictor of future market values, while Zhang et al. (2011) find that both "hope" and "fear" as keywords in tweets predict negative movements in the market. A third paper, Ranco et al. (2015), finds that the emotional content of messages containing a certain stock symbol can predict the movement of those stocks. In Mittal and Goel (2012), the authors attempt to replicate the work by Bollen et al. (2010) and arrive at a similar predictive value, but this time find an additional mood profile (happy) that contributes significant predictive power. With so many positive results, it seems a natural conclusion that Twitter is indeed a perfect candidate for prediction of future events by sentiment analysis. Indeed, Mao et al. (2011) finds that Twitter beats out several other Internet-based sentiment measurements in forecasting accuracy. Given these overwhelming results, Twitter appears to be a viable and natural choice for a sentiment data source to use in my own analysis.

In a sense, the results of these studies are indication that the random walk hypoth-

esis, and by extension the efficient market theory, may be less correct than previously thought. This hypothesis states that the current price of assets in a market is inclusive of all available information, so there should be no way to use a piece of information, assuming that information is available to anyone, to predict the price. White (1988) explores this concept by constructing a neural network designed to to forecast the end-of-day prices for IBM stock, using the past prices as input. While the results of the autoregression neural network on the training data were promising, out-of-sample forecasts were inaccurate, and so the efficient market hypothesis remained unchallenged by this result. If the efficient market hypothesis were correct, it would mean that the information regarding the emotions of people, both investors and the general public, which can be observed by anyone who is so inclined, should already be incorporated into and reflected in the existing asset prices. However, the conclusions of the works referenced above, which, excepting White (1988), all find predictive ability in their various inputs, represent a mounting pile of evidence against efficient markets. As my own results fall more in line with White, it appears that there is much more work to be done in this area before a solid conclusion can be made from so many conflicting results.

2 Data

2.1 ANEW Wordlist

The development of the Affective Norms for English Words [ANEW] wordlist is described in Bradley and Lang (1999). The dimensional model of emotion used in this work describes emotions as points in 3-d space, defined by dimensions of Valence, Arousal, and Dominance. Valence, or Pleasure, and Arousal are considered primary dimensions, respectively corresponding to the psychological and physiological aspects of a particular emotion. Dominance is a more recent addition to the affect model,

corresponding with whether a particular emotion is associated with being in or out of control. Within this particular wordlist, the values for each dimension range from 1 to 9. The scale is arranged such that greater numbers in valence, arousal, and dominance indicate a greater feeling of pleasure, excitedness, or control respectively. Given these values, any emotional state can be plotted and quantitatively compared to others within the same model.

The affect model used in the ANEW study is presented in Russell and Mehrabian (1977). The primary factors of pleasure-displeasure, degree of arousal, and dominance-submissiveness had previously been shown to be sufficient to define nonverbal emotional expression. Russell shows that not only can the three-dimensional space also be used to define verbal expression of emotion, but that these dimensions, particularly the addition of the dominance-submissiveness dimension, are necessary to fully define emotional state as expressed in English. Also important to note is that in these psychological studies, emotional state is not defined by fleeting passions. It is meant to represent the overall state of mind, like a mental backdrop that influences an individual's thoughts, decisions, and reactions. This influence should extend to economic activity as well, and so it would seem a reasonable assumption that the emotional state of the agents within an economy should have at least some correlation to economic indicators such as stock indices. The question is whether or not this correlation is directly observable and, if so, if it is a leading indicator for asset pricing.

In order to gather the emotional ratings for each word, the authors employ the Self-Assessment Manikin devised by Lang. Participants used the graphical representation of the varying degrees of each emotional dimension (i.e. a smiling face for high Valence, or a figure that appears to be asleep for low Arousal) that they felt in response to each word. The resulting mean and standard deviation of responses for each word and dimension were reported for both men and women separately, and for

the study participants as a whole. Here, because it is impossible to determine the gender of all Twitter users, I have used the results for all participants.

The decision to use this particular wordlist came after consideration of previous works, which typically use dimensions that consist of single emotions based mainly on the frequency of those emotions being stated outright or within certain, fixed, phrases. I questioned whether using a more generalized framework such as the three-dimensional PAD model, paired with a list such as ANEW that extracts the inherent, non-explicitly-stated emotion conveyed by a choice of words, might provide a more accurate picture of the prevailing emotional state of social media users. ANEW is also freely available both in the original paper and by request from the authors, allowing for easy replication and extension of research done using it. In this analysis the values published in the original article are used.

It is worth noting that following in the footsteps of ANEW are a host of other wordlists, among them an expanded version of ANEW containing over ten-thousand words Warriner et al. (2013) and one developed solely for the purposes of measuring Valence of social media status updates Nielsen (2011). Both of these wordlists, and potentially more that remain unknown to me, are worthy of consideration and study, but due to their size and the finite nature of time I must leave that duty to researchers who have considerably greater computational resources available to them.

2.2 Acquiring the Twitter Dataset

The Twitter data was acquired directly through the streaming API provided on Twitter's developer site [https://dev.twitter.com]. The "sample" stream provides a random selection amounting to about 1% of all tweets sent. However, this stream provides a volume of tweets that numbers in the millions each day, a number that far surpasses both the needs and computing capability of this study. To cut back the stream to a more appropriate number, a further randomized selection of about 5% is taken at the

time of downloading. The resulting selection amounted to around ninety-thousand tweets per day. For each selected tweet the script extracted only the relevant data: the Date and Time the message was sent, the text of the message, the number of followers the sender has, and an identification number unique to each tweet. All other information, including user names, locations, etc. are discarded. The records kept are compiled into a daily tweet dataset to be scored by sentiment analysis.

2.3 Twitter Sentiment Analysis: Creating the time-series

To begin analyzing the textual content of the tweets, a simple filter is applied which removes all punctuation and converts the remaining text to lower-case. The scoring script then cycles through each converted tweet and searches for words that are contained within the selected sentiment dictionary, in this case the 1034 words of the ANEW list. Any matching words and their respective sentiment scores for each of the three dimensions are then saved in a separate variable with the corresponding tweets. Tweets that don't contain any matching words are discarded.

The sentiment scores for each tweet were calculated using the method outlined by Healey and Ramaswamy (2011). To be specific, the scores of the ANEW words found in each Tweet are used to form a weighted average score that is inversely proportional to the relative standard deviation of their rating. The formula for a single dimension of this weighted average is:

$$Score_t = \frac{1}{n-1} \sum_{1}^{n} \bar{x}_i * (1 - \frac{\sigma_i}{S})$$

Where n is the number of ANEW words found, $\bar{x}$ is the mean value of the dimension, σ is the standard deviation of the value, and S is the sum of all standard deviations of contained words.

Note that in order to use this function, the number of words contained in the

message must be at least two. For messages containing only a single ANEW word, the scores for each dimension are equal to the scores of the single contained word.

To form the daily sentiment time-series, the values of the sentiment score for each matching tweet within each 24-hour period of the GMT time zone are averaged. A total of 185 days of tweet values were collected, to form a final time series of 180 days with corresponding lagged values.

A second time series is extracted using the same method, but this time using only tweets that contain two or more matching words, which in theory should improve the accuracy of the sentiment evaluation function.

2.4 Additional Data

As an additional sentiment proxy, the daily cloud cover in Manhattan was collected from [http://forecast.io]. This gives the percentage of cloud cover in the Manhattan area on that day, which I subtracted from one to arrive at an estimate of the daily amount of sunshine at the New York Stock Exchange, in attempt to emulate the sunshine effect described in Hirschleifer and Shumway (2003).

Daily closing data for the DJIA and VIX were collected from FRED. Because this research is concerned only with the movements of these indices, a categorical variable was constructed by taking the first difference, and applying a value of 1 for an increase over the previous day, and 0 for a decrease. Narrowing the focus in this way transforms a regression problem into a classification problem, making it much clearer when comparing between different model predictions which is performing accurately and which isn't.

Before analysis, all sentiment score data is normalized to the range of $0, 1$. This normalization is necessary for the neural network to function as intended.

3 Analysis

3.1 Models

To test the predictive effectiveness of the acquired data, I constructed three different prediction models for each index. In each case, a present value and 3 one-day lags of the chosen predictors are used. Additionally, the models are repeated using only lagged values to determine if the movement can be predicted rather than just correlated.

The first model uses the percentage of sun, defined as $1 - \%cloudcover$, mimicking the sentiment variable used in Hirschleifer and Shumway (2003).

$$Y = \beta_0 + \beta_1 Sun_{(t,t-1,t-2,t-3)} + \epsilon \tag{1}$$

And the corresponding model using only lags:

$$Y = \beta_0 + \beta_1 Sun_{(t-1,t-2,t-3)} + \epsilon \tag{2}$$

A second set includes the full array of Tweet sentiments:

$$Y = \beta_0 + \beta_1 Sentiment_{(t,t-1,t-2,t-3)} + \epsilon \tag{3}$$

$$Y = \beta_0 + \beta_1 Sentiment_{(t-1,t-2,t-3)} + \epsilon \tag{4}$$

And finally a third model set that adjusts the Tweet sentiment to include only the messages with two or more ANEW words:

$$Y = \beta_0 + \beta_1 Sentiment^{(Words \geq 2)}_{(t,t-1,t-2,t-3)} + \epsilon \tag{5}$$

$$Y = \beta_0 + \beta_1 Sentiment^{(Words \geq 2)}_{(t-1,t-2,t-3)} + \epsilon \tag{6}$$

For each of these models, I use a training set comprised of the initial 170 days of the dataset to fit the models, which are then used to predict the daily change in the index for the remaining 15 days and compared to the real data to determine accuracy.

3.2 Neural Network Structure

A multi-layer feed-forward neural network, as described by Hyndman and Athanasopoulos (2016), is one of the primary models used in this prediction of stock index movements. Using the guideline that the hidden layers should be smaller in size than inputs but greater than the number of outputs, I am using a network with input nodes equal in number to the regressors, a single hidden layers of 5 nodes, and a single output node. Learning is done through backpropagation, and a logistic activation function is used.

Fadlalla and Lin (2001) summarizes the findings of several papers in which both standard methods and neural networks are used, and find that, on average, neural networks outperform the standard statistical methods such as regression in several financial applications, including bond rating, bankruptcy prediction, and asset value forecasting. Given the differing conclusions reached, it would seem that while there is a place for neural networks in economic research, the effectiveness of such methods seem to vary by application. One major cause of the variation in results is the flexibility that the neural network model offers in terms of topology. Many decisions must be made about the size and number of hidden layers, the learning rules applied, activation functions and so on. More problematic is that there seems to be no defined rules to how best construct the network, making the discovery of an appropriate topology largely a matter of trial-and-error.

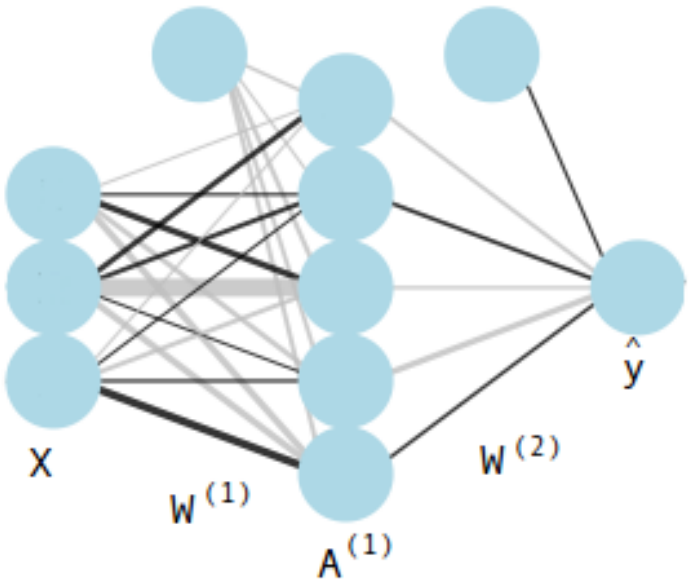


Figure 1: Neural Network Structure

Input nodes represent the regressors, while the output node is the predicted value. Each node is connected to each node of the following layer by a particular weight, which is found through an algorithm called back-propagation. The weights in the diagram above are visualized by the thickness of the line connecting each node, with thicker lines representing a larger weight.

To arrive at an output value, the inputs are multiplied by the weights that connect them to a particular node and summed, then multiplied by the node's activation function (a sigmoid logistic) to determine a value for that node. Then, the process is repeated with these nodes becoming the inputs for the next layer. In each layer except the output layer, a bias node is also present. This node always holds a value of 1, and functions similarly to the constant term of a regression.

For the network used here, represented by the diagram above, the values of each weight and node can be computed as follows: The value of the nodes in the input layer are just the values of the input, in this case the training data, represented by

the matrix X. The input nodes pass a value to the hidden layer that is equal to the values of X are multiplied by the matrix of weights connecting them to the nodes of the hidden layer, or generally: $z_2 = XW_1$. The values passed to the nodes are then transformed according to the activation function, and this value is assigned to the node: $a_2 = f(z_2)$. Then, values for the remaining layers of nodes are calculated as:

$$z_3 = a_2 W_2$$

$$\hat{y} = f(z_3)$$

In contrast, if we consider the logistic regression, which is also employed herein, we can imagine it as a neural network consisting only of input and output layers:

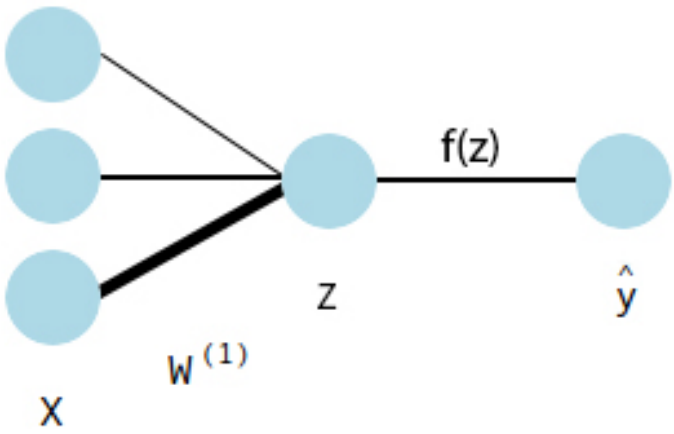


Figure 2: Logistic Regression imagined as a neural network

With this visual it becomes easy to see how we arrive at the final value. The inputs are multiplied by their weights (coefficients), summed to get the value of Z, and passed through an activation function (in this case, the logistic function) to arrive

at a final output value.

3.3 Back-Propagation

The algorithm for determining the weights between nodes, back-propagation, is in many ways similar to the way in which a simple linear regression is calculated. The commonly used linear regression is essentially the same as a neural network with only input and output nodes. The addition of hidden layers allows for more accurate prediction of complex non-linear patterns, such as those observed in stock market movements.

The general idea is to use a gradient descent method to minimize the error produced at the output. To begin, we start by initializing the weights, or coefficients, to random values. Then, the weights are shifted slightly higher or lower in order to decrease the total error, or cost. To make this movement possible, the partial derivatives of the cost function with respect to each weight must be found.

A cost function $J = \sum \frac{1}{2}(y - \hat{y})^2$ for a single hidden layer network such as the one used here would look like:

$$J = \sum \frac{1}{2}(y - f(f(XW_1)W_2))^2 \tag{7}$$

Where y are the actual output values, function f is the activation function of the neural network, X is the matrix of inputs, and W_i is the matrix of weights for the ith layer. In most cases, the activation function is either a sigmoid (when outputs are between 0 and 1) or hyperbolic tangent function (for outputs between -1 and 1). So, the partial derivative of the cost function for the set of weights connecting the hidden layer to the output would be:

$$\frac{\partial J}{\partial W_2} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W_2} \tag{8}$$

or:

$$\frac{\partial J}{\partial W_2} = a_2^T \Delta_3 | \Delta_3 = -(y - \hat{y}) f'(z_3) \tag{9}$$

Then, the partials for the first layer of weights, which connect the input layer to the hidden layer, can be represented by:

$$\frac{\partial J}{\partial W_1} = X^T \Delta_2 | \Delta_2 = \Delta_3 W_2^T f'(z_2) \tag{10}$$

In this way the errors of the second set of weights are taken into account when finding the gradient of the first layer, since changes to the first layer will also affect the error caused by the second. This method of calculating the gradients backwards from the output gives the algorithm its name.

To update the weights, a value equal to the partial of the cost function times the learning rate, α, is subtracted from each weight and the resulting value assigned as the new weight.

$$W_{i,j} \leftarrow W_{i,j} - \alpha * \frac{\partial J}{\partial W_{i,j}} \tag{11}$$

This update occurs simultaneously for all weights in the network. Then, the same algorithm is repeated until the gradient of the cost converges to some acceptably low level, ideally something close to zero, or after a specified number of iterations have been performed without finding any convergence.

3.4 Back-propagation Caveats

The primary issue with the use of a neural network, especially in contrast to simpler functions like logistic regression, is that the cost function of a logistic regression is always convex, while that of a neural network rarely is. The added layers of the neural network bring with them local minima that badly complicate the minimization

of error. To limit the chance that the algorithm would converge to the incorrect minimum, it can be run multiple times with different initial values, and either the best model can be chosen, or an average of all models can be created. Here, being heavily limited by available processing power, I have chosen to average the results of 25 fitting procedures for each model.

Also worth noting is the potential for over-fitting, which is alleviated by adding a regularization term, λW_i to each partial derivative of the cost function. This discourages having an overly fitted model by penalizing large weights more than smaller ones.

The implementation of the backpropagation algorithm used in this research is provided by the *nnet* and *neuralnet* packages in R. The networks' output are then used to predict values for the test set comprised of the remaining 15 days, and compared to the actual values for the change and movement direction of the indices.

4 Results

After the models are trained on the first 170 days of the data, the resulting coefficients were applied to the independent variables for the remaining 15 days to arrive at a predicted value. To translate the model predictions to the binary result, the outputs are rounded to 0 if they are less than 0.5, or 1 if they are equal or greater than 0.5. These final results are compared to the actual market movement for the corresponding day.

The results of the prediction for each model, predicted by both logistic regression and neural network, can be seen in the charts below. Particularly notable is the accuracy of the logistic models for the DJIA. The most accurate of these models predicted the correct direction of movement 11 out of 15 days in this test set. It also appears that the choice of the sentiment proxy may be less important than expected.

While there were differences in the number of correct predictions, they are not striking enough to conclusively say one is more effective than the other, especially with only one trial.

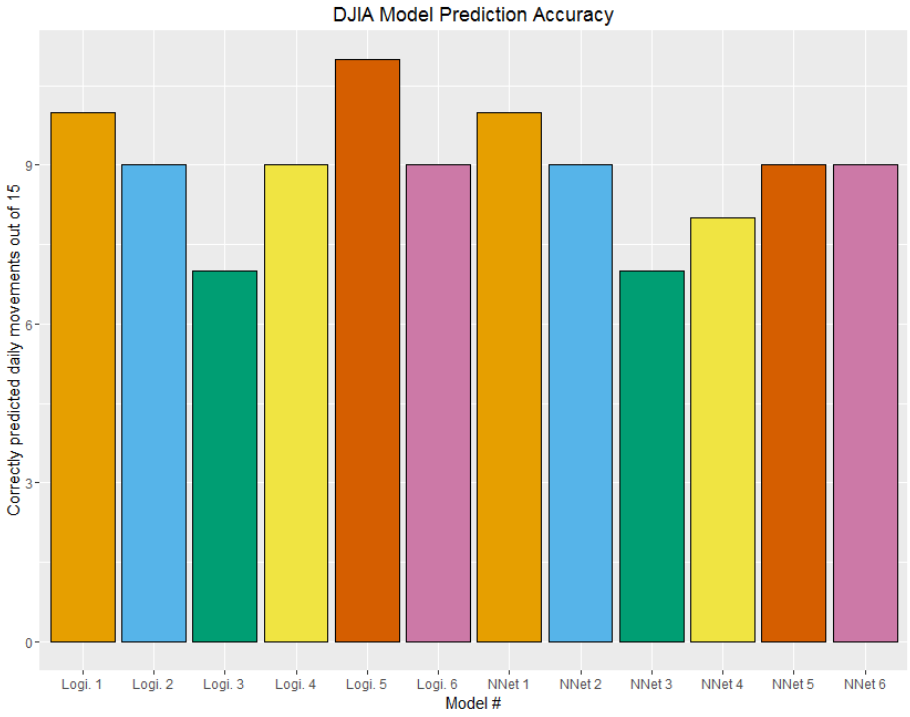


Figure 3: Prediction accuracy for DJIA forecast models

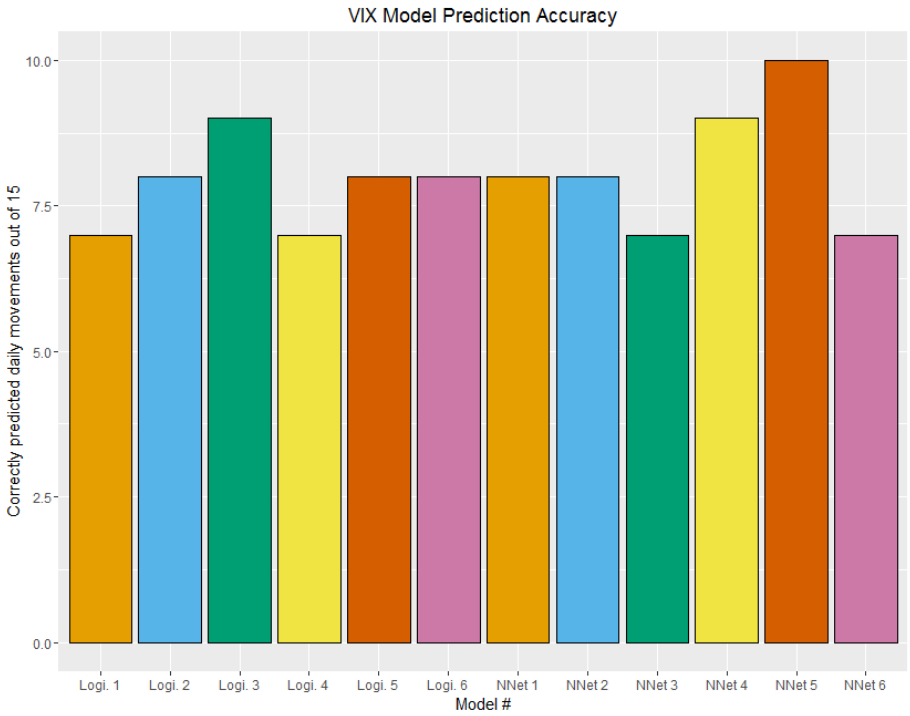


Figure 4: Prediction accuracy for VIX forecast models

Table 1: Accuracy of DJIA prediction by model and method

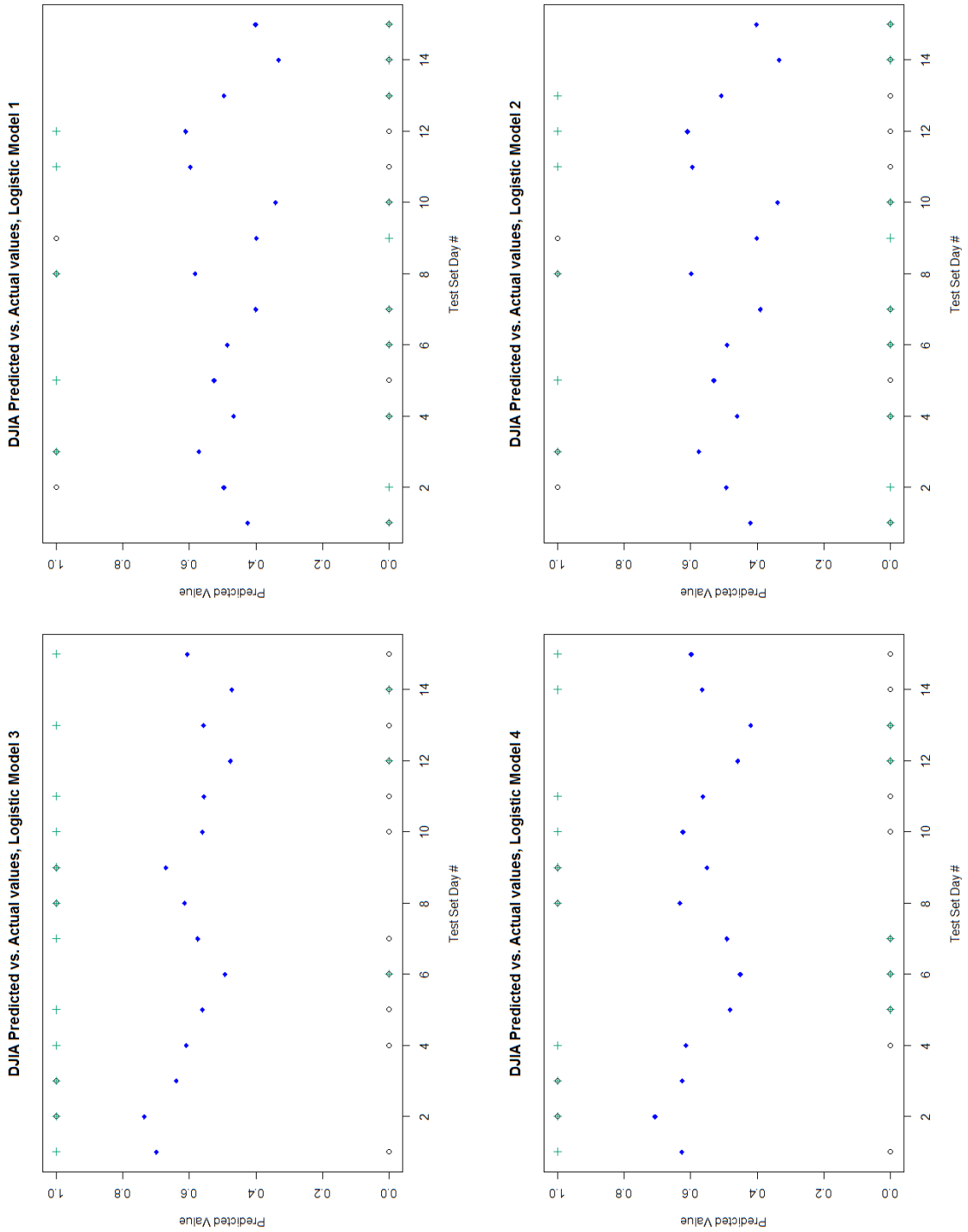
	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Log	10	9	7	9	11	9
Nnet	10	9	7	8	9	9

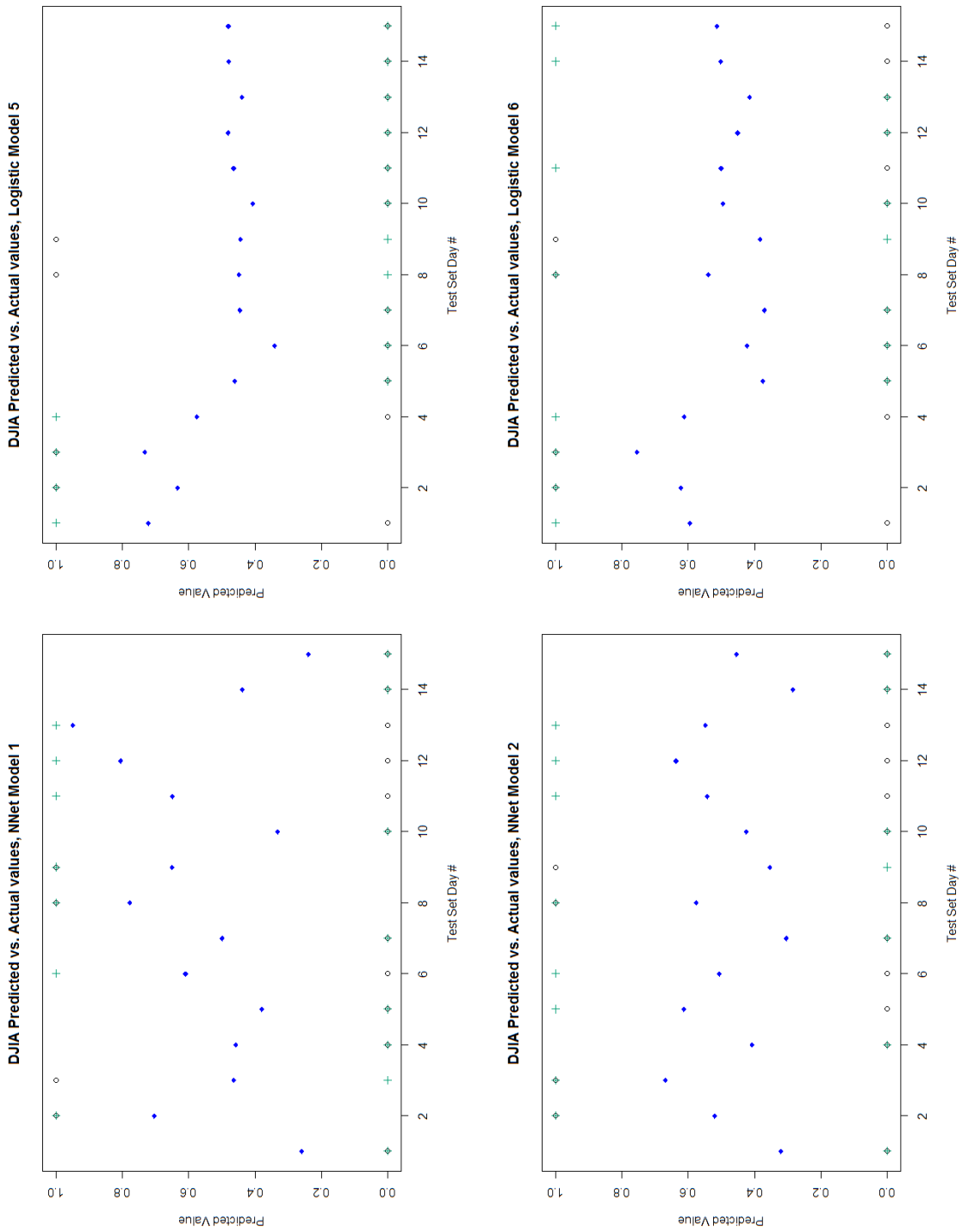
Table 2: Accuracy of VIX prediction by model and method

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Log	7	8	9	7	8	8
NNet	8	8	7	9	10	7

The next set of graphs show the individual predictions and how they compare to the actual index movements. In these graphs, the predicted value as output by the model appears as a blue dot. The green crosses represent the implied movement of the model output, defined as 1 if the output is greater than 0.5 and 0 if the model output is less than 0.5, while the actual movements are depicted as black circles.

The days on which the green cross and black circle are overlapping indicate a correct prediction from the model.




DJIA Predicted vs. Actual values, Logistic Model 5
DJIA Predicted vs. Actual values, Logistic Model 6
DJIA Predicted vs. Actual values, NNet Model 1
DJIA Predicted vs. Actual values, NNet Model 2
Predicted Value
Test Set Day #

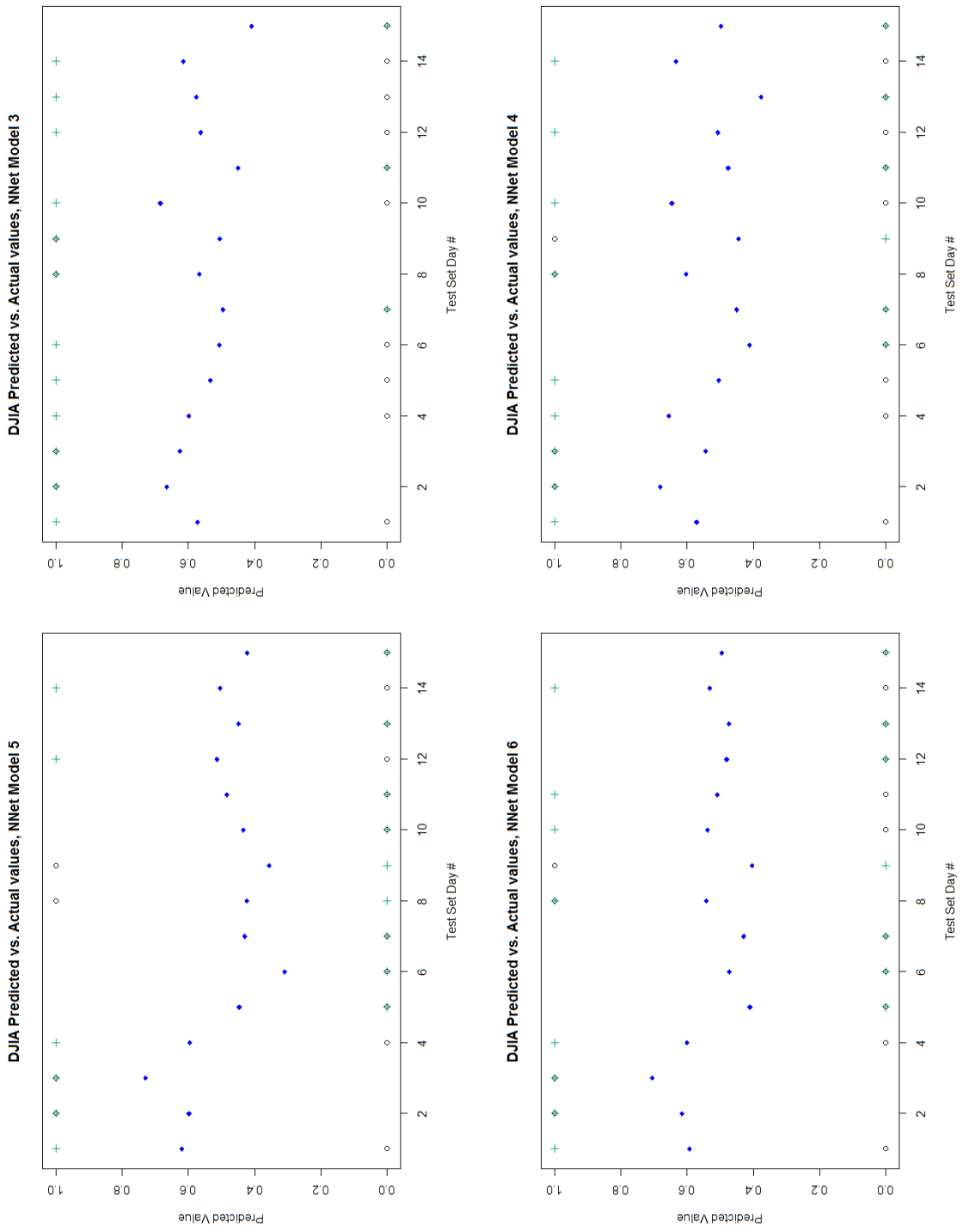
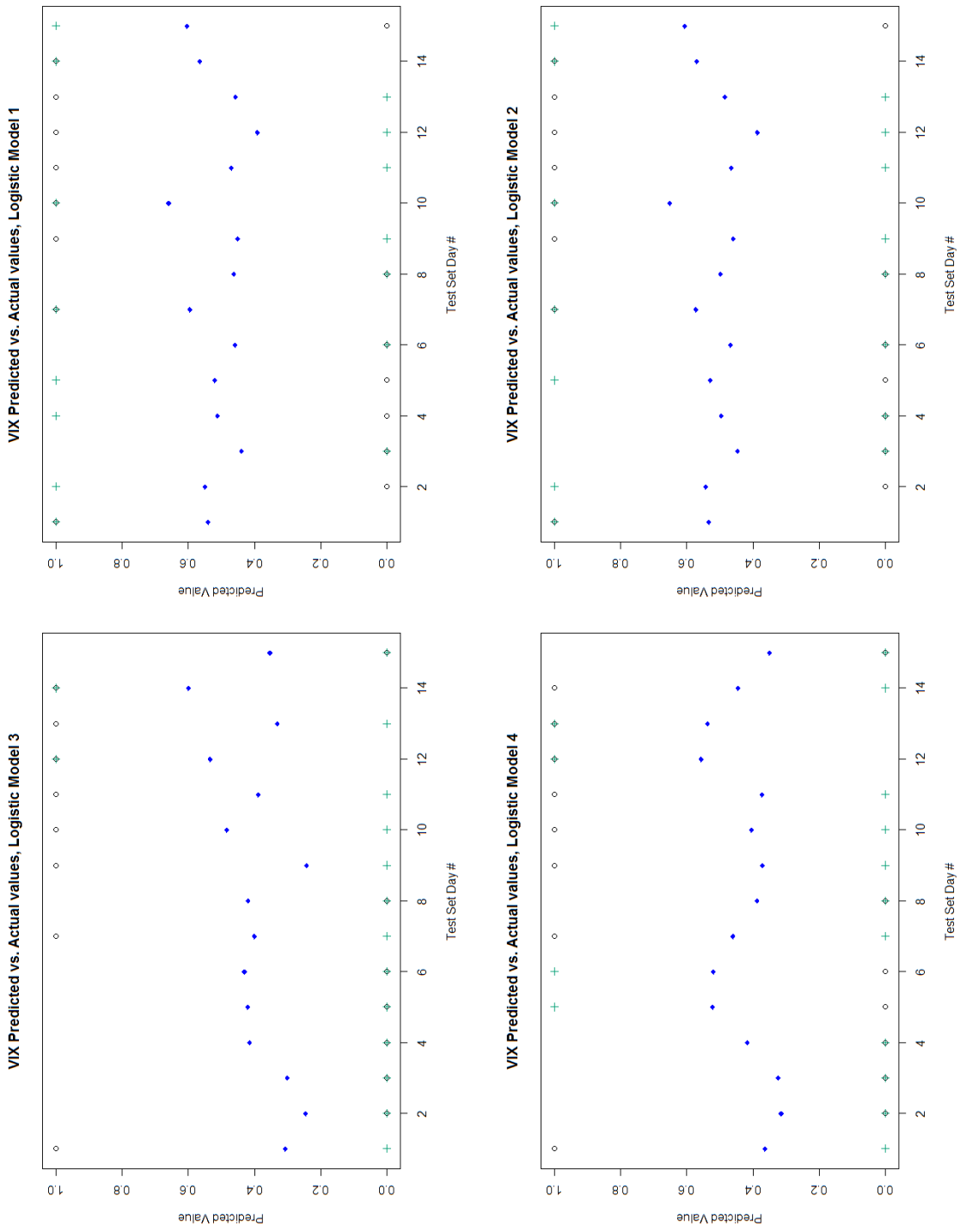
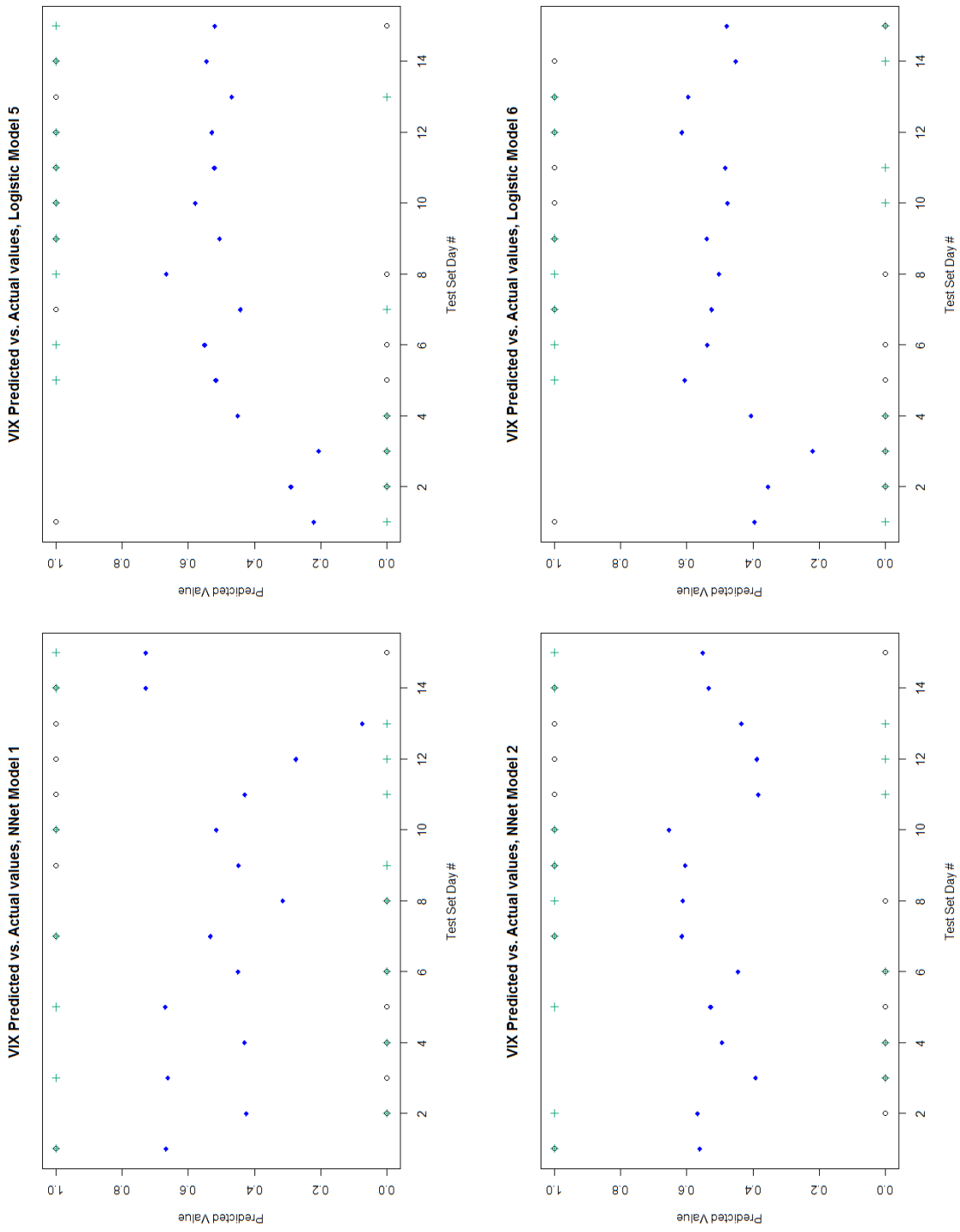


Figure 5: Predicted values for DJIA forecast models


VIX Predicted vs. Actual values, Logistic Model 1
VIX Predicted vs. Actual values, Logistic Model 2
VIX Predicted vs. Actual values, Logistic Model 3
VIX Predicted vs. Actual values, Logistic Model 4
Predicted Value
Test Set Day #
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
12
14


VIX Predicted vs. Actual values, NNet Model 1
VIX Predicted vs. Actual values, Logistic Model 5
VIX Predicted vs. Actual values, NNet Model 2
VIX Predicted vs. Actual values, Logistic Model 6
Predicted Value
Test Set Day #

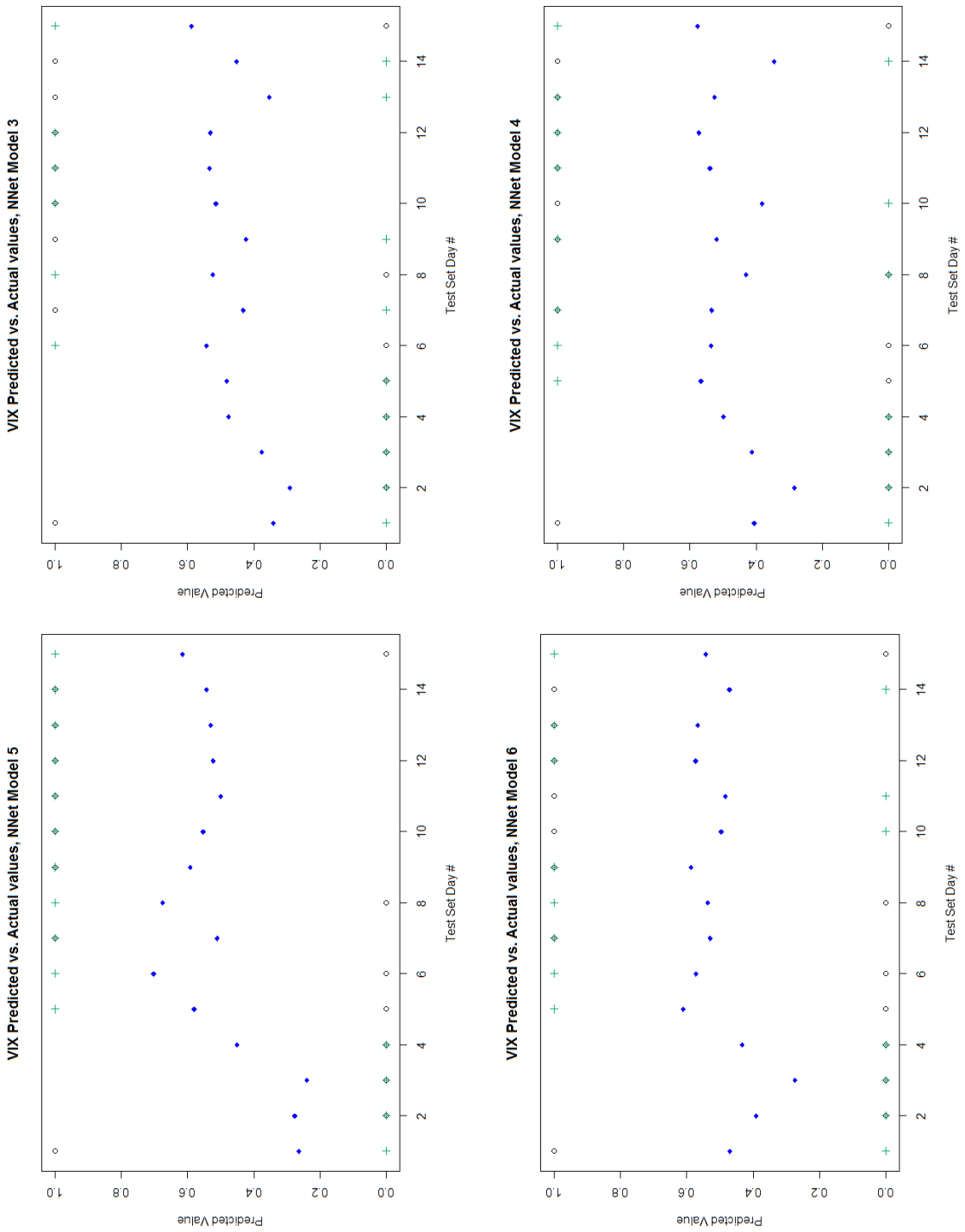


Figure 5: Predicted values for VIX forecast models

Interestingly, although some of the regressions on DJIA have formed a fairly accurate prediction for the test set, there were no statistically significant coefficients in any of the regressions. The full output summaries for the regressions can be found in the appendix.

These results are somewhat in line with previous works, but are also extremely curious in their own right. While in theory the neural network should be as good or better than a logistic regression, it seems that the logistic regression is a better first choice for those without the ability to re-run the hundreds of iterations it might take to achieve the absolute minimum error. Also, while the DJIA results are promising, the VIX results are not quite as impressive. It may be the case that volatility is less correlated to sentiment than is the ending price.

Also worth taking into consideration is some curious differences between the results when including same-day sentiment versus those without. It would be reasonable to assume that including the same-day data can only help, but in the case of DJIA's model 3 and 4, the lag-only model performs better using both methods. Because both time-series are daily, it is also unclear whether the same-day sentiment is occurring before, therefore being a predictor of movement, or occurring after and being a result of the movement. Accurate tick data for the stock prices, as well as a corresponding Twitter data stream, would be necessary to determine this relationship.

5 Conclusion

Most of the models for index movement prediction fell just above fifty-percent accurate, with the average number of correct predictions being 8.9 for DJIA and 8 for VIX. While the predicted values vary considerably between sentiment variables and categorization method, the accuracy for each remains within a similar range. It could be assumed from this result that the actual difference in usefulness between sentiment

variables is very low, as they all measure the same thing: general public sentiment. One possible concern is the lack of statistically significant regressors even in the most accurate models. The only explanation for this that seems in any way plausible is that, while individually the variables are next to useless, some combinations of them may be very significant.

As seen in the graph of the Valence dimension seen below, the sentiment measures do seem to coincide with other events, such as a huge increase around Christmas and New Years, and a sharp decrease during the time of the November terror attacks in Paris. So the sentiment data derived from ANEW does appear to be valid and satisfy expectations with regard to real life correlation.

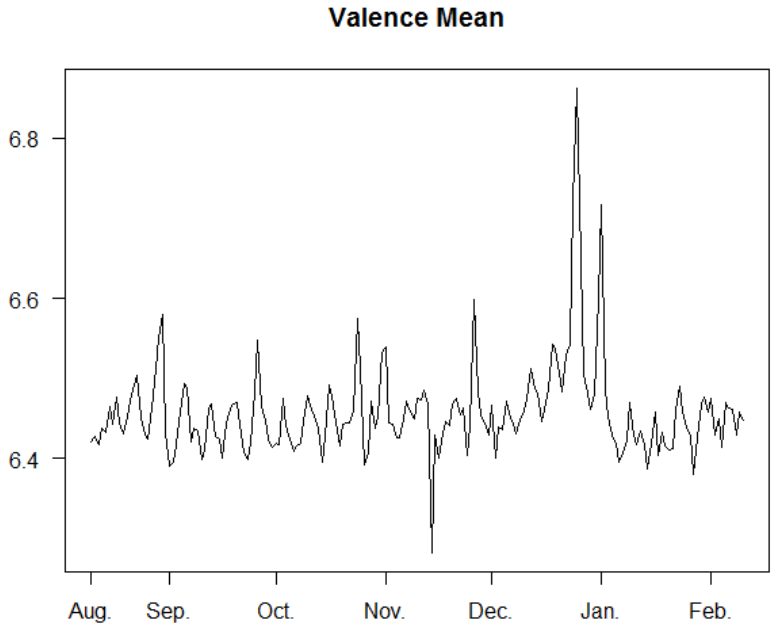


Figure 6: Valence score means across the study period

For both indices the neural network provides far less advantage than expected, and in the DJIA actually performs much worse. The most likely explanation for the lesser predictive ability of the neural network, when compared to the logistic regression, is the previously mentioned lack of a convex cost function. Because gradient

descent stops immediately upon finding any minimum point, the logistic regression can always find its global minimum error level, while it could potentially take hundreds of iterations before finding the global minimum for a neural network's error. It may also be the case that this neural network's structure or one of its learning parameters is not ideal for the task.

The results of these models suggest that, at least for the tested period, the sentiment of the public is a stronger predictor for the movement of the DJIA index than for the VIX index. While there is a strong possibility that this particular set of predictions are just a fluke, and would not carry over to another set of tweet and stock data, based only on the results here it appears that there are some predictable patterns in stock movement.

The lack of significant predictive ability found in this set of sentiment variables, in contrast to the existing studies on the topic, supports the efficient market hypothesis. While it cannot be stated confidently that there is no value in the information found in the Twitter sentiment variable, it also does not differ enough from randomness in this small sample size to positively state that there is. However, the differences in results may also be caused by the choice of sentiment variable, or even the possibility that increased awareness of the phenomenon has changed peoples' behavior since the original study. The previous works have shown that there is a very real potential for finding a very powerful predictive method using social network sentiment, but continued iteration and experimentation with methods and models are necessary to conclusively determine the most effective techniques.

References

Werner Antweiler and Murray Z. Frank. Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, 59(3):1259–1294, Jun. 2004.

Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, Oct. 2010.

Margaret M. Bradley and Peter J. Lang. Affective norms for english words (anew): Instruction manual and affective ratings. *Technical Report C-1, The Center for Research in Psychophysiology, University of Florida*, 1999.

Jon Elster. Emotions and economic theory. *Journal of Economic Literature*, 36(1): 47–74, Mar. 1998.

Adam Fadlalla and Chien-Hua Lin. An analysis of the applications of neural networks in finance. *Interfaces*, 31(4):112–122, 2001.

Christopher G. Healey and S. Ramaswamy. Visualizing twitter sentiment. `https://www.csc.ncsu.edu/faculty/healey/tweet_viz/`, 2011.

David Hirschleifer and Tyler Shumway. Good day sunshine: Stock returns and the weather. *The Journal of Finance*, 58(3):1009–1032, Jun. 2003.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. 2016.

George Loewenstein. Emotions in economic theory and economic behavior. *The American Economic Review*, 90(2):426–432, May. 2000.

Huina Mao, Scott Counts, and Johan Bollen. Predicting financial markets: Comparing survey, news, twitter and search engine data. *arXiv preprint:1112.1051*, 2011.

Anshul Mittal and Arpit Goel. Stock prediction using twitter sentiment analysis. *Stanford University Working Paper*, 2012.

Robert Neal and Simon M. Wheatley. Do measures of investor sentiment predict returns? *The Journal of Financial and Quantitative Analysis*, 33(4):523–547, Dec. 1998.

Finn Arup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages 718 in CEUR Workshop Proceedings*, pages 93–98, 2011.

Gabriele Ranco, Darko Aleksovski, Guido Caldarelli, Miha Grar, and Igor Mozeti. The effects of twitter sentiment on stock price returns. *PLoS ONE*, 10(9):1–21, 2015.

James A. Russell and Albert Mehrabian. Evidence for a three-factor theory of emotions. *Journal of Research in Personality*, 11:273–294, 1977.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavioral Research Methods*, 45(4):1191–1207, 2013.

Halbert White. Economic prediction using neural networks: The case of ibm daily stock returns. *Neural Networks, 1988., IEEE International Conference on*, pages 451–458, 1988.

Xue Zhang, Hauke Fuehres, and Peter A. Gloor. Predicting stock market indicators through twitter: i hope it is not as bad as i fear. *Procedia - Social and Behavioral Sciences*, 26:55–62, 2011.

6 Appendix: Logarithmic Regression Output

```
Call:
glm(formula = djia_model_anew1, family = binomial(link = "logit"),
    data = train.set)


Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6341  -1.1568  -0.1535   1.0779   1.8052


Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.0264     1.1909   3.381 0.000722 ***
vMean_        1.5586     6.4155   0.243 0.808051
vMean.l1     -6.0903     9.0873  -0.670 0.502729
vMean.l2      3.8954     7.6440   0.510 0.610334
vMean.l3     -1.0116     5.4452  -0.186 0.852622
aMean_       -1.1390     2.6863  -0.424 0.671568
aMean.l1     -0.3374     3.1655  -0.107 0.915115
aMean.l2     -1.2736     3.1568  -0.403 0.686609
aMean.l3      1.8023     2.4625   0.732 0.464235
dMean_       -3.9835     5.2678  -0.756 0.449529
dMean.l1      2.2963     7.4003   0.310 0.756331
dMean.l2     -5.4015     6.5212  -0.828 0.407499
dMean.l3     -1.8215     4.6693  -0.390 0.696465
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 212.05  on 154  degrees of freedom
AIC: 238.05

Number of Fisher Scoring iterations: 4

Call:
glm(formula = djia_model_anew1f, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5472  -1.1768  -0.1421   1.0862   1.8151

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.1665     1.0191   3.107  0.00189 **
vMean.l1     -7.1769     6.4730  -1.109  0.26754
vMean.l2      5.0267     7.1222   0.706  0.48032
vMean.l3     -1.4784     5.2634  -0.281  0.77880
aMean.l1     -1.1672     2.6629  -0.438  0.66115
aMean.l2     -0.7189     3.0178  -0.238  0.81171
aMean.l3      1.7620     2.4194   0.728  0.46646
dMean.l1      1.7557     5.1159   0.343  0.73145
dMean.l2     -6.0482     6.1334  -0.986  0.32408
```

```
dMean.l3     -1.2400     4.5036  -0.275  0.78306
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 215.10  on 157  degrees of freedom
AIC: 235.1

Number of Fisher Scoring iterations: 4

Call:
glm(formula = djia_model_anew2, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.716  -1.139  -0.103   1.071   1.628

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   6.11503    1.62178   3.771 0.000163 ***
vMean2_       0.02235    8.47127   0.003 0.997895
vMean2.l1    -3.30324    9.79063  -0.337 0.735825
vMean2.l2     6.55276    9.78901   0.669 0.503240
vMean2.l3    -2.70340    8.03101  -0.337 0.736403
```

```
aMean2_      -1.55549    3.40654  -0.457 0.647946
aMean2.l1    -2.15642    3.94054  -0.547 0.584214
aMean2.l2     3.22841    3.92711   0.822 0.411029
aMean2.l3     0.94237    3.34373   0.282 0.778073
dMean2_      -3.98118    5.95023  -0.669 0.503445
dMean2.l1     0.86459    6.85064   0.126 0.899569
dMean2.l2   -10.44351    6.96462  -1.500 0.133742
dMean2.l3    -1.41226    5.76891  -0.245 0.806607
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 207.46  on 154  degrees of freedom
AIC: 233.46

Number of Fisher Scoring iterations: 5

Call:
glm(formula = djia_model_anew2f, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6758  -1.1532  -0.1435   1.0937   1.5424
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.7190     1.3726   3.438 0.000586 ***
vMean2.l1    -6.7003     8.3935  -0.798 0.424710
vMean2.l2     6.8524     9.5858   0.715 0.474700
vMean2.l3    -3.0682     7.8329  -0.392 0.695278
aMean2.l1    -3.4311     3.4621  -0.991 0.321668
aMean2.l2     3.6723     3.8218   0.961 0.336603
aMean2.l3     1.0336     3.2944   0.314 0.753721
dMean2.l1     1.6621     5.7301   0.290 0.771760
dMean2.l2   -10.0894     6.7606  -1.492 0.135599
dMean2.l3    -0.8557     5.6337  -0.152 0.879276
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 212.07  on 157  degrees of freedom
AIC: 232.07

Number of Fisher Scoring iterations: 4

Call:
glm(formula = djia_model_sun, family = binomial(link = "logit"),
    data = train.set)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4968  -1.1574  -0.8568   1.1597   1.5324

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1657     0.4540  -0.365    0.715
sun           0.0987     0.6994   0.141    0.888
sun.l1        1.0835     0.8134   1.332    0.183
sun.l2       -0.2350     0.7935  -0.296    0.767
sun.l3       -0.6483     0.6905  -0.939    0.348

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 227.41  on 162  degrees of freedom
AIC: 237.41

Number of Fisher Scoring iterations: 4

Call:
glm(formula = djia_model_sunf, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4863  -1.1583  -0.8619   1.1631   1.5441
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1421     0.4220  -0.337    0.736
sun.l1        1.1425     0.6981   1.637    0.102
sun.l2       -0.2577     0.7771  -0.332    0.740
sun.l3       -0.6336     0.6824  -0.928    0.353

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 227.43  on 163  degrees of freedom
AIC: 235.43

Number of Fisher Scoring iterations: 4

Call:
glm(formula = vix_model_anew1, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8008  -1.0955   0.1015   1.1531   1.7709

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.6064     1.1716  -3.078  0.00208 **
```

```
vMean_       -6.7123     6.3138  -1.063  0.28773
vMean.l1      8.4328     9.1417   0.922  0.35629
vMean.l2     -0.9185     8.6335  -0.106  0.91527
vMean.l3     -1.6199     5.6226  -0.288  0.77327
aMean_        1.2968     2.6741   0.485  0.62770
aMean.l1      0.5965     3.0575   0.195  0.84532
aMean.l2      1.0482     3.2581   0.322  0.74767
aMean.l3     -0.4440     2.4176  -0.184  0.85430
dMean_        8.1882     5.3002   1.545  0.12237
dMean.l1     -5.8371     7.4540  -0.783  0.43358
dMean.l2      5.0235     7.1627   0.701  0.48309
dMean.l3      1.4216     4.7557   0.299  0.76500
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 211.56  on 154  degrees of freedom
AIC: 237.56

Number of Fisher Scoring iterations: 5

Call:
glm(formula = vix_model_anew1f, family = binomial(link = "logit"),
    data = train.set)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8470  -1.1032   0.1701   1.1797   1.5264

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.6363     0.9841  -2.679  0.00739 **
vMean.l1      4.8833     6.3205   0.773  0.43975
vMean.l2     -1.6685     7.7547  -0.215  0.82964
vMean.l3     -1.3046     5.3109  -0.246  0.80595
aMean.l1      1.0958     2.5822   0.424  0.67129
aMean.l2      0.7067     3.0937   0.228  0.81930
aMean.l3     -0.5046     2.3519  -0.215  0.83011
dMean.l1     -1.4958     5.0781  -0.295  0.76834
dMean.l2      5.3151     6.5510   0.811  0.41717
dMean.l3      0.9249     4.5027   0.205  0.83726
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 216.34  on 157  degrees of freedom
AIC: 236.34

Number of Fisher Scoring iterations: 4
```

```
Call:
glm(formula = vix_model_anew2, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6960  -1.0503   0.1513   1.0995   1.8138

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.408      1.621  -3.337 0.000847 ***
vMean2_       -9.651      8.676  -1.112 0.265943
vMean2.l1      6.552      9.998   0.655 0.512227
vMean2.l2     -4.624     10.270  -0.450 0.652515
vMean2.l3     -3.980      8.056  -0.494 0.621269
aMean2_        3.939      3.519   1.119 0.263004
aMean2.l1      2.830      4.008   0.706 0.480079
aMean2.l2     -5.288      4.143  -1.276 0.201864
aMean2.l3      2.725      3.275   0.832 0.405346
dMean2_        9.849      6.242   1.578 0.114579
dMean2.l1     -4.470      7.051  -0.634 0.526116
dMean2.l2     12.363      7.235   1.709 0.087497 .
dMean2.l3      2.526      5.853   0.432 0.666032
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 203.93  on 154  degrees of freedom
AIC: 229.93

Number of Fisher Scoring iterations: 5

Call:
glm(formula = vix_model_anew2f, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5897  -1.0886   0.1454   1.1230   1.6878

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -4.125      1.346  -3.065  0.00217 **
vMean2.l1      5.100      8.349   0.611  0.54129
vMean2.l2     -3.995      9.991  -0.400  0.68924
vMean2.l3     -4.139      7.748  -0.534  0.59319
aMean2.l1      4.402      3.421   1.287  0.19824
aMean2.l2     -5.431      3.959  -1.372  0.17011
aMean2.l3      2.636      3.189   0.827  0.40842
dMean2.l1     -2.159      5.768  -0.374  0.70813
dMean2.l2     11.336      6.945   1.632  0.10261
dMean2.l3      2.280      5.642   0.404  0.68608
```

```
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 210.39  on 157  degrees of freedom
AIC: 230.39

Number of Fisher Scoring iterations: 5

Call:
glm(formula = vix_model_sun, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4410  -1.1767   0.9421   1.1680   1.4127

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.0643     0.4522  -0.142    0.887
sun           0.2172     0.6977   0.311    0.756
sun.l1       -0.5183     0.8027  -0.646    0.519
sun.l2       -0.3492     0.7892  -0.442    0.658
sun.l3        0.8071     0.6888   1.172    0.241
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 229.24  on 162  degrees of freedom
AIC: 239.24

Number of Fisher Scoring iterations: 4

Call:
glm(formula = vix_model_sunf, family = binomial(link = "logit"),
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4057  -1.1701   0.9252   1.1658   1.3900

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.01205    0.41976  -0.029    0.977
sun.l1      -0.38827    0.68482  -0.567    0.571
sun.l2      -0.39832    0.77316  -0.515    0.606
sun.l3       0.83821    0.68171   1.230    0.219

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 231.51  on 166  degrees of freedom
Residual deviance: 229.33  on 163  degrees of freedom
```

```
AIC: 237.33

Number of Fisher Scoring iterations: 4
```