Machine Learning for Finance: Assignment 3

Due on Wednesday, June 20, 2018

Argimiro Arratia

Bradley Kester & Simon Neumeyer (Team Crow)

1 Sentiment Analysis Implementing Deep Neural Networks and Support Vector Machine Models

In this section we analysed the potential predictive ability of sentiment indicators in forecasting one-step ahead realisations of the NETFLIX share price. Two analyses were conducted across two epochs, firstly, from January 1, 2017 until June 30 of that year, and secondly, from July 1 until December 30, again in 2017. For each epoch, we used two different machine learning techniques being support vector machines and neural networks.

Table 1: **Epoch One Analysis**

	SVM Lags	**NN Lags**	**SVM Bullish**	**NN Bullish**	**SVM Bearish**	**NN Bearish**
MSE	3.82	13.29	135.67	43.05	13.77	17.40
SSR	91.45	417.719	3798.77	1205.27	330.42	417.72
NRMSE	.24	.46	.87	.49	.42	.47
% Outperforming	75.51%	54.28%	13.37%	51.20%	58.26%	53.07%

In terms of NRMSE and percentage that outperform sample expected value (% outperforming) for the first epoch the best performing forecast was the model that only considered the three previous lags of the time-series utilising a support vector machine. This model exhibits a NRMSE of .24 and a % outperforming of 75.51%, both of which are significantly better than all alternatives. Looking across the different models we find that between support vector machines and neural networks tend to perform similarly with models excluding sentiment measures the most accurate. This said, support vector machine models that incorporate bearish sentiment measures are the second best model as indicated by NRMSE and % outperforming.

Table 2: **Epoch Two Analysis**

	SVM Lags	**NN Lags**	**SVM Bullish**	**NN Bullish**	**SVM Model**	**NN Bearish**
MSE	14.25	49.16	101.31	26.29	58.72	44.06
SSR	399.15	1376.45	2836.64	736.02	1644.19	1233.62
NRMSE	.31	.57	.76	.39	.63	.55
% Outperforming	69.34%	43.06%	24.06%	61.32%	36.36%	44.87%

In terms of NRMSE and percentage that outperform sample expected value (% outperforming) for the second epoch the best performing forecast was the model that only considered the three previous lags of the time-series utilising a support vector machine. This model exhibits a NRMSE of .31 and a % outperforming of 69.34%, both of which are significantly better than all alternatives. Looking across the different models again we find that between support vector machines and neural networks tend to perform similarly, although, this time with greater variance. In this case, the second best model as indicated by NRMSE and % outperforming the second best model is the neural network model that includes bullish sentiment data.

As a general comment we can determine that the optimal model is that including only lag data, discounting the sentiment indices that relies on the use of support vector machines rather than neural networks. Visual plots of each neural network can be found in the appendix.

2 Portfolio Optimisation

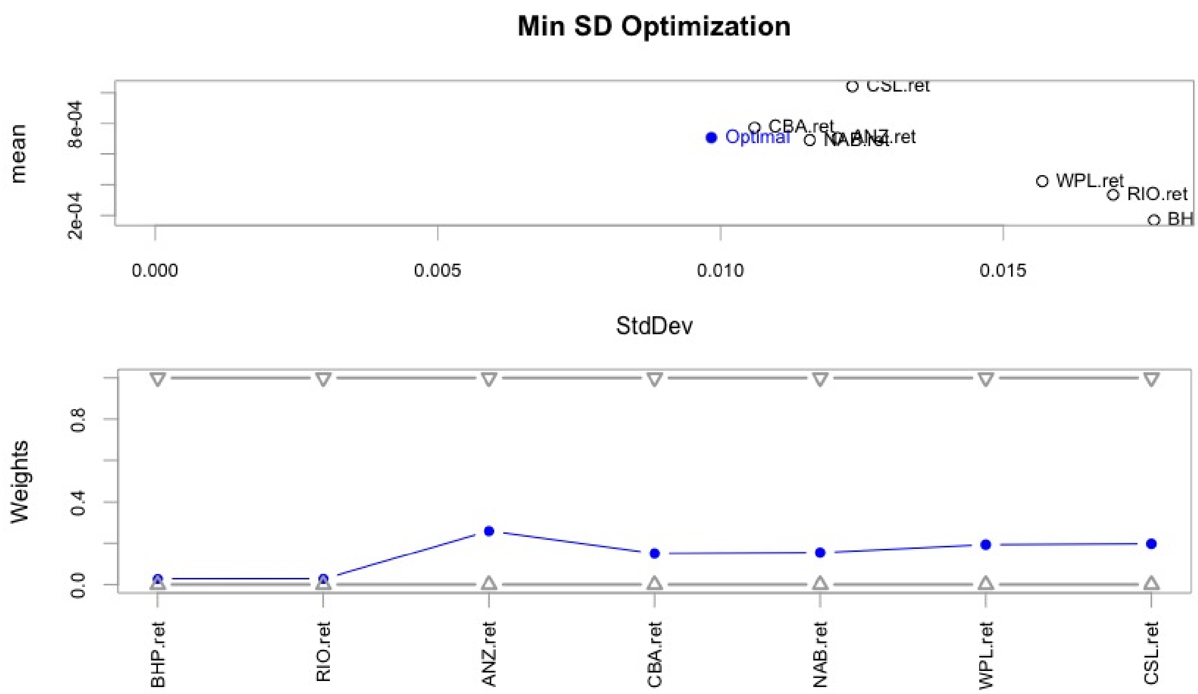
In this section we analysed 7 stocks: BHP Billiton (BHP), Rio Tinto (RIO), Woodside Petroleum (WPL), CSL Behring (CSL), The Commonwealth Bank of Australia (CBA), The National Australian Banking Group (NAB) and The Australia, and New Zealand Banking Group (ANZ). We did so in order to attain optimal portfolios in terms of a minimum-variance long-only constraint set and a maximum return long-short (unrestricted) constraint set.

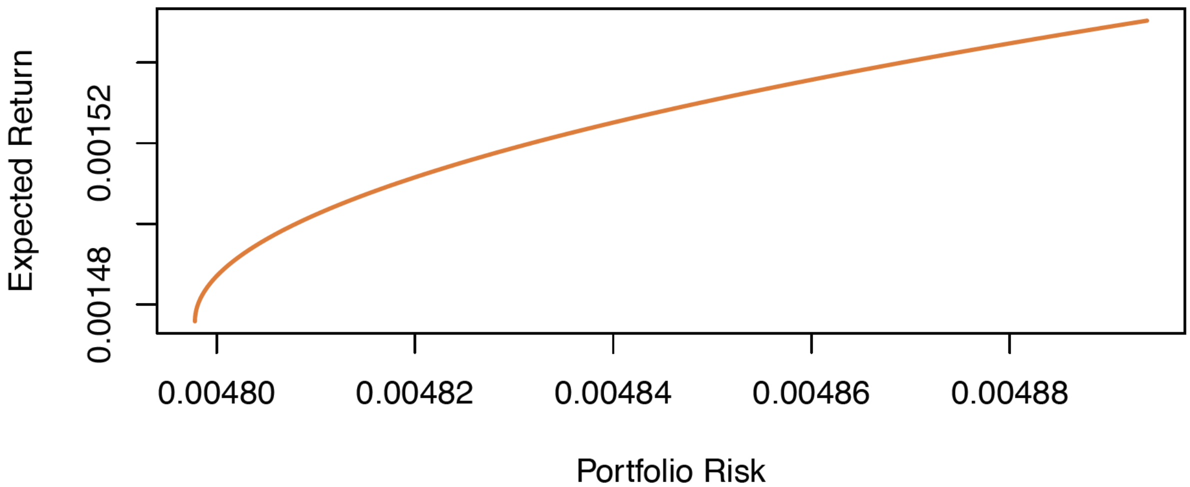
We found that for the long-only portfolio optimised to minimise the standard deviation of returns to include greater proportions of of ANZ, WPL, and NAB. These shares exhibited lesser variance than others in our sample during the period of analysis. Moreover, when examining the portfolio optimised for greatest returns we note the greater presence of CSL and CBA as the overall highest returning shares during the epoch investigated.

Under each constraint set the optimal allocation were determined to be the following:

Table 3: **Optimal Portfolio Allocations**

	BHP	**RIO**	**WPL**	**CSL**	**CBA**	**ANZ**	**NAB**
Long-only	0.0274	0.0279	0.1933	0.1975	0.1507	0.2587	0.1545
Long-short	0.0057	0.2412	0.0336	0.3715	0.2719	0.0105	0.0756




Expected Return
0.00148
0.00152
0.00480
0.00482
0.00484
0.00486
0.00488
Portfolio Risk

3 Circular Autocovariance

To start with, we assume the following framework with some values s and l: At each step j of the bootstrap,

let $Y^*_{j-1} = yI_{l+s}$, so:
$Y^*_j = yI_j$ with probability p
$Y^*_j = yI_{l+s+1}$ with probability $1-p$

Therefore, we can express $E[Y^*_j]$ as the following:

$$E[Y^*_j|y_1,...,y_n] = E[y^*_1|y_1,...,y_n]p + E[y^*_{I\,l+s+1}|y_1,...,y_n](1-p)$$

For the following step we assume y_t to be covariance stationary, which means $E[y_t|y_1,...,y_n = \hat{y}\forall t$ Thus:

$$\begin{aligned} E[Y^*_j] &= E[y_t]p + E[y_t](1-p) \\ &= \bar{y}p + \bar{y}(1-p) \\ &= \bar{y}(p+1+p) \\ &= \bar{y} \end{aligned}$$

When we check $cov(Y^*_1, Y^*_{j+1}$, recall that when $Y^*1 = y_{I1}$. Furthermore, we assume covariance stationarity of the time series again $cov(y_s, y_{s+1} = covcov(y_j, y_{j+1})\forall s, j$. This depends on order, not on time. using the algorithm given in 8.3, we show that the covariance between the first and second bootstrap values is given by:

$$\begin{aligned} cov(y^*_1, Y^*_2) &= E[(y_{I1}-\mu)[p(y_{I2}-\mu) + (1-p)(y_{I1}-\mu)]] \\ &= pE[(y_{I1}-\mu)(y_{I2}-\mu) + (1-p)E[(y_{I1}-\mu)(y_{I1+1}-\mu)] \end{aligned}$$

Now, we can see the first part of the equation will be zero (as there are two random numbers), especially if we are dealing with a large sample. Therefore, we are left with:

$$\begin{aligned} cov(Y^*_1, Y^*_2) &= (1-p)E[(y_{I1}-\mu)(y_{I1+1}-\mu \\ &= (1-p)cov(y_t, y_{t+1}) \\ &= (1-p)c_1 \end{aligned}$$

Moving on, to make the pattern clearer we can show the second order covariance in the following step:

$$\begin{aligned} cov(Y^*_1, Y^*_3) &= E[(y_{I1}-\mu)[p(y_{Is}-\mu) + (1-p)[p(y_{Is}-\mu) + (1-p)[p(y_{I1+2}-\mu)]]] \\ &= (1-p)^2 E[(y_{I1}-\mu)(y_{I1+2}-\mu)] \\ &= (1-p)^2 cov(y_t, y_{t+2} \\ &= (1-p)^2 c_2 \end{aligned}$$

Henceforth, as we find we iterate the previous until we arrive the $(j+1)^{th)}$ order autocovariance:

$$\begin{aligned} cov(Y^*_1, Y*_{1+j} &= (1-p)^j E[(y_{I1}-\mu)(y_{I1+j}-\mu)] \\ &= (1-p)^j cov(y_t, y_{t+j} \\ &= (1-p)^2 c_j \end{aligned}$$

4 Mean-Variance Approximation to Expected Utility

In order to show the mean-variance approximation of the expected utility, first we must take a second order Taylor expansion around E(C):

$$u(C) \approx u[E(C)] + u'[E(C)][C - E(C)] + \frac{1}{2}u'[E(C)][C - E(C)]^2$$

Taking expectation on the Taylor expansion of u(C):

$$\begin{aligned} E[u(C)] &\approx u[E(C)] + u'[E(C)]E[C - E(C)] + 1/2u''[E(C)]E[C - E(C)]^2 \\ &\approx u[E(C)] + 1/2u''[E(C)var(C) \end{aligned}$$

Since we are implementing quadratic approximation of a utility function, we can interchange utility with expectations:

$$E[u(C)] = u[E(C)]$$

Moreover, as utility is unaffected by linear transformations we can change our function to:

$$\hat{u}[E(C)] \approx \frac{u[E(C)}{u'[E(C)]} + \frac{1}{2}\frac{u''[E(C)]}{u'[E(C)]}var(C)$$

Finally, we can add the mean of C and subtract the first term:

$$\hat{u}[E(C)] \approx E(C) - \frac{u[E(C)}{u'[E(C)]} + \frac{u[E(C)}{u'[E(C)]} + \frac{1}{2}\frac{u''[E(C)]}{u'[E(C)]}var(C)$$

Which yields:

$$E[\hat{u}(C)] \approx u[E(C)] - \frac{1}{2}var(C) \text{ for some } \lambda > 0$$

Where λ represents the risk aversion parameter

5 Portfolio Return Proof

To be proven:

$$Var\left(\sum_{i=1}^{n} a_i X_i\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \sigma_i \sigma_j \rho_{ij}$$

We start with the case of only two variables and then expand to n variables.

$$\begin{aligned} Var(a_1X_1 + a_2X_2) &= Var(a_1X_1) + Var(a_2X_2) + 2Cov(a_1X_1, a_2X_2) \\ &= a_1^2 Var(X_1) + a_2^2 Var(X_2) + 2a_1a_2Cov(a_1X_1, X_2) \\ &= a_1^2\sigma_1^2 + a_2^2\sigma_2^2 + 2a_1a_2Cov(a_1X_1, X_2) \\ &= a_1^2\sigma_1^2 + a_2^2\sigma_2^2 + 2a_1a_2\sigma_1\sigma_2\rho_{12} \end{aligned}$$

$$\rho_{12} = \frac{Cov(X_1, X_2)}{\sigma_1\sigma_2} \rightarrow Cov(X_1, X_2) = \sigma_1\sigma_2\rho_{12}$$

Taking sums:

$$\begin{aligned} Var\left(\sum_{i=1}^{2} a_i X_i\right) &= \sum_{i=1}^{2}\sum_{j=1}^{2} a_i a_j \sigma_i \sigma_j \rho_{ij} \\ &= a_1^2\sigma_1^2 + a_2^2\sigma_2^2 + a_1a_2\sigma_1\sigma_2\rho_{12} + a_1a_2\sigma_1\sigma_2\rho_{21} \\ &\text{where } \rho_{12} = \rho_{21} \text{ and } \rho_{11} = \rho_{22} = 1 \end{aligned}$$

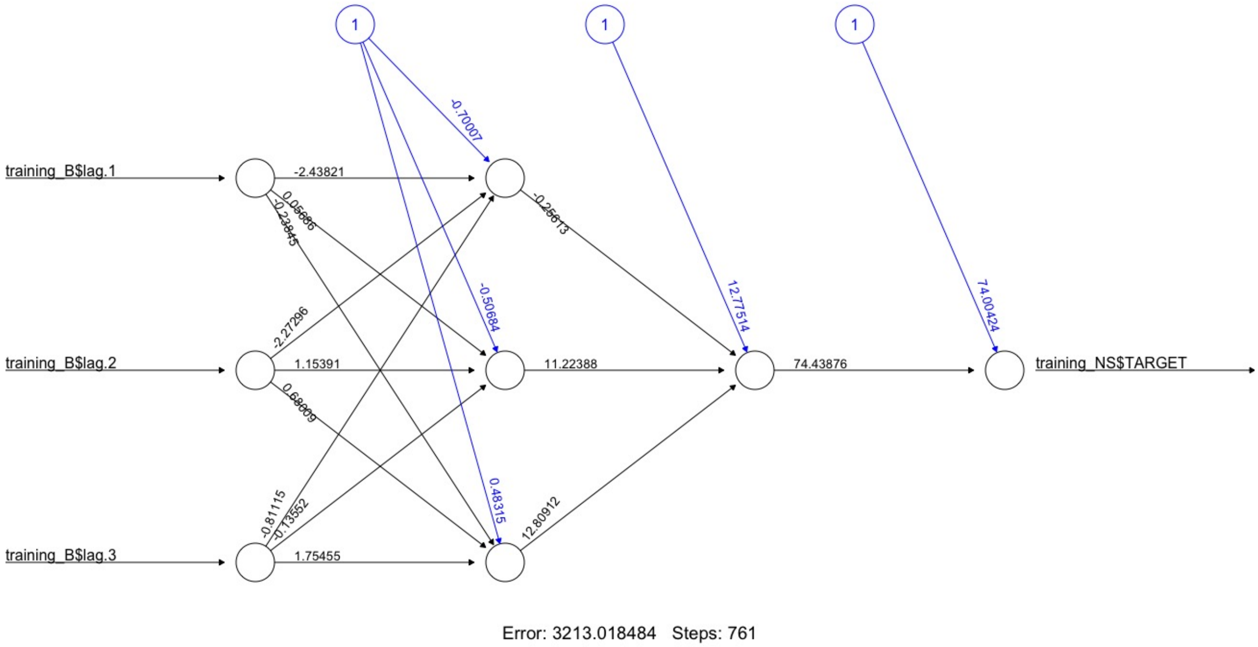
Thus, by induction we can generalize for n variables:

$$Var\left(\sum_{i=1}^{n} a_i X_i\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \sigma_i \sigma_j \rho_{ij}$$

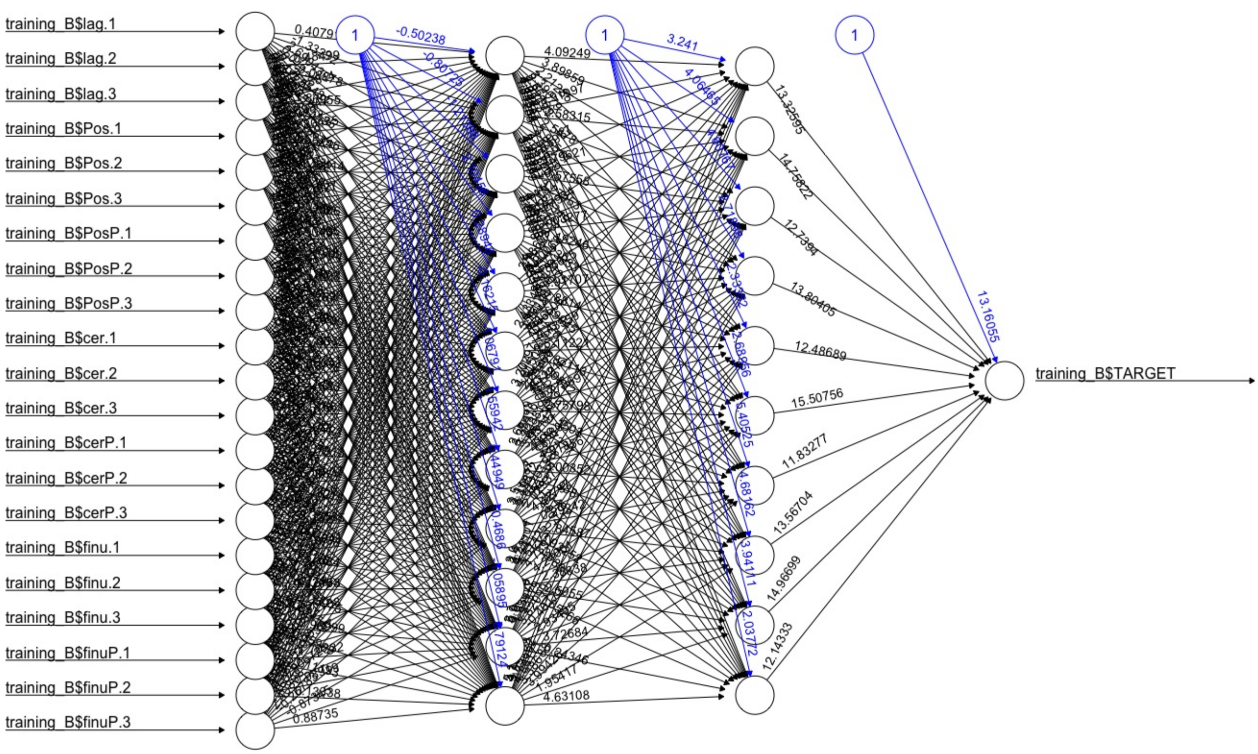
6 Appendix

6.1 Epoch One

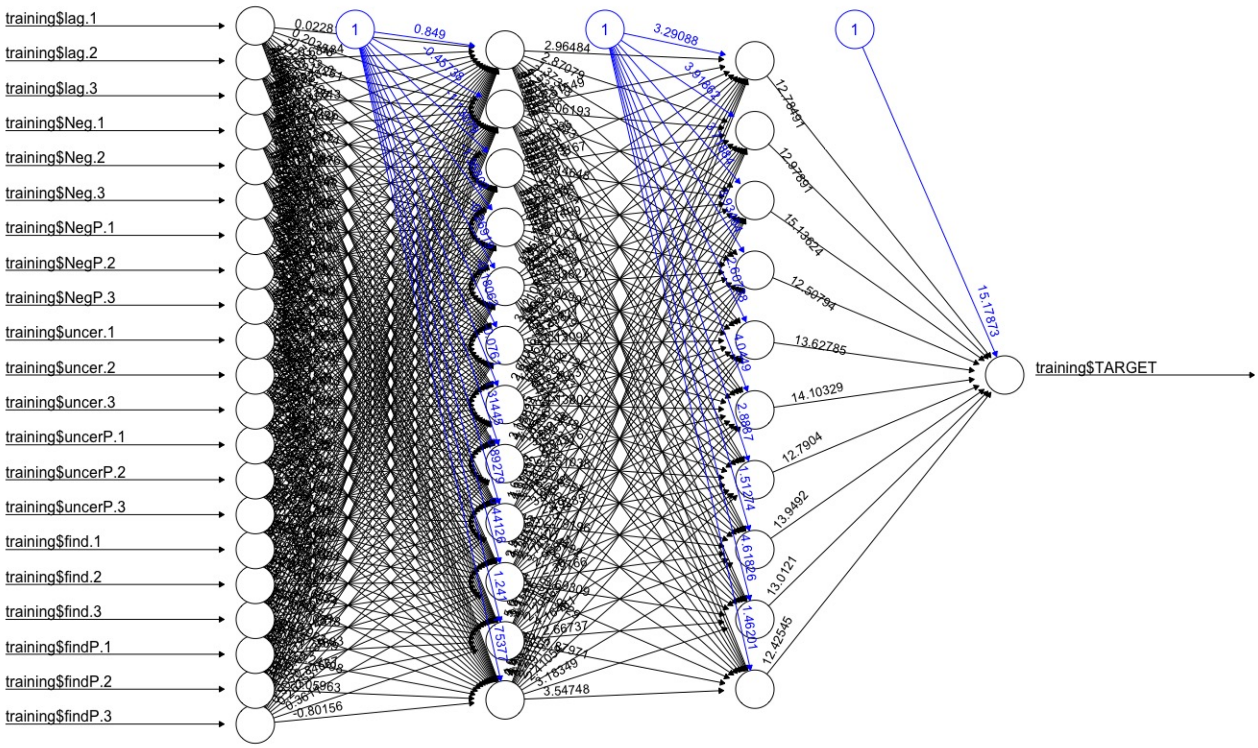
Neural Network Plot - Lags Only



Neural Network Plot - Bullish Sentiment Included

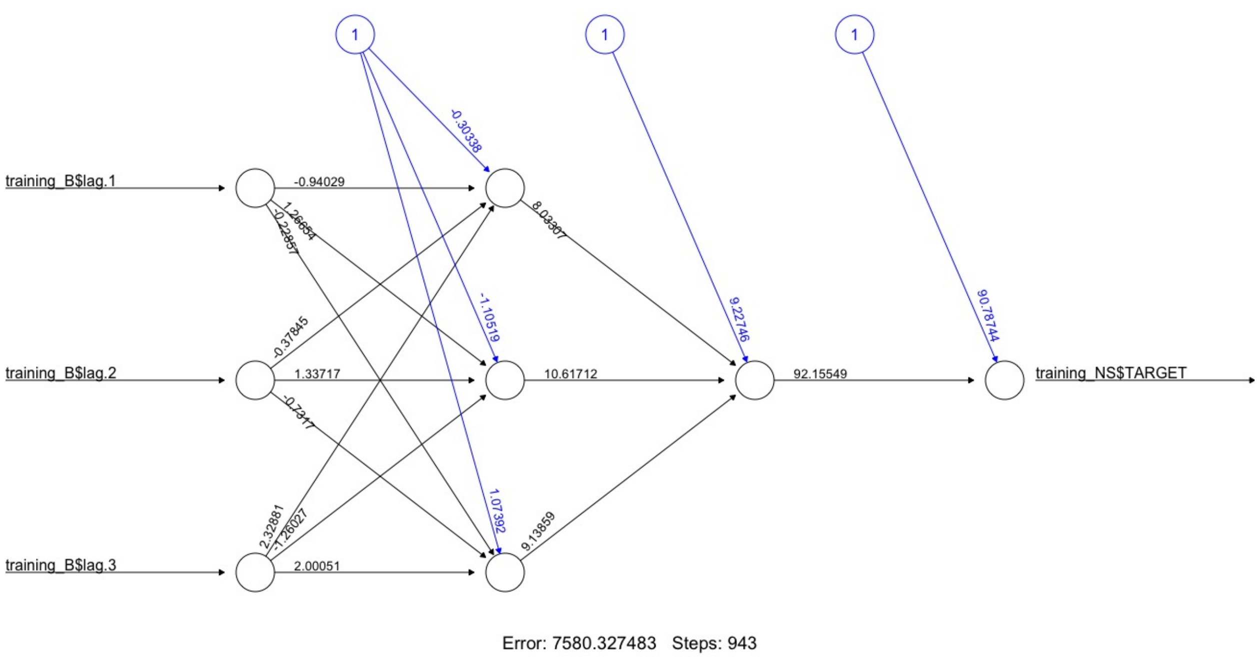


Neural Network Plot - Bearish Sentiment Included

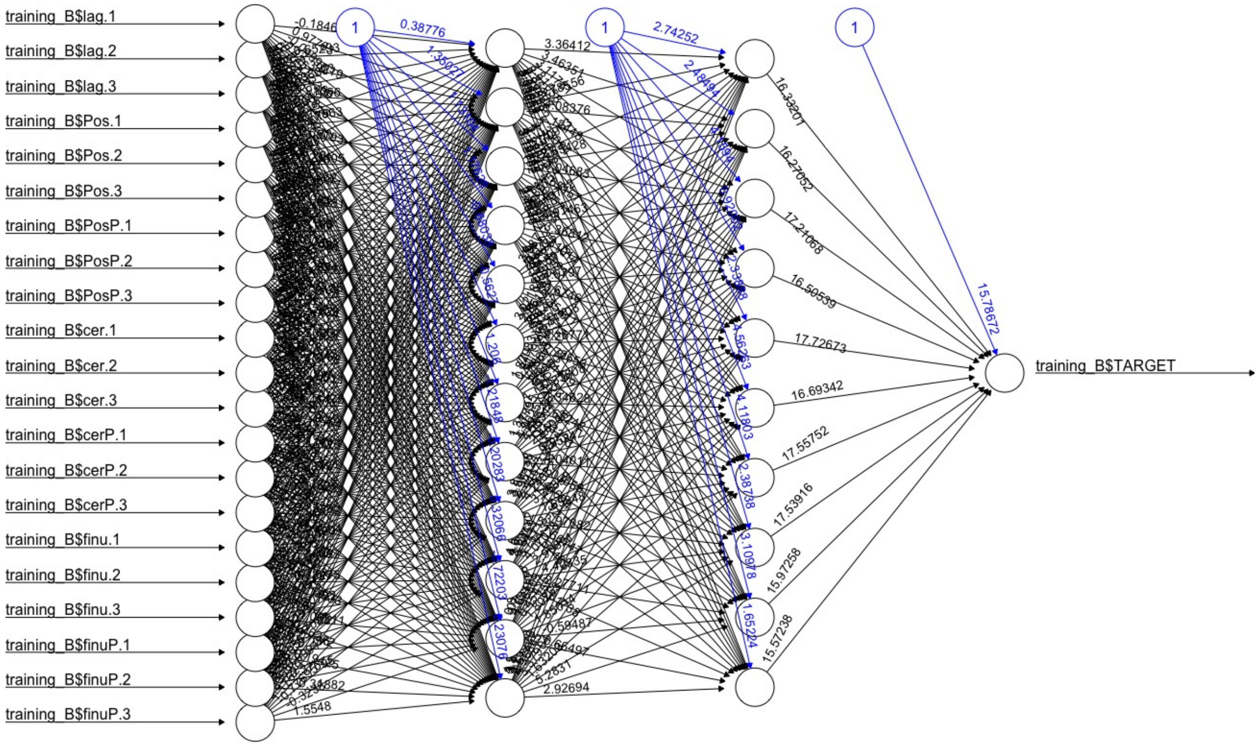


6.2 Epoch Two

Neural Network Plot - Lags Only



Neural Network Plot - Bullish Sentiment Included



Neural Network Plot - Bearish Sentiment Included