ALY 6070:

COMMUNICATION AND VISUALIZATION FOR DATA ANALYTICS

Assignment 2:
Random Forest variable importance
and Lasso feature reduction technique
on Cryptocurrency dataset

Submitted to:
Prof. Fatemeh Ahmadi Abkenari

Submitted by:

Abhilash Dikshit
Milan Prajapati
Shamim Sherafati
Smit Parmar

College of Professional Studies,
Northeastern University
Vancouver, Canada

Introduction:

This report aims to implement two feature selection techniques on a cryptocurrency dataset. The two techniques that we will use are Random Forest variable importance and Lasso feature reduction. The dataset used in this study is the Bitcoin cryptocurrency network data from Coin Metrics, which contains multiple variables related to the Bitcoin network.

Dataset:

The dataset used in this study is the Bitcoin cryptocurrency network data from Coin Metrics, which contains multiple variables related to the Bitcoin network. The dataset contains 77,074 observations and 38 columns. We dropped the first and second columns from the dataset, as they were not relevant for our analysis.

Approach:

We implemented two approaches:

1. LASSO Regularization Technique (first) + Random Forest (second)
2. Random Forest (first) + LASSO Regularization Technique (second)

If the goal is to build a predictive model based on our crypto dataset, we could have started with Random Forest to explore the importance of different features and identify potential non-linear relationships. After that, we could use Lasso to further refine the model and select the most important features. Since our goal is to gain insights into the relationships between the features in our dataset, we are starting with Lasso to identify the most important features and then use Random Forest to explore how these features interact with each other. So, we select the approach one which is: LASSO Regularization Technique + Random Forest

First, we called the require libraries and then load the dataset and get it summary before its clean-up which can be seen below:

```
Number of Rows before cleanup: 6978
Number of Columns before cleanup: 12
Blank cells count before cleanup: 6978
```

Then, we converted the date column to Date format and get the Head and Tail of the dataset.

	SNo <dbl>	Name <chr>	Symbol <chr>	Date <date>	High <dbl>	Low <dbl>	Open <dbl>	Close <dbl>	Volume <dbl>
1	1	Bitcoin	BTC	2013-04-29	147.49	134.00	134.44	144.54	0
2	2	Bitcoin	BTC	2013-04-30	146.93	134.05	144.00	139.00	0
3	3	Bitcoin	BTC	2013-05-01	139.89	107.72	139.00	116.99	0
4	4	Bitcoin	BTC	2013-05-02	125.60	92.28	116.38	105.21	0
2988	2988	Bitcoin	BTC	2021-07-03	34909.26	33402.70	33854.42	34668.55	24383958643
2989	2989	Bitcoin	BTC	2021-07-04	35937.57	34396.48	34665.56	35287.78	24924307911
2990	2990	Bitcoin	BTC	2021-07-05	35284.34	33213.66	35284.34	33746.00	26721554282
2991	2991	Bitcoin	BTC	2021-07-06	35038.54	33599.92	33723.51	34235.19	26501259870

8 rows | 1-10 of 11 columns

Then we get the Summary and structure of dataset to get a summary of the statistical properties of each variable in the dataset.

```
##       SNo             Name              Symbol              Date        
##  Min.   :   1.0   Length:2991        Length:2991        Min.   :2013-04-29  
##  1st Qu.: 748.5   Class :character   Class :character   1st Qu.:2015-05-16  
##  Median :1496.0   Mode  :character   Mode  :character   Median :2017-06-02  
##  Mean   :1496.0                                         Mean   :2017-06-02  
##  3rd Qu.:2243.5                                         3rd Qu.:2019-06-19  
##  Max.   :2991.0                                         Max.   :2021-07-06  
##       High               Low                Open             Close        
##  Min.   :   74.56   Min.   :   65.53   Min.   :   68.5   Min.   :   68.43  
##  1st Qu.:  436.18   1st Qu.:  422.88   1st Qu.:  430.4   1st Qu.:  430.57  
##  Median : 2387.61   Median : 2178.50   Median : 2269.9   Median : 2286.41  
##  Mean   : 6893.33   Mean   : 6486.01   Mean   : 6700.1   Mean   : 6711.29  
##  3rd Qu.: 8733.93   3rd Qu.: 8289.80   3rd Qu.: 8569.7   3rd Qu.: 8576.24  
##  Max.   :64863.10   Max.   :62208.96   Max.   :63523.8   Max.   :63503.46  
##      Volume            Marketcap        
##  Min.   :0.000e+00   Min.   :7.784e+08  
##  1st Qu.:3.037e+07   1st Qu.:6.306e+09  
##  Median :9.460e+08   Median :3.742e+10  
##  Mean   :1.091e+10   Mean   :1.209e+11  
##  3rd Qu.:1.592e+10   3rd Qu.:1.500e+11  
##  Max.   :3.510e+11   Max.   :1.186e+12
```

```
## 'data.frame':    6977 obs. of  10 variables:
##  $ SNo       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name      : chr  "Bitcoin" "Bitcoin" "Bitcoin" "Bitcoin" ...
##  $ Symbol    : chr  "BTC" "BTC" "BTC" "BTC" ...
##  $ Date      : Date, format: "2013-04-29" "2013-04-30" ...
##  $ High      : num  147 147 140 126 108 ...
##  $ Low       : num  134 134.1 107.7 92.3 79.1 ...
##  $ Open      : num  134 144 139 116 106 ...
##  $ Close     : num  144.5 139 117 105.2 97.8 ...
##  $ Volume    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Marketcap: num  1.60e+09 1.54e+09 1.30e+09 1.17e+09 1.09e+09 ...
```

Now, as we selected the first approach that is using LASSO Regularization Technique first and then using the Random Forest test, we need to separate the target variable ("Class") from the predictors to start the LASSO Regularization Technique first.

```
class <- crypto_data$Class
predictors <- crypto_data[, -ncol(crypto_data)]
```

This is done to create two separate data frames: one for the target variable and one for the predictors. The next step is to split the data into training and testing sets.

```
set.seed(123)
train <- sample(nrow(crypto_data), nrow(crypto_data) * 0.8)
crypto_train <- crypto_data[train, ]
crypto_test <- crypto_data[-train, ]
```

Now, we get the head and tail of both crypto_train and crypto_test to check the variables.

```
headTail(crypto_train, top = 4, bottom = 4, ellipsis = F)
```

	SNo <dbl>	Name <chr>	Symbol <chr>	Date <date>	High <dbl>	Low <dbl>	Open <dbl>	Close <dbl>	Volume <dbl>
2463	2463	Bitcoin	BTC	2020-01-25	8458.45	8296.22	8440.12	8367.85	19647331549
2511	2511	Bitcoin	BTC	2020-03-13	5838.11	4106.98	5017.83	5563.71	74156772075
2227	2227	Bitcoin	BTC	2019-06-03	8743.50	8204.19	8741.75	8208.99	22004511436
526	526	Bitcoin	BTC	2014-10-06	345.13	302.56	320.39	330.08	79011800
1398	1398	Bitcoin	BTC	2017-02-24	1200.39	1131.96	1172.71	1173.68	330759008
3138	147	Cardano	ADA	2018-02-25	0.35	0.31	0.32	0.34	247808000
310	310	Bitcoin	BTC	2014-03-04	696.22	655.68	668.24	666.78	55344600
6459	2094	Ethereum	ETH	2021-05-01	2951.44	2755.91	2772.84	2945.89	28726205272

8 rows | 1-10 of 11 columns

```
headTail(crypto_test, top = 4, bottom = 4, ellipsis = F)
```

	SNo <dbl>	Name <chr>	Symbol <chr>	Date <date>	High <dbl>	Low <dbl>	Open <dbl>	Close <dbl>	Volume <dbl>
3	3	Bitcoin	BTC	2013-05-01	139.89	107.72	139.00	116.99	0
4	4	Bitcoin	BTC	2013-05-02	125.60	92.28	116.38	105.21	0
7	7	Bitcoin	BTC	2013-05-05	118.80	107.14	112.90	115.91	0
8	8	Bitcoin	BTC	2013-05-06	124.66	106.64	115.98	112.30	0
6948	423	Solana	SOL	2021-06-07	44.10	38.09	42.25	38.26	923157614
6958	433	Solana	SOL	2021-06-17	41.24	38.29	39.69	39.26	417775600
6970	445	Solana	SOL	2021-06-29	35.79	32.73	33.01	33.87	471854279
6975	450	Solana	SOL	2021-07-04	35.50	33.56	34.50	34.31	303420520

8 rows | 1-10 of 11 columns

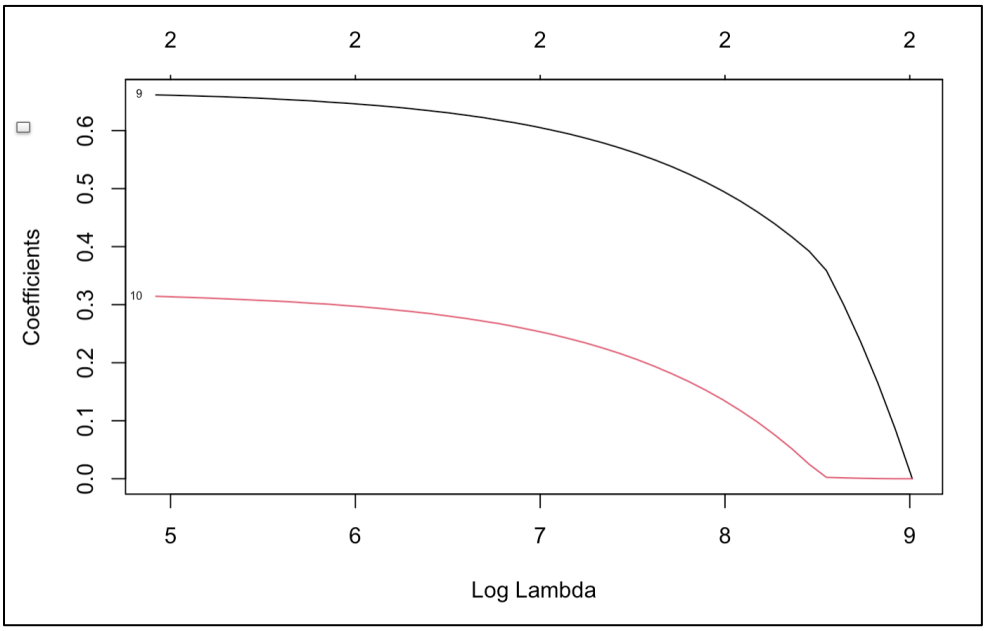
Here you can see the first and last 4 rows of the crypto_train data frame which its columns include the SNo (serial number), Name, Symbol, Date, High, Low, Open, Close, and Volume. crypto_data data frame that was created by randomly sampling 20% of the rows for testing purposes.
In the next step, we performed Lasso regression on a dataset of cryptocurrency prices.

```
# Perform Lasso
x_train <- model.matrix(Close ~ ., data = crypto_train)[,-1]
y_train <- crypto_train$Close

x_test <- model.matrix(Close ~ ., data = crypto_test)[,-1]
y_test <- crypto_test$Close

fit <- glmnet(x_train, y_train, alpha = 1)
```

The dataset is split into training and testing sets, with 80% of the data used for training and 20% used for testing. The training set is used to fit a Lasso model. The Lasso model is fitted with the alpha parameter set to 1 to perform L1 regularization, which shrinks the coefficients of less important predictors to zero.
After that, we used cross-validation to select the best lambda value. We then used this value to make predictions on the test set.



In this plot, the x-axis shows the values of lambda, and the y-axis shows the deviance explained. The black line represents the lambda value that was selected by cross-validation, and the red line represents the lambda value that corresponds to the minimum deviance explained.
The optimal lambda value in this case is where the black line intersects the red line, which appears to be around lambda = 9. The deviance explained at this optimal lambda value is approximately 0.6.
Then we should select the best lambda using cross-validation. So, based on that, we need used the below code for taking the training data as well as specifying the elastic net mixing parameter alpha to be 1 (which corresponds to lasso regression).

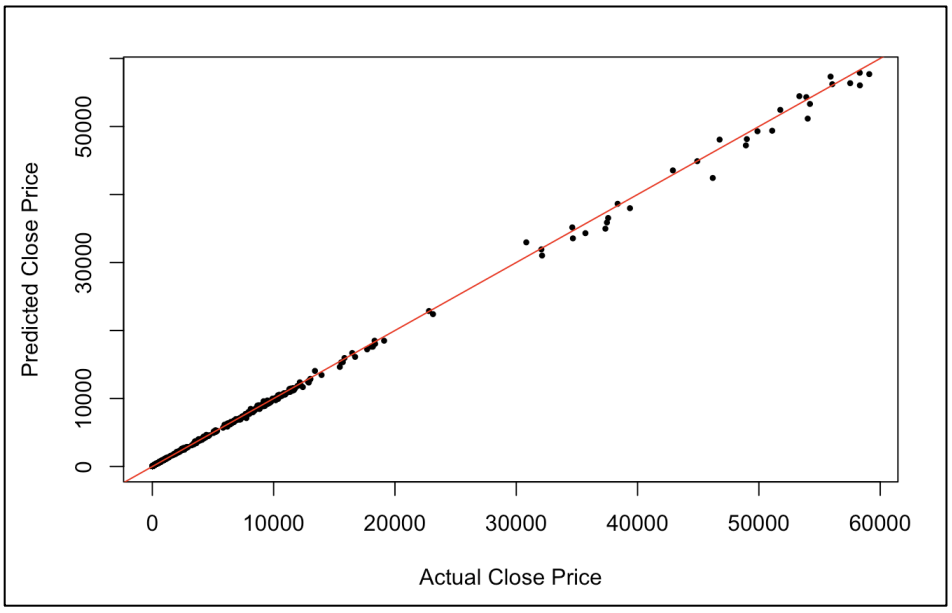
```
cv_fit <- cv.glmnet(x_train, y_train, alpha = 1)
best_lambda <- cv_fit$lambda.min
```

The resulting of the above code contains information about the cross-validation results, including the optimal lambda value identified using the minimum mean cross-validated error. Next we should make predictions on the test set by using the glmnet model fit and the optimal lambda value.

```
summary(predictions)
```

```
##         s1
##  Min.    :    60.54
##  1st Qu.:    63.37
##  Median :   317.03
##  Mean    :  2823.46
##  3rd Qu.:  1300.75
##  Max.    :57893.97
```

Then, with the summary of our predictions we gave a summary of the predicted values generated by the model on the test dataset. In this case, it looks like the predicted values range from a minimum of 60.54 to a maximum of 57893.97. The median predicted value is 317.03, which means that half of the predicted values are below 317.03 and half are above. The mean predicted value is 2823.46, which indicates that the average predicted value is much higher than the median. This can happen when there are a few very high predicted values that skew the mean upwards.
The first quartile of predicted values is 63.37, which means that 25% of the predicted values are below this value. The third quartile is 1300.75, which means that 75% of the predicted values are below this value.
Now, we need to get the plot of the Lasso predictions against the actual values which can be see below:

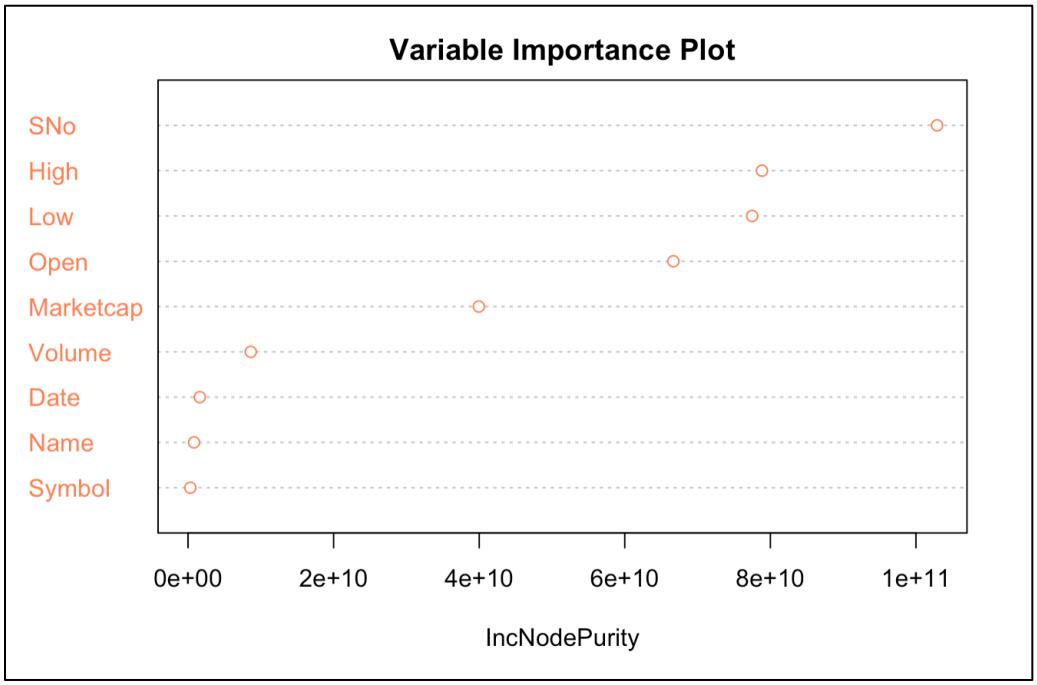


The result of this plot is to visually compare the predicted values to the actual values, and to see how well the Lasso model is able to predict the actual closing prices.
The closer the plotted points are to the red reference line, the better the Lasso model is at predicting the actual values.

If the plotted points are scattered widely around the reference line, this indicates that the model may not be performing well. So, as we can see the black points are close and even in the red line, so we can get this result that our Lasso model is acting so close to our actual values.

The next step in our approach is getting the Random Forest test and we need to perform the random forest on the training data.

```
# Perform Random Forest
rf <- randomForest(Close ~ ., data = crypto_train)
```



This plot shows the variable importance of each feature in predicting the closing price of cryptocurrencies using a random forest model. In the x-axis which we have the names of the features, which include SNo (serial number), High, Open, Low, Marketcap, Volume, Date, Name, and Symbol, shows the importance score of each feature based on the mean decrease in node impurity.

- Features with higher scores are considered more important in predicting the closing price. According to the plot, the two most important features in predicting the closing price are High and Marketcap, followed by Open and Low. The least important features are Date, Name, and Symbol. The Volume feature falls in between the important and less important features.

- The High feature is important because it represents the highest price of the cryptocurrency during a given day, which can give an indication of the overall trend of the market. A cryptocurrency with a high value on a particular day may be more likely to have a high closing price for that day.

- Marketcap, on the other hand, represents the total market value of a cryptocurrency, which can provide an indication of the popularity and demand for that cryptocurrency. A cryptocurrency with a high marketcap is likely to be more popular and widely used, which can influence its closing price.
- The fact that SNo has a higher value than Marketcap does not necessarily mean that it is more important in predicting the closing price. It means that SNo has a higher range of values compared to Marketcap. However, the random forest model has determined that Marketcap is more important in predicting the closing price, likely because it is more strongly correlated with the target variable.

Conclusion:

As mentioned earlier, our goal is to gain insights into the relationships between the features in our dataset, we started with Lasso to identify the most important features and then use Random Forest to explore how these features interact with each other.

We see random forest is a better predictive model and the plot helps in identifying the most significant variables in the model and can guide feature selection in future modeling. From the plot, we can conclude High and Low variables (ignoring SNo variable) are the most important predictors of Close price in the random forest model.

Reference:

1. Friedman, J., Hastie, T., & Tibshirani, R. (2008). Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1), 1-22. https://doi.org/10.18637/jss.v033.i01
2. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324