Yes Bank
Stock Closing Price Prediction

Rudrajit Bhattacharyya
Cohort Zanskar
AlmaBetter

Abstract:

Yes Bank is a well-known bank in the Indian financial domain, headquartered in Mumbai, India and was founded by Rana Kapoor and Ashok Kapur in 2004. Since 2018, it has been in the news because of the fraud case involving Rana Kapoor. Owing to this fact, it was interesting to see how that impacted the stock prices of the company and whether any predictive models can do justice to such situations. The task here is to model the data efficiently and accurately predict the stock's closing prices using regression models. We will be using a few different regression models to predict the stock's closing price and compare their results or performances with each other. We will be using different evaluation metrics such as MSE, RMSE, R-Squared and Adjusted R-Squared to compare the different regression techniques and see which model performs better or predicts more accurately.

Problem Statement:

Yes Bank is a well-known bank in the Indian financial domain. Since 2018, it has been in the news because of the fraud case involving Rana Kapoor. Owing to this fact, it was interesting to see how that impacted the stock prices of the company and whether time series models or any other predictive models can do justice to such situations.

The dataset has monthly stock prices of the bank since its inception and includes closing, starting, highest, and lowest stock prices of every month. The main objective is to predict the stock's closing price of the month.

Data Description:

The dataset contains 185 records and 5 columns which consists of,

- **Date**: It contains the month and the year information which is of object data type. We will need to convert it into a datetime object and in a suitable date format.
- **Open**: It contains the information about the opening price on a particular month and year. Opening price is the price at which a security/share first trades upon the opening of an exchange on a trading day.
- **High**: It contains the information about the highest price at which a stock traded on a particular month and year.
- **Low**: It contains the information about the lowest price at which a stock traded on a particular month and year.
- **Close**: It contains the information about the closing price on a particular month and year. Closing price is the price at which a security/share last trades before the market officially closes for trading on a trading day.

Introduction:

Yes Bank is an Indian bank headquartered in Mumbai, India and was founded by Rana Kapoor and Ashok Kapur in 2004. Since 2018, it has been in the news because of the fraud case involving Rana Kapoor. On 5 March 2020, in an attempt to avoid the collapse of the bank, which had an excessive amount of bad loans, the RBI took control of it. RBI later reconstructed the board and named Prashant Kumar, former CFO and deputy managing director of SBI, as MD and CEO of Yes Bank, along with Sunil Mehta, former non-executive chairman of PNB, as Yes Bank's non-executive chairman. The bank's management under the new leadership, immediately repositioned itself and dealt with all internal and market related challenges to restore customer and depositor confidence. Under the coordinated efforts of the new board and management, Mehta assured shareholders of speedy recovery, even as the RBI, SBI, HDFC Bank, ICICI Bank, Axis Bank and other banks lent it support through a historic Yes Bank Reconstruction Scheme 2020.

Stock price analysis has been a critical area of research and is one of the top applications of machine learning. A stock market is a public market where we can buy or sell shares for publicly listed companies. The stocks, also known as equities, represent ownership in

the company. The stock exchange is the mediator that allows the buying and selling of shares. The stock market is known for being volatile, dynamic and nonlinear. Accurate stock price prediction is extremely challenging because of multiple factors such as politics, global economic conditions, unexpected events, a company's financial performance and so on. Stock price prediction using machine learning helps us discover the future value of the stock and the entire idea of predicting stock prices is to gain significant profits.

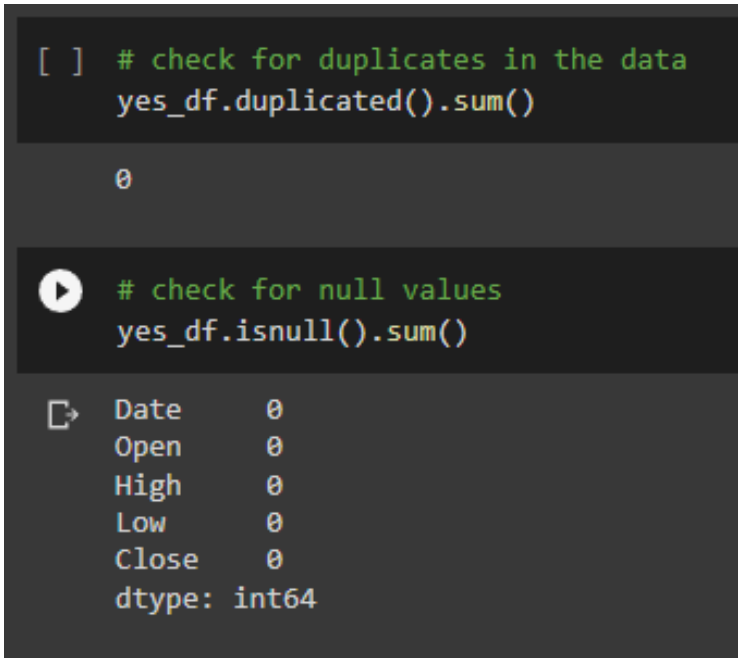
Technical analysis analyzes measurable data from stock market activities, such as stock prices, historical returns etc that could identify trading signals and capture the movement patterns of the stock market. It focuses on historical data and current data just like fundamental analysis, but is mainly used for short term trading purposes.

The dataset we have in our hand is relatively simpler, cleaner and our goal here is to predict the stock's closing price of the month using regression techniques.

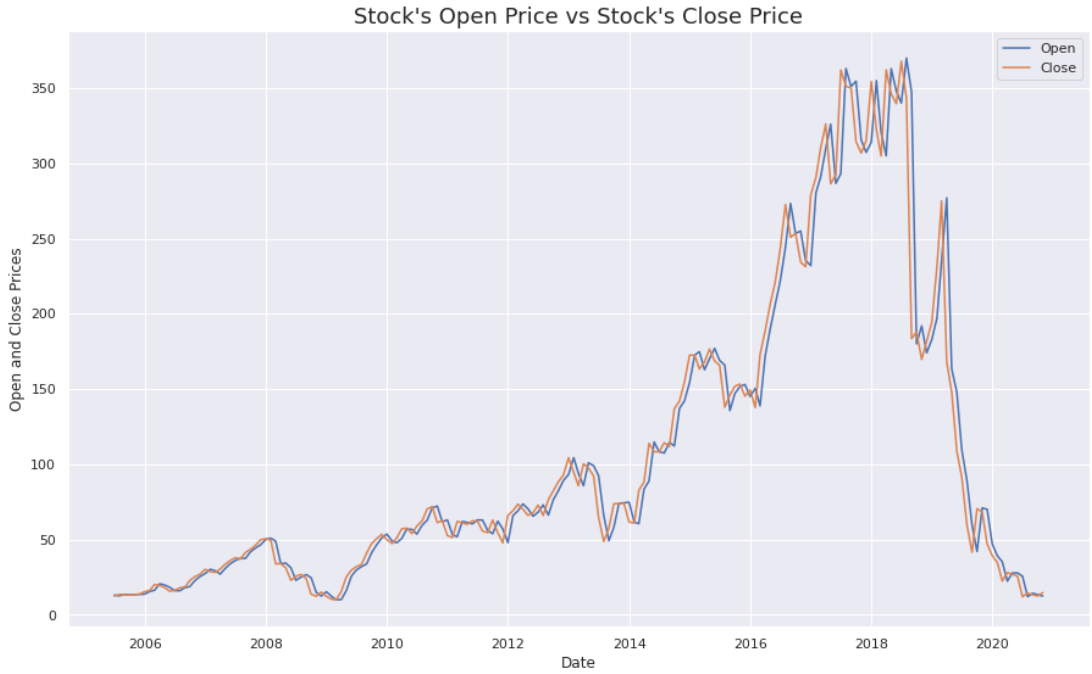
Exploratory Data Analysis:

- **Data Cleaning**:

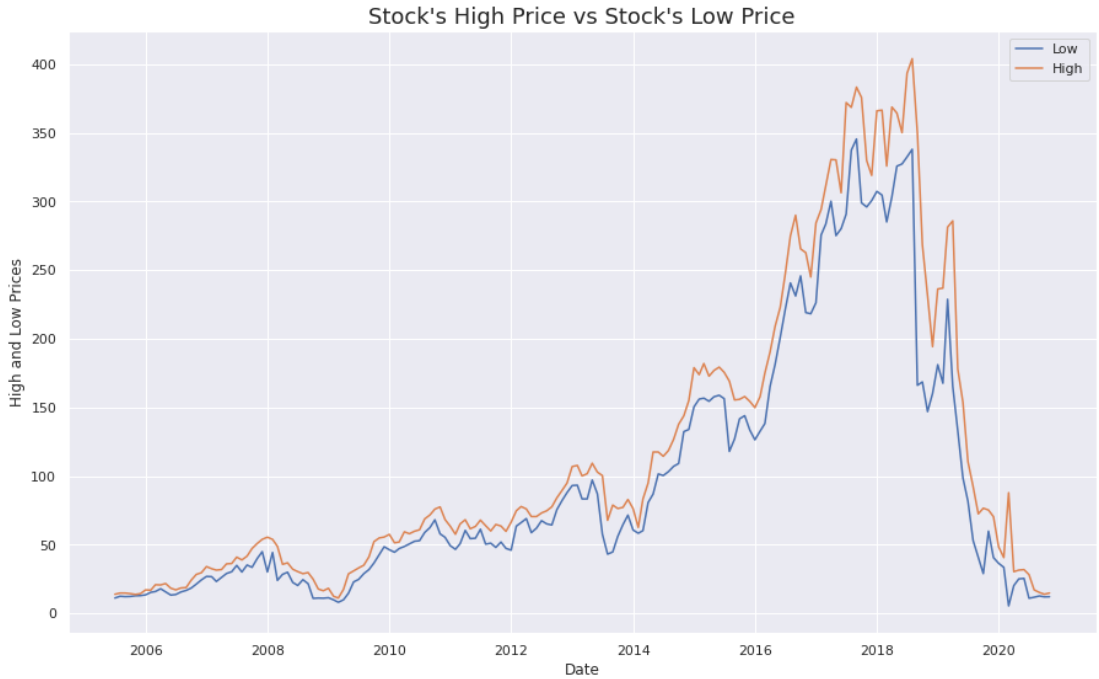
  Data cleaning is one of the most time consuming aspects of data analysis. Formatting issues, missing values, duplicated rows, spelling mistakes, and so on could all be present in a dataset. These difficulties make data analysis difficult, resulting in inappropriate results. Thankfully our dataset was mostly a cleaned one with no duplicated records or null values present. We had to convert the 'Date' column into a datetime object and into a suitable format like 'YYYY-MM-DD'. The 'Date' column was dropped from the dataframe and set as an index before modeling.



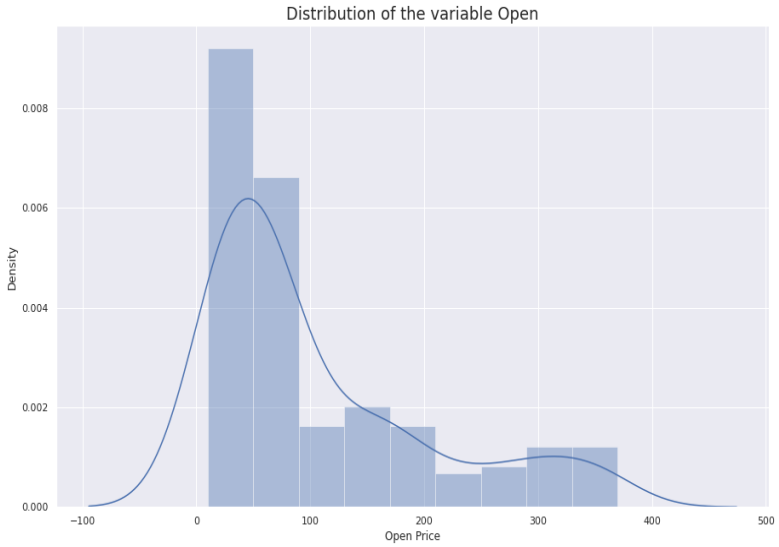
- **Data Visualization/Exploration**:
    - We found out that there isn't much difference between the stock's opening price and the closing price on any particular day/month of the year. The best time to sell the shares was between 2016 to 2018 as the stock prices grew a lot between that period and fell off sharply as soon as the Rana Kapoor incident came into the news.

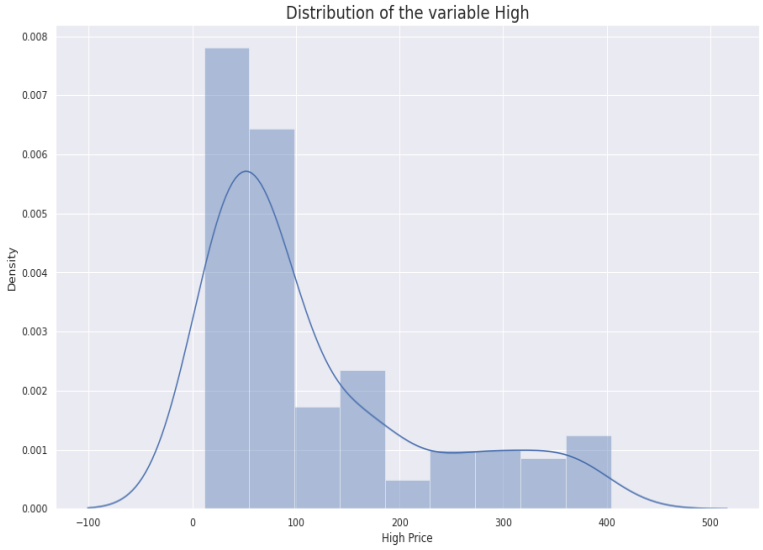


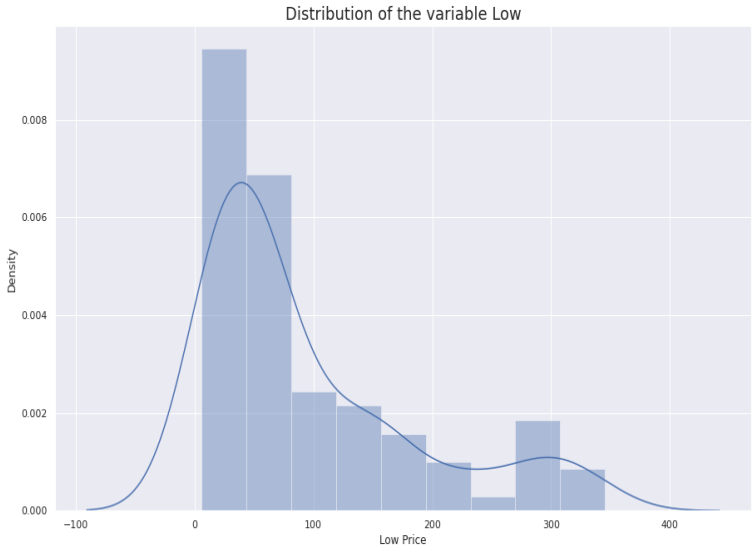
- It touched an all time high of Rs 404 during Aug 2018 and fell off sharply after that to touch an all time low of Rs 5.55 during Mar 2020. It wasn't a very volatile stock before 2018 but after the Rana Kapoor fraud case happened, it is better to stay away from these kinds of stocks while investing.



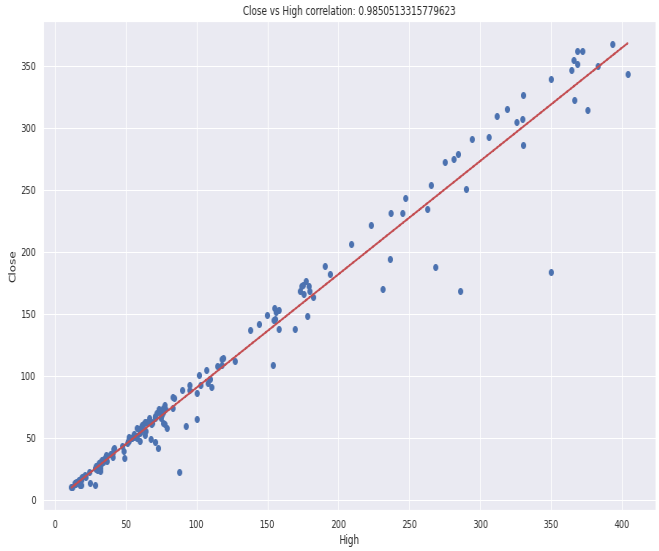
- We found out that the distribution of all the variables, independent and dependent, are right or positively skewed. This can be made normal by applying log transformation.

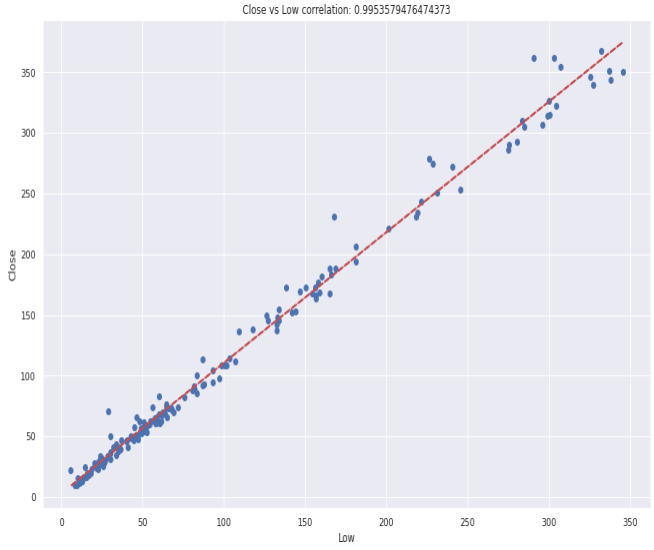


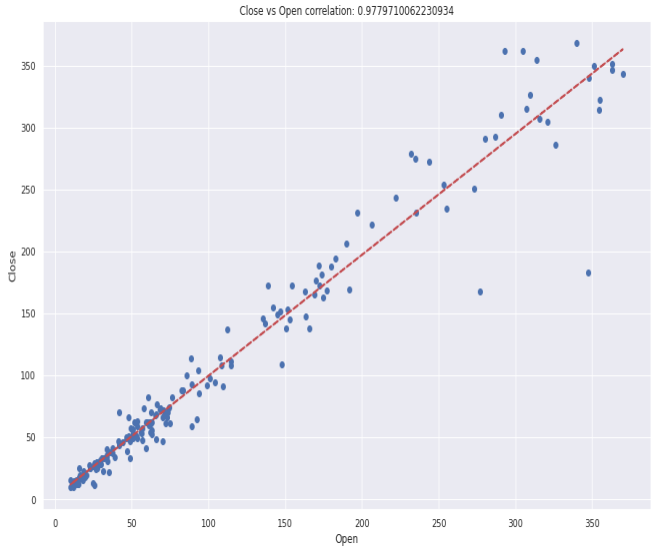




- While doing bivariate analysis we found out that, all the independent variables follow a linear relationship with the dependent variable and have a high positive correlation score. This is an indication that linear regression might perform very well on this dataset.


Close vs High correlation: 0.9850513315779623
Close
High
0
50
100
150
200
250
300
350
400


Close vs Low correlation: 0.9953579476474373
Close
Low
0
50
100
150
200
250
300
350

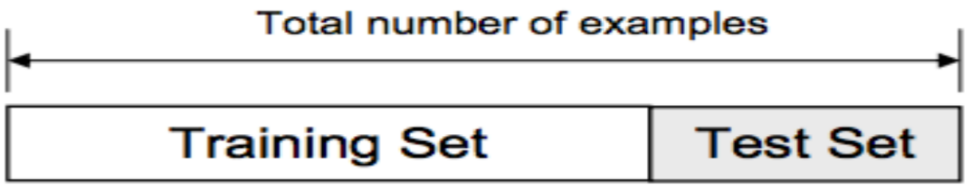

Close vs Open correlation: 0.9779710062230934
Close
Open
0
50
100
150
200
250
300
350

Data Preparation:

Before diving into modeling the data and predicting the stock's closing price, we will be doing the following steps:

- **Train-Test Split**:

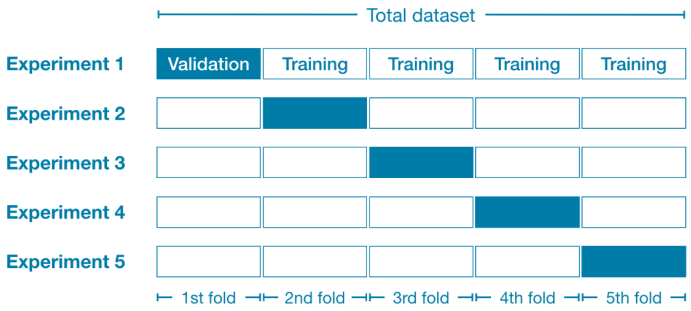
  The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which will allow us to compare the performance of machine learning algorithms for our prediction. Although it is simple to use and interpret, there are times when this procedure should not be used, such as when we have a small dataset like in our case. The train-test split procedure is appropriate when we have a large dataset, a costly model to train, or require a good estimate of model performance quickly. We have chosen to use a 70:30 split for implementing linear regression with our dataset.



  The train-test procedure is not appropriate when the dataset is small. The reason is that when the dataset is split into train and test sets, there will not be enough data in the train set to train the model well and there will not be enough data in the test set to effectively evaluate the model performance. The estimated performance could be overly optimistic or overly pessimistic. A suitable alternate model evaluation procedure would be the k-fold cross validation.

  Cross validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As

such the procedure is often called k-fold cross validation. When a specific value for k is chosen, such as k=10 it becomes 10-fold cross validation. Cross validation is primarily used in machine learning to estimate the skill of a machine learning model on unseen data i.e, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used to train the model. It is a popular method because it is simple to understand and generally results in a less biased or less optimistic estimate of the model than a simple train-test split. We have used cross validation to evaluate model performance for all other models except linear regression.



- **Feature Scaling/Normalization**:

  Some machine learning algorithms are sensitive to feature scaling while others are not. Machine learning algorithms like linear regression, logistic regression, neural networks etc, that use gradient descent as an optimization technique require data to be scaled. The presence of feature value X in the formula will affect the step size of the gradient descent. The difference in ranges of features will cause different step sizes for each feature. To ensure that the gradient descent moves smoothly towards the minima and that the steps for gradient descent are updated at the same rate for all the features, we need to scale the data before feeding it to the model. We have chosen normalization as a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. This is also known as Min-Max scaling.

$$X^{'} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Normalization is good to use when we know that our distribution does not follow a Gaussian distribution. We know that our variables don't follow a gaussian distribution by default.

Evaluation Metrics:

After the model is built, if we see that the difference in the values of the predicted and actual data is not much, it is considered to be a good model and can be used to make future predictions. Few of the regression evaluation metrics we have used here are:

- **Mean Squared Error (MSE)**:

  MSE is one of the most preferred metrics for regression tasks. It is simply the average of the squared difference between the target value and the predicted value by the model.

$$MSE = \frac{1}{n}\sum(Actual - Predicted)^2$$

- **Root Mean Squared Error (RMSE):**

  RMSE is the most widely used metric for regression tasks and is the square root of the averaged squared difference between the target value and the predicted value by the model. It is preferred more in some cases because the errors are first squared before averaging which poses a high penalty on large errors.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted - Actual)^2}{N}}$$

- **R-Squared:**

  Coefficient of determination or R^2 is another metric used for evaluating the performance of a regression model. This metric helps us to compare our current model with a constant baseline and tells us how much our model is better. The

constant baseline is chosen by taking the mean of the data and drawing a line at the mean.
R^2 is a scale free score that implies it doesn't matter whether the values are too large or too small, the R^2will always be less than or equal to 1.

$$R^2 = 1 - \frac{MSE(current\ model)}{MSE(baseline)}$$

- **Adjusted R-Squared:**

  Adjusted R^2depicts the same meaning as R^2 but it is an improvement. R^2 suffers from the problem that the scores improve on increasing terms even though the model is not improving which may misguide the researcher. Adjusted R^2 is always lower than R^2 as it adjusts for the increasing predictors and only shows improvement if there is a real improvement.

$$Adj\ R^2 = 1 - [(\frac{n-1}{n-k-1}) \times (1 - R^2)]$$

Data Modeling:

We have used several regression models to predict the stock's closing price and compare their performances. Let us see each model in depth:

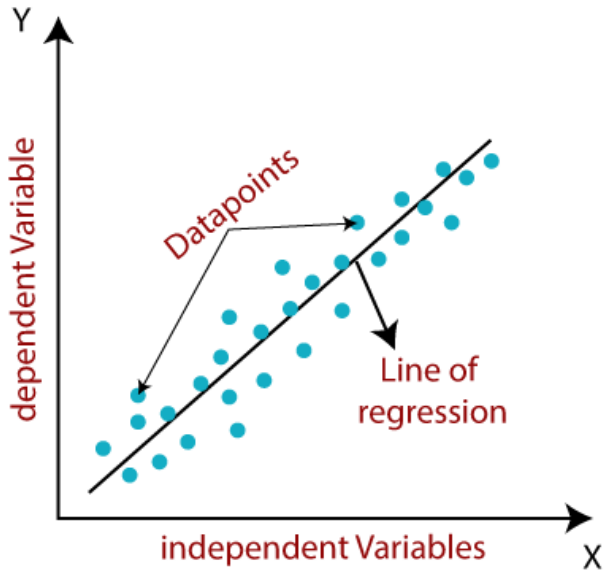
- **Linear Regression:**

  Linear regression is a linear model, e.g a model that assumes a linear relationship between the input variables (x) and the output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). When there is a single input variable, the method is referred to as simple linear regression. When there are multiple input variables, it is referred to as multiple linear regression.

The representation is a linear equation that combines a specific set of input values (x), the solution to which is the predicted output for that set of input values (y). As such, both the input and output values are numeric. The linear equation assigns one scale factor to each input value, called a coefficient and represented by Greek letter β. One additional coefficient is also added, giving the line an additional degree of freedom and is often called the intercept or the bias coefficient.

$$y = \beta_0 + \beta_1 x$$

In higher dimensions when we have more than one input variable (x), the line is called a plane or a hyper-plane.



The results obtained by implementing linear regression to predict stock's closing price is:

MSE	RMSE	R-Squared	Adjusted R-Squared
62.378016	7.897	0.994	0.994

- **Lasso Regression:**

  Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point like the mean. The lasso procedure encourages simple, sparse models. This particular type of regression is well suited for models showing high levels of multicollinearity or when we want to automate certain parts of feature selection/elimination. Lasso stands for Least Absolute Shrinkage and Selection Operator.

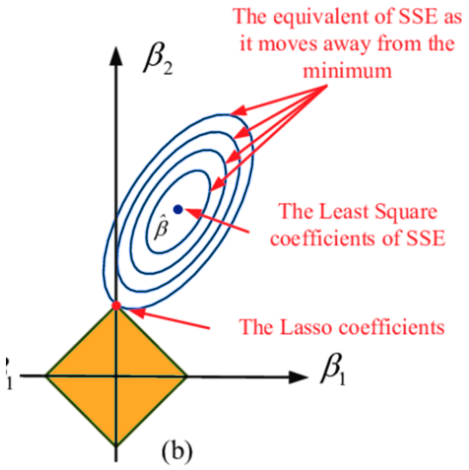
  Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients, some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is ideal for producing simpler models. On the other hand L2 regularization doesn't result in elimination of coefficients or sparse models. This makes the lasso far easier to interpret than ridge.

$$\sum_{i=1}^{n}(y_i - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p}\left|\beta_j\right|$$

  A tuning parameter λ controls the strength of the L1 penalty. λ is basically the amount of shrinkage.

  - When $\lambda = 0$, no parameters are eliminated. The estimate is equal to the one found with linear regression.
  - As λ increases, more and more coefficients are set to zero and eliminated.
  - As λ increases, bias increases.
  - As λ decreases, variance increases.

  If an intercept is included in the model, it is usually left unchanged.



The results obtained by implementing lasso regression with cross validation is:

MSE	RMSE	R-Squared	Adjusted R-Squared
62.378020	7.897	0.994	0.994

As we can observe that the estimate achieved by lasso is the same as we got in linear regression which implies that no penalty was included in the model. With $\lambda = 0$, it behaves the same as linear regression which gave us the best result till now.

- **Ridge Regression:**

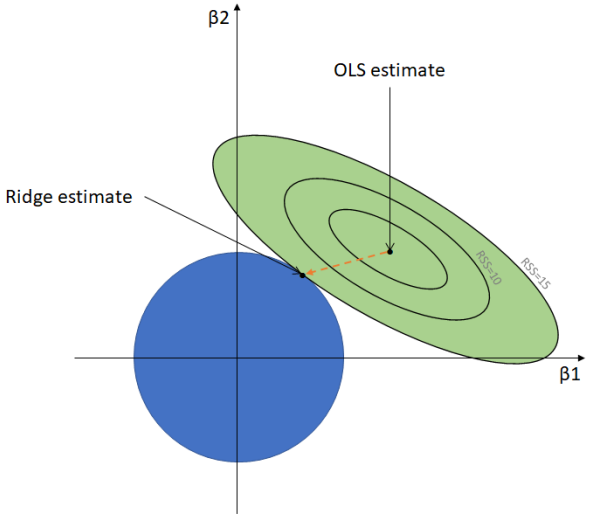
  Ridge regression is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the

issue of multicollinearity occurs, least-squares are unbiased and variances are large, which results in predicted values being far away from the actual values. The cost function for ridge regression is:

$$\sum_{i=1}^{n}(y_i - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right|^2$$

By changing the values of λ, we are controlling the penalty term. The higher the values of λ, the bigger the penalty and therefore the magnitude of coefficients is reduced.

It shrinks the parameters and therefore is used to prevent multicollinearity. It reduces the model complexity by coefficient shrinkage.



The results obtained by implementing ridge regression with cross validation is:

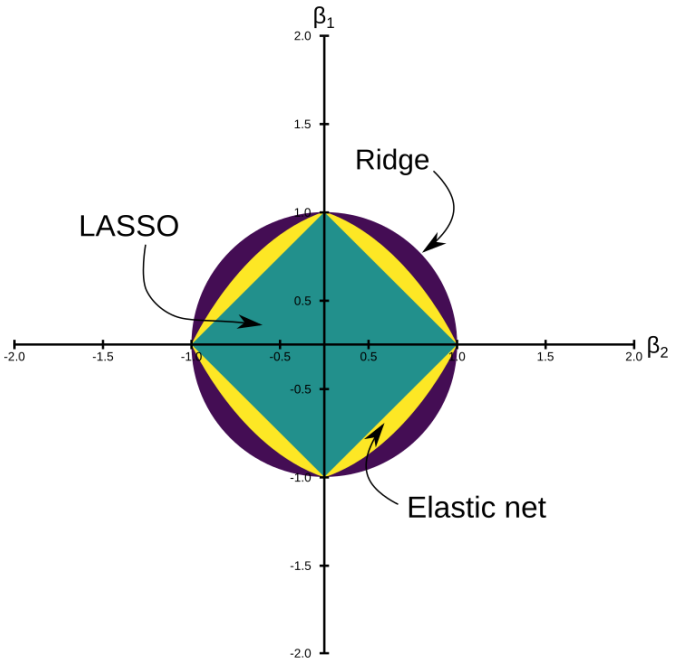
MSE	RMSE	R-Squared	Adjusted R-Squared
67.591	8.221	0.993	0.993

- **Elastic Net Regression:**

  Elastic net regression uses the penalties from both the lasso and ridge regularizations to regularize regression models. The technique combines both the lasso and ridge regression methods by learning from their shortcomings to improve the regularization of statistical models.

  The elastic net method improves lasso's limitations i.e, where lasso takes a few samples for high dimensional data. The elastic net procedure provides the inclusion of 'n' number of variables until saturation. If the variables are highly correlated groups, lasso tends to choose one variable from such groups and ignore the rest.
  To eliminate the limitations found in lasso, the elastic net includes a quadratic expression in the penalty, which when used in isolation becomes ridge regression. The quadratic expression in the penalty elevates the loss function towards being convex. The elastic net draws on the best of both worlds.

  In the procedure for finding the elastic net method's estimator, two stages involve both lasso and ridge regression techniques. It first finds the ridge regression coefficients and then conducts the second step by using a lasso sort of shrinkage of the coefficients. This method subjects the coefficients to two types of shrinkages which causes low efficiency in predictability and high bias. To correct such effects, the coefficients are rescaled by $(1 + \lambda_2)$.



The results obtained by implementing Elastic Net regression with cross validation is:

MSE	RMSE	R-Squared	Adjusted R-Squared
67.110	8.192	0.993	0.993

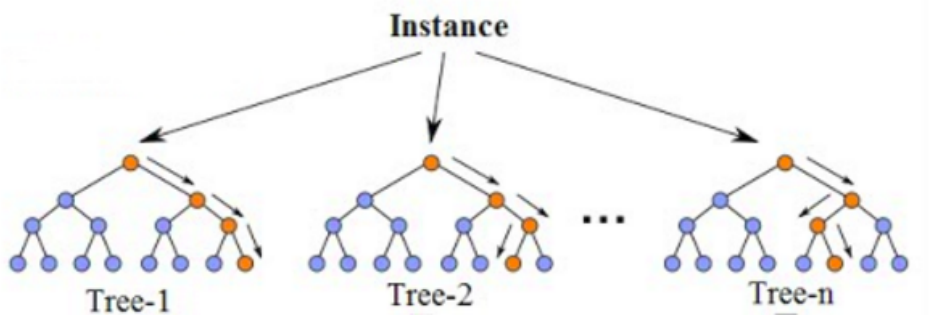
- **Random Forest Regression:**

  Random forest is an ensemble of decision trees. Decision trees are great for obtaining non-linear relationships between input features and the target variable. The inner workings of a decision tree can be thought of as a bunch of if-else conditions.

It starts at the very top with one node which then splits into a left and right node known as decision nodes. These nodes then split into their respective left and right nodes. At the end of the leaf node, the average of the observation that occurs within that area is computed. The most bottom nodes are referred to as leaves or terminal nodes.

Many trees constructed in a certain random way form a Random Forest.

- Each tree is created from a different sample of rows and at each node, a different sample of features is selected for splitting.
- Each of the trees makes its own individual prediction.
- These predictions are then averaged to produce a single result.



The averaging makes a random forest better than a single decision tree hence improves its accuracy and reduces overfitting. A prediction from the random forest regressor is an average of the predictions produced by the trees in the forest.

The results obtained by implementing Random Forest Regressor with cross validation is:

MSE	RMSE	R-Squared	Adjusted R-Squared
180.358	13.429	0.983	0.982

- **Gradient Boosting Regression:**

  Gradient boosting is one of the most powerful techniques for building predictive models. The idea of boosting came out of the idea of whether a weak learner can be modified to become better. Gradient boosting involves three elements:

  - A loss function to be optimized: The loss function depends on the type of problem being solved. For e.g, regression may use a squared error and classification may use logarithmic loss. It must be differentiable.

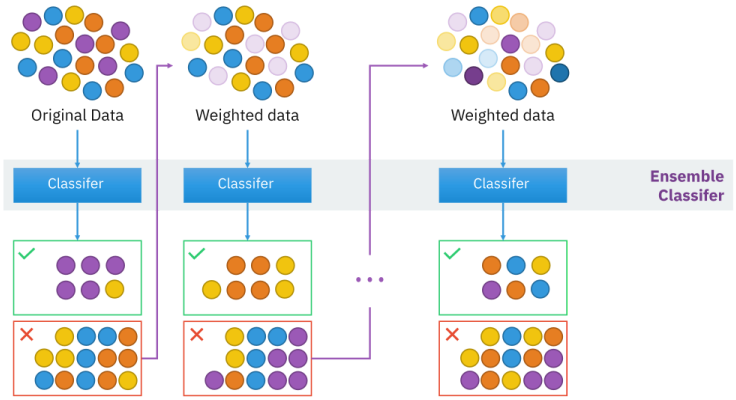
  - A weak learner to make predictions: Decision trees are used as the weak learner in gradient boosting. Specifically regression trees are used that output real values for splits and whose output can be added together, allowing subsequent models outputs to be added and correct the residuals in the predictions. Trees are constructed in a greedy manner, choosing the best split points based on purity scores like Gini or to minimize the loss.

  - An additive model to add weak learners to minimize the loss function: Trees are added one at a time, and existing trees in the model are not changed. A gradient descent procedure is used to minimize the loss when adding trees.

    Traditionally, gradient descent is used to minimize a set of parameters, such as the coefficients in a regression equation or weights in a neural network. After calculating error or loss, the weights are updated to minimize that error.

    Instead of parameters, we have weak learner sub-models or more specifically decision trees. After calculating the loss, to perform gradient descent procedure, we must add a tree to the model that reduces the loss. We do this by parameterizing the tree, then modify the parameters of the tree and move in the right direction by reducing the loss.

    The output for the new tree is then added to the output of the existing sequence of trees in an effort to correct or improve the final output of the model.



The results obtained by implementing Gradient Boosting Regressor with cross validation is:

MSE	RMSE	R-Squared	Adjusted R-Squared
148.650	12.192	0.986	0.985

Model Comparison and Performance:

The final matrix of results after applying all the algorithms:

Models	MSE	RMSE	R-Squared	Adjusted R-Squared
Linear Regression	62.378	7.897	0.994	0.994
Lasso	62.378	7.897	0.994	0.994
Elastic Net	67.110	8.192	0.993	0.993

Ridge	67.591	8.221	0.993	0.993
Gradient Boosting	148.650	12.192	0.986	0.985
Random Forest	180.358	13.429	0.983	0.982

All the algorithms have performed fairly well with the dataset in predicting the stock's closing price as the R-Squared value achieved by all of them is around 98-99% which proves that 98-99% of the variation in the dependent variable can be predicted with the help of the independent variables.

We can clearly observe that linear regression performs the best out of all the algorithms applied and this could be due to the simplicity of the data, cleanliness of the data, and the linear relationship between the independent variables and the dependent variable.

Out of the two ensemble methods, gradient boosting performed better than random forest as boosting tries to correct the residuals of the previous model in its sequence.

As our data had a linear relationship, it was obvious that linear regression will perform better than any other methods.

Challenges:

- It is a very small dataset, thus achieving a good accuracy is difficult. The results could be overly optimistic or overly pessimistic.
- As the independent variables follow a linear relationship with the dependent variable, it is difficult to achieve a good R-squared value for boosting or bagging techniques.

Conclusion:

We have reached the end of our stock price prediction project and have discovered a lot of insights through EDA and have used a few modeling techniques to predict the stock's closing price of the month. All the models have performed fairly well on our dataset.

Let us summarize what we have done so far:

- The dataset had 185 rows and 5 columns consisting of Date, Open, High, Low and a dependent variable Close.
- There were no missing values or duplicated records found in the dataset. The Date column was converted into a datetime object and in a suitable format of 'YYYY-MM-DD'.
- The stock's closing price grew a lot between the period 2016 to 2018 and it was the best time to sell the shares to get good profits.
- There wasn't much fluctuation between the stock's opening price and the closing price through the whole period. It was a stable stock in the market till 2018. As soon as the Rana Kapoor fraud case came into the news, the stock started falling drastically and is trading at the level where it was in its initial days.
- The stock touched an all time high of Rs 404 around the period of Aug 2018 and then started falling drastically as soon as the Rana Kapoor fraud case occurred. The stock then went on to touch an all time low of Rs 5.55 during Mar 2020. It hasn't yet recovered much from those levels till date.
- All the variables in the dataset had a right skewed distribution.
- All the independent variables were linearly related to the dependent variable with a high positive correlation score. This was an indication that the dataset should be a good fit for linear regression.
- We have used 6 regression models to model the data and predict the stock's closing price (Linear Regression, Lasso, Ridge, Elastic Net, Random Forest, Gradient Boosting).
- All the models have performed fairly well as the R-Squared value achieved by them is around 98-99%.
- Linear Regression has performed the best out of all the algorithms applied and this could be due to the simplicity of the data, cleanliness of the data, and the linear relationship between the independent variables and the dependent variable.
- We have used evaluation metrics like MSE, RMSE, R-Squared, and Adjusted R-Squared to evaluate and compare our regression models. Linear regression has given the best results in terms of all the metrics used here.

- Out of the two ensemble methods, gradient boosting has performed better than random forest and this could be due to the way boosting works by giving weightage to the misclassified results of the previous iteration.

References:

- Analytics Vidhya
- Statistics How To
- Machine Learning Mastery
- Neptune AI
- Wikipedia
- Towards Data Science