YES Bank Stock Closing Price Prediction

Karan Tiwari

Abstract

Predicting stock market returns with any degree of accuracy is quite challenging due to the volatility and non-linearity of the financial stock markets.

The advancement of artificial intelligence and increased computing power has increased the accuracy of numerous programme techniques for predicting stock values. Here, I used regression to search for associations. As financial information for variables that act as model inputs, the stock's open, high, low, and close prices are used. Common strategic measures used to evaluate the models include RMSE, and R2. These indicators' low values show how successfully the models predict the closing price of equities.

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. This 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.

Objective

The project's main goal is to do analyze the Data and predict the stock's closing price of the month.

Data Description

- The dataset has 185 rows and 5 columns.

- Dataset does not have any null values
- No duplicate rows

Attribute Information:

- Date: Date of record
- Open: Opening Price
- High: Highest price in the day
- Low: Lowest price in the day
- Close: Closing Price

Challenges

- Data was not normally distributed so we had to fix it.
- We changed the date format.
- Implementation of model

Tools Used

This project is performed on Google collaboratory and Python was used throughout the entire project. The following libraries were imported for data analysis and visualization:.

- Pandas: Extensively used to load and wrangle with the dataset.
- Matplotlib: Used for visualization.
- Seaborn: Used for visualization.
- Numpy: For some math operations in predictions.
- Sklearn: For the purpose of analysis and prediction.

Steps involved

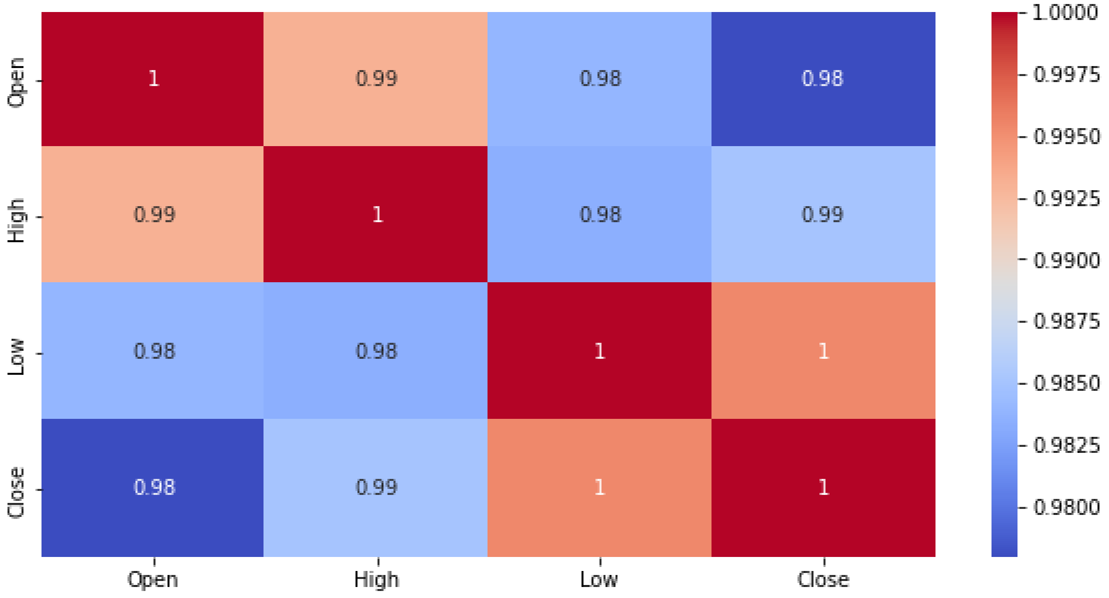
The following steps are involved in the project.

- **Exploratory Data Analysis:** After loading and reading the dataset in notebook, we performed EDA. Exploratory Data Analysis is the crucial process of doing preliminary investigations on data in order to uncover patterns, spot anomalies, test hypotheses, and validate assumptions using summary statistics and graphical representations.

- **Correlation**

  Refers to process for establishing the relationships between two variables called correlation.

We plot the heatmap to find the correlation between all the columns and observed that:



• DATA MODELLING

In this we assign independent and dependent features.

Independent variable: An independent variable is exactly what it sounds like. It is a variable that cannot be influenced by the other elements you are seeking to evaluate.

Dependent variable:: A dependent variable is exactly what it sounds like: something on which other components are reliant.

- **Train test split**

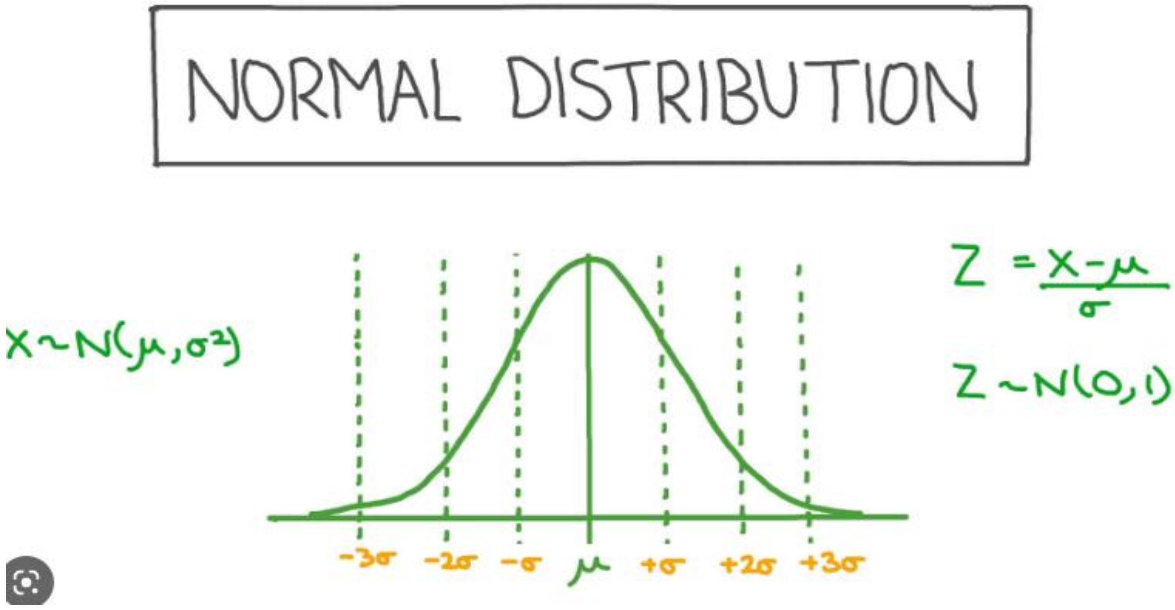
  In train test split we take ‘x’ as dependent variables and ‘y’ take as independent variable then train the model.

- **Data Normalization**

  Our data was rightly skewed so we use standard scaler in x_train and x_test to make data standard normally distributed .

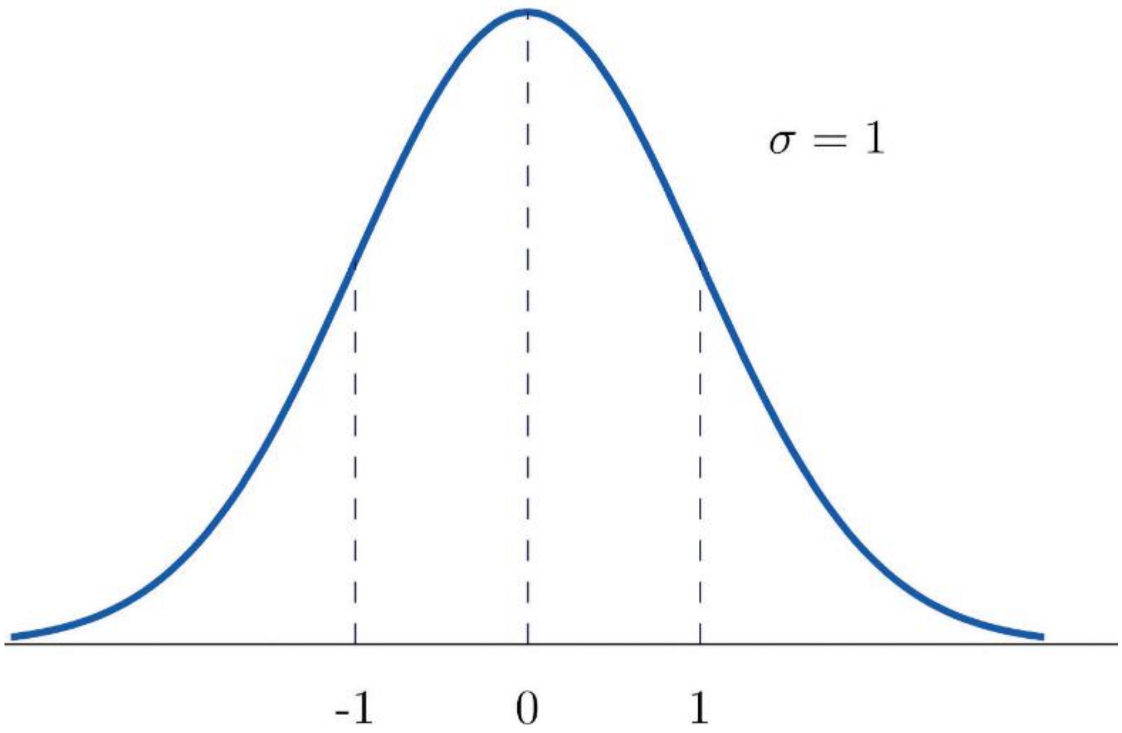
Normal distribution

Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean.



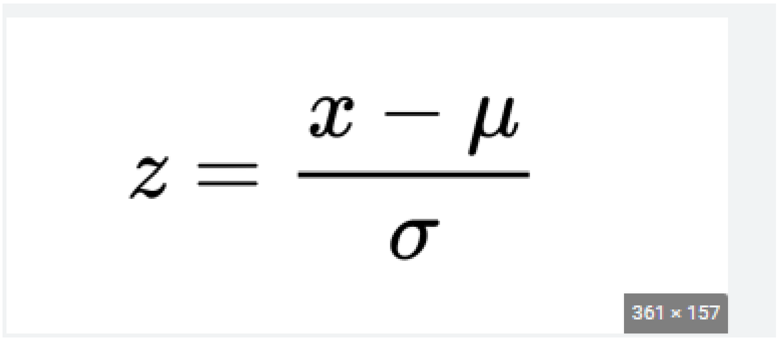
Standard Normal distribution

The standard normal distribution, also called the z-distribution, is a special normal distribution where the mean is 0 and the standard deviation is 1.



We can make data standard normal distributed with standard scaler.

Standard ScalerFormula



$$z = \frac{x - \mu}{\sigma}$$

- **Model Implemented**
    1. **Linear Regression**
    2. **Lasso Regression**
    3. **Ridge Regression**
    4. **Elastic Net Regression**

- **Algorithms**

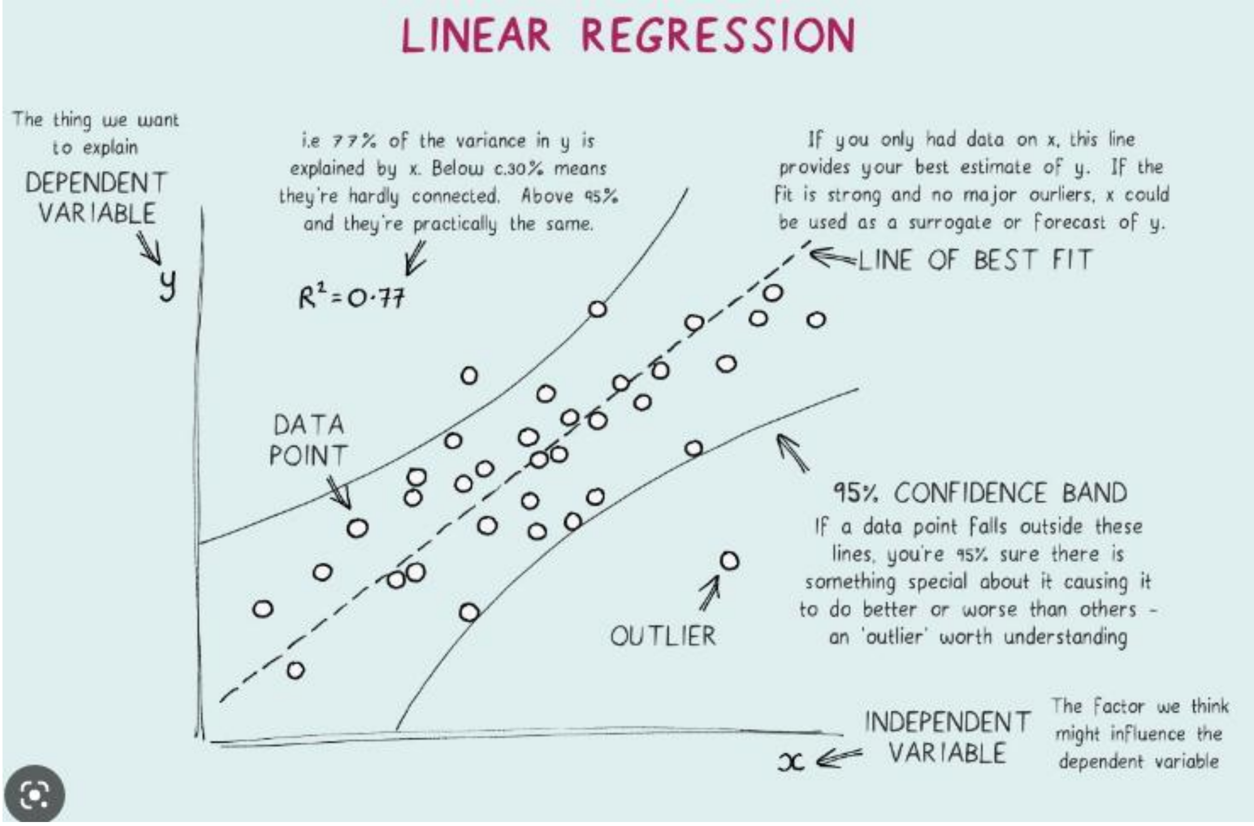
1. Linear Regression

One of the simple and well-liked Machine Learning techniques is linear regression. This statistical technique is employed in predictive analysis. Predictions are made for continuous or numerical variables using linear regression.

A linear connection between a dependent and independent variables is demonstrated using the linear regression procedure.

The linear regression model can be represented by the following equitation.

$$\underset{\text{Dependent Variable}}{Y_i} = \overset{\text{Constant/Intercept}}{\beta_0} + \underset{\text{Slope/Coefficient}}{\beta_1}\,\overset{\text{Independent Variable}}{X_i}$$



The thing we want to explain

i.e 77% of the variance in y is explained by x. Below c.30% means they're hardly connected. Above 95% and they're practically the same.

If you only had data on x, this line provides your best estimate of y. If the fit is strong and no major ourliers, x could be used as a surrogate or forecast of y.

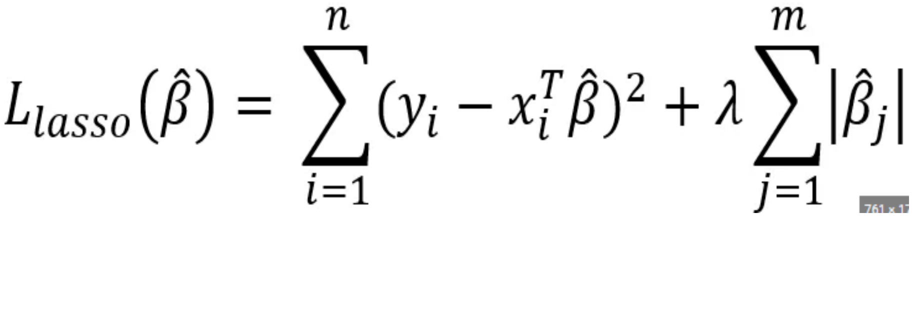
If a data point falls outside these lines, you're 95% sure there is something special about it causing it to do better or worse than others - an 'outlier' worth understanding

The factor we think might influence the dependent variable

2. Lasso Regression

Lasso regression is a type of linear regression that employs the "shrinkage" strategy, in which the coefficients of determination gradually approach zero. Regression coefficients as observed in the dataset are provided via linear regression. With the lasso regression, it is possible to reduce or regularize coefficients to prevent over fitting and improve their performance across various datasets.

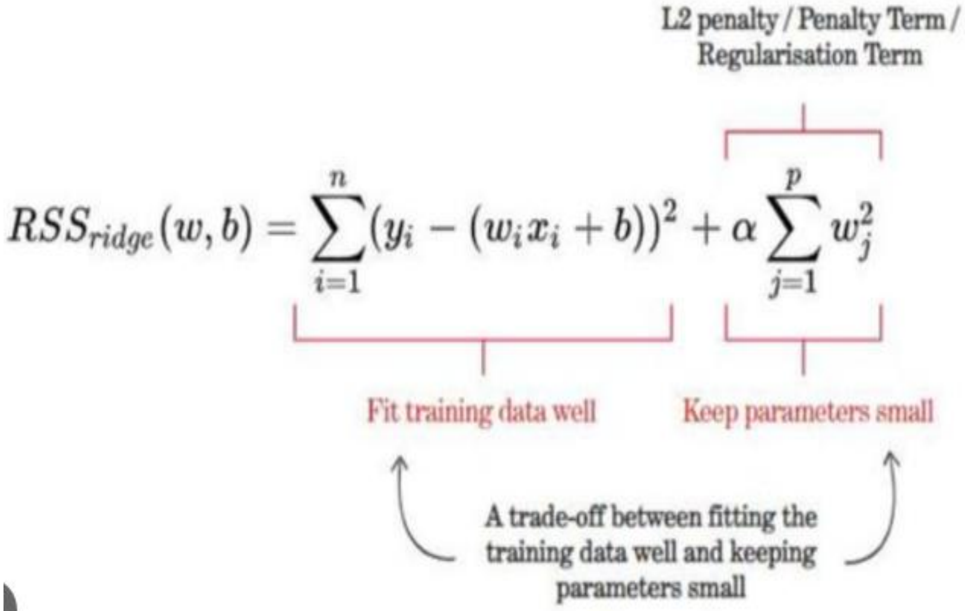
When the dataset exhibits strong multicollinearity or when you wish to automate feature selection and variable exclusion, this sort of regression is utilized.



$$L_{lasso}(\hat{\beta}) = \sum_{i=1}^{n}(y_i - x_i^T\hat{\beta})^2 + \lambda\sum_{j=1}^{m}|\hat{\beta}_j|$$

3. **Ridge Regression**

A regularized variation of linear least squares regression is ridge regression. It operates by reducing the regression model's coefficients or weights until they are zero. By applying a squared penalty to their size, this is accomplished.
This is one regularization technique where the data has multicollinearity problems. The variance is high and the least squares are unbiased in this multicollinearity, which causes a deviation between the predicted value and the actual value.



$$RSS_{ridge}(w,b) = \underbrace{\sum_{i=1}^{n}(y_i - (w_i x_i + b))^2}_{\text{Fit training data well}} + \alpha \overbrace{\underbrace{\sum_{j=1}^{p} w_j^2}_{\text{Keep parameters small}}}^{\text{L2 penalty / Penalty Term / Regularisation Term}}$$

A trade-off between fitting the training data well and keeping parameters small

4. Elastic Net Regression

Elastic Net Regression is the third type Of Regularization technique. It came into Existence due to the limitation of the Lasso regression. Lasso regression Cannot take correct alpha and lambda values as per the requirement of the data. The solution to the problem is to Combine the penalties of both ridge regression and lasso regression.

$$L_{enet}(\hat{\beta}) = \frac{\sum_{i=1}^{n}(y_i - x_i^T\hat{\beta})^2}{2n} + \lambda\left(\frac{1-\alpha}{2}\sum_{j=1}^{m}\hat{\beta}_j^{\ 2} + \alpha\sum_{j=1}^{m}|\hat{\beta}_j|\right)$$

- ## Model Performance

  Model performance can be calculated as:

Mean square Error (MSE)

The mean square is defined as the arithmetic mean of a set of integers or a random variable's squares.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE	= mean squared error
n	= number of data points
Y_i	= observed values
$\hat{Y}_i$	= predicted values

Root mean square Error (RMSE)

It is also known as Root Mean Squared Deviation (RMSD).
Model performance RMSE or is a measure of accuracy, to compare forecasting errors of different models for a particular dataset and not between datasets, as it is scale-dependent. RMSD is always non-negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2}{N}}$$

R-Squared

R-Squared is a statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variable(s) in a regression model.

In investing, R-squared is
generally interpreted as the percentage of a fund or security's movements that can be explained by movements in a benchmark index.

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination

RSS = sum of squares of residuals

TSS = total sum of squares

- **Hyperparameter Tuning:**

Finding the optimal mix of hyper parameters to enhance the model's performance is known as hyperparameter tweaking. It operates by doing several trials within a single training procedure. Every trial entails the full execution of your training application with the values of the selected hyperparameters set within the predetermined bounds. Once complete, this procedure will provide you with the set of hyperparameter values that are ideal for the model to produce the best outcomes.

Grid search CV

Grid Search combines a set of hyperparameters chosen by the scientist and goes through them all to assess the model's performance. It has the benefit of being a straightforward procedure that will go through all of the preset combinations.

- ## Conclusion

  That's all! We had completed our exercise. So far, we have done EDA, data modelling, and model construction, beginning with data loading.In all of these models, our accuracy was 98%.