Yes Bank Stock Closing Price Prediction

Ashfaque sayyed
Data science trainees,
AlmaBetter, Bangalore

Abstract:

The term stock price refers to the current price that a share of stock is trading for on the market. The price of a stock will go up and down in relation to a number of different factors, including changes within the economy as a whole, changes within industries, political events, war, and environmental changes.

Accurate prediction of stock market returns is a very challenging task due to the volatile and non-linear nature of the financial stock markets. With the introduction of machine learning, time series forecasting and increased computational capabilities, programmed methods of prediction have proved to be more efficient in predicting stock prices

1.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.

2. Introduction

To determine the YES bank's stock's future value on the national stock exchange. The advantage of a successful prediction of a stock's future price could result in insignificant profit. In this work, regression and time series forecasting techniques have been utilized for predicting the next day closing price for one company belonging to the Finance sector of operation. The financial data: Open, High, Low and Close prices of stock are used for creating new variables which are used as inputs to the model. The models are evaluated using standard strategic indicators: RMSE and MAPE.

3. Variables Description

Date: It specifies the month and year of particular price.

Open: it specifies the opening price of the stock in that month. (Numeric)

High: it specifies the highest price of stock in that month. (Numeric)

Low: it specifies the lowest price of stock in that month. (Numeric)

Close: it specifies the close price of stock in that month. (Numeric)

4. Steps involved:

Data collection:

Data collection is the process of collecting, measuring and analyzing different types of information using a set of standard validated techniques. The main objective of data collection is to gather information-rich and reliable data, and analyses them to make critical business decisions. Once the data is collected, it goes through a rigorous process of data cleaning and data processing to make this data truly useful for businesses. It refers to the process of finding and loading data into our system. Pandas library is used to loading our data in our system in python. Using pandas we can manipulate data easily.

Data Cleaning:

The next task was data cleaning which was easy with this dataset. Data cleaning refers to the process of removing unwanted variables and values from your dataset and getting rid of any irregularities in it. Such anomalies can disproportionately skew the data and hence adversely affect the results. Some steps that can be done to clean data are:

• Handling missing values: There are always some missing values in dataset. If we don't remove or handle those missing values then that can cause a trouble in our analysis. Removing or replacing those missing values with something meaningful is very important so that our data will have no missing values.

• Removing duplicates: Drop the duplicates rows.

• Formatting data to proper dtype.

• Adding or removing columns required for analysis.

Exploratory Data Analysis:

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling and thereby contrasts traditional hypothesis testing. It is crucial to understand it in depth before you perform data analysis and run your data through an algorithm. You need to know the patterns in your data and determine which variables are important and which do not play a significant role in the output. Further, some variables may have correlations with other variables. You also need to recognize errors in your data.

Model Training:

Model training is the process of fitting a data into machine learning model from which model learns the patterns in data to predict the dependent variable. Model do it so by assigning a weight to each variable. After our model is trained, we test our model on test data to

check how our model is performing. 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. In this project we have used 80% data for training purpose and 20% data for test set. The train-test procedure is appropriate when there is a sufficiently large dataset available.

Standardization of features

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.
The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

Fitting different models

For modelling we tried various regression algorithms like:

1. **Linear Regression**
2. **Lasso Regression**
3. **Ridge Regression**
4. **Random Forest Regression**
5. **XG Boost Regression**

Tuning the hyperparameters for better accuracy

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of linear based models like Lasso and Ridge.

We used Grid Search CV for hyperparameter tuning.

Grid Search CV:

Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

5. Algorithms:

1. Linear Regression:

Linear regression is one of the most basic types of regression in supervised machine learning. The linear regression model consists of a predictor variable and a dependent variable related linearly to each other. We try to find the relationship between independent variable(input) and a corresponding dependent variable (output). This can be expressed in the form of a straight-line

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Linear regression, also known as ordinary least squares (OLS) and

linear least squares, is the real workhorse of the regression world.

.

2. Lasso Linear Regression:

This is a regularization technique used in feature selection using a Shrinkage method also referred to as the penalized regression method. Lasso is short for Least Absolute Shrinkage and Selection Operator, which is used both for regularization and model selection. If a model uses the L1 regularization technique, then it is called lasso regression. In this shrinkage technique, the coefficients determined in the linear model from equation are shrunk towards the central point as the mean by introducing a penalization factor called the alpha α (or sometimes lamda) values.

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

Alpha (α) is the penalty term that denotes the amount of shrinkage (or constraint) that will be implemented in the equation. With alpha set to zero, you will find that this is the equivalent of the linear regression model and a larger value penalizes the optimization function. Therefore, lasso regression shrinks the coefficients and helps to reduce the model complexity and multi-collinearity.
Alpha (α) can be any real-valued number between zero and infinity; the larger the value, the more aggressive the penalization is.

3. Ridge Linear Regression:

Similar to the lasso regression, ridge regression puts a similar constraint on the coefficients by introducing a penalty factor. Ridge regression shrinks coefficients toward zero, but they rarely reach zero. However, while lasso regression takes the magnitude of the coefficients, ridge regression takes
the square.

$$L_{hridge}(\hat{\beta}) = \sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2 + \lambda \sum_{j-1}^{m} w_j\hat{\beta}_j^2.$$

4. Random Forest Regressor:

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree. The basis for the Random Forest is formed by many individual decision trees, the so-called Decision Trees. A tree consists of different decision levels and branches, which are used to classify data. The Decision Tree algorithm tries to divide the training

data into different classes so that the objects within a class are as similar as possible and the objects of different classes are as different as possible.

5. XG Boost:

XG Boost is one of the most popular variants of gradient boosting. It is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XG Boost is basically designed to enhance the performance and speed of a Machine Learning model. In prediction problems involving unstructured data (images, text, etc.), artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree-based algorithms are considered best-in-class right now.

XG Boost uses pre-sorted algorithm & histogram-based algorithm for computing the best split. The histogram-based algorithm splits all the data points for a feature into discrete bins and uses these bins to find the split value of the histogram. Also, in XG Boost, the trees can have a varying number of terminal nodes and left weights of the trees that are calculated with less evidence is shrunk more heavily.

6. Model performance:

Model can be evaluated by various metrics such as:

1. **Train Score:**
   A data with lots of variance then this causes over-fitting. This causes poor result on Test Score. Because the model curved a lot to fit the training data and generalized very poorly. So, generalization is the goal.

2. **Test Score:**
   This is when our model is ready. Before this step we have not touched this data-set. So, this represents real life scenario. Higher the score, better the model generalized.

3. **Mean Score Error:**
   The Mean Squared Error (MSE) is perhaps the simplest and most common loss function, often taught in introductory Machine Learning courses. To calculate the MSE, you take the difference between your model's predictions and the ground truth, square it, and average it out across the whole dataset.
   The MSE will never be negative, since we are always squaring the errors. The MSE is formally defined by the following equation:
   Where N is the number of samples we are testing against.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

4. **Root Mean Square Error:**
   Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data. Formally it is defined as follows:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are predicted values
$y_1, y_2, \ldots, y_n$ are observed values
n is the number of observations

   In data science, RMSE has a double purpose:

   To serve as a heuristic for training models
   To evaluate trained models for usefulness / accuracy.

5. **Mean Absolute Error:**
   The Mean Absolute Error (MAE) is only slightly different in definition from the MSE, but interestingly provides almost exactly opposite properties! To calculate the MAE, you take the difference between your model's predictions and the ground truth, apply the absolute value to that difference, and then average it out across the whole dataset.

   The MAE, like the MSE, will never be negative since in this case we are always taking the absolute value of the errors. The MAE is formally defined by the following equation:

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^{n} |y_j - \hat{y}_j|$$

6. **Mean Absolute Percentage Error:**
   The Mean Absolute Percentage Error (MAPE) is one of the most commonly used KPIs to measure forecast accuracy.

   MAPE is the sum of the individual absolute errors divided by the demand (each period separately). It is the average of the percentage errors.

$$MAPE = \frac{1}{n} \sum \frac{|e_t|}{d_t}$$

   MAPE is a really strange forecast KPI. It is quite well-known among business managers, despite being a poor-accuracy indicator. As you can see in the formula, MAPE divides each error individually by the demand, so it is skewed: high errors during low-demand periods will significantly impact MAPE. Due to this, optimizing MAPE will result in a strange forecast that will most likely undershoot the demand.

7. **R2 Score:**
   R-squared (aka coefficient of determination) measures the variation that is explained by a regression model. R-squared of a regression model is positive if the model's prediction is better than a prediction which is just the mean of the already available 'y' values, otherwise it is negative. Below is the theoretical formula of R-squared.

$$R^2 = \frac{SS_{explained}}{SS_{Total}}$$

not always correct

The above formula is theoretically correct, but only when the R-squared is positive. The formula doesn't return a negative R-squared as we are computing the sum of squares in both the numerator and denominator, which makes them always positive, thereby returning a positive R-squared. We can derive the right formula (the one used in practice and also returns negative R-squared) from the above formula as shown below.

$$SS_{Total} = SS_{residual} + SS_{explained}$$

$$SS_{explained} = SS_{Total} - SS_{residual}$$

$$R^2 = \frac{SS_{explained}}{SS_{Total}}$$

$$R^2 = \frac{SS_{Total} - SS_{residual}}{SS_{Total}}$$

$$\boxed{R^2 = 1 - \frac{SS_{residual}}{SS_{Total}}}$$

Correct formula

8. **Adjusted R2 Score:**
For a multiple regression model, R-squared increases or remains the same as we add new predictors to the model, even if the newly added predictors are independent of the target variable and don't add any value to the predicting power of the model. Adjusted R-squared eliminates this drawback of R-squared. It only increases if the newly added predictor improves the model's predicting power. Adding independent and irrelevant predictors to a regression model results in a decrease in the adjusted R-squared.

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

$where:$

$R^2 = R - squared$

$n = number\ of\ samples/rows\ in\ the\ data\ set$

$p = number\ of\ predictors/features$

8. Conclusion:

- Features are multicollinear but can't drop the column because features are limited.
- High, low, open are directly correlate with the closing price of stocks.
- The test results of all the regression models are evaluated and compared. We checked performance metrics such as R2 score, Mean Scored Error, and Root Mean Score Error etc.
- In linear, lasso and ridge accuracy are approximately equal even after applying cross validation.
- Results are not up to the mark with linear regression, ridge and lasso regression.

- Other models such as random forest and XG boost. With the help of this model, we got better R2 scores and metrics.

- Out of all the model random forest and XG boost gives best result.

Model name	Train score	Test score	MSE	RMSE	MAE	MAPE	R2 score
Linear regression	0.815	0.823	0.032	0.178	0.151	0.095	0.823
Lasso regression	0.815	0.821	0.032	0.179	0.152	0.096	0.821
Ridge regression	0.815	0.822	0.032	0.178	0.151	0.095	0.822
Random forest	0.998	0.992	0.001	0.037	0.029	0.018	0.992
XGboost	0.999	0.991	0.002	0.039	0.030	0.020	0.991

References-

1. stackoverflow.com
2. towardsdatascience.com
3. GeeksforGeeks
4. Analytics Vidhya