Yes Bank Stock Closing Price Prediction

Rajat Chaudhary, Anukriti Shakyawar
Raman Kumar, Deepmala Srivastava

Team: Web Crawlers

Abstract:

Rana Kapoor and Ashok Kapoor launched Yes Bank, an Indian bank with its headquarters in Mumbai, India, in 2004. Through retail banking and asset management services, it provides a wide range of unique solutions for corporate and retail consumers. The Reserve Bank of India (RBI) took control of the bank on March 5 in an effort to prevent its collapse due to an overwhelming volume of bad loans. Following a board restructuring, the RBI appointed Prashant Kumar, a former State Bank of India CFO and deputy managing director, as MD & CEO of Yes Bank. Sunil Mehta, a former non executive chairman of Punjab National Bank, was designated as Yes Bank's nonexecutive chairman.

1. Introduction:

Retail, MSME, and corporate banking are all areas of interest for Yes Bank. YES Securities (India) Limited, YES Trustee Limited, and YES Asset Management (India) Limited are its three subsidiaries. As of September 2018, Yes Bank had borrowed syndicated loans of US$30 million to US$410 million from eight major international organizations, including ADB, OPIC, European investment bank, banks in Taiwan, and Japan. In order to support female entrepreneurs, it also collaborated with Wells Fargo and the US government's OPIC.

Yes Bank provides (UPI) Unified Payments Interface facility to allow customers to easily and securely perform various financial transactions from their mobile devices via third-party app providers like PhonePe and Yuva Pay. According to the data shared by NPCI (National Payments Corporation of India), Yes Bank processed 25.94 million transactions amounting to INR 14811.73 crores through its own UPI app in July 2021.Yes Bank acquired over 24.19% stake in Dish TV, India's largest direct-to-home (DTH) company in terms of subscribers, on 30 May 2020.

2. Data Description:

In this Dataset we have 185 rows and 5 columns namely Date, Open, High, Low, Close.

a. **Date:** *We will use it as an index.*
b. **Open:** *opening price of the stock of a particular day.*
c. **High:** *It's the highest price at which a stock traded during a period.*
d. **Low:** *It's the lowest price at which stock traded during a period.*
e. **Close:** *Closing price of a stock at the end of a trading day.*

3. Analysis Methodology:

Integral research, data cleaning and filtering, data visualization, data transformation, that make up for our three-part strategy. We started by performing some fundamental research on our dataset. While doing this, we found out the basic information regarding our dataset such as columns, data types, shape, info and we also wanted to find in our dataset that if there is some missing values, duplicate values present or not but fortunately we found no duplicate values in our data set, as well as no missing values. Next

we did data visualization. In Data Visualization we plot graphs between independent variables vs date, dependent variables vs date, relation between dependent and independent variables.

3.1. Data Cleaning:

No null, duplicate, or missing values were discovered throughout the data cleaning process. Date was identified as an object data type in the pandas operation, thus we converted it to the date datatype. Additionally, there has been a lot of diversity within features.

3.2. Data Visualization:

In our Data Visualization we performed many analysis to find relations in our data set. First we plot closing price vs Date.

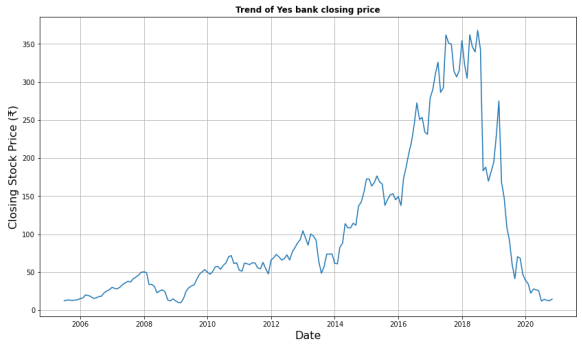


Fig.1: Trend of Yes Bank Closing Price

The closing price declines after 2018 that were seen in the graph above indicates investors' concern.

Fig.2: Distribution of Closing Price

This plotting revealed that the closing price is appropriately skewed. It may lead us to misleading results in view of statistical hypotheses. Applying log transformation can fix that, and after that, we'll examine how the data behaves.

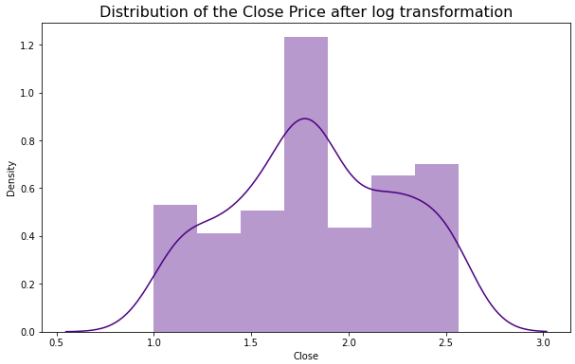
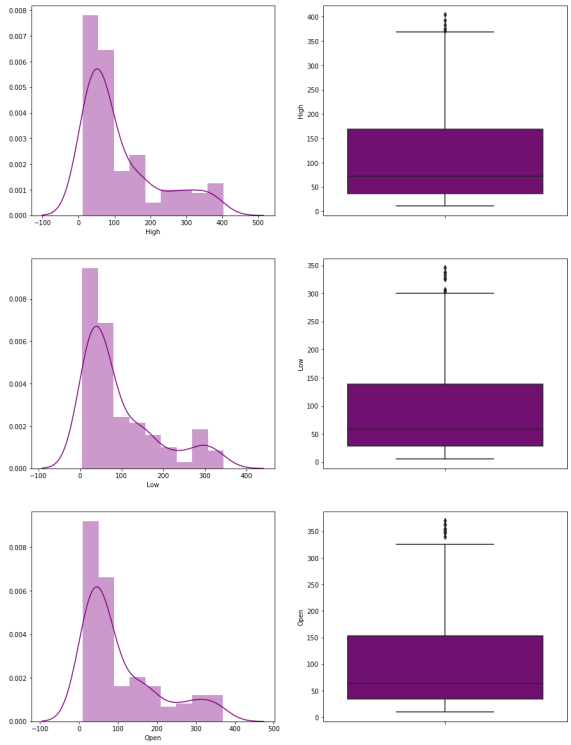


Fig.3: Distribution of the Close Price after log transformation

This graph demonstrates that the closing price distribution is uniform, not right- or left-skewed. The previous graphs' log transformations were used to create this graph.



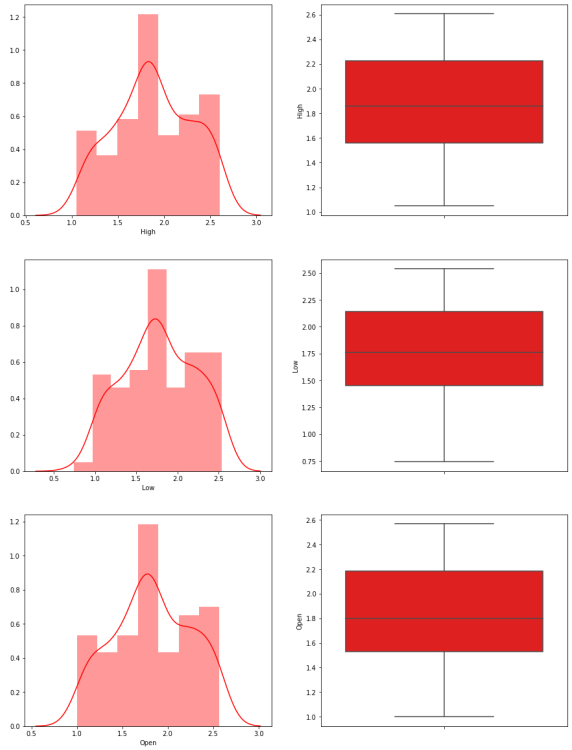
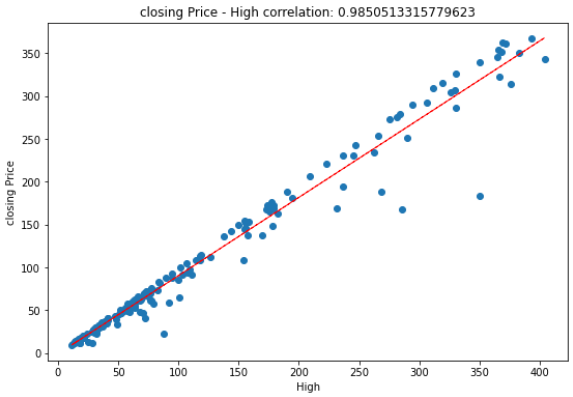


Fig.4: Distribution of High, Low & Open before and after log transformation

To ensure uniform distribution, we used the same data transformation process and for all features, including High, Low & Open.



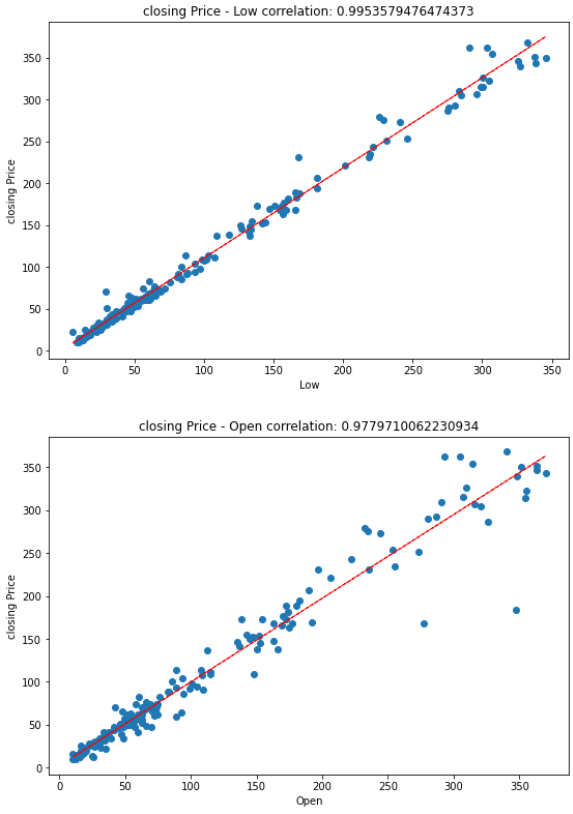


Fig.5: Correlation between Independent variables and Dependent variables

We can observe from the graph above that each independent variable and each dependent variable have a linear relationship and high correlation.

4. Feature Engineering

This is the pre-processing step of machine learning, which is used to transform raw data into features that can be used for creating a predictive model using Machine learning or statistical Modeling. Feature engineering in machine learning aims to improve the performance of models.

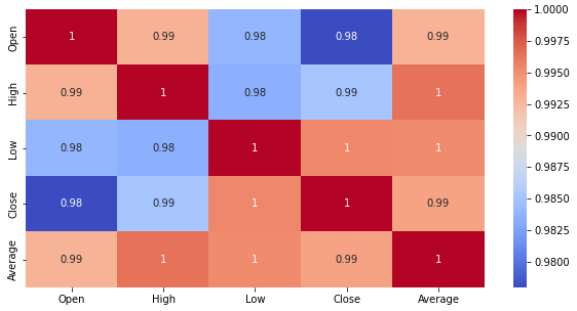


Fig.6: HeatMap for correlation between variables

Independent variables have extremely strong correlations with one another, which causes multicollinearity. High multicollinearity makes it difficult to fit models and make predictions since even small changes to just one independent variable can lead to wildly unanticipated outcomes. Calculating the VIF (Variation Inflation Factor) will allow us to determine which variables to keep in our analysis and prediction model, as well as how much multicollinearity our dataset has.

VIF: Variance Inflation Factor (VIF) is used to detect the presence of multicollinearity.

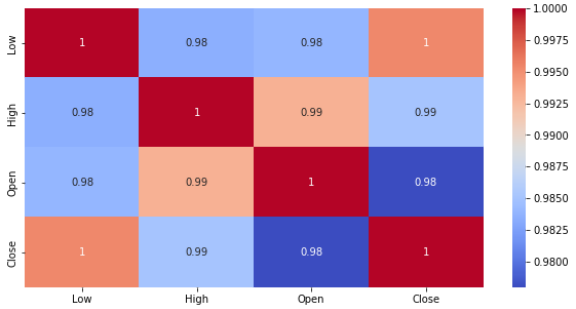
	Variables	**VIF**
0	Open	175.185704
1	High	inf
2	Low	inf
3	Average	inf

Through this table we can see that we have very high VIF in our dataset, so we have to drop one of them which is least correlated with the dependent variable.

Now, let's check the correlation in all independent variables.

	Variables	**VIF**
0	Open	175.185704
1	High	167.057523
2	Low	71.574137

In this table we can see that the "Open" variable has a very high correlation which leads to less reliability in our regression result. But, before deleting any variable again we have to plot a heatmap between the left independent variables and the dependent variable.
So, we can decide which variable we can drop.



Our final dropping variable will be the High feature because it has less correlation with the dependent variable in comparison with the dependent variable(Close). We've dropped 3 features from our dataset. It can affect our model efficiency but neglecting high VIF is far more dangerous than dropping features. So, we preferred to drop the features and move forward with the Low Variable.

5. Model Building:

For our Machine Learning model we have divided our data set in an 80:20 ratio with 80% of our data in the training module and 20% of our data in the testing module.

In this project, we are going to apply 4 models to train our dataset-

1. Linear regression
2. Ridge Regression
3. Lasso Regression
4. Elastic Net Regression

In terms of handling bias, Elastic Net is considered better than Ridge and Lasso regression, Small bias leads to the disturbance of prediction as it is dependent on a variable.

Therefore Elastic Net is better at handling collinearity than the combined ridge and lasso regression.

Also, When it comes to complexity, again, Elastic Net performs better than ridge and lasso regression as in both ridge and lasso, the number of variables is not significantly reduced. Here, the incapability of reducing variables causes declination in model accuracy.

Ridge and Elastic Net could be considered better than the Lasso Regression as Lasso regression predictors do not perform as accurately as Ridge and Elastic Net. Lasso Regression tends to pick non-zero as predictors and sometimes it affects accuracy when relevant predictors are considered as non-zero.

5.1. Linear Regression Model:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis.

Mathematically, we can represent a linear regression as

$$y= a0+a1x+ \varepsilon$$

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

It also has a cost function-

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

After using it in our model we get the following results-

Mean Squared Error: 0.008378716531125619
Root Mean Squared Error: 0.09153532941507131
R2: 0.9550214108859424
Adjusted R2: 0.9147774100996803

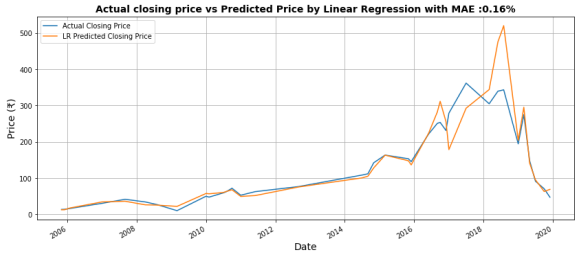


Fig.7: Actual closing price vs Predicted Price by Linear Regression

After the removal of highly correlation columns and the introduction of dummy variables, this graph accurately predicts the closing price with a training accuracy of 95.5%.

5.2. Ridge Regression Model:

Ridge regression is one of the types of linear regression in which a small amount of bias is introduced so that we can get better long-term predictions. It is a regularization technique, which is used to reduce the complexity of the model. It is also called L2 regularization. In this technique, the cost function is altered by adding the penalty term to it. The amount of bias added to the model is called the Ridge Regression penalty. We can calculate it by multiplying the lambda by the squared weight of each individual feature.

The equation for the cost function in ridge regression will be:

$$\sum_{i=1}^{M}(y_i - y'_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{n}\beta_j * x_{ij}\right)^2 + \lambda\sum_{j=0}^{n}\beta_j{}^2$$

In the above equation, the penalty term

regularizes the coefficients of the model, and hence ridge regression reduces the amplitudes of the coefficients that decrease the complexity of the model. As we can see from the above equation, if the values of λ tend to zero, the equation becomes the cost function of the linear regression model. Hence, for the minimum value of λ, the model will resemble the linear regression model. A general linear or polynomial regression will fail if there is high collinearity between the independent variables, so to solve such problems, Ridge regression can be used.

After using it in our model we get the following results-

Mean Squared Error: 0.009365359873489867
Root Mean Squared Error:
0.0967747894520565
R2: 0.9497249164487018
Adjusted R2: 0.9047419469554351

After applying Cross Validation and Hyperparameter Tuning in Ridge we can get-
By Using {'alpha': 3} Negative mean squared error is: -0.012538304166010358

After applying this to our equation we get the following result-

Mean Squared Error: 0.008847513525776934
Root Mean Squared Error:
0.09406122222136461
R2: 0.9525048169276678
Adjusted R2: 0.9100091268103179

In this, we can see that after applying results from cross-validation our model has improved very much and is much more reliable than before.

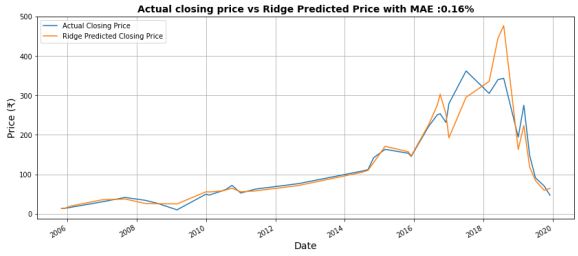


Fig 8. Actual closing price vs Predicted Price by Ridge Regression

After hyperparameter adjustment, cross validation, dummy variable introduction, and removal of strongly correlated, multicollinear columns, this graph predicts closing price with a training accuracy of 94.58%.

5.3. Lasso Regression Model:

Lasso regression is another regularization technique to reduce the complexity of the model. It stands for Least Absolute and Selection Operator. It is similar to the Ridge Regression except that the penalty term contains only the absolute weights instead of a square of weights. Since it takes absolute values, hence, it can shrink the slope to 0, whereas Ridge Regression can only shrink it near to 0. It is also called L1 regularization.
The equation for the cost function of Lasso regression will be:

$$\sum_{i=1}^{M}(y_i - y'_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{n}\beta_j * x_{ij}\right)^2 + \lambda\sum_{j=0}^{n}|\beta_j|$$

Some of the features in this technique are completely neglected for model evaluation.
Hence, the Lasso regression can help us to reduce the overfitting in the model as well as the feature selection.

After using this method we get the following results-
Mean Squared Error: 0.00940109189845321
Root Mean Squared Error:
0.09695922802112861
R2: 0.9495330999499494
Adjusted R2: 0.9043785051683252

After applying Cross Validation and Hyperparameter Tuning in Lasso we can get-
By Using {'alpha': 0.0014} Negative mean squared error is: -0.0126307155201585

After applying this to our equation we get the following result-
Mean Squared Error: 0.009376701436556797
Root Mean Squared Error:
0.09683336943717696
R2: 0.9496640327198869
Adjusted R2: 0.9046265883113646

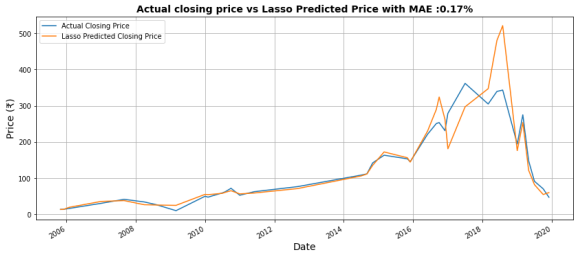


Fig 9. Actual closing price vs predicted price by Lasso Regression

After the removal of highly correlated columns and the introduction of dummy variables, this graph accurately predicts the closing price with a training accuracy of 94.58%.

5.4. Elastic Net Regression:

Coefficients to the variables are considered to be information that must be relevant, however, ridge regression does not promise to remove all irrelevant coefficients which is one of its disadvantages over Elastic Net Regression(ENR)
It uses both Lasso as well as Ridge Regression regularization in order to remove all unnecessary coefficients but not the informative ones.
ENR = Lasso Regression + Ridge Regression
The equation for ENR is given below-:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - (mx_i + z))^2 + \lambda \sum_{i=1}^{p}(mx_i + z)^2 + \lambda \sum_{i=1}^{p}(mx_i + z)$$

Mean Squared Error: 0.030334217104865353
Root Mean Squared Error:
0.17416721018855802
R2: 0.8371599895774178
Adjusted R2: 0.6914610328835284

After applying Cross Validation and Hyperparameter Tuning in Elastic Net we can get-
{'alpha': 0.01, 'l1_ratio': 0.3} Negative mean squared error is: -0.01233335084816529

Mean Squared Error: 0.009096377849834535
Root Mean Squared Error:
0.09537493302663198
R2: 0.9511688645612937
Adjusted R2: 0.9074778486424512

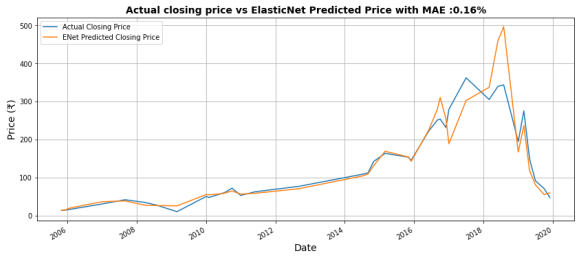


Fig 10. Actual closing price vs predicted price by ElasticNet Regression

After the removal of highly correlated columns and the introduction of dummy variables, this graph accurately predicts the closing price with a training accuracy of 94.58%.

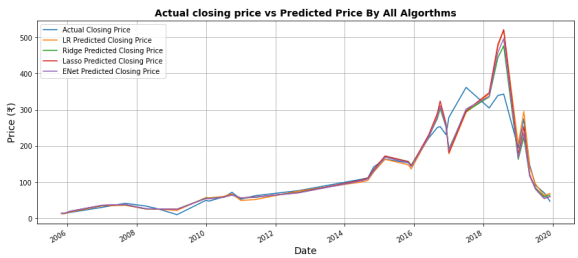


Fig 11. Actual closing price vs predicted by all algorithms

This graph shows the combined result of the predicted price of all models in comparison to the actual closing price.
Linear Regression and Lasso are performing better than other models with training accuracy of 94.0359% and 94.45777% respectively.
Apart from Linear Regression and Lasso, Ridge and Elastic Net are also performing better but they have less training accuracy.

Conclusion:

1. Target Variable is strongly dependent on Independent Variables.
2. Linear Regression and Lasso are performing better than other models with training accuracy 94.0359% and 94.45777% respectively.
3. Apart from Linear Regression and Lasso, Ridge and Elastic Net are also performing better but they have less training accuracy.
4. Ridge and ElasticNet are performing far much better after Applying Hyperparameter Tuning and Cross validation, it is because we have a small set of datasets.
5. R2 and Adjusted R2 are around 95 and 91% in each model.

Future Work:

We can explore hyperparameter tuning and cross validation in order to gain more accuracy and also find ways to reduce multicollinearity.We can also apply more models in our dataset like Random Forest or XGBoost or we can also explore this project with time series analysis method to get more accurate results.

References:

1. https://pandas.pydata.org/
2. https://matplotlib.org/stable/api/index.html
3. https://scikit-learn.org/stable/
4. https://seaborn.pydata.org/examples/index.html