



**YES BANK STOCK CLOSING PRICE PREDICTION**



# TEAMMATES

S.NO.	NAME	REGISTRATION NUMBER
1.	YASH SARIN	RA2211026030015
2.	AKSHAY MALVIYA	RA2211026030034
3.	RISHABH JHA	RA2211026030062
4.	SAARANSH YADAV	RA2211026030063



# ABSTRACT

- ❑ Yes Bank is a banking company that was founded in 2004 that offers a wide range of differentiated products for its corporate and retail customers through retail banking and asset management services. It is also a publically traded company. That provides an opportunity for anyone to invest in Yes bank and become a shareholder. But at the same time, it means that the valuation of the company is now in the hands of investors and speculators as share prices are often heavily impacted by public opinion.
- ❑ We have used yes bank stock price data set. This dataset contains 4 different features that can be used for predicting close price prediction using machine learning. We have built machine learning regression model for price prediction. We have used some of best models.



# OUTLINE

- **Introduction**
- **Problem Statement**
- **Proposed System/Solution**
- **Technology Used**
- **Architectural Diagram**
- **Algorithm & Deployment**
- **Methodologies**
- **Results**
- **Conclusions**
- **References**





# OVERVIEW AND OBJECTIVE

## YES BANK OVERVIEW

YES bank stands for Youth Enterprise Scheme Bank. Stock market is one of the major fields that attracts people, thus stock market price prediction is always a hot topic for researchers from both financial and technical domains. In our project our objective is to build a prediction model for close price prediction. A stock market is a public market where you can buy and sell shares for publicly listed companies. Stock Price Prediction using machine learning helps you get an estimate of value of company stock going forward and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are numerous other factors involved in the prediction, such as the psychological factor – namely crowd behavior etc. All these factors combine to make share prices very difficult to predict with high accuracy.

## YES BANK OBJECTIVE

Yes Bank is a well-known bank in the Indian financial domain. It has been in the headlines since 2018 as a result of the Rana Kapoor fraud case. Due to this, it was interesting to observe how it affected the company's stock prices and whether Time series models or other prediction models could properly reflect for such circumstances. Since the bank's founding, this dataset has included closing, starting, highest, and lowest stock prices for each month.



# DATASETS

## YES BANK STOCK CLOSING PRICE PREDICTION DATASET

We have 185 rows and 5 columns in our dataset. Here our dependent variable is Close and Independent variable is Open, High and Low.

**Date :-** It denotes the month and year for a specific pricing.

**Open :-** The price at which a stock started trading that month is referred to as the "Open."

**High :-** The highest price for that particular month.

**Low :-** It describes the monthly minimum price.

**Close :-** It refers to the final trading price for that month, which we have to predict using regression.

**Independent or input variables :-** ' Open ', ' High ', ' Low '

**Dependent or target variable :-** ' Close '

' Date 'is only useful for EDA purpose & don't have any influence for closing price prediction.



# 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.



# PROPOSED SOLUTION

## 1. Data Understanding and Preparation

### •Dataset Review:

- Examine the provided dataset containing monthly stock prices (closing, opening, highest, lowest) since the bank's inception.
- Ensure data consistency, handle missing values, and potentially outliers.

### •Feature Selection:

- Focus primarily on the closing price as the target variable for prediction.
- Other features like opening, highest, and lowest prices can be used for additional analysis or feature engineering.

## 2. Exploratory Data Analysis (EDA)

- Visualize the trends, seasonality, and any underlying patterns in the closing prices over time.
- Use statistical techniques like autocorrelation function and partial autocorrelation function plots to understand dependencies.

## 3. Model Training :

- Spilt data into training and validation sets.
- Train model on historical data and validate using out of sample data to assess performance.

## 4. Deployment:

- Implement the model in a production environment for ongoing predictions.
- Monitor model performance and retrain periodically if necessary.

## 5. Model Evaluation

- Use metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) to evaluate model accuracy.
- Compare different models to choose the best performing one.

## 6. Result





# TECHNOLOGY USED

- ❑ **System requirements : Windows 10/11 operating system , 6/8GB RAM, i5/i7 Processor, Python 3.11/3.10, Jupyter Notebook.**
  
- ❑ **Library required to build the model :**
  - NumPy
  - Panda
  - Matplotlib
  - Seaborn
  - Datetime
  - **Sklearn**

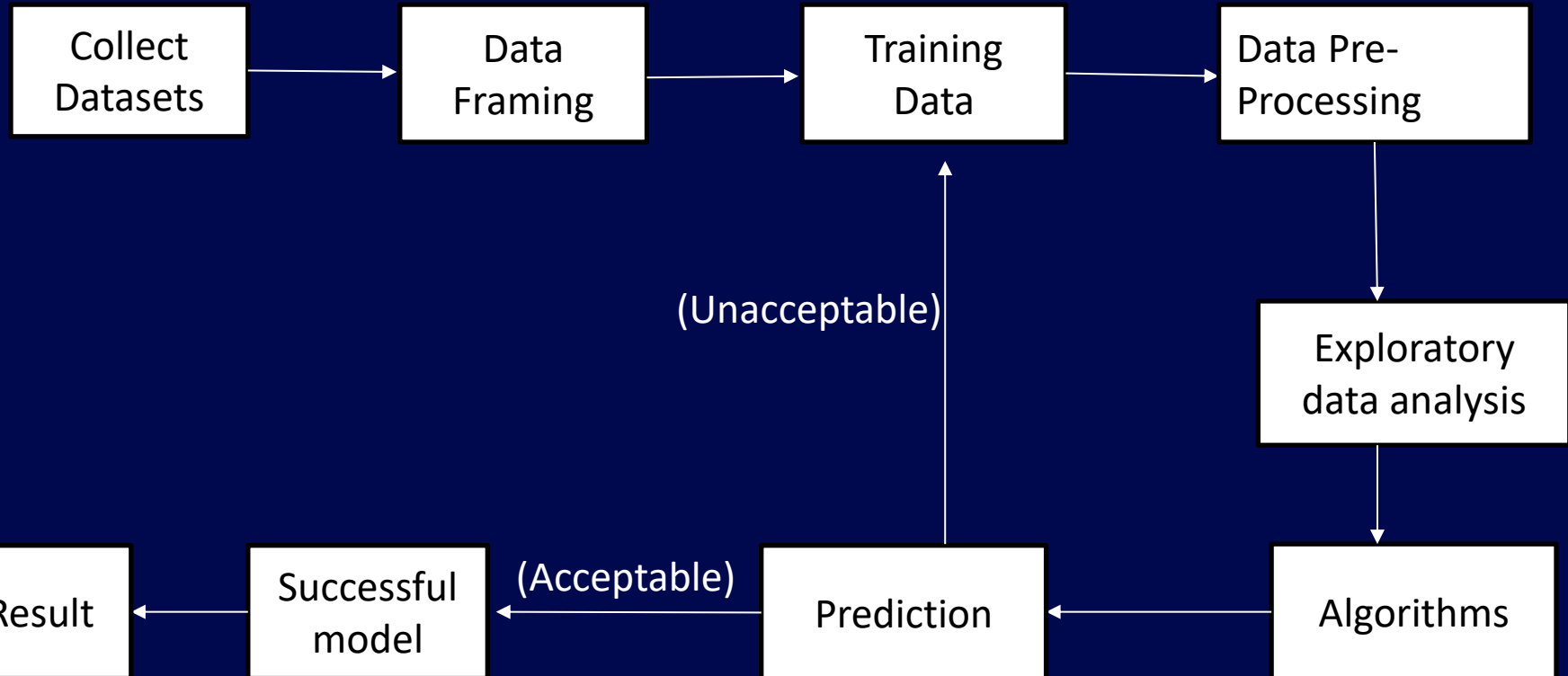


# ALGORITHM & DEPLOYMENT

- ❑ **Step 1- Dataset- Collection of dataset from Kaggle.**
- ❑ **Step 2- Data Framing – Define the success metrics used , define the ideal outcome and models goals. Or what is observed and what should be predicted based on Problem Statement.**
- ❑ **Step 3-Data Pre-Processing - Data processing is a crucial step in the machine learning (ML) , as it prepares the data for use in building and training the models.**
- ❑ **Step 4- Exploratory data analysis (EDA)- Summarize the main characteristics according to the dataset.**
- ❑ **Step 5- K-Nearest Neighbours (KNN)-The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, normalizing the training data can improve its accuracy dramatically.**
- ❑ **Step 6- Random Forest: Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and aggregation , commonly known as bagging.**
- ❑ **Step 7- Implementation of Regression- We create 4 regression models for our data such as Linear Regression, Lasso Regression , Ridge Regression , Elastic Net Regression.**
- ❑ **Step 8- Conclusion**



# ARCHITECTURAL DIAGRAM





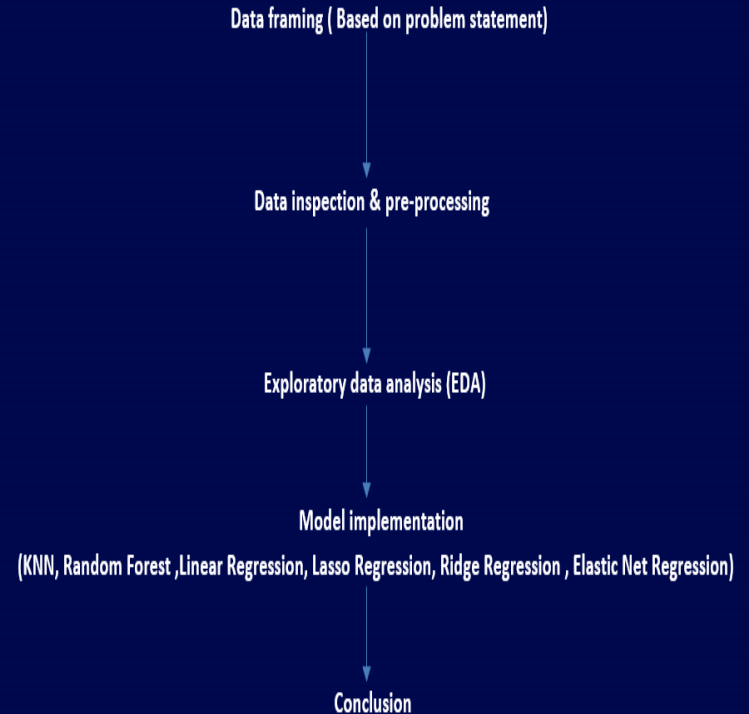
# METHADOLOGY

1.Data Collection: Gathered comprehensive historical stock data for Yes Bank.

2.Feature Engineering: Engineered relevant features to enhance model performance.

3.Model Training: Utilized advanced machine learning models for accurate predictions.

4.Evaluation: Rigorous evaluation to ensure the model's effectiveness and reliability.







# RESULTS

## Data Cleaning: -

The Given Date in data is of Month-year format (mmm-yy) is converted to proper date of YYYY-MM-DD and given date column has dtype as object converting it into date time format.

value in the datasets

DATA CLEANING

Out[38]:

	Date	Open	High	Low	Close
0	Jul-05	13.00	14.00	11.25	12.46
1	Aug-05	12.58	14.88	12.55	13.42
2	Sep-05	13.48	14.87	12.27	13.30
3	Oct-05	13.20	14.47	12.40	12.99
4	Nov-05	13.35	13.88	12.88	13.41

Out[42]:

	Open	High	Low	Close
count	185.000000	185.000000	185.000000	185.000000
mean	105.541405	116.104324	94.947838	105.204703
std	98.879850	106.333497	91.219415	98.583153
min	10.000000	11.240000	5.550000	9.980000
25%	33.800000	36.140000	28.510000	33.450000
50%	62.980000	72.550000	58.000000	62.540000
75%	153.000000	169.190000	138.350000	153.300000
max	369.950000	404.000000	345.500000	367.900000

Description of datasets



# RESULTS

## Data Frame Information



In [43]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 185 entries, 0 to 184  
Data columns (total 5 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   Date    185 non-null     object  
1   Open    185 non-null     float64  
2   High    185 non-null     float64  
3   Low     185 non-null     float64  
4   Close   185 non-null     float64  
dtypes: float64(4), object(1)  
memory usage: 7.4+ KB
```

## Data Preprocessing



In [81]: `ys.head()`

Out[81]:

	Open	High	Low	Close
0	-1.671724	-1.762828	-1.622235	-1.709751
1	-1.704449	-1.700176	-1.518365	-1.635775
2	-1.635576	-1.700867	-1.539820	-1.644731
3	-1.656505	-1.728898	-1.529800	-1.668243
4	-1.645239	-1.771672	-1.493670	-1.636518

# RESULTS

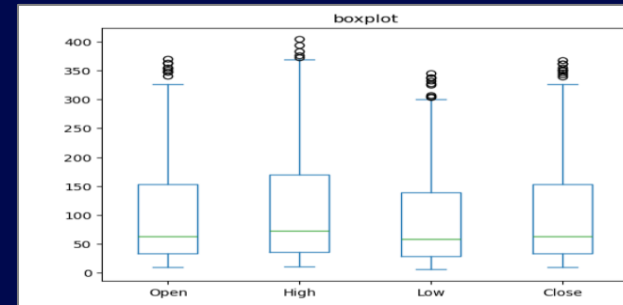
❑ **Univariate Analysis:** In our yes bank stock market dataset all the features have positively skewed distributions.

Power Transform



```
pw = PowerTransformer(method='box-cox', standardize=True)
cf = df[list(df.columns)]
df['Close']
```

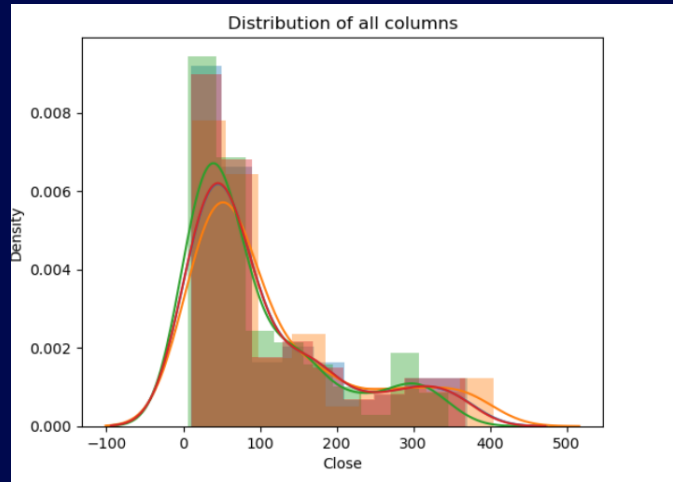
```
Date
Jul-05    12.46
Aug-05    13.42
Sep-05    13.30
Oct-05    12.99
Nov-05    13.41
...
Jul-20    11.95
Aug-20    14.37
Sep-20    13.15
Oct-20    12.42
Nov-20    14.67
Name: Close, Length: 185, dtype: float64
```



All the features have normally distributed after log transformations.

# RESULTS

## VISUALIZATION OF DISTRIBUTION OF ALL COLUMNS OF CLOSING PRICE:



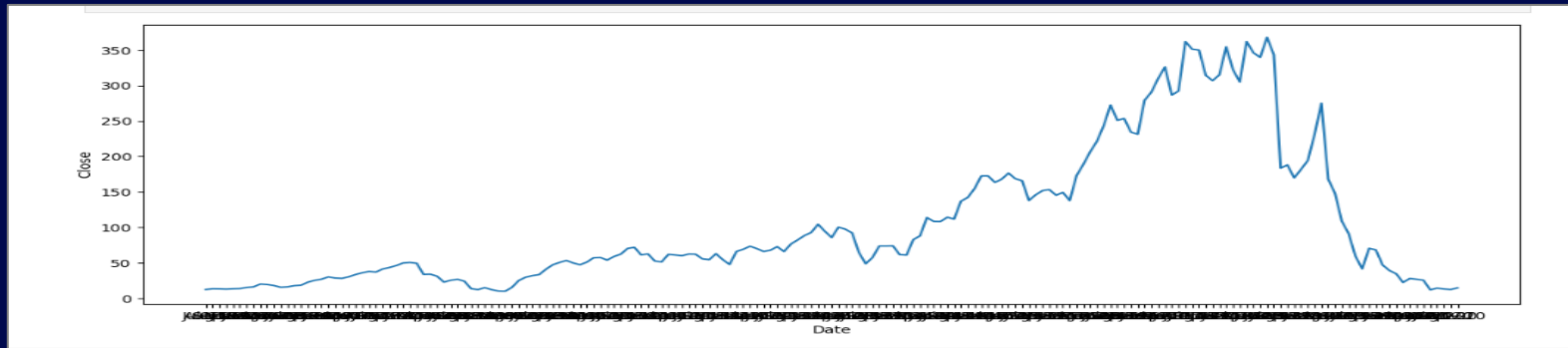
The above graph shows that they are not normally distributed. The mean and median should be equal for perfect normal distribution curve. So we log transform all the features to normal distribution.



# RESULTS

## □ Bivariate Analysis:

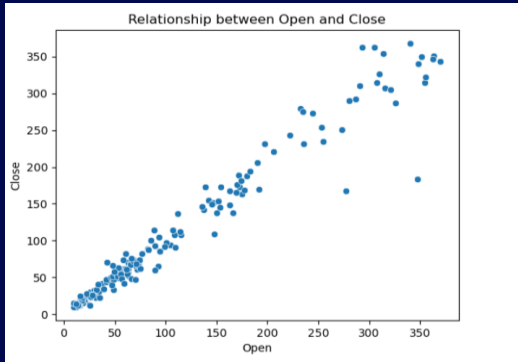
In the context of supervised learning, it can help determine the essential predictors when the bivariate analysis is done by plotting one variable against another. The graphs below depict that there is a high correlation between the dependent (Close) and independent variables.



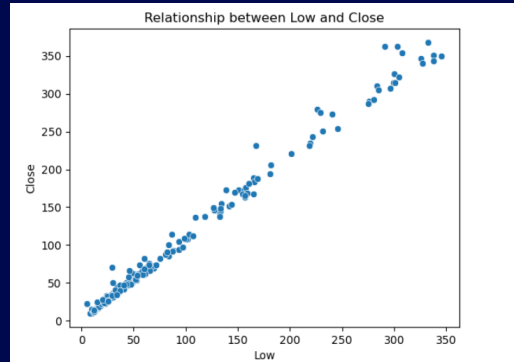
From the above line plot, We conclude that the stock price is keep on increasing till 2018. But after 2018, the stock price is kept on decreasing due the fraud case involving Rana Kapoor.

# RESULTS

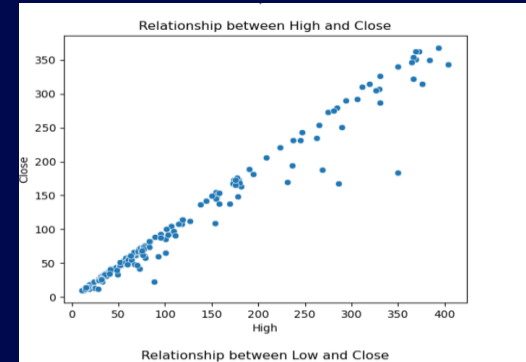
## VISUALIZATION OF RELATIONSHIP OF ALL DATASETS OF CLOSING PRICE



Relationship between Open and Close



Relationship between Low and Close

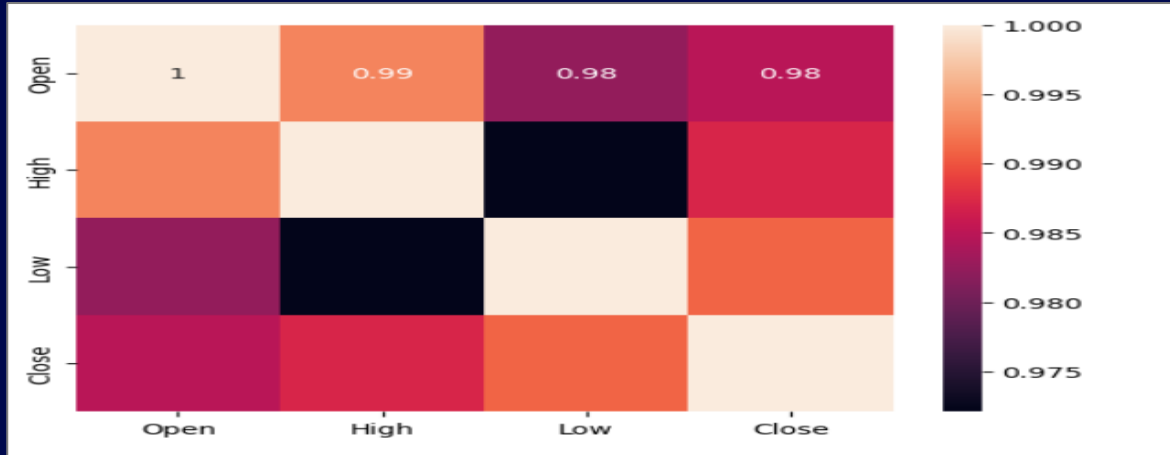


Relationship between High and Close



# RESULTS

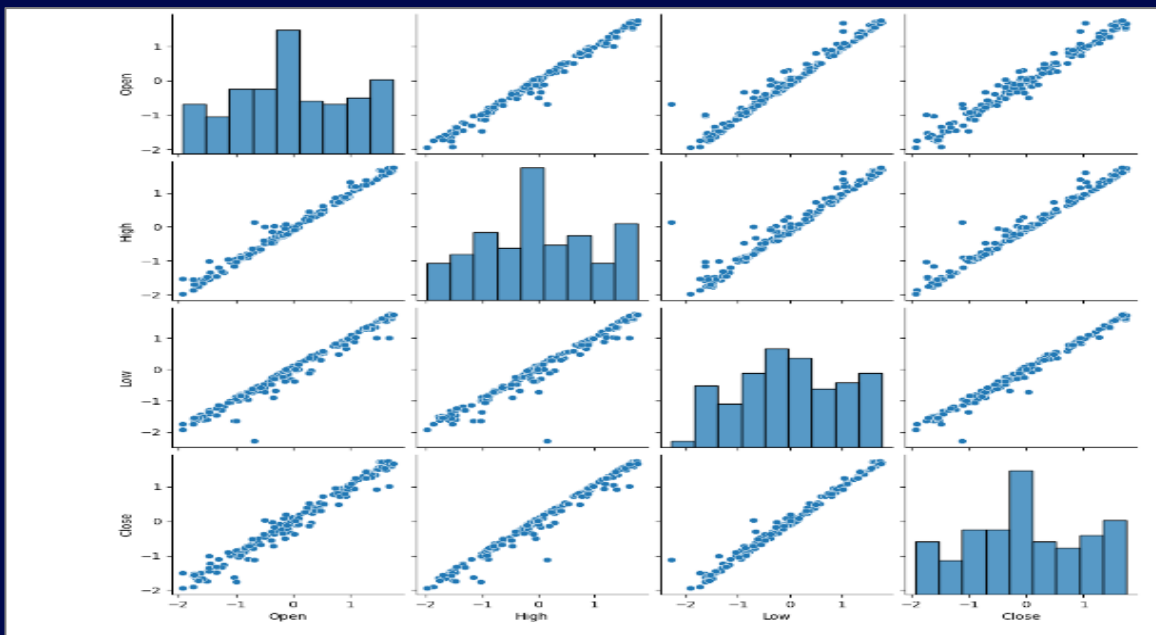
Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between numerical variables. This heatmap shows us the correlation between all numerical variables in our data. Every feature is extremely correlated with each other, so taking just one feature or average of these features would suffice for our regression model as linear regression assumes there is no multicollinearity in the features. We will try to reduce multicollinearity using the transformation of variables.





# RESULTS

VISUALIZATION OF EVERY SINGLE COLUMN OF OUR DATA FRAME AGAINST EVERY OTHER COLUMN:





# RESULTS

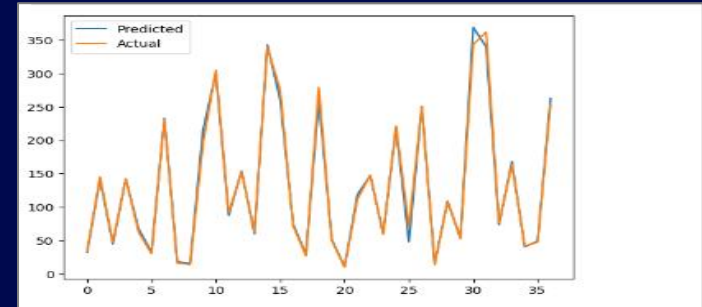
## K-Nearest Neighbors (KNN):

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, normalizing the training data can improve its accuracy dramatically.

```
GridSearchCV
GridSearchCV(cv=5, estimator=KNeighborsRegressor(),
             param_grid={'n_neighbors': [2, 3, 4, 5, 6, 7, 8, 9]})
  estimator: KNeighborsRegressor
    KNeighborsRegressor()
      KNeighborsRegressor
        KNeighborsRegressor()
```

```
knn = KNeighborsRegressor(n_neighbors=3)
knn.fit(x_train,y_train)
```

```
KNeighborsRegressor
KNeighborsRegressor(n_neighbors=3)
```



# RESULTS

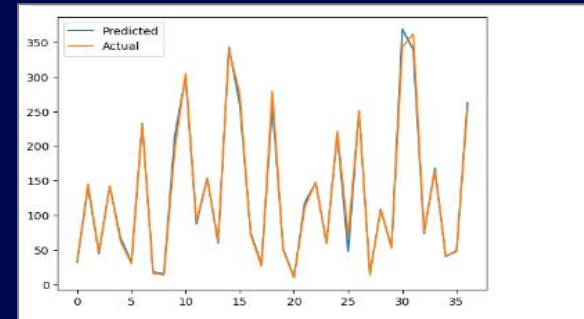
## Random Forest:

Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging.

```
GridSearchCV
GridSearchCV(cv=7, estimator=RandomForestRegressor(),
             param_grid={'criterion': ['squared_error', 'absolute_error',
                                         'friedman_mse', 'poisson'],
                           'max_features': ['sqrt', 'log2', None],
                           'n_estimators': [100, 200, 300]})
  estimator: RandomForestRegressor
    RandomForestRegressor()
      RandomForestRegressor
        RandomForestRegressor()
```

```
rf = RandomForestRegressor(criterion='friedman_mse', max_features=None, n_estimators=300)
rf.fit(x_train, y_train)
```

```
RandomForestRegressor
RandomForestRegressor(criterion='friedman_mse', max_features=None,
                      n_estimators=300)
```



# RESULTS

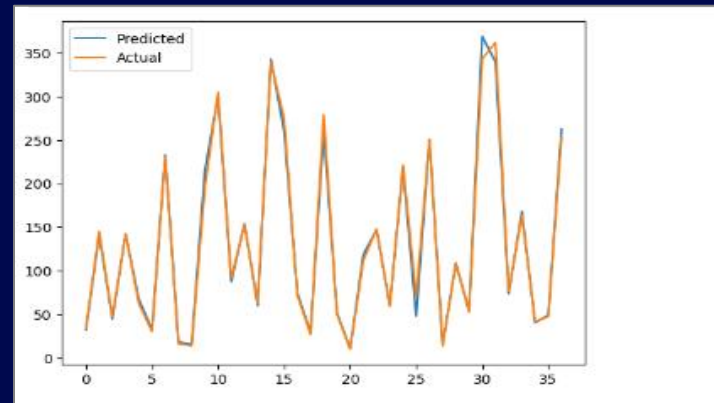
## Regression Model:

### ❑ Linear Regression:

Linear regression is the most basic machine learning approach that can be applied to this data. The result of the linear regression model is an equation showing how the independent variables and dependent variable related to each other.

▼ LinearRegression

LinearRegression()



# RESULTS

## ❑ Ridge Regression:

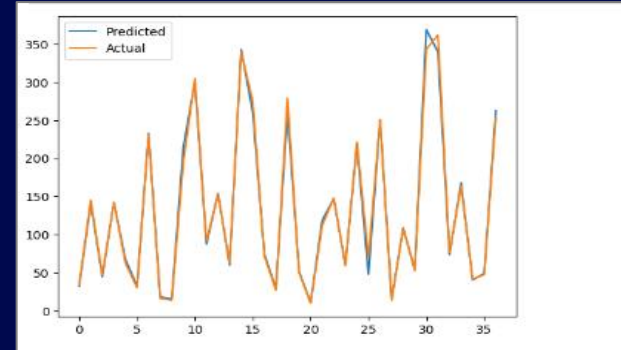
Ridge regression is a model-tuning technique that is used to analyze any multicollinear data. L2 regularization is done using this technique. The projected values vary significantly from the actual values when the problem of multicollinearity is present, least-squares are unbiased, and variances are large.

```
ridge = Ridge()
param = {'alpha': [1e-15, 1e-10, 1e-8, 1e-5, 1e-4, 1e-3, 1e-2, 0.3, 0.7, 1, 1.2, 1.33, 1.365, 1.37, 1.375, 1.4, 1.5, 1.6, 1.8, 2.5, 5, 10, 20, 30, 40, 45, 50, 55, 60, 100]}
ridge_regressor = GridSearchCV(ridge, param, scoring='neg_mean_squared_error', cv=3)
ridge_regressor.fit(x_train, y_train)
ridge_regressor.best_params_

{'alpha': 100}
```

```
ridge = Ridge(alpha= 150)
ridge.fit(x_train, y_train)
```

```
▼ Ridge
Ridge(alpha=150)
```





# RESULTS

## ❑ Elastic Net Regression:

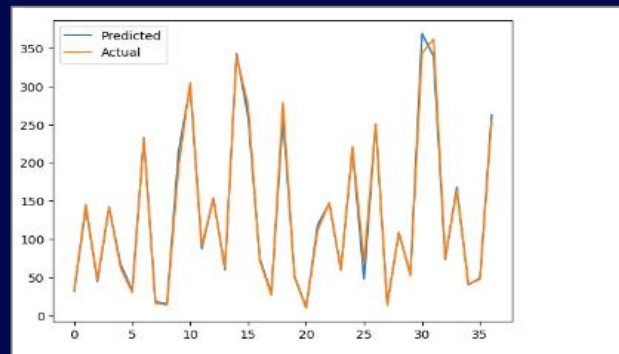
Elastic net regression works in a manner that takes the best of lasso and ridge regressions. It adds up the penalty terms for regularization in lasso and ridge (L1 and L2) and uses that for regularization. It is used for regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

```
GridSearchCV
GridSearchCV(cv=5, estimator=ElasticNet(),
             param_grid={'alpha': [1e-15, 1e-13, 1e-10, 1e-08, 1e-05, 0.0001,
                                   0.001, 0.001, 0.01, 0.02, 0.03, 0.04, 1, 5,
                                   10, 20, 40, 50, 60, 100],
                        'l1_ratio': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,
                                    0.9]},
             scoring='neg_mean_squared_error')
  ▾ estimator: ElasticNet
    ElasticNet()
      ▾ ElasticNet
        ElasticNet()
```

```
elastic =ElasticNet(alpha = 5, l1_ratio = 0.1)

elastic.fit(x_train,y_train)

ElasticNet
ElasticNet(alpha=5, l1_ratio=0.1)
```



# RESULTS

## ❑ Lasso Regression:

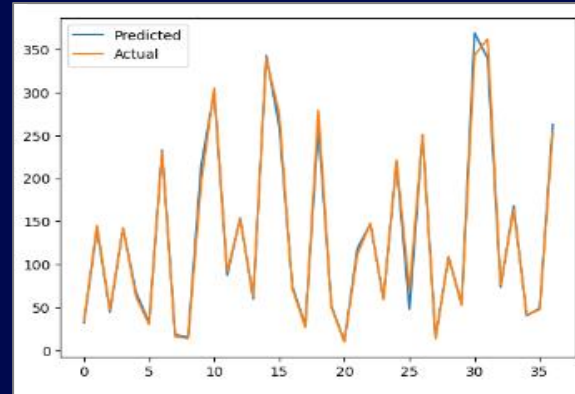
Lasso (least absolute shrinkage and selection operator) regression is another technique of Parameter estimation regression method. This method is usually used in machine learning for the selection of the subset of variables. It provides greater prediction accuracy as compared to other regression models. Lasso Regularization enhances the accessibility of models.

```
GridSearchCV
GridSearchCV(cv=3, estimator=Lasso(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 1e-05, 0.0001, 0.001,
                                   0.01, 0.3, 0.7, 1, 1.2, 1.33, 1.365, 1.37,
                                   1.375, 1.4, 1.5, 1.6, 1.8, 2.5, 5, 10, 20,
                                   30, 40, 45, 50, 55, 60, 100]}),
             scoring='neg_mean_squared_error')
  estimator: Lasso
    Lasso()
      Lasso
        Lasso()
```

```
lasso = Lasso(alpha=1.6)

lasso.fit(x_train, y_train)
```

```
Lasso
Lasso(alpha=1.6)
```





# CONCLUSION

- ❑ The trend of the price of Yes Bank's stock increased until 2018 and then Close, Open, High, Low price decreased.
- ❑ Based on the open vs. close price graph, we concluded that Yes Bank's stock fell significantly after 2018.
- ❑ Both duplicate and null values are absent, as we have seen. But object data type values are available for the Date feature.
- ❑ The dependent and independent values were found to be linearly related.
- ❑ The data contained a significant amount of multicollinearity.
- ❑ Decision Tree regression is best model for yes bank stock closing price data this model use for further prediction
- ❑ Visualization has allowed us to notice that the closing price of the stock has suddenly fallen starting in 2018. It seems reasonable that the Yes Bank stock price was significantly impacted by the Rana Kapoor case fraud.
- ❑ KNN performed the worst out of all.
- ❑ In this work, we create 4 regression models for our data:-
  1. Linear Regression • 2. Lasso Regression • 3. Ridge Regression 4. **Elastic Net** Regression
- ❑ These four models gives us the following results: High, Low, Open are directly correlate with the closing price of the stocks.



# REFERENCES

- ❑ <https://www.kaggle.com/datasets/simranjain17/yes-bank-stock-prices>
- ❑ <https://www.youtube.com/watch?app=desktop&v=lx-dcCr0JCI>
- ❑ <https://www.indiapropertydekho.com/article/154/yes-bank-share-price-target>



THANK YOU