Stock Market Analysis – Price Movement Prediction

Apoorv Verma

Contents

Abstract

Introduction

Exploratory Data Analysis

Data Cleaning and transformation

Feature Creation

Outlier Analysis

Model Performance Analysis

Results and Insights

Conclusion

Abstract

Our project focus is to Predict stock performance within the NIFTY-50 index using machine learning techniques. By studying these price patterns, the project helps uncover trends that make it easier for investors to make smarter financial decisions.

Classification Strategy:

Stocks are categorized into **High**, **Medium**, and **Low** based on price changes.

Stock Data Analysis:

Implemented machine learning models to predict stock performance:

- **Baseline Model**: Serves as a benchmark for comparison.
- **Naïve Bayes**
- **Support Vector Machines (SVM)**
- **Logistic Regression**

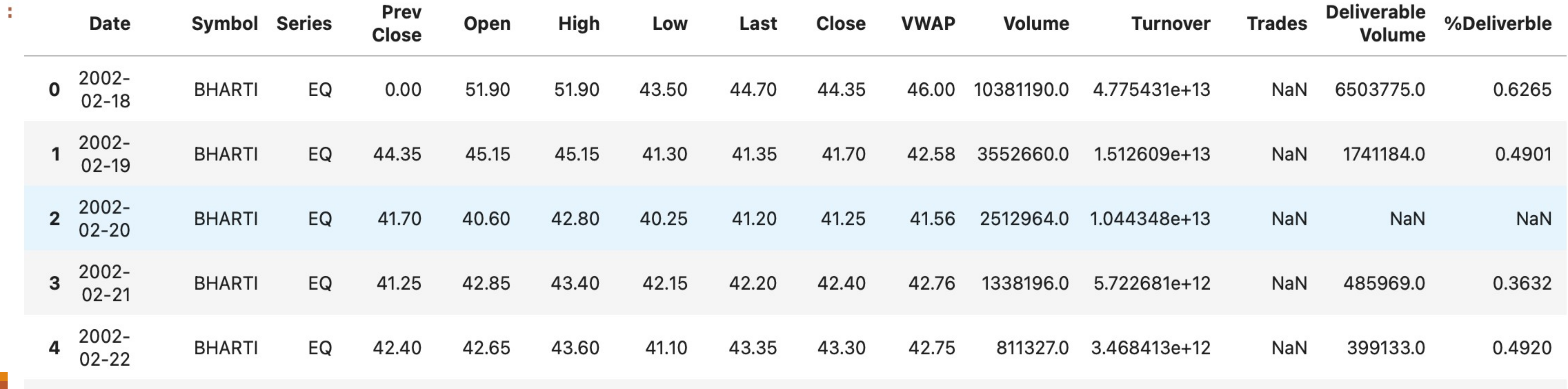
Dataset Details

The NIFTY-50 index, represents 50 of India’s top companies, serves as a key benchmark for understanding the country’s stock market trends.

Dataset contains low frequency stocks data.

The data set contains the stock details from the year 2000 to 2021.

It consists of 470434 rows, 18 columns.



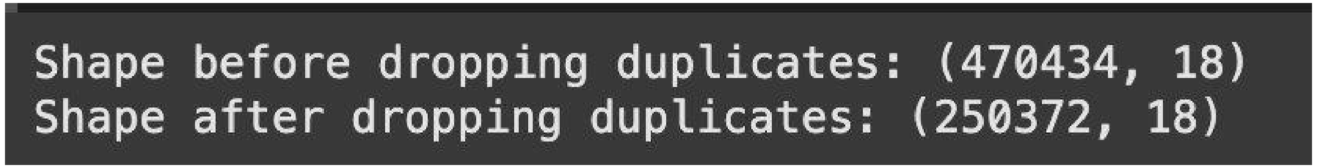
	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
0	2002-02-18	BHARTI	EQ	0.00	51.90	51.90	43.50	44.70	44.35	46.00	10381190.0	4.775431e+13	NaN	6503775.0	0.6265
1	2002-02-19	BHARTI	EQ	44.35	45.15	45.15	41.30	41.35	41.70	42.58	3552660.0	1.512609e+13	NaN	1741184.0	0.4901
2	2002-02-20	BHARTI	EQ	41.70	40.60	42.80	40.25	41.20	41.25	41.56	2512964.0	1.044348e+13	NaN	NaN	NaN
3	2002-02-21	BHARTI	EQ	41.25	42.85	43.40	42.15	42.20	42.40	42.76	1338196.0	5.722681e+12	NaN	485969.0	0.3632
4	2002-02-22	BHARTI	EQ	42.40	42.65	43.60	41.10	43.35	43.30	42.75	811327.0	3.468413e+12	NaN	399133.0	0.4920

Exploratory Data Analysis

Handling Duplicates

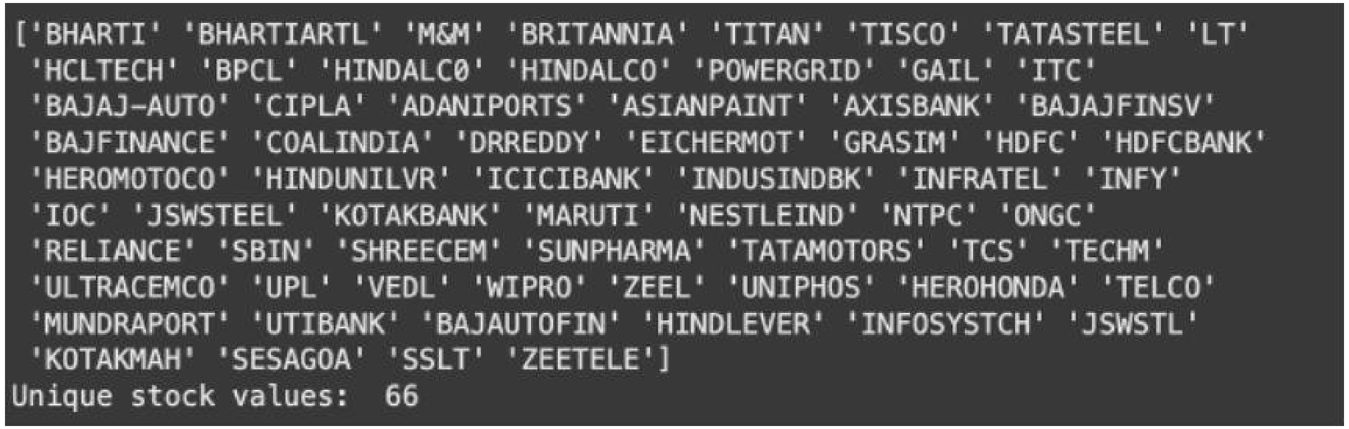
Data Preprocessing and Cleaning

- Drop duplicate rows while keeping the first occurrence



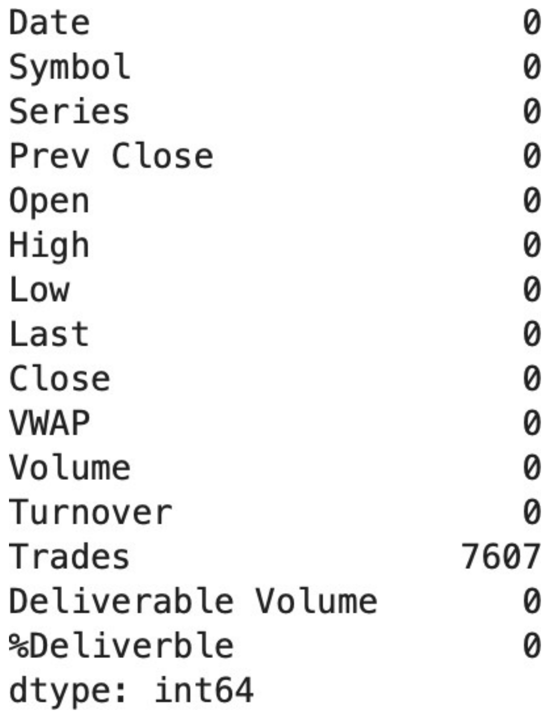
```
Shape before dropping duplicates: (470434, 18)
Shape after dropping duplicates: (250372, 18)
```

- Renaming stocks that got changed overtime, Replace old stock names with current names in dataframe.



```
['BHARTI' 'BHARTIARTL' 'M&M' 'BRITANNIA' 'TITAN' 'TISCO' 'TATASTEEL' 'LT'
 'HCLTECH' 'BPCL' 'HINDALC0' 'HINDALCO' 'POWERGRID' 'GAIL' 'ITC'
 'BAJAJ-AUTO' 'CIPLA' 'ADANIPORTS' 'ASIANPAINT' 'AXISBANK' 'BAJAJFINSV'
 'BAJFINANCE' 'COALINDIA' 'DRREDDY' 'EICHERMOT' 'GRASIM' 'HDFC' 'HDFCBANK'
 'HEROMOTOCO' 'HINDUNILVR' 'ICICIBANK' 'INDUSINDBK' 'INFRATEL' 'INFY'
 'IOC' 'JSWSTEEL' 'KOTAKBANK' 'MARUTI' 'NESTLEIND' 'NTPC' 'ONGC'
 'RELIANCE' 'SBIN' 'SHREECEM' 'SUNPHARMA' 'TATAMOTORS' 'TCS' 'TECHM'
 'ULTRACEMCO' 'UPL' 'VEDL' 'WIPRO' 'ZEEL' 'UNIPHOS' 'HEROHONDA' 'TELCO'
 'MUNDRAPORT' 'UTIBANK' 'BAJAUTOFIN' 'HINDLEVER' 'INFOSYSTCH' 'JSWSTL'
 'KOTAKMAH' 'SESAGOA' 'SSLT' 'ZEETELE']
Unique stock values:  66
```

Handling Missing Data

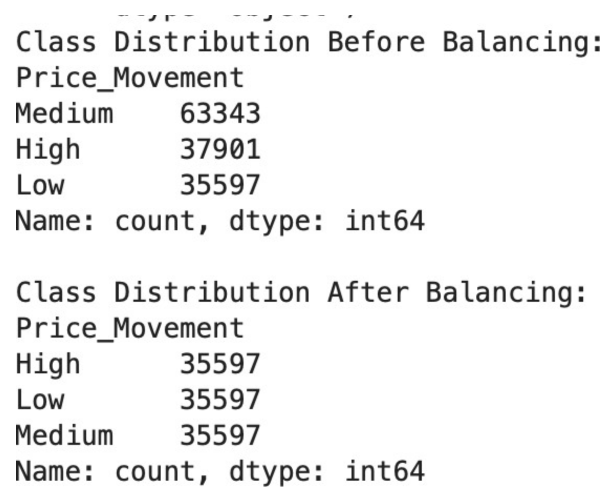


```
Date                   0
Symbol                 0
Series                 0
Prev Close             0
Open                   0
High                   0
Low                    0
Last                   0
Close                  0
VWAP                   0
Volume                 0
Turnover               0
Trades              7607
Deliverable Volume     0
%Deliverble            0
dtype: int64
```

Handled missing values using KNN imputation to handle missing values. (3-NN Imputer)

Feature Creation

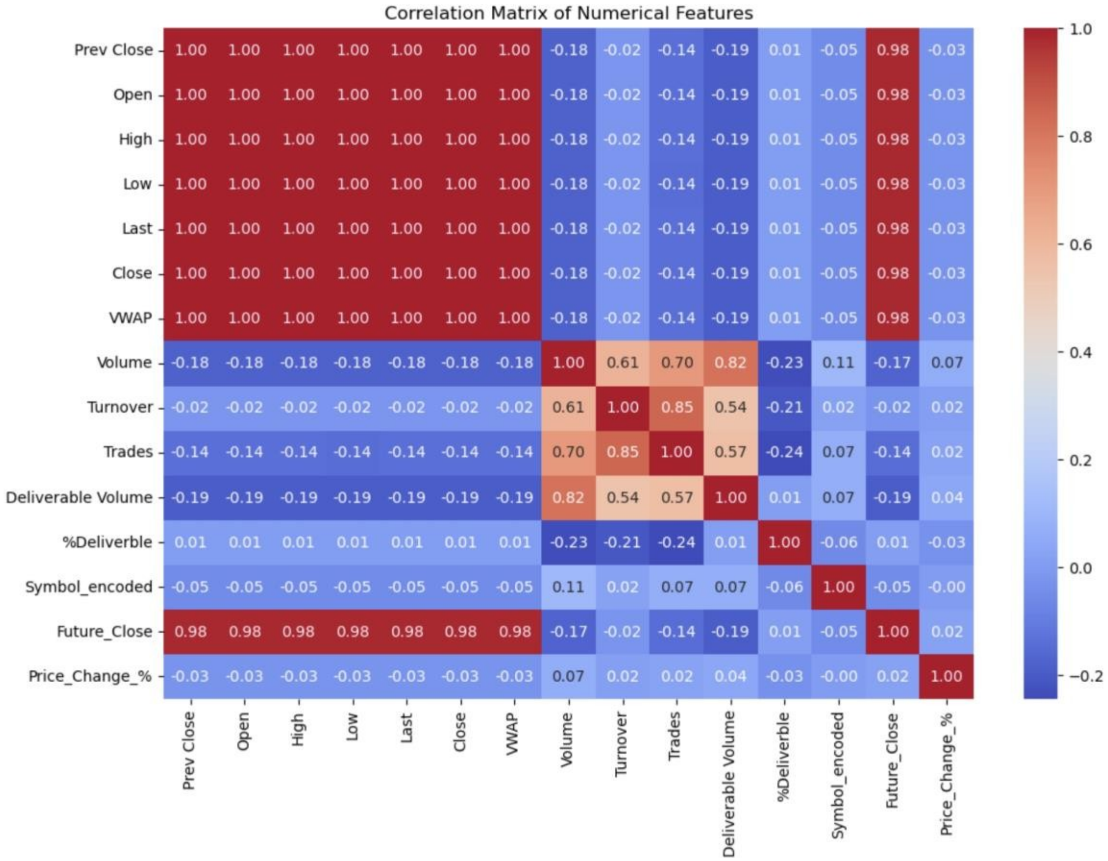
- Future_Close: Calculated by shifting the Close prices forward by a time horizon of 10 trading days.
- ((Future_Close−Close)/Close)×100 to measure the percentage change in stock price over 10 trading days.
- Price_Movement Classification:
  Categorized Price_Change_% into: High: >4%, Low: < −3%, Medium –3 to 4
- **Class Balancing**: with **stratified sampling 36,172 per class** to avoid bias during model training.
- Applied Label Encoding to transform categorical columns (Symbol and Series) into numerical format.



```
Class Distribution Before Balancing:
Price_Movement
Medium    63343
High      37901
Low       35597
Name: count, dtype: int64

Class Distribution After Balancing:
Price_Movement
High      35597
Low       35597
Medium    35597
Name: count, dtype: int64
```

Correlation Analysis Among Input Features

- High correlation among price-related variables, indicating multicollinearity.
- Volume and %Deliverable exhibit weak correlations with price-related variables, indicating they provide unique insights.
- Strong positive correlations exist between Turnover, Trade and Volume, reinforcing their interdependence in transactional metrics.



Correlation Matrix of Numerical Features

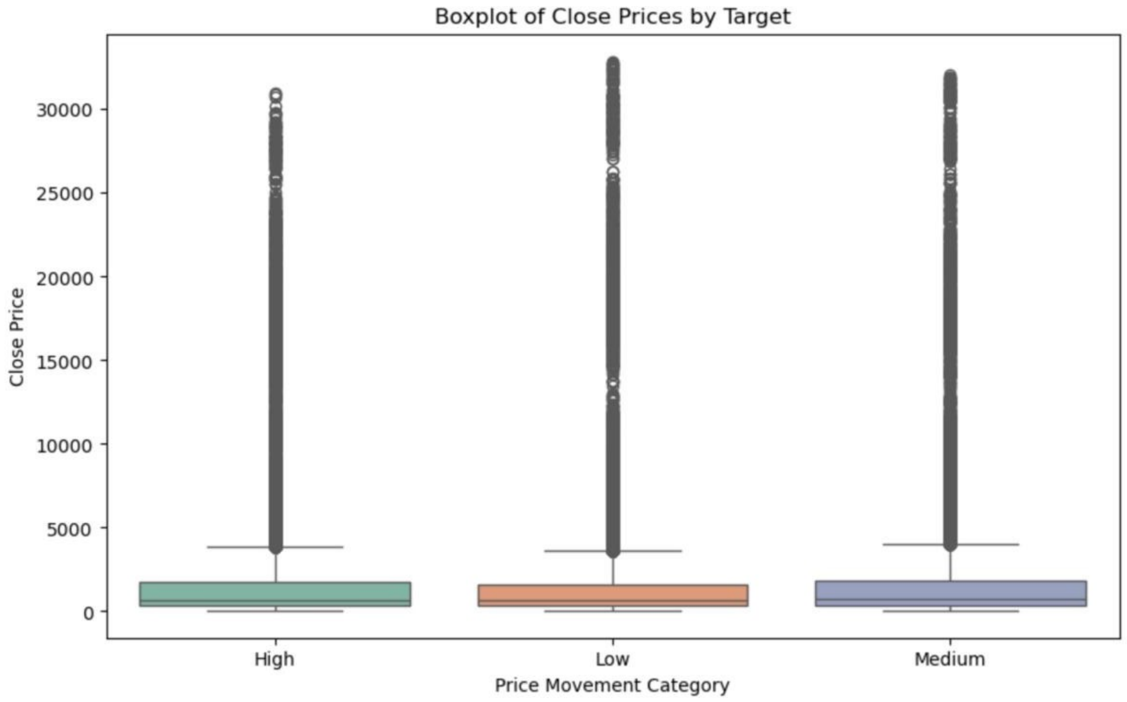
	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble	Symbol_encoded	Future_Close	Price_Change_%
Prev Close	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.18	-0.02	-0.14	-0.19	0.01	-0.05	0.98	-0.03
Open	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.18	-0.02	-0.14	-0.19	0.01	-0.05	0.98	-0.03
High	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.18	-0.02	-0.14	-0.19	0.01	-0.05	0.98	-0.03
Low	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.18	-0.02	-0.14	-0.19	0.01	-0.05	0.98	-0.03
Last	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.18	-0.02	-0.14	-0.19	0.01	-0.05	0.98	-0.03
Close	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.18	-0.02	-0.14	-0.19	0.01	-0.05	0.98	-0.03
VWAP	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.18	-0.02	-0.14	-0.19	0.01	-0.05	0.98	-0.03
Volume	-0.18	-0.18	-0.18	-0.18	-0.18	-0.18	-0.18	1.00	0.61	0.70	0.82	-0.23	0.11	-0.17	0.07
Turnover	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	0.61	1.00	0.85	0.54	-0.21	0.02	-0.02	0.02
Trades	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	0.70	0.85	1.00	0.57	-0.24	0.07	-0.14	0.02
Deliverable Volume	-0.19	-0.19	-0.19	-0.19	-0.19	-0.19	-0.19	0.82	0.54	0.57	1.00	0.01	0.07	-0.19	0.04
%Deliverble	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.23	-0.21	-0.24	0.01	1.00	-0.06	0.01	-0.03
Symbol_encoded	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	0.11	0.02	0.07	0.07	-0.06	1.00	-0.05	-0.00
Future_Close	0.98	0.98	0.98	0.98	0.98	0.98	0.98	-0.17	-0.02	-0.14	-0.19	0.01	-0.05	1.00	0.02
Price_Change_%	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	0.07	0.02	0.02	0.04	-0.03	-0.00	0.02	1.00

Distribution For Different Class Labels

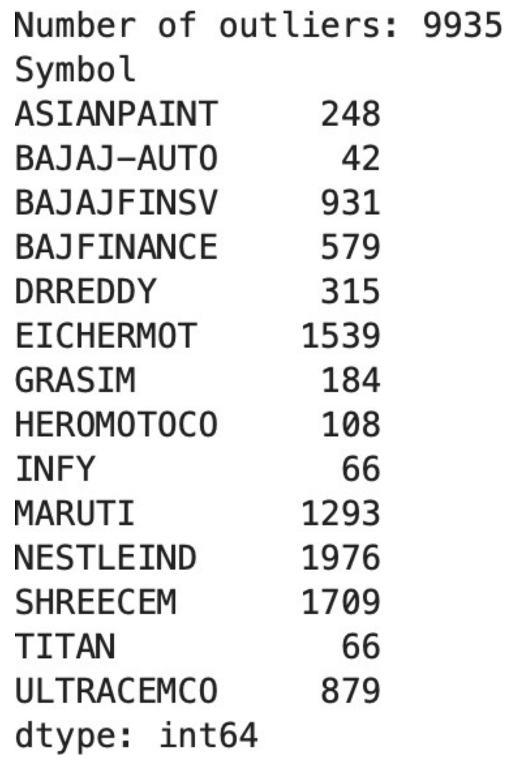
- The boxplot displays the distribution of Close prices across three price movement categories: High, Medium, and Low.

- All categories show highly skewed distributions with several outliers, as indicated by the extended whiskers and scattered points.

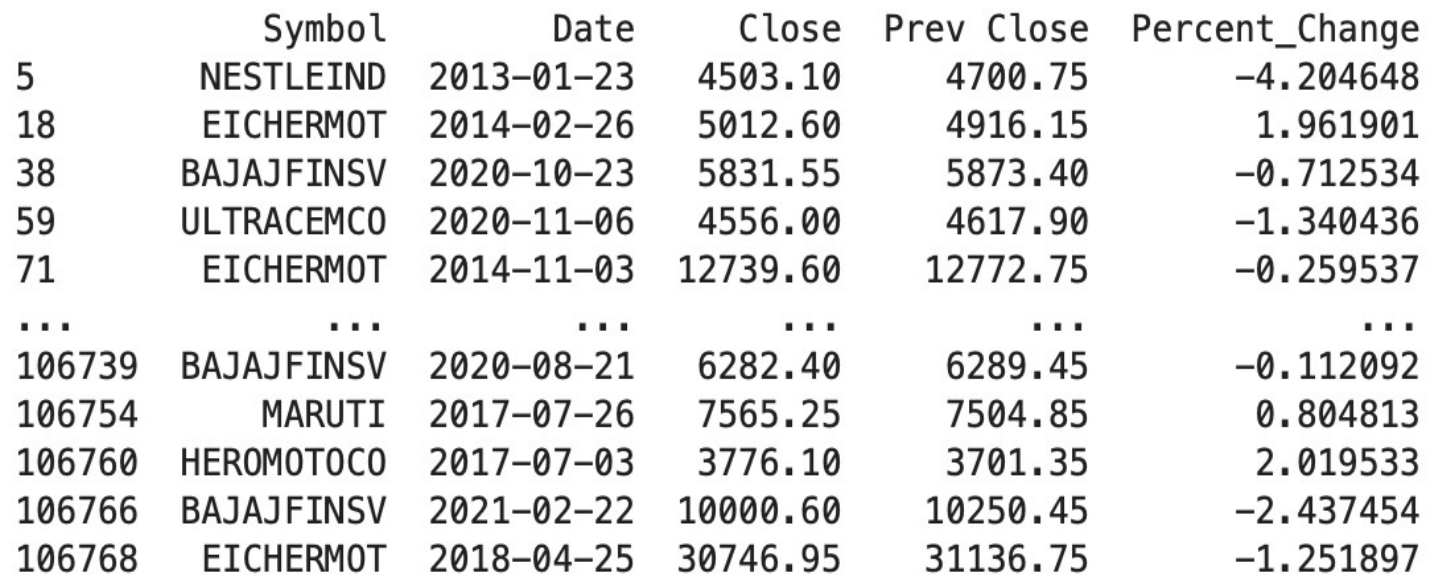
- Price movements are likely driven by other features, as the spread of Close prices across categories does not reveal significant differentiation.



Outliers Analysis



```
Number of outliers: 9935
Symbol
ASIANPAINT      248
BAJAJ-AUTO       42
BAJAJFINSV      931
BAJFINANCE      579
DRREDDY         315
EICHERMOT      1539
GRASIM          184
HEROMOTOCO      108
INFY             66
MARUTI         1293
NESTLEIND      1976
SHREECEM       1709
TITAN            66
ULTRACEMCO      879
dtype: int64
```

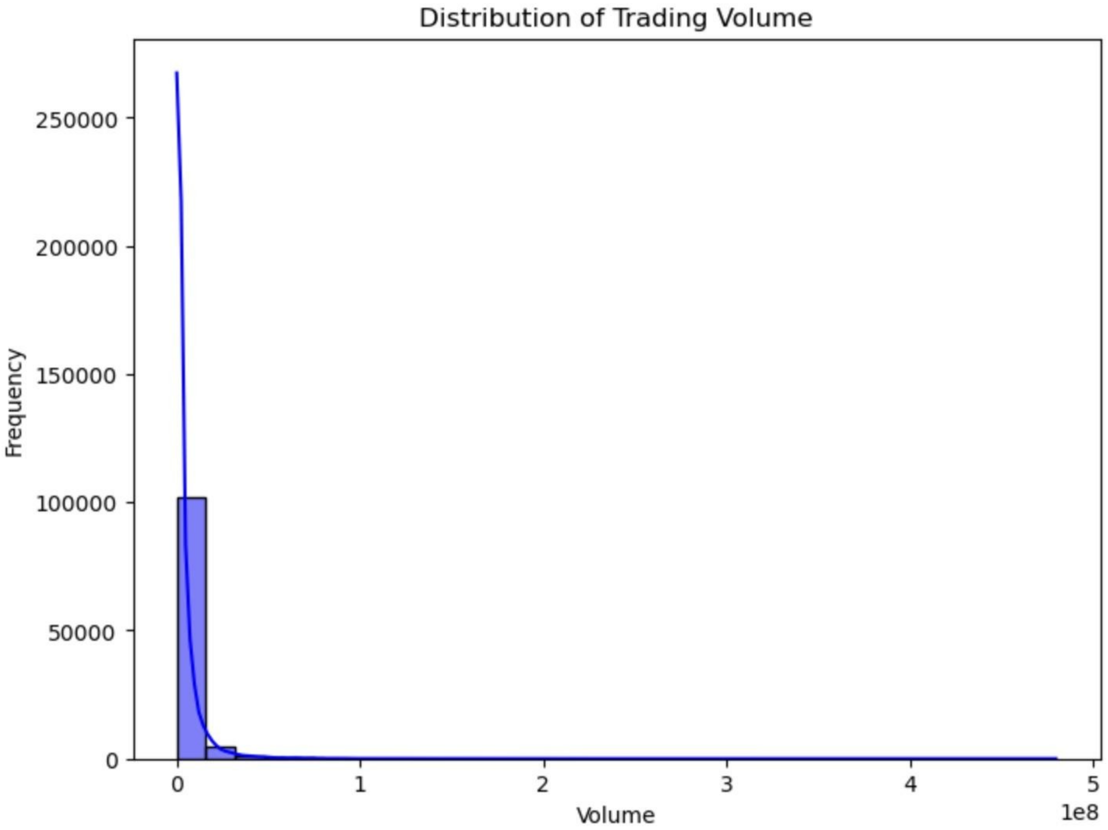


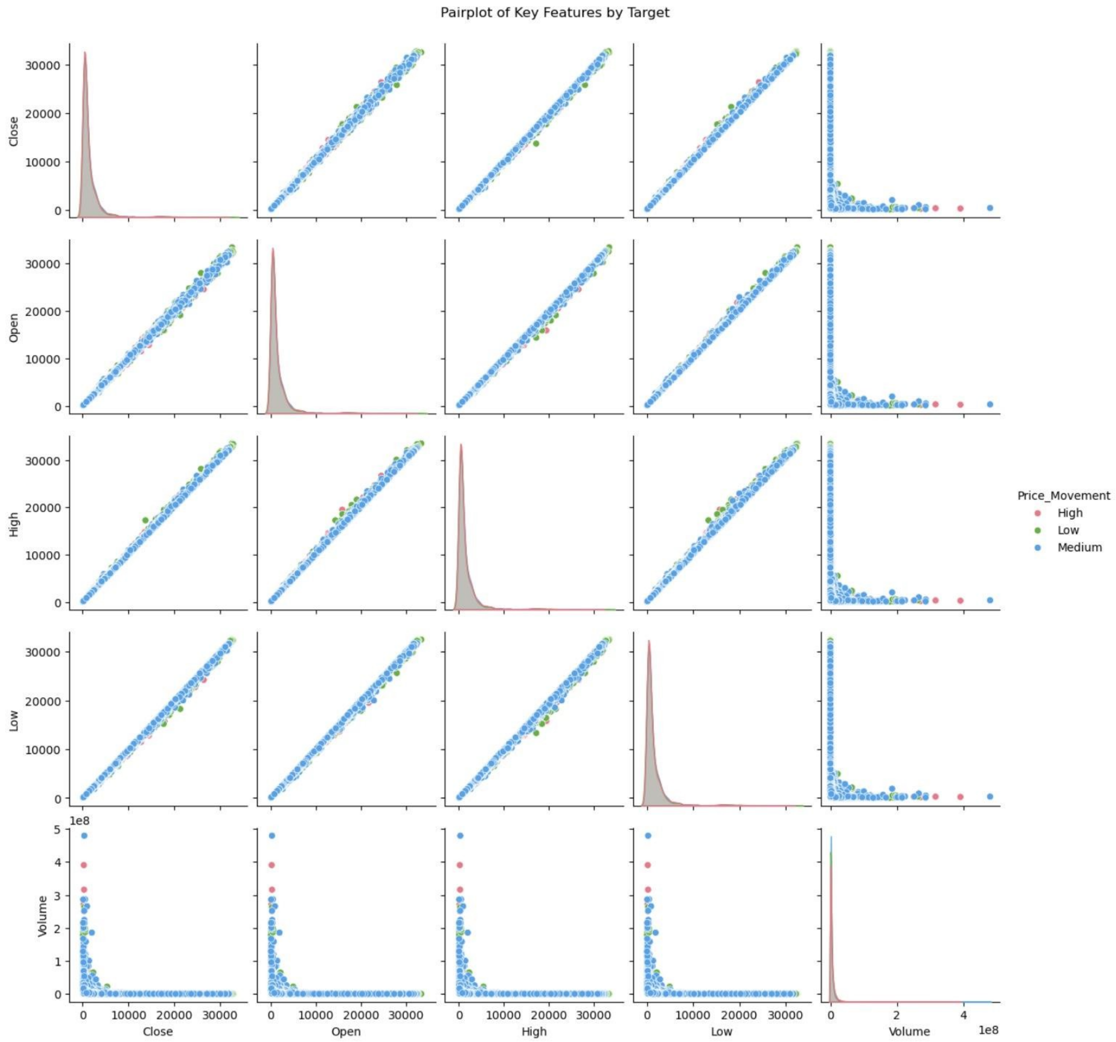
```
            Symbol        Date     Close  Prev Close  Percent_Change
5        NESTLEIND  2013-01-23   4503.10     4700.75       -4.204648
18       EICHERMOT  2014-02-26   5012.60     4916.15        1.961901
38      BAJAJFINSV  2020-10-23   5831.55     5873.40       -0.712534
59      ULTRACEMCO  2020-11-06   4556.00     4617.90       -1.340436
71       EICHERMOT  2014-11-03  12739.60    12772.75       -0.259537
...            ...         ...       ...         ...             ...
106739  BAJAJFINSV  2020-08-21   6282.40     6289.45       -0.112092
106754      MARUTI  2017-07-26   7565.25     7504.85        0.804813
106760  HEROMOTOCO  2017-07-03   3776.10     3701.35        2.019533
106766  BAJAJFINSV  2021-02-22  10000.60    10250.45       -2.437454
106768   EICHERMOT  2018-04-25  30746.95    31136.75       -1.251897
```

Distribution For Trading Volume

- The Volume distribution is heavily right-skewed, with most data points concentrated in lower ranges.

- Most data points lie in lower ranges, with a few outliers having very high trading volumes.





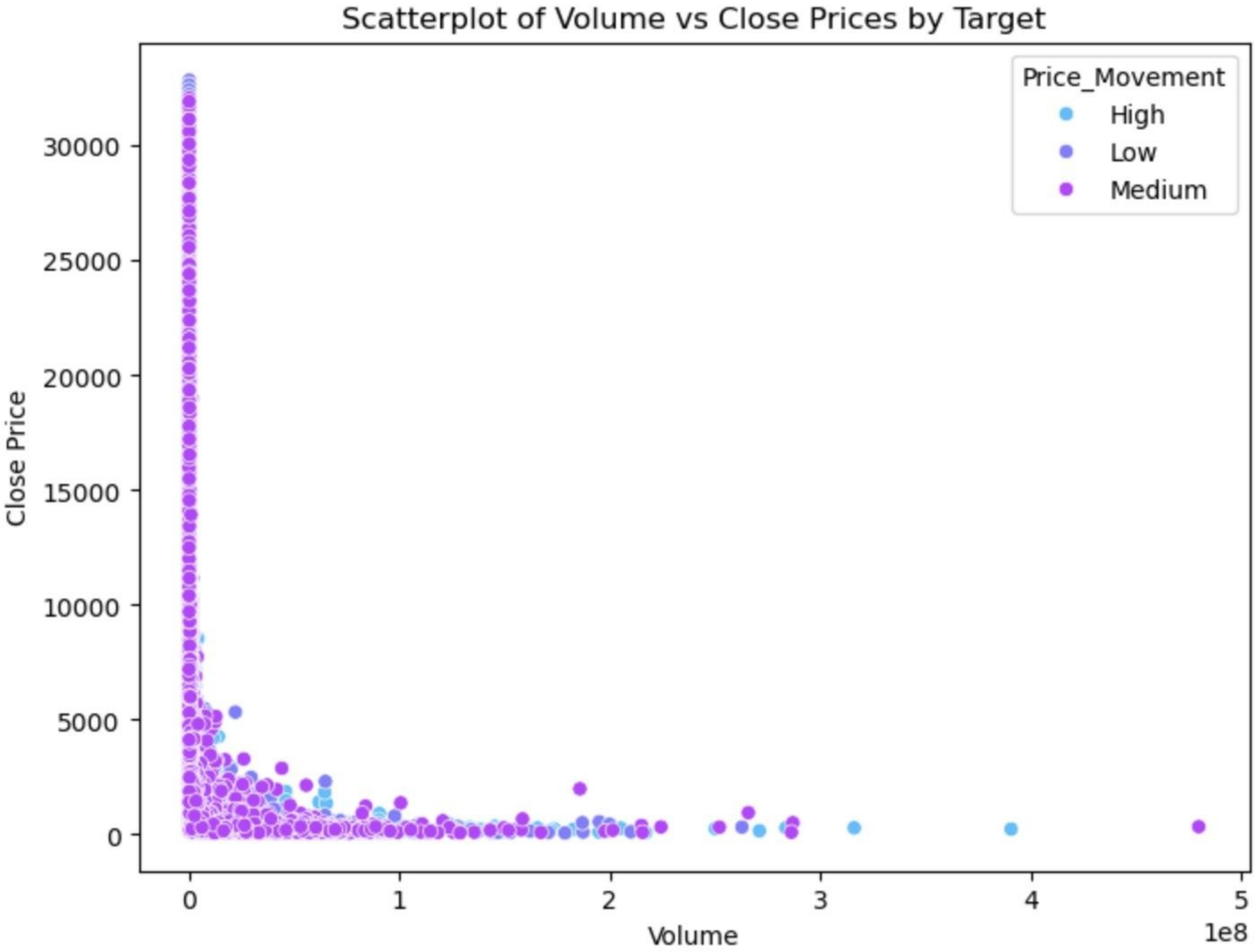
Pairplots

Pair plot of key features by Target

- Linear relationships among price based features like open, high, low, close.
- Class separability is not evident from pairwise plots.
- Volume shows a distinct pattern, with most data points clustered around low values irrespective of the target class.

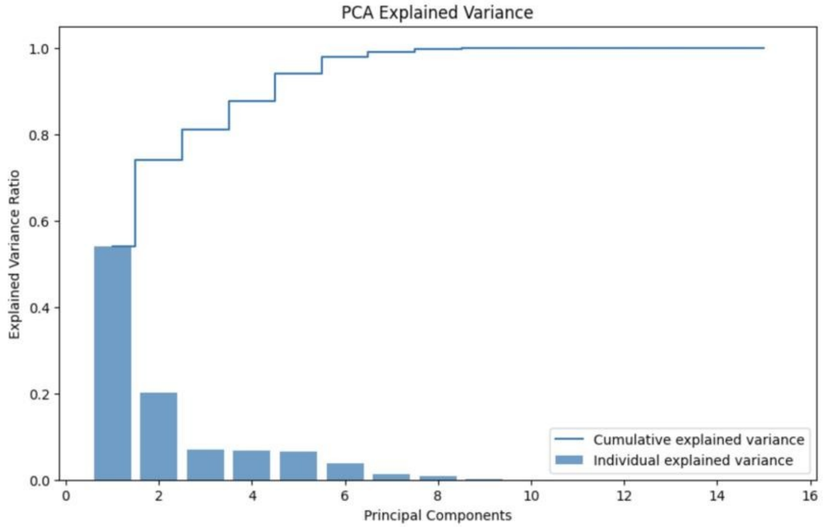
Scatterplot of Volume vs. Close Prices by Target

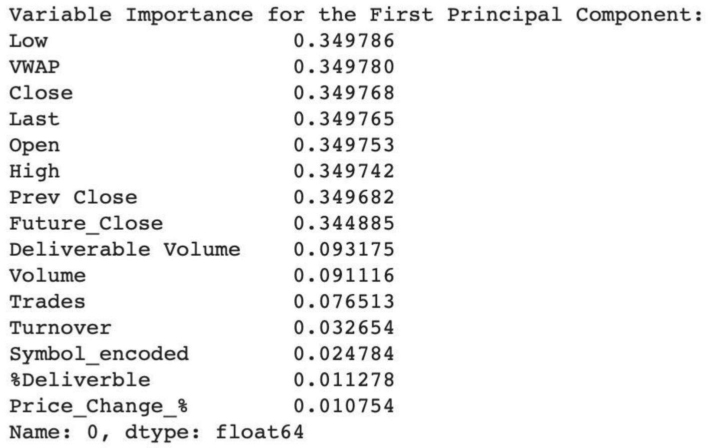
- No clear relationship between Close Price and Volume.
- Significant overlap among target classes.
- Outliers are present in both variables.



PCA Variable Importance And Explained Variance

- The first 5 components explain ~90% of the variance.
- Price-related variables contribute the most to the first component.
- Features like Volume and %Deliverable have lower contributions.

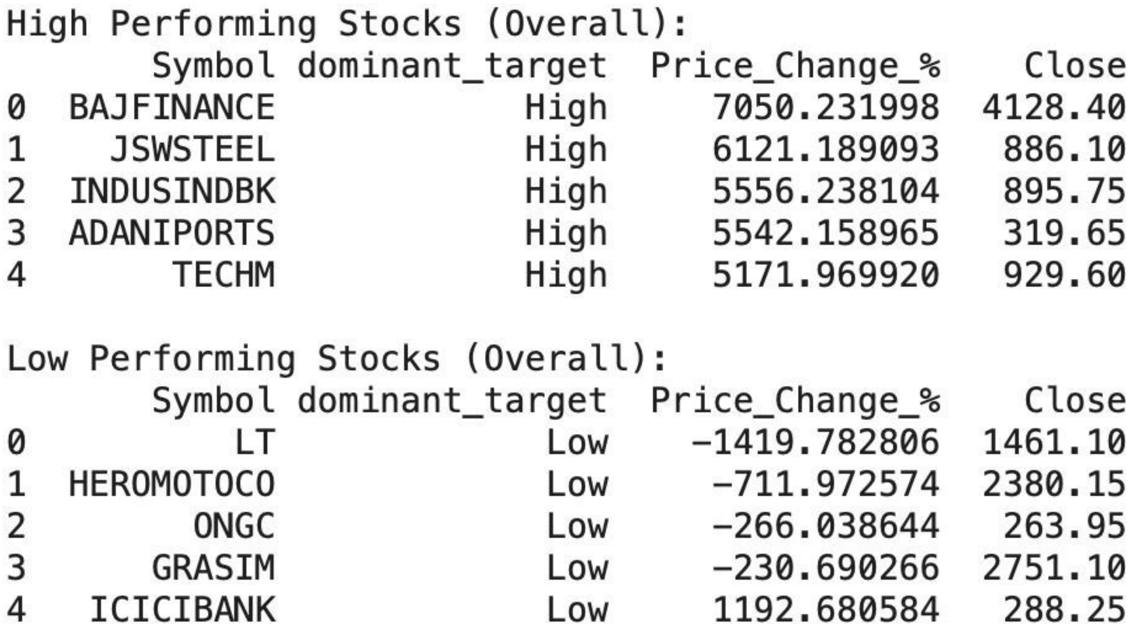




```
Variable Importance for the First Principal Component:
Low                   0.349786
VWAP                  0.349780
Close                 0.349768
Last                  0.349765
Open                  0.349753
High                  0.349742
Prev Close            0.349682
Future_Close          0.344885
Deliverable Volume    0.093175
Volume                0.091116
Trades                0.076513
Turnover              0.032654
Symbol_encoded        0.024784
%Deliverble           0.011278
Price_Change_%        0.010754
Name: 0, dtype: float64
```

High and Low Performing Stocks

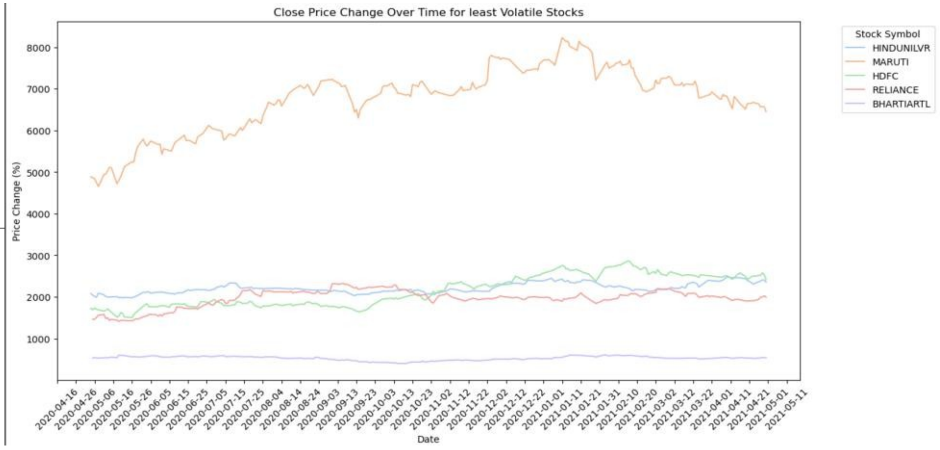
- High-performing stocks: BajFinance, JSWSteel, and TechM top the list with significant price changes (>5000%).
- Low-performing stocks: LT and Heromotoco show the largest declines in Price_Change_%.
- Recent closing prices for both high and low performers vary significantly, showing diverse market behaviors.

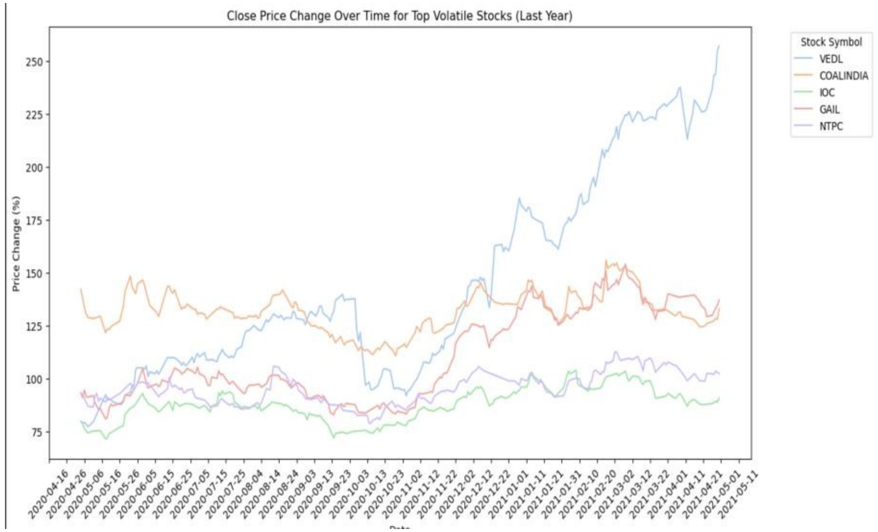


```
High Performing Stocks (Overall):
       Symbol dominant_target  Price_Change_%    Close
0  BAJFINANCE            High     7050.231998  4128.40
1    JSWSTEEL            High     6121.189093   886.10
2  INDUSINDBK            High     5556.238104   895.75
3  ADANIPORTS            High     5542.158965   319.65
4       TECHM            High     5171.969920   929.60

Low Performing Stocks (Overall):
       Symbol dominant_target  Price_Change_%    Close
0          LT             Low    -1419.782806  1461.10
1  HEROMOTOCO             Low     -711.972574  2380.15
2        ONGC             Low     -266.038644   263.95
3      GRASIM             Low     -230.690266  2751.10
4   ICICIBANK             Low     1192.680584   288.25
```

Volatility Analysis and Trends

- Most volatile stocks like Vedl, Gail, and CoalIndia exhibit sharp fluctuations in price trends.
- Least volatile stocks like ITC and TCS show steady and predictable growth.
- Volatility is a key factor in understanding stock performance and risk.





Models and Performance Evaluation

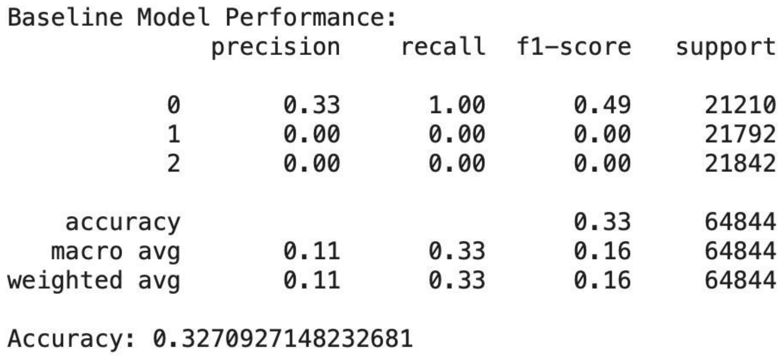
Baseline Model

- The baseline model predicts only the majority class (**High**) across all instances, failing to identify the **Low** and **Medium** categories.

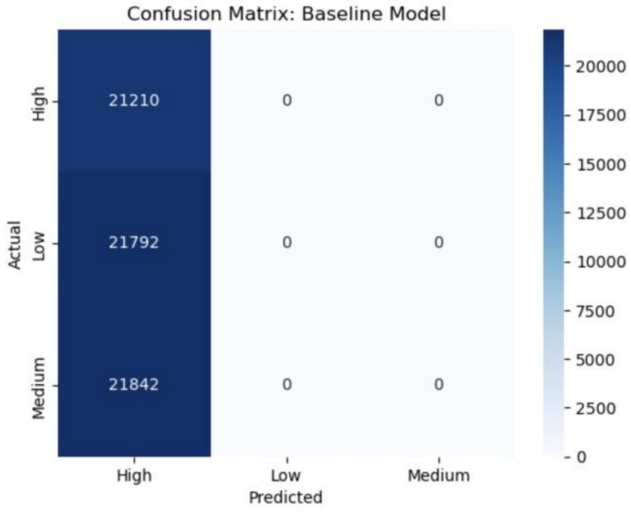
- Achieved accuracy of **32.7%**, which is equivalent to predicting the largest class and highlights the need for model improvements.

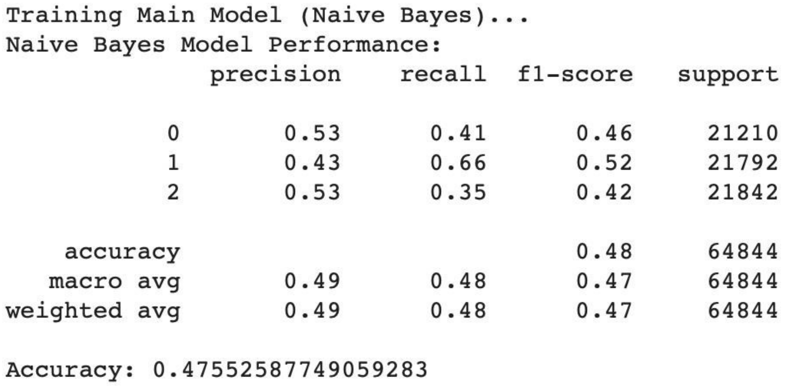
- **Precision, Recall, and F1-scores** for classes **1 (Low)** and **2 (Medium)** are zero, indicating no correct predictions for these categories.

- All predictions are concentrated in the **High** class, with no instances predicted for other classes, confirming the baseline model's inability to distinguish between categories.

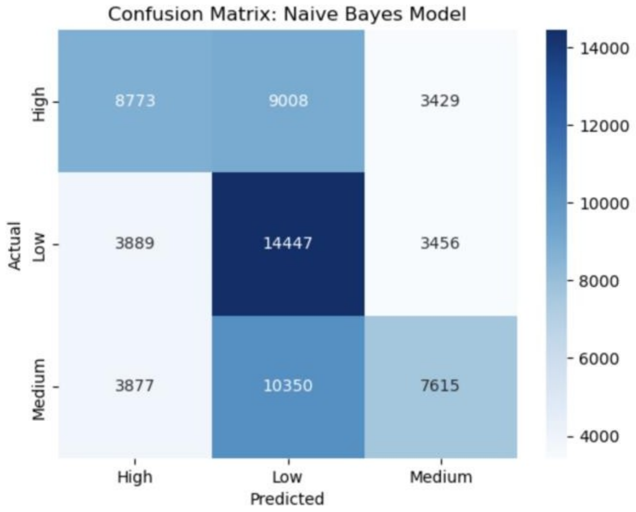


```
Baseline Model Performance:
              precision    recall  f1-score   support

           0       0.33      1.00      0.49     21210
           1       0.00      0.00      0.00     21792
           2       0.00      0.00      0.00     21842

    accuracy                           0.33     64844
   macro avg       0.11      0.33      0.16     64844
weighted avg       0.11      0.33      0.16     64844

Accuracy: 0.3270927148232681
```





```
Training Main Model (Naive Bayes)...
Naive Bayes Model Performance:
              precision    recall  f1-score   support

           0       0.53      0.41      0.46     21210
           1       0.43      0.66      0.52     21792
           2       0.53      0.35      0.42     21842

    accuracy                           0.48     64844
   macro avg       0.49      0.48      0.47     64844
weighted avg       0.49      0.48      0.47     64844

Accuracy: 0.47552587749059283
```

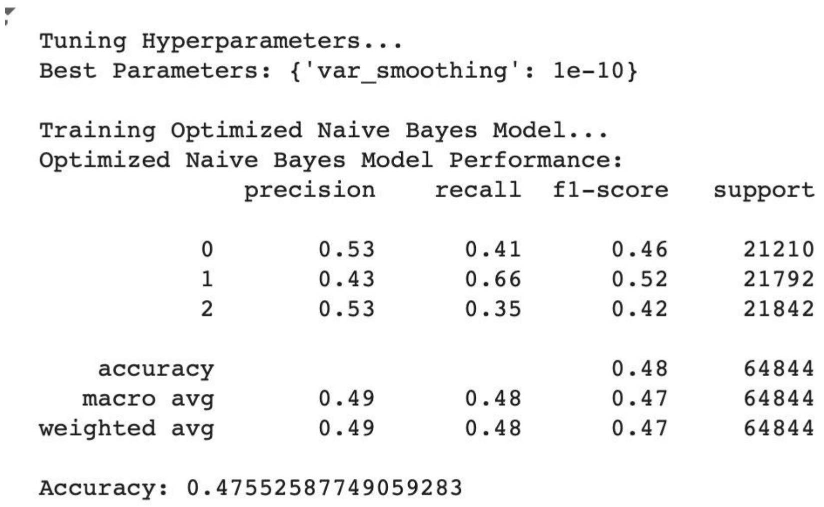


Naïve Bayes Model

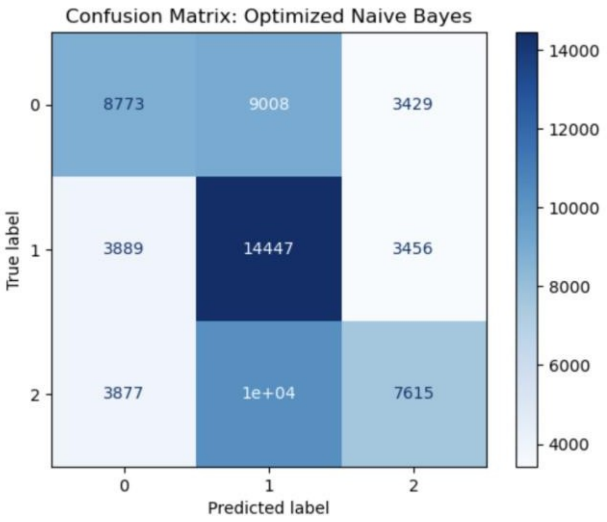
- Achieved an accuracy of **47.55%**, a noticeable improvement over the baseline model's performance.

- Performs relatively well for **Low** (class 1) with a recall of **66%**, indicating the model captures a significant portion of these instances.

- Macro and weighted averages for precision, recall, and F1-score are **49%**, reflecting the model's balanced but suboptimal performance.

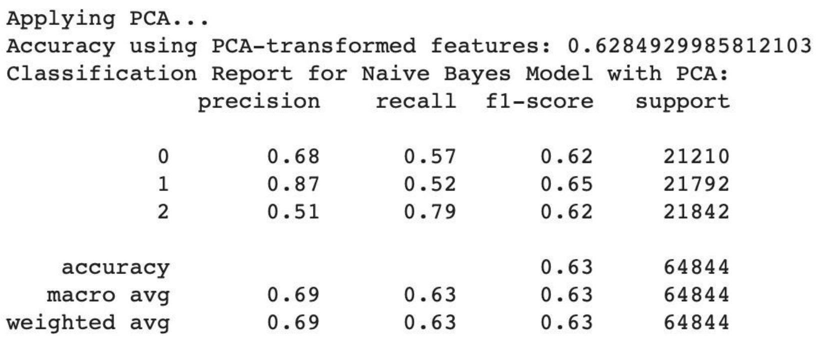


```
Tuning Hyperparameters...
Best Parameters: {'var_smoothing': 1e-10}

Training Optimized Naive Bayes Model...
Optimized Naive Bayes Model Performance:
              precision    recall  f1-score   support

           0       0.53      0.41      0.46     21210
           1       0.43      0.66      0.52     21792
           2       0.53      0.35      0.42     21842

    accuracy                           0.48     64844
   macro avg       0.49      0.48      0.47     64844
weighted avg       0.49      0.48      0.47     64844

Accuracy: 0.47552587749059283
```

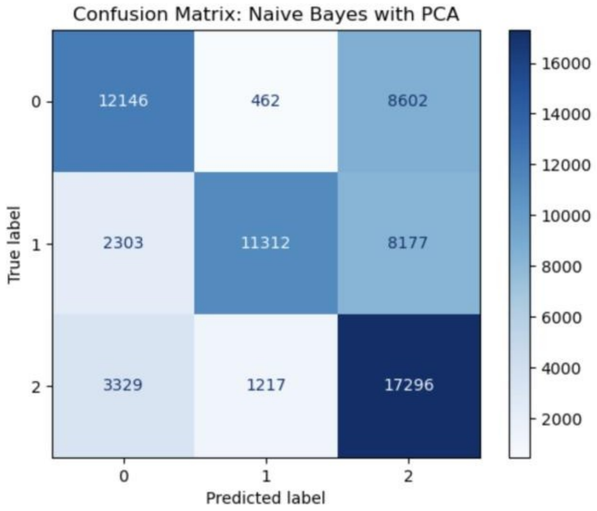


Naïve Bayes Model:
Hyper Parameter Tuning

- Best parameter found during tuning: **var_smoothing = 1e-10**
- Achieved an accuracy of **47.55%**, similar to the initial Naive Bayes model.
- Recall for **Class 1 (Low)** remains strong at **66%**, reflecting good detection of this category.
- **Misclassifications** are still prominent between **High** and **Low** or **Medium** categories, demonstrating the challenge of clearly separating adjacent classes.

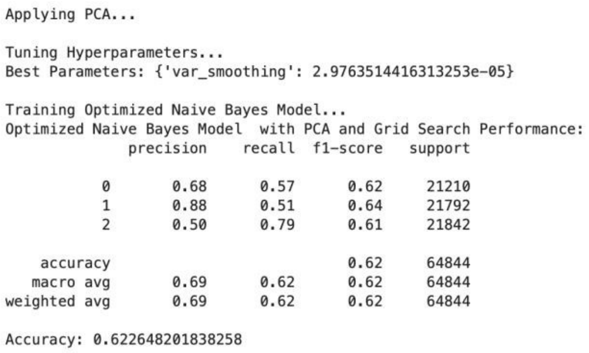


```
Applying PCA...
Accuracy using PCA-transformed features: 0.6284929985812103
Classification Report for Naive Bayes Model with PCA:
              precision    recall  f1-score   support

           0       0.68      0.57      0.62     21210
           1       0.87      0.52      0.65     21792
           2       0.51      0.79      0.62     21842

    accuracy                           0.63     64844
   macro avg       0.69      0.63      0.63     64844
weighted avg       0.69      0.63      0.63     64844
```

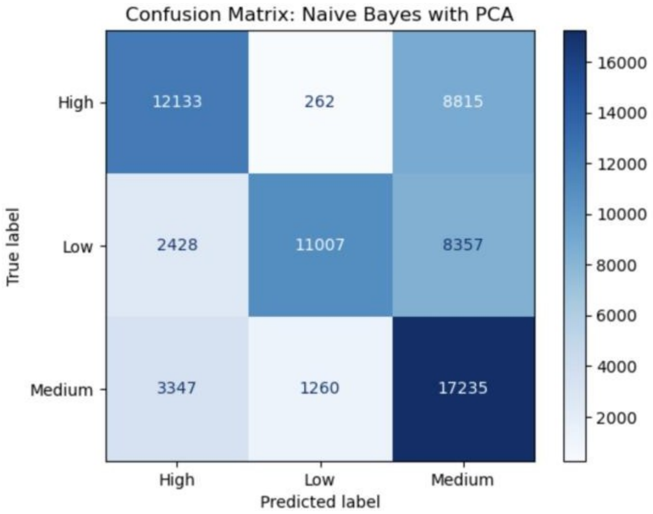


Naïve Bayes Model: PCA

- Principal Component Analysis (PCA) was applied to reduce feature dimensionality before training the Naive Bayes model.
- Achieved an accuracy of **62.84%**, demonstrating a significant improvement over the previous models.
- **Class 1 (Low)** shows the highest precision (**87%**) but lower recall (**52%**), indicating accurate but incomplete predictions.
- Significant misclassifications between **Class 0 (High)** and **Class 2 (Medium)**, indicating areas for further improvement.



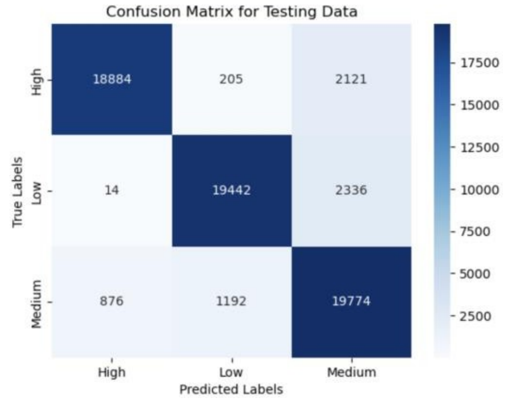
```
Applying PCA...

Tuning Hyperparameters...
Best Parameters: {'var_smoothing': 2.9763514416313253e-05}

Training Optimized Naive Bayes Model...
Optimized Naive Bayes Model  with PCA and Grid Search Performance:
              precision    recall  f1-score   support

           0       0.68      0.57      0.62     21210
           1       0.88      0.51      0.64     21792
           2       0.50      0.79      0.61     21842

    accuracy                           0.62     64844
   macro avg       0.69      0.62      0.62     64844
weighted avg       0.69      0.62      0.62     64844

Accuracy: 0.622648201838258
```

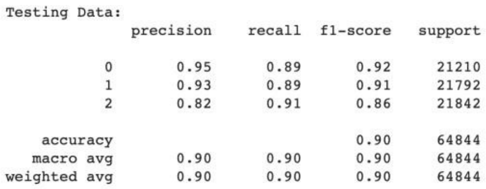


Naïve Bayes Model: Hyper Parameter Tuning and PCA

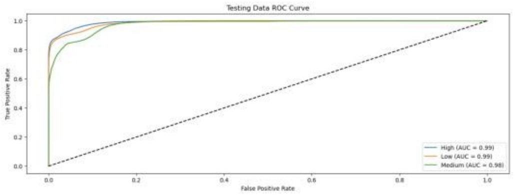
- PCA was applied to reduce feature dimensionality, retaining the most critical information for classification and tuning helped refine the model's ability to handle class boundaries effectively.
- Best parameter identified during tuning: **var_smoothing = 2.976e-05**.
- Achieved an accuracy of **62.26%**, slightly lower compared to PCA without tuning (**62.84%**).
- **Class 1 (Low)** retains high precision (**88%**) and relatively balanced recall (**51%**).
- Misclassifications remain evident, particularly between **Class 0 (High)** and **Class 2 (Medium)**.

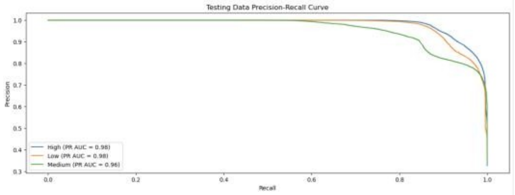
Support Vector Machine Model



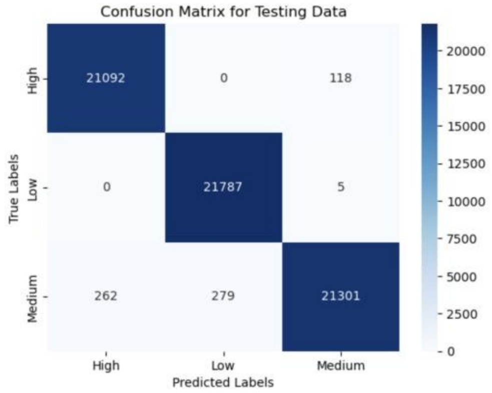


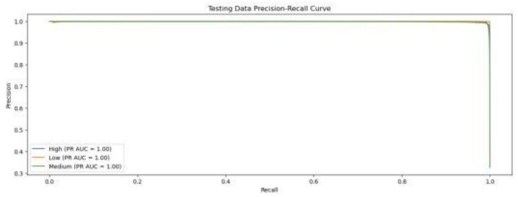
```
Testing Data:
              precision    recall  f1-score   support

           0       0.95      0.89      0.92     21210
           1       0.93      0.89      0.91     21792
           2       0.82      0.91      0.86     21842

    accuracy                           0.90     64844
   macro avg       0.90      0.90      0.90     64844
weighted avg       0.90      0.90      0.90     64844
```

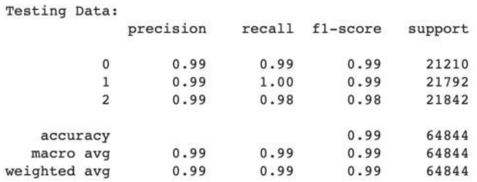




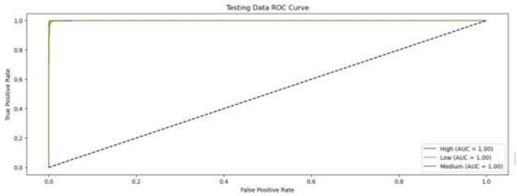
- The model achieved an **accuracy of 90%**, demonstrating strong generalization and effective classification.
- **Class 0 (High)** and **Class 1 (Low)** exhibit high precision (**95%** and **93%**) and recall (**89%** each). **Class 2 (Medium)** shows slightly lower precision (**82%**) but excellent recall (**91%**), indicating effective identification despite minor challenges.
- Minor misclassifications are observed between **Class 0 (High)** and **Class 2 (Medium)**.
- AUC scores for all classes range between **0.98–0.99**, validating the model’s ability to distinguish among categories effectively.
- Precision-Recall metrics are also robust across all classes.







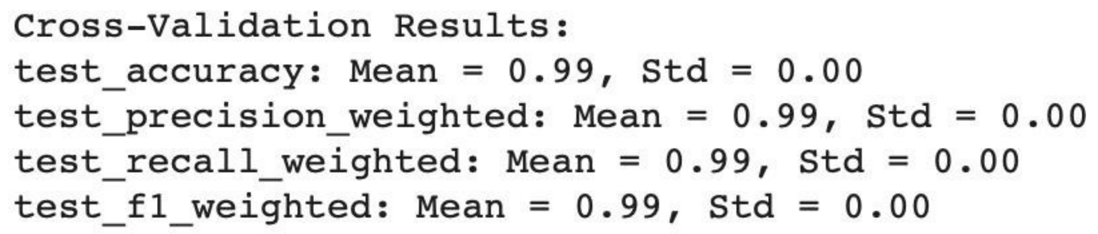
```
Testing Data:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     21210
           1       0.99      1.00      0.99     21792
           2       0.99      0.98      0.98     21842

    accuracy                           0.99     64844
   macro avg       0.99      0.99      0.99     64844
weighted avg       0.99      0.99      0.99     64844
```



Logistic Regression Model

- The model achieved an outstanding **accuracy of 99%**, demonstrating exceptional generalization capabilities.
- Precision, recall, and F1-scores for all classes are near-perfect at **0.99**, reflecting the model's ability to handle imbalanced data effectively.
- Negligible misclassifications for **Class 0 (High)** and **Class 1 (Low),** whereas minor confusion observed between **Class 2 (Medium).**
- AUC scores of **1.00** across all classes validate the model's reliability in distinguishing between categories effectively.
- Near-perfect precision-recall curves indicate a well-calibrated model with consistent performance across varying thresholds.

Logistic Regression Model: Cross Validation



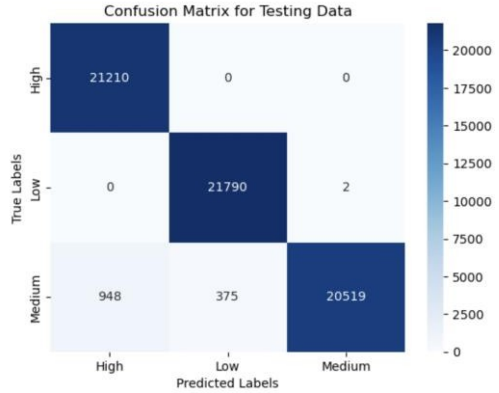
```
Cross-Validation Results:
test_accuracy: Mean = 0.99, Std = 0.00
test_precision_weighted: Mean = 0.99, Std = 0.00
test_recall_weighted: Mean = 0.99, Std = 0.00
test_f1_weighted: Mean = 0.99, Std = 0.00
```

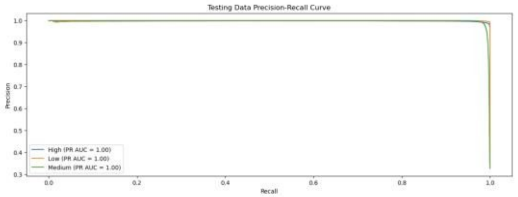
- Cross-validation confirms the model's stability with a **mean accuracy of 99%** and negligible standard deviation across folds.

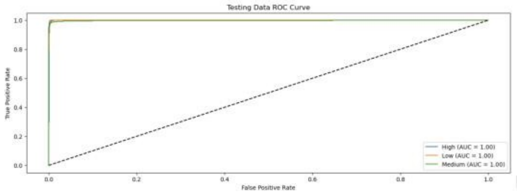
- Precision, recall, and F1-scores maintain a mean of **0.99** with near-zero variation, showcasing consistent performance.

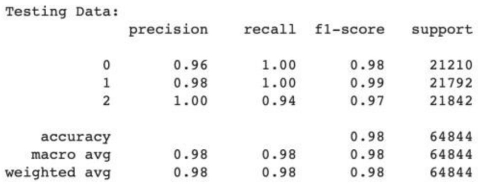
- These results validate the model's robustness, making it a strong candidate for real-world deployment with minimal risk of overfitting.

Lasso Logistic Regression Model





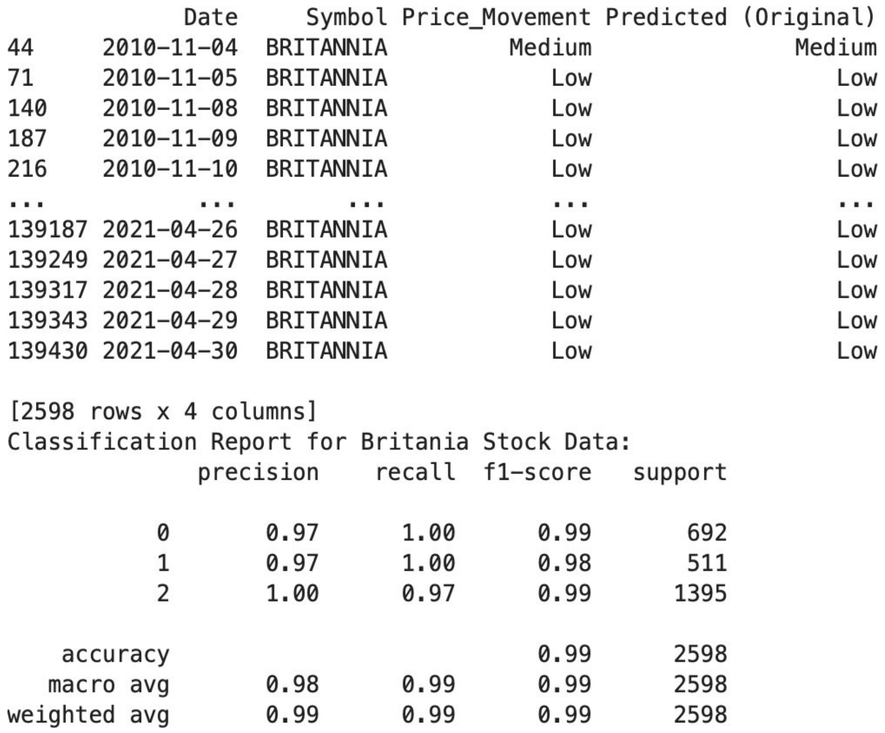




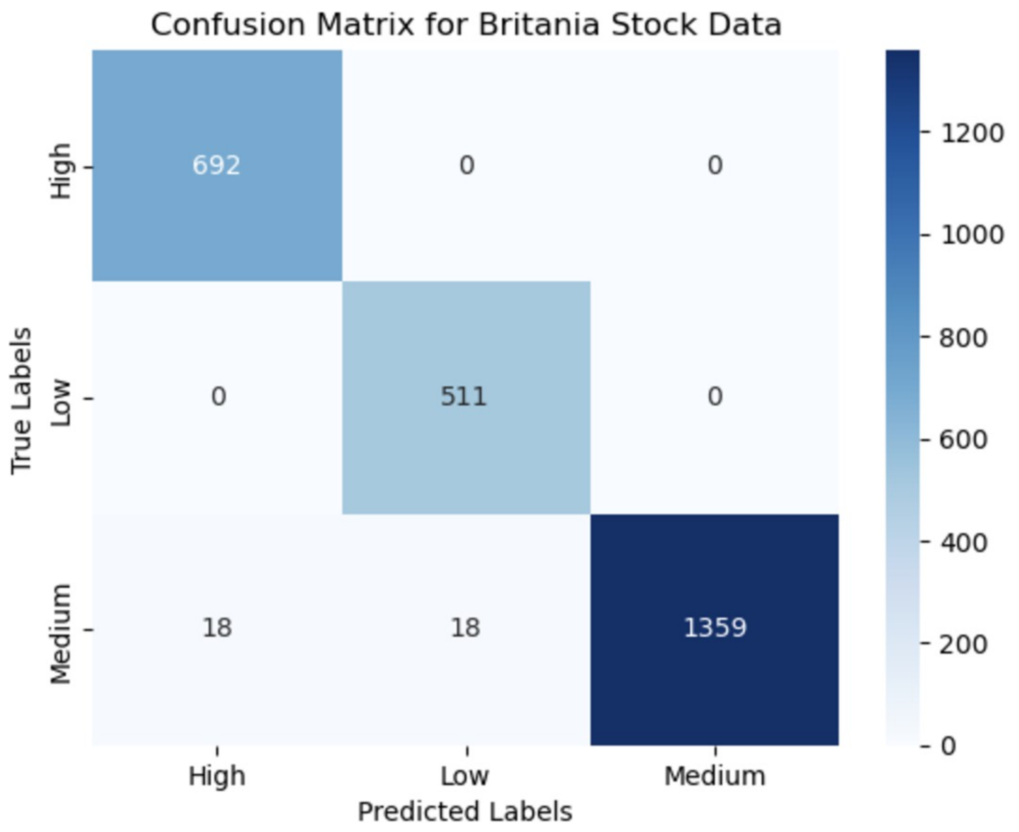
```
Testing Data:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     21210
           1       0.98      1.00      0.99     21792
           2       1.00      0.94      0.97     21842

    accuracy                           0.98     64844
   macro avg       0.98      0.98      0.98     64844
weighted avg       0.98      0.98      0.98     64844
```

- The model achieved an **accuracy of 98%**, demonstrating strong predictive performance and generalization capabilities.
- **Class 0 (High)** and **Class 1 (Low)** exhibit near-perfect recall (**100%**) and high F1-scores (**0.98** and **0.99**, respectively).
- Negligible misclassifications for **Class 0 (High)** and **Class 1 (Low),** whereas minor misclassification for **Class 2 (Medium)**, with some overlap into the other categories.
- AUC scores of **1.00** across all classes confirm the model's exceptional ability to distinguish between categories effectively.
- Precision-Recall metrics also remain strong and consistent,

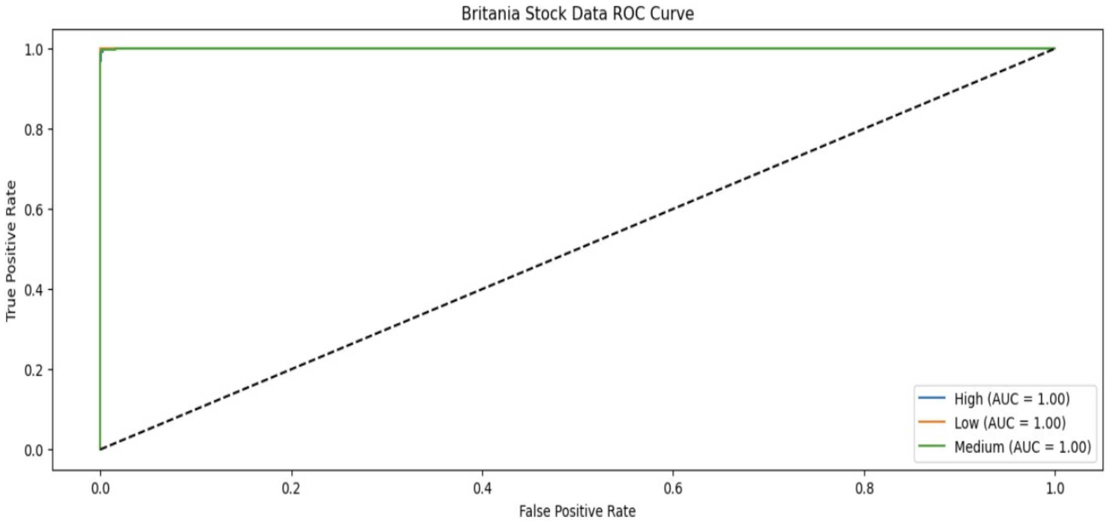
Does Logistic Regression perform well on completely new stock – Britannia ?

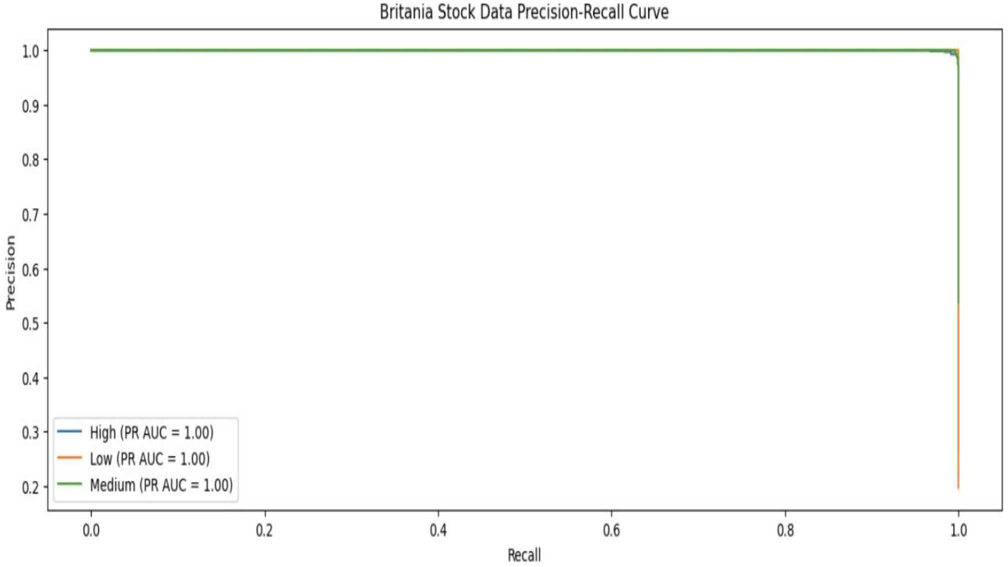


```
            Date     Symbol Price_Movement Predicted (Original)
44     2010-11-04  BRITANNIA         Medium               Medium
71     2010-11-05  BRITANNIA            Low                  Low
140    2010-11-08  BRITANNIA            Low                  Low
187    2010-11-09  BRITANNIA            Low                  Low
216    2010-11-10  BRITANNIA            Low                  Low
...           ...        ...            ...                  ...
139187 2021-04-26  BRITANNIA            Low                  Low
139249 2021-04-27  BRITANNIA            Low                  Low
139317 2021-04-28  BRITANNIA            Low                  Low
139343 2021-04-29  BRITANNIA            Low                  Low
139430 2021-04-30  BRITANNIA            Low                  Low

[2598 rows x 4 columns]
Classification Report for Britania Stock Data:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       692
           1       0.97      1.00      0.98       511
           2       1.00      0.97      0.99      1395

    accuracy                           0.99      2598
   macro avg       0.98      0.99      0.99      2598
weighted avg       0.99      0.99      0.99      2598
```

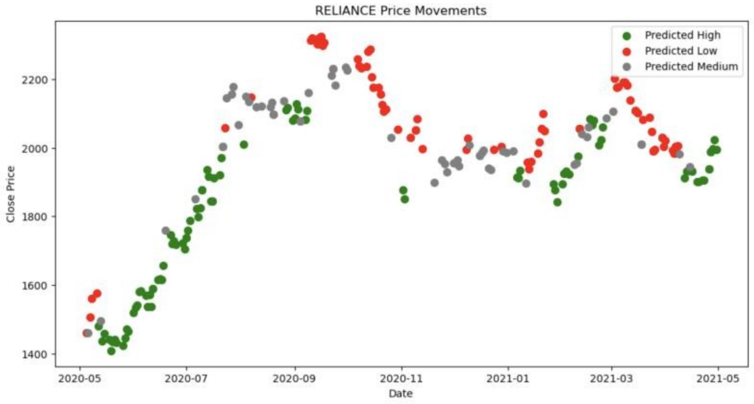


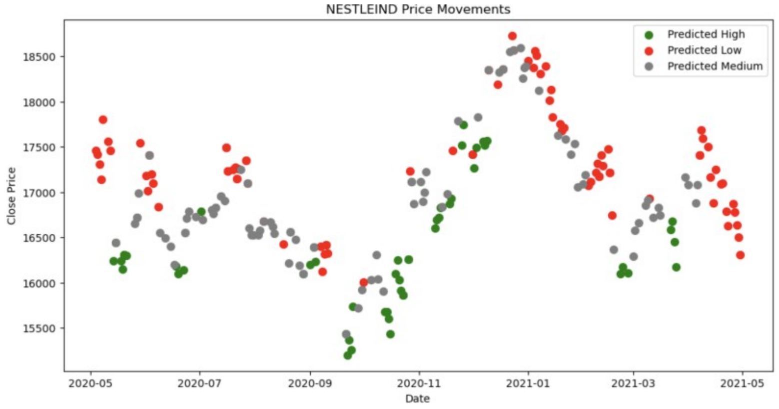
Does Logistic Regression perform well on completely new stock – Britannia ?





Visualization of Price Movements Based on Models Classification (For Stocks: Reliance and Nestle)





- The predicted price movements of stocks are visualized over time, categorized as High, Low, or Medium based on the model predictions.
- Scatter plot shows predicted price movements classified as High, Low, and Medium.
- Visualizations validate the model's ability to capture trends in stock price behavior.

Conclusio n

The project showcased the importance of iterative model refinement, careful evaluation, and validation in building reliable and interpretable models for complex classification tasks.

Logistic Regression and Lasso Logistic Regression demonstrated exceptional accuracy (98-99%), with balanced precision, recall, and F1-scores across all classes.

Robust ROC and Precision-Recall metrics validated the reliability of the selected models.

Logistic Regression with Lasso is well-suited for deployment in real-world applications due to its scalability and robustness.

Thank You