**Stock Price Prediction Project**

**1.** Project Background and Objectives

The aim of this project is to use historical stock price information (opening, closing, high, low, trading volume) in order to generate a classification model that predicts if a stock is going to go up or down in the coming time interval. It trains on the historical price signals and forecasts direction of the price movement for upcoming time intervals.

This project shows that ML algorithm (Logistic Regression model) is implemented in financial data analysis to predict stock market trend and also analyzes the performance of this algorithm.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2.**  Dataset

The dataset we will be using in this project has these fields:

• date: Date

• symbol: Stock name (e.g., stock code)

• open: Opening price

• close: Closing price

• low: Lowest price

• high: Highest price

• volume: Trading volume

In order to make stock market predictions, the data was preprocessed and cleaned, for example:

• Taking Care of Missing Data: Rows where data was missing were removed.

• Date Conversion: Date field was remapped to datetime for the sake of order.

• Feature Engineering:

o MA10,MA20: Reconstructed the 10- and 20-day MAs of last price to flatten prices and identify patterns.

o Price Change Rate (price\_change): Calculated the daily rate of price change to track stock price movement.

o 7-Day Volume Moving Average (volume\_MA7): Displayed the trend of volume.

At last, we created a target variable direction when the next day’s closing price was greater than the current day’s closing price:

• 1: Stock price increases

• 0: Stock price decreases

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3.**  Model and Techniques

For the classification, Logistic Regression model was applied to forecast the movement of the stock price. Logistic Regression is an older classification algorithm, it can be used for binary classification problems. In this project, it tells you if a stock will go up (1) or down (0) tomorrow.

**3.1** Model Tuning: GridSearchCV

GridSearchCV was used for hyperparameter tuning of Logistic Regression model to get optimal performance. Grid search gave the following parameters as the best choices:

• C: Regularization function (factor = 0.1)

• solver: Algorithm to solve (chosen = liblinear)

• penalty: Regularization scheme (chosen = l2)

GridSearchCV cross-validated all hyperparameters, and finally chose the optimal model configuration.

**3.2** Techniques Used

• Data Preprocessing: Date Conversion, missing values, feature engineering for data quality and model stability.

• Logistic Regression Model: Prediction of binary classification, training the model and predictions on test set.

• Hyperparameter Tuning: Tweaked the hyperparameters of the model through GridSearchCV for better prediction.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4.**  Evaluation Metrics

Here are the metrics we used to measure the performance of the model:

• **Correctness:** Denotes how many of the samples are correctly predicted among the total samples. Calculated using the accuracy\_score function:

Correctness = Proper predictions / Total predictions.

The project was an accuracy of 0.79 which means that about 79% of predictions were accurate.

• **F1 Score:** F1 score is the relative precision and recall that are the harmonic mean (usually helpful when you have unbalanced datasets). Since investors are in the business of picking up or pulling out of a stock, calculating the rise or falling accurately is important in stock market prediction, so the F1 score becomes a very significant measure. The project got an F1 score of 0.81 which means that the model was highly accurate and recallable for stock movements.

• **Confusion Matrix:** Shows model classification success and failure matrix of the model. It plots the spread of true vs. predicted values to measure the performance of the model on each class:

o True Positives (TP): Expected 1 and actual 1 (prediction correct)

o True Negatives (TN): Predicted 0 and true 0 (Wrong prediction)

o False Positives (FP): Expected 1 but actual 0 (wrong prediction)

o False Negatives (FN): Expected 0 but got 1 (wrong prediction).

• **Residual plot:** The prediction error (error) was defined as the difference between the measured and calculated direction and a residual plot was created. Residual plot: The residual plot allows us to visualize model bias and how prediction errors are distributed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5.** Model Results and Visualization

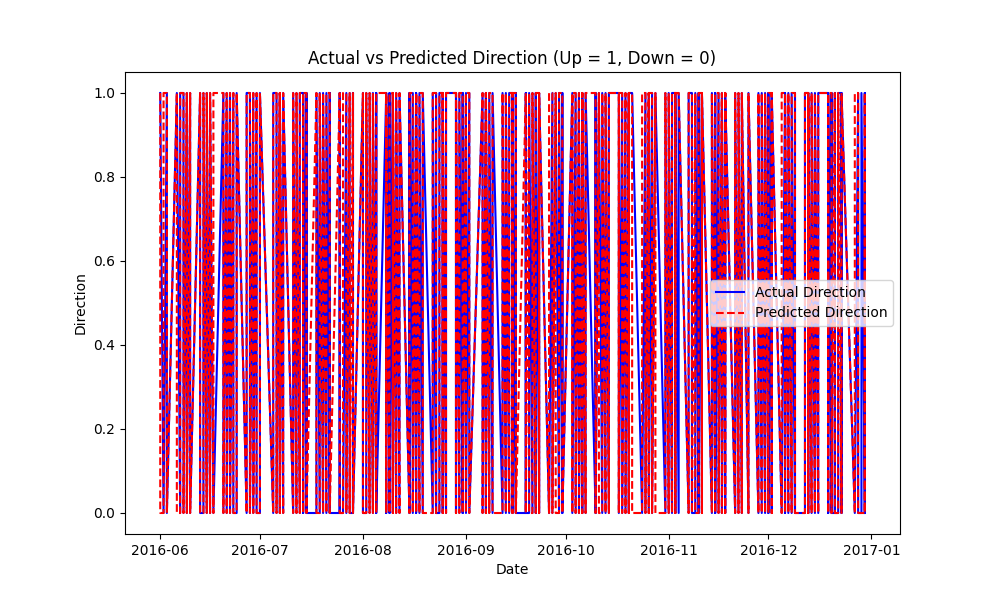
The comparison of model predictions and the real stock movement directions was visualized by way of visualization. The following plots were generated:

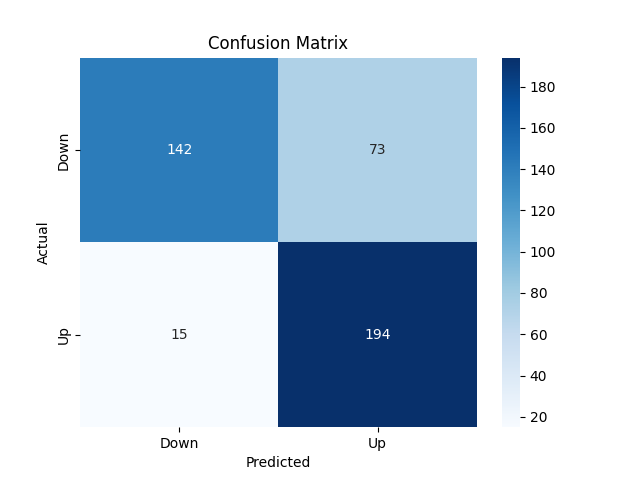
• **Actual vs. Predicted Directions plot:** Using actual stock price and predicted movements shows the model to be correct.

• **Confusion Matrix Plot:** Gives a plot of the classification performance of the model.

• **Residual Plot:** Shows prediction error and any model biases.

1. \*\*Actual vs. Predicted Directions Plot\*\*: Shows us the time series relationship between the stock movement direction and the model predictions to see the predictive capabilities of the model.

  
  
2. \*\*Confusion Matrix\*\*: Displays model’s classification result (from real-world and predicted value), which visually displays correct and false predictions.

  
  
3. \*\*Residual Plot\*\*: Displays the model prediction error and the residual plot shows a detailed overview of model performance.

