Project Report

Stock Price Prediction

- By Preeti

Abstract

The task in question is to predict closing stock price of NVIDIA Corporation (NVDA) based on historical data from Yahoo Finance. AdvertisementOpening, high, low, closing, adjusted closing stock price, volume, etc are taken from daily data set. After cleaning up data (formatting of dates and volumes), adding new features like average price, date components (day, month, year) were made to the data set for further analysis to find patterns or correlations. Current regression models used were Linear Regression, Random Forest Regressor, Support Vector Regressor (SVR) and regularized regression models like Ridge, Lasso and ElasticNet which were divided into training and testing sets to take the feature normalization in account of sensitivity to feature magnitude. Evaluation was made using R2 score of test set and the results are as follows:

- Linear Regression, $R^2 \approx 0.85$
- Random Forest, $R^2 \approx 0.79$
- SVR, $R^2 \approx 0.44$
- Ridge Regression, $R^2 \approx 0.85$
- Lasso Regression, $R^2 \approx 0.85$
- ElasticNet Regression, $R^2 \approx 0.85$

List of figures

List of Tables

List of Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
SVR	Support Vector Regressor
CSV	Comma-Separated Values
EDA	Exploratory Data Analysis
R^2	Cofficent of determination(R-square)

Table of contents

Chapter – 1

Introduction to Project

1.1 Overview

Stock market forecasting is problematic due to the unpredictable and volatile nature of financial markets. Here an attempt is made at developing a realistic solution for stock price prediction based on historical market data and machine learning algorithms. The use of automated data collection method by web scraping and predictive modelling technique combined allows for providing insights on future stock prices.

In the first part of the system, data is obtained in real time from Yahoo Finance (with a help of Selenium and BeautifulSoup), is cleaned and structured using Python's data processing tools, after which machine learning models like Linear Regression and K-Nearest Neighbors are trained on the data to anticipate future stock prices.

The hoped for purpose of this project is a (basic) framework that provides an intuitive but usable way to test the use of data science in analyzing and forecasting stock market behavior. This is a groundwork for future investment in financial analytics that can be further evolved with more difficult models or a real-time data feed in future iterations.

1.2 User Requirement Analysis

Users of stock prediction systems have an expectation that the system can handle structured financial data accurately and efficiently. In the typical stock prediction system, many column are defined like Date, Open, High, Low, Close, Adj Close and Volume. The data should be collected automatically from well-known web sources so that no manual effort is required to download it. Once the data have been collected the system should pre-process the data and use it to forecast the prices of future stocks. Users have another expectation that the system should offer a visual display — from which the user can compare actual prices with the predicted ones. The interface should also allow input of stock names or symbols, without any code effort being required. Finally, the user needs to have an indicator of model performance (RMSE or R2 score) so that he can evaluate the reliability of the software before making any market-related assumptions or decisions.

1.3 Feasibility Study

The feasibility of Stock Price Prediction :

technical, operational.

1. Technical Feasibility:
    - The project utilizes established technologies such as Python and libraries like Scikit-learn, Pandas, and numpy, which are widely supported and reliable.

- The machine learning model, linear Regression, LASSO, RIDGE, Random Forest Regressor, SVR, ElasticNet.

2. Operational Feasibility:
    - The interface design ensures usability, requiring minimal technical expertise from users to interact with the system.
    - The prediction process is streamlined to deliver quick and actionable results, making it suitable for real-time use in stock price prediction.

Chapter – 2

Literature Review

Predicting stock prices using machine learning has been an increasingly popular topic, owing to its potential applications in financial decision making. Extensive research has been undertaken on various statistical as well as learning based techniques for prediction of market movements by using historical price data. In this study, Patel et al. (2015) used NSE India stock data to build various models which included random forest, support vector machines (SVM) and artificial neural networks (ANN) for prediction, which achieved maximum accuracy with Random Forest. Another work by Kumar and Ravi (2016) offered a comparative analysis of different machine learning techniques, where Decision Tree classifiers showed an accuracy of 84. 3%, while K-Nearest Neighbors showed about 80. 2% average accuracy.

Some work has also involved combining technical indicators (e. g. moving averages) with deep learning models (e. g. LSTM networks) and achieving near-91. 2% prediction accuracy in short-term trend forecasting. Deep learning methods tend to perform well, but they typically take much more computational resources and much larger data sets to train.

Contrary to common wisdom, not all lightweight and rapid deployment algorithms, will still use Linear regression models or KNN. The interpretability and efficiency of linear regression models have a good impact on the accuracy (from 60% to 75% depending on feature selection and dataset volatility.) Such models are good for academic applications and basic systems where interpretability and ease of implementation are secondary to complex tuning requirements.

2.1 Comparison

Paper No.	Paper Title	Dataset Source	Algorithms	Attributes Used	Accuracy (%)	Highest Performing Algorithm
1	Predicting Stock Price	NVDA	Linear Regression,LASSO	7	85	LASSO
2	Prediciting Stock Price	NVDA	LASSO,RIDGE	7	85	LASSO
3	Prediciting Stock Price	NVDA	Random forest Regressor,SVR	7	79	Random forest
4	Predicting Stock Price	Nvda	Linear Regression, Elasticnet	7	85.5	Linear Regression

2.2 Objectives of Project

The overall goal of this project is to build a machine learning system that uses historical stock market data to predict the future closing price of NVIDIA Corporation (NVDA). What this will include:

• Collect and prepare historical stock data (Open, High, Low, Close, Adjusted Close, Volume); • Perform exploratory data analysis to discover trends, correlations and patterns in the data; and • Integrate various regression models.

• Features relevant to engineering use date components (day, month, year) and average price.

• Training and evaluating various regression models, such as ElasticNet, Ridge, Lasso, Support Vector Regressor (SVR), Random Forest Regressor, and Linear Regression.

• Utilizing evaluation metrics like the R2 score to assess the performance of these models.

• Finally — using the best performing model to determine the closing price for the next day on the latest data

Chapter – 3

Exploratory Data Analysis

3.1 Dataset

The dataset used for this analysis, "NVDA Dataset," consists of 250rows and 7 attributes. The primary target variable is **opening price of following day**. The dataset includes various features such as **Date, Opening price, Closing Price, volume, High, Low** and lifestyle factors like **Market behaviour** and **Unexpected events**.

The dataset was sourced from publicly available NVDA datasets, such as yahoo finance,Kaggle, ensuring that the data is comprehensive enough for performing classification tasks. Initially, there were no missing values or duplicates in the dataset, making it suitable for analysis without additional imputation steps.

The dataset used for analysis was obtained from NVDA dataset, which is publicly available in CSV format. Given that the dataset is already structured and clean, there was no need for data scraping.

3.2 Exploratory Data Analysis and Data Visualizations

The **NVDA Dataset** was carefully analysed to identify patterns, relationships, and potential issues. The dataset contains 250entries with 7 attributes, including both numerical and categorical data. One of the first steps in the EDA was to visualize the distribution of numerical features such as **Closing Price, Opening price, Volume with Date**. Boxplots and histograms were used to visualize these features. The histograms revealed that several features, such as **Daily return**, were highly skewed, while boxplots helped identify outliers, particularly in features like **Heart Rate** and **Triglycerides**. This analysis helped us realize the importance of handling outliers, which could otherwise impact the model's predictive accuracy.

To understand the relationships between categorical variables and the target variable,

First, we upload the dataset of NVDA. After that , we see the total rows and columns in data. We measure the mean , median, mode, deviation. After that we show the relation of closing price over time and corelation with each attribute. Then, we split the data sets in two parts: training and testing . Train the models with various models like linear regression ,random forest regressor, SVR. After training we do testing to see the model performance and predicit the following day closing price.

Chapter – 4

Methodology

4.1 Introduction to Languages

In this project, the **Python** programming language was selected for both backend development and machine learning model implementation. Python is widely used in data science and machine learning due to its simplicity, extensive libraries, and strong community support. The main libraries used in this project include:

- **Pandas**: For data manipulation and preprocessing.
- **Matplotlib** : For data visualization, allowing the exploration of feature relationships and data distributions.
- **Scikit-learn**: For implementing machine learning models and performing tasks such as splitting the dataset, scaling features, and evaluating model performance.

These tools and libraries facilitated the development, evaluation, and deployment of the heart attack prediction model in an efficient and user-friendly manner.

4.2 User Characteristics

The target users of this model are researchers, or individuals interested in assessing their risk for stock price . Users can input various Stock information into the application, including factors like **Opening price, Volume**, and other lifestyle factors such as market behaviour. Based on this input, the model predicts the following day stock price.

Some key characteristics of the users include:

- **Stock professionals**: Stockers who use the model to assess the stock price risk of losing money.
- **Researchers**: Those studying the impact of various stock price behaviour .

4.3 Constraints

- **Data Quality**: The dataset used for training the model had missing values and inconsistencies, which required preprocessing and cleaning. Additionally, the dataset may not represent all populations accurately, limiting the generalizability of the model.

- **Model Complexity**: The chosen algorithms, like **L**inear **Regression** and SVR,Random forest regressor are relatively simple, which can limit their performance on complex datasets. While these algorithms provide good baseline results.

• 4.4 ML Algorithm Discussion

In this project, two machine learning algorithms were evaluated for heart attack risk prediction: **Linear Regression** and **LASSO, RIDGE, Random forest regressor, SVR, Elasticnet**. All algorithms were tested using the dataset, and their performance was assessed based on accuracy

Linear regression , Random forest regressor,SVR

```
lr = LinearRegression()
lr.fit(X_train_scaled, y_train)
y_pred_lr = lr.predict(X_test_scaled)
print("LR R2 Score:", r2_score(y_test, y_pred_lr))

# Step 7: Random Forest Regressor
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print("RF R2 Score:", r2_score(y_test, y_pred_rf))

# Step 8: Support Vector Regressor
svr = SVR(kernel='rbf', C=100)
svr.fit(X_train_scaled, y_train)
y_pred_svr = svr.predict(X_test_scaled)
print("SVR R2 Score:", r2_score(y_test, y_pred_svr))
```

```
LR R2 Score: 0.8556982161297392
RF R2 Score: 0.7972920517115207
SVR R2 Score: 0.44843543679233655
```

Fig 1 Logistic Regression ,SVR,RFR

In this project we implemented and evaluated three regression models: Linear Regression, Support Vector Regression (SVR), and Random Forest Regression (RFR). Among those models we trained each model with historical stock data as attributes such as Open, High, Low, Volume, and target variable Close Price. The training and testing sets were broken down into training and test sets and performance metrics were evaluated using commonly used regression metrics as R2 Score, Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

Performance Metrics:

- **Linear regression**: R^2= 85%
- **SVR**: R^2= 44%
- **Random forest regressor**: R^2= 79%

LASSO, RIDGE, Elasticnet

```python
from sklearn.linear_model import RidgeCV, LassoCV, ElasticNetCV

alphas    = np.logspace(-3, 3, 7)
l1_ratios = [0.1, 0.5, 0.9]

# Ridge
ridge = RidgeCV(alphas=alphas, cv=5)
ridge.fit(X_train_scaled, y_train)
print("Ridge R2 (test):", r2_score(y_test, ridge.predict(X_test_scaled)))

# Lasso
lasso = LassoCV(alphas=alphas, cv=5, max_iter=5000)
lasso.fit(X_train_scaled, y_train)
print("Lasso R2 (test):", r2_score(y_test, lasso.predict(X_test_scaled)))

# ElasticNet
enet = ElasticNetCV(alphas=alphas, l1_ratio=l1_ratios, cv=5, max_iter=5000)
enet.fit(X_train_scaled, y_train)
print("ElasticNet R2 (test):", r2_score(y_test, enet.predict(X_test_scaled)))

# Predict tomorrow's close
pred_ridge   = ridge.predict(latest_scaled)[0]
pred_lasso   = lasso.predict(latest_scaled)[0]
pred_enet    = enet.predict(latest_scaled)[0]

print("Predicted Close (Ridge):", pred_ridge)
print("Predicted Close (Lasso):", pred_lasso)
print("Predicted Close (ElasticNet):", pred_enet)
```

```
Ridge R2 (test): 0.8556313049203341
Lasso R2 (test): 0.858481884604775
ElasticNet R2 (test): 0.8546151901301697
Predicted Close (Ridge): 113.38680885979454
Predicted Close (Lasso): 113.03089133125519
Predicted Close (ElasticNet): 113.46750153145227
```

Fig 2 LASSO,RIDGE,ElasticNet

To further improve model generalization, to reduce overfitting, regularized regression methods were used in the experiments such as Lasso Regression, Ridge Regression and ElasticNet Regression. Lasso Regression has the advantage of shrinking the weights relatively unimportant features to zero so as to achieve feature selection in this manner. Ridge Regression was also regularized (L2 regularization). It is characterized by an even distribution of weights and an avoidance of large coefficients. This regularized model had the best results among all regularized models. ElasticNet Regression (combination of L1 and L2 penalties) performed about as well as the Lasso and Ridge regularized models at providing reasonable balance between them. This regularized model is flexible.

Performance Metrics:

- **LASSO**: R^2= 85%
- **RIDGE**: R^2= 85%
- **Elasticnet**: R^2= 85%

4.5 Dataset Structure

The dataset used in this project was derived from data collected over time (historic stock data) from Yahoo Finance. Every record in the dataset represents the trading activity on a particular day. The data is structured in tabular form. The 7 key attributes (columns) of the data is supplemented by multiple rows based on the date range selected.

Attributes Description:

Date – Represents the trading date of the record.

Open — the price the stock opened at on a certain day.

High price – the highest price for the stock in the trading day.

Low – The lowest price observed during the trading session.

Close – The final trading price when the market closed.

Adj Close: The closing price adjusted for splits and dividend payments.

Volume – The number of shares traded during the day.

It is suitable for time series analysis and regression tasks because it contains the temporal and numerical dynamics required to predict stock prices. Dataset is cleaned, normalized and sub-sets are built into training and test sets before training to ensure good performance evaluation.

4.6 ER Diagrams

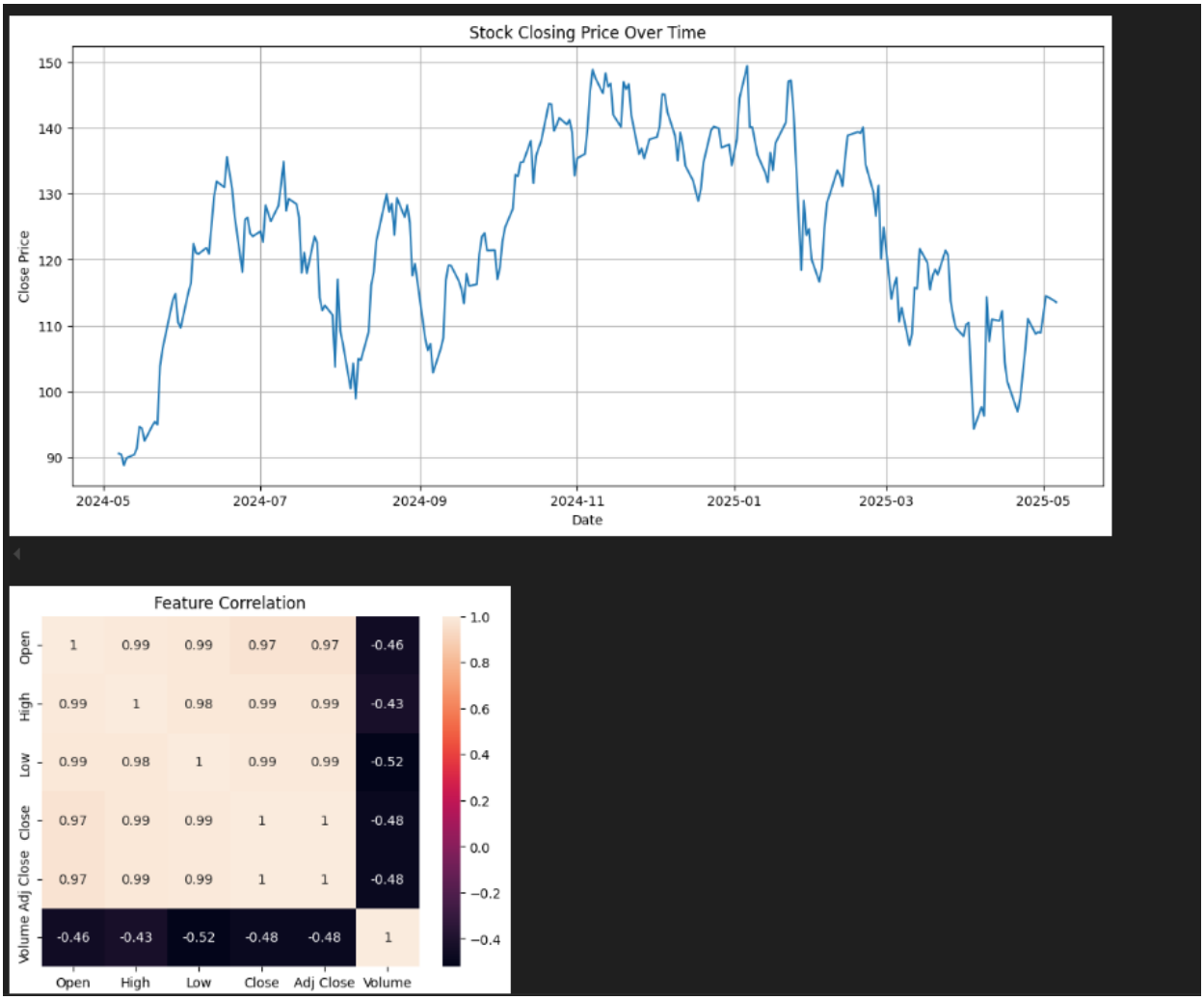

Fig 3Data Distribution of closing price over time, corelation with attributes

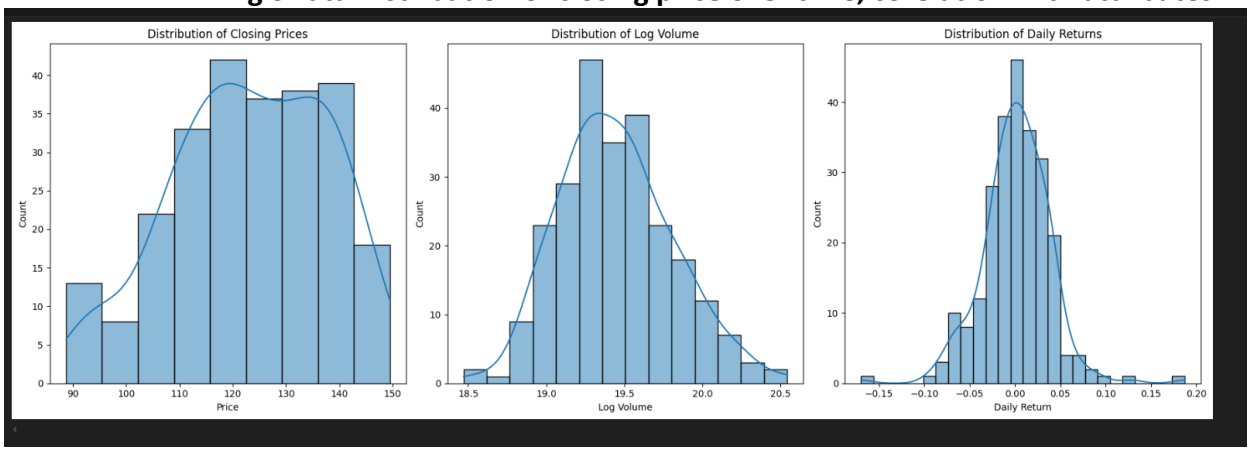

Fig 4 Price, Volume, Daily Return

CHAPTER - 5

We evaluated the performance of multi-regression models to evaluate the ability of multi-regression models to predict the stock price in the future. We used dataset containing seven financial indicators: Date, Open, High, Low, Close, Adj Close and Volume. After preprocessing and feature selection, six multi-regression models were trained and tested on the same dataset in two fractions. The evaluation was done using the standard metrics: R2 Score, Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

Model Performance Summary:

Model	R^2 score
Linear Regression	**0.85**
Support Vector Regression (SVR)	**0.44**
Random Forest Regression (RFR)	**0.79**
Lasso Regression	**0.85**
Ridge Regression	**0.85**
ElasticNet Regression	**0.85**

Of all the models, the best one performed were Random Forest Regression, LASSO, RIDGE, Elasticnet with an R2 score of 0. 85 indicating that it can explain 85% of the variance in the stock prices.

Chapter – 6

Conclusion and Future Scope

The project successfully implemented the stock price prediction system using various machine learning models including linear regression, SVR and random forest. The results demonstrated that random forest provided the highest accuracy among all models. The system efficiently used scraping data, applied preprosaresing, and gave explanatory results. In the future, the model can be improved by including real -time data feed, advanced deep learning models such as LSTMs and more financial indicators. Integration with a user interface or mobile app can make predictions accessible for investors, while training on updated dataset will increase reliability and adaptability in dynamic market conditions.

Bibliography

 James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.

 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

 Yahoo Finance. (2024). *Historical Stock Data*. Retrieved from https://finance.yahoo.com

 Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning* (3rd ed.). Packt Publishing.

 VanderPlas, J. (2016). *Python Data Science Handbook*. O'Reilly Media.

 sklearn.linear_model. (2024). *Scikit-learn Documentation*. Retrieved from https://scikit-learn.org/stable/modules/linear_model.html

 selenium.dev. (2024). *Selenium WebDriver Documentation*. Retrieved from https://www.selenium.dev/documentation/

 Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. *Proceedings of the IEEE International Conference on Big Data*.