International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Assessing Machine Learning's Accuracy in Stock Price Prediction

Aryan Bhatta[a*], Pranshu Poudyal[b], Drishant Kumar Maharjan[c], Aryaa Thapa[d]

[a,b,c,d]*Premier International School, Kathmandu, Nepal*

[d]*Lancers International School, Gurgaon, India*

[a]*Email: aryan.b11@premier.edu.np*

Abstract

This research examines how well machine learning models can predict the closing price of traded stocks. The financial industry has seen an increase in the use of these models due to the availability of datasets and technological advancements. The study compares machine learning models such as Linear Regression, Random Forest and K Nearest Neighbor (KNN) to determine which ones are the accurate predictors and what factors contribute to their effectiveness. To gain insights into model performance a diverse dataset consisting of five stocks from sectors is used. Data analysis and modelling are conducted using Python programming language with libraries, like Pandas, NumPy, Matplotlib and Scikit learn. The performance evaluation metric utilized is Mean Squared Error (MSE). The research findings have the potential to assist investors and traders in making decisions while also contributing to the growth of the financial industry.

Keywords: Machine Learning; Stock Price Prediction; Linear Regression; Random Forest K Nearest Neighbor (KNN); Mean Squared Error (MSE); Financial Industry.

1. Introduction

Predicting the closing price of traded stocks in the market has been popularized by the emergence of machine learning models that offer comprehensive features and tools for this problem. These models have become more accessible due to advances in technology, the availability of vast amounts of historical data, and the need for accurate stock price predictions in the financial industry. As a result, both retail investors and companies are using machine learning models to gain insight into future stock prices. This research will focus on the effectiveness of machine learning models commonly used to predict the closing price of stocks. Several machine learning models are available for this task. each with advantages and disadvantages. This investigation aims to compare and contrast these models in order to identify the most effective ones and the factors contributing to their effectiveness.

Received: 8/7/2023
Accepted: 9/12/2023
Published: 9/26/2023

----

* Corresponding author.

Furthermore, the relevance of this research goes beyond individual or organizational levels due to the intersection of the country's financial sector and the use of machine learning.

Accurate stock price forecasts are essential for investment decisions and can significantly affect a country's economic growth. Therefore, this research can influence foreign investment decisions, profits and losses. Understanding the effectiveness of various machine learning models in predicting stock prices can inform investment decisions and contribute to the growth of the financial industry.

1.1 Background Information

The closing cost of a stock is a key factor in determining investment trends for both investors and traders. This figure shows the final market rate that stock holds during a given day, as well as how much it costs to purchase a single share. There are numerous external elements that have the potential to drastically alter stock price, such as economic conditions, information, company efficiency, and global activity. Since these variabilities are harder to foresee, stocks normally undergo heightened fluxes.

Utilizing machine learning to select stocks has grown in popularity lately. It can make use of expansive datasets to find patterns that the typical individual may not notice. This has been used to generate multiple machine learning algorithms that are specifically made to estimate stock expenses. They can be prepared by utilizing past data in order to anticipate stock prices in the future.

As machine learning models evolve, they have become increasingly precise in predicting stock prices. This has led to their widespread use in the finance industry, where they are used by hedge funds, banks, and other financial institutions to make investment decisions. Using machine learning algorithms, investors and traders can gain a competitive edge against their peers by accurately predicting stock prices and making more informed investment decisions.

1.2 Supervised and Unsupervised Learning Models

Supervised and unsupervised learning models are two of the main machine learning models. A supervised learning model is trained using labelled data, where the inputs and the expected output are pre-defined. Supervised learning models are widely used in the finance industry to predict stock prices because of their ability to use historical data to predict future outcomes.

This is so to build a function that can predict the output of future data inputs. In unsupervised learning models, the algorithm is trained on unlabeled data without a specified output variable. Therefore, supervised learning models can provide a more accurate output as they are trained on labelled data and while unsupervised learning models can find patterns and relationships in unlabeled data, they are often relatively more difficult to interpret.

	Date	Open	High	Low	Close
0	01/03/2021	123.75	127.93	122.790001	127.790001
1	02/03/2021	128.410004	128.720001	125.010002	125.120003
2	03/03/2021	124.809998	125.709999	121.839996	122.059998
3	04/03/2021	121.75	123.599998	118.620003	120.129997
4	05/03/2021	120.980003	121.940002	117.57	121.419998
5	08/03/2021	120.93	121	116.209999	116.360001
6	09/03/2021	119.029999	122.059998	118.790001	121.089996
7	10/03/2021	121.690002	122.169998	119.449997	119.980003
8	11/03/2021	122.540001	123.209999	121.260002	121.959999
9	12/03/2021	120.400002	121.169998	119.160004	121.029999

Figure 1: An example dataset of a supervised machine learning model.

In figure 1. the dataset shows an example of a supervised machine learning model. The inputs such as X1. X, and Xs could be Open, High, and Low to predict Close (Y). If the "Close" column is omitted, this dataset becomes an example of an unsupervised learning model where there is no corresponding output.

Other than the ability of supervised machine learning models to predict stock prices, they are also used in the finance industry to classify loan applications and detect fraudulent transactions. Therefore, the real-world applicability of these models is not just limited to this research, they can be used for other similar problems.

1.3 Stock Selection

The selection of stocks is paramount in any stock market investigation, and this research is no exception. The selected stocks represent different sectors of the economy, allowing for a more comprehensive view of how machine learning models perform across distinct industries. Therefore, five stocks from diverse sectors, such as technology, banking, and pharmaceutical, are chosen to carry out this investigation. The stocks used in this investigation are as follows:

1. Apple (AAPL): Undoubtedly. Apple is one of the most heavily valued companies in the world, known for its flawless track record of innovation.
2. Microsoft (MSFT): Microsoft offers various products, such as Windows, Office, and cloud-based services, such as Azure. 3. Johnson & Johnson (JNJ): A healthcare company with a diverse portfolio of products, including health products.
3. Pfizer (PFE): Pfizer is a pharmaceutical company that has been in the news lately due to its development of a COVID-19 vaccine. This company also has a broad portfolio of products in areas like cardiology and neuroscience.
4. JPMorgan Chase (JPM): It is one of the oldest multinational investment banks and financial institutions in the United States. Studying the historical data of these stocks in the investigation will provide valuable insights into how machine learning models perform across different sectors and industries, which can have substantial implications for investors and traders alike.

2. Choice of programming language

Python programming language is used for this investigation because it is widely med for data analysis and machine learning. Python also provides an interactive environment; therefore, several plots or graphs could be created to help visualize the data.

Jupyter Notebook is used for this investigation to write the code. Jupyter Notebook is a web- based interactive computing environment that could contain live code and visualizations. They are widely used in data analysis because they provide an interactive environment and allow for data manipulation.

Pandas, NumPy, Matplotlib, and Scikit-learn are Python libraries used commonly for data analysis in machine learning models.

Here is an explanation of Python libraries used for this investigation

Pandas: Pandas in a Python library for data manipulations and analysis. It provides tools

for data cleaning such as filling null values) and data merging (such as merging multiple Data Frames or CSV files) Pandas is used for this investigation because it also provides tools for data visualization.

NumPy: NumPy is a Python library used for scientific computing, which applies mathematical and computational methods to solve scientific problems. NamPy supports arrays and matrices, which are vital for numerical calculation in machine learning. It is used in this investigation to handle the data as arrays and matrices, making it easier to work with large historical datasets of the stocks.

Matplotlib: Matplotlib is a Python library widely used for data visualization, much as creating various graphs and plots. In the content of this investigation, Matplotlib can be used to create a visualization of the historical stock price data and the subsequent predicted closing prices generated by the machine learning models. Furthermore, it can be used to create histograms, scatter plots, and other visualizations to explore the relationships between multiple variables.

Scikit-learn: Scikit-learn is a popular machine-learning Python library. It provides tools for various machine-learning problems, such as regression, classification, and clustering, making it a comprehensive library. Scikit-learn is used for this problem because it provides many machine-learning models which can be used for the stock price prediction problem.

3. Data Collection

To predict the closing price of these stocks, stock data of 500 business days were extracted for each stock from Yahoo Finances and the organized on their separate CSV (Comin Separated Values) files. The following snapshot shows the first few rows of the AAPL file extracted from Yahoo Finances.

Date	Open	High	Low	Close
01/03/2021	123.75	127.93	122.79	127.79
02/03/2021	128.41	128.72	125.01	125.12
03/03/2021	124.81	125.71	121.84	122.06
04/03/2021	121.75	123.6	118.62	120.13
05/03/2021	120.98	121.94	117.57	121.42
08/03/2021	120.93	121	116.21	116.36
09/03/2021	119.03	122.06	118.79	121.09
10/03/2021	121.69	122.17	119.45	119.98
11/03/2021	122.54	123.21	121.26	121.96
12/03/2021	120.4	121.17	119.16	121.03
15/03/2021	121.41	124	120.42	123.99

Figure 2: First few rows of AAPL.cr.

The following steps were taken care of when preparing the data for the machine learning models :

- A large and diverse historical dataset (500 business days was considered).
- The data was pre-processed, such as checking for missing (NaN) values and deleting data not needed for the scope of this investigation (each as trading volume).
- The denser size for each stack was kept unconscientious to ensure that the results were affected by a correct distribution of data among the stocks.

4. Choosing Machine Learning Models For The Investigation

For the investigation, the effectiveness of the machine learning models are compared by using Linear Regression, Random Forest, and K Nearest Neighbor.

5. Machine Learning Model: Linear Regression

Linear regression is a statistical modeling technique that relates two variables: a dependent variable and one or more independent variables [1]. This machine learning algorithm can be used to predict the value of a dependent variable based on the values of one or more independent variables. In this example, the dependent variable is the closing price of a stock, while the independent variables are the opening price, high price, and low price of the stock on a given day. The simplest form of linear regression is simple linear regression, which involves one independent variable and one dependent variable.

The equation of a simple linear equation is given as follows [4]:

$Y = b0 + b1 * x$

where:

- y is the dependent variable, which is the variable to be predicted.
- x is the independent variable, which is the variable we use to predict y.
- b0 is the intercept, which is the value of y when x is zero.
- b1 is the slope of the regression line, which represents the change in y for a one-unit change in x.

The form of linear regression used in this investigation is the multiple linear regression model, which inputs various independent variables and then outputs the value of the dependent variable [2, 8]. The formula for the multiple linear regression model is given as follows:

where:

$$y = b_0 + bo\ +\ b^1 x^1 +\ b^2 \bullet\ x2\ +\ bz \bullet\ X3 + \ldots + bn\ \times\ xn$$

y is the dependent variable, which is the closing price of a stock in this case.

b_0 is the intercept, which represents the value of y when all independent variables are set to zero.

b1, b2, b3,...,bn, are the coefficients for the independent variables X1, X2, X3, Xn These coefficients represent the change in y for a one-unit change in each independent variable. holding all other independent variables constant [6].

• X1, X2, X3, X, are the independent variables: the stock's opening price, highest, and lowest price.

Machine Learning Model: Random Forest

6. Decision Trees

Decision trees are a machine learning algorithm that can be used for classification and regression problems. They work by recursively splitting the dataset into smaller subsets based on the values of the input features until the subsets become as pure as possible. The resulting tree-like structure consists of decision nodes representing the input features and leaf nodes representing the output classes or values. Suppose we have historical data on a particular stock's daily low, high, and opening prices and the corresponding closing price. We want to use this data to build a machine-learning model to accurately predict the stock's closing price for the coming days. To illustrate how a decision tree might be used for this problem, we have the following data for the last five days:

Day	Low	High	Open	Close
1	100	110	105	105
2	95	105	100	100
3	98	103	101	102
4	101	112	105	110
5	107	115	112	113

Figure 10: Using this data, a decision tree could be represented (see figure 3(ii) for the code)

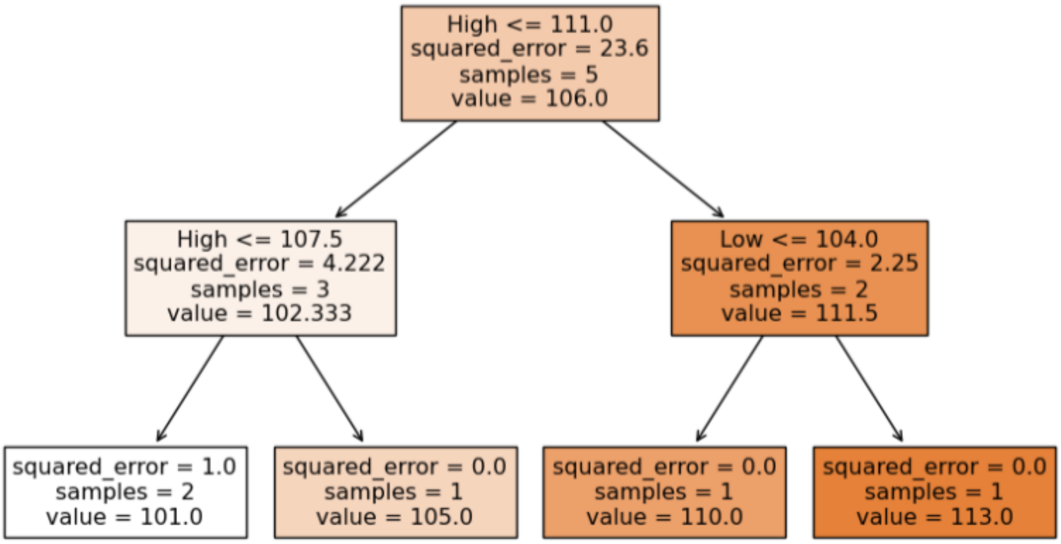
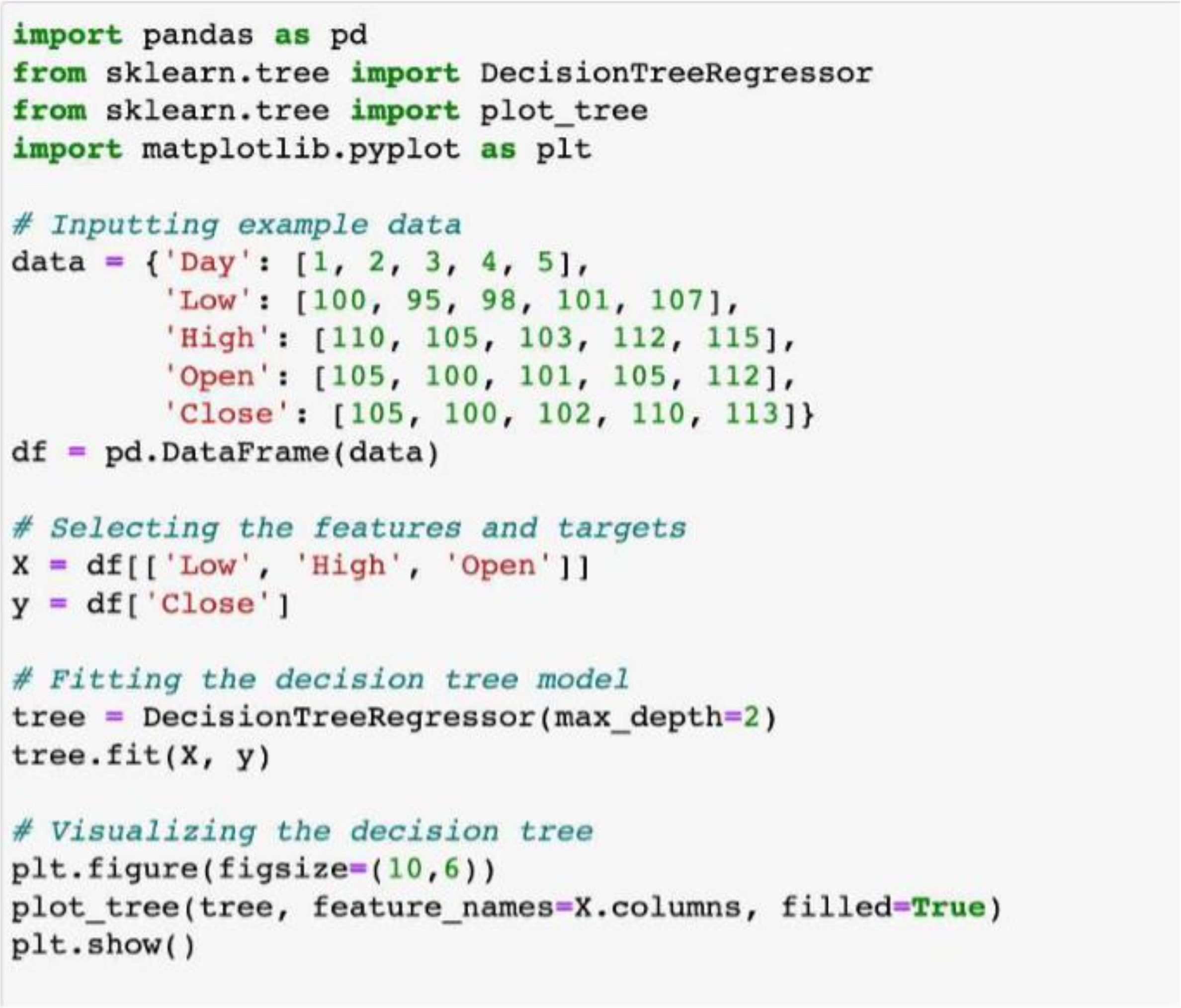


Figure 3(i): Decision tree representation of the dataset.



```
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

# Inputting example data
data = {'Day': [1, 2, 3, 4, 5],
        'Low': [100, 95, 98, 101, 107],
        'High': [110, 105, 103, 112, 115],
        'Open': [105, 100, 101, 105, 112],
        'Close': [105, 100, 102, 110, 113]}
df = pd.DataFrame(data)

# Selecting the features and targets
X = df[['Low', 'High', 'Open']]
y = df['Close']

# Fitting the decision tree model
tree = DecisionTreeRegressor(max_depth=2)
tree.fit(X, y)

# Visualizing the decision tree
plt.figure(figsize=(10,6))
plot_tree(tree, feature_names=X.columns, filled=True)
plt.show()
```

Figure 3(ii): shows the code for the decision tree represented in figure 2.

Random forest is a machine learning algorithm that can be used for regression problems. An ensemble learning model combines the speculation of multiple machine learning models to improve accuracy and minimize overfitting. Random forest is a specific type of ensemble learning model (used for this investigation) that combines multiple decision trees. One of the reasons behind choosing this model for this investigation is that it can help avoid overfitting. The problem of overfitting occurs when a statistical machine learning model fits against its training data, which means that it cannot perform well against unseen data. Each decision tree is trained on a different subset of data, and the final prediction is based on the aggregated prediction of all the decision trees.

Machine Learning Model: K-Nearest Neighbor (KNN)

K-Nearest Neighbor is a supervised machine-learning model used for classification and regression problems [5]. It can be used to predict the future price of a stock based on its historical data. In KNN, data points are represented as n-dimensional vectors and classified based on the "distance" between these

vectors.

These are the steps a KNN machine learning model undergoes when predicting the data:

First, the model chooses a number k, representing the number of "nearest" data points to consider. This is typically done through trial and error or cross-validation to find the optimal k value.

Next, the model calculates the distance between the data points. This can be done using metrics such as Euclidean or Manhattan distance.

Once the model has the distances, it can sort the data points by their distance to the one being predicted and select the k-nearest data points.

Finally, it can use the price of the k-nearest data points to estimate the stock's future price.

For a given data point, x, and its k-nearest neighbors, N, the distance between x and each neighbor, n, can be calculated using the Euclidean distance formula:

$$\text{distance}(x,n) = \sqrt{\sum_{i=1}^{n}(x_i - n_i)^2}$$

Where x, and n, are the values of the i^{th} feature for x and n, respectively. The summation is taken over all the features. Once the distances are calculated, the k-nearest neighbors can be selected and used to make the prediction.

For instance, the following table shows test data, and the following plot shows how a KNN model visualizes such data. Once the model is trained, it can predict new input data by finding the k nearest neighbors to the input data point and taking the average output values as the predicted value.

Day	Low	High	Open	Close
1	50	60	55	58
2	55	70	60	65
3	65	75	70	73
4	70	80	73	77
5	75	85	78	83
6	80	90	83	88
7	85	95	88	91
8	90	100	91	98
9	95	105	98	103
10	100	110	103	108

Figure 4: KNN representation of a dataset.

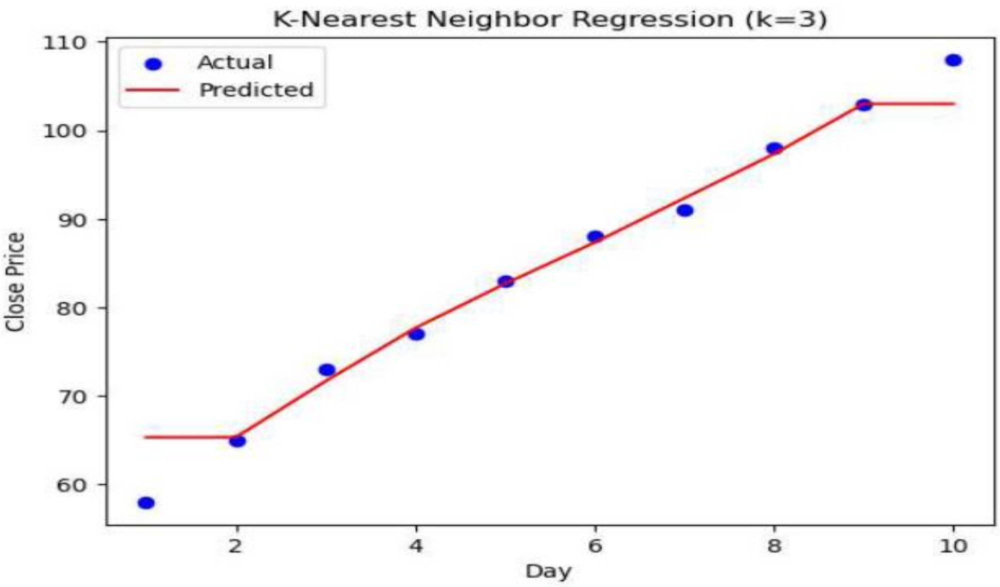
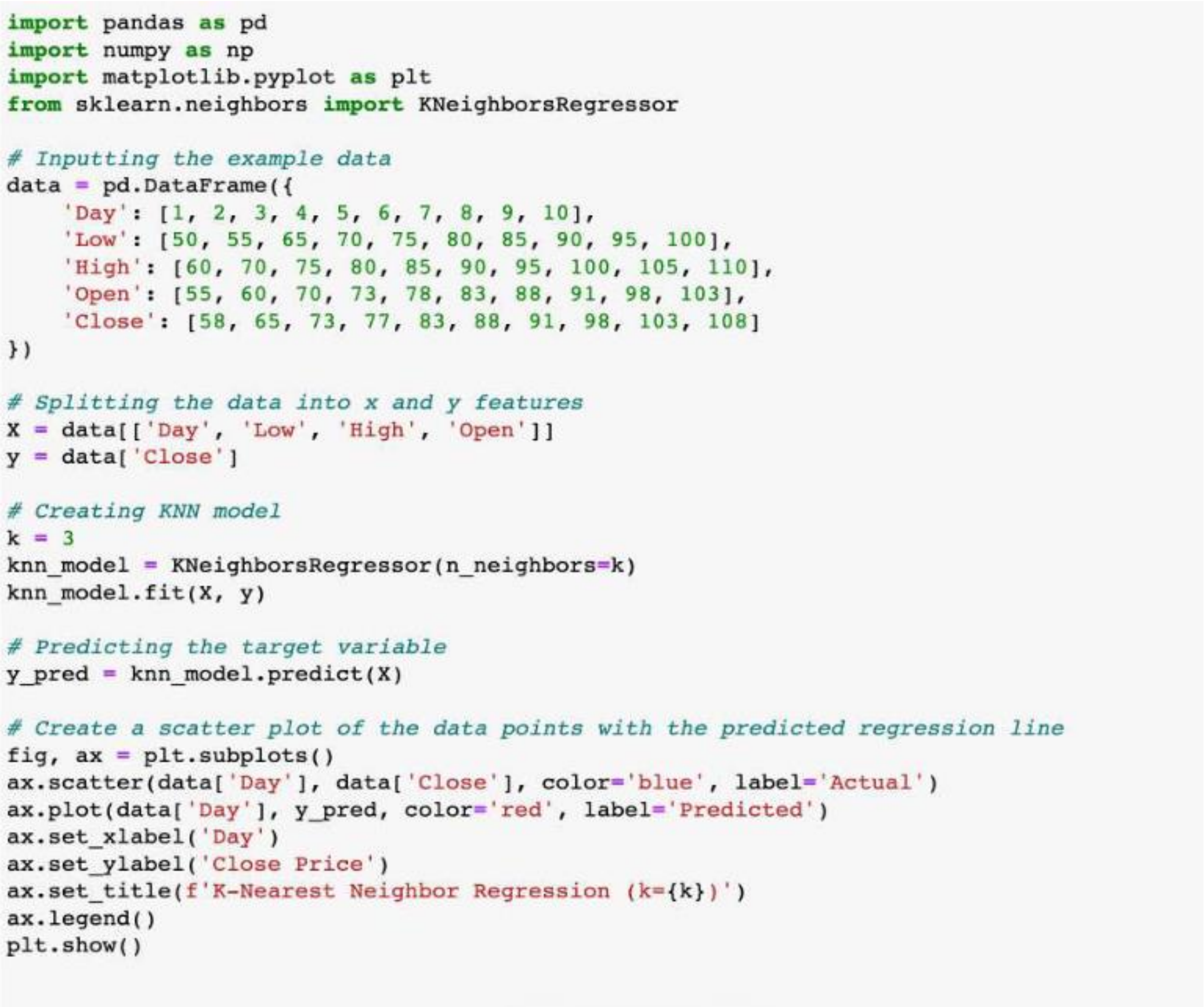


Figure 5: shows the code for the KN.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsRegressor

# Inputting the example data
data = pd.DataFrame({
    'Day': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Low': [50, 55, 65, 70, 75, 80, 85, 90, 95, 100],
    'High': [60, 70, 75, 80, 85, 90, 95, 100, 105, 110],
    'Open': [55, 60, 70, 73, 78, 83, 88, 91, 98, 103],
    'Close': [58, 65, 73, 77, 83, 88, 91, 98, 103, 108]
})

# Splitting the data into x and y features
X = data[['Day', 'Low', 'High', 'Open']]
y = data['Close']

# Creating KNN model
k = 3
knn_model = KNeighborsRegressor(n_neighbors=k)
knn_model.fit(X, y)

# Predicting the target variable
y_pred = knn_model.predict(X)

# Create a scatter plot of the data points with the predicted regression line
fig, ax = plt.subplots()
ax.scatter(data['Day'], data['Close'], color='blue', label='Actual')
ax.plot(data['Day'], y_pred, color='red', label='Predicted')
ax.set_xlabel('Day')
ax.set_ylabel('Close Price')
ax.set_title(f'K-Nearest Neighbor Regression (k={k})')
ax.legend()
plt.show()
```

Figure 11: N model plot represented in Figure 4.

7. Evaluation Metrics

A vital aspect of this investigation is comparing the results of the different machine learning algorithms used in the study. This will help determine the most efficient and accurate algorithm for stock price prediction. Several metrics exist to evaluate the performance of a machine learning model, such as MSE (Mean Square Error) and MAE (Mean Average Error). The metric used for evaluating the performance of machine learning models for this investigation is MSE, a commonly used metric for evaluating regression problems [3]. It is calculated by taking the average squared differences between the actual and the predicted values. The formula for MSE is given as follows:

$$MSE = \frac{1}{n} * \sum_{i=1}^{n} \left(y_{actual} - y_{predicted}\right)^{2}$$

Where:

- n is the total number of data points in the test set
- y $_{actual}$ is the actual value for a data point
- y $_{predicted}$ IS the predicted value for the same data point

It is essential to note that while MSE is a commonly used metric for evaluating regression problems, it is not the only metric that can be used. Other metrics such as MAE, R-squared, and RMSE (Root Mean Square Error) can also be used to evaluate the performance of machine learning models for stock price prediction.

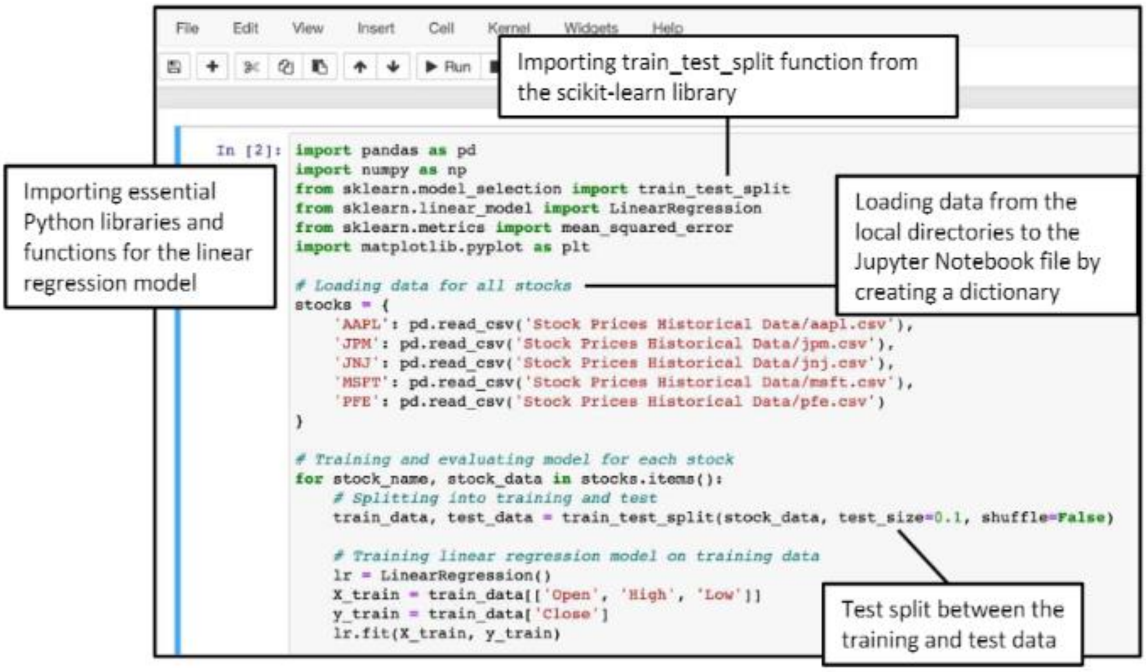
MAE, for instance, measures the average absolute difference between the actual and predicted values rather than the squared difference used in MSE. R-squared, on the other hand, measures the proportion of the variance in the dependent variable that is predictable from the independent variables. RMSE is similar to MSE but takes the square root of the average squared differences between the actual and predicted values [7].

In addition, it is also essential to consider the use case and specific goals of the stock price prediction model when choosing an evaluation metric. For instance, a metric that penalizes significant errors, such as MSE or RMSE, may be more appropriate if the goal is to minimize financial losses. However, if the goal is to predict the direction of the stock price movement (i.e., up or down), a classification metric such as accuracy or F1 score may be more relevant.

8. Investigation

For this investigation, the dataset has been split into a ratio of 9:1, where 90% of the data is

used to train the model, and the rest 10% is used to test the model and evaluated using MSE. The dataset can be split into train and test data using the scikit-learn feature train test_split, which inputs a decimal ratio value to determine the training and test data. For example. a test_size of 0.1 means that 10% of the dataset is allotted to the test data, and the rest 90% is allotted to the training data.



```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Loading data for all stocks
stocks = {
    'AAPL': pd.read_csv('Stock Prices Historical Data/aapl.csv'),
    'JPM': pd.read_csv('Stock Prices Historical Data/jpm.csv'),
    'JNJ': pd.read_csv('Stock Prices Historical Data/jnj.csv'),
    'MSFT': pd.read_csv('Stock Prices Historical Data/msft.csv'),
    'PFE': pd.read_csv('Stock Prices Historical Data/pfe.csv')
}

# Training and evaluating model for each stock
for stock_name, stock_data in stocks.items():
    # Splitting into training and test
    train_data, test_data = train_test_split(stock_data, test_size=0.1, shuffle=False)

    # Training linear regression model on training data
    lr = LinearRegression()
    X_train = train_data[['Open', 'High', 'Low']]
    y_train = train_data['Close']
    lr.fit(X_train, y_train)
```

Importing train_test_split function from the scikit-learn library

Importing essential Python libraries and functions for the linear regression model

Loading data from the local directories to the Jupyter Notebook file by creating a dictionary

Test split between the training and test data

Figure 6: shows how the investigation was carried out.

This image is a snapshot of the Notebook file where the Python code for the line regression model is compiled.

Findings

For the linear regression model, the following are MSE values for each stock:

AAPL MSE: 1.033035087925765

JPM MSE: 0.548108827098076

JNJ MSE: 0.3937946267801689

MSFT MSE: 2.3108724065298287 PFE MSE: 0.08021163233490894

A lower MSE represents a better performance of the model.

For instance, AAPL's MSE error of-1.033 suggests that the linear regression model built using the historical data of AAPL's stocks has a root mean square error of -1.033 when unpredicting closing prices on a test data set (see Evaluation Metrics section).

The linear regression model has performed best for PFE stock, as it has the lowest MSE.

The following figure shows an actual vs. predicted closing price graph of the AAPL stock outputted by the linear regression model (see Appendix for all the graphs outputted by the Linear Regression model).

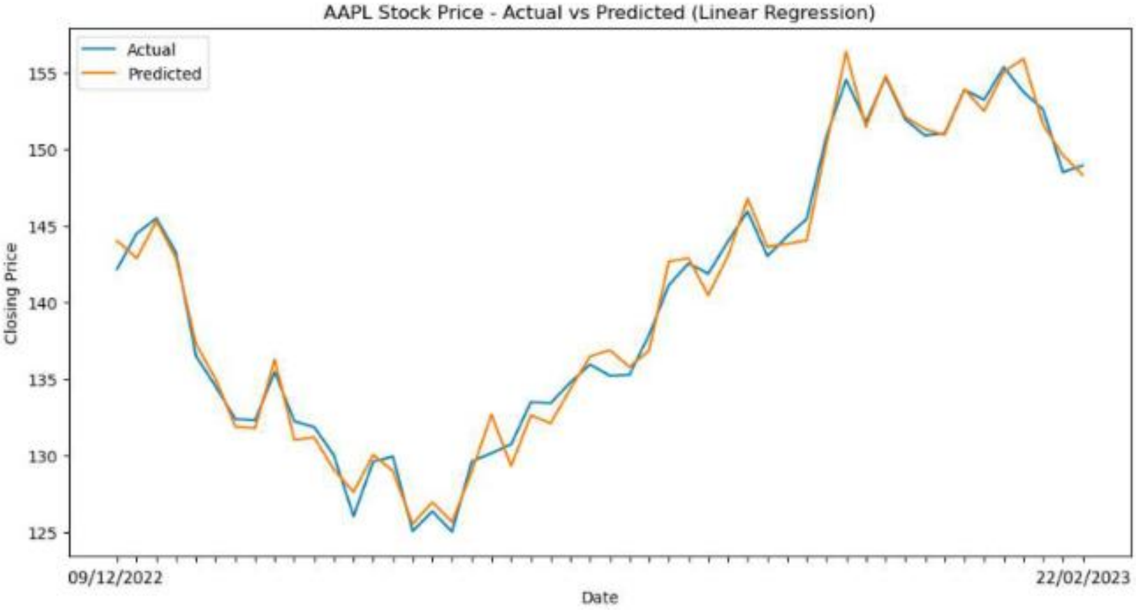


Figure 7: Actual vs Predicted graph of AAPL stock using Linear Regression model.

The following are the finds for the Random Forest Model:

AAPL MSE: 1.5949276403353148

JPM MSE: 0.944748974029316

JNJ MSE: 1.0165604889915791

MSFT MSE: 4.391515902170421

PFE MSF: 0.12070477770278848

Judging by the MSE values, the Random Forest model has performed the best for the PFE stock.

The following figure shows an actual vs. predicted closing price graph of the AAPL stock outputted by the random forest model.

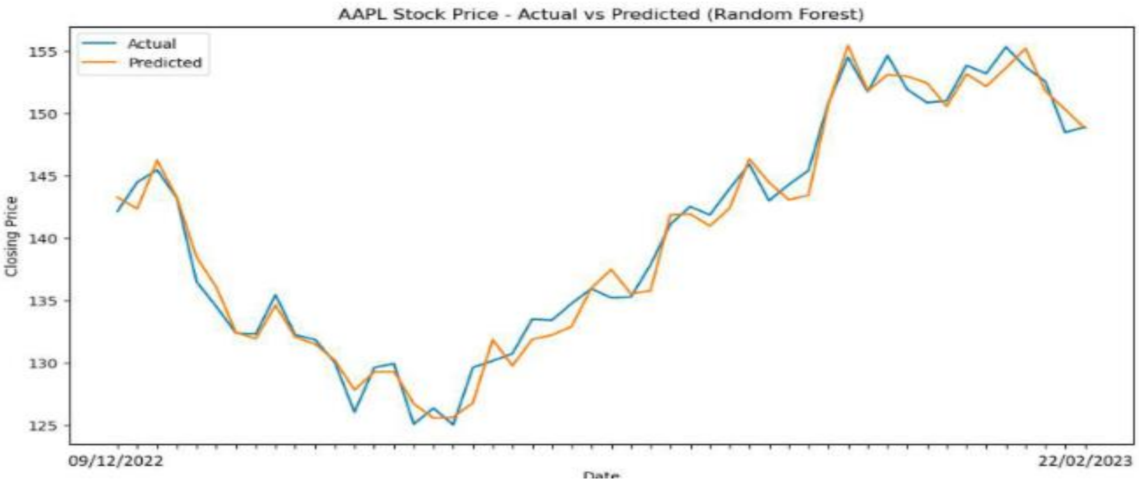


Figure 8: Actual vs Predicted graph of AAPL stock using random forest model.

The following are the finds for the KNN model:

AAPL MSE: 1.7398663994784973

JPM MSE: 1.096230416540109

JNJ MSE: 0.6654033782269775

MSFT MSE: 3.7815499162569095

PFE NISEI 0.1110422397615104

The following figure shows an actual vs. predicted closing price graph of the AAPL stock outputted by the KNN model.

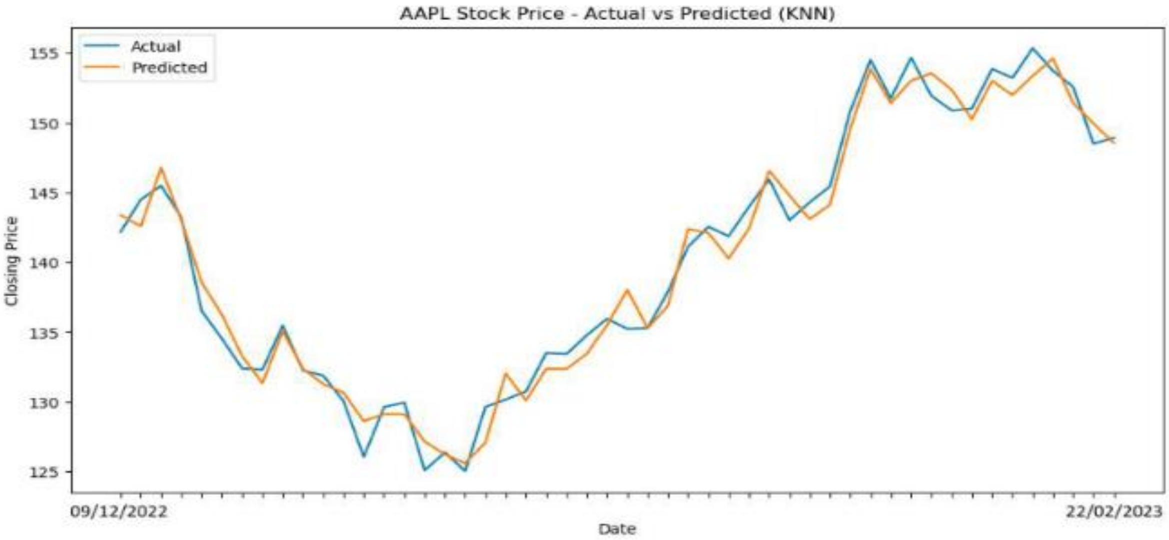


Figure 9: Actual vs Predicted graph of AAPL stock using KNN model.

9. Limitations of this investigation

Market volatility: Stock prices are affected by various factors that a machine-learning model might not be capable of predicting. Unexpected events such as natural disasters and political instability can affect stock prices. Sophisticated machine learning models must be built on a system that can consider these factors by analyzing factual data. In addition, some machine learning models assume an underlying stationary pattern over time, which may not be accurate for the stock market due to its volatility.

Limited Historical Data: While 500 business days of stock price data are used for this investigation, it is incorrect to assume that all machine learning models require the same amount of data to reach a good outcome. For instance, a linear regression model might require more inputs to reach a valid output than a random forest model. The performance of a machine learning model is affected if there is a lack of data available.

Randomness: Some models, such as Random Forest, have a degree of randomness ingrained into them. The same model may output different results every time the model is run with the same data.

Quality of Data: The stock price data for this investigation was sourced from a single source: Yahoo Finances. Multiple data sources and external factors could be introduced into the analysis to avoid introducing biases and keep the data as accurate as possible.

10. Conclusion

The comparative analysis of the three machine learning models—K-Nearest Neighbor (KNN), Random Forest, and Linear Regression—provides nuanced insights into their performance across different stocks. The KNN model exhibited varying effectiveness, performing sub-optimally for AAPL and MSFT while yielding improved results for JNJ and PFE. Conversely, the random forest model demonstrated superior performance for JPM and PFE, although it faced challenges with MSFT. The linear regression model excelled, showcasing the lowest Mean Squared Error (MSE) and thus outperforming the others for PFE, while also offering competitive performance for JNJ and AAPL. However, it's imperative to acknowledge that these findings are influenced by factors that were not exhaustively addressed within the scope of this investigation.

These outcomes illustrate the potential of machine learning models in delivering reasonably accurate predictions of stock closing prices. Nevertheless, it's noteworthy that the efficacy of these models exhibits variability among different stocks, suggesting the intricate nature of stock price movements. Notably, the random forest model displayed notable underperformance for MSFT, indicating the challenges of predicting stocks characterized by its dynamics.

While these models demonstrate the capacity to effectively predict closing prices by leveraging historical data, it's crucial to recognize that their predictive prowess correlates with the availability and comprehensiveness of information. Models achieve greater accuracy when equipped with comprehensive inputs encompassing factors beyond historical stock prices, such as market sentiment and macroeconomic indicators. Importantly, the models' predictions should not be relied upon as the sole basis for investment decisions. Instead, prudent decision-making necessitates the integration of predictive insights with broader fundamental analyses, encompassing considerations like company financials, prevailing market trends, and industry dynamics. By factoring in these broader aspects, the accuracy and robustness of predictive models can be further enhanced, leading to more informed investment strategies.

The findings from the three machine learning models—K-Nearest Neighbor (KNN), Random Forest, and Linear Regression—shed light on their predictive abilities across various stocks. Comparing these outcomes with prior studies reveals intriguing insights into the interplay between machine learning algorithms and stock price prediction.

In line with existing research, the KNN model's diverse performance reflects its sensitivity to data distribution and feature scaling, aligning with observations by authors in [9]. Similarly, the mixed results of the random forest model mirror by the authors in [9]. findings, emphasizing the algorithm's susceptibility to overfitting and the influence of industry dynamics on performance.

The linear regression model's success with PFE stocks aligns with conclusions drawn by authors in [10] on its efficacy in capturing linear relationships in stock data. However, acknowledging that these outcomes are context-specific and influenced by unexplored variables emphasizes the need for comprehensive investigations.

While these models exhibit impressive predictive capabilities, they may not account for market-altering events highlighted by the authors in [11], underlining the importance of real-world context. By contextualizing our findings within prior research, we validate their significance and pave the way for further exploration of machine learning's potential in stock price prediction.

11. Limitations of the study

While this study provides valuable insights into the potential of machine learning models for predicting stock prices, it’s important to recognize limitations impacting the broader applicability and reliability of the findings. The reliance on a relatively small dataset of 500 business days from Yahoo Finance might restrict the models' ability to capture complex stock price patterns and overlooks crucial factors like market sentiment and macroeconomic indicators. The evaluation of only a subset of machine learning algorithms, excluding advanced methods like neural networks and time series models, limits the exploration of potentially superior predictive models. The study's generalizability to diverse stocks and market conditions could be constrained by variations in model effectiveness. Diversifying evaluation metrics beyond Mean Squared Error (MSE) and considering external factors could enhance real-world relevance. Acknowledging and addressing these limitations will contribute to the development of more robust and accurate stock price prediction models.

12. Further scope of research

Possibility to conduct a more detailed analysis of market trends: While this investigation provides some insights into the behavior of stock prices based on historical data, there is still much to be known about the functioning of the stock market. For instance, future research could focus on exploring the topic more thoroughly by doing an in-depth analysis of market trends, such as how political events, news releases, or economic indicators affect stock prices.

Explore using other machine learning algorithms: This investigation used several machine learning algorithms to model the stock price data. However, many other machine learning algorithms could be used for this purpose (which could not be possible for this investigation due to the lack of advanced resources), such as support vector machines, decision trees, and neural networks. Future research could involve comparing the accuracy and performance of different algorithms on the same dataset to determine the best algorithm for the purpose.

Investigate the impact of other variables on the closing price of a stock: This investigation focused on the relationship between the closing price of a stock and some independent variables affecting it. However, many other variables, such as trading volume, dividend yields, and interest rates, could influence stock prices. Future research could investigate these variables' impact on stock prices to understand in more detail the functioning of the market and the factors that influence a stock's price.

References

[1] "2.1 Introduction to Linear Regression - Module 2: Fundamental Algorithms I," *Coursera*. https://www.coursera.org/lecture/machine-learning-accounting-python/2-1-introduction-to-linear-regression-MziWZ

[2] P. Agarwal, "Machine Learning For Prognosis of Life Expectancy and Diseases," *VOLUME-8 ISSUE-10, AUGUST 2019, REGULAR ISSUE*, vol. 8, no. 10, pp. 1765–1771, Aug. 2019, doi: https://doi.org/10.35940/ijitee.j9156.0881019.

[3] "Calculate Mean Squared Error using TensorFlow 2," *lindevs.com*, Oct. 24, 2020. https://lindevs.com/calculate-mean-squared-error-using-tensorflow-2 (accessed Mar. 13, 2023).

[4] Y. Choudhary, "Linear Regression Implementation in Python," *Linear Regression Implementation in Python*, Jun. 07, 2017. https://yasirchoudhary.blogspot.com/2017/06 (accessed Mar. 14, 2023).

[5] JavaTpoint, "K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint," *www.javatpoint.com*, 2021. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[6] HKT Consultant, "Multiple Coefficient of Determination in Multiple Regression," *HKT Consultant*, Aug. 31, 2021. https://phantran.net/multiple-coefficient-of-determination-in-multiple-regression/ (accessed Mar. 18, 2023).

[7] I. Inada, "Comprehensive Guide on Root Mean Squared Error (RMSE)," Aug. 12, 2023. https://www.skytowner.com/explore/comprehensive_guide_on_root_mean_squared_error (accessed Mar. 18, 2023).

[8] MLTut, “Multiple Linear Regression: Everything You Need to Know About,” *MLTut*, May 19, 2020. https://www.mltut.com/multiple-linear-regression/ (accessed Mar. 16, 2023).

[9] O. Altay, *Performance of different KNN models in prediction english language readability*. IEEE, 2022, pp. 1–5. doi: https://doi.org/10.1109/ICMI55296.2022.9873670.

[10] Rishab Mamgai *et al.*, “Stock prediction & recommendation system using KNN and linear regression,” *Nucleation and Atmospheric Aerosols*, Jan. 2022, doi: https://doi.org/10.1063/5.0108799.

[11] R. Ruhal and E. Prashar, *A Comparative Study Of Statistical Methods And Machine Learning Approaches For Stock Price Prediction*. 2023. doi: https://doi.org/10.13140/RG.2.2.19210.44483.

[12] Yahoo Finance, “Yahoo Finance - Business Finance, Stock Market, Quotes, News,” *Yahoo Finance*, 2023. https://finance.yahoo.com/