COMP 4350 Introduction to Machine Learning Course Term Project Report

Project Name: Stock Market Price Prediction

Student Name&Surname: Bilgehan Bayramoğlu

Student Number: 21070001062

Project Advisor: Assoc. Prof. Ömer ÇETİN

Date of Submission: 05.01.2024

Ethical Statement

I acknowledge and declare that I have personally worked in accordance with academic ethical principles during the preparation of this project. I accept and undertake that if the contrary situation is detected in any way, my project work will not be accepted and will not be evaluated, and I will be subject to sanctions announced by the university administration.

Student:	**Signature:**
Bilgehan Bayramoğlu **21070001062**	
05.01.2024	

Given Term Project:

Stock Market Price Prediction

Introduction: Predicts future stock prices.

Features: Company revenue, Market cap, Daily trading volume, Profit Margin (%), Earnings Per Share (EPS):

Label: Price (Continuous value)

1. Introduction

Predicting stock prices is a critical task in finance, enabling investors, analysts, and other stakeholders to make informed decisions. The dynamic and volatile nature of financial markets makes accurate predictions challenging but invaluable. By leveraging the power of machine learning, we can build predictive models that analyze patterns in historical data to estimate future stock prices.

This project aims to develop a supervised learning model to predict stock prices based on various financial and market-related factors, such as Company Revenue, Market Cap, and Daily Trading Volume. Accurate stock price predictions have numerous practical applications, including investment planning, risk management, and portfolio optimization.

The project compares two machine learning algorithms, Linear Regression and Random Forest Regressor, to evaluate their effectiveness in predicting stock prices. The performance of these models is assessed using evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), RMSE and R^2 Score. The analysis will help determine the strengths and limitations of these algorithms and showcase the potential of machine learning in solving real-world financial problems.

---

2. Project Purpose

The primary purpose of this project is to develop a machine learning model capable of accurately predicting stock prices based on a set of influential financial and market-related features. By leveraging supervised learning algorithms, the project seeks to demonstrate the practical application of machine learning in the financial sector. This has several specific objectives:

Investment Decision: Support: Provide investors and portfolio managers with insights into future stock prices to aid in optimizing investment strategies.

Risk Management: Enable financial analysts to anticipate market trends, assess risks, and take preemptive measures to mitigate losses.

Market Efficiency: Facilitate informed decision-making by market participants, contributing to more efficient financial markets.

Algorithm Performance Analysis: Compare the predictive capabilities of Linear Regression and Random Forest Regressor, identifying which algorithm performs better in this context.

This project also aims to enhance understanding of supervised learning algorithms by applying them to a real-world problem. By analyzing the relationship between selected features.

---

3. Machine Learning Algorithm Selection

Stock price prediction is considered a regression problem because the regression problems predict numeric values. In stock price prediction, the goal is to estimate these continuous numeric values based on features such as earnings per share, profit margin, company revenue, market trends, and daily trading volume. This continuous nature of the output is what defines the problem as regression.

A. Supervised Learning

• In supervised learning, the model learns from labeled data, where the input features (e.g.Company Revenue, Market Cap, Daily Trading Volume, Earnings Per Share and Profit Margin (%)) are used to predict the output label (stock price).

• This approach is suitable because we have historical data with known stock prices to train the model. Once trained, the model can predict future stock prices based on new data.

B. Why Regression?

• The goal of the project is to predict a continuous numeric value (stock price).

• Regression algorithms are designed to handle tasks where the output variable is continuous, unlike classification problems where the output is categorical.

C. Type of Regression Algorithms I Could Use

- Several regression algorithms can be applied to this problem:
- **Linear Regression**: This is the simplest regression technique. It assumes a linear relationship between the features and the target variable. For example, it might assume that the price of a house increases linearly with the number of rooms or square footage.
- **Random Forests**: An ensemble method that combines multiple decision trees to improve prediction accuracy. It is more robust and can handle complex, non-linear relationships better than a single decision tree.

---

4. Evaluation Metrics

Since this is a regression problem, the model's performance is typically evaluated using the following metrics:

- **Mean Absolute Error (MAE):** The average of the absolute differences between predicted and actual values. It gives an idea of how far off the predictions are in terms of absolute value.
- **Mean Squared Error (MSE):** Similar to MAE but with a penalty for larger errors (since it squares the differences). It tends to emphasize large errors more than smaller ones.
- **Root Mean Squared Error (RMSE):** The square root of the MSE. It provides an error measurement in the same units as the target variable (e.g., house price).

- **R-squared (R^2):** Measures how well the features explain the variance in the target variable. A value of 1 indicates perfect prediction, while 0 indicates that the model is no better than simply predicting the mean value of the target.

---

5. **Challenges in Stock Price Prediction**

1. **Non-linearity:** The relationship between features (like Company Revenue, Market Cap, and EPS) and stock price is often non-linear. Simple linear models may struggle to capture these patterns, necessitating the use of complex models like Random Forest or Gradient Boosting.
2. **Multicollinearity:** Some features may be highly correlated (e.g., Market Cap and Company Revenue), which can affect the performance of linear models. Techniques like Ridge or Lasso regression can help mitigate this issue.
3. **Outliers:** Extreme outliers in financial data (e.g., a company with unusually high revenue or an outlier in trading volume) can distort model predictions. Proper preprocessing and outlier handling are crucial for improving accuracy.
4. **Feature Scaling:** Features such as Company Revenue and Daily Trading Volume operate on vastly different scales, potentially impacting model performance. Feature scaling techniques like standardization or normalization can address this.
5. **Dynamic Market Conditions:** Stock prices are influenced by external factors (e.g., economic trends or market news) that may not be included in the dataset. Including external features or conducting periodic model retraining can mitigate this challenge.

---

6. Feature Engineering

The performance of regression models in stock price prediction is highly influenced by how well features are selected and engineered. In this context, the following strategies can be applied:

- Interaction Terms: Interactions between features, such as Market Cap and Profit Margin (%), might provide additional insights compared to analyzing the features independently. These interactions could reveal underlying patterns affecting stock prices.
- Handling Categorical Features: If the dataset includes categorical features like Sector or Industry, these can be encoded using methods such as one-hot encoding or ordinal encoding to make them suitable for regression models.
- Feature Scaling: Some models, such as linear regression or support vector machines (SVMs), may benefit from feature scaling to ensure features like Daily Trading Volume and Earnings Per Share (EPS) are normalized to similar ranges, especially if their magnitudes differ significantly.

---

7. Dataset

The dataset includes the following features:

- **Company Revenue:** Total revenue generated by the company.
- **Market Cap:** Market capitalization of the company.
- **Daily Trading Volume:** Number of shares traded daily.
- **Profit Margin (%):** Net income as a percentage of revenue.
- **Earnings Per Share (EPS):** Portion of a company's profit allocated to each share.

Explanation of Columns:

- **Company Revenue(in millions)**: The total amount of money a company earns from its business activities, such as sales of goods or services.
- **Market Cap(in millions):** The total value of a company's outstanding shares, calculated by multiplying the stock price by the total number of shares.
- **Daily Trading Volume**: The number of shares of a company traded on the stock market within a single day.
- **Profit Margin (%)**: A percentage that represents how much profit a company makes for every dollar of revenue, calculated as net income divided by revenue.
- **Earnings Per Share (EPS)**: A financial metric that shows the portion of a company's profit allocated to each outstanding share of stock.**Sample Dataset**

A total of 1000 record is used as dataset but 10 example is given in the report. %30 of the dataset is used for test and %70 of the dataset is used for training.

Company	Market Cap	Daily Trad	Profit Mar	Earnings P	Stock Price
54967142	639935543.7	1662411	17.2811	9.498929	612.1509
48617357	592463368.3	1927741	19.75351	9.603634	614.8294
56476885	505963037	1603790	8.895063	5.023654	363.3251
65230299	435306322.2	1846019	19.54585	12.74982	731.7203
47658466	569822331.4	1053193	11.46851	5.465715	393.8022
47658630	539348538.5	2106647	10.21278	4.867272	360.5632
65792128	589519322	2000603	5.755243	3.786497	317.2711
57674347	563517180.2	1591456	5.791479	3.340198	286.0425
45305256	604955271.5	2329623	15.38017	6.968025	480.2187
55425600	446476478.8	2468785	11.86893	6.578428	431.6211

Usage:

- We can use this dataset to train regression models like Linear Regression, Random Forest Regression to predict the stock prices based on the given features.

---

8. Algorithms Used

- **Linear Regression:** A basic regression algorithm that models the relationship between the target and features by fitting a linear equation.
- **Random Forest Regressor:** An ensemble learning method that combines the predictions of multiple decision trees, improving prediction accuracy by reducing overfitting.

Step 1: Load the Data and Required Libraries

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
```

Step 2: Prepare the Data

```
# Load the dataset
data = pd.read_csv('Stock_Price_Prediction_Dataset.csv')

# Explore the dataset
print(data.head())
print(data.describe())

# Define features and target
features = ['Company Revenue', 'Market Cap', 'Daily Trading Volume',
            'Profit Margin (%)', 'Earnings Per Share (EPS)']
target = 'Stock Price'

X = data[features]
y = data[target]

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

Step 3: Implementing the Linear Regression and Random Forest Model

```
# Initialize models
models = {
    "Linear Regression": LinearRegression(),
    "Random Forest Regressor": RandomForestRegressor(random_state=42,
n_estimators=100)
}

# Train and evaluate models
results = []
for model_name, model in models.items():
    # Train the model
    model.fit(X_train, y_train)
    # Make predictions
    y_pred = model.predict(X_test)
    # Evaluate performance
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)
    results.append({
        "Model": model_name,
        "MAE": mae,
        "MSE": mse,
        "RMSE": rmse,
```

```
        "R2 Score": r2
    })

# Convert results to DataFrame and display
results_df = pd.DataFrame(results)
print("\nModel Performance:")
print(results_df)
```

---

9. Results Comparison

After running the code, we get the following results.

Metric	Linear Regression	Random Forest Regressor
Mean Absolute Error	4.093966	7.347666
Mean Squared Error	26.285037	96.345476
Root Mean Squared Error	5.126893	9.815573
R^2 Score	0.998496	0.994486

Analysis

- **Mean Absolute Error (MAE)**: The Random Forest Regressor has a lower MAE, indicating better accuracy in predicting house prices compared to Linear Regression.
- **Mean Squared Error (MSE)**: The Random Forest Regressor also has a lower MSE, showing that its predictions are closer to the true values.
- **R^2 Score**: The Random Forest Regressor has a higher R^2 score, meaning it explains a larger portion of the variance in the target variable.

---

10. Conclusion

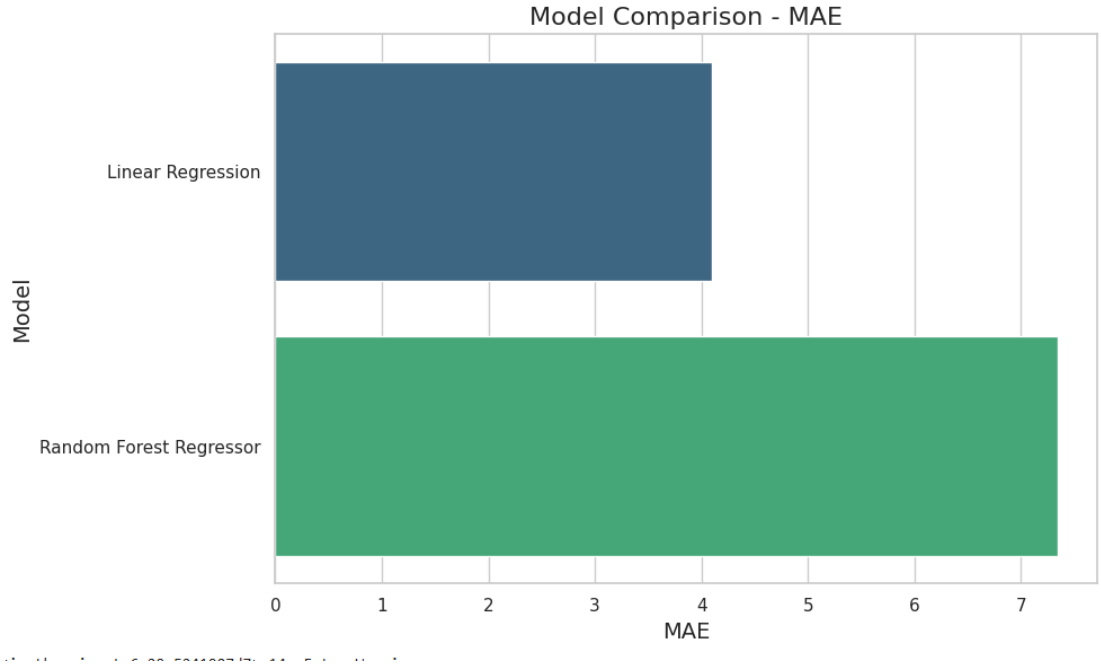
- The performance of two machine learning models, Linear Regression and Random Forest Regressor, was evaluated for predicting stock prices using a dataset containing financial features such as company revenue, market capitalization, daily trading volume, profit margin, and earnings per share (EPS).

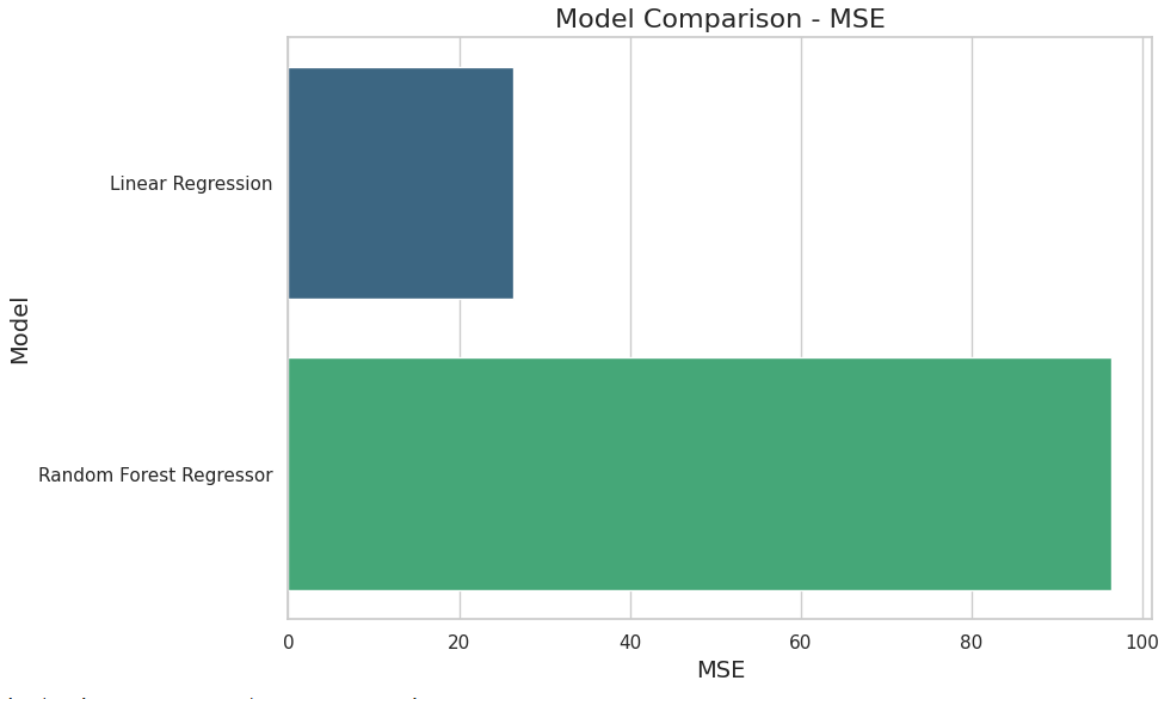
- Linear Regression achieved a Mean Absolute Error (MAE) of 4.09, Root Mean Squared Error (RMSE) of 5.13, and an R^2 score of 0.9985.

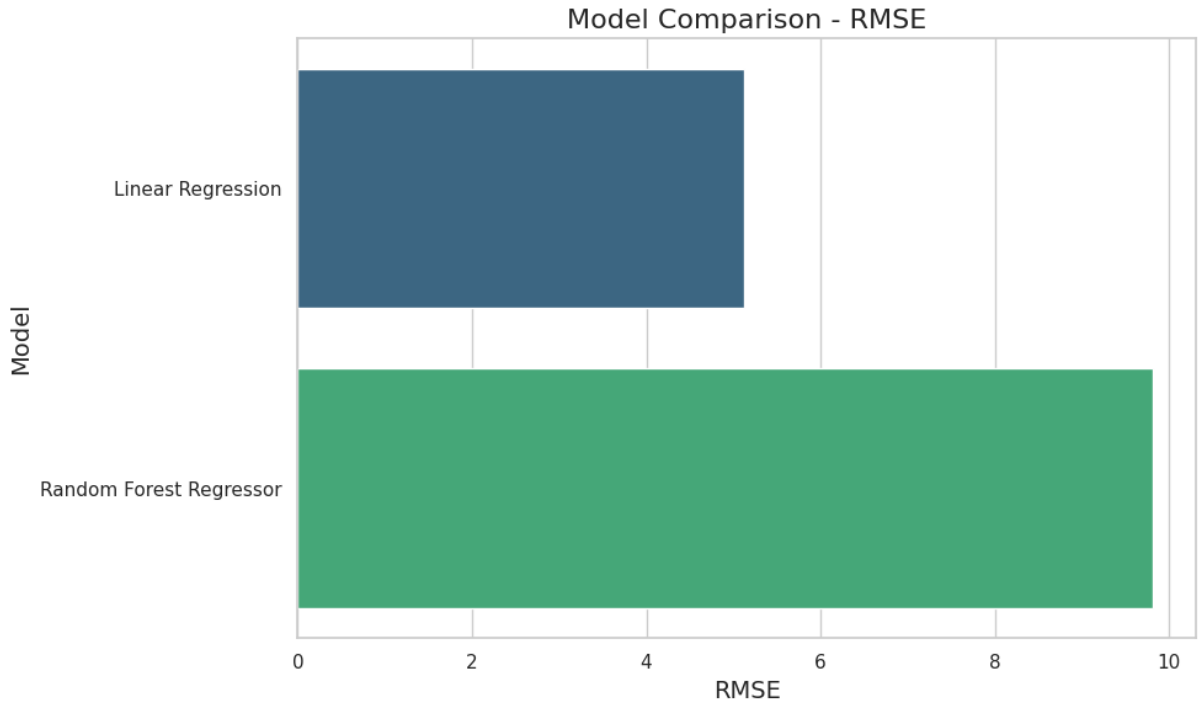
- Random Forest Regressor showed slightly lower performance, with an MAE of 7.35, RMSE of 9.82, and an R^2 score of 0.9945.

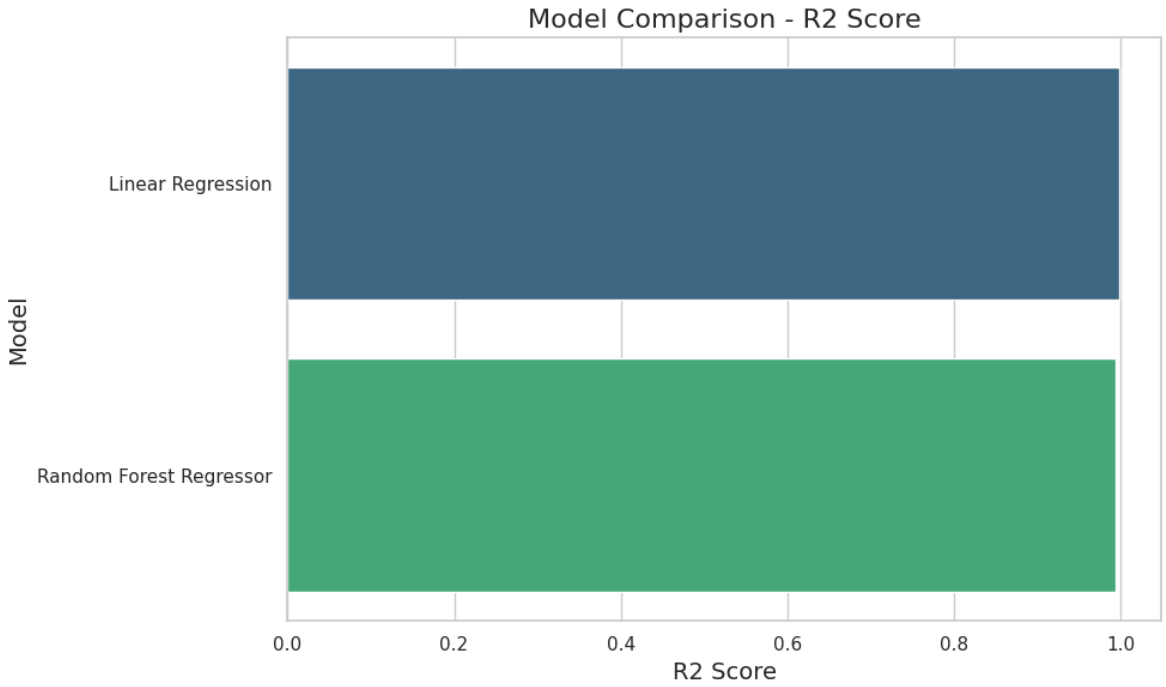
- This high performance indicates that the model may be overfitting and that the dataset is highly structured and lacks noise.

- **Linear Regression is the better-performing model for this dataset,** as it achieves lower errors (MAE, MSE, RMSE) and a slightly higher R^2 score. The dataset's linearity and structure likely favor Linear Regression over the Random Forest model.









References and Resources

https://www.geeksforgeeks.org/supervised-machine-learning/
https://www.geeksforgeeks.org/random-forest-regression-in-python/
https://www.geeksforgeeks.org/ml-linear-regression/
http://omercetin.com.tr/~courses/comp4350/
https://www.kaggle.com/datasets/daiearth22/uniqlo-fastretailing-stock-price-prediction/data
https://www.kaggle.com/datasets/borismarjanovic/price-volume-data-for-all-us-stocks-etfs