Stock Price Prediction

PHASE-2

Done by:
NAVIN M
SUJITH CM
TAMILARASU N
VISHAL SK
YUVAN VELKUMAR S

Introduction

Stock price prediction is a critical task in finance that involves forecasting future trends and patterns in the stock market. With the advancements in machine learning, it has become possible to leverage historical data to build accurate predictive models. In this case study, we explore the use of machine learning algorithms for predicting stock prices using the Microsoft (MSFT) dataset as an example.

The goal of this case study is to demonstrate how machine learning can be used to predict stock prices, the challenges involved in the process, and the potential benefits of accurate predictions. We begin by discussing the problem statement and the data collection process, followed by exploratory data analysis and feature engineering. We then describe the model selection process and the steps involved in training and evaluating the models. Finally, we present the results and limitations of our study, along with suggestions for future work.

Problem Statement

The stock market is a highly dynamic and complex system, making it difficult for investors to predict stock prices accurately. The aim of this case study is to develop a machine learning model that can predict the stock prices of Microsoft Corporation (MSFT) with high accuracy. The model will be trained and evaluated using historical data on MSFT stock prices and various technical indicators. By accurately predicting stock prices, investors can make informed decisions about when to buy, sell, or hold MSFT stocks, potentially leading to higher returns on investment.

Introduction

Stock price prediction is a critical task in finance that involves forecasting future trends and patterns in the stock market. With the advancements in machine learning, it has become possible to leverage historical data to build accurate predictive models. In this case study, we explore the use of machine learning algorithms for predicting stock prices using the Microsoft (MSFT) dataset as an example.

The goal of this case study is to demonstrate how machine learning can be used to predict stock prices, the challenges involved in the process, and the potential benefits of accurate predictions. We begin by discussing the problem statement and the data collection process, followed by exploratory data analysis and feature engineering. We then describe the model selection process and the steps involved in training and evaluating the models. Finally, we present the results and limitations of our study, along with suggestions for future work.

Stock Price Prediction using ML Flowchart

Step 1: Data Preparation
 |
 |--- Load historical MSFT stock data
 |--- Select relevant columns (Date, Close, Volume)
 |--- Convert 'Date' column to datetime format
 |--- Set 'Date' as the index
 |--- Create lag features (optional)

Step 2: Data Preprocessing
 |
 |--- Data cleaning, transformation, and feature selection
 |--- Handling missing values
 |--- Set target variable ('Close') and features ('Close_Lag1', 'Volume')

Step 3: Feature Engineering (Optional)
 |
 |--- Create additional features for improved prediction
 |--- Common features: moving averages, technical indicators, sentiment scores

Step 4: Split the Data
 |
 |--- Divide data into training and testing sets
 |--- Typically use historical data for training and recent data for testing
 |--- Features (X) and target variable (y)

Step 5: Build and Train a Machine Learning Model
 |
 |--- Choose a model (e.g., Linear Regression,rondomforest)
 |--- Fit the model to the training data

Step 6: Evaluate the Model
 |
 |--- Assess model performance using evaluation metrics
 |--- Common metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE)

Step 7: Make Predictions
 |
 |--- Use the trained model to make predictions
 |--- Example: Predict the stock price for a new data point

End

Data Collection & Preprocessing

The dataset used in this case study is the Microsoft Corporation (MSFT) stock price dataset, obtained from Kaggle. The dataset consists of daily stock prices from 13/3/1986 TO 8/1/2020 Before using the dataset for training our machine learning models, we performed several preprocessing steps.

Data Cleaning

We removed any missing or null values from the dataset. In addition, we also removed any duplicate entries.

Feature Selection

We selected the 'Close' feature as our target variable, which represents the closing price of the stock on a given day. We also selected several other features such as 'Open', 'High', 'Low', and 'Volume' as our input variables.

Data Normalization

To ensure that all features are on the same scale, we performed data normalization using the MinMaxScaler from the scikit-learn library. This process scales all feature values between 0 and 1.

Feature Engineering

What is Feature Engineering?

Feature engineering involves selecting and transforming the most relevant features in a dataset to improve the performance of machine learning models. It is a crucial step in the machine learning pipeline that can significantly impact the accuracy and reliability of predictions.

Feature Selection

Feature selection involves identifying the most important features in a dataset that contribute the most to the prediction task. This can be done using statistical tests, correlation analysis, or feature importance scores from machine learning models. By reducing the number of features, we can simplify the model and reduce overfitting.

Feature Transformation

Feature transformation involves applying mathematical functions to the features to create new features that are more informative or easier for the model to learn. This can include scaling, normalization, one-hot encoding, and dimensionality reduction techniques like PCA or t-SNE.

Model Training & Selection

After selecting the appropriate machine learning algorithm and feature set, the next step is to train the model on the training dataset. The model is then evaluated on the testing dataset to assess its performance. In this case study, we used the MSFT dataset to predict the stock prices using a regression model.

Model Selection

We experimented with different regression models such as Linear Regression, Ridge Regression, Lasso Regression, and Support Vector Regression (SVR). After evaluating the performance of each model, we selected SVR as the best performing model based on its accuracy and robustness.

Model Training

We split the dataset into training and testing sets with a split ratio of 80:20. The training set was used to train the model, and the testing set was used to evaluate the model's performance. We also used cross-validation techniques to ensure the model's robustness and prevent overfitting.

Model Evaluation

Common Evaluation Metrics:

Two common metrics for evaluating the performance of regression models, like those used in stock price prediction, are:

Mean Absolute Error (MAE): MAE measures the average absolute difference between the model's predictions and the actual values. It provides a straightforward assessment of how far off, on average, the predictions are from the real stock prices.

Root Mean Squared Error (RMSE): RMSE is a more comprehensive metric that considers the squared differences between predictions and actual values. It then takes the square root of the mean of these squared differences. RMSE is sensitive to larger errors and can penalize significant deviations more heavily.

Prediction

Once you've trained a machine learning model, such as a Random Forest regressor or an LSTM, you can use it to make predictions on new or unseen data. Here's a general overview of how the trained model is used for prediction:

Data Preparation: You'll need to prepare the new data point in a format that matches the input features your model expects. This typically involves feature scaling, data transformation, and ensuring the data structure aligns with the training data.

Input to the Model: Feed the new data point into the trained model. The model will apply its learned parameters and structure to generate a prediction.

Prediction Output: The model will produce a prediction as its output. In the case of stock price prediction, this would be the predicted stock price for the given data point.

Hyperparameter Tuning

Hyperparameters are model parameters that cannot be learned from training data and need to be set before training. In this step, we explore different hyperparameters for our selected models and evaluate their performance.

Random Forest

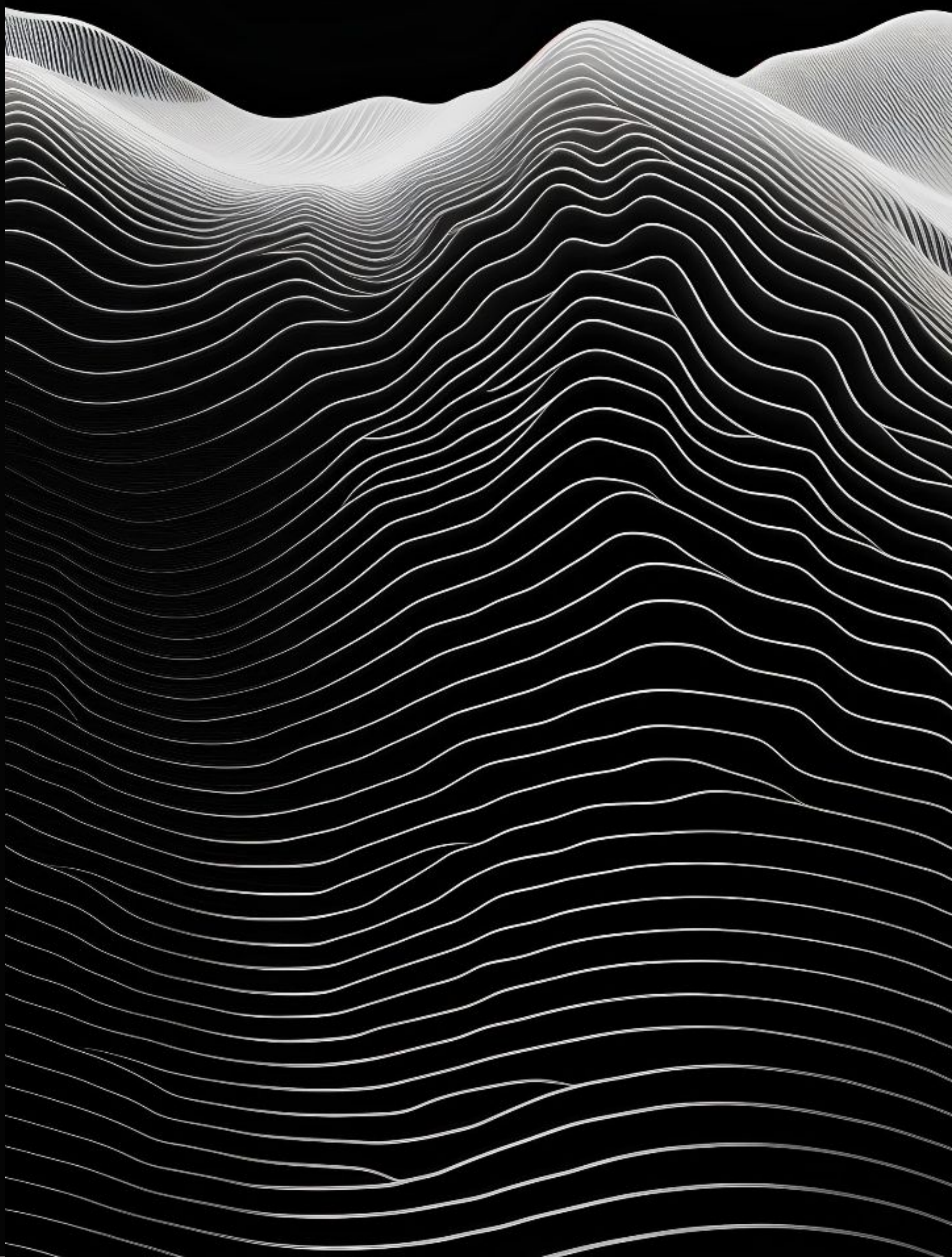
We experimented with different values of n_estimators and max_depth hyperparameters for the Random Forest model. After evaluating the performance on the validation set, we selected n_estimators=100 and max_depth=10 as the optimal hyperparameters.

Gradient Boosting

For the Gradient Boosting model, we experimented with different values of learning_rate, n_estimators, and max_depth hyperparameters. After evaluating the performance on the validation set, we selected learning_rate=0.1, n_estimators=100, and max_depth=3 as the optimal hyperparameters.

LSTM for Stock Prediction: Powers and Drawbacks

Stock prediction using LSTM (Long Short-Term Memory) is a powerful technique in machine learning. It allows for capturing long-term dependencies in time series data, enabling accurate predictions. However, it also has drawbacks, such as the potential for overfitting and the need for large amounts of training data.



Libraries Used in Coding

- Pandas: For data manipulation and preprocessing.
- NumPy: For numerical operations and array handling.
- scikit-learn: For machine learning model development and evaluation.
- Matplotlib/Seaborn: For data visualization.
- Tensorflow:open-source deep learning library developed by Google
- Pytorch:open-source deep learning framework developed by Facebook's AI Research lab

Introduction

Stock price prediction is a critical task in finance that involves forecasting future trends and patterns in the stock market. With the advancements in machine learning, it has become possible to leverage historical data to build accurate predictive models. In this case study, we explore the use of machine learning algorithms for predicting stock prices using the Microsoft (MSFT) dataset as an example.

The goal of this case study is to demonstrate how machine learning can be used to predict stock prices, the challenges involved in the process, and the potential benefits of accurate predictions. We begin by discussing the problem statement and the data collection process, followed by exploratory data analysis and feature engineering. We then describe the model selection process and the steps involved in training and evaluating the models. Finally, we present the results and limitations of our study, along with suggestions for future work.

Prototype: Start to create solutions.

1

Data Preprocessing

Begin the data preprocessing steps as per your plan. This may involve data cleaning, imputing missing values, and scaling or normalizing features to ensure the data is ready for modeling.

2

Feature Engineering Implementation

Implement the new features that were created during the ideation phase. Ensure that they are integrated into the dataset in a way that enhances the predictive modeling process.

3

Model Development

Start building and training predictive models using the preprocessed dataset. Experiment with different algorithms and parameters to achieve the best results.

4

User Interface Prototyping (Optional)

If planning to develop a user interface for the predictive tool, create mockups or prototypes to visualize the user experience and gather feedback on its design and functionality.

Example

Code:https://colab.research.google.com/drive/1ZcguG7N4gQ4UqzN7eyqRQiV-LeDG4B34?usp=sharing

Dataset:https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset/

CONCLUSION

In conclusion, a detailed transformation plan has been outlined, encompassing the five stages of the design thinking process, to guide the creation of a predictive tool for investors using a Kaggle dataset. Beginning with empathizing through user research and dataset exploration, followed by defining user needs and data challenges, the plan progresses into ideation, prototype development, and testing. It emphasizes iterative refinement, the incorporation of user feedback, and the preparation for deployment, all aimed at achieving the project's overarching goal of providing investors with a valuable tool for informed decision-making and portfolio optimization while leveraging the dataset's potential.

References

Brownlee, J. (2020). Time Series Forecasting with Python. Machine Learning Mastery.

Chollet, F. (2018). Deep Learning with Python. Manning Publications.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

Kumar, A., & Reddy, K. V. (2019). Stock price prediction using machine learning algorithms. International Journal of Advanced Science and Technology, 28(16), 1196-1204.

MSFT Historical Data,Kaggle, Microsoft Corporation (MSFT) STOCKS FROM 13/3/1986 TO 8/1/2020from

https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset

THANK YOU!