CS-GY 6923 Machine Learning
Apple Stock Price Prediction

Runze Li
rl50xx@nyu.edu

April 29, 2024

1 Background and Motivation

The stock market are filled with uncertainty and volatility. In that case, investors are in need of reliable tools to help them make informed investment decisions. In recent years, machine learning and artificial intelligence technology have made tremendous progress, providing more possibilities for stock predictions. By utilizing these advanced technologies, we can better understand market behavior and patterns so that we can improve the accuracy and reliability of stock prediction.

This project aims to solve the problem of stock market prediction. Specifically, we make predictions about Apple's future stock price. This is an important question because stock market predictions can help investors make more informed investment decisions, reduce investment risks, and increase investment returns. By using advanced machine learning models, we can try to capture the patterns behind stock prices, providing valuable insights to investors.

2 Process of Project Implementation

1. Data collection: Download Apple Stock Price, including 'Date', 'Open', 'Close', etc.
2. Data preprocessing: Clean data, deal with missing values, outliers, etc. Explore how the dataset is formatted and determine if there are any relationships between the features. Normalize or standardize the data so that the model can learn better.
3. Model selection: Choose appropriate machine learning algorithms, such as support vector machine (SVM), long short-term memory network (LSTM) and convolutional neural network (CNN).
4. Model training: Divide the dataset into training set and testing set. Use the training set to train the model for model training.
5. Model evaluation: Evaluate the performance of the model, such as mean squared error (MSE), Mean absolute error (MAE), etc. Use visualization tools to display model prediction results, feature importance and other information to make it easier to understand.
6. Improvement: Use new methods and models to improve prediction performance. Collect new data and update models to adapt to market changes.

3 Dataset

Apple Company is an American technology company in California. Apple's stock is usually known as AAPL. Apple's stock always attracts much attention since it is one of the largest technology companies in the world. Generally speaking, the performance, products and technological innovation of the company will have an impact on its stock price.

AAPL.csv contains historical data about Apple stock. Each row records stock transaction information on the corresponding date. The data columns include:

- Date: the date corresponding to the stock
- Open: opening price
- High: highest price
- Low: lowest price
- Close: closing price
- Adj Close: Adjusted closing price
- Volume: trading volume

4 Data Preprocessing

Data preprocessing is a crucial step in data analysis and machine learning. It involves operations such as cleaning, transforming and normalizing raw data to make the data suitable for modeling and analysis.

4.1 Clean missing data

Cleaning missing values is to ensure data quality and accuracy of analysis. Missing values will affect statistical indicators, such as mean, standard deviation, etc., thus affecting the overall understanding of the data. Moreover, missing values may cause gaps or discontinuities in data visualization results, making the results difficult to interpret and understand. So cleaning missing values can improve the accuracy and reliability of data and ensure that analysis results are more trustworthy.

```
# check for missing values
print('Checking for missing values')
print(df.isnull().sum())
```

4.2 Convert the date column to datetime format

We need to convert the date column to datetime format because this improves the efficiency and convenience of data operations, especially when processing time series data.

```
# Convert 'Date' column to datetime format and set it as index
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)
```

4.3 Normalization

In the field of machine learning and data analysis, it is often necessary to process data with different scale ranges. Normalization can scale the data to the same range, which is beneficial to the convergence and accuracy of the algorithm. So normalizing data helps improve data analysis and modeling and is a common preprocessing step in many cases.

```
# Normalize the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_prices = scaler.fit_transform(closing_prices)
```

5 Exploratory Data Analysis

Exploratory data analysis (EDA) is a critical preliminary step in understanding the dataset. This method involves using statistical charts and mathematical tools to reveal key features of data, discover patterns, anomalies, and relationships between variables in data. The purpose of EDA is to have an intuitive understanding of the data before building a complex model so that it can guide subsequent data processing and analysis strategies.

EDA is the basis for reporting and communicating results, making data-driven decisions more credible. Through EDA, we can gain insights into Apple's stock, which helps us formulate effective data processing strategies and select appropriate analysis models. So exploratory data analysis is an integral part of data analysis.

5.1 Scatter chart of Apple's stock

Scatter plots are a common tool in EDA and this method can demonstrate the relationship between two variables. We use scatter plots 1 to show how Apple's stock has changed over time.

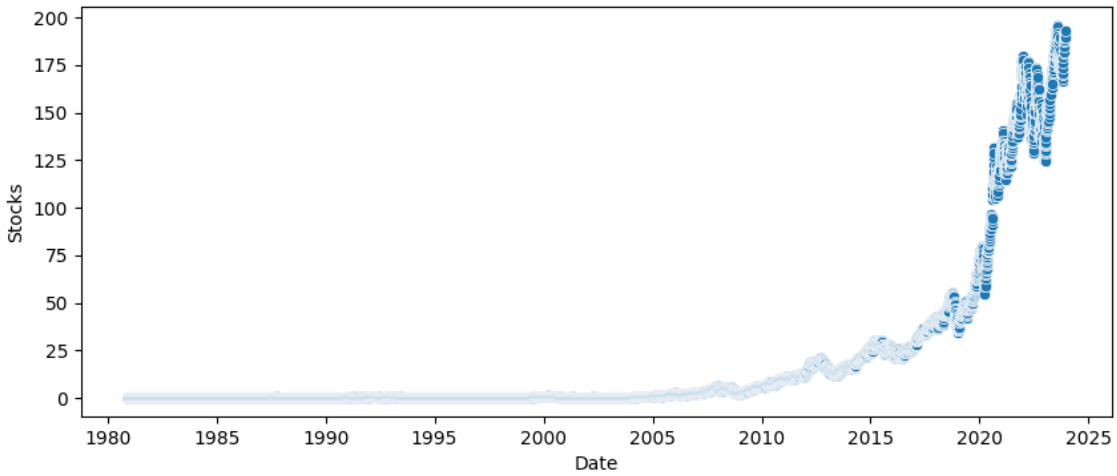


Figure 1: Apple Stocks from 1980 to 2023

From 1980, Apple's stock price remained low and changed little until 2000. After about 2000, Apple's stock price has began to rise significantly because of company's innovations in consumer

electronics. In 2010, the stock price has shown a trend of accelerating growth, which is corresponding to Apple's successful products, such as iPhone and Macbook.

Overall, Apple's stock price shows a transition from stability to rapid growth, reflecting the company's dramatic changes and success.

5.2 Visualization of relationships between features

It is crucial to understand the relationships between features in the dataset. Pairplot is a powerful tool for exploratory data analysis, which can display the relationship between two variables. Figure 2 shows the pairing relationship between two features of the dataset.

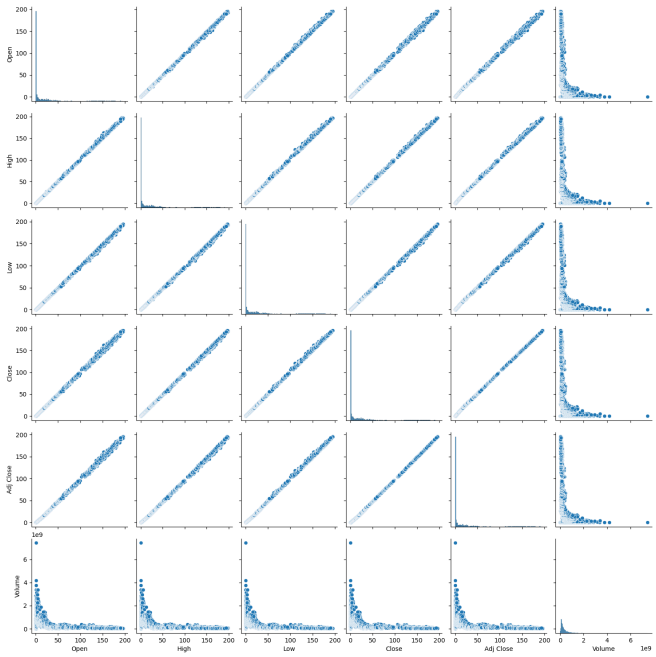


Figure 2: Pairplot of Features

The subplots show the relationships between features. For the features 'Open', 'High', 'Low', 'Close' and 'Adj Close', we can find that these features display strong positive correlations with each other, because they are usually very close on any trading day and they are related to each other.

5.3 Correlation matrix

The correlation matrix is a useful statistical tool used to calculate the strength of linear relationships between in the dataset. Each element value ranges from -1 to 1. The values close to -1 or 1 indicate the strong negative and positive correlation, while the values close to 0 indicate non-linear relationship.
Figure 3 shows the correlation matrix between various features in stock data. The features (Open, High, Low, Close and Adj Close) show the strong positive correlation, which means that they always move at the same time during the trading process.

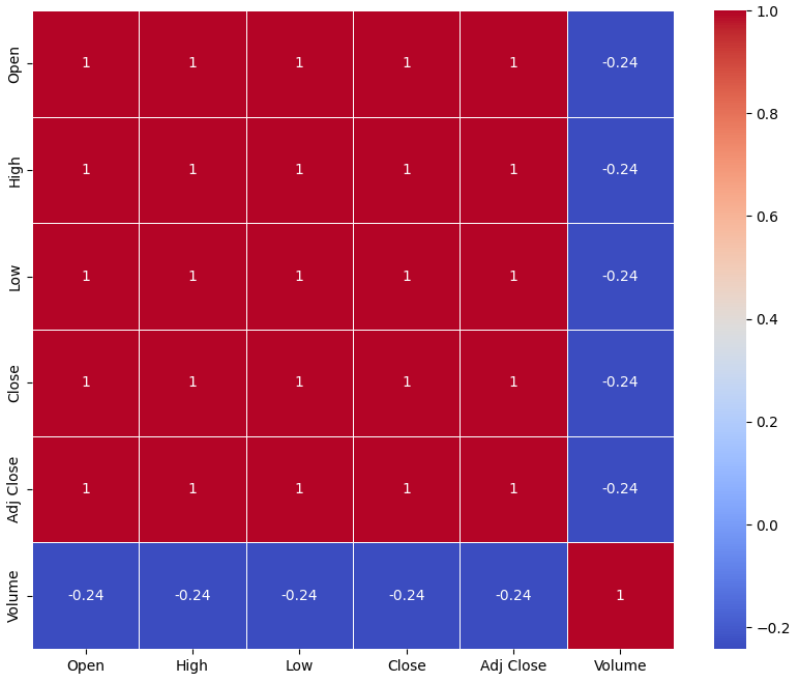


Figure 3: Correlation Matrix

6 Model

6.1 Long Short Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) aimed at dealing with learning long-term dependencies. This model is very effective in learning the characteristics in time and space because LSTM is able to utilize its internal state, which is named as memory, to handle temporal intervals and long-term dependencies in sequence data.

We use three LSTM layers to help model learn complex patterns in the data and predict output values through Dense layer.

```
from keras.models import Sequential
from keras.layers import LSTM, Dense
# LSTM
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(
   train_features.shape[1], 1)))
model.add(LSTM(units=50, return_sequences=True))
model.add(LSTM(units=50))
model.add(Dense(units=1))
```

6.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm widely used in the field of machine learning for classification and regression analysis. Its main idea is to find an optimal hyperplane to separate different categories of data points, so that the distance (i.e., interval) between

the data points projected on this hyperplane and the hyperplane is as large as possible. Support vector machines are characterized by their ability to handle high-dimensional data and perform well on small sample data sets.

We create and train a support vector regression model to solve regression problems, using sigmoid kernel function as the kernel function.

```
from sklearn.svm import SVR
# SVM
model = SVR(kernel='sigmoid', C=100, gamma=.0073)
model.fit(x_train, y_train)
```

6.3 Convolutional Neural Network (CNN)

CNN is a deep learning model that is widely used in image recognition, computer vision and other fields. The core components of CNN include convolutional layers, pooling layers, activation functions and fully connected layers. The training of CNN usually uses optimization methods such as backpropagation algorithm and gradient descent to minimize the loss function of the model, so that the model can learn the feature representation of the input data and accurately predict new data.

In the experiment, we use a one-dimensional convolutional layer to extract features, then reduce the feature map size through a pooling layer, then extract time-related features in the sequence data through two LSTM layers, and finally output through a fully connected layer.

```
from keras.models import Sequential
from keras.layers import Conv1D, MaxPooling1D, LSTM, Dense, Flatten
# CNN
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=3, activation='relu',
   input_shape=(60, 1)))
model.add(MaxPooling1D(pool_size=2))
model.add(LSTM(units=50, return_sequences=True))
model.add(LSTM(units=50))
model.add(Dense(units=1))
```

7 Result

In the project, we use three different models (LSTM, SVM, and CNN) to predict the closing price of Apple's stock. The following table 1 is the performance evaluation results of each model:
By combining the training set and the test set, we can get the stock prediction curves obtained using different models. In figure 4a, 5a and 6a, the blue curves represent the actual stock price, while the orange curves represent the predicted stock price. What's more, figure 4b, 5b and 6b show the curve combination of the training data and the testing data, where the blue curves represent the training data, the orange curves represent the actual stock price in the testing data, and the green curves represent the predicted stock price in the testing data.

Model	MSE	$RMSE$	MAE	R^2
LSTM	10.92	3.30	2.64	0.96
SVM	76.97	8.77	7.26	0.72
CNN	17.36	4.17	3.41	0.94

Table 1: Evaluation results of stock prediction models

(a) From 2021 to 2024 (Testing data)

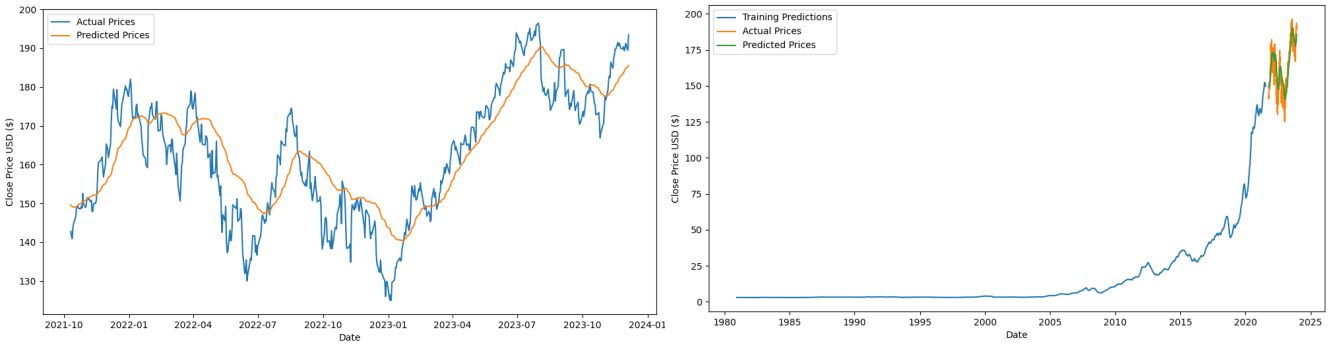
(b) From 1980 to 2024 (Training and testing data)

Figure 4: Stock price prediction using LSTM

From the result table 1 and figure 4, 5 and 6, we can find that LSTM model performs best in the stock price prediction. LSTM model can better capture the time series characteristics of stock prices so the prediction results have high accuracy and low volatility. The performance of SVM model is relatively weak, due to the limited ability of SVM model to process nonlinear and high-dimensional time series data. In the subsequent improvement space, we can use more complex or more adaptable models to process the dataset. For CNN model, it shows relatively high prediction ability. Through its powerful feature extraction capabilities, CNN model can display good adaptability and prediction accuracy when processing complicated stock market data. As a result, these results can provide valuable reference for future stock market prediction research.

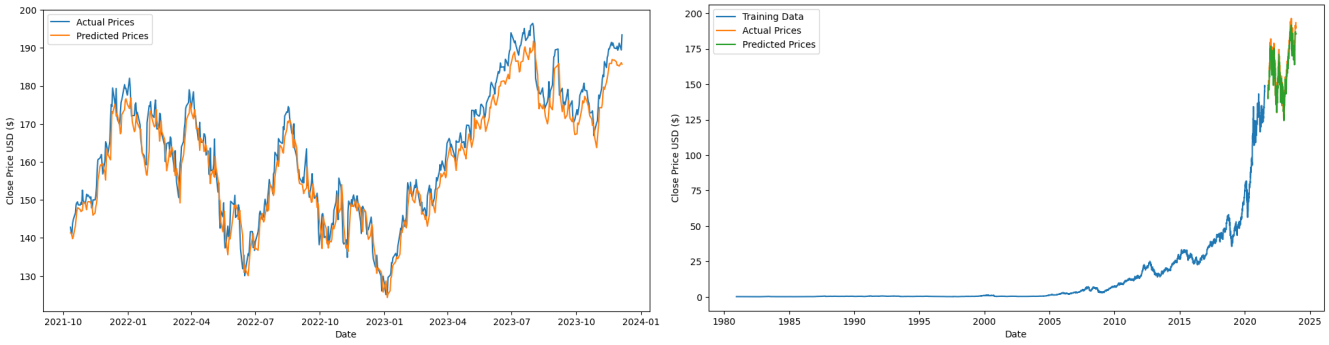
8 Future improvement

We can consider the performance and application scope of the enhanced model from multiple perspectives. In the field of data augmentation, we can enhance the generalization ability of the model by extending the time range of the data set or adding more stock data. What's more, we can perform model improvement and algorithm optimization, such as adjusting the parameters of existing models or utilizing more complex neural network to improve prediction accuracy.



(a) From 2021 to 2024 (Testing data) (b) From 1980 to 2024 (Training and testing data)

Figure 5: Stock price prediction using SVM



(a) From 2021 to 2024 (Testing data) (b) From 1980 to 2024 (Training and testing data)

Figure 6: Stock price prediction using CNN