Decision Support Systems

Shivaram Rammohan
202202968@post.au.dk
Department of Computer Engineering
Aarhus University
Aarhus, Denmark

Instructors:
Christian Fische Pedersen & Christian Marius Lillelund

Group 11

Project 3 - Forecasting

Contents

Introduction:

A time series is a sequence of observations recorded over time that is then used later for analysis and time series forecasting involves analysing this data to identify patterns and trends that can be used to make predictions about the values that can take place in the future. Time series data can have a variety of characteristics such as trend, seasonality, cyclicality, and randomness which are then later used for analysis and building forecasting models.

In this project, we are focusing on building a time series model based on stock market data. The data will then later be used to build an ARIMA, SARIMAX, and LSTM model and each of those models will then be compared with their results and we will conclude the results based on the observation from those models that we build.

Project description:

The project is done for the Forecasting part in which the time series models are designed and built upon and must be optimized based on forecasting accuracy, simplicity, and explain-ability/interpret-ability in such a way that the user can easily understand. To check for their accuracy, they are compared against the state-of-the-art models in the market and then evaluated on their accuracy.

Report structure:

The report will contain the first theory on how the data is obtained and what kind of models have been used and how its accuracy stands against the existing system and the other models that we have implemented against the data collected. The results and observations are then discussed to find the best model for the stock market data and are better at predicting future values.

Repository:

Material related to this project is stored in a GitHub account where the report and source code can be found. And, in the ZIP file that has been submitted for this project.

Project Background:

The project discusses predicting the time series data for future values and I have taken the stock market data as it's mostly been used to predict the future values. The stock market is a very complex system that has multiple external factors that affect the future values and hence any model that is being used to predict must consider the past values but also for seasonal and external variables that would affect them.

Approach:

This project achieves the problem statement by using a Python package known as yfinance (yahoo finance API) which allows the user to gather stock data and then the data obtained is then modeled into different models and they are then optimized and then compared to the previous result of those models and choosing the best model that is available. The best of each model is then compared and discussed and finally we conclude the best available model and our observations of why it is the best of all the models.

Dataset:

The dataset that we are using in our models is obtained through the Python package yfinance which will obtain the data from a given interval then this is then stored in a panda's data frame which then makes a panda's series of the Close value which as shown in Fig 1

```
import yfinance as yf
import matplotlib.pyplot as plt
```

```
# Define stock ticker and date ra
nge
ticker = "AAPL"
start_date = "2015-01-01"
end_date = "2022-01-01"

# Download stock data from Yahoo
Finance
data = yf.download(ticker, start=
start_date, end=end_date)

# Select the "Close" column
df = data["Close"]
plt.plot(df, label="APPLE Stock D
ata")
plt.show()
```

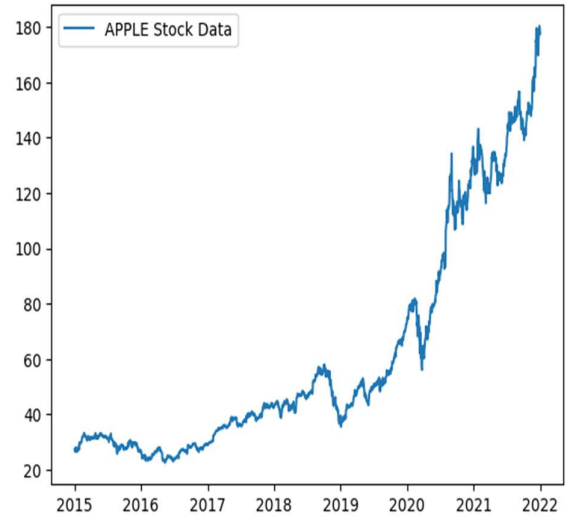

FIGURE 1: APPLE STOCK DATA

Basic Flowchart algorithm to train models:

The Flow diagram as shown in Fig 2 is a basic sketch of the steps that is repeated throughout the project to obtain the forecasted data for each model and then compared with each other for evaluation.

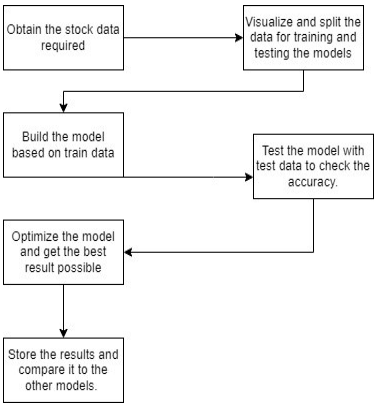

FIGURE 2: FLOW DIAGRAM FOR FORCASTING THE STOCK DATA

Training the ARIMA Model:

The ARIMA (Autoregressive Integrated Moving Average) model is the most widely used to predict time series data. It uses a combination of autoregression (AR), differencing (I), and moving average (MA) components. When dealing with the stock market data we have a lot of non-stationary data and hence we can use differentiation in the data to make the data stationary and achieve stationary data.

The AR component of the ARIMA model refers to the autoregression model where the past values are used to predict the future value of the same time series. The AR component of the model is represented by the parameter p which means the amount of weightage to be given to the past value to predict the future values.

The MA component of the ARIMA model refers to the moving average model, which predicts future values based on past errors, the error usually means the difference between the actual and predicted value. The MA component of the model is represented by the parameter q which means the amount of weightage to be given to the past errors to predict the future values.

The ARIMA model combines the AR, MA, and I components to model the time series data.

The parameters p, d, and q are chosen based on the characteristics of the data and the results of model diagnostics, such as the autocorrelation function (ACF) and partial autocorrelation function (PACF). The ARIMA model is estimated using maximum likelihood estimation (MLE), and the parameters are chosen to minimize the residual sum of squares (RSS).

Once the ARIMA model is estimated, it can be used to make forecasts by iteratively applying the prediction equation to the observed data. The accuracy of the ARIMA model can be evaluated using various metrics, such as mean squared error (MSE) and mean absolute error (MAE).

Experiments and results:

The ARIMA model is constructed and trained against 80% of the data (test data) and then tested against the rest of the 20% and the AIC value is obtained which is then iteratively made to traverse through multiple parameters and the best model is obtained by adjusting the (p, q,d) values in the model.

The basic model is first built upon with the parameters (1,1,1) and the results are shown in Fig 3

FIGURE 3:ARIMA MODEL RESULT FROM ORDER (1,1,1)

The model is then fed into multiple combinations of the order and the best value obtained is of order (8,2,7).And the graph for stated value is as shown in Fig 4.

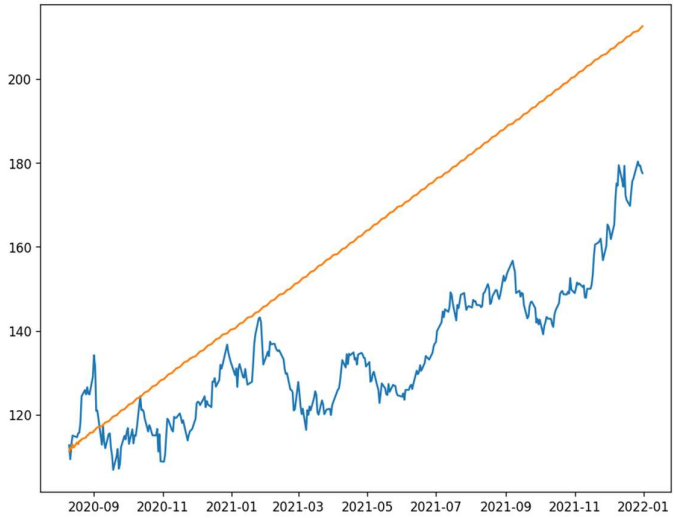

FIGURE 4: ARIMA BEST MODEL PREDICTED

SARIMAX Model:

SARIMAX is a model best suited for occasional peak changes in the data. In the sense that the SARIMAX is best suitable for values that have occasionally outside factors in play. Hence, the stock market has a lot of outside factors influencing the closing value. Hence, we use this model to predict and see if it's a better solution than ARIMA.

SARIMAX considers three main components:

Autoregressive (AR): This looks at how the current value of the time series is related to its past values. It considers that what happened in the past can help predict what will happen next.

Integrated (I): This considers how the data changes over time. It looks at whether the time series is getting bigger or smaller over time and adjusts predictions accordingly.

Moving Average (MA): This component looks at the relationship between the current value and the average of past values. It considers how the data fluctuates around its average.

The "X" in SARIMAX stands for eXogenous variables. These are additional factors that can influence the time series but are not

directly related to its past values. For example, if the quarterly earnings report influences the stock value hence this can be used to add as an additional external factor.

Experiments:

The SARIMAX model is fed into an automated method that can automatically figure out the best model when iteratively measuring the AIC and other factors in determining the best model for the given dataset. The below code represents the above method to get the best SARIMAX model.

```
sxmodel = pm.auto_arima(df,
exogenous=quarterly_data["Close"],
                        start_p=8, start_q=7,
                        test='adf',
                        max_p=8, max_q=8, m=12,
                        start_P=0, seasonal=True,
                        d=None, D=1, trace=True,
                        error_action='ignore',
                        suppress_warnings=True,
                        stepwise=True)
sxmodel.summary()
```

The obtained model is then made to predict the values for the stock market and is plotted as shown in the Fig 5.

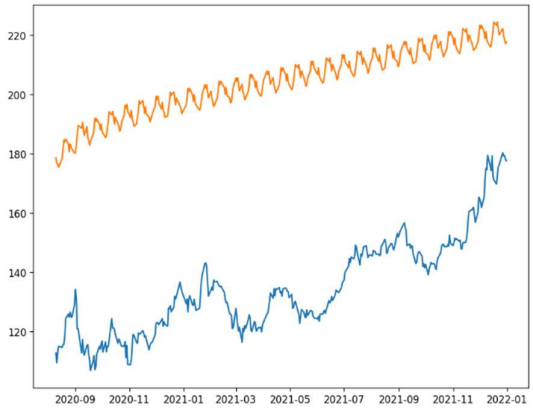

FIGURE 5: SARIMAX MODEL PREDICTED BY USING ACCURACY FACTOR

LSTM Method:

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture used for processing sequential data. Let's break it down in simpler terms.

Let's take the stock market as an example now if the stock has certain peaks and pits where the stock prices can fall, and rise based on certain patterns the LSTM remembers these important patterns and uses them to predict the next value.

Long-Term Memory: LSTM can remember information from earlier parts of the sequence as it stores important information in its "long-term memory" to help understand the pattern from previous values.

Short-Term Memory: LSTM also has a "short-term memory" that quickly processes new information as it comes in. It decides which parts of the new information to pay attention to and how much importance to give them as when the stock has a huge rally, and they automatically make much higher peaks than previous data.

Forget and Remember: LSTM can forget or discard irrelevant information from its long-term memory. This can be used as certain one-off incidents like an irrelevant spike due to very rare external events and such.

Sequences and Patterns: LSTM is designed to handle sequential data, which means it can understand and make predictions based on the order of the numbers in the input. This is crucial when dealing with the stock market predictions as data needs to be fed and patterns need to be recognized as prediction are made.

By using this combination of long-term and short-term memory, LSTM can effectively learn patterns and dependencies in the sequential data. It can capture complex relationships between past and current inputs and use them to make predictions about what might happen next. Hence, it is especially powerful when dealing with data that has long-term dependencies and requires the

model to remember and utilize information from the past to make accurate predictions. Thus, this is very effective in predicting the stock market and is widely used in algorithmic trading.

Explaining the logic used in LSTM:

The LSTM as we did before I have obtained the data from yfinance and separated the data into train (before 2020 data) and test data(after 2020 data). The Model Building is as follows.

The construction of the LSTM model using the Keras library. The model is defined as a Sequential object, which allows for building a linear stack of layers.

The first LSTM layer is defined with LSTM (units=50, return_sequences=True, input_shape=(self.X_train.shape[1], 1)). This layer has 50 LSTM units, returns sequences, and expects input of shape (self.X_train. shape[1], 1).

The second LSTM layer is defined with LSTM (units=50). This layer has 50 LSTM units and doesn't return sequences.

A dense output layer with one unit is added to the model using Dense (1).The model is compiled with model.compile(loss='mean_squared_error', optimizer='adam'), specifying the loss function as mean squared error and the optimizer as Adam.

Model Training:

The fit method of the model to train on the training data (self.X_train and self. y_train).The epochs parameter determines the number of times the model will iterate over the entire training dataset.The batch size parameter specifies the number of samples per gradient update. The LSTM model is built with two LSTM layers and a dense output layer. The model is trained using the mean squared error loss function and the Adam optimizer.

```
# Build the LSTM model
from keras. models import Sequen-
tial
from keras. layers import Dense
from keras.layers import LSTM
model = Sequential()
model.add(LSTM(units=50, re-
turn_sequences=True, in-
put_shape=(X_train.shape[1], 1)))
model.add(LSTM(units=50))
model.add(Dense(1))
model.com-
pile(loss='mean_squared_error',
optimizer='adam')

# Fit the model to the training
data
model.fit(X_train, y_train,
epochs=25, batch_size=64)
```

Results for LSTM model:

The LSTM model is plotted and the results are shown below:

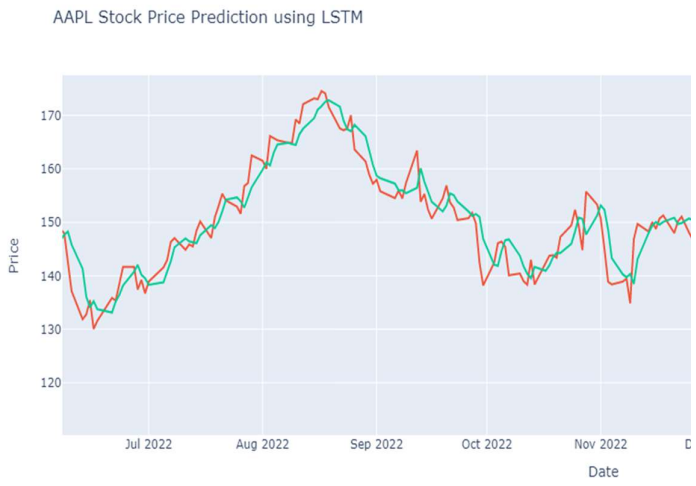

FIGURE 6: RESULTS FOR LSTM MODEL PREDICTED

The LSTM model can predict the stock values with greater accuracy then ARIMA and SARIMA models that we discussed before.The LSTM model has better accuracy than ARIMA and SARIMAX as predicted results are much closer to the stock closing prices than the other models.

Comparison to other models:

The LSTM (Long Short-Term Memory) models are often considered better for stock market predictions compared to ARIMA (Autoregressive Integrated Moving Average) and SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Variables) models for the below reasons as listed.

Capturing Nonlinear Patterns:

Stock market data is known to exhibit complex and nonlinear patterns, including trends, seasonality, and irregularities. LSTM models are well-suited for capturing such patterns because they can learn long-term dependencies and nonlinear relationships within the data. In contrast, ARIMA and SARIMAX models assume linear relationships and struggle to capture nonlinear patterns effectively.

Handling Long-Term Dependencies:

LSTM models have memory cells that can retain information over extended periods, making them capable of handling long-term dependencies. In stock market prediction, past prices and trends often have an impact on future prices, and LSTM models can effectively capture and utilize this historical information. ARIMA and SARIMAX models, on the other hand, consider only a fixed number of lagged values and may not adequately capture long-term dependencies.

Dealing with Irregularities and Outliers:

Stock market data is susceptible to irregularities, sudden changes, and outliers due to various factors like economic events, news, and market behaviour. LSTM models are more robust in handling such irregularities because they can adapt and adjust their predictions based on the changing patterns in the data. ARIMA and SARIMAX models, which rely on past observations and assume stationarity, can be more sensitive to outliers and may produce less accurate predictions.

Incorporating Exogenous Variables:

SARIMAX models allow for the inclusion of exogenous variables, which can provide additional information to enhance predictions. However, LSTM models can also handle exogenous variables by incorporating them as additional input features. LSTM models can effectively learn and leverage the relationships between the exogenous variables and the target variable, potentially improving prediction accuracy.

Handling Variable Time Intervals:

Stock market data may have irregular time intervals between data points, such as trading days, hours, or minutes. LSTM models can handle variable time intervals more flexibly by working with sequences of data rather than relying on fixed time steps. ARIMA and SARIMAX models typically assume equally spaced time intervals and may not handle irregular time intervals as effectively.

Overall, LSTM models' ability to capture nonlinear patterns, handle long-term dependencies, adapt to irregularities, and incorporate exogenous variables make them well-suited for stock market predictions compared to ARIMA and SARIMAX models, which are limited by their assumptions and linear nature. However, it's important to note that the choice of the best model depends on the specific characteristics of the data, the problem at hand, and empirical evaluation.

Conclusion:

The Forecasting time series we have tried to predict the stock market value using ARIMA , SARIMA and LSTM models and have discussed the results of all the models and obtained the best results for such models. The LSTM has shown cased that it can be the best option for predicting the stock market data as LSTM could Handle lot of irregularities and can have the ability to predict newer patterns

and while withholding older patterns for predicting the stock market. Hence , we can conclude that LSTM is a better fit for predicting the stock market data.

References:

1. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control. Wiley.
2. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. OTexts.
3. Taylor, S. J. (2017). Forecasting in Univariate Time Series: ARIMA and SARIMA Models. Handbook of Economic Forecasting, 2, 89-141.
4. Brownlee, J. (2018). Long Short-Term Memory Networks with Python. Machine Learning Mastery.
5. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.