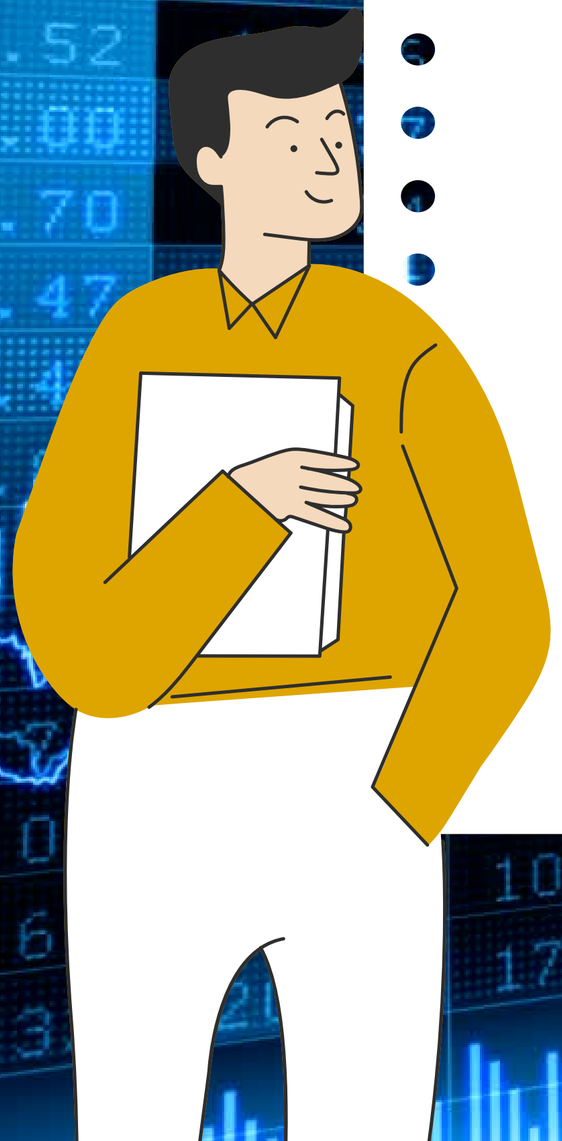
PROJECT PRESENTATION

DS303

STOCK MARKET TECHNICAL ANALYSIS



Contributors

Abhimanyu Chauhan
(200040006)

Avichal Jain
(200020035)

Balbir Yadav
(200100042)

Eknoor Singh
(200020051)

CONTRIBUTIONS

1. Abhimanyu Chauhan
2. Avichal Jain
3. Balbir Yadav
4. Eknoor Singh

1. Reading and understanding technical indicators in the stock market and implementaion in the Linear Regression Model to predict the future stock price.
2. Linear Regression, DecisionTree Classifier, RandomForest Classifier and SVM Implementation
3. Reading and understanding Stock market, ARIMA model and Implementation
4. LSTM Implementation, Conclusions and Documentation

PROBLEM

To technically analyze the stock market using concepts learnt in the course

Motivation / Understanding the Problem Statement

Broadly, stock market analysis is divided into two parts – **Fundamental Analysis and Technical Analysis.**

- Fundamental Analysis involves analyzing the company's future profitability on the basis of its current business environment and financial performance.
- Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market.

As you might have guessed, our focus will be on the technical analysis part. Our interest in using Tech in finance motivated us to pursue this project.

ARIMA MODEL

AIM: To predict the future stock price using ARIMA model

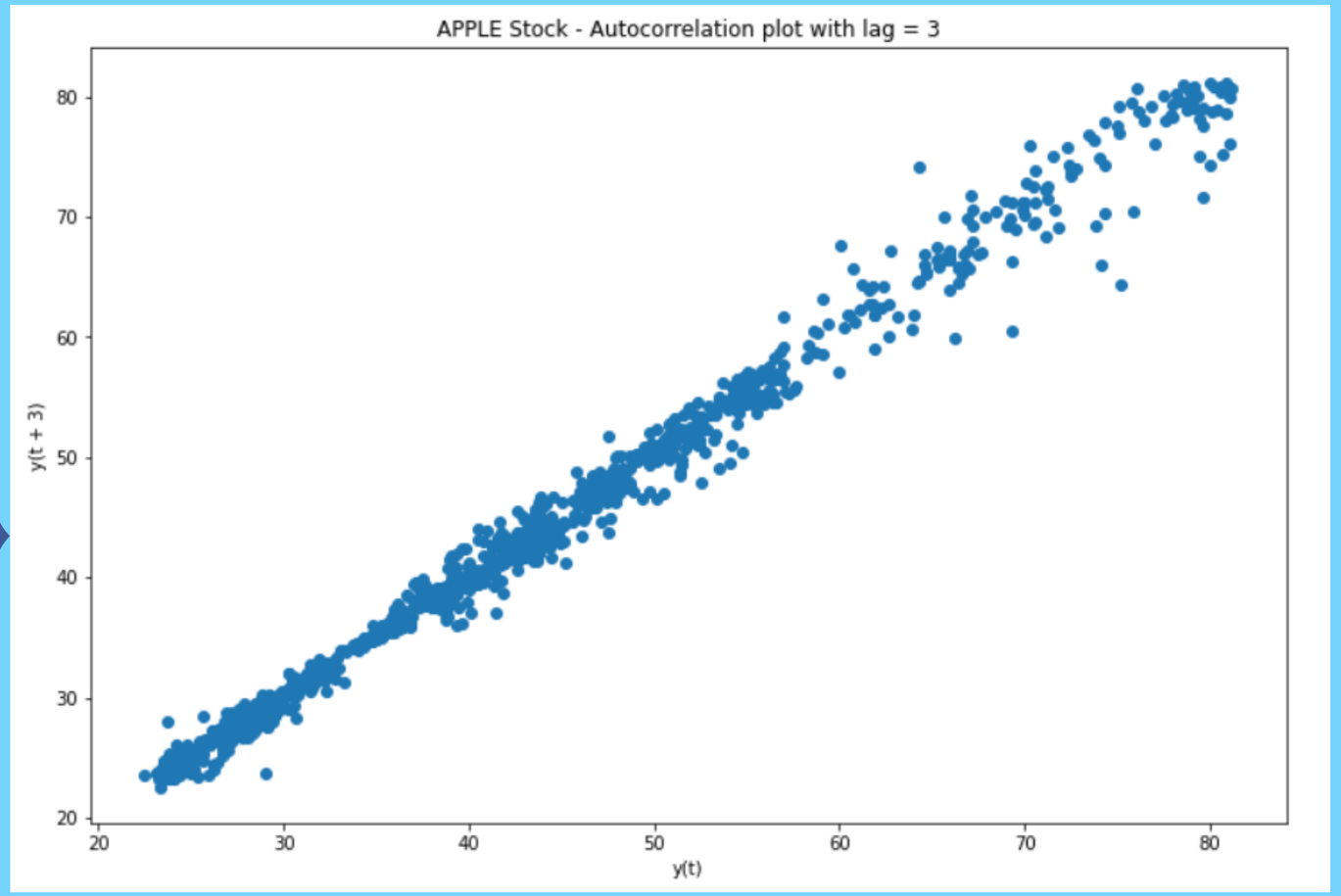
ARIMA stands for Auto Regressive Integrated Moving Average
It is a time series model which is used to as the machine learning technique to analyse and predict future stock prices based on historical prices.

DATASETS, BASIC ANALYSIS

we have used APPLE stock dataset available at https://finance.yahoo.com/

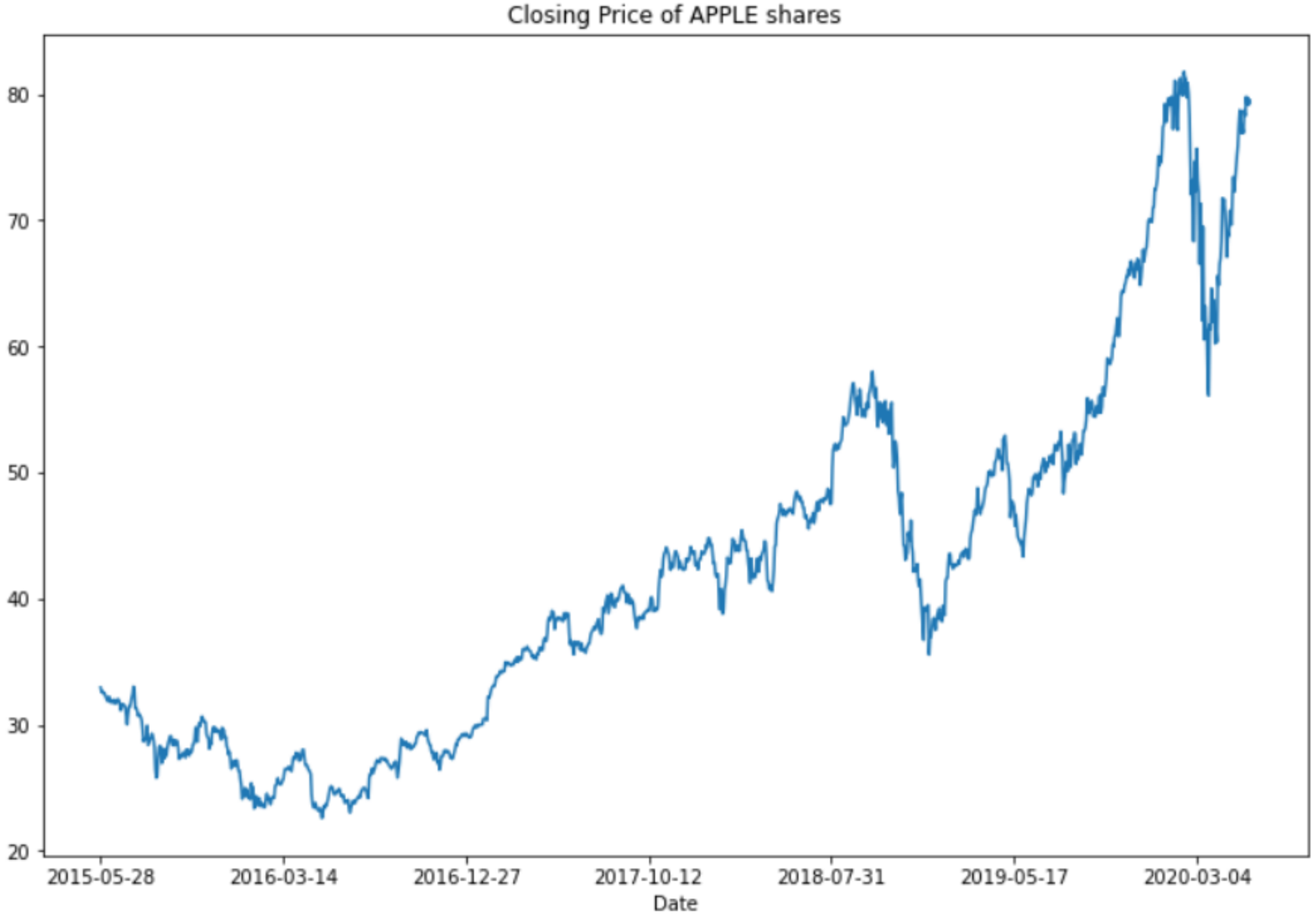
Imports

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from pandas.plotting import lag_plot
import datetime
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
```



As there is autocorrelation in the data, ARIMA model works well. Generally, ARIMA is applied to this type of data

Dataset Visualisation | Dataset split | Parameters



Data Split for this project

- 80% training data
- 20% test data

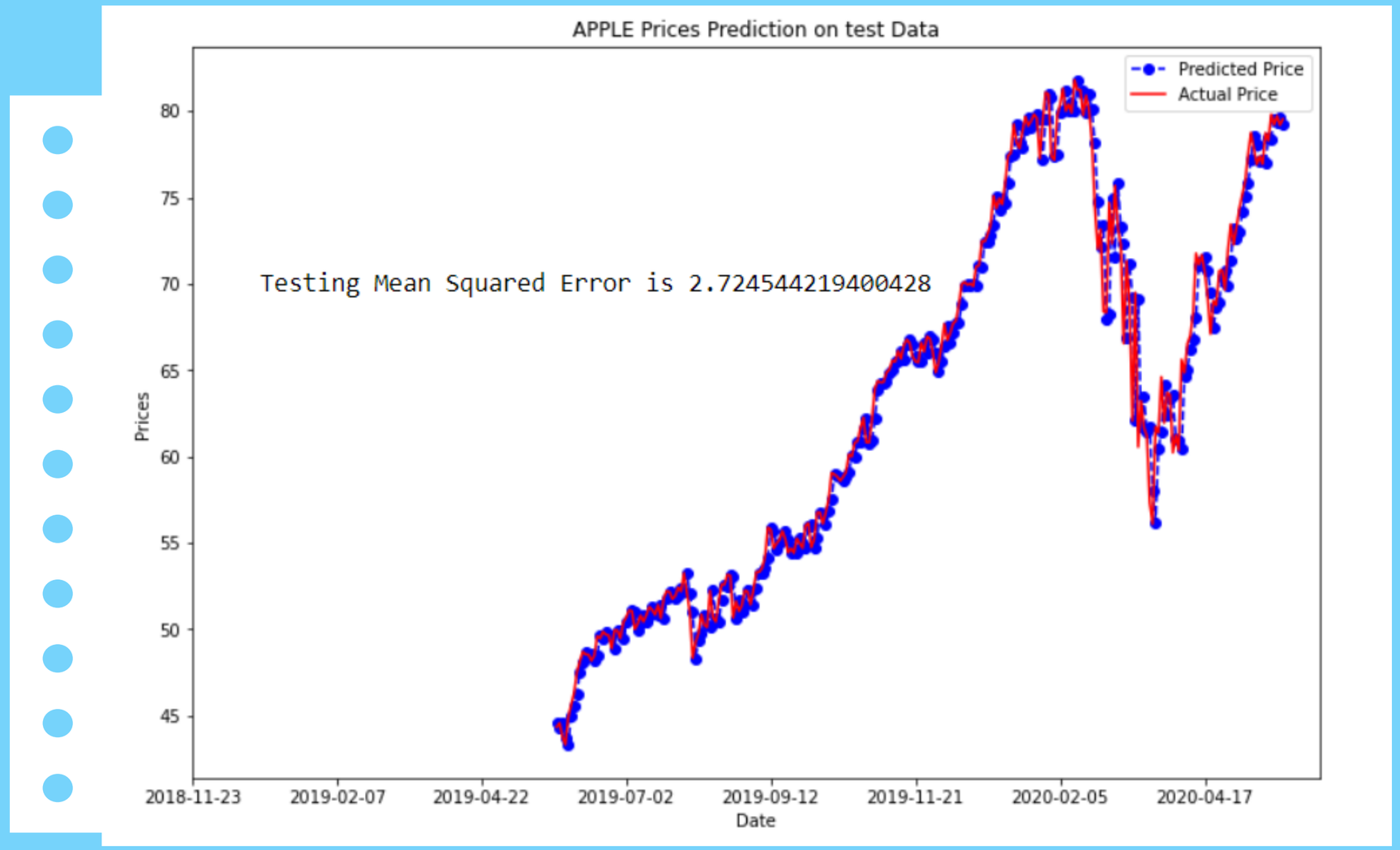
ARIMA Model parameters:

- p : number of lag observations
- d : degree of differencing
- q : size of moving avg window

p=4 d=1 q=0

Results

ARIMA Implementation on APPL Stock

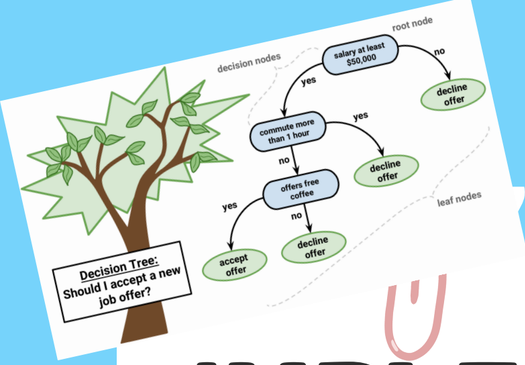


Note: In this graph only test data has been plotted

CONCLUSIONS

(ARIMA MODEL)

- ARIMA model on APPL dataset seems to be giving out very good results.
- MSE is approximately around 2.7245 which is very low. This implies that model have good prediction as it can be seen clearly in the comparision graph.
- The results of this model seems to be very good and therefore this model can be used to predict the future stock price and these predictions can be made to gain profit or avoid losses in the stock market.



IMPLEMENTATION OF MULTIPLE MODELS

AIM: To perform the analysis of the Stock market using models like Linear Regression, DecisionTree Classifier, RandomForest Classifier and Support Vector Machine and comparing their accuracies.

LINEAR REGRESSION

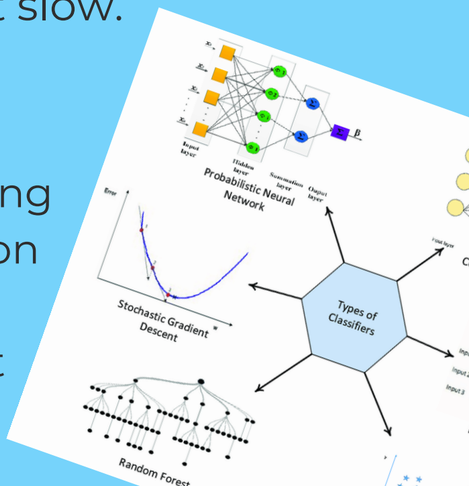
Linear regression is a linear model, which means it works really nicely when the data has a linear shape. But, when the data has a non-linear shape, then a linear model cannot capture the non-linear features.

DECISION TREE AND RANDOM FOREST

used at capturing the non-linearity in the data by dividing the space into smaller sub-spaces depending on the questions asked.
Decision trees are very easy as compared to random forests. A decision tree combines some decisions, whereas a random forest combines several decision trees. Thus, it is a long process, yet slow.

SUPPORT VECTOR MACHINE

SVM is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification.



```
test_score': 0.9839825556009695,
```

```
Model Accuracy:  0.57155983329
```

```
Model Accuracy 0.851752021563342
              precision    recall  f1-score   support

           0       0.85      0.85      0.85       185
           1       0.85      0.85      0.85       186

    accuracy                           0.85       371
   macro avg       0.85      0.85      0.85       371
weighted avg       0.85      0.85      0.85       371
```

```
[[144  33]
 [ 33 161]]
              precision    recall  f1-score   support

           0       0.81      0.81      0.81       177
           1       0.83      0.83      0.83       194

    accuracy                           0.82       371
   macro avg       0.82      0.82      0.82       371
weighted avg       0.82      0.82      0.82       371
```

LINEAR REGRESSION

SVM

RANDOM FOREST

DECISION TREE

Results

Accuracies of various models

CONCLUSIONS

- The Linear Regression model has the best test score among all the four. It has a test score of 98.4%. Since this is true, it clearly indicates that our true model is linear.
- SVM has the least accuracy of 57.2%. This means that our dataset lacks a clear classification boundary
- The Random Forest can generalize the data in a better way. This randomized feature selection makes Random Forest much more accurate than a Decision Tree as clear from our model.

LSTM MODEL

AIM: To predict the future stock price using LSTM model

Long Short Term Memory Network is an advanced RNN, a sequential network, that allows information to persist.

The model is extremely accurate and has been covered in the project for comparison purposes.

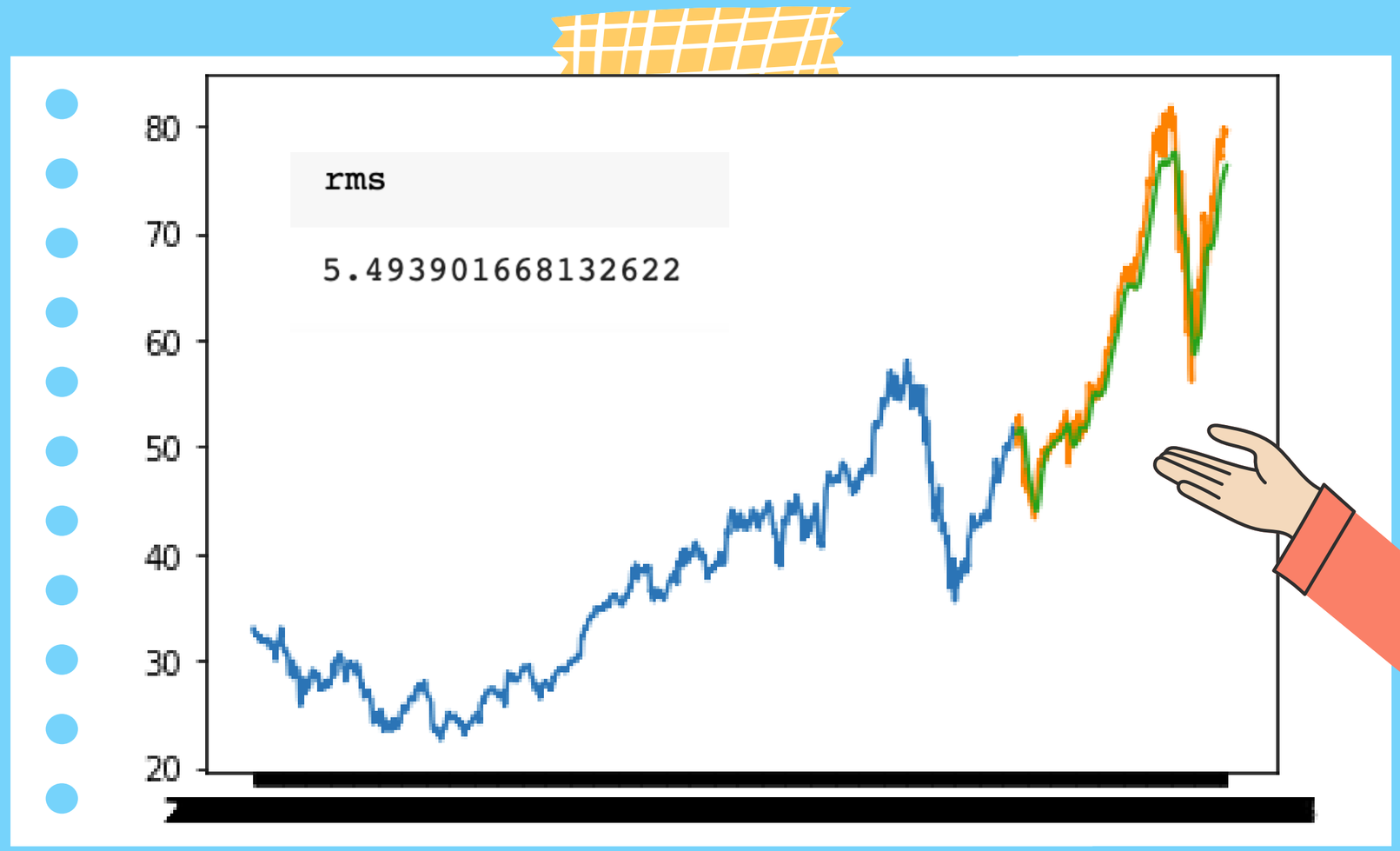
LSTMs are widely used for sequence prediction problems and have proven to be extremely effective. The reason they work so well is that LSTM can store past important information and forget the information that is not.

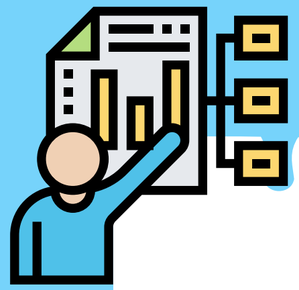
Resarch Paper [Reference/Motivation]

Adil Moghar, Mhamed Hamiche,
Stock Market Prediction Using LSTM Recurrent Neural Network,
Procedia Computer Science,
Volume 170,
2020,
Pages 1168-1173,
ISSN 1877-0509,
https://doi.org/10.1016/j.procs.2020.03.049.
(https://www.sciencedirect.com/science/article/pii/S1877050920304865)

Results

LSTM Implementation on APPL Stock





CONCLUSIONS
(LSTM MODEL)

Evidently, the best model so far proves to be the LSTM model, and hence was covered in the project.

The LSTM model can be tuned for various parameters such as changing the number of LSTM layers, adding dropout value or increasing the number of epochs. **But are the predictions from LSTM enough to identify whether the stock price will increase or decrease? Certainly not!**

1. LSTMs are **prone to overfitting** and it is difficult to apply the dropout algorithm to curb this issue.

   Dropout is a regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network

2. The prices of these stocks also depend on the news about the company and other factors like demonetization or merger/demerger of the companies. There are certain intangible factors as well which can often be impossible to predict beforehand.

LINEAR REGRESSION WITH TECHNICAL INDICATORS

Linear regression requires an assumption that variables in the data are independent to be made to be effective. our date values aren't suitable as our independent variable. In order to tackle this problem, we will be using technical indicators such as Moving Averages.

LINEAR REGRESSION

Linear regression is utilized in business, science, and just about any other field where predictions and forecasting are relevant.

Adding Technical Indicators

Commonly used technical indicators include moving averages (SMA, EMA, MACD), the Relative Strength Index (RSI), Bollinger Bands (BBANDS), and several others. To add our technical indicators we'll be using the pandas_ta library.

DATASETS, BASIC ANALYSIS

we have used APPLE stock dataset available at https://finance.yahoo.com/

The plot of APPLE historic pricing

Imports

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
```

Dataset Visualisation | Dataset split | Parameters

Data Split for this project

- 80% training data
- 20% test data

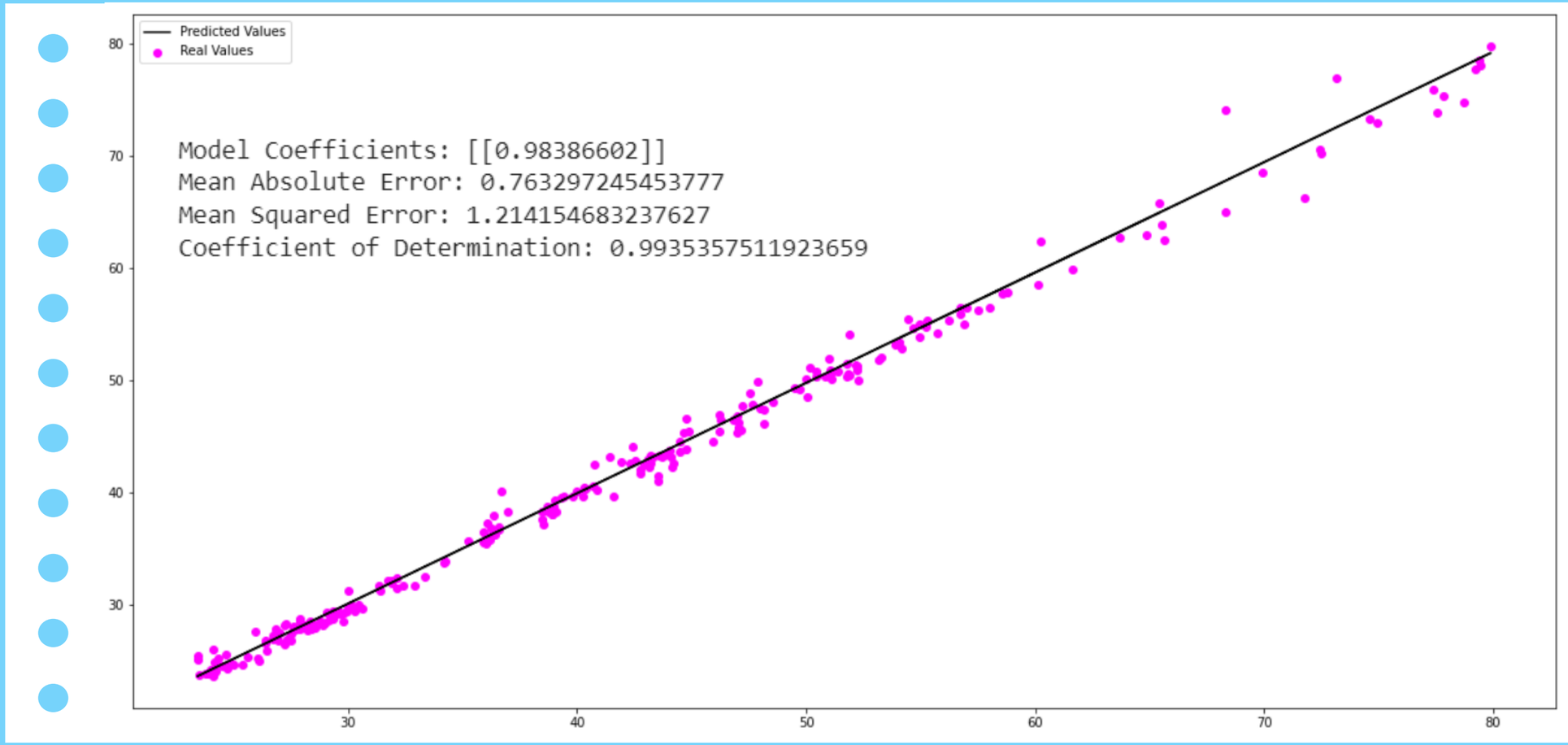
exponential moving

average calculated over a 10-day period.

```
#inspecting how the EMA value tracks with the close prices
```

Results

Linear Regression Implementation



Note: In this graph only test data has been plotted

“

The Project provides us the motivation to explore sophisticated models further!

THANK YOU!

Looking forward to implementing the models further in Summer!