

Solent University

Faculty of Business, Law, and Digital Technologies



Software Engineering: 2021/2022

COM616: Dissertation

‘Intelligent Income and Expenditure System’

Rahul Shrestha

Q13671294

GitHub Repository: <https://github.com/shrestha-rahul/Dissertation-Project>



Supervisor: Darren Cunningham

Date of Submission: May 2022

Acknowledgement

My sincere thanks go to my supervisor and mentor, Darren Cunningham, for guiding me through this work, offering me a great deal of advice, encouragement, and feedback.

Abstract

Financial struggles are frequent within the population, with stress arising from living from pay check to pay check. In order to support those who find themselves to be worried about their finances, I propose an application that will allow them to clearly see their income and expenditures, allowing them to budget their money. Further, I propose providing stock market guidance by the use of technical analysis and machine learning models as a means to forecast stock prices and forecast trading signals (buy, sell or hold). By doing so, not only stress will be relieved, but also capital gains can be increased. In my research, LSTM and ARIMA models were used to forecast stock prices. However, the results were insignificant. In contrast, classification models, especially Random Forest, produced promising results in the classification of trading signals. To address the limitations presented by this project, it is recommended that the current models used in this app should be revisited and revised using more technical indicators and other forms of technical analysis as input data. Additionally, the prototype application should be improved based on feedback from a target audience.

Table of Contents

Acknowledgement	1
Abstract	1
1. Introduction	3
2. Literature Review.....	5
2.1 Debt.....	5
2.2 Budget Apps	6
2.3 Stock Investment	6
3. Design Implementation	9
4. Methodology	13
4.1 Regression	21
4.1.1 Regression: Exploratory Data Analysis & Data Cleaning	21
4.1.2 Regression: Data Pre-Processing	31
4.1.3 Regression Model: Modelling and Results	32
4. 2 Classification	37
4.2.1 Classification: Exploratory Data Analysis & Data Cleaning.....	38
4.2.2 Classification: Data Pre-Processing.....	41
4.2.3 Classification Model: Modelling & Results	43
5. Results.....	53
6. Limitations & Recommendations	55
References	56
Appendix- A (Mobile Wireframes).....	73
Appendix B-(Web Wireframe)	76
Appendix-C (Mobile-Mock-up)	77
.....	78
.....	79
Appendix D-(Web-Mock-Up)	80

1. Introduction

Debt is a term that no one wants to hear when it comes to their personal finance. Debt can be defined as owing something, usually money, to someone (Chen, 2022). Research from Close Brothers (2019) has found that almost 94% of UK employees are suffering from money worries and almost 77% of employees have said that it has affected them at work. Additionally, The Money Charity has stated that as of November 2021, the average total household debt in the UK is £63,122. There are many factors as to why someone can get into debt; Zhen (2022) has stated one of the most common causes was due to poor money management. This can take form in several different ways, such as impulsive buying, using overdraft and simply spending more than you are earning. Whistl (2017) have found that 91% of the nation have admitted to making impulsive purchases every month and on top of that, Hall (2018) stated in a news article that an average UK adult will spend over £144,000 on impulsive buying during their lifetime. This could be due to the advancement of technology over the years which has allowed the rapid growth of e-commerce and in turn have amplified impulsive buying behaviours. Moreover, the rise of contactless payments and mobile wallets have also been seen to contribute overspending. A study conducted by Xu et al. (2019) found that using mobile wallets can lead to people spending more money, more frequently. Additionally, due to the COVID-19 pandemic, the UK's top retailers have stated that their online traffic has increased by 52% (Jobling, 2021).

Other common causes of debt include spending future money and having no savings, such as an emergency fund. Even though one of the most common attempts to save money is to store it away in a savings account, Money Helper (2022) have stated that leaving your money in a savings account is not the best option as the interest rate in the saving account is nearly always lower than the rate of inflation. Furthermore, Inman (2022) has reported that inflation in the UK has risen to its highest levels in 30 years, currently around 5.4%. The rise of inflation refers to an increase in prices and the decrease of purchasing power. This means that consumers can purchase less goods and services compared to before

(Davies, 2022). Moreover, Clark (2020) has reported that the average saving account interest rates have fallen to their lowest levels on record at 0.64% in 2020, meaning that the return on your money will be non-existent, especially when factoring in the increase of inflation rates.

Nevertheless, Barclays (2021) have stated in terms of accumulating more wealth for the future, it is better to invest into the stock markets rather than leave it in a savings account. The S&P 500, a stock market index that tracks the US 500 large-cap companies (Amadeo, 2022), has reported an average annual return of around 10.5% since its inception in 1957, beating the inflation rate and any other savings account (Maverick, 2022). However, investing is a lot riskier than storing your money in a savings account, therefore it is not advised for short-term goals, such as anything less than 5 years (Barclays, 2021). Stock markets are volatile, meaning the values of stocks can fluctuate and even drop in value drastically. For this reason, it is advised to aim to invest for at least 5 years as a longer time frame will allow your investments to recover over time (HSBC, n.d). Additionally, HSBC (n.d) have advised that before participating in saving or investing any money, it is important to have an emergency fund in case of any unexpected expense.

One of the reasons in why people don't invest in stocks is said to be due to the lack of knowledge around investing and stock markets. A news article in 2015 stated that one in five Americans that have not invested in stocks due to the fact they simply do not know enough (Gold, 2015). Additionally, Abhishek (2021) stated that 98% of India's population have nothing invested in the stock due to their lack of awareness, knowledge and the bad connotation the stock market gets in idea, where investing in stocks is considered gambling in India. He also mentioned that even the handful of people who are willing to invest are unable to due to the lack of proper guidance and platforms available, where they can learn about investing.

One of the best ways to manage expenses and control spending is to budget. By budgeting, individuals can be guided on their future income and expenses, which

can aid achieving financial goals as well as give the individual peace of mind surrounding their finances. Even with the advocacy from experts in favour of budgeting, less than half of U.S. citizens were reported to partake in financial budgeting in the millennium (Hogarth, Hilgarth & Schudardt, 2002). However, in more recent years, most Americans have since indicated the use of budgeting within their household (Bankrate Inc., 2015). With the growth of budgeting app development in more recent years, a quarter of respondents reported to have used an app or computer program to manage their budgets. Taking advantage of a budgeting app comes with many benefits, such as helping to keep track of spending and bills and allows the user to become more aware of their finances in an easily accessible format (Lake & Foreman, 2021).

2. Literature Review

2.1 Debt

A scoping review conducted by Harper et al. (2021) found that people involved in the criminal justice system are disproportionally indebt compared to the average person. They suggested that reducing debt in this population can improve re-entry outcomes and quality of life. Furthermore, Van Beek et al. (2021) systematic review found debt to be a risk factor for criminal behaviour. Thus, utilisation of a financial app could be particularly beneficial for those who have a criminal background, manage their debt. This could result in fewer crimes.

Additionally, A review conducted Swanton and Gainsbury (2020) found that debt problems led people to take part in gambling addiction which in turn resulted in bigger mental health problems. It was also stated gambling-related debt problem increased the likelihood of psychological distress, substance use, crime, and suicidality.

In a study by Ong et al. (2019), it was found that when those in debt were given debt relief, they had experienced significant improvements in their cognitive function and reports of less anxiety.

2.2 Budget Apps

It was found that individuals who follow a budgeting plan to guide their saving and spending demonstrated more financial practices, as well as health practices, than those who do not (O'Neill, Xiao & Ensle, 2017). However, this study relied entirely on self-report measures, heightening the risk of bias in the results. Additionally, the findings are correlational in nature, limiting causal inferences to be concluded. To form conclusions of causality, the literature would benefit from the implementation of prospective studies in future research.

Furthermore, French, McKillop and Stewart (2020) found that utilisation of finance smartphone apps significantly improved financial knowledge which translated to improved financial behaviours. Through further investigation, French, McKillop and Stewart (2021) found that those from low-income households specifically had improved financial literacy, displayed more financially capable behaviours, and self-confidence in financial decision making.

2.3 Stock Investment

Investing in the stock market is one of the ways that individuals can accumulate their wealth. However, most households (89%) in the U.S that are representative of the bottom quintile of the income distribution do not invest in stocks. Conversely, 82% of those within the upper quintile have stock holdings (Survey of Consumer Finances Chartbook, 2016). Research has shown that individuals reared in an economically adverse household results in a pessimistic view towards financial matters, and hence stock investment, limiting wealth accumulation (Kuhnen & Miu, 2017). Such findings provide potential reasoning for the disparity in stock investments regarding social economic status (SES).

When contemplating investing in stocks, many factors are considered. Namely, background risks, investment horizon, rare disaster, transaction factors, and costs of stock market participation are seen to be some of the strongest determinant amongst primary household financial decision-makers in the U.S (Choi & Robertson, 2020).

Plus, findings from research conducted by Franzen and Bradaric (2018) showed that there is a gap of knowledge when it comes to managing money and being financially aware, especially in college students. It was also stated that due to the poor money management skills, students had increased stress levels and were not performing well in their academics. It also led to some students dropping out of school. Additionally, they suggested that utilisation of budgeting apps could lead to student maintaining and attaining financial wellness.

Another factor that has been demonstrated to effect investment is gender. Montford and Goldsmith (2015) found that female participants invested less than males. Moreover, females reported lower financial self-efficacy, which is defined as the belief in one's ability to achieve once financial goals (Forbes & Kara, 2010), than males (Montford & Goldsmith, 2015).

Likewise, financial literacy has been shown to be correlated with stock market investment, with individuals being less likely to participate if they are less financially literate (Chu et al., 2017; van Rooij, Lusardi & Alessie, 2011).

Stock market prediction has gained a lot of focus in the past 10 years, in regard to using machine learning algorithms to predict stock trends and prices. Gandhmal and Kumar (2019) conducted a systematic review of 50 research papers and discovered that the most common technique used for effective stock market prediction were artificial neural networks (ANN) and fuzzy-based technique.

Additionally, they discovered that despite the numerous amount of research efforts, stock market prediction still has many limits such and proven to be a very complex task. Therefore, Pang et al (2020) aimed to develop an innovative neural network approach to achieve better stock market predictions by utilising a deep long short-term memory neural network (LSTM) with an embedded layer and a LSTM neural network with an automatic encoder to predict the stock market. However, their experiments accuracy only reached up 52.5% for an individual stock.

Furthermore, Zhou and Qu (2020) also conducted research to improve stock price prediction by incorporating company's accounting statistics and accounting data into the models' input data. They experimented 3 variation of the LSTM model including deep learning sequential LSTM, Stacked-LSTM and Attention-Based LSTM, along with a traditional ARIMA(Auto-Regressive Integrated Moving Average) model. They discovered that Attention-LSTM performed the best out of the other models in terms of prediction errors and even though Stacked-LSTM has a complex model structure, it did not improve the predictive power.

3. Proposal

Considering the aforementioned factors which result in financial insecurity and the evidence in support for budgeting and investing into stock, for my dissertation project I have proposed to develop a prototype app that functions as a financial budgeting system, with an additional element which will aid users to invest in the stock market. This additional element will consist of utilising machine learning models to forecast stock prices and forecast trading signals (buy, sell or hold) for a particular stock. Ultimately, this app will serve a large variety of people, particularly those who are at higher risk of economic adversity, by providing a user-friendly experience that will help to make understanding investing into the stock market easier and more accessible, whilst budgeting their finances.

4. Design Implementation

As the main intention of the project is to research and experiment how viable machine learning models can perform on the stock market, there are no legal or ethical issues concerned with this project. Subsequently, ethical clearance from my supervisor was sought and approved.

ID	Investigator	Project name	Status	Last updated	Option	Download
26320	Rahul Shrestha	Intelligent Income and Expenditure System	●●● Approved	03/05/22 10:08	Edit	Download as PDF

Ethical clearance for research and innovation projects

Project status

Status

●●● Approved

Actions

Date	Who	Action	Comments
10:08:00 03 May 2022	Darren Cunningham	Supervisor approved	
22:04:00 02 May 2022	Rahul Shrestha	Principal investigator submitted	

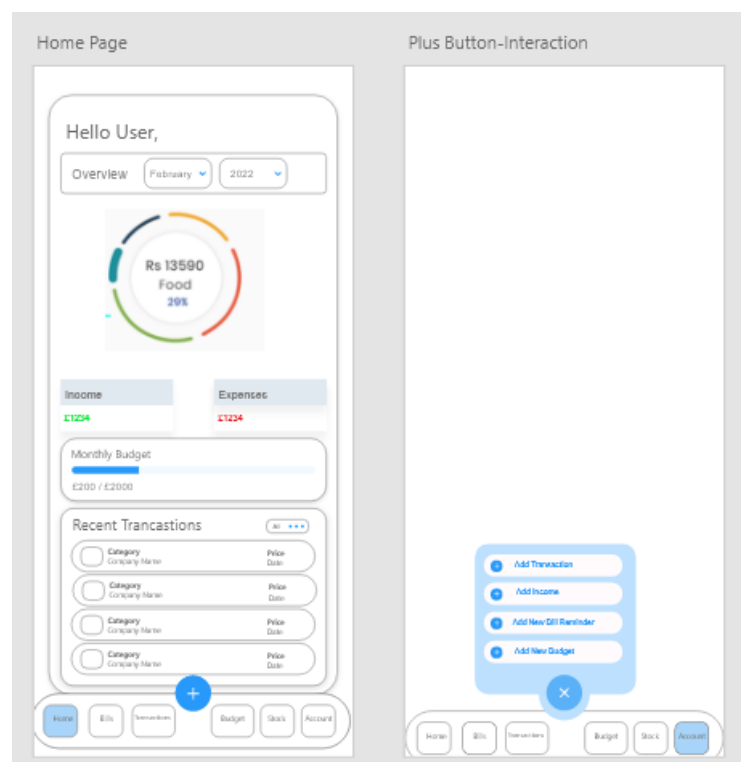
Get Help

The main project requirements and objectives include:

- Designing and creating a clean, aesthetic front end user interface (UI) for a web version and a mobile phone app version.
- A web application prototype such that users can:
 - Sign up and log in to access the app.
 - View, add and manage their budgets.
 - View, add and manage their income and expenditures.
 - View, add and manage any upcoming bills.
- A machine learning application such that users can:
 - Enter and receive stock data information.
 - View technical analysis on the stock.
 - Forecast stock prices using machine learning models.
 - View forecasted trading signal from machine learning model.

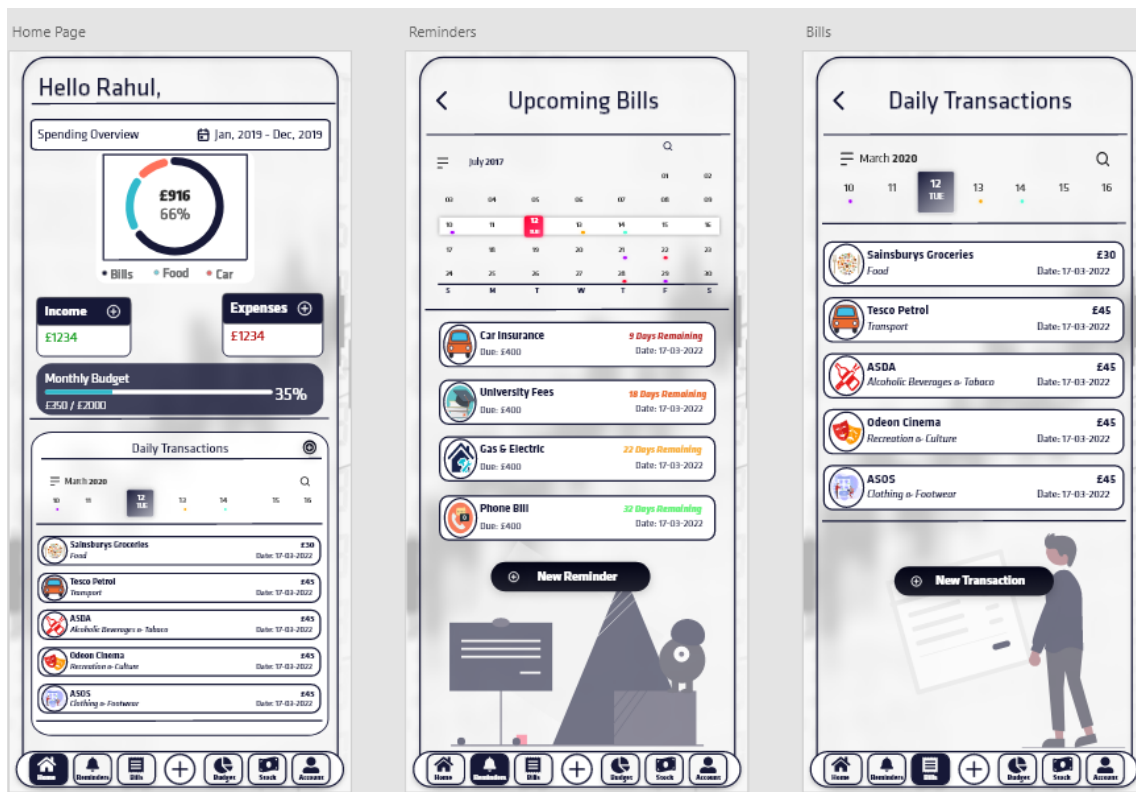
The project followed a waterfall methodology that started with a design phase which consisted of creating wireframes.

Wireframes are considered as blueprints for applications that allows programmers to visualise the structure of the application. Hannah (2021) describes that wireframe allow for a clear overview of the page structure, user flow and functionality. Additionally, a mobile-first approach was taken when creating the wireframes as Morales (2021) stated that designing for smaller screens first allows designers to focus more on the core functions of the application. Moreover, it has been stated that by 2025, it is expected that 72.5% of people will access the internet from their mobile phones only (Morales, 2021). Here are screenshots displaying one of the mobile wireframes. The complete wireframes for mobile version can be found in [Appendix A \(Mobile\)](#) and the web version on [Appendix B \(Web\)](#).

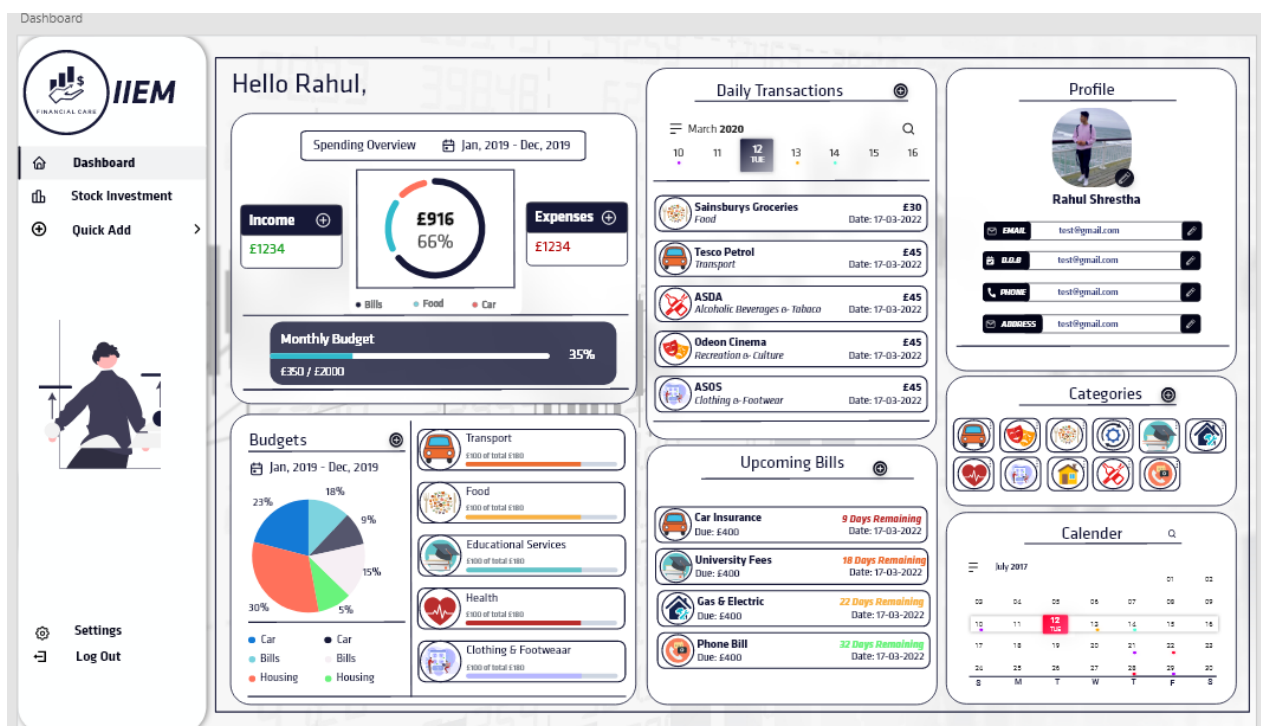


After the initial wireframe design phase, the next step included creating a mock-up of the application using Adobe XD. The mock up consisted of changes in the layout, implementation of colour scheme, images, styles and dictating the flow of the application. The purpose of mock up acted as a visual draft of the application.

Here is a screenshot of one of the mobile version mock-ups. The complete mobile mock-ups can be found at [Appendix C \(Mobile-Mockup\)](#):



As well, here is a screenshot displaying one the web version mock-ups. The remaining screens can be found at [Appendix D \(Web-Mock-Up\)](#):



Following the design phase, the prototype for the web application was implemented using React.js as the front end and Flask for the backend.

React.Js is stated to be one of the most popular frameworks when it comes to front-end development. In a survey conducted by StackOverflow (2019), consisting of 90,000 developers, React.Js came at the top for the most liked frameworks scoring at 74.5%. Plus, Rafal and Matt (2021) described react to increase the development process because of its ability to reuse components and development tools. Additionally, it was stated that react allowed for a production of rich, quality user interfaces, so much that big companies such as Facebook, PayPal and many more use React.

Python's Flask was chosen for the backend web framework for this project. Flask is classed as a microframework which allows for a rapid development of web applications. The main reason Flask was chosen over other python web frameworks like Django was because of Flask's simplicity when it came to the development process and its potential to scale as the project gets bigger (Deery, 2021). Furthermore, Flask is stated to be more user-friendly and requires less of a learning curve which will result in a cleaner codebase compared to Django (Adhikari, 2022). Additionally, as the other half of this project utilises machine learning, Python was chosen due to its flexibility as it can be used in many different fields such as web development, artificial intelligence (AI), machine learning and many more (Maderia, 2020). Moreover, Python was stated to be the lead choice for machine learning and AI, due to its extensive collection of libraries and packages that are available for use (Gupta, 2021).

5. Methodology

Advancing on to the stock market-machine learning aspect. The stock market can be explored using two methods known as technical analysis and fundamental analysis. Fundamental analysis is defined as a method to determine the real (intrinsic) value of a stock by examining economic and financial factors of the company (Segal, 2021). Investors and traders that use fundamental analysis believe that the market does not accurately estimate the value of stocks and therefore they try and find a true worth of a company (The Street, 2022). They find and invest in stocks they believe are undervalued by the market and hope the stock's value increases over time.

On the other hand, technical analysis is defined as using historical market data to evaluate the price trends and patterns, to predict future markets behaviour (Chen, 2021). Saravanan (2019) has stated that fundamental analysis is more theoretical and that using technical analysis is seen to be more practical as it uses more factual, concrete data. Additionally, The Street (2022) has claimed that trading decisions are best made from technical analysis using trend evaluation and pattern recognition as they believe that stocks are accurately valued, thus fundamental analysis is unnecessary.

Technical Indicators fall into the realm of technical analysis, and Chen (2021) defined it as mathematical calculations and patterns derived from historical data. There are many technical indicators available out there and they can be classed into five categories: trend, momentum, relative strength, mean reversion, and volume (Barone, 2022). Folger (2022) has advised that when developing a trading strategy, it is recommended not to use different indicators from the same category as this can result in multicollinearity but as this project is aimed towards beginners, I have chosen easy to understand and beginner-friendly indicators which goes against Folgers' advice.

The dataset used to train and test the models will be historical stock data, in this case Apple's (AAPL) stock data was used, sourced from Yahoo Finance using the python library 'yfinance'. Additionally, extra columns will be added to the dataset which will consist of four technical indicators data and for the classification models, an extra column will be added which will act as a target variable; this column will be an overall recommendation for a trading signal derived from the four technical indicators. This project takes on a supervised learning approach; Petersson (2021) defines supervised learning as models that are trained on input data labelled to specific output. This allows the model to learn and detect underlying patterns and relationships between the input and output data so that it can accurately predict on unseen input data. The aim of this project is to utilise technical analysis to predict future stock prices and trading signals of a stock such as buy, hold, or sell based on its historical data and the technical indicators data.

The technical indicators that are used and added to the dataset are:

COLUMN	EXPLANATION
Stochastic Oscillator (SO)	<p>Stochastic Oscillator (SO), which was developed by George Lane in the 1950's, is a popular technical indicator when it comes to generating oversold and overbought signals (Hayes, 2021). Anderson (2022) defines SO to describe the relationship between the stock price, relative to its high and low prices over a predetermined period (14 days being the most popular period). Additionally, Anderson (2022) has stated that SO has a history of being accurate when it comes to generating buy and sell signals.</p> <p>SO has two components that work together in building a trading signal, the fast line denoted as '%K' and the slow line denoted as '%D' (West, n.d). Both signals produce a value that ranges between 0 to 100. Typically, values below 20 are seen as oversold which</p>

infers a buy signal and values over 80 are seen as overbought which infers a sell signal (West, n.d).

$K\%$ is calculated by $= 100 * ((14 \text{ Day Closing Price} - 14 \text{ Day Lowest Price}) / (14 \text{ Highest Price} - 14 \text{ Day Lowest Price}))$

$D\%$ is calculated by $=$ moving average of $\%K$ over 3 days.

(For Clasifcation Model Only)

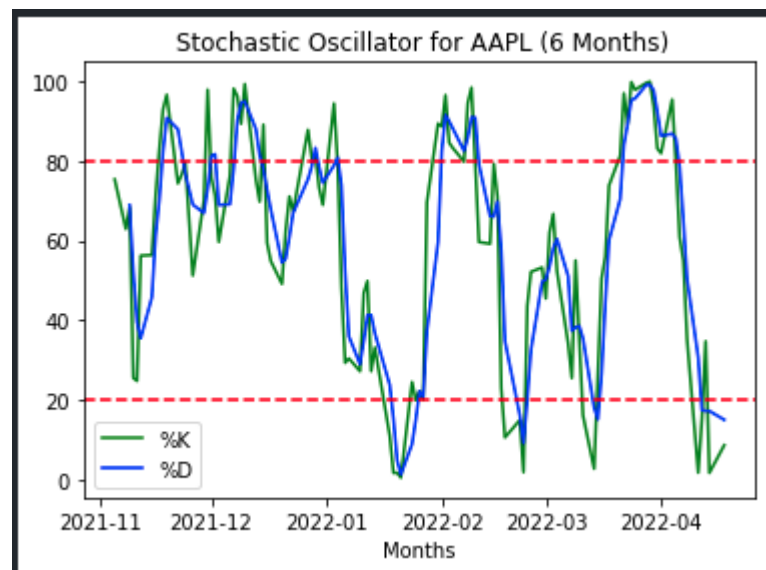
For this project, the SO indicator will follow the traditional rules when producing a trading signal such that :

A 'buy' signal will be created when:

- The $\%K$ value/line is below 20
- The $\%D$ value/line is below 20

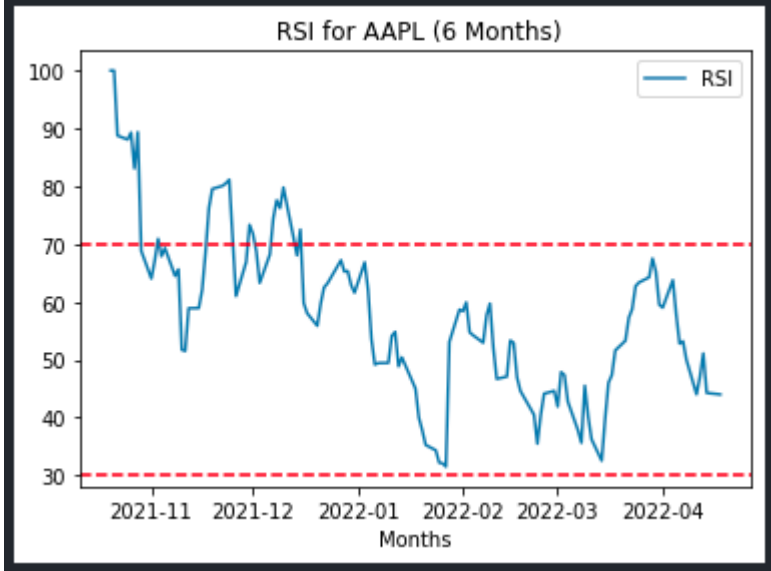
A sell signal will be created when:

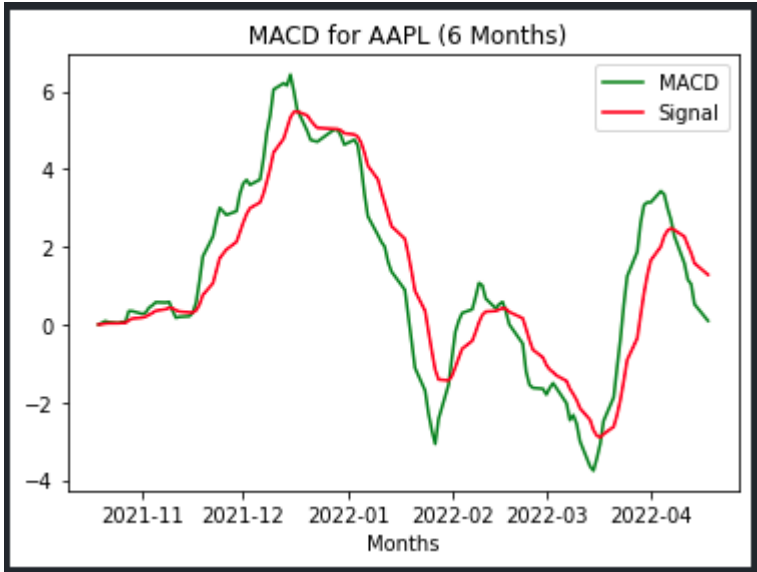
- The $\%K$ value/line is above 80
- The $\%D$ value/line is above 80



Here is a graph displaying the SO indicator based on the past 6 months of the Apple (AAPL) stock:

<p>Relative Strength Index (RSI)</p>	<p>The Relative Strength Index (RSI), which was developed by J. Welles Wilder in 1970, is also a momentum indicator like the stochastic oscillator that is used by traders to identify whether the market is an overbought or oversold state. Gumparthi (2017) describes RSI to measure the speed and change of price movements over a previous trading period.</p> <p>The RSI also produces a value ranging from 0 to 100 but unlike the SO, values over 70 are seen as overbought and values under 30 are seen as oversold, according to Fernando (2022).</p> <p>Even though the RSI and SO are both momentum indicators, they both have different underlying methods and theories. Ross (2021) has stated the RSI is more useful in trending markets whereas SO is more useful when the market is trading in consistent ranges.</p> <p>A study conducted by Gumparthi (2017) to test the validity of RSI signals in trading strategies found the RSI to be an effective indicator that was able to produce accurate buy and sell signals for both short-term and long-term investments. It was also discovered that RSI successfully predicted future trends in the market.</p> <p>Fernando (2022) described the RSI to be calculated using the following formulas:</p> <ol style="list-style-type: none"> 1. Avg Loss = Sum of Losses over the past 14 periods / 14 2. Avg Gain = Sum of Gains over the past 14 periods / 14
--------------------------------------	--

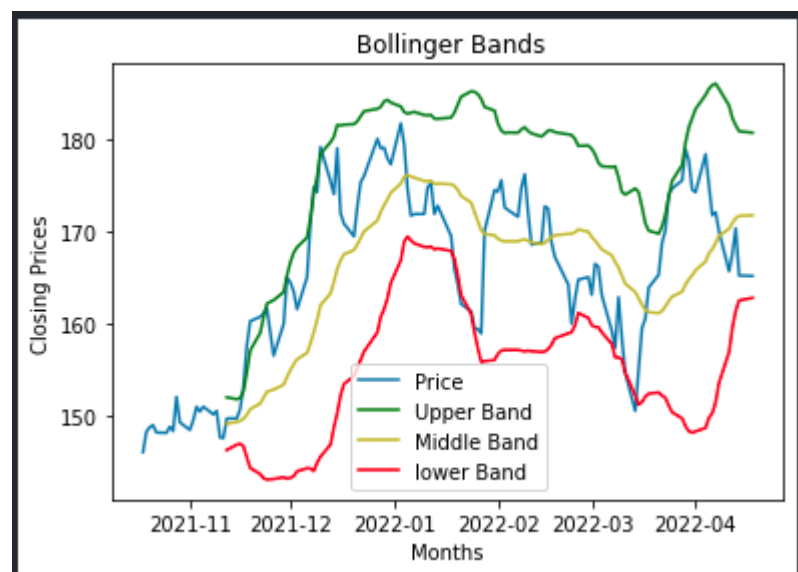
	<p>3. $RS = \text{Average Gain} / \text{Average Loss}$</p> <p>4. $RSI = 100 - 100 / (1 + RS)$.</p> <p>(For Clasifcation Model Only)</p> <p>For this project, the traditional boundaries will be used to create a trading signal for the RSI; such that values under 30 will be seen as buy signals and values over 70 will be seen as sell signals.</p> <p>Here is a graph displaying the RSI indicator over the past 6 months of the Apple (AAPL) stock:</p> 
Moving Average Convergence Divergence (MACD)	<p>The Moving Average Convergence Divergence (MACD) was developed by Gerald Appel in 1979 and is used as a trend-following momentum indicator (Schlossberg, 2022). Silberstein (2022) defined MACD to describe the relationship between two moving averages of a stock and it is calculated by subtracting the 26-period exponential moving average (EMA) from the 12-period EMA; this is referred to as the MACD line. Additionally, there is another component referred to as the signal line, that works with the MACD line to come up with a</p>

	<p>trading signal. The signal line is calculated by finding out the 9-period EMA of the MACD. Mathematically written as:</p> <ul style="list-style-type: none"> • $MACD = 12D\ EMA - 26DEMA$ • $Signal = 9D\ EMA\ of\ MACD$ <p>Here is a diagram displaying the MACD line and the Signal line for the past 6 months of the Apple (AAPL) stock:</p>  <p>(For Clasifcation Model Only)</p> <p>For this project, MACD indicator will produce a buy signal when the MACD line crosses above the signal line thus the sell signal will be created when the MACD line crosses below the signal line.</p>
Bollinger Bands (BB)	<p>Bollinger Bands (BB) was created by John Bollinger in the 1980's and it has been described to offer numerous insights into price and volatility, such as monitoring breakouts, following trends and determining overbought and oversold levels (Mitchell, 2022).</p>

BB consist of three components that work together to highlight how prices are distributed around an average value. Binance Academy (2018) described the components to be calculated using the following formulas:

- Middle Band= 20-day simple moving average (SMA)
- Upper Band = Middle Band + (2 x 20-day stand deviation)
- Lower Band = Middle Band - (2 x 20-day stand deviation)

Here is a diagram displaying the BB for the past 6 months of the Apple (AAPL) stock:



(For Clasifcation Model Only)

For this project, the BB indicator will be used to determine overbought and oversold level to create buy and sell signals. Buy signals will be created when the price crosses below the lower band and alternatively, sell signals when the price cross above the upper band.

Recommender
(Target Variable for
Classification Model)

(For Clasifcation Model Only)

The Recommender column (dependant variable) contains an overall recommendation in whether to buy, sell, or hold the stock based on the signals from the other indicators.

Upon further inspection, the function I created to derive trading signals the MACD indicators were producing inaccurate signals. Therefore, they have not been taken in consideration when creating the overall signals. However, the MACD line and the signal line will still be used when training the models.

To ensure signals were as accurate as possible, I followed the following steps:

A simple if-else function, where if all three of the indicators stated the same signal, the value would be declared as that signal or if at least two out of three indicators stated the same signal, it was declared as that signal. Everything else that did not fit into the above statements were labelled as 'Unclassed' and they were manually given a signal. This table outlines

the above
function:

RSI	SO	BB	Recommender
Buy	Buy	Buy	Buy
Sell	Sell	Sell	Sell
Hold	Hold	Hold	Hold
Buy	Buy	?	Buy
?	Buy	Buy	Buy
Buy	?	Buy	Buy
Sell	Sell	?	Sell
?	Sell	Sell	Sell
Sell	?	Sell	Sell
Hold	Hold	?	Hold
?	Hold	Hold	Hold
Hold	?	Hold	Hold

5.1 Regression

Predicting future stock prices can also be classed as a time series problem in which time series forecasting methods can be applied. Tableau (2022) defined time series forecasting as making scientific predictions based on historical timed stamped data. In relation to stock prices, Christie (2020) stated that stock prices should be treated as discrete time series data as stock prices are taken sequentially in time. As mentioned above, AAPL's historical stock data will be used as the dataset to train and test the regression models. Unlike with the classification models, this dataset will contain all AAPL's stock data available, dating back to when the company first went public on December 12, 1980.

5.1.1 Regression: Exploratory Data Analysis & Data Cleaning

Conducting an exploratory data analysis (EDA) is an important step in which preliminary investigation are conducted to provide insight into the data and their interactions (Sonal, 2021). EDA normally consists of using graphical representations and summary of statistics.

Table containing the EDA that was conducted:

1. Gathered historical data for the AAPL stock using 'yfinance' and stored in a panda's data frame. Applied the function '.shape' which returns the dimensions of the dataset. From the result, you can interpret the base dataset to contains 10427 rows and 7 columns.

```
#Getting data for the Apple Stock
aapl = yf.Ticker("AAPL")
# Get historical market data
aapl.dataset = aapl.history(period="max")
```

✓ 1.4s Python

```
#Dataframe basic information on rows and
aapl.dataset.shape
```

✓ 0.6s Python

(10427, 7)

2. Explored the dataset by applying the function `‘.columns’` which returned the columns headers in the dataset. From the result, you can see it to contain:

```
#Dataframe columns
aapl.dataset.columns
✓ 0.1s Python
Index(['Open', 'High', 'Low', 'Close',
       'Volume', 'Dividends', 'Stock Splits'],
      dtype='object')
```

- Open
 - Open represents the stock’s initial price at the start of the trading day.
- Close
 - Close represents the stock’s final price at the end of the trading day.
- High
 - High represents the stock’s highest trading price for the day.
- Low
 - Low represents the stock’s lowest trading price of the day.
- Volume
 - Volume represents the number of shares that was traded in the day.
- Dividends
 - Dividends represent the number of shares that was paid to the shareholders instead of cash.
- Stock Splits
 - Stock Splits represents the ratio in which the stocks are split. This occurs when a company wants to boot its stock liquidity by increasing the number of its outstanding shares (Hayes, 2022).

3. Explored the values of the columns “Dividends” and “Stock Splits”. From the results, you can see the majority of the value contained in these columns were ‘0’. As these two columns are not of any use, I decided to drop them before moving on.

```
# Distribution of the Dividends
print(aapl.dataset['Dividends'].value_counts())
✓ 0.7s
```

0.000000	10354
0.001071	21
0.000714	4
0.000893	4
0.000982	4
0.108929	4
0.117500	4
0.130000	4
0.142500	4
0.157500	4
0.182500	4
0.192500	4
0.205000	4
0.220000	4
0.094643	3
0.000536	2

```
Name: Dividends, dtype: int64
```

```
# Distribution of Stock Splits
print(aapl.dataset['Stock Splits'].value_counts())
✓ 0.7s
```

0.0	10423
2.0	3
7.0	1
4.0	1

```
Name: Stock Splits, dtype: int64
```

4. When I came to dropping these two columns, I encountered a bug where that even after dropping the columns successfully, calling the data frame again would result in the columns being reinstated. To resolve this issue, I utilised the ‘.copy’ function to copy the relevant columns on to a new data frame.

```
# Dropping Dividends and Stock Splits Columns
aapl.dataset.drop(['Dividends', 'Stock Splits'], axis = 1)
✓ 0.9s Python
```

```
#YFinance Bug
aapl.dataset.columns
✓ 0.7s Python
```

```
Index(['Open', 'High', 'Low', 'Close', 'Volume',
      'Dividends', 'Stock Splits'], dtype='object')
```

```
#YFinance Bug- Dropped columns appears after being dropped
dataset = aapl.dataset[['Open', 'Close', 'High', 'Low', 'Volume']]
dataset.tail()
✓ 0.7s Python
```

5. Calculated and inserted the technical indicators data on to the dataset.

```
#Columns After Adding All The Technical Indicators
dataset.columns
✓ 0.1s
```

```
Index(['Open', 'Close', 'High', 'Low', 'Volume', '%K', '%D', 'RSI',
      'Bollinger_Upper', 'Bollinger_Lower', 'MACD', 'Signal'],
      dtype='object')
```


6. Explored the dataset using the '.info' function which provides a brief overview of the dataset. From the result, you can see the date range, the column headers, the number of values per each column, and its data types. Being able to view the data types of the columns is especially valuable as if there any non-numerical values, you would need to encode these values before moving onto modelling. In this case, all the values are numerical.

```
#Information regarding the dataset
dataset.info()

✓ 0.1s Python

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 10428 entries, 1980-12-12 to 2022-04-21
Data columns (total 12 columns):
#   Column              Non-Null Count  Dtype  
---  -
0   Open                 10428 non-null  float64
1   Close                10428 non-null  float64
2   High                 10428 non-null  float64
3   Low                  10428 non-null  float64
4   Volume               10428 non-null  int64   
5   %K                   10415 non-null  float64
6   %D                   10413 non-null  float64
7   RSI                  10427 non-null  float64
8   Bollinger_Upper      10409 non-null  float64
9   Bollinger_Lower      10409 non-null  float64
10  MACD                 10428 non-null  float64
11  Signal               10428 non-null  float64
dtypes: float64(11), int64(1)
memory usage: 1.0 MB
```

7. Explored the dataset columns using the function '.describe' which provides a statistical summary of the numerical columns in the dataset.

```
#Summary of Statistics
aapl.dataset.describe()

✓ 0.1s Python


```

	Open	High	Low
count	10427.000000	10427.000000	10427.000000
mean	13.616999	13.765218	13.470239
std	30.579462	30.928255	30.240507
min	0.038822	0.038822	0.038385
25%	0.235284	0.240213	0.229512
50%	0.387617	0.395262	0.380475
75%	12.288757	12.457872	12.176728
max	182.397624	182.707227	178.892080

8. Explored the dataset rows by applying the '.tail' function which returns the last 5 rows in the data frame. From the result, you can see we have the latest stock data available.

```
#Latest Stock Data in the Dataset
aapl.dataset.tail()

✓ 0.1s Python


```

	Open	High	Low	Close
Date				
2022-04-13	167.389999	171.039993	166.770004	170.3999
2022-04-14	170.619995	171.270004	165.039993	165.2899
2022-04-18	163.919998	166.600006	163.570007	165.0700
2022-04-19	165.020004	167.820007	163.910004	167.3999
2022-04-20	168.759995	168.860001	166.514999	166.8999

9. Explored the dataset rows by applying the `‘.head’` function which returns the first 5 rows in the data frame. From the result, you can see, we have stock data from as far back as `‘1980-12-12’` in the dataset.

```
#Start Stock Data in the Dataset
apl.dataset.head()
```

✓ 0.1s Python

	Open	High	Low	Close	Volume
Date					
1980-12-12	0.100326	0.100762	0.100326	0.100326	469
1980-12-15	0.095528	0.095528	0.095092	0.095092	175
1980-12-16	0.088548	0.088548	0.088112	0.088112	105
1980-12-17	0.090293	0.090729	0.090293	0.090293	86
1980-12-18	0.092911	0.093347	0.092911	0.092911	73

10. Performed checks on the dataset by applying the `‘.isna’` function which returns if there are any null values present in the dataset. It is important to handle any missing values as most machine learning models do not support missing values. Ergo, missing values can result in building a biased model which can lead to inaccurate results (Tamboli, 2021). From the result, you can see there are some null values:

```
#Checking for Null Values
dataset.isna().sum()
```

✓ 0.9s

Open	0
Close	0
High	0
Low	0
Volume	0
%K	13
%D	15
RSI	1
Bollinger_Upper	19
Bollinger_Lower	19
MACD	0
Signal	0

dtype: int64

11. Dropped null values using the function `‘.dropna’`. There are multiple ways in which you can handle missing data but, in this case, I decided to drop the rows entirely as there were only a small percentage of null values and dropping these rows would not impact anything later.

```
#Dropping Null Values
dataset = dataset.dropna()
dataset.isna().sum()

✓ 0.1s

Open      0
Close     0
High      0
Low       0
Volume    0
%K        0
%D        0
RSI       0
Bollinger_Upper  0
Bollinger_Lower  0
MACD      0
Signal    0
dtype: int64
```

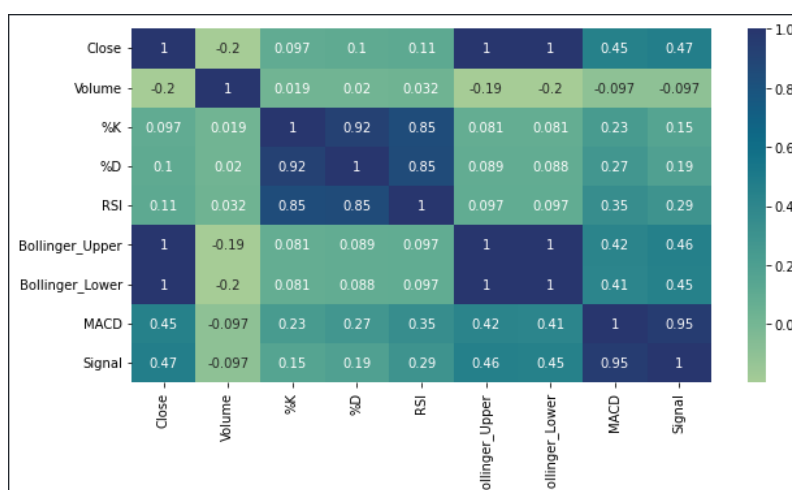
12. Performed more checks on the dataset using the `‘.duplicated’` function which returns if there are any duplicated values present in the dataset. From the result, you can see that for this instance, there were no duplicates.

```
#let's check if there is any duplicate data
print(aapl.dataset.duplicated().any())
print(aapl.dataset.duplicated().sum())

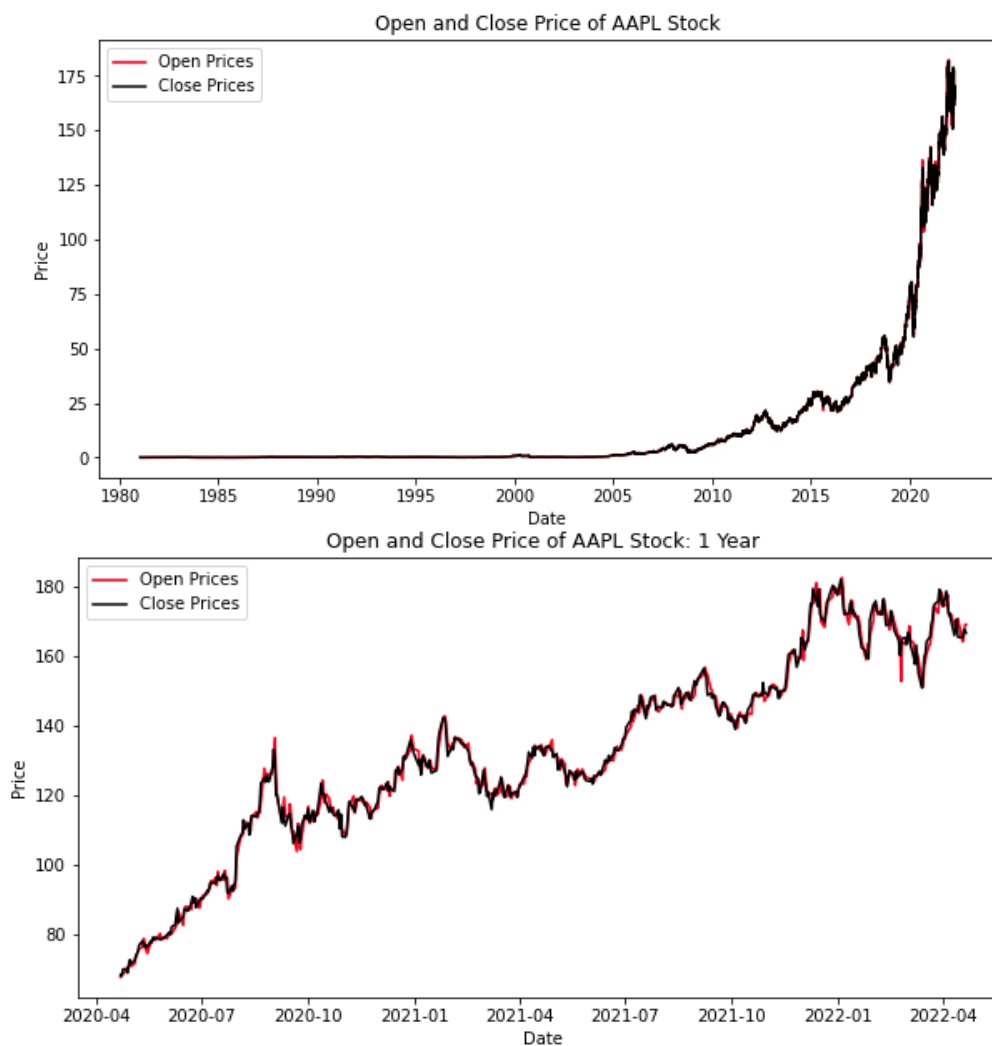
✓ 0.4s Python

False
0
```

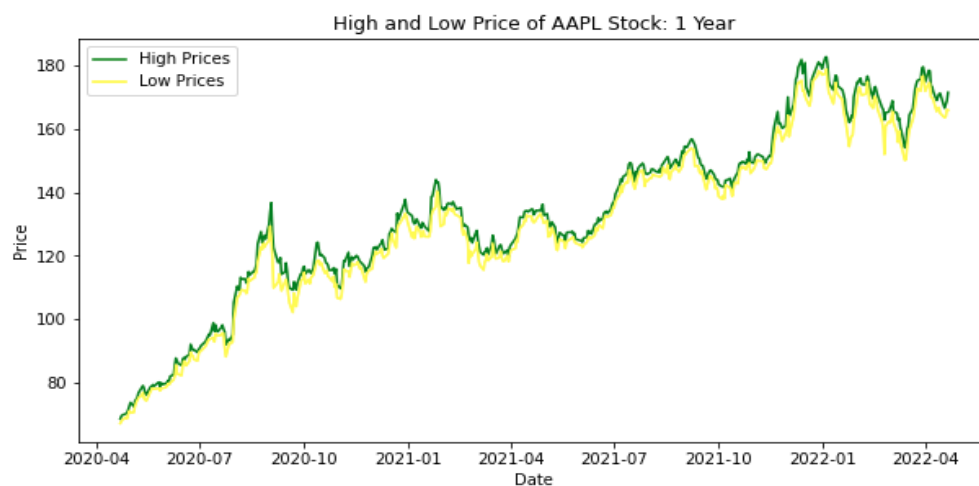
13. Plotting Heatmap: Heatmaps display the correlation between the different variables on scale from -1 to 1. From the results, you can interpret how the different technical indicators, closing price and volume are correlated.



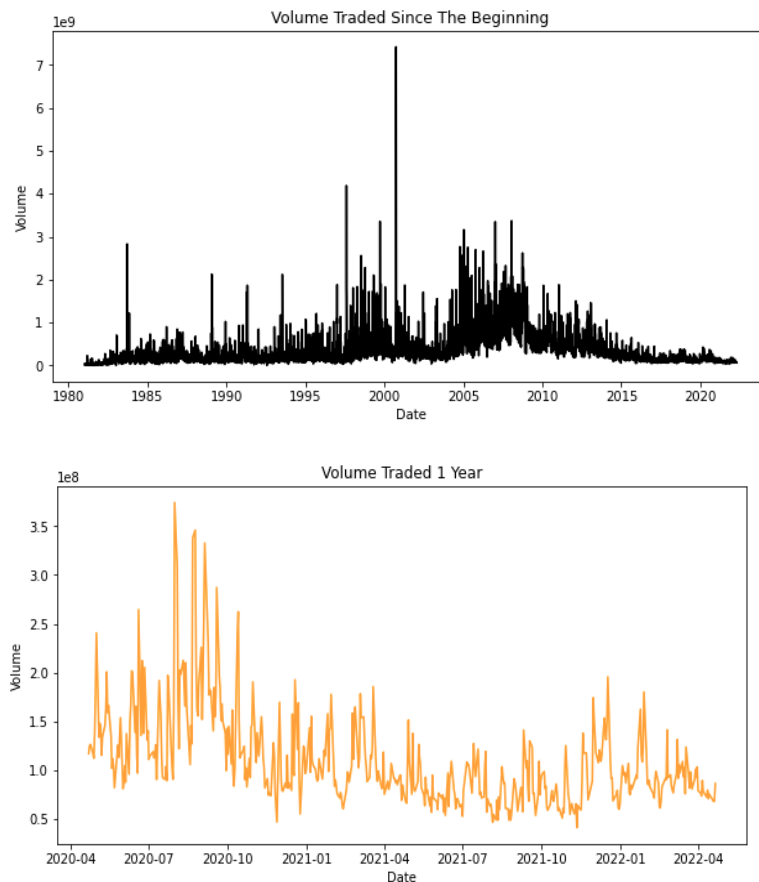
14: Plotting Line Graph: This first graph represents the stocks opening and closing price throughout the years. To get a better insight, the second chart looks back over the past 1 year.



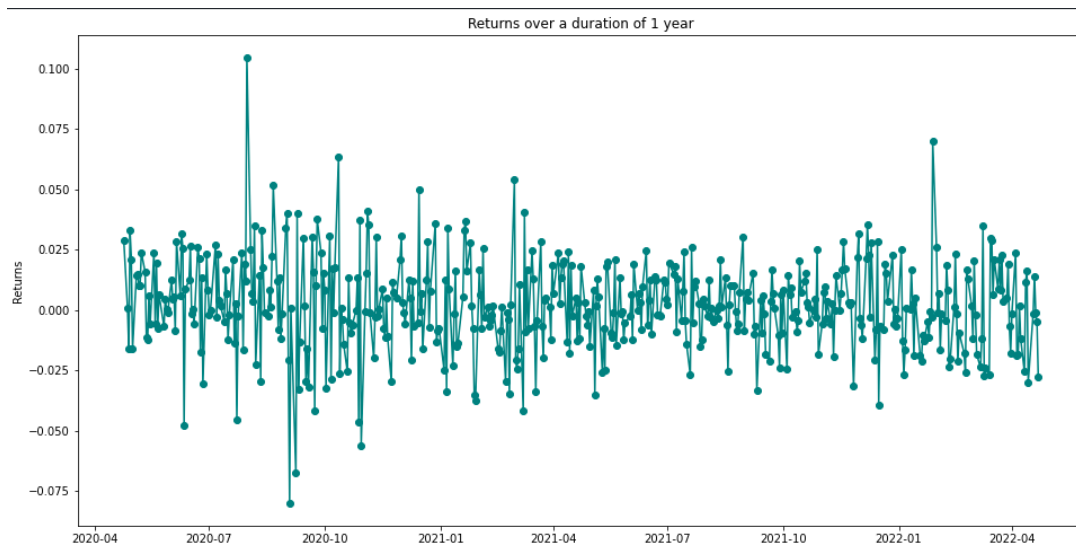
15. Plotting Line Graph: This graph represents the stocks high and low prices over the past 1 year.



16. Plotting Bar Graph: These graphs display the number of shares being traded throughout the years. The second graph displays the result from the past 1 year.



17. Plotting Scatter Graph: This graph displays the returns of the stock over the 1 past year.

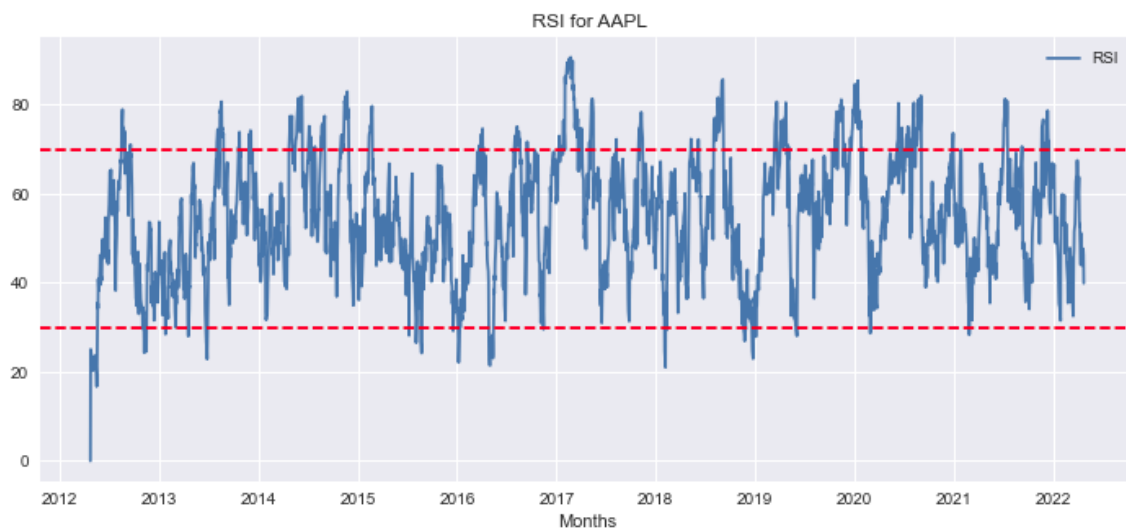


18. Plotting Line Graph: This graph displays the closing price against the moving averages of 50, 100 and 200 days. Moving Averages (MA) are popular tool when to comes to technical analysis, Potters (2022) defines MA to smooth out the price

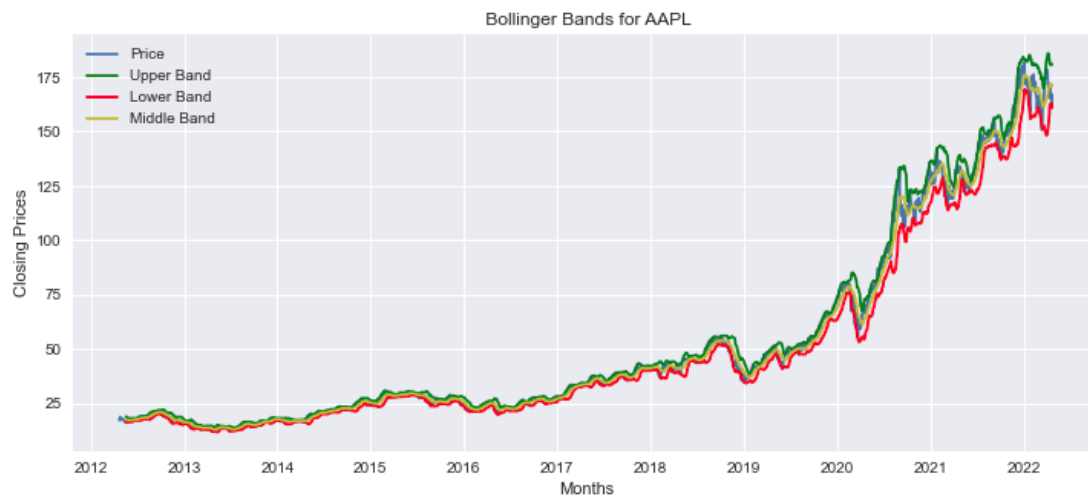


trend from short-flucations by filtering out the noise.

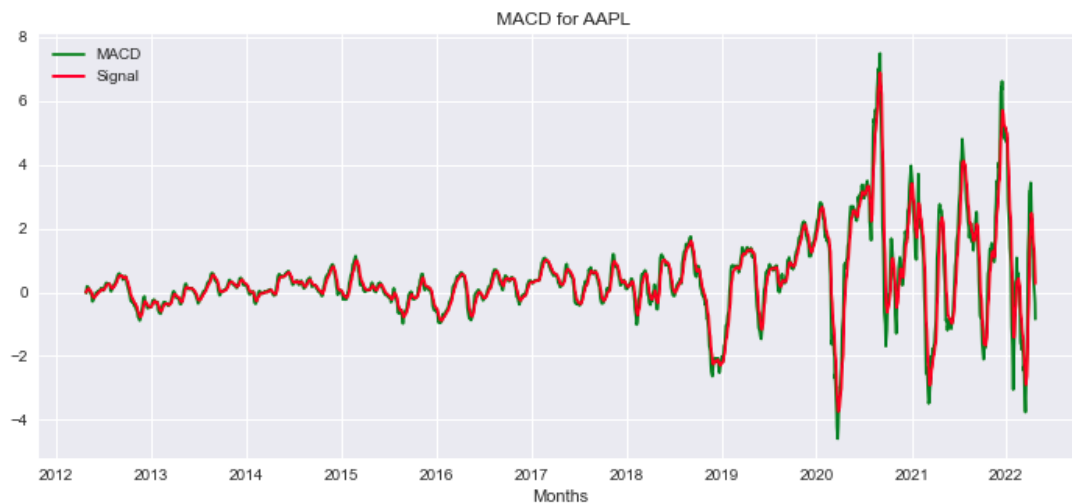
19. Plotting Technical Indicator: RSI (10 Years Duration)



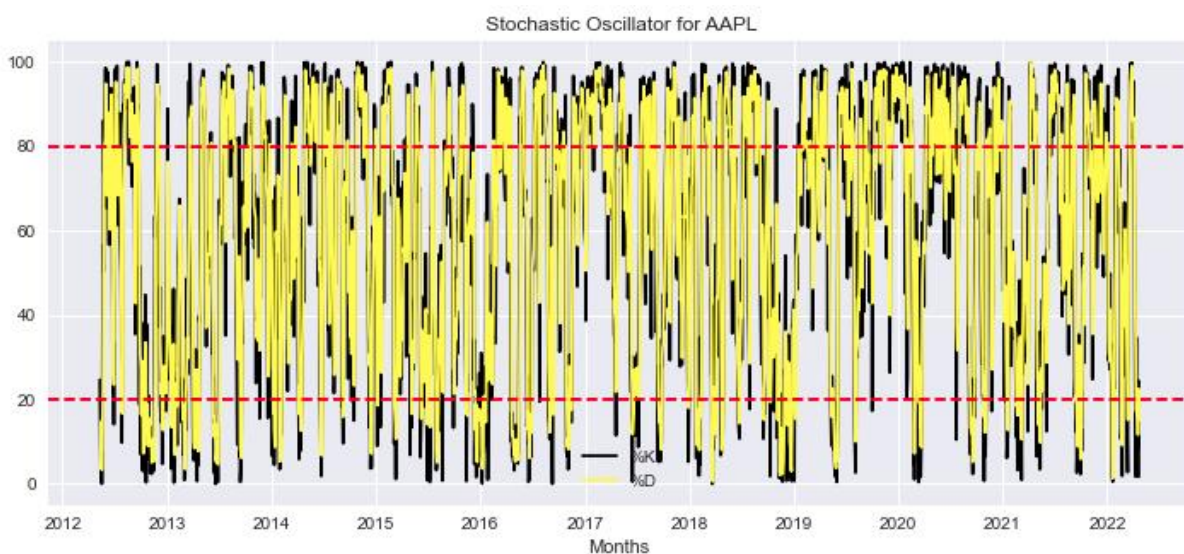
20. Plotting Technical Indicator: BB (10 Years Duration)



21. Plotting Technical Indicator: MACD (10 Years Duration)



22. Plotting Indicator: SO (10 Years Duration)



5.1.2 Regression: Data Pre-Processing

Preparing and pre-processing the dataset prior to modelling is a critical step that must be taken to ensure that the machine learning model is accurate and efficient. Baheti (2022) defined data pre-processing as a series of step that must be followed to transform and encode the data so that it can be easily parsed by the machine learning model. As I have already cleaned the data by removing null values and duplicates, the next steps include feature scaling and splitting the dataset into train and test sets.

Feature Scaling can be defined as a method to transform the numeric features in the dataset, to a standard range (Munagala, 2021). Roy (2020) has stated that implementing feature scaling is a crucial step that can determine the difference between a weak and a strong model. Additionally, Brownlee (2020b) and Sharma (2021) have reported that many machine learning algorithms perform better when the numerical features are scaled to a standard range, such that scaling increases precision and reduces memory consumption. The main reasoning behind this is because most machine learning algorithms cannot comprehend the true meaning behind numbers such that they interpret features with higher range values as more important and tend to ignore features with smaller range values which can lead to inaccurate predictions (Munagala, 2021). Additionally, Roy (2020) explained that machine learning algorithms cannot differentiate between 10g of weight and £10 in price, this leads to hierarchy and bias between the variables. However, some models can perform well without feature scaling as its accuracy is not dependent on the range, for example: tree-based models.

Normalisation and Standardisation are the two most popular techniques that are used to scale numerical data. Brownlee (2020b) describes normalisation as rescaling the data so that all the values fit into a range of 0 and 1, where 1 represents the highest feature value and the '0' the lowest. Secondly, Liu (2020) describes standardisation as transforming the data so that the features are

rescaled to have a standard deviation of 1 and a mean of 0. To determine which technique to use, Brownlee (2020b) expressed that there is no correct answer such that it depends on many different factors like the specifics of the problem, the choice of models, and the state of the variables.

For this project, ‘MinMaxScaler’ (MMX) from scikit-learn was used to scale the dataset between the default range of 0 to 1. MMX has been described to preserve the original shape of the distribution without changing the meaning of the values from the original data, such that the importance of outliers is not reduced (Hale, 2019).

5.1.3 Regression Model: Modelling and Results

Furthermore, the following metrics will be used to evaluate the models, however the RMSE score will be considered as the main metric. Models scoring less than 5 RMSE will be classed as successful, thus be tuned, and implemented on the app.

Mean Absolute Error (MAE)	The MAE refers to the average in the absolute difference between the actual value and the predicted value (Bajaj, 2022). Brownlee (2021b) has stated that the MAE increases the scores linearly with the increases in errors and that it does not give any sort of weight to the different types of errors.
---------------------------	---

Formula used to calculate the MAE:

$$MAE = \frac{1}{N} \sum_{j=1}^N |y_j - \hat{y}_j|$$

Image by Ghosh (2022)

Mean Squared Error (MSE)	The MSE refers to the average of the squared differences between the predicted and the actual values (Brownlee, 2021b). In other words, the MSE provides an absolute number
--------------------------	---

on how much the predicted values deviate from the actual values. Additionally, Ghosh (2022) has stated that the MSE is highly sensitive to outliers and small errors which can give a high error score and can lead to a misinterpretation on how poorly the model performed.

Formula used to calculate the MSE:

$$MAE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2$$

Image by Ghosh (2022).

Root Mean
Squared Error
(RMSE)

The RMSE is the squared root of the MSE and the purpose of it is to have the error score in the same scale as the original units. Additionally, it has been stated to handle the penalisation of small errors from the MSE (Ghosh, 2022).

Formula used to calculate the RMSE:

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (y_j - \tilde{y}_j)^2}$$

Image by Ghosh (2022).

R-Squared (R^2)

R^2 , also known as the Coefficient of determination, refers to the variance between the original values (independent variable) in the dataset and the predictions (dependant variable) made by the model (Kharwal, 2021). In other words, Wu (2020) described R^2 to measure how much variability in the dependant variable can be explained by the model.

Formula used to calculate the R^2

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Image by Wu (2020)

For this project, I have explored the two most popular time series models used to predict stock prices.

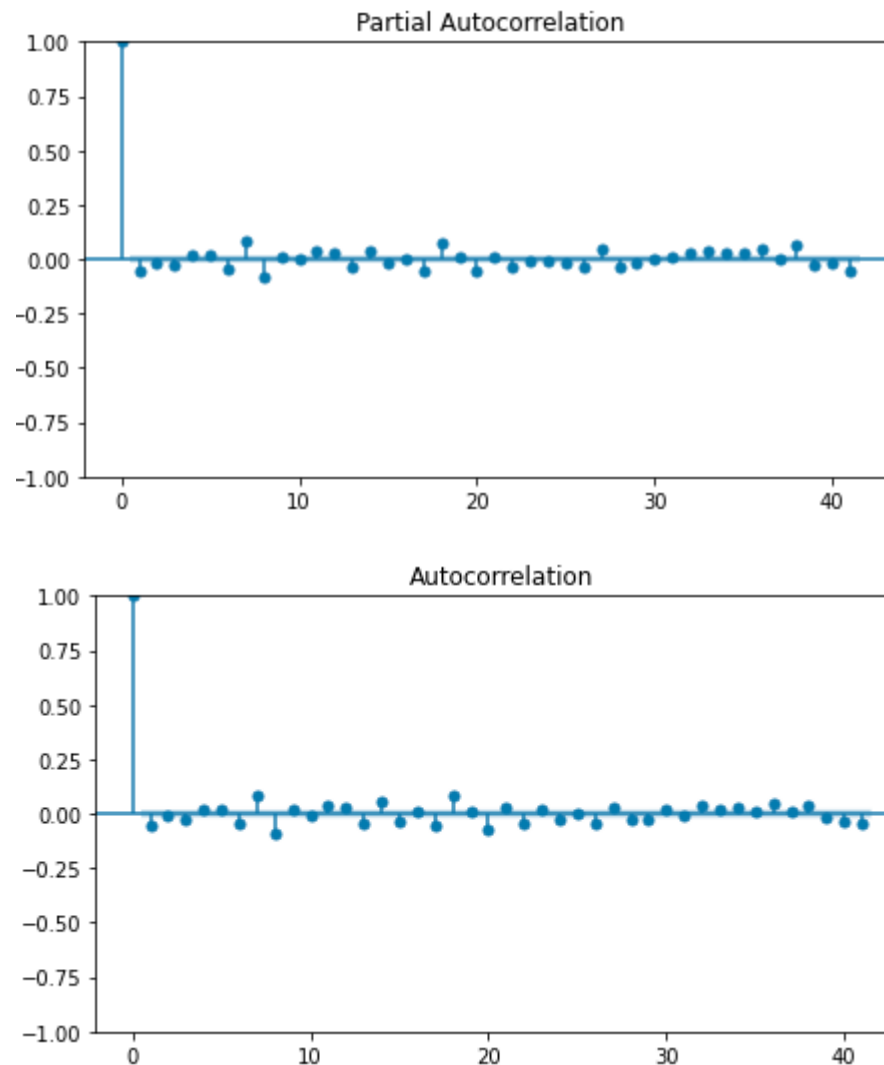
Auto-Regressive Integrated Moving Average (ARIMA)

The ARIMA model has been defined as a statistical analysis model that is used to forecast time series problem such that it examines the difference between the values against the time series (Hayes, 2021). A glaring drawback of this model is that it assumes that future prices and trends will resemble the past, causing the model to consequently make false predictions.

The ARIMA model can be broken down into three components, which are also important parameters to the model. Hayes (2021) described them as:

- AR (Auto Regression), denoted as p , representing the amount of lag observations in the model, also known as the lag order.
- I (Integrated), denoted as d , representing the number of differences with the raw observations.
- MA (Moving Average), denoted as q , represents the size of the moving average window.

To find these parameters, Partial Autocorrelation (PACF) and the Auto Correlation (ACF) were inspected.



Additionally, “Auto-ARIMA” model was implemented to find best p,d,q parameters. The results were:

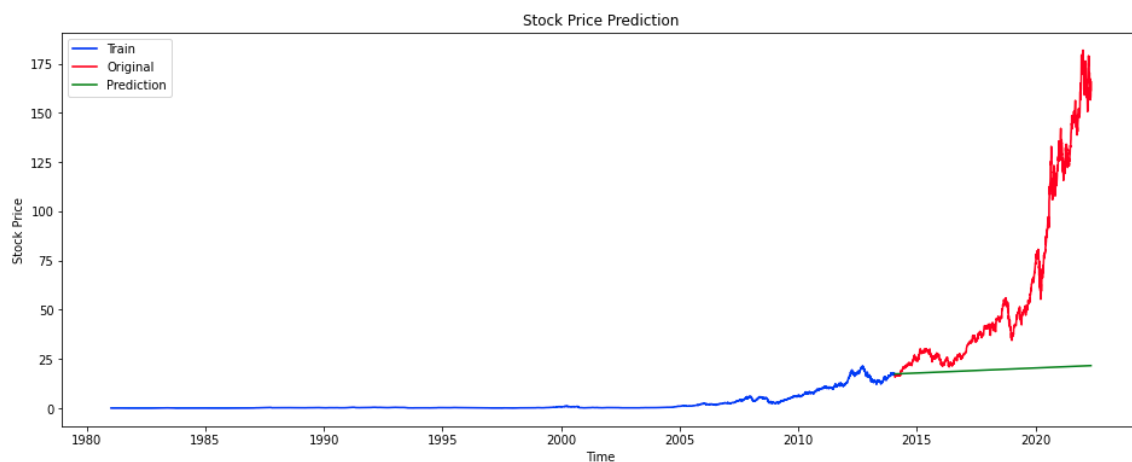
```
ARIMA(order=(2, 1, 3), scoring_args={},  
suppress_warnings=True)
```

Models performance

```
MAE: 40.40606103198367  
MSE: 3597.3934877090496  
RMSE: 59.978275131159364  
R2: -0.7427742965419393
```

As you can see from the results, the model very poorly scored 59.9 on the RMSE.

From the graph you can see that the ARIMA model failed to predict the growth and future trend of the Apples' stock, proving the limitation of this model.



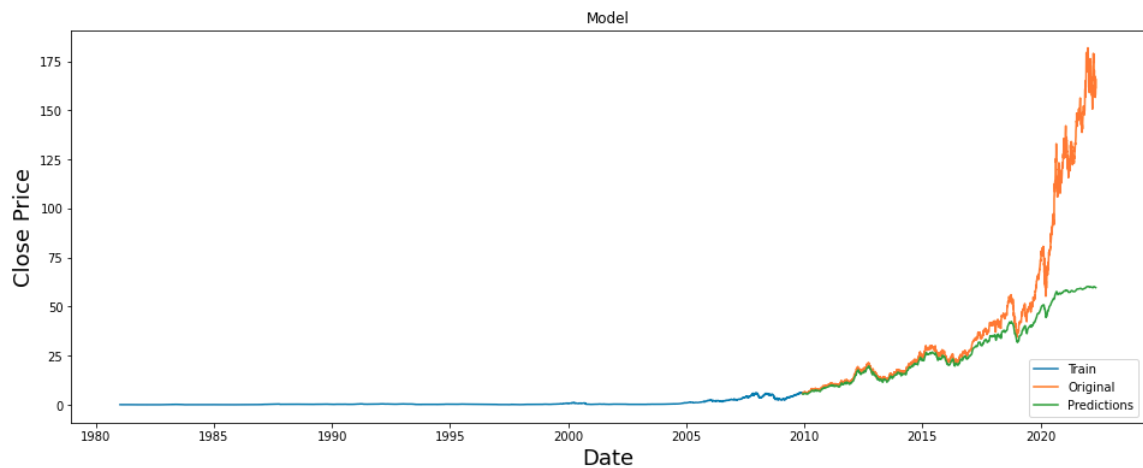
Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) model is described by Brownlee (2021a) as an advanced recurrent neural network (RNN) that is capable of learning and remembering selective patterns over a long period of time. RNN are an extension of artificial neural network (ANN) which consist of a set of algorithms that try to imitate how a human brain would function. Additionally, Saxena (2021) has stated the LSTM has been explicitly designed to avoid the shortcomings of RNN, such that RNN were not able to remember long term dependencies due to the vanishing gradient.

The model was composed of a sequential input layer, followed by four extra layers consisting of 50, 60, 80 and 120 neurons. Here are the results from the model:

```
MAE: 15.70348991256714
MSE: 1062.9650678386315
RMSE: 32.60314506054027
R2: -2.820984317807399
```

The LSTM model also did not perform that well. Although, the LSTM model performed significantly better than the ARIMA, such that it was predicting well until 2020.



To conclude, neither of these models were able to accurately predict the growth in the Apples' stock and neither scored a RMSE less than 5. Therefore, the regressor side of this project was deemed to be unsuccessful and no further research was conducted.

5. 2 Classification

Classification is defined as a process of predicting which class label or category, a given observation belongs to (Nabi, 2018). The aim of the classifier is seen as a multi-class classification problem where we are trying to predict stock signals into 3 classes: buy, hold, and sell. Like the regression model, AAPL's historical stock data will be used to train and test the models but in this case the dataset will only contain 10 years' worth of data. As described above, this dataset will contain an extra target variable column.

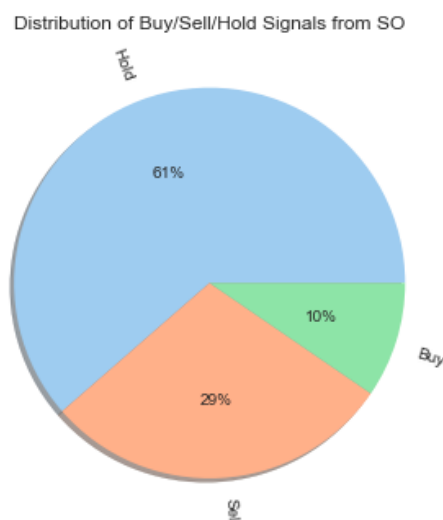
5.2.1 Classification: Exploratory Data Analysis & Data Cleaning

The EDA for this dataset will mostly follow the same process as the regression dataset. Therefore, only the differences are listed below:

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2516 entries, 2012-04-27 to 2022-04-26
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Open                   2516 non-null   float64
1   Close                  2516 non-null   float64
2   High                   2516 non-null   float64
3   Low                    2516 non-null   float64
4   Volume                 2516 non-null   int64  
5   %K                     2503 non-null   float64
6   %D                     2501 non-null   float64
7   SO Indicator           2516 non-null   object  
8   RSI                    2515 non-null   float64
9   RSI Indicator          2516 non-null   object  
10  Bollinger_Upper        2497 non-null   float64
11  Bollinger_Lower        2497 non-null   float64
12  Bollinger Indicator     2516 non-null   object  
13  MACD                   2516 non-null   float64
14  Signal                  2516 non-null   float64
15  MACD Indicator          2516 non-null   object  
16  Recommender            2516 non-null   object  
dtypes: float64(11), int64(1), object(5)
memory usage: 353.8+ KB
```

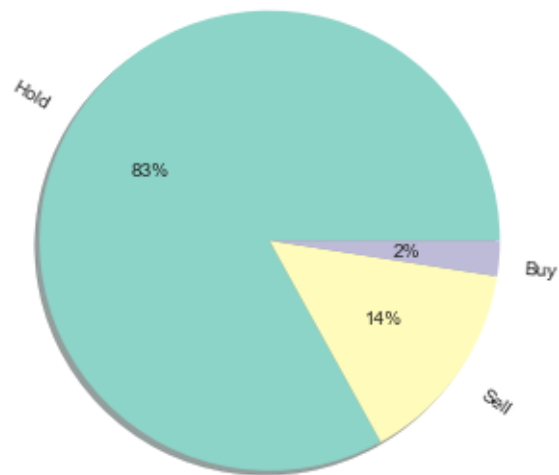
1. The state of the dataset after data cleaning. From the results, you can see that this dataset contains records from '2012-05-23' to the '2022-04-26' and has the extra trading signals columns that was used to calculate the target variable column, their datatypes are denoted as objects. The extra columns containing the trading signal from each indicator (highlighted in red) will be dropped prior to modelling.

2. Plotting Pie Chart: Distribution of trading signals from SO.



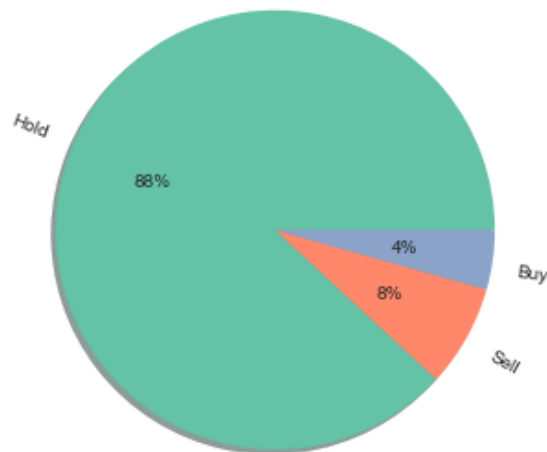
3. Plotting Pie Chart: Distribution of trading signals from SO.

Distribution of Buy/Sell/Hold Signals from RSI



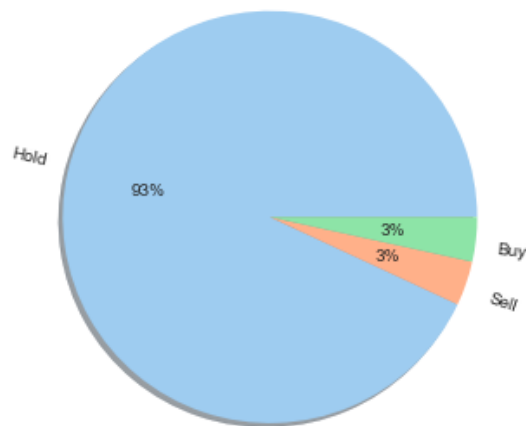
4. Plotting Pie Chart: Distribution of trading signals from BB.

Distribution of Buy/Sell/Hold Signals from Bollinger Indicator



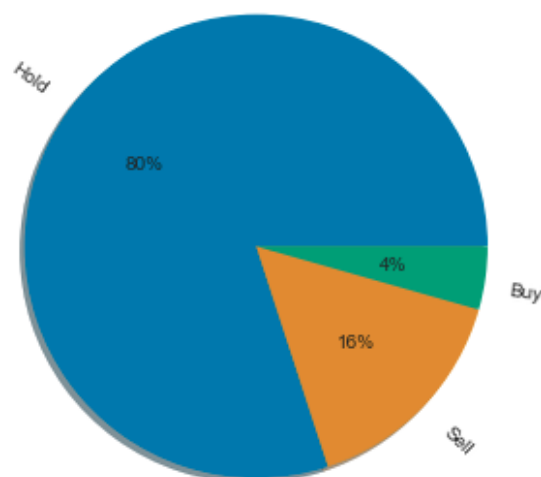
5. Plotting Pie Chart: Distribution of trading signals from MACD.

Distribution of Buy/Sell/Hold Signals from MACD Indicator



6. Plotting Pie Chart: Distribution of trading signals in the target variable.

Distribution of Buy/Sell/Hold Signals: Recommender



7. From the above charts, you can see collectively, there are significantly low 'buy' and 'sell' signals produced compared to 'hold' signals. This makes the dataset imbalanced which will be rectified in the pre-processing stage.

5.2.2 Classification: Data Pre-Processing

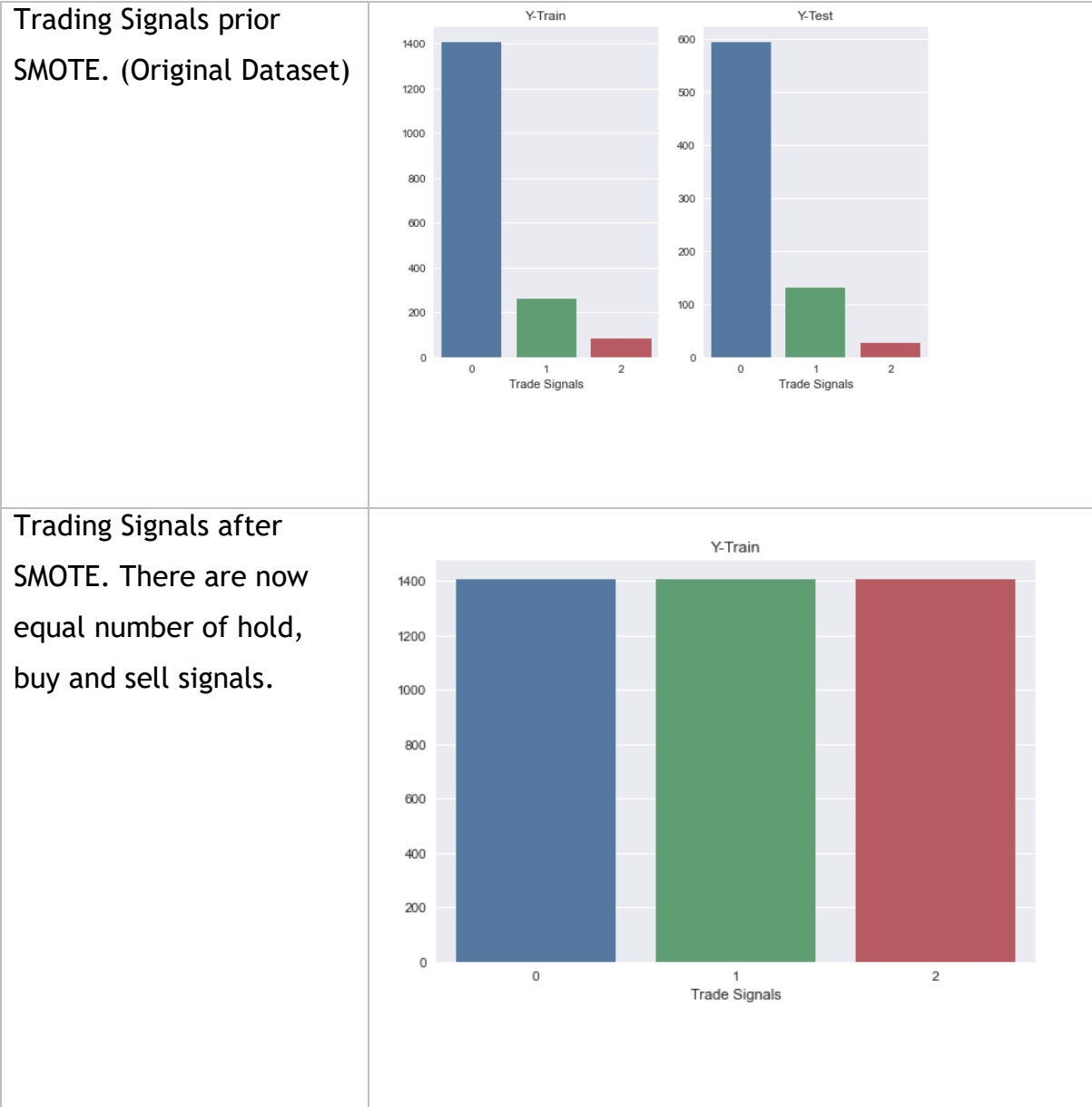
In addition to feature scaling, data encoding and handling of imbalanced dataset must be completed prior to modelling.

Data Encoding is defined as the process of converting categorical data into integer format (Verma, 2021). Dhandare (2020) described that it is necessary to encode categorical variables as many machine learning models can only work with numerical variables as they are required to perform mathematical operations. Categorical variables can be divided into 2 parts: nominal and ordinal; Singh (2020) defined nominal variables to have no intrinsic ordering whereas ordinal to have a clear ranked ordering. In this project, python dictionary was used to map every category to a numerical value such that 0 = Hold, 1 = Sell and 2 = Buy. This may have not been the best method to use as it is important to choose the right technique depending on the type of categorical data as forcing an ordinal relationship between nominal variables can be misleading to the model and result in poor performance.

Next, handling the imbalanced dataset; as mentioned above, the 'hold' signals make up majority of the dataset, accounting for 80% of the records, whereas 'sell' signals account for 16% and 'buy' signals for only 4%. It is essential to resolve this to produce good, accurate results as models trained on an imbalanced dataset will cause a bias and falsely predict on the majority class. The main techniques used to manage imbalanced datasets are over-sampling and under-sampling. Over-sampling is described as duplicating or creating new synthetic examples in the minority class whereas under-sampling includes merging or deleting examples in the majority class (Brownlee, 2020a). For this project, I have chosen to oversample my dataset using Synthetic Minority Oversampling Technique (SMOTE) as stock data are classed as discrete time series data. Therefore, deleting record from the majority class could lead to removal of valuable information which in turn lead to inaccurate

predictions. SMOTE has been described to generate synthetic samples from interpolating between the positive instances that lie together (Satpathy, 2020).

Contrastingly, the models' performance will be analysed using both the original dataset and the SMOTE dataset.



5.2.3 Classification Model: Modelling & Results

Furthermore, cross validation has been implemented to evaluate if the models have been generalised to the dataset. Seldon (2021) has described cross validation as a technique to assess a machine learning model's accuracy on new and unseen data, beyond the training dataset. It is also said that cross validation is a useful tool to highlight if the model has been overfitting to the training data. There are numerous cross validation techniques out there, for this project I have chosen a technique called Stratified K-Fold (SKF) cross validation. SKF is a variation of the K-Fold cross validation technique such that the dataset is split up into 'k' folds. The 'k-1' fold will be assigned as the training data and the rest will be testing data. At each fold, the model is trained with the assigned training/test data and evaluated. This process iterates through until all the folds have been used as testing data (Krishni, 2018). Unlike K-Fold, where the data are split up randomly, SKF splits up the data in a stratified manner to account for the class imbalance in the dataset. Lendave (2021) has described SKF to maintain the same class ratio through the K-folds as the ratio in the original dataset.

Moreover, hyperparameter tuning will also be implemented to the best performing model. Rouse (2021) defined hyperparameters as machine learning parameters that manage and control the behaviour of the model, such that hyperparameter tuning is the process of finding the optimal combination hyperparameters to maximise the model's performance and minimise the loss function. Additionally, Lee (2019) stated that failing to utilise hyperparameter tuning can lead to sub-optimal results. On the other hand, Lee (2019) also expressed that hyperparameter tuning may not be worth its time-consuming and computationally expensive process, just to achieve minor improvements.

Nonetheless, I have utilised two hyperparameter tuning methods called Grid Search and Randomised Search in this project. Grid search works in a similar way to a brute force algorithm such that it iterates through all the possible

combinations of hyperparameters to find the best performing one (Badr, 2019). A disadvantage of this method is that it is very time consuming and requires lots of computational space. Alternatively, random search iterates through randomly picked sets of hyperparameters which is less time consuming and less computationally demanding. However, a drawback of this method is that it may not return the best possible combinations. Additionally, random search does not remember its past iterations which makes this method inefficient (Badr, 2019). To mitigate the weaknesses of these methods, I ran a random search. First to narrow down the range of values of each hyperparameter, and afterwards grid search was ran focusing on the most promising hyperparameter ranges found in the random search.

The following metrics will be used to evaluate the models. However, the F1 score will be considered as the main metric. Thus, models scoring over 85 will be deemed as successful.

Accuracy	Accuracy score represents the percentage of correct predictions out of the total predictions. Accuracy has been stated to only be a useful metrics when there is a balanced dataset (Sunasra, 2027).
----------	--

Formula used to calculate accuracy:

$$Accuracy = \frac{\# \text{ of correct predictions}}{\# \text{ of total predictions}}$$

Image by Korstanje (2022)

Precision	Precision represents the number of true positives predicted out of all the positive predictions. It is calculated by:
-----------	---

$$Precision = \frac{\# \text{ of True Positives}}{\# \text{ of True Positives} + \# \text{ of False Positives}}$$

Image by Korstanje (2022)

Recall	Recall represents the number of true positives correctly identified out of the total number of positive predictions. It is calculated by:
--------	---

$$Recall = \frac{\# \text{ of True Positives}}{\# \text{ of True Positives} + \# \text{ of False Negatives}}$$

Image by Korstanje (2022)

F1 F1 is a combination of the precision and recall metric, where it has been described as a harmonic mean of precision and recall (Korstanje, 2021). Additionally, Korstanje (2021) has described the F1 score to work well on imbalanced dataset. It is calculated by:

$$F1 \text{ score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Image by Korstanje (2022)

For this project, I have explored seven classification models and as I have mentioned above, the best performing model will get further tuned and be implemented on to the application as the main model, granted it scored over 85 on the F1 metric.

MODELS	RESULTS	
	No SMOTE	SMOTE
Logistic Regression (LR) uses mathematical probability to predict the target variable into categories (Agrawal, 2021). Normally, LR is used for binary classification but as I have 3 classes for my target variable, multinomial LR was used.	Avg Accuracy : 0.897 Avg Recall : 0.545 Avg Precision: 0.620 Avg F1 : 0.569	Avg Accuracy : 0.785 Avg Recall : 0.8974 Avg Precision: 0.6279 Avg F1 : 0.6804
	Although accuracy seemed to be good, the recall, precision and F1 have terrible scores. LR did not	A slight improvement was produced when using SMOTE. Recall significantly improved, however

	perform well against an imbalanced dataset.	the precision and F1 are still low. To conclude, the F1 score in both scenarios were low. Therefore, the LR model did not perform well in this case.
Decision Tree (DT) has been described to predict classes by learning simple decision rules from the training data (Chauhan, 2022). Additionally, Roy (2020) has defined DT as a graphical representation that maps all possible solutions to a decision based on certain conditions.	<p>Avg Accuracy : 0.928 Avg Recall : 0.851 Avg Precision: 0.847 Avg F1 : 0.846</p> <p>DT model performed well scoring over 80 in all the metrics.</p>	<p>Avg Accuracy : 0.9303 Avg Recall : 0.8891 Avg Precision: 0.8349 Avg F1 : 0.8556</p> <p>Considering DT performed well on the imbalanced dataset, there was only a small improvement in the scores.</p>
Random Forest (RF) consists of building multiple decision trees that work together as an ensemble. Yiu (2019) explained that each individual tree in RF make a class prediction and the class with most votes becomes the models' predictions.	<p>Avg Accuracy : 0.949 Avg Recall : 0.830 Avg Precision: 0.926 Avg F1 : 0.869</p> <p>RF also performed well on the imbalanced data, scoring 92% on precision.</p>	<p>Avg Accuracy : 0.9415 Avg Recall : 0.902 Avg Precision: 0.8558 Avg F1 : 0.8746</p> <p>Like DT, RF also behaved in the same way such that recall improved, and precision decreased</p>

		when trained on the SMOTE dataset.
<p>Support Vector Machine (SVM) is made up of finding a decision boundary line, known as the hyperplane, to separate data into different classes (Pupale, 2018).</p>	<p>Avg Accuracy : 0.927 Avg Recall : 0.779 Avg Precision: 0.888 Avg F1 : 0.821</p> <p>SVM, although performed satisfactorily, recall was significantly lower than the other metrics.</p>	<p>Avg Accuracy : 0.867 Avg Recall : 0.9163 Avg Precision: 0.7115 Avg F1 : 0.7756</p> <p>Unlike other models, SVM performed significantly worse on the SMOTE dataset. Although it improved recall, all the other metrics were worse.</p>
<p>K-Nearest Neighbours (kNN), also known as the lazy learning algorithm, uses feature similarity to predict the correct classes. Dwivedi (2020) explained kNN to classify new data points based on similarity such that data points falling near to each other will fall in the same category.</p>	<p>Avg Accuracy : 0.925 Avg Recall : 0.799 Avg Precision: 0.883 Avg F1 : 0.833</p> <p>KNN, mimicking the results of SVM, performed better when trained on the original dataset, although recall scored higher in the SMOTE dataset.</p>	<p>Avg Accuracy : 0.8907 Avg Recall : 0.9057 Avg Precision: 0.7445 Avg F1 : 0.8032</p>
<p>Naive Bayes (GNB) is a statistical model that uses conditional probability to make predictions derived from Bayes theorem. Additionally, Navlani (2018) explained that GNB assumes</p>	<p>Avg Accuracy : 0.818 Avg Recall : 0.864 Avg Precision: 0.651 Avg F1 : 0.708</p> <p>GNB was the first model to score</p>	<p>Avg Accuracy : 0.7989 Avg Recall : 0.8619 Avg Precision: 0.6366 Avg F1 : 0.6925</p>

independence between every pair of features in the data.	significantly lower on precision than recall.	The results did not seem to improve when using SMOTE dataset, making it the first model to score better on the original dataset in every single metric.
Multi-Layer Perceptron (MLP) is a deep learning method that relies on it underlying neural networks to make predictions (Nair, 2019).	<p>Avg Accuracy : 0.928</p> <p>Avg Recall : 0.826</p> <p>Avg Precision: 0.869</p> <p>Avg F1 : 0.843</p> <p>MLP also performed decently, scoring over 80 in every single metric. Additionally, the model seemed to perform better on the original dataset.</p>	<p>Avg Accuracy : 0.888</p> <p>Avg Recall : 0.917</p> <p>Avg Precision: 0.7455</p> <p>Avg F1 : 0.8051</p> <p>On the other hand, whilst the recall improved scoring over 90 using the SMOTE dataset, all other metrics score lowered such that precision dropped quite significantly.</p>

Overview of Results

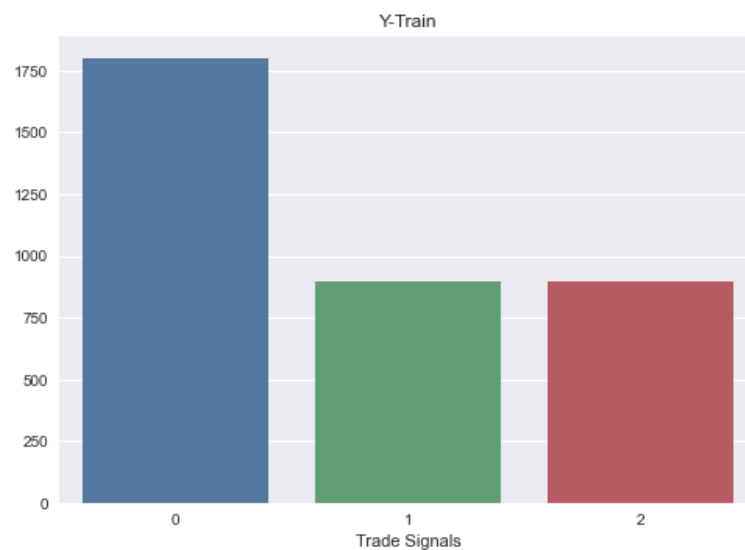
	Model	Accuracy	Precision	Recall	F1
2	Random Forest	0.9495	0.9268	0.8309	0.8692
1	Decision Tree	0.9283	0.8475	0.8512	0.8469
6	MLP	0.9287	0.8697	0.8265	0.8436
4	K-Nearest Neighbour	0.9255	0.8830	0.7992	0.8337
3	Support Vector Machine	0.9271	0.8880	0.7798	0.8214
5	GaussianNB	0.8181	0.6510	0.8642	0.7088
0	Logistic Regression	0.8971	0.6207	0.5459	0.5696

	Models with SMOTE	Accuracy	Precision	Recall	F1
2	Random Forest	0.9415	0.8558	0.9020	0.8746
1	Decision Tree	0.9303	0.8349	0.8891	0.8556
6	MLP	0.8886	0.7455	0.9171	0.8051
4	K-Nearest Neighbour	0.8907	0.7445	0.9057	0.8032
3	Support Vector Machine	0.8670	0.7115	0.9163	0.7756
5	GaussianNB	0.7989	0.6366	0.8619	0.6925
0	Logistic Regression	0.7853	0.6279	0.8974	0.6804

To conclude, more models performed better on the original dataset than the SMOTE dataset in regards to the F1 score. Additionally, the ranking of models performance stayed consistent and followed the same order, between both datasets. On the other hand, the general trend was that training the models on

the SMOTE dataset greatly improved the recall score but decreased precision slightly. RF managed to perform the best all-around between both datasets, scoring the highest in accuracy, precision and F1 score. Thus, RF will be tuned and implemented on to the app.

Furthermore, I decided to experiment on one more scenario which consisted of balancing the dataset in the following ratio: 50% Hold, 25% Buy, 25% Sell. This was done to explore if the models performed better on a dataset that mimicked real world data.



From the results, you can see the models performed quite well such that 6/7 models scored over 80 in the F1 metric. Additionally, 3 models highlighted in green performed better in this dataset compared to before. However, RF still managed to hold the lead, scoring the highest F1 again.

	Models with SMOTE 2	Accuracy	Precision	Recall	F1
5	Random Forest	0.942667	0.851067	0.903528	0.872668
4	Decision Tree	0.930667	0.849990	0.882102	0.865144
6	MLP	0.905333	0.772229	0.928076	0.832063
1	SVM	0.901333	0.758270	0.916864	0.817402
2	KNeighbors	0.909333	0.765858	0.898812	0.816190
0	Logistic Regreesion	0.890667	0.743223	0.924258	0.807865
3	GaussianNB	0.821333	0.656741	0.875907	0.719462

Hyperparameter Tuning Main Model: Random Forest

1. Retrieving the current parameters that my default model is using by applying the function 'get_params()'.

Parameters currently in use:

```
{'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'auto', 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 100, 'n_jobs': None, 'oob_score': False, 'random_state': None, 'verbose': 0, 'warm_start': False}
```

2. From the results above, you can see there are numerous parameters. For this project, I only investigated the following five parameters:

- n_estimators = the number of trees in the model.
- max_features = the maximum number of features required to split a leaf node.
- max_depth = the maximum of depth of each decision tree.
- min_samples_split = the minimum number of samples placed in a leaf node before the node splits.
- min_samples_leaf = the minimum number of data points required in a leaf node.

(Meinert, 2019).

3. Created a grid containing the above hyperparameters so that it can be passed on to random search for sampling.

```
{'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000], 'max_features': ['auto', 'sqrt'], 'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4]}
```

4. Carried out random search with the settings of 'n_iter' set to 100; this controls the number of different combinations to try and 'cv' set to 5; this represents the number of folds for cross validation (Koehrsen, 2018). After its lengthy search, it provided the best parameters of:

```
#Random Search Best Paramters
# print the best parameters
print ('Best Parameters: ', rf_random_search.best_params_)
```

✓ 0.1s Python

```
Best Parameters: {'n_estimators': 800,
'min_samples_split': 2, 'min_samples_leaf': 1,
'max_features': 'sqrt', 'max_depth': None}
```

5. Evaluated the model using the hyperparamters from random search. From the results you can see the results have improved, even if it is marginally.

```
Accuracy Score : 0.9426666666666667
Precision Score : 0.8609392082350725
Recall Score : 0.9032695277732454
TData F1 Score: 0.8797321676380795
```

6. As utilising random search narrowed the range of each hyperparameters, I utilised grid search to exhaustively search for every combinations. Grid search provided the best parameters of:

```
✓ 110m 26.9s Python
```

Fitting 3 folds for each of 1500 candidates, totalling 4500 fits

```
('Best Parameters: ', grid_search.best_params_, ' \n')
```

✓ 0.3s Python

```
Best Parameters: {'max_depth': None, 'max_features': 3,
'min_samples_leaf': 1, 'min_samples_split': 4,
'n_estimators': 600}
```

7. Evaluated the model using the hyperparameteres produced from grid search. As you can see from the result, the improvements were miniscule. However, the time computation required for grid search was far greater than random search.

```
Accuracy Score : 0.9426666666666667  
Precision Score : 0.8592040564938523  
Recall Score : 0.9056665886830525  
F1 Score: 0.8800809917802693
```

8. Additionally, expiremented with another hyperparamter turning method named 'HalvingGridSearch'. Gilde (2021) has described this method to evaluate all the candidates with a small number of resources at first and then iteratively selects the back candidates, using more and more resources. This method also produced good results considering the computation time was dramatically lower than gird search.

```
Accuracy Score : 0.9426666666666667  
Precision Score : 0.8573305467247537  
Recall Score : 0.8988125196217996  
F1 Score: 0.8763259443855976
```

8. From the comparison of results, the grid search model performed the best. Therefore, I exported the model to a file to be tested against other stock data and be implemented on to the app.

```
import pickle  
  
rf_tuned_model = 'rf_tuned_model.sav'  
pickle.dump(gs_rf, open(rf_tuned_model, 'wb'))  
✓ 0.5s
```

9. Tested the exported model on Google's stock data. As you can see from the result, the model performed quite well considering it was trained on a completely different stocks' data.

```
#Loading Model
loaded_model = pickle.load(open(rf_tuned_model, 'rb'))

#Making Prediction with Model
y_pred = loaded_model.predict(X_test)
```

```
Accuracy Score : 0.9373333333333334
Precision Score : 0.7941407821111945
Recall Score : 0.9145034466520734
F1 Score: 0.8252154717777175
```

6. Results

In conclusion, I deem this project to be mostly satisfactory. Although the stock price prediction models did not perform well, the classification model performed well and achieved great scores on the evaluation metrics. Further, the web application prototype met all the user requirements stated at the beginning. Although the backend of the prototype is not deployed and the machine learning aspect has not been integrated into the web app, I managed to create a separate app to showcase the machine learning aspects using Streamlit.

Streamlit was released in 2019 and is an open-source framework that allows for a fast, easy way to convert Python scripts into interactive applications (Kilcommins, 2021). This framework is mainly targeted towards machine learning engineers that work with Python as it allows them to create an adequate web application without adding in any extra HTML/CSS code. Moreover, Streamlit supports many visualisation libraries such as 'matplotlib' and 'plotly' (Grootendorst, 2019).

Here is a screenshot displaying the Streamlit app:

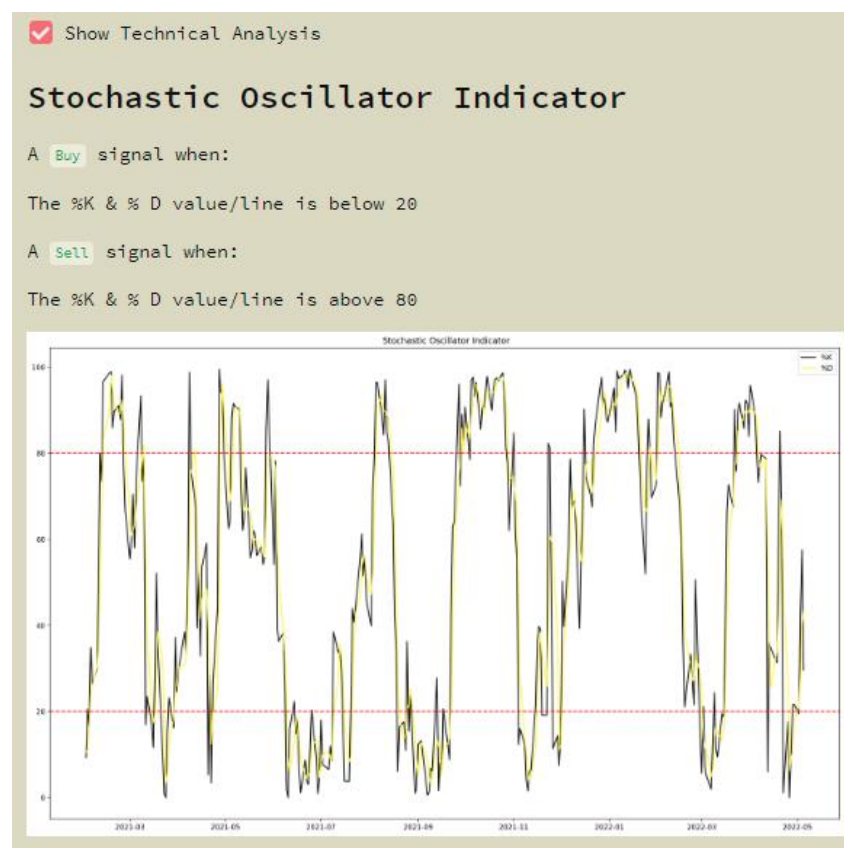
**Machine Learning: Stock
Rregresor & Classifier**

Please Enter A Stock Ticker

HSBC

- ☐ Want to Enter A Custom Start Date for Dataset?
- ☐ Show Stock Dataset Information....
- ☐ Show Stock Data Analysis
- ☐ Show Technical Analysis
- ☐ Classifier Model Performance
- ☐ Regression Model Performance

The users can enter and stock ticker and perform the listed services, for example get guidance regarding technical analysis:



7. Limitations & Recommendations

One of limitations regarding the app that I have developed is that although it allows the user to add and manage their transactions, it is quite inconvenient for the user to manually add all their transactions. Therefore, one way to improve the app would be to implement a system so that user transactions history could be automatically transferred from their banks. However, this would mean that the app's security, safety, and authentication system would be required at the highest level. Additionally, the app would benefit from some sort of classification model to be implemented to automatically categorise user's transactions rather than having the user categorise their transactions manually. Furthermore, a mobile app version to be developed following the mock-ups outlined above.

Secondly, the app was not able to receive feedback from a target audience. This means that the app could not be evaluated by users by means of the proposed measures of Customer Satisfaction Score (CSAT), Customer Effort Score (CES) and Net Promoter Score (NES). Due to this, it is unknown how the user experience and practicality of the app would be in reality. In future, a demonstration showcasing the app to receive feedback will allow for improvements and developments moving forwards, following guidance from user reviews.

Further, the LSTM and ARIMA models did not perform well initially. This should be further explored using tuned and optimal model parameters and increased number of epochs. Additionally, multi-variate models should be explored to predict stock prices including a vast array of technical analysis data as the input data. This research only utilised the data of 4 technical indicators, therefore the implementation of other indicators from other categories and other forms of technical analysis incorporated as input data could improve stock market prediction.

8. References

ABHISHEK, K., 2021. #9 Reasons Why Most Indians Do Not Invest In Stocks [viewed 10/04/ 2022]. Available from: <https://tradebrains.in/reasons-why-most-indians-do-not-invest-in-stocks/>

ADHIKARI, D., 2022. Django vs. Flask: What Works Better for Machine Learning? [viewed 23/02/ 2022]. Available from: <https://datasciencenerd.com/django-vs-flask-what-works-better-for-machine-learning/>

AGRAWAL. SARVAGYA, 2021. Logistic Regression- Supervised Learning Algorithm for Classification [viewed 03/05/ 2022]. Available from: <https://www.analyticsvidhya.com/blog/2021/05/logistic-regression-supervised-learning-algorithm-for-classification/>

AMADEO, K., 2022. The S&P 500 and How It Works [viewed 21/02/ 2022]. Available from: <https://www.thebalance.com/what-is-the-sandp-500-3305888#:~:text=The%20S%26P%20500%20is%20a,all%20other%20investments%20are%20compared.>

ANDERSON, S., 2022. Stochastics: An Accurate Buy and Sell Indicator [viewed 15/04/ 2022]. Available from: <https://www.investopedia.com/articles/technical/073001.asp>

BADR, W., 2019. 3 Different Ways to Tune Hyperparameters (Interactive Python Code) [viewed 04/05/ 2022]. Available from: <https://towardsdatascience.com/3-different-ways-to-tune-hyperparameters-interactive-python-code-87548d7f2365>

BAHETI, P., 2022. A Simple Guide to Data Preprocessing in Machine Learning [viewed 25/04/ 2022]. Available from: <https://www.v7labs.com/blog/data-preprocessing-guide>

BAJAJ, A., 2022. Performance Metrics in Machine Learning [Complete Guide] [viewed 22/02/ 2022]. Available from: <https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide>

BANKRATE INC, 2015. 3 in 5 Americans Don't Have Enough Savings to Pay for Unexpected Expenses [viewed 03/05/ 2022]. Available from: <https://www.prnewswire.com/news-releases/3-in-5-americans-dont-have-enough-savings-to-pay-for-unexpected-expenses-300016840.html>

BARCLAYS, 2021. Should you save cash or invest?[viewed 17/02/ 2022]. Available from: <https://www.barclays.co.uk/smart-investor/new-to-investing/before-you-start/should-you-save-cash-or-invest/#:~:text=Savings%20are%20ideal%20for%20short,cash%20over%20the%20longer%2Dterm.>

BARONE, A., 2022. Top Technical Indicators for Rookie Traders [viewed 14/04/ 2022]. Available from: <https://www.investopedia.com/articles/active-trading/011815/top-technical-indicators-rookie-traders.asp>

BINANCE ACADEMY, 2018. Bollinger Bands Explained [viewed 18/04/ 2022]. Available from: <https://academy.binance.com/en/articles/bollinger-bands-explained>

BROWNLEE, J., 2020a. How to Combine Oversampling and Undersampling for Imbalanced Classification [viewed 26/04/ 2022]. Available from:

<https://machinelearningmastery.com/combine-oversampling-and-undersampling-for-imbalanced-classification/#:~:text=Oversampling%20methods%20duplicate%20or%20create,of%20methods%20are%20used%20together.>

BROWNLEE, J., 2020b. How to Use StandardScaler and MinMaxScaler Transforms in Python [viewed 25/04/ 2022]. Available from: <https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/>

BROWNLEE, J., 2021a. A Gentle Introduction to Long Short-Term Memory Networks by the Experts [viewed 04/05/ 2022]. Available from: <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/>

BROWNLEE, J., 2021b. Regression Metrics for Machine Learning [viewed 22/02/ 2022]. Available from: <https://machinelearningmastery.com/regression-metrics-for-machine-learning/>

CFI, n.d. Technical Indicator [viewed 14/04/ 2022]. Available from: <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/technical-indicator/>

CHAUHAN, N., 2022. Decision Tree Algorithm, Explained [viewed 03/05/ 2022]. Available from: <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html>

CHEN, J., 2021a. Technical Indicator [viewed 14/04/ 2022]. Available from: <https://www.investopedia.com/terms/t/technicalindicator.asp>

CHEN, J., 2021b. What Is Technical Analysis?[viewed 14/04/ 2022]. Available from: <https://www.investopedia.com/terms/t/technical-analysis-of-stocks-and-trends.asp#toc-the-difference-between-technical-analysis-and-fundamental-analysis>

CHEN, J., 2022. What Is Debt?[viewed 10/02/022 Available from: <https://www.investopedia.com/terms/d/debt.asp>

CHOI, J.J. and A.Z. ROBERTSON, 2020. What Matters to Individual Investors? Evidence from the Horse's Mouth. *The Journal of finance* (New York), 75(4), 1965-2020

CHRISTIE, A., 2020. Time Series & Stock Analysis [viewed 20/04/ 2022]. Available from: <https://www.rpubs.com/AurelliaChristie/time-series-and-stock-analysis>

CHU, Z. et al., 2017. Financial Literacy, Portfolio Choice and Financial Well-Being. *Social Indicators Research*, 132(2), 799-820

CLARK, D., 2020. Average saving rates at their lowest levels on record [viewed 21/02/ 2022]. Available from: <https://moneyfacts.co.uk/news/savings/average-saving-rates-at-their-lowest-levels-on-record/>

CLOSE BROTHERS, 2019. 25 million UK employees affected by money worries while at work [viewed 17/02/ 2022]. Available from: <https://www.closebrothersam.com/for-employers/news-and-insights/25-million-uk-employees-affected-by-money-worries-while-at-work/#:~:text=The%20vast%20majority%20of%20UK,new%20research%20from%20Clo>

se%20Brothers.&text=Additionally%2C%20two%20in%20five%20(40,their%20finances%20always%20or%20often

DAVIES, R., 2022. What does high inflation mean? What happens when inflation rises and what it means for Scotland [viewed 21/02/ 2022]. Available from: <https://www.scotsman.com/lifestyle/money/what-does-high-inflation-mean-what-happens-when-inflation-rises-3460604>

DEERY, M., 2021. What Is Flask and How Do Developers Use It? A Quick Guide [viewed 23/02/ 2022]. Available from: <https://careerfoundry.com/en/blog/web-development/what-is-flask/>

DHANDARE, S., 2021. What Is Encoding? And Its Importance in Data Science![viewed 26/04/ 2022]. Available from: <https://medium.datadriveninvestor.com/what-is-encoding-and-its-importance-in-data-science-6a2b0cce8e8e#:~:text=Models%20only%20work%20with%20numerical,numerical%20data%20is%20called%20Encoding.>

DWIVEDI, R., 2020. How Does K-nearest Neighbor Works In Machine Learning Classification Problem?[viewed 03/05/ 2022]. Available from: <https://www.analyticssteps.com/blogs/how-does-k-nearest-neighbor-works-machine-learning-classification-problem>

FERNANDO, J., 2022. Relative Strength Index (RSI) [viewed 15/04/ 2022]. Available from: <https://www.investopedia.com/terms/r/rsi.asp>

FOLGER, J., 2022. Using Technical Indicators to Develop Trading Strategies [viewed 14/04/ 2022]. Available from:

<https://www.investopedia.com/articles/trading/11/indicators-and-strategies-explained.asp>

FORBES, J. and S.M. KARA, 2010. Confidence mediates how investment knowledge influences investing self-efficacy. *Journal of Economic Psychology*, 31(3), 435-443

FRANZEN, E. and L. BRADARIC, 2018. Financial Wellness Applied Market Research: Review of Financial Wellness Literature and Budgeting Apps Dr. Kelly LaVenture June 23, 2018.

FRENCH, D., D. MCKILLOP and E. STEWART, 2020. The effectiveness of smartphone apps in improving financial capability. *null*, 26(4-5), 302-318

FRENCH, D., D. MCKILLOP and E. STEWART, 2021. Personal finance apps and low-income households. *Strategic Change*, 30(4), 367-375

GANDHMAL, D.P. and K. KUMAR, 2019. Systematic analysis and review of stock market prediction techniques. *Computer Science Review*, 34, 100190

GHOSH, S., 2022. The Ultimate Guide to Evaluation and Selection of Models in Machine Learning [viewed 22/02/ 2022]. Available from:
<https://neptune.ai/blog/the-ultimate-guide-to-evaluation-and-selection-of-models-in-machine-learning>

GILDE, K., 2021. Faster Hyperparameter Tuning with Scikit-Learn's HalvingGridSearchCV [viewed 05/05/ 2022]. Available from:
<https://towardsdatascience.com/faster-hyperparameter-tuning-with-scikit-learn-71aa76d06f12>

GOLD, H., 2015. Here's the real reason why people don't invest enough in stocks [viewed 10/04/ 2022]. Available from: <https://www.marketwatch.com/story/heres-the-real-reason-why-people-dont-invest-enough-in-stocks-2015-04-15>

GROOTENDORST, M., 2019. Quickly Build and Deploy a Dashboard with Streamlit [viewed 12/01/ 2022]. Available from: <https://towardsdatascience.com/quickly-build-and-deploy-an-application-with-streamlit-988ca08c7e83>

GUMPARTHI, S., 2017. Relative strength index for developing effective trading strategies in constructing optimal portfolio. International Journal of Applied Engineering Research, 12(19), 8926-8936

GUPTA, S., 2021. <https://careerfoundry.com/en/blog/web-development/what-is-flask/> [viewed 23/02/ 2022]. Available from: <https://www.springboard.com/blog/data-science/best-language-for-machine-learning/>

HALE, J., 2019. Scale, Standardize, or Normalize with Scikit-Learn [viewed 04/05/ 2022]. Available from: <https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02>

HALL, A., 2018. Brits spend £144,000 on 'impulse buys' during lifetime, research finds [viewed 15/02/ 2021]. Available from: <https://www.independent.co.uk/news/business/news/consumer-spending-impulse-buys-lifetime-average-sweets-clothes-takeaways-coffee-lunch-a8159651.html>

HANNAH, J., 2021. What Exactly Is Wireframing? A Comprehensive Guide [viewed 05/04/ 2022]. Available from: <https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide/>

HARPER, A. et al., 2021. Debt, Incarceration, and Re-entry: a Scoping Review. *American Journal of Criminal Justice*, 46(2), 250-278

HAYES, A., 2021. Stochastic Oscillator [viewed 14/04/ 2022]. Available from: <https://www.investopedia.com/terms/s/stochasticoscillator.asp>

HAYES, A., 2022. What Is a Stock Split?[viewed 24/04/ 2022]. Available from: <https://www.investopedia.com/terms/s/stocksplit.asp>

HOGARTH, J., M. HILGERT and J. SCHUCHARDT, 2002. Money managers: the good, the bad, and the lost.

HSBC, Saving vs investing [viewed 21/02/ 2022]. Available from: <https://www.hsbc.co.uk/wealth/articles/saving-vs-investing/>

INMAN, P., 2022. UK inflation rises to highest level in almost 30 years at 5.4% [viewed 21/02/ 2022]. Available from: <https://www.theguardian.com/business/2022/jan/19/uk-inflation-hits-near-three-decade-high-rising-to-54#:~:text=Britain's%20cost%20of%20living%20crisis%20worsened%20in%20December%20after%20inflation,of%20clothes%2C%20food%20and%20footwear.>

JOBLING, P., 2021. Online shopping has surged during the pandemic - but can retailers keep pace with continuous high demand?[viewed 17/02/ 2022]. Available

from: <https://www.business-live.co.uk/special-features/online-shopping-surged-during-pandemic-21202301>

KHARWAL, A., 2021. R2 Score in Machine Learning [viewed 22/02/ 2022]. Available from: <https://thecleverprogrammer.com/2021/06/22/r2-score-in-machine-learning/>

KILCOMMINS, S., 2021. Streamlit – Everything You Need To Know [viewed 13/01/ 2022]. Available from: <https://medium.datadriveninvestor.com/streamlit-everything-you-need-to-know-665eb90fcf4a>

KOEHRSEN, W., 2018. Hyperparameter Tuning the Random Forest in Python [viewed 04/05/ 2022]. Available from: <https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>

KORSTANJE, J., 2021. The F1 score [viewed 03/05/ 2022]. Available from: <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>

KRISHNI, 2018. K-Fold Cross Validation [viewed 03/05/ 2022]. Available from: <https://medium.datadriveninvestor.com/k-fold-cross-validation-6b8518070833>

KUHNEN, C.M. and A.C. MIU, 2017. Socioeconomic status and learning from financial information. *Journal of Financial Economics*, 124(2), 349-372

LAKE, R. and FOREMAN, D., 2021. Are Budgeting Apps Worth It?[viewed 17/02/ 2022]. Available from: <https://www.forbes.com/advisor/banking/are-budgeting-apps-worth-it/>

LEE, A., 2019. Why you should do Feature Engineering first, Hyperparameter Tuning second as a Data Scientist [viewed 04/05/ 2022]. Available from: <https://towardsdatascience.com/why-you-should-do-feature-engineering-first-hyperparameter-tuning-second-as-a-data-scientist-334be5eb276c>

LENDAVE, V., 2021. Performance Measure of Stratified K-Fold Cross-Validation [viewed 03/05/ 2022]. Available from: <https://analyticsindiamag.com/hands-on-tutorial-on-performance-measure-of-stratified-k-fold-cross-validation/#:~:text=The%20stratified%20k%20fold%20cross,ratio%20in%20the%20original%20dataset.>

LIU, C., 2020. Data Transformation: Standardization vs Normalization [viewed 25/04/ 2022]. Available from: <https://www.kdnuggets.com/2020/04/data-transformation-standardization-normalization.html>

MADEIRA, TIAGO, 2020. Why use Python for Web Development?[viewed 23/02/ 2022]. Available from: <https://www.imaginarycloud.com/blog/why-use-python-for-web-development/>

MAVERICK, J.B., 2022. What Is the Average Annual Return for the S&P 500?[viewed 21/02/ 2022]. Available from: <https://www.investopedia.com/ask/answers/042415/what-average-annual-return-sp-500.asp>

MEINERT, R., 2019. Optimizing Hyperparameters in Random Forest Classification [viewed 04/05/ 2022]. Available from: <https://towardsdatascience.com/optimizing-hyperparameters-in-random-forest-classification-ec7741f9d3f6>

MITCHELL, C., 2022. Using Bollinger Bands to Gauge Trends [viewed 18/04/ 2022]. Available from: <https://www.investopedia.com/trading/using-bollinger-bands-to-gauge-trends/>

MONEY HELPER, 2022. Inflation - what does it mean for your savings [viewed 17/02/ 2022]. Available from: <https://www.moneyhelper.org.uk/en/savings/how-to-save/inflation-what-the-saver-needs-to-know#:~:text=The%20only%20rule%20is%20that,get%20your%20hands%20on%20soon.>

MONTFORD, W. and R.E. GOLDSMITH, How gender and financial self-efficacy influence investment risk taking. International journal of consumer studies, 40(1), 101-106

MORALES, J., 2021. Mobile First Design Strategy: The When, Why and How [viewed 05/04/ 2022]. Available from: <https://xd.adobe.com/ideas/process/ui-design/what-is-mobile-first-design/#:~:text=Mobile%2Dfirst%20design%20is%20a,core%20functions%20of%20their%20product.>

MUNAGALA, R., 2021. Feature Scaling [viewed 25/04/ 2022]. Available from: <https://www.numpyninja.com/post/feature-scaling>

NABI, J., 2018. Machine Learning — Multiclass Classification with Imbalanced Dataset [viewed 26/04/ 2022]. Available from: <https://towardsdatascience.com/machine-learning-multiclass-classification-with-imbalanced-data-set-29f6a177c1a>

NAIR, A., 2019. A Beginner's Guide To Scikit-Learn's MLPClassifier [viewed 03/05/ 2022]. Available from: <https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/>

NAVLANI, A., 2018. Naive Bayes Classification Tutorial using Scikit-learn [viewed 03/05/ 2022]. Available from: <https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn>

O'NEILL, B., J.J. XIAO and K. ENSLE, 2017. Positive Health and Financial Practices: Does Budgeting Make a Difference? Journal of family and consumer sciences, 109(2), 27-36

ONG, Q., W. THESEIRA and I.Y.H. NG, 2019. Reducing debt improves psychological functioning and changes decision-making in the poor. Proceedings of the National Academy of Sciences - PNAS, 116(15), 7244-7249

PANG, X. et al., 2020. An innovative neural network approach for stock market prediction. The Journal of Supercomputing, 76(3), 2098-2118

PETERSSON, D., 2021. Supervised learning [viewed 20/04/ 2022]. Available from: <https://www.techtarget.com/searchenterpriseai/definition/supervised-learning>

POTTERS, C., 2022. How To Use a Moving Average to Buy Stocks [viewed 24/04/ 2022]. Available from: <https://www.investopedia.com/articles/active-trading/052014/how-use-moving-average-buy-stocks.asp>

PUPALE, R., 2018. Support Vector Machines(SVM) – An Overview [viewed 03/05/ 2022]. Available from: <https://towardsdatascience.com/https-medium-com->

pupalerushikesh-svm-

f4b42800e989#:~:text=SVM%20or%20Support%20Vector%20Machine,separates%20the%20data%20into%20classes.

RAFAL and MATT, 2021. React Front-End Development - 6 Things To Consider Before Choosing [viewed 06/04/ 2022]. Available from: <https://brainhub.eu/library/reasons-to-choose-react/>

ROSS, S., 2021. Relative Strength Index vs. Stochastic Oscillator [viewed 18/04/ 2022]. Available from: <https://www.investopedia.com/ask/answers/012015/what-are-differences-between-relative-strength-index-rsi-stochastic-oscillator.asp#:~:text=While%20relative%20strength%20index%20was,in%20sideways%20or%20choppy%20markets.>

ROUSE, M., 2021. Hyperparameter [viewed 04/05/ 2022]. Available from: <https://www.techopedia.com/definition/34625/hyperparameter-ml-hyperparameter>

ROY, A., 2020. A Dive Into Decision Trees [viewed 03/05/ 2022]. Available from: <https://towardsdatascience.com/a-dive-into-decision-trees-a128923c9298#:~:text=A%20decision%20tree%20is%20a,dataset%20to%20the%20fullest%20purity.>

ROY, B., 2020. All about Feature Scaling [viewed 25/04/ 2022]. Available from: <https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35>

SARAVANAN, P., 2019. Fundamental or technical analysis: Which is better?[viewed 14/04/ 2022]. Available from:

<https://www.financialexpress.com/market/fundamental-or-technical-analysis-which-is-better/1659945/>

SATPATHY, S., 2020. Overcoming Class Imbalance using SMOTE Techniques [viewed 26/04/ 2022]. Available from:
<https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/#:~:text=SMOTE%20is%20an%20oversampling%20technique,problem%20posed%20by%20random%20oversampling.>

SAXENA, S., 2021. Introduction to Long Short Term Memory (LSTM) [viewed 24/02/ 2022]. Available from:
<https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>

SCHLOSSBERG, B., 2022. How to Trade the MACD Divergence [viewed 18/04/ 2022]. Available from:
<https://www.investopedia.com/articles/forex/05/macddiverge.asp>

SEGAL, T., 2021. Fundamental Analysis [viewed 14/04/ 2022]. Available from:
<https://www.investopedia.com/terms/f/fundamentalanalysis.asp>

SELDON, 2021. What is Cross Validation in Machine Learning [viewed 03/05/ 2022]. Available from: <https://www.seldon.io/cross-validation-in-machine-learning>

SHARMA, P., 2021. Machine Learning for Stock Market Prediction [viewed 04/05/ 2022]. Available from: <https://www.analyticsvidhya.com/blog/2021/10/machine-learning-for-stock-market-prediction-with-step-by-step-implementation/>

SILBERSTEIN, S., 2022. Moving Average Convergence Divergence (MACD) [viewed 18/04/ 2022]. Available from: <https://www.investopedia.com/terms/m/macd.asp>

SONAL, 2021. Importance Of Exploratory Data Analysis Before ML Modelling [viewed 20/04/ 2022]. Available from: <https://blog.eduonix.com/bigdata-and-hadoop/importance-exploratory-data-analysis-ml-modelling/>

STACKOVERFLOW, 2019. Developer Survey Results

2019 [viewed 05/05/ 2022]. Available from: https://insights.stackoverflow.com/survey/2019#technology_-_most-loved-dreaded-and-wanted-web-frameworks

SUNASRA, M., 2022. Performance Metrics for Classification problems in Machine Learning [viewed 03/05/ 2022]. Available from: <https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>

SURVEY OF CONSUMER FINANCES CHARTBOOK, 2016. 2016 SCF Chartbook [viewed 03/05/ 2022]. Available from: <https://www.federalreserve.gov/econres/files/BulletinCharts.pdf>

SWANTON, T.B. and S.M. GAINSBURY, 2020. Gambling-related consumer credit use and debt problems: a brief review. *Current Opinion in Behavioral Sciences*, 31, 21-31

TABLEAU, 2022. Time Series Forecasting: Definition, Applications, and Examples [viewed 20/04/ 2022]. Available from:
<https://www.tableau.com/learn/articles/time-series-forecasting>

TAMBOLI, N., 2021. All You Need To Know About Different Types Of Missing Data Values And How To Handle It [viewed 24/04/ 2022]. Available from:
<https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/#:~:text=It%20is%20important%20to%20handle,support%20data%20with%20missing%20values.>

THE MONEY CHARITY, 2022. The Money Statistics January 2022 [viewed 17/02/ 2022]. Available from: <https://themoneycharity.org.uk/money-statistics/january-2022/>

THE STREET, 2022. What Are Stock Fundamentals? Definition, Examples & FAQ [viewed 14/04/ 2022]. Available from:
<https://www.thestreet.com/dictionary/f/fundamentals>

VAN BEEK, G., V. DE VOGEL and D. VAN DE MHEEN, 2021. The relationship between debt and crime: A systematic and scoping review. *European Journal of Probation*, 13(1), 41-71

VAN ROOIJ, M., A. LUSARDI and R. ALESSIE, 2011. Financial literacy and stock market participation. *Journal of Financial Economics*, 101(2), 449-472

VERMA, Y., 2021. A Complete Guide to Categorical Data Encoding [viewed 26/04/ 2022]. Available from: <https://analyticsindiamag.com/a-complete-guide-to-categorical-data-encoding/#:~:text=float%20or%20integer.->

,Encoding%20categorical%20data%20is%20a%20process%20of%20converting%20categorical%20data,give%20and%20improve%20the%20predictions.

WEST, Z., n.d. Stochastic Oscillator: Predicting Trend Reversals for Better Entries in Trading [viewed 15/04/ 2022]. Available from: <https://www.alpharithms.com/stochastic-oscillator-574218/>

WHISTL, 2017. Brits spend over £3 billion on impulse buys every month [viewed 17/02/ 2022]. Available from: <https://www.whistl.co.uk/news/brits-spend-over-3-billion-on-impulse-buys-every-month>

WU, S., 2020. 3 Best metrics to evaluate Regression Model?[viewed 04/05/ 2022]. Available from: <https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b>

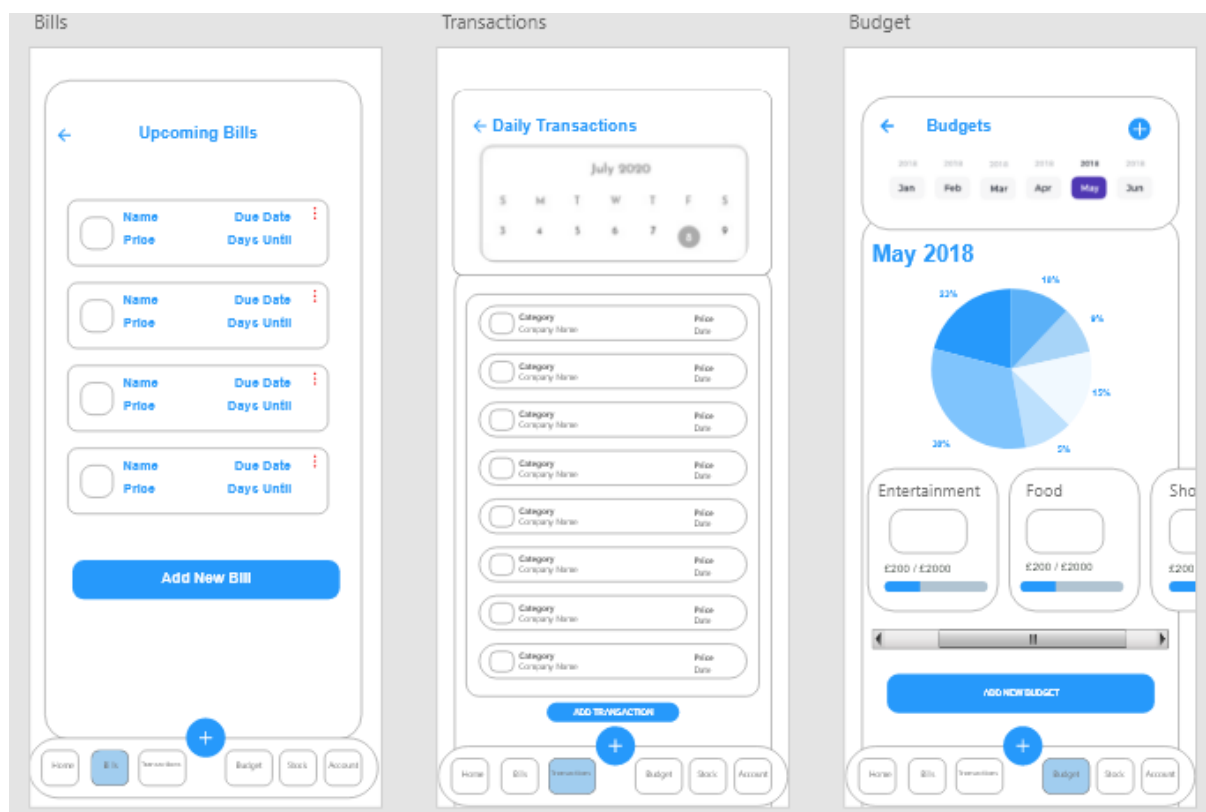
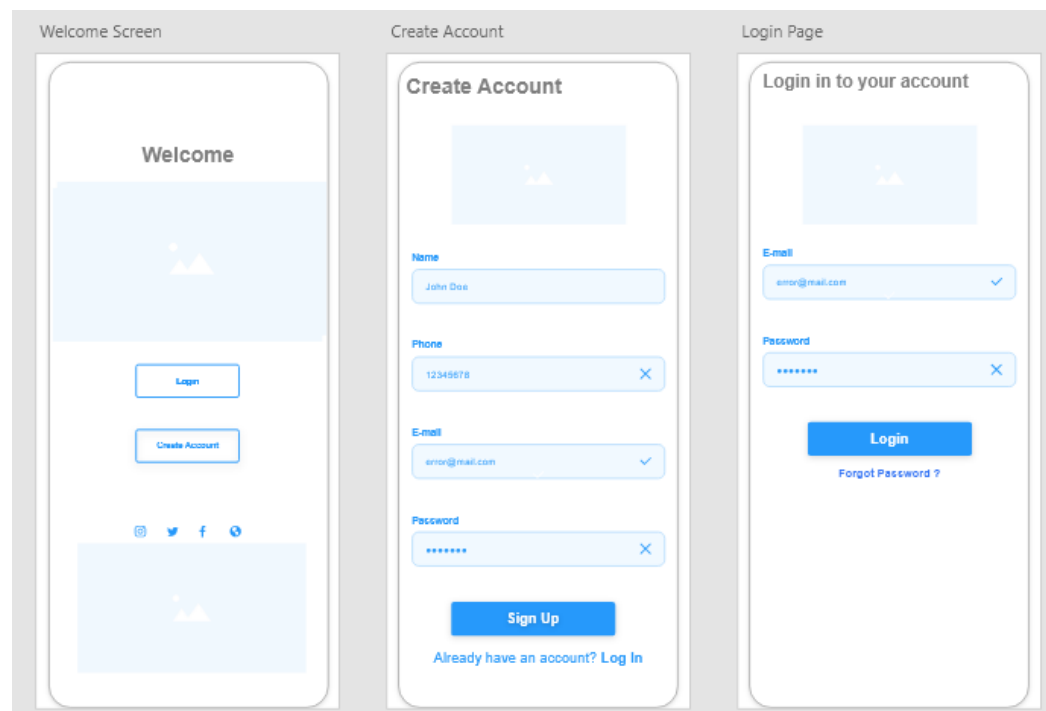
XU, Y., A. GHOSE and B. XIAO, 2019. Mobile payment adoption: An empirical investigation on Alipay. Available at SSRN,

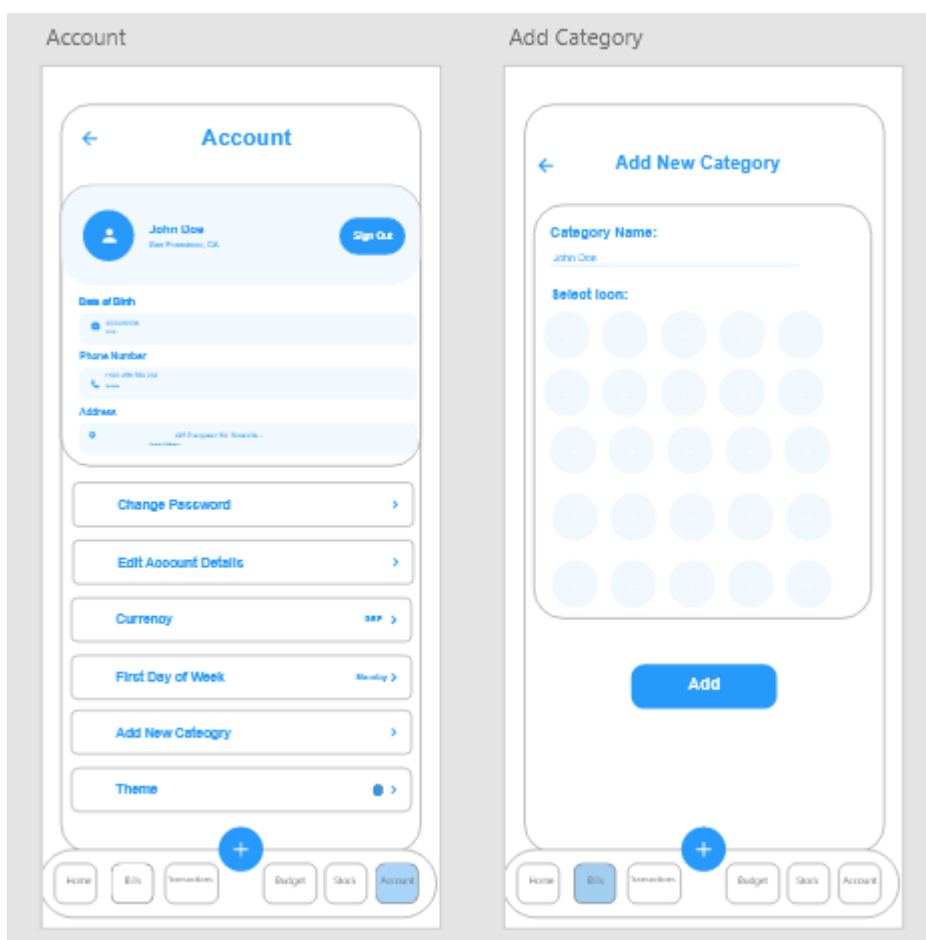
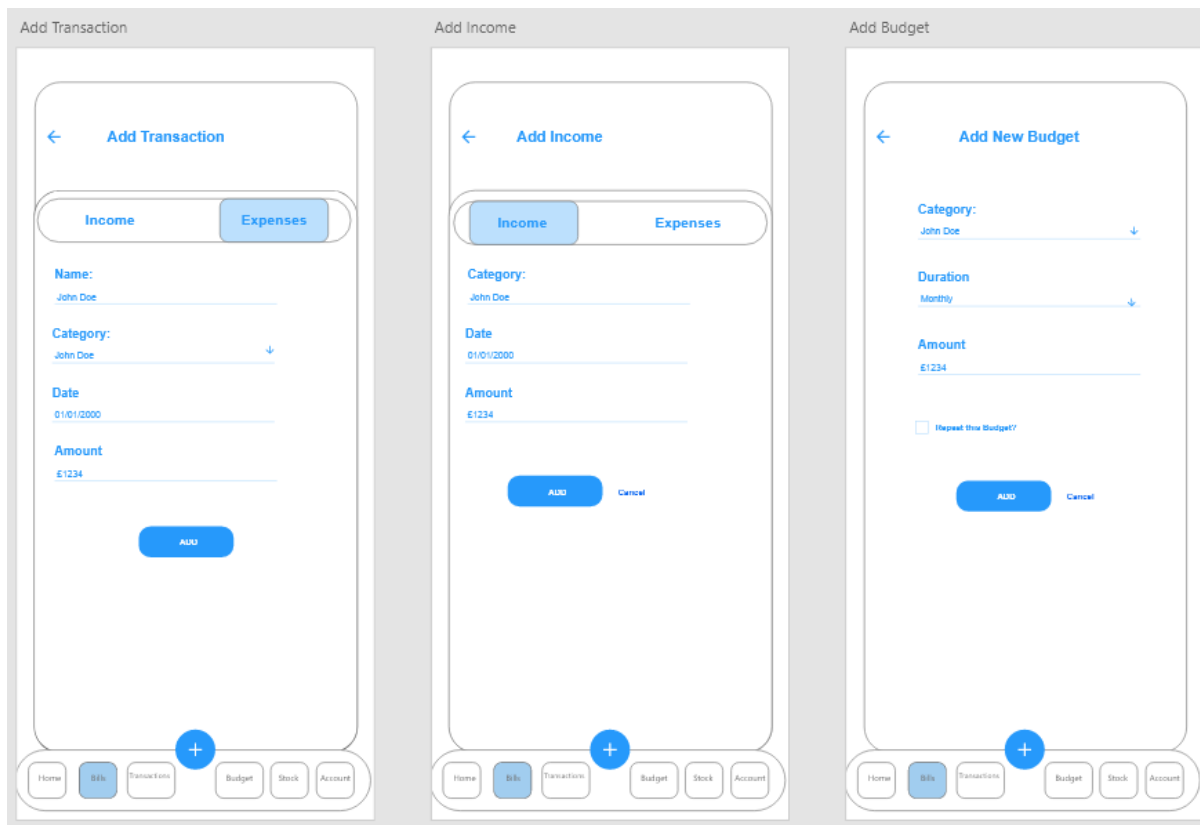
YIU, T., 2019. Understanding Random Forest [viewed 03/05/ 2022]. Available from: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

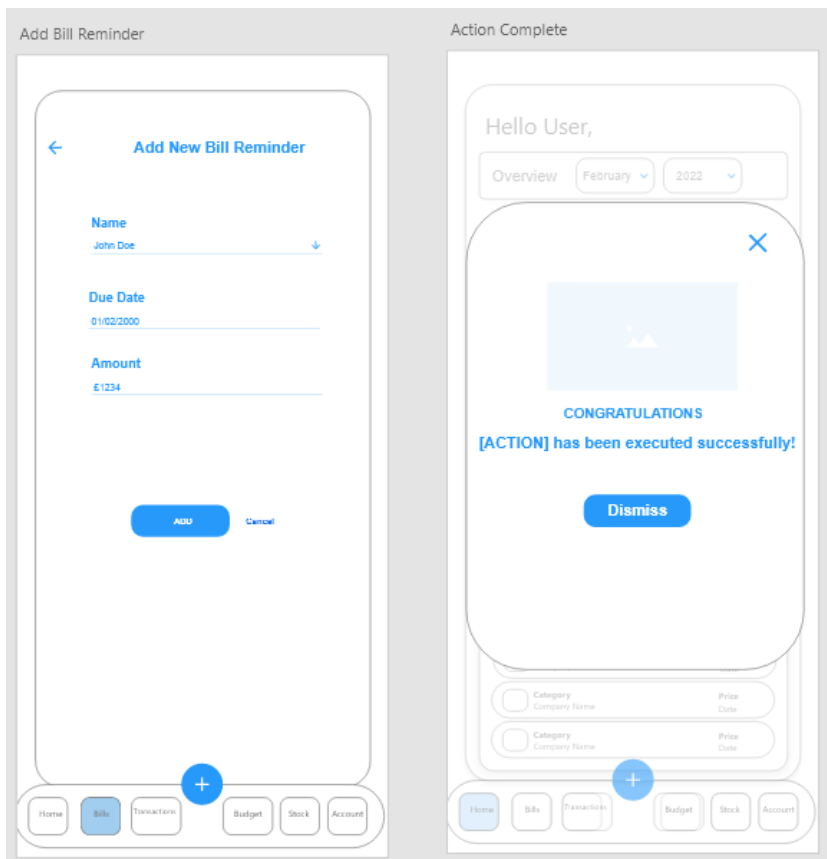
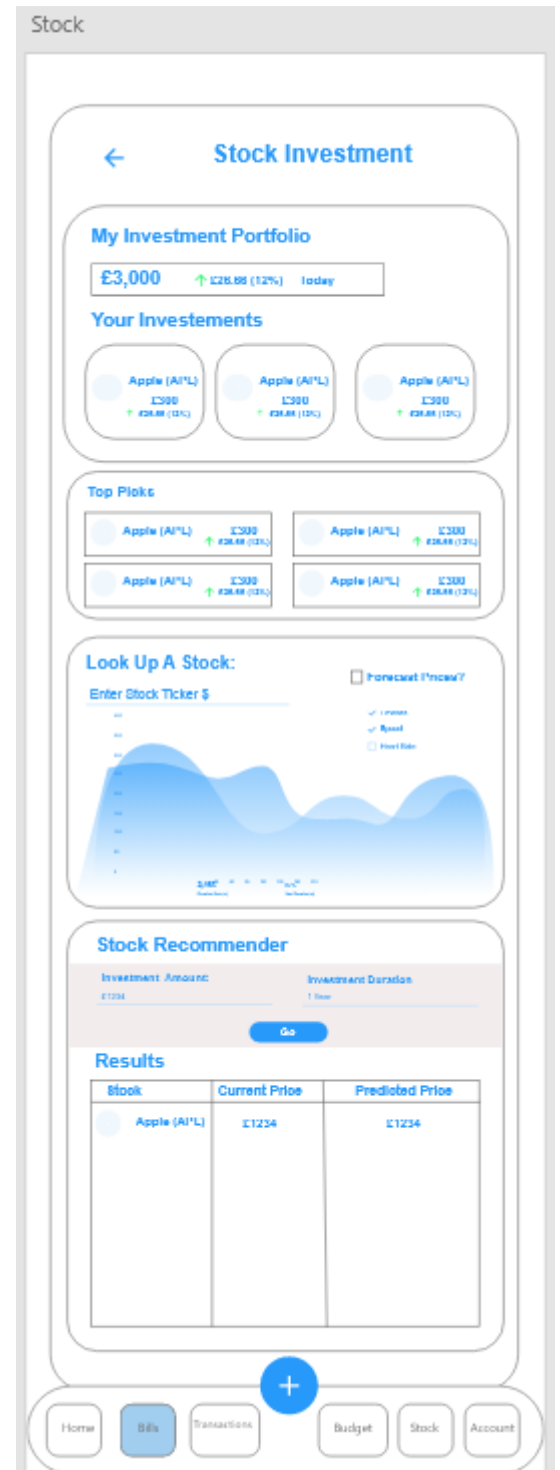
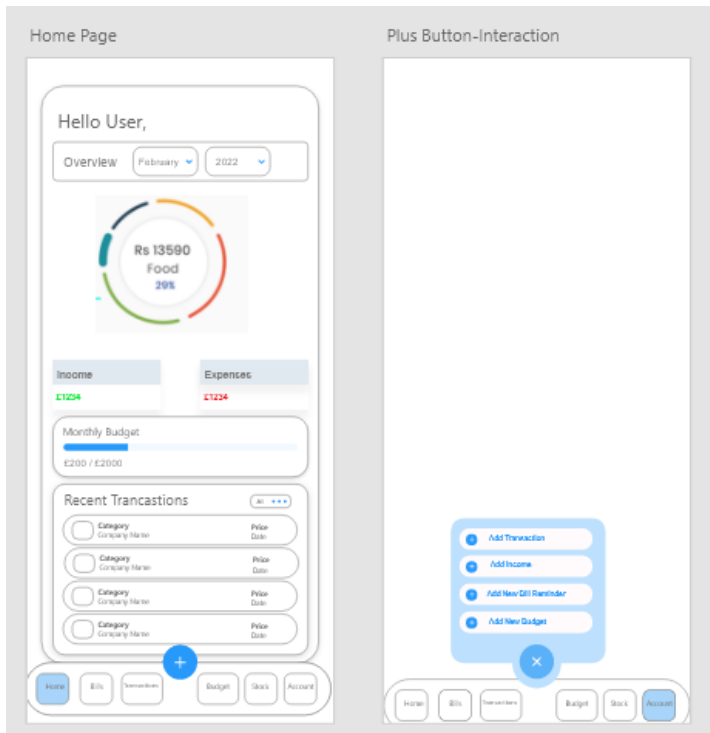
ZHEN, S., 2022. Top 10 Most Common Causes of Debt [viewed 17/02/ 2022]. Available from: <https://www.mybanktracker.com/news/top-10-debt>

ZOU, Z. and Z. QU, 2020. Using LSTM in Stock prediction and Quantitative Trading. CS230: Deep Learning, Winter,

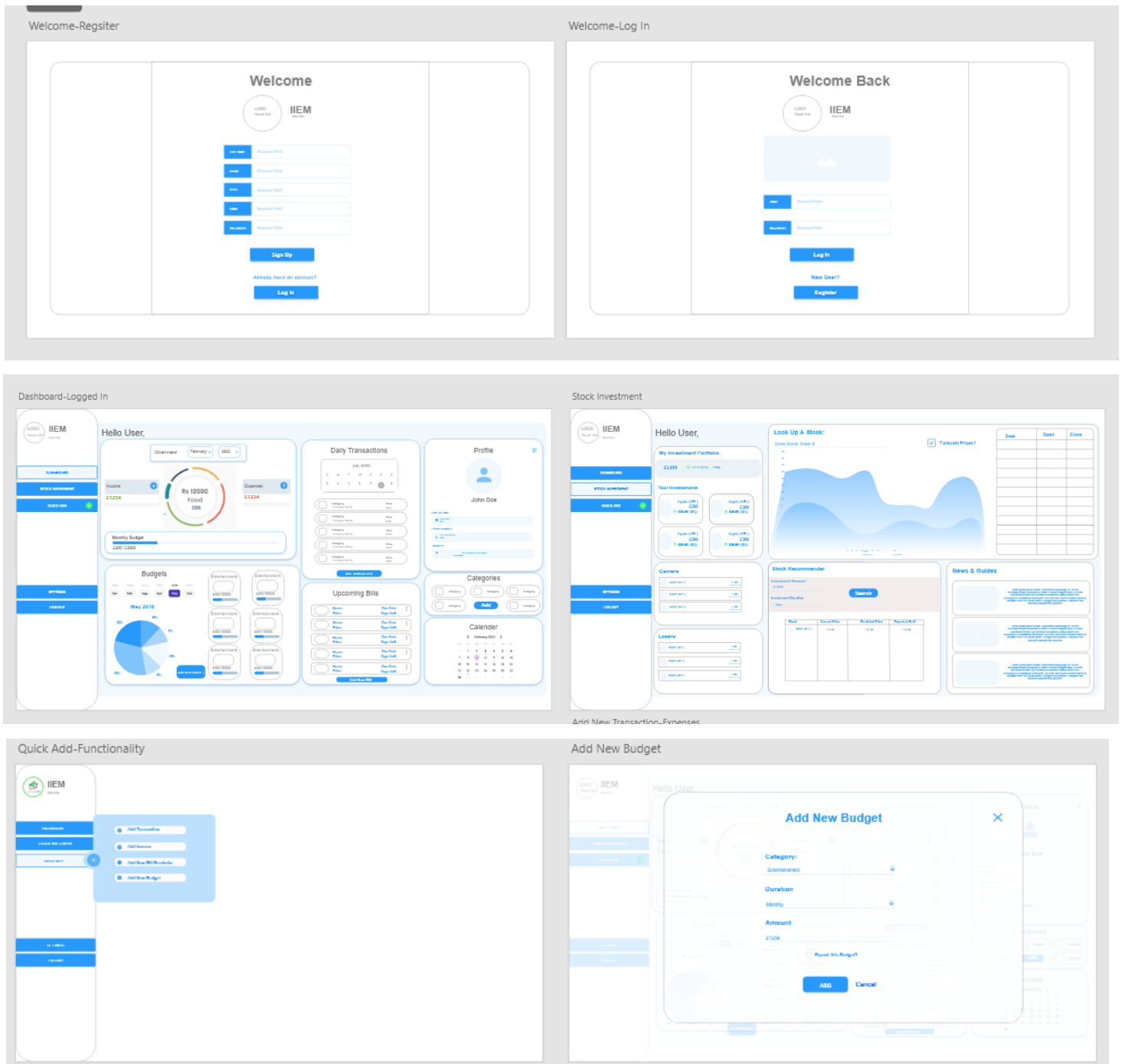
Appendix- A (Mobile Wireframes)

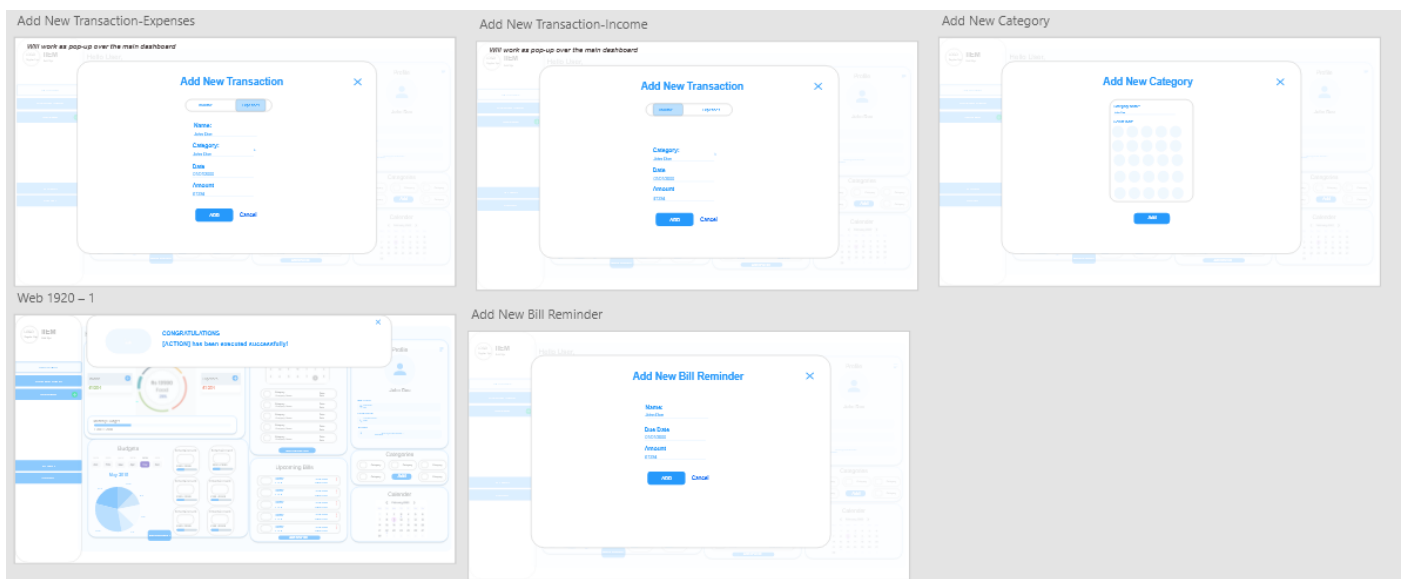




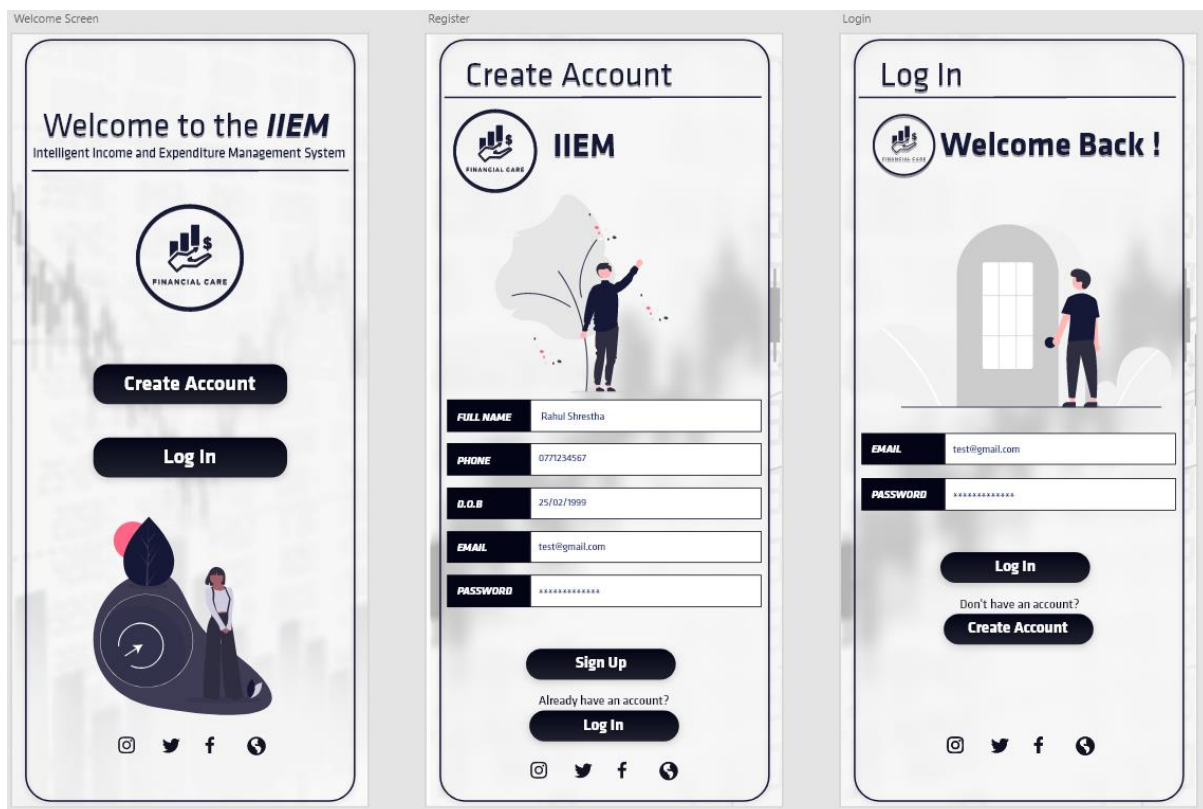


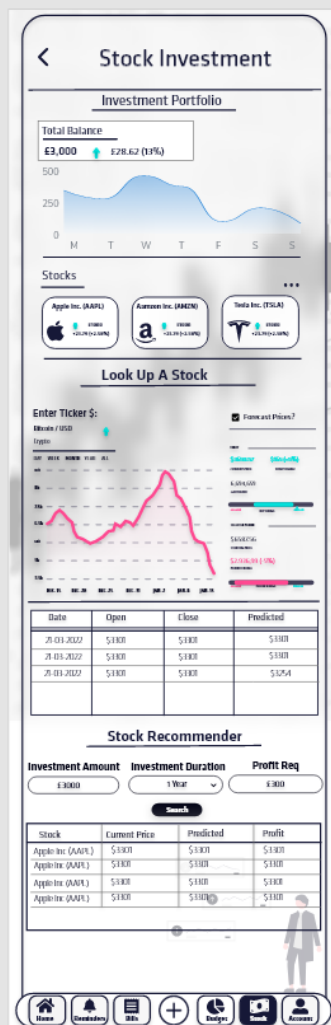
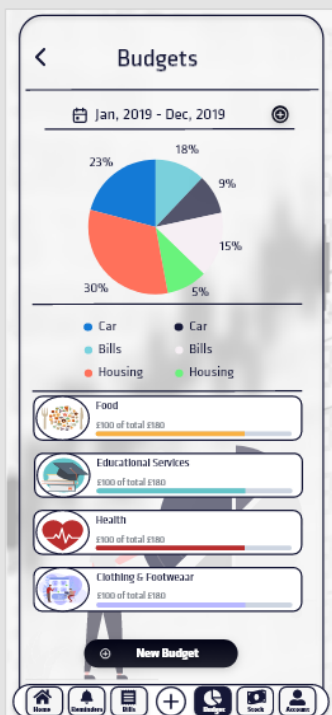
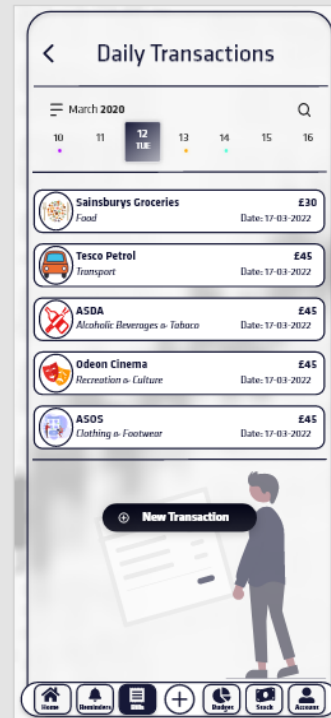
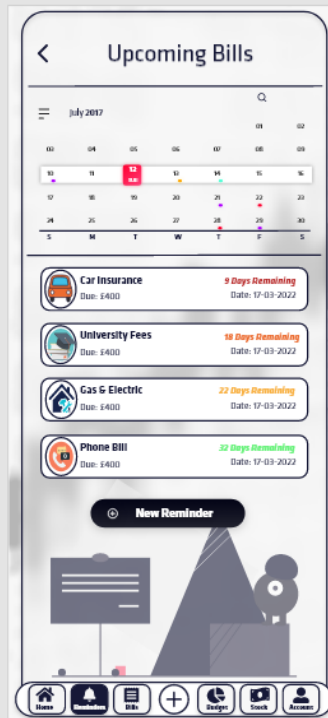
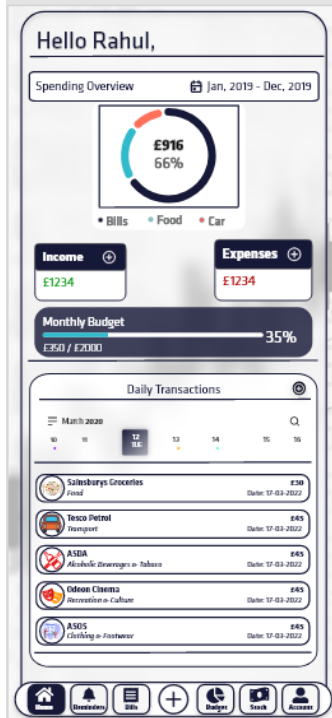
Appendix B-(Web Wireframe)





Appendix-C (Mobile-Mock-up)





Add Transactions-Expenses

Add New Transaction

Income **Expenses**

Name:
Taxes Paid

Category:
Transport

Date:
22-01-2022

Amount:
£70

ADD Cancel

Bottom navigation bar: Home, Dashboard, Add, Budget, Spend, Profile

Action Complete

Hello Rahul,

Spending Overview Jan, 2019 - Dec, 2019

SUCCESS!
[ACTION] has been executed successfully!

Dismiss

Subscriptions Expenses	£30
Taxes Paid	£70
Alcohol	£40
Alcoholic Beverages & Tobacco	£40
Alcohol Expenses	£40
Alcohol Expenses	£40
Alcohol Expenses	£40

Bottom navigation bar: Home, Dashboard, Add, Budget, Spend, Profile

Add Transactions – Income

Add New Transaction

Income **Expenses**

Name:
Work Salary

Category:
Income

Date:
22-01-2022

Amount:
£70

ADD Cancel

Bottom navigation bar: Home, Dashboard, Add, Budget, Spend, Profile

Add New Bill Reminder

Add New Reminder

Name:
Work Salary

Category:
Alcoholic Beverages & Tobacco

Due Date:
Monthly

Amount:
£100

ADD Cancel

Bottom navigation bar: Home, Dashboard, Add, Budget, Spend, Profile

Add New Budget

Add New Budget

Category:
Alcoholic Beverages & Tobacco

Duration:
Monthly

Amount:
£100

ADD Cancel

Bottom navigation bar: Home, Dashboard, Add, Budget, Spend, Profile

Appendix D-(Web-Mock-Up)

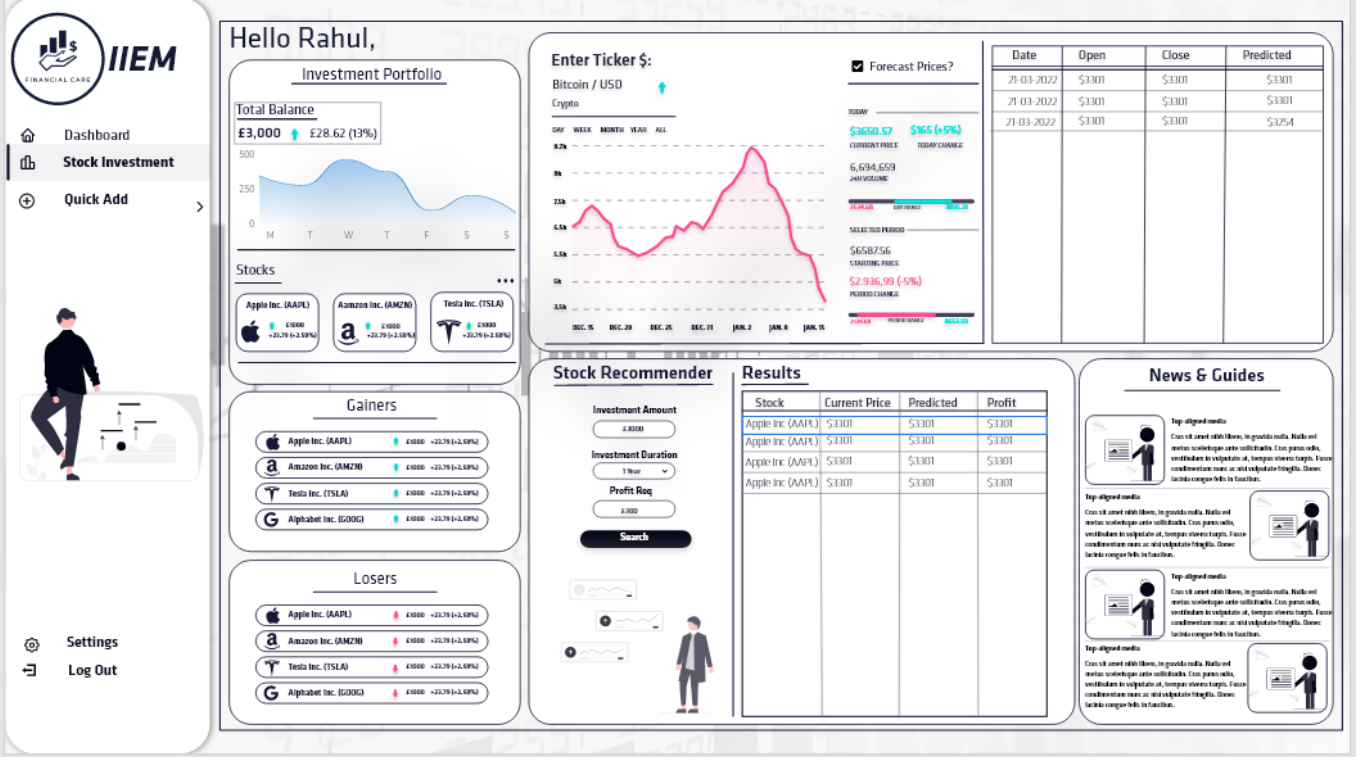
Welcome-Register -Dark

Welcome-Login-Dark

Welcome-Register

Welcome-Login-Dark

Dashboard



Add New Transaction-Expense

Add New Transaction

Income Expense

Name: [Text Field]

Category: [Dropdown]

Date: 21-03-2022

Amount: £100

ADD Cancel

Add New Budget

Add New Budget

Category: [Dropdown]

Duration: [Dropdown]

Amount: £100

ADD Cancel

Add New Transaction-Income

Add New Transaction

Income Expense

Name: [Text Field]

Category: [Dropdown]

Date: 21-03-2022

Amount: £100

ADD Cancel

Add New Bill Reminder

Add New Reminder

Name: [Text Field]

Category: [Dropdown]

Due Date: [Dropdown]

Amount: £100

ADD Cancel



CATEGORIES

RECREATION AND CULTURE 🎭

TRANSPORT 🚗

FOOD AND NON-ALCOHOLIC BEVERAGES 🍽️

MISCELLANEOUS GOODS AND SERVICES 🛒

EDUCATIONAL SERVICES 🎓

FURNISHINGS, HOUSEHOLD EQUIPMENT AND ROUTINE HOUSEHOLD MAINTENANCE 🏠

HEALTH ❤️

CLOTHING AND FOOTWEAR 👕

HOUSING AND UTILITIES 🏠

ALCOHOLIC BEVERAGES AND TOBACCO 🚭

COMMUNICATION 📞

INCOME 💰