PERSONAL ASSET MANAGEMENT BASED ON RISK RETURN INDEX

A Project Report

Submitted by

Gaurav Sherwani	**111608025**
Murtaza Laljiwala	**111608043**
Ayeshabi W Tigdikar	**111608076**

in partial fulfillment for the award of the degree

of

B.Tech Information Technology

Under the guidance of

Prof. Rahul Adhao

College of Engineering, Pune

AND

Prof. Nikita Mahajan

College of Engineering, Pune

DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE-5

May, 2020

DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE

CERTIFICATE

Certified that this project, titled "PERSONAL ASSET MANAGEMENT BASED ON RISK RETURN INDEX" has been successfully completed by

Gaurav Sherwani	**111608025**
Murtaza Laljiwala	**111608043**
Ayeshabi W Tigdikar	**111608076**

and is approved for the partial fulfillment of the requirements for the degree of "B.Tech. Information Technology".

SIGNATURE

Prof. Rahul Adhao and Prof. Nikita Mahajan
Project Guides
Department of Computer Engineering and Information Technology, College of Engineering Pune, Shivajinagar, Pune - 5.

SIGNATURE

Prof. Vahida Attar
Head
Department of Computer Engineering and Information Technology, College of Engineering Pune, Shivajinagar, Pune - 5.

Abstract

Finance is an integral part of everyone's life and financial principles are based on pure and simple common sense. For one's financial well-being, management of personal finances is very much essential. Money management aims to give the power and the knowledge to take control of the money. Keeping track of personal expenses, personal debt and savings helps in calculation of net worth. Personal Finance for the mass has for long been ignored by many investment banks. Moreover, the Banking strategies used by investment banks are very expensive as they require individual portfolio managers which can't be afforded by a common man. Hence, we're automating the process of investing by creating an investment software application. This application can be used as a one stop shop for maintaining a user's personal finances. It eliminates intervention by humans on the Bank's side and is a direct link between financial instruments and the person. The Key highlight is creating our proprietary Risk Return Index for the calculations and calibration.

Keywords: Personal Finance, Stocks, Risk Return Index, Investment.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Everyone sets financial goals throughout their life once they are drawing an income. These financial goals may be short or long term (retirement, college education are some to name). An overview of it would be to manage, save and invest capital in a diverse array of financial instruments readily available. Usually these services are devised for High Net-Worth Individuals or Companies with surplus cash in their bank accounts.

In this project, asset allocation will be replicated for common individuals by creating a Robo Asset Manager. Since it is based on a model of extensive research it is scalable to any individual. This also eradicates the need of a human asset manager that all individuals can't afford. To attain an expertise in personal finance, it is essential to be financially literate first but with a new user friendly Robo Asset Manager guiding you at your fingertips, investing has never been easier.

Chapter 2

Literature Survey

Financial Independence is the ultimate goal of an individual. Because it would mean finally not having to work for generating income. It is the status of having enough income being generated to pay for your living expenses for the rest of one's life without having to be dependent on others. Passive income is the income you earn without having a fulltime job which means you earn money from other sources like existing investments. But to generate such an income one should have enough savings in his/her pocket. Savings can mean a lot of different things to different people, to some it can mean to physically save the money in your bank account, to others it means to invest in real estate property, but to economists it means to consume less in the present in order to consume more in the future. Economists firmly use their current savings for generating future passive income by forming a solid capital.

In economics, capital formation is an addition of capital stock to one's portfolio. Capital formation is crucial for one's economic growth because it leads to increased productivity and the ability to innovate and ultimately leads to financial independence. Investing in productivity is crucial for financial growth. It leads to increased gain and the ability to invest in yourself and your family.

Capital formation by allocating assets wisely into financial products like stocks, commodities, fixed deposits, mutual funds, options and other alternative investments is the right path to gain financial independence but one should be financially literate to make such decisions. There are firms which provide these services where they advise people to make decisions which are fruitful in future and beneficial to one's portfolio. But these services are so exorbitant that the working class who want to invest in today's stock have funds lower than the fees charged by these firms. The major cost goes into the human resources that provide personal advisers to their customers. Taking it online and building a robo-adviser is the way to go in the future since certain algorithms can be designed and inculcated into the software. And the cost of building a software for the above purpose is considerably low as it requires a minimal investment in the beginning. A few key terms and definitions are mentioned below before we dive into the crux of our survey.

Asset Management: Asset management is a methodology by which one can mitigate risk to an extent by dividing assets into various financial instruments such as stocks, commodities, FDs', Real Estate and derivatives. The rate of Risk and Return varies for each asset, time is also an important factor determining the above. It is one of the best ways to protect anyone from major losses.

Risk/Return: Every Individual wants to maximise their savings, but doing it in one particular way, for example stocks isn't the right way to go about it. There should be a diversified portfolio that not only maximises capital gain over time but also hedges against your losses by investing in safe havens like sovereign bonds, etc.

Thumb rule for investing in Stocks: The golden rule for investing in stocks as followed by experts of finance and economics is that a person subtract one's

age from 100. Let's say you are 25 years old, you should allocate 75% of your money in stocks and 25% into real estate or cash. Due to increase in average age this rule has been amended to (110 - age) or (120 - age).

Goals (Asset Allocation Plan): Depending on the investors goals (Retirement, Children's education, etc.) their funds would require different investment models which will be taken care of by our proprietary investment model based on a questionnaire which will determine the risk an investor is willing to take based on his/her goals in the future.

Time: Looking at return on investment is one aspect however maturity date is equally important for reallocation of assets or liquidation. To sum it up, Asset allocation is not a one-time event, it's a life-long process of progression and fine-tuning.

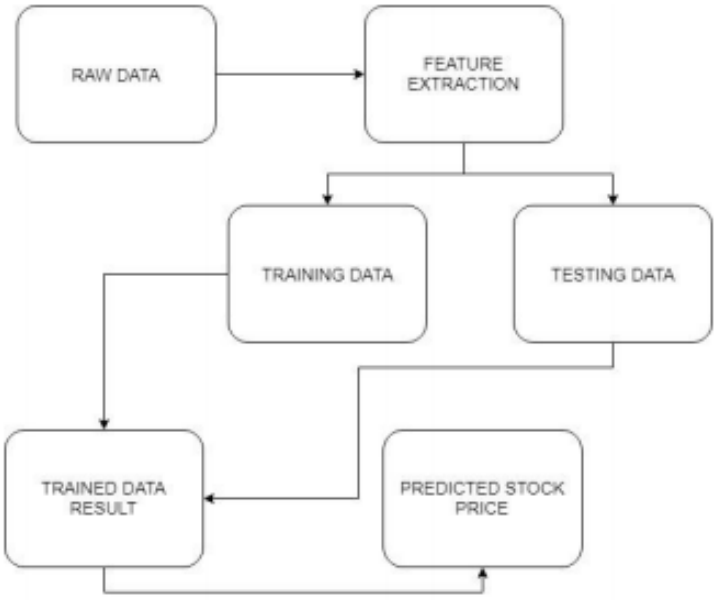

Figure 2.1: An abstract view of Stock and Commodity predictions using Neural Networks

2.1 LSTM

Table 1: Comparison of RNN and LSTM	
Conventional RNN's	**LSTM**
RNN's (Recurrent Neural Networks) lack the ability to retain long term memory. So RNN's may leave out long term memory	LSTM (Long short-term memory) on the other hand considers both long term and short term memory hence enhancing the time series model
Back Propagation problem: As the neural network starts updating weights the gradient that back propagates over time reduces to a negligible amount hence reducing the contribution of old data	LSTM's are created to overcome the Back-propagation flaw with RNN's. These Neural Networks have additional gates designed to control the way information is flowing through a neuron

Table 2.1: Comparison of RNN and LSTM

Apart from a detailed comparison between RNN's and LSTM in Table 2.1 we also analyzed KNN, ARIMA, Linear Regression for stock price modelling but found out that they were not even close to the accuracy LSTM had (denoted in the bibliography section - research papers marked 1-4). Based on the above pros of LSTM we decided to use this for our stock price prediction model.

Steps for implementing LSTM:

1. Combination of previous state and current input.
2. This is then passed on to the forget layer which removes non relevant data.

3. Now a layer is generated which holds possible values for a cell state (Candidate Layer)

4. This layer is then added to the new cell state.

5. The new cell state is calculated using the above vectors of the 3 layers above and the hidden cell state.

6. Finally, this gives some output.

7. Multiplication of this output and new cell state gives us a new hidden state which is updated to our model

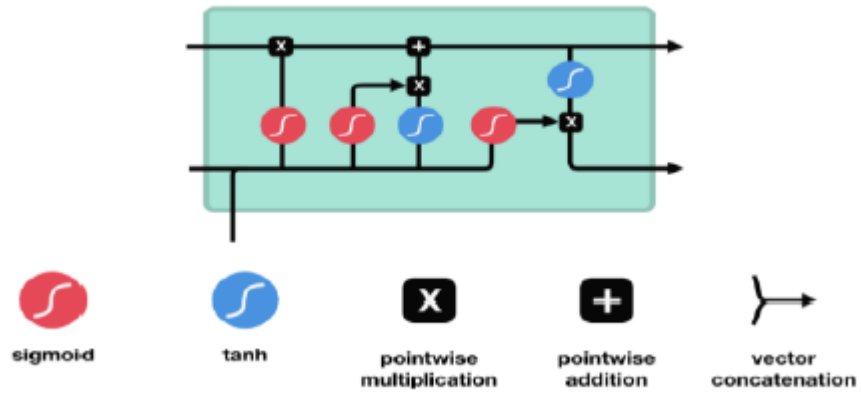

Figure 2.2: Working of LSTM

2.2 Stocks

A stock is a general term used to describe the ownership certificates of any company. A share on the other hand, refers to the stock certificate of a particular company. Holding a particular company's share makes you a shareholder. There are various types of stocks, however common stocks that are ETF (Exchange Traded Funds) are most accessible to a common user. Any

Demat (Short for dematerialsed) account in a registered bank lets users hold financial securities like Equity and Debt in a digital form. Stocks are one of the most volatile financial instruments and need to be invested with in depth knowledge to gain profits. The LSTM AI model gives accurate stock predictions of the near future.

2.3 Commodities

2.3.1 Gold

Gold is a safe haven and has been used to hedge since a very long time. Common people do not think Gold as an investible product, although Indians have an affinity towards Gold. Few countries even have their bank reserves linked to Gold. Few analysts say Gold moves opposite to the market. Everyone falls back to Gold when market trends are on a downturn in general. Gold futures are also tracked very diligently when the believe a market downfall is near.

2.3.2 Oil

Oil is the most important thing in day to day life, life would be un-imaginable. Investors are well aware that Oil has a big effect on stock market. Oil can go up only till the stock market bubble bursts or is on the verge to. Oil has recently affected the stock prices of transportation companies, this will reduce consumer spending. Once Oil rallies oil and energy stocks become defensive.

There are three main factors that commodities traders look at when developing the bids that create oil prices. First, the current supply in terms of output. Second, access to future supply. That depends on oil reserves. It includes what's available in U.S. refineries as well as in the Strategic Petroleum

Reserves. These reserves can be accessed very easily to increase oil supply if prices get too high. Third, oil demand, particularly from the United States. These estimates are provided monthly by the Energy Information Agency. Potential world crises in oil-producing countries dramatically increase oil prices. That's because traders worry the crisis will limit supply.

2.4 Fixed Deposit

A Fixed Deposit (FD) is a financial instrument used for investment which is offered by the banking and non-banking financial companies. These companies provide investors a high rate of interest as compared to a regular savings account. The major drawback here is that the money cannot be withdrawn before the maturity date unless a penalty amount is paid which is usually less than 1 percent. Also, some banks do allow investors to withdraw money with zero penalty charges. Fixed deposits are considered to be one the safest option to invest money for a long period among a large section of investors in India. The returns on an FD depends on various factors like age of investor, tenure of account, type of FD account etc. The rate of interest is usually ranging from 4.5% per annum to 8% per annum which would depend on the tenure (mostly 10 - 12 years). There are two kinds of FD's. One is the Bank FD's and the other one being Corporate FD. Company Fixed Deposit are very likely to be of high risk also known as 'Default Risk'. The difference between the two kinds are shown in Table 2.2.

Table 2: Bank Fixed Deposit VS Corporate Fixed Deposit		
	Bank FD	**Corporate FD**
Offered by	Bank	Companies
Rate of Interest	Average	High
Tenure	Month/s to year/s	6 months to 3 years
Risk	Low	High

Table 2.2: Bank Fixed Deposit VS Corporate Fixed Deposit

After the budget is declared by the Government of India, the interest rate of the fixed deposit is revised every year based on repo rate else it changes on a negligible rate.

2.5 Research - Asset Management

Families employ wealth management services not merely to monitor their money, but to employ and invest their wealth. The Wealth Management Services can be used by High Net-worth Individuals (HNWIs), small beginners/entrepreneurs and other individuals who want their hard-earned income to be professionally managed. Most of these renowned services are quite personal and portfolio management is the most common one among them. With the increasing use of technology current wealth management clients are interested to use mobile applications to access and manage their accounts. Wealth Managers need to build robust and secure IT infrastructure to safeguard their investor's wealth. Every individual that thinks wisely and logically uses the available information to reach his/her decisions. It has been stated that an investor does consider the consequences and benefits of their actions before making any decisions about their conduct or behaviour related to any investment they make. Hence it results into a conceptual theory known as "a theory

of reasoned action". It basically refers to a framework build from behavioural intention and fundamentally describes the relationship between attitude and behaviour of an individual. According to this theory, an individual's intention is derived from a function of two essential attributes. The nature of the first attribute is personal to the individual and the second attribute reflects the influence of the society on that individual. This evidently concludes that an attitude and social pressure forms a person's intention. Attitude, a personal factor, determines whether a certain behaviour should be exhibited or not. The second determinant referred to as subjective norm, determines the influence of social pressures on an investor's behaviour. This theory provides us with one of the most influential techniques to effectively predict human behaviour and the related behavioural characteristics. Using this theory, we provide the user with a basic questionnaire in order to read his mind virtually and calculate the amount of risk he is willing to take and then categorise him into 3 broad categories namely Conservative, Moderately Aggressive and Aggressive. There are three generic approaches for asset management that are represented in Fig. 2.3, 2.4 and 2.5:

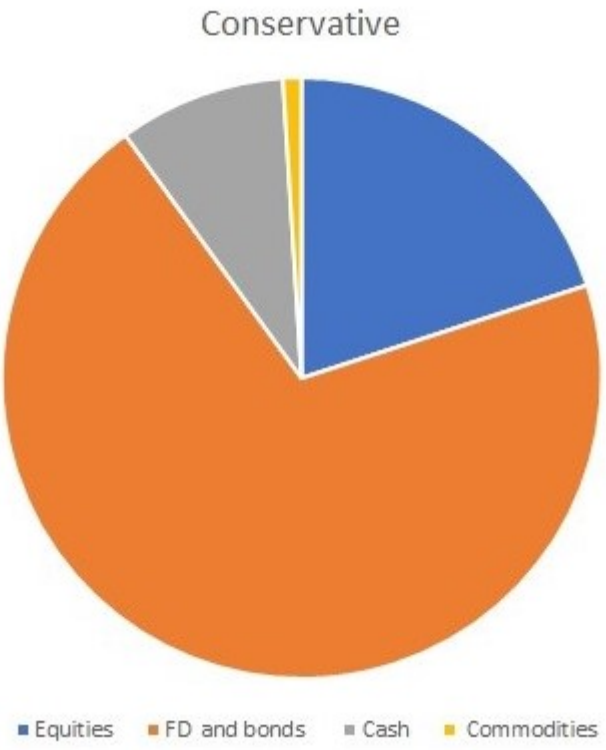

Figure 2.3: Conservative - Low Risk

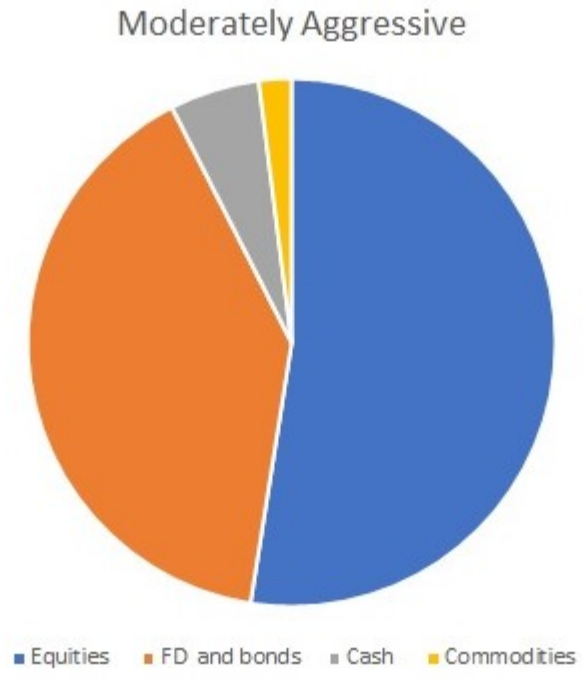

Figure 2.4: Moderately Aggressive - Balanced

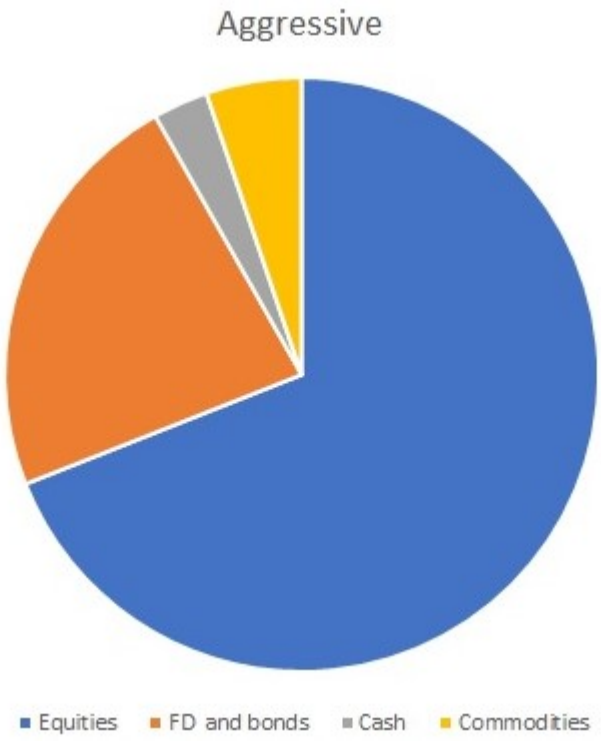

Figure 2.5: Aggressive - High Risk

2.6 Conclusion

To help an individual gain financial independence, we can conclude that there are many financial instruments. Making sure that the investor is a common man we will have to combine stocks, commodities (Oil and Gold) and FD's and allocate a portfolio. This portfolio will have liquid cash as well. A psychological questionnaire and algorithm would categorise the investor as Conservative, Moderately Aggressive or Aggressive. This categorisation will give the portfolio breakup as a percentage of Stocks, Commodities, FD's and Cash. Comparing various neural networks we will use LSTM for stocks and commodities prediction which will give on insight on whether to invest and how much to invest. To maximise profits we will use a proprietary greedy algorithm based on the LSTM predictions. The FD rate will be dynamically fetched to give accurate everyday rates.

Chapter 3

Problem Statement

Investment strategies have been studied for long, but no one has focused on creating a robo advisor for the common man. Existing software used for asset allocation are sometimes so exorbitant that the fee required to get started with such a software is more than the amount that a working middle class can manage to invest. Also, a huge portion of this cost goes into the human resource allocated for such a process. We are optimizing capital formation and gain over time that is accessible not only to High NetWorth Individuals but also to middle class families. Technological advancements have helped us in various facets of finance and other sectors. The proposal is to automate this process and use robo-advisors which does quantitative calculations based on different machine learning models and various other algorithmic simulations.

Chapter 4

System Requirement Specifications

4.1 Introduction

This document provides an overview of the SRS with the purpose, intended audience use, scope, user needs, functional and non-functional requirements of the project. The aim is to analyse our project Personal Asset Manager by clearly defining our problem statement in detail.

4.1.1 Purpose

The purpose of this document is to analyse and study the ideas which have come up to define our system and its requirements with respect to the users. In this project, the purpose is to maximize savings while minimizing the risk by managing money for saving and investing. Based on the risk/return index obtained from the questionnare the software will rebalance the different monetary assets in a portfolio.

4.1.2 Document Conventions

The font used in the document is "Times New Roman" with size "14". Topic headings have been typed in bold.

4.1.3 Intended Audience and Reading Suggestions

The audience we are targeting are everyday Working Individuals, Retired Individuals, Financial planners and Investors.

4.1.4 Product Scope

Users can use this application so that they can invest their savings in different financial instruments. The report focuses on allocating the assets to an individual based on a questionnaire. The project can be further implemented in investment banks, retail banks and e-wallet service providers.

4.1.5 References

(Refer Bibliography - Page 56)

4.2 Overall Description

4.2.1 Product Perspective

The user can input his/her savings after which user will be asked to attempt a questionnaire.Based on which our application will categorize the individual into one of the three main categories - Conservative, Moderately Aggressive and Aggressive. Depending on the category, a breakup percentage for each asset will be displayed.

4.2.2 Product Functions

The application will perform functions like:

1. Helps an individual put his savings into a portfolio to generate passive income.

2. Determines risk the person is willing to take based on a questionnaire.
3. Categorizes him into one of the three broad categories of Risk
    - Conservative
    - Moderately Aggressive
    - Aggressive
4. Asset Allocation on various assets are Stocks, Fixed Income, Commodities and Cash.

4.2.3 User Classes and Characteristics

The use of the project is to help an individual make the most out of his income by investing in various assets like Stocks, Cash, Commodities, FD's. This SRS is intended mainly for developers, project managers, users and testers. Index page is for everyone.

4.2.4 Operating Environment

The application will be operated on a web server based on Tkinter. Any web browser can be used to run this application.

4.2.5 Design and Implementation Constraints

Implementing neural networks requires a lot of computational memory. Hence to cater to this we will use cloud computational services (Google Collab) and an Nvidia 1050 GeForce GPU. We have used models based on the literature reviews. These don't consider news-based sentiment analysis, so sometimes (for example news based on a disastrous events) the predictions maybe inaccurate

4.2.6 User Documentation

The user interface is very intuitive however we will provide a Help page to the user if necessary.

4.2.7 Assumptions and Dependencies

Tensor flow and Yahoo API's for various stocks, and other financial indexes.The user knows how much he will save every month

4.3 System Requirements

4.3.1 User Interfaces

A web based application based on the Tkinter frameworks will be developed.

4.3.2 Hardware Interfaces

Deep Learning requires computers with high computational ability so we will use Google Collab (Compute Engine) to train our model which includes Deep Learning AMI (Ubuntu) with 4 vCPUs, 1 NVIDIA Tesla K80. Other Requirements are 50GB of disk space. TensorFlow integrates GPU's for training when required. We also have our own GPU (Nvidia GeForce 1050) for training offline.

4.3.3 Software Interfaces

A web based interface will be developed using Tkinter framework in Python.

4.3.4 Communication Interfaces

The communication between different segments of the system i.e (interface and deep learning) is very essential as they are contingent on each other.

Communication between the deep learning model and web interface is established using the tkinter framework in Python.

4.4 System Features

4.4.1 Input

1. Login Credentials: Authenticate and validate users
2. Behaviour categorisation based on psychological questionnaire
3. Savings
4. Age and Savings Maturity Date

4.4.2 Output

1. One of the three risk categorisation
2. Asset allocation
3. Break-up of individual components (Stocks, FDs and Commodities)

4.5 Other Non-Functional Requirements

4.5.1 Performance Requirements

1. Standard PC
2. Internet with good connectivity
3. NVIDIA GEFORCE GTX 1050

4.5.2 Safety Requirements

Not Applicable

4.5.3 Security Requirements

1. Data will be kept confidential and won't be shared with third party organisation
2. Data that will be used for the purpose of allocating assets will only be shared with our clients.

4.5.4 Software Quality attributes

1. Adaptability: Can be incorporated into any web browser.
2. Flexibility: There is a fixed probability of outcome, so the software is inflexible.
3. Maintainability: Debugging can be done easily. Modification and extension of functionality is possible but with limitations.
4. Reusability: A Modular programming approach ensures that it is reusable.
5. Testability: Program is modular and structured which makes testing easy.

4.5.5 Business Rules

All Individuals can use the web application, since it is user friendly. No specific prerequisite knowledge is required to use the application.

Chapter 5

System Design and Architecture

Fig 5.1 shows the overall system working. The database will consist our AI model predictions for stocks and commodities as well as the dynamically fetched interest rates for FD. The Graphical User interface used will be based on Tkinter and this is how the user will interact with the overall system.

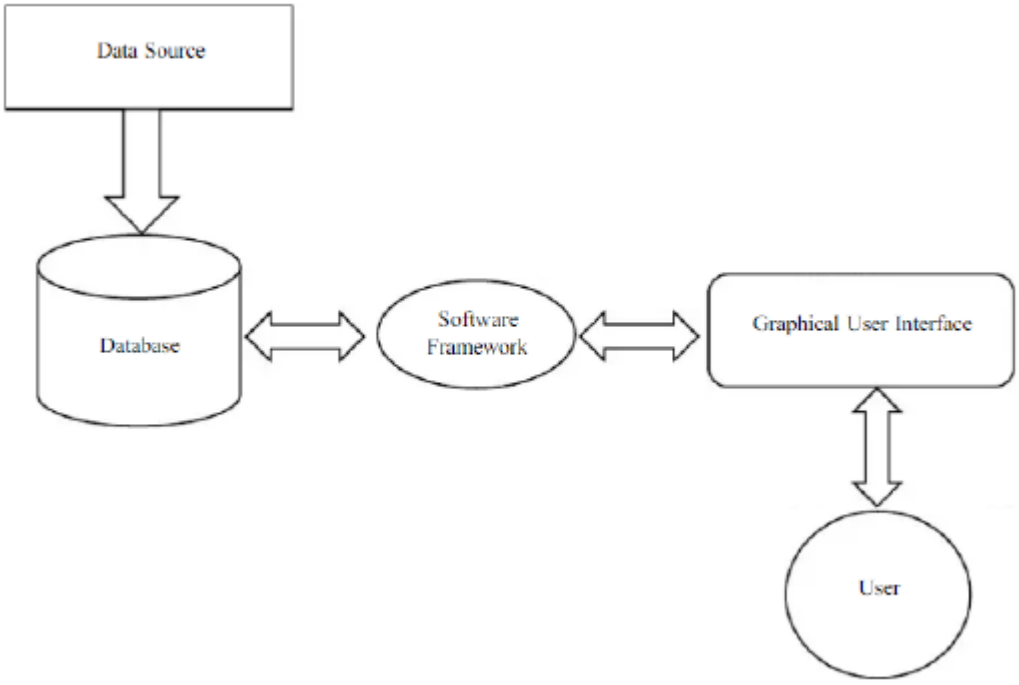

Figure 5.1: System Design

Fig 5.2 shows the steps the user will have to go through in order to get the required portfolio distribution of funds.

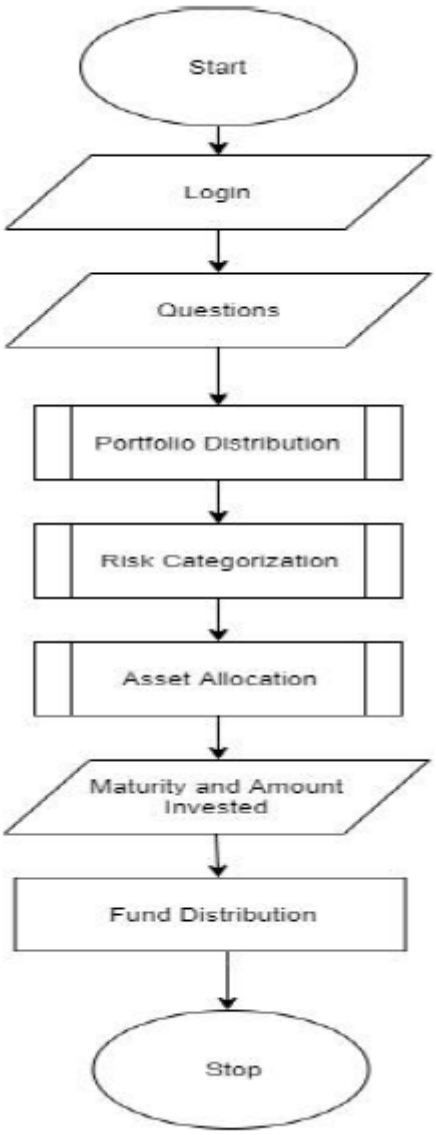

Figure 5.2: System Workflow

Fig 5.3 shows the financial instruments for investment purpose along with the algorithm used.

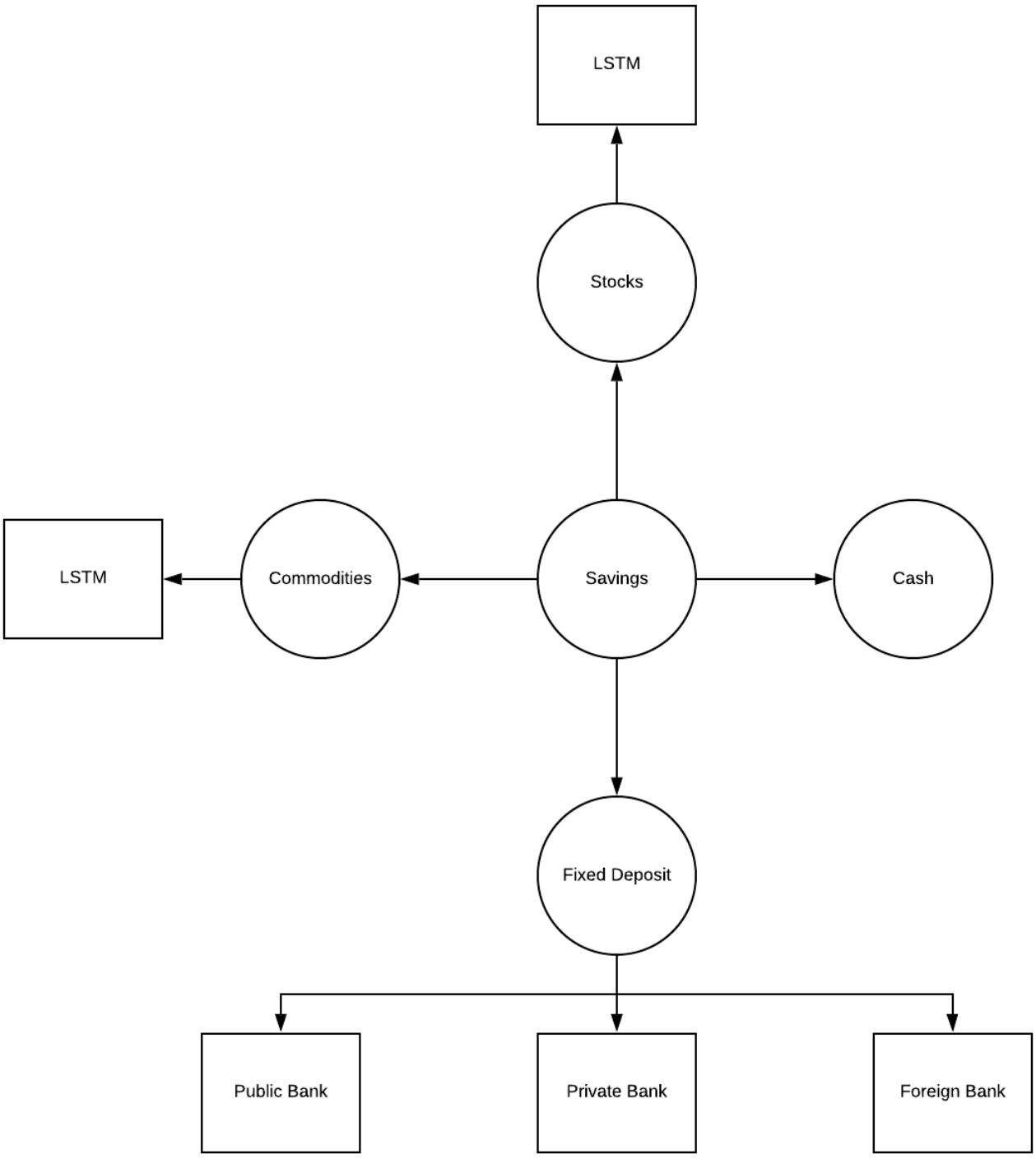

Figure 5.3: Overview

Chapter 6

Code and Implementation

6.1 Functions for Questionnaire

```
def bnext():

  global windows
  windows = Toplevel(root)
  windows.title("Question 1")
  windows.geometry("900x200")
  root.withdraw()

  lbl1 = Label(windows,text = q[0],font = ('arial',10,'bold')).pack(side = TOP)
  cb1 = Radiobutton(windows, text=a0[0], value=4,variable = v0).pack(side=BOTTOM)
  cb2 = Radiobutton(windows, text=a0[1], value=3,variable = v0).pack(side=BOTTOM)
  cb3 = Radiobutton(windows, text=a0[2], value=2,variable = v0).pack(side=BOTTOM)
  cb4 = Radiobutton(windows, text=a0[3], value=1,variable = v0).pack(side=BOTTOM)
  btn1 = Button(windows,text = "next",font = ('arial',12,'bold'),fg = 'blue', command = bnext2).pack(side
= RIGHT)
  btn2 = Button(windows,text = "back",font = ('arial',12,'bold'),fg = 'blue', command = bback).pack(side =
 LEFT)
  windows.mainloop()
```

```
def bnext2():

  global windows3
  windows3 = Toplevel(windows)
  windows3.title("Question 2")
  windows3.geometry("900x200")
  windows.withdraw()

  lbl2 = Label(windows3, text=q[1], font=('arial', 10, 'bold')).pack(side=TOP)
  cb5 = Radiobutton(windows3, text=a1[0], value=1,variable = v1).pack(side=BOTTOM)
  cb6 = Radiobutton(windows3, text=a1[1], value=2,variable = v1).pack(side=BOTTOM)
  cb7 = Radiobutton(windows3, text=a1[2], value=3,variable = v1).pack(side=BOTTOM)
  cb8 = Radiobutton(windows3, text=a1[3], value=4,variable = v1).pack(side=BOTTOM)

  btn3 = Button(windows3,text = "next",font = ('arial',12,'bold'),fg = 'blue',command = bnext3).pack(side
=RIGHT)
  btn4 = Button(windows3,text = "back",font = ('arial',12,'bold'),fg = 'blue',command = bback2).pack(side
=LEFT)
  windows3.mainloop()
```

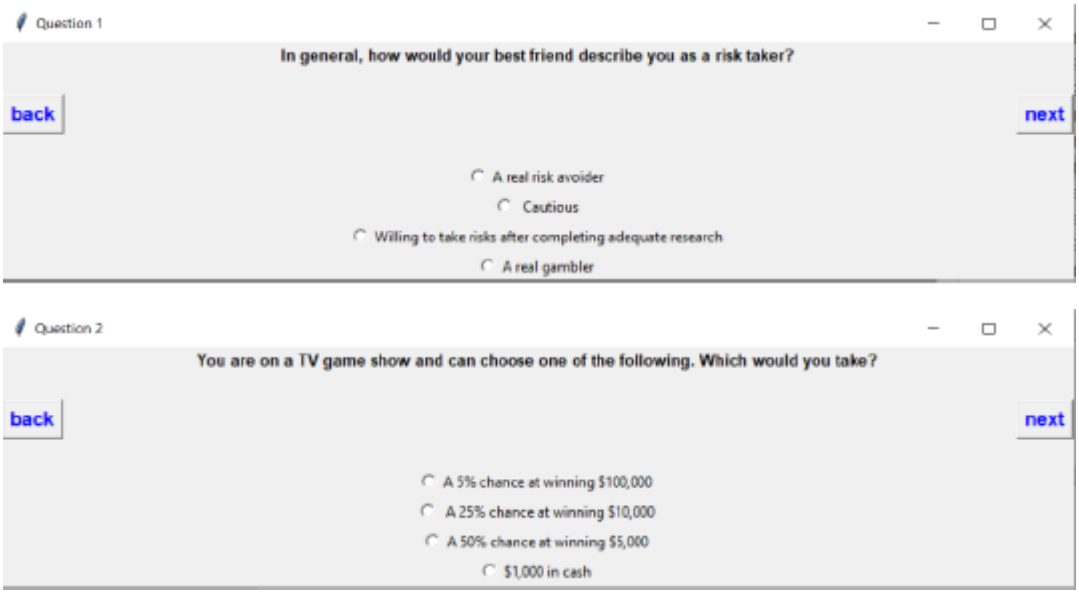

Figure 6.1: Output : Questionnaire

6.2 Stocks and Commodities Prediction with LSTM

```
## Import Modules
from tensorflow.keras.layers import Input, LSTM, GRU, SimpleRNN, Dense, GlobalMaxPool1D, Dropo
ut
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import SGD, Adam
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler

## Stock Data Fetching
from pandas_datareader import data as pdr

#Yahoo API
import yfinance as yf
yf.pdr_override()

stock = "HDFCNIFETF.NS"
df = pdr.get_data_yahoo(stock, start="2018-09-01", end="2020-02-21")
```

```
## Pre processing
# Remove irrelevant fields
series = df['Close'].values.reshape(-1, 1)

# Normalize the data
scaler = StandardScaler()
scaler.fit(series[:len(series) // 2])

#Check Results (Debug)
print(series)
series = scaler.transform(series).flatten()
print(series)

## Building Actual Dataset for the Model

# Uses 30 Timestamps to predict next Stock Value
T = 30
D = 1
X = []
Y = []
for t in range(len(series) - T):
  x = series[t:t+T]
  X.append(x)
  y = series[t+T]
```

```
  Y.append(y)

# Data Format : N x T x D
X = np.array(X).reshape(-1, T, 1)
Y = np.array(Y)
N = len(X)
print("X.shape", X.shape, "Y.shape", Y.shape)

## Autoregressive RNN model - (LSTM)

i = Input(shape=(T, 1))

# Layer
x = LSTM(20)(i)
x = Dropout(0.2)(x)

# Layer
x = Dense(10, activation = 'relu')(x)

# Output
x = Dense(1)(x)

# Using MSE
model = Model(i, x)
model.compile(
  loss='mse',
  optimizer=Adam(lr=0.1),
)
```

```
# train the RNN (Test 50, Train 50)
r = model.fit(
  X[:-N//2], Y[:-N//2],
  epochs=80,
  validation_data=(X[-N//2:], Y[-N//2:]),
)

outputs = model.predict(X)
predictions = outputs[:,0]
only_pred = np.array([])
```

```
## Output
# Next 10 Days Prediction (Future)
param = X[-1:]
for i in range(10):
    out = model.predict(param)
    out = out[:,0]
    predictions = np.append(predictions, out)
    only_pred = np.append(only_pred, out)
    tmp = param[0]
    tmp = np.append(tmp, out)
    tmp = np.delete(tmp, [0])
    param = tmp.reshape(-1, T, 1)
    print(param)

# Transforming to actual Stock Values
pred = scaler.inverse_transform(predictions)
print(pred)

Y = scaler.inverse_transform(Y)
predictions = scaler.inverse_transform(predictions)
only_pred = scaler.inverse_transform(only_pred)

# Plotting Graph
plt.plot(Y, label='targets')
plt.plot(predictions, label='predictions')
plt.legend()
plt.show()
```

```
# Plot loss per iteration
import matplotlib.pyplot as plt
plt.plot(r.history['loss'], label='loss')
plt.plot(r.history['val_loss'], label='val_loss')
plt.legend()
plt.show()

# Adding Results to CSV Files
stock = stock + ".csv"
stockall = stock + "_all" + ".csv"
pd.DataFrame(only_pred).to_csv(stock)
pd.DataFrame(predictions).to_csv(stockall)
```

6.3 Greedy Algorithm to Maximize Profits

```
def Max(File, Amount):
    with open(File, newline='') as csvfile:
        data = list(csv.reader(csvfile))

    X = [row[1] for row in data]

    X.pop(0)

    Y = [float(i) for i in X]

    Profit = []
    ProfitPercentage = []
```

```
    def stockBuySell(price, n):
        if (n == 1):
            return
        i = 0
        sum = 0
        total = 0

        while (i < (n - 1)):

            while ((i < (n - 1)) and (price[i + 1] <= price[i])):
                i += 1

            if (i == n - 1):
                break

            buy = i
            i += 1
            while ((i < n) and (price[i] >= price[i - 1])):
                i += 1

            sell = i - 1
            sum = (price[sell] - price[buy])
            total = total + sum

            print("Buy on day: ",buy,"\t",
                "Sell on day: ",sell)
            Profit.append(total)
            #print(price[buy])
            temp =1 + (sum/price[buy])
            ProfitPercentage.append(temp)
```

```
price = Y
n = len(price)

stockBuySell(price, n)
print(Profit)
print(ProfitPercentage)


Final = Amount
for i in range(0, len(ProfitPercentage)):
    Final = Final * (ProfitPercentage[i])

return(Final)
```

```
Buy on day:  0   Sell on day:  3
Buy on day:  4   Sell on day:  9
[0.6451583073499734, 1.2782414742224972]
[1.0062070840116235, 1.006056471112066]

Out[1]: 5263.9659703732195
```

Figure 6.2: Output : Stocks

6.4 Fixed Deposit Calculator

```
def home() :
        PERCENTAGE = 100
        amount  request = int(e1.get())
        subs = str(choices[tkvar.get()])
        results = []
        with open('output.csv') as csvfile :
                readCSV = csv.reader(csvfile, delimiter=',')
                banks = []
                tenures = []
                interests = []
                for row in readCSV:
                        bank = row[0]
                        tenure = row[1]
                        interest = row[2]
                        banks.append(bank)
                        tenures.append(tenure)
                        interests.append(interest)
        prd = [i for i in tenures if subs in i]
        for i in range(1, len(banks)) :
                if tenures[i] in prd :
                        b  n = banks[i]
                        t  n = tenures[i]
                        perc_val = float(interests[i].rstrip("%"))
                        cal = round((perc  val*amount  request)/ PERCENTAGE, 2)
                        tot = amount  request + cal
                        temp_text = str(b_n) + '(' + str(perc_val) + '% ) = '+ str(tot) + 'Rs (Interest Earned
= '+ str(cal) + 'Rs) \n'
                        results.append(str(temp  text))

        #Display the Results
        mylist = Listbox(master, yscrollcommand = scrollbar.set, width = 70, height = int(80/3) )
        for line in range(len(results)):
                mylist.insert(END, results[line])
                mylist.pack( side = LEFT, fill = BOTH )
                mylist.place(relx = 0.5, rely = 0.5, anchor="center")
                scrollbar.config( command = mylist.yview )
```

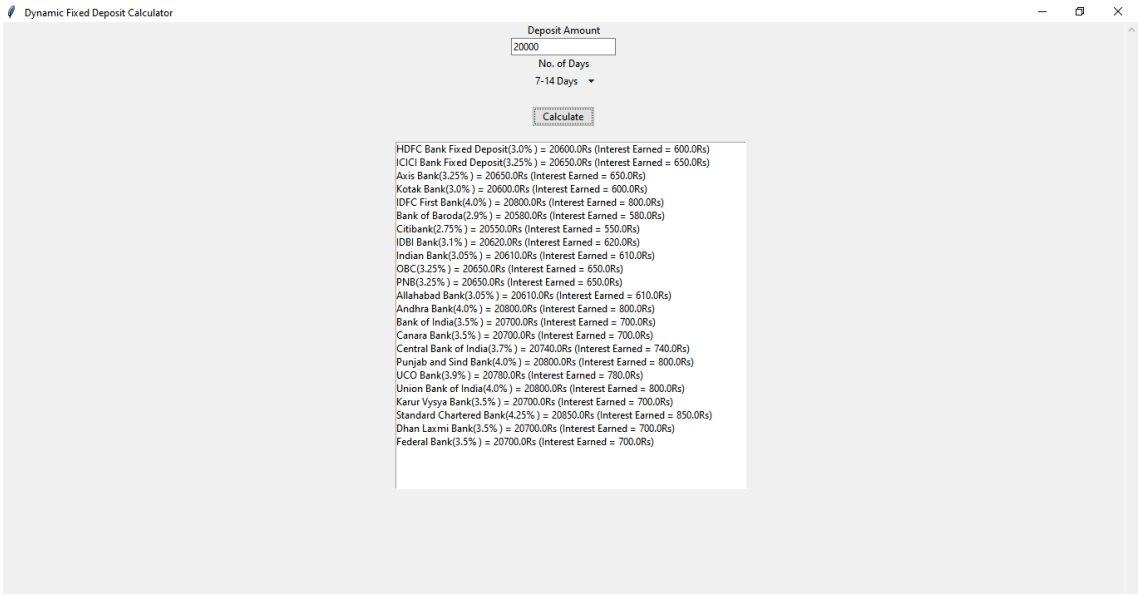

Figure 6.3: Output : Fixed Deposit 1

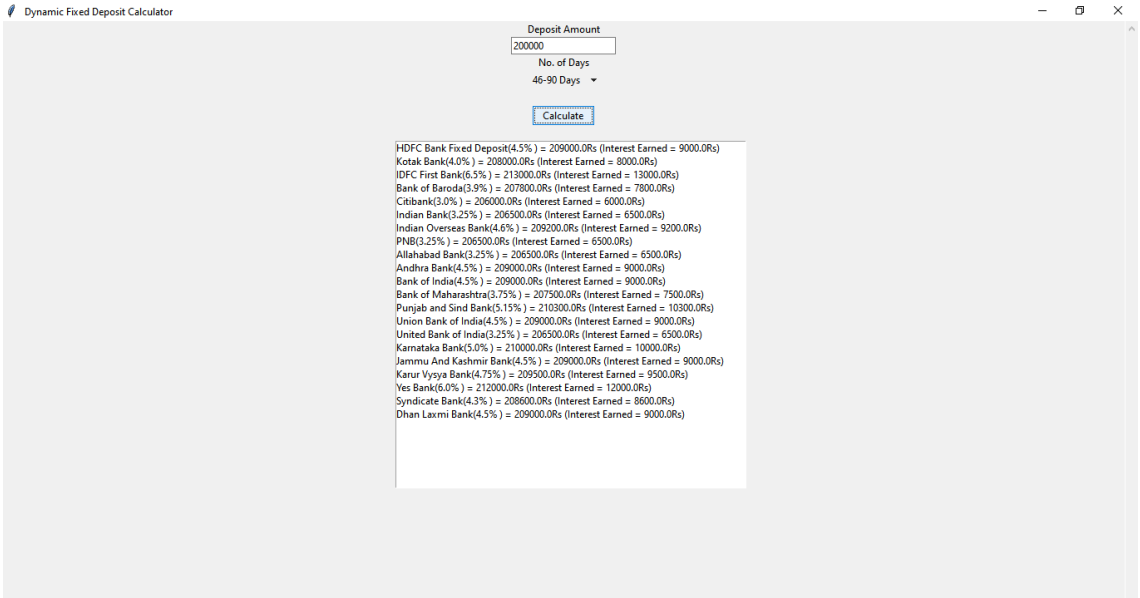

Figure 6.4: Output : Fixed Deposit 2

6.5 Break-up of savings into assets based on the risk class alloted to the user

```
if riskclass == 'Aggressive':
     stockperc = 0.69
     fdperc = 0.23
     cashperc = 0.03
     commperc = 0.05
if riskclass == 'Moderately Aggressive':
     stockperc = 0.52
     fdperc = 0.40
     cashperc = 0.06
     commperc = 0.02
if riskclass == 'Conservative':
     stockperc = 0.2
     fdperc = 0.7
     cashperc = 0.09
     commperc = 0.01
s = Sav.get() #savings
global stockamt
global fdamt
global cash
global commamt
stockamt = s * stockperc
fdamt = s * fdperc
cash = s * cashperc
commamt = s * commperc
bf = Label(windows25,text = "Money into stocks: Rs " + str(stockamt),font = ("arial",16),fg = "blue").place(x = 5, y = 10)
bf1 = Label(windows25,text = "Money into FDs: Rs " + str(fdamt),font = ("arial",16),fg = "blue").place(x = 5, y = 50)
bf2 = Label(windows25,text = "Money as cash: Rs " + str(cash),font = ("arial",16),fg = "blue").place(x = 5, y = 100)
bf3 = Label(windows25,text = "Money into commodities: Rs " + str(commamt),font = ("arial",16),fg = "blue").place(x = 5, y = 150)
bf4 = Label(windows25,text = "Do you wish to change the breakup?" ,font = ("arial",16),fg = "blue").place(x = 5, y = 200)
bnext = Button(windows25, text="Yes", font=('arial', 12, 'bold'), fg='green', command=bnextyes).pack(side=LEFT)
bnext = Button(windows25, text="No", font=('arial', 12, 'bold'), fg='red', command=bnext25).pack(side=RIGHT)
windows25.mainloop()
```

6.6 Predicting returns from stocks and commodities for 5 different elements of each class

```
HDFC =  Max(path + r'/HDFCNIFETF.NS.csv', stockamt)
m50 = Max(path + r'/M50.BO.csv', stockamt)
m100 =  Max(path + r'/M100.BO.csv', stockamt)
fbse = Max(path + r'/SETFBSE100.BO.csv', stockamt)
nifbk = Max(path + r'/SETFNIFBK.NS.csv', stockamt)
global stockmax
stockmax = max(HDFC, m50, m100, fbse, nifbk)

bf1 = Label(windows26,text = "Amount invested in Stocks = " + str(stockamt),font = ("arial",16),fg = "blue").place(x = 5, y = 10)

if HDFC > stockamt:
     bf = Label(windows26,text = "HDFC returns: " + str(HDFC),font = ("arial",16),fg = "green").place(x = 5, y = 40)
else:
    bf = Label(windows26,text = "HDFC returns: Not profitable ",font = ("arial",16),fg = "red").place(x = 5, y = 40)
if m50 > stockamt:
     bf1 = Label(windows26,text = "M50 returns:   " + str(m50),font = ("arial",16),fg = "green").place(x = 5, y = 70)
else:
     bf1 = Label(windows26,text = "M50 returns:   Not profitable",font = ("arial",16),fg = "red").place(x = 5, y = 70)
if m100 > stockamt:
    bf2 = Label(windows26,text = "M100 returns: " + str(m100),font = ("arial",16),fg = "green").place(x = 5, y = 100)
else:
     bf2 = Label(windows26,text = "M100 returns: Not profitable",font = ("arial",16),fg = "red").place(x = 5, y = 100)
if fbse > stockamt:
    bf3 = Label(windows26,text = "SETFBSE returns: " + str(fbse),font = ("arial",16),fg = "green").place(x = 5, y = 130)
else:
    bf3 = Label(windows26,text = "SETFBSE returns: " + str(fbse),font = ("arial",16),fg = "red").place(x = 5, y = 130)
if nifbk > stockamt:
    bf4 = Label(windows26,text = "NIFBK returns: " + str(nifbk),font = ("arial",16),fg = "green").place(x = 5, y = 160)
else:
    bf4 = Label(windows26,text = "NIFBK returns: Not profitable",font = ("arial",16),fg = "red").place(x = 5, y = 160)
```

6.7 Predicting returns from Fixed Deposit for 8 different banks

```
sbi = fd(fdamt,'SBI Fixed Deposit', '7 days to 45 days')
bf = Label(windows27,text = "SBI returns: " + str(sbi),font = ("arial",16),fg = "green").place(x = 5, y = 40)
hdfc = fd(fdamt,'HDFC Bank Fixed Deposit', '7 days')
bf = Label(windows27,text = "HDFC returns: " + str(hdfc),font = ("arial",16),fg = "green").place(x = 5, y = 70)
icici = fd(fdamt,'ICICI Bank Fixed Deposit', '7 days')
bf = Label(windows27,text = "ICICI returns: " + str(icici),font = ("arial",16),fg = "green").place(x = 5, y = 100)
axis = fd(fdamt,'Axis Bank', '7 days')
bf = Label(windows27,text = "Axis Bank returns: " + str(axis),font = ("arial",16),fg = "green").place(x = 5, y = 130)
kotak = fd(fdamt,'Kotak Bank', '7 days')
bf = Label(windows27,text = "kotak returns: " + str(kotak),font = ("arial",16),fg = "green").place(x = 5, y = 160)
bob = fd(fdamt,'Bank of Baroda', '7 days')
bf = Label(windows27,text = "Bank of Baroda returns: " + str(bob),font = ("arial",16),fg = "green").place(x = 5, y = 190)
citi = fd(fdamt,'Citibank', '7 days')
bf = Label(windows27,text = "Citibank returns: " + str(citi),font = ("arial",16),fg = "green").place(x = 5, y = 220)
idfc = fd(fdamt,'IDFC First Bank', '7 days')
bf = Label(windows27,text = "IDFC returns: " + str(idfc),font = ("arial",16),fg = "green").place(x = 5, y = 250)
global stockmax
global commax
global fdmax
global cash
fdmax = max(sbi, hdfc,icici,axis,kotak,bob,citi,idfc)
#bf = Label(windows27,text = "TOTAL returns: Rs" + str(stockmax+commax+fdmax+cash),font = ("arial",16),fg = "green").place(x = 5, y =
280)
```

6.8 Calculation and displaying of final result and the percentage gain from investing using our algorithm

```
global stockmax
global commax
global fdmax
global cash
treturn = stockmax+commax+fdmax+cash
global stockamt
global fdamt
global commamt
s = stockamt + fdamt + cash + commamt
gainperc = ((treturn - s)/s) * 100
bf = Label(windows28,text = "Total returns:  Rs",font = ("arial",32),fg = "blue").place(x = 5, y = 10)
bf = Label(windows28,text = str(treturn),font = ("arial",32),fg = "green").place(x = 350, y = 10)
bf = Label(windows28,text = "Return Percent: ",font = ("arial",32),fg = "blue").place(x = 5, y = 100)
bf = Label(windows28,text =  str(gainperc) + "%",font = ("arial",32),fg = "green").place(x = 350, y = 100)
bnext = Button(windows28, text="Quit", font=('arial', 12, 'bold'), fg='blue', command=quit).pack(side=BOTTOM)
windows28.mainloop()
```

Chapter 7

Results

7.1 Questionnaire

The risk analysis score is one of the Preliminary Result as it is the magnitude of the risk the user of the software is willing to take. It is a score given out of 69 based on the answer given to the questionnaire asked. As we can see in Figure 7.1, the user has received a score of 58.

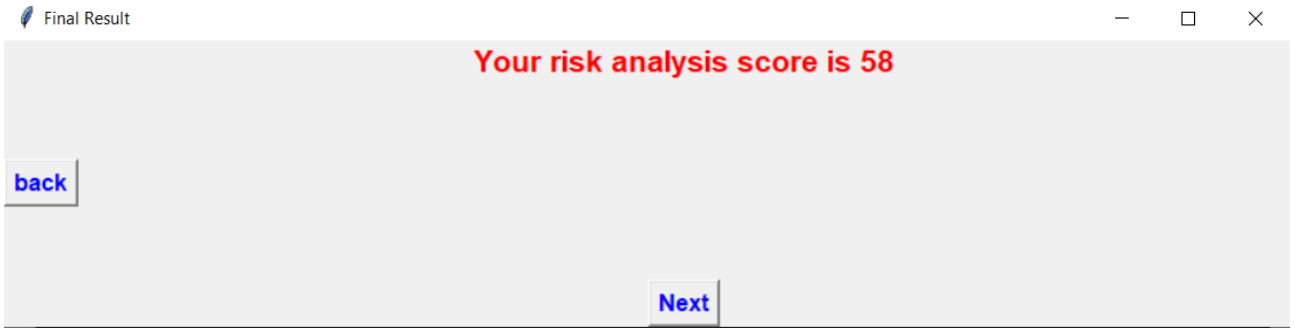

Figure 7.1: Risk Analysis Score

Also, the risk class, allocated using the risk analysis score, is one of the preliminary result as it classifies the user into 3 different classes. The risk classes are conservative, moderately aggressive and aggressive. Afterwards, when the user enters its savings as input, using his savings and the risk class, a pie chart is displayed which shows the breakup of the portfolio into different classes.

As seen in Figure 7.2, for the class 'Aggressive', the split up is done as 69 percent is given to equities (stocks indices), 23 percent is put into FDs, 5 percent into commodities and just 3 percent is kept as cash.

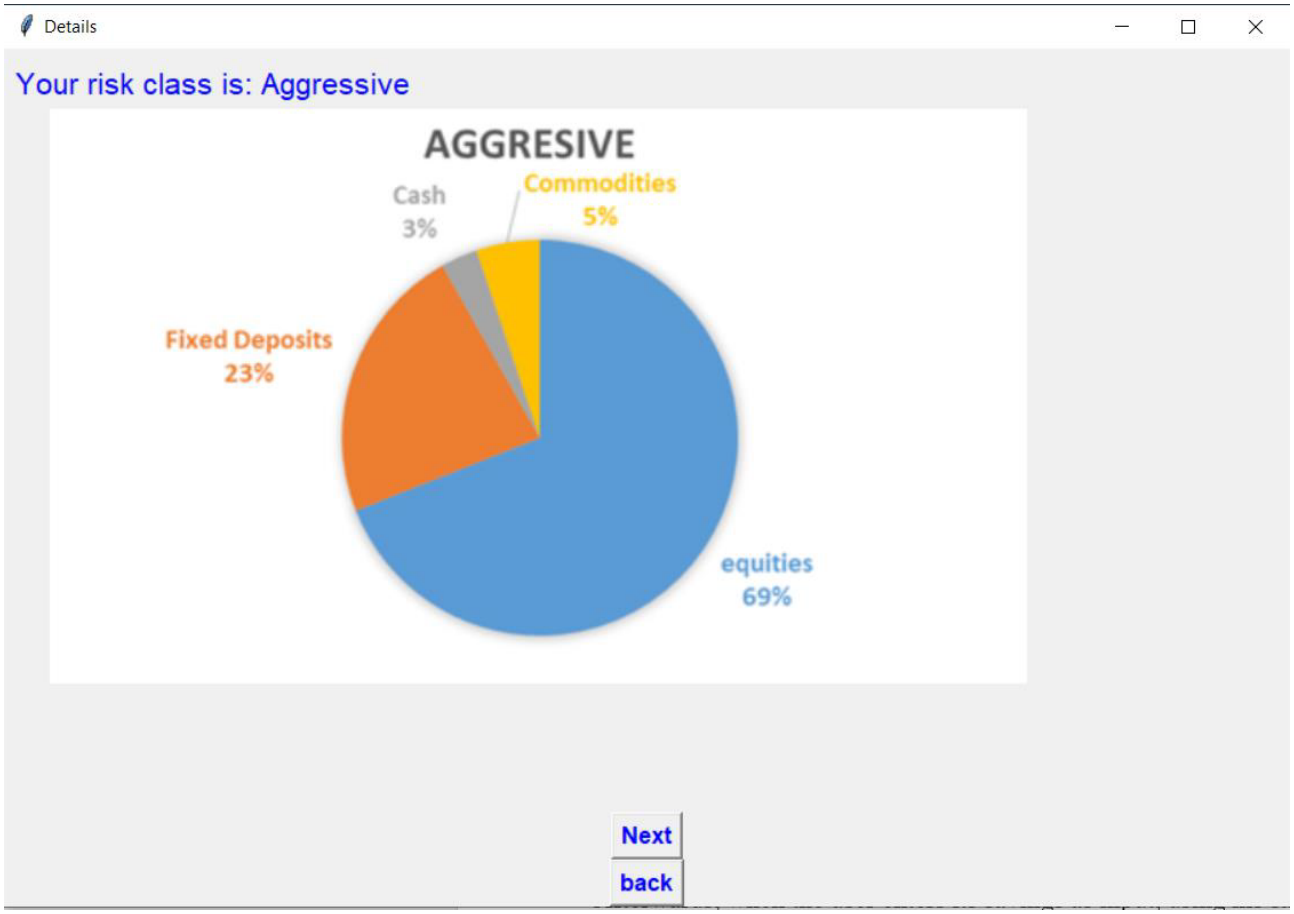

Figure 7.2: Risk Class

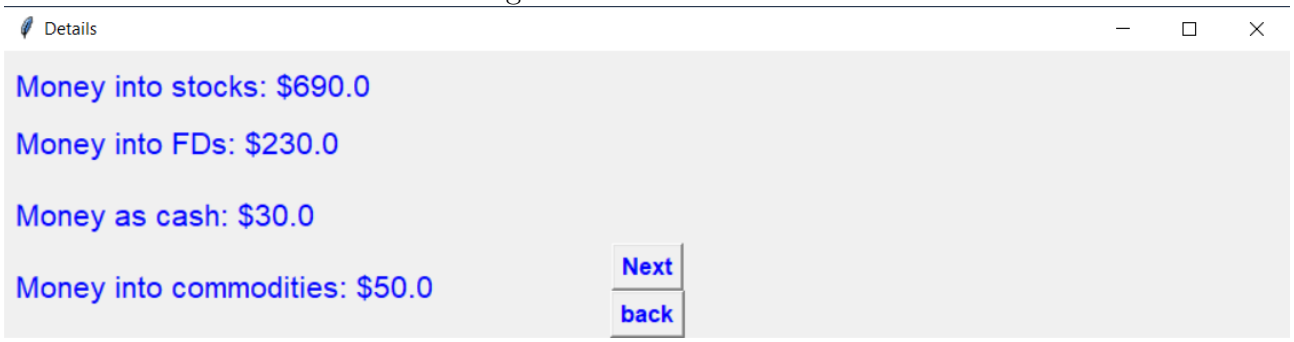

Figure 7.3: Breakup of Savings

Now, as seen iin Figure 7.3, based on the pie chart from Figure 7.2, the savings amount entered by the user are split up and displayed as a percentage of the savings amount.

The results displayed is the amount of savings going into each of the classes.

7.2 Stocks

Predicted Values for the next 10 days	
	0
0	312.0935
1	312.5988
2	312.6568
3	312.7124
4	312.8123
5	312.8834
6	312.945
7	313.0035
8	313.0575
9	313.1068

Table 7.1: Stocks - Predicted Values for the next 10 days

Since predictions were made using LSTM, the preliminary module gives preliminary result in the form of future predictions (10) and a graph where the actual and the predicted values are plotted against spot versus time (in days). Predicted values are in Fig 7.4 and TABLE 2.1 for reference. The MSE obtained for the SETFNIFBK.NS Index Fund is very close to 0 represented in Fig.7.5 below.

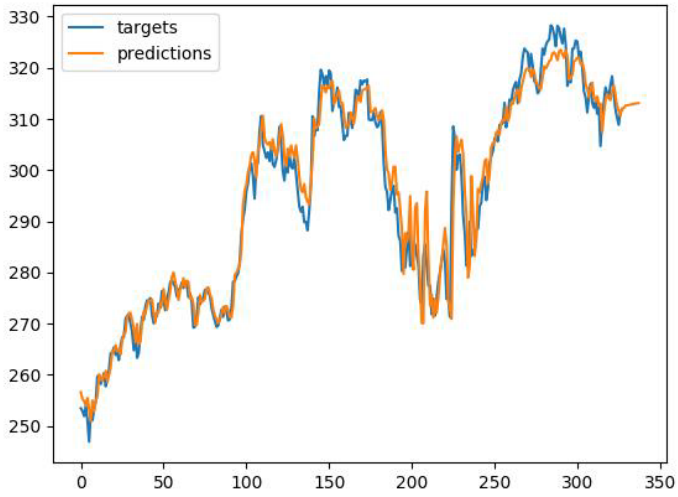

Figure 7.4: SETFNIFBK.NS Stock prediction

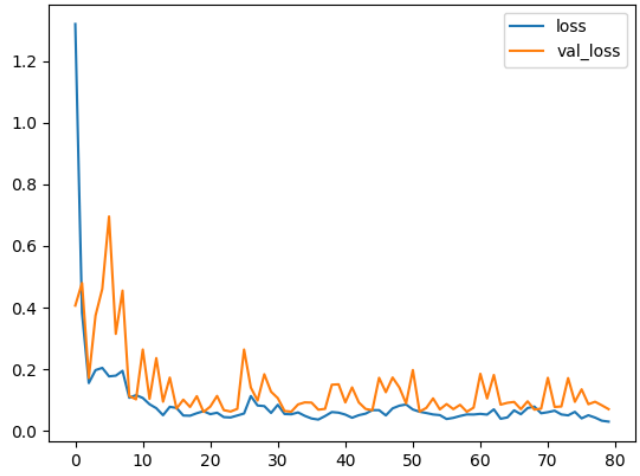

Figure 7.5: SETFNIFBK.NS Mean Square Error

All other predictions are displayed below. The predicted stock always seems to move with the actual stock.

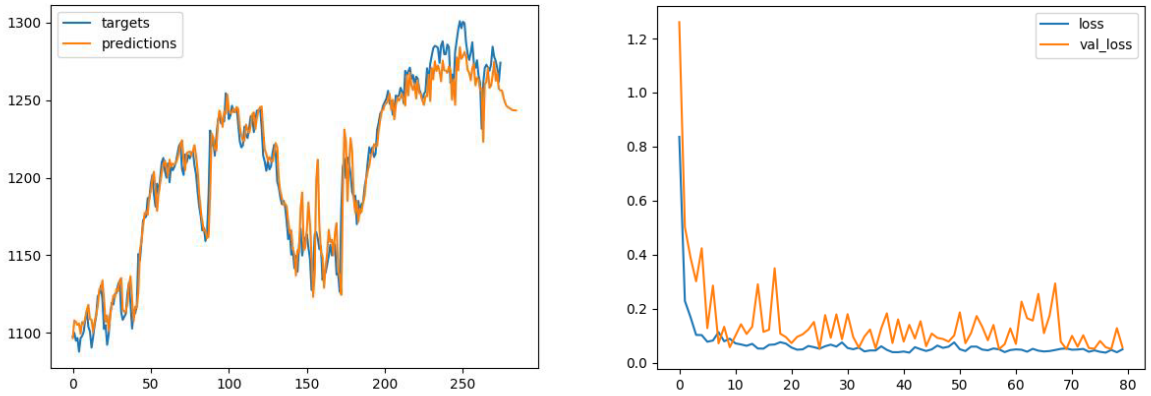

Figure 7.6: HDFCNIFETF.NS

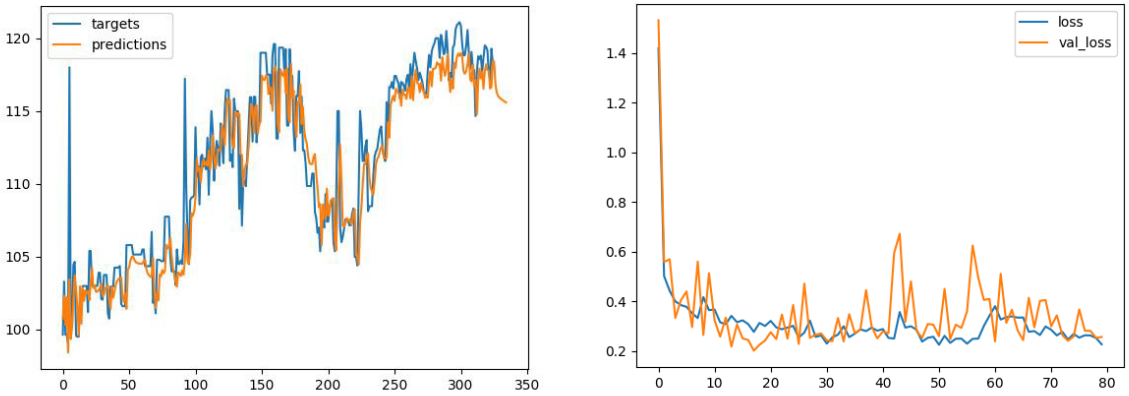

Figure 7.7: M50.BO

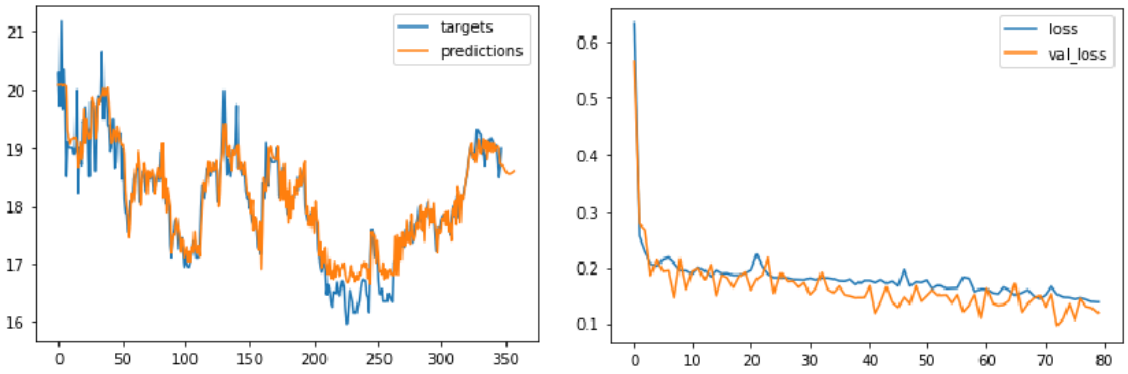

Figure 7.8: M100.BO

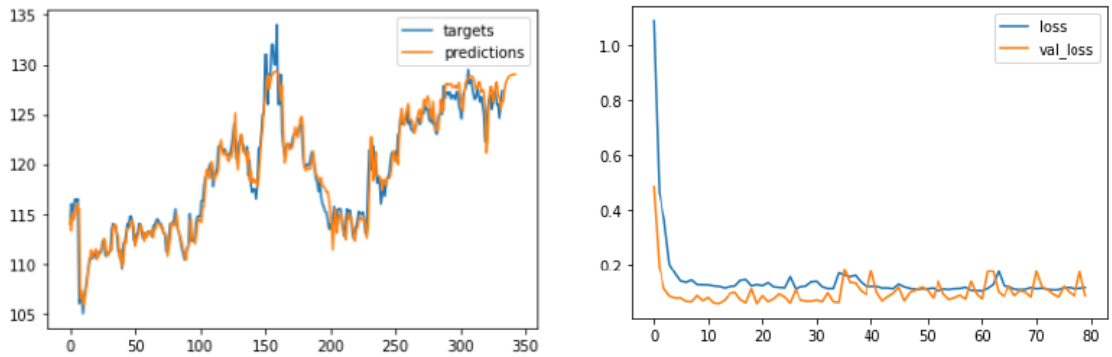

Figure 7.9: SETFBSE100.BO

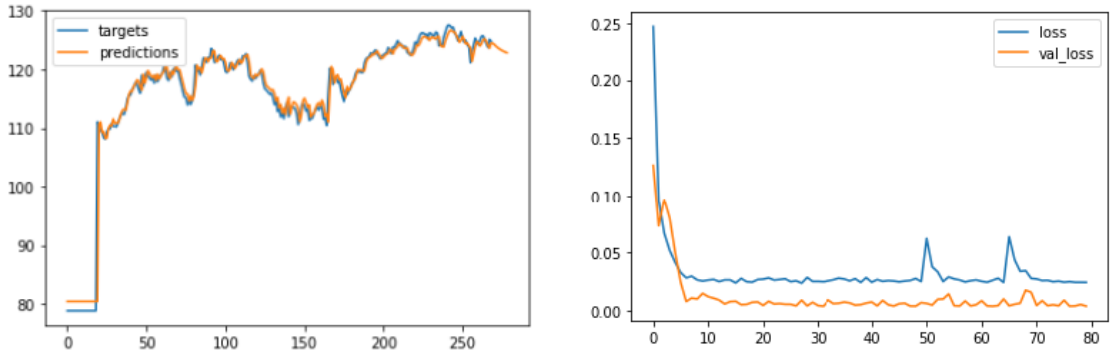

Figure 7.10: SETFNIF50.NS

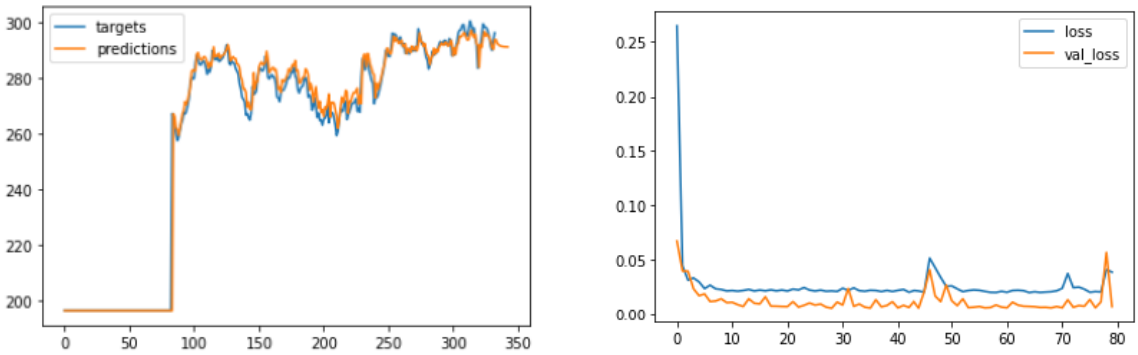

Figure 7.11: SETFNN50.NS

Observing the stock predictions in Fig.7.9, 7.10 and 7.11 the mean squared error is below 0.1 and almost zero for Fig. 7.10 and 7.11. This shows how highly accurate the predictions are for the stock time series. However in Fig. 7.6, 7.7 and 7.8 the mean squared error lies between 0.1 and 0.2 due to which in some cases the predictions do not indicate the exact stock value but they always move along the trend and give peaks and troughs which is sufficient to predict maxima's and minima's for profitable stocks. Using these peaks troughs we'll help our investor and tell him when to buy or sell the particular stock.

7.3 Commodities

Predicted Values for the next 10 days	
	0
0	224.8382
1	236.0543
2	244.7556
3	247.1265
4	249.0646
5	251.5855
6	253.3573
7	255.0003
8	256.3522
9	257.3673

Table 7.2: Commodities - Predicted Values for the next 10 days

Using the same LSTM, we modelled each commodity. The Above table displays predictions for the next 10 days of HINDPETRO.NS. MSE is also very close to 0, indicating accurate results.

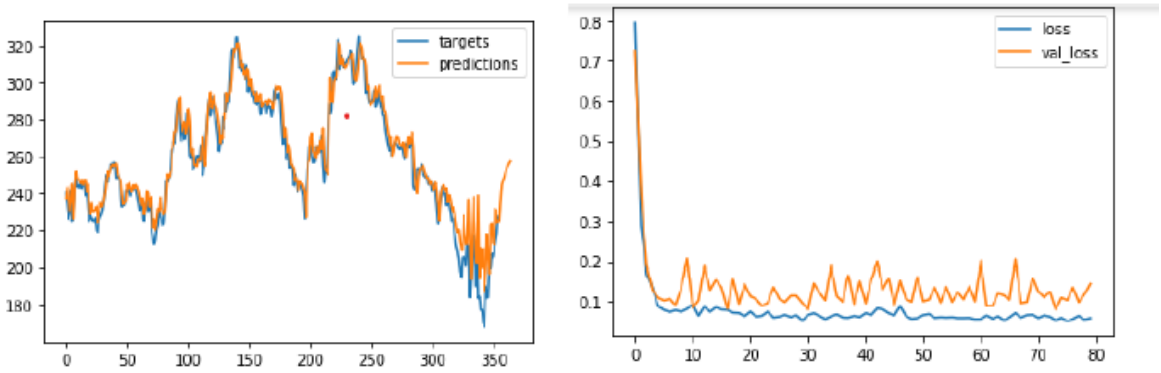

Figure 7.12: HINDPETRO.NS

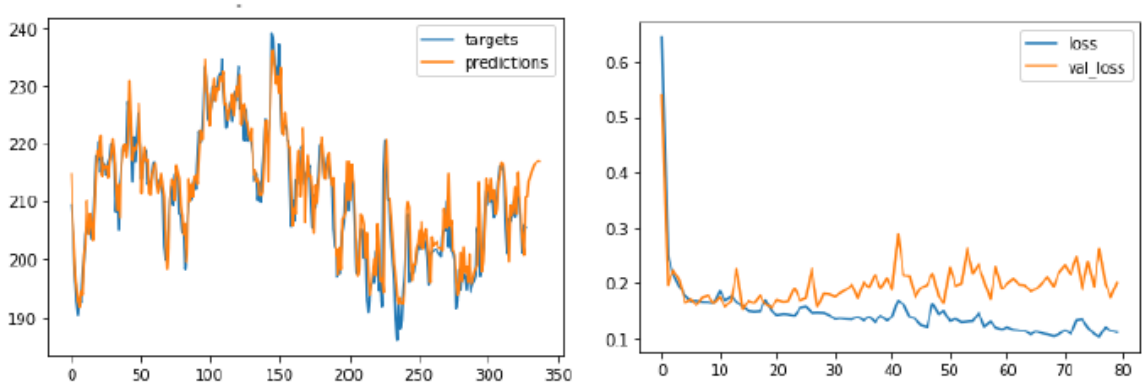

Figure 7.13: AMBUJACEM.NS

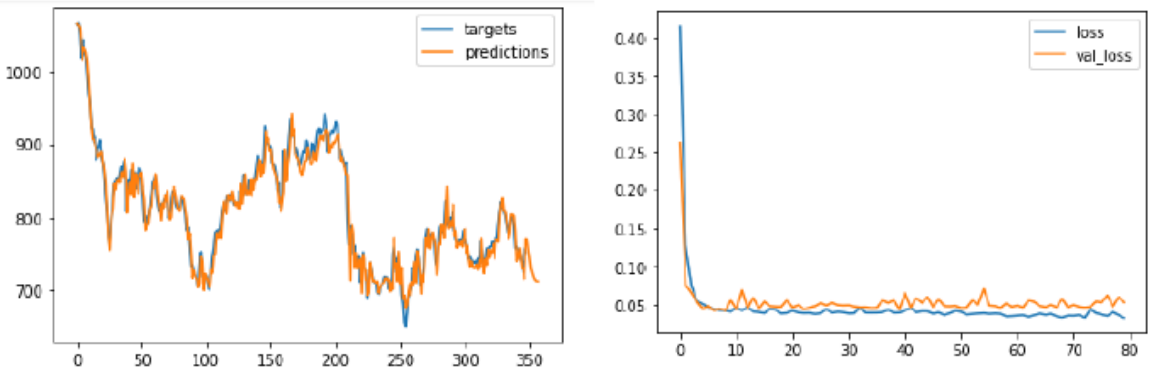

Figure 7.14: GRASIM.NS

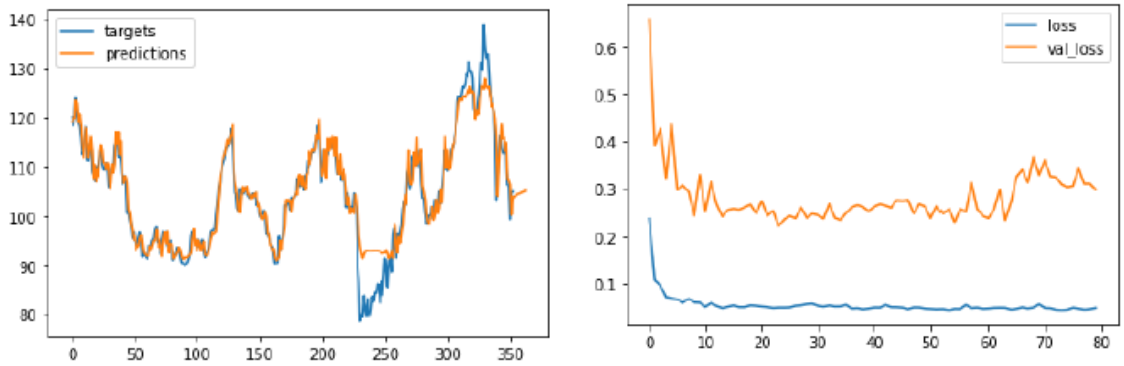

Figure 7.15: NMDC.NS

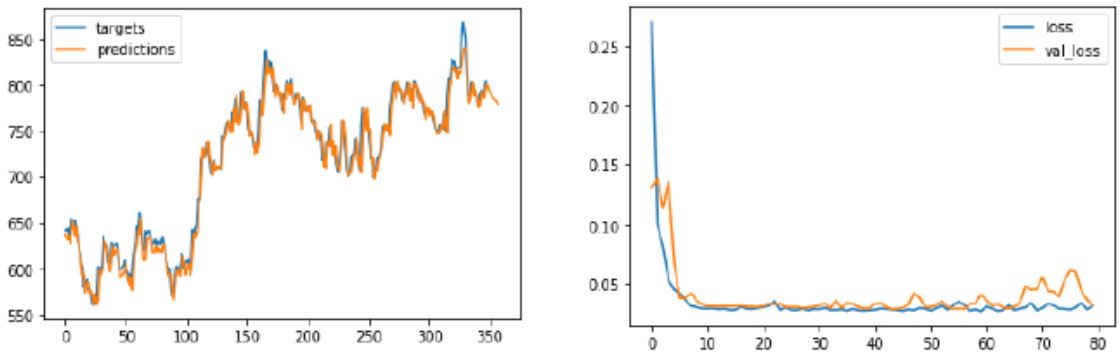

Figure 7.16: RAMCOCEM.NS

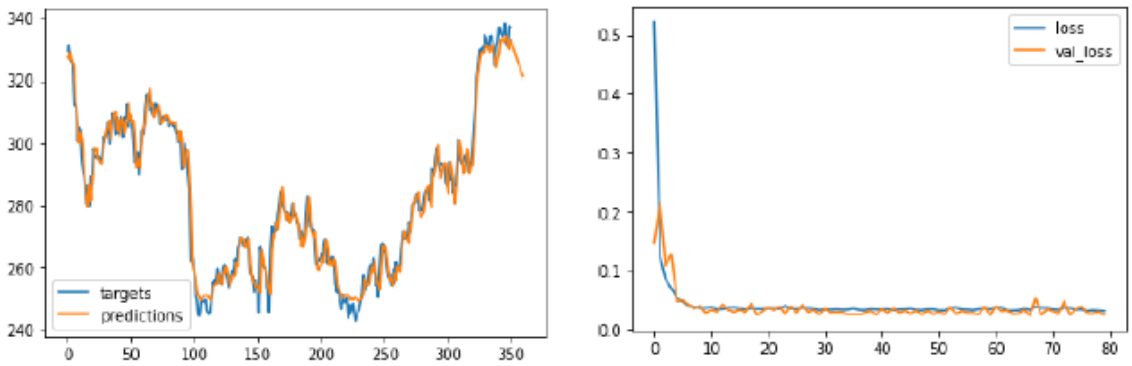

Figure 7.17: TATACHEM.NS

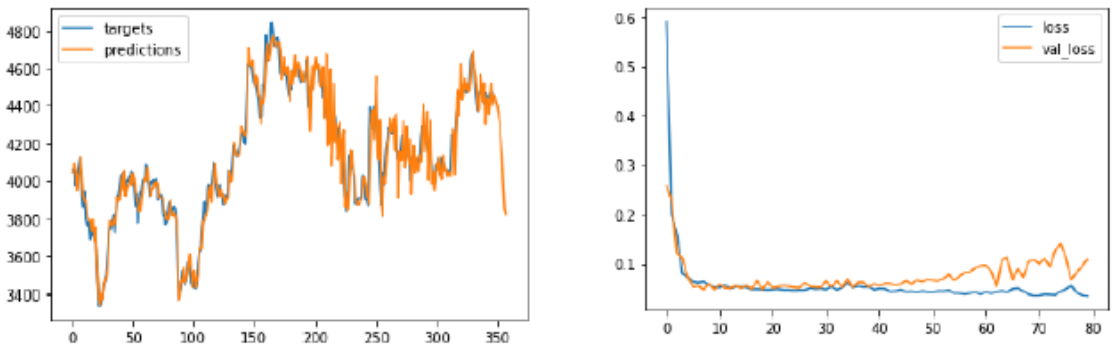

Figure 7.18: ULTRACEMCO.NS

Observing the commodity predictions in Fig.7.14 and 7.16the mean squared error is below 0.05. In Fig.7.12, 7.15, 7.17 and 7.18 it is below 0.1. It ranges between 0.1 to 0.2 online in Fig. 7.13. Again just like stocks the prediction always moves with the trend and using the peaks and troughs it is easy to tell when to buy and sell commodities to make a profit in the near future.

7.4 Fixed Deposit

Based on the amount invested in Fixed Deposit it gives the best returns for each bank. The interest rates for each bank are web scraped using Beautiful Soup from the link, the data is then webscraped into an intermediate file and then sorted and stored in a particular format in a csv file as shown in the figure 7.19. Using this data stored in the csv file this module calculates the interest earned on the principal amount for 8 banks those of which include Public, Private and Foreign Banks. The figure 7.24 shows the Fixed Deposit Prediction.

	A	B	C
1	Name of Bank	Tenure	Interest
2	SBI Fixed Deposit	7 days to 45 days	2.90%
3	SBI Fixed Deposit	46 days to 179 days	3.90%
4	SBI Fixed Deposit	180 days to 210 days	4.40%
5	SBI Fixed Deposit	211 days to 364 days	4.40%
6	SBI Fixed Deposit	1 year to 1 year 364 days	5.10%
7	HDFC Bank Fixed Deposit	7 days to 14 days	3.00%
8	HDFC Bank Fixed Deposit	15 days to 29 days	3.50%
9	HDFC Bank Fixed Deposit	30 days to 45 days	4.00%
10	HDFC Bank Fixed Deposit	46 days to 90 days	4.50%
11	HDFC Bank Fixed Deposit	91 days to 6 months	4.50%
12	ICICI Bank Fixed Deposit	7 days to 14 days	3.25%
13	ICICI Bank Fixed Deposit	15 days to 29 days	3.50%
14	ICICI Bank Fixed Deposit	30 days to 45 days	3.75%
15	ICICI Bank Fixed Deposit	46 days to 60 days	4.25%
16	ICICI Bank Fixed Deposit	61 days to 184 days	4.25%
17	Axis Bank	7 days to 14 days	3.25%
18	Axis Bank	15 days to 29 days	3.50%
19	Axis Bank	30 days to 45 days	4.00%
20	Axis Bank	46 days to 60 days	4.50%
21	Axis Bank	61 days to 3 months	4.50%
22	Kotak Bank	7 days to 14 days	3.00%
23	Kotak Bank	15 days to 30 days	3.00%

output

Figure 7.19: Data stored in CSV File

7.5 Predictions

Before starting predictions of the assets, we ask for confirmation from the user if he accepts the breakup of portfolio. This is to create a free and interactive environment. If the user agrees with the break up he selects 'No' else he goes for 'Yes' and he is made to enter the breakup he wishes manually. As seen in Figure 7.20, the question is asked and two options are available for the user to enter as per his/her own convenience.

Figure 7.20: Breakup of Savings - Application based

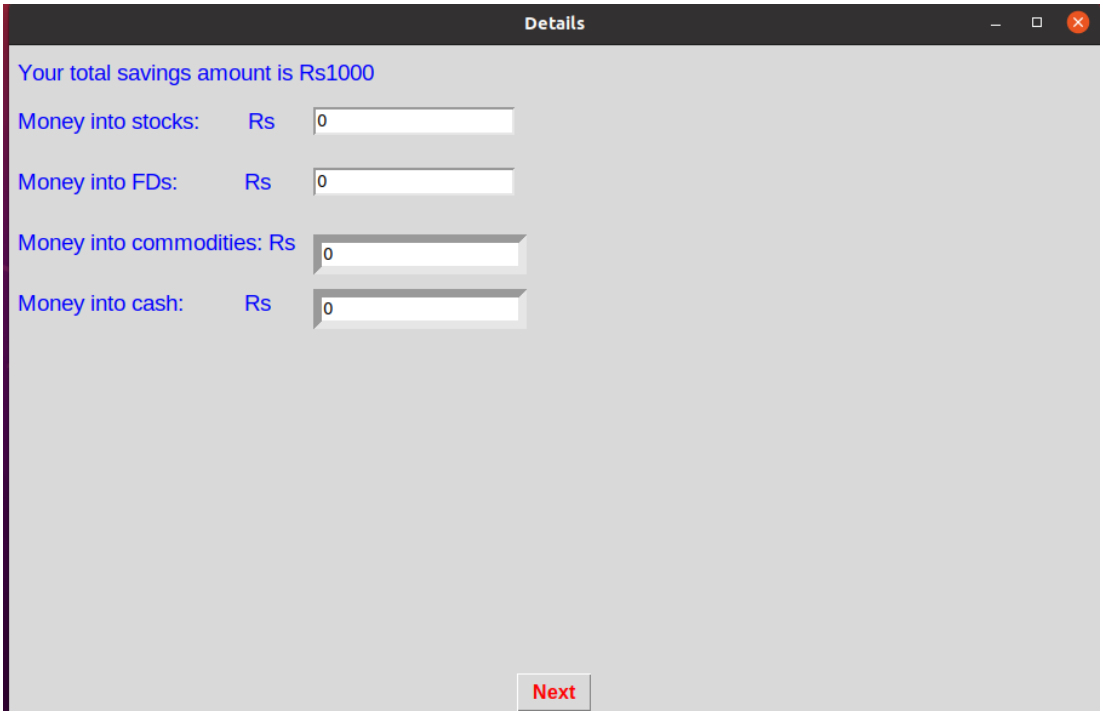

Figure 7.21: Breakup of Savings - User based

Figure 7.21 shows the possibility where the user chose 'Yes', so now he has the freedom to enter the amount that he wishes to enter in any of the classes.

The user enters the amount that he thinks is more suitable for him according to his own strategy as seen in Fig. 7.22.

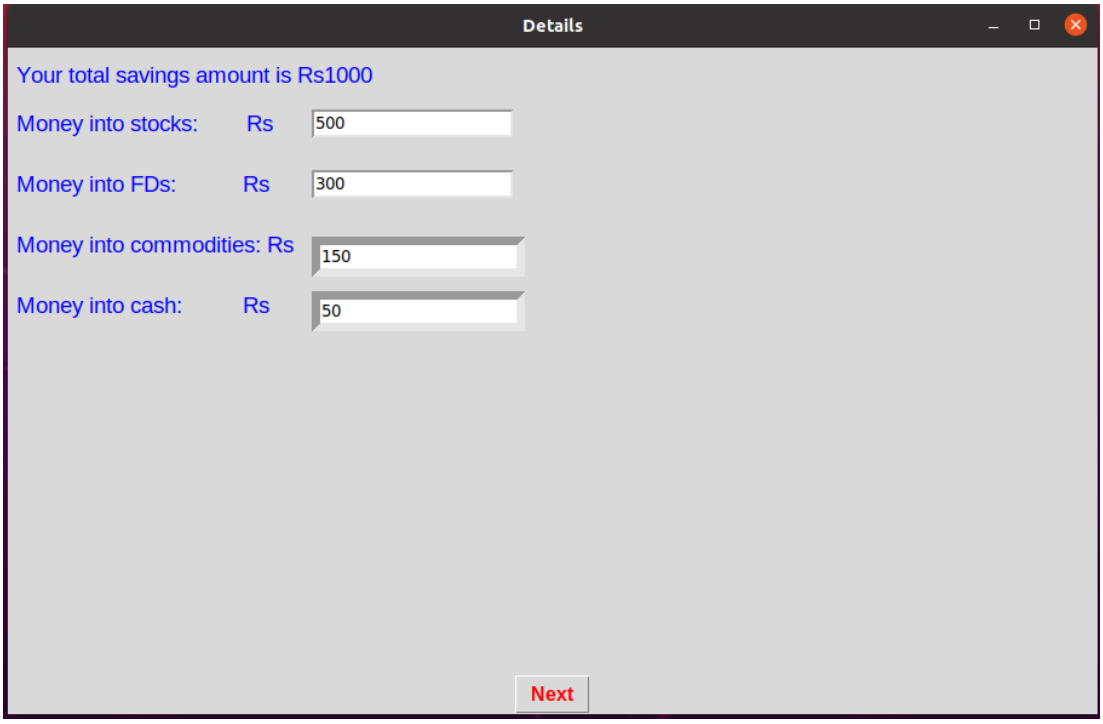

Figure 7.22: User Input Breakup of Savings

Then based on the user input we display our predictions using the alogoriths shown before for stocks, commodities and fixed deposits.

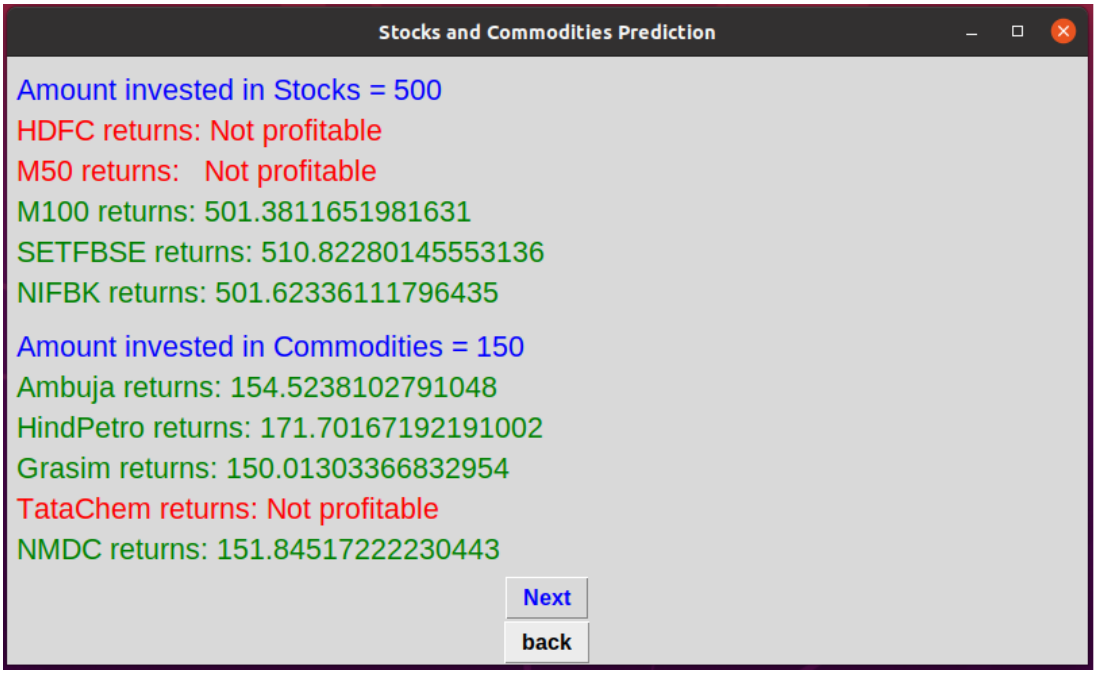

Figure 7.23: Stocks and Commodities Prediction

Now, as seen in Fig. 7.23, the predictions using our own algorithms on the amount entered by the user are diplayed. They are in green accent if they are making profit and the profit amount is diplayed next to the name of the stock. If they are in loss or are not making any profit they are given a red accent and a 'Not profitable' tag is dislpayed next to the stock name. This scheme is followed along in commodities and Fixed Desposits too.

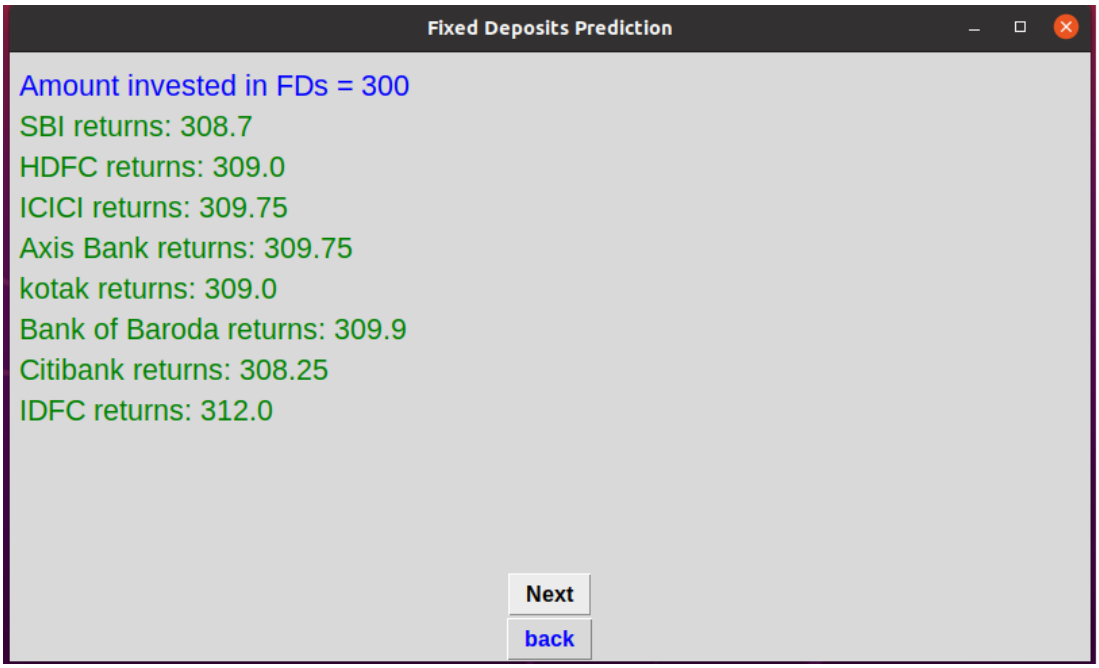

Figure 7.24: Fixed Deposit Prediction

Figure 7.24 shows the Fixed Deposits' predictions which are all in green as they never incur any loss.

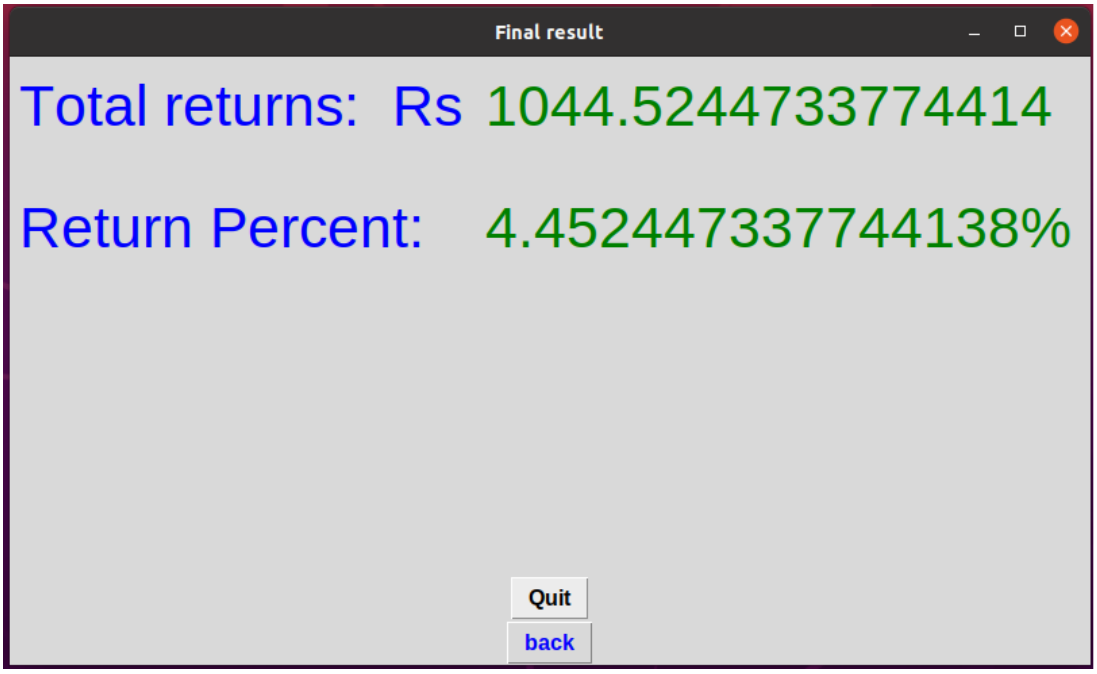

Figure 7.25: Total Return with Percentage

The final output with the percentage gain is displayed which is the total gain the user gains from our program. This obviously is bound to change with the change in data that is feeded to the algorithm.

The final compounded return from investing according to the algorithm is shown in Fig. 7.25, also with it the percentage return is displayed.

Chapter 8

Timeline required for project Implementation

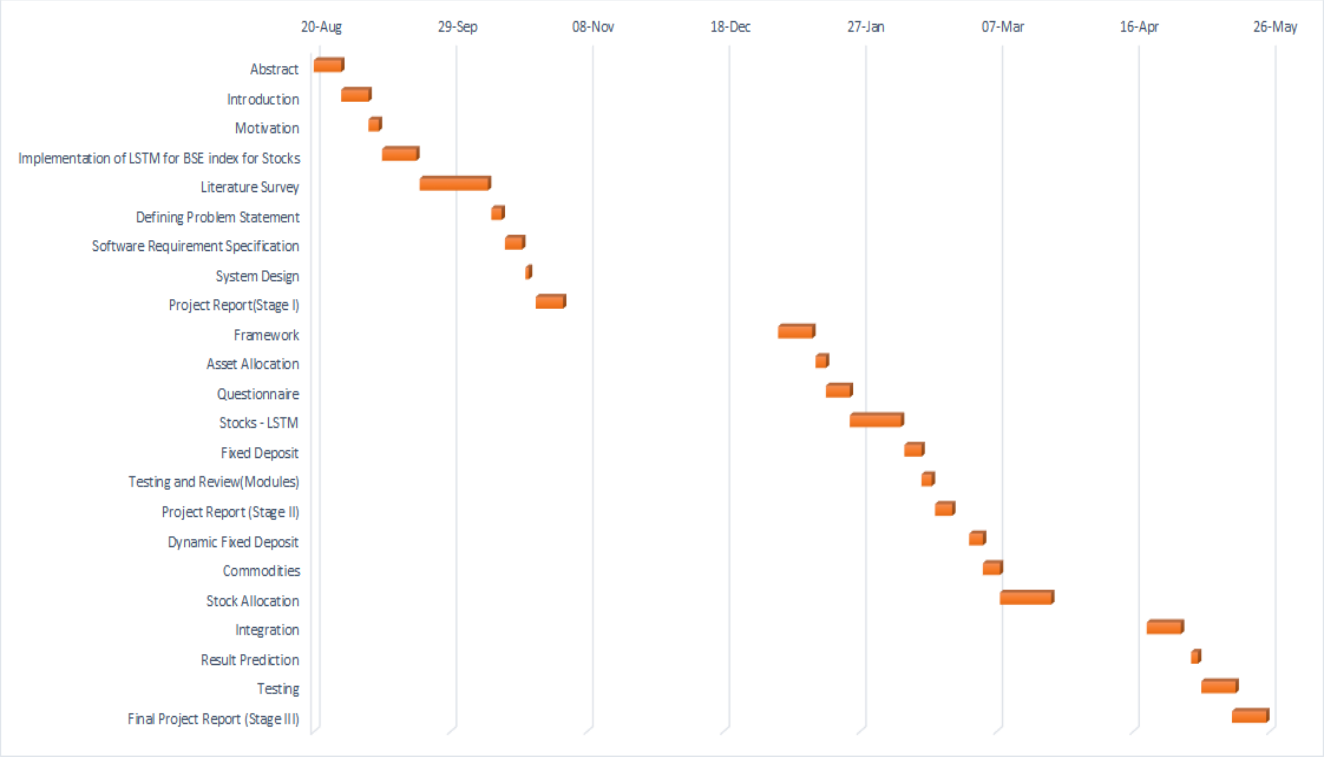

Figure 8.1: Timeline For Project Implementation

Chapter 9

Future Scope

The Personal Finance Asset Management Application can be replicated for all ETF markets across the world. In fact major retail banks can tweak an incorporate this app based on their needs so that it reaches their customers. The model can be used for all existing stocks and commodity ETF's if implemented in a distributed system. This will give people insights of stocks that they didn't even know existed. Instead of a customer visiting individual banks physically or their websites a person can get all interest rates in one platform.

At a later stage one stock can be shared by multiple investors so even if they can't afford it they can share it. Other Complex financial instruments like Options, Futures/Forwards as well as derivatives can also be added.

Chapter 10

Conclusion

The Risk Return Scores were used to categorise investors as conservative, moderately aggressive or aggressive. After the classification, a portfolio breakup is done based on the three categories mentioned above. The LSTM model was chosen based on our literature survey to predict stocks as well as commodities. LSTM prediction results show good accuracy level, even if at times It does not predict exact values but the stock market graph mostly moves with the actual trend. Since the trend is mostly along the actual market values an investor always seems to get the right predictions and it is unlikely that he will make a loss. The last element FD is one of the safest investments and we dynamically fetch data to give an investor the most optimal yield based on maturity. Lastly our portfolio breakup even leaves some amount as cash so that in an emergency he does not have to sell stocks, commodities or even break FD's for that matter. This project proves it is possible to make a personal asset management application without the need of a human to manage funds thus saving the exorbitant fees charged by banks.

Bibliography

[1] Dharmaraja Selvathu, Vineet Kumar, Abhishek Mishra, Indian stock market prediction using artificial neural networks on tick data, 2019

[2] Manna Majumder, MD Anwar Hussian, Forecasting of Indian Stock market Index using Artificial Neural Network, 2015

[3] Özgür İcan, Taha buğra Çelik, Stock Market Prediction Performance of Neural Networks: A Literature Review, 2017

[4] Raghav Nandakumar, Uttamraj K R, Vishal R, Y V Lokeswari, Stock Price Prediction Using Long Short Term Memory, 2018

[5] Swetava Ganguli, Jared Dunnmon, Machine Learning for Better Models for Predicting Bond Prices, 2015

[6] Singhal, Tanvi A Study of Factors Influencing Switching Behaviour of Fixed Deposit Investors of Indian Banks, 2015

[7] Ahamed, H. a. , Perception And Behaviour Of Bank Deposit Investors In Tumkur, Karnataka, 2013.

[8] Salam Abdul and Kulsum, Umma, Savings Behavior in India: An Empirical Study. The Indian Economic Journal. Vol. 50, Issue 1, 2002

[9] Gaurav Kabra, P. K.. , Factors Influencing Investment Decision of Generations in India: An Econometric Study. Asian Journal of Management Research, 2010

[10] Brian J. Scott, CFA; James Balsamo; Kelly N. McShane; Christos Tasopoulos, The global case for strategic asset allocation and an examination of home bia, 2017

[11] Julie R. Agnew and Lisa R. Szykman, Asset Allocation and Information Overload: The Influence of Information Display, Asset Choice, and Investor Experience, 2005

[12] John P. Hussman, Strategic Allocation White Paper, 2019

[13] https://www.investopedia.com/terms/i/investing.asp last accessed on 17th October 2019

[14] https://www.investopedia.com/financial-edge/0312/how-commodities-predict-market-movement.aspx last accessed on 21st October 2019

[15] https://www.investopedia.com/articles/economics/08/determining-oil-prices.asp last accessed on 1st November 2019

[16] https://www.investopedia.com/articles/investing/112514/monte-carlo-simulation-basics.asp last accessed on 15th November 2019

[17] https://www.investopedia.com/articles/07/montecarlo.asp last accessed on 15th November 2019

[18] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 last accessed on 21st January 2020

[19] https://static.arnaudsylvain.fr/2017/03/Grable-Lyton-1999-Financial-Risk-revisited.pdf2017last accessed on 15th February 2020

Appendix A

List of Abbreviations

- LSTM - Long Short Term Memory
- RNN - Recurrent Neural Network
- FD - Fixed Deposit
- KNN - K-Nearest Neighbors
- ARIMA - Auto Regressive Integrated Moving Average
- ETF - Exchange-traded Fund
- DEMAT - Dematerialised Account
- AI - Artificial Intellegence
- HNWI – High Net Worth Individual
- IT - Information Technology
- SRS - System Requirement Specification
- GPU - Graphics Processing Unit
- API - Application Program Interface
- AMI - Amazon Machine Image

- MSE - Mean squared error
- CSV - Comma Separated Values
- SETFNIFBK.NS - SBI MF ETF NIFTY BANK/ETF
- HDFCNIFETF.NS - HDFC Nifty 50 ETF
- M50.BO - Motilal Oswal M50 ETF
- M100.BO - Motilal Oswal Midcap 100 ETF
- SETFBSE100.BO - SBI ETF BSE 100 ETF
- SETFNIF50.NS - SBI ETF Nifty 50
- SETFNN50.NS - SBI ETF Nifty Next 50
- HINDPETRO.NS - Hindustan Petroleum Corporation Limited
- AMBUJACEM.NS - Ambuja Cements Limited
- GRASIM.NS - Grasim Industries Limited
- NMDC.NS - NMDC Limited
- RAMCOCEM.NS - The Ramco Cements Limited
- TATACHEM.NS - Tata Chemicals Limited
- ULTRACEMCO.NS - UltraTech Cement Limited