Project Report

On

Cricket Pitch Player Analysis

Developed By:

Jainam Shah (18162121033)

Jinesh Shah (18162121034)

Krutik Shah (18162121035)

Guided By:

Prof. Rahul Shrimali Parekh (internal)

Prof. Krishna Pal (external)

Submitted to

Department of Computer Science & Engineering

Institute of Computer Technology

Year: 2020

CERTIFICATE

This is to certify that the Application Development Project work entitled **"Cricket Pitch Player Analysis"** by Jainam Shah(Enrolment No.18162121033) of Ganpat University, towards the fulfilment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE(CBA/BDA/CS) Department. The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name & Signature of IBM Mentor

Name & Signature of Internal Guide

Place:

Date:

ACKNOWLEDGEMENT

Application Development project is a golden opportunity for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank **Dr. Hemal Shah,** Head of Department, Computer Science and Engineering, who gave us an opportunity to undertake this project. My grateful thanks to **Prof. Rahul Shrimali & Prof. Krishna Pal (Internal & External Guides)** for their guidance in project work **Cricket Pitch Player Analysis**, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where would have been without his/her help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Jainam Shah (Enrollment No:18162121033)

Jinesh Shah (Enrollment No:18162121034)

Krutik Shah (Enrollment No:18162121035)

ABSTRACT

Modern Day cricket is very fast and increasing and evolving in every aspects. In order to stay with it and provide better understanding and visualization any batsman is very necessary. In this case Data science can be very profound and can be used to explain performance analysis of any batsman around the world. With the advent and benefit of graph monitoring the data analysis of any batsman can be obtained easily and in an understandable way.

INDEX

CHAPTER 1 INTRODUCTION

The cricket battlefield is competitive and challenging. Players have become extremely professional and disciplined about their training. The Cricket Board and training centre have analysed the power of the shot a batsman hits, speed of his/her bat, directional and shaped shot what not to ensure top performance from the players. The cut-throat competition doesn't make it easy either. In spite of all the hoopla, a few players still make it to the top. They raise the bar of the game every time they walk to the field.

By considering this above scenario we decided to make a UI based application which helps to analyses each and every shot a batsman tries to play in the field by using a mobile based sensor called HyperIMU which initiates the phone's Accelerometer and gyroscopic sensor and takes reading according to the shot played by a batsman; to make this application we have first extracted data from device and then displayed it using graphs; after getting clean data we have done feature selection and extraction;

The main focus of this project is to make a proper and predictive model of the shots played by a batsman in his/her practice sessions and also during a live cricket match. Below is the list of the tools and technologies which we have used in this project:-

- Python (3.7) and HyperIMU for data fetching from device
- Matplotlib for plotting graphs according to sensor data
- Pandas for providing surface analysis
- Python Tkinter for providing proper User Interface

CHAPTER: 2 PROJECT SCOPE

The project is limited to Desktop or Web based application.

CHAPTER: 3 SOFTWARE AND HARDWARE REQUIREMENTS

CHAPTER 3 SOFTWARE AND HARDWARE REQUIREMENTS

Minimum Hardware Requirements

Processor	Intel(R) Core(TM) i3-7020U CPU 2.3 GHZ
RAM	8 GB
HDD	40 GB

Table 3.1 Minimum Hardware Requirements

Minimum Software Requirements

Operating System	Any operating system which can support an internet browser.
Programming language	Python
Other tools & tech	Python IDLE,Matplotlib,Visual Code

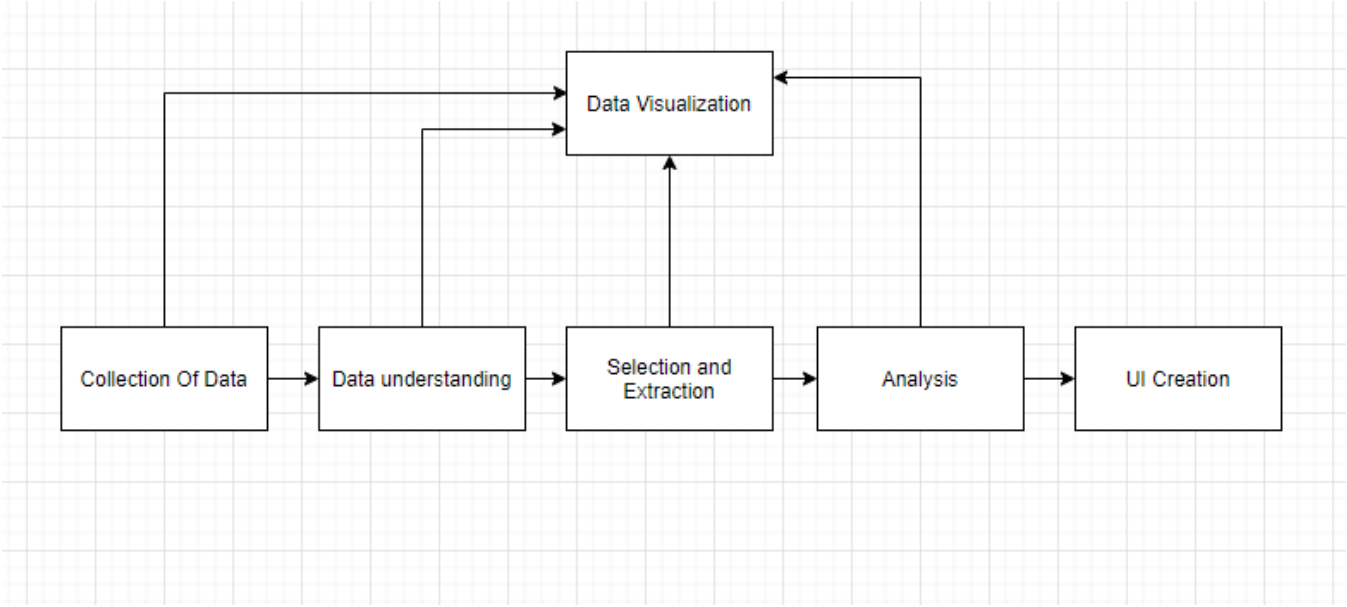
Table 3.2 Minimum Software Requirements

CHAPTER: 4 PROJECT PLAN

List of Activities

Task: - 1 Configuring the Sensors
Task: - 2 Implementing Data Extraction Strategies
Task: - 3 Collecting Data
Task: - 4 Analysing the Data
Task: - 5 UI Creation
Task: - 6 Performing Data Visualization

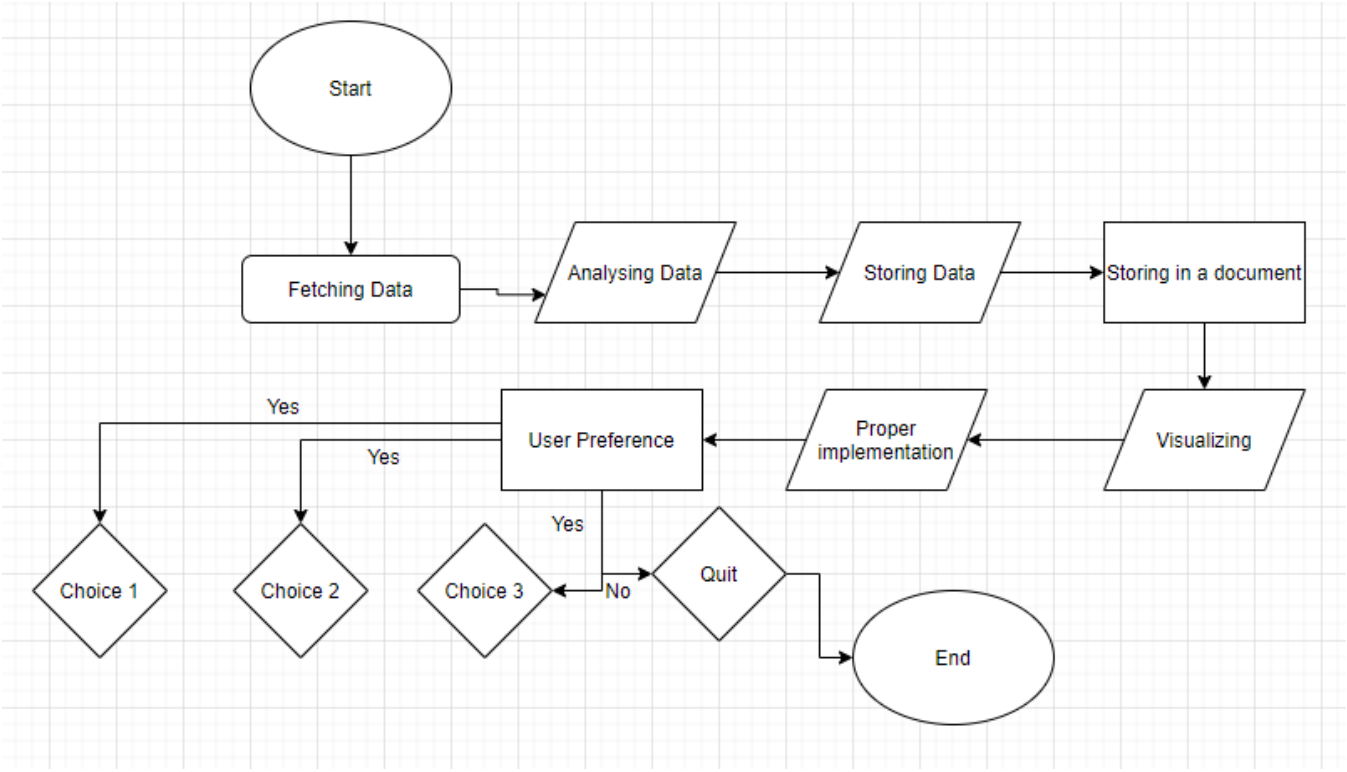
Process Model Of Project



CHAPTER 5 IMPLEMETATION DETAILS

5.1 DATA COLLECTION

FLOWCHART

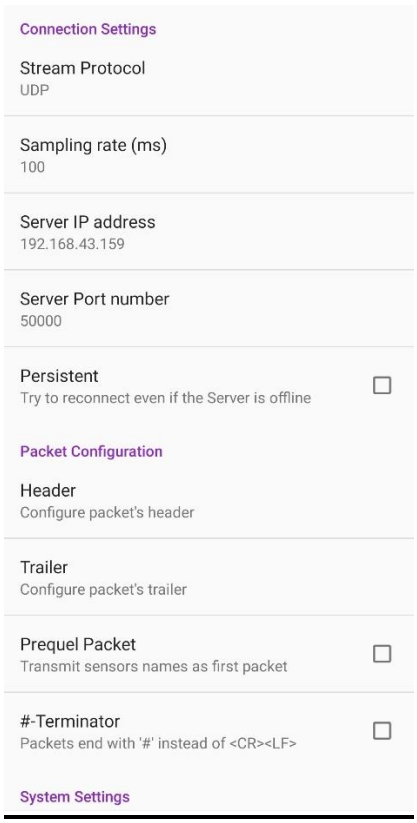


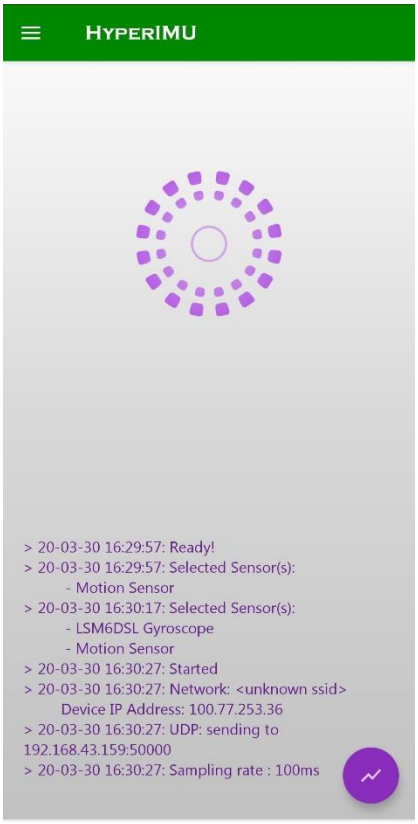
Extraction of sensor data from HyperIMU app and tools which were used for extracting it were:

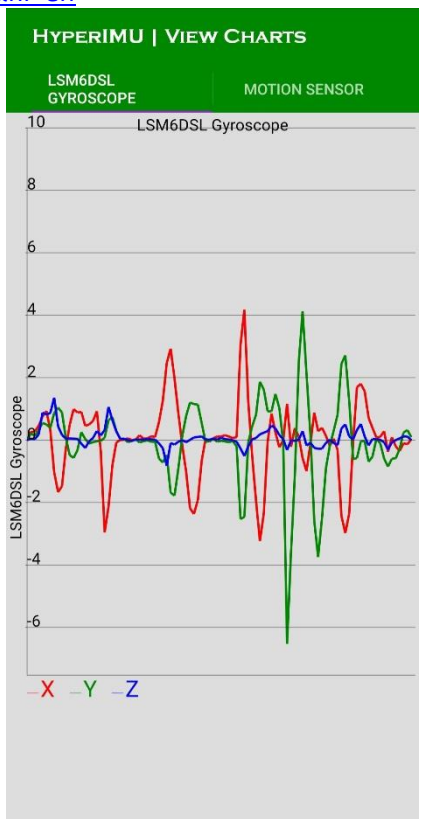
- PYTHON PROGRAMMING LANGUAGE
- HYPERIMU App

Configuring the Sensors

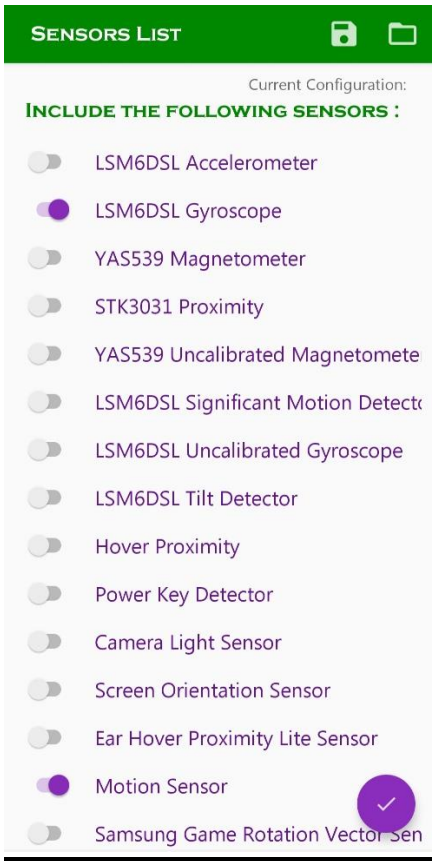
HyperIMU App was used to configure sensors from mobile phones and extract data accordingly. You can find the app in the below mentioned link.
https://play.google.com/store/apps/details?id=com.ianovir.hyper_imu&hl=en







Sensor Variations



Various Sensors like Gyroscope, Accelerometer, Magnetometer and Motion sensor were being equipped for determining area, distance, elevation (height) respectively

Further multiple data extraction strategies were implemented using Python Programming Language which resulted in data extraction from sensors as given below.

5.2 Understanding Data

Data Extraction Code

```
import socket
import sys
UDP_IP = "192.168.43.159"
```

```
UDP_PORT_NO = 50000
#target = "www.google.com"
ServerSock=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
#ServerSock.connect((target,8968))
ServerSock.bind((UDP_IP,UDP_PORT_NO))
i=0
splitData=[]

while(i<30):
    data, addr = ServerSock.recvfrom(1024)
    newData = data.decode().split(",")
    newData[2]=newData[2][:-1]
    splitData.append(newData)
    print(data)
    i=i+1
```

OUTPUT

```
============ RESTART: C:\Users\Admin\Desktop\WebProject1\module1.py ============
b'0.0,0.0,0.0\r\n'
b'0.02873042,0.17477672,9.791088\r\n'
b'0.019153614,0.20350714,9.7863\r\n'
b'0.01436521,0.18195933,9.785102\r\n'
b'0.01436521,0.19273324,9.787497\r\n'
b'0.005985504,0.19393034,9.776723\r\n'
b'0.020350715,0.17836803,9.785102\r\n'
b'0.019153614,0.19752164,9.78989\r\n'
b'0.011971008,0.19512744,9.794679\r\n'
b'0.019153614,0.17956513,9.805453\r\n'
b'0.01436521,0.19393034,9.7958765\r\n'
b'0.01436521,0.19153613,9.803059\r\n'
b'0.009576807,0.18674773,9.805453\r\n'
b'0.011971008,0.19632454,9.799467\r\n'
b'0.005985504,0.19153613,9.801862\r\n'
b'0.009576807,0.18195933,9.8042555\r\n'
b'0.017956512,0.19512744,9.817424\r\n'
b'0.011971008,0.18195933,9.822212\r\n'
b'0.017956512,0.19273324,9.813833\r\n'
b'0.021547815,0.18914193,9.829395\r\n'
b'0.01316811,0.19632454,9.831789\r\n'
b'0.01436521,0.19033903,9.832987\r\n'
b'0.016759412,0.18555063,9.824607\r\n'
b'0.009576807,0.19632454,9.827001\r\n'
b'0.017956512,0.18794483,9.836577\r\n'
b'0.01316811,0.18315643,9.841366\r\n'
b'0.015562311,0.18674773,9.846154\r\n'
b'0.021547815,0.18195933,9.836577\r\n'
b'0.022744916,0.18914193,9.832987\r\n'
b'0.017956512,0.18315643,9.835381\r\n'
>>>
```

Collection of Data

The above given coordinates/data obtained from the sensor were stored and saved in a .txt file and also a .csv file in the same folder which was mapped in a separate file using URL during main execution

UI Creation

Creation of proper user interface was made using PYTHON Tkinter which is used for creating Desktop Based Applications.

There are two parameters covered in the UI which includes Fetch Data and implementation of Graphs using data visualization

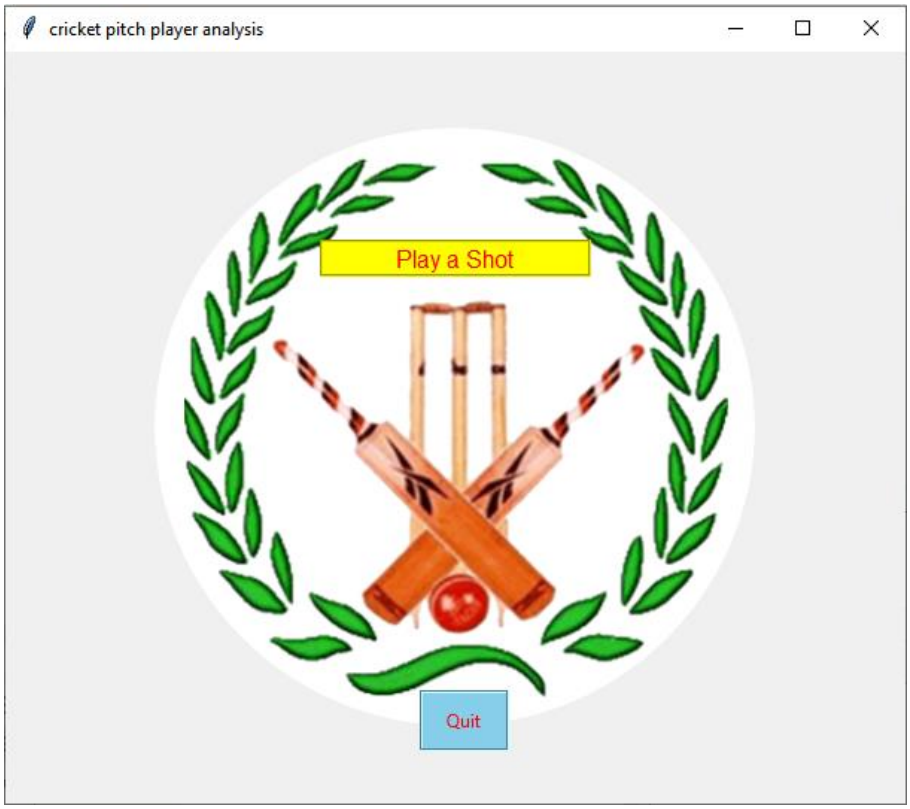
Code for UI Creation

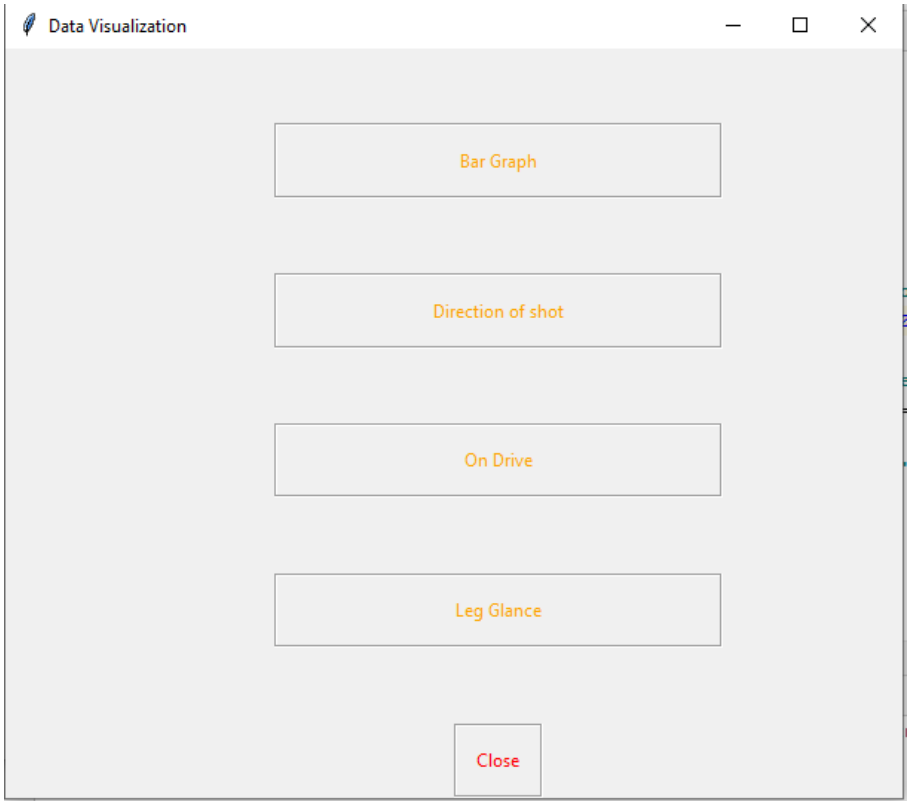
```
import tkinter as tk
import requests
from tkinter import *
import os
import socket
from PIL import ImageTk
from PIL import Image


HEIGHT = 500
WIDTH = 600

root = tk.Tk()
root.title('cricket pitch player analysis')

canvas = tk.Canvas(root, height=HEIGHT, width=WIDTH)
canvas.pack()


#canvas.create_image(0,0,image=photo)
background = tk.PhotoImage(file='cricket1.png')
label = tk.Label(root, image=background)
label.place(relwidth=1, relheight=1)

def FetchData():
    r = tk.Tk()
    r.title("Data Visualization")
    canvas = tk.Canvas(r, height=HEIGHT, width=WIDTH)
    canvas.pack()
    import socket
    import sys
    btn = Button(r, text="Bar Graph", fg="orange", command=Img1, height="2", width="40",
relief=GROOVE)
    btn.place(relx=0.3, rely=0.1, relheight=0.1, relwidth=0.5)
    btn = Button(r, text="Direction of shot", fg="orange", command=Img2, height="2",
width="40", relief=GROOVE)
```

```
    btn.place(relx=0.3, rely=0.3, relheight=0.1, relwidth=0.5)
    btn = Button(r, text="On Drive", fg="orange", command=Img3, height="2", width="40",
relief=GROOVE)
    btn.place(relx=0.3, rely=0.5, relheight=0.1, relwidth=0.5)
    btn = Button(r, text="Leg Glance", fg="orange", command=Img4, height="2", width="40",
relief=GROOVE)
    btn.place(relx=0.3, rely=0.7, relheight=0.1, relwidth=0.5)
    btn = Button(r, text="Close", fg="red", command=r.destroy, relief=GROOVE)
    btn.place(relx=0.5, rely=0.9, relheight=0.1, relwidth=0.1)


UDP_IP = "192.168.43.159"
UDP_PORT_NO = 50000
# target = "www.google.com"
ServerSock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
# ServerSock.connect((target,8968))
ServerSock.bind((UDP_IP, UDP_PORT_NO))
i = 0
splitData = []

while (i < 30):
    data, addr = ServerSock.recvfrom(1024)
    newData = data.decode().split(",")
    newData[2] = newData[2][:-1]
    splitData.append(newData)
    print(data)
    i = i + 1


# Function for BarGraph Graph
def Img1():
    r = Toplevel()
    r.title("BarGraph")
    canvas = Canvas(r, height=600, width=1200)
    canvas.pack()
    widget = Label(r, text='The last three shots  played by batsman', fg='red')
    widget.place(relx=0.4, rely=0.1, relheight=0.1, relwidth=0.3)
    widget.pack()
    my_image = PhotoImage(file='BarGraph.png', master=root)
    canvas.create_image(0, 0, anchor=NW, image=my_image)
    r.mainloop()


# Function for Direction Graph
def Img2():
    r = Toplevel()
    r.title("Direction Of Shot")
    canvas = Canvas(r, height=460, width=600)
    canvas.pack()
    widget = Label(r, text=' That is a fine sweep shot near the fine leg ', fg='red')
    widget.place(relx=0.4, rely=0.1, relheight=0.1, relwidth=0.3)
    widget.pack()
    my_image = PhotoImage(file='PolarGraph1.png', master=root)
    canvas.create_image(10, 10, anchor=NW, image=my_image)
    r.mainloop()
```

```
# Function for Sweep Shot Graph
def Img3():
    r = Toplevel()
    r.title("Sweep shot elevation")
    canvas = Canvas(r, height=450, width=600)
    canvas.pack()
    widget = Label(r, text='A perfect bat elevation and timing fot the shot played',
fg='red')
    widget.place(relx=0.4, rely=0.1, relheight=0.1, relwidth=0.3)
    widget.pack()
    my_image = PhotoImage(file='OnDrive.png', master=root)
    canvas.create_image(10, 10, anchor=NW, image=my_image)
    r.mainloop()

def Img4():
    r = Toplevel()
    r.title("3D presentation")
    canvas = Canvas(r, height=450, width=600)
    canvas.pack()
    widget = Label(r, text='Visual Presentation of shot along with bat of player',
fg='red')
    widget.place(relx=0.4, rely=0.1, relheight=0.1, relwidth=0.3)
    widget.pack()
    my_image = PhotoImage(file='leg glance.png', master=root)
    canvas.create_image(0, 0, anchor=NW, image=my_image)
    r.mainloop()


btn = tk.Button(root, text="Play a Shot", fg='red', font=40, bg='yellow',
command=FetchData, relief=GROOVE)
btn.place(relx=0.35, rely=0.25, relheight=0.05, relwidth=0.3)
btn = tk.Button(root, text="Quit", fg='red', command=root.destroy, bg='skyblue',
relief=GROOVE)
btn.place(relx=0.46, rely=0.85, relheight=0.08, relwidth=0.1)

root.mainloop()
```

USER INTERFACE

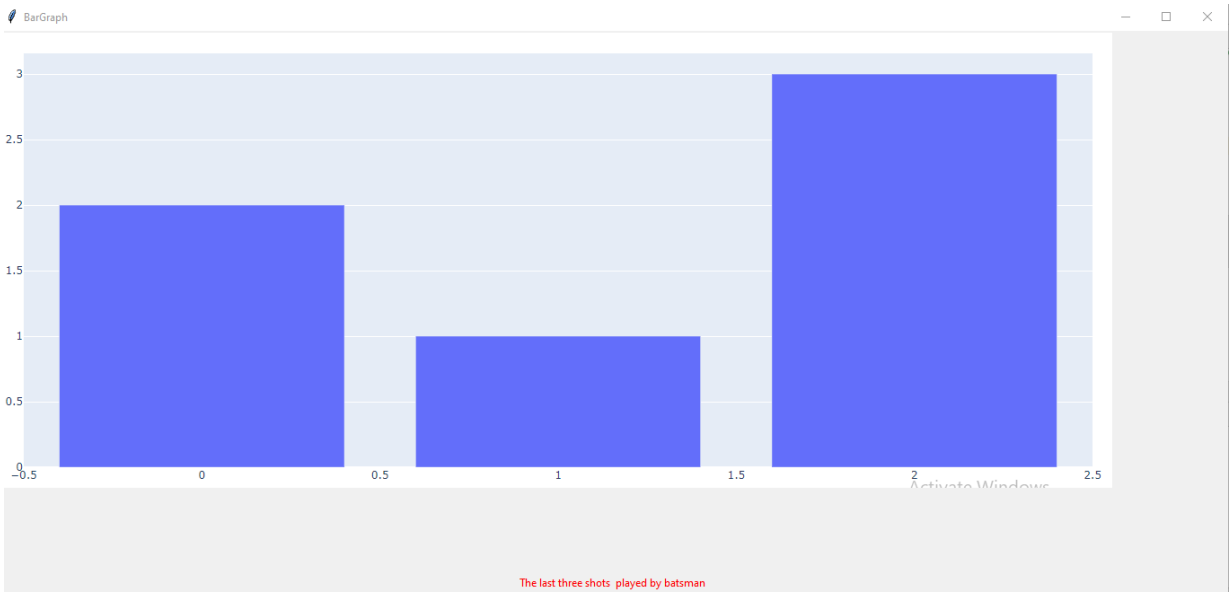




Data Visualization

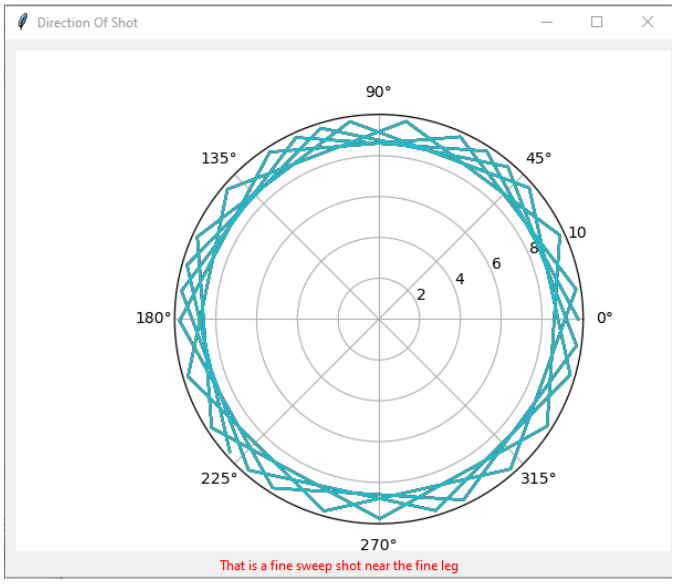
Furthermore Data visualization was performed using Matplotlib

Bar Graph (The following figure shows last three shots played by the batsman)



Polar Graph

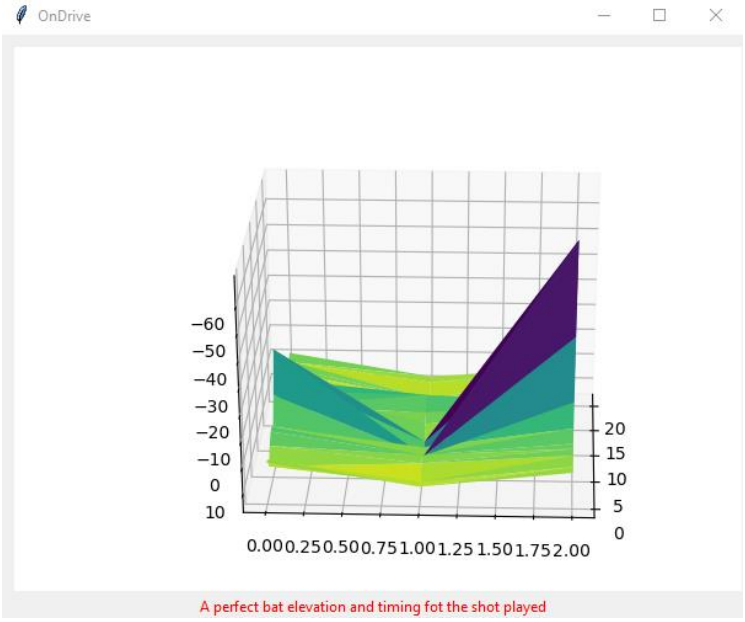
The following figure shows pictorial presentation of **Sweep shot** played by the batsman near the fine leg



Code

```
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt
import matplotlib.animation as animation

fig = plt.figure()
ax1 = fig.add_subplot(111,projection="polar")

def animate(i):
    pullData = open("module1.txt",'r').read()
    dataArray = pullData.split('\n')
    xar=[]
    yar=[]
    zar=[]
    for eachline in dataArray:
        if len(eachline)>1:
            x,y,z=eachline.split(',')
            xar.append(float(x))
            yar.append(float(y))
            zar.append(float(z))
    ax1.plot(xar,yar,zar)
ani= animation.FuncAnimation(fig,animate,interval=1000)
plt.show()
```

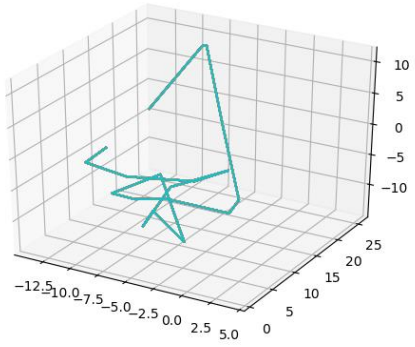
The following figure represents the bat elevation of the batsman while playing the above shot



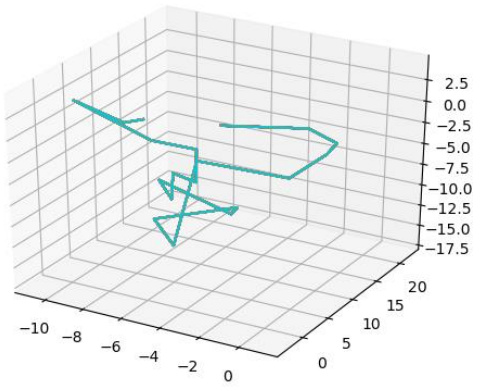
Graphs For Shots Played:-

3D Graps:-

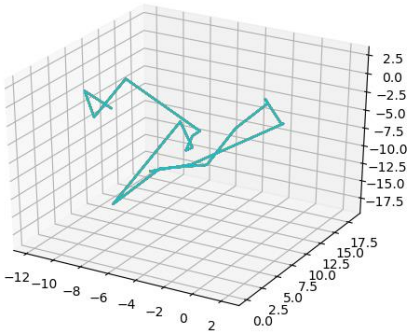
Upper Cut:-



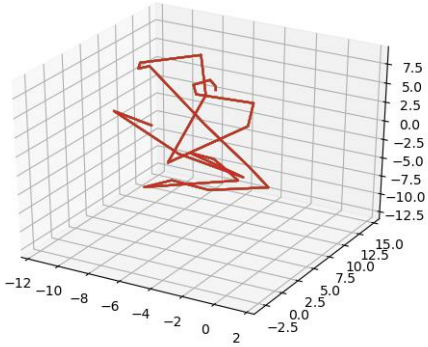
Straight Drive:-



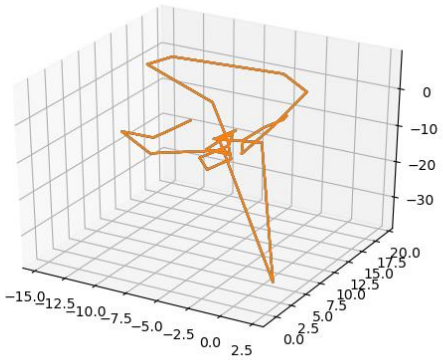
Leg Glance:-



Square Drive:-



Cover Drive:-



CHAPTER 6: CONCLUSION AND FUTUTRE WORK

CONCLUSION

To reiterate my views, as we all know that the ratio of Desktop applications has rapidly decreased in the last few years. Before developing a Desktop application if we can know
The approximate rating with the help of its properties which we are going to consider while Development then by this we can conclude that will our application will become a successful app, in future? If not then we can also know what kind of properties will help to make it popular in the real world. There is single constraint that will make this app slower which is time elapsed for updating shot and creating proper analysis but still we have tried developing a UI based application which will predict performance of the any player based on the shots played by him/her.

FUTURE

Till now we have clean application properties data and on the other side. Moreover, we have also done data visualization for original data-set. Following are the task which we are going to implement now.

- **Proper data analysis of the shots played**
- **Data visualization in an effective manner**

REFERENCES

[1] www.matplotlib.org
[2] www.python.org
[3] Python Data Science Handbook – Essential Tools working with data by Jake Vanderplas