

Fundamentals of Big Data Analytics

PROJECT Weightage: 16%

Submitted Date: 10 June 2022

Group Members

Roll No

Rabia Mustafa :

201-1853

Neha Shabbir :

201-0677

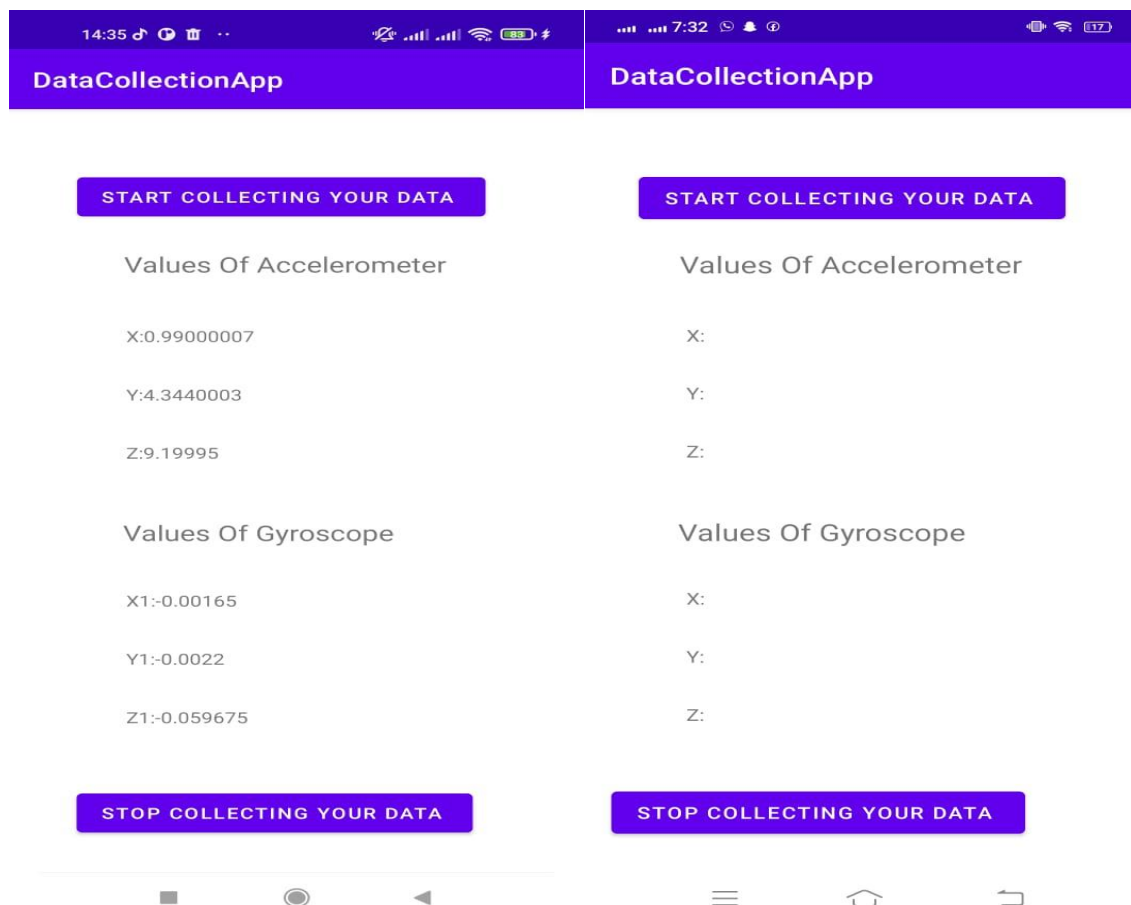
Muhammad Ahsan :

201-1787

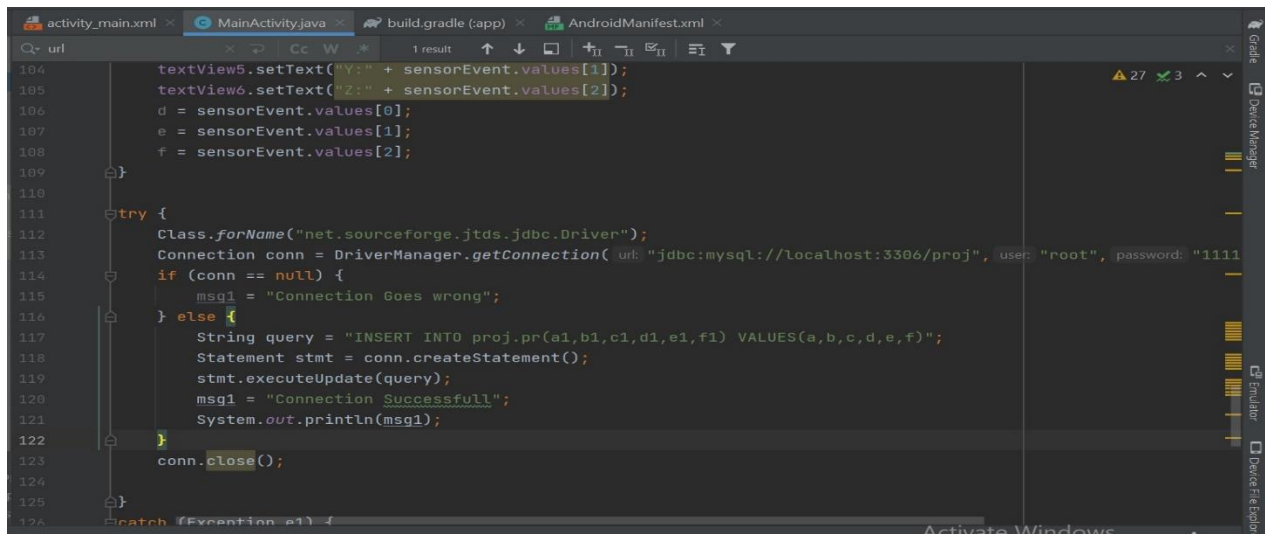
Making Mobile APP:

1- We make mobile App on android studio in java language. There are two main files in it, one is xml file in which app interface is declared and other is java file that contains connection for xml file and has functions to collect accelerometer and gyroscope data from mobile phone.

2- App contains start and stop button to start generating data and to stop when required data comes.



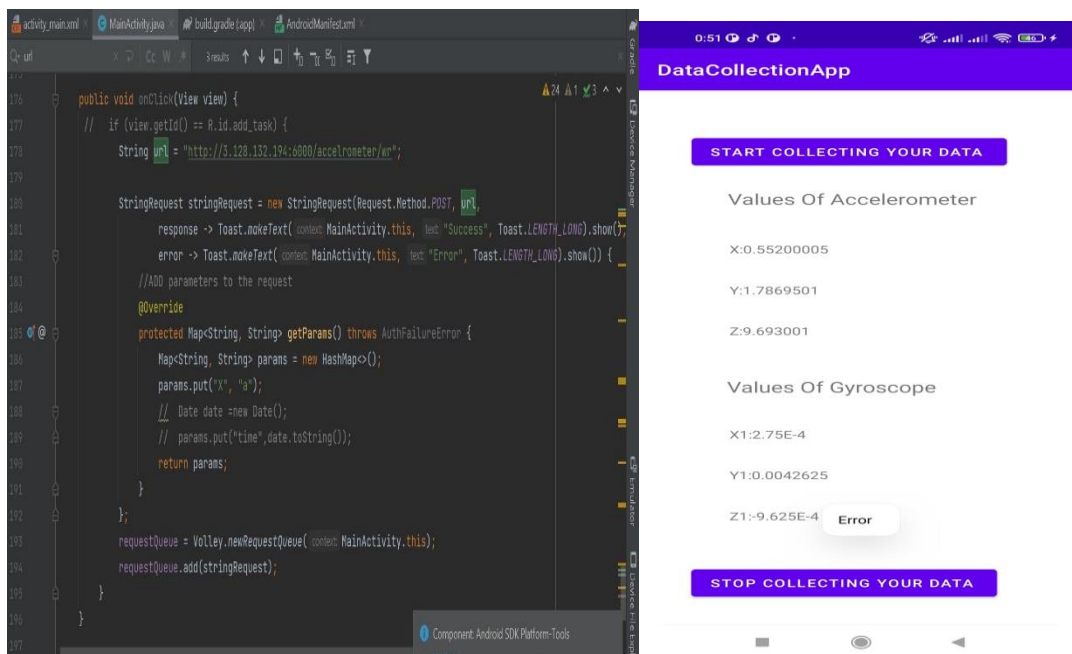
3- First, we tried to save data in MySQL by using such piece of code which did not work for any one of us, as we got connection failed error.



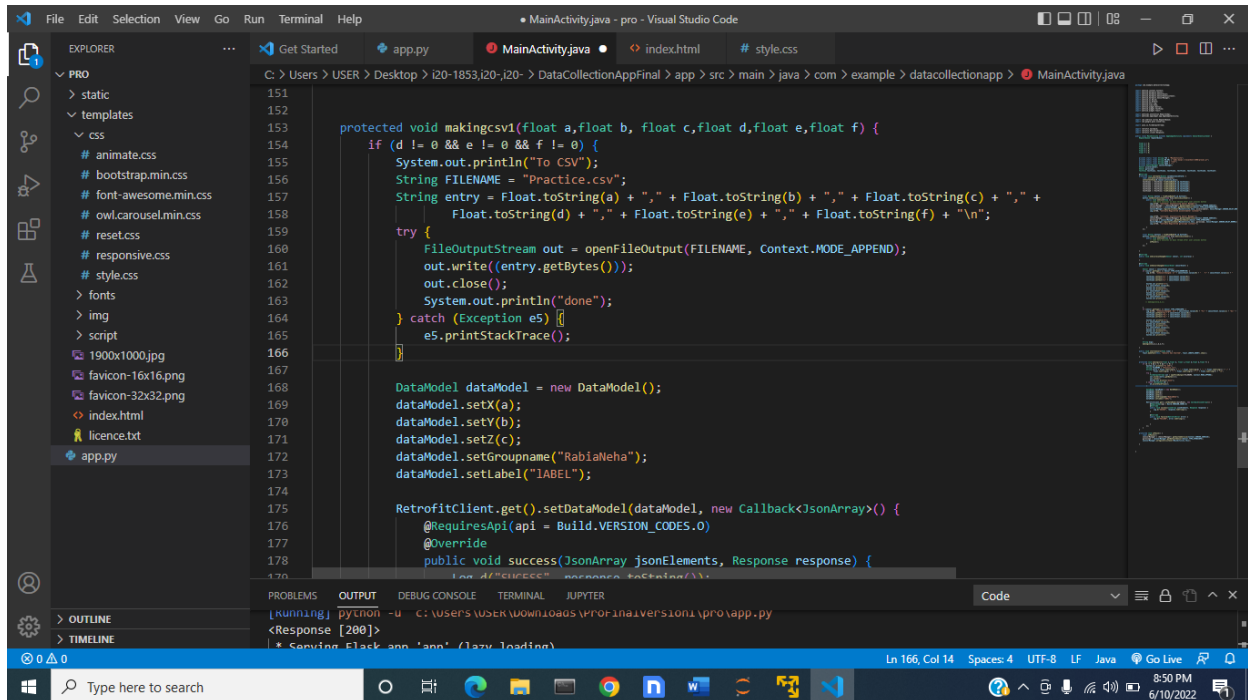
```
104     textView5.setText("Y:" + sensorEvent.values[1]);
105     textView6.setText("Z:" + sensorEvent.values[2]);
106     d = sensorEvent.values[0];
107     e = sensorEvent.values[1];
108     f = sensorEvent.values[2];
109
110
111     try {
112         Class.forName("net.sourceforge.jtds.jdbc.Driver");
113         Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/proj", user: "root", password: "1111");
114         if (conn == null) {
115             msg1 = "Connection Goes wrong";
116         } else {
117             String query = "INSERT INTO proj.pr(a1,b1,c1,d1,e1,f1) VALUES(a,b,c,d,e,f)";
118             Statement stmt = conn.createStatement();
119             stmt.executeUpdate(query);
120             msg1 = "Connection Successfull";
121             System.out.println(msg1);
122         }
123         conn.close();
124     }
125 }
126 catch (Exception e1) {
```

4- Then we started saving data in CSVs which were later used to train model.

5- Then we got a solution to send data in mongo DB as you can see in below image which also did not work due to connection issues.



6- Then one more solution we used which result useful after solved some issues. You can see code in below image.



The screenshot shows the Visual Studio Code editor with the file `MainActivity.java` open. The code is in Java and implements a method `makingcsv1` that checks if any of the input float values (a, b, c, d, e, f) are non-zero. If they are, it creates a CSV entry and writes it to a file named `Practice.csv`. The entry format is `a,b,c,d,e,f`. The code also initializes a `DataModel` object and sets its properties. The bottom status bar shows the file is at line 166, column 14.

```
151 protected void makingcsv1(float a,float b, float c,float d,float e,float f) {
152
153     if (d != 0 && e != 0 && f != 0) {
154         System.out.println("To CSV");
155         String FILENAME = "Practice.csv";
156         String entry = Float.toString(a) + "," + Float.toString(b) + "," + Float.toString(c) + "," +
157             Float.toString(d) + "," + Float.toString(e) + "," + Float.toString(f) + "\n";
158
159         try {
160             FileOutputStream out = openFileOutput(FILENAME, Context.MODE_APPEND);
161             out.write(entry.getBytes());
162             out.close();
163             System.out.println("done");
164         } catch (Exception e5) {}
165         e5.printStackTrace();
166     }
167
168     DataModel dataModel = new DataModel();
169     dataModel.setX(a);
170     dataModel.setY(b);
171     dataModel.setZ(c);
172     dataModel.setGroupname("RabiaNeha");
173     dataModel.setLabel("LABEL");
174
175     RetrofitClient.get().setDataModel(dataModel, new Callback<JSONArray>() {
176         @RequiresApi(api = Build.VERSION_CODES.O)
177         @Override
178         public void success(JSONArray jsonElements, Response response) {
179             // ...
180         }
181     });
182 }
```

Making CSV:

- 1. CSVs of all labels were made to train model by using csv data.**
- 2. Making csv code is declared in java file, that first checking whether gyroscope values are not zero, if values are not zero then sending data to mongo dB and make csv of coming data in mobile internal storage.**

```
146
147
148 protected void makingcsv1() {
149     if (d != 0 && e != 0 && f != 0) {
150         System.out.println("To CSV");
151
152         String FILENAME = "Phone in Use.csv";
153         String entry = Float.toString(a) + "," + Float.toString(b) + "," + Float.toString(c) + "," +
154             Float.toString(d) + "," + Float.toString(e) + "," + Float.toString(f) + "\n";
155
156         try {
157             FileOutputStream out = openFileOutput(FILENAME, Context.MODE_APPEND);
158             out.write((entry.getBytes()));
159             out.close();
160             System.out.println("done");
161         } catch (Exception e5) {
162             e5.printStackTrace();
163         }
164     }
165 }
166
167
```

3. Then downloading csv file from internal storage of mobile phone and store it for later use.

```
146
147
148 protected void makingcsv1() {
149     if (d != 0 && e != 0 && f != 0) {
150         System.out.println("To CSV");
151
152         String FILENAME = "Phone in Use.csv";
153         String entry = Float.toString(a) + "," + Float.toString(b) + "," + Float.toString(c) + "," +
154             Float.toString(d) + "," + Float.toString(e) + "," + Float.toString(f) + "\n";
155
156         try {
157             FileOutputStream out = openFileOutput(FILENAME, Context.MODE_APPEND);
158             out.write((entry.getBytes()));
159             out.close();
160             System.out.println("done");
161         } catch (Exception e5) {
162             e5.printStackTrace();
163         }
164     }
165 }
166
167
```

Training Of Machine Learning Models:

All the data that was collected through the mobile app was in the form of csv. First of all, all the data was cleaned, Labels were added and then we trained the machine learning Models on the data. We

trained two ML models (KNN and Naïve Bays). Naïve Bays model was giving accuracy of almost 85% and KNN was giving Accuracy 91% So we choose KNN model to predicts the labels Of Live Data.

Naive Bays

```
[52]: # Naive Bays
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
pred=gnb.predict(X_test)
gnb.score(X_test,y_test)
print ("Accuracy : " , accuracy_score(y_test,pred)*100)
```

Accuracy : 85.52143739899229

```
#KNN
from sklearn.neighbors import KNeighborsClassifier
clf1=KNeighborsClassifier(7)
clf1=clf1.fit(X_train, y_train)
pred=clf1.predict(X_test)
print ("Accuracy : " , accuracy_score(y_test,pred)*100)
```

Accuracy : 91.41553379598821

Kafka Consumer Producer:

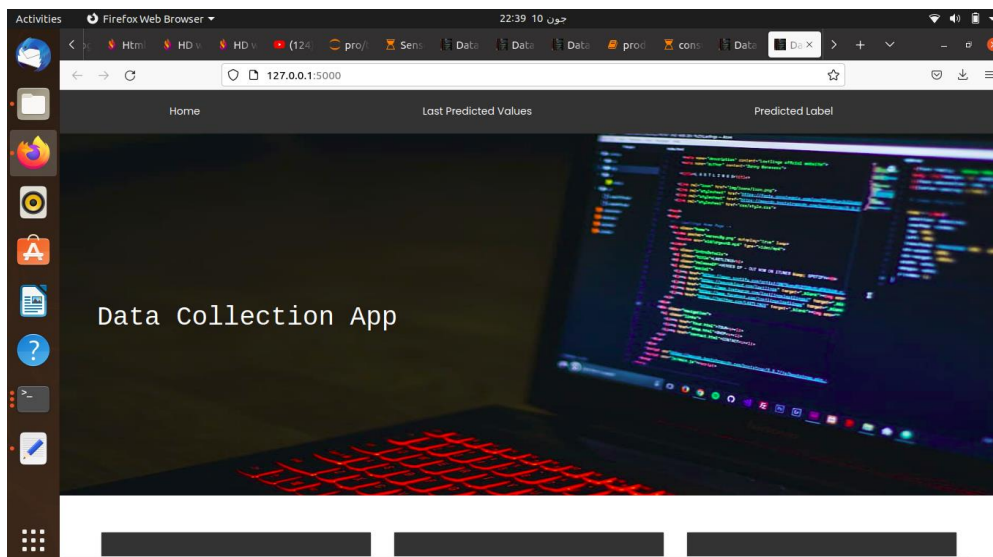
1. In Kafka producer, one csv is made from all 5 CSVs of all labels.
2. Then sending this csv to consumer.
3. Kafka consumer accept data frame from producer and use trained KNN model that is stored in file which is giving 91% accuracy.

4. We made another python files as sensors producer and sensors consumer which is used for predicting the labels at run time as if we train model at run time, it will decrease accuracy.
5. In sensors producer, we are taking live data through API by using request module and sending this to sensors consumer.
6. In sensors consumer, trained model is loaded by using Pickle library and then predict label.
7. Predicted label is then displayed on the website using Flask.

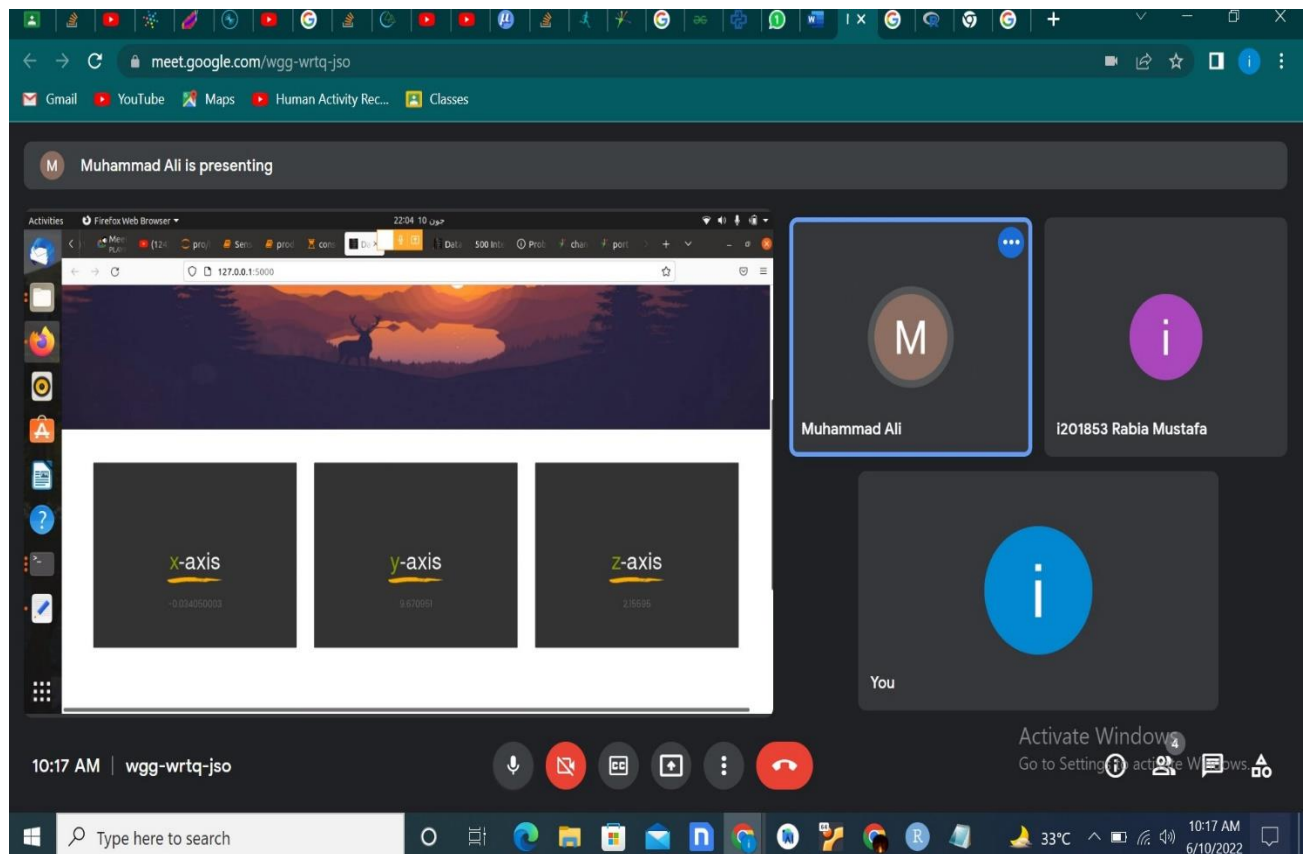
Connection With Flask:

<https://www.free-css.com/free-css-templates>

We have updated data on our website through flask. The label was stored in a variable after being predicted. Images were stored in variables before being sent through Flask.



This is the front page of our website



INDIVIDUAL LEARNING OUTCOME

RABIA MUSTAFA:

Through this project I learned how we can make a mobile application. And how we can add Sensors in a mobile app and then how we can transfer the Live data of these Sensors to API's as well as in CSV. Other Than that, it was a good practice of training the Machine Learning Models on the data that was collected through the sensors. And then Implementation of these ML models in Kafka (Consumer and producer) Clarify a lot of concepts about Kafka.

NEHA SHABBIR:

While working on this project, I was thinking that what we are getting from all this procedure as we have given all things on internet like sensors data is available on many links, we can get mongo dB and Kafka skills by simple programs but after finished our project now my perspective is totally different. We have got such skills which cannot get by simple procedures, the struggle of sending data in someone else mongo DB, making CSVs after solving tons of errors, retrieving CSV, collecting data from sensors, sending data from producer and accepting it from consumer in Kafka environment after solving difficult errors in ubuntu, all these things make me more confident in my field. I learned a lesson that if you work hard with great hope and effort then you can make impossible things possible.

MUHAMMAD AHSAN:

This project has helped me gain vital knowledge required in our field and learn how to work with my team mates. I learned how to send data to websites through Flask. However, the main problem I faced while working in this project was to solve this error:

Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

Which I was able to fix by freeing port:5000 I was able to learn more about HTML, CSS and Kafka which will greatly improve my ability to solve future problems in a time-effective manner.

The End