PROJECT TITLE: ANALYSIS OF GPS TRAJECTORIES USING (CLASSIFICATION AND CLUSTERING TECHNIQUES)

J COMPONENT

COURSE: DATAMINING TECHNIQUES – SWE2009

M.TECH (Software Engineering)

SUBMITTED BY:

S. No	NAME	REG. No
1)	B KRISHNA KARTHIKEYA	16MIS0158
2)	A VENKATA KIRAN	16MIS0192

UNDER THE GUIDENCE OF:

Prof. PRABAVATHY P

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering

April 2019

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled **“ANALYSIS OF GPS TRAJECTORIES”** submitted by me to VIT University, Vellore in partial fulfilment of the requirement for the award of the degree of **M.Tech(Software Engineering)** is a record of Jth component of project work carried out by me under the guidance of **Prof. PRABAVATHY P**. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: VELLORE

Date: 09/04/19

SUBMITTED BY:

B KRISHNA KARTHIKEYA

A VENKATA KIRAN

CONTENT:

TOPICS: **PAGE NO:**

1) INTRODUCTION:

GPS trajectories generated by moving objects provide researchers with an excellent resource for revealing patterns of human activities. Relevant research based on GPS trajectories includes the fields of location-based services, transportation science, and urban studies among others. Research relating to how to obtain GPS data (e.g., GPS data acquisition, GPS data processing) is receiving significant attention because of the availability of GPS data collecting platforms. One such problem is the GPS data classification based on transportation mode. The challenge of classifying trajectories by transportation mode has approached detecting different modes of movement through the application of several strategies. The advances in location-acquisition and mobile computing techniques have generated massive spatial trajectory data, which represent the mobility of a diversity of moving objects, such as people, vehicles and animals. Many techniques have been proposed for processing, managing and mining trajectory data. Recent advances in geo-positioning mobile phones have made it possible for users to collect a large number of GPS trajectories by recording their location information. these mobile phones with built-in GPS devices usually record far more data than needed, which brings about both heavy data storage and a computationally expensive burden in the rendering process for a web browser. By these smartphones, location-based services provide the potential to understand people's mobility pattern at an unprecedented level. How to discover people's personal mobility patterns is key phase for location-based services to provide high-quality service. Therefore, we focus on mining popular mobility patterns from user GPS trajectories. that means GPS trajectories collected based on one type of transportation mode are regarded as single-mode data; otherwise it is considered as mixed-mode data. The key to GPS trajectories classification based on the underlying transport mode is recognizing the transport mode of each GPS point, GPS segment, or an entire GPS trajectory.

2) LITERATURE SURVEY: -

2.1) Clustering:

1) Multispectral images segmentation for biomedical applications diagnosis:

K-means oriented approach

The segmentation of multispectral images is considered as a key step in image processing for biomedical applications. Performing this step using the appropriate methodology is a real issue that being investigated by the research community. In this paper, we propose a new algorithm to perform automatic segmentation based on kmeans methodology within an automatic generation of the optimal value of "K". We applied the new algorithm on a dataset of a real medical image. -[1] Bouzid-Daho

2) Primary cloud assessment in THEOS imagery using k-means clustering

and morphological transformation algorithms.

THEOS is the earth observation satellite system, which acquires earth images via its optical instruments. As its instruments consists of passive type CCD sensors, the instruments require sunlight reflected from the Earth surface for imaging. Then, cloud presence above imaging area directly affects the image usability, image interpretation, image classifying accuracy, calibration activity and so on. As THEOS mainly focuses on country needs, imaging over Thailand and Asian countries has been main priority since its launch. These countries are in the equatorial region and subjected to heavy cloud throughout the year. The results after implementing the cloud mask investigation indicates that this approach is capable of providing accurate cloud coverage assessment for THEOS images.-[2] Prayot Puangjaktha

3) Motion-based moving object detection and tracking using automatic K-means

Multiple objects detection and tracking are amongst the most important tasks in computer vision-based surveillance and activity recognition. This paper proposes a real-time multiple objects detection method and compares its performance with three existing methods. 'Good Features to Track' algorithm is used to extract feature points from each frame. Based on the motion-based information, feature points corresponding to moving objects are extracted from next frame. In terms of accuracy and efficiency, the proposed method is shown to be highly accurate in determining the number of moving objects and also fast in tracking them in the scene. - [3] Jules- Raymond

4) An effective method determining the initial cluster centres for K-means for clustering gene expression data.

Clustering is an important tool for analysing gene expression data. Many clustering algorithms have been proposed for the analysis of gene expression data. In this article we have clustered real life gene expression data via K-Means which is one of clustering algorithms. The comparison results show that the K-means algorithm which uses the proposed methods converges to better clustering results than other clustering algorithms. - [4] Deniz Tanır

5) A GPS data based distributed K-means for cabstand location selection.

Taxi has become an important component of public transportation system. A proper cabstand location can alleviate the traffic pressure. In this paper, a large set of global positioning system (GPS) data of taxi in Jinan City is employed to help locating the cabstand. By analyzing more than 300 million taxi driving data in Jinan, Shandong Province, the parallel K- means algorithm is applied on the cluster analysis based on Spark distributed computing framework. Although the results and conclusion are specific to Jinan City, the methods and models used in this paper can be employed on other cities as well.-[5] Wei Gui

2.2) Classification:

1) Fixture identification from aggregated hot water consumption data.

Activity identification in smart housing utilizes smart meters to label consumption of utilities, such as cold and hot water, into human activities, such as cooking and cleaning. Typical approaches utilize a large array of high sampling rate sensors installed at each fixture location. This high density-high sampling rate approach raises computational challenges due to the volume of data generated over time. In this paper, we present a novel approach for identifying water usage patterns using a sparse array of sensors. Their results show that our approach reduces the number of fixture level smart meters from 7 to 3, while achieving an average accuracy between 70% to 80% for identifying hot water fixtures used in the kitchen sink, bathroom sink and shower.- [6] Yan Gao

2) Software and machine learning tools for monitoring railway track switch performance

Trackside data logging hardware is often used in the UK, and increasingly elsewhere in the world, to record and transmit processed condition data from track switching equipment (points) in order to gauge asset health. This paper presents a novel implementation of three tools which can be used together to make the analysis and handling of this data easier. The first of these tools is a statistical classifier which automatically assigns labels to the process data. The classifier is trained using historical data containing examples of events of interest, such as recordings taken when maintenance activity or failures have developed. The final tool is a statistical technique which is used to extract simple features from the data and raise alarms if they indicate poor track switch performance. The effectiveness of these tools is tested using real world data taken from three different railways. – [7] N P Wright

3) Optimizing indoor location recognition through wireless fingerprinting at the Ian Potter Museum of Art

Indoor tracking of smartphones adds context to smartphone applications, enabling a range of smarter behaviours. The predicted use cases are many and varied, and include navigation, planning, advertising and communication. Potentially, indoor tracking could become as ubiquitous as GPS - however, all of these possibilities depend on being able to produce a reasonably accurate, reliable system which does not require specialised infrastructure. Large Random Forests (200 trees) were found to have the best performance, which was further improved by calibrating for average differences in RSSI between phone models, to achieve an average of 90% correct classification of exhibits within the top five hits.- [8] Işıl Karabey, Levent Bayındır.

4) Predicting Transportation Carbon Emission with Urban Big Data

Transportation carbon emission is a significant contributor to the increase of greenhouse gases, which directly threatens the change of climate and human health. Under the pressure of the environment, it is very important to master the information of transportation carbon emission in real time. In the traditional way, we get the information of the transportation carbon emission by calculating the combustion of fossil fuel in the transportation sector The results show that our method has advantages over the well-known three machine learning methods (Gaussian Naïve Bayes, Linear Regression, Logistic Regression) and two deep learning methods (Stacked Denoising Autoencoder, Deep Belief Networks). - [9] Xiangyong Lu

5) Comparing Classification Methods for Longitudinal fMRI Studies.

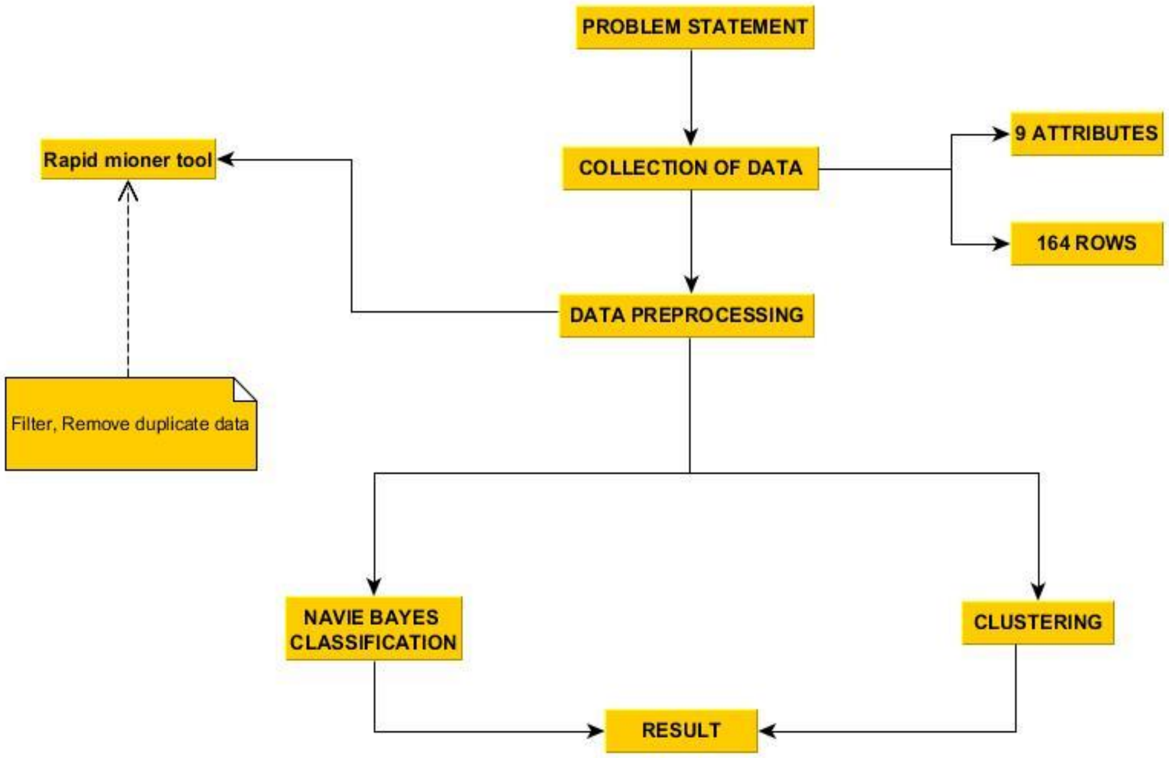
We compare 10 methods of classifying fMRI volumes by applying them to data from a longitudinal study of stroke recovery: adaptive Fisher's linear and quadratic discriminant; gaussian naive Bayes; support vector machines with linear, quadratic, and radial basis function (RBF) kernels; logistic regression; two novel methods based on pairs of restricted Boltzmann machines (RBM); and K-nearest neighbours. All methods were tested on three binary classification tasks, and their out-of-sample classification accuracies are compared. The relative performance of the methods varies considerably across subjects and classification tasks. The best overall performers were adaptive quadratic discriminant, support vector machines with RBF kernels, and generatively trained pairs of RBMs. - [10] Orhan Firat.

2.3) LITERATURE SUMMARY:

It gives a detailed analysis over the different applications over the gps trajectories. This gives the different analysis in cab stand analysis, rating of car or bus, moving object detection, monitoring railway track switch performance, determining the initial cluster centres for K-means. This helps in get the output using easy way for the given data. For example, Transportation agencies have an opportunity to leverage increasingly available trajectory datasets to improve their analyses and decision-making processes. It explores trajectory data from the perspective of a road transportation agency interested in acquiring trajectories to enhance its analysis. Systems analysis: demand estimation, modelling human behaviour, designing public transit, traffic performance measurement and prediction, environment, and safety.

3) SYSTEM DESIGN:

3.1) FRAMEWORK:



4) SYSTEM IMPLEMENTATION:

Tool used for execution of algorithms:

- Programming language-Python
- Tools – spyder, anaconda
- Algorithm- K-means, Guassian naïve bayes.

4.1) CODE:

K-MEANS:

```
import csv, math, random
import pandas as pd
from sklearn.naive_bayes import GaussianNB
import csv
import matplotlib.pyplot as plt
pd.set_option('display.max_rows',165)
x=pd.read_csv('go_track_tracks.csv').as_matrix()
df=pd.DataFrame(x)
data= x[1:30, 0:2]
target = x[1:30, 3]
df
from sklearn.cluster import KMeans
kmeans=KMeans(n_clusters=2)
kmeans
KMmodel=kmeans.fit(data)
KMmodel
KMmodel.labels_
KMmodel.cluster_centers_
pd.crosstab(target,KMmodel.labels_)
```

NAVIE – BAYES:

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB
data =pd.read_csv('go_track_tracks.csv').as_matrix()
clf = GaussianNB()
traindata = data[1:130, 0:4]
print(traindata)
trainLable = data[1:130, 4]
```

```
trainLable=trainLable.astype('int')
print(trainLable)
clf.fit(traindata, trainLable)
testData = data[130:160,0:4]
testlabel=data[130:160,4]
for i in range(1,30):
    print(testData[i] )
    print(testlabel[i])
    x=clf.predict([testData[i]])
    print(x)
```

4.2) TEST RESULTS:

1) By using k-means we get the results into the clusters like 0 and 1.
2) By using naïve bayes we get the results in the form of matrix of the taken data set according to the output.

K-MEANS CODE:

Python code:

Steps:

1) Read the dataset and assign the attributes to one variable & class label to another variable.
2) Partion the whole dataset into two, one for training & remaining for testing the data.
3) Find the unique class labels present in the database and length of the classes.
4) Calculate the total number of independent variables and the length of the test set.
5) All the above values are to be assigned to some variable.
6) Probability calculation using normal distribution, by calculating the
mean, standard deviation, and use normal cumulative distributive function
for the probability calculation.

7) Probability calculation using kernel distribution, where probability density estimates are used to calculated the probability of the set.

8) In the above two probabilities, consider the probability having more value, & use it for confusion matrix calculation.

9) Calculate the confusion matrix, and from the obtained matrix calculate the accuracy and error.

NAÏVE BAYES:

Python code

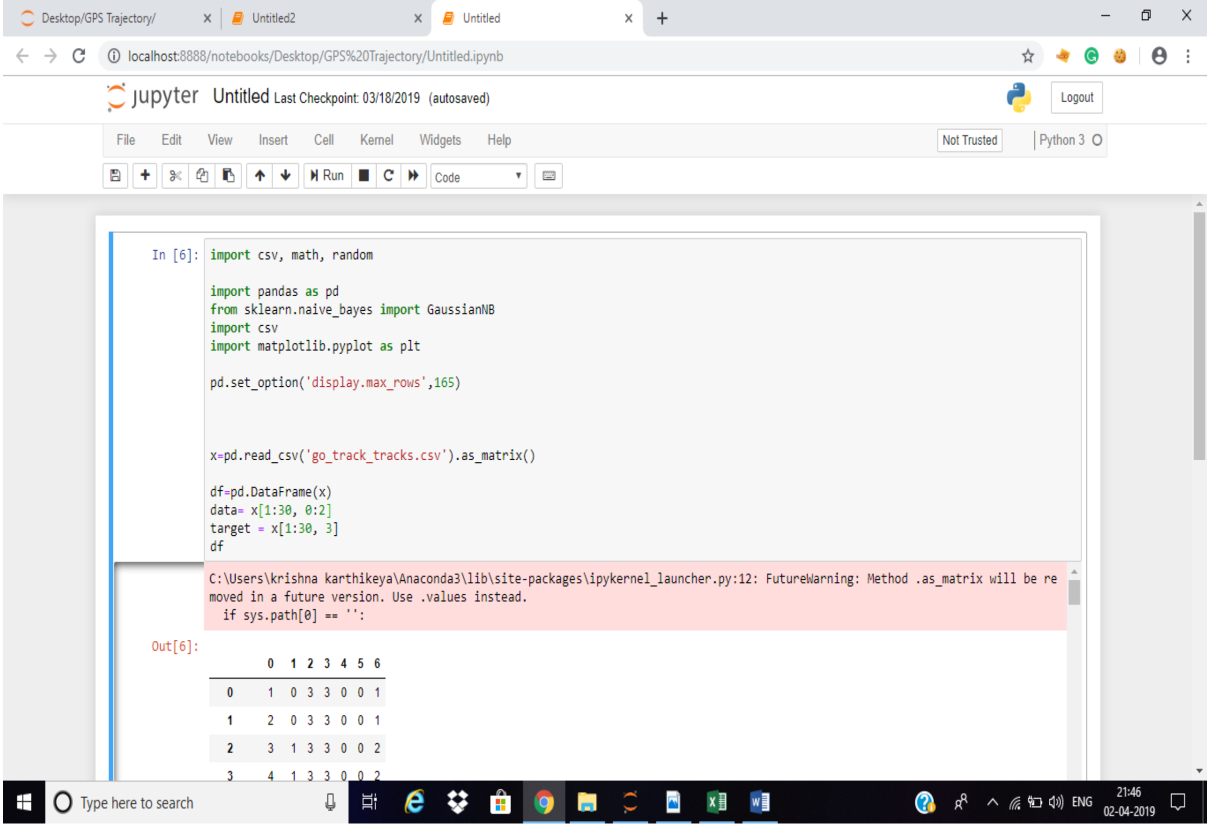
Steps:

1. Load the dataset and assign it to some variable.
2. Now plot the dataset using plot function just to visualize the dataset and to get a raw view of the data points.
3. Now use inbuilt, k-means function by choosing no of clusters is three.
4. Now plot the graph by shading the overall grid into k clusters (3-clusters) region, so that we can see each cluster boundary.
5. Now use centroid formula, to locate the centroid for each of the cluster, and also calculate the distance measures.
6. Finally, plot the centroids and visualize all the three clusters.

5) RESULTS AND DISCUSSION:

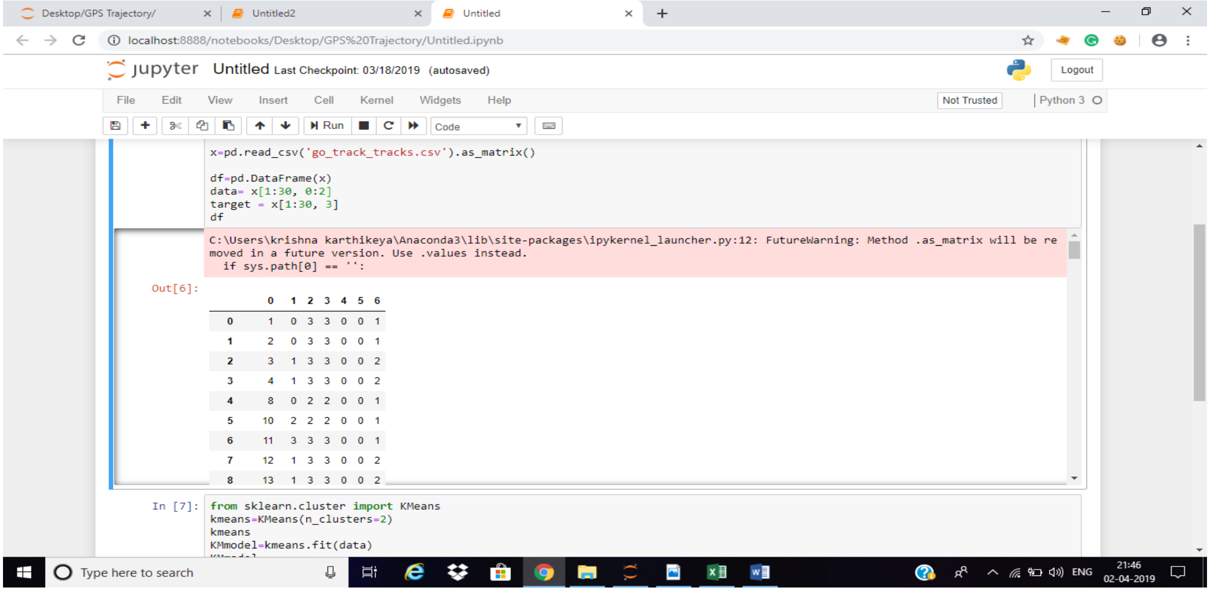
5.1) RESULTS AND OUTPUT:

K-MEANS:


Desktop/GPS Trajectory/
Untitled2
Untitled
localhost:8888/notebooks/Desktop/GPS%20Trajectory/Untitled.ipynb
jupyter Untitled Last Checkpoint: 03/18/2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
Type here to search
ENG
21:46
02-04-2019

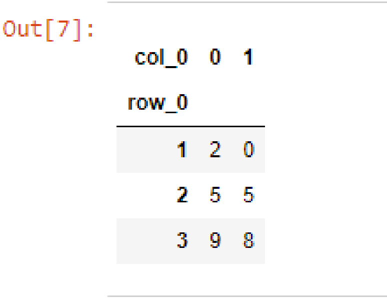
```
In [6]: import csv, math, random

        import pandas as pd
        from sklearn.naive_bayes import GaussianNB
        import csv
        import matplotlib.pyplot as plt

        pd.set_option('display.max_rows',165)



        x=pd.read_csv('go_track_tracks.csv').as_matrix()

        df=pd.DataFrame(x)
        data= x[1:30, 0:2]
        target = x[1:30, 3]
        df

C:\Users\krishna karthikeya\Anaconda3\lib\site-packages\ipykernel_launcher.py:12: FutureWarning: Method .as_matrix will be re
moved in a future version. Use .values instead.
  if sys.path[0] == '':

Out[6]:
       0  1  2  3  4  5  6
   0   1  0  3  3  0  0  1
   1   2  0  3  3  0  0  1
   2   3  1  3  3  0  0  2
   3   4  1  3  3  0  0  2
```

DATASET VISUALIZATION:

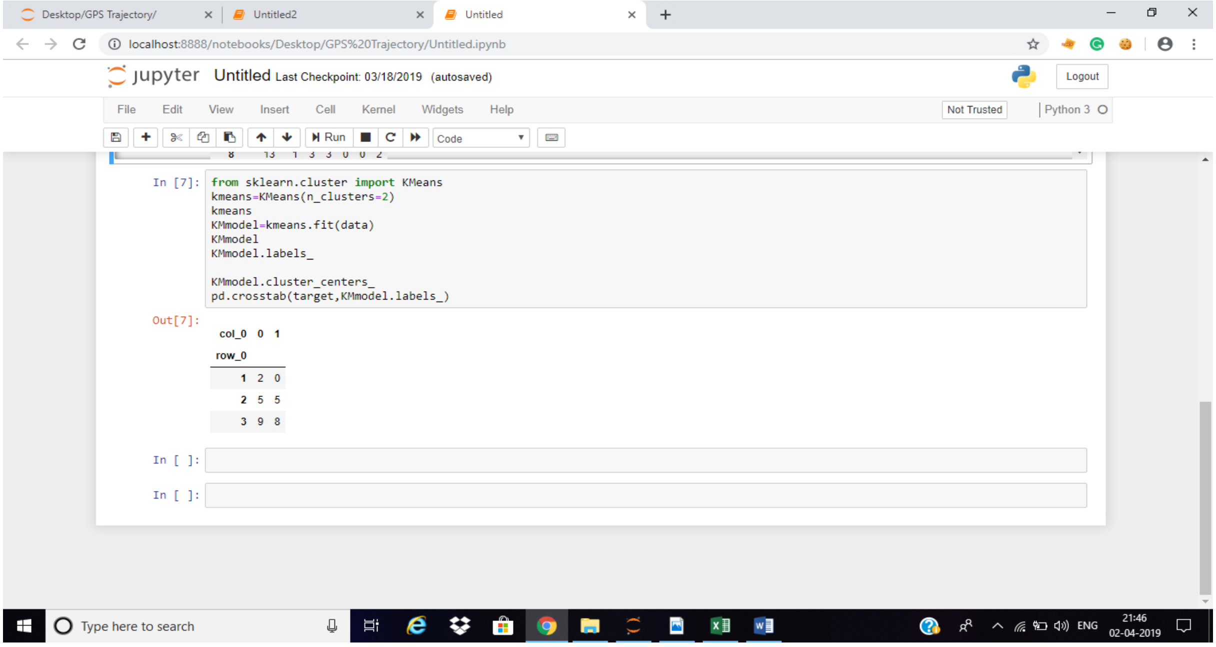

Desktop/GPS Trajectory/
Untitled2
Untitled
localhost:8888/notebooks/Desktop/GPS%20Trajectory/Untitled.ipynb
jupyter Untitled Last Checkpoint: 03/18/2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
Type here to search
ENG
21:46
02-04-2019

```
        x=pd.read_csv('go_track_tracks.csv').as_matrix()

        df=pd.DataFrame(x)
        data= x[1:30, 0:2]
        target = x[1:30, 3]
        df

C:\Users\krishna karthikeya\Anaconda3\lib\site-packages\ipykernel_launcher.py:12: FutureWarning: Method .as_matrix will be re
moved in a future version. Use .values instead.
  if sys.path[0] == '':

Out[6]:
       0  1  2  3  4  5  6
   0   1  0  3  3  0  0  1
   1   2  0  3  3  0  0  1
   2   3  1  3  3  0  0  2
   3   4  1  3  3  0  0  2
   4   8  0  2  2  0  0  1
   5  10  2  2  2  0  0  1
   6  11  3  3  3  0  0  1
   7  12  1  3  3  0  0  2
   8  13  1  3  3  0  0  2

In [7]: from sklearn.cluster import KMeans
        kmeans=KMeans(n_clusters=2)
        kmeans
        KMmodel=kmeans.fit(data)
```

CLUSTERED OUTPUT:

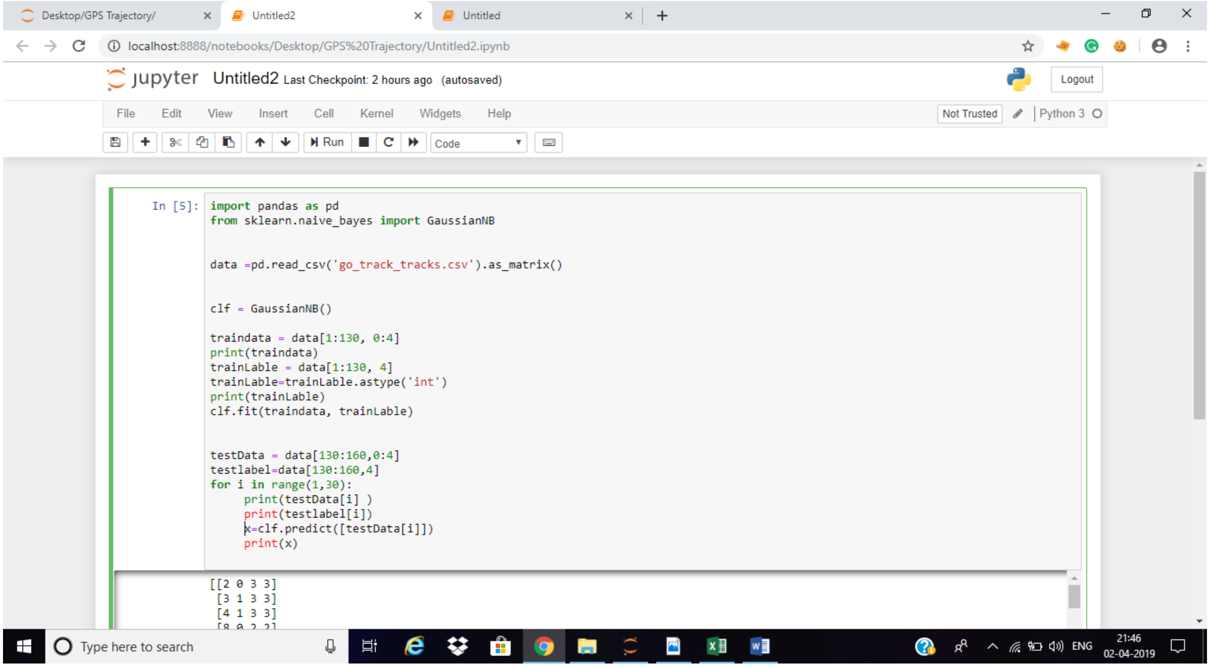


```
Out[7]:
          col_0  0  1
          row_0
              1  2  0
              2  5  5
              3  9  8
```



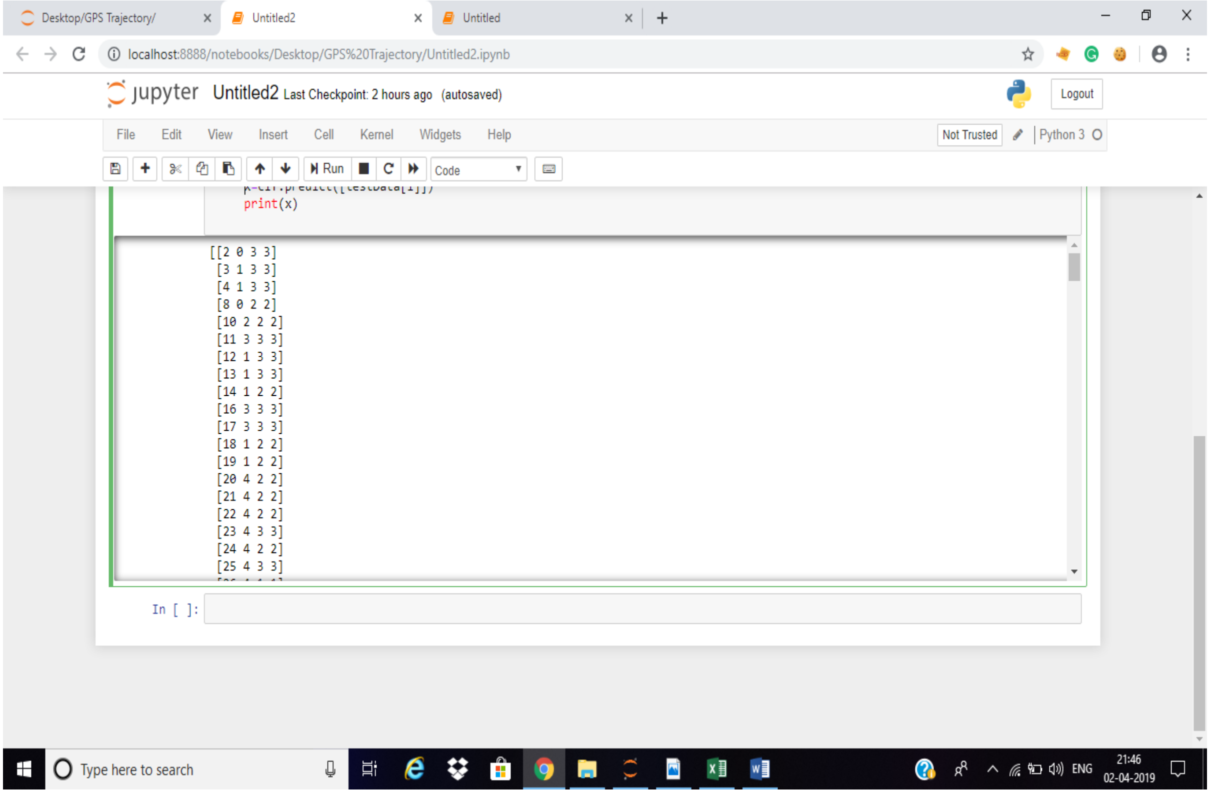
```
In [7]: from sklearn.cluster import KMeans
        kmeans=KMeans(n_clusters=2)
        kmeans
        KMmodel=kmeans.fit(data)
        KMmodel
        KMmodel.labels_

        KMmodel.cluster_centers_
        pd.crosstab(target,KMmodel.labels_)

Out[7]:
          col_0  0  1
          row_0
              1  2  0
              2  5  5
              3  9  8

In [ ]:

In [ ]:
```

NAÏVE BAYES:



```
In [5]: import pandas as pd
        from sklearn.naive_bayes import GaussianNB


        data =pd.read_csv('go_track_tracks.csv').as_matrix()


        clf = GaussianNB()

        traindata = data[1:130, 0:4]
        print(traindata)
        trainLable = data[1:130, 4]
        trainLable=trainLable.astype('int')
        print(trainLable)
        clf.fit(traindata, trainLable)


        testData = data[130:160,0:4]
        testlabel=data[130:160,4]
        for i in range(1,30):
            print(testData[i] )
            print(testlabel[i])
            x=clf.predict([testData[i]])
            print(x)
```

```
[[2 0 3 3]
 [3 1 3 3]
 [4 1 3 3]
```

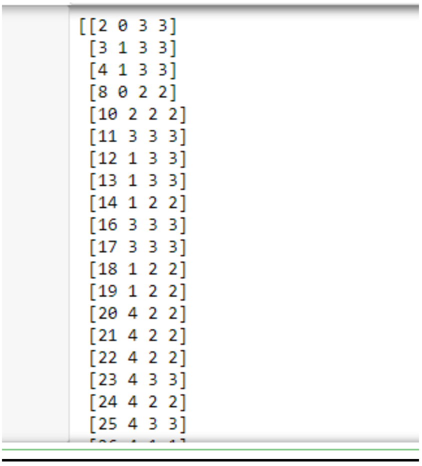


```
    print(x)
```

```
[[2 0 3 3]
 [3 1 3 3]
 [4 1 3 3]
```

```
 [8 0 2 2]
 [10 2 2 2]
 [11 3 3 3]
 [12 1 3 3]
 [13 1 3 3]
 [14 1 2 2]
 [16 3 3 3]
 [17 3 3 3]
 [18 1 2 2]
 [19 1 2 2]
 [20 4 2 2]
 [21 4 2 2]
 [22 4 2 2]
 [23 4 3 3]
 [24 4 2 2]
 [25 4 3 3]

In [ ]:
```

OUTPUT IN MATRIX FORM:



```
[[2 0 3 3]
 [3 1 3 3]
 [4 1 3 3]
 [8 0 2 2]
 [10 2 2 2]
 [11 3 3 3]
 [12 1 3 3]
 [13 1 3 3]
 [14 1 2 2]
 [16 3 3 3]
 [17 3 3 3]
 [18 1 2 2]
 [19 1 2 2]
 [20 4 2 2]
 [21 4 2 2]
 [22 4 2 2]
 [23 4 3 3]
 [24 4 2 2]
 [25 4 3 3]
```

:

5.2) DISCUSSION:

- Import all the required packages for classification and cluster.
- Import or open the dataset using pandas data frame.
- Train and test the dataset using the python built-in functions to predict the outcome.
- Then use confusion matrix and accuracy score functions to get the results.

6) CONCLUSION:

The data taken will be made into the clusters by using k -means clustering algorithm. This is done by taking dataset then like when two types of data like taking car and bus and give them based on rating etc. When we use naïve Bayes then we get to matrix by different columns. K-means algorithm here proposed can be used for real – world problems like carbon emission prediction in polluted provinces, and naïve Bayes algorithm can be used for weather prediction.

7) REFERENCES:

1) Prayot Puangjaktha Satellite Engineering Division, Geo-Informatics and Space Technology Development Agency (Public Organization), GISTDA, Bangkok, Thailand

2) Jules-Raymond Tapamo School of Engineering, University of KwaZulu-Natal, South Africa

3) Bouzid-Daho LERICA Laboratory, Faculty of Sciences of engineers, University Badji Mokhtar, Annaba, Algeria.

4) Wei Gui School of MS&E, Anhui Technology of University, Maanshan, China

5) Yan Gao Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY, USA 13699