**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JnanaSangama, Belagavi-590018, Karnataka**



**MINI PROJECT SYNOPSIS**

**on**

**ELECTRIC CAR RECOMMENDATION**

**Submitted by**

**USN Name**

**1BI17CS190 ANURAG KUMAR**

**1BI17CS191 RISHABH MISHRA**

Under the Guidance of

**Dr. Suneetha K R**

Associate Professor

Department of CS&E, BIT

Bengaluru-560004

****

**Department of Computer Science & Engineering**

**BANGALORE INSTITUTE OF TECHNOLOGY**

K.R. Road, V.V.Pura, Bengaluru-560 004

**2020-21**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”,** Belagavi-590018, Karnataka

**BANGALORE INSTITUTE OF TECHNOLOGY**

Bengaluru-560 004

****

**Department of Computer Science And Engineering**

***Certificate***

Certified that the Mini Project work entitled **“ELECTRIC CAR RECOMMENDATION”** carried out by

**USN NAME**

**1BI17CS190 ANURAG KUMAR**

**1BI17CS191 RISHABH MISHRA**

of V semester, Computer Science and Engineering branch as partial fulfillment of the course **Data Mining and Warehousing (17CS651)** prescribed by **Visvesvaraya Technological University, Belgaum** during the academic year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report.

The Mini Project report has been approved as it satisfies the academic requirements in respect of project work in Artificial Intelligence.

**Dr. Suneetha K R**

Associate Professor,

Department of CS&E, BIT,

Bengaluru-560004

**CONTENTS**

1. **INTRODUCTION** 
   1. Introduction 1-2

**2.** **PROBLEM STATEMENT**

2.1 Problem Statement 3

2.2 Objectives 3

**3.** **SYSTEM REQUIREMENTS**

3.1 Software Requirements 4

**4.** **ARCHITECTURE**

4.1 Architecture 5-6

**5.** **MODULE DESCRIPSIONS**

5.1 Module Description 7-9

**6. IMPLEMENTATION** **DETAILS**

6.1 Source Code 10-13

**7.** **RESULTS**  14-17

7.1 Output 1

**8. APPLICATIONS** 18

**9.** **CONCLUSION AND FUTURE WORK**

9.1 Conclusion 19

9.2 Future Work

**BIBLIOGRAPHY**

**CHAPTER 1**

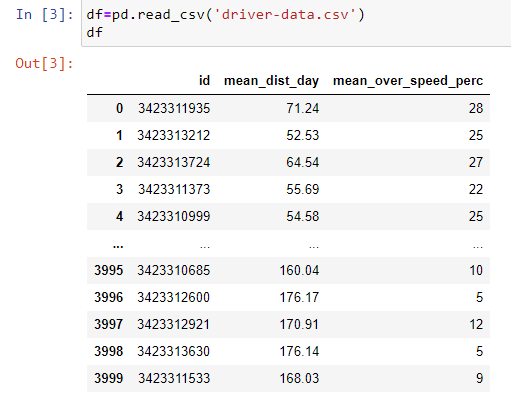
**INTRODUCTION**

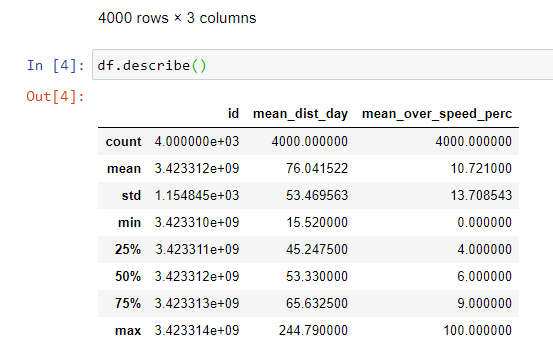
**1.1 Introduction**

**ELECTRIC CAR RECOMMENDATION**

* We refer to a dataset where there is a driver ID, mean distance and mean speed that a particular driver travels in a day.
* Then we divide the drivers into 6 clusters based on the mean distance and mean speed that they have travelled and plot a graph.
* Then after implementing the K-Means Clustering algorithm, the user can input the mean distance and mean speed to find out(predict) which cluster they belong to.
* The compiler then displays the respective cluster group number and then opens a window which displays the cars that belong to that cluster group such that the user can refer to these and buy a car accordingly.

ABOUT THE DATASET



****

**CHAPTER 2**

**PROBLEM STATEMENT**

**2.1 Problem Statement**

To segregate groups based on the mean distance and the mean speed using K-means Clustering algorithm and recommend a vehicle according to the cluster to which they belong.

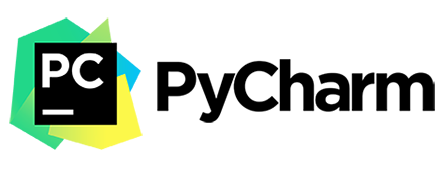
**2.2 Objectives**

* The primary objective of this work is to recommend a car based on the user model and item profile.
* Another objective is to prevent pollution, since electric cars are more eco-friendly!

**CHAPTER 3**

**SYSTEM REQUIREMENTS**

**3.1 Software requirements**

****

**PyCharm** is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), specifically for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) language. It is developed by the Czech company [JetBrains](https://en.wikipedia.org/wiki/JetBrains).[[6]](https://en.wikipedia.org/wiki/PyCharm#cite_note-6) It provides code analysis, a graphical debugger, an integrated unit tester, integration with [version control systems](https://en.wikipedia.org/wiki/Revision_control) (VCSes), and supports web development with [Django](https://en.wikipedia.org/wiki/Django_(web_framework)) as well as [Data Science](https://en.wikipedia.org/wiki/Data_science) with [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)).[[7]](https://en.wikipedia.org/wiki/PyCharm#cite_note-7)

PyCharm is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), with [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) versions. The Community Edition is released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License),[[8]](https://en.wikipedia.org/wiki/PyCharm#cite_note-community-8) and there is also Professional Edition with extra features – released under a [proprietary license](https://en.wikipedia.org/wiki/Proprietary_software).



Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

**CHAPTER 4**

**ARCHITECTURE**

**Clustering** is one of the most common exploratory data analysis technique used to get an intuitionabout the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

**Kmeans** algorithm is an iterative algorithm that tries to partition the dataset into*K*pre-defined distinctnon-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster’s centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

The way kmeans algorithm works is as follows:

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing.

Compute the sum of the squared distance between data points and all centroids. Assign each data point to the closest cluster (centroid).

Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

In the mini project the above algorithm has been used to divide the drivers based on their mean speed And mean distance into different subgroups or clusters. For each cluster a suitable electric car is getting recommended.

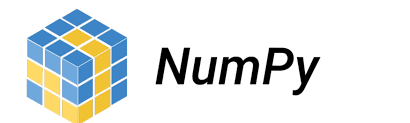
**CHAPTER 5**

**MODULE DESCRIPTIONS**

**5.1 Module Description**

**Python Libraries Used: •** Numpy

* Matplotlib
* Scikit-learn
* Pandas
* Opencv



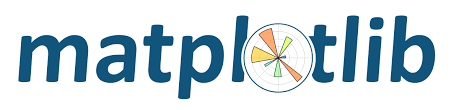
Numpy arrays are a special class of arrays that do these operations within milliseconds. These arrays are implemented in C programming language. In tasks like Natural Language Processing where you have a large set of vocabulary and hundreds of thousands of sentences, a single matrix can have millions of numbers. As a beginner, you have to master using this library.



In simple terms, [Pandas](https://www.edureka.co/blog/python-pandas-tutorial/) is the Python equivalent of **Microsoft Excel** . Whenever you have tabular data, you should consider using Pandas to handle it. The good thing about Pandas is that doing operations is just a matter of a couple of lines of code. If you want to do something complex, and you find yourself thinking about a lot of code, there is a high probability that there exists a Pandas command to **fulfill** **your wish in a line or two**.

****

**Scikit-learn** is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language.[[3]](https://en.wikipedia.org/wiki/Scikit-learn#cite_note-jmlr-3) It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [*k*-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy).

****

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.



OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source BSD license.

**CHAPTER 6**

**IMPLEMENTATION DETAILS**

**6.1 Source Code**

import pandas as pd

import numpy as np

import cv2

from os import listdir

from os.path import isfile, join

import matplotlib.pyplot as plt

df=pd.read\_csv('driver-data.csv')

print(df.head())

x=df.iloc[:,[1,2]].values

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

abc=[]

for i in range(2,10):

model=KMeans(n\_clusters=i)

model.fit(x)

abc.append(model.inertia\_)

plt.plot(range(2,10),abc)

plt.xlabel('number of clusters')

plt.ylabel('within cluster of square sum')

plt.title('elbow method')

plt.show()

model=KMeans(n\_clusters=6,random\_state=0)

model.fit(x)

y\_pred=model.predict(x)

y\_pred

a=float(input('Enter distance travelled per day:'))

b=float(input('Enter speed value:'))

c=model.predict([[a,b]])

if(c==0):

print('CLUSTER 0')

elif(c==1):

print('CLUSTER 1')

elif(c==2):

print('CLUSTER 2')

elif(c==3):

print('CLUSTER 3')

elif(c==4):

print('CLUSTER 4')

else:

print('CLUSTER 5')

#scatter plot for first cluster

plt.scatter(x[y\_pred==0,0],x[y\_pred==0,1],label='cluster 0',c='r')

#scatter plot for second cluster

plt.scatter(x[y\_pred==1,0],x[y\_pred==1,1],label='cluster 1',c='k')

#scatter plot for third cluster

plt.scatter(x[y\_pred==2,0],x[y\_pred==2,1],label='cluster 2',c='b')

#scatter plot for fourth cluster

plt.scatter(x[y\_pred==3,0],x[y\_pred==3,1],label='cluster 3',c='c')

#scatter plot for fifth cluster

plt.scatter(x[y\_pred==4,0],x[y\_pred==4,1],label='cluster 4',c='g')

#scatter plot for sixth cluster

plt.scatter(x[y\_pred==5,0],x[y\_pred==5,1],label='cluster 5',c='#FFF800')

plt.scatter(model.cluster\_centers\_[:,0],model.cluster\_centers\_[:,1],c='y',s=200,label='centroid')

plt.legend()

plt.xlabel('Distance travelled per day')

plt.ylabel('average speed')

plt.show()

if(c==0):

data\_path = 'Cluster 0/'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path,f))]

print(onlyfiles[0])

for i in onlyfiles:

img = cv2.imread(i)

cv2.imshow('recomended cars', img)

cv2.waitKey(0)

continue

elif(c==1):

data\_path = 'Cluster 1/'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path,f))]

print(onlyfiles[0])

for i in onlyfiles:

img = cv2.imread(i)

cv2.imshow('recomended cars', img)

cv2.waitKey(0)

continue

elif(c==2):

data\_path = 'Cluster 2/'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path,f))]

print(onlyfiles[0])

for i in onlyfiles:

img = cv2.imread(i)

cv2.imshow('recomended cars', img)

cv2.waitKey(0)

continue

elif(c==3):

data\_path = 'Cluster 3/'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path,f))]

print(onlyfiles[0])

for i in onlyfiles:

img = cv2.imread(i)

cv2.imshow('recomended cars', img)

cv2.waitKey(0)

continue

elif(c==4):

data\_path = 'Cluster 4/'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path,f))]

print(onlyfiles[0])

for i in onlyfiles:

img = cv2.imread(i)

cv2.imshow('recomended cars', img)

cv2.waitKey(0)

continue

else:

data\_path = 'Cluster 5/'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path, f))]

print(onlyfiles[0])

for i in onlyfiles:

img = cv2.imread(i)

cv2.imshow('recomended cars', img)

cv2.waitKey(0)

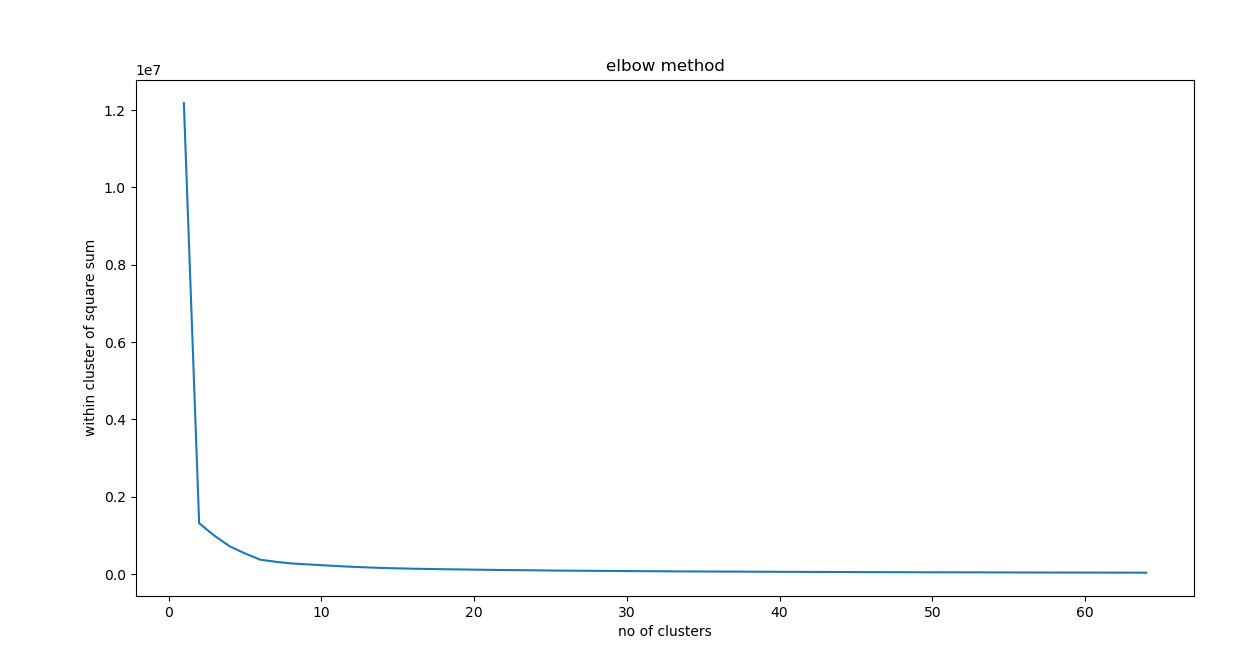
continue

**CHAPTER 7**

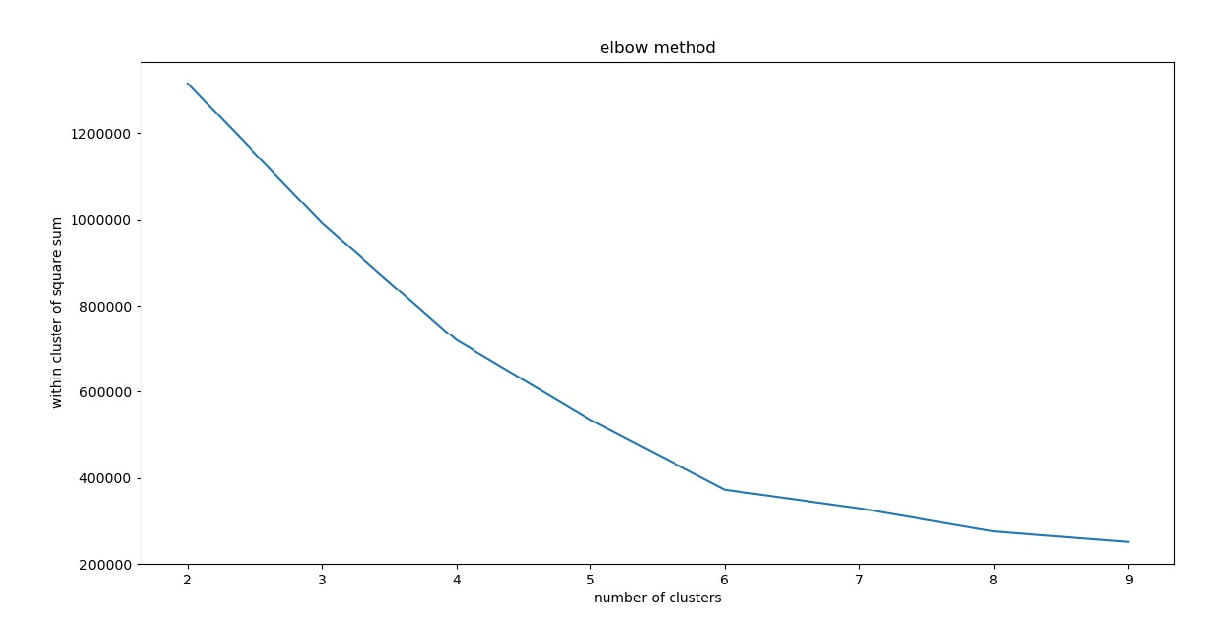
**RESULTS**

**7.1. Output 1**

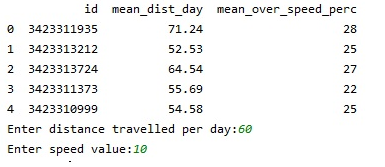
The first thing we need to do is dividing in the different number of clusters. For this, first we’ll find the value of k. The value of k from our dataset is approx. 64.



For precision, we shall take help of the elbow method. We find that there are 2 joints in the graph, at position 2 and 6. For more precision we shall take the range 2 to 10. Here we notice that there is a more clear joint at point 6,hence we will take 6 as the number of clusters.

****

Input for the mean distance and mean speed from the user

****

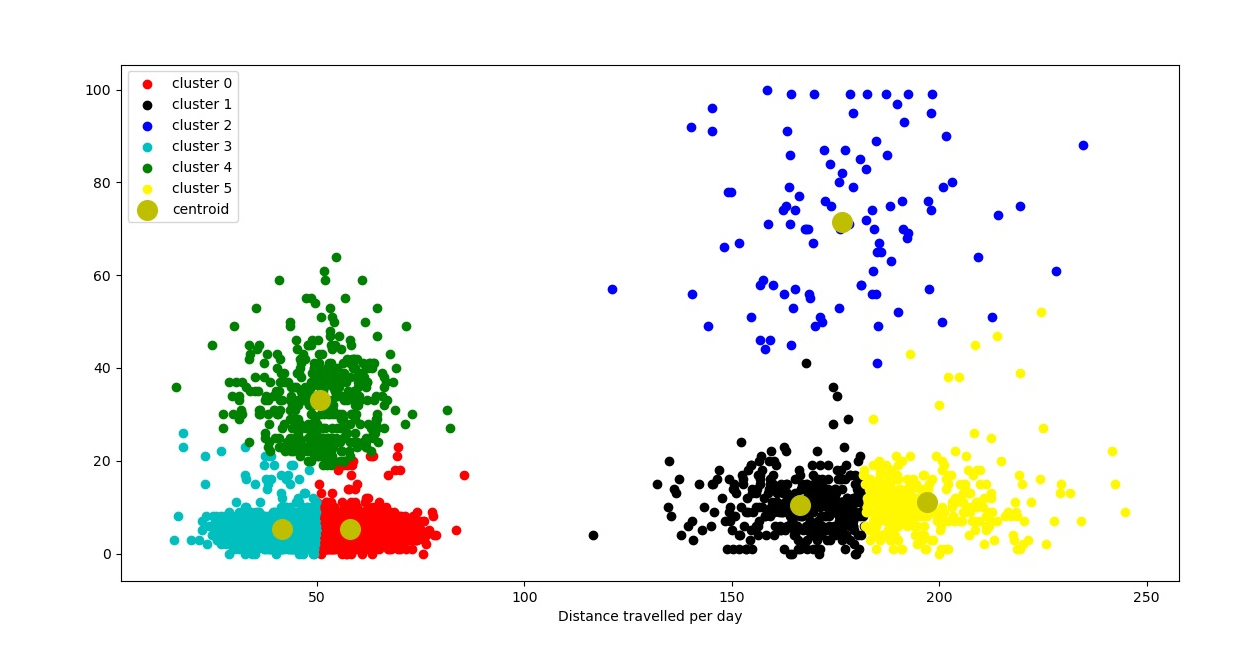
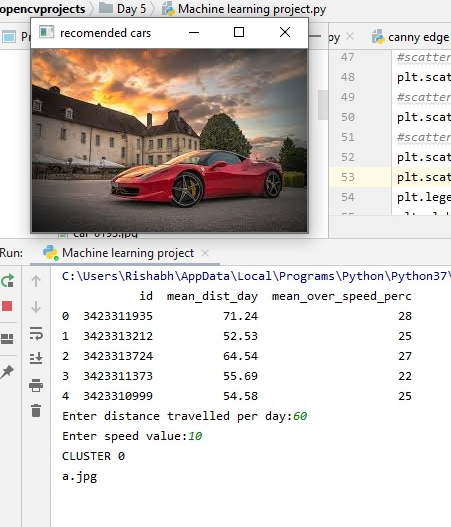
****

Fig. Cluster Graph

Recommendation (Output)

* We see that the user can input the distance travelled and the speed.
* That will open a window showing the cluster graph and also tells the respective cluster group.
* And finally, it opens another window which displays the recommended car. In this window we can press the enter key to view the other cars that belong to this cluster group so that the user can buy anyone among them.

****

**CHAPTER 8**

**APPLICATIONS**

* This project can be used by different drivers to select a suitable car for them.
* This project can be used by different car companies to produce different types of electric car based on the clusters that are formed.
* Another application is to prevent pollution, since electric cars are more eco-friendly!

**CHAPTER 9**

**CONCLUSION AND FUTURE ENHANCEMENT**

**9.1 Conclusion**

Overall we can say that this project is very useful for companies and drivers in their respective fields. For the companies this project is useful because they can understand the need of different types of customers and produce the cars accordingly. For drivers this project is like a boon because after a little analysis on mean distance and mean speed they can buy a suitable car for them as this project will suggest them the list of electric cars which are suitable for them using kmeans clustering algorithm. Here the algorithm has been used to classify the driver into 6 types or 6 clusters and for each cluster we have a list of cars. Here the elbow method has been successfully implemented to identify the k value which can be seen in the project screenshots as the graph bends near 6 indicating 6 clusters. Each cluster has been shown with a different color to distinguish among different clusters and finally the list of cars are shown.

**9.2 Future Enhancement**

The drawbacks that this project has can be seen as the drawbacks of kmeans clustering algorithm.

K-Means Disadvantages :

1) Difficult to predict K-Value.

2) With global cluster, it didn't work well.

3) Different initial partitions can result in different final clusters.

4) It does not work well with clusters (in the original data) of Different size and Different density.

Hence for the future enhancement all these disadvantages must be taken care of.

**BIBLIOGRAPHY**

The following websites are being used to take help-

* [www.youtube.com](http://www.youtube.com/)
* [www.edureka.co](http://www.edureka.co/)
* [https://www.geeksforgeeks.org](https://www.geeksforgeeks.org/)