

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JnanaSangama, Belagavi-590018, Karnataka



MINI PROJECT SYNOPSIS
on
ELECTRIC CAR RECOMMENDATION

Submitted by

USN

Name

1BI17CS190
1BI17CS191

ANURAG KUMAR
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

1.1 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 DESCRIPTIONS

5.1 Module Description	7-9
------------------------	-----

6. IMPLEMENTATION DETAILS

6.1 Source Code	10-13
-----------------	-------

7. RESULTS

7.1 Output 1	14-17
--------------	-------

8. APPLICATIONS

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

```
In [3]: df=pd.read_csv('driver-data.csv')
df
```

```
Out[3]:
```

	id	mean_dist_day	mean_over_speed_perc
0	3423311935	71.24	28
1	3423313212	52.53	25
2	3423313724	64.54	27
3	3423311373	55.69	22
4	3423310999	54.58	25
...
3995	3423310685	160.04	10
3996	3423312600	176.17	5
3997	3423312921	170.91	12
3998	3423313630	176.14	5
3999	3423311533	168.03	9

4000 rows × 3 columns

```
In [4]: df.describe()
```

Out[4]:

	id	mean_dist_day	mean_over_speed_perc
count	4.000000e+03	4000.000000	4000.000000
mean	3.423312e+09	76.041522	10.721000
std	1.154845e+03	53.469563	13.708543
min	3.423310e+09	15.520000	0.000000
25%	3.423311e+09	45.247500	4.000000
50%	3.423312e+09	53.330000	6.000000
75%	3.423313e+09	65.632500	9.000000
max	3.423314e+09	244.790000	100.000000

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 (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.^[6] It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda.^[7]

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License,^[8] and there is also Professional Edition with extra features – released under a proprietary license.



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 intuition about 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 distinct non-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

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 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 machine learning library for the Python programming language.^[3] It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k -means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and 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='#FF8000')
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

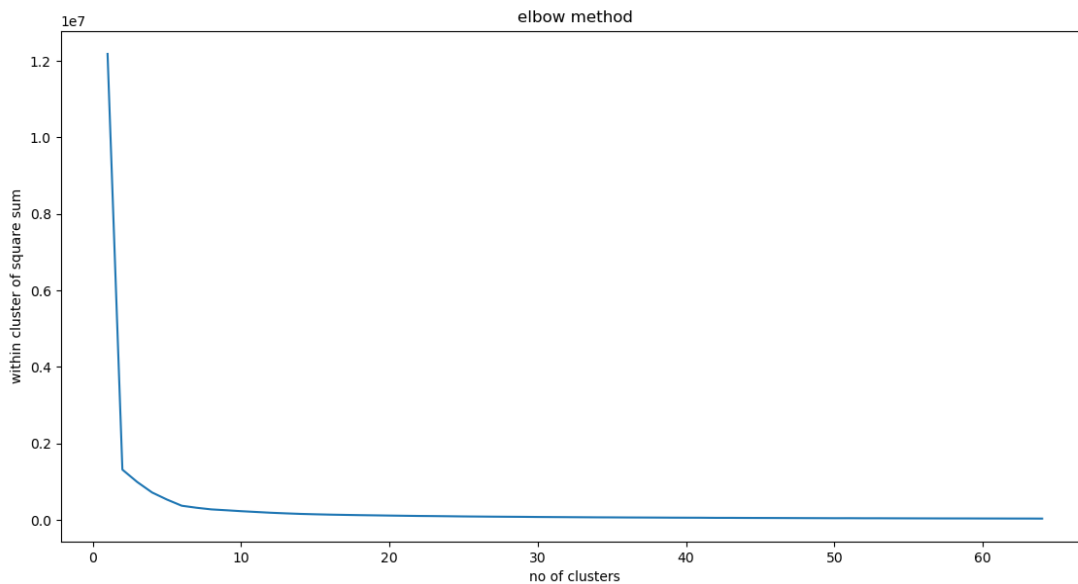
```


CHAPTER 7

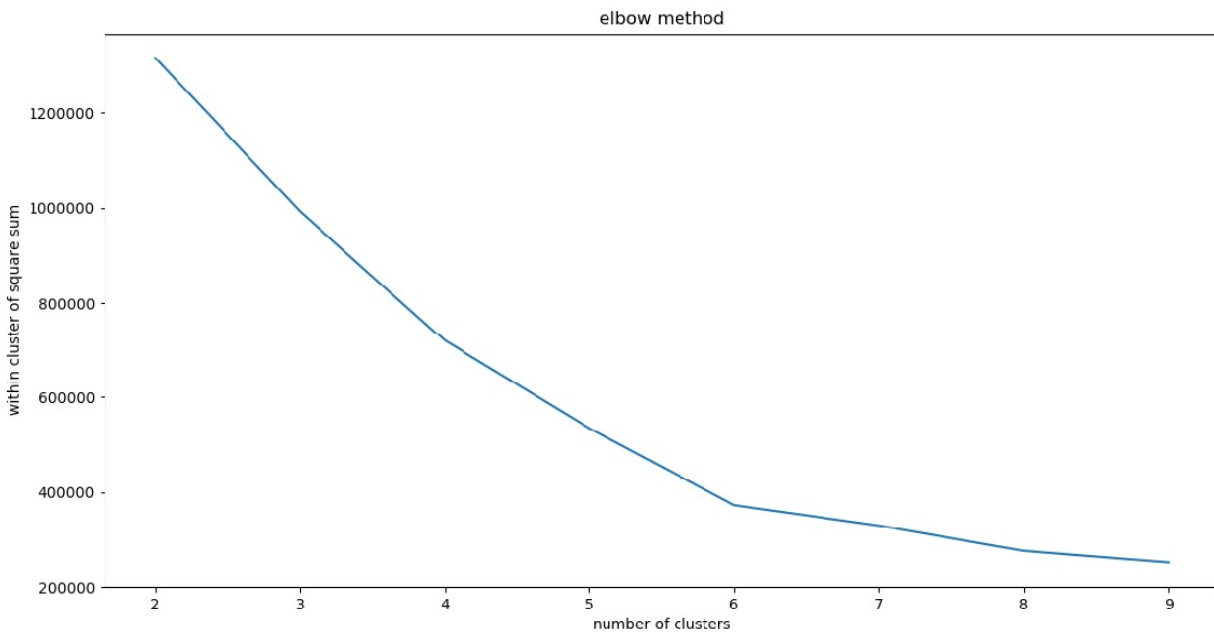
RESULTS

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

	id	mean_dist_day	mean_over_speed_perc
0	3423311935	71.24	28
1	3423313212	52.53	25
2	3423313724	64.54	27
3	3423311373	55.69	22
4	3423310999	54.58	25

Enter distance travelled per day:60

Enter speed value:10

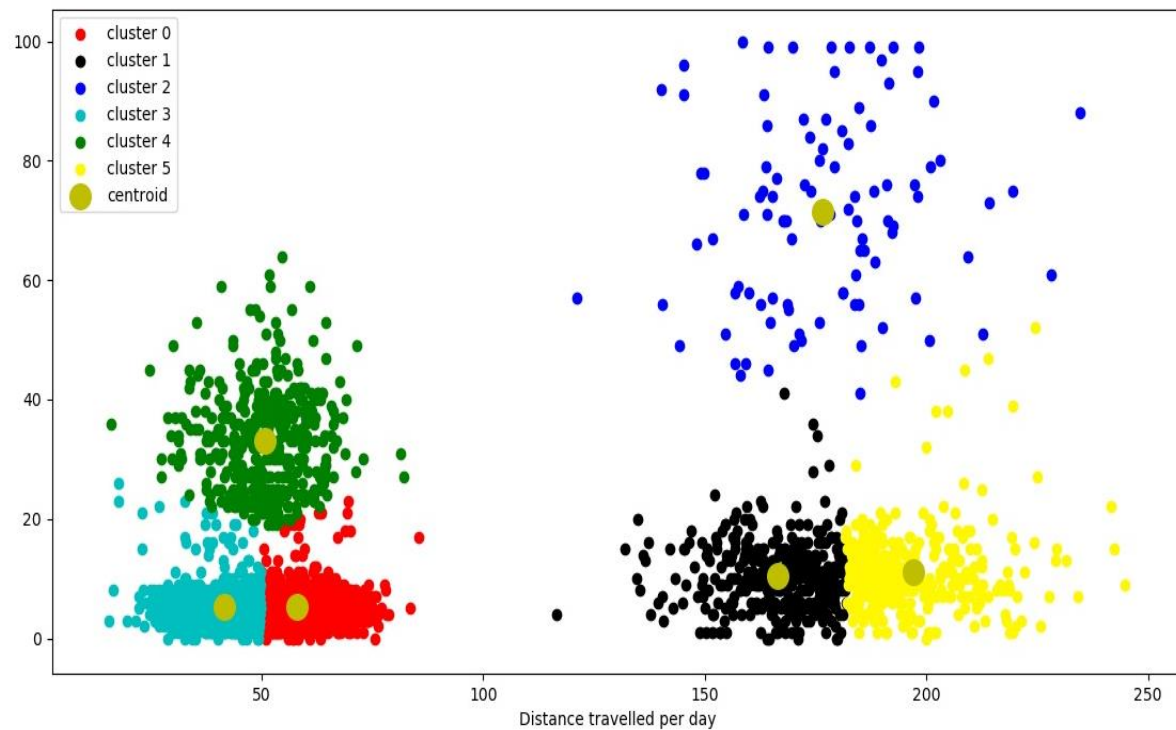



Fig. Cluster Graph

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.

opencvprojects
Day 5
Machine learning project.py

recomended cars


canny edge

```

47 #scatter
48 plt.scat
49 #scatter
50 plt.scat
51 #scatter
52 plt.scat
53 plt.scat
54 plt.lege

```

Run: Machine learning project

```

C:\Users\Rishabh\AppData\Local\Programs\Python\Python37\
      id  mean_dist_day  mean_over_speed_perc
0  3423311935         71.24             28
1  3423313212         52.53             25
2  3423313724         64.54             27
3  3423311373         55.69             22
4  3423310999         54.58             25
Enter distance travelled per day:60
Enter speed value:10
CLUSTER 0
a.jpg

```

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
- www.edureka.co
- <https://www.geeksforgeeks.org>