

# Major project report

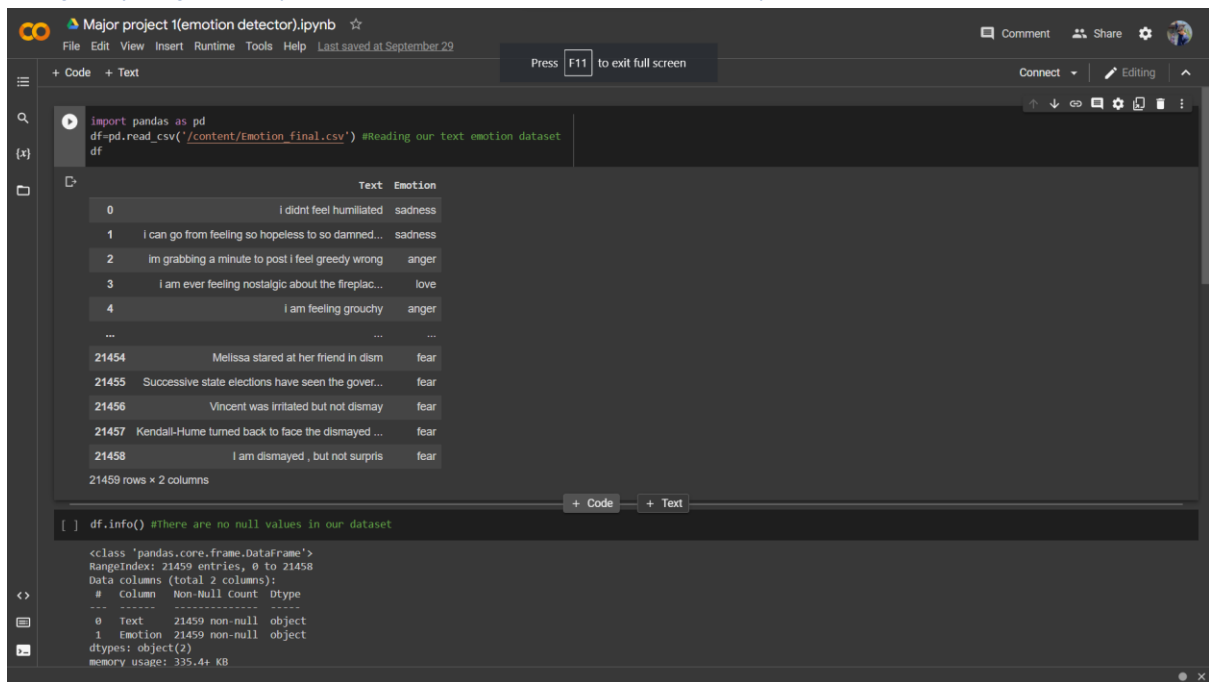
## Self introduction

Name: Sanyam Jain

College: Thakur college of engineering and technology

Year: ST

## Major project 1(Classifier-Emotion Detector)



The screenshot shows a Jupyter Notebook titled "Major project 1(emotion detector).ipynb". The code cell contains the following Python code:

```
import pandas as pd
df=pd.read_csv('/content/Emotion final.csv') #Reading our text emotion dataset
df
```

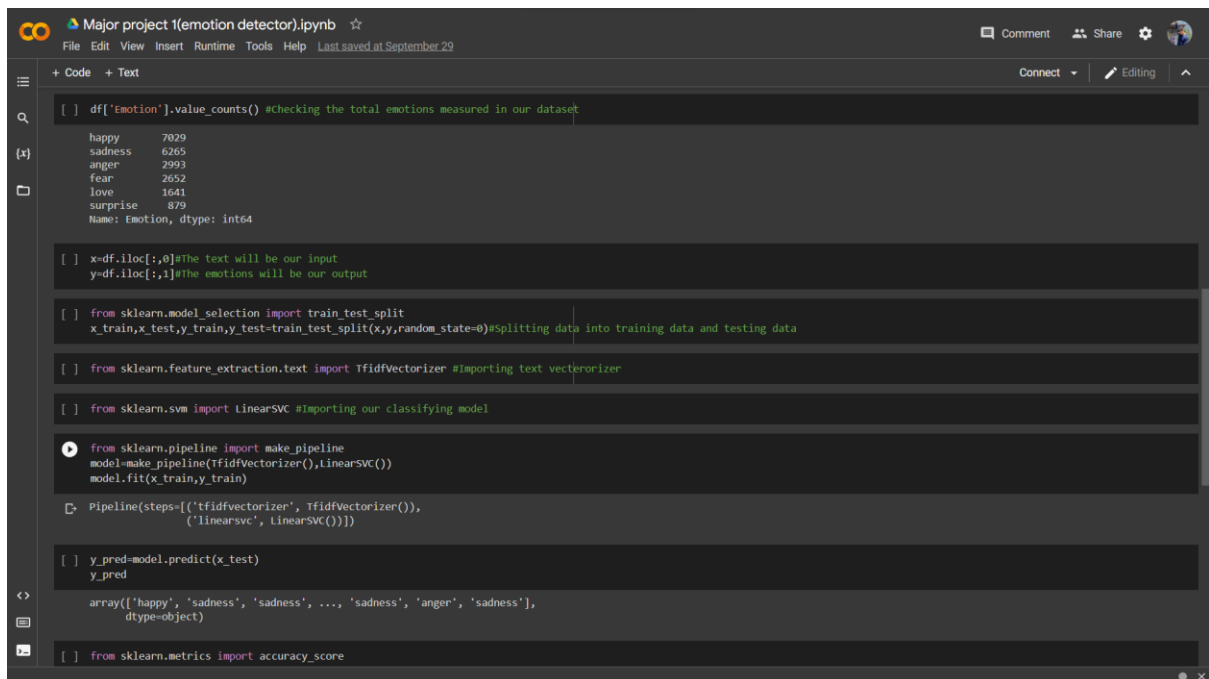
The output of the code is a preview of the dataset, showing columns "Text" and "Emotion". The preview displays rows 0 through 21458, with a total of 21459 rows and 2 columns.

	Text	Emotion
0	I didnt feel humiliated	sadness
1	I can go from feeling so hopeless to so damned...	sadness
2	im grabbing a minute to post I feel greedy wrong	anger
3	i am ever feeling nostalgic about the fireplac...	love
4	i am feeling grouchy	anger
...	...	...
21454	Melissa stared at her friend in dism	fear
21455	Successive state elections have seen the gover...	fear
21456	Vincent was irritated but not dismay	fear
21457	Kendall-Hume turned back to face the dismayed ...	fear
21458	I am dismayed , but not surpris	fear

The output also includes the result of `df.info()`, which shows that there are no null values in the dataset and provides details about the data types and memory usage.

```
[ ] df.info() #there are no null values in our dataset

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21459 entries, 0 to 21458
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  --
 0   Text    21459 non-null   object  
 1   Emotion  21459 non-null   object  
dtypes: object(2)
memory usage: 335.4+ KB
```



The screenshot shows the continuation of the Jupyter Notebook, focusing on data preprocessing and model training. The code cell contains the following Python code:

```
[ ] df['Emotion'].value_counts() #Checking the total emotions measured in our dataset

happy      7029
sadness    6265
anger      2993
fear       2652
love       1641
surprise    879
Name: Emotion, dtype: int64

[ ] x=df.iloc[:,0]#The text will be our input
y=df.iloc[:,1]#The emotions will be our output

[ ] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0)#splitting data into training data and testing data

[ ] from sklearn.feature_extraction.text import TfidfVectorizer #importing text vectorizer

[ ] from sklearn.svm import LinearSVC #importing our classifying model

[ ] from sklearn.pipeline import make_pipeline
model=make_pipeline(TfidfVectorizer(),LinearSVC())
model.fit(x_train,y_train)

[ ] Pipeline(steps=[('tfidfvectorizer', TfidfVectorizer()),
                    ('linearsvc', LinearSVC())])

[ ] y_pred=model.predict(x_test)
y_pred

array(['happy', 'sadness', 'sadness', ..., 'sadness', 'anger', 'sadness'],
      dtype=object)

[ ] from sklearn.metrics import accuracy_score
```

```
[ ] from sklearn.metrics import accuracy_score
    accuracy_score(y_pred,y_test)*100 #The accuracy of our model is 88.59%

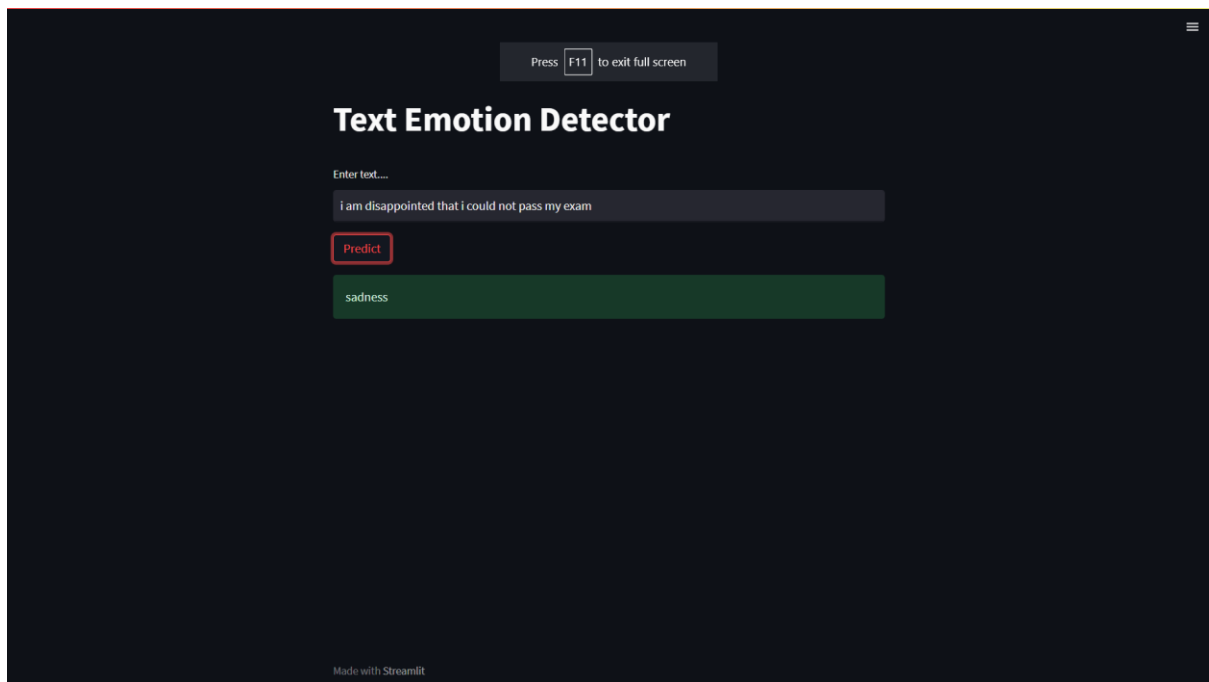
88.59273066169618

[ ] import joblib
    joblib.dump(model,'Emotion Detector') #Dumping our emotion detection model

['Emotion Detector']
```

This app has been deployed successfully on Heroku with the name emotion-detector-sanyam-jain.

Heroku was connected to my github with the repository for this project and was then automatically deployed using streamlit.



## Major project 1(Regressor – Mumbai house prices)

Major project 1(Mumbai house prices).ipynb

File Edit View Insert Runtime Tools Help Last saved at September 29

Press F11 to exit full screen

+ Code + Text

```
import pandas as pd
df=pd.read_csv('/content/Mumbai1.csv')#importing the house prices dataset
df
```

Unnamed: 0	Price	Area	Location	No. of Bedrooms	New/Resale	Gymnasium	Lift Available	Car Parking	Maintenance Staff	24x7 Security	Children's Play Area	Clubhouse	Intercom	Landscaped Gardens	Indoor Games	Gas Connection	Jog T
0	4850000	720	Kharghar	1	0	0	1	1	1	1	0	0	0	0	0	0	0
1	4500000	600	Kharghar	1	0	1	1	1	1	1	0	1	0	0	0	0	0
2	6700000	650	Kharghar	1	0	1	1	1	1	1	1	1	1	0	0	0	0
3	4500000	650	Kharghar	1	0	0	1	1	1	1	0	0	1	1	0	0	0
4	5000000	665	Kharghar	1	0	0	1	1	1	1	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6342	6342	2485000	700	Shirgaon	1	1	0	0	0	0	0	0	0	0	0	0	0
6343	6343	14500000	900	Thane West	2	0	0	0	0	0	0	0	0	0	0	0	0
6344	6344	14500000	900	Thane West	2	0	0	1	0	0	0	0	1	0	0	1	0
6345	6345	4100000	1380	Boisar	3	0	0	0	0	0	0	0	0	0	0	0	0
6346	6346	2750000	700	Badlapur East	1	1	1	1	1	0	1	1	0	1	1	0	0

6347 rows x 19 columns

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6347 entries, 0 to 6346
Data columns (total 19 columns):
 #   Column              Non-Null Count  Dtype
...
```

```
Major project 1(Mumbai house prices).ipynb
File Edit View Insert Runtime Tools Help Last saved at September 29

+ Code + Text
[ ]
4 No. of Bedrooms 6347 non-null int64
5 New/Resale 6347 non-null int64
6 Gymnasium 6347 non-null int64
7 Lift Available 6347 non-null int64
8 Car Parking 6347 non-null int64
9 Maintenance Staff 6347 non-null int64
10 24x7 Security 6347 non-null int64
11 Children's Play Area 6347 non-null int64
12 Clubhouse 6347 non-null int64
13 Intercom 6347 non-null int64
14 Landscaped Gardens 6347 non-null int64
15 Indoor Games 6347 non-null int64
16 Gas Connection 6347 non-null int64
17 Jogging Track 6347 non-null int64
18 Swimming Pool 6347 non-null int64
dtypes: int64(18), object(1)
memory usage: 942.3+ KB

df.isnull().sum()#checking if there are any null values in the dataset
Unnamed: 0 0
Price 0
Area 0
Location 0
No. of Bedrooms 0
New/Resale 0
Gymnasium 0
Lift Available 0
Car Parking 0
Maintenance Staff 0
24x7 Security 0
Children's Play Area 0
Clubhouse 0
Intercom 0
Landscaped Gardens 0
Indoor Games 0
Gas Connection 0
Jogging Track 0
Swimming Pool 0
dtype: int64
```

```
Major project 1(Mumbai house prices).ipynb
File Edit View Insert Runtime Tools Help Last saved at September 29

+ Code + Text
#getting the input values for the regressor model, including area
x=df.iloc[:,4:19]
x['Area']=df['Area']
x

No. of Bedrooms New/Resale Gymnasium Lift Available Car Parking Maintenance Staff 24x7 Security Children's Play Area Clubhouse Intercom Landscaped Gardens Indoor Games Gas Connection Jogging Track Swimming Pool Area
0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 720
1 1 0 1 1 1 1 1 0 1 0 0 0 0 1 1 600
2 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 650
3 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 650
4 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 665
... ..
6342 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 700
6343 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 900
6344 2 0 0 1 0 0 0 0 0 0 1 0 0 1 0 900
6345 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1380
6346 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 700
6347 rows x 16 columns

#getting the output values of price
y=df.iloc[:,1:2]
y

Price
0 4850000
1 4500000
```

```
Major project 1(Mumbai house prices).ipynb
File Edit View Insert Runtime Tools Help Last saved at September 29

+ Code + Text
Connect Editing

#Getting the output values of price
y=df.iloc[:,1:2]
y

Price
0 4850000
1 4500000
2 6700000
3 4500000
4 5000000
...
6342 2485000
6343 14500000
6344 14500000
6345 4100000
6346 2750000
6347 rows x 1 columns

[ ] from sklearn.model_selection import RepeatedKFold #RepeatedKFold is used to estimate the performance of our model
cv=RepeatedKFold(n_splits=10,n_repeats=3,random_state=1)

[ ] from sklearn.linear_model import LassoCV
from numpy import arange
model=LassoCV(alphas=arange(0,1,0.01),cv=cv,n_jobs=-1)
model.fit(x,y)
model.alpha_#The alpha value we get for our model

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_coordinate_descent.py:1571: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, 1) before passing to fit.

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_coordinate_descent.py:644: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations.
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_coordinate_descent.py:644: UserWarning: Coordinate descent with alpha=0 may lead to unexpected results and is discouraged.
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_coordinate_descent.py:644: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations.
0.99

[ ] test=[[3,1,0,1,1,1,1,0,1,1,0,1,0,0,1000]]#Test values with no. of bedrooms, parking availability, area,etc
model.predict(test)

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but LassoCV was fitted with feature names
"X does not have valid feature names, but"
array([18109310.9330524])

[ ] import joblib
joblib.dump(model,'Mumbai House Prices')#Dumping our created ML model

['Mumbai House Prices']
```

Following is the app deployed on Heroku using streamlit:

## Mumbai house price predictor

This model can predict the price of a house in Mumbai by accepting info such as no.

Select no. of bedrooms

1 3 10

Condition:

☒ New  
☐ Resale

Gym:

☐ Yes  
☒ No

Lift available:

☒ Yes  
☐ No

Parking available:

☒ Yes  
☐ No

Maintenance staff:

☒ Yes  
☐ No

24x7Security:

24x7Security:  
☒ Yes  
☐ No

Play area:  
☒ Yes  
☐ No

Clubhouse:  
☐ Yes  
☒ No

Intercom:  
☒ Yes  
☐ No

Landscaped garden:  
☒ Yes  
☐ No

Indoor games:  
☐ Yes  
☒ No

Gas connection:  
☒ Yes  
☐ No

Jogging track:  
☐ Yes  
☒ No

Swimming pool:

☐ Yes  
☒ No

Gas connection:  
☒ Yes  
☐ No

Jogging track:  
☐ Yes  
☒ No

Swimming pool:  
☐ Yes  
☒ No

Select the area  
1010  
200 9000

Predict

Price of the home in rupees is:  
18366199.208116412

Made with Streamlit

## Major project 2(Clustering – Star Clusters)

Following is the code for this project:

```
Major project 2(star clusters).ipynb
File Edit View Insert Runtime Tools Help Last edited on September 25

+ Code + Text

#Applying clustering on a data set

[ ] #importing all required libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import MiniBatchKMeans
from sklearn.metrics import silhouette_score

#Reading the data - star clusters data
df=pd.read_csv('/content/star_clusters.csv')
df
```

	x	y	z	vx	vy	vz	m	id
0	0.485939	-0.524359	-0.531982	0.461539	-0.033776	-0.322765	0.000016	1
1	-0.065961	0.080844	-0.276031	-0.575780	1.107815	-0.293408	0.000016	2
2	-0.034809	0.076795	-0.390880	-0.553991	-0.173861	0.059251	0.000016	3
3	1.502104	1.442983	1.449747	-0.090266	0.326612	-0.310594	0.000016	4
4	-0.009554	0.538341	-0.267264	0.011855	-0.011874	-0.405565	0.000016	5
...	...	...	...	...	...	...	...	...
63995	-0.515370	0.834457	1.609291	0.317309	0.082150	-0.514357	0.000016	63996
63996	0.725611	-0.579058	0.973140	0.129194	-0.207819	0.259824	0.000016	63997
63997	-0.175569	-0.397390	0.163967	-0.137939	0.234582	-0.060062	0.000016	63998
63998	0.239999	0.625631	0.715540	-0.148245	0.351681	0.784334	0.000016	63999
63999	-0.158667	-1.150932	0.237001	-0.396418	-0.076439	-0.445377	0.000016	64000

64000 rows x 8 columns

```
Major project 2(star clusters).ipynb
File Edit View Insert Runtime Tools Help Last edited on September 25

+ Code + Text

#checking if there are any null values, there are none
df.info()

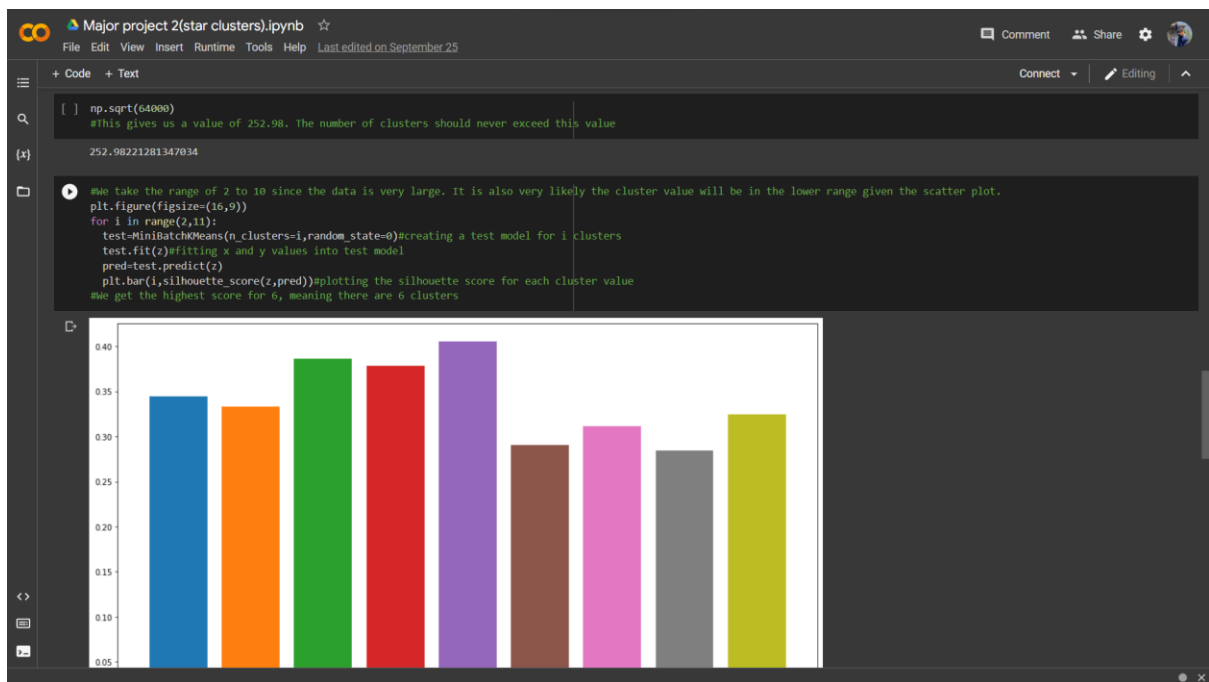
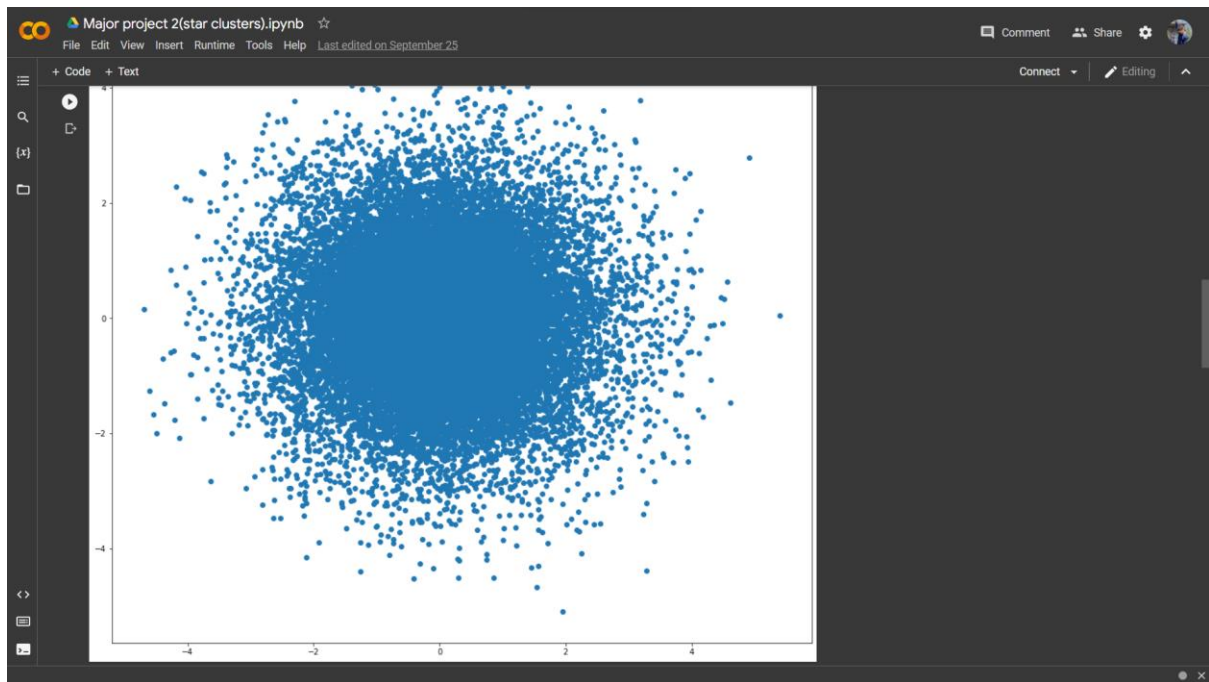
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64000 entries, 0 to 63999
Data columns (total 8 columns):
 #   column  Non-Null Count  Dtype
---  -
 0   x       64000 non-null     float64
 1   y       64000 non-null     float64
 2   z       64000 non-null     float64
 3   vx      64000 non-null     float64
 4   vy      64000 non-null     float64
 5   vz      64000 non-null     float64
 6   m       64000 non-null     float64
 7   id      64000 non-null     int64
dtypes: float64(7), int64(1)
memory usage: 3.9 MB

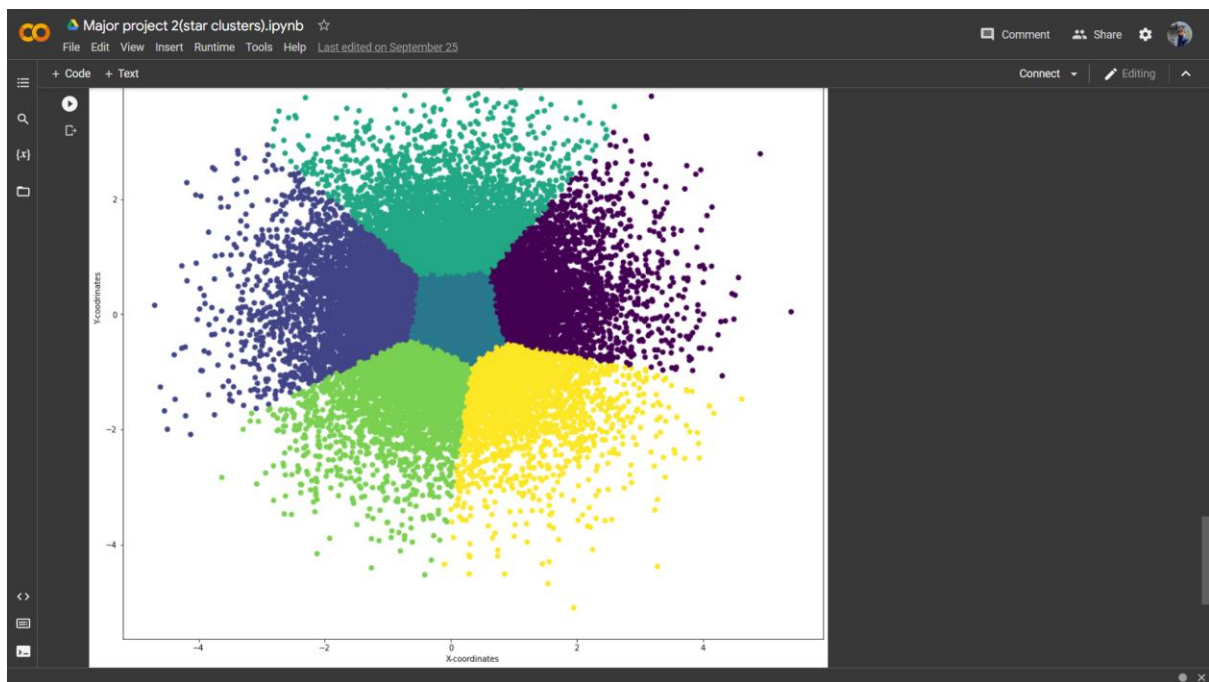
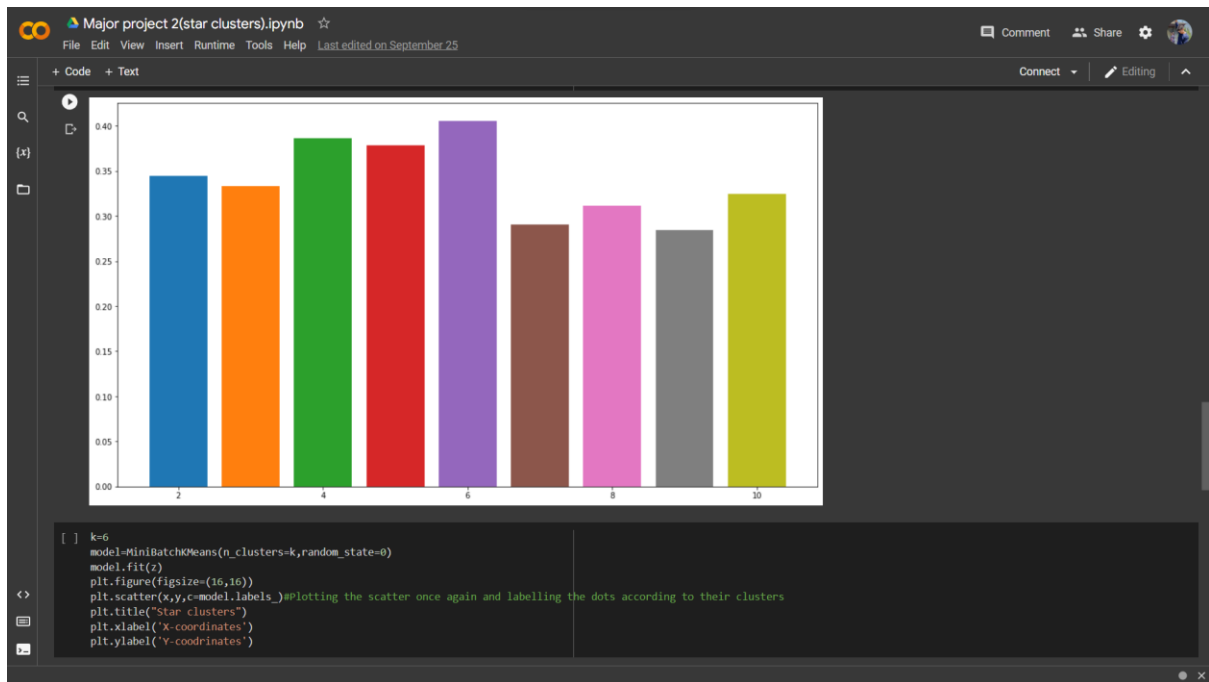
[ ]
x=df['x']
y=df['y']
z=df.loc[:,('x','y')].values

array([[ 0.48593906, -0.52435857],
       [-0.06596069,  0.08084424],
       [-0.03480916,  0.07679548],
       ...,
       [-0.17556863, -0.39738956],
       [ 0.23999888,  0.62563109],
       [-0.15866721, -1.1509324 ]])

[ ] #Plotting the x and y coordinates of stars in a scatter plot
plt.figure(figsize=(16,16))
plt.scatter(x,y)

<matplotlib.collections.PathCollection at 0x7f64fdcd350>
```





## Resources

Github account containing repositories for all 3 major projects and includes all the files used:

<https://github.com/sanyamjain0315>

Heroku account with the app deployments:

<https://emotion-detector-sanyam-jain.herokuapp.com/>

<https://mumbai-house-price-detection.herokuapp.com/>

Google drive link with google colab notebooks and resources:



<https://drive.google.com/drive/folders/1wLGEPkK8zN3mhfl2XSGT1723Lc2jZvLy?usp=sharing>