Name: R.Iniyavan
Roll No:20I120

Assesment-2

Link:

https://colab.research.google.com/drive/1_I-w6KmzNVa-s9iQ0hObjCUo053h85O5?usp=sharing

1)

```
[1] from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[2] import pandas as pd
    import matplotlib.pyplot as plt
    import numpy as np
```

```
[3] data=pd.read_csv('/content/drive/MyDrive/Datasets for Assesments/voice.csv')
```

```
data.head()
```

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...	centroid	meanfun	minfun	maxfun
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402905	0.893369	0.491918	...	0.059781	0.084279	0.015702	0.275862
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	...	0.066009	0.107937	0.015826	0.250000
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	...	0.077316	0.098706	0.015656	0.271186
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232	...	0.151228	0.088965	0.017798	0.250000
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713	0.971955	0.783568	...	0.135120	0.106398	0.016931	0.266667

5 rows × 21 columns

```
from sklearn.preprocessing import LabelEncoder
```

```
[6] labelencoder=LabelEncoder()
```

```
[9] data['label']=labelencoder.fit_transform(data['label'])
```

```
[10] data['label']
```

```
0       1
1       1
2       1
3       1
4       1
       ..
3163    0
3164    0
3165    0
3166    0
3167    0
Name: label, Length: 3168, dtype: int64
```

```
[11] from sklearn.model_selection import train_test_split
```

```
[13] x=data.iloc[:,0:-1].values
     y=data.iloc[:,-1].values
```

2)

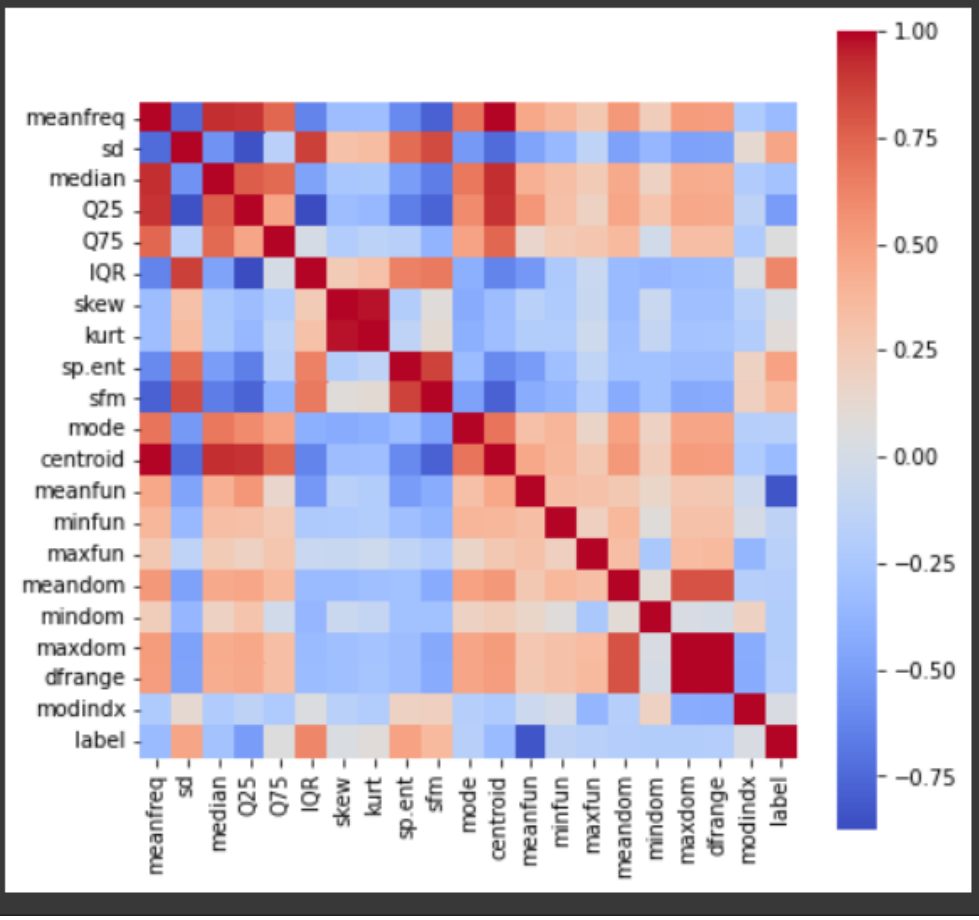
```
[17] x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=120)

     from sklearn.linear_model import LogisticRegression

[19] model=LogisticRegression()

[20] model.fit(x_train,y_train)

     /usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1):
     STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

     Increase the number of iterations (max_iter) or scale the data as shown in:
         https://scikit-learn.org/stable/modules/preprocessing.html
     Please also refer to the documentation for alternative solver options:
         https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
       extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
     LogisticRegression()
```

```
[21] y_pred=model.predict(x_test)

[22] from sklearn.metrics import accuracy_score

     ac=accuracy_score(y_test,y_pred)
     print(ac)

     0.9100946372239748
```

3)

```
[27] import seaborn as sns
```

```
corr = data.corr()
plt.figure(figsize=(7,7))
sns.heatmap(corr, cbar = True,  square = True,
            cmap= 'coolwarm')
plt.show()
```



4)

```
[103] x=data.iloc[:,[1,3,5,8,9,11,12]].values
      y=data.iloc[:,-1].values
```

```
data.head()
```

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...	centroid	meanfun	minfun	maxfun	meandom	mindom	maxdom	dfrange	
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402905	0.893369	0.491918	...	0.059781	0.084279	0.015702	0.275862	0.007812	0.007812	0.007812	0.000000	0
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	...	0.066009	0.107937	0.015826	0.250000	0.009014	0.007812	0.054688	0.046875	0
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	...	0.077316	0.098706	0.015656	0.271186	0.007990	0.007812	0.015625	0.007812	0
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232	...	0.151228	0.088965	0.017798	0.250000	0.201497	0.007812	0.562500	0.554688	0
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713	0.971955	0.783568	...	0.135120	0.106398	0.016931	0.266667	0.712812	0.007812	5.484375	5.476562	0

5 rows × 21 columns

```
[104] x
```

```
array([[0.06424127, 0.01507149, 0.07512195, ..., 0.49191777, 0.05978098,
        0.08427911],
       [0.06731003, 0.01941387, 0.07325232, ..., 0.51372384, 0.06600874,
        0.10793655],
       [0.08382942, 0.00870106, 0.12320696, ..., 0.47890498, 0.0773155 ,
        0.09870626],
       ...,
       [0.09579843, 0.03342387, 0.19093638, ..., 0.65419636, 0.14205626,
        0.20991768],
       [0.09062826, 0.0435081 , 0.1764347 , ..., 0.67546972, 0.14365874,
        0.172375  ],
       [0.09288354, 0.0700715 , 0.18075587, ..., 0.60152881, 0.16550895,
        0.18560693]])
```

```
[112] ac=accuracy_score(y_test,y_pred)
      print(ac)
```

```
0.9100946372239748
```