Krish Sukhani

TE IT

Batch : D

2018140059

ML - Exp 4 - Wine Quality Classification Daataset

```
#importing necessary libraries
import numpy as np
import pandas as pd
import warnings
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.neural_network import MLPClassifier
import matplotlib.pyplot as plt
```

```
from google.colab import drive
drive.mount("/content/gdrive")
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

```
wine = pd.read_csv('/content/gdrive/My Drive/datasets/wine.csv',encoding= 'unicode_escape')
```

```
wine.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	bad
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	bad
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	good
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad

```
wine.isnull().sum()
```

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

```
# plt.figure(figsize=(40,25))
# plt.subplots_adjust(left=0, bottom=0.5, right=0.9, top=0.9, wspace=0.5, hspace=0.8)
# plt.subplot(141)
# plt.title('Percentage of good and bad quality wine',fontsize = 20)
# wine['quality'].value_counts().plot.pie(autopct="%1.1f%%")
```

```
wine['quality'].replace({'bad': 0 , 'good': 1}, inplace=True)
```

```
wine.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	0
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	0
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	0
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	1
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	0

```
# Y = wine['quality']
```

```
# Y
```

```
# X = wine.drop(['quality'],axis = 1)
```

```
# X
```

```
y = wine['quality'].values
y = y.reshape(-1,1)
x = wine.drop(['quality'],axis = 1)

x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.3,random_state=100)
y_train = y_train.reshape(-1,1)
y_test = y_test.reshape(-1,1)
```

```
print("x_train: ",x_train.shape)
print("x_test: ",x_test.shape)
print("y_train: ",y_train.shape)
print("y_test: ",y_test.shape)
```

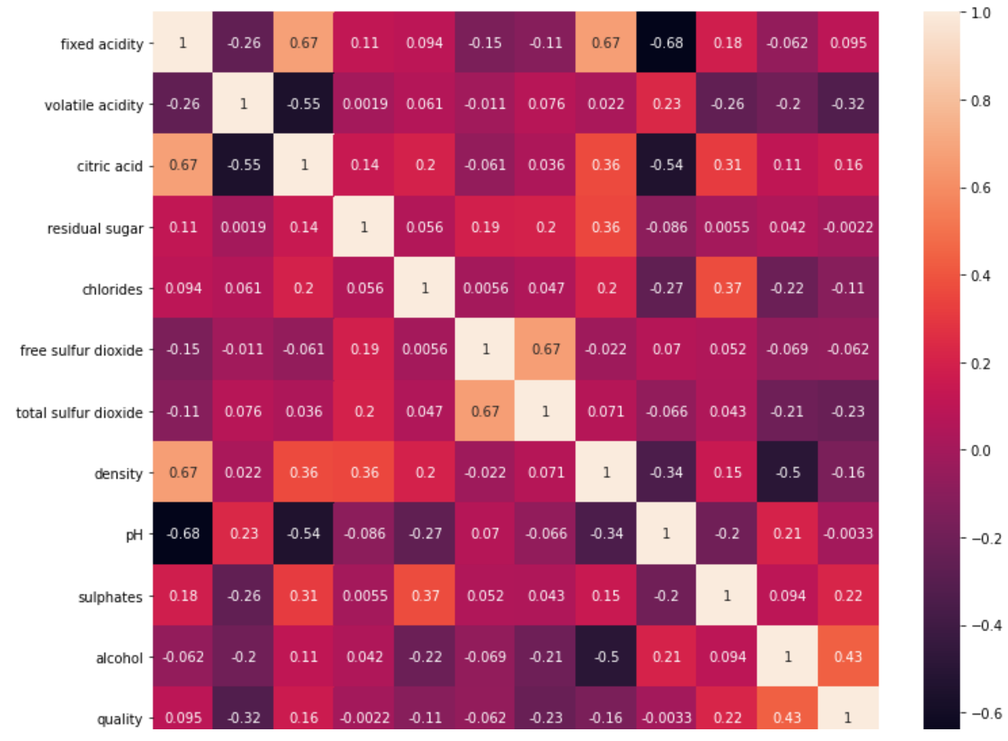
```
x_train:  (1119, 11)
x_test:  (480, 11)
y_train:  (1119, 1)
y_test:  (480, 1)
```

```
from sklearn.neural_network import MLPClassifier
classifier = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(10, 3), random_state=2)


classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.5333333333333333
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:934: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for exam
  y = column_or_1d(y, warn=True)
```

```
plt.figure(figsize=(12,10))
sns.heatmap(wine.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5668ffba90>
```




```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_train, y_train=scaler.fit_transform(x_train),scaler.fit_transform(y_train)
```

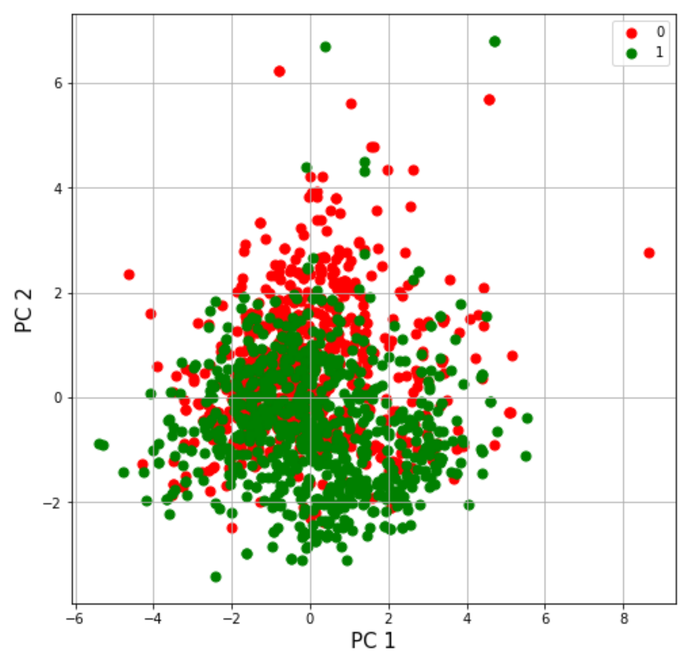
```
from sklearn.decomposition import PCA
pca=PCA(n_components=2)
x_pca=scaler.fit_transform(x)
m=pca.fit_transform(x_pca)


m_df=pd.DataFrame(data = m, columns= ['PC 1', 'PC 2'])


final_df = pd.concat([m_df, wine[['quality']]], axis=1)


final_df.head()
```

	PC 1	PC 2	quality
0	-1.619530	0.450950	0
1	-0.799170	1.856553	0
2	-0.748479	0.882039	0
3	2.357673	-0.269976	1
4	-1.619530	0.450950	0

```
pca.explained_variance_ratio_
```

```
array([0.28173931, 0.1750827 ])
```

```
fig=plt.figure(figsize=(8,8))
ax=fig.add_subplot(1,1,1)
ax.set_xlabel('PC 1',fontsize = 15)
ax.set_ylabel('PC 2',fontsize = 15)
targets = [0,1]
colors=['r','g']
for target , color in zip(targets,colors):
  indicesToKeep = final_df['quality'] == target
  ax.scatter(final_df.loc[indicesToKeep,'PC 1']
             ,final_df.loc[indicesToKeep,'PC 2']
             ,c = color
             ,s = 50)
ax.legend(targets)
ax.grid()
```




```
y = final_df['quality'].values
y = y.reshape(-1,1)
x = final_df.drop(['quality'],axis = 1)


x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.3,random_state=100)


classifier = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(10, 3), random_state=2)


classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:934: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for exam
  y = column_or_1d(y, warn=True)
Accuracy: 0.6375
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:470: 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
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
```