```
#KMEANS
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from google.colab import files
uploaded = files.upload()
from sklearn.cluster import KMeans
data = pd.read_csv('student_scores.csv')
data
plt.scatter(data['Hours'],data['Scores'])
plt.xlim(0,10)
plt.ylim(0,100)
plt.show()
x = data.iloc[:,0:2]
kmeans = KMeans(2)
kmeans.fit(x)
print(kmeans.labels_)
identified_clusters = kmeans.fit_predict(x)
print(identified_clusters)
data_with_clusters = data.copy()
data_with_clusters['Clusters'] = identified_clusters
plt.scatter(data_with_clusters['Hours'],data_with_clusters['Scores'],c=
data_with_clusters['Clusters'],cmap='rainbow')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=500);
predicted_class = kmeans.predict([[3.6, 42], [2.8, 28]])
print(predicted_class)

#PCA
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from google.colab import files
uploaded = files.upload()
df = pd.read_csv('diabetes.csv')
df.head()
#y = df.pop('quality')
from sklearn.preprocessing import StandardScaler
scalar = StandardScaler()
print(scalar)
df_scaled = pd.DataFrame(scalar.fit_transform(df), colu
mns=df.columns)
df_scaled
pca = PCA(n_components=4)
pca
df_pca = pd.DataFrame(pca.fit_transform(df_scaled))
df_pca

#Without Bagging
import pandas as pd
from google.colab import files
from google.colab import files
uploaded = files.upload()
df = pd.read_csv("winequality-red.csv")
df.head()
df.isnull().sum()
X = df.drop("quality",axis="columns")
y = df.quality
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled[:3]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
stratify=y, random_state=10)
X_train.shape
X_test.shape
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
scores = cross_val_score(DecisionTreeClassifier(), X, y, cv=5)
scores
scores.mean()

#with bagging
import pandas as pd
from google.colab import files
uploaded = files.upload()
df = pd.read_csv("winequality-red.csv")
df.head()
df.isnull().sum()
X = df.drop("quality",axis="columns")
y = df.quality
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled[:3]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
stratify=y, random_state=10)
X_train.shape
X_test.shape
from sklearn.ensemble import BaggingClassifier
bag_model = BaggingClassifier(
base_estimator=DecisionTreeClassifier(),
n_estimators=100,
max_samples=0.8,
bootstrap=True,
oob_score=True,
random_state=0
)
bag_model.fit(X_train, y_train)
bag_model.oob_score_

#Boosting : Adaboost
import pandas as pd
from google.colab import files
uploaded = files.upload()
uploaded = files.upload()
df = pd.read_csv("winequality-red.csv")
df.head()
df.isnull().sum()
X = df.iloc[:, 0:11]
y = df.iloc[:, 11]
#X = df.drop("Outcome",axis="columns")
#y = df.Outcome
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
X_train.shape
from sklearn.ensemble import AdaBoostClassifier
adaboost = AdaBoostClassifier(n_estimators = 50, learning_rate =
0.2).fit(X_train, y_train)
score = adaboost.score(X_test, y_test)
score

import matplotlib.pyplot as plt
import numpy as np

#day one, the numbers and marks of B16 students:
x = np.array([10,12,14,15,16,17,21,22,23,24,26,29,25])
y = np.array([99,86,87,88,99,86,93,87,94,78,77,85,86])

plt.scatter(x, y,color='blue')
#day two, the numbers and marks of B17 students:
x = np.array([13,25,32,36,45])
y = np.array([85,96,77,82,96])
plt.scatter(x, y,color='yellow')
plt.show()

import matplotlib.pyplot as plt
import numpy as np

y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
mycolors = ["black", "hotpink", "yellow", "red"]

plt.pie(y, labels = mylabels, colors = mycolors)
plt.show()

import matplotlib.pyplot as plt
import numpy as np

y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]

plt.pie(y, labels = mylabels)
plt.show()

import matplotlib.pyplot as plt
import numpy as np

#day one, the numbers and marks of B16 students:
x = np.array([10,12,14,15,16,17,21,22,23,24,26,29,25])
```

```
y = np.array([99,86,87,88,99,86,93,87,94,78,77,85,86])

plt.plot(x,linestyle = 'dashed')
plt.plot(y,linestyle = 'dashed')
#day two, the numbers and marks of B17 students:
x = np.array([13,25,32,36,45])
y = np.array([85,96,77,82,96])
plt.plot(x,linestyle = 'dashed')
plt.plot(y,linestyle = 'dashed')


plt.show()
```