PRN - 240340128031

import pandas as pd

import numpy as np

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

from sklearn.datasets import load\_iris

from sklearn.compose import make\_column\_transformer, make\_column\_selector

from sklearn.preprocessing import OneHotEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.impute import SimpleImputer

from sklearn.model\_selection import train\_test\_split, StratifiedKFold, GridSearchCV

from sklearn.metrics import log\_loss, accuracy\_score, confusion\_matrix, silhouette\_score, ConfusionMatrixDisplay

import matplotlib.pyplot as plt

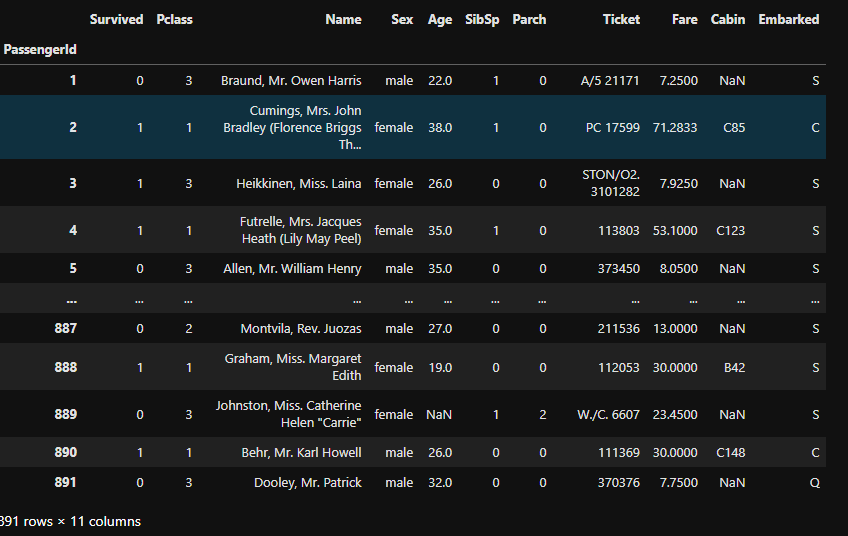
Q1.

a

titanic = pd.read\_csv("Titanic-Dataset.csv",index\_col=0)

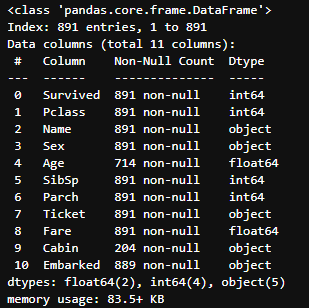
titanic

1\_1



titanic.info()

1\_2



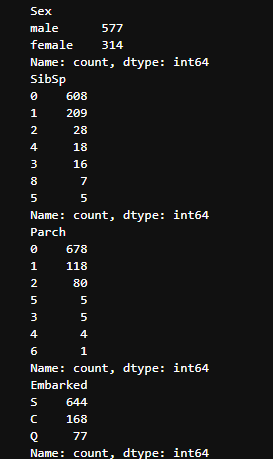
print(titanic['Sex'].value\_counts())

print(titanic['SibSp'].value\_counts())

print(titanic['Parch'].value\_counts())

print(titanic['Embarked'].value\_counts())

1\_3

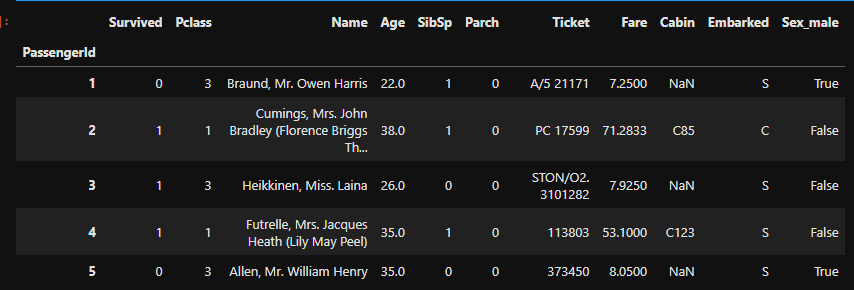


titanic['Sex\_male'] = pd.get\_dummies(titanic['Sex'],drop\_first = True)

titanic.drop("Sex",axis=1,inplace=True)

titanic.head()

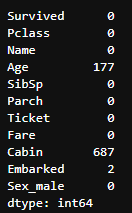
1\_4



b

titanic.isna().sum()

1\_5



imp\_mean = SimpleImputer(missing\_values=np.nan, strategy='mean')

ct = make\_column\_transformer(("passthrough",make\_column\_selector(dtype\_include = object)),

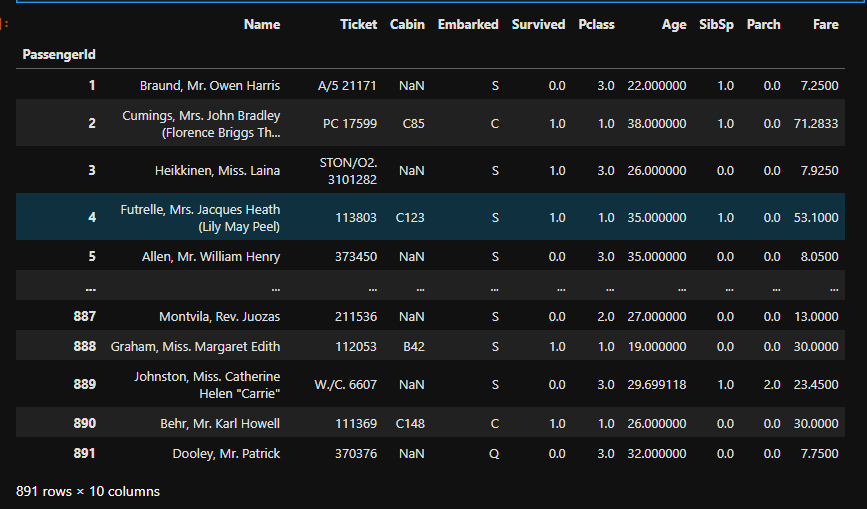
(imp\_mean,make\_column\_selector(dtype\_include = ["int64","float64"])),

verbose\_feature\_names\_out=False).set\_output(transform='pandas')

dum\_np = ct.fit\_transform(titanic)

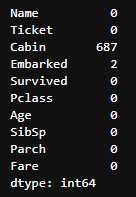
dum\_np

1\_6



dum\_np.isna().sum()

1\_7



Q2

titanic = pd.read\_csv("Titanic-Dataset.csv",index\_col=0)

titanic

2\_1

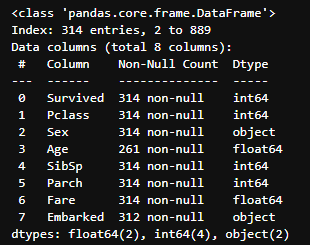


female\_titanic = titanic[titanic['Sex']=='female']

female\_titanic.drop(['Name','Ticket','Cabin'],inplace=True,axis=1)

female\_titanic.info()

2\_2



impute\_age = SimpleImputer(strategy='mean').set\_output(transform='pandas')

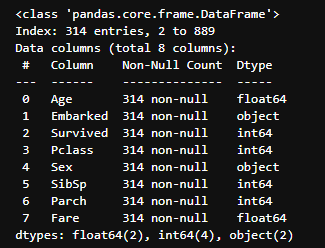
impute\_embarked = SimpleImputer(strategy='most\_frequent').set\_output(transform='pandas')

ct = make\_column\_transformer((impute\_age,['Age']),(impute\_embarked,['Embarked']),remainder='passthrough',verbose\_feature\_names\_out=False).set\_output(transform='pandas')

titanic\_fe = ct.fit\_transform(female\_titanic)

titanic\_fe.info()

2\_3



X = pd.get\_dummies(titanic\_fe.drop('Survived',axis=1),drop\_first=True)

y = titanic\_fe['Survived']

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=24, stratify=y)3

rf = RandomForestClassifier()

rf.fit(X\_train,y\_train)

y\_pred = rf.predict(X\_test)

print(accuracy\_score(y\_test,y\_pred))

y\_pred\_prob = rf.predict\_proba(X\_test)

print(log\_loss(y\_test,y\_pred\_prob))

2\_4



b

confusion\_matrix(y\_test,y\_pred)

2\_5



C

rf = RandomForestClassifier()

kfold = StratifiedKFold(n\_splits=5,shuffle=True, random\_state=24)

params = {'max\_depth':[None,2,3],'min\_samples\_split':[2,10,15],'min\_samples\_leaf':[1,5,9]}

gcv = GridSearchCV(rf,param\_grid=params,cv=kfold,scoring='neg\_log\_loss')

gcv.fit(X,y)

print(gcv.best\_params\_)

print(gcv.best\_score\_)

2\_6



From comparing results, absolute value of neg loss after tuning (-0.414932) is closer to zero than log loss value (0.417944)without tuning. Hence it can be concluded that results have improved.

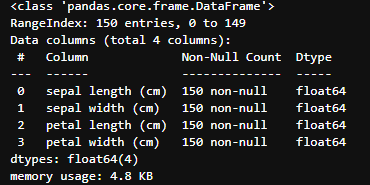
Q3

iris = load\_iris()

df = pd.DataFrame(data = iris.data,columns=iris.feature\_names)

df.info()

3\_1



scaler = StandardScaler().set\_output(transform='pandas')

iris\_scaled = scaler.fit\_transform(df)

Ks = np.arange(2,8)

scores = []

for i in Ks:

clust = KMeans(n\_clusters=i,random\_state=24,init='random')

clust.fit(iris\_scaled)

scores.append(silhouette\_score(iris\_scaled, clust.labels\_))

i\_max = np.argmax(scores)

print("Best no. of clusters:", Ks[i\_max])

print("Best Score:", scores[i\_max])

3\_2



plt.plot(Ks,scores)

plt.title("Number of Centroids vs silhouette score")

plt.xlabel('Number of Centroids')

plt.ylabel('silhouette score')

plt.show()

3\_3

