**ML Practical Questions**

**Q. 1 Write a Python program to prepare Scatter Plot for Iris Dataset.**

import pandas as pd

df=pd.read\_csv("iris.csv")

df.plot.scatter(x='sepal\_length',y='sepal\_width',title='scatter plot between sepal length & width' )

df.plot.scatter(x='petal\_length',y='petal\_width',title='scatter plot between petal length & width' )

**Q. 2 Write a python program to find all null values in a given dataset and remove them**.

import pandas as pd

import numpy as np

df=pd.read\_csv("nullvalues.csv")

df

print(df.head())

print(df['ST\_NUM'])

print(df['ST\_NUM'])

print(df['ST\_NUM'].isnull())

print(df['NUM\_BEDROOMS'])

print(df['NUM\_BEDROOMS'].isnull())

print(df['OWN\_OCCUPIED'])

print(df['OWN\_OCCUPIED'].isnull())

print(df.isnull().sum())

(df['ST\_NUM'].fillna(125,inplace=True))

df

**Q. 3 Write a python program to make Categorical values in numeric format for a given dataset**

import pandas as pd

df=pd.read\_csv('categorical.csv')

df1=pd.get\_dummies(df['Purchased'])  
df

df1

df=pd.concat([df,df1],axis=1).reindex(df.index)

df

df.drop('Purchased',axis=1,inplace=True)

df

**Q. 4 Write a python program to Implement Simple Linear Regression for predicting house price**.

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

def estimate\_coef(x,y):

n=np.size(x)

m\_x=np.mean(x)

m\_y=np.mean(y)

SS\_xy =np.sum(y\*x) - n\*m\_y\*m\_x

SS\_xx =np.sum(x\*x) - n\*m\_x\*m\_x

b\_1 = SS\_xy / SS\_xx

b\_0 = m\_y - b\_1\*m\_x

return(b\_0, b\_1)

def plot\_regression\_line(x,y,b):

plt.scatter(x,y,color ="m",marker="o", s=30)

y\_pred =b[0] +b[1]\*x

plt.plot(x,y\_pred,color="g")

plt.xlabel('x')

plt.ylabel('y')

plt.show()

def main():

# observations / data

df=pd.read\_csv("kc\_house\_data.csv")

y=df['price']

x=df['bedrooms']

b=estimate\_coef(x,y)

print("coefficients are b\_0 ={},b\_1 = {}".format(b[0], b[1]))

plot\_regression\_line(x,y,b)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Q. 5 Write a python program to implement Multiple Linear Regression for given dataset**.

import pandas as pd

from sklearn import linear\_model

df=pd.read\_csv("multiple\_regression.csv")

df

df=df.head(4)

df

x=df[['weight','volume']]

y=df['co2']

x

y

regr=linear\_model.LinearRegression()

regr.fit(x,y)

predicted\_co2=regr.predict([[1200,1300]])

print(predicted\_co2)

**Q. 6 Write a python program to implement Polynomial Linear Regression for given dataset**

import numpy as np

import pandas as pd

import matplotlib.pyplot asplt

x=np.arange(0,30)

y=[3,4,5,7,10,8,9,10,10,23,27,44,50,63,67,60,62,70,75,88,81,87,95,100,108,135,151,160,169,179]

plt.figure(figsize=(10,6))

plt.scatter(x,y)

plt.show()

from sklearn.preprocessing import PolynomialFeatures

poly=PolynomialFeatures(degree=2,include\_bias=False)

poly\_features=poly.fit\_transform(x.reshape(-1,1))

from sklearn.linear\_model import LinearRegression

poly\_reg\_model=LinearRegression()

poly\_reg\_model.fit(poly\_features,y)

y\_predicted=poly\_reg\_model.predict(poly\_features)

plt.figure(figsize=(10,6))

plt.title("polynomial regression",size=16)

plt.scatter(x,y)

plt.plot(x,y\_predicted,c="red")

plt.show()

y1=poly\_reg\_model.predict(poly.fit\_transform(np.array([34]).reshape(-1,1)))

print(y1)

**Q. 7 Write a python program to implement Naive Bayes.**

%matplotlib inline

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns;sns.set()

from sklearn.datasets import make\_blobs

X,y=make\_blobs(100,2,centers=2,random\_state=2,cluster\_std=1.5)

plt.scatter(X[:,0],X[:,1],c=y,s=50,cmap='RdBu');

from sklearn.naive\_bayes import GaussianNB

model=GaussianNB()

model.fit(X,y);

rng=np.random.RandomState(0)

Xnew=[-6,-14]+[14,18]\*rng.rand(2000,2)

ynew=model.predict(Xnew)

plt.scatter(X[:,0],X[:,1],c=y,s=50,cmap='RdBu')

lim=plt.axis()

plt.scatter(Xnew[:,0],Xnew[:,1],c=ynew,s=20,cmap='RdBu',alpha=0.1)

plt.axis(lim);

yprob=model.predict\_proba(Xnew)

yprob[-8:].round(2)

**Q. 8 Write a python program to implement Decision Tree whether or not to play Tennis**

import numpy as np

import pandas as pd

data=pd.read\_csv("PlayTennis.csv")

data

from sklearn.preprocessing import LabelEncoder

outlook=LabelEncoder()

temp=LabelEncoder()

humidity=LabelEncoder()

windy=LabelEncoder()

play=LabelEncoder()

data['outlook']=outlook.fit\_transform(data['outlook'])

data['temp']=outlook.fit\_transform(data['temp'])

data['humidity']=outlook.fit\_transform(data['humidity'])

data['windy']=outlook.fit\_transform(data['windy'])

data['play']=outlook.fit\_transform(data['play'])

data

features\_cols=['outlook','temp','humidity','windy']

x=data[features\_cols]

y=data.play

x

y

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)

from sklearn.tree import DecisionTreeClassifier

classifier=DecisionTreeClassifier(criterion='gini')

classifier.fit(x\_train,y\_train)

classifier.predict(x\_test)

x\_test

y\_test

classifier.score(x\_test,y\_test)

from sklearn import tree

tree.plot\_tree(classifier)

**Q. 9 Write a python program to implement Linear SVM.**

import numpy as np

import matplotlib.pyplot as plt

from sklearn import svm,datasets

iris=datasets.load\_iris()

X=iris.data[:,:2]

y=iris.target

C=1.0

svc=svm.SVC(kernel='linear',C=1).fit(X,y)

x\_min, x\_max =X[:,0].min() -1, X[:,0].max()+1

y\_min, y\_max =X[:,1].min() -1, X[:,0].max()+1

h=(x\_max / x\_min)/100

xx,yy =np.meshgrid(np.arange(x\_min, x\_max,h),

np.arange(y\_min,y\_max,h))

plt.subplot(1,1,1)

Z=svc.predict(np.c\_[xx.ravel(), yy.ravel()])

Z=Z.reshape(xx.shape)

plt.contourf(xx,yy,Z,cmap = plt.cm.Paired,aplha=0.8)

plt.scatter(X[:,0],X[:,1],c=y, cmap=plt.cm.Paired)

plt.xlabel('Sepal length')

plt.xlabel('Sepal width')

plt.xlim(xx.min(),xx.max())

plt.title('SVC with linear kernel')

plt.show()