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())
```

Edit with WPS Office

```
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

Edit with WPS Office

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():
```

Edit with WPS Office

```
# 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
```

Edit with WPS Office

```
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 as plt

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()
```

Edit with WPS Office

```
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');
```

Edit with WPS Office

```
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")
```

Edit with WPS Office

```
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)
```

Edit with WPS Office

```
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
```

Edit with WPS Office

```
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()
```

Edit with WPS Office