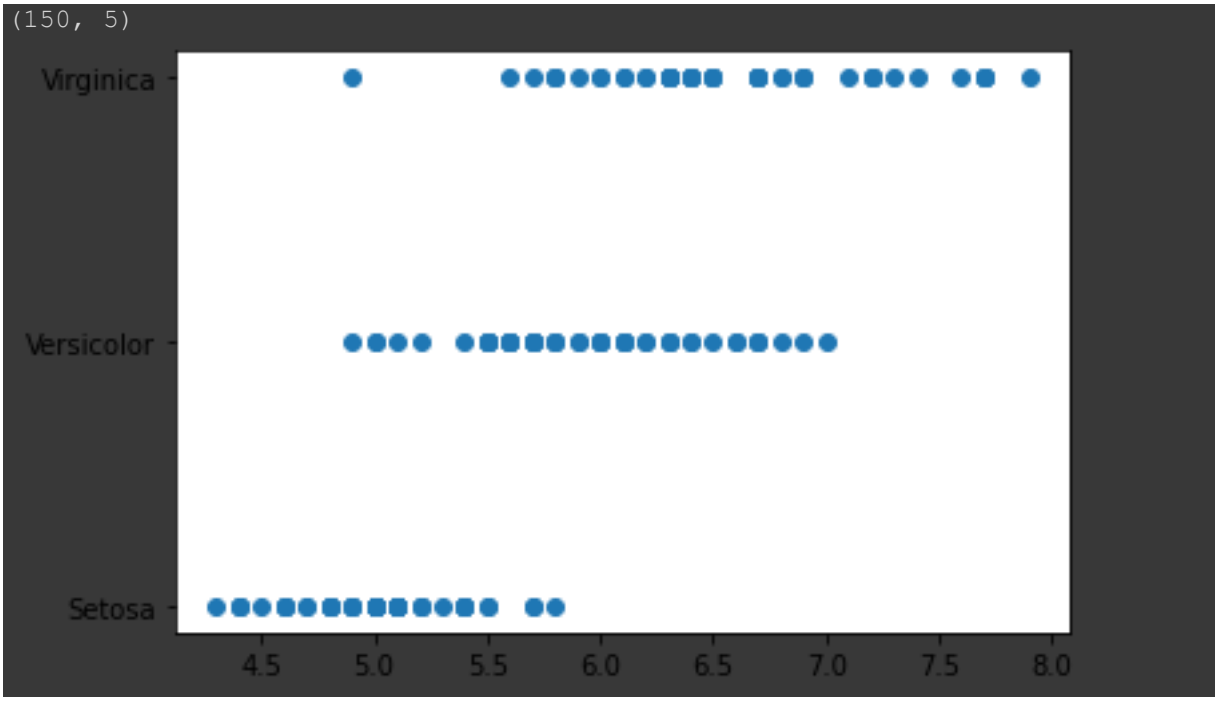
Subject: Machine Learning slip solution

1)write a program to prepare scatter plot(use iris dataset).

```
import pandas as pd
import matplotlib. pyplot as plt
import numpy as np
iris=pd.read_csv("/iris.csv")
print(iris.shape)
df=pd.DataFrame(iris)
x=df['sepal.length'].values
y=df['variety'].values
plt.scatter(x,y)
plt.show()
```

OUTPUT



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

```
import pandas as pd
data = pd.read_csv("Maths.csv")
print(data)
df = data.dropna(axis = 0, how = 'any', inplace = False)
print(df)
```

OUTPUT

```
   experience  salary
0         2.0   100.0
1         4.0   300.0
6         7.0  1000.0
```

3)Write a python program th categorical values in numeric format for a given datasets.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer

df=pd.read_excel("simple_regression.xlsx")
x=df.iloc[:,:-1].values
y=df["price"].values

si=SimpleImputer(missing_values=np.nan,strategy="mean")
si.fit(x[:,1:2])
x[:,1:2]=si.fit_transform(x[:,1:2])
print(x)
```

OUTPUT

```
[[23.     1.   ]
 [34.     3.   ]
 [21.     4.   ]
 [45.     2.   ]
 [35.     5.   ]
 [23.     8.   ]
 [23.     3.125]
 [56.     1.   ]
 [30.     1.   ]]
```

4) Write a python program to implement simple Linear Regression for predicting house price.

import pandas as pd

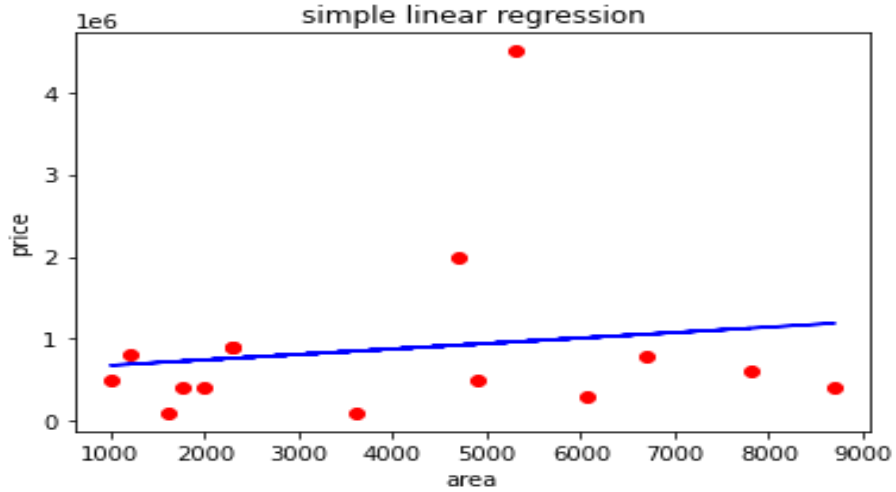
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

```
import matplotlib.pyplot as plt

df = pd.read_excel('House.xlsx')

x = df['Area'].values
y = df['Price'].values
xtr,xte,ytr,yte = train_test_split(x,y,test_size=0.2,random_state=1)


xtr = xtr.reshape(-1,1) #Feature Skaling
xte = xte.reshape(-1,1)
ytr = ytr.reshape(-1,1)


reg = LinearRegression()
reg.fit(xtr,ytr)

print("House Price of Area 8000 = ",reg.predict([[8000]]))

plt.scatter(xtr,ytr,color="red")
plt.title("Simple Linear Regression")
plt.xlabel("Area in Sq/ft")
plt.ylabel("Price")
plt.plot(xtr,reg.predict(xtr),color="blue")
plt.show()


print("Accuracy of the Model = ",reg.score(xtr,ytr))
```

OUTPUT

```
[[878310.02225477]]
```



5) Write a python program to implement multiple Linear Regression for a given dataset.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
df = pd.read_excel('House.xlsx')
x = df.iloc[:,:-1].values
y = df['Price'].values
xtr,xte,ytr,yte = train_test_split(x,y,test_size=0.2,random_state=1)
reg = LinearRegression()
reg.fit(xtr,ytr)
print("House Price of Area 8000 = ",reg.predict([[8000,10,10]]))
print("Accuracy of the Model = ",reg.score(xtr,ytr))
```

OUTPUT

```
[1110753.21217973]
accuracy 0.14327117826328462
```

6) Write a python program to implement Polynomial Regression for given dataset.

```
import pandas as pd
from sklearn .model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

df=pd.read_excel("House.xlsx")
x=df.iloc[:,:-1].values
y=df['price'].values

xtr,xte,ytr,yte=train_test_split(x,y,test_size=0.2,random_state=1)

poly=PolynomialFeatures(degree=3)
x_poly=poly.fit_transform(xtr)

reg=LinearRegression()
xtr=xtr.reshape(-1,1)
xte=xte.reshape(-1,1)

reg.fit(x_poly,ytr)
print(reg.score(x_poly, ytr))
```

OUTPUT

```
accuracy 1.0
```

7) Write a python program to Implement Naïve Bayes.

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB
df=pd.read_csv("area.csv")
```

```
x=df.iloc[:,:-1].values
y=df["price"].values
na=GaussianNB()
na.fit(x,y)
print("predict ",na.predict([[34500,5,22]]))
print("Accuracy",na.score(x,y))
```

OUTPUT

```
predict  [13000]
Accuracy 1.0
```

8) Write a python program to Implement Decision Tree whether or not to play tennis.

```
import pandas as pd
from sklearn.tree import DecisionTreeRegressor

df=pd.read_excel("Play.xlsx")

x=df.iloc[:,:-1].values
y=df['playtennis'].values

dec_tree=DecisionTreeRegressor()
dec_tree.fit(x,y)
print("Prediction:",dec_tree.predict([[2,6,2,10]]))
print("Accuracy:",dec_tree.score(x,y))
```

OUTPUT

```
prediction [0.]
accuracy 1.0
```

9) Write a python program to implement linear SVM.

import pandas as pd

```
from sklearn.linear_model import LinearRegression

from sklearn.svm import SVR

df=pd.read_csv("area.csv")

x=df['area'].values

y=df["price"].values

x=x.reshape(-1,1)

#y=y.reshape(-1,1)

sv=SVR();

sv.fit(x,y)

print("Prediction of support vector machine is",sv.predict(([[50000]])))

print("Accuracy is",sv.score(x,y))
```

```
Prediction of support vector machine is [45000.66143784]
Accuracy is -0.0018832525802194855
```