1. Write a python program to Prepare Scatter Plot (Use Forge Dataset / Iris Dataset)

```
import pandas as pd
iris=pd.read_csv("C:/Users/ADMIN/Documents/JAVASCRIPT/Machine-Learning-main/Iris.csv")
iris.plot(kind='scatter',x='SepalLengthCm',y='SepalWidthCm')
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
iris=pd.read_csv("C:/Users/ADMIN/Documents/JAVASCRIPT/Machine-Learning-main/Iris.csv")
sns.set_style("whitegrid");
sns.FacetGrid(iris,hue="Species",size=5).map(plt.scatter,"SepalLengthCm","SepalWidthCm").add_leg
end()
import pandas as pd
import matplotlib.pyplot as plt
data=pd.read_csv("C:/Users/ADMIN/Documents/JAVASCRIPT/Machine-Learning-main/Iris.csv")
SepalLengthCm=data["SepalLengthCm"]
SepalWidthCm=data["SepalWidthCm"]
x=[]
y=[]
x=list(SepalLengthCm)
y=list(SepalWidthCm)
plt.scatter(x,y)
plt.xlabel('SepalLengthCm')
plt.ylabel("SepalWidthCm")
plt.title('SepalLengthCm Vs SepalWidthCm')
plt.show()
```

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

```
import pandas as pd
result=pd.read_csv("C:/Users/User/Documents/Python Scripts/xyz.csv")
print(result)
print(result.isnull().sum())
```

result.dropna()

3. Write a python program the Categorical values in numeric format for a given dataset.

```
import pandas as pd
from pandas import ExcelFile
from pandas import ExcelWriter
import numpy as np

df = pd.read_csv("C:/Users/User/Documents/Python Scripts/Machine-Learning-main/Iris.csv")
df.head()

dummy = pd.get_dummies(df['Species'])

dummy.head()
df2 = pd.concat((df, dummy),axis = 1)
df2.head()

df2 = df2.drop(('Species'), axis = 1)
df2.head()
df2.tail()
```

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

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
housing = pd.read_csv('housing.csv')
housing.head()
housing.info()
housing.describe()
housing.columns
X = housing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
```

```
        'Avg. Area Number of Bedrooms', 'Area Population']]
y = housing['Price']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train,y_train)
# print the intercept
print(lm.intercept_)
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
predictions = lm.predict(X_test)
plt.scatter(y_test,predictions)
sns.distplot((y_test-predictions),bins=50);
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

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

```
import pandas as pd
import numpy as nm
import matplotlib.pyplot as plt
df = pd.read_csv('housing.csv')
df.describe()
fetures=df.iloc[:,1:-2]
target=df.iloc[:,-2]
fetures
target
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)
```

```
from sklearn.linear_model import LinearRegression
mlr=LinearRegression()
mlr.fit(x_train,y_train)

print("Intercept",mlr.intercept_)
print("Coefficient",mlr.coef_)
predicted_price = mlr.predict([[5.682861,7.009188,4.09,23086.800503]])
print(predicted_price)
```

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

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset=pd.read_csv('salary.csv')
dataset.head()
x=dataset.iloc[:,1:-1].values
y=dataset.iloc[:,-1].values
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x,y)
from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=4)
x_poly=poly_reg.fit_transform(x)
lin_reg2=LinearRegression()
lin_reg2.fit(x_poly,y)
plt.scatter(x,y,color='red',marker='*',s=100)
plt.plot(x,lr.predict(x),color='blue')
plt.title('True or False (Linear Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
```

```
plt.show()
plt.scatter(x,y,color='red',marker='*',s=100)
plt.plot(x,lin_reg2.predict(poly_reg.fit_transform(x)),color='blue')
plt.title('True or False (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
lr.predict([[6.5]])
lin_reg2.predict(poly_reg.fit_transform([[6.5]]))
```

7. Write a python program to Implement Naïve Bayes.

```
import pandas as pd
import numpy as np
from sklearn import datasets
iris = datasets.load_iris() # importing the dataset
iris.data # showing the iris data
X=iris.data #assign the data to the X
y=iris.target #assign the target/flower type to the y

print (X.shape)
print (y.shape)
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=9) #Split the dataset
from sklearn.naive_bayes import GaussianNB
nv = GaussianNB() # create a classifier
nv.fit(X_train,y_train) # fitting the data
from sklearn.metrics import accuracy_score
y_pred = nv.predict(X_test) # store the prediction data
accuracy_score(y_test,y_pred) # calculate the accuracy
```