Salary Prediction System

In [1]:

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score #Importing the r2_score function from the sklearn.metrics module for finding


```

In [2]:

```

def welcome():
    print("Welcome to salary prediction system")
    print("Press ENTER KEY to proceed")
    input()
```

In [3]:

```
def checkcsv():
    csv_files = []# Initializing list to hold CSV files
    cur_dir = os.getcwd()# Getting current working directory
    content_list = os.listdir(cur_dir)
    for x in content_list:
        if x.split(".")[-1] == 'csv':
            csv_files.append(x)

    if len(csv_files) == 0:
        return "No CSV files in the directory"
    else:
        return csv_files

```

In [4]:

```python
def display_and_select_csv (csv_files):
    # Displaying each CSV file with an index number
    i = 0
    for file_name in csv_files:
        print(i,"...", file_name)
        i +=1
    return csv_files[int(input("select file to create ML Model" ))]# Printing instruction for the user to select

#creating function
def graph(X_train, Y_train, regressionObject, X_test, Y_test, Y_pred):#Function to plot the training and test dat
    plt.scatter(X_train, Y_train, color = "red", label= "training data")# Ploting training data points
    plt.plot(X_train, regressionObject.predict(X_train), color = "blue", label = "Best fit")# Ploting the regress
    plt.scatter(X_test, Y_test, color = "green", label = "test data")# Ploting test data points
    plt.scatter(X_test, Y_pred, color = "black", label = "Pred test data")#Ploting predicted test data points
    plt.title("Salary vs Experience")## Seting graph title
    plt.xlabel("Years of Experience")#Seting graph label for x axis
    plt.ylabel("Salary")#Seting graph label for y axis
    plt.legend()# show legend
    plt.show()# display the graph
```

In [*]:

```
ain():
elcome()
ry:
csv_files = checkcsv()
if csv_files == "No csv files in the directory":
    raise FileNotFoundError("No csv files in the directory")
csv_files = display_and_select_csv(csv_files)
print(csv_files, "is selected")
print("Reading csv file")
print("Creating Dataset")
dataset = pd.read_csv(csv_files)
print("Dataset Created")
X = dataset.iloc[:,:-1].values
Y = dataset.iloc[:,-1].values
s = float(input("Enter test data size (between 0 to 1)"))
X_train,X_test,Y_train, Y_test = train_test_split(X, Y, test_size=s)
print("Model created in progression")
regressionObject = LinearRegression()
regressionObject.fit(X_train, Y_train)
print("Model is created")
print("Press ENTER key to predict test data in trained model")
input()



Y_pred = regressionObject.predict(X_test)
i = 0 #
print(X_test, "  ...", Y_test, "  ...", Y_pred)#Printing test data, actual salaries, and predicted salaries
while i<len(X_test): ##Loop for picking individual test data
    print(X_test[i], "...", Y_test[i], "...", Y_pred[i])#Printing individual test data, actual salaries, and predic
    i+=1 #Moves to the next test data point
print("press ENTER key to see above graphical format")# Printing instruction for the user to press ENTER to view re
input()#Press enter button for going to the next step
graph(X_train, Y_train, regressionObject, X_test, Y_test, Y_pred)#Call the graph function to plot the results
r2 = r2_score(Y_test, Y_pred)# Calculate the R^2 score for the model

print("our model is %2.2f%% accurate" %(r2*100))# Printing the accuracy of the model
print("Now you can predict salary of an employee using our model")#Inform the user that they can now predict salari
print("\nEnter experience in years of the candidates, seperated by comma")# Printing instruction for the user to en


```

```
exp = [float(e) for e in input().split(",")]# Reading experience years from user input and convert to a list of flo
ex = []# Initializing an empty list to hold experience values
for x in exp: # Looping over the list of experience years
    ex.append([x])# Append each experience year as a list to the 'ex' list
experience = np.array(ex)# Converting the 'ex' list to a numpy array
salaries = regressionObject.predict(experience)# Predicting salaries for the given experience years


plt.scatter(experience, salaries, color="black")  # Plotting the predicted salaries against experience years
plt.xlabel("Years of Experience")  # Setting the label for the x-axis
plt.ylabel("Salaries")  # Setting the label for the y-axis
plt.show()  # Displaying the plot


d = pd.DataFrame({"Experience": exp, "Salaries": salaries})  # Creating a DataFrame with experience and predicted s
print(d)  # Printing the DataFrame

xcept FileNotFoundError:
print("No csv file in the directory")
print("Press ENTER key to exit")# Instruction for the user to press ENTER to exit
input()# Wait for the user to press ENTER
exit()# Exit the program

_name__ == "__main__":
ain()
nput()


```

```
Welcome to salary prediction system
Press ENTER KEY to proceed

0 ... attachment_Salary-Data_lyst5512 (1).csv
select file to create ML Model0
attachment_Salary-Data_lyst5512 (1).csv is selected
Reading csv file
Creating Dataset
Dataset Created
Enter test data size (between 0 to 1)0.8
Model created in progression
Model is created
Press ENTER key to predict test data in trained model

[[ 5.3]
 [ 2. ]
 [ 8.7]
 [ 2.2]
 [ 8.2]
```

In [ ]: