RIPHAH INTERNATIONAL UNIVERSITY OF ISLAMABAD

Assignment #03
7th Semester (BSSE)

Submitted to: Mr. Waheed
Submitted by: M.Usama Nazir
Sap Id: 30445
Submission date: 24/04/2024

Question:

Follow the given steps to complete your assignment.

Part A 1. Import appropriate libraries 2. Read the csv file into a dataframe using appropriate function 3. Describe your dataset using appropriate function of pandas 4. Plot each input feature against the output feature/target into a scatter plot to see if there is a linear trend 5. Define a separate dataframe X and y representing input and target features 6. Use appropriate function to split the dataset into training and testing partitions 7. Create an instance of LinearRegression 8. Call the fit method for multiple linear regression using all input features 9. Predict the values for y_test and plot the true and predicted values 10.Print the score (r2)

Part B 7. After step 6 in above, perform Kfold validation using 3, 5 and 10 splits and report the validation score after each fold.

Solution:

```
# Step 1: Import appropriate libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, cross_val_score, KFold

from sklearn.linear_model import LinearRegression

from sklearn.metrics import r2_score


# Step 2: Read the csv file into a dataframe

df = pd.read_csv("/content/compensation-a1 (1).csv")


# Step 3: Describe the dataset

print(df.describe())
```

```
# Step 4: Plot each input feature against the output feature/target with custom colors

features = df.columns[:-1]  # Exclude the target column

target = df.columns[-1]

colors = ['red', 'orange', 'purple']  # Define custom colors

for i, feature in enumerate(features):

    plt.scatter(df[feature], df[target], alpha=0.5, color=colors[i % len(colors)])  # Use modulo to cycle through colors

    plt.xlabel(feature)

    plt.ylabel(target)

    plt.title(f"{feature} vs {target}")

    plt.show()


# Step 5: Define separate dataframe X and y


X = df.iloc[:, :-1]  # Input features

y = df.iloc[:, -1]   # Target feature


# Step 6: Split the dataset into training and testing partitions


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Step 7: Create an instance of LinearRegression

model = LinearRegression()
```

```
# Step 8: Fit multiple linear regression using all input features
model.fit(X_train, y_train)

# Step 9: Predict values for y_test and plot true vs predicted values

y_pred = model.predict(X_test)
plt.scatter(y_test, y_pred, color='blue')  # You can change the color here
plt.xlabel("True Values")
plt.ylabel("Predictions")
plt.title("True vs Predicted Values")
plt.show()

# Step 10: Print the score (r2)

r2 = r2_score(y_test, y_pred)
print("R2 Score:", r2)

# Part B: Perform k-fold cross-validation

# Define the number of splits
splits = [3, 5, 10]
```

 # Perform k-fold cross-validation

for num_splits in splits:

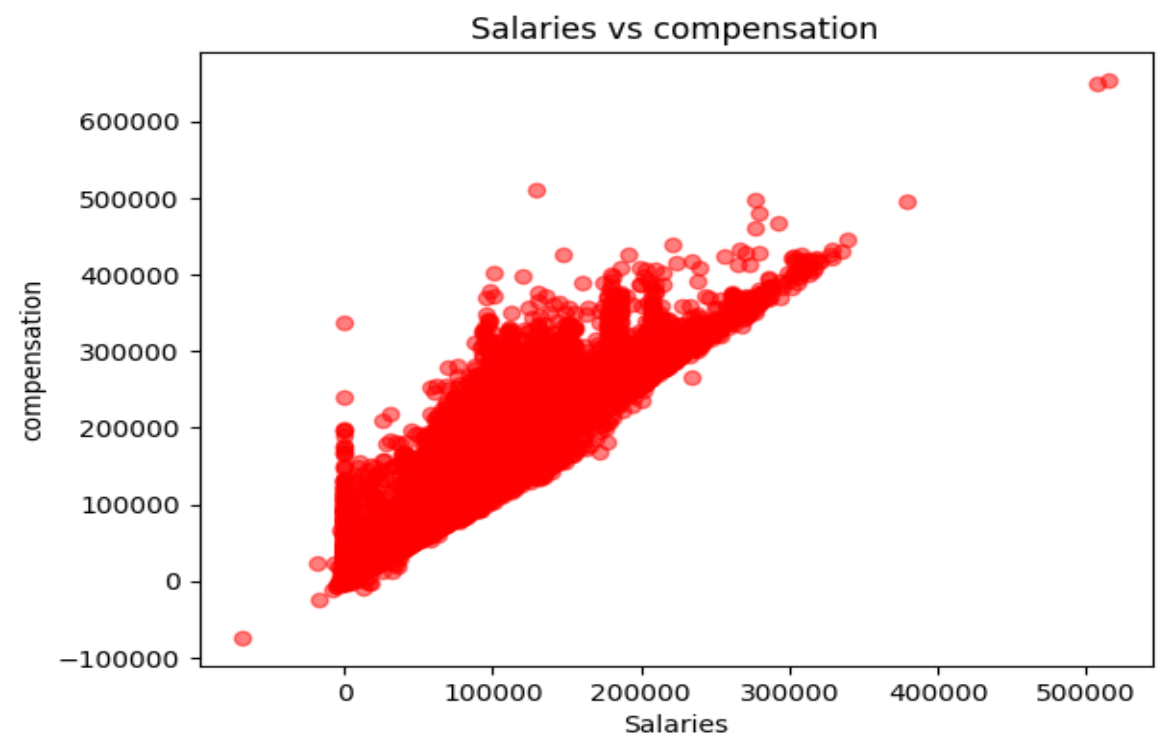
kf = KFold(n_splits=num_splits, shuffle=True, random_state=42)

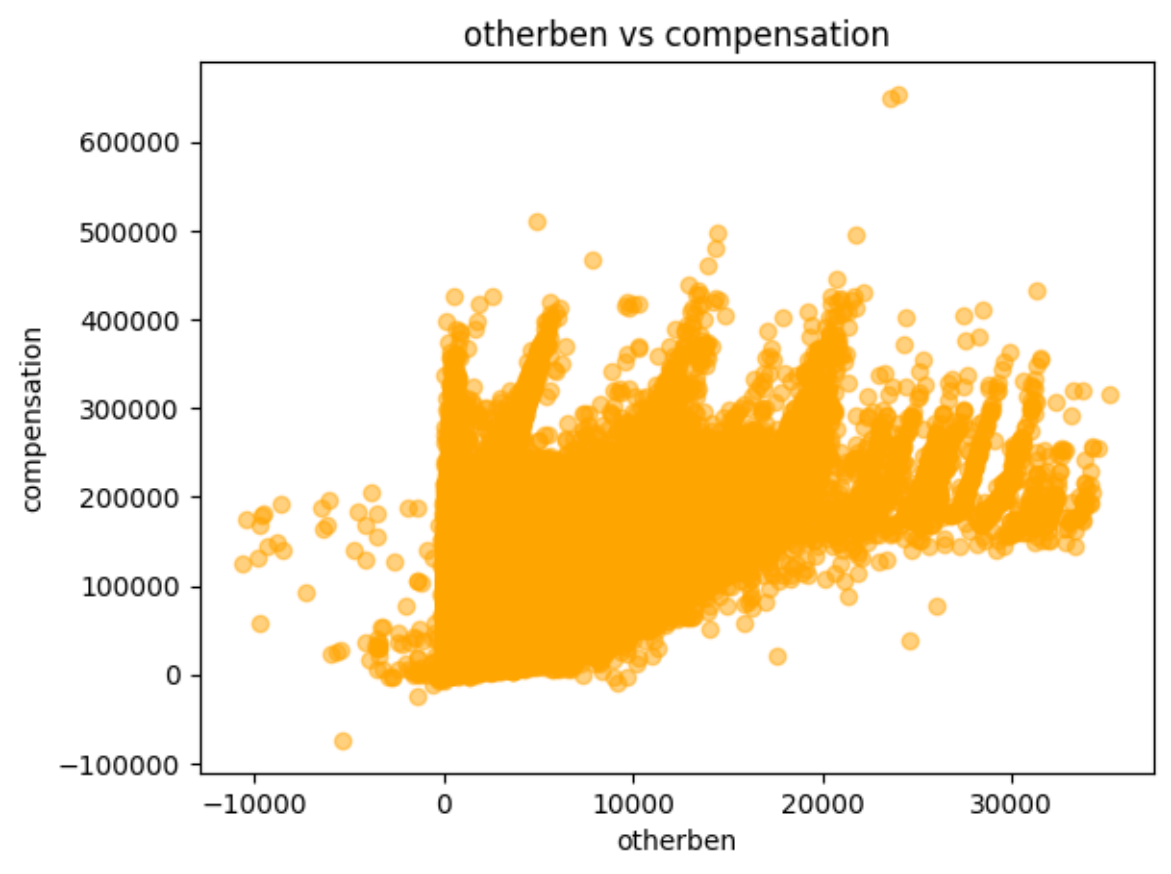
scores = cross_val_score(model, X, y, cv=kf)

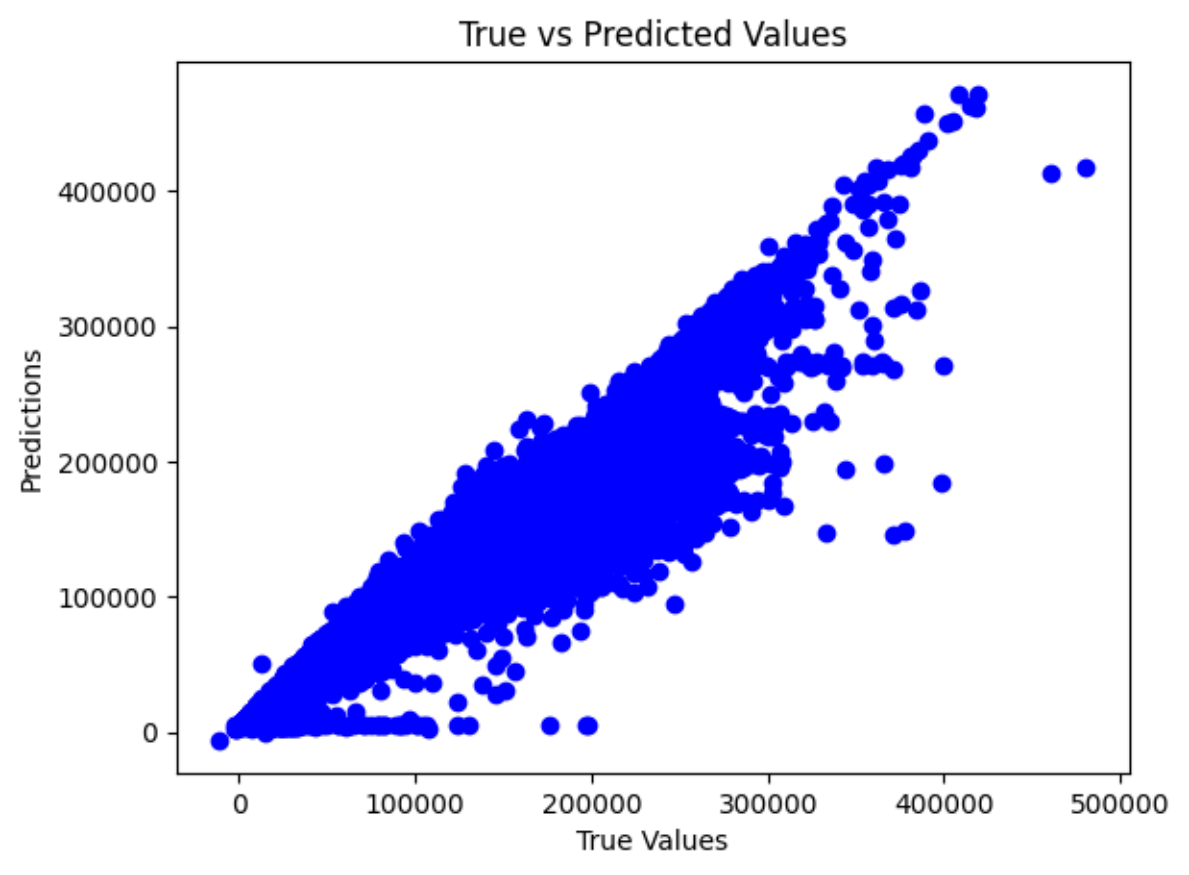
print(f"Validation score after {num_splits} splits: {scores.mean()}")

Output:

	Salaries	otherben	compensation
count	291825.000000	291825.000000	291825.000000
mean	63210.172887	4644.276407	97901.982292
std	44660.462305	3787.907010	67777.549152
min	-68771.780000	-10636.500000	-74082.610000
25%	23281.920000	1587.190000	35817.450000
50%	62519.120000	4349.090000	98022.370000
75%	92910.710000	6829.480000	142058.420000
max	515101.800000	35157.630000	653498.150000




otherben vs compensation
compensation
otherben
600000
500000
400000
300000
200000
100000
0
−100000
−10000
0
10000
20000
30000


True vs Predicted Values
Predictions
True Values
400000
300000
200000
100000
0
0
100000
200000
300000
400000
500000

“R2 Score: 0.9461601976224221”

Validation score after 3 splits: 0.9460823543532891
Validation score after 5 splits: 0.9460816769591134
Validation score after 10 splits: 0.9460808052679411