unit-2-assig1

November 18, 2024

[1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

[2]:
```
data = pd.read_csv('RJsKXWqDBZc3m0GG.csv')
print(data.head())
```

```
   Age          Eduacation      Race         Hisp MaritalStatus  Nodeg  \
0   45  LessThanHighSchool  NotBlack  NotHispanic       Married      1
1   21        Intermediate  NotBlack  NotHispanic    NotMarried      0
2   38          HighSchool  NotBlack  NotHispanic       Married      0
3   48  LessThanHighSchool  NotBlack  NotHispanic       Married      1
4   18  LessThanHighSchool  NotBlack  NotHispanic       Married      1

   Earnings_1974  Earnings_1975  Earnings_1978
0      21516.670      25243.550      25564.670
1       3175.971       5852.565      13496.080
2      23039.020      25130.760      25564.670
3      24994.370      25243.550      25564.670
4       1669.295      10727.610       9860.869
```

[5]:
```
data['Race'] = data['Race'].apply(lambda x: 1 if x == 'Black' else 0)
data['Hisp'] = data['Hisp'].apply(lambda x: 1 if x == 'Yes' else 0)
data['MaritalStatus'] = data['MaritalStatus'].apply(lambda x: 1 if x == 'Yes'␣
 ↪else 0)
```

[10]:
```
X = data[['Age', 'Race', 'Eduacation', 'Hisp', 'MaritalStatus',␣
 ↪'Earnings_1974', 'Earnings_1975']]
y = data['Earnings_1978']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[13]:
```
model = LinearRegression()

data['Race'] = data['Race'].apply(lambda x: 1 if x == 'Black' else 0)
data['MaritalStatus'] = data['MaritalStatus'].apply(lambda x: 1 if x == 'Yes'␣
 ↪else 0)
```

[16]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
preprocessor = ColumnTransformer(
    transformers=[
        ('edu', OneHotEncoder(), ['Eduacation'])  # OneHotEncode the␣
 ↪'Education' column
    ],
    remainder='passthrough'  # Keep the remaining columns as they are
)
```

[19]:
```
from sklearn.pipeline import Pipeline
X = data[['Age', 'Race', 'Eduacation', 'MaritalStatus', 'Earnings_1974',␣
 ↪'Earnings_1975']]
y = data['Earnings_1978']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Create a pipeline with preprocessing and model training
pipeline = Pipeline(steps=[
    ('preprocessing', preprocessor),
    ('regression', LinearRegression())
])

# Train the model
pipeline.fit(X_train, y_train)

# Make predictions on the test set
y_pred = pipeline.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared Score: {r2}')

# Visualize the predictions
plt.scatter(y_test, y_pred, color='blue')
plt.xlabel('Actual Earnings in 1978')
```

```
plt.ylabel('Predicted Earnings in 1978')
plt.title('Actual vs Predicted Earnings in 1978')
plt.show()
```

```
Mean Squared Error: 48659344.27465752
R-squared Score: 0.47635223896831325
```