

Write Python code to implement the following

1. Predict canada's per capita income in year 2020. Use the data file `canada_per_capita_income.csv` file. If required, apply the necessary data processing steps. Using this build a regression model and predict the per capita income for canadian citizens in year 2020

[1]
✓ 5s

```
from google.colab import files
uploaded = files.upload()
```

Choose files `canada_per..._income.csv`
canada_per_capita_income.csv(text/csv) - 874 bytes, last modified: 25/02/2026 - 100% done
Saving `canada_per_capita_income.csv` to `canada_per_capita_income (1).csv`

[2]
✓ 2s

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

[3]
✓ 0s

```
df = pd.read_csv('canada_per_capita_income.csv')
df.head()
```

	year	per capita income (US\$)
0	1970	3399.299037
1	1971	3768.297935
2	1972	4251.175484
3	1973	4804.463248
4	1974	5576.514583

Next steps: [Generate code with df](#) [New interactive sheet](#)

[4]
✓ 0s

```
df.columns = ['year', 'income']
df.head()
```

	year	income
0	1970	3399.299037
1	1971	3768.297935
2	1972	4251.175484
3	1973	4804.463248
4	1974	5576.514583

Next steps: [Generate code with df](#) [New interactive sheet](#)

[5]
✓ 0s

```
X = df[['year']] # Independent variable
y = df['income'] # Dependent variable
```

[6]
✓ 0s

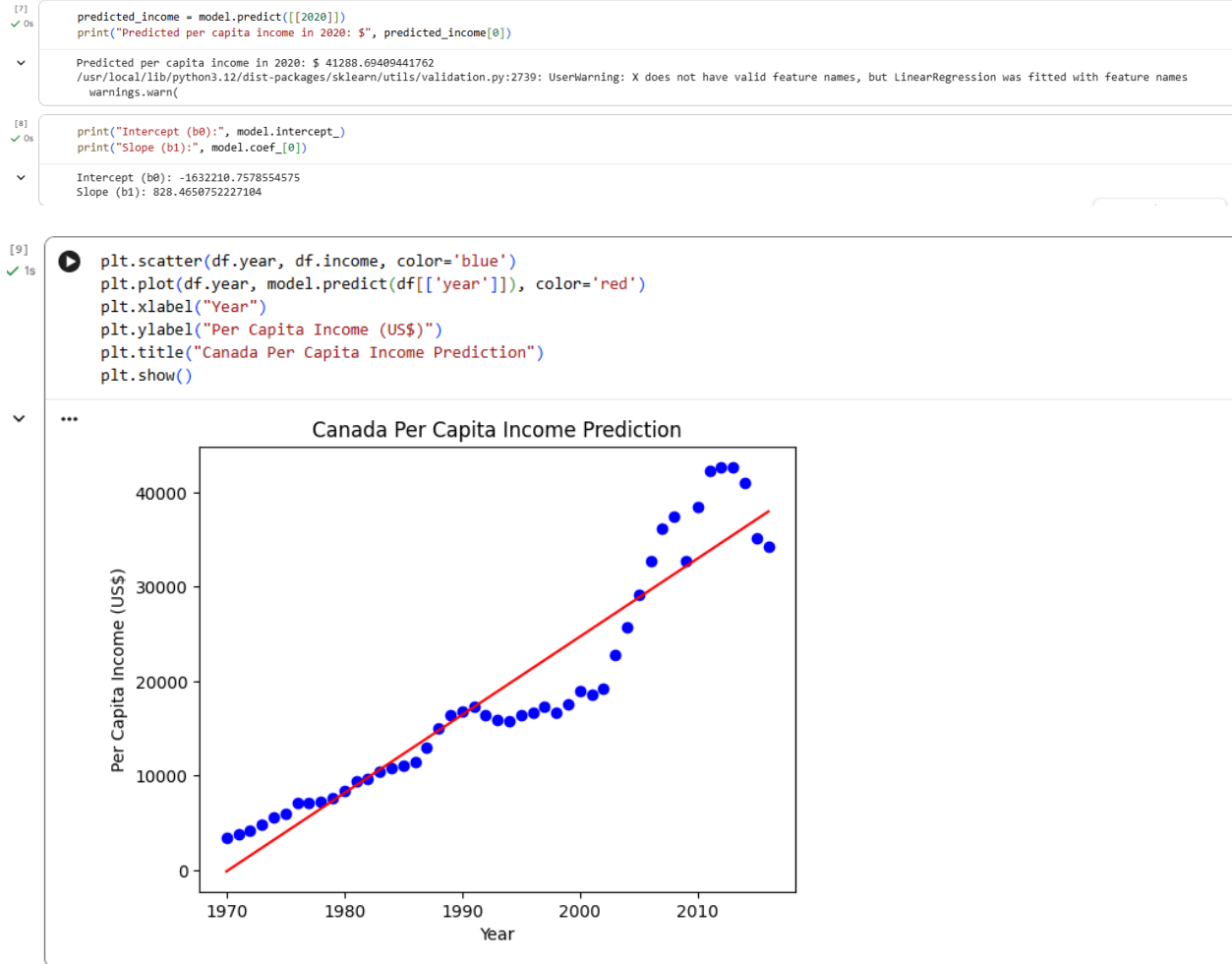
```
model = LinearRegression()
model.fit(X, y)
```

LinearRegression ⓘ
LinearRegression()

[7]
✓ 0s

```
predicted_income = model.predict([[2020]])
print("Predicted per capita income in 2020: $", predicted_income[0])
```

Predicted per capita income in 2020: \$ 41288.69409441762
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(



2. Predict Salary of the employee. Use the data file salary.csv file. If required, apply the necessary data processing steps. Using this build a regression model and predict the salary of the employee with 12 years of experience.

[10]



```
from google.colab import files
uploaded = files.upload()

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

df = pd.read_csv('salary.csv')

print("Before Cleaning:")
print(df.info())

df = df.dropna()

print("\nAfter Cleaning:")
print(df.info())

X = df[['YearsExperience']]
y = df['Salary']

model = LinearRegression()
model.fit(X, y)

salary_12 = model.predict([[12]])

print("\nPredicted Salary for 12 Years Experience:", salary_12[0])

plt.scatter(X, y, color='blue')
plt.plot(X, model.predict(X), color='red')
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Salary Prediction")
plt.show()
```

Choose files salary.csv

salary.csv(text/csv) - 346 bytes, last modified: 25/02/2026 - 100% done

Saving salary.csv to salary.csv

Before Cleaning:

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 30 entries, 0 to 29

Data columns (total 2 columns):

#	Column	Non-Null Count	Dtype
0	YearsExperience	28 non-null	float64
1	Salary	30 non-null	int64

dtypes: float64(1), int64(1)

memory usage: 612.0 bytes

None

After Cleaning:

```
<class 'pandas.core.frame.DataFrame'>
```

Index: 28 entries, 0 to 29

Data columns (total 2 columns):

#	Column	Non-Null Count	Dtype
0	YearsExperience	28 non-null	float64
1	Salary	28 non-null	int64

dtypes: float64(1), int64(1)

memory usage: 672.0 bytes

None

Predicted Salary for 12 Years Experience: 139049.6749539778

/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names

warnings.warn(



- Write Python code to implement the following
Considering the data file **hiring.csv**. The file contains hiring statics for a firm such as experience of candidate, his written test score and personal interview score. Based on these 3 factors, HR will decide the salary. Given this data, you need to build a Multiple Linear Regression model for HR department that can help them decide salaries for future candidates. Using this predict salaries for following candidates,
2 yr experience, 9 test score, 6 interview score
12 yr experience, 10 test score, 10 interview score

[11]
✓ 6s

```
1. # Upload hiring.csv
from google.colab import files
uploaded = files.upload()

# Import libraries
import pandas as pd
from sklearn.linear_model import LinearRegression

# Load dataset
df = pd.read_csv('hiring.csv')

print("Original Data:")
print(df)

word_to_num = {
    'zero': 0,
    'one': 1,
    'two': 2,
    'three': 3,
    'four': 4,
    'five': 5,
    'six': 6,
    'seven': 7,
    'eight': 8,
    'nine': 9,
    'ten': 10
}

df['experience'] = df['experience'].replace(word_to_num)

# Convert to numeric
df['experience'] = pd.to_numeric(df['experience'], errors='coerce')

# Fill missing values
df['experience'] = df['experience'].fillna(0)
df['test_score(out of 10)'] = df['test_score(out of 10)'].fillna(df['test_score(out of 10)'].mean())

print("\nCleaned Data:")
print(df)
```

```
X = df[['experience', 'test_score(out of 10)', 'interview_score(out of 10)']]
y = df['salary($)']
```

```
model = LinearRegression()
model.fit(X, y)
```

```
salary1 = model.predict([[2, 9, 6]])
salary2 = model.predict([[12, 10, 10]])
```

```
print("\nPredicted Salary for (2 yr, 9 test, 6 interview):", salary1[0])
print("Predicted Salary for (12 yr, 10 test, 10 interview):", salary2[0])
```

```

... Choose files hiring.csv
hiring.csv(text/csv) - 198 bytes, last modified: 25/02/2026 - 100% done
Saving hiring.csv to hiring.csv
Original Data:
  experience  test_score(out of 10)  interview_score(out of 10)  salary($)
0         NaN                    8.0                        9         50000
1         NaN                    8.0                        6         45000
2         five                    6.0                        7         60000
3         two                     10.0                       10         65000
4         seven                    9.0                        6         70000
5         three                    7.0                       10         62000
6         ten                      NaN                        7         72000
7        eleven                    7.0                        8         80000

Cleaned Data:
  experience  test_score(out of 10)  interview_score(out of 10)  salary($)
0         0.0                    8.000000                        9         50000
1         0.0                    8.000000                        6         45000
2         5.0                    6.000000                        7         60000
3         2.0                    10.000000                       10         65000
4         7.0                    9.000000                        6         70000
5         3.0                    7.000000                       10         62000
6         10.0                   7.857143                        7         72000
7         0.0                    7.000000                        8         80000

Predicted Salary for (2 yr, 9 test, 6 interview): 57403.24743480464
Predicted Salary for (12 yr, 10 test, 10 interview): 79095.98147979788
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(

```

- 4. Considering the data file **1000_companies.csv**. The file contains profit statics for a firm such as R&D Spend, Administration, Marketing Spend and State. Based on these four factors build a Multiple Linear Regression model to predict the profit. Using this predict profit for following,
 - 91694.48 R&D Spend, 515841.3 Administration, 11931.24 Marketing Spend, Florida State**
- Note: If required, apply the necessary data processing steps to data files.



```
from google.colab import files
uploaded = files.upload()

import pandas as pd
from sklearn.linear_model import LinearRegression

df = pd.read_csv('1000_Companies.csv')

print("Original Data:")
print(df.head())

df = df.dropna()

df = pd.get_dummies(df, columns=['State'], drop_first=True)

print("\nEncoded Data:")
print(df.head())

X = df.drop('Profit', axis=1)
y = df['Profit']

model = LinearRegression()
model.fit(X, y)

print("\nColumns Used for Training:")
print(X.columns)
|
profit = model.predict([[91694.48, 515841.3, 11931.24, 1, 0]])

print("\nPredicted Profit:", profit[0])
```

```
*** Choose files | 1000_Companies.csv
1000_Companies.csv(text/csv) - 52203 bytes, last modified: 25/02/2026 - 100% done
Saving 1000_Companies.csv to 1000_Companies.csv
Original Data:
  R&D Spend Administration Marketing Spend State Profit
0 165349.20 136897.80 471784.10 New York 192261.83
1 162597.70 151377.59 443898.53 California 191792.06
2 153441.51 101145.55 407934.54 Florida 191050.39
3 144372.41 118671.85 383199.62 New York 182901.99
4 142107.34 91391.77 366168.42 Florida 166187.94

Encoded Data:
  R&D Spend Administration Marketing Spend Profit State_Florida \
0 165349.20 136897.80 471784.10 192261.83 False
1 162597.70 151377.59 443898.53 191792.06 False
2 153441.51 101145.55 407934.54 191050.39 True
3 144372.41 118671.85 383199.62 182901.99 False
4 142107.34 91391.77 366168.42 166187.94 True

  State_New York
0 True
1 False
2 False
3 True
4 False

Columns Used for Training:
Index(['R&D Spend', 'Administration', 'Marketing Spend', 'State_Florida',
      'State_New York'],
      dtype='object')

Predicted Profit: 510570.9926108309
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(
```