**Introduction to Data Science**

**Name:** Aoun-Haider

**ID:** FA21-BSE-133

**Section:** A

**Assignment:** 03

**Question:** 01

|  |  |  |
| --- | --- | --- |
| **Age** | **Income** | **Savings** |
| 20 | 100,000 | 50,000 |
| 22 | 120,000 | 40,000 |
| 32 | 80,000 | 5,000 |
| 40 | 350,000 | 10,000 |
| 50 | 50,000 | 0 |
| 36 | 150,000 | 30,000 |
| 47 | 250,000 | 8,000 |

1. You are required to generate the R and R2 factors of the Linear and Polynomial regressions respectively and fill in the following data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Predicted Income | | Predicted Savings | |
|  | Linear Regression | Polynomial Regression | Linear Regression | Polynomial Regression |
| 15 | 105356.50 | -4270.84 | 49679.68 | 60191.52 |
| 55 | 207470.44 | 85699.15 | -7998.56 | 3677.72 |
| 65 | 232998.92 | -94231.53 | -22418.12 | 8959.034 |

1. By using the percentile calculation please find out the income and saving of the following percentage of the population:

|  |  |  |
| --- | --- | --- |
| Percentile | Income | Savings |
| 25 | 90000 | 6500 |
| 50 | 120000 | 10000 |
| 75 | 200000 | 35000 |

**Linear Regression R Factors:**

Age vs. Income: 0.277

Age vs. Savings: -0.855

**Polynomial R^2 Factors:**

Age vs. Income: 0.221

Age vs. Savings: 0.771

**Implementation:**

import numpy

from sklearn.metrics import r2\_score

from scipy import stats

age = [20, 22, 32, 40, 50, 36, 47]

income = [100000, 120000, 80000, 350000, 50000, 150000, 250000]

savings = [50000, 40000, 5000, 10000, 0, 30000, 8000]

slope\_income, intercept\_income, r\_income, p\_income, std\_err\_income = stats.linregress(age, income)

# Linear regression between age and savings

slope\_savings, intercept\_savings, r\_savings, p\_savings, std\_err\_savings = stats.linregress(age, savings)

# List of ages for which expected income is required

target\_ages = [15, 55, 65]

# Function that calculates linear regression predicted income

def calculate\_linear\_income(ages):

return slope\_income \* ages + intercept\_income

def calculate\_linear\_savings(ages):

return slope\_savings \* ages + intercept\_savings

linear\_income\_predictions = list(map(int, map(calculate\_linear\_income, target\_ages)))

linear\_savings\_predictions = list(map(int, map(calculate\_linear\_savings, target\_ages)))

# Calculate polynomial regression predicted Income

income\_model = numpy.poly1d(numpy.polyfit(age, income, 2))

# Calculate polynomial regression predicted Savings

savings\_model = numpy.poly1d(numpy.polyfit(age, savings, 2))

polynomial\_income\_predictions = list(map(int, map(income\_model, target\_ages)))

polynomial\_savings\_predictions = list(map(int, map(savings\_model, target\_ages)))

# Print Population Data

print("Population Data")

print("Age:", end="\t")

for a in age:

print(a, end="\t")

print("\nIncome:", end="\t")

for i in income:

print(i, end="\t")

print("\nSavings:", end="\t")

for s in savings:

print(s, end="\t")

# Print Linear and Polynomial Regression Results

print("\n(1)")

print("Linear Regression R Factors:")

print("Age vs. Income:", round(r\_income, 3))

print("Age vs. Savings:", round(r\_savings, 3))

print("Polynomial R^2 Factors:")

print("Age vs. Income:", round(r2\_score(income, income\_model(age)), 3))

print("Age vs. Savings:", round(r2\_score(savings, savings\_model(age)), 3))

# Print Predicted Income and Savings

print("\nPredicted Income values by Linear and Polynomial regression for the following ages:")

print("Ages:", target\_ages)

print("Linear Regression Predicted Income:", end="\t")

for i in linear\_income\_predictions:

print(i, end="\t")

print("\nPolynomial Regression Predicted Income:", end="\t")

for i in polynomial\_income\_predictions:

print(i, end="\t")

print("\nPredicted Savings values by Linear and Polynomial regression for the following ages:")

print("Ages:", target\_ages)

print("Linear Regression Predicted Savings:", end="\t")

for s in linear\_savings\_predictions:

print(s, end="\t")

print("\nPolynomial Regression Predicted Savings:", end="\t")

for s in polynomial\_savings\_predictions:

print(s, end="\t")

# Print Percentile Calculations

print("\n(2)")

print("Calculating Income and Savings of the population by percentile calculation:")

percentiles = [25, 50, 75]

print("Population Percentile\tIncome\t\tSavings")

for percentile in percentiles:

income\_percentile = numpy.percentile(income, percentile)

savings\_percentile = numpy.percentile(savings, percentile)

print(f"{percentile}%\t\t\t{income\_percentile} \t{savings\_percentile}")