Experiment 2 - Linear Regression

April 28, 2023

1 Experiment Details

1.1 Submitted By

Desh Iyer, 500081889, Year III, AI/ML(H), B5

[ ]:

```python
from sklearn.linear_model import LinearRegression
from sklearn.datasets import make_regression
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:

```python
# Function to print the summary statistics of a variable
def printSummary(var, x):
    print("\nSummary Statistics for '{}'".format(var))
    print(f"Mean = {np.mean(x)}\nStandard Deviation = {np.std(x)}")
```

[ ]:

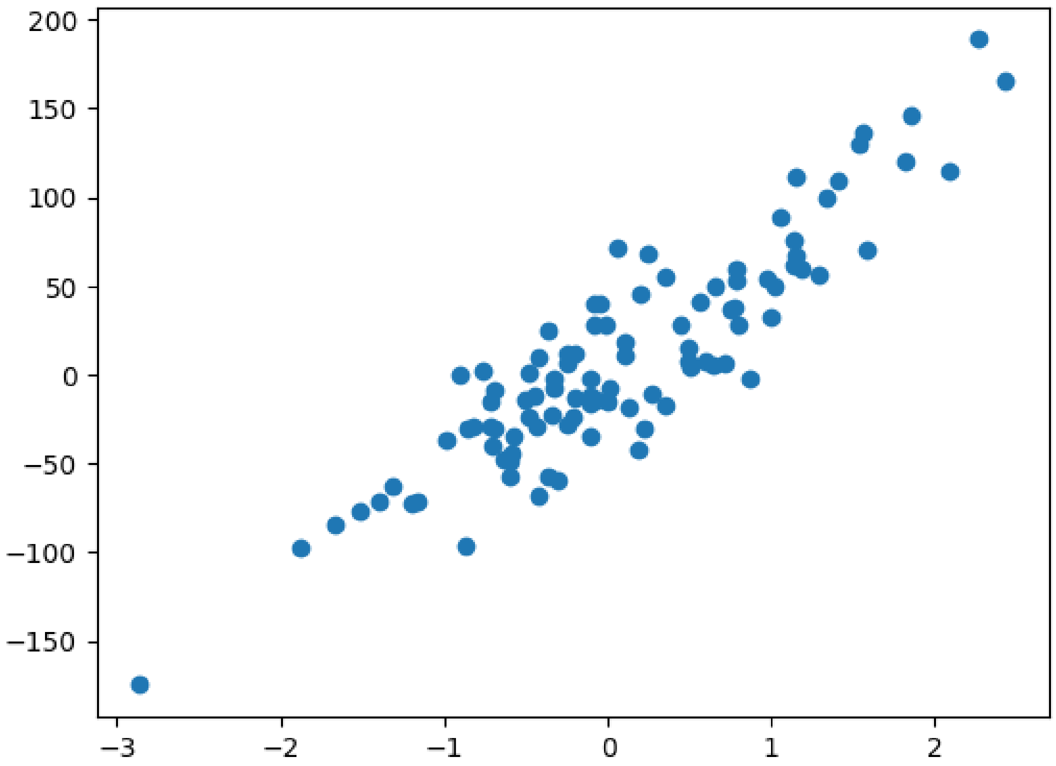
```python
# Create training dataset to test linear regression
X, t = make_regression(100, 1, shuffle=True, bias=0, noise=25, random_state=5)

# Printing the summary statistics
printSummary('x', X)
printSummary('t', t)

# Plotting scatter plot
plt.scatter(X, t)
plt.show()
```

```
Summary Statistics for 'x'
Mean = 0.09154664386777672
Standard Deviation = 0.9309767032155012

Summary Statistics for 't'
Mean = 9.478600133225886
Standard Deviation = 60.8084611275079
```



[ ]:

```
# Calculate mean of both variables
xMean = np.mean(X)
tMean = np.mean(t)

# Calculate deltaX, deltaT and deltaXSquare
deltaX = [float(i - xMean) for i in X]
deltaXSquare = [float(pow(i, 2)) for i in X]
deltaT = [float(i - tMean) for i in t]

# Calculate the list in the numerator
productDelta = []
for i in range(0, len(deltaT)):
    productDelta.append(deltaX[i] * deltaT[i])
```

[ ]:

```
# Calculate slope and y-intercept
m = sum(productDelta) / sum(deltaXSquare)
c = tMean - (m * xMean)

# Print final output
print('\n\nSlope = {}\nBias (or) y-Intercept = {}\n'.format(m, c))
```

```
Slope = 58.27910999576034
Bias (or) y-Intercept = 4.143343205513028
```

[ ]:

```python
# Note: slope (m) ==> beta_1 || y-intercept (c) ==> beta_0

# Using an alternate method using sklearn sub-modules
reg = LinearRegression().fit(X, t)

print("Regression score:", reg.score(X, t))

# Obtain m and c from attributes of variable reg
slope = float(reg.coef_)
bias = reg.intercept_

# Printing details
print("Slope = {}\nBias / y-Intercept = {}".format(slope, bias))
print(f'\nMethod #1 (From scratch)\nm = {m} | c = {c}')
print(f'\nMethod #2 (Using sklearn)\nm = {slope} | c = {bias}')
```

```
Regression score: 0.8115849010376874
Slope = 58.84264401600654
Bias / y-Intercept = 4.091753557254172

Method #1 (From scratch)
m = 58.27910999576034 | c = 4.143343205513028

Method #2 (Using sklearn)
m = 58.84264401600654 | c = 4.091753557254172
```