feature-engineering-5-6

August 13, 2023

[ ]:
```
# Q1. Pearson correlation coefficient is a measure of the linear relationship␣
↪between two variables.

# Suppose you have collected data on the amount of time students spend studying␣
↪for an exam and their final exam scores.

# Calculate the Pearson correlation coefficient between these two variables and␣
↪interpret the result.
```

[1]:
```
import numpy as np

# Sample data for study time and exam scores
study_time = [10, 15, 20, 5, 8, 12, 18, 6, 9, 14]
exam_scores = [85, 90, 92, 70, 78, 88, 94, 75, 80, 89]

# Calculate the Pearson correlation coefficient
correlation_coefficient = np.corrcoef(study_time, exam_scores)[0, 1]

print("Pearson Correlation Coefficient:", correlation_coefficient)
```

```
Pearson Correlation Coefficient: 0.9416113724269137
```

If the calculated Pearson correlation coefficient is close to 1 (positive value), it indicates a strong positive linear relationship between study time and exam scores. This means that students who study more tend to score higher on the exam.

If the coefficient is close to -1 (negative value), it indicates a strong negative linear relationship. This would imply that students who study more tend to score lower on the exam, which is unlikely in this context.

If the coefficient is close to 0, it suggests no strong linear relationship between study time and exam scores. This means that variations in study time do not correspond to consistent variations in exam scores.

[ ]:
```
# Q2. Spearman's rank correlation is a measure of the monotonic relationship␣
↪between two variables.

# Suppose you have collected data on the amount of sleep individuals get each␣
↪night and their overall job satisfaction level on a scale of 1 to 10.
```

```
# Calculate the Spearman's rank correlation between these two variables and␣
 ↪interpret the result.
```

[2]:
```
import numpy as np
from scipy.stats import spearmanr

# Sample data for sleep and job satisfaction
sleep = [7, 6, 8, 5, 4, 6, 7, 8, 7, 5]
job_satisfaction = [8, 6, 9, 5, 4, 7, 8, 9, 6, 5]

# Calculate the Spearman's rank correlation
correlation_coefficient, p_value = spearmanr(sleep, job_satisfaction)

print("Spearman's Rank Correlation Coefficient:", correlation_coefficient)
print("p-value:", p_value)
```

```
Spearman's Rank Correlation Coefficient: 0.9436153964994065
p-value: 4.129787098050796e-05
```

The Spearman's rank correlation coefficient ranges from -1 to 1, where:

1 indicates a perfect monotonic positive relationship. -1 indicates a perfect monotonic negative relationship. 0 indicates no monotonic relationship.

The p-value indicates the significance of the correlation. A low p-value (typically < 0.05) suggests that the observed correlation is statistically significant.

[ ]:
```
# Q3. Suppose you are conducting a study to examine the relationship between␣
 ↪the number of hours of exercise per week and body mass index (BMI) in a␣
 ↪sample of adults.

# You collected data on both variables for 50 participants.

# Calculate the Pearson correlation coefficient and the Spearman's rank␣
 ↪correlation between these two variables and compare the results.
```

[3]:
```
import numpy as np
from scipy.stats import pearsonr, spearmanr

# Sample data for exercise hours and BMI
exercise_hours = [4, 5, 3, 6, 2, 4, 5, 3, 2, 6, 1, 2, 3, 5, 4, 6, 1, 2, 3, 4,␣
 ↪5, 3, 2, 6, 4, 5, 2, 3, 1, 6, 3, 4, 5, 2, 1, 6, 4, 3, 5, 2, 1, 4, 3, 6, 5,␣
 ↪2, 1, 4, 3, 5]
bmi = [24, 25, 23, 26, 22, 24, 25, 23, 22, 26, 21, 22, 23, 25, 24, 26, 21, 22,␣
 ↪23, 24, 25, 23, 22, 26, 24, 25, 22, 23, 21, 26, 23, 24, 25, 22, 21, 26, 24,␣
 ↪23, 25, 22, 21, 24, 23, 26, 25, 22, 21, 24, 23, 25]
```

```
# Calculate Pearson correlation coefficient
pearson_coefficient, pearson_p_value = pearsonr(exercise_hours, bmi)

# Calculate Spearman's rank correlation coefficient
spearman_coefficient, spearman_p_value = spearmanr(exercise_hours, bmi)

print("Pearson Correlation Coefficient:", pearson_coefficient)
print("Pearson p-value:", pearson_p_value)
print("Spearman's Rank Correlation Coefficient:", spearman_coefficient)
print("Spearman p-value:", spearman_p_value)
```

```
Pearson Correlation Coefficient: 1.0
Pearson p-value: 0.0
Spearman's Rank Correlation Coefficient: 1.0
Spearman p-value: 0.0
```

[ ]:
```
# Q4. A researcher is interested in examining the relationship between the␣
 ↪number of hours individuals spend watching television per day and their␣
 ↪level of physical activity.

# The researcher collected data on both variables from a sample of 50␣
 ↪participants.

# Calculate the Pearson correlation coefficient between these two variables.
```

[4]:
```
import numpy as np

# Sample data for TV hours and physical activity level
tv_hours = [3, 4, 5, 2, 6, 3, 4, 5, 1, 2, 6, 3, 4, 5, 2, 6, 3, 4, 5, 2, 6, 3,␣
 ↪4, 5, 2, 6, 3, 4, 5, 2, 6, 3, 4, 5, 2, 6, 3, 4, 5, 2, 6, 3, 4, 5, 2, 6, 3, 4]
physical_activity = [2, 3, 4, 1, 5, 2, 3, 4, 1, 2, 5, 2, 3, 4, 1, 5, 2, 3, 4,␣
 ↪1, 5, 2, 3, 4, 1, 5, 2, 3, 4, 1, 5, 2, 3, 4, 1, 5, 2, 3, 4, 1, 5, 2, 3, 4,␣
 ↪1, 5, 2, 3]

# Calculate the Pearson correlation coefficient
correlation_coefficient = np.corrcoef(tv_hours, physical_activity)[0, 1]

print("Pearson Correlation Coefficient:", correlation_coefficient)
```

```
Pearson Correlation Coefficient: 0.9908066573771616
```

[ ]:
```
# Q5. A survey was conducted to examine the relationship between age and␣
 ↪preference for a particular brand of soft drink.

# The survey results are shown below:

# Age(Years)       Soft drink Preference
```

```
# 25                    Coke
# 42                    Pepsi
# 37                Mountain dew
# 19                    Coke
# 31                    Pepsi
# 28                    Coke
```

[ ]:
```
# Q6. A company is interested in examining the relationship between the number␣
 ↪of sales calls made per day and the number of sales made per week.

# The company collected data on both variables from a sample of 30 sales␣
 ↪representatives.

# Calculate the Pearson correlation coefficient between these two variables.
```

[8]:
```
import numpy as np

# Sample data for sales calls per day and sales per week
sales_calls_per_day = [20, 15, 18, 25, 12, 24, 16, 22, 19, 14, 21, 17, 23, 27,␣
 ↪13, 28, 10, 26, 11, 29, 30, 9, 31, 8, 32, 7, 33, 6, 34, 5]
sales_per_week = [8, 6, 7, 10, 4, 9, 6, 8, 7, 5, 8, 6, 9, 11, 4, 12, 3, 10, 4,␣
 ↪11, 12, 3, 13, 2, 14, 1, 15, 0, 16, 1]

# Calculate the Pearson correlation coefficient
correlation_coefficient = np.corrcoef(sales_calls_per_day, sales_per_week)[0, 1]

print("Pearson Correlation Coefficient:", correlation_coefficient)
```

```
Pearson Correlation Coefficient: 0.9911908887402545
```

[ ]:
```

```