

# Statistics for AI - Exercise

Ngày 4 tháng 8 năm 2024

## 1. BASIC PROBABILITY

**Câu hỏi 1:** Kết quả của đoạn chương trình tính mean sau đây là:

Cho Data  $X = \{2, 0, 2, 2, 7, 4, -2, 5, -1, -1\}$

Hãy hoàn thiện function `compute_mean()` để tính mean  $\mu$  của  $X$  đã cho.

- Data:  $X = \{x_1, \dots, x_N\}$

- Mean:  $\mu = \frac{1}{N} \sum_{i=1}^N x_i$

```
1 ### Question 1
2 import numpy as np
3
4 def compute_mean(X):
5     #your code here *****
6     return
7
8 X = [2, 0, 2, 2, 7, 4, -2, 5, -1, -1]
9
10 print("Mean : ", compute_mean(X))
```

Kết quả của đoạn chương trình trên là:

- a) 1.8
- b) 2.8
- c) 2.8
- d) 3.8

**Câu hỏi 2:** Kết quả của đoạn chương trình tính median sau đây là:

Cho Data  $X = \{1, 5, 4, 4, 9, 13\}$ . Hãy hoàn thiện function `compute_median()` để tìm median của  $X$  đã cho.

- Data:  $X = \{x_1, \dots, x_N\}$

- Median:

– Sort  $X \rightarrow S$  (tăng dần)

- If  $N$  is odd:  $m = S_{\frac{N+1}{2}}$
- If  $N$  is even:  $m = \frac{1}{2}(S_{\frac{N}{2}} + S_{\frac{N}{2}+1})$

```

1 ### Question 2
2
3 def compute_median(X):
4     size = len(X)
5     X = np.sort(X)
6     print(X)
7     if (size % 2 == 0):
8         return # your code here *****
9     else:
10        return # your code here *****
11
12 X = [1, 5, 4, 4, 9, 13]
13 print("Median: ", compute_median(X))

```

Kết quả của đoạn chương trình trên là:

- a) 4.0
- b) 4.5
- c) 4.6
- d) 4.7

**Câu hỏi 3:** Kết quả của đoạn chương trình tính variance và standard deviation sau đây là:

Cho Data  $X = \{171, 176, 155, 167, 169, 182\}$

Hãy hoàn thiện function **compute\_std()** để tìm standard deviation  $\sigma$  của  $X$  đã cho.

- Data:  $X = \{x_1, \dots, x_N\}$
- Mean:  $\mu = \frac{1}{N} \sum_{i=1}^N x_i$
- Variance:  $var(X) = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$
- Standard Deviation:  $\sigma = \sqrt{var(X)}$

```

1 ### Question 3
2
3 def compute_std(X):
4     mean = compute_mean(X)
5     variance = 0
6     # your code here *****
7
8     return np.sqrt(variance)
9
10 X = [ 171, 176, 155, 167, 169, 182]
11 print(compute_std(X))

```

Kết quả của đoạn chương trình trên là:

- a) 8.0
- b) 8.23
- c) 8.33
- d) 8.13

**Câu hỏi 4:** Kết quả của đoạn chương trình tính correlation coefficient sau đây là:

Cho Data  $X = \{-2, -5, -11, 6, 4, 15, 9\}$  và

$Y = \{4, 25, 121, 36, 16, 225, 81\}$

Hoàn thiện function `compute_correlation_coefficient()` để tìm correlation coefficient của  $X$  và  $Y$  đã cho?

- Random variables  $X, Y$ :  $X = \{x_1, \dots, x_N\}$   $Y = \{y_1, \dots, y_N\}$

- Correlation Coefficient: 
$$p_{xy} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$
$$= \frac{N(\sum_i x_i y_i) - (\sum_i x_i)(\sum_i y_i)}{\sqrt{N\sum_i x_i^2 - (\sum_i x_i)^2} \sqrt{N\sum_i y_i^2 - (\sum_i y_i)^2}}$$

```

1 ### Question 4
2
3 def compute_correlation_coefficient(X, Y):
4     N = len(X)
5     numerator = 0
6     denominator = 0
7     # your code here *****
8
9     return np.round(numerator / denominator, 2)
10
11 X = np.asarray([-2, -5, -11, 6, 4, 15, 9])
12 Y = np.asarray([4, 25, 121, 36, 16, 225, 81])
13 print("Correlation: ", compute_correlation_coefficient(X,Y))

```

Kết quả của đoạn chương trình trên là:

- a) 0.41
- b) 0.44
- c) 0.43
- d) 0.42

## 2. TABULAR DATA ANALYSIS

**Câu hỏi 5:** Kết quả của đoạn chương trình sau đây là:

```
1 # Download dataset: !gdown 1iA0WmVfW88HyJvTBSQDI5vesf-pgKabq
2 import pandas as pd
3
4 data = pd.read_csv("advertising.csv")
5
6 def correlation(x, y):
7     # Your code here #
8
9 # Example usage:
10 x = data['TV']
11 y = data['Radio']
12 corr_xy = correlation(x, y)
13 print(f"Correlation between TV and Sales: {round(corr_xy, 2)}")
```

- a) 0.04
- b) 0.05
- c) 0.06
- d) 0.07

**Câu hỏi 6:** Kết quả của đoạn chương trình sau đây là:

```
1 data = pd.read_csv("advertising.csv")
2
3 def correlation(x, y):
4     # Your code here #
5
6 features = ['TV', 'Radio', 'Newspaper']
7
8 for feature_1 in features:
9     for feature_2 in features:
10         correlation_value = correlation(data[feature_1], data[feature_2])
11         print(f"Correlation between {feature_1} and {feature_2}: {round(
            correlation_value, 2)}")
```

- a) TV and TV: -1.0  
TV and Radio: 0.05  
TV and Newspaper: 0.06  
Radio and TV: -0.05  
Radio and Radio: 1.0  
Radio and Newspaper: 0.35  
Newspaper and TV: -0.06  
Newspaper and Radio: 0.35  
Newspaper and Newspaper: 1.0

b) TV and TV: 1.0  
TV and Radio: 0.05  
TV and Newspaper: 0.06  
Radio and TV: 0.05  
Radio and Radio: 1.0  
Radio and Newspaper: 0.35  
Newspaper and TV: -0.06  
Newspaper and Radio: 0.35  
Newspaper and Newspaper: 1.0

c) TV and TV: 1.0  
TV and Radio: 0.05  
TV and Newspaper: 0.06  
Radio and TV: 0.05  
Radio and Radio: 0.0  
Radio and Newspaper: 0.35  
Newspaper and TV: 0.06  
Newspaper and Radio: 0.35  
Newspaper and Newspaper: 1.0

d) TV and TV: 1.0  
TV and Radio: 0.05  
TV and Newspaper: 0.06  
Radio and TV: 0.05  
Radio and Radio: 1.0  
Radio and Newspaper: 0.35  
Newspaper and TV: 0.06  
Newspaper and Radio: 0.35  
Newspaper and Newspaper: 1.0

**Câu hỏi 7:** Hãy cho biết đoạn code phù hợp với kết quả sau đây là:

```
1 data = pd.read_csv("advertising.csv")
2 x = data['Radio']
3 y = data['Newspaper']
4
5 result = # Your code here #
6 print(result)
7
8 # Expected output: [[1.          0.35410375]
9 #                  [0.35410375 1.          ]]
```

- a) np.correlation(x, y)
- b) np.coefficient(x, y)
- c) np.corrcoef(x, y)
- d) correlation(x,y)

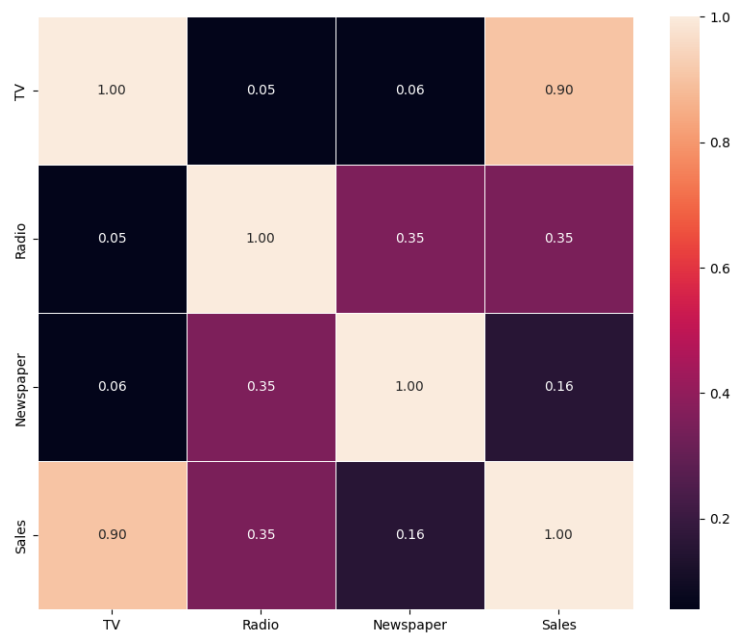
**Câu hỏi 8:** Hãy cho biết đoạn code phù hợp với kết quả sau đây là:

```
1 data = pd.read_csv("advertising.csv")
2
3 # Your code here #
```

	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.901208
Radio	0.054809	1.000000	0.354104	0.349631
Newspaper	0.056648	0.354104	1.000000	0.157960
Sales	0.901208	0.349631	0.157960	1.000000

- a) np.corr(x, y)
- b) data.corr(x, y)
- c) data.correlation(x, y)
- d) data.corr()

**Câu hỏi 9:** Hãy cho biết đoạn code phù hợp với kết quả sau đây là:



```
1 #Question 09
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
```

```
5 data = pd.read_csv("advertising.csv")
6
7 plt.figure(figsize=(10,8))
8 # Your code here #
9 plt.show()
```

- a) `sns.heatmap(data, annot=True, fmt=".2f", linewidth=.5)`
- b) `data_corr = kết quả câu số 08`  
`sns.heatmap(data_corr, annot=True, fmt=".2f", linewidth=.5)`
- c) `data_corr = kết quả câu số 08`  
`sns.hatmap(data_corr, annot=False, fmt=".2f", linewidth=.5)`
- d) `data_corr = kết quả câu số 08`  
`sns.heatmap(data_corr @ data, annot=False, fmt=".2f", linewidth=.5,)`

### 3. TEXT RETRIEVAL

**Câu hỏi 10:** Kết quả của đoạn chương trình đọc file và sử dụng TF-IDF để biểu diễn văn bản thành vector sau đây là:

```
1 # Download dataset: !gdown 1jh2p2DlaWsDo_vEWIcTrNh3mUuXd-cw6
2 import pandas as pd
3 import numpy as np
4 from sklearn.metrics.pairwise import cosine_similarity
5 from sklearn.feature_extraction.text import TfidfVectorizer
6
7 vi_data_df = pd.read_csv("./vi_text_retrieval.csv")
8 context = vi_data_df['text']
9 context = [doc.lower() for doc in context]
10
11 tfidf_vectorizer = TfidfVectorizer()
12 context_embedded = # Your Code *****
13 context_embedded.toarray()[7][0]
```

- a) 0.30
- b) 0.31
- c) 0.32
- d) 0.33

**Câu hỏi 11:** Kết quả của đoạn chương trình tính độ tương đồng cosine là:

```
1 def tfidf_search(question, tfidf_vectorizer, top_d=5):
2     # lowercasing before encoding
3     query_embedded = # Your Code Here *****
4     cosine_scores = # Your Code Here *****
5
6     # Get top k cosine score and index its
7     results = []
8     for idx in cosine_scores.argsort()[-top_d:][::-1]:
9         doc_score = {
10             'id': idx,
11             'cosine_score': cosine_scores[idx]
12         }
13         results.append(doc_score)
14     return results
15
16 question = vi_data_df.iloc[0]['question']
17 results = tfidf_search(question, tfidf_vectorizer, top_d=5)
18 results[0]['cosine_score']
```

- a) 0.60
- b) 0.61
- c) 0.62
- d) 0.63



**Câu hỏi 12:** Kết quả của đoạn chương trình tính độ tương đồng correlation là:

```
1 def corr_search(question, tfidf_vectorizer, top_d=5):
2     # lowercasing before encoding
3     query_embedded = # Your Code Here *****
4     corr_scores = # Your Code Here *****
5     corr_scores = corr_scores[0][1:]
6     # Get top k correlation score and index its
7     results = []
8     for idx in corr_scores.argsort()[-top_d:][::-1]:
9         doc = {
10             'id': idx,
11             'corr_score': corr_scores[idx]
12         }
13         results.append(doc)
14     return results
15
16 question = vi_data_df.iloc[0]['question']
17 results = corr_search(question, tfidf_vectorizer, top_d=5)
18 results[1]['corr_score']
```

- a) 0.20
- b) 0.21
- c) 0.22
- d) 0.23