AI VIET NAM – COURSE 2024

Statistic Exercise

(Correlation, Cosine and Applications)

Ngày 27 tháng 7 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 μ của X đã cho.

- Data: $X = \{x_1, ..., x_N\}$
- Mean: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$

```python
### Question 1
import numpy as np

def compute_mean(X):
    #your code here **********************
    return

X = [2, 0, 2, 2, 7, 4, -2, 5, -1, -1]

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: $X = \{x_1, ..., x_N\}$
- Median:

- Sort $X \longrightarrow S$ **(tăng dần)**
- If N is odd: $m = S_{\frac{N+1}{2}}$
- If N is even: $m = \frac{1}{2}(S_{\frac{N}{2}} + S_{\frac{N}{2}+1})$

```
### Question 2

def compute_median(X):
  size = len(X)
  X = np.sort(X)
  print(X)
  if (size % 2 == 0):
    return # your code here **************************
  else:
    return # your code here **************************

X = [1, 5, 4, 4, 9, 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 σ của X đã cho.

- Data: $X = \{x_1, ..., x_N\}$
- 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: $\sigma = \sqrt{var(X)}$

```
### Question 3

def compute_std(X):
  mean = compute_mean(X)
  variance = 0
  # your code here ******************

  return np.sqrt(variance)

X = [ 171, 176, 155, 167, 169, 182]
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_cofficient()** để tìm correlation coefficient của X và Y đã cho?

- Random variables X, Y: $X = \{x_1, ..., x_N\}$ $Y = \{y_1, ..., y_N\}$
- Correlation Coefficient: $p_{xy} = \dfrac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$
  $= \dfrac{N(\Sigma_i x_i y_i) - (\Sigma_i x_i)(\Sigma_i y_i)}{\sqrt{N\Sigma_i x_i^2 - (\Sigma_i x_i)^2}\sqrt{N\Sigma_i y_i^2 - (\Sigma_i y_i)^2}}$

```
### Question 4

def compute_correlation_cofficient(X, Y):
  N = len(X)
  numerator = 0
  denominator = 0
  # your code here ****************

  return np.round(numerator / denominator,2)

X = np.asarray([-2, -5, -11, 6, 4, 15, 9])
Y = np.asarray([4, 25, 121, 36, 16, 225, 81])
print("Correlation: ", compute_correlation_cofficient(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:

```
# Download dataset: !gdown 1iA0WmVfW88HyJvTBSQDI5vesf-pgKabq
import pandas as pd

data = pd.read_csv("advertising.csv")

def correlation(x, y):
  # Your code here #

# Example usage:
x = data['TV']
y = data['Radio']
corr_xy = correlation(x, y)
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:

```
data = pd.read_csv("advertising.csv")

def correlation(x, y):
  # Your code here #

features = ['TV', 'Radio', 'Newspaper']

for feature_1 in features:
  for feature_2 in features:
      correlation_value = correlation(data[feature_1], data[feature_2])
      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:

```
data = pd.read_csv("advertising.csv")
x = data['Radio']
y = data['Newspaper']

result = # Your code here #
print(result)

# Expected output: [[1.         0.35410375]
#                   [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:

```
data = pd.read_csv("advertising.csv")

# 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

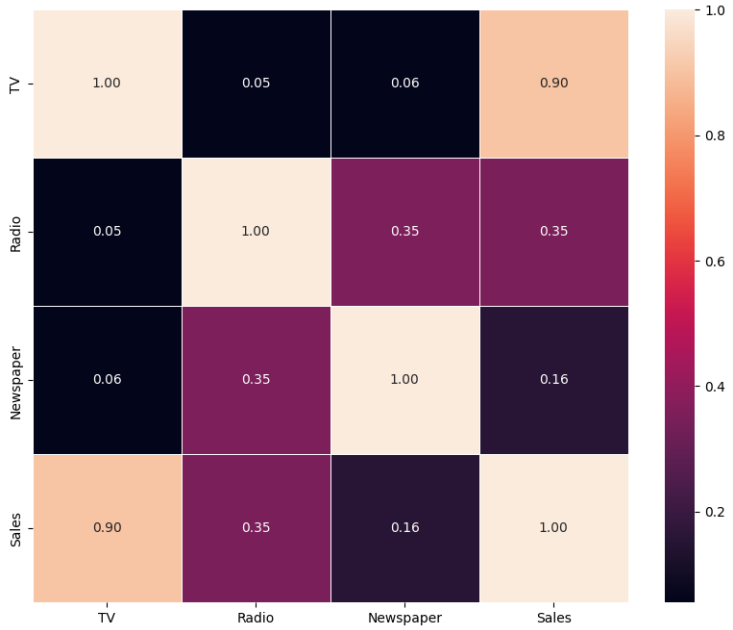
a) np.corr(x, y)

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:




```
#Question 13
import matplotlib.pyplot as plt
import seaborn as sns

```

```
data = pd.read_csv("advertising.csv")

plt.figure(figsize=(10,8))
# Your code here #
plt.show()
```

a) sns.heatmap(data, annot=True, fmt=".2f", linewidth=.5)

b) data_corr = kết quả câu số 12
sns.heatmap(data_corr, annot=False, fmt=".2f", linewidth=.5)

c) data_corr = kết quả câu số 11
sns.hatmap(data_corr, annot=False, fmt=".2f", linewidth=.5)

d) data_corr = kết quả câu số 12
sns.heatmap(data_corr, 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:

```
# Download dataset: !gdown 1jh2p2DlaWsDo_vEWIcTrNh3mUuXd-cw6
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer

vi_data_df = pd.read_csv("./vi_text_retrieval.csv")
context = vi_data_df['text']
context = [doc.lower() for doc in context]

tfidf_vectorizer = TfidfVectorizer()
context_embedded = # Your Code *******
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:

```
def tfidf_search(question, tfidf_vectorizer, top_d=5):
    # lowercasing before encoding
    query_embedded = # Your Code Here ********
    cosine_scores = # Your Code Here ********

    # Get top k cosine score and index its
    results = []
    for idx in cosine_scores.argsort()[-top_d:][::-1]:
        doc_score = {
            'id': idx,
            'cosine_score':cosine_scores[idx]
        }
        results.append(doc_score)
    return results

question = vi_data_df.iloc[0]['question']
results = tfidf_search(question, tfidf_vectorizer, top_d=5)
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:

```
def corr_search(question, tfidf_vectorizer, top_d=5):
    # lowercasing before encoding
    query_embedded = # Your Code Here ******
    corr_scores = # Your Code Here ******
    corr_scores = corr_scores[0][1:]
    # Get top k correlation score and index its
    results = []
    for idx in corr_scores.argsort()[-top_d:][::-1]:
        doc = {
            'id': idx,
            'corr_score':corr_scores[idx]
        }
        results.append(doc)
    return results

question = vi_data_df.iloc[0]['question']
results = corr_search(question, tfidf_vectorizer, top_d=5)
results[1]['corr_score']
```

a) 0.20

b) 0.21

c) 0.22

d) 0.23