A. Jashwanth

231FA04902

Machine Learning

# Pre-requasite Assignment-1

## Section A: Python Programming

1. Write a python function to compute the mean and standard deviation of a List of numbers.

```
import math
def compute_mean_std(numbers):
    if not numbers:
        return None, None
mean = sum(numbers)/len(numbers)
variance = sum((x-mean)**2 for x in numbers)/len(numbers)
std_dev = math.sqrt(variance)
return mean, std_dev.
input_str = input("Enter numbers separated by spaces")
numbers = list(map(float, input_str.strip().split()))
mean, std = compute_mean_std(numbers)
print("Mean", mean)
print("Standard Deviation", std)
```

2. What is the difference between a list, a tuple, and a dictionary in python? Give an example of each

| Feature | List | Tuple | Dictionary |
|---|---|---|---|
| Syntax | [] | () | {} |
| Ordered | ✓ | ✓ | ✓ |
| Mutable | ✓ | X | ✓ |
| Duplicates | ✓ | ✓ | x (keys must be unique) |

Example of list

Fruits = ["apple", "banana", "cherry"]

Tuple

coordinates = (10.5, 20.3)

Dictionary

student = {"name": "Alice", "age": 21, "grade": "A"}

3. Implement a simple linear regression using numpy without using scikit-learn.

```
import numpy as np
import matplotlib.pyplot as plt.
x = np.array ([1,2,3,4,5])
y = np.array([2,4,5,4,5])
n = len(x)
m = (n * np.sum(x*y) - np.sum(x) * np.sum(y)) /
        (n*np.sum(x**2) - (np.sum(x))**2)
c = (np.sum(y) - m * np.sum (x))/n
y_pred = m*x+c
print (f" slope (m): {m}")
print (f" Intercept (c): {c}")
plt.scatter( x,y, color= "blue", label = "original Data")
plt.plot ( x,y_pred, color="red", label="fitted line")
plt.xlabel ("x")
plt.ylabel ("y")
plt.title (" simple linear Regression")
plt.lergend ()
plt.show ()
```

Scanned with OKEN Scanner

(4) Explain how to handle missing data in a Pandas DataFrame. Provide an example.

Handling missing data in a pandas. DataFrame is a common data preprocessing task. Missing values are typically represented as.

Example

```
import pandas as pd
import numpy as np
data = { 'Name': ['Alice', 'Bob', 'charlie', 'David'],
         'Age': [25, np.nan, 30, 22],
         'city': ['New York', 'Los Angeles', np.nan,
                  'chicago']
       }
df = pd.DataFrame(data)
print("original Data frame")
print(df)
print("\n Missing value locations")
print(df.isnull())
df-dropped = df.dropna()
print("\n After dropping rows with missing data")
print(df-dropped)
df-filled = df.fillna({
        'Age': df['Age'].mean(),
        'city': 'unknown'
  })
print("\n After filling missing values")
print(df-filled)
```

and stastics.

Orgiral

| | Name | Age | City |
|---|---|---|---|
| 0 | Alice | 25.0 | newyork |
| 1 | Bob | NaN | los Angeles |
| 2 | charlie | 30.0 | NAN |
| 3 | David | 22.0 | Chicago |

After Drop

| | NAME | Age | City |
|---|---|---|---|
| 0 | Alice | 25.0 | New york |
| 3 | David | 22.0 | Chicago |

(5) write a python script to load a csv file and normalize its numeric columns.

```
import pandas as pd
from sklearn.preprocessing import MinMaxscaler
file_path = 'your_file.csv'
 df = pd.read_csv(file_path)
 print("original Data:")
 print(df.head())
numeric_cols = df.select_dtypes(include=['number']).
                                                  columns
 scaler = MinMaxScaler()
df[numeric_cols] = scaler.fit_transform(df[numeric_
                                                 cols])
  print("\n Normalized Data:")
  print(df.head())
  df.to_csv('normalized_output.csv', index=false)
```

A: ... null hypothesis")



Section B: probability and stastics.

1. Define the following with examples: conditional probability, Baye's theorem.

Conditional probability: probability is the probability of an event A, given that another event B has already occured

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{where } P(B) > 0$$

Ex:

A = card is a king

B = card is a face card

$P(A \cap B) = P(\text{king}) = \frac{4}{52}$

$P(B) = \frac{12}{52}$

$P(\text{king} \mid \text{Face card}) = \frac{P(A \cap B)}{P(B)} = \frac{4/52}{12/52} = \frac{1}{3}$

Bayes theorem: Bayes theorem gives the probability of an event based on prior knowledge of related events

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Ex: 1% of people have a rare disease, $P(D) = 0.01$

- A tests detects the disease with 99% accuracy $P(\text{positive}|D) = 0.99$

5% false $P(\text{positive} \mid \text{No Disease}) = 0.05$.

$$P(D \mid \text{positive}) = \frac{P(\text{positive} \mid D) \cdot P(D)}{P(\text{positive})}$$

$P(\text{positive}) = P(\text{positive} \mid D) \cdot P(D) + P(\text{positive} \mid \sim D) \cdot P(\sim D)$

$= 0.0594$

$\boxed{P(D \mid \text{positive}) = 0.1667}$

2. Given two datasets of exam scores, How would you test if their means are significantly different?

1. Hypothesis

Null hypothesis $(H_0)$: $\mu_1 = \mu_2$

Alternative hypothesis $(H_1)$: $\mu_1 \neq \mu_2$.

The two datasets are independent

The data in each group is approximately normally distributed

variances are equal.

```
import numpy as np.
from scipy. stats import ttest_ind
group1 = [85, 90, 88, 92, 87]
group2 = [78, 75, 80, 74, 79]
print(f"T- statistic: {t_stat}")
print(f"p value: {p_value}")
```

```
if p_value < 0.05:
    print("Reject the null hypothesis")
else:
    print("Fail to reject the null hypothesis")
```

P-value < 0.05 → significant diff b/w means

P-value ≥ 0.05 → No significant difference

3. Explain bias and variance with help of probability distribution.

Bias → Error due to simplifying assumptions in the model. It causes the model to miss relevant relationships.

Variance → Error due to model sensitivity to training data fluctuations. It causes the model to learn noise.

High Bias

models are too simple

predictions from multiple datasets are clustered together but far from the true function.

Distribution : predictions are narrowly centred, but not near correct answer.

High variance

model is too complex

predictions vary widely depending on the data set

Distribution predictions are spread out, some near the true value, some far programming

the mean

4. You roll two dice. what is the probability that the sum is 7? what is the probability that both numbers are even?

Total possible outcomes: $6 \times 6 = 36$.

1. probability that the sum is 7

(1,6) (2,5) (3,4), (4,3), (5,2), (6,1) = 6 outcomes

probability: $\frac{6}{36} = \frac{1}{6}$

2. probability that both numbers are even = 9.

probability = $\frac{9}{36} = \frac{1}{4}$.

5. Describe the central limit Theorem and its importance in Machine learning.

If you take many random sampels from any population, and calculate the mean of each sample:

The distribution of those sample means will look like a Normal curve. as the no. of samples increase even if the original data is not normal.

Example:

Imagine collecting the average test scores from many small classrooms.

Importance

1. Makes predictions easier
2. confidence intervals
3. model evaluation
4. Sampling stablity