

Bytes of Intelligence: Set - 05

Department of AI Engineering

Semester Examination

Course Title: Deep Learning Techniques in Python for Vision and Language

Course Code: DL-101

Duration: 3 Hours

Total Marks: 100

Instructions:

- Attempt all questions.
 - Read each question carefully before answering.
 - All code should be properly commented and indented.
 - Use appropriate data structures and algorithms where necessary.
 - Marks are indicated next to each question.
 - Assume any necessary imports (e.g., `import numpy as np`, `import pandas as pd`).
 - Write your code in Python programming language.
-

Question 1: Python Fundamentals and Control Structures (15 Marks)

(a) Basic Input/Output and Data Types (5 Marks)

Write a Python program that:

- Prompts the user to input a sentence.
- Counts the number of words, characters (excluding spaces), and vowels in the sentence.
- Prints the results in a formatted manner.

Example:

```
Enter a sentence: Data Science is awesome!  
Number of words: 4
```

Number of characters (excluding spaces): 19
Number of vowels: 9

(b) Functions and Error Handling (10 Marks)

Define a function `factorial(n)` that:

- Calculates the factorial of a positive integer `n`.
- Raises a `ValueError` if `n` is negative or not an integer.
- Includes a docstring explaining the function's purpose.

Demonstrate the use of this function with at least two examples, one of which should trigger the `ValueError`.

Question 2: Data Structures in Python (20 Marks)

(a) List Manipulation and Comprehensions (10 Marks)

Given a list of integers:

```
numbers = [3, 7, 2, 9, 12, 15, 8, 6]
```

Perform the following tasks:

- Create a new list containing the squares of all even numbers from the original list using a list comprehension. (5 Marks)
 - Sort the original list in descending order without using the built-in `sort()` method. (5 Marks)
-

(b) Dictionary Operations (10 Marks)

You have a dictionary representing employees and their departments:

```
employees = {  
    "E001": "HR",  
    "E002": "IT",  
    "E003": "Finance",  
    "E004": "IT",  
    "E005": "HR",  
    "E006": "Marketing",  
    "E007": "Finance",  
}
```

```
"E008": "IT"  
}
```

Perform the following tasks:

- Create a new dictionary that maps each department to a list of employee IDs in that department. (6 Marks)
 - Find the department with the highest number of employees. (4 Marks)
-

Question 3: NumPy Array Operations (15 Marks)

(a) Array Creation and Statistical Analysis (10 Marks)

- Create a NumPy array `data` of 100 random integers between 50 and 150. (2 Marks)
 - Reshape the array into a `(20, 5)` matrix. (2 Marks)
 - Calculate the following statistics for the entire dataset:
 - Mean
 - Median
 - Standard Deviation
 - Variance (4 Marks)
 - Find the maximum value in each column and store the results in a one-dimensional array. (2 Marks)
-

(b) Logical Indexing and Filtering (5 Marks)

Using the `data` array from part (a):

- Create a boolean mask to filter out all values greater than 100.
 - Use the mask to create a new array containing only the values less than or equal to 100.
 - Calculate the percentage of values that are less than or equal to 100.
-

Question 4: Data Analysis with Pandas (25 Marks)

You have a CSV file `customer_data.csv` containing the following columns: `'CustomerID'`, `'Name'`, `'Age'`, `'Gender'`, `'AnnualIncome'`, `'SpendingScore'`.

(a) Data Loading and Exploration (10 Marks)

- Load the dataset into a Pandas DataFrame. (2 Marks)
- Display the basic statistics of numerical columns using `.describe()`. (2 Marks)

- Check for missing values in the dataset and handle them appropriately (e.g., fill with mean or median, or drop rows). Explain your choice. (4 Marks)
 - Convert the 'Gender' column into numerical values (e.g., 0 for 'Male', 1 for 'Female'). (2 Marks)
-

(b) Data Visualization and Interpretation (10 Marks)

- Create a scatter plot using Matplotlib or Seaborn to visualize the relationship between 'AnnualIncome' and 'SpendingScore'. Include appropriate labels and title. (5 Marks)
 - Use Seaborn to create a histogram of the 'Age' distribution, separated by gender. Include a legend, labels, and title. (5 Marks)
-

(c) Data Segmentation (5 Marks)

- Segment the customers into three age groups: 'Youth' (18-30), 'Adult' (31-50), 'Senior' (51 and above).
 - Add a new column 'AgeGroup' to the DataFrame with the corresponding age group labels.
-

Question 5: Data Visualization with Matplotlib and Seaborn (15 Marks)

(a) Cluster Analysis Visualization (7 Marks)

Using the `customer_data.csv` DataFrame:

- Perform K-Means clustering on 'AnnualIncome' and 'SpendingScore' to segment customers into 5 clusters. (You can use `from sklearn.cluster import KMeans`.)
 - Visualize the clusters using a scatter plot with different colors for each cluster.
 - Include appropriate labels, title, and legend.
-

(b) Heatmap and Correlation Analysis (8 Marks)

- Calculate the correlation matrix for the numerical features in the DataFrame.
 - Use Seaborn to create a heatmap of the correlation matrix.
 - Interpret the heatmap and discuss any strong positive or negative correlations.
-

Question 6: Hands-On Project (15 Marks)

You are provided with a dataset `energy_consumption.csv` containing hourly energy consumption data with the following columns: 'Timestamp', 'EnergyConsumption', 'Temperature', 'Humidity'.

Tasks:

- Load the dataset into a Pandas DataFrame and parse the `'Timestamp'` column as datetime. (3 Marks)
 - Set the `'Timestamp'` column as the index and resample the data to calculate daily total energy consumption. (4 Marks)
 - Plot the daily energy consumption over time using Matplotlib. Include labels, title, and gridlines. (4 Marks)
 - Perform a rolling average with a window of 7 days on the daily energy consumption and plot it on the same graph. (4 Marks)
-

Question 7: Capstone Coding Challenge (Bonus Question - Optional) (10 Marks)

Time Series Decomposition and Forecasting

Write a Python script that:

- Loads the `energy_consumption.csv` dataset and sets the `'Timestamp'` column as the index. (1 Mark)
- Decomposes the time series into trend, seasonal, and residual components using `statsmodels`. (3 Marks)
- Plots each component separately. (3 Marks)
- Uses the decomposed components to forecast energy consumption for the next 7 days. (3 Marks)

Note: You may assume the necessary libraries are installed (e.g., `from statsmodels.tsa.seasonal import seasonal_decompose`).