

Course Code: CS481	Course Name: Data Science
Instructor Name: Dr Muhammad Atif Tahir	
Student Roll No:	Section No:

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **6 questions and 4 pages**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- All relevant formulas are provided in Appendix.
- In each question, Show all steps clearly.

Time: 180 minutes.

Total Marks: 50 points

Question 1: Briefly answer the following questions. Each questions should be answered in maximum 20 words including articles. Otherwise, answer will not be checked. [10 Points]

- What is data science?
- As a programmer, list two possible tools that can be used for ETL process
- Why it is important to scale attributes in kNN classifier?
- What are the two main advantages of Decision Tree based Classifiers?
- What is Hyperplane?
- Briefly explain the difference between Homogenous and Heterogeneous Classifiers.
- Why the base classifiers should be unstable for bagging?
- List two solutions to Initial Centroids Problem in kmeans?
- Briefly discuss any two issues to consider during data integration.
- Can you think of a real world application in which stop words would be useful for text classification?

Question 2: Table 1 shows an example of a Distance Matrix. Draw Dendrogram using Hierarchical Clustering Process using [8 Points]

- single link clustering [4 Points]
- complete link clustering [4 Points]

	a	b	c	d	e	f
a	0	15	6	8	1	9
b	15	0	4	9	8	5
c	6	4	0	14	7	5
d	8	9	14	0	7	3
e	1	8	7	7	0	16
f	9	5	5	3	16	0

Table1: Example of a Distance Matrix.

Question 3 (show all formulas used clearly)**[7 Points]**

- (a) Suppose the fraction of undergraduate students who smoke is 15% and the fraction of graduate students who smoke is 23%. If one-fifth of the college students are graduate students and the rest are undergraduates, what is the probability that a student who smokes is a graduate student? (Hint: Use Bayes Classifier Theorem) [2 Points]
- (b) Given the information in part (a), is a randomly chosen college student more likely to be a graduate or undergraduate student? [1 Point]
- (c) Repeat part (b) assuming that the student is a smoker [1 Point]
- (d) Suppose 30% of the graduate students live in a dorm but only 10% of the undergraduate students live in a dorm. If a student smokes and lives in the dorm, is he or she more likely to be a graduate or undergraduate student? You can assume independence between students who live in a dorm and those who smoke [3 Points]

Question 4: Consider a Database D (Table 2) consists of 8 transactions**[5 Points]**

- (a) Find frequent Itemsets using Apriori Algorithm. Suppose min. support count = 3 [3 Points]
- (b) Generate Association Rules from Largest Frequent Itemsets. Suppose min. confidence = 100% [2 Points]

TID	Items
1	I_1, I_3
2	I_1, I_4, I_5
3	I_1, I_4
4	I_2, I_3, I_4, I_8
5	I_2, I_4, I_7
6	I_4, I_5, I_6, I_7, I_8
7	I_4, I_6, I_7, I_8
8	I_4, I_7, I_8

Table2: Database D for Transactions.

Question 5:**[10 Points]**

- (a) **[4 Points]** Consider a document containing 100 words wherein the word student appears 10 times and word university appears 50 times. Let's also assume that there 10,000 documents and the word student appears in one thousands of these while word university appears in 100 of these. Calculate TFIDF weight for student and university
- (b) **[4 Points]** Explain the difference between Feature Selection and Feature Extraction. List any 3 feature selection methods and 3 feature extraction methods. You do not need to explain these methods.
- (c) **[2 Points]** What is the best topic that you enjoy most in Data Science course and why?

Question 6:**[10 Points]**

(a) [4 Points] Write down in the cells below seaborn and matplotlib functions according to their correct definitions: **matplotlib.axes.Axes.hexbin**, **seaborn swarmplot**, **seaborn regplot**, **seaborn jointplot**, **matplotlib.pyplot.xlim**, **seaborn heatmap**, **matplotlib.axes.Axes.hist2d**, **seaborn stripplot**, **seaborn violin**

For example; seaborn tsplot	Plot one or more timeseries with flexible representation of uncertainty
	Draw a scatterplot where one variable is categorical
	Make a hexagonal binning plot
	Draw a combination of boxplot and kernel density estimate.
	Plot rectangular data as a color-encoded matrix
	Draw a plot of two variables with bivariate and univariate graphs
	Plot data and a linear regression model fit
	Make a 2D histogram plot
	Draw a categorical scatterplot with non-overlapping points
	Set the x-axis range

(b) [3 Points] In order to visualize two-dimensional arrays of data, it is necessary to understand how to generate and manipulate 2-D arrays. Many Matplotlib plots support arrays as input and in particular, they support NumPy arrays. The purpose of meshgrid is to create a rectangular grid out of an array of x values and an array of y values. The simplest way to generate a meshgrid is as follows:

```
import numpy as np
Y,X = np.meshgrid(range(10), range(20))
```

This will create two arrays with a shape of (20,10), which corresponds to 20 rows along the Y-axis and 10 columns along the X-axis. Based on the above information and following instructions, write down a program script.py to do the following

- import the numpy and matplotlib.pyplot modules using the respective aliases np and plt
- Generate two one-dimensional arrays u and v using np.linspace(). The array u should contain 30 values uniformly spaced between -2 and +2. The array v should contain 10 values uniformly spaced between -1 and +1.
- Construct two two-dimensional arrays X and Y from u and v using np.meshgrid().
- Compute Z, $Z = \sin(3 \cdot \sqrt{X^2 + Y^2})$
- After the array Z is computed using X and Y, visualize the array Z using plt.pcolor() and plt.show() functions

(c) [3 Points] Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. One of the simplest things one can do using seaborn is to fit and visualize a simple linear regression between two variables using seaborn lmplo. One difference between seaborn and regular matplotlib plotting is that you can pass pandas DataFrames directly to the plot and refer to each column by name. For example, if you were to plot the column 'price' vs the column 'area' from a DataFrame df, you could call lmplo(x='area', y='price', data=df) from seaborn.

Residuals on the other hand visualize how far datapoints diverge from the regression line. Seaborn residplot function will regress y on x. Below is the syntax of that function

seaborn.residplot(x, y, data=None, lowess=False, x_partial=None, y_partial=None, order=1, robust=False, dropna=True, label=None, color=None, scatter_kws=None, line_kws=None, ax=None)

Based on the above information and following instructions, write down the program regression.py; that will do the following

- Import matplotlib.pyplot and seaborn using the standard names plt and sns respectively.
- Plot a linear regression between the 'weight' column (on the x-axis) and the 'hp' column (on the y-axis) from the DataFrame auto.
- Generate a green residual plot of the regression between 'hp' (on the x-axis) and 'mpg' (on the y-axis). Ignore observations with missing data

Appendix: TDIDF (Help Notes Only)

A more complex way of calculating the weights is called TFIDF, which stands for Term Frequency Inverse Document Frequency. This combines term frequency with a measure of the rarity of a term in the complete set of documents. It has been reported as leading to improved performance over the other methods.

The TFIDF value of a weight X_{ij} is calculated as the product of two values, which correspond to the term frequency and the inverse document frequency, respectively.

The first value is simply the frequency of the jth term, i.e. t_j , in document i . Using this value tends to make terms that are frequent in the given (single) document more important than others.

We measure the value of inverse document frequency by $\log_2(n/n_j)$ where n_j is the number of documents containing term t_j and n is the total number of documents. Using this value tends to make terms that are rare across the collection of documents more important than others.

Appendix (Formulas)

Bayes Classifier	$P(C A) = \frac{P(A C)P(C)}{P(A)}$ <p style="text-align: center;">Or</p> $P(C A) = \frac{P(A C)P(C)}{P(A C)P(C) + P(A \sim C)P(\sim C)}$ <p>where \sim = NOT</p>
Conditional Probability	$P(C A) = \frac{P(A, C)}{P(A)}$ $P(A C) = \frac{P(A, C)}{P(C)}$

BEST OF LUCK!