National University of Computer & Emerging Sciences, Karachi

Spring 2016/17 CS-Department

Final - Solution

18th May 2017, 9:00 am – 12:00 am

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]

a) What is data science?
Ans: Data science is an interdisciplinary field about processes and systems to extract knowledge or insights from data in various forms. (Wikipedia)
Data Science closes the circle from collecting real-world data, to processing and analyzing it, to influence the real world again

b) As a programmer, list two possible tools that can be used for ETL process
Ans: wget, curl, Beautiful Soup, lxml, or any other possibility

c) Why it is important to scale attributes in kNN classifier?
Ans: Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes

d) What are the two main advantages of Decision Tree based Classifiers?
Ans: Inexpensive to construct
Extremely fast at classifying unknown records
Easy to interpret for small-sized trees
Accuracy is comparable to other classification techniques for many simple data sets

e) What is Hyperplane?
An hyperplane is a generalization of a plane, in one dimension, an hyperplane is called a point
in two dimensions, it is a line, in three dimensions, it is a plane, in more dimensions you can call it an hyperplane

f) Briefly explain the difference between Homogenous and Heterogeneous Classifiers.
Ans: Homogenous Classifiers (Same Classifiers but different training data) e.g. Bagging, Boosting etc
Heterogeneous Classifiers (Different Classifiers but same training data) e.g. Majority Voting, Mean etc)

g) Why the base classifiers should be unstable for bagging?
Ans: due to high variance

h) List two solutions to Initial Centroids Problem in kmeans?

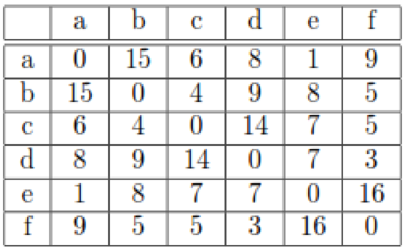
Multiple runs – Helps, but probability is not on your side, Sample and use hierarchical clustering to determine initial centroids, Select more than k initial centroids and then select among these initial centroids – Select most widely separated

i) Briefly discuss any two issues to consider during data integration.
Answer: Data integration involves combining data from multiple sources into a coherent data store. Issues that must be considered during such integration include: • Schema integration: The metadata from the different data sources must be integrated in order to match up equivalent real-world entities. This is referred to as the entity identification problem. • Handling redundant data: Derived attributes may be redundant, and inconsistent attribute naming may also lead to redundancies in the resulting data set. Duplications at the tuple level may occur and thus need to be detected and resolved. • Detection and resolution of data value conflicts: Differences in representation, scaling, or encoding may cause the same real-world entity attribute values to differ in the data sources being integrated.

j) Can you think of a real world application in which stop words would be useful for text classification?
One example: Forensic or artistic situation.

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

(a) single link clustering [4 Points]
(b) 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.

Solution: (Single Link)

	ae	b	c	d	f
ae	0	8	6	7	9
b	8	0	4	9	5
c	6	4	0	14	5
d	7	9	14	0	3
f	9	5	5	3	0

	ae	b	c	df
ae	0	8	6	7
b	8	0	4	5
c	6	4	0	5
df	7	5	5	0

	ae	bc	df
ae	0	6	7
bc	6	0	5
df	7	5	0

	ae	bcdf
ae	0	6
bcdf	6	0

Solution: (Complete Link)

	ae	b	c	d	f
ae	0	15	7	8	16
b	15	0	4	9	5
c	7	4	0	14	5
d	8	9	14	0	3
f	16	5	5	3	0

	ae	b	c	df
ae	0	15	7	16
b	15	0	4	9
c	7	4	0	14
df	16	9	14	0

	ae	bc	df
ae	0	15	16
bc	15	0	14
df	16	14	0

	ae	bcdf
ae	0	14
bcdf	14	0

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]

(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?

Answer:

Given $P(S|UG) = 0.15$, $P(S|G) = 0.23$, $P(G) = 0.2$, $P(UG) = 0.8$. We want to compute $P(G|S)$.

According to Bayesian Theorem,

$$P(G|S) = \frac{0.23 \times 0.2}{0.15 \times 0.8 + 0.23 \times 0.2} = 0.277. \tag{5.1}$$

(b) Given the information in part (a), is a randomly chosen college student more likely to be a graduate or undergraduate student?

Answer:

An undergraduate student, because $P(UG) > P(G)$.

(c) Repeat part (b) assuming that the student is a smoker.

Answer:

An undergraduate student because $P(UG|S) > P(G|S)$.

(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.

Answer:

First, we need to estimate all the probabilities.

$P(D|UG) = 0.1$, $P(D|G) = 0.3$.
$P(D) = P(UG).P(D|UG) + P(G).P(D|G) = 0.8*0.1 + 0.2*0.3 = 0.14$.
$P(S) = P(S|UG)P(UG) + P(S|G)P(G) = 0.15*0.8 + 0.23*0.2 = 0.166$.
$P(DS|G) = P(D|G) \times P(S|G) = 0.3 \times 0.23 = 0.069$ (using conditional independent assumption)
$P(DS|UG) = P(D|UG) \times P(S|UG) = 0.1 \times 0.15 = 0.015$.
We need to compute $P(G|DS)$ and $P(UG|DS)$.

$$P(G|DS) = \frac{0.069 \times 0.2}{P(DS)} = \frac{0.0138}{P(DS)}$$

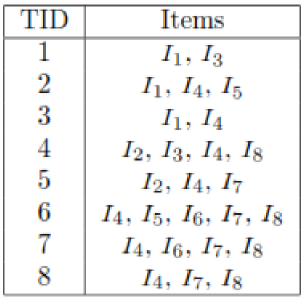
$$P(UG|DS) = \frac{0.015 \times 0.8}{P(DS)} = \frac{0.012}{P(DS)}$$

Since $P(G|DS) > P(UG|DS)$, he/she is more likely to be a graduate student.

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.

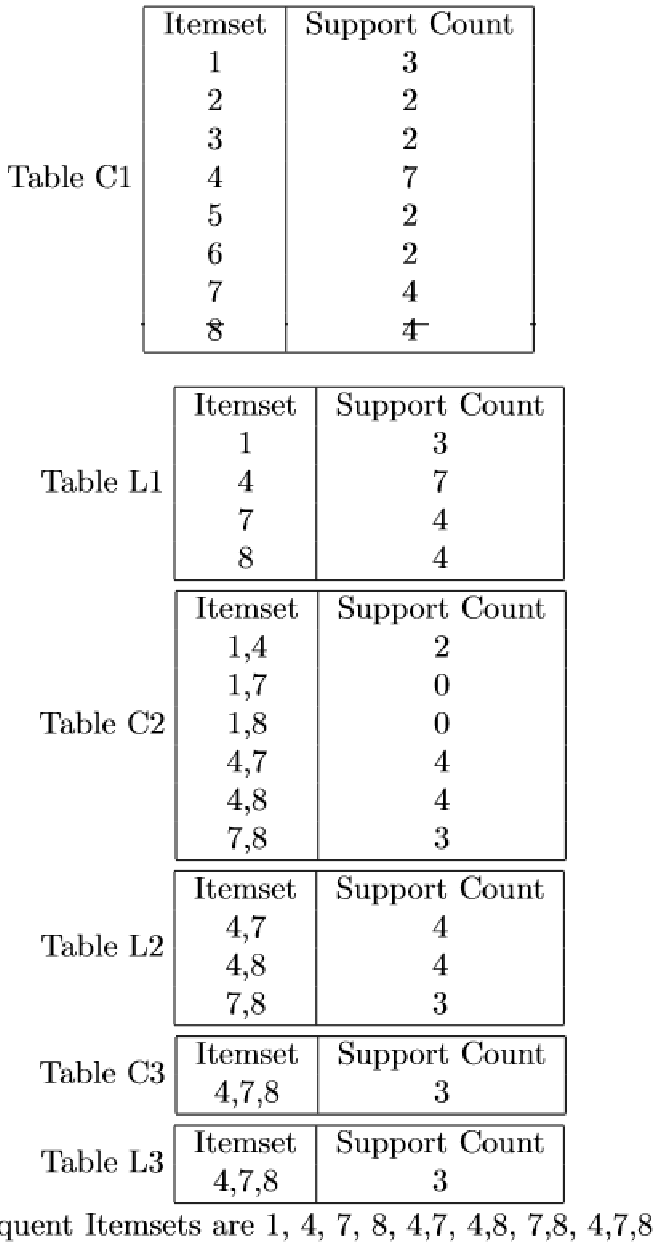


Table C1

Itemset	Support Count
1	3
2	2
3	2
4	7
5	2
6	2
7	4
~~8~~	~~4~~

Table L1

Itemset	Support Count
1	3
4	7
7	4
8	4

Table C2

Itemset	Support Count
1,4	2
1,7	0
1,8	0
4,7	4
4,8	4
7,8	3

Table L2

Itemset	Support Count
4,7	4
4,8	4
7,8	3

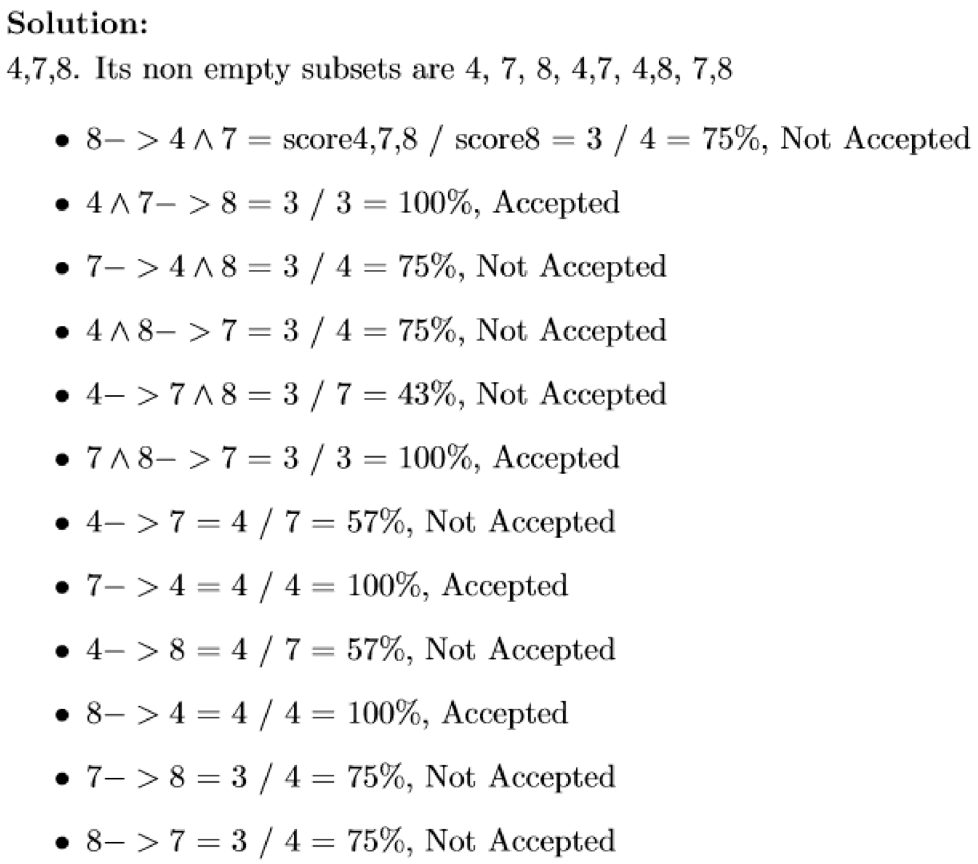
Table C3

Itemset	Support Count
4,7,8	3

Table L3

Itemset	Support Count
4,7,8	3

Frequent Itemsets are 1, 4, 7, 8, 4,7, 4,8, 7,8, 4,7,8



Solution:

4,7,8. Its non empty subsets are 4, 7, 8, 4,7, 4,8, 7,8

- $8- > 4 \wedge 7 =$ score4,7,8 / score8 $= 3 / 4 = 75\%$, Not Accepted
- $4 \wedge 7- > 8 = 3 / 3 = 100\%$, Accepted
- $7- > 4 \wedge 8 = 3 / 4 = 75\%$, Not Accepted
- $4 \wedge 8- > 7 = 3 / 4 = 75\%$, Not Accepted
- $4- > 7 \wedge 8 = 3 / 7 = 43\%$, Not Accepted
- $7 \wedge 8- > 7 = 3 / 3 = 100\%$, Accepted
- $4- > 7 = 4 / 7 = 57\%$, Not Accepted
- $7- > 4 = 4 / 4 = 100\%$, Accepted
- $4- > 8 = 4 / 7 = 57\%$, Not Accepted
- $8- > 4 = 4 / 4 = 100\%$, Accepted
- $7- > 8 = 3 / 4 = 75\%$, Not Accepted
- $8- > 7 = 3 / 4 = 75\%$, Not Accepted

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

Answer:

TFIDF (student) = tf * idf = 100/10 * log2(10000/1000) = 10 * log2(10) = 33.2
TFIDF (university) = tf * idf = 100/50 * log2(10000/100) = 2 * log2(100) = 13.2

(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.
Feature Selection to select the best subset of features while feature extraction to transform data into another domain.
FS (Brute Force, Sequential Forward Selection, Sequential Backward Selection)
Feature Extraction (PCA, LDA, MDS, ISoMAP)

(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 matlabplot functions according to their correct definitions: **matplotlib.axes.Axes.hexbin, seaborn swarmplot, seaborn regplot, seaborn jointplot, matlabplot.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
seaborn stripplot	Draw a scatterplot where one variable is categorical
matplotlib.axes.Axes.hexbin	Make a hexagonal binning plot
Seaborn violin	Draw a combination of boxplot and kernel density estimate.
seaborn heatmap	Plot rectangular data as a color-encoded matrix
seaborn jointplot	Draw a plot of two variables with bivariate and univariate graphs
seaborn regplot	Plot data and a linear regression model fit
matplotlib.axes.Axes.hist2d	Make a 2D histogram plot
Seaborn swarmplot	Draw a categorical scatterplot with non-overlapping points
matlabplot.pyplot.xlim	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 = np.sin(3*np.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

```
# Import numpy and matplotlib.pyplot
import numpy as np
import matplotlib.pyplot as plt

# Generate two 1-D arrays: u, v
u = np.linspace(-2,2,41)
v = np.linspace(-1,1,21)

# Generate 2-D arrays from u and v: X, Y
X,Y = np.meshgrid(u,v)

# Compute Z based on X and Y
Z = np.sin(3*np.sqrt(X**2 + Y**2))

# Display the resulting image with pcolor()
plt.pcolor(Z)
plt.show()
```

(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 lmplot**.** 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 lmplot(x='area', y='price', data=df) from seaborn.

Residuals on the other hand visualizie 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

```
# Import plotting modules

import matplotlib.pyplot as plt

import seaborn as sns
```

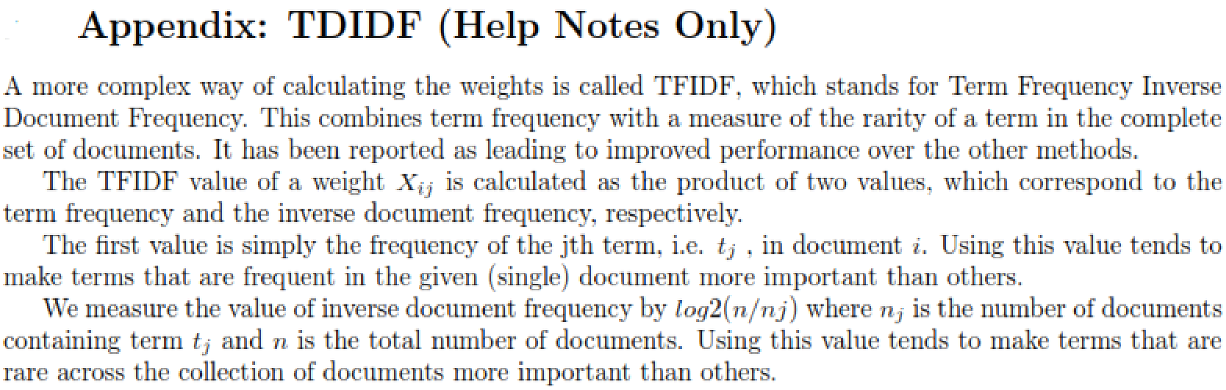
```
# Plot a linear regression between 'weight' and 'hp'

sns.lmplot(x = 'weight', y = 'hp', data=auto)

sns.resigplot(x=’hp’, y = ‘mpg’, dropna = True)

# Display the plot

plt.show()
```



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 $log2(n/nj)$ 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	$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$ **Or** $P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A \mid C)P(C) + P(A \mid \sim C)P(\sim C)}$ where ~ = NOT
Conditional Probability	$P(C \mid A) = \frac{P(A,C)}{P(A)}$ $P(A \mid C) = \frac{P(A,C)}{P(C)}$

BEST OF LUCK!