FAST- National University of Computer & Emerging Sciences, Karachi.
Department of Computer Science,
Final Examinations, Spring 2020
22nd June, 2020, 9:00 am – 12:30 pm

Course Code: CS 481	**Course Name:** Data Science
Instructor Names: Dr. Muhammad Nouman Durrani	
Student Roll No:	**Section No:**

Instructions:

- Read each question completely before answering it. There are 6 questions on 8 pages
- The Exam will start on: 9:00 am; and will End at: 12:30 pm, including the submission time
- You will attempt this paper offline, in your hand writing, however, for some questions, you may use your PC
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper
- All the answers must be solved according to the SEQUENCE given in the question paper.
- Show all steps clearly
- You may use cam-scanner, MS lens or any equivalent application to scan and convert your hand-written answer sheets + any screenshot in a single PDF file
- The paper should be submitted using our Google classroom Data Science Spring 2020 having Code: eka5v6p. For this purpose, you are given 30 minutes, already mentioned in the above instructions. Additionally, after submitting over there, you should also submit it over the slate (if possible), otherwise email exactly the same copy to your instructor.
- Put your **signature on every page** along with **YOUR Roll No/ID, and page number**.
- Please fill the below table with your details. A sample value for a student having Roll number: K162345 and Name: Muhamamd Nouman Durrani is provided.

Description	Sample Value	Value for you
Your Full Name	YFN = Muhammad Nouman Durrani	
Your First Name	FiN = Muhammad	
Your Last Name	LN = Durrani	
Number of words/parts in your full name	NP = 3	
The last 4 digits of your Roll Number	RollNo: 2345 a[0] = 2, a[1] = 3 a[2] = 4, a[3] = 5	
RollNo/2.3 means your Roll No divided by 2.3, i.e., 2345/2.3=1019.56 RollNo/NP means your Roll No divided by 3, i.e., 2345/3=781.66 88 x (a[0] / 9) means 88 multiply by (2/9) i.e., 88 x 2/9 = 19.55		

Time Allowed: 180 minutes **Maximum Points**: 96 Points

==========

Question 1: **Classification and Cross Validation** **[70-85 Minutes]** **[30 Points]**

(i) Briefly answer the following short questions: [1 x 8 = 8 Points]

a) Why the training error of 1-NN classifier is 0.

b) Why variance is substantially more harmful to the test data than the training error?

c) The bootstrap method is a resampling technique that is used to estimate statistics on a population by sampling a dataset with replacement. In our class, we discussed that cross validation can be used to select the number of iterations in boosting; this procedure may help reduce overfitting. How?

d) Can we ensemble multiple models of same ML algorithm?

e) Why do we want to use "weak" learners when boosting?

f) What makes Adaboost adoptable?

g) Why the depth of a learned decision tree can't be larger than the number of training examples used to create the tree?

h) Briefly discuss, what happen if we do not use any activation function(s) in a neural network?

(ii) Use 2-Fold CV using a kNN classifier to find the *accuracy*, *precision*, *recall* and *F-measure* of the following data Points. Use k=1 and Euclidian distance function: $d(x, y) = (\sum_{i=1}^{n} (x_i - y_i)^2)^{1/2}$. [5 Points]

Attribute 1	Attribute 2	Attribute 3	Label
1	2	a[3]	A
2	1	5	A
a[0]	0	1	B
1	2	4	B
1	a[1]	3	A
4	3	5	B
a[2]	2	3	A
3	2	3	A

(iii) The following questions are related to k-means clustering: [1 + 2 +1+ 1 = 5 Points]

a) Can k-means clustering ever give results which contain less or more than k clusters?

b) Use k-means clustering to divide the following data into two clusters:

(a[2], 2), (3, 2), (1.5, a[3]), (1, 1)

Assume (1,1) and (3,2) are the initial centers of the two clusters.

c) The sum of squares is the sum of the square of variation, where variation is defined as the spread between each individual value and the mean. Explain what is the sum-of-squares for k-means?

d) The following images show the results of clustering the same data with k-means, with k running from 2 to 6; also a plot of the sum-of-squares versus k. How many clusters would you guess this data has, and why? Does it matter whether the plot is an average over many runs of the algorithm?

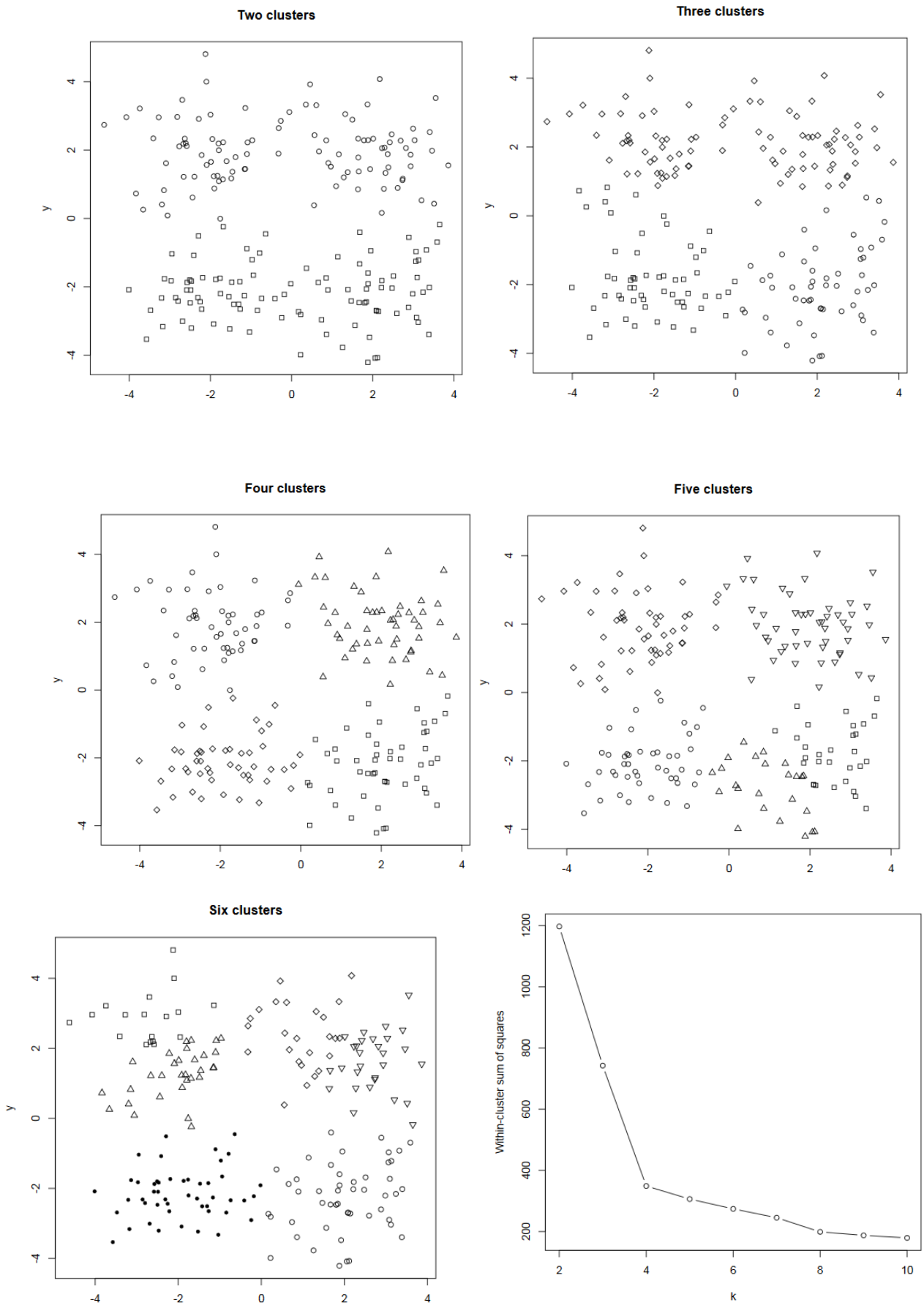


Figure 1: Clustering the same data with k-means and the cluster sum of squares

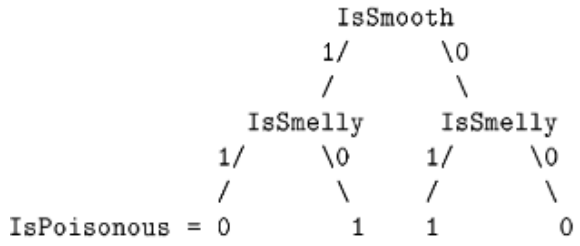
(iv) Suppose you are stranded on a Thar Cholistan deserted area. Mushrooms of various types grow widely all over the desert. Some of the mushrooms have been determined as poisonous and other are not. Your job is to identify the Mushroom type IsPoisonous True (1) or False(0) based on the following data. [2 + 2 + 2 = 6 Points]

Sample Type	IsSmooth	IsHeavy	IsSmelly	IsSpotted	IsPoisonous
A	0	0	0	0	0
B	0	0	0	1	0
C	1	1	1	0	0
D	1	1	0	0	1
E	0	0	1	1	1
F	1	0	0	1	1
G	1	0	0	0	1
H	0	1	1	0	1
I	1	0	1	0	?
J	0	1	1	0	?
K	1	1	1	1	?

a) In view of the above data given in the table, what is the entropy of IsPoisonous?

b) Which attribute you consider should be used as the root of a decision tree? What was the information gain of the attribute you chose as the root node? Hint: The root of a decision tree can be figure out by computing the information gain of all four attributes.

c) Suppose the following tree was built after some calculations. Classify Mushroom I, J and K using the below decision tree as poisonous or not poisonous.



(v) Implement the model shown in Figure 2. Expected output is shown below. [6 Points]

Size of Training Data: (398, 30)
Size of Testing Data: (171, 30)
Fold1: Accuracy using Naïve Bayes: 0.935
Fold2: Accuracy using Random Forest: 0.914
Fold3: Accuracy using KNN: 0.873

Consider the following Initial Coding:

```
# Load libraries
from sklearn.metrics import accuracy_score
from sklearn import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.datasets import load_breast_cancer

breast_data = load_breast_cancer()

#…… Complete the program as explained in block diagram
```

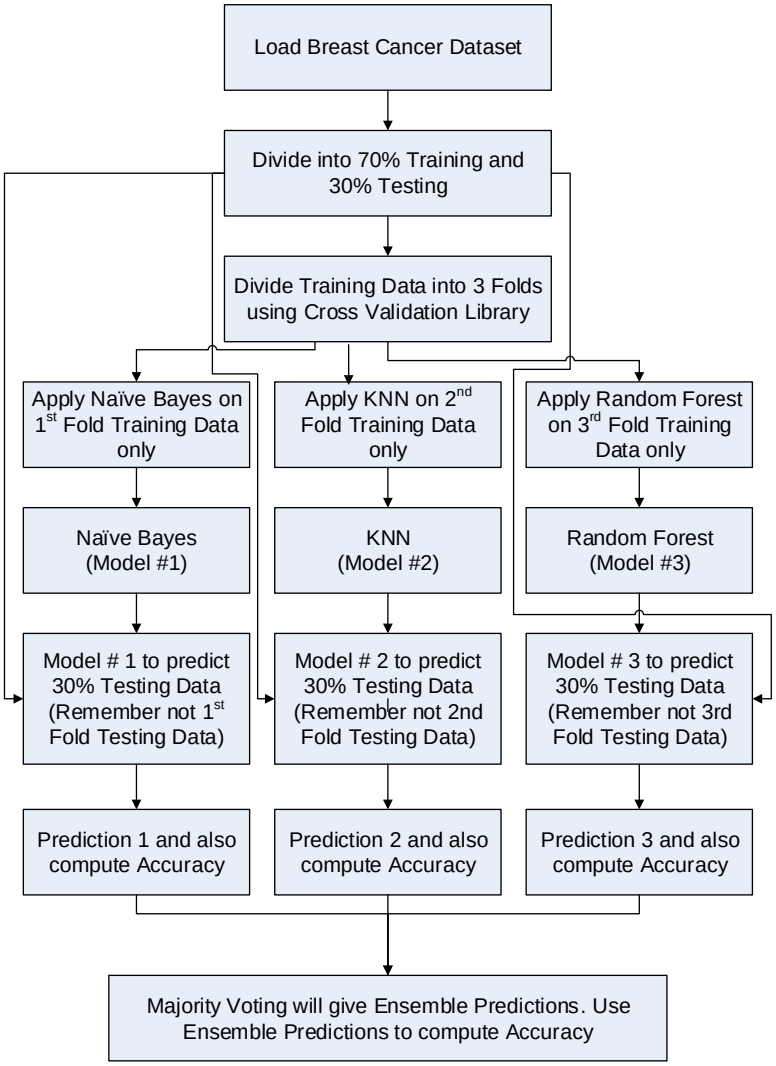


Figure 2: Model for Question 1 (v). Prediction 1, Prediction 2 and Prediction 3 are of type ndarray.

Question No 2: **Dimensionality Reduction** **[20-25 Minutes]** **[6 x 2 = 12 Points]**

Let A be an *m* x *n* matrix of data points:

$$A = \begin{pmatrix} a[0] & 4 & 11 & 10 \\ 4 & 5 & 10 & a[3] \\ 11 & a[1] & 25 & 24 \\ 10 & 11 & 24 & 25 \\ 11 & 10 & a[2] & 24 \\ 10 & 11 & 24 & 24 \end{pmatrix}$$

Also consider the following initial code:

```
from numpy import array
from numpy.linalg import eig
A = array([[a[1], 4, 11, 10], [4, 5, 10, a[3]], [11, a[1], 25, 24], [10, 11, 24, 25], [11, 10, a[2], 24],
           [10, 11, 24, 24]])
```

Perform the following operation using a Python code. You must attach a screenshot of your code here for your ease.

(i) If A is symmetric, then $A = A^T$. Calculate AA^T.

(ii) Calculate the eigenvalues λ_i for AA^T. What do the numerical values of the eigenvalues tell you about the data? You can answer the description part of this question as a comment in your Python notebook.

(iii) Select top 3 eigenvalues λ_i retaining maximum variance of the total sample variance.

(iv) Find the eigenvectors V_i of AA^T, using your eigenvalues λ_i from part (c).

(v) Now, construct a matrix *E*, the matrix of eigenvectors V_i for the matrix AA^T and use the concept of PCA to compute the resultant matrix *EA*.

(vi) Find the SVD for the matrix A using some of the above calculations with suitable assumptions, if any.

Question 3 **Regression and Correlation** **[25-30 Minutes]** **[14 Points]**

In the table below, the x_i column shows scores in the Aptitude Test. Similarly, the y_i column shows the Statistics course grades. The last two columns show deviations scores - the difference between the student's score. The average score on each test are also given.

Student	x_i	y_i	(x_i-x)	(y_i-y)
1	88 x (a[0] / 9)	85		
2	92 x (a[1] / NP)	95		
3	87	70 x (a[3] / NP)		
4	77 x (a[2] / 6)	65		
5	62	70 x (a[2] / NP)		
Sum				
Mean				

Required: [2 x 7 = 14 Points]

(i) The regression equation is a linear equation of the form: $\hat{y} = b_0 + b_1x$. Show the computation steps for the regression coefficient (b_1) and slope (b_0).

(ii) In the given data, independent variable is the student's score on the aptitude test. The dependent variable is the student's statistics grade. If a student made 76 on the aptitude test, the estimated statistics grade ($\hat{y}$) would be?

(iii) Compute the coefficient of determination R^2 (R^2 give some information about the goodness of fit of a model). What coefficient of determination R^2 indicates in this example?

(iv) Calculate the total sum of squares (SST) using sum of squares due to regression (SSR) and sum of squares due to error (SSE).

(v) Calculate a Pearson's correlation on the data *xi* and *yi* given in the above table.

(vi) How do we interpret a statistically significant Spearman correlation?

(vii) With suitable assumptions, calculate the Kendall's Tau coefficient for the above data?

Question 4: N**atural Language Processing** **[25-30 Minutes]** **[18 Points]**

We will grab some web page contents and save them in a data frame. Then, we will analyze the text to see what the page is about by performing the following NLP operations. You can write code for the below questions. Take screenshots and paste in your solution copy.

(i) Part-of-speech (POS) tagging is used to assign parts of speech to each word of a given text (such as nouns, verbs, pronouns, adverb, conjunction, adjectives, interjection) based on its definition and its context. Write code to assign parts of speech to the following text: [2 Points]

text = " YFN vote to choose a particular man or a group (party) to represent them in parliament"

(ii) Chunking means picking up individual pieces of information and grouping them into bigger pieces. In the context of NLP and text mining, what will be the output of the following code snippet: [2 Points]

```
sentence = "the little yellow dog barked at the cat in front of LN"
token = word_tokenize(sentence)
tags = nltk.pos_tag(token)
grammar = "NP: {<DT>?<JJ>*<NN>}"
cp = nltk.RegexpParser(grammar)
result = cp.parse(tags)
print result
result.draw()
```

(iii) Calculate and plot (line) the frequency distribution of Q 4(ii) tokens using Python NLTK and matplotlib. [1 Point]

(iv) Remove stop words from the above data. [1 Point]

(v) Differentiate the concept of stemming and lemmatizing and *apply* it on the extracted text. [2 Point]

(vi) Define a conditional frequency distribution over the Names Corpus that allows you to see which initial letters are more frequent for males versus females. [2 Points]

(vii) Write a program to find all words that occur at least NP times in the BrownCorpus. [2 Points]

(viii) Briefly explain how the Word2Vec approach converts words in Q 4(ii) into corresponding vectors? [2 Points]

(ix) Consider the following two Documents:
Document 1: The bus is driven on the motorway by YFN. Document 2: The truck is driven on the highway by FiN
Calculate the TF-IDF for the above two documents, which represent our corpus. [2 Points]

(x) Build your Word Frequencies model using TfidfVectorizer with suitable assumptions [2 Points]

Question 5: **Data Visualizations** **[20-25 Minutes]** **[14 Points]**

(i) This Problem is about calculating different aggregates using data visualization techniques we studied in the class. Consider the following financial data of K-Electric between June 13, 2020 to June 18, 2020. Also, consider financial data stored in **fdata.csv**, which is loaded into the Pandas DataFrame df as:

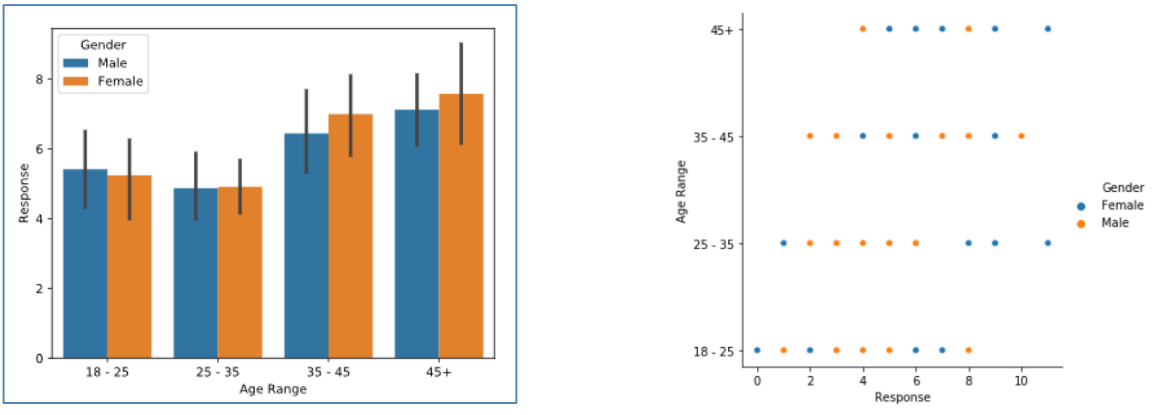
```
# For this problem you can use Jupitor notebook for coding

import matplotlib.pyplot as plt

import pandas as pd

df = pd.read_csv('fdata.csv')

============================================================

Sample Financial data (fdata.csv):
Date,Open,High,Low,Close
06-13-16,RollNo/NP,776.065002,769.5,772.559998
06-14-16,776.030029,778.710022,772.890015,776.429993
06-15-16,779.309998,782.070007,RollNo/2.3,776.469971
06-16-16,779,780.47998,775.539978,776.859985
06-17-16,779.659973,779.659973,770.75,RollNo/NP.080017

============================================================
```

a) Write a Python code to draw a line charts considering the above data. [Python code's screenshot required here] [1 Point]

b) Suppose we are now working on another dataset (already uploaded with this paper, please go through the dataset, a column named response has "NP" that must be first adjusted accordingly). We found that the survey *response* data is categorical; and we might want to count how many times each category appears. Plot the number of times a particular value appears in the Response data shown in Figure 3 (a). [Python code's screenshot required here] [1 Point]

c) Suppose our data has many outliers, in that situation, we might also want to plot the median. Plot the median of the Response column represented by black vertical lines in Figure 3 (a) [Python code's screenshot required here] [1 Point]

d) Also draw the scatter plot of the results.csv dataset as shown in Figure 3(b).[Python code's screenshot required here] [1 Point]



(a) *Figure 3: Aggregate Plots* (b)

(ii) In this Problem, we'll work on the **Error Bars** to display error visually in a bar chart. [Python code's screenshot required here] For someone who is learning about the different drink types at Macdonald, a bar chart of milk amounts in each drink may be useful. We have provided the ounces_of_milk list, which contains the amount of milk in each 14oz drink in the drinks list. According to different barista styles and measurement errors, there might be variation on how much milk actually goes into each drink. We've included a list error on each amount of milk.

```
drinks = ["latte", "americano", "espresso", "cappuccino","chai","mocha"]
ounces_of_milk = [a[1], 9, 10.75, a[3], a[2], 15]
error = [0.8, 0.75, 1.25, 1.0, 0.25, 1.7]
```

a) Plot this information in a line and as well as in a bar chart. [2 Point]
b) Display this error as error bars on the bar graph and add caps of size 5 to your error bars. [1 Point]
c) Set the axis to go from 'latte' to 'mocha' on the y-axis and 3 to 15 on the x-axis. Also add the title "Drinks to milk ratio", x-axis label "Milk amount in ounces", and y-axis label "Drinks". [1 Point]

(iii) In the previous problem, we saw bar plots to find out information about the mean, median, error bars etc. - but it doesn't give us a sense of how spread out the data is in each set. [Python code's screenshot required here]

To find out more about the distribution, we can use a KDE plot.

Another plot, called the *box plot* (also known as a box-and-whisker plot), not only tell us about how our dataset is distributed, like a KDE plot. But it shows us the range of our dataset, gives us an idea about where a significant portion of our data lies, and whether or not any outliers are present.

a) In this problem, you'll plot both KDE and box plots on the two datasets "flights", and "tips" (these two datasets can directly be loaded using the following code), to visualize the distribution of the datasets. [2 Points]

```
a = sns.load_dataset("tips")
f = sns.load_dataset("flights")
```

b) Color is basically the feature that approaches the human eyes beyond any other feature. Seaborn allows you to play with colors using various functions such as color_palette(), hls_palette(), husl_palette(). Draw a violinplot for the tips dataset while using the following color palette code for your ease. [2 Points]

```
qualitative_colors = sns.color_palette("Set3", NP)
sns.palplot(qualitative_colors)
```

c) Plot a graph for a variable "mynormaldistribution" whose values are generated by the normal() function using distplot. [2 Points]

Question 6 **Frequent Itemsets Mining:** **[5-10 Minutes]** **[2 + 4 + 2 = 8 Points]**

Here is a collection of ten baskets. Each contains three of the six items 1 through 6.

{1, 2, 3}, {2, 3, 4}, {3, 4, 5}, {4, 5, a[1]}, {a[2], 3, 5} {1, 3, 4}, {2, 4, 5}, {3, 5, 6} {a[0], 2, 4} {2, 3, a[3]}

(i) If the support threshold is 3, which items are frequent?
(ii) Find all the frequent itemsets using Apriori algorithm.
(iii) Obtain at least 2 significant decision rules.

BEST OF LUCK!

Submission: 1. Google classroom Data Science Spring 2020 having Code: eka5v6p

2. In the Slate Assignment Section or Through Email