[This sheet must be completed and attached to the first page of your homework]

ISE 291

Introduction to Data Science

Term 211

Homework #4

Student Name	ID#	Signature

Homework Guidelines

To receive full credit, you should make sure you adhere to the following guidelines. For any questions/comments contact your section instructor.

Homework Presentation & Submission:

- Every sub-problem (part) should be answered on a DIFFERENT CELL.
- EVERY CELL should have problem and part number clearly written in the first line.
- You should submit the solutions for the FIRST TWO problems only.
- All cells of your homework should be in CHRONOLOGICAL order. One cell per sub-problem.
- Submit entire HW as ONE single .ipynb document.
- ONE HW per group should be submitted.
- Your NAMEs, IDs, and the homework number should be clearly indicated in the FIRST CELL of the notebook.

Problem # A 50 marks

Consider data given in CSV file **HW4DataA** obtained from a public repository[1]. Consider the following data description:

Table 1: Data Description

Field	Description
Channel	The mechanism through which the goods were consumed. Contains two values: Hotels or Retail.
City_Town	The City/Town from which the data is collected.
Fresh	Annual spending (in SAR) on the fresh products.
Milk	Annual consumption (in liter) of milk products.
Grocery	Annual spending (in SAR) on the grocery products.
Frozen	Annual spending (in SAR) on the frozen products.
Detergents_Paper	Annual spending (in SAR) on the detergents and paper products.
Delicassen	Annual spending (in SAR) on the delicatessen products.

Do the following tasks using data given in **HW4DataA** and Table-1:

☞ *Note: Solve all the above questions using Python. Use* ***Pandas*** *&* ***Sklearn*** *library for all the above analysis.*

A-1: **Given Data.** Read and display the random 20 rows of data. The column with "Index" labe should be the index of the data. Identify the fields of the data. Count the number of rows and columns in the data. Count the number of non-null rows for each column.

A-2: **Type Consistency.** Identify the type for each field based on value. Also, identify the datatypes in Python. Report any inconsistency.

A-3: **Filter noise.** Any record whose channel value is neither "Retail" nor "Hotels" should be removed.

A-4: **Data Wrangling/Munging.** Columns "Fresh", "Milk", "Grocery", "Frozen", "Detergents_Paper" and "Delicassen" should be numeric. Resolve the inconsistency. The value in the "Milk" column is in liters, and should be converted to SAR. The conversion can be done by multiplying values in the "Milk" column by 9.8. Round the values in the milk column to the nearest integer.

A-5: **Handling NaN values.** All missing values in "Channel" field corresponds to "Hotels". Similarly, all the missing values in "City_Town" fields corresponds to "Khobar". The missing values in "Detergents_Paper" field should be replaced by mean, rounded to the nearest integer.

A-6: **Encoding.** Pick field "Channel", and relabel "Hotels" as 1, and "Retail" as 0.

A-7: **Feature Generation.** Create a new filed, called "Region". The values in region should be as follows:

- Region value for "Riyadh", "Qaseem" or "Hail" should be "Central".
- Region value for "Tabuk", "Makkah" or "Madinah" should be "Western".
- Region value for "Khobar", "Dammam" or "Dhahran" should be "Eastern".

A-8: **One-Hot-Encoding.** Do One-Hot-Encoding for "Region" column. Do not delete the original column.

A-9: **Standardization.** For all the following columns, do standard scalarizaiton, such that the mean value is 0, and the standard deviation is 1: "Fresh", "Milk", "Grocery", "Frozen", "Detergents_Paper" and "Delicassen"

A-10: **Clean & Prepared Data.** Display the modified data, and display the default summary statistics for all the columns.

[1] modified data for ISE 291 HW. Reference will be given in the solution.

Problem #B 50 marks

Consider data given in CSV file **HW4DataB** obtained from a public repository[2]. Consider the following data description:

Table 2: Data Description

Field	**Description**
gender	Gender of the student
race/ethnicity	The ethnicity the student belongs to (masked)
parental level of education	The highest level of education of either of the parent of the student
lunch	Whether the student pays for lunch at standard or free/reduced rate
test preparation course	Student took and completed the preparation course before the test or not
math score	Score student received in the math assessment (out of 30)
reading score	Score student received in the reading assessment (out of 25)
writing score	Score student received in the writing assessment (out of 40)
gk score	Score student received in the general knowledge (GK) assessment (out of 100)

Do the following tasks (in exact sequence) using data given in **HW4DataB** and Table-2:
☞ *Note: Solve all the above questions using Python. Use* ***Pandas*** *&* ***Sklearn*** *library for all the above analysis.*

B-1: **Given Data.** Read the data and display the data. Identify the number of rows and columns. Does any column have missing data? Display the description of both numeric and non-numeric columns.

B-2: **Type Consistency.** For each column, identify type of each field and verify that each column in Python is identified correctly.

B-3: **Inconsistent Data.** Looking at the data, two types of inconsistencies were discovered, some of the scores were entered with negative sign (by mistake). Or in some cases, we found larger values than the possible maximum score. For all such entries, assume the score is out of 100, and scale it out of corresponding maximum score. (*For example, if you see 70 in math score, you scale the score by dividing it with 100 and multiplying it by maximum score of math score (i.e. 30)*).

B-4: **Handling NaN values.** For any missing data in the score columns, take the average score of the student from other exam scores and replace the NaN value. (*HINT: Taking average across the row and not column. Remember the scores are based on different scales.*)

B-5: **Handling NaN values.** For any missing data in the non-numeric columns, fill the hole (or replace the NaN value) as follows: NaN value is to be replaced by the mode based on the race/ethnicity. (*For example, if for any student, lunch field is empty, then use the mode of the same race/ethnicity the student belongs to, not the mode across all records.*)

B-6: **Label Encoding.** Convert "parental level of education" and lunch using label encoder.

B-7: **One-Hot-Encoding**. For the 'test preparation course', convert it using one-hot-encoding. Since it has only two values, dropping one column will still give us the same information. 0 for completed and 1 for none. (Hint: Use the option drop_first)

B-8: **Normalization.** To be able to compare scores, scale all the scores between [0,1]. You can use normalize() from sklearn.preprocessing (make sure you understand the options).

B-9: **Clean & Prepared Data.** Display the modified data, and display the default summary statistics for all the columns.

[2] modified data for ISE 291 HW. Reference will be given in the solution.

Problem #C (Practice only. No submission required.)

Consider the following python methods, available in naive python, or pandas/sklearn libraries:

C-1: `pandas.DataFrame.index`

C-2: `pandas.DataFrame.columns`

C-3: `pandas.DataFrame.dtypes`

C-4: `pandas.DataFrame.select_dtypes()`

C-5: `pandas.DataFrame().apply()`

C-6: `pandas.DataFrame.map()`

C-7: `pandas.DataFrame.get_dummies()`

Answer the following questions for each of the above methods:

- List all the argument of the method.
- Classify the arguments as positional or keyword arguments.
- Identify the data types for each of the arguments.
- Write the default values for each of the arguments.

Consider the following python classes, available in sklearn library:

C-8: `sklearn.preprocessing.LabelEncoder.fit()`

C-9: `sklearn.preprocessing.LabelEncoder.transform()`

C-10: `sklearn.preprocessing.StandardScaler.transform()`

Answer the following questions for each of the above classes:

- List all the methods and properties.
- Discuss the `.fit()` method.
- Discuss the `.transform()` method.

☞ *Note: You can use the following online references to answer the above questions:*

♣ https://docs.python.org/3.8/library/functions.html#help

♦ https://docs.python.org/3/library/index.html

♥ https://pandas.pydata.org/pandas-docs/stable/index.html

♠ https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing

Problem #D (Practice only. No submission required.)

Consider data given in **HW4DataC.csv**. Following table gives meta information:

Table 3: Data Description

Field	Description
gender	Gender of the student
city	Home city of the student
region	Home Province of the student
Fjob	Father's job
Mjob	Mother's job
scores	Scores of the student in math, physics, chemistry, biology, reading, writing, and general knowledge. All are out of 30.

Do the following tasks (in exact sequence) using data given in **HW4DataC.csv** and Table-3:
☞ *Note: Solve all the above questions using Python (not by hand). Use **Pandas & Seaborn** libraries for all the above analysis. "differentiated by" in the above problem means that either have multiple plots, one plot with different colors, or one plot with sub-plots.*

D-1: **Given Data.** Read the data and display the data. Identify the number of rows and columns. Does any column have missing data? Display the description of both numeric and non-numeric columns.

D-2: **Compounded Data.** The column corresponding to 'scores' combines scores for all the subjects. Make new columns for each subject, and fill the data according to the 'scores' column.

D-3: **Handling NaN Values.** The fill NaN values in column 'Mjob' as 'HomeMaker'.

D-4: **Handling NaN Values.** The fill NaN values in column 'Fjob' according to the mode of the jobs in that city.

D-5: **Data Wrangling.** The values in column 'city' are written in full or short form. Reset all values to their full forms.

D-6: **Data Wrangling.** The values in column 'region' are written in many styles. Pick one style and reset all values in that style.

D-7: **Feature Generation.** Create a column called 'traits' and fill the values as follows:

- Fill the value as 'Engineering' if the student's top two scores are from math and physics.
- Fill the value as 'Medicine' if the student's top two scores are from biology and chemistry.
- Fill the value as 'Admin' if the student's top two scores are from writing and reading.
- Fill the value as 'Law' if the student's top score is from general knowledge.
- Fill the value as 'Unclear' for any student who does not satisfy any of the above conditions.

D-8: **Standardization.** Scale all the exam column to have mean of 0 units and standard deviation of 1 unit.

D-9: **Visualization.** Draw box-plots for each exam score column. Group/differentiate your data based on 'city'. Do you see outliers? Any difference in the distribution between different groups?

D-10: **Visualization.**Draw box-plots for each exam score column. Group/differentiate your data based on 'region'. Do you see outliers? Any difference in the distribution between different groups?

D-11: **Visualization.**Compute the average scores for each gender for each exam. Is there any difference in the average score w.r.t gender? Use any plot to show the distribution.

Problem #E (Practice only. No submission required.)

Explain the following *Python* codes. Assume `df` represents an existing pandas’ dataframe, where the columns are `C1, c2,....` The columns with odd numbers are categorical, and columns with even numbers are numerical:

Code-1:

In [1]:
```
df["C1"]=df["C1"].apply(lambda x:"_".join(x.split()))
```

Code-2:

In [2]:
```
s = "$10,000.00"
x = int(s.replace(’$’, ’’).replace(’,’,’’).replace(’.00’,’’))
print(x)
```

Code-3:

In [3]:
```
n=df.columns[df.isna().any()]
m = df.loc[:,’C5’:’C15’].columns
o = df[ df[’C2’] == 22 ].index
p = df.select_dtypes(exclude=’object’).columns
```

Code-4:

In [4]:
```
for c in df.columns:
   if df[c].isna().any():
      value = df[c].mean() if df[c].dtype!=’object’ else df[c].mode()
      df[c].fillna(value,inplace=True)
```

Code-5:

In [5]:
```
df = pd.get_dummies(df, columns=[’C1’,’C5’,’C13’],drop_first=True)
```

Code-6:

In [6]:
```
dfrom sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(df[[’C2’]])
df[’C4’]=scaler.transform(df[[’C4’]])
```

Code-7:

In [7]:
```
m = {’Value11’:’S’, ’Value12’:’S’, ’Value13’:’S’, ’Value14’:’S’,
   ’Value21’:’T’, ’Value22’:’T’, ’Value23’:’T’,
   ’Value31’:’W’, ’Value32’:’W’, ’Value41’:’E’}
df[’C3’] = df[’C3’].map(m)
```