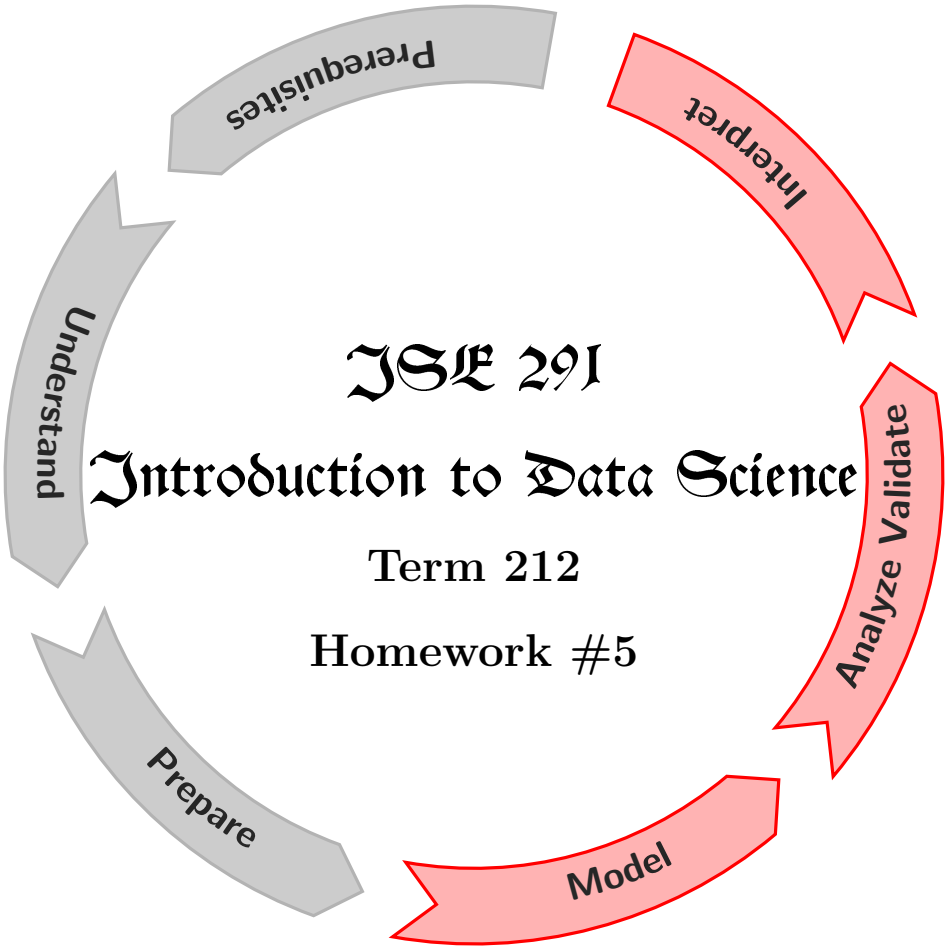
[The HW must be submitted as one .ipynb file. Write names & IDs of all the group members.]



JSE 291

Introduction to Data Science

Term 212

Homework #5

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:

- You should submit the solutions for the **FIRST TWO** problems only.
- Every sub-problem (part) should be answered on a DIFFERENT CELL as given in the template.
- EVERY CELL should have problem and part number clearly written in the first line.
- All cells of your homework should be in CHRONOLOGICAL order. One cell per sub-problem.
- Any text should be written as comment in the code cell. Do NOT modify code cell into markdown cell.
- Submit entire HW as ONE single .ipynb document.
- **Do NOT add/delete** any cell in the given template.
- 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 **HW5Data1** obtained from a public repository[1]. Consider the following data description:

Table 1: Data Description

Field	Description
Application. ID	Serial number of the application file
Age	Age of the applicant in years at the time of applying for admission
Gender	Gender of the applicant
Location	Home province of the applicant (Western, Central, Eastern)
Parents Edu.	University level education of the applicant's parents ('M' mother is graduated, 'F' father is graduated, or 'M+F both graduated from a university)
Siblings	Are there any siblings of the applicant currently in the university (Yes or No)
RAM-1 Score	RAM1 test score of the applicant (out of 100)
RAM-2 Score	RAM2 test score of the applicant (out of 100)
HS Score	High school total score of applicant (out of 100)
School Rank	Rating of the high school the applicant graduated from. (Low, Avg, Top)
SOP	Rating of Statement of Purpose of a student (out of 5)
Admission Status	The applicant has been admitted into the university (Yes or No)

☞ *Note: Solve all the above questions using Python. Use* ***Pandas***, ***Seaborn***, ***Sklearn***, *etc. libraries for all the above analysis*

Do the following tasks using data given in **HW5Data1** and Table-1:

A-1: **Given Data.** Read the data and display the data. Count the number of rows and columns in the data. Count the number of non-null rows for each column. Display the description of both numeric and non-numeric columns.

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

A-3: **Filtering.** Drop the column 'Application. ID'.

A-4: **Visualization-I.** Do the following:

- Draw the histogram of all numerical columns, differentiated by 'Admission Status'. Draw all the histograms in one figure (Use subplots).
- Draw the histogram of all non-numerical columns (except 'Admission Status'), differentiated by 'Admission Status'. Draw all the histograms in one figure (Use subplots).

A-5: **Visualization-II.** Do the following:

- Depict each pairwise combination of numerical columns, differentiated by 'Admission Status'. Draw all the plots in one figure. Are there any patterns in Ram-1, Ram-2 and SOP scores, that are visible from the plot.
- Depict each combination of numerical and non-numerical column (except 'Admission Status'), differentiated by 'Admission Status'. Draw all the plots in one figure (Use subplots).
- Draw a count plot of 'Admission Status', differentiated by 'Parents Edu.', 'School Rank' and 'Siblings'. If any of siblings is currently enrolled, and both parents are educated, then what are the chances of getting admitted?
- Depict 'School Rank' vs 'HS Score', differentiated by 'Admission Status', 'Parents Edu.', and 'Siblings'. Is there any difference between getting a High value in 'HS Score' from High 'School Rank'?
- Depict 'RAM-1 Score' vs 'RAM-2 Score', differentiated by 'Admission Status', 'Parents Edu.', and 'Siblings'.

A-6: **Tables.** Do the following:

- Which combination of 'Location' and 'Gender' has max number of records, and which combination has minimum number of records?
- Which combination of 'Location' and 'Gender' has max number of admitted applicants, and which combination has minimum number of admitted applicants?
- What is the average number of admitted applicants for each combination of 'Parents Edu.', 'School Rank' and 'Siblings'?

[1] data created for ISE 291 HW.

Problem #B **50 marks**

Consider data given in CSV file **HW5Data2** [2]. Consider the following data description: ☞ *Note: Solve*

Table 2: Data Description

Field	Description
Gender	Gender of the student
Location	Home City of the student
Quiz-1	Score of the student in Quiz-1
Quiz-2	Score of the student in Quiz-2
Quiz-3	Score of the student in Quiz-3
Quiz-4	Score of the student in Quiz-4
Major-1	Score of the student in Major-1
Major-2	Score of the student in Major-2
Major-3	Score of the student in Major-3
Final	Score of the student in the final exam.

*all the above questions using Python. Use **Pandas**, **Seaborn**, **Sklearn**, etc. libraries for all the above analysis.*

Do the following tasks (in exact sequence) using data given in **HW5Data2** and Table-2:

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 in **HW5Data2**, identify type of each field and verify that each column in Python is identified correctly. If there is any inconsistency, then resolve it.

B-3: **Normalization.** For each score column in **HW5Data2**, apply the standard scaler, such that the mean is zero and standard deviation is one.

B-4: **Correlation Analysis.** Do the following:

- Calculate the correlation between all the score columns of **HW5Data2**.
- Identify top 3 variables that are highly correlated with 'Final' score column.
- Which pair of score columns are strongly correlated?

B-5: **PCA.** Do the following:

- Get first two principal components of the data without considering 'Gender', 'Location' and 'Final' columns.
- Add the two principal components to the dataframe, and rename the components 'PC1' and 'PC2' respectively.
- Construct a scatter plot using the first two principal components of the data. Can the principal components separate 'Final' variable? To help in visualization, use the following color style: Anyone scoring above 85 in Final is depicted in green color, anyone scoring between 65 and 84.9 in Final is depicted in blue color, and the rest in red color.
- Differentiate the above plot using 'Gender'. In a separate plot, differentiate the above plot using 'Location'.
- How much variation do each principal component capture?
- What are the coefficients (the u vector) of the linear combination of input variables for the first PC?

[2] data created for ISE 291 HW.

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

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

C-1: `pandas.DataFrame.corr`

C-2: `pandas.DataFrame.concat`

C-3: `pandas.DataFrame.from_records`

C-4: `pandas.crosstab`

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

C-6: `matplotlib.pyplot.subplots()`

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

C-8: `pandas.DataFrame.max()`

Answer the following questions for each of the above methods:

- State the purpose/usage of the method/attribute.
- List all the argument of the method.
- Classify the arguments as positional or keyword arguments.
- Write the default values for each of the keyword arguments.

Consider the following python class, available in sklearn library:

C-9: `sklearn.decomposition.PCA`

Answer the following questions for the above class:

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

☞ *Note: You must use* ***help()*** *function from python to answer all the above questions.*

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

Consider data given in **HW5Data3.csv**[3].

Table 3: Data Description

Field	Description
risk	-3, -2, -1, 0, 1, 2, 3; where 3 implies highest risk.
make	company and model/name of the car
fuel-type	diesel, gas.
aspiration	std, turbo.
num-of-doors	four, two.
body-style	hardtop, wagon, sedan, hatchback, convertible.
drive-wheels	4wd, fwd, rwd.
engine-location	front, rear.
wheel-base	continuous from 86.6 120.9.
length	continuous from 141.1 to 208.1.
width	continuous from 60.3 to 72.3.
height	continuous from 47.8 to 59.8.
curb-weight	continuous from 1488 to 4066.
engine-type	dohc, dohcv, l, ohc, ohcf, ohcv, rotor.
num-of-cylinders	eight, five, four, six, three, twelve, two.
engine-size	continuous from 61 to 326.
fuel-system	1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi.
bore	continuous from 2.54 to 3.94.
stroke	continuous from 2.07 to 4.17.
compression-ratio	continuous from 7 to 23.
horsepower	continuous from 48 to 288.
peak-rpm	continuous from 4150 to 6600.
city-mpg	continuous from 13 to 49.
highway-mpg	continuous from 16 to 54.
price	continuous from 5118 to 45400.

Do the following tasks (in exact sequence) using data given in **HW5Data3**:

D-1: **Given Data.** Does any column have missing data? If yes, then drop all the rows that contain any missing values.

D-2: **Type Consistency.** For each column in **HW5Data3**, identify type of each field and verify that each column in Python is identified correctly. If there is any inconsistency, then resolve it.

D-3: **Normalization.** For each score column in **HW5Data3**, apply the standard scaler, such that the mean is zero and standard deviation is one.

D-4: **Correlation Analysis.** Identify top 5 numerical variables that are highly correlated with 'price' column.

D-5: **PCA.** Do the following:

- Get first two principal components of the numerical data without considering 'price' column.
- Add the two principal components to the dataframe, and rename the components 'pc1' and 'pc2' respectively.
- Construct a scatter plot using the first two principal components of the data, differentiate the plot using 'price' column. Can the principal components separate 'price' column?
- Drop 'make','pc1' and 'pc2' column from the dataframe, and convert all other non-numerical columns to numerical column using one hot encoding.
- Repeat the first three steps using numerical (and encoded) columns as inputs to the pca.

☞ *Note: Solve all the above questions using Python (not by hand). Use* ***Pandas***, ***Seaborn***, ***SkLearn***, *etc. libraries for all the above analysis.*

[3]Kibler, D., Aha, D. W., & Albert, M. (1989). Instance-based prediction of real-valued attributes. *Computational Intelligence*, *5*,51–57

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. Also, assume that relevant libraries are imported before executing the following code:

Code-1:

In [1]:
```
corr = df.corr()
sns.heatmap(corr)
```

Code-2:

In [2]:
```
print(df['C1'].values.reshape(-1,1))
print(df[['C1']]
```

Code-3:

In [3]:
```
ndf=pd.concat([df['C1'], df['C2'], axis=1)
display(ndf.sample(5))
```

Code-4:

In [4]:
```
plt.figure()
sns.relplot(x='C2',y='C4',hue='C1', palette=['r','b','g','m','c'],
kind='scatter',alpha=0.75,height=5, aspect=1,data=df)
plt.show()
```

Code-5:

In [5]:
```
cat_columns = df.select_dtypes('object').columns.drop('C1')
num_columns = df.select_dtypes(exclude='object').columns
fig,axes = plt.subplots(len(cat_columns), len(num_columns), figsize=(9,9))
for c,nCol in enumerate(num_columns):
   for r,cCol in enumerate(cat_columns):
      sns.boxplot(y=cCol,x=nCol,hue='C1',data=df, ax=axes[r][c])
plt.show()
```

Code-6:

In [6]:
```
X = df.iloc[:,:-1].values
y = df.iloc[:, -1].values
print(np.c_[np.ones(len(df.index)), X])
```

Code-7:

In [7]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(df[['C2','C4']])
df[['C2','C4']] = scaler.transform(df[['C2','C4']])
df[['C6','C8']] = scaler.transform(df[['C6','C8']])
```