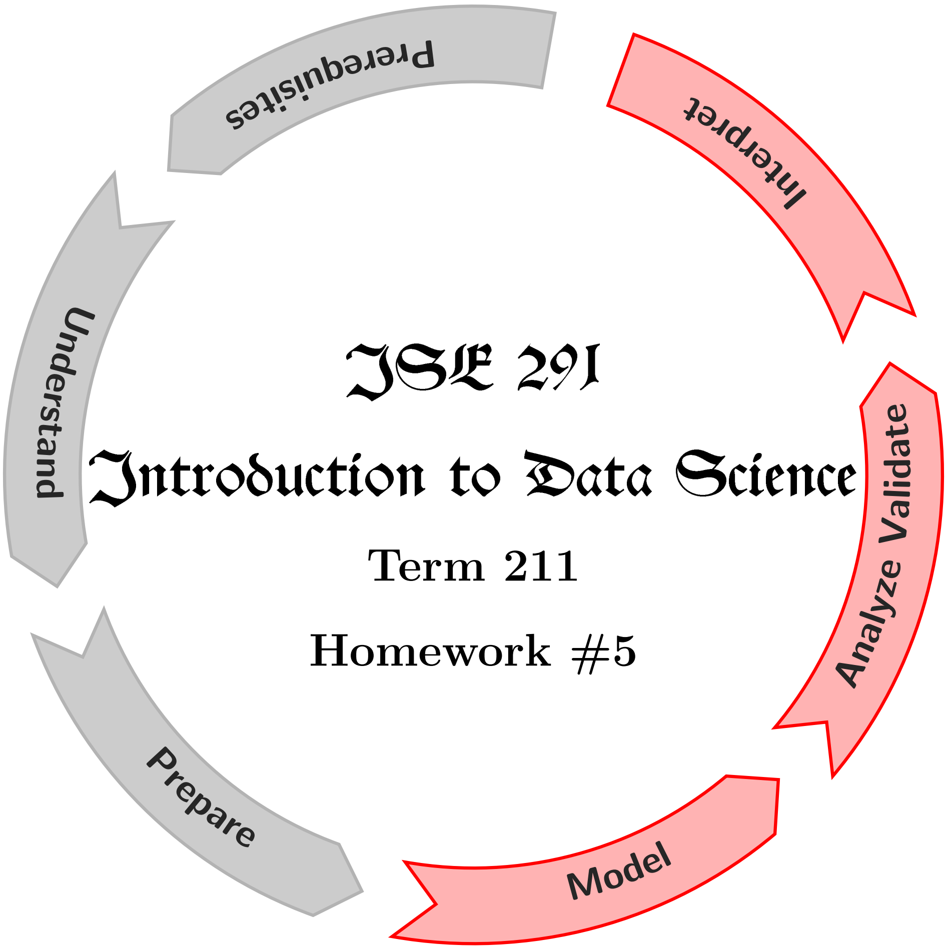
[The HW must be submitted as one .ipynb file. Write names & IDs of all the group members.]



ISE 291

Introduction to Data Science

Term 211

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:

- 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

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

Consider data given in file **HW5DataA** [1]. Consider the following data description:

Table 1: 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.

Do the following tasks (in exact sequence) using data given in **HW5DataA** 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 **HW5DataA**, 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 **HW5DataA**, apply the standard scaler, such that the mean is zero and standard deviation is one.

B-4: **Visualization.** Draw pairwise scatter plots each pair of score columns in **HW5DataA**. Also, for each score column draw KDE (Kernel Density Estimation). Differentiate the pairwise plots and KDE plot by 'Gender' column.

B-5: **Correlation Analysis.** Do the following:

- Calculate the correlation between all the score columns of **HW5DataA**.
- Identify top 3 variables that are highly correlated with 'Final' score column.
- Which pair of score columns are strongly correlated?

B-6: **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?

[1] data created for ISE 291 HW.

Problem #B 50 marks

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

Consider data given in CSV file **HW5DataB** [2]. Consider the following data description:

Table 2: Data Description

Field	Description
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.

Do the following tasks (in exact sequence) using data given in **HW5DataB** 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 statistical summaries of all the columns.

B-2: **Type Consistency.** For each column in **HW5DataB**, identify the type for each field based on value. Also, identify the datatypes in Python. Report and resolve any inconsistency.

B-3: **Normalization.** For each column in **HW5DataB**, apply the standard scaler, such that the mean is zero and standard deviation is one. Display the summaries of all the columns.

B-4: **Cross Normalization.** For each column in **HW5DataC**, apply the standard scaler fitted (learned) from **HW5DataB** data. Display the summaries of all the columns in **HW5DataC** data.

B-5: **OLS Regression.** The hypothesis is that the quiz and major exam scores are linearly related to final exam score. Use the following formula to calculate the OLS coefficient estimates of all **HW5DataB** data. Take column 'Final' as the output column, and all other columns as input column.

$$\boldsymbol{\theta} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{y}$$

B-6: **OLS Regression.** The hypothesis is that the quiz and major exam scores are linearly related to final exam score. Do the following:

- Use the sklearn library to calculate the OLS coefficient estimates of all **HW5DataB** data. Take column 'Final' as the output column, and all other columns as input column.
- Compare the coefficients obtained in Part *B*-5 with the above coefficients. Report any differences in between the coefficients from Parts *B*-5 and *B*-6.
- Using the above OLS coefficient estimates, calculate the MSE for data given in **HW5DataC**.

B-7: **Ridge Regression.** It may be possible that the quiz and major exam scores are not really independent. Thus, the coefficients needs regularization (penalization). Do the following:

- Do the ridge analysis, taking all **HW5DataB** data as the training data. Use 10-fold cross validation, and pick the best value of alpha from $10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}$
- Using the above coefficient estimates, calculate the MSE for data given in **HW5DataC**.

B-8: **Lasso Regression.** It may be possible that not all the quiz and major exam scores are helpful in predicting final score. Thus, the coefficients needs selection (penalization). Do the following:

- Do the lasso analysis, taking all **HW5DataB** data as the training data. Use 10-fold cross validation, and pick the best value of alpha from $10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}$
- Using the above coefficient estimates, calculate the MSE for data given in **HW5DataC**.

B-9: **Regression Analysis.** Compare and contrast the coefficient estimates obtained from Parts *B*-6, *B*-7, *B*-8 and *B*-9.

[2] modified data from Problem-A

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()`

C-9: `sklearn.model_selection.train_test_split()`

C-10: `sklearn.metrics.mean_squared_error()`

C-11: `numpy.linalg.inv()`

C-12: `numpy.c_`

C-13: `numpy.linspace()`

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`

C-10: `sklearn.linear_model.LinearRegression`

C-11: `sklearn.linear_model.RidgeCV`

C-12: `sklearn.linear_model.LassoCV`

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 **HW5DataD.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 **HW5DataD**:

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 **HW5DataD**, 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 **HW5DataD**, 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’]])
```