COMP20008 - 2018 - SM2 - Project Phase 2

Release Date: 11:59am Monday, 3^{rd} September 2018

Due Date: 11:59am Friday, 21^{st} September 2018

Submission is via the LMS

Please, make sure you get a submission confirmation email once you submit your assignment. Otherwise, it will be considered as a late submission.

Phase 2: Python Data Wrangling (15 marks, worth 15% of subject grade)

For banks, risk management and default detection has always been a crucial part in issuing credit cards. Defaults in credit cards can result in a great financial loss. In this phase, you will practice your Python wrangling skills, specifically correlation, classification and clustering parts with a modified version of an available default credit cards dataset at the UCI Machine Learning Repository. The dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

In this phase, you will be working with the "UCI_Credit_Card_Modified.csv" dataset. It has 200 records for credit card users, each is described by 25 variables.

Libraries to use are Pandas, Matplotlib, NumPy, SciPy, seaborn and sklearn. You will need to write Python 3 code (Jupyter notebook) and work with the topics discussed in workshops weeks 6-8. If you are using other packages, you must provide an explanation in your code about why it is necessary.

Dataset Content

There are 25 variables: (1 ID, 1 label, 23 features/attributes)

- ID: ID of each client
- limit_bal: Amount of given credit in NT dollars (includes individual and family/supplementary credit
- is_male: **Binary** : Gender (1=male, 0=female)
- education: **Categorical** : (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- is_married: **Binary** : Marital status (1=married, 0= not married, i.e. single or others)
- age: Age in years
- pay_1: **Numerical** : Repayment status in September, 2005 (0=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above)
- pay_2: **Numerical** : Repayment status in August, 2005 (scale same as above)
- pay_3: **Numerical** : Repayment status in July, 2005 (scale same as above)
- pay_4: **Numerical** : Repayment status in June, 2005 (scale same as above)
- pay_5: **Numerical** : Repayment status in May, 2005 (scale same as above)
- pay_6: **Numerical** : Repayment status in April, 2005 (scale same as above)
- bill_amt1: **Numerical** : Amount of bill statement in September, 2005 (NT dollar)
- bill_amt2: **Numerical** : Amount of bill statement in August, 2005 (NT dollar)
- bill_amt3: **Numerical** : Amount of bill statement in July, 2005 (NT dollar)
- bill_amt4: **Numerical** : Amount of bill statement in June, 2005 (NT dollar)
- bill_amt5: **Numerical** : Amount of bill statement in May, 2005 (NT dollar)
- bill_amt6: **Numerical** : Amount of bill statement in April, 2005 (NT dollar)
- pay_amt1: **Numerical** : Amount of previous payment in September, 2005 (NT dollar)
- pay_amt2: **Numerical** : Amount of previous payment in August, 2005 (NT dollar)
- pay_amt3: **Numerical** : Amount of previous payment in July, 2005 (NT dollar)
- pay_amt4: **Numerical** : Amount of previous payment in June, 2005 (NT dollar)
- pay_amt5: **Numerical** : Amount of previous payment in May, 2005 (NT dollar)
- pay_amt6: **Numerical** : Amount of previous payment in April, 2005 (NT dollar)
- label: **Binary** :Default payment (1=yes, 0=no)

Import Required Python Libraries and Load the Data

Please write here all the Python libraries you will be using! Also load the dataset (.csv) in a dataframe object.

In [ ]:
```python
#import ....
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

from scipy.cluster.hierarchy import dendrogram, linkage
from scipy.spatial.distance import pdist, squareform

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

from sklearn.tree import DecisionTreeClassifier
```

Helper Functions

This section includes few functions discussed in workshops week 6 and 7, and will be used in this assignment.

In [ ]:
```
def VAT(R):

    R = np.array(R)
    N, M = R.shape
    if N != M:
        R = squareform(pdist(R))

    J = list(range(0, N))

    y = np.max(R, axis=0)
    i = np.argmax(R, axis=0)
    j = np.argmax(y)
    y = np.max(y)


    I = i[j]
    del J[I]

    y = np.min(R[I,J], axis=0)
    j = np.argmin(R[I,J], axis=0)

    I = [I, J[j]]
    J = [e for e in J if e != J[j]]

    C = [1,1]
    for r in range(2, N-1):
        y = np.min(R[I,:][:,J], axis=0)
        i = np.argmin(R[I,:][:,J], axis=0)
        j = np.argmin(y)
        y = np.min(y)
        I.extend([J[j]])
        J = [e for e in J if e != J[j]]
        C.extend([i[j]])

    y = np.min(R[I,:][:,J], axis=0)
    i = np.argmin(R[I,:][:,J], axis=0)

    I.extend(J)
    C.extend(i)

    RI = list(range(N))
    for idx, val in enumerate(I):
        RI[val] = idx

    RV = R[I,:][:,I]

    return RV.tolist(), C, I

def my_entropy(probs):
    return -probs.dot(np.log2(probs))
def mutual_info(X,Y):

    df = pd.DataFrame.from_dict({'X' : X,  'Y' :Y})
    Hx = my_entropy(df.iloc[:,0].value_counts(normalize=True, sort=False))
    Hy = my_entropy(df.iloc[:,1].value_counts(normalize=True, sort=False))


    counts = df.groupby(["X","Y"]).size()
    probs = counts/ counts.values.sum()
    H_xy = my_entropy(probs)

    # Mutual Information
    I_xy = Hx + Hy - H_xy
    MI = I_xy
    NMI = I_xy/min(Hx,Hy)
    return NMI
```

In [ ]:
```
default_credit_card_df =  pd.read_csv("UCI_Credit_Card_Modified.csv", index_col="ID")
default_credit_card_df.head(3)
```

1 Data Preparation and Dimension Reduction (5 Marks)

1.1 Categorical Features:

In this assignment, you will use the dataset to perform clustering and classification methods. In this regard, using categorical features might lead to inaccurate evaluation results. Therefore, categorical features should be converted to numerical first. The provided default credit card dataset contains one categorical attribute "education" (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown). Write code to replace the "education" column with another three numerical columns of type integer, with the names "graduate_school", "university" and "high_school". The three numerical columns will have the value 0 or 1, where each categorical value will be converted to a binary vector as follow: **(1 Mark)**

education	graduate_school	university	high_school
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	0
5	0	0	0
6	0	0	0

The resulted dataframe should include 26 columns in the following order:

limit_bal	*is_male*	*grad_school*	*university*	*high_school*	*is_married*	*age*	*pay_1*	*pay_2*
pay_3	*pay_4*	*pay_5*	*pay_6*	*bill_amt1*	*bill_amt2*	*bill_amt3*	*bill_amt4*	*bill_amt5*
bill_amt6	*pay_amt1*	*pay_amt2*	*pay_amt3*	*pay_amt4*	*pay_amt5*	*pay_amt6*	*label*	

Note that the "ID" column should be the dataframe index. The output of this step should print the first two rows of preprocessed dataframe using df.head(2).

```
In [ ]: ### answer Q1.1
```

1.2 Feature Scaling:

The second step in the data preparation is the feature scaling/transformation. You will use **sklearn.preprocessing.StandardScaler()** function to normalise each attribute separately, to have 0 mean and unit variance. To do so, implement the following steps: **(1 Mark)**

- Create a features matrix "X" which contains all columns in your dataframe except the "label" column.
- Create a vector "y" which contains the values in the column "label" in your dataframe.
- Use the StandardScaler() function to normalise the features matrix "X". Store the transformed features in "X_scaled" matrix.

The output of this question should print shape, min, max, average and standard deviation of "X_scaled" matrix in the following format:

```
***
Q1.2: X_scaled matrix details
Shape: #
Min: #
Max: #
Average: #
Standard Deviation: #
***
```

where # is the vector of calculated values rounded to 4 decimal places.

```
In [ ]: ### answer Q1.2
```

1.3 Dimension Reduction: PCA

It is hard to visualise default clients versus non-default clients since each client has 25 attributes. Write code to reduce number of dimensions for "X" matrix to two dimensions by using PCA. Store the result in "X_reduced" matrix. Then, use "X_reduced" matrix to display a scatter plot. X-axis should contain the first principal component and y-axis should contain the second principal component. Default clients should be colored with red, while other non-default clients should be colored with blue. **(3 Marks)**

Then, implement the same steps with "X_scaled" data matrix instead of "X" matrix. Now, you should have two plots. Assign the titles "Dimension reduction without feature scaling" and "Dimension reduction with feature scaling" for the plots repectively. The final output of this question should be one figure with the two required plots.

- Based on the visualised plots, which features data matrix is better for PCA, "X_scaled" or "X"? Why?
- Do you think using PCA is a good idea for visualising the default credit card clients? Yes/No, Why?

In [ ]:
```
### answer Q1.3
```

```
### answer Q1.3 justification
```

2 Clustering and Clustering Visualisation (5 Marks)

In this section you will perform some of the clustering and clustering visualisation techniques.

2.1 Hierarchical Clustering:

Write code to plot the dendrogram using "X_scaled" matrix for each of the linkage methods: **COMPLETE** and **SINGLE**. You should also use Euclidean distance to calculate the dissimilarity matrix. **(2 Marks)**

The output of this question should be one figure with two dendrogram sub-plots. First sub-plot should have the title "Agglomerative clustering with complete linkage method", while the second sub-plot should have the title "Agglomerative clustering with single linkage method".

Following this, you need to look at the y-axis range for the generated dendrograms and justify the different y-axis range/scale for each dendrogram?

In [ ]:
```
### answer Q2.1
```

```
### answer Q2.1 justification
```

2.2 Clustering Visualisation:

In this question, you will plot the heatmap for both dissimilarity matrix and ordered dissimilarity matrix. First, calculate the "Euclidean distance" dissimilarity matrix for the "X_scaled" data matrix. Next, use the VAT() function discussed in workshop-week6, in order to get the ordered dissimilarity matrix for the "X_scaled" data matrix. Finally, plot the heatmap for "Euclidean" dissimilarity matrix and the ordered one (i.e. VAT heatmap). **(3 Marks)**

The final output of this question should be one figure with two sub-plots. Use appropriate titles and xy-labels for each sub-plot.

Is there any relation between the ordered dissimilarity matrix (VAT heatmap) and the complete linkage dendrogram plots? If yes, describe one relation between both plots?

Knowing that the dataset contains two classes (default and non-default clients), does VAT heatmap give an accurate number of clusters, Yes|No? Why?

In [ ]:
```
### answer Q2.2
```

```
### answer justification 2.2
```

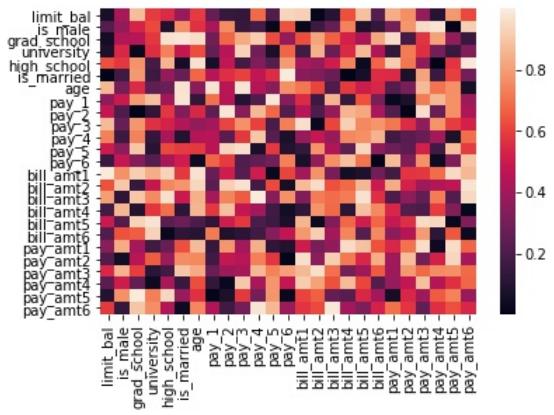
3 Correlation and Mutual Information (7 Marks)

In this section you will investigate the correlation and mutual information between credit card dataset attributes and the class label.

3.1 Pearson Correlation:

Write code to calculate a correlation matrix of size 25 x 25 (25 is the number of attributes) using the "X_scaled" data matrix. The caculated square symmetric matrix will contain the correlation between every of attributes. For example, a value in row i and column j should contain the Pearson Correlation r_{ij} between two attributes i and j. Then, plot the heatmap for the calculated correlation matrix. **(3 Marks)**

The final output for this question should look like the below heatmap plot. Note that the values in this plot are randomly generated and you should get different colors/values for your visualised heatmap.



From your visualised heatmap, report three different/interesting findings/explanations regarding the calculated correlation between every pair of the 25 attributes? For example you might focus the highest and/or lowest r values, and justify/explain whether you expected such high or low correlation between these specific attributes or not.

In [ ]:
```
### answer Q3.1
```

```
### answer Q3.1 justification
```

3.2 Mutual Information:

In this question, you will use the provided mutual_info() and my_entropy() functions discussed in workshop-week7 to calculate the mutual information between different attributes and the class label. Since discretisation is an important preprocessing for MI calculations, you will be using the "X" data matrix (i.e. without scaling) in this question. By looking at the values of the attributes in the "X" data matrix, they can be grouped into 11 discrete and 14 numerical attributes. **(4 Marks)**

Discrete columns are ['is_male', 'grad_school', 'university', 'high_school', 'is_married', 'pay_1', 'pay_2', 'pay_3', 'pay_4', 'pay_5', 'pay_6'] and
Numerical columns are ['limit_bal', 'age', 'bill_amt1', 'bill_amt2', 'bill_amt3', 'bill_amt4', 'bill_amt5', 'bill_amt6', 'pay_amt1', 'pay_amt2', 'pay_amt3', 'pay_amt4', 'pay_amt5', 'pay_amt6']

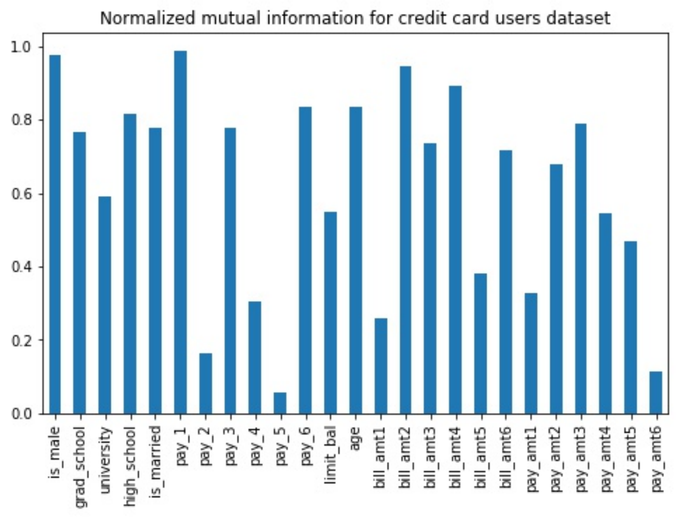
First, you need to discretize each of the numerical attributes by using the 4-bin equal-width technique. The output for this step should have the following format:

################# limit_bal #################
bin# 1: range [a,b)
bin# 2: range [a,b)
bin# 3: range [a,b)
bin# 4: range [a,b)
.
.
.
.

################# pay_amt6 #################
bin# 1: range [a,b)
bin# 2: range [a,b)
bin# 3: range [a,b)
bin# 4: range [a,b)

where a and b are the calculated range for each bin.

Next, you will calculate the normalised mutual information (NMI) between each of the columns (i.e. 25 in total) and the class label "y". Then display a bar plot for the calculated NMI values. The format of the bar plot should be similar to the below plot. Note that the NMI values in this plot are randomly generated and you should get different bars/values for your bar plot.



In [ ]:
```
### answer Q3.2
```

4 Classification (7 Marks)

In this section you will create a model to predict credit card default. This will enable you to answer the question: who is going to default into the credit card payments next month. Functions used in this section are covered in workshop-week8 materials.

4.1 Train-Test Split:

To evaluate the performance of any model/classifier, you should test it on unseen set of instances. Therefore, as a preparation for the model training, you will use train_test_split() from sklearn.model_selection to split your features matrix "X" into training and testing sets. Specifically, you will use train_test_split() function to randomly select 80% of the instances to be training and the rest to be testing. **(1 Mark)**

The output of the function should contain two matrices: **X_train** and **X_test**, and two vectors: **y_train** and **y_test**.

- X_train contains features of training instances (i.e. 80%) and y_train contains labels for the training instances.
- X_test contains features of test instances (i.e. 20%) and y_test contains labels for test instances.

The output of this question should print shape of X_train and X_test matrices as well as shape of y_train and y_test vectors, in the following format:

```
***
Q4.1: Train Test Split Results
X_train matrix: #
y_train labels: #
X_test matrix: #
y_test labels: #
***
```

where # is the shape values.

In [ ]:
```
### answer Q4.1
```

4.2 Feature Scaling:

Another preprocessing step before training any classifier is normalizing the data features. Because features with large scale might bias the trained classifier. Write code to transform each feature (i.e. column) in both training and testing sets, (i.e. X_train and X_test matrices), to have 0 mean and unit variance. Each transformed matrix should be stored in the matrix name, i.e. X_train and X_test. **(1 Mark)**

Is it a good idea to use the transformed matrix "X_scaled" as input for the split function and in this case, you don't have to scale/transform X_train and X_test again, Yes/No? Why?

In [ ]:
```
### answer Q4.2
```

```
### answer justification 4.2
```

4.3 K-nearest Neighbor Classifier:

In this question, you will build a K-NN classifier by using the built-in functions from sklearn package. To do so, implement the following steps: **(3 Marks)**

- Create an instance from the model/estimator (i.e. KNeighborsClassifier) and set the K (number of neighbors) to 3.
- Train/fit the model using X_train and y_train
- Evaluate the model by calculating the accuracy of the model on the test set (i.e. X_test)

The output of this question should be in the following format:

```
***
Q4.3: Default credit card user prediction using K-NN
Test accuracy: # %
***
```

Where # is the calculated classifier accuracy rounded to 2 decimal places.

As you might know, choosing the value of k-neighbors used for training the K-NN model is very important. Using the provided credit card clients dataset, what is the best value/s for k? You need to justify your choice either by providing extra code, visualisation plot and/or explanation?

In [ ]:
```
### answer Q4.3
```

```
### answer justification 4.3
```

4.4 Decision Tree Classifier:

In this question, you will build a decision tree classifier by following same steps provided in the previous question. Only, you will be using "DecisionTreeClassifier" instead of "KNeighborsClassifier". You should use the default values for all DT parameters. **(2 Marks)**

The output of this question should be in the following format:

```
***
Q4.4: Default credit card user prediction using DT
Test accuracy: # %
***
```

Where # is the calculated classifier accuracy rounded to 2 decimal places.

Do you expect the decision tree classifier to perform better than K-NN, Yes/No? why?

Does the resulted accuracies follow your expectations? If not, justify why this has happened? You might add extra code, visualisations and/or explanation to support your answer.

In [ ]:
```
### answer Q4.4
```

```
### answer Q4.4 justification
```

Marking scheme

Correctness (24 marks): For each of the 4 questions a mark will be allocated for level of correctness (does it provide the right answer, is the logic right), according to the number in parentheses next to each question. Correctness will also take into account the readability and labelling provided for any plots and figures (plots should include title of the plot, labels/scale on axes, names of axes, and legends for colours where appropriate).

Coding style **(1 Mark)**: Mark will be allocated for coding style. In particular the following aspects will be considered:

- Formatting of code (e.g. use of indentation and overall readability for a human)
- Code modularity and flexibility. Use of functions or loops where appropriate, to avoid redundant or excessively verbose definitions of code.
- Use of python library functions (you should avoid reinventing logic if a library function can be used instead)
- Code commenting and clarity of logic. You should provide comments about the logic of our code for each question, so that it can be easily understood by the marker.

The final mark of the assignment will be scaled from 25 to 15 using the following formula:
$your\_final\_mark = \frac{your\_mark\_out\_of\_25}{25} \times 15$

Submission Instructions

Via the LMS, submit a jupyter notebook containing the code. Make sure you get a submission receipt via email. If you didn't get a receipt via email, this means we didn't receive your submission and it will be considered as late submission.

Other

Extensions and Late Submission Penalties: If requesting an extension due to illness, please submit a medical certificate to the lecturer. If there are any other exceptional circumstances, please contact the lecturer with plenty of notice. Late submissions without an approved extension will attract a penalty of 10% of the marks available per 24hr period (or part thereof) that it is late. E.g. A late submission will be penalised 1.5 marks if 4 hours late, 3 marks if 28 hours late, 4.5 marks if 50 hours late, 6 marks if 73 hours late, 7.5 marks if 106 hours late, etc.

Phase 2 is expected to require 15-18 hours work.

Academic Honesty

You are expected to follow the academic honesty guidelines on the University website https://academichonesty.unimelb.edu.au (https://academichonesty.unimelb.edu.au)

Further Information

A project discussion forum has also been created on the subject LMS. Please use this in the first instance if you have questions, since it will allow discussion and responses to be seen by everyone. The Phase 1 project page will also contain a list of frequently asked questions.

Acknowledgements

Dataset used in this assignment was sampled from the original dataset available at the UCI Machine Learning Repository. Further, few changes have been applied to some attributes.

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] (http://archive.ics.uci.edu/ml]). Irvine, CA: University of California, School of Information and Computer Science.

The original dataset can be found here at the UCI Machine Learning Repository. https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset/home (https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset/home)