Tutorial 2

for

Data Mining - Transformation Version 1

Adam Ryan (14395076)

COMP47530

September 30, 2021

Contents

1 Question 1 **3**

1.1 Exercise 1 - Questions . . . 3

1.2 Question 1 - Answer . . . 3

2 Question 2 **8**

2.1 Exercise 2 - Questions . . . 8

2.2 Exercise 2 - Answers . . . 8

3 Question 3 **10**

3.1 Exercise 3 - Questions . . . 10

3.2 Exercise 3 - Answers . . . 10

1 Question 1

1.1 Exercise 1 - Questions

A module coordinator has just completed the module assessments, and s/he would like to perform a quick analysis on the students results in various components of the module. The main objective is to see if there is any correlation between the assessment components. The students' results are given in the file "Students_Results.csv". Using Python script, answer the following questions:

1. Find the minimum, maximum, mean and standard deviation for each Homework column and the exam column

2. Add an additional named as 'Homework Avg' for the average homework mark for each student. Assume that the weighting of the homework average is 25% and that of the examination is 75%, add an additional column named 'Overall Mark' for the overall folded mark.

3. Construct a correlation matrix of homework and exam variables. What can you conclude from the matrix?

4. Discuss various ways of treating the missing values in the dataset.

5. Use UCD grading system to convert the final mark into a grade (column named 'Grade'). Produce a histogram for the grades.

6. Save the newly generated dataset to "./output/question1_out.csv".

1.2 Question 1 - Answer

This is the answer to question one on the tutorial sheet.

1. Find the minimum, maximum, mean and standard deviation for each Homework column and the exam column

   - For the beginning step, we observe there are 54 rows of data. In total, one student has Homework 1 not completed, no students have not completed homework 2, and 6 students have not completed Homework 3. As the context of the question is in relation to assessment, we can assume that homeworks without grades were homeworks which were not submitted. Therefore, as an initial data cleansing exercise, we replace missing values with 0.

- Of the students who did complete the homeworks (i.e. of students with data populated) the stats are:
  - Homework 1
    - Minimum: 31
    - Maximum: 90
    - Standard Deviation: 17.877782
    - Mean: 55.641509
  - Homework 2
    - Minimum: 0
    - Maximum: 98
    - Standard Deviation: 15.441612
    - Mean: 89.83...
  - Homework 3
    - Minimum: 5
    - Maximum: 100
    - Standard Deviation: 21.585216
    - Mean: 47.687500
- After fixing the data quality error described above by replacing all missing values with 0 due to the context, we calculate the values as:
  - Homework 1
    - Minimum: 0.0
    - Maximum: 90.0
    - Mean: 54.611111111111114
    - Std. Deviation: 19.259319397315124
  - Homework 2
    - Minimum: 0.0
    - Maximum: 98.0
    - Mean: 89.83333333333333
    - Std. Deviation: 15.441612487898249 Count: 54.0
  - Homework 3
    - Minimum: 0.0
    - Maximum: 100.0
    - Mean: 42.388888888888886
    - Std. Deviation: 25.338029190733053

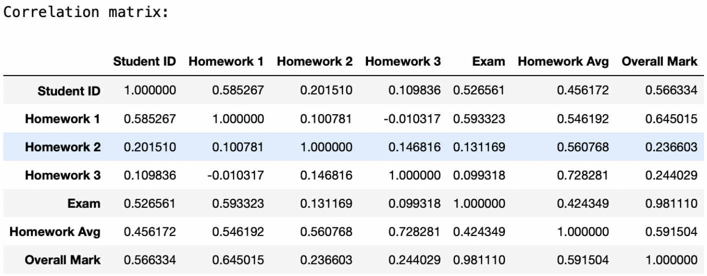
  - I use the dataframe describe feature which is present in pandas and provides descriptive statistics for numerical columns as this process is quick and the volume of data small enough to allow pandas usage.

2. Add an additional named as ‘Homework Avg’ for the average homework mark for each student. Assume that the weighting of the homework average is 25% and that of the examination is 75%, add an additional column named ’Overall Mark’ for the overall folded mark.

   - I continue using the cleansed dataset as described in the previous question, where missing Homework values were replaced by zero, cleanse the exam column in the same method (as this has one missing value), and add on the described column using pandas.

3. Construct a correlation matrix of homework and exam variables. What can you conclude from the matrix?

   - I use the correlation method on dataframe to produce the matrix. We see that the homeworks are, in general, only weakly positively correlated with one another. The first homework’s grade is much more strongly correlated with the exam results (with a value of 0.6 rounded to one decimal place) compared to the other homeworks. However, while Homework 1 was more strongly correlated with the exam results, the other two homeworks were more strongly correlated with the homework average. This homework average is elevated by the overall strong performance of students on the second homework which, as we see from the descriptive statistics, are highly inflated compared to Homework 1 and Homework 3. The correlation matrix, again after cleansing the missing values, is below:



```
Correlation matrix:
```

	Student ID	Homework 1	Homework 2	Homework 3	Exam	Homework Avg	Overall Mark
Student ID	1.000000	0.585267	0.201510	0.109836	0.526561	0.456172	0.566334
Homework 1	0.585267	1.000000	0.100781	-0.010317	0.593323	0.546192	0.645015
Homework 2	0.201510	0.100781	1.000000	0.146816	0.131169	0.560768	0.236603
Homework 3	0.109836	-0.010317	0.146816	1.000000	0.099318	0.728281	0.244029
Exam	0.526561	0.593323	0.131169	0.099318	1.000000	0.424349	0.981110
Homework Avg	0.456172	0.546192	0.560768	0.728281	0.424349	1.000000	0.591504
Overall Mark	0.566334	0.645015	0.236603	0.244029	0.981110	0.591504	1.000000

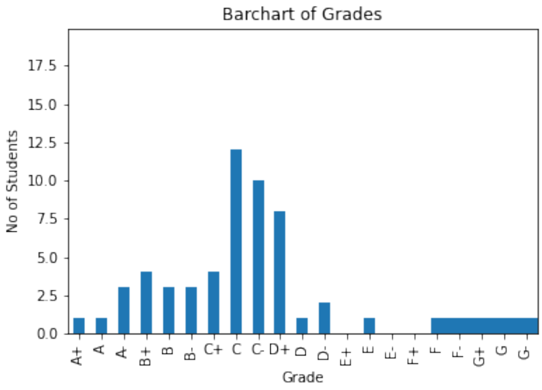
   - I do not remove the student ID (from the photo above) in creating the correlation matrix as this can be ignored in interpretting the result as the first row and column in the set, but it can be seen that this is removed in the submission file.

4. Discuss various ways of treating the missing values in the dataset.

   - As described in the tutorial, we were instructed to cleanse the homework and exam columns prior to calculations in question 1.

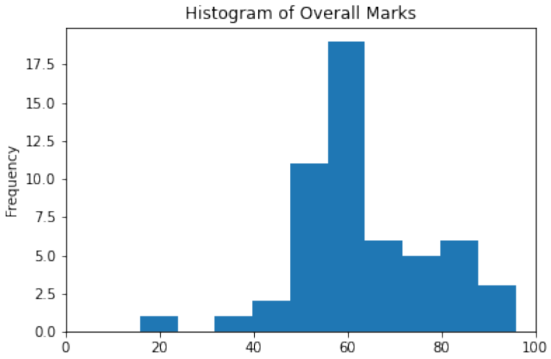
   - The first method of cleansing, which I implemented, is to populate the missing values with 0. This method is sensible in the context of homework assignments and exams, as students may have not completed the assignment or exam in which case they should receive a zero.
   - The second method is to leave these data as empty; students may have received extenuating circumstances resulting in these components not being weighted for grading or no homework grade being assigned. In this instance, a null value is more sensible for the assignment than any numerical grade as it's a more accurate description of the grade (i.e. it is not applicable).
   - Another method to cleanse the data would be to assign the class average to the homework or exam columns. This method is not sensible given the context, as it punishes students who received below the mean grade who submitted the assignments while rewarding those who did not complete the assignment.

5. Use UCD grading system to convert the final mark into a grade (column named 'Grade'). Produce a histogram for the grades.
   - A histogram looks at numeric values rather than the categorical grades, so I have produced a barplot of the grades and a histogram of the marks as displayed below. We see the grades are concentrated around a C grade, and skews leans towards higher marks as opposed to being more evenly distributed across all grades:



6. Save the newly generated dataset to "./output/question1_out.csv".
   - Done.

Please note that in the test script, there are roundings of grades which occur; in order to pass the test script I have had to adjust some calculations to round however I would not ordinarily this. Similarly, there is a customer which is flaged as being an A grade when with the computer science conversion grading scheme they are actually an A-. In my grade bucketing, I have explicilty used the Computer Science grading scheme rather than the general UCD grading scheme.


Histogram of Overall Marks
Frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
20
40
60
80
100

2 Question 2

2.1 Exercise 2 - Questions

The file "Sensor_Data.csv" contains data obtained from a sensory system. Some of the attributes in the file need to be normalised, but you don't want to lose the original values.

1. Generate a new attribute called "Original Input3" which is a copy of the attribute "Input3". Do the same with the attribute "Input12" and copy it into Original "Input12".

2. Normalise the attribute "Input3" using the z-score transformation method.

3. Normalise the attribute "Input12" in the range [0:0; 1:0].

4. Generate a new attribute called "Average Input", which is the average of all the attributes from "Input1" to "Input12". This average should include the normalised attributes values but not the copies that were made of these.

5. Save the newly generated dataset to "./output/question2_out.csv".

2.2 Exercise 2 - Answers

This is solution to the question in Exercise 2.

1. Generate a new attribute called "Original Input3" which is a copy of the attribute "Input3". Do the same with the attribute "Input12" and copy it into Original "Input12".

   - Done.

2. Normalise the attribute "Input3" using the z-score transformation method.

   - Done, wrote the method.

3. Normalise the attribute "Input12" in the range

   $$0:0;1:0$$

   .

   - Done, wrote the method.

4. Generate a new attribute called "Average Input", which is the average of all the attributes from "Input1" to "Input12". This average should include the normalised attributes values but not the copies that were made of these.
    - Done, mean over list of columns in df with format of column name as 'input' and number less than thirteen with axis of one.
5. Save the newly generated dataset to "./output/question2_out.csv".
    - Done

3 Question 3

3.1 Exercise 3 - Questions

The files "DNA_Data.csv" contains biological data arranged into multiple columns. We need to compress the information contained in the data.

1. Reduce the number of attributes using Principal Component Analysis (PCA), making sure at least 95% of all the variance is explained.
2. Discretise the PCA-generated attribute subset into 10 bins, using bins of equal width. For each component X that you discretise, generate a new column in the original dataset named "pcaX_width". For example, the first discretised principal component will correspond to a new column called "pca1_width".
3. Discretise PCA-generated attribute subset into 10 bins, using bins of equal frequency (they should all the same number of points). For each component X that you discretise, generate a new column in the original dataset named "pcaX_freq". For example, the first discretised principal component will correspond to a new column called "pca1_freq".
4. Save the generated dataset to "./output/question3_out.csv".

3.2 Exercise 3 - Answers

The following are the answers to exercise 3 in the first tutorial sheet:

1. Reduce the number of attributes using Principal Component Analysis (PCA), making sure at least 95% of all the variance is explained.
   - This is complete. I wrote a script which adds components until the cumulative amount reaches a cut off and then returns that data. I used SKLearn to do this.
2. Discretise the PCA-generated attribute subset into 10 bins, using bins of equal width. For each component X that you discretise, generate a new column in the original dataset named "pcaX_width". For example, the first discretised principal component will correspond to a new column called "pca1_width".
   - Pandas contains a function qcut and cut to achieve this, and I implemented a method that loops over all PCA columns and adds on the width and frequency using cut and qcut with the desired number of bins.

3. Discretise PCA-generated attribute subset into 10 bins, using bins of equal frequency (they should all the same number of points). For each component X that you discretise, generate a new column in the original dataset named "pcaX_freq". For example, the first discretised principal component will correspond to a new column called "pca1_freq".
    - Pandas contains a function qcut and cut to achieve this, and I implemented a method that loops over all PCA columns and adds on the width and frequency using cut and qcut with the desired number of bins.
4. Save the generated dataset to "./output/question3_out.csv".
    - Done