Data Mining – Preprocessing Report

Background:

Data mining is a process to extract useful information from a vast amount of data. It is used to discover new, accurate and useful patterns in data, looking for meaning.

Data: refers to characteristics /numerical/categorical which are collected through observation.

Datasets: collections of items which are described by a set of attributes.

Data Mining starts with collection or sourcing of raw data. Raw data can have multiple issues like poor data quality such as noisy data, dirty data, missing values, inexact or incorrect values, inadequate data size and poor representation in data sampling.

Issues with Raw data should be resolved before starting with data mining. Raw data issues can be resolved using data cleaning techniques based on the type of discrepancy of data.

Handling Missing data:
- Ignore tuple
- Fill in missing values manually
- Fill in missing values automatically

Handling Noisy data:
- Binning
- Regression
- Outliers
- Semi-supervised

This report covers the Preprocessing of raw data to pre-pare for the next stage of data mining.

Pre-processing of Raw Data:

1. Binning Technique
    a. What is Binning technique and why it is used?

    Data binning is a data pre-processing technique for reducing the cardinality of continuous and discrete data. Binning groups in related values together in bins of either equal width or equal frequency this will then reduce the number of distinct values.

    b. How?

    There are two types :
    - Equal – width
    - Equal – frequency

    Binning Method:
    1. First sort data and partition them into bins of equal frequency .
    2. Then one can smooth the data by bin means, bin median or by bin boundaries.

    Example:
    Q1: Consider the following sales data: [3, 16, 20, 4, 2, 5, 10, 9, 13, 7, 14, 8]. Apply the following binning techniques on the data, assuming 3 bins in each case:

Solution:
Before sorting data: [2, 3 ,4 ,5 ,7 ,8 ,9 ,10 ,13 ,14 ,16 ,20]

Step 1: Sorting of data in ascending order:

New data: [2, 3 ,4 ,5 ,7 ,8 ,9 ,10 ,13 ,14 ,16 ,20]

1. Equal-frequency binning

   Bin 1: [2,3,4,5]

   Bin 2: [7,8,9,10]

   Bin 3: [13,14,16,20]

2. Smoothing by bin boundaries

	Original Bins	Smoothening after bin boundaries
1	[2,3,4,5]	[2,2,5,5]
2	[7,8,9,10]	[7,7,10,10]
3	[13,14,16,20]	[13,13,13,20]

2. Normalization
   a. What is Normalization of data and why?

   The Normalization scales the data of an attribute so that it falls in a smaller range, such as -1.0 to 1.0 or 0.0 to 1.0. It is generally useful for classification algorithms. When multiple attributes have values on different scales, this may lead to poor data models while performing data mining operations. These can be normalized by bring all the attributes on the same scale.

   b. How?

   Q2 Use the below methods to normalize the following data: [10, 5, 25, 50, 35]:

   Solution:

   Before sorting data: [10, 5, 25, 50, 35]
   Step 1: Sorting of data:
   New data: [5, 10, 25, 35, 50]

   1. Min-max normalization with min=0 and max=1.
   Formula:
   $v' = ((v - min_A)/(max_A - min_A)) * (new\_max_A - new\_min_A) + new\_min_A$
   $min_A = 5$, $max_A = 50$, $new\_min_A = 0$, $new\_max_A = 1$
   Normalized data: [0, 0.11, 0.44, 0.67, 1]

   2. Z-score normalization
   Z = (v – Mean)/(Standard deviation)
   Here values of ‘v’ = [5, 10, 25, 35, 50]
   Mean = 25, standard deviation = 16.431676725155, Variance, σ^2: 270
   Z-score normalization values: [-1.21716, -0.91287, 0, 0608581, 1.521452]

3. Chi-Square test
    a. What is Chi-square test and why?

    A chi-square (χ^2) statistic test which is a measure of the difference between the observed and expected frequencies of the outcomes of a set of events or variables. Chi-square is beneficial for analyzing such differences in categorical variables, especially those nominal in nature.

    b. How?

Q3: Students at two universities, University A and University B, have been provided with feedback forms on student satisfaction, with the below responses recorded. Is student satisfaction correlated with a specific university? Use a chi-square test to find out, assuming a significance level of 0.001 and a corresponding chi-square significance value of 10.828. [1 mark out of 5]

Solution:

Observed frequencies:

Rating/University	University A	University B	Total
Satisfied	71	129	200
Dissatisfied	37	73	110
Total	108	202	310

Expected frequencies:
Formula:
$\mathbf{e_{ij} = (count(A = a_i)*count(B = b_j))/n}$
where n= 310

e(ij)1	(108*200)/310	69.67	$x^2(1)$	$(71-69.677)^2/(69.677)$	0.0251206
e(ij)2	(108*110)/310	38.322	$x^2(2)$	$(37-38.322)^2/(38.322)$	0.0456052
e(ij)3	(202*200)/310	130.322	$x^2(3)$	$(129-130.322)^2/(130.322)$	0.0134105
e(ij)4	(202*110)/310	71.67	$x^2(4)$	$(73-71.677)^2/(71.677)$	0.0244196

χ^2 = ∑c i=1 ∑r j=1 (oij – eij) ^{2}/ eij
= (0.0251206 + 0.0456052 + 0.0134105 + 0.0244196)
= 0.1085559

4. Load the CSV file country-income.csv which includes both numerical and categorical attributes. Perform data cleaning in order to replace any NaN values with the mean of the value for a given field. Then replace any categorical labels with numerical labels. Display the resulting dataset. You can use the sklearn. impute and sklearn. preprocessing packages to assist you. [1 mark out of 5]

Solution:

There are two tasks to be attained in the question:

- Replace NaN values with the mean of the table:
  Firstly, we create a new data2 attribute which contains the values of 'Income' and 'Age' columns only from the country-income.csv file. It would be useful to apply the fillna() function with along with mean() function.
- Replacing the categorical labels with numerical labels
  The categorical labels can be labelled with numerical labels by importing the 'LabelEncoder' from sklearn. Preprocessing package

<table>
<tr><th>#CODE</th><th>Output</th></tr>
<tr><td>import pandas as pd
df = pd. read_csv('/country-income.csv')
df</td><td>import pandas as pd
df = pd.read_csv('/country-income.csv')
#df.shape
df

<table>
<tr><th></th><th>Region</th><th>Age</th><th>Income</th><th>Online Shopper</th></tr>
<tr><td>0</td><td>India</td><td>49.0</td><td>86400.0</td><td>No</td></tr>
<tr><td>1</td><td>Brazil</td><td>32.0</td><td>57600.0</td><td>Yes</td></tr>
<tr><td>2</td><td>USA</td><td>35.0</td><td>64800.0</td><td>No</td></tr>
<tr><td>3</td><td>Brazil</td><td>43.0</td><td>73200.0</td><td>No</td></tr>
<tr><td>4</td><td>USA</td><td>45.0</td><td>NaN</td><td>Yes</td></tr>
<tr><td>5</td><td>India</td><td>40.0</td><td>69600.0</td><td>Yes</td></tr>
<tr><td>6</td><td>Brazil</td><td>NaN</td><td>62400.0</td><td>No</td></tr>
<tr><td>7</td><td>India</td><td>53.0</td><td>94800.0</td><td>Yes</td></tr>
<tr><td>8</td><td>USA</td><td>55.0</td><td>99600.0</td><td>No</td></tr>
<tr><td>9</td><td>India</td><td>42.0</td><td>80400.0</td><td>Yes</td></tr>
</table></td></tr>
<tr><td>#Replacing NaN values
data2 = df[['Income','Age']]
print ('Before replacing missing values:')
print(data2)
data2 = data2.fillna(data2.mean())
print ('\n After replacing missing values:')
print(data2)</td><td>Before replacing missing values:
<pre>
    Income   Age
0 86400.0 49.0
1 57600.0 32.0
2 64800.0 35.0
3 73200.0 43.0
4   NaN 45.0
5 69600.0 40.0
6 62400.0   NaN
7 94800.0 53.0
8 99600.0 55.0
9 80400.0 42.0

 After replacing missing values:
          Income   Age
0 86400.000000 49.000000
1 57600.000000 32.000000
2 64800.000000 35.000000
3 73200.000000 43.000000
4 76533.333333 45.000000
5 69600.000000 40.000000
6 62400.000000 43.777778
7 94800.000000 53.000000
8 99600.000000 55.000000
9 80400.000000 42.000000
</pre></td></tr>
</table>

#Categorical labels to numerical labels

```
from sklearn.
preprocessing import LabelEncoder
le = LabelEncoder()
features = df[['Online Shopper','Region']]
features.head()
df['OS'] = le.fit_transform
(features['Online Shopper'])
df['RG'] = le.fit_transform(features['Region'])

df
```

	Region	Age	Income	Online Shopper	OS	RG
0	India	49.0	86400.0	No	0	1
1	Brazil	32.0	57600.0	Yes	1	0
2	USA	35.0	64800.0	No	0	2
3	Brazil	43.0	73200.0	No	0	0
4	USA	45.0	NaN	Yes	1	2
5	India	40.0	69600.0	Yes	1	1
6	Brazil	NaN	62400.0	No	0	0
7	India	53.0	94800.0	Yes	1	1
8	USA	55.0	99600.0	No	0	2
9	India	42.0	80400.0	Yes	1	1

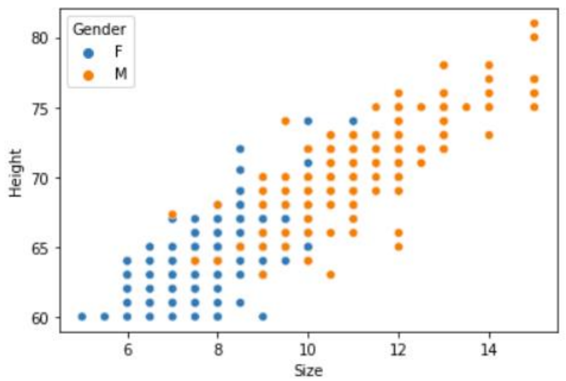
5. Load the CSV file shoesize.csv, which includes measurements of shoe size and height (in inches) for 408 subjects, both female and male. Plot the scatterplots of shoe size versus height for female and male subjects separately. Compute the Pearson's correlation coefficient of shoe size versus height for female and male subjects separately. What can be inferred by the scatterplots and computed correlation coefficients? You can implement your own formulation of the correlation coefficient or use the scipy. stats package to assist you. [1 mark out of 5]

#CODE

```
# for Data visualization
import seaborn as sns
# for Data visualization
import matplotlib. pyplot as plt
import pandas as pd
df = pd. read_csv('./shoesize.csv')
sns.scatterplot(x = "Size", y = "Height",
data = df, hue = "Gender",
hue_order= ['F','M'])
```

Output

```
&lt;matplotlib.axes._subplots.AxesSubplot at 0x7f6cc8b6ab10&gt;
```



There is a positive linear relationship between height and shoe size in this sample. The magnitude of the relationship between the shoe size and height appears to be strong.

Pearson’s correlation formula:

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Correlations: Size for F, Height for F	
SUM((X-Average of female size)*(Y-Average of female height))	526.9465
SUM((X-Average of female size)**2)	354.4438
SUM((Y-Average of female height)**2)	1563.6872
Pearson correlation (Size for F , Height for F)	0.7078
Correlations: Size for M, Height for M	
SUM((X-Average of male size)*(Y-Average of male height))	781.677
SUM((X-Average of male size)**2)	475.1063
SUM((Y-Average of male height)**2)	2812.0817
Pearson correlation (Size for M , Height for M)	0.7677