CPS 844 Lab 2: Data Preprocessing

02/09/2022
Section 1
Ryan Soliven
500840952

Question #2

2) Read the dataset located here 'https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data'

Code

```
data = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data', header=None)
```

Question #3

3) Assign new headers to the DataFrame

Code

```
data.columns = ['Sample code number', 'Clump Thickness', 'Uniformity of Cell Size', 'Uniformity of Cell Shape',
        'Marginal Adhesion', 'Single Epithelial Cell Size', 'Bare Nuclei', 'Bland Chromatin',
        'Normal Nucleoli', 'Mitoses','Class']
```

Question #4

4) Drop the 'Sample code number' attribute

Code

```
data = data.drop(['Sample code number'],axis=1)
```

Missing Values

Question #5

5)Convert the '?' to NaN

Code

```
data = data.replace('?', np.nan)
```

Question #6

6) Count the number of missing values in each attribute of the data.

Code

```
print('Number of missing values:')
for col in data.columns:
    print('\t%s: %d' % (col,data[col].isna().sum()))
```

Question #7

7) Discard the data points that contain missing values

Code

```
data = data.dropna()
```

Outliers

Question #8

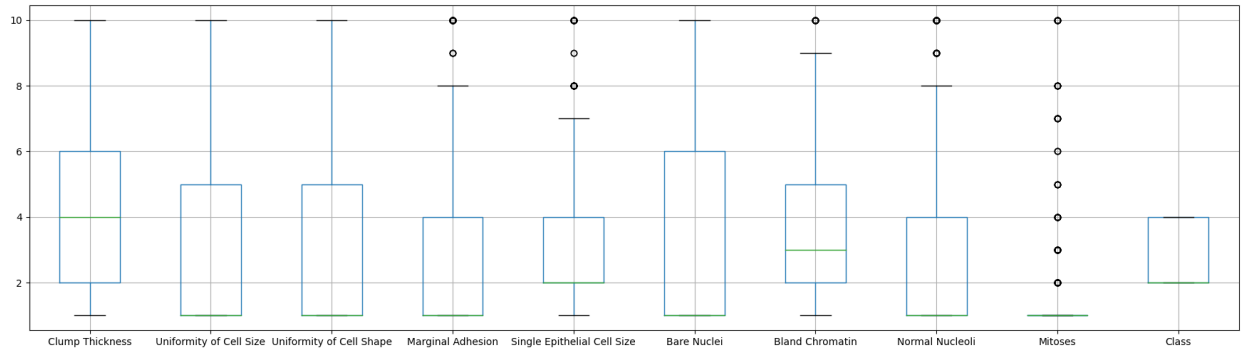
8)Draw a boxplot to identify the columns in the table that contain outliers

Code

```
data['Bare Nuclei'] = data['Bare Nuclei'].astype(str).astype(int)
#need to convert Bare Nuclei column to int in order to draw the boxplot
plot1 = plt.figure(1)
data.boxplot(column = ['Clump Thickness', 'Uniformity of Cell Size', 'Uniformity of Cell
Shape',
                'Marginal Adhesion', 'Single Epithelial Cell Size', 'Bare Nuclei', 'Bland
Chromatin',
               'Normal Nucleoli', 'Mitoses','Class'])
```

Result

The attributes with outliers are: 'Marginal Adhesion', 'Single Epithelial Cell Size', 'Bland Chromatin', ‘Normal Nucleoli', 'Mitoses'



Duplicate Data

Question #9

9) Check for duplicate instances.

Code

```
dups = data.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
```

Result

Number of duplicate rows = 234

Question #10

10) Drop row duplicates

Code

```
print('Number of rows before discarding duplicates = %d' % (data.shape[0]))
data = data.drop_duplicates()
print('Number of rows after discarding duplicates = %d' % (data.shape[0]))
```

Result

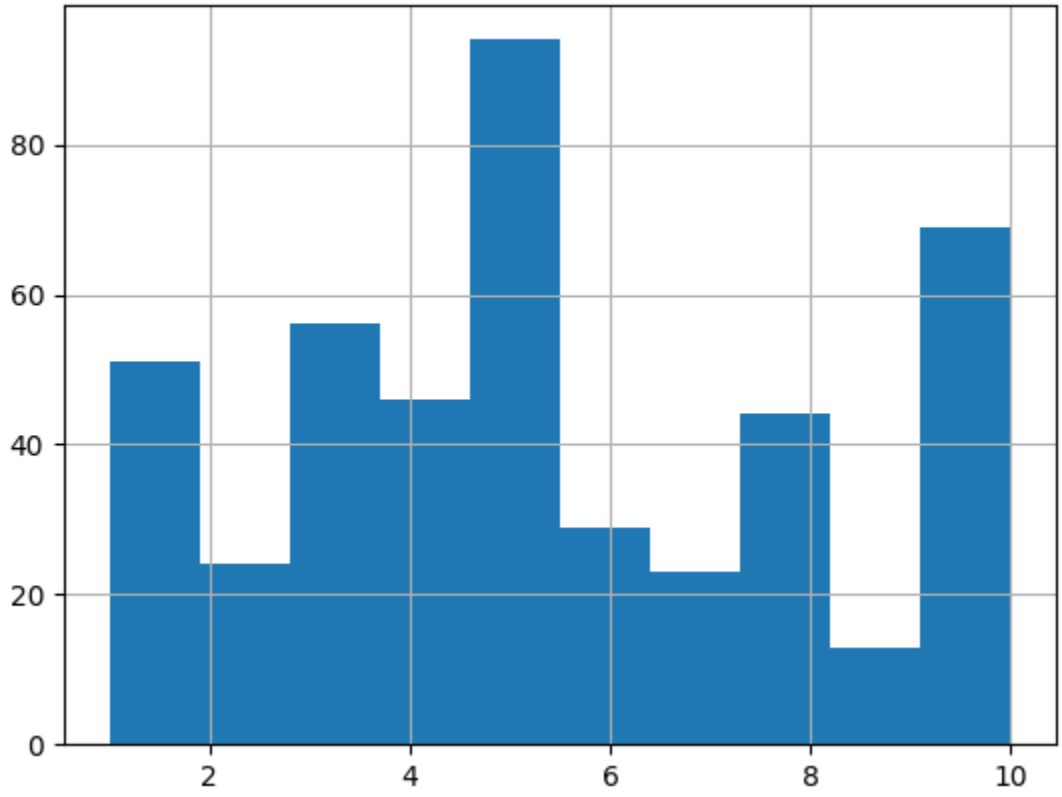
Number of rows before discarding duplicates = 683
Number of rows after discarding duplicates = 449

Discretization

Question #11
11) Plot a 10-bin histogram of the attribute values 'Clump Thickness' distribution

Code
```
plot2 = plt.figure(2)
data['Clump Thickness'].hist(bins=10)
plt.show()
```

Result



Question #12
12)Discretize the ‘Clump Thickness' attribute into 4 bins of equal width.

Code
```
data['Clump Thickness'] = pd.cut(data['Clump Thickness'], 4)
data['Clump Thickness'].value_counts(sort=False)

#print(data['Clump Thickness'].value_counts(sort=False))
```

Result

Range of Values and number of records of each category:

(0.991, 3.25] 131
(3.25, 5.5] 140
(5.5, 7.75] 52
(7.75, 10.0] 126

Sampling

Question #13

13) Randomly select 1% of the data without replacement. The random_state argument of the function specifies the seed value of the random number generator.

Code

```
sample = data.sample(frac=0.01, replace=False, random_state=1)
sample

#print(sample)
```

Result

	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
415	(0.991, 3.25]	3	2	6	3	3	3	5	1	2
400	(0.991, 3.25]	10	8	7	6	9	9	3	8	4
240	(3.25, 5.5]	1	3	3	2	2	2	3	1	2
33	(0.991, 3.25]	1	1	2	2	1	3	1	1	2