Assignment 8

In [1]:
```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import linear_model
from sklearn import tree
from sklearn import naive_bayes
from sklearn.preprocessing import MinMaxScaler
from sklearn.neural_network import MLPClassifier
```

1. Perform Data exploratory analysis on the data (10 points)

In [2]:
```
os.chdir(r"E:\GoogleDriveNew\PSU\DAAN862\Course contents\Lesson 8")
breast_cancer = pd.read_csv("dataR2.csv")
breast_cancer.columns
```

Out[2]:
```
Index(['Age', 'BMI', 'Glucose', 'Insulin', 'HOMA', 'Leptin', 'Adiponectin',
       'Resistin', 'MCP.1', 'Classification'],
      dtype='object')
```

In [3]:
```
breast_cancer.shape
```

Out[3]:
```
(116, 10)
```

In [4]:
```
breast_cancer.isnull().sum()
```

Out[4]:
```
Age               0
BMI               0
Glucose           0
Insulin           0
HOMA              0
Leptin            0
Adiponectin       0
Resistin          0
MCP.1             0
Classification    0
dtype: int64
```

In [5]:
```
breast_cancer.describe()
```

Out[5]:

	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Classification
count	116.000000	116.000000	116.000000	116.000000	116.000000	116.000000	116.000000	116.000000	116.000000	116.000000
mean	57.301724	27.582111	97.793103	10.012086	2.694988	26.615080	10.180874	14.725966	534.647000	1.551724
std	16.112766	5.020136	22.525162	10.067768	3.642043	19.183294	6.843341	12.390646	345.912663	0.499475
min	24.000000	18.370000	60.000000	2.432000	0.467409	4.311000	1.656020	3.210000	45.843000	1.000000
25%	45.000000	22.973205	85.750000	4.359250	0.917966	12.313675	5.474282	6.881763	269.978250	1.000000
50%	56.000000	27.662416	92.000000	5.924500	1.380939	20.271000	8.352692	10.827740	471.322500	2.000000
75%	71.000000	31.241442	102.000000	11.189250	2.857787	37.378300	11.815970	17.755207	700.085000	2.000000
max	89.000000	38.578759	201.000000	58.460000	25.050342	90.280000	38.040000	82.100000	1698.440000	2.000000

In [6]:
```
breast_cancer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 116 entries, 0 to 115
Data columns (total 10 columns):
Age               116 non-null int64
BMI               116 non-null float64
Glucose           116 non-null int64
Insulin           116 non-null float64
HOMA              116 non-null float64
Leptin            116 non-null float64
Adiponectin       116 non-null float64
Resistin          116 non-null float64
MCP.1             116 non-null float64
Classification    116 non-null int64
dtypes: float64(7), int64(3)
memory usage: 9.1 KB
```

In [7]:
```
breast_cancer.corr()
```

Out[7]:

	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Classification
Age	1.000000	0.008530	0.230106	0.032495	0.127033	0.102626	-0.219813	0.002742	0.013462	-0.043555
BMI	0.008530	1.000000	0.138845	0.145295	0.114480	0.569593	-0.302735	0.195350	0.224038	-0.132586
Glucose	0.230106	0.138845	1.000000	0.504653	0.696212	0.305080	-0.122121	0.291327	0.264879	0.384315
Insulin	0.032495	0.145295	0.504653	1.000000	0.932198	0.301462	-0.031296	0.146731	0.174356	0.276804
HOMA	0.127033	0.114480	0.696212	0.932198	1.000000	0.327210	-0.056337	0.231101	0.259529	0.284012
Leptin	0.102626	0.569593	0.305080	0.301462	0.327210	1.000000	-0.095389	0.256234	0.014009	-0.001078
Adiponectin	-0.219813	-0.302735	-0.122121	-0.031296	-0.056337	-0.095389	1.000000	-0.252363	-0.200694	-0.019490
Resistin	0.002742	0.195350	0.291327	0.146731	0.231101	0.256234	-0.252363	1.000000	0.366474	0.227310
MCP.1	0.013462	0.224038	0.264879	0.174356	0.259529	0.014009	-0.200694	0.366474	1.000000	0.091381
Classification	-0.043555	-0.132586	0.384315	0.276804	0.284012	-0.001078	-0.019490	0.227310	0.091381	1.000000

2. Use 30% of data as test set and build a Logistic regression model to predict Labels variable (20 points)

In [8]:
```
X = breast_cancer.iloc[:, :9]
y = breast_cancer.Classification
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 123)
```

In [9]:
```
lr = linear_model.LogisticRegression()
lr.fit(X_train, y_train)
```

Out[9]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

In [10]:
```
y_trian_lr = lr.predict(X_train)

metrics.accuracy_score(y_train, y_trian_lr)
```

Out[10]: 0.8271604938271605

In [11]:
```
y_test_lr = lr.predict(X_test)
metrics.accuracy_score(y_test, y_test_lr)
```

Out[11]: 0.6857142857142857

3. Build the Naïve Bayes model to predict Labels variable (20 points)

In [12]:
```
NB = naive_bayes.GaussianNB()
NB.fit(X_train, y_train)
y_train_NB = NB.predict(X_train)
metrics.accuracy_score(y_train, y_train_NB)
```

Out[12]: 0.6172839506172839

In [13]:
```
y_test_NB = NB.predict(X_test)
metrics.accuracy_score(y_test, y_test_NB)
```

Out[13]: 0.6857142857142857

4. Build the Decision tree model to predict Labels variable (20 points)

In [14]:
```
dt = tree.DecisionTreeClassifier()
dt.fit(X_train, y_train)
y_train_dt = dt.predict(X_train)
metrics.accuracy_score(y_train, y_train_dt)
```

Out[14]: 1.0

In [15]:
```
y_test_dt = dt.predict(X_test)
metrics.accuracy_score(y_test, y_test_dt)
```

Out[15]: 0.5142857142857142

5. Build Neural network model to predict Labels variable (20 points)

In [16]:
```
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [17]:
```
nn = MLPClassifier(hidden_layer_sizes= (10, 5), max_iter=10000)
nn.fit(X_train_scaled, y_train)
y_train_nn = nn.predict(X_train_scaled)
metrics.accuracy_score(y_train, y_train_nn)
```

Out[17]: 0.9629629629629629

In [18]:
```
y_test_nn = nn.predict(X_test_scaled)
metrics.accuracy_score(y_test, y_test_nn)
```

Out[18]: 0.7142857142857143

6. Compare their performance, which one is better? (10 points)

Based on the test accuracy, Neural network model provide the best performance.