Machine Learning Assignment

Classification of Breast Cancer from

**SE4060**

**Bachelor of Science (Honors) in Information Technology**

**Department of Software Engineering**

**Sri Lanka Institute of Information Technology**

**Sri Lanka**

**April 2018**

# Description on the problem addressed

Breast cancer is a type of cancer that is developed in the breast tissues.

# Dataset used

## Description

To create the classification algorithm for breast cancers, we need to find a dataset to base the algorithm on. Breast Cancer Wisconsin (Original) is a breast cancer database posted in UCI machine learning repository. This database is obtained from the University of Wisconsin Hospitals. Data is present in the form of a comma separated values(CSV) file although having the extension txt.

## Features

The dataset contains 11 features and 699 instances having a shape of (11\*699). The attributes and the domain are described as follows.

|  |  |
| --- | --- |
| Attribute | Domain |
| Sample code number | id number |
| Clump Thickness | 1 - 10 |
| Uniformity of Cell Size | 1 - 10 |
| Uniformity of Cell Shape | 1 - 10 |
| Marginal Adhesion | 1 - 10 |
| Single Epithelial Cell Size | 1 - 10 |
| Bare Nuclei | 1 - 10 |
| Bland Chromatin | 1 - 10 |
| Normal Nucleoli | 1 - 10 |
| Mitoses | 1 – 10 |
| Class | (2 - Benign, 4 - Malignant) |

There are two classes in this dataset, if the tumor is benign or malignant. The other features except the Sample code number represent different attributes of the tumor that may be used to classify the tumor as benign or malignant.

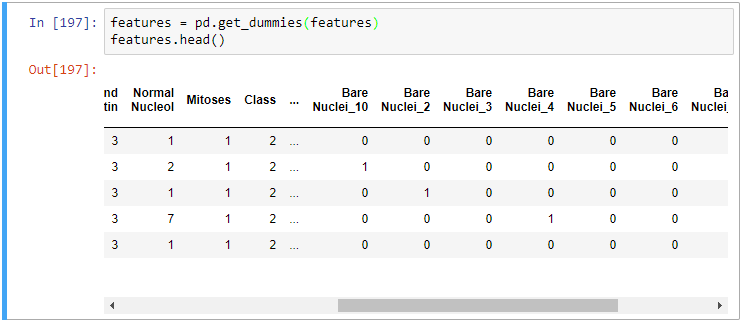
# Methodology (algorithm used)

## Algorithm

For this classification we use **Random Forest** which is an ensemble algorithm. It combines multiple decision trees to classify data. It can help avoid overfitting problem.

## Convert data

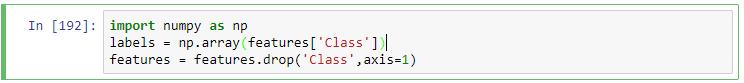
Before we can feed this data to the random forest generator we should convert the categorical data into discrete values. We use pandas to do this. The code given in the following image will convert all categorical data into multiple columns.



For example, the column Bare Nuclei which has a domain of 1-10 will be split into 10 columns Bare Nuclei \_1, Bare Nuclei and so on as shown in the image above. We have 9 categorical columns in our database which all have the same domain of 1-10. This will result in 90 columns.

## Split Label and Features

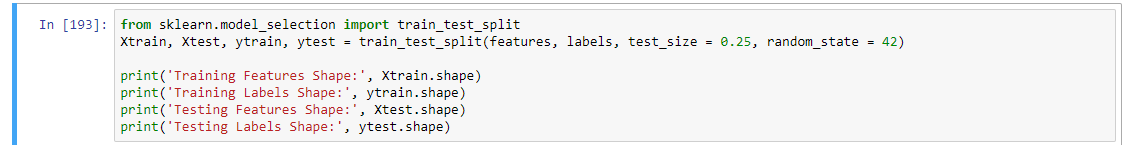
Next, we must divide the labels and features. Labels are what we are trying to predict and features are the attributes that are used to predict them. In this case, we are using all columns in the table except Sample code number to predict the Class (benign or malignant).



The above code will split the Class attribute into the label array and remove it from the features array. Now we have the features and the label separated in two arrays.

## Split Train and Test Data

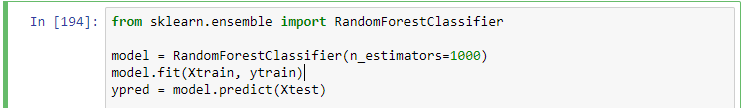
Next, we can split the data into training and test data. The training data will be used to create the random forest classifier model and the training data will be used to test the accuracy of the classifier.



We use sklearn to split the data. Here we specify the test size as 0.25 so 25% of the data is split into test and the remaining 75% is used for training the model.

## Generate the classifier

RandomForestClassifier is used to generate a model as shown below. the parameter **n\_estimators** indicate the number of decision trees in the forest. We then fit the model using the label and class of the training data set.

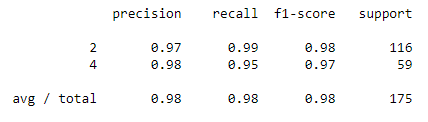


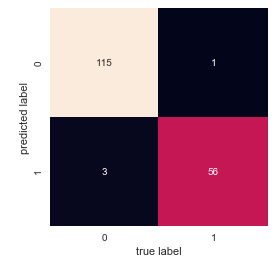
Now we have a model, next we can predict the validity of the model by using it against the test data.

## Predict the test data

With the model we have created using 75% of the data, we can use that to predict the class of the tumor represented in the test data.

# Results and discussion (including possible limitations/future work)





# Appendix

import pandas as pd

features = pd.read\_csv('./data/breast-cancer-wisconsin.data.txt',names = ["Sample Code number", "Clump Thickness", "Uniformity of Cell Size", "Unfiformity of Cell Shape", "Marginal Adhesion", "Single Epithelial Cell Size", "Bare Nuclei", "Bland Chromatin", "Normal Nucleol", "Mitoses", "Class"])

features.head()

features = pd.get\_dummies(features)

features.head()

import numpy as np

labels = np.array(features['Class'])

features = features.drop('Class',axis=1)

from sklearn.model\_selection import train\_test\_split

Xtrain, Xtest, ytrain, ytest = train\_test\_split(features, labels, test\_size = 0.25, random\_state = 42)

print('Training Features Shape:', Xtrain.shape)

print('Training Labels Shape:', ytrain.shape)

print('Testing Features Shape:', Xtest.shape)

print('Testing Labels Shape:', ytest.shape)

from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n\_estimators=1000)

model.fit(Xtrain, ytrain)

ypred = model.predict(Xtest)

from sklearn import metrics

print(metrics.classification\_report(ypred, ytest))

from sklearn.metrics import confusion\_matrix

import seaborn as sns; sns.set()

%matplotlib inline

import matplotlib.pyplot as plt

mat = confusion\_matrix(ytest, ypred)

sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False)

plt.xlabel('true label')

plt.ylabel('predicted label');