Requirements

- Machine Learning Understanding
- Python 3.5+
- Python libraries : Scikit Learn, Pandas, Numpy

1 Introduction

This is the beginner project 1 for the ML Course. The aim of the project is to predict whether Cancer is Benign or Malignant based on a set of real valued features. In essence, this is a Binary classification problem.

2 Description

Cancer is one of the most dreaded disease in living beings. There are two types of cancers depending on how they spread : **Benign and Malignant**. Thus with the necessary features, this can be seen as a binary classification problem. The original Wisconsin-Breast Cancer (Diagnostics) dataset (WBC) from UCI machine learning repository is a classification dataset, which records the measurements for breast cancer cases.

The dataset can be downloaded from https://www.kaggle.com/uciml/breast-cancer-wisconsin-data. A simpler way to use the dataset is to import it through **scikit-learn** using

```
from sklearn.datasets import load_breast_cancer
```

3 Data set Description

The attributes of the dataset are:

- ID number
- Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension (”coastline approximation- 1)

Thus we can see that the **Diagnosis** is the target. All the rest are features.

4 Problem Statement

Given the features and the target, build a **Machine Learning Classification model** that can **classify** from a given set of features, if the cancer is Benign or Malignant.
You can use any classification algorithm. Please make sure you answer the following are in your report with the code:

- Briefly explain the algorithm.
- What was the accuracy?
- Build and Understand items of **Classification Report** using scikit learn
- Observe and note changes in accuracy as you **vary parameters**. Ex. Number of Nodes in Random Forest classifier.

Optional: An interesting extra task that can be done after you are able to classify is **compare the performance** on different algorithms. Also looking into **Exploratory data Analysis** will be useful.

5 Report

A simple report containing the following can be submitted:

- Name, SRN, Section
- Answers to the questions above
- Code : Direct python or jupyter notebook screenshots
- Results (Can be tables or Graphs or simple sentences)

6 Contact for Doubts:

- Dr. B N Krupa : bnkrupa@pes.edu
- venkatramnank@pesu.pes.edu