Delhi Technological University

Computer Science and Engineering Department

Modelling and Simulation (SE-207)

Mid-Term Examination Project

Project Information

Title: Breast Cancer Detection

Faculty: Mr Rahul, Assistant Professor
Computer Science and Engineering Department
Delhi Technological University
rahul@dtu.ac.in

Prepared By: Anushka Sethi (2K19/SE/015) &
Aseem Sangalay (2K19/SE/021)

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We would like to take this opportunity to express our profound gratitude and deep regards to our teacher **Prof. Rahul Chandra** for his exemplary guidance, monitoring and constant encouragement, throughout the course of this project. The blessing help and guidance given by him from time to time shall carry us a long way in the journey of the life on which we are about to embark.

OBJECTIVES

This project aims to:

- To observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in the model selection.
- To achieve this we are simulating the real-time model using machine learning techniques to fit a function that can predict the discrete class of new input.
- Evaluate and interpret the results and justify the interpretation based on the observed and available data set.
- Create notebooks that serve as computational records and document our thought process.

The analysis is divided into four sections, saved in jupyter notebooks

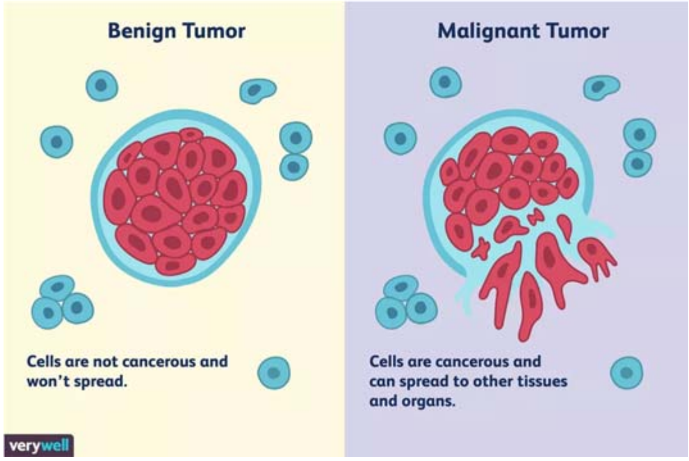
1. Identifying the problem and Data Sources
2. Exploratory Data Analysis
3. Pre-Processing the Data
4. Build a model to predict whether breast cell tissue is malignant or Benign

1. OVERVIEW

1.1 Introduction

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today’s society.

Breast cancer occurs when a *malignant (cancerous)* tumour originates in the breast. As breast cancer tumours mature, they may metastasize (spread) to other parts of the body. The primary route of metastasis is the lymphatic system which, ironically enough, is also the body's primary system for producing and transporting white blood cells and other cancer-fighting immune system cells throughout the body. Metastasized cancer cells that aren't destroyed by the lymphatic system's white blood cells move through the lymphatic vessels and settle in remote body locations, forming new tumours and perpetuating the disease process.


Benign Tumor
Malignant Tumor
Cells are not cancerous and won’t spread.
Cells are cancerous and can spread to other tissues and organs.
verywell

The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of *benign tumours* can prevent patients from undergoing unnecessary treatments. Thus, the correct diagnosis of BC and the classification of patients into malignant or benign groups is the subject of much research. Because of its unique advantages in critical features detection from complex BC datasets, machine learning (ML) is widely recognized as the methodology of choice in BC pattern classification and forecast modelling.

Classification and *data mining methods* are an effective way to classify data. Especially in the medical field, where those methods are widely used in diagnosis and analysis to make decisions.

1.1.1 Recommended Screening Guidelines:

Mammography. The most important screening test for breast cancer is the mammogram. A mammogram is an X-ray of the breast. It can detect breast cancer up to two years before the tumour can be felt by you or your doctor. *Women age 40–45 or older* who are at average risk of breast cancer should have a mammogram once a year. *Women at high risk* should have yearly mammograms along with an MRI starting at age 30.

1.1.2 Some Risk Factors for Breast Cancer

The following are some of the known risk factors for breast cancer. However, most cases of breast cancer cannot be linked to a specific cause. Talk to your doctor about your specific risk.

Age: The chance of getting breast cancer increases as women age. Nearly 80 percent of breast cancers are found in women over the age of 50.

Personal history of breast cancer: A woman who has had breast cancer in one breast is at an increased risk of developing cancer in her other breast.

Family history of breast cancer: A woman has a higher risk of breast cancer if her mother, sister or daughter had breast cancer, especially at a young age (before 40). Having other relatives with breast cancer may also raise the risk.

Genetic factors: Women with certain genetic mutations, including changes to the BRCA1 and BRCA2 genes, are at higher risk of developing breast cancer during their lifetime. Other gene changes may raise breast cancer risk as well.

Childbearing and menstrual history: The older a woman is when she has her first child, the greater her risk of breast cancer. Also at higher risk are:

- Women who menstruate for the first time at an early age (before 12)
- Women who go through menopause late (after age 55)
- Women who've never had children

1.2 Role of Machine Learning in the Detection of Breast Cancer

A *mammogram* is an x-ray picture of the breast. It can be used to check for breast cancer in women who have no signs or symptoms of the disease. It can also be used if you have a lump or other sign of breast cancer.

Screening mammography is the type of mammogram that checks you when you have no symptoms. It can help reduce the number of deaths from breast cancer among women ages 40 to 70 but it can also have drawbacks. Mammograms can sometimes find something that looks abnormal but isn't cancer. This leads to further testing and can cause you anxiety. Sometimes mammograms can miss cancer when it is there. It also exposes you to radiation. You should talk to your doctor about the benefits and drawbacks of mammograms. Together, you can decide when to start and how often to have a mammogram.

Now while it's difficult to figure out for physicians by seeing only images of x-ray that whether the tumour is toxic or not, training a *machine learning model* according to the identification of tumour can be of great help.

1.3 Aim and Identification of the problem in the project

The objective of this report is to train machine learning models to predict whether a breast cancer cell is Benign or Malignant. Data will be transformed and its dimension reduced to reveal patterns in the dataset and create a more robust analysis. As previously said, the optimal model will be selected following the *resulting accuracy*, *sensitivity*, and *f1 score*, amongst other factors. We will later define these metrics. We can use machine learning methods to extract the features of cancer cell nuclei image and classify them. It would be helpful to determine whether a given sample appears to be *Benign ("B")* or *Malignant ("M")*. The machine learning models that we will apply in this project create a classifier that provides a high accuracy level combined with a low rate of false-negatives (high sensitivity).

2. METHODS AND ANALYSIS

2.1 Data Preparation

The project covers the *Breast Cancer Wisconsin (Diagnostic) DataSet* (http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29)
Using python language and libraries such as *Numpy*, *Pandas*, *Sci-kit learn*, we have prepared the dataset that we will use for this project.

The Breast Cancer dataset is an available machine learning repository maintained by the University of California, Irvine. The dataset contains **569 samples of malignant and benign tumour cells**.

- The first two columns in the dataset store the unique ID numbers of the samples and the corresponding diagnosis (M=malignant, B=benign), respectively.
- The columns 3-32 contain 30 real-value features that have been computed from digitized images of the cell nuclei, which can be used to build a model to predict whether a tumour is benign or malignant.

2.1.1 Load libraries and dataset

Load the supplied CSV file using additional options in the Pandas read_csv function.

2.1.2 Inspecting the data

The first step is to visually inspect the new data set. There are multiple ways to achieve this:

- The easiest being to request the first few records using the DataFrame data.head()* method. By default, “data.head()” returns the first 5 rows from the DataFrame object df (excluding the header row).
- Alternatively, one can also use “df.tail()” to return the five rows of the data frame.
- For both head and tail methods, there is an option to specify the number of records by including the required number in between the parentheses when calling either method.
- The **“info()”** method provides a concise summary of the data; from the output, it provides the type of data in each column, the number of non-null values in each column, and how much memory the data frame is using.
- The method **get_dtype_counts()** will return the number of columns of each type in a DataFrame
- Diagnosis is a categorical variable because it represents a fixed number of possible values (i.e, Malignant, of Benign. The machine learning algorithm wants numbers, and no strings, as their inputs so we need some method of coding to convert them.

2.2 Exploratory Data Analysis

After having a good intuitive sense of the data, the next step involves taking a closer look at attributes and data values. In this section, we get familiar with the data, which will provide useful knowledge for data pre-processing.

2.2.1 Objectives of Data Exploration

Exploratory data analysis (EDA) is a very important step which takes place after feature engineering and acquiring data and it should be done before any modelling. This is because it is very important for a data scientist to be able to understand the nature of the data without making assumptions. The results of data exploration can be extremely useful in grasping the structure of the data, the distribution of the values, and the presence of extreme values and interrelationships within the data set.

The purpose of EDA is:

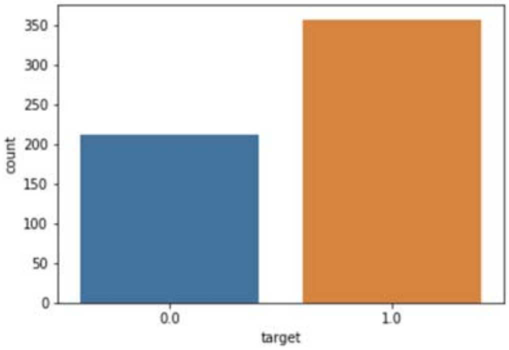
- To use summary statistics and visualizations to better understand data, find clues about the tendencies of the data, its quality and to formulate assumptions and the hypothesis of our analysis
- For data preprocessing to be successful, it is essential to have an overall picture of your data, basic statistical descriptions can be used to identify properties of the data and highlight which data values should be treated as noise or outliers.

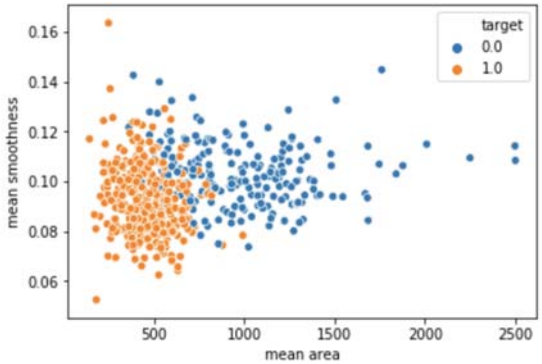
Next step is to explore the data. There are two approaches used to examine the data using:

1. ***Descriptive statistics*** is the process of condensing key characteristics of the data set into simple numeric metrics. Some of the common metrics used are mean, standard deviation, and correlation.
2. ***Visualization*** is the process of projecting the data, or parts of it, into Cartesian space or into abstract images. In the data mining process, data exploration is leveraged in many different steps including preprocessing, modelling, and interpretation of results.

2.3 Unimodal Data Visualizations

One of the main goals of visualizing the data here is to observe which features are most helpful in predicting malignant or benign cancer. The other is to see general trends that may aid us in model selection and hyperparameter selection.





2.4 Multimodal Data Visualizations

The Multimodal Data Visualizations consists of

- Correlation Matrix
- Scatter Plots

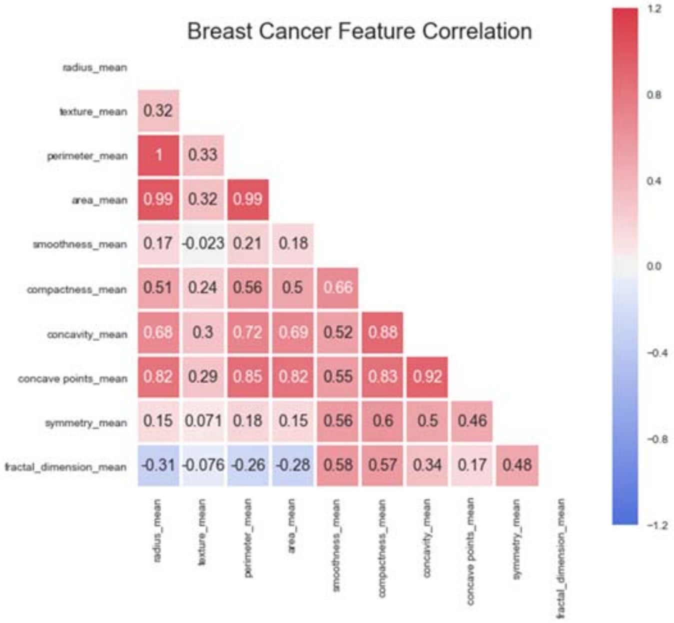


Figure: Correlation Matrix

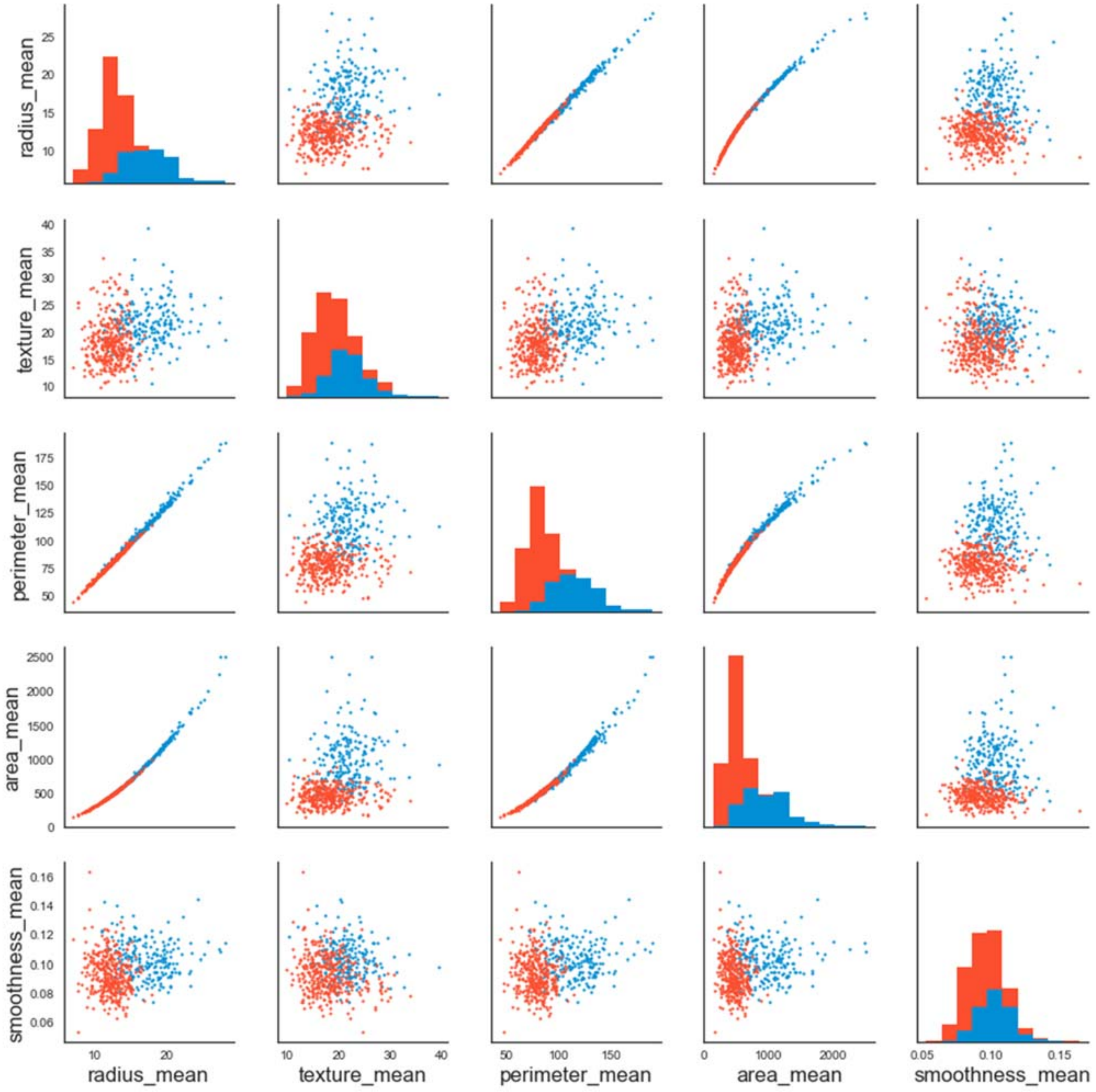


Figure: Scatter Plots

3. PRE-PROCESSING THE DATA

3.1 Introduction

Data preprocessing is a crucial step for any data analysis problem. It is often a very good idea to prepare your data in such way to best expose the structure of the problem to the machine learning algorithms that you intend to use.This involves a number of activities such as:

- Assigning numerical values to categorical data;
- Handling missing values; and
- Normalizing the features (so that features on small scales do not dominate when fitting a model to the data).

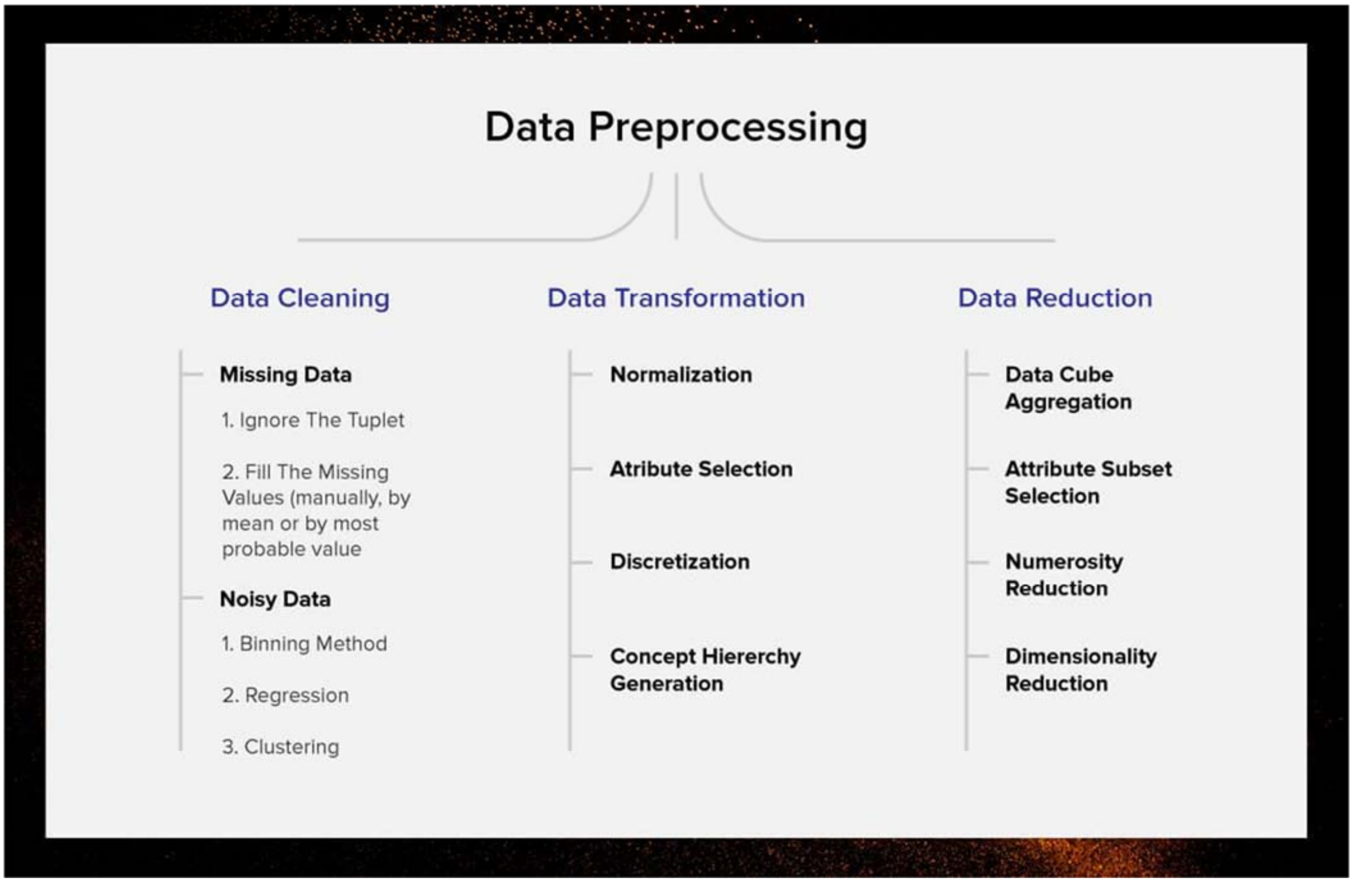
Earlier we explored the data, to help gain insight on the distribution of the data as well as how the attributes correlate to each other. We identified some features of interest. Here, we use feature selection to reduce high-dimension data, feature extraction and transformation for dimensionality reduction.

Why preprocessing?

1. Real world data are generally
   - Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
   - Noisy: containing errors or outliers
   - Inconsistent: containing discrepancies in codes or names

2. Tasks in data preprocessing

   - *Data cleaning*: fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
   - *Data integration*: using multiple databases, data cubes, or files.
   - *Data transformation*: normalization and aggregation.
   - *Data reduction*: reducing the volume but producing the same or similar analytical results.
   - *Data discretization*: part of data reduction, replacing numerical attributes with nominal ones.



Data Preprocessing

Data Cleaning

- Missing Data
  1. Ignore The Tuplet
  2. Fill The Missing Values (manually, by mean or by most probable value
- Noisy Data
  1. Binning Method
  2. Regression
  3. Clustering

Data Transformation

- Normalization
- Atribute Selection
- Discretization
- Concept Hiererchy Generation

Data Reduction

- Data Cube Aggregation
- Attribute Subset Selection
- Numerosity Reduction
- Dimensionality Reduction

Goal:

Find the most predictive features of the data and filter it so it will enhance the predictive power of the analytics model.

3.2 Split data into training and test sets

The simplest method to evaluate the performance of a machine learning algorithm is to use different training and testing datasets. Here we:

- Split the available data into a training set and a testing set. (70% training, 30% test)
- Train the algorithm on the first part,
- make predictions on the second part and
- evaluate the predictions against the expected results.

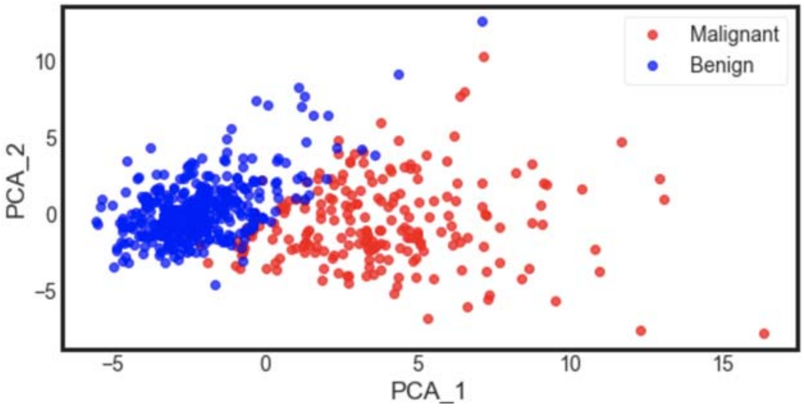
3.3 Feature Standardization

- Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.
- The raw data has differing distributions which may have an impact on the most ML algorithms. Most machine learning and optimization algorithms behave much better if features are on the same scale.

We used sklearn to scale and transform the data such that each attribute has a mean value of zero and a standard deviation of one.

3.4 Feature decomposition using Principal Component Analysis(PCA)

From the pair plot obtained earlier, lots of feature pairs divide nicely the data to a similar extent, therefore, it makes sense to use one of the dimensionality reduction methods to try to use as many features as possible and maintain as much information as possible when working with only 2 dimensions. Therefore, we used PCA.



4. MODEL TRAINING

4.1 Brief introduction ML Algorithms

4.1.1 Machine learning algorithm classification

According to the way of learning, machine learning mainly includes:

- ***Supervised learning*****:** Supervised learning establishes a learning process, compares the predicted results with the actual results of the "training data" (ie, input data), and continuously adjusts the predictive model until the predicted results of the model reach an expected accuracy, such as classification and regression problems.

- ***Unsupervised learning*****:** Input data has no tags, but algorithms to infer the intrinsic links of data, such as clustering and association rule learning.

- ***Semi-supervised learning*****:** Input data part tags, is an extension of supervised learning, often used for classification and regression

- ***Reinforcement learning*****:** Input data as feedback to the model, emphasizing how to act based on the environment to maximize the expected benefits.

4.1.2 Some of the algo to be used in project:

1. **Linear Regression:** For statistical technique linear regression is used in which value of dependent variable is predicted through independent variables. A relationship is formed by mapping the dependent and independent variable on a line and that line is called regression line which is represented by $Y= a*X + b,$ where

   Y= Dependent variable (e.g weight)

   X= Independent Variable (e.g height)

   b= Intercept and a = slope

2. **Logistic Regression:** In logistic regression we have a lot of data whose classification is done by building an equation. This method is used to find the discrete dependent variable from the set of independent variables. Its goal is to find the best fit set of parameters. In this classifier, each feature is multiplied by a weight and then all are added. Then the result is passed to the sigmoid function which produces the binary output. Logistic regression generates the coefficients to predict a logit transformation of the probability.

3. **Support vector machine:** Support vector machine is a binary classifier. Raw data is drawn on the n- dimensional plane. In this a separating hyperplane is drawn to differentiate the datasets.

The line drawn from the centre of the line separating the two closest data-points of different categories is taken as an optimal hyperplane. This optimised separating hyperplane maximizes the margin of training data. Through this hyperplane, new data can be categorised.

4. **Naive-Bayes:** It is a technique for constructing classifiers which is based on Bayes theorem used even for highly sophisticated classification methods. It learns the probability of an object with certain features belonging to a particular group or class. In short, it is a probabilistic classifier. In this method occurrence of each feature is independent of occurrence of another feature. It only needs a small amount of training data for classification, and all terms can be precomputed thus classifying becomes easy, quick and efficient.

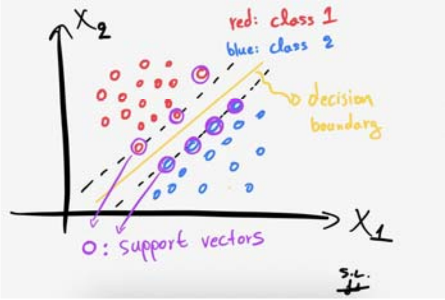
5. **KNN:** This method is used for both classification and regression. It is among the simplest methods of machine learning algorithms. It stores the cases and for new data it checks the majority of the k neighbours with which it resembles the most. KNN makes predictions using the training dataset directly.

4.2 Predictive model using *Support Vector Machine (SVM)*

Support vector machines (SVMs) learning algorithm will be used to build the predictive model. SVMs are one of the most popular classification algorithms, and have an elegant way of transforming nonlinear data so that one can use a linear algorithm to fit a linear model to the data.
Kernelized support vector machines are powerful models and perform well on a variety of datasets.

1. SVMs allow for complex decision boundaries, even if the data has only a few features.
2. They work well on low-dimensional and high-dimensional data (i.e., few and many features), but don’t scale very well with the number of samples.
3. SVMs require careful preprocessing of the data and tuning of the parameters. This is why, these days, most people instead use tree-based models such as *random forests* or *gradient boosting* (which require little or no preprocessing) in many applications.
4. SVM models are hard to inspect; it can be difficult to understand why a particular prediction was made, and it might be tricky to explain the model to a non-expert.

Figure: Support Vector Machines Algorithm Representation



4.2.1 Important Parameters

The important parameters in kernel SVMs are the:

- Regularization parameter C,
- The choice of the kernel,(linear, radial basis function(RBF) or polynomial)
- Kernel-specific parameters.

gamma and C both control the complexity of the model, with large values in either resulting in a more complex model. Therefore, good settings for the two parameters are usually strongly correlated, and C and gamma should be adjusted together.

4.3 Classification with cross-validation

As discussed earlier, splitting the data into test and training sets is crucial to avoid overfitting. This allows generalization of real, previously-unseen data. Cross-validation extends this idea further. Instead of having a single train/test split, we specify *so-called folds* so that the data is divided into similarly-sized folds.

- Training occurs by taking all folds except one – referred to as the holdout sample.
- On the completion of the training, you test the performance of your fitted model using the holdout sample.
- The holdout sample is then thrown back with the rest of the other folds, and a different fold is pulled out as the new holdout sample.
- Training is repeated again with the remaining folds and we measure performance using the holdout sample. This process is repeated until each fold has had a chance to be a test or holdout sample.
- The expected performance of the classifier, called cross-validation error, is then simply an average of error rates computed on each holdout sample.

This process is demonstrated by first performing a standard train/test split, and then computing cross-validation error which came out to a *“classifier accuracy score” of 0.95.*

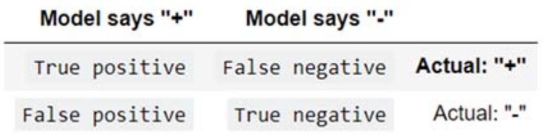
To get a better measure of prediction accuracy, you can successively split the data into folds that you will use for training and testing. *The 3-fold cross-validation accuracy score for this classifier is 0.97.*

4.4 Model Accuracy: Receiver Operating Characteristic (ROC) curve

In statistical modeling and machine learning, a commonly-reported performance measure of model accuracy for binary classification problems is Area Under the Curve (AUC).

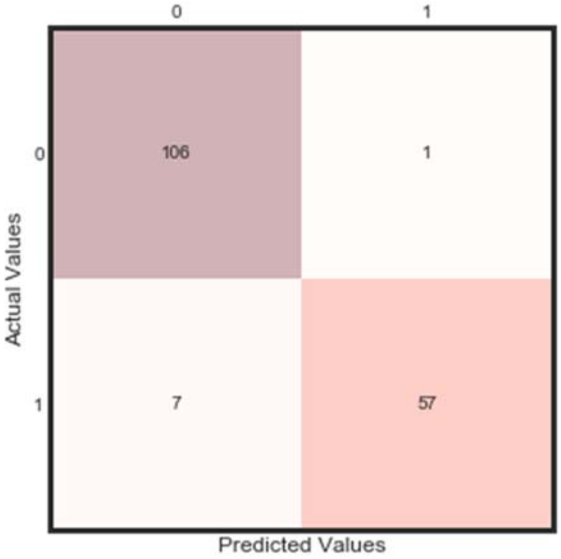
To understand what information the ROC curve conveys, consider the so-called confusion matrix that essentially is a two-dimensional table where the classifier model is on one axis (vertical), and ground truth is on the other (horizontal) axis, as shown below. Either of these axes can take two values (as depicted)

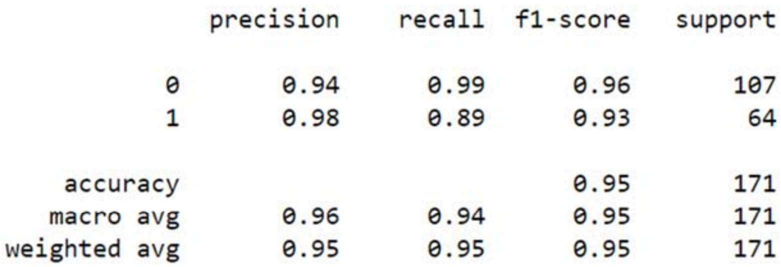
In an ROC curve, we plot “True Positive Rate” on the Y-axis and “False Positive Rate” on the X-axis, where the values “true positive”, “false negative”, “false positive”, and “true negative” are events (or their probabilities) as described above. The rates are defined according to the following:



Model says "+"	Model says "-"	
True positive	False negative	**Actual: "+"**
False positive	True negative	Actual: "-"

- True positive rate (or sensitivity)}: tpr = tp / (tp + fn)
- False positive rate: fpr = fp / (fp + tn)
- True negative rate (or specificity): tnr = tn / (fp + tn)-





```
              precision    recall  f1-score   support

           0       0.94      0.99      0.96       107
           1       0.98      0.89      0.93        64

    accuracy                           0.95       171
   macro avg       0.96      0.94      0.95       171
weighted avg       0.95      0.95      0.95       171
```

Here is the explanation of terminologies used:

- *Precision* - It is define the ability of the classifier not to label as positive a sample that is negative
- *Recall* - It is defined as the ability of the classifier to find all the positive samples.
- *f1-score* - The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative 25 contributions of precision and recall to the F1 score are equal. The formula used is as follows : F1 = 2 * (precision * recall) / (precision + recall)
- *Support* - Support is the number of actual occurrences of the class in the specified dataset.
- *Accuracy* - It gives the fraction of correct scores.

- *macro avg* - It simply calculates the mean of the binary metrics, giving equal weight to each class.
- *weighted avg* - It accounts for class imbalance by computing the average of binary metrics in which each class's score is weighted by its presence in the true data sample.

Observation

There are two possible predicted classes: "1" and "0". Malignant = 1 (indicates presence of cancer cells) and Benign = 0 (indicates absence).

- The classifier made a total of 174 predictions (i.e 174 patients were being tested for the presence of breast cancer).
- Out of those 174 cases, the classifier predicted "yes" 58 times, and "no" 113 times.
- In reality, 64 patients in the sample have the disease, and 107 patients do not.

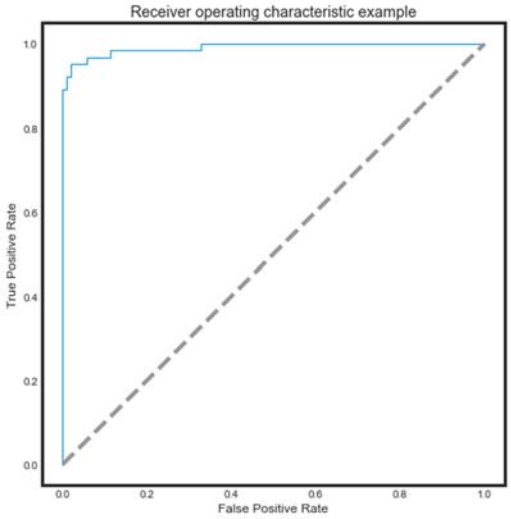


Figure: Example of ROC curve

5. MODEL EVALUATION AND PREDICTIONS

5.1 Optimizing the SVM Classifier

Machine learning models are parameterized so that their behavior can be tuned for a given problem. Models can have many parameters and finding the best combination of parameters can be treated as a search problem. Here, we aim to tune parameters of the SVM Classification model using scikit-learn.

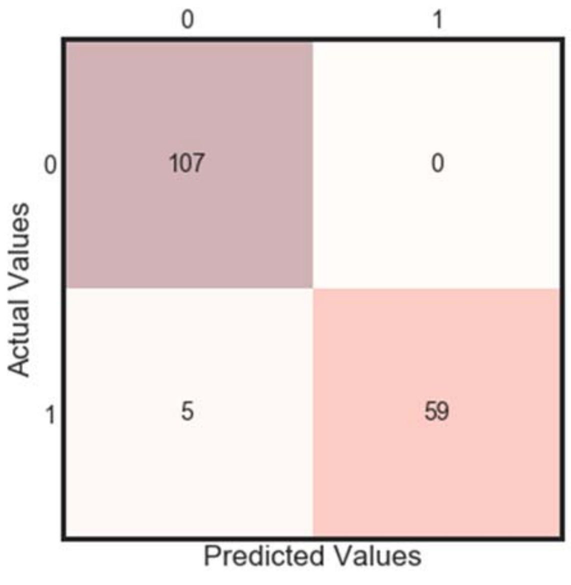
5.1.1 Importance of optimizing a classifier

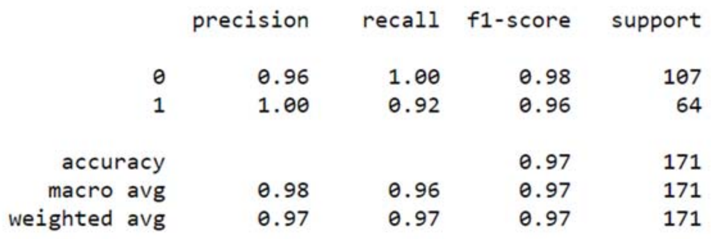
We can tune two key parameters of the SVM algorithm:

- the value of C (how much to relax the margin)
- and the type of kernel.

Python scikit-learn provides two simple methods for algorithm parameter tuning:

- Grid Search Parameter Tuning.
- Random Search Parameter Tuning.



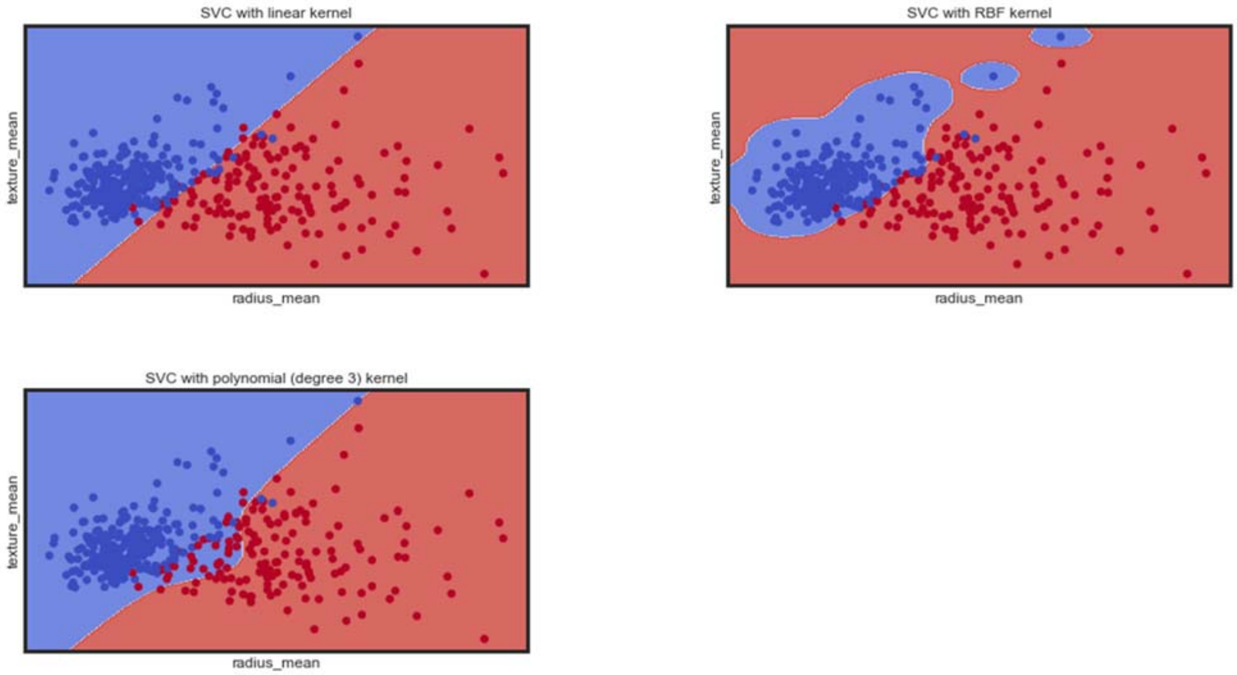


```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       107
           1       1.00      0.92      0.96        64

    accuracy                           0.97       171
   macro avg       0.98      0.96      0.97       171
weighted avg       0.97      0.97      0.97       171
```

Figure: Improvement of metrics after optimization of the classifier

5.1.2 Decision boundaries of different classifiers

Let's see the decision boundaries produced by the linear, Gaussian and polynomial classifiers.



The SVM performs better when the dataset is standardized so that all attributes have a mean value of zero and a standard deviation of one. We can calculate this from the entire training dataset and apply the same transform to the input attributes from the validation dataset.

6. AUTOMATE THE ML PROCESS USING PIPELINES

There are standard workflows in a machine learning project that can be automated. In Python scikit-learn, Pipelines help to clearly define and automate these workflows.

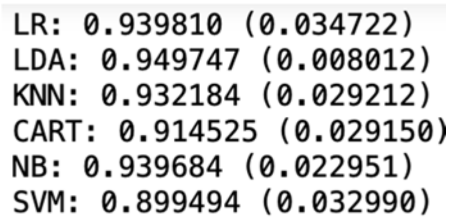
- Pipelines help overcome common problems like data leakage in your test harness.
- Python scikit-learn provides a Pipeline utility to help automate machine learning workflows.
- Pipelines work by allowing for a linear sequence of data transforms to be chained together culminating in a modeling process that can be evaluated.

6.1 Evaluate Some Algorithms

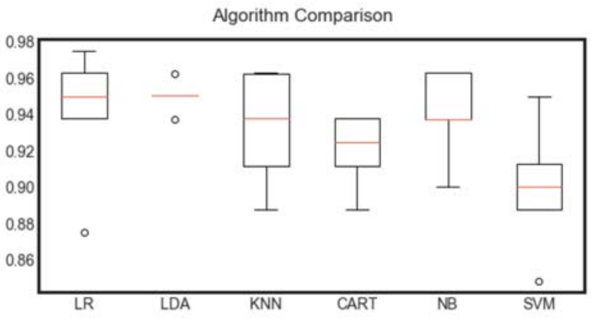
Now we create some models of the data and estimate their accuracy on unseen data. Here is what we are going to cover in this step:

1. Separate out a validation dataset.
2. Setup the test harness to use 10-fold cross validation.
3. Build 5 different models
4. Select the best model

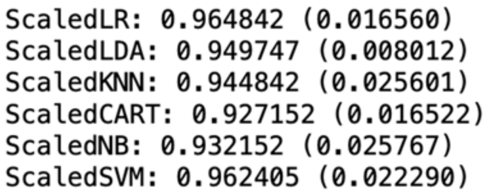
The results suggest That both Logistic Regression and LDA may be worth further study. These are just mean accuracy values. It is always wise to look at the distribution of accuracy values calculated across cross validation folds. We can do that graphically using box and whisker plots.



```
LR: 0.939810 (0.034722)
LDA: 0.949747 (0.008012)
KNN: 0.932184 (0.029212)
CART: 0.914525 (0.029150)
NB: 0.939684 (0.022951)
SVM: 0.899494 (0.032990)
```



After standardisation of data



```
ScaledLR: 0.964842 (0.016560)
ScaledLDA: 0.949747 (0.008012)
ScaledKNN: 0.944842 (0.025601)
ScaledCART: 0.927152 (0.016522)
ScaledNB: 0.932152 (0.025767)
ScaledSVM: 0.962405 (0.022290)
```

The results show that standardization of the data has lifted the skill of SVM to be the most accurate algorithm tested so far.

6.2 Algorithm Tuning

Tuning hyper-parameters - SVC estimator

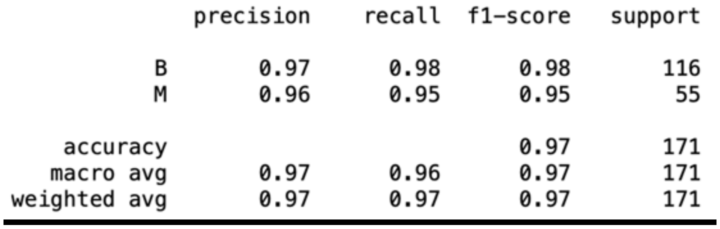
- Model Training Accuracy: 0.940 +/- 0.034
- Tuned Parameters Best Score: 0.9446794871794871
- Best Parameters:
  {'clf__C': 1.0, 'clf__kernel': 'linear'}

Tuning the hyper-parameters - k-NN hyperparameters

- Model Training Accuracy: 0.927 +/- 0.044
- Tuned Parameters Best Score: 0.9396153846153847
- Best Parameters:
  {'clf__n_neighbors': 19}

Final Model:

- Final Model Training Accuracy: 0.945 +/- 0.041
- Final Accuracy on Test set: 0.97076



```
              precision    recall  f1-score   support

           B       0.97      0.98      0.98       116
           M       0.96      0.95      0.95        55

    accuracy                           0.97       171
   macro avg       0.97      0.96      0.97       171
weighted avg       0.97      0.97      0.97       171
```

7. PROJECT CODE

Project Code attached separately.

CONCLUSION AND SUMMARY

Worked through a classification predictive modeling machine learning problem from end-to-end using Python. Specifically, the steps covered were:

1. Problem Definition (Breast Cancer data).
2. Loading the Dataset.
3. Analyze Data (same scale but different distributions of data).
    - Evaluate Algorithms (KNN looked good).
    - Evaluate Algorithms with Standardization (KNN and SVM looked good).
4. Algorithm Tuning (K=19 for KNN was good, SVM with an RBF kernel and C=100 was best)..
5. Finalize Model (use all training data and confirm using validation dataset)

REFERENCES

- https://www.verywellhealth.com/what-does-malignant-and-benign-mean-514240
- https://towardsdatascience.com/a-tour-of-the-top-10-algorithms-for-machine-learning-newbies-dde4edffae11
- https://machinelearningmastery.com/start-here/#algorithms