# Udacity - Machine Learning Engineer Nanodegree Program

## Capstone Project

Murilo Venturin
March 10th, 2020

# 1. Proposal

## 1.2. Domain Background

Breast cancer is a major cause of mortality in women worldwide. Thus, considering the statistics are not promising and the fact that the effectiveness of therapies used against breast cancer is limited. Most cases are diagnosed in late stages. It is emphasized that the chances of a cure are increased the sooner the disease is diagnosed.

The project's proposal is to build a model capable of classifying breast cancer between benign and malignant based on tumor data, such as radius, texture, perimeter, among other features.

## 1.3. Problem Statement

Conventional methods of monitoring and diagnosing diseases depend on the detection of the presence of particular signs characteristics by a human observer. Due to the large number of patients in intensive care units and the need for continuous observation, this work ends up taking a lot of time from health professionals.

An automated diagnostic approach can solve this problem. A machine learning model can transform this diagnosis based on qualitative criterion faster and more objective, occupying less time of professionals and analyzing quantitative criterion.

## 1.4. Datasets and Inputs

The dataset is from the University of California, Irvine, School of Information and Computer Sciences, and can be accessed through the link:

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29 (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)

and on kaggle:

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data)

The fine needle aspiration data set for breast lesions contains 569 samples of fine needle aspirate from breast nodules (FNAB), including 212 positive samples (malignancy) and 357 negative (benign) samples. All the samples were confirmed by biopsy.

Each sample contains the following information:

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

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

- a) radius (mean of distances from center to points on the perimeter)
- b) texture (standard deviation of gray-scale values)
- c) perimeter
- d) area
- e) smoothness (local variation in radius lengths)
- f) compactness (perimeter ^ 2 / area - 1.0)
- g) concavity (severity of concave portions of the contour)
- h) concave points (number of concave portions of the contour)
- i) symmetry
- j) fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

#### 1.4.1. Data Exploration

The dataset consists of a single file called data.csv, which contains the following columns:

- id
- diagnosis
- radius_mean
- texture_mean
- perimeter_mean
- area_mean
- smoothness_mean
- compactness_mean
- concavity_mean
- concave points_mean
- symmetry_mean
- fractal_dimension_mean
- radius_se
- texture_se
- perimeter_se
- area_se
- smoothness_se

- compactness_se
- concavity_se
- concave points_se
- symmetry_se
- fractal_dimension_se
- radius_worst
- texture_worst
- perimeter_worst
- area_worst
- smoothness_worst
- compactness_worst
- concavity_worst
- concave points_worst
- symmetry_worst
- fractal_dimension_worst

| | id | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | compactness_mean | concavity_mean | concave points_mean | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 842302 | M | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0.27760 | 0.3001 | 0.14710 | ... |
| 1 | 842517 | M | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0.07864 | 0.0869 | 0.07017 | ... |
| 2 | 84300903 | M | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0.15990 | 0.1974 | 0.12790 | ... |
| 3 | 84348301 | M | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0.28390 | 0.2414 | 0.10520 | ... |
| 4 | 84358402 | M | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0.13280 | 0.1980 | 0.10430 | ... |

Before training the model, it will be necessary to perform a pre-processing of the data, which includes removing the "id" column as it is not a relevant feature, checking for missing data and finally normalizing the data between the values of 0.0 and 1.0.

## 1.5. Solution Statement

Before building the classifier model, python libraries such as numpy, pandas, matplotlib and seaborn will be used to visualize and clean the data. SKlearn mathematical techniques such as MinMaxScaler will be used in the data pre-processing step, to make the data less sparse for the model.

The k-fold cross-validation method with k = 5 will be used to verify that the results of the models are consistent.

Subsequently, models will be built using SKlearn supervised estimators, such as:

- SVM
- Gradient Boosting Classifier
- Stochastic Gradient Descent
- Decision tree
- Naive Bayes

## 1.6. Benchmark Model

the results of each model will be compared to define the best solution. It is expected to achieve an accuracy greater than 95% in the average K-fold training.

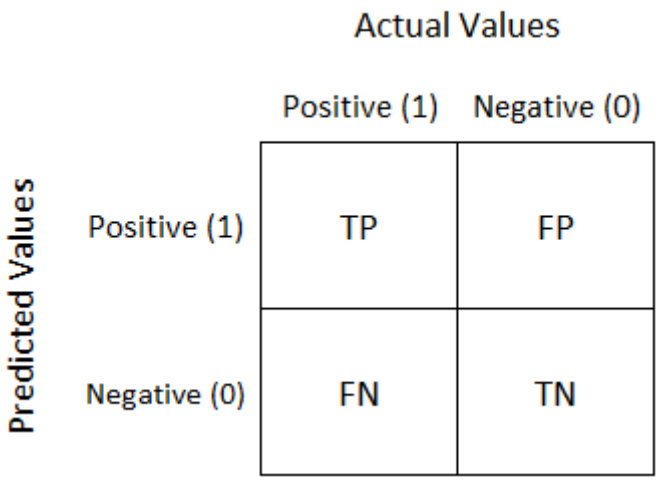
## 1.7. Evaluation Metrics

To validate the models, the following metrics will be used:

### 1.7.1. Confusion Matrix

The Confusion matrix is one of the most intuitive and easiest metrics used for finding the correctness. The Confusion matrix in itself is not a performance measure as such, but almost all of the performance metrics are based on Confusion Matrix and the numbers inside it.

- True Positives (TP): True positives are the cases when the actual class of the data point was 1(True) and the predicted is also 1(True)
- True Negatives (TN): True negatives are the cases when the actual class of the data point was 0(False) and the predicted is also 0(False)
- False Positives (FP): False positives are the cases when the actual class of the data point was 0(False) and the predicted is 1(True). False is because the model has predicted incorrectly and positive because the class predicted was a positive one. (1)
- False Negatives (FN): False negatives are the cases when the actual class of the data point was 1(True) and the predicted is 0(False). False is because the model has predicted incorrectly and negative because the class predicted was a negative one. (0)




### 1.7.2. Accuracy

Accuracy in classification problems is the number of correct predictions made by the model over all kinds predictions made.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{All Samples}}$$

### 1.7.3. Precision

Let’s use the same confusion matrix as the one we used before for our cancer detection example. Precision is a measure that tells us what proportion of patients that we diagnosed as having malignant cancer, actually had malignant cancer. The predicted positives (People predicted as malignant cancerous are TP and FP) and the people actually having a malignant cancer are TP.

$$\text{Precision} = \frac{TP}{TP + FP}$$

#### 1.7.4. Recall

Recall is a measure that tells us what proportion of patients that actually had malignant cancer was diagnosed by the algorithm as having malignant cancer. The actual positives (People having malignant cancer are TP and FN) and the people diagnosed by the model having a malignant cancer are TP.

$$\text{Recall} = \frac{TP}{TP + FN}$$

#### 1.7.5. F1 Score

The F1 score combines recall with precision so that they bring in a single number.

$$F1 = \frac{2 * precisão * recall}{precisão + recall}$$

## 1.8. Project Design

#### 1.8.1. Data visualization

- View dataset graphs to understand it better

#### 1.8.1. Data preprocessing

- Check for missing data.
- Transform non-numeric fields into numeric ones, if necessary.
- Normalize the data using MinMaxNormalization.

#### 1.8.1. Model Building

- Train different types of models, using K-fold in all.
- Check which was the best model using the metrics defined above