# Problem Statement:

Breast Cancer Prediction using Machine Learning.

In 2020, approximately 2.3 million women were diagnosed with breast cancer with a mortality rate of ~30%. According to the last 5 years' records, this has become the world's most prevalent cancer. Based on the above facts, effective breast cancer prediction using machine learning methods can help reduce the mortality rate, prevent cancer proliferating into later severe stages as well as curation of the affected people. The objective of this project is to identify an approximately accurate model to predict the incidence of breast cancer based on various patients' clinical records.

# Solution:

# We are using Breast Cancer dataset available at [Data World](https://data.world/gerry2994/breast-cancer/workspace/file?filename=Breast_cancer_data.csv). The solution has been designed using Python 3.7 and Jupyter Notebook as IDE.

# Step 1: Import Libraries and Necessary Installation

Load the necessary libraries and do required installation.

Text, letter

Description automatically generated

Text

Description automatically generated

**Step 2: Read file.**

Loaded the Breast Cancer dataset ("Breast\_cancer\_data.csv") using pandas read\_csv into a dataframe named “df”. Checked memory usage using the info method in Pandas.

**Step 3: Feature Descriptions.**

The feature descriptions are given below:

*Id* - ID number

*Diagnosis* - The diagnosis of breast tissues (M = malignant, B = benign)

*Radius\_mean* - mean of distances from center to points on the perimeter

*Texture\_mean* - standard deviation of gray-scale values

*Perimeter\_mean* - mean size of the core tumor

*area\_mean* – mean area of the tumor

*smoothness\_mean* - mean of local variation in radius lengths

*compactness\_mean* - mean of perimeter^2 / area

*concavity\_mean* - mean of severity of concave portions of the contour

*concave points\_mean* - mean for number of concave portions of the contour

*symmetry\_mean* – mean of symmetry

*fractal\_dimension\_mean* - mean for "coastline approximation"

*radius\_se* - standard error for the mean of distances from center to points on the perimeter

*texture\_se* - standard error for standard deviation of gray-scale values

*perimeter\_se* – standard error for perimeter of the tumor

*area\_se* – standard error for area of the tumor

*smoothness\_se* - standard error for local variation in radius lengths

*compactness\_se* - standard error for perimeter^2 / area

*concavity\_se* - standard error for severity of concave portions of the contour

*concave points\_se* - standard error for number of concave portions of the contour

*symmetry\_se* – standard error for symmetry

*fractal\_dimension\_se* - standard error for "coastline approximation"

*radius\_worst* - "worst" or largest mean value for mean of distances from center to points on the perimeter

*texture\_worst* - "worst" or largest mean value for standard deviation of gray-scale values

*perimeter\_worst* – “worst” or largest mean value for perimeter of the tumor

*area\_worst* – “worst” or largest mean value for area of the tumor

*smoothness\_worst* - "worst" or largest mean value for local variation in radius lengths

*compactness\_worst* - "worst" or largest mean value for perimeter^2 / area

*concavity\_worst* - "worst" or largest mean value for severity of concave portions of the contour

*concave points\_worst* - "worst" or largest mean value for number of concave portions of the contour

*symmetry\_worst* - “worst” or largest mean value for symmetry of the tumor

*fractal\_dimension\_worst* - "worst" or largest mean value for "coastline approximation"

**Step 4: Exploratory Data Analysis (EDA)**

Analyzed the shape, feature size, data types and statistics for the dataset. The percentage of Malignant cases (37.26%) vs Benign cases (62.74%) in the dataset shows the imbalance in the dataset. Checked for numerical and categorical variables so that we know how to handle the variables with respect to the machine learning algorithms.

While checking for duplicate values, we found 569 duplicate records. There is no columns with NULL records. Hence, we are confirmed that NULL handling is not required as well as no imputation needs to be carried out. The dataset has multiple duplicates. Hence the duplicates need to be removed as part of Feature Engineering. Checked for presence of negative values. We did not find any negative values in the dataset.

We created a new dataframe named “df1” where we have converted the categorical variable Diagnosis into numerical variable which will help in visualization of the dataset.

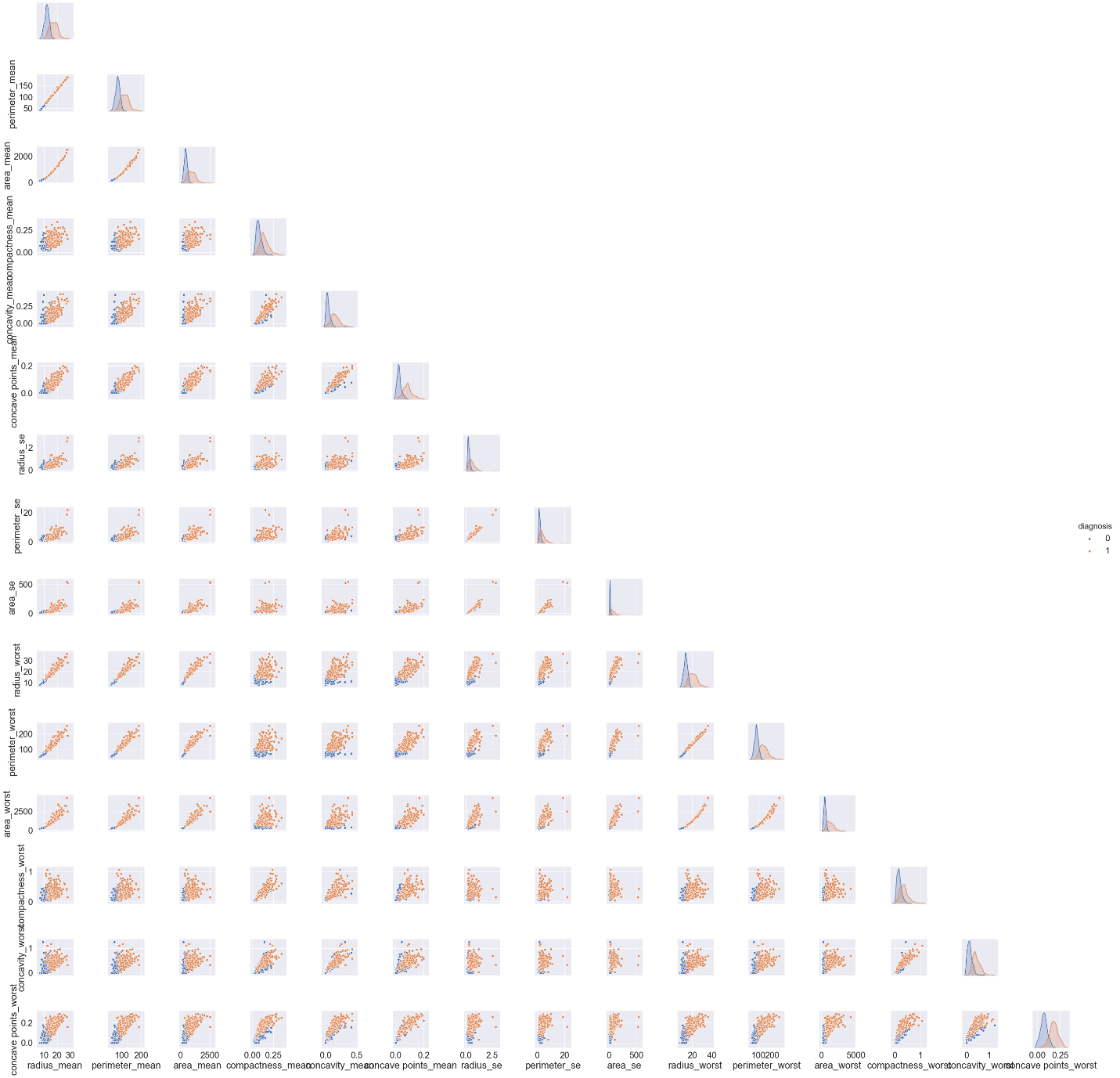
The correlation matrix has been created to identify any meaningful insights from the datasets and how different variables impact each other. Diagnosis is highly positively related with radius\_mean, perimeter\_mean, area\_mean, compactness\_mean, concavity\_mean, concave points\_mean, radius\_se, perimeter\_se, area\_se, radius\_worst, perimeter\_worst, area\_worst, compactness\_worst, concavity\_worst and concave point\_worst.

A picture containing text

Description automatically generated

Area, perimeter, and radius are highly correlated with each other based on normal geometry. Smoothness\_mean is related to compactness\_mean, concavity\_mean, concave points\_mean, symmetry\_mean and fractal\_dimension\_mean. Compactness\_se is highly correlated with concavity\_mean, fractal\_dimension\_mean, concavity\_se, concave points\_se, fractal\_dimension\_se, concavity\_worst and fractal\_dimension\_worst. Concavity\_se is related to compactness\_mean, compactness\_se, compactness\_worst, fractal\_dimension\_se, fractal\_dimension\_worst, concave\_point\_se.

We created a new dataframe named “df\_sliced” which contains the attributes which are highly correlated with diagnosis in order to create pair plots to find hidden patterns.



We created distribution plots for diagnosis vs each of the correlated attributes.

Chart, histogram

Description automatically generated Chart, histogram

Description automatically generated Chart

Description automatically generated

Chart, histogram

Description automatically generated Chart, histogram

Description automatically generated Chart

Description automatically generated Chart

Description automatically generated Chart

Description automatically generated Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated Chart

Description automatically generated Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated Chart, histogram

Description automatically generated

We created histogram for the diagnosis attribute which shows the class imbalance.

Chart, bar chart

Description automatically generated

Visualization through boxplots has been created which show that area\_worst and area\_mean have the most outliers.

Chart, box and whisker chart

Description automatically generated

We separated mean, se and worst attributes in order to plot them to find out any hidden patterns.

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart

Description automatically generated

**Step 5: Feature Engineering**

For a cleaner dataset to start our solution, we dropped duplicates.

As part of Outlier handling, we found out the outlier values for area\_worst and area\_mean as those were having the most outliers. We checked the diagnosis type for the corresponding outliers and found that all those belong to malignant cases.

Hence, we didn’t remove those as that may lead to Type II error or False Negatives, that is the model may predict a malignant case as benign one which is detrimental. We found out revised statistics after removing duplicates. We converted the categorical variable Diagnosis into numerical variable with Label Encoding on the main dataframe “df” because most of the ML algorithms perform better and produce better accuracy with numerical variables.

**Step 6: Split the dataset into train and test.**

We split the dataset df into train and test, i.e. 80% of the dataset is used for training the model and 20% of the dataset is used for testing the model. Then we separated the output (Y) variable (‘Diagnosis’) from input (X) variable for both train and test datasets.

**Step 7: Model Selection**

We checked the Accuracy and F1 scores for some ML models and selected the model with highest Accuracy and F1 score.

1. Random Forest Model

Roc Auc Score: 96.49%

F1 score: 0.95

1. Decision Tree

Roc AUC Score: 92.98%

F1 Score: 0.91

1. K-Nearest Neighbors

Roc AUC Score: 72.81%

F1 Score: 0.54

1. XGBoost

Roc AUC Score: 95.61%

F1 Score: 0.94

1. SVM

Roc AUC Score: 62.28%

F1 Score: 0

The F1 score was an important differentiator in this model selection, because F1 score being harmonic mean of Precision and Recall, handles the imbalance of the dataset while calculating the metric. This metrics shows that XGBoost is having a high F1-score along with greater accuracy with respect to other models.

**Step 8: Balancing the unbalanced data**

We balanced our imbalanced Train dataset using SMOTE. In the original dataset, we noticed that it contains only 37.26% malignant cases which represents the minority class. Using SMOTE data augmentation methodology, helps to synthesize examples of the minority class without adding any new information.

**Step 9: Model Training after balancing the data**

After applying the SMOTE, we re-executed the XGBoost model and captured the performance

metrics. We noticed the Accuracy score of the XGBoost model is 97.37% and the F1 score of the XGBoost model is 0.96, which is still the highest among the other Models tested for the

purpose. We created Confusion Matrix to check true positives, true negatives, false positives

and false negatives.

The ROC Curve shows that the model has higher chance of predicting the probability that a medical case is malignant based on the AUC score of 1 which is much higher compared to no-skill line of 0.5.

Chart, line chart

Description automatically generated