Breast Cancer Wisconsin (Diagnostic) Prediction

CS504 - Programming Languages for Data Analysis
Final Report

Reza Zare

Overview:

Each year, cancer affects millions worldwide, with more than 18.1 million cases reported in 2020. 8.8 million women were diagnosed with cancer during this period, representing over 25.8 of all possible cancers. The American Cancer Society projects that by 2022 more than 43,250 women will die from breast cancer, and more than 287,850 new cases of invasive breast cancer will be diagnosed in the USA only. Our project focuses on this disease, specifically on recognizing breast tumors and their potential condition. In medical terms, a tumor can be benign (non-cancerous) or malignant (cancerous). We used a dataset from researchers at the University of Wisconsin in the Clinical Sciences Center and the Department of Computer Science. This dataset was created on the 1st of November 1995 and the method of data measurement is as follows:

Features are computed from a digitized image of a fine needle aspirate FNA of a breast mass. They describe the characteristics of the cell nuclei present in the image. The aim of this report is to analyze the selected dataset and apply different statistical and machine-learning models. We want to predict tumor type using these features and compare the best models that can remain globally accurate and perform better at correctly categorizing malignancies to allow patients to be followed up for possible treatment almost immediately. We will use a partition to train our models and we will use a test partition to see the prediction stability of our best models. Thus, the malignant tumor type will be considered as our positive output (M / 1) and the benign tumors as our negative output (B / 0), so we want to maximize the accuracy and sensitivity of our models and we try to optimize the hyperparameter of our best model to obtain the best result and accuracy.

Keywords: Machine Learning, malignancy, cancerous, non-cancerous, benign, breast cancer · Breast cancer Wisconsin (BCW), classification model, logistic regression, random forest.

Introduction:

Breast cancer is a type of cancer that starts in the breast. Cancer starts when cells begin to grow out of control. Breast cancer cells usually form a tumor that can often be seen on an x-ray or felt as a lump. Breast cancer occurs almost entirely in women, but men can get breast cancer, as well. It's important to understand that most breast lumps are benign and not cancer (malignant). Non-cancerous breast tumors are abnormal growths, but they do not spread outside of the breast. They are not life-threatening, but some types of benign are. Breast lumps can increase women's risk of getting breast cancer. Any breast lump or change needs to be checked by a healthcare professional to determine if it is benign or malignant (cancer) and if it might affect your future cancer risk.

1. Data Understanding

The key challenge against its detection is how to classify tumors into "malignant (cancerous)" or "benign(non-cancerous)". The dataset contains information about various features of breast cancer cells and a corresponding classification of whether the cells are malignant or benign.

1.2. Understand the data

Attribute Information:

- ID number
- Diagnosis (M = malignant, B = benign)
- radius (mean of distances from the center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

Attribute Interpretation: The mean, standard error (SE), and worst of these features were computed for each image, each contains 10 parameters (radius, texture, area, perimeter, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension), resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, and field 23 is the Worst Radius.

1.3. Read and Analyze Data

inspecting the first three rows of the dataset:

2. Data Processing / Preparation

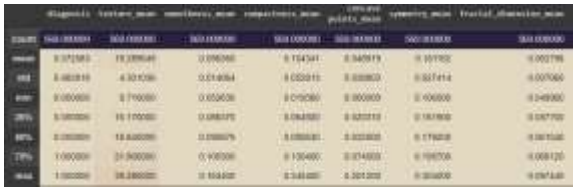
It involves data cleaning, feature selection, data splitting, and encoding categorical variables which leads to more accurate predictions.

2.1. Descriptive Statistics

Structural Analysis: Before we look into the content of the data, we first need to look into the general structure of the data, i.e., the number of rows (data points) and the number of columns (features) in it. getting the shape, size, and data types.

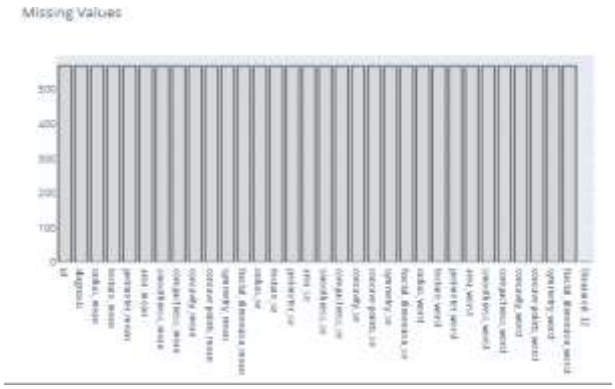
- the shape of the dataset: **(569, 33)**
- The size of the dataset: **18777**
- data types of the dataset: **int64, object, and float64**
- All feature values are recorded with four significant digits.

Inspecting the descriptive statistics of part of the data:



We have features but what do they mean or actually how much do we need to know about these features that we do not know, we should know something like variance, standard deviation, number of the sample (count), or max-min values. This type of information helps to understand what is going on in our data. For example, **area_mean** feature's max value is 2501 and the **smoothness_mean** features' max is 0.16340. Therefore do we need standardization or normalization before visualization, feature selection, or classification? The answer is yes.

2.2. Inspect & Drop Missing values



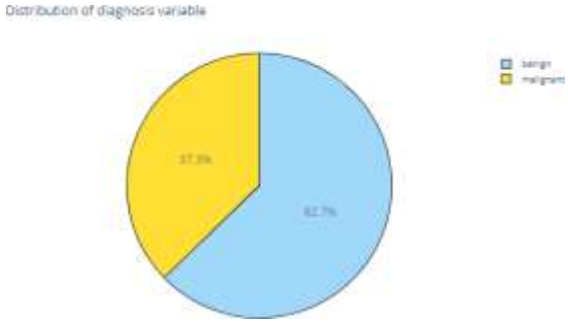
Note: we can see that the last column is null.

2.3. Reassign target and drop useless features:

- There are 4 things that catch the attention
- There is an **id** that cannot be used for classification
- **Diagnosis** is our **class label**
- **Unnamed: 32** features include **NaN** so we do not need it.

2.4. Target Distribution

The column "**diagnosis**" has two values: Malignant and Benign. Machine learning models can be built on data that is made of just numbers. Hence, we will replace Malignant with the number 1 and Benign with the number 0. Once replaced, the code **df.diagnosis.unique()** will serve as a check that we get a resulting column of numbers 1 and 0.



- Number of cells labeled Benign: **357**
- Number of cells labeled Malignant: **212**
- % of cells labeled Benign **62.74 %**
- % of cells labeled Malignant **37.26 %**

3. EDA (Exploratory Data Analysis)

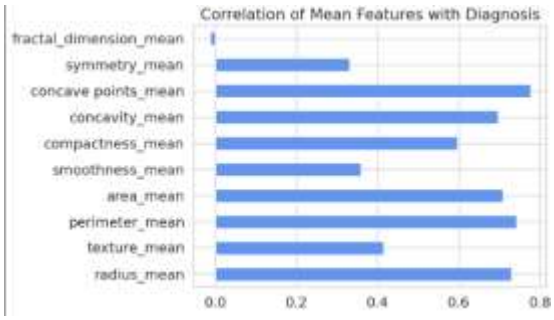
Exploratory Data Analysis (EDA) involves descriptive statistics, correlation analysis, and analyzing the various features of breast cancer cells and their classifications to gain insights into the data.

3.1. Kernel Subplot Correlation

Extracting Mean, Squared Error, and Worst Features

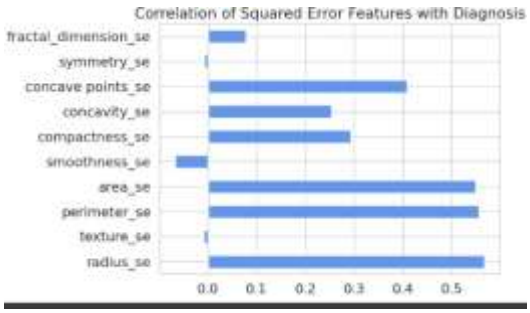
Dividing features into groups for easier kernel subplot readability

Correlation of our subplots with Diagnosis



Observations:

- fractal_dimension_mean is the least correlated with the target variable.
- All other mean features have a significant correlation with the target variable.



Observations:

- texture_se, smoothness_se, symmetry_se, and fractal_dimension_se are least correlated with the target variable.
- All other squared error features have a significant correlation with the target variable.

Observation:

All worst features have a significant correlation with the target variable.

3.2. Correlation Matrix

A correlation matrix shows the relation between two given variables in the form of a matrix. Such a matrix colored with a heat map will make it much easier to read. It's also known as the error matrix,

- -1 indicates a perfect negative linear correlation between two variables
- 0 indicates no linear correlation between the two variables
- 1 indicates a perfect positive linear correlation between two variables

Confusion Matrix Squares Interpretation:

- true positive (TP): Malignant tumor correctly identified as malignant
- true negative (TN): Benign tumor correctly identified as benign
- false positive (FP): Benign tumor incorrectly identified as malignant
- false negative (FN): Malignant tumor incorrectly identified as benign

Correlation Matrix for the subsets:

One of the assumptions in most of the key Machine learning models is that no variable in the model is highly correlated to any other variable. A high correlation between variables causes the problem of multicollinearity and hence it's important to be aware of the relationships between each variable to better interpret the results of a model.

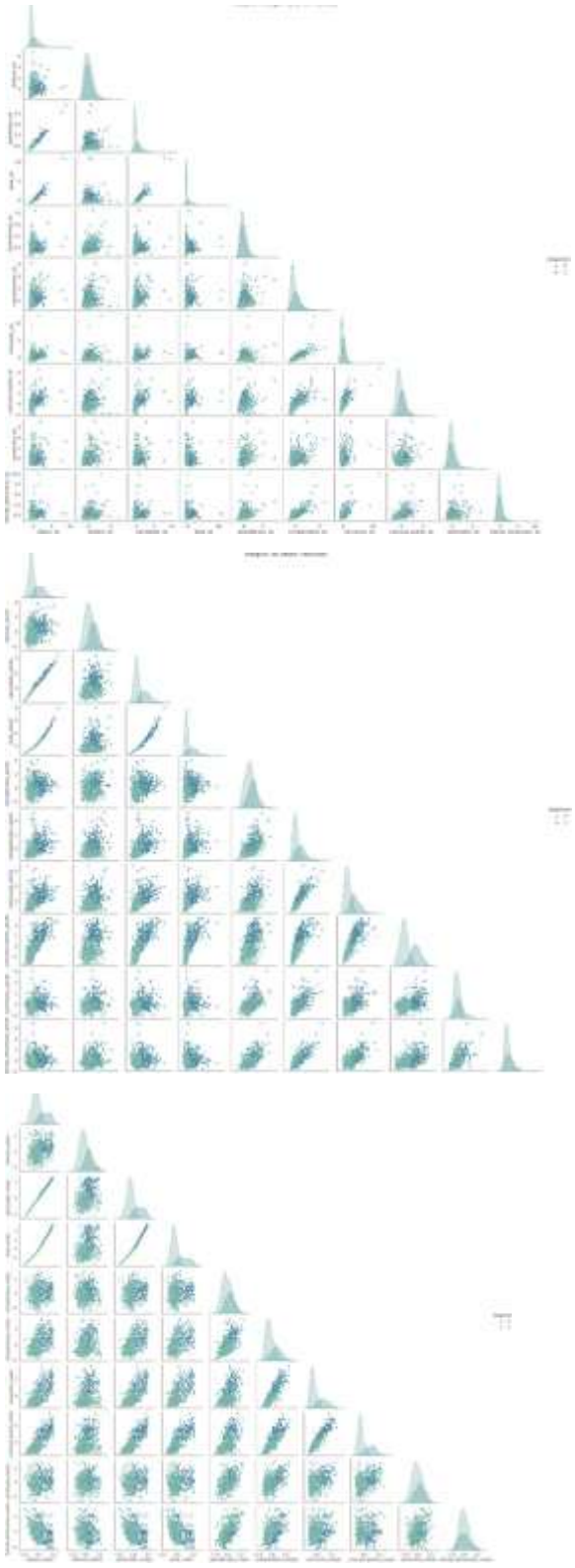
Looking at the matrix, we can immediately verify the presence of multicollinearity between some of our variables. For instance, the **radius_mean** column has a correlation of 1 and 0.99 with the **perimeter_mean** and **area_mean** columns, respectively. This is probably because the three columns essentially contain the same information, which is the physical size of the observation (the cell). Therefore we should only pick one of the three columns when we go into further analysis.

Another place where multicollinearity is apparent is between the "**mean**" columns and the "**worst**" column. For instance, the **radius_mean** column has a correlation of 0.97 with the **radius_worst** column. In fact, each of the 10 key attributes displays very high (from 0.7 up to 0.97) correlations between its "**mean**" and "**worst**" columns.

Similarly, it seems like there is multicollinearity between the attributes **compactness, concavity, and concave_points**. Just like what we did with the size attributes, we should pick only one of these three attributes that contain information on the shape of the cell. I think **compactness** is an attribute name that is straightforward, so we can remove the other two attributes.

We can now go ahead and drop all unnecessary columns, but let's have a deeper understanding of the data and we will do the feature selection later.

Checking Multicollinearity Between Distinct Features:



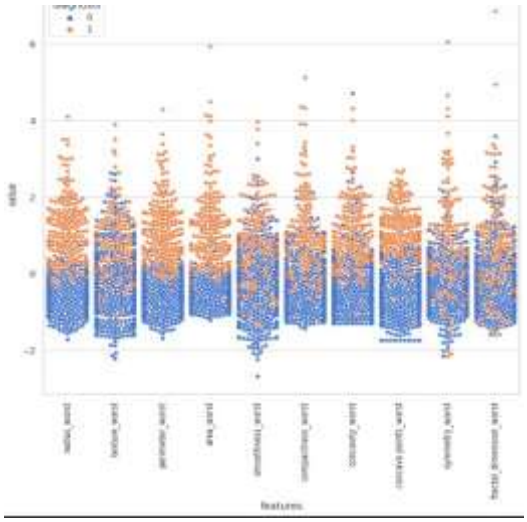
Observations:

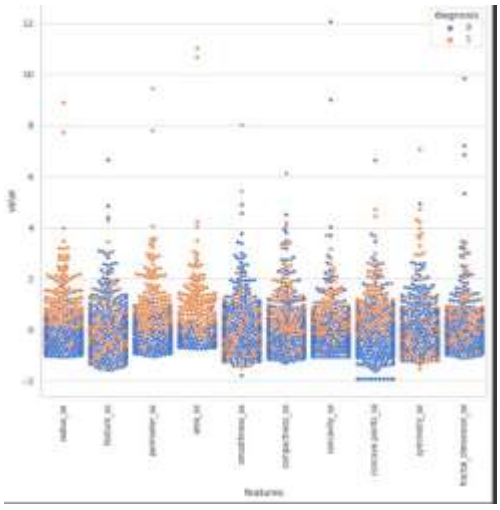
Almost perfectly linear patterns between the **radius, perimeter, and area** attributes are hinting at the

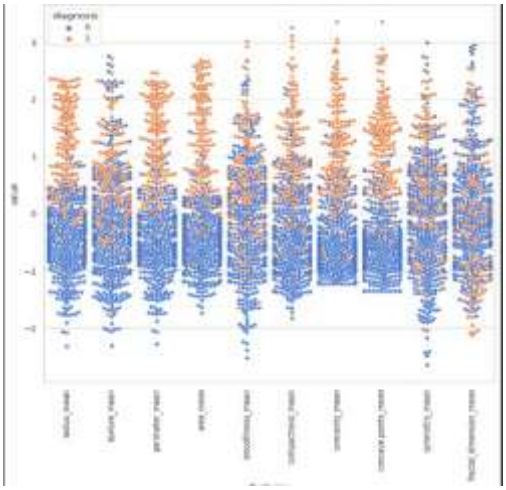
presence of multicollinearity between these variables. Another set of variables that possibly imply multicollinearity are **concavity, concave points, and compactness.**

3.3. Relationship Between Variables

- We can say that two variables are related to each other if one of them gives information about the others
- For example, price and distance. If you go a long distance by taxi you will pay more. Therefore we can say that price and distance are positively related to each other.







Observations:

- A certain level of separation in the values for the radius, perimeter, and area in the benign & malignant data points (as also observed in the diagonal plot of a pair plot)
- Each of the features appears to have outliers.
- The distribution appears to be Gaussian with a right skew.

- **area_worst** in the last strip plot looks like malignant and benign are separated not totally but mostly. However, **smoothness_se** in strip plot 2 looks like malignant and benign are mixed so it is hard to classify while using this feature.

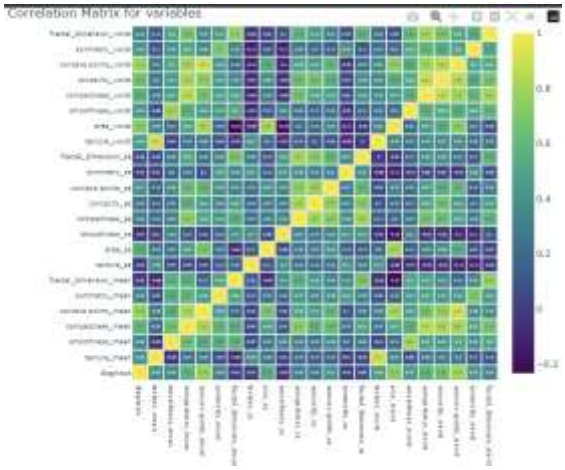
4. Feature Selection:

a process of selecting a subset of relevant features from a large set of features to improve model performance, reduce the risk of overfitting, and reduce computational time and cost.

> We selected features based on correlation.

Feature Selection with Correlation

- Based on our plots and correlation matrix, we will select the best features with the best correlation with the target variable.
- We can see that there are many columns that are very highly correlated which causes multi-collinearity so we have to remove highly correlated features to have a better result.
- We will remove columns with more than 0.92 correlation to avoid multicollinearity.



after removing, the number of features drops to 22 features excluding the label. As can be seen, there is no multicollinearity.

> There is no high correlation between variables as can be seen from the correlation matrix.

5. Model Selection & Model Building

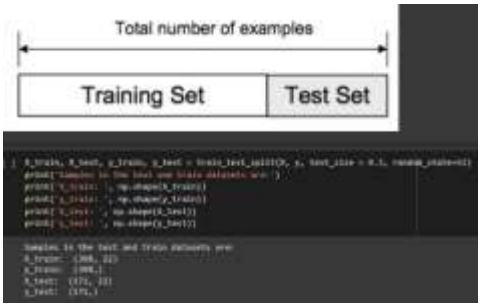
Involves building and developing a model that can accurately classify breast cancer cells as malignant or benign based on their features, which is used to improve the diagnosis and treatment of breast cancer.

We will start by first splitting our dataset into two parts; one as a training set for the model, and the other as a test set to validate the predictions that the model will make. If we omit this step, the model will be trained and tested on the same dataset, and it will underestimate the true error rate, a phenomenon known as overfitting. It is like writing an exam after taking a look at the questions and answers beforehand. We want to make sure that our model truly has predictive power and is able to accurately label unseen data. We will set the test size to 0.3; i.e., 70% of the data will be assigned to the training set, and the remaining 30% will be used as a test set. In order to obtain consistent results, we will set the random state parameter to a value of 42.

5.2. Splitting the dataset

```
[ ]  X = data.drop('diagnosis',axis = 1)
     y = data.diagnosis  # M or B
```

5.3. Splitting to train and test sets



5.4. Feature Scaling

Why do you need to standardize your data? For example, a variable that ranges between 0 and 100 will outweigh a variable that ranges between 0 and 1. Using these variables without standardization in effect gives the variable with the larger range a bigger weight in the analysis

We need to standardize our data to put all data points into a mean of 0 and standard deviation of 1 since the total distribution of data follows the Gaussian Distribution and has a bell-shaped trend and is skewed to the sides for most plots.

5.5. Initialization, Fitting, and Predicting the Model

In our dataset, we have the outcome variable or Dependent variable i.e **y** having only two sets of values, either M (Malign) or B(Benign). So we will use the Classification algorithm of supervised learning.

We have tested different classifiers to select the best model amongst them.

Result of running multiple classifiers:

Here we introduce the most important metrics to read a confusion matrix that is necessary to understand this report.

- **Accuracy**: The ratio of correctly classified records of both classes to the total number of records. **(TP +TN) / (TP + TN + FP +FN)**
- **Precision**: The ratio of correctly classified malign cancers to the total number of records whose diagnosis is malign. **TP / (TP + FP)**
- **Recall**: It measures the proportion of actual positive cases that are correctly identified by the model (true positives) out of all positive cases, including those that were missed by the model (false negatives). **TP / (TP + FN)**
- F1-score: It is the harmonic mean of precision and recall, which provides a balance between the two measures. **2 * (precision * recall) / (precision + recall)**
- **Balanced Accuracy:** It measures the average of the true positive rate and true negative rate of the model. **1/2 (TP/(TP+FN)) (TN/(TN+FP))**

	accuracy	f1_score	precision	recall	balanced_accuracy
LogisticRegression	0.970760	0.976959	0.972477	0.981481	0.966931
RandomForest	0.964912	0.972222	0.972222	0.972222	0.962302
SVM	0.959064	0.967742	0.963303	0.972222	0.954365
KNearsNeighbors	0.953216	0.963303	0.954545	0.972222	0.946429
DecisionTree	0.947368	0.958140	0.962617	0.953704	0.945106
NaiveBayes	0.929825	0.944444	0.944444	0.944444	0.924603

Among all 6 models tested, **Logistic Regression** performed better than the others so we will choose this classifier for our further evaluation.

6. Model Evaluation:

This involves measuring how accurately the model can predict the target on the testing data and using metrics such as accuracy, precision, recall, and F1-score, tuning the hyperparameters and their curve plots to assess its predictive power.

6.1. Evaluating & Plotting Model Performance

ROC & AUC:

ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0 classes as 0 and 1 class as 1.

The ROC Curve:

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

Cross-validation:

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

6.2. Optimize & Tune Hyperparameters

Need of hyperparameter tuning:

- To avoid over-fitting
- To avoid under-fitting

Process of hyperparameter tuning:
In model optimization, we split the data into three parts

- train-train the model on the given parameter
- cv-optimize the model's parameter valu**es**
- test-evaluate the optimized model

Hyperparameter tuning is an optimization technique and is an essential aspect of the machine-learning process. A good choice of hyperparameters may make your model meet your desired metric.

GridSearchCV:

It is used for finding the optimal parameter values from a given set of parameters in a grid. It's essentially a cross-validation technique.

Important GridSearchCV argument:

- **model:** Choose the model which you want to pass like-Random forest, decision tree, etc
- **param_grid**: Dictionary with parameters names (str) as keys and lists of parameter settings to try as values or a list of such dictionaries
- **scoring**: Strategy to evaluate the performance of the cross-validated model on the test set
- **n_jobs**: Number of jobs to run in parallel. -1 means using all processors.
- **cv**: Determines the cross-validation splitting strategy. Possible inputs for the cv are:
    - **None**, to use the default 5-fold cross-validation
    - **integer**, to specify the number of folds in a (Stratified)KFold
    - **CV splitter**
    - An iterable yielding (train, test) splits as arrays of indices.

Logistic Regression Hyperparameter Tuning:

We now try to tune Logistic Regression Hyperparameters although it does not really have any critical hyperparameters to tune.

Sometimes, you can see useful differences in performance or convergence with different solvers.

solver in ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']

Regularization (penalty) can sometimes be helpful.

penalty in ['none', 'l1', 'l2', 'elasticnet']

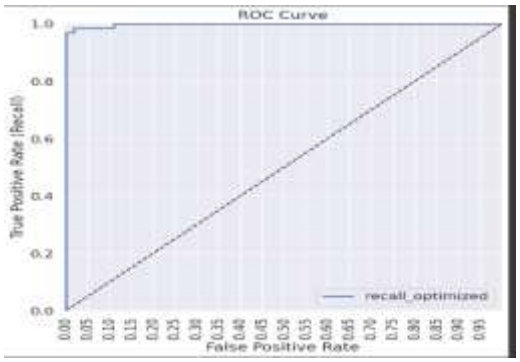
Note: not all solvers support all regularization terms.

The C parameter controls the penalty strength, which can also be effective.

C in [100, 10, 1.0, 0.1, 0.01]

The best parameter selected by GridSearchCv is:

{'C': 1, 'penalty': 'l2', 'solver': 'saga'}



The AUC (a measure of separability) **score is around 99.79% which means** it successfully predicts 0 classes as 0 and 1 classes as 1.

Before hyperparameter tuning:

	accuracy	f1_score	precision	recall	balanced_accuracy
LogisticRegression	0.970760	0.976939	0.972477	0.981481	0.966931

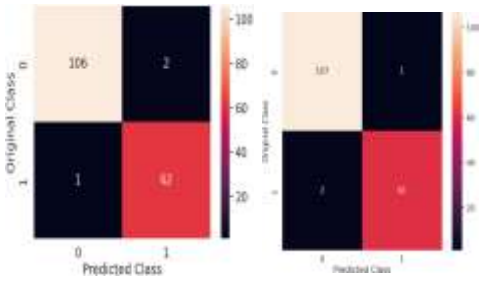
After hyperparameter tuning with penalty L1:

	accuracy	f1_score	precision	recall	balanced_accuracy
LogisticRegression	0.982456	0.976	0.983871	0.968254	0.979497

After hyperparameter tuning with penalty L2:

	accuracy	f1_score	precision	recall	balanced_accuracy
LogisticRegression	0.982456	0.976378	0.96875	0.984127	0.982904

The balanced accuracy of Logistic Regression increased using GridSearchCV from 96.69 to 97.94 with FP=1 & FN=2 with penalty=L1 and 98.28 with FP=2 & FN=1 with penalty=L2



Conclusion:

- We started by analyzing the data types of the feature and converted the string data type to integer (using hot encoding). We Also Checked the distribution of the target variable.
- We then checked for missing values and duplicates and dealt with them.
- Further, we dropped certain features that upon analysis, we found to be irrelevant to the target variable and have multicollinearity.
- After performing preliminary analysis, we plotted the data to get more insights using EDA (exploratory data analysis) methods.
- We found the correlation of all the features with the target variable as well as among each other. We then found the 22 most optimal features by removing highly correlated variables. We have reduced the number of features from 30 to 22.
- We then implemented multiple classifiers and selected **Logistic Regression** as our best classifier for further analysis with **96.69%** accuracy.
- We also increased the accuracy of our model from **~96.6% to ~98.2%** during the process using hyperparameter tuning.

References:

[1] Breast Cancer Wisconsin (Diagnostic) Data Set: Predict whether the cancer is benign or malignant," [source]
[2] Wisconsin Diagnostic Breast Cancer (WDBC) Dataset, Scikit-Learn Machine Learning Repository - [source]
[3] American Institute of Cancer Research Statistics - [source]
[4] Diagnosis of Breast Cancer Pathology on the Wisconsin Dataset with the Help of Data Mining Classification and Clustering Techniques. [source]
[5] A semantic rule-based approach to diagnosing breast cancer based on Wisconsin datasets. [source]

[6] Building a Simple Machine Learning Model on Breast Cancer Data. [source]