Predictive Analytics for Breast Cancer Diagnosis Using PCA-Enhanced Logistic Regression

Prepared for:

UNT ADTA 5230

Prepared by:

Bryanna Williams, Biniam Abebe, Guttikonda Bharat Kumar, and Maybel Hernandez

April 26, 2024

1. Abstract

Breast cancer is a global health concern, with millions of cases diagnosed annually. This research analyzes the Breast Cancer Wisconsin Data Set to identify critical attributes associated with positive diagnoses. The study evaluates model performance using accuracy, precision, recall, and F1-score metrics by reviewing existing machine-learning methodologies for breast cancer prediction and focusing on logistic regression. Results show that the logistic regression model accurately distinguishes between benign and malignant cases with a low false positive rate. The research emphasizes the importance of algorithm selection and feature analysis in improving breast cancer diagnosis and treatment outcomes.

***Keywords****: Machine Learning, Supervised Machine Learning, Predictive Analytics, Breast Cancer Diagnosis, PCA, Logistic Regression*

Contents

[**Introduction 4**](#_30j0zll)

[**Literature Review 4**](#_1fob9te)

[**Data Description 5**](#_3znysh7)

[**Exploratory Data Analysis 7**](#_2et92p0)

[**Methodology 9**](#_tyjcwt)

[Data Preprocessing 9](#_ac4whxk6zwxa)

[PCA Implementation 9](#_1t3h5sf)

[Model Development 10](#_4d34og8)

[Performance Metrics 11](#_2s8eyo1)

[Results 12](#_17dp8vu)

[**Discussion 13**](#_w3r45hf5ia73)

**Conclusion and Recommendation 14**

[**References 16**](#_1ft6r2tfexzn)

[**Appendix 17**](#_lnxbz9)

## Introduction

This study analyzes the Breast Cancer Wisconsin (Diagnostic) Data Set from the UC Irvine Machine Learning Repository to conduct our research. This research aims to determine which attributes tested tend to have the most significant connection with a positive diagnosis. These findings could contribute significantly to breast cancer research by allowing clinicians to identify which factors have a heavier impact, guiding research and study practices more heavily toward identifying the cause of those features and how to mitigate them. Accurately detecting a positive diagnosis as early as possible is crucial for getting patients the needed treatment and ultimately leading to a drastic reduction in breast cancer mortality rate. Knowing what features to focus research and healing efforts can drastically impact this goal.

## Literature Review

According to the World Cancer Fund International, breast cancer was the most commonly assigned form of cancer in the year 2020, with over 2.2 million cases that were reported. It was also the culprit of approximately 685,000 deaths in women that same year. With this, it is clear to see that breast cancer poses a significant global health challenge. To improve the prediction and diagnosis of breast cancer, researchers have turned toward machine learning and data mining algorithms to better understand this issue and help resolve this urgent problem. This literature review aims to provide an overview of current developments in machine-learning techniques for detecting and predicting breast cancer based on knowledge from two relevant scientific publications.

In their study to predict the diagnosis of breast cancer, Naji et al. (2001) tested five different machine learning algorithms: Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision Tree (C4.5), and K-Nearest Neighbors (KNN) to understand their performance better. Their study based on the Wisconsin Breast Cancer Diagnostic dataset showed that SVM outperformed other classifiers with 97.2% accuracy. Their research sheds light on the potential machine-learning algorithms have to improve diagnostic accuracy and patient outcomes in breast cancer care.

In a recent study, Fatima et al. (2020) performed an in-depth evaluation of machine-learning methodologies for predicting breast cancer. Their review examined several linear, nonlinear, and ensemble algorithms. Their analysis included algorithms such as Linear Regression, Logistic Regression, Classification and Regression Tree, Naive Bayes, K-nearest neighbor, Support Vector Machine, Decision Tree, Random Forest, Boosting, and AdaBoost. The authors aimed to determine the most efficient and precise algorithm for predicting breast cancer through their comprehensive comparative analysis. Their review offers valuable insights into the various machine-learning techniques used in recent breast cancer research, highlighting the importance of algorithm selection in predictive modeling.

In conclusion, machine-learning approaches show promising potential for improving breast cancer prediction and diagnosis. The studies mentioned reinforce the importance of strategically selecting algorithms and evaluating their performance to create reliable predictive models of breast cancer. In the future, it will significantly benefit researchers to focus on addressing issues related to limited data sets, unbalanced data, and algorithm refinement to further enhance diagnostic accuracy and patient outcomes when receiving breast cancer treatment.

## Data Description

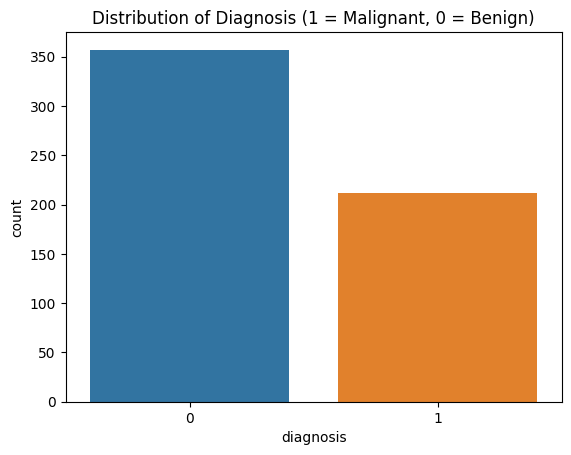
The Breast Cancer Wisconsin (Diagnostic) Data Set includes features calculated from images of fine needle aspirate (FNA) of breast masses. The dataset contains 32 columns and 569 samples. The column description is as follows:

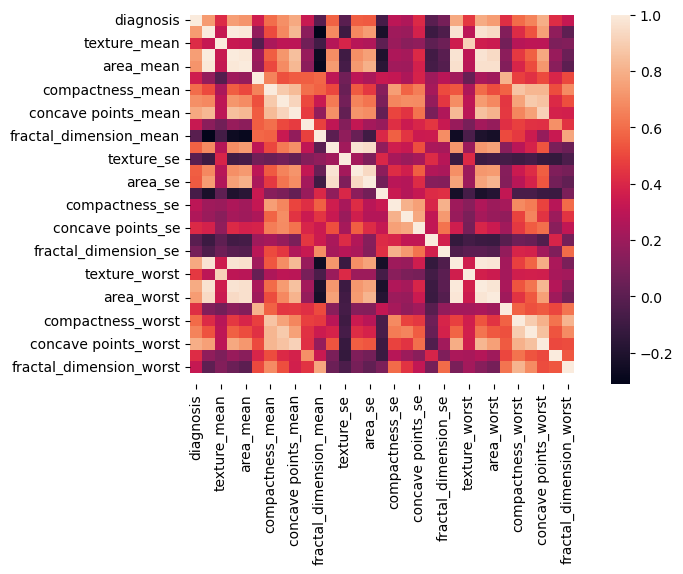
**Column Description:**

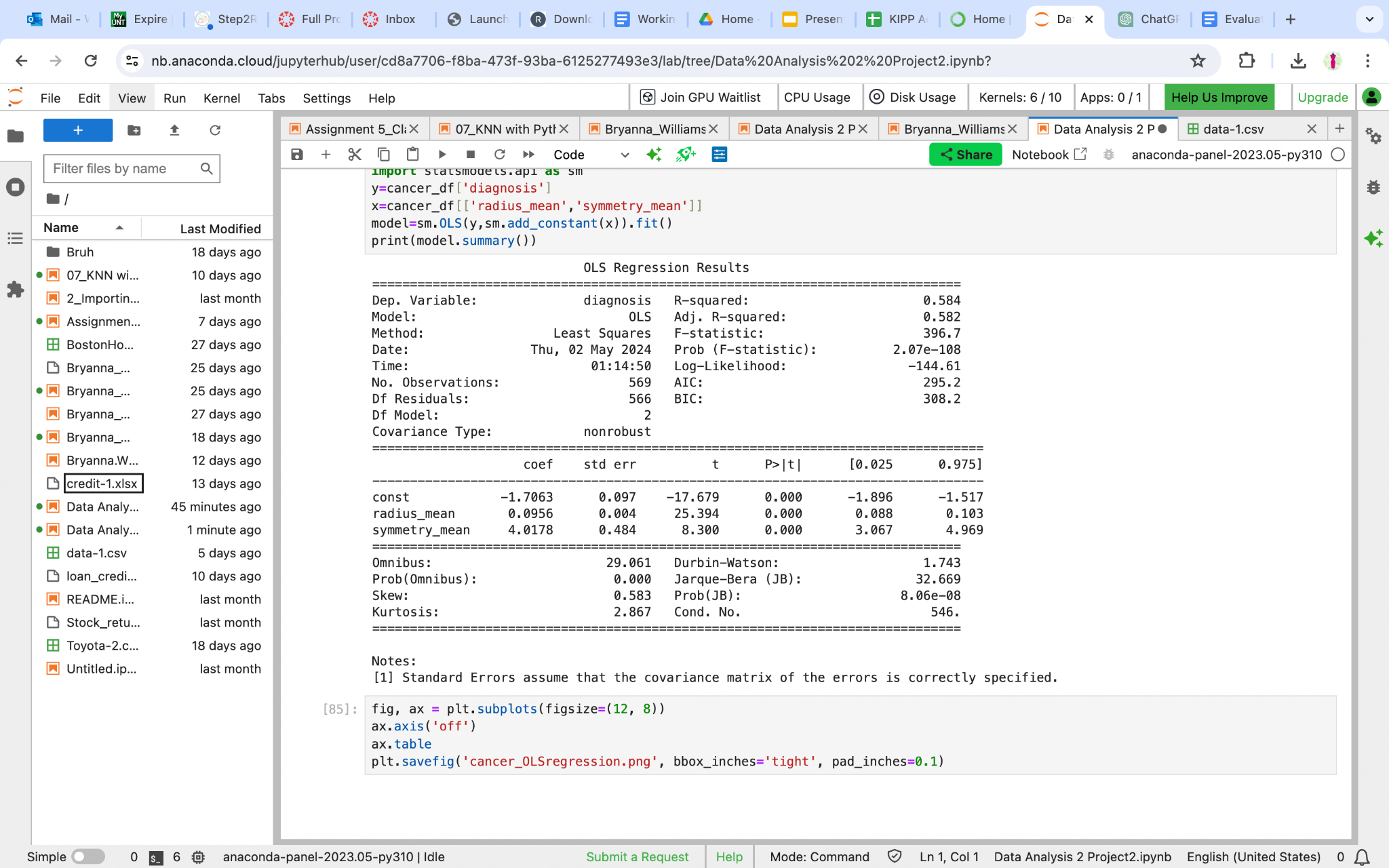
* ID number: Unique identifier for each sample.
* Diagnosis: The diagnosis of breast tissues as malignant (M) or benign (B).
* Radius: Mean distances from the center to points on the perimeter.
* Texture: Standard deviation of gray-scale values.
* Perimeter: Perimeter of the cell nucleus.
* Area: Area of the cell nucleus.
* 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: Symmetry of the cell nucleus.
* Fractal dimension: "Coastline approximation" - 1.

This project will use the Diagnosis column in this dataset as the target variable, which contains the diagnosis of breast tissues as either M (malignant) or B (benign).

## Exploratory Data Analysis

  
 **Table 1- Distribution of Benign and Malignant Cancer Diagnoses**This graphic displays the distribution of Malignant compared to Benign cancer diagnoses. Out of the 569 patients included in the data, 357 had benign diagnoses, and 212 had malignant diagnoses.

**Table 2- Correlation Matrix Heatmap**This heat map illustrates the correlation matrix among the variables in our dataset. Variables highlighted in pale orange display the strongest positive correlation with each other, like texture mean and worst, area worst and perimeter.



**Table 3 - Regression Analysis for Diagnosis Prediction**This table shows the results of a regression analysis where we looked at how the radius influences the diagnosis (dependent variable, y) mean and symmetry mean (independent variables). Our findings suggest that combined two variables can account for about 58.4% of the variation we see in diagnosis results.

## Methodology

### Data Preprocessing

We used data normalization as a component of the pipeline for preprocessing the data. The process of normalization involves adjusting the values of a feature vector so that it can be evaluated on a standardized scale, enabling comparisons to be made with other feature vectors. By employing this approach, it becomes feasible to enhance the efficiency and speed of the machine learning algorithms, specifically when the features exhibit disparate value ranges. The software being utilized is Sklearn, a Python library. For the purpose of normalizing the data, the StandardScaler class preprocessing module was used. StandardScaler standardizes features by removing the mean from the dataset and scaling the variance to a unit of measure. As a result of this operation, each feature is dealt with separately.

**Code Snippet**

from sklearn.preprocessing import StandardScaler  
scaler = StandardScaler()  
X\_scaled = scaler.fit\_transform(X)

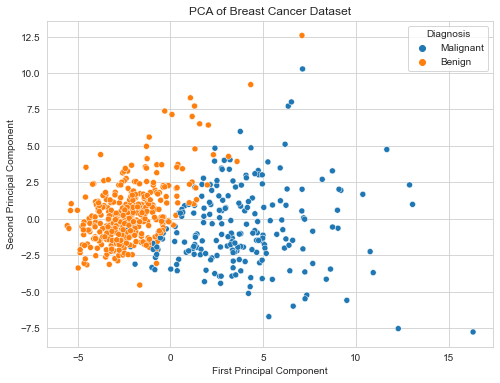
### PCA Implementation

Principal Component Analysis (PCA) was used to reduce the dimensionality of the Breast Cancer dataset (Pedregosa et al., 2011). This method simplified the conversion of the original features into a new set of features called principle components, which are linear combinations of the original features. A machine-learning model will be built using the PCA-reduced data. Logistic regression is frequently used to predict breast cancer diagnosis because of its simplicity and effectiveness. The study used the Sklearn decomposition PCA approach with two components.

**Code Snippet**

pca = PCA(n\_components=2) # Reduce dimensions to 2 for visualization or further analysis

X\_pca = pca.fit\_transform(X\_scaled)



Clustering is seen in the plot. Malignant samples, represented by the color orange, usually group closely together and show spread along the primary and secondary principal components. Non-cancerous samples (blue) show clustering, particularly along the main principal component.PCA can visually illustrate the differentiation between malignant and benign scenarios in the dataset pertaining to the diagnosis via data clustering.

### Model Development

We used various features of the Scikit-learn package, a well-known Python machine-learning framework, to prepare our data and create our machine-learning model.

We imported Scikit-learn modules in the initial stage. We partitioned our dataset into training and testing sets using the train\_test\_split function from sklearn.model\_selection. The logistic regression model was constructed using the LogisticRegression class from the sklearn.linear\_model package. We imported various performance metrics from sklearn.metrics, such as Accuracy\_score, Precision\_score, Recall\_score, F1\_score, and Confusion\_matrix, to assess our model's performance.

Once the required modules are imported, we divide our data. The dataset included the principal component features X\_pca and the target variable y. We utilized the train\_test\_split function to partition the data, reserving 30% of it for testing using test\_size=0.3. The random\_state option was assigned the value 42 to guarantee the reproducibility of our results. This is crucial for troubleshooting and evaluating models.The train\_test\_split function yielded four results: X\_train, X\_test, y\_train, and y\_test. These correspond to the training data, testing data, training labels, and testing labels, respectively. The data was utilized in the following stages of our machine-learning process, during which the model was trained and assessed.

**Code Snippet**

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

# Splitting the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_pca, y, test\_size=0.3, random\_state=42)

#Initializing and training the logistic regression model

log\_reg = LogisticRegression()

log\_reg.fit(X\_train, y\_train)

### Performance Metrics

Measuring accuracy alone in binary classification may be misleading. Therefore, we have used various evaluation metrics to assess the performance of supervised models on the dataset. These metrics include:

* **Accuracy score:** This metric calculates the accuracy of a classification model, which is the percentage of correct predictions over total predictions. In other words, it measures the model's overall correctness by measuring how many accurate predictions (both true positives and true negatives) are out of the total number of cases.
* **Precision score:** Precision measures the accuracy of the positive predictions. Specifically, it provides information about this case's percentage of accurate malignant predictions.
* **Recall score:** Recall, also known as sensitivity, computes the proportion of accurate positive predictions over the sum of true positive and false negative predictions. It measures the model’s ability to identify all relevant instances correctly.
* **F1 score:** The F1 score of a classification model is the harmonic mean of precision and recall. It provides a balanced measure of precision and recall.
* **Confusion matrix**: This matrix summarizes the correct and incorrect classifications produced by a classifier for the dataset. It provides insight into the model’s performance across different classes.

Using these metrics collectively gives a comprehensive understanding of how well the supervised models performed in classifying the data.

### Results

A diagram of a logistic regression model

Description automatically generated

*Figure 1: Confusion Matrix for Logistic Regression*

As a result of the Logistic Regression Model's confusion matrix, 60 cases were correctly predicted to be malignant. True Negative (TN): 106 cases were correctly predicted to be benign. False Positives (FP): Two cases were incorrectly predicted as malignant, whereas three cases were incorrectly predicted as benign. False Negatives (FN): Three cases were incorrectly predicted as benign when they were malignant. This model effectively distinguishes benign and malignant cases with few errors, demonstrating many true positives and negatives.

A graph of a graph showing different colored bars

Description automatically generated with medium confidence  
*Figure 2: Report for Logistic Regression*

Using the Logistic Regression Model, the following performance metrics are shown: Accuracy: 0.9708 (97.8%), Precision: 0.9677 (96.77%), Recall: 0.9524 (95.24%), and F1-Score: 0.9600 (96.00%). Based on these metrics, the Logistic Regression Model performs exceptionally well on all key evaluation aspects. It is highly reliable in identifying true malignant cases and maintains low false positive rates while accurately classifying cases.

## Discussion

Findings from the analysis and insights from the previous literature review highlight the revolutionary capacity of machine-learning methods, especially logistic regression boosted with PCA, in breast cancer diagnosis. The research findings align with earlier research that has succeeded in showing the power of machine-learning algorithms in enhancing both the level of diagnostic accuracy and the patient’s outcomes. Naji et al. (2001) provided evidence of the high degree of success of Support Vector Machine (SVM) in diagnosing breast cancer by accurately diagnosing 97.2% of the cases. Thus, it is crucial to choose suitable algorithms for logistic regression in our study that gave close performance (97.08% accuracy, with good precision (96.77%) and recall (95.24%). Further, Fatima et al. (2020) thoroughly assessed several machine-learning algorithms, including logistic regression, emphasizing algorithm choice's importance in predictive modeling. In addition, the study adds that using PCA for dimensionality reduction helps visualize data and improves model performance by concentrating on the most informative features. Results from the logistic regression model validate its ability to discriminate between benign and malignant cases, with appreciable sensitivity in identifying actual malignant cases at the expense of false positives. This function is essential in clinical settings where the correctness of diagnosis is critical for prompt intervention and treatment planning. High precision and recall of the model indicate its efficacy in recognizing important characteristics of breast cancer diagnosis, which is in line with the aim of improving early detection and reducing mortality rates.

## Conclusion and Recommendation

In conclusion, this study used predictive analytics with PCA-enhanced logistic regression to diagnose breast cancer. By analyzing the Breast Cancer Wisconsin Data Set, positive diagnoses related to significant attributes were identified and used to develop a robust predictive model. The research conducted by our group demonstrates the effectiveness of machine-learning approaches in enhancing the accuracy of breast cancer diagnosis and patient outcomes.

The logistic regression model performed well in differentiating benign and malignant cases in a range of evaluation metrics, including accuracy, precision, recall, and F1-score. The model produced a high accuracy (97.08%) and showed good precision (96.77%), recall (95.24%), and F1-score (96.00). As such, it was good in detecting the right malignant cases and minimizing false positives.

These findings underscore the advantages of predictive analytics in breast cancer diagnosis, allowing physicians to have increased knowledge of the situation and early detection and intervention. Continuing with the future direction, more feature selection approaches and algorithm adjustments must be investigated to improve prediction modeling in breast cancer diagnosis and treatment planning.

## References

* Wolberg,William, Mangasarian,Olvi, Street,Nick, and Street,W.. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. <https://doi.org/10.24432/C5DW2B>.
* Fatima, N., Liu, L., Hong, S., & Ahmed, H. (2020). Prediction of Breast Cancer, Comparative Review of Machine Learning Techniques, and Their Analysis. *IEEE Access*, *8*, 150360–150376. <https://doi.org/10.1109/access.2020.3016715>
* Naji, M. A., Filali, S. E., Aarika, K., Benlahmar, E. H., Abdelouhahid, R. A., & Debauche, O. (2021). Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis. *Procedia Computer Science*, *191*, 487–492. <https://doi.org/10.1016/j.procs.2021.07.062>
* Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

### 

## **Appendix**

#This code installs the `ucimlrepo` package.

#pip install ucimlrepo

import pandas as pd

import numpy as np

import dmba

import seaborn as sns

import matplotlib.pyplot as plt

from pandas.plotting import scatter\_matrix

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix

from sklearn.decomposition import PCA

# filter warnings

import warnings

warnings.filterwarnings("ignore")

import dmba

from dmba import classificationSummary, gainsChart, liftChart

from dmba.metric import AIC\_score

from ucimlrepo import fetch\_ucirepo

# fetch dataset

breast\_cancer\_wisconsin\_diagnostic = fetch\_ucirepo(id=17)

# data (as pandas dataframes)

X = breast\_cancer\_wisconsin\_diagnostic.data.features

y = breast\_cancer\_wisconsin\_diagnostic.data.targets

# variable information

#breast\_cancer\_wisconsin\_diagnostic.data

#create dataframe using both x and y

df = pd.concat([X, y], axis=1)

#save the data to a csv file

#df.to\_csv('breast\_cancer\_wisconsin\_diagnostic.csv', index=False)

# ### Basic information about the dataset

#

print("First 5 rows of the dataset:")

df.head()

df.shape

print("\nDataset summary:")

df.describe()

print("\nMissing values in the dataset:")

df.isnull().sum()

#print the information about the dataset

print(df.info())

# #### Data Preparation

#

#replace the target values with 0 and 1

#y = np.where(y == 'M', 1, 0)

#df['diagnosis'] = y

# ### EDA

# #### Distribution Analysis:

# Set the aesthetic style of the plots

sns.set\_style("whitegrid")

# Create a figure to hold the subplots

plt.figure(figsize=(20, 15))

# Plot histograms for each numeric feature to understand distributions

for index, column in enumerate(df.columns[:-1], 1):  # Exclude the 'diagnosis' column

    plt.subplot(6, 5, index)

    sns.histplot(df[column], kde=True, element='step', color='blue')

    plt.title(column)

plt.tight\_layout()

plt.show()

#HeatMap Correlations between the features

plt.figure(figsize=(20,20))

sns.heatmap(df.corr(), annot=True, fmt='.1f')

plt.show()

# Count plot for the 'diagnosis' to see class distribution

plt.figure(figsize=(6, 4))

sns.countplot(x='Diagnosis', data=df)

plt.title('Class Distribution')

plt.show()

model = LogisticRegression()

model.fit(X, y)

feature\_importance = pd.Series(model.coef\_[0], index=X.columns)

feature\_importance.plot(kind='barh')

plt.show()

# Normalization

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Dimensionality Reduction using PCA

pca = PCA(n\_components=2)  # Reduce dimensions to 2 for visualization or further analysis

X\_pca = pca.fit\_transform(X\_scaled)

# Explained variance ratio for PCA components

explained\_variance = pca.explained\_variance\_ratio\_

print("Explained Variance Ratio for PCA Components:", explained\_variance)

# Plotting the first two principal components

plt.figure(figsize=(8, 6))

sns.scatterplot(x=X\_pca[:, 0], y=X\_pca[:, 1], hue=y['Diagnosis'].map({'M': 'Malignant', 'B': 'Benign'}))

plt.title('PCA of Breast Cancer Dataset')

plt.xlabel('First Principal Component')

plt.ylabel('Second Principal Component')

plt.legend(title='Diagnosis')

plt.show()

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

# Splitting the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_pca, y, test\_size=0.3, random\_state=42)

# Initializing and training the logistic regression model

log\_reg = LogisticRegression()

log\_reg.fit(X\_train, y\_train)

#Predicting on the test set

y\_pred = log\_reg.predict(X\_test)

# Calculating performance metrics

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, pos\_label= 'M')

recall = recall\_score(y\_test, y\_pred, pos\_label= 'M')

f1 = f1\_score(y\_test, y\_pred,pos\_label= 'M')

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

# Visualizing the confusion matrix

plt.figure(figsize=(7, 5))

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap='Blues', cbar=False,

            xticklabels=['Predicted Benign', 'Predicted Malignant'],

            yticklabels=['Actual Benign', 'Actual Malignant'])

plt.title('Confusion Matrix of Logistic Regression Model')

plt.show()

# Visualization of Performance Metrics

metrics = ['Accuracy', 'Precision', 'Recall', 'F1-Score']

values = [accuracy, precision, recall, f1]

plt.figure(figsize=(8, 5))

barplot = sns.barplot(x=metrics, y=values, palette='viridis')

plt.ylim(0.9, 1)

plt.title('Enhanced Performance Metrics of Logistic Regression Model')

plt.ylabel('Score')

for p in barplot.patches:

    barplot.annotate(format(p.get\_height(), '.4f'),

                     (p.get\_x() + p.get\_width() / 2., p.get\_height()),

                     ha='center', va='center',

                     size=10, xytext=(0, 8),

                     textcoords='offset points')

plt.show()