Analysis and Comparison of Methods for Learning Correlations of Breast Cancer Wisconsin (Diagnostic) Dataset

Geethanjali Nallani, Jacob Rubinstein, Mehedi Galib, Sachin Phogat, Shalima Manir

***Abstract*—Our goal is to compare various machine learning algorithms to classify the Brest Cancer Wisconsin dataset from the UCI Machine Learning repository. To achieve this goal, we will first preprocess the data to remove any obvious outliers or data points without complete information. Next, we will perform exploratory data analysis to find the statistical distribution of data. Then, we perform principal component analysis (PCA) to find out correlation among different features. Next, we will train three classifiers to identify the cancer. Finally, we will evaluate the performance of each technique and analyze efficacy of these algorithms.**

***Index Terms*—Naive Bayes, Logistic Regression, KNN, Exploratory Data Analysis, PCA.**

I. Introduction

THE main motivation of this project is to analyze and compare different machine learning methods such as logistic regression [1], naive bayes [2], and K-nearest neighbor (KNN)[3] for learning the correlations between features of the Breast Cancer Wisconsin (Diagnostic) Dataset. This dataset consists of features computed from digitized images of fine needle aspirate (FNA) biopsies of breast masses. The features describe the characteristics of the cell nuclei present in the image, and the dataset is labeled as either benign or malignant.

First, we did Exploratory Data analysis on the data to familiarize ourselves with the data and to discover any correlations. We then used this information to improve our learning models. Then we performed Principal Component Analysis and t-Distributed Stochastic Neighbor Embedding on the variables. This has the purpose of examining the variables in the dataset to determine which are the most important, and to examine any trends in the variables. We then trained our three learning models on the data using this learned information.

In this report, section II describes the related work, and section III describes all the proposed method we appliedand then comes to result and discussion on section IV and finally we concludes our work.

II. Related Work

A. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a statistical approach and methodology used to analyze and understand datasets before applying formal statistical techniques. It involves visually exploring and summarizing the data to gain insights, identify patterns, detect outliers, and formulate hypotheses. EDA helps researchers and analysts familiarize themselves with the data, discover relationships between variables, and guide subsequent analysis.

According to [4], EDA focuses on understanding the structure and properties of the data, rather than making specific statistical inferences. It involves the use of graphical and numerical tools to examine data distributions, relationships, and anomalies. The aim is to generate hypotheses and insights that guide further investigation and analysis. In [5], authors emphasize the importance of graphical techniques in EDA, advocating for the use of plots, charts, and diagrams to effectively summarize and visualize data. He introduces methods such as histograms, scatterplots, box plots, and stem-and-leaf displays to represent data visually and uncover patterns or trends. Moreover, in [6], authors emphasize the exploratory nature of data analysis and the need for flexibility and creativity in approaching datasets through promoting the idea of ”data-driven thinking” and encourages analysts to let the data speak for itself, allowing unexpected patterns or outliers to spark new lines of inquiry.

Overall, [4], [5], [6] serve as a foundational reference for understanding the concept and principles of EDA for this project.

B. Principal Component Analysis (PCA) & t-Distributed Stochastic Neighbor Embedding (t-SNA)

PCA transforms a large set of variables into a smaller one that still contains most information in the large data set. The principal components are ordered in such a way that the maximum variance of data will be explained by the first component, and the preceding components explain the rest of the variance in a descending way. t-Distributed Stochastic Neighbor Embedding (t-SNE)is an iterative non-linear dimensionality reduction technique. Data points that are close to each other in the high-dimensional space will tend to be close to each other in the low-dimensional space as well. t-SNE is a non-linear methodology, whereas PCA is a linear method. Finding linear combinations of the initial variables that capture the most variance in the data is the goal of PCA. It concentrates on maintaining relationships and global structure. The local structure and connections between data points are intended to be preserved by t-SNE, which frequently reveals non-linear patterns and clusters.

Special Thanks to Dr. Abhijit Dutt for fruitful discussion related to the project

C. k-nearest Neighbors Algorithm(KNN)

K-Nearest Neighbors (KNN) is a popular classification algorithm in machine learning. It is a non-parametric method, meaning it does not make assumptions about the underlying data distribution. It classifies new data points based on their proximity to labeled data points in the feature space. The "K" in KNN represents the number of nearest neighbors considered for classification. The algorithm calculates distances, usually using Euclidean distance, between the new data point and its K nearest neighbors. The class label is determined by a majority vote among these neighbors. KNN is simple to implement, interpretable, and can handle binary and multi-class classification problems. However, choosing the right value of K is important, and computational complexity increases with larger datasets. Feature scaling and handling imbalanced datasets can also affect KNN's performance. Nonetheless, KNN is widely used and researchers are continuously exploring ways to enhance its performance.

D. Logistic Regression

Logistic regression is a statistical modeling technique used to analyze the relationship between a binary or categorical dependent variable and one or more independent variables [7]. It is widely employed in various research fields, including social sciences, health sciences, economics, and more. In logistic regression, the dependent variable is typically binary, meaning it takes on one of two possible outcomes, such as "success" or "failure," "yes" or "no," or "positive" or "negative." The goal of logistic regression is to estimate the probability of the dependent variable belonging to a particular category based on the values of the independent variables.

The logistic regression model assumes a logistic or sigmoidal relationship between the independent variables and the log-odds (also known as the logit) of the dependent variable. The log-odds represent the logarithm of the odds ratio, which is the ratio of the probability of the event occurring to the probability of it not occurring.

The logistic regression model estimates coefficients for each independent variable, representing their impact on the log-odds or probability of the dependent variable. These coefficients allow us to assess the direction and magnitude of the relationship between the independent variables and the probability of the outcome.

In practice, logistic regression involves estimating the model parameters using a maximum likelihood estimation (MLE) approach. This estimation process finds the coefficients that maximize the likelihood of observing the actual outcomes based on the model's predicted probabilities.

E. Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm that is widely used for classification tasks. It is based on Bayes' theorem, which is a fundamental concept in probability theory. Naive Bayes is called "naive" because it makes a strong assumption of independence among the features, meaning that it assumes that the presence or absence of a particular feature does not affect the presence or absence of any other feature.

The general idea behind Naive Bayes is to calculate the probability of a certain class label given a set of features. It predicts the class label by maximizing this conditional probability. The algorithm assumes that the features are conditionally independent of each other, given the class label.

There are several types of Naive Bayes classifiers, each with its own assumptions and characteristics:

Gaussian Naive Bayes: This variant of Naive Bayes assumes that the features follow a Gaussian distribution (normal distribution). It calculates the mean and standard deviation of each feature for each class and uses these parameters to estimate the probability of a feature value belonging to a particular class.

Multinomial Naive Bayes: This variant is specifically designed for discrete features, typically used when working with text data. It models the feature probabilities using a multinomial distribution. It is commonly used for tasks like document classification or text categorization, where each feature represents the frequency of a term in a document.

Bernoulli Naive Bayes: Similar to the multinomial variant, Bernoulli Naive Bayes is suitable for binary features, where each feature can take only two values (usually 0 and 1). It models the feature probabilities using a Bernoulli distribution. It is often used in sentiment analysis or spam filtering, where the presence or absence of certain words or features is important.

III. Proposed method

In this project, we applied several machine-learning techniques to our dataset. First, we did exploratory data analysis, then PCA and t-SNE dimensionality reduction techniques. Next, we applied KNN, Logistic regression and Naive Bayes algorithm to both balanced and unbalanced data.

A. Exploratory Data Analysis

The method proposed for this project involves preprocessing the data [8], performing exploratory data analysis, and applying different machine learning algorithms to learn the correlations between features and classify the dataset. The preprocessing step will involve handling missing data, scaling the features, and encoding the class labels. The exploratory data analysis will help to identify any patterns, correlations, or outliers in the dataset.

B. PCA & t-SNA

Moreover, principal component analysis (PCA) will be done to find out the correlation among different features, which will allow the implementation of dimensionality reduction (if application) [9], [10].

C. KNN

In this study, we employed the K-nearest neighbors (KNN) algorithm as a classification technique to address the task of classifying the dataset into distinct categories. The KNN algorithm is a simple yet effective non-parametric method that determines the class of a test instance based on the majority vote of its nearest neighbors in the feature space.

To begin, we divided our dataset into two subsets: the unbalanced dataset and the balanced dataset. The unbalanced dataset represents the original distribution of the data, while the balanced dataset was obtained through a data balancing technique to mitigate class imbalance issues. Both subsets were preprocessed to ensure data integrity and eliminate missing values.

For the unbalanced dataset, we selected a set of relevant features, namely "$radius_mean$" "$texture_mean$," "$smoothness_mean$," "$compactness_mean$," and "$concavity_mean$," as the predictor variables (X), while the "diagnosis" column was considered as the target variable (Y). The dataset was then split into training and testing sets using a test size of 33% and a random state of 42. Similarly, for the balanced dataset, we utilized the same set of relevant features and performed the same train-test split procedure as the unbalanced dataset.

Next, we instantiated two KNN classifiers: KNN for the unbalanced dataset and knn for the balanced dataset. The value of K was set to 5 for both classifiers, as it has been empirically shown to yield good results in various scenarios.

To evaluate the performance of the KNN algorithm, we computed several metrics including accuracy, precision, recall, and F1-score. Additionally, we generated confusion matrices to gain insights into the classification results.

D. Logistic Regression

For our Logistic Regression we used R to do the calculations. We split he data first into an unbalanced dataset and a balanced dataset. The unbalanced dataset was the original dataset after exploratory data analysis, while for the balanced dataset after exploratory data analysis we selected an equal number of benign and malignant samples. For both of these datasets we did a split between training and testing data of 80% for training data and 20% for testing data.

For both of these datasets we then used the glm function to train our Logistic Regressions. We then used the VarImp function to examine the importance of the variables on the model. We then used the predict function to make predictions on our test data using our trained models.

Next we used the ConfusionMatrix function on our predictions to calculate the confusion matrices and related accuracy scores. These scores were then used to calculate ROC curves along with corresponding AUC values for both regressions.

E. Naive Bayes

Naive Bayes is commonly used in medical applications, including the detection of breast cancer, due to its simplicity, efficiency, and good performance even with relatively small datasets. Here are some reasons why Naive Bayes is suitable for breast cancer detection:

Handling High-Dimensional Data: Breast cancer detection often involves analyzing a large number of features, such as patient age, tumor size, tumor shape, and various measurements from medical imaging. Naive Bayes can handle high-dimensional data efficiently because it assumes feature independence, which reduces the computational complexity and avoids the curse of dimensionality.

Good Performance with Small Datasets: In medical applications, obtaining large labeled datasets for training machine learning models can be challenging due to the need for expert annotations and privacy concerns. Naive Bayes is known to work well even with limited data, making it a suitable choice for breast cancer detection where acquiring large datasets can be difficult.

Fast Training and Prediction: Naive Bayes has a fast training time compared to more complex models like neural networks or support vector machines. This is particularly advantageous in medical settings where quick results are often required. Naive Bayes also has fast prediction times, making it efficient for real-time or near real-time decision-making.

Interpretable Results: Naive Bayes provides interpretable results by calculating probabilities and conditional probabilities. This can be valuable in medical applications, as doctors and medical professionals often need to understand and interpret the model's predictions and decision-making process.

Handling Categorical and Discrete Data: Naive Bayes works well with categorical or discrete data, which is often the case in medical diagnosis. For instance, patient attributes such as family history, presence of certain symptoms, or results from medical tests are often represented as discrete or categorical variables, which can be directly incorporated into a Naive Bayes classifier.

IV. Results & Discussion

A. Data-set

We have chosen the Breast Cancer Wisconsin dataset that contains ten real-valued features computed for all cell nuclei and two types of tumor classes. Two types of tumors are Benign(367 cases) and Malignant(212 cases). All features are recorded with four digits. The ten feature values are radius, texture, perimeter, area, smoothness, compactness, concavity, concave points symmetry, and fractal dimension.

B. Tools & Software

We adopt R and $PyTorch$ as the primary tools to implement our three machine-learning algorithms and preprocess the raw data. Later, after extracting the test results, we will adopt various data visualization tools such as $Matplotlib$ and $Seaborn$ to provide a better depiction of our prediction about benign or malignant cancer.

C. Exploratory Data Analysis

1) Data Cleaning of BCD: In Fig. 1, we illustrate the overview of the Wisconsin Breast Cancer data-set. The data set contains 569 samples as depicted by the number of rows, and 33 columns give us the information about the number of features. It is to be noted that, we have one column for diagnosis, one column is for reference ID. Therefore, we have 31 features in total. However, one columns contains unknown garbage data, hence, we need to do the data cleaning as illustrated in Fig. 2. Moreover, we need to change the characters types of diagnosis column 'M' and 'B' to 1 and 0 for implementing for our machine learning model.

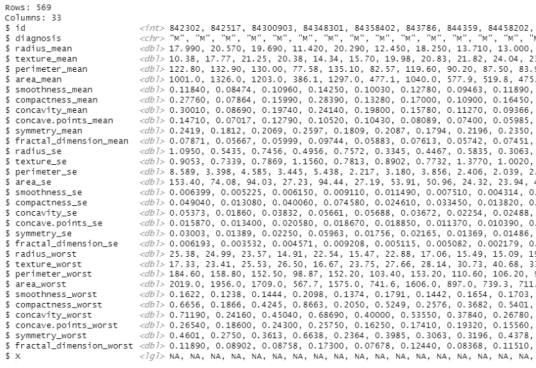


Fig. 1. Overview of Breast Cancer Dataset (BCD)

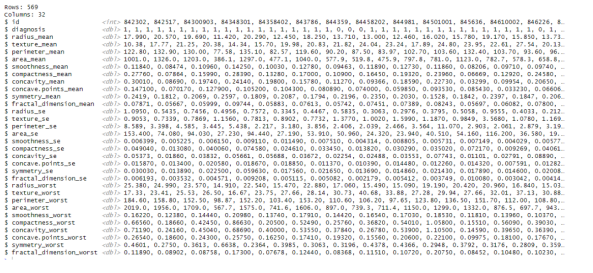
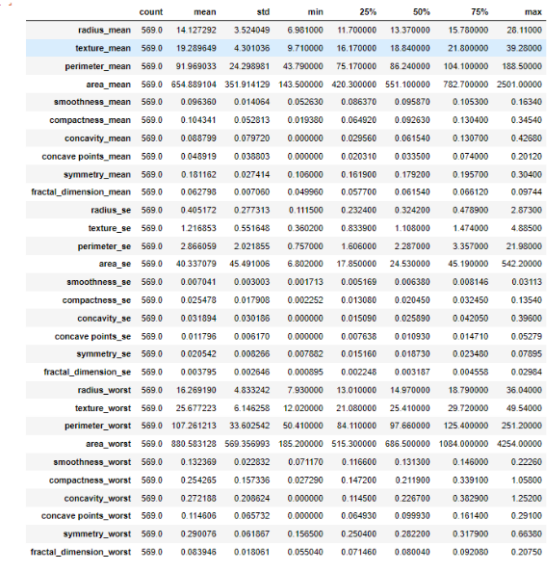


Fig. 2. Data cleaning of BCD



	count	mean	std	min	25%	50%	75%	max
radius_mean	569.0	14.127292	3.524049	6.981000	11.700000	13.370000	15.780000	28.11000
texture_mean	569.0	19.289649	4.301036	9.710000	16.170000	18.840000	21.800000	39.28000
perimeter_mean	569.0	91.969033	24.298981	43.790000	75.170000	86.240000	104.100000	188.50000
area_mean	569.0	654.889104	351.914129	143.500000	420.300000	551.100000	782.700000	2501.00000
smoothness_mean	569.0	0.096360	0.014064	0.052630	0.086370	0.095870	0.105300	0.16340
compactness_mean	569.0	0.104341	0.052813	0.019380	0.064920	0.092630	0.130400	0.34540
concavity_mean	569.0	0.088799	0.079720	0.000000	0.029560	0.061540	0.130700	0.42680
concave points_mean	569.0	0.048919	0.038803	0.000000	0.020310	0.033500	0.074000	0.20120
symmetry_mean	569.0	0.181162	0.027414	0.106000	0.161900	0.179200	0.195700	0.30400
fractal_dimension_mean	569.0	0.062798	0.007060	0.049960	0.057700	0.061540	0.066120	0.09744
radius_se	569.0	0.405172	0.277313	0.111500	0.232400	0.324200	0.478900	2.87300
texture_se	569.0	1.216853	0.551648	0.360200	0.833900	1.108000	1.474000	4.88500
perimeter_se	569.0	2.866059	2.021855	0.757000	1.606000	2.287000	3.357000	21.98000
area_se	569.0	40.337079	45.491006	6.802000	17.850000	24.530000	45.190000	542.20000
smoothness_se	569.0	0.007041	0.003003	0.001713	0.005169	0.006380	0.008146	0.03113
compactness_se	569.0	0.025478	0.017908	0.002252	0.013080	0.020450	0.032450	0.13540
concavity_se	569.0	0.031894	0.030188	0.000000	0.015090	0.025890	0.042050	0.39600
concave points_se	569.0	0.011796	0.006170	0.000000	0.007638	0.010930	0.014710	0.05279
symmetry_se	569.0	0.020542	0.008266	0.007882	0.015160	0.018730	0.023480	0.07895
fractal_dimension_se	569.0	0.003795	0.002646	0.000895	0.002248	0.003187	0.004558	0.02984
radius_worst	569.0	16.269190	4.833242	7.930000	13.010000	14.970000	18.790000	36.04000
texture_worst	569.0	25.677223	6.146258	12.020000	21.080000	25.410000	29.720000	49.54000
perimeter_worst	569.0	107.261213	33.602542	50.410000	84.110000	97.660000	125.400000	251.20000
area_worst	569.0	880.583128	569.356993	185.200000	515.300000	686.500000	1084.000000	4254.00000
smoothness_worst	569.0	0.132369	0.022832	0.071170	0.116600	0.131300	0.146000	0.22260
compactness_worst	569.0	0.254265	0.157336	0.027290	0.147200	0.211900	0.339100	1.05800
concavity_worst	569.0	0.272188	0.208624	0.000000	0.114500	0.226700	0.382900	1.25200
concave points_worst	569.0	0.114606	0.065732	0.000000	0.064930	0.099330	0.161400	0.29100
symmetry_worst	569.0	0.290076	0.061867	0.156500	0.250400	0.282200	0.317900	0.66380
fractal_dimension_worst	569.0	0.083946	0.018061	0.055040	0.071460	0.080040	0.092080	0.20750

Fig. 3. Statistics of BCD

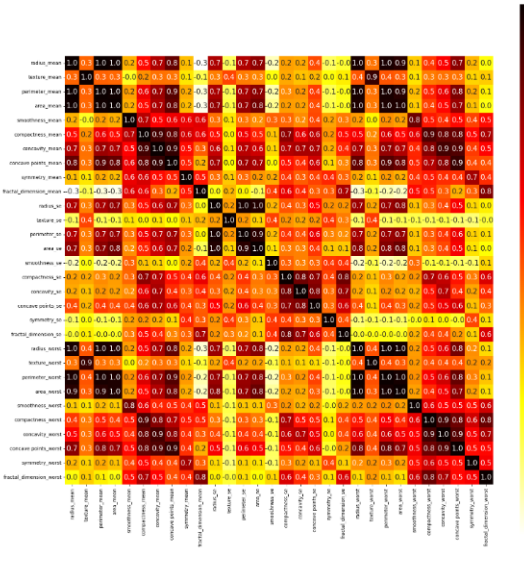
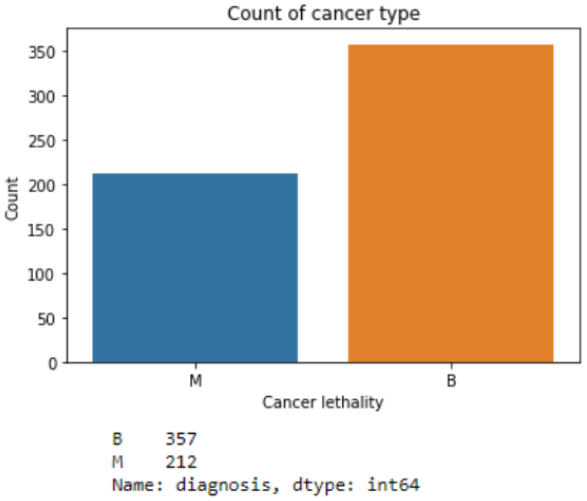


Fig. 4. Heatmap Correlation Between Features



W

Fig. 5. Dataset unbalancing

Fig. 6. Dataset balancing

Fig. 7. Histograms of Feature distribution of breast cancer dataset

2) Statistics of BCD: Fig. 3 represents statistics of different features of the data-set such as count, mean, standard deviation, minimum value, first quartile data (25%), median, third quartile data (75%), and maximum value. Through this analysis, we can easily understand the nature of distribution of our data-set. Later, we calculate the correlation analysis among different features and illustrated in Fig. 4 as heat-map. These correlation are more studies during PCA ans t-SNE analysis in following sub-section.

3) Balanced Vs Unbalanced Data-set: In our considered data-set, as shown in Fig. 5 there were 357 sample of benign cancer and 212 of the malignant cancer. Hence, in our study, we perform various machine learning technique through both balancing and unbalanced data set. For this purpose, we take all the malignant sample and on the other hand, we only take 212 sample out of 357 benign samples. The overall balanced data-set is depicted in Fig. 6, where total number of sample is 424 and number of features are 30.

Additionally, we generate histogram to understand the distribution of different features of our data sets as shown in Fig. 7. As mentioned before, more discussion related to the features is done in following subsection.

D. PCA and t-SNE Analysis

Moreover, by examining the ROC curves, in the figure 18 we can visualize the trade-off between true positive rate and false positive rate for both before and after balancing the dataset. The ROC curve for the balanced dataset showed a higher area under the curve (AUC) compared to the unbalanced dataset, indicating improved overall performance.

In figure 8 before applying PCA a feature distribution plot has been made to better understand the data distribution.

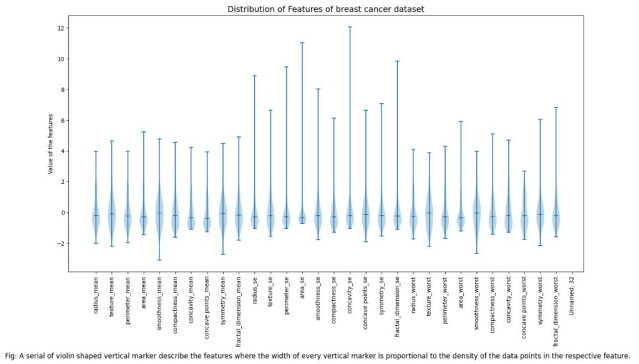


Fig. 8. Feature distribution of breast cancer dataset using Violin plot

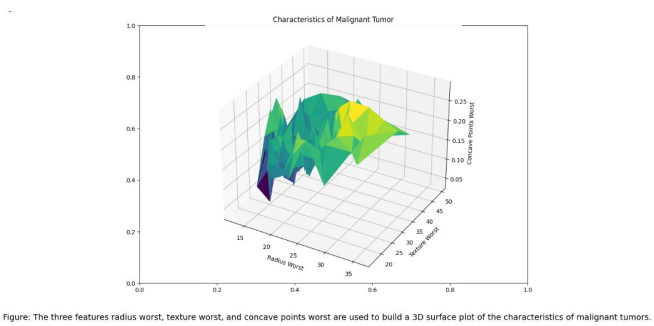


Fig. 9. Glimpse of malignant tumor features

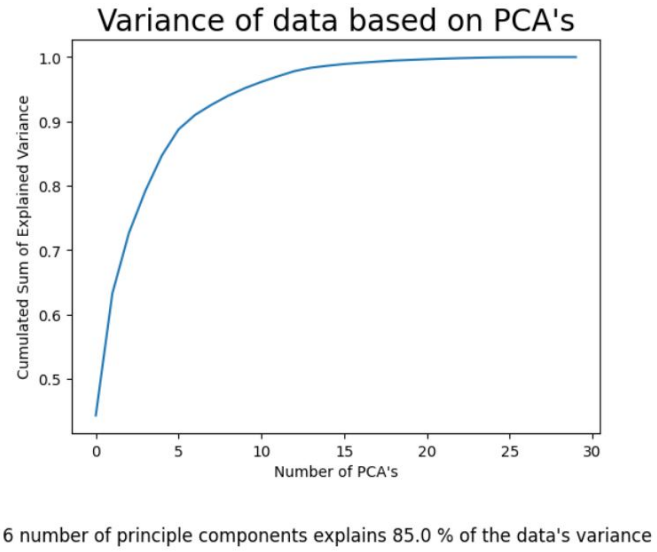


Fig. 10. Variance of data based on PCA's

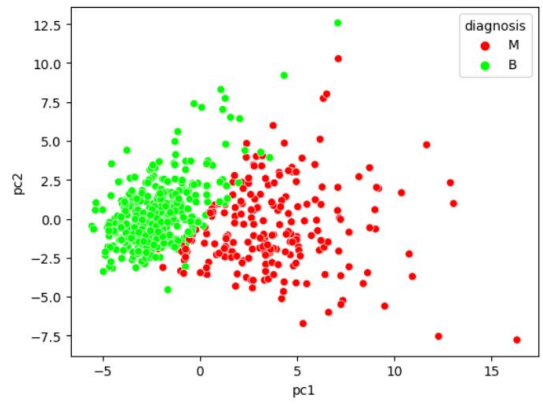


Fig. 11. PCA analysis plot for principal components 1 and 2

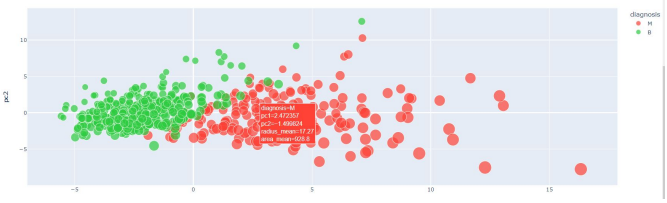


Fig. 12. PCA analysis results on hover plot

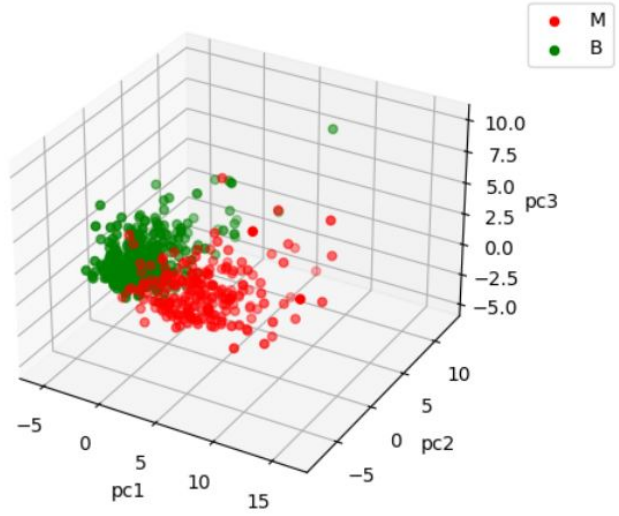


Fig. 13. 3-D plot of three principal components

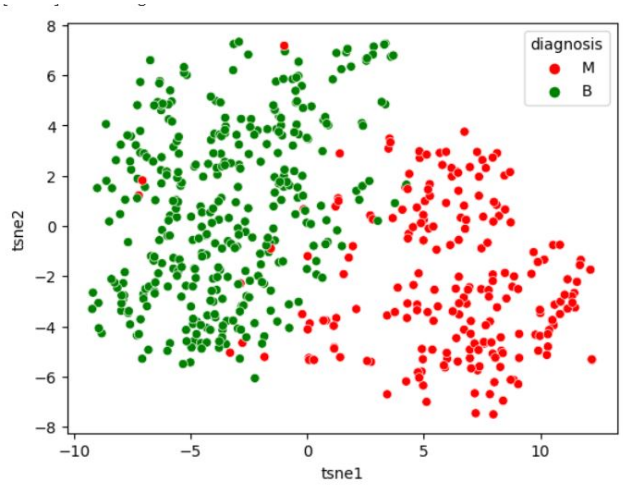


Fig. 14. t-SNE analysis plot for tsne1 and tsne2

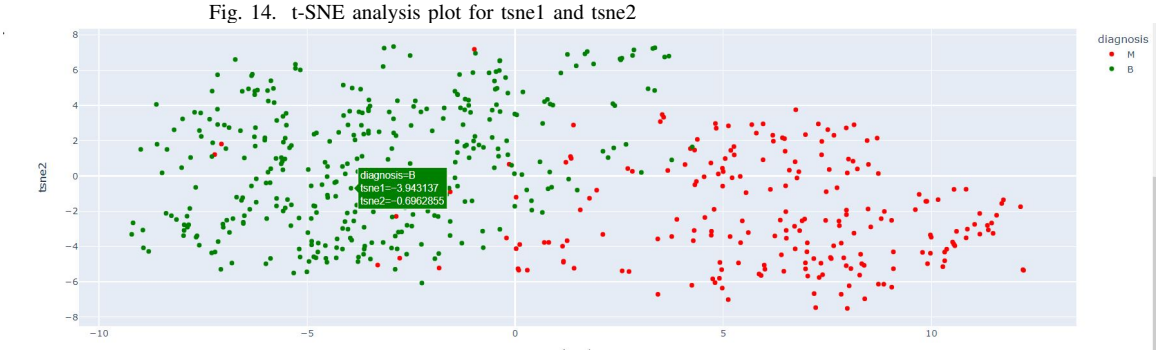


Fig. 15. Hover plot of t-SNE analysis

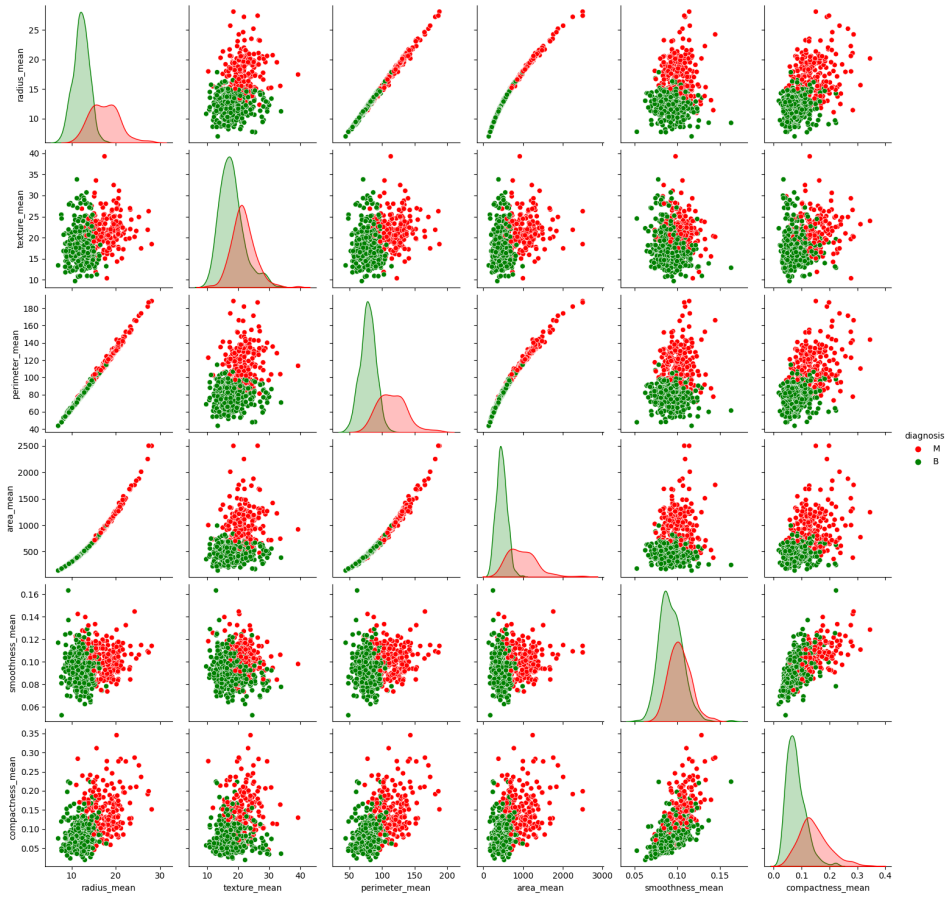


Fig. 16. Pairplot of six important features

```
Before Balancing the dataset
Recall: 0.7761194029850746
Precision: 0.8387096774193549
f1_score: 0.8062015503875968
Accuracy: 0.8670212765957447
confusion matrix:  [[111  10]
 [ 15  52]]
After Balancing the dataset
Recall: 0.8289473684210527
Precision: 0.9692307692307692
f1_score: 0.8936170212765957
Accuracy: 0.8928571428571429
confusion matrix:  [[62  2]
 [13 63]]
```

Fig. 17. KNN Evalution metrics

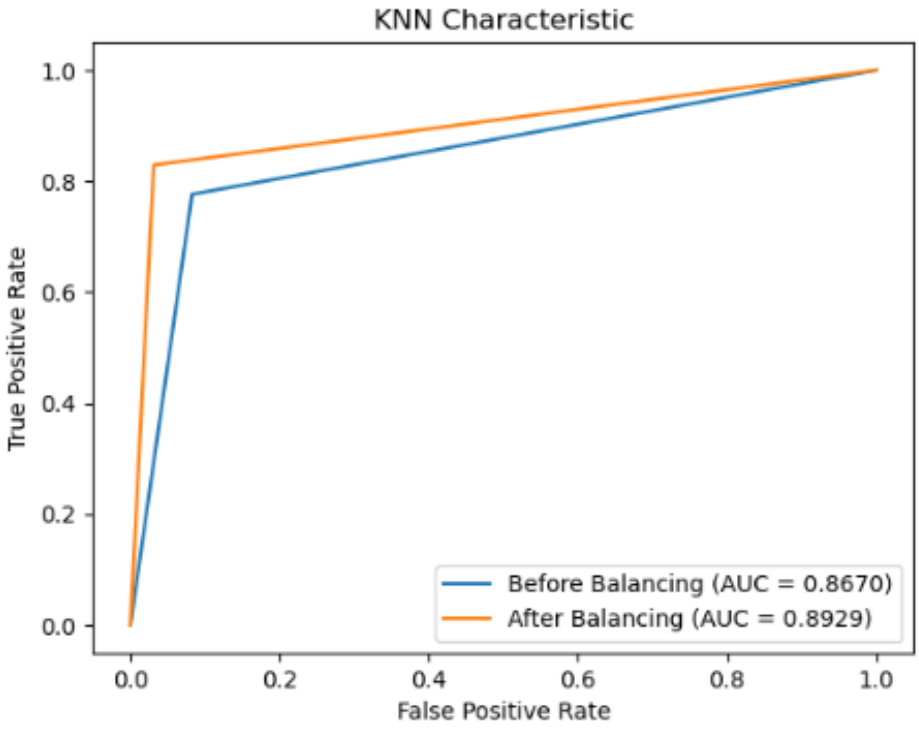


Fig. 18. The KNN ROC Curve comparing the accuracy of both the balanced and unbalanced datsets at k=5

In figure 9 a glimpse of malignant tumor features is depicted via a 3-d plotting. Figure 10 shows the variance of data based on PCA analysis. From this analysis it depicted six number of principal components explain 85 percent of the data variance. In our project, we can change the threshold to different values to see how many components we need. Figure 11 the plot for two principal components of the dataset.

Figure 12 shows a hover plot of PCA analysis it also shows the two important feature values $radiusmean$ and $areamean$ of each data point along with pc1 and pc2 while hovering.

Figure 13 depicted a 3-d scatter plot of three principal components for both Malignant and benign tumor.

In figure 14 shows t-SNE analysis for t-sne1 and t-sne2. Also a hover plot of t-SNE analysis is depicted in figure 15 for each datapoint. And a pairplot of six important features is depicted in figure 16.

E. KNN

The K-nearest neighbors (KNN) algorithm was applied to both unbalanced and balanced datasets for classification purposes. The performance of the algorithm was evaluated using various metrics, including recall, precision, F1-score, and accuracy.

We have tried running the code by varying the k values form 1 to 10 and the following is the summary that we got. For the unbalanced dataset, the results showed that the KNN

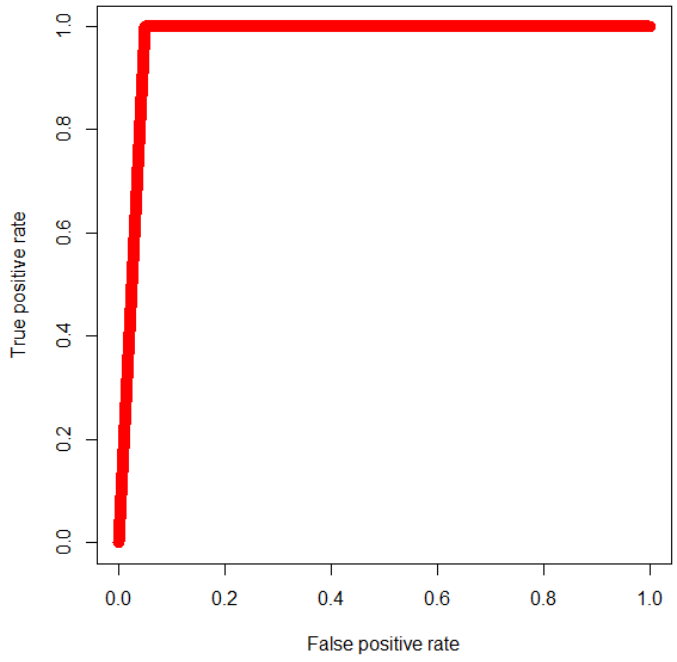


Fig. 19. ROC curve for Logistic Regression Unbalanced

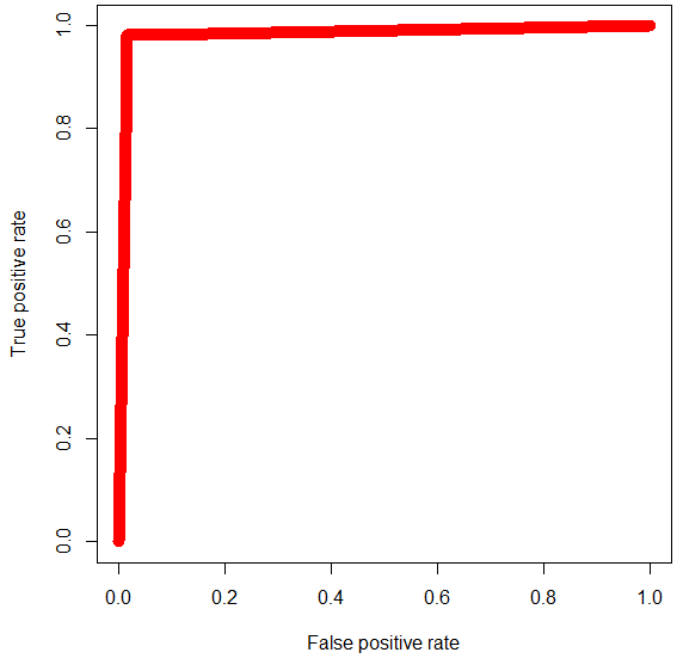


Fig. 20. ROC curve for Logistic Regression Balanced

algorithm achieved a range of recall values from 0.686 to 0.83 for different values of K. The highest recall was obtained at K=1, indicating the ability of the algorithm to correctly identify positive instances. The precision values ranged from 0.823 to 0.92, with K=2 achieving the highest precision. The F1-score values ranged from 0.786 to 0.829, and the highest F1-score was achieved at K=2. The accuracy values ranged from 0.867 to 0.882, with the highest accuracy achieved at K=6.

In contrast, for the balanced dataset, the KNN algorithm demonstrated improved performance. The recall values ranged from 0.71 to 0.828, with K=5 achieving the highest recall. This indicates the effectiveness of the algorithm in capturing a larger proportion of positive instances. The precision values ranged from 0.898 to 0.981, with K=2 achieving the highest precision. The F1-score values ranged from 0.824 to 0.893, and the highest F1-score was obtained at K=5. The accuracy values ranged from 0.835 to 0.892, with K=5 achieving the

highest accuracy.

Comparing the results between the unbalanced and balanced datasets, it is evident that balancing the dataset has a positive impact on the performance of the KNN algorithm. The balanced dataset achieved higher recall, precision, F1-score, and accuracy values compared to the unbalanced dataset for various values of K. This suggests that balancing the dataset helps mitigate the bias towards the majority class and improves the overall predictive capability of the KNN algorithm.

Overall, the optimal value of K differed between the unbalanced and balanced datasets. For the unbalanced dataset, K=2 yielded the highest precision and F1-score, while K=1 achieved the highest recall. The highest accuracy was obtained at K=6. In contrast, for the balanced dataset, K=2 achieved the highest precision, while K=5 yielded the highest recall, F1-score, and accuracy.

So, after careful evaluations we came to conclusion to consider the value of k to be 5 and the corresponding metrics output is shown in the figure and their corresponding outputs for k=5 is shown in the figure 17.

These results emphasize the importance of carefully considering the class distribution and the impact of data balancing techniques when using the KNN algorithm. Balancing the dataset can help enhance the algorithm's performance and improve its ability to classify instances accurately.

	Overall
id	0.00241
radius_mean	0.01201
texture_mean	0.00098
perimeter_mean	0.00772
area_mean	0.01220
smoothness_mean	0.02123
compactness_mean	0.01007
concavity_mean	0.00225
concave.points_mean	0.00391
symmetry_mean	0.00872
fractal_dimension_mean	0.00770
radius_se	0.00386
texture_se	0.00133
perimeter_se	0.00602
area_se	0.00555
smoothness_se	0.00202
compactness_se	0.02068
concavity_se	0.02142
concave.points_se	0.01056
symmetry_se	0.00766
fractal_dimension_se	0.01732
radius_worst	0.00536
texture_worst	0.00686
perimeter_worst	0.00101
area_worst	0.00575
smoothness_worst	0.00615
compactness_worst	0.00645
concavity_worst	0.01717
concave.points_worst	0.00354
symmetry_worst	0.01186
fractal_dimension_worst	0.00868

TABLE I
LOGISTIC REGRESSION UNBALANCED DATA VARIABLE IMPORTANCE

F. Logistic Regression Results

We trained the Logistic Regression model on both the balanced and unbalanced data. Both regressions were trained on all of the labels. Table I shows the variable importance for regression trained on the unbalanced data, while Table II shows the variable importance for the regression trained on the balanced data.

These trained classifiers were then used to make predictions on the test data we had set aside. Table IV-F shows the confusion matrix for the regression trained on the unbalanced data, while Table IV-F shows the confusion matrix for the regression trained on the balanced data.

The information from these confusion matrices was then used to calculate information such as Accuracy, Kappa, and P-Values. Table V shows these values for the regression trained on the unbalanced data, while Table VI shows these values for the the regression trained on the balanced data.

ROC curves and associated AUC values were also calculated for both of the Logistic Regression models. Fig. 19 shows the ROC curve for the regression trained on the unbalanced data, while Fig. 20 shows the ROC curve for the regression trained on the balanced data. The AUC value associaetd with the unbalanced data is 0.975409836065574 and the ROC value associated with the balanced data is 0.981596040828951.

In these particular trainings of the model, the balanced dataset performed better than the unbalanced dataset in nearly all metrics. This was not true in all trainings of the models. The accuracy of the unbalanced model varied wildly, while the accuracy of the balanced model was very consistent. We attribute these results to the fact that the balanced dataset was much smaller than the untrained dataset, meaning that the variance in the training and testing split has the potential to play a very large role in the accuacy of the model.

G. Naive Bayes

The accuracy of Naive Bayes on the Wisconsin Breast Cancer dataset can vary depending on various factors such as the preprocessing of the data, feature selection, and the specific implementation of Naive Bayes. However, it has been reported to achieve relatively high accuracy on this dataset.

In general, studies have reported Naive Bayes achieving accuracy rates ranging from approximately 0.90 to 0.97 on the Wisconsin Breast Cancer dataset. These accuracy rates demonstrate the effectiveness of Naive Bayes for breast cancer detection.

From our training and testing we got the accuracy of 0.92 for unbalanced dataset and 0.91 for balanced dataset.

V. CONCLUSION

We did EDA Analysis to perform data cleaning, statistical analysis, and data balancing. Then we performed PCA and t-SNE analysis to capture the important features. Applying Logistic Regression we get the accuracy: 0.97 for the unbalanced dataset and 0.89 for the balanced dataset. KNN classification accuracy: 0.86 for the unbalanced dataset and 0.89 for the balanced dataset. And finally applying Naive Bayes we get classification accuracy: 0.92 for the unbalanced dataset and 0.91 for the balanced dataset.

	Overall
id	0.0001
radius_mean	0.0039
texture_mean	0.0034
perimeter_mean	0.0045
area_mean	0.0014
smoothness_mean	0.0026
compactness_mean	0.0024
concavity_mean	0.0021
concave.points_mean	0.0007
symmetry_mean	0.0015
fractal_dimension_mean	0.0021
radius_se	0.0000
texture_se	0.0039
perimeter_se	0.0007
area_se	0.0017
smoothness_se	0.0003
compactness_se	0.0003
concavity_se	0.0024
concave.points_se	0.0022
symmetry_se	0.0025
fractal_dimension_se	0.0035
radius_worst	0.0030
texture_worst	0.0027
perimeter_worst	0.0004
area_worst	0.0017
smoothness_worst	0.0007
compactness_worst	0.0002
concavity_worst	0.0004
concave.points_worst	0.0001
symmetry_worst	0.0031
fractal_dimension_worst	0.0026

TABLE II
LOGISTIC REGRESSION BALANCED DATA VARIABLE IMPORTANCE

	0	1
0	58	0
1	3	53

TABLE III
LOGISTIC REGRESSION UNBALANCED DATA CONFUSION MATRIX

	0	1
0	41	5
1	5	34

TABLE IV
LOGISTIC REGRESSION BALANCED DATA CONFUSION MATRIX

	Measure	Value
1	Accuracy	0.97
2	Kappa	0.95
3	AccuracyLower	0.93
4	AccuracyUpper	0.99
5	AccuracyNull	0.54
6	AccuracyPValue	0.00
7	McnemarPValue	0.25

TABLE V
LOGISTIC REGRESSION UNBALANCED DATA CONFUSION MATRIX INFORMATION

	Measure	Value
1	Accuracy	0.88
2	Kappa	0.76
3	AccuracyLower	0.79
4	AccuracyUpper	0.94
5	AccuracyNull	0.54
6	AccuracyPValue	0.00
7	McnemarPValue	1.00

TABLE VI
LOGISTIC REGRESSION BALANCED DATA CONFUSION MATRIX INFORMATION

VI. MISCELLANEOUS

A. Data Source

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

REFERENCES

[1] S. Nusinovici, Y. C. Tham, M. Y. C. Yan, D. S. W. Ting, J. Li, C. Sabanayagam, T. Y. Wong, and C.-Y. Cheng, "Logistic regression was as good as machine learning for predicting major chronic diseases," *Journal of clinical epidemiology*, vol. 122, pp. 56–69, 2020.

[2] J. Wolfson, S. Bandyopadhyay, M. Elidrisi, G. Vazquez-Benitez, D. M. Vock, D. Musgrove, G. Adomavicius, P. E. Johnson, and P. J. O'Connor, "A naive bayes machine learning approach to risk prediction using censored, time-to-event data," *Statistics in medicine*, vol. 34, no. 21, pp. 2941–2957, 2015.

[3] Z. Zhang, "Introduction to machine learning: k-nearest neighbors," *Annals of translational medicine*, vol. 4, no. 11, 2016.

[4] J. W. Tukey *et al.*, *Exploratory data analysis*. Reading, MA, 1977, vol. 2.

[5] E. J. Sweetlin and S. Saudia, "Exploratory data analysis on breast cancer dataset about survivability and recurrence," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 2021, pp. 304–308.

[6] D. C. Hoaglin, "John w. tukey and data analysis," *Statistical Science*, pp. 311–318, 2003.

[7] R. E. Wright, "Logistic regression." 1995.

[8] S. A. Alasadi and W. S. Bhaya, "Review of data preprocessing techniques in data mining," *Journal of Engineering and Applied Sciences*, vol. 12, no. 16, pp. 4102–4107, 2017.

[9] A. George and A. Vidyapeetham, "Anomaly detection based on machine learning: dimensionality reduction using pca and classification using svm," *International Journal of Computer Applications*, vol. 47, no. 21, pp. 5–8, 2012.

[10] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.