Breast Cancer Analysis and Detection

**ABSTRACT**

*Cancer, particularly breast cancer, remains a significant health concern. This document explores the use of machine learning techniques for breast cancer diagnosis. By analyzing fine needle aspiration images and extracting cell characteristics, we aim to differentiate between benign and cancerous cells. The Breast Cancer Wisconsin dataset from the UCI machine learning repository serves as the basis for our analysis. Various data preparation steps, including outlier detection and feature engineering, are applied to improve model performance. Our investigation includes exploratory analysis, addressing outliers, handling multicollinearity, and feature selection. We delve into modeling, considering various classifiers and cross-validation techniques. Finally, hyperparameter tuning is performed to optimize model performance. Our findings shed light on the potential of machine learning in breast cancer detection.*

1. **INTRODUCTION**

As most of us know, cancer is an uncontrolled growth of the cells in each area, and if that place is the breast, it causes breast cancer. Based on an imaging procedure called Fine Needle Aspiration, an expert will classify the cells as malignant or benign. But how can we diagnose breast cancer with machine learning? That's the question of the hour.

With image processing techniques or manual measurements, cell characteristics are measured from Fine Needle aspiration images. These characteristics will be used to classify the cells into Benign and Cancerous. A few more specifics are given about the datasets.

We use the Breast Cancer Wisconsin dataset from UCI machine learning repository:

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

Data File: breast-cancer-wisconsin.data (class: 2 for benign, 4 for malignant)

Data Metafile: breast-cancer-wisconsin.names

1. **DATASET DESCRIPTION**
2. **DATA CLEANING TECHNIQUES**
   1. **OUTLIER DETECTION**

In the dataset, most of the Standard error features and factual dimensional features have significant outliers. Outliers affect the machine Learning model's predictions. There are several techniques to handle outliers. As our used data has Multivariate feature relations, we tried to apply some Multivariate Features Outlier Handling Methods as mentioned below:

1. Parametric Methods:

Elliptic Envelope

1. Non-Parametric Methods:

* DBSCAN
* Local Outlier Factor
* Isolation Forest

Here, we applied all automatic or unsupervised outlier detection algorithms. To understand their performance, we chose ordinary simple regression fitting and compared the accuracy of original and cleaned data.

Based on the two provided Figure4 and Figure5 we can say the following:

Figure4: Bar Chart (Skewness, Kurtosis, and Accuracy Comparison): The bar chart shows the accuracy values for each algorithm on a line plot overlaid on the same graph as the skewness and kurtosis bars. The accuracy values are explicitly annotated on the graph.

Figure5: Scatter Plots (Outlier Detection with Accuracy): The scatter plots show the results of the outlier detection algorithms on a dataset, with accuracy values provided in the titles of each subplot.

*Figure 4: Comparison of skewness and accuracy between algorithms.*

A graph with numbers and a line

Description automatically generated

*Figure 4: Comparison of skewness and accuracy between algorithms.*

A collage of multiple images of a number of dots

Description automatically generated with medium confidence

By comparing the accuracy values from both images:

Elliptic Envelope: 0.82 (Bar Chart), 0.70 (Scatter Plot)

DBSCAN: 0.83 (Bar Chart), 0.69 (Scatter Plot)

Local Outlier Factor: 0.82 (Bar Chart), 0.62 (Scatter Plot)

Isolation Forest: Not annotated in the Bar Chart, 0.66 (Scatter Plot)

Original (Baseline): Not applicable in Scatter Plot

The DBSCAN algorithm shows the highest accuracy value in the bar chart (0.83), which is different from the scatter plot accuracy (0.69).

Analyzing both the plots the Elliptic Envelope algorithm should be selected since it has the highest accuracy according to the scatter plots and has a very close value to the highest value for the bar chart.

* 1. **CORRELATION AND MULTIMODALITY**

Figure6 is a heatmap of a correlation matrix for the features in the Breast Cancer dataset. A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. The value is in the range of -1 to 1. If two variables have a high correlation, they may have a linear relationship.

*Figure 6: Correlation matrix and Multicollinearity.*

A screenshot of a computer screen

Description automatically generated

Here’s how to interpret the heatmap:

* Color Coding: Typically, red, or darker colors represent a higher positive correlation, and blue or lighter colors represent a higher negative correlation. White or neutral colors represent no correlation.
* Correlation Coefficient Values: These values are within the cells, indicating the strength and direction of the relationship:
* 1 or -1: Perfect positive or negative linear relationship.
* Close to 0: No linear relationship.
* Positive Value: Positive linear relationship.
* Negative Value: Negative linear relationship.

From the correlation plot, we can see that there are plenty of highly correlated data, so we can safely say we have multicollinearity in our data. But what is collinearity? This could be a big issue if we take a simple regression model like linear regression, as the highly correlated features highly influence the model weights and predictions.

We can take the high road to circumvent this issue with a better model or other techniques. Here we apply the following two scenarios and select the one that gives the best AUC\_ROC score.

The first idea is to use the Variance Inflation Factor over the dataset and remove features that have VIF > threshold. This is kind of iterative in nature. Go on till all the features VIF is less than our threshold. This threshold varies based on situation and application, and the general value of the threshold is 5.

The second idea is to use permutation feature-important techniques to remove these redundant features.

From Figure7, we can see that permutation feature selection did well on our data based on AUC\_ROC metrics. So now we try to find and select features based on metric roc\_auc.

*Figure 7: Linear model Performance comparison with PFI, VIF, and Original Data.*

A diagram of red and blue dots

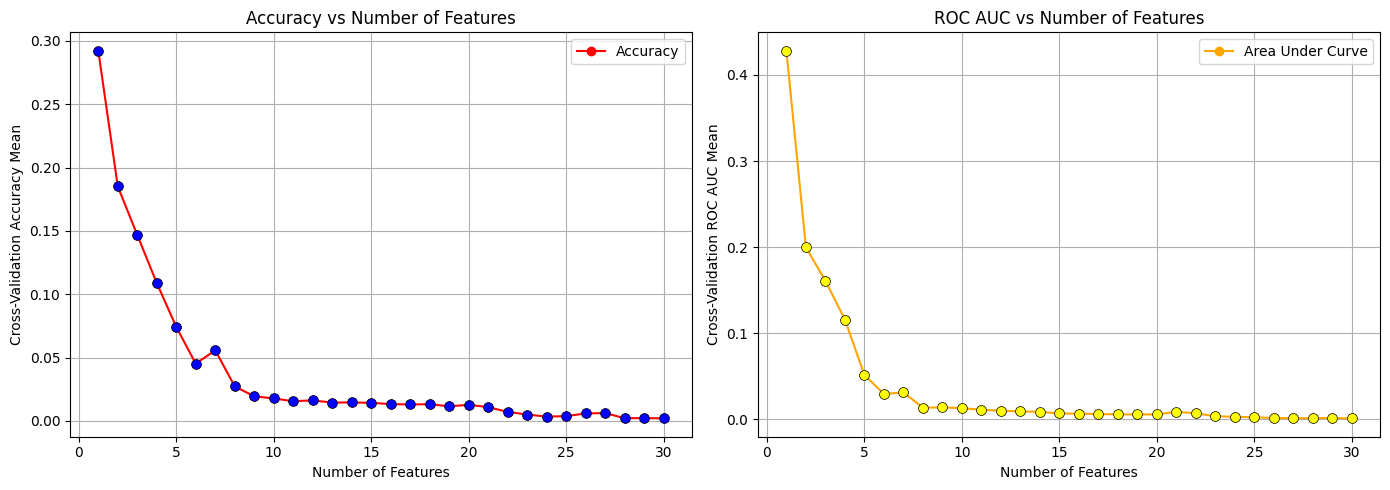
Description automatically generated

1. **FEATURE ENGINEERING AND FEATURE CONTRIBUTION**

From the above section, we found that permutation did a good job. Now we apply Xgboost on the data set detected after performing permutation feature selection and see Feature Importance with the roc\_auc metric for this technique.

From Figure8 the Accuracy vs Number of Features graph on the left shows the relationship between the number of features used in the model and the accuracy of the model. The horizontal axis represents the number of features, ranging from 0 to over 30. The vertical axis represents the cross-validation accuracy mean, ranging from 0 to about 0.3. The plot has a line marked with red color and labeled "Accuracy." From the plot, we can see that the accuracy is highest when the number of features is low (around 1 to 5). As the number of features increases, the accuracy rapidly decreases and then levels off, indicating that adding more features beyond a certain point does not significantly improve or may even reduce the model's accuracy.

*Figure 8: Influence of Number of Features on Metric.*



The ROC AUC vs Number of Features graph on the right shows the relationship between the number of features and the area under the ROC curve (AUC), which is a performance measurement for classification problems. The horizontal axis is the same as the first graph, representing the number of features. The vertical axis represents the cross-validation ROC AUC mean, which also ranges from 0 to a value that appears to be just under 0.5. The plot has a line marked with orange color and labeled "Area Under Curve." Like the accuracy graph, the AUC is highest when using a low number of features and decreases sharply as more features are added. The AUC then levels off, indicating that additional features do not improve the model's ability to distinguish between classes.

Both graphs suggest that the model performs best with a small number of features. This could indicate that many features contribute to noise or are not informative, and the model benefits from feature selection or dimensionality reduction. The leveling off of both accuracy and AUC suggests that after a certain point, adding more features does not contribute to model performance and could be a sign of overfitting or irrelevant feature inclusion.

In addition, we draw the SHAP value in Figure9. This visualization enables us to understand the feature's importance and global interpretation.

*Figure 9: SHAP Values and Features.*

A graph of blue and pink lines

Description automatically generated

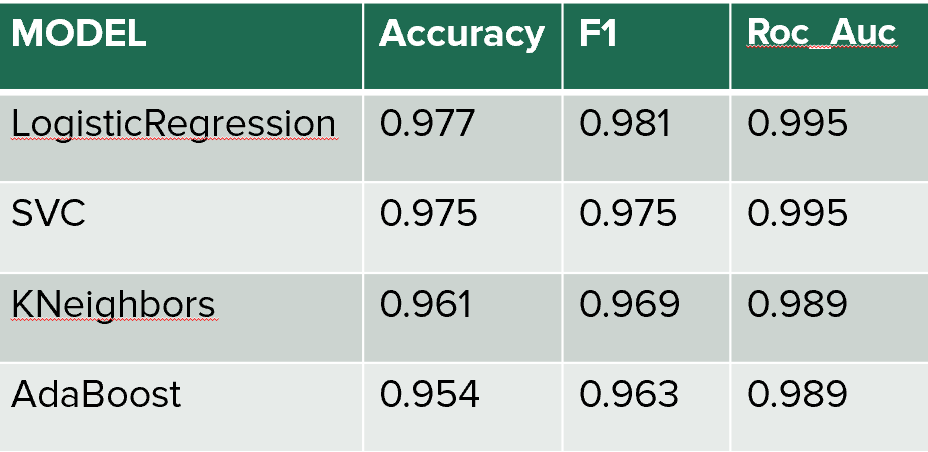
1. **EXPERIMENTAL RESULTS**

We have explored all the fundamentals in the machine learning pipeline until now. Now we will make predictions with different ML models. The idea is something like this.

1. List append a few base classifiers and get cross-validation scores,
2. Perform hyperparameter tuning on these five models

Logistic Regression, SVC, KNeighbors, AdaBoost, RandomForest, GradientBoosting, ExtraTrees, and XGBoost were used in our experiment and the result is shown in Figure 10 and Figure 11. Figure 12 and Figure 13 shows the confusion matrix of these models.

*Figure 10: Result Table 1.*

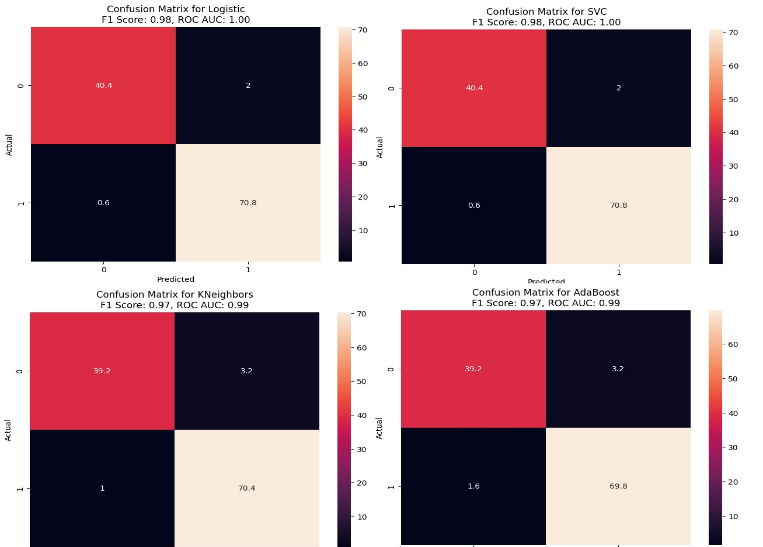


*Figure 11: Result Table 2.*

A table with numbers and text

Description automatically generated

*Figure 12: Confusion matrix1.*



*Figure 13: Confusion matrix2.*

A screenshot of a computer screen

Description automatically generated

1. **HYPERPARAMETER TUNING**
   1. **EXPLORING HYPERPARAMETER TUNING**

Hyperparameters play a crucial role in creating a robust and accurate model. Hyperparameters help us balance between bias and variance, and in this way, they cooperate to prevent overfitting or underfitting the model. To tune hyperparameters, we employ Grid search CV implementation with predefined grid params in this work.

The result of these tuned classifiers is shown in Figure 14 and Figure 15.

*Figure 14: Confusion matrix1.*

A screenshot of a computer screen

Description automatically generated

*Figure 15: Confusion matrix2.*

A screenshot of a computer screen

Description automatically generated

* 1. **STACKED ENSEMBLED CLASSIFICATION**

Ensembling is a way of supervised learning, using multiple learning algorithms to obtain an optimal predictive model. It combines different sets of supervised learners to get an improved predictive power model. It has been established that two optimization (or machine learning) algorithms are equivalent when their contribution is averaged out of all possible problems.

In our work, we make our ensemble classifier ensembling  **LogisticRegression, SVC , RandomForestClassifier, AdaBoostClassifier.** Our stacked classifier gained an **Accuracy of  0.98%,  Roc\_auc of  1.00% and  F1 scores of 0.98** %. We also visualized the results of stacked classification and their respective results in Figure 16.

*Figure 16: Confusion matrix for ensembled result.*

A diagram of a graph

Description automatically generated with medium confidence

1. **CONCLUSION**

In conclusion, the analysis presented in this document underscores the significant potential of machine learning in the early detection and diagnosis of breast cancer. Through comprehensive data preparation, including the elimination of outliers and careful feature engineering, the study has demonstrated that machine learning models can effectively distinguish between benign and malignant cells based on features extracted from Fine Needle Aspiration images.

The use of the Breast Cancer Wisconsin dataset has provided a robust foundation for exploratory analysis and the application of various classification algorithms. Addressing outliers with techniques like the Elliptic Envelope has been shown to enhance model performance significantly. Furthermore, the study effectively illustrates the necessity of handling multicollinearity, where the application of the Variance Inflation Factor and permutation feature importance has streamlined the feature set, thus bolstering the model's predictive accuracy.

Feature importance analysis, particularly using SHAP values, has offered profound insights into the contribution of individual features to model predictions, facilitating a deeper understanding of model behavior. The evaluation of different classifiers via cross-validation has not only validated the efficacy of the models but also set the stage for rigorous hyperparameter tuning, which is critical to optimizing model performance.

The experimental results, encapsulated in various figures throughout the document, paint a clear picture of the classifiers' performance, with confusion matrices providing tangible evidence of their predictive capabilities. Hyperparameter tuning further refines the models, ensuring that they are neither overfitting nor underfitting, thereby enhancing their generalizability.

Overall, this study lays the groundwork for future research, particularly in refining machine learning models for clinical applications. The promising results invite continued exploration into more sophisticated algorithms, larger datasets, and real-world clinical validation, with the goal of deploying these models in medical settings to aid in the timely and accurate diagnosis of breast cancer.

1. **FUTURE WORK**

In the future, we can use interpretable Machine Learning to remove the myth of the Black-box model. We can further implement the significance of this study in real life implementing knowledge distillation. A global model can be made using federated learning from the knowledge of this study.