

**Breast Cancer Diagnosis with Machine Learning**

By DAWRA Tony, ABDELBAKI Kayan &SLEIMAN Rebecca

Rubric

* Introduction
* Dataset
  + Dividing into X and Y variables
  + Overview of the dataset’s features and targets
* Data visualization
* Preprocessing
  + Encoding
  + Handling class imbalance
  + Feature scaling, standardization
* Feature Selection
  + Correlation analysis
  + PCA
* Model Training and Evaluation
  + Model selection and regularization
  + Cross-validation results
  + Hyper parameter tuning
* Results and Comparison
* Conclusion
  + Summary of results
  + Potential improvements.

1. **Introduction**

Breast cancer is a significant global health challenge, affecting millions of individuals worldwide. Early and accurate diagnosis is critical for improving patient outcomes. With advancements in artificial intelligence, machine learning (ML) techniques have emerged as powerful tools to enhance diagnostic accuracy. This study explores the application of ML algorithms to classify breast tumors as benign or malignant using the Breast Cancer Wisconsin Diagnostic dataset. Through data preprocessing, feature selection, model training, and hyper parameter tuning, this report aims to identify the most effective model for breast cancer diagnosis.

1. **Dataset** 
   1. **Source**

The Breast Cancer Wisconsin Diagnostic dataset, sourced from the University of Wisconsin Hospitals, Madison, provides cytological data obtained from fine-needle aspirates of breast masses. This dataset is widely recognized for its contributions to cancer research and ML applications.

**2.2. Structure**

- Features (X): Thirty numerical attributes derived from tumor images, such as radius, texture, and smoothness.

- Target (y): Diagnosis outcomes labeled as M (malignant) and B (benign).

- Check for missing values in both the feature set X and the target variable y. This step is essential for data preprocessing to avoid issues during model training caused by incomplete data.

**2.3 Dividing into X and Y**

Dividing the dataset into input features X and output targets Y is essential for:

- Model Training: Ensuring separation of predictors and the target variable for supervised learning.

- Data Manipulation: Allowing independent preprocessing of features and targets.

- Evaluation: Comparing model predictions against true outputs.

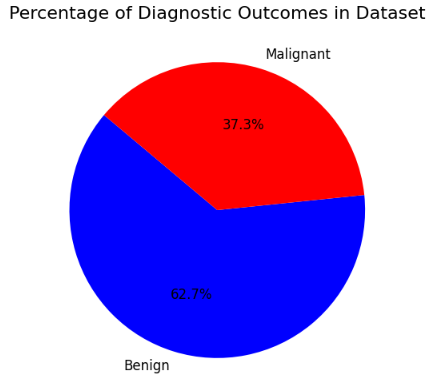
**2.4 Splitting the Data**

The dataset was split into training and testing subsets using stratified sampling to preserve class distribution:

- Training Set: 80% for model learning.

- Test Set: 20% for performance evaluation.

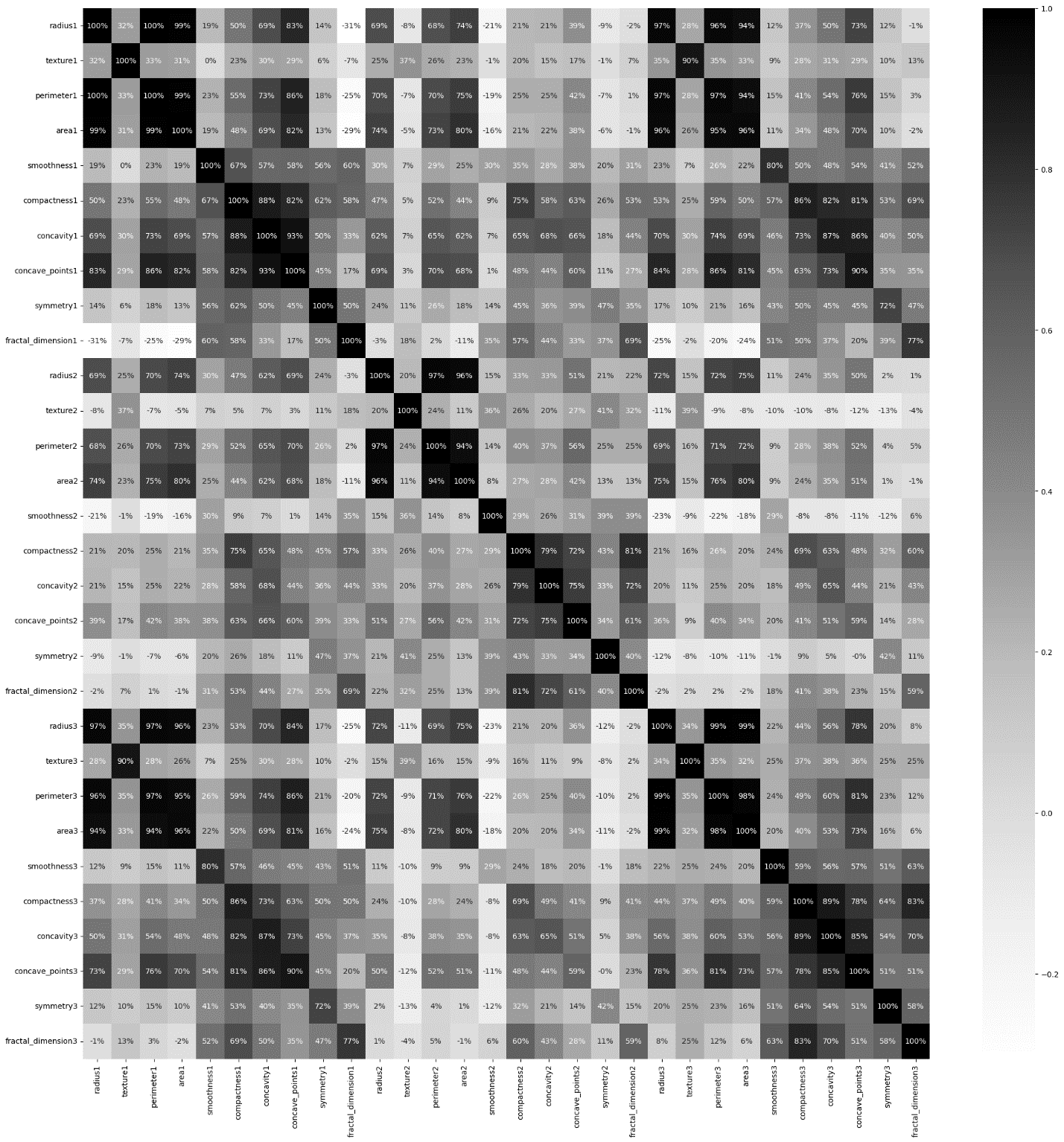
1. **Data Visualization**

**3.1. Class Distribution**

The dataset comprises 62.74% benign and 37.26% malignant cases, indicating a class imbalance. Visualized using a pie chart, this imbalance necessitates mitigation techniques to avoid biased predictions.

**3.2. Correlation Analysis**

A heat map of feature correlations highlighted strong linear relationships among certain features, suggesting potential redundancies. Addressing this was essential to improve computational efficiency and model interpretability.



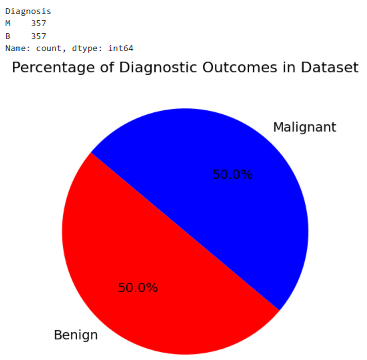
**4. Preprocessing**

**4.1. Encoding the Target Variable**

The target variable ‘B’ and ‘M’ was encoded into binary values: benign = 0, malignant = 1, using Label Encoder. This transformation enabled compatibility with ML algorithms while preserving the ordinal relationship between classes.

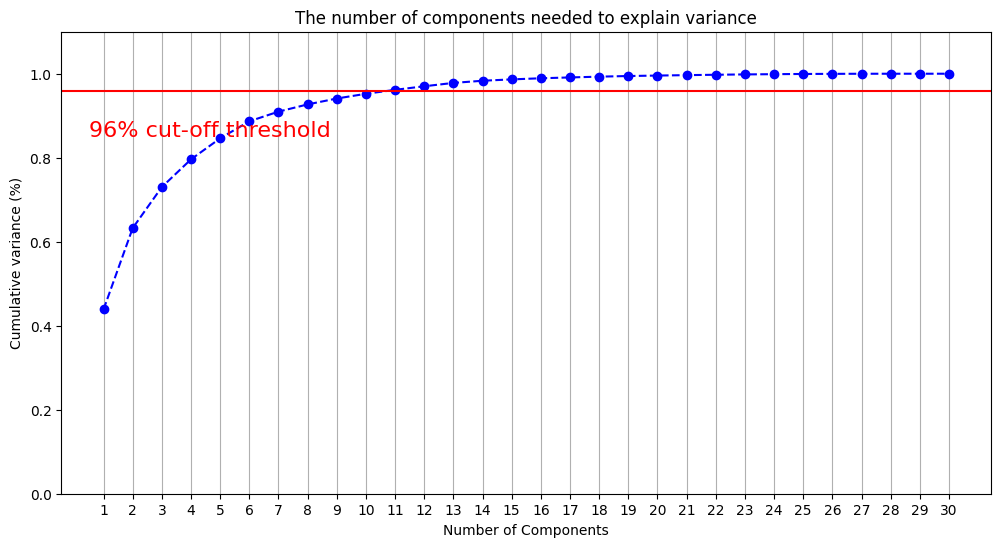
**4.2 Handling Class Imbalance**

Random Over-Sampling was employed to balance the class distribution, ensuring that both benign and malignant cases were equally represented in the training set.

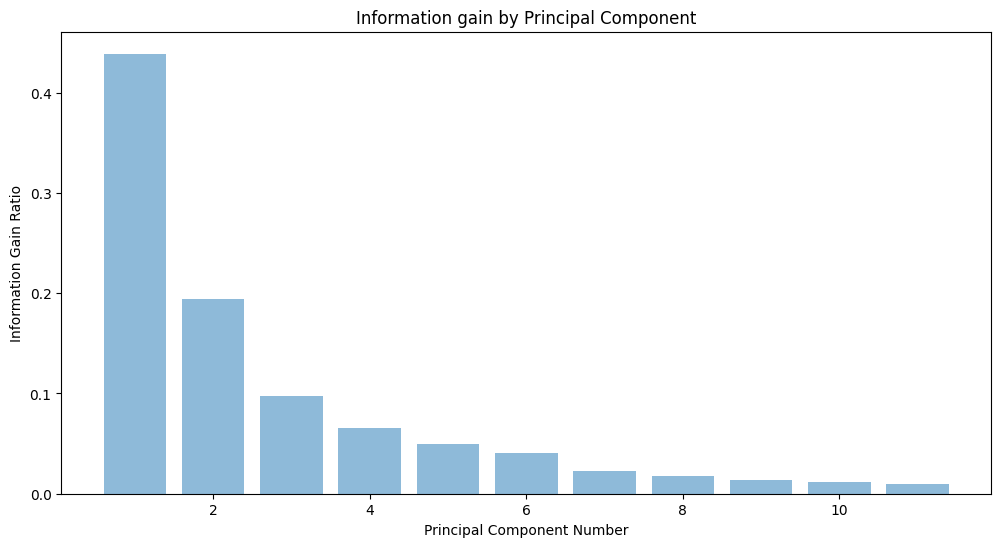


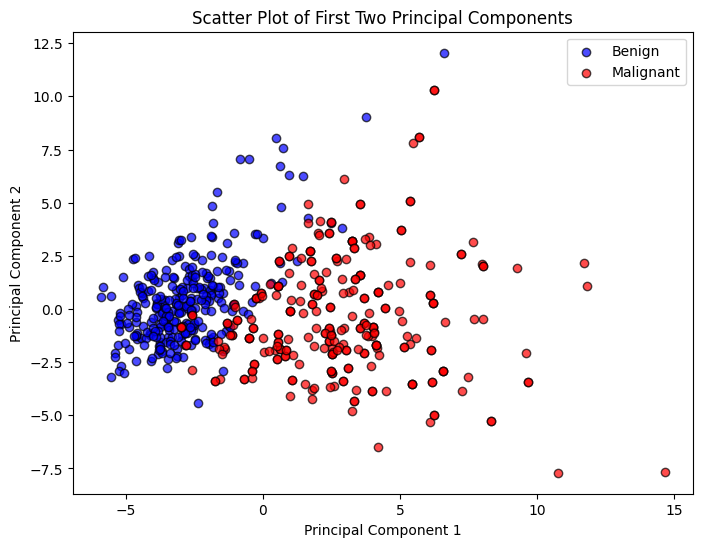
**4.3 Scaling Features**

StandardScaler was applied to standardize features, transforming them to a mean of 0 and a standard deviation of 1. Standardization prevents features with larger magnitudes from dominating the learning process, facilitating better model convergence. Standardization is particularly important for distance-based models (e.g., SVM, k-NN)

**5. Feature Selection**

Principal Component Analysis (PCA) was utilized to reduce the dimensionality of the dataset while retaining significant information. PCA transforms the 30 original features into a smaller set of uncorrelated components ranked by the amount of variance they explain. Analysis revealed that the first 11 principal components accounted for 96% of the variance, achieving a balance between reducing complexity and preserving essential information.





By applying PCA, the model's computational overhead was reduced, and its resistance to overfitting improved. Visualizations of the first two principal components highlighted their ability to provide partial separation of benign and malignant classes, showcasing PCA’s contribution to feature engineering.

**6. Model Training and Evaluation**

**6.1 Algorithms Used**

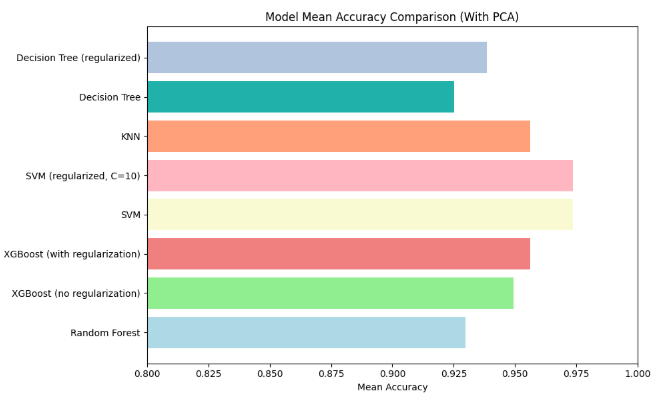
Various machine learning algorithms were employed to assess their classification effectiveness, including Random Forest, XGBoost (with and without regularization), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree models. These algorithms were chosen for their proven performance in binary classification tasks.

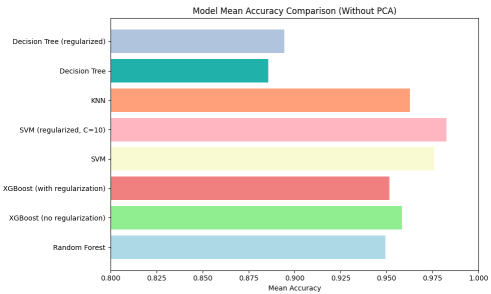
**6.2 Evaluation Metrics**

Model performance was assessed using a five-fold cross-validation strategy, ensuring that results were robust and not overly dependent on any specific data split. Key metrics included accuracy to measure overall correctness, precision to quantify the relevance of positive predictions, recall to evaluate sensitivity, and F1 score as a harmonic mean of precision and recall. Among the models tested, XGBoost and SVM exhibited the highest accuracy, emerging as top candidates for further optimization, which were used, without PCA.

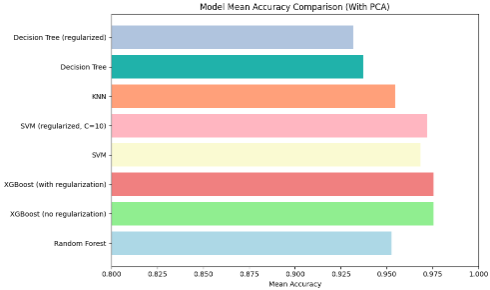
Before balancing the dataset, SVM regularized had the highest accuracy. And after balancing the dataset, we notice that XGBoost regularized has the highest accuracy.

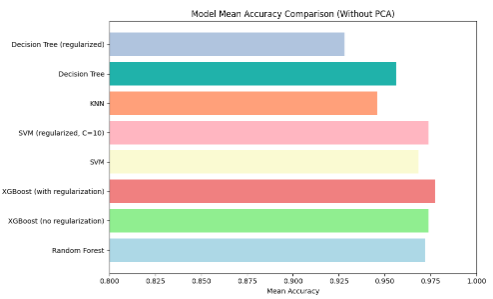
This information will be followed by the bar plots:

Without balancing:



With balancing:





**7. Hyperparameter Tuning**

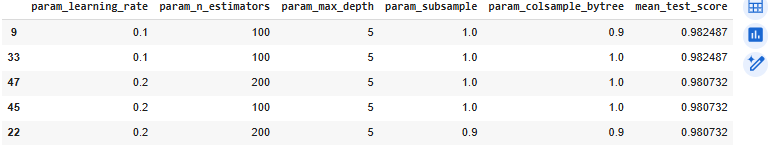
**7.1 Optimization of XGBoost**

XGBoost was fine-tuned using GridSearchCV to optimize hyperparameters, such as learning rate, number of estimators, maximum depth, and subsampling ratios. This systematic search resulted in an optimal configuration that achieved 98.2487% accuracy. The tuning process emphasized finding a balance between model complexity and generalization to prevent overfitting while maintaining high predictive accuracy. Best hyperparameters for XGBoost without PCA:

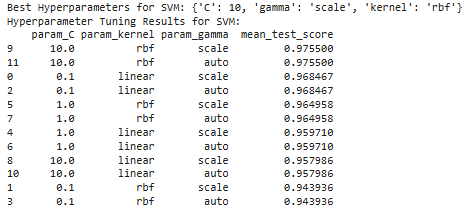
{colsample\_bytree=0.9, learning\_rate= 0.1, max\_depth= 5 , n\_estimators: 100, subsample= 1}

Key steps:

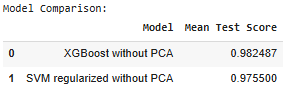
1. Define a grid of hyperparameters (param\_grid\_xgb)
2. Use GridSearchCV with 5-fold cross-validation to evaluate all combinations of hyperparameters.
3. Fit the model to the training data and identify the best hyperparameters (best\_params\_xgb).
4. Extract and display all tested hyperparameter combinations and their corresponding mean test scores.
5. Sort the results by mean test scores in descending order for better readability. This process helps ensure the XGBoost classifier is optimized for performance on the given dataset.



**7.2 Optimization of SVM**

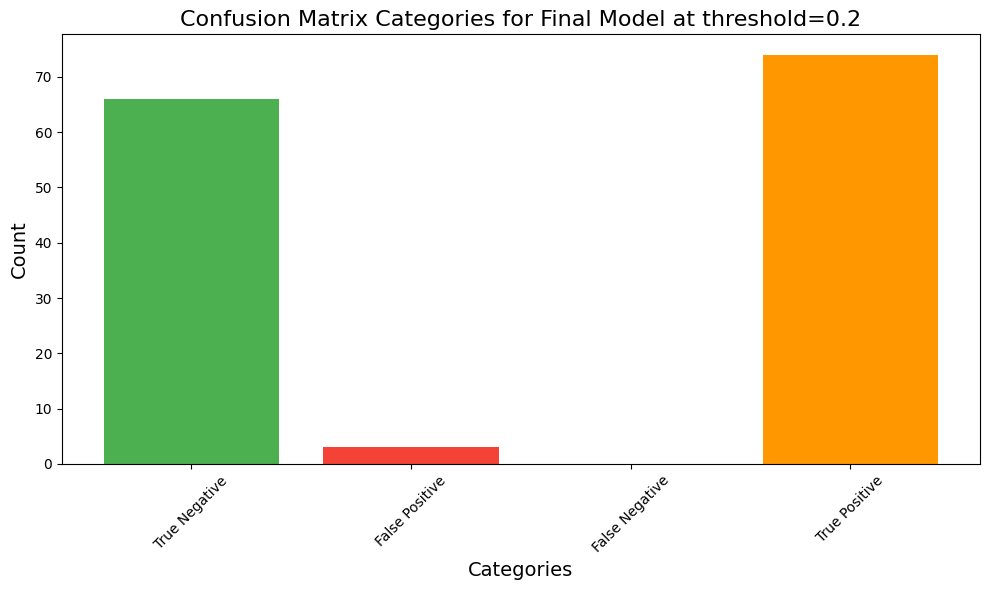
Similarly, SVM hyperparameters were optimized using GridSearchCV. The search space included regularization parameters (C), kernel types (linear and RBF), and gamma values for the RBF kernel. The best-performing configuration achieved an accuracy of 94.8%, demonstrating the effectiveness of regularization in enhancing model performance and generalization.

**7.3 Importance of Hyperparameter Tuning**

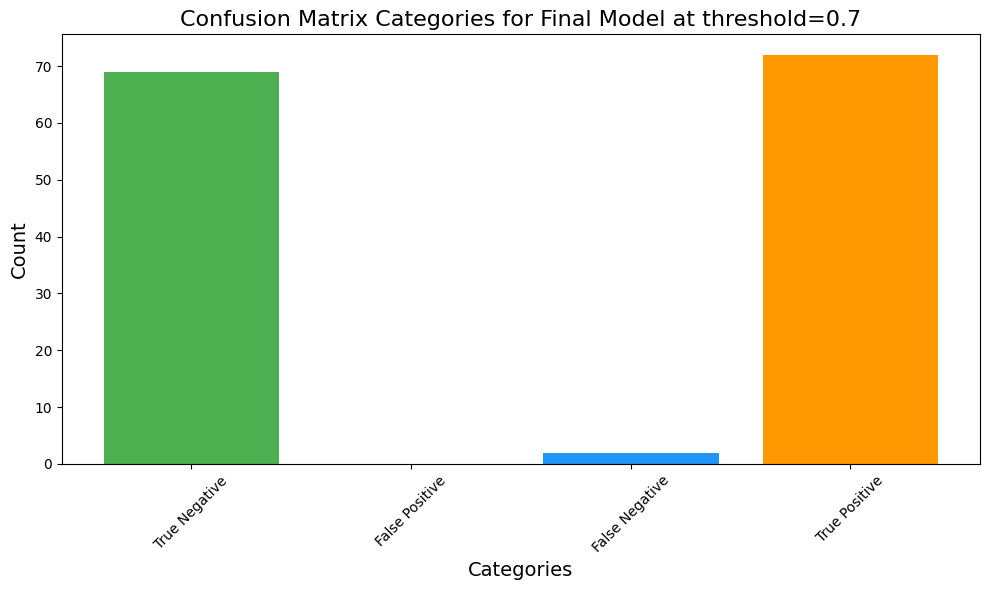
Hyperparameter tuning proved essential in maximizing the potential of both XGBoost and SVM. By systematically exploring different parameter combinations, the models were optimized to deliver not only high accuracy but also stability and efficiency. This process underscored the importance of fine-tuning in deploying machine learning models for critical applications like cancer diagnosis.

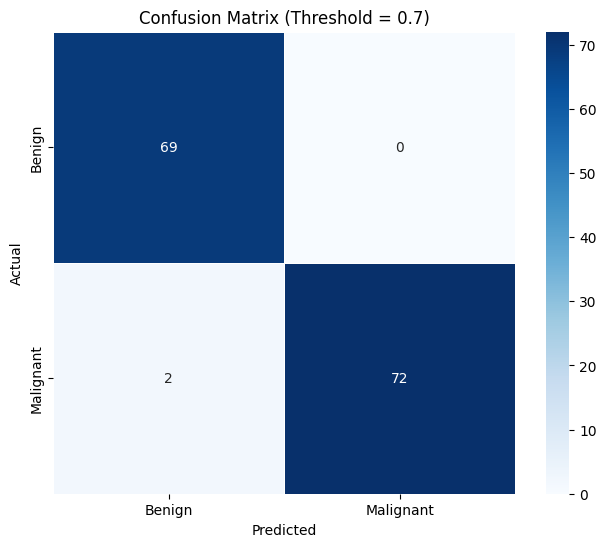
**8. Results and Comparison**

**8.1 Threshold 0.2 Analysis**

At a threshold of 0.2, the XGBoost model showed a lower bar for classifying a case as malignant. This setting resulted in higher sensitivity, which is particularly important in medical diagnostics where missing malignant cases can have severe consequences. However, this threshold also increased the number of false positives, as benign cases were more likely to be misclassified as malignant.

This threshold maximized recall, making it suitable for applications where identifying all malignant cases takes priority.

**8.2 Threshold 0.7 Analysis**

Using a stricter threshold of 0.7 reduced the likelihood of classifying a benign case as malignant. This threshold improved precision by reducing false positives, but it came at the cost of a slightly lower recall. The confusion matrix at this threshold:

This threshold provided a balanced trade-off between precision and recall, making it ideal for scenarios where minimizing false positives is critical.

**8.3 Comparison and Insights**

* **Threshold 0.2:** Prioritized sensitivity, capturing more malignant cases but with an increased rate of false positives.
* **Threshold 0.7:** Balanced sensitivity and specificity, achieving higher precision with fewer false positives.

The choice of threshold should depend on the specific use case and the relative importance of minimizing false negatives versus false positives. In life-critical applications like cancer diagnosis, thresholds can be dynamically adjusted based on patient-specific or institutional priorities.

**10. Conclusion and Future Work**

The XGBoost model proved to be the most effective, achieving high accuracy and robust generalization. Key takeaways include:

- Summary of Results: The model achieved 98.248% accuracy without PCA and demonstrated balanced precision and recall.

- Future Directions: Future work could focus on expanding the dataset, incorporating advanced techniques like deep learning, and employing interpretability tools to enhance model transparency and clinical applicability.

**Thank you Dr. for this great semester**