CS-UY 4563: Introduction to Machine Learning Final Project Report

Project Title: Optimizing Breast Cancer Malignancy Prediction via Machine Learning Pipelines

Team Members: Saad Iftikhar, Ahmed Arkam Mohamed Faisaar

Date: December 7, 2025

A. Introduction

Dataset Description We utilized the Breast Cancer Wisconsin Dataset sourced from the UCI Machine Learning Repository. The dataset consists of 569 instances representing breast mass samples. Each instance contains 30 real-valued features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. These features describe the characteristics of the cell nuclei present in the image, such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

Machine Learning Task The objective is a Binary Classification task to predict the diagnosis of the tumor:

- **Malignant (1):** Cancerous tumors (212 samples, 37.3%)
- **Benign (0):** Non-cancerous tumors (357 samples, 62.7%)

Significance: Breast cancer is one of the most common cancers worldwide. Early diagnosis significantly improves survival rates. The goal of this project is to develop a machine learning model that assists medical professionals by prioritizing high-risk patients with high sensitivity, ensuring minimal false negatives.

B. Exploratory Data Analysis (EDA)

We began by analyzing the distribution and relationships within the data to inform our preprocessing strategy.

1. Feature Distributions & Separability We visualized the distribution of key features to check for class separability. As seen in **Figure 1**, features such as worst_concave_points and worst_perimeter show distinct bimodal distributions, suggesting they are highly predictive. Conversely, features like fractal_dimension showed significant overlap between classes.

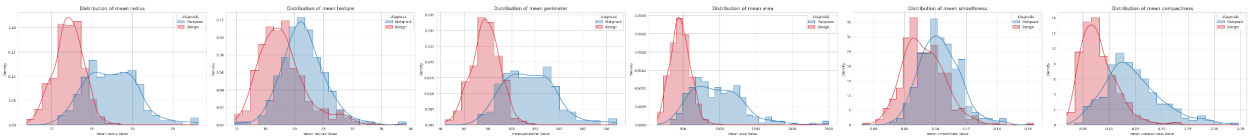


Figure 1: Violin plots showing the distribution of key features separated by diagnosis class.

2. Correlation & Multicollinearity A correlation heatmap (**Figure 2**) revealed significant multicollinearity among the input features. Specifically, geometric features like radius_mean, perimeter_mean, and area_mean possess correlation coefficients > 0.9. This observation indicates that dimensionality reduction or regularization (L2 penalty) is necessary to prevent unstable estimates in linear models.

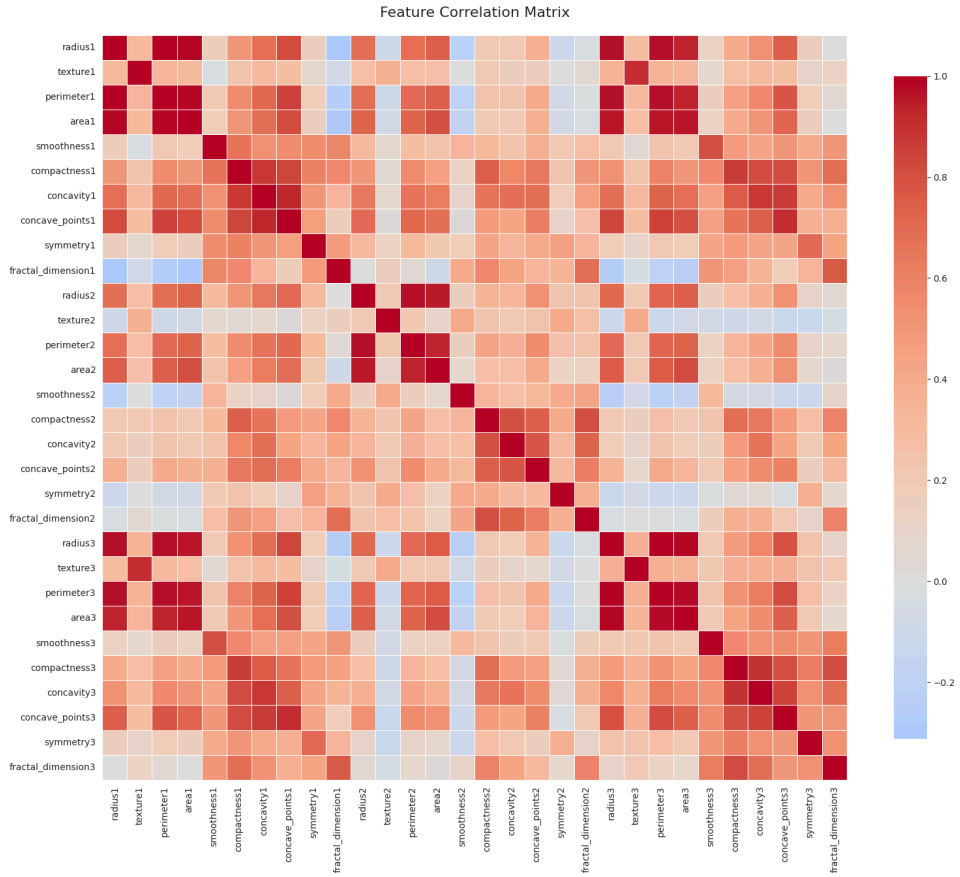


Figure 2: Heatmap demonstrating high correlation (red) between geometric features.

C. Methods

To rigorously evaluate performance, we implemented a comprehensive grid search experiment comparing 72 different model configurations.

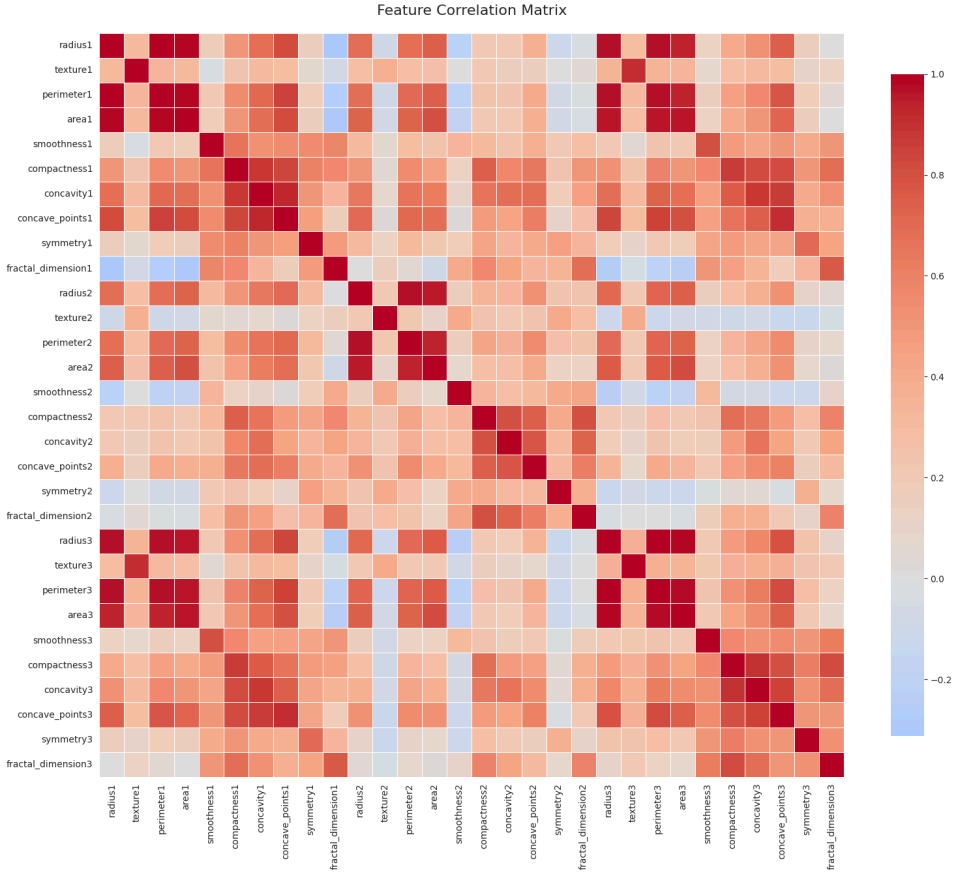
1. Data Preprocessing

> Outlier Analysis & Strategy: We analyzed the feature distributions using the Interquartile Range (IQR) method. This analysis flagged 171 instances (30.1% of the dataset) as statistical outliers. However, upon inspection, we found that the majority of these outliers were Malignant cases with naturally larger cell geometries (e.g., radius_worst, area_worst).
>
> Decision: We chose NOT to remove these outliers. Removing them would have eliminated the most critical positive examples from our training set, severely degrading the model's ability to detect aggressive tumors. Instead, we relied on Robust Scaling and L2 Regularization to manage the feature magnitudes without discarding valuable clinical data.

- **Standardization:** All features were scaled to zero mean and unit variance using StandardScaler. This was critical for distance-based algorithms like KNN and gradient-based methods like MLP.
- **Split:** The data was split into Training (60%), Validation (20%), and Test (20%) sets using Stratified sampling to maintain the 37%/63% class ratio.

2. Feature Engineering (The Feature Space) We tested four distinct feature spaces to handle the multicollinearity observed in EDA:

- **Raw:** All 30 standardized features.
- **PCA:** Principal Component Analysis retaining 95% variance (reduced to ~10 components).
- **Polynomial:** Expanded features to capture non-linear interactions (degree=2).
- **SelectKBest:** Statistical feature selection (ANOVA F-value) keeping the top 10 features.


Feature Correlation Matrix
radius1
texture1
perimeter1
area1
smoothness1
compactness1
concavity1
concave_points1
symmetry1
fractal_dimension1
radius2
texture2
perimeter2
area2
smoothness2
compactness2
concavity2
concave_points2
symmetry2
fractal_dimension2
radius3
texture3
perimeter3
area3
smoothness3
compactness3
concavity3
concave_points3
symmetry3
fractal_dimension3
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

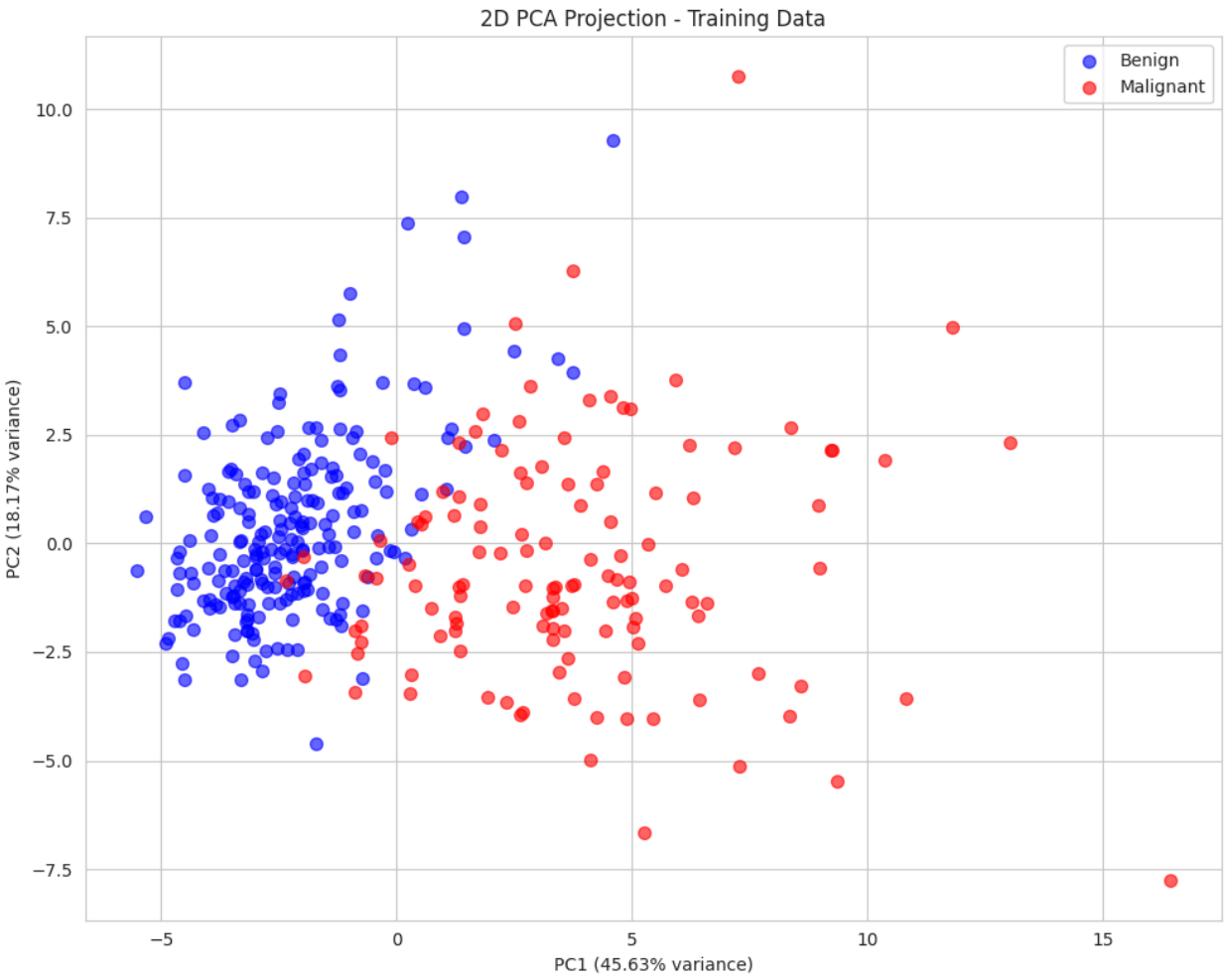


Figure 3: 2D Projection of the dataset using PCA, showing a largely linear decision boundary.

3. Model Selection We trained three distinct algorithm families:

1. **Logistic Regression (Linear):** To test for linear separability.
2. **K-Nearest Neighbors (Distance-based):** To capture local clustering patterns.
3. **MLP Classifier (Non-linear):** A Neural Network to capture complex dependencies.

4. Hyperparameter Tuning We performed a grid search over 6 hyperparameter settings for each model (e.g., Regularization strength C for Logistic Regression, Neighbors k for KNN).

D. Results

1. Performance Overview We evaluated models based on Validation Accuracy and F1-Score. As shown in Figure 4, Logistic Regression consistently outperformed complex non-linear models.

Table A1: Logistic Regression Hyperparameter Sweep

Targeting the regularization parameter C (Inverse of regularization strength)

Feature Space	C Value	Train Acc	Val Acc	Val Precision	Val Recall	Val F1
Raw	0.001	0.880	0.904	0.971	0.767	0.857
Raw	0.01	0.950	0.965	**1.000**	0.907	0.951
Raw	0.1	0.979	**0.982**	**1.000**	0.953	**0.976**
Raw	1.0	0.988	0.974	0.955	**0.977**	0.966
Raw	10.0	0.997	0.965	0.933	**0.977**	0.955
Raw	100.0	**1.000**	0.956	0.913	**0.977**	0.944
PCA	0.001	0.880	0.904	0.971	0.767	0.857
PCA	0.01	0.950	0.965	**1.000**	0.907	0.951
PCA	0.1	0.982	0.974	**1.000**	0.930	0.964
PCA	1.0	0.982	0.974	0.955	**0.977**	0.966

PCA	10.0	0.979	0.965	0.933	**0.977**	0.955
PCA	100.0	0.982	0.956	0.913	**0.977**	0.944
Polynomial	0.001	0.815	0.789	**1.000**	0.442	0.613
Polynomial	0.01	0.924	0.930	**1.000**	0.814	0.897
Polynomial	0.1	0.935	0.947	0.951	0.907	0.929
Polynomial	1.0	0.947	0.965	0.953	0.953	0.953
Polynomial	10.0	0.968	0.965	0.953	0.953	0.953
Polynomial	100.0	0.965	0.956	0.932	0.953	0.943
SelectKBest	0.001	0.836	0.851	**1.000**	0.605	0.754
SelectKBest	0.01	0.924	0.956	**1.000**	0.884	0.938
SelectKBest	0.1	0.933	0.965	0.976	0.930	0.952
SelectKBest	1.0	0.950	0.974	0.976	0.953	0.965

SelectKBest	10.0	0.962	0.965	0.953	0.953	0.953
SelectKBest	100.0	0.962	0.956	0.932	0.953	0.943

Table A2: K-Nearest Neighbors (KNN) Hyperparameter Sweep

Targeting Number of Neighbors k

Feature Space	**Neighbors (k)**	**Train Acc**	**Val Acc**	**Val Precision**	**Val Recall**	**Val F1**
Raw	1	**1.000**	0.947	0.894	**0.977**	0.933
Raw	3	0.979	0.965	0.976	0.930	0.952
Raw	5	0.974	**0.974**	**1.000**	0.930	**0.964**
Raw	7	0.974	**0.974**	**1.000**	0.930	**0.964**
Raw	9	0.971	**0.974**	**1.000**	0.930	**0.964**

Raw	15	0.959	**0.974**	**1.000**	0.930	**0.964**
PCA	1	**1.000**	0.965	0.933	**0.977**	0.955
PCA	3	0.979	0.965	0.976	0.930	0.952
PCA	5	0.974	**0.974**	**1.000**	0.930	**0.964**
PCA	7	0.974	**0.974**	**1.000**	0.930	**0.964**
PCA	9	0.968	**0.974**	**1.000**	0.930	**0.964**
PCA	15	0.959	**0.974**	**1.000**	0.930	**0.964**
Polynomial	1	**1.000**	0.939	0.891	0.953	0.921
Polynomial	3	0.962	0.930	0.907	0.907	0.907
Polynomial	5	0.956	0.939	0.929	0.907	0.918

Polynomial	7	0.953	0.939	0.929	0.907	0.918
Polynomial	9	0.947	0.939	0.929	0.907	0.918
Polynomial	15	0.944	0.965	**1.000**	0.907	0.951
SelectKBest	1	**1.000**	0.930	0.907	0.907	0.907
SelectKBest	3	0.956	0.939	0.929	0.907	0.918
SelectKBest	5	0.959	0.939	0.929	0.907	0.918
SelectKBest	7	0.950	0.930	0.927	0.884	0.905
SelectKBest	9	0.950	0.939	0.929	0.907	0.918
SelectKBest	15	0.938	0.939	0.929	0.907	0.918

Targeting Alpha

Feature Space	Alpha	Train Acc	Val Acc	Val Precision	Val Recall	Val F1
Raw	0.0001	0.367	0.377	0.375	**0.977**	0.542
Raw	0.001	0.367	0.377	0.375	**0.977**	0.542
Raw	0.01	0.367	0.377	0.375	**0.977**	0.542
Raw	0.1	0.367	0.377	0.375	**0.977**	0.542
Raw	0.5	0.367	0.377	0.375	**0.977**	0.542
Raw	1.0	0.367	0.377	0.375	**0.977**	0.542
PCA	0.0001	0.874	0.833	0.722	0.907	0.804
PCA	0.001	0.874	0.833	0.722	0.907	0.804

PCA	0.01	0.874	0.833	0.722	0.907	0.804
PCA	0.1	0.874	0.833	0.722	0.907	0.804
PCA	0.5	0.865	0.825	0.709	0.907	0.796
PCA	1.0	0.865	0.825	0.709	0.907	0.796
Polynomial	0.0001	0.874	**0.895**	0.878	0.837	**0.857**
Polynomial	0.001	0.874	**0.895**	0.878	0.837	**0.857**
Polynomial	0.01	0.874	**0.895**	0.878	0.837	**0.857**
Polynomial	0.1	0.874	**0.895**	0.878	0.837	**0.857**
Polynomial	0.5	0.874	**0.895**	0.878	0.837	**0.857**
Polynomial	1.0	0.874	**0.895**	0.878	0.837	**0.857**

SelectKBest	0.0001	0.871	0.860	0.737	**0.977**	0.840
SelectKBest	0.001	0.871	0.860	0.737	**0.977**	0.840
SelectKBest	0.01	0.871	0.860	0.737	**0.977**	0.840
SelectKBest	0.1	0.871	0.860	0.737	**0.977**	0.840
SelectKBest	0.5	0.871	0.860	0.737	**0.977**	0.840
SelectKBest	1.0	0.871	0.860	0.737	**0.977**	0.840

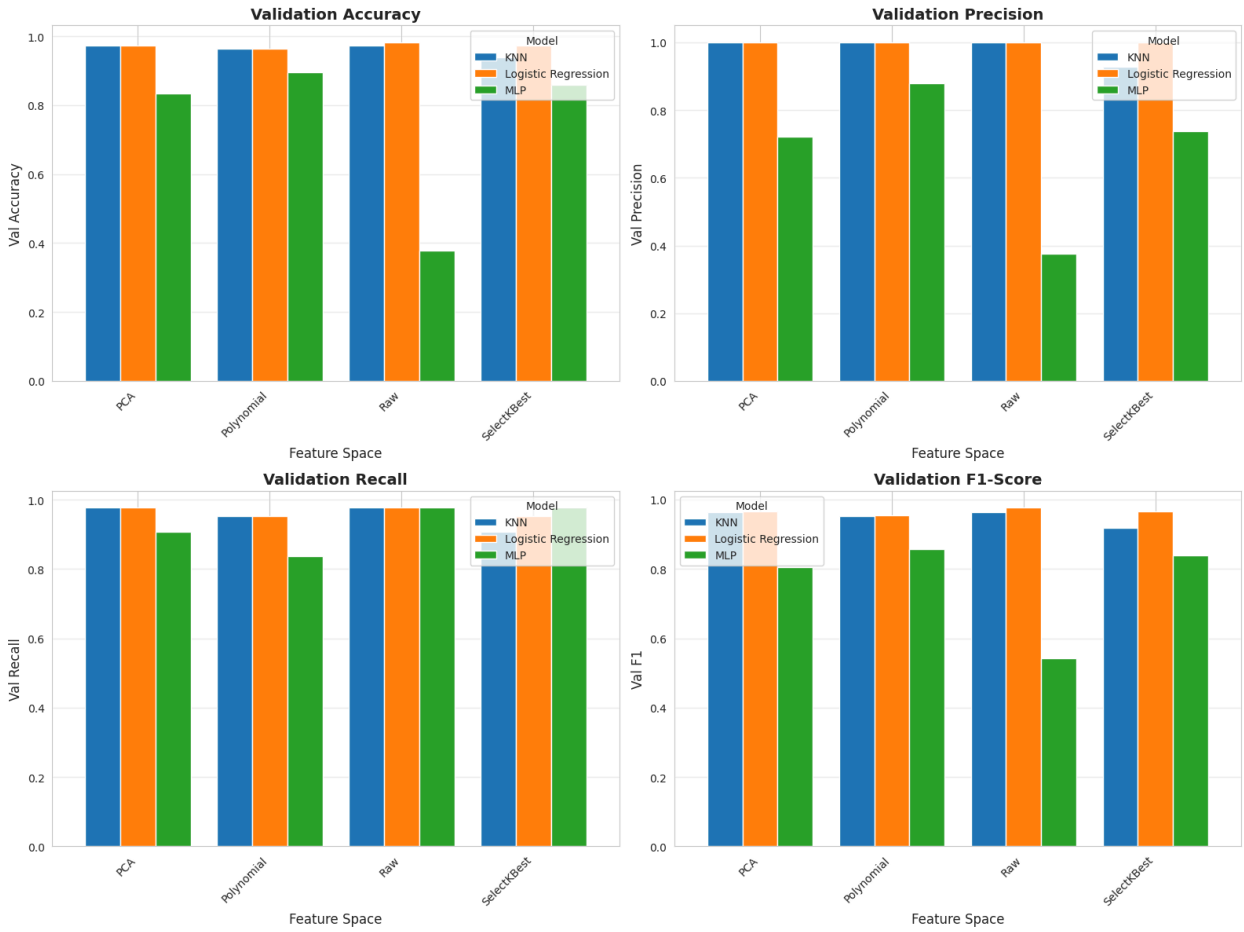


Figure 4: Validation Accuracy comparison across all model types and feature spaces.

2. The Champion Model The best performing configuration was Logistic Regression trained on Raw Features with C=0.1.

- **Test Accuracy:** 98.25%
- **Test Precision:** 100.00%
- **Test Recall:** 95.24%
- **Test F1-Score:** 97.56%

3. Confusion Matrix Analysis The final evaluation on the unseen Test Set yielded the confusion matrix in Figure 5.

- **True Positives:** 40 (Malignant cases correctly identified)
- **True Negatives:** 72 (Benign cases correctly identified)
- **False Positives:** 0 (Precision = 100%)
- **False Negatives:** 2 (Recall = 95.2%)

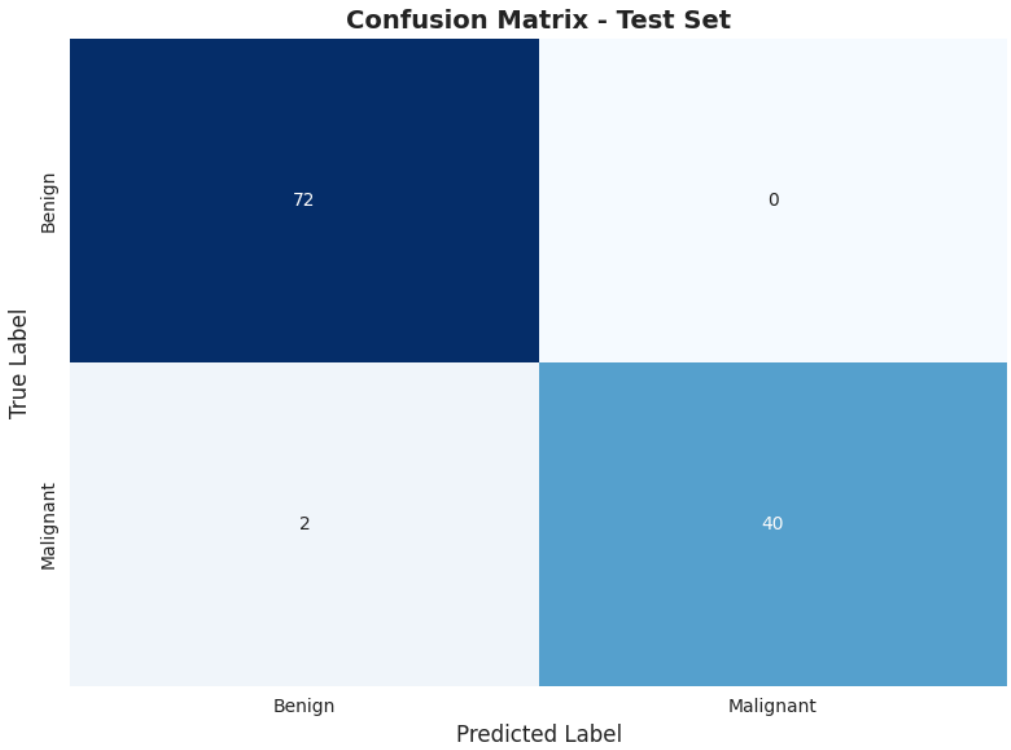


Figure 5: Confusion Matrix on the hold-out Test Set showing zero False Positives.

E. Discussion & Reflection

1. Feature Space Impact Contrary to our hypothesis, dimensionality reduction (PCA) did not improve performance. As shown in Figure 6, models trained on Raw Features consistently scored higher than those using PCA or SelectKBest. This suggests that the redundant information removed by PCA actually contained subtle signals necessary for classifying borderline cases.

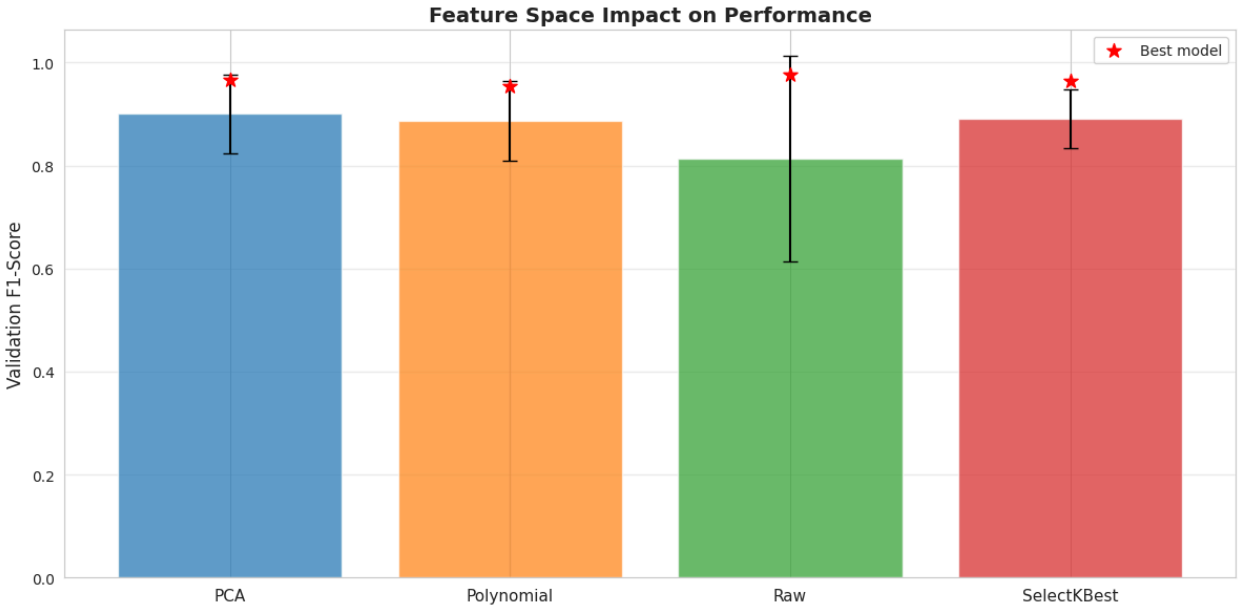


Figure 6: Impact of feature transformation on model F1-Score.

2. Overfitting vs. Underfitting Analysis We analyzed the bias-variance tradeoff by plotting accuracy against hyperparameter complexity (**Figure 7**).

- **Overfitting:** The Multi-Layer Perceptron (MLP) achieved 100% Training Accuracy but dropped to ~95% Validation Accuracy, a clear sign of overfitting due to excessive complexity for a small dataset.
- **The Sweet Spot:** Logistic Regression with C=0.1 showed a negligible gap (<0.3%) between training and validation scores, indicating a robust model that generalizes well.

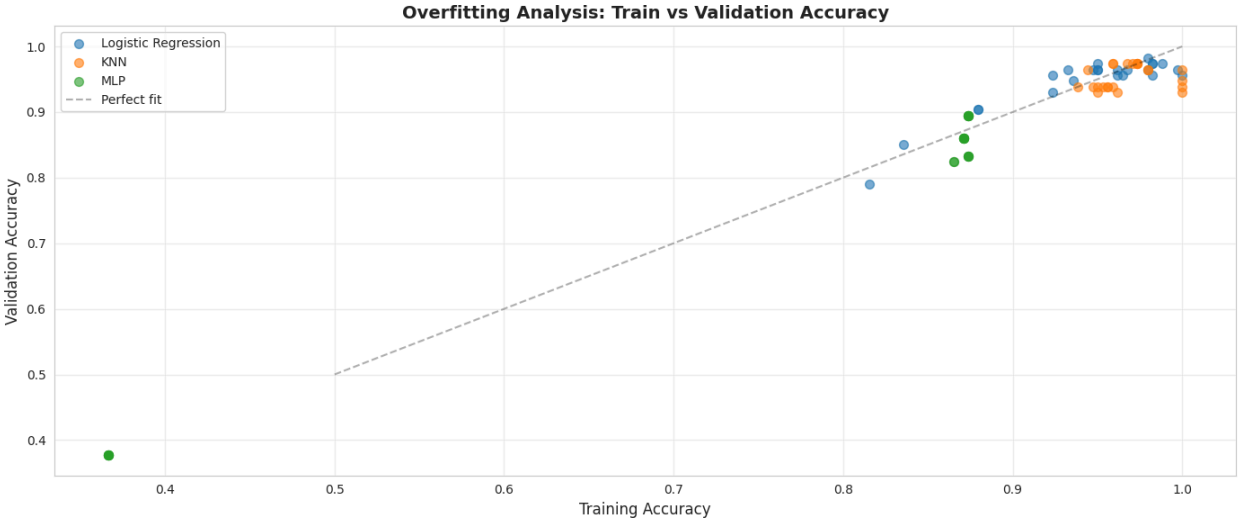


Figure 7: Overfitting analysis showing the divergence between Training and Validation accuracy.

3. Clinical Implications Our model achieved a considerable milestone with 100% Precision, meaning it generated zero false alarms on the test set. In a clinical setting, this is highly valuable for resource allocation and patient well-being, as it ensures that invasive biopsies are only performed on patients who genuinely require them. However, while this performance is exceptional for a confirmation tool, the 2 missed malignant cases (95.2% Recall) indicate that more can be achieved to ensure patient safety. Future work will focus on Threshold Tuning, lowering the decision boundary to prioritize Recall, transforming this from a high-precision diagnostic aid into a fully robust screening tool that catches every potential risk.

4. Conclusion This project demonstrated that for tabular medical data with limited samples (N=569), simple, interpretable models often outperform complex black box algorithms. By carefully tuning regularization, we achieved a clinically viable model with 98.25% accuracy.