Histopathologic Cancer Detection

Group C : Jing Wang,Vikash Bajiya, Deviprasad Saka

Date: December 12, 2024

1. Introduction

- Problem Statement:
  Histopathologic cancer detection involves identifying metastatic cancer in small image patches extracted from larger pathology scans. Accurate detection aids in timely diagnosis and treatment planning.
- Objective:
  To design and evaluate multiple machine learning models for binary image classification, leveraging advanced preprocessing, augmentation, and optimization techniques.
- Dataset:
  The dataset consists of 32x32-pixel pathology images labeled as positive (cancerous) or negative (non-cancerous). Class imbalance is inherent, with more negative samples than positive.

2. Data Preprocessing

2.1 Data Introduction

The dataset used for our project is the Histopathologic Cancer Detection Dataset, which consists of labeled image patches extracted from pathology slides. Each image is a 32x32 pixel RGB patch categorized into two classes:

- Label 0: Represents a negative sample, where the patch does not contain metastatic cancer.
- Label 1: Represents a positive sample, indicating the presence of metastatic cancer.

The dataset is characterized by a class imbalance, with significantly more negative samples than positive samples. To mitigate this, techniques like SMOTE (Synthetic Minority Oversampling Technique) and data augmentation were applied during analytics.

Below is a sample visualization of the dataset, showcasing a mix of positive (Label: 1) and negative (Label: 0) image patches:

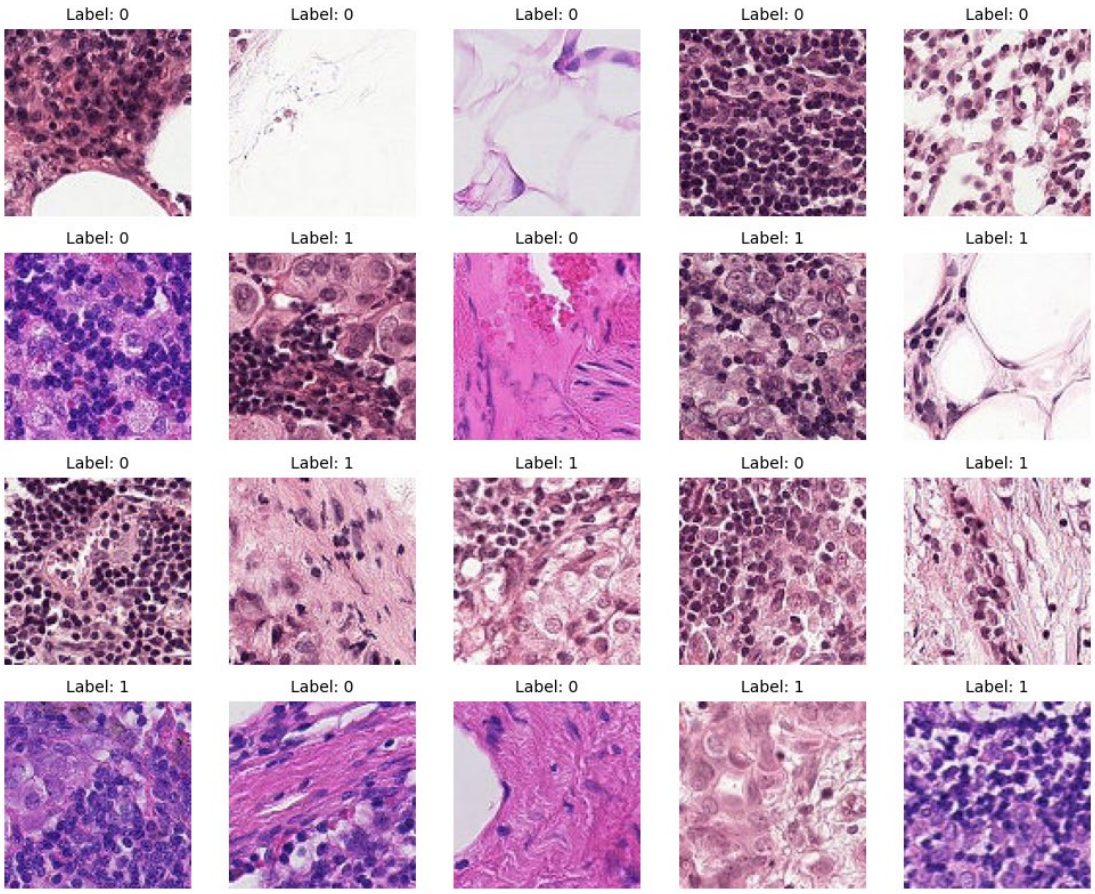

Fig 2.1 Sample Visualization of Dataset

Observations:

- Positive samples often exhibit denser cell clusters and darker regions, suggesting the presence of cancer.
- Negative samples tend to have more uniform structures or appear lighter in texture.

This dataset forms the foundation for training and evaluating our machine learning models, with a focus on binary classification for detecting metastatic cancer.

2.2 Data Cleaning

- Verified consistency between the dataset and the label file.
- Identified and removed missing, extra, or corrupted images.
- Ensured there were no duplicate images in the dataset.

2.3 Exploratory Data Analysis

- Analyzed the class distribution:
    - Positive samples: 89,117

- Negative samples: 130,908

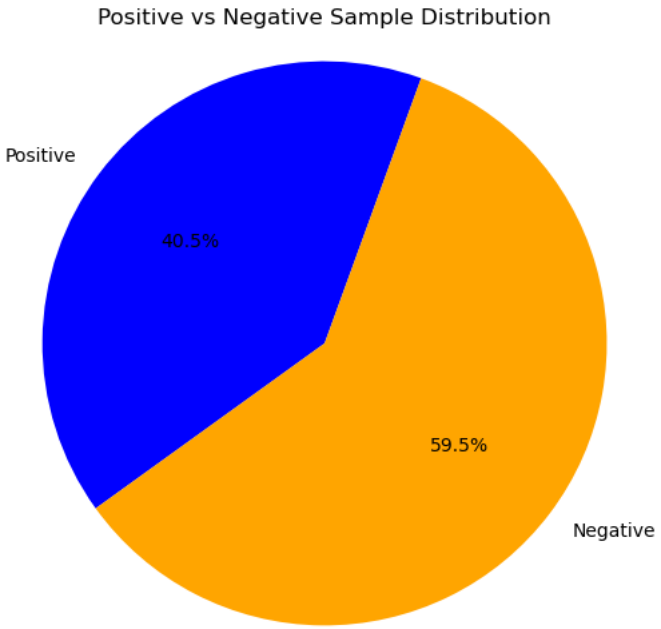

Fig 2.2 Sample Distribution

- Visualized image features (e.g., brightness, color channel distributions) to check for distinct patterns.

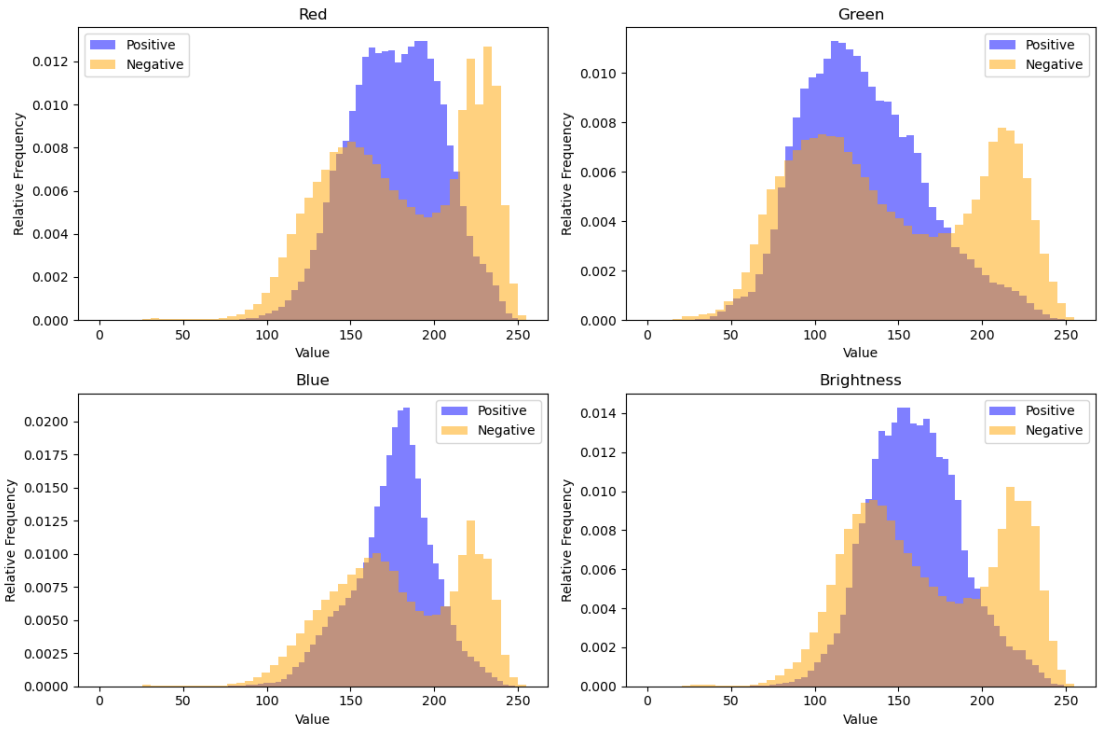

Fig 2.3 Sample Features Comparison

2.4 Preprocessing Steps

- Normalization: Scaled pixel values to the range [0, 1] for neural network models.
- Standardization: Standardized data using z-scores for models like Logistic Regression and SVM.
- Data Augmentation: Applied random flips, rotations, and resizing to enhance diversity and generalization.
- Class Imbalance Handling: Used SMOTE to oversample minority classes, ensuring balanced training.

3. Model Design and Implementation

Six models were designed, trained, and evaluated to identify the most effective classifier:

1. Naive Bayes
2. Random Forest
3. Logistic Regression
4. Support Vector Machine (SVM)
5. Neural Network
6. XGBoost

3.1 Model 1: Naive Bayes

- Key Features:
    - Assumes independence between features.
    - Efficient for large datasets.
- Preprocessing: Used raw pixel values without scaling or normalization.
- Performance:

  Accuracy: 73%; Precision: 75%; Recall: 73%; F1-score: 73%

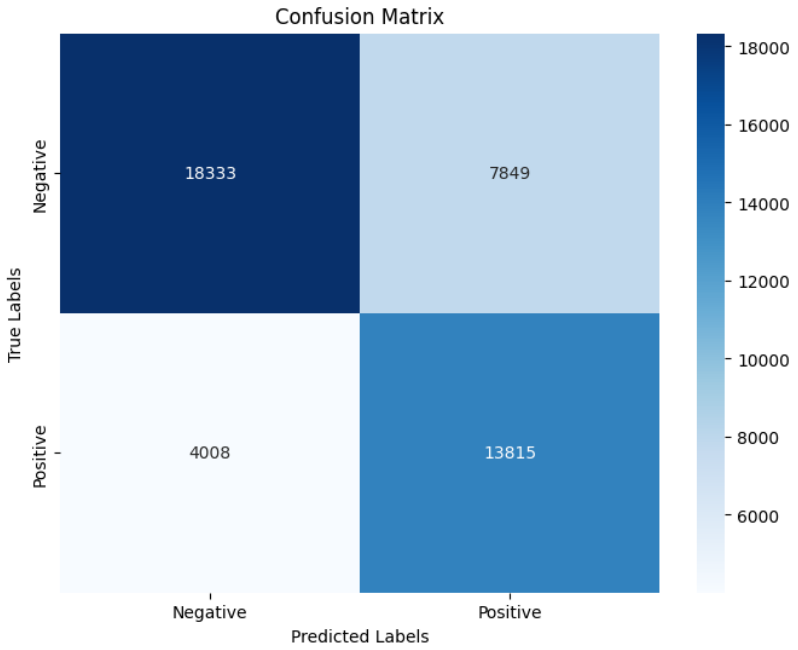

Fig 3.1 Confusion Matric

3.2 Model 2: Random Forest

- Key Features:
    - Ensemble method using decision trees.
    - Captures feature importance and non-linear relationships.
- Performance:

  Accuracy: 75%; Precision: 76%; Recall: 75%; F1-score: 74%

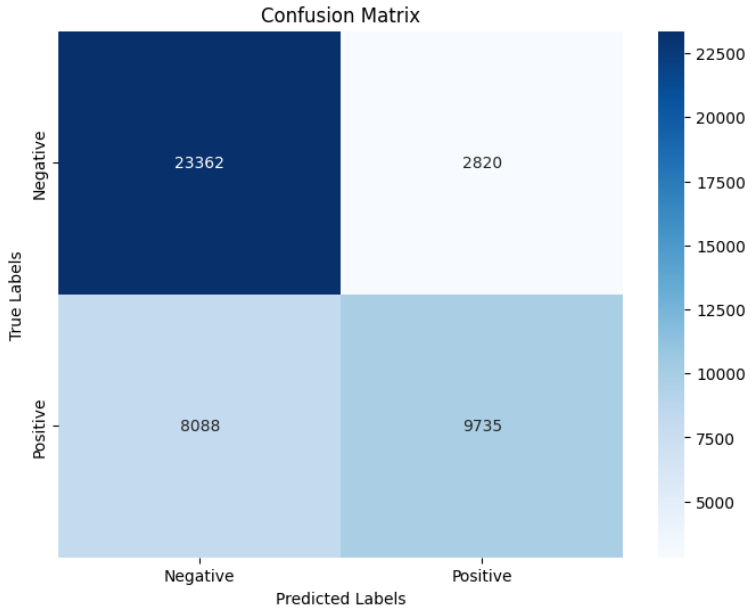

Fig 3.2 Confusion Matric

3.3 Model 3: Logistic Regression

- Key Features:
    - Linear classification model.
    - Uses sigmoid activation to predict probabilities.
- Enhancements:
    - Added L2 regularization for improved generalization.
    - Applied probability calibration (Isotonic Regression).
- Performance:

  Accuracy: 63%; Precision: 62%; Recall: 63%; F1-score: 62%

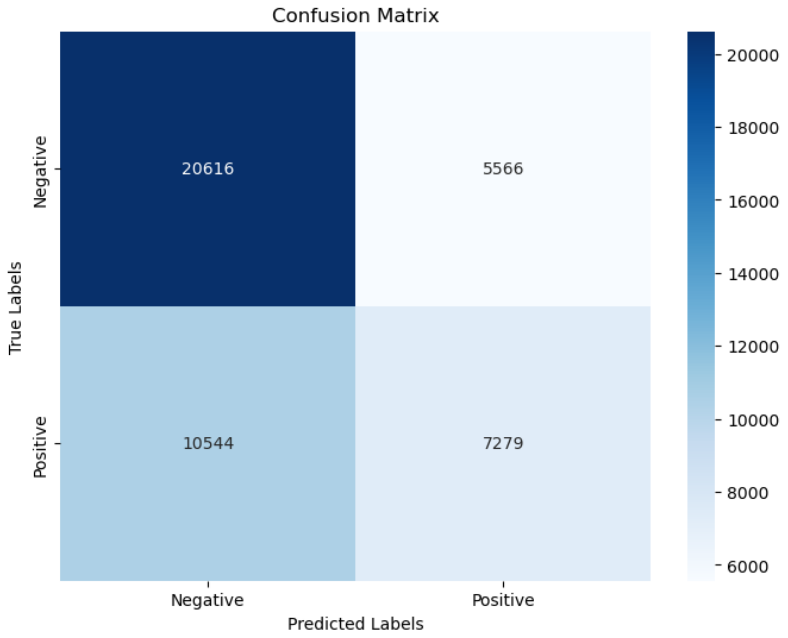

Fig 3.3 Confusion Matric

3.4 Model 4: Support Vector Machine (SVM)

- Key Features:
    - Finds an optimal hyperplane for classification.
    - Uses kernel functions for non-linear decision boundaries.
- Performance:

  Accuracy: 76%; Precision: 77%; Recall: 76%; F1-score: 77%

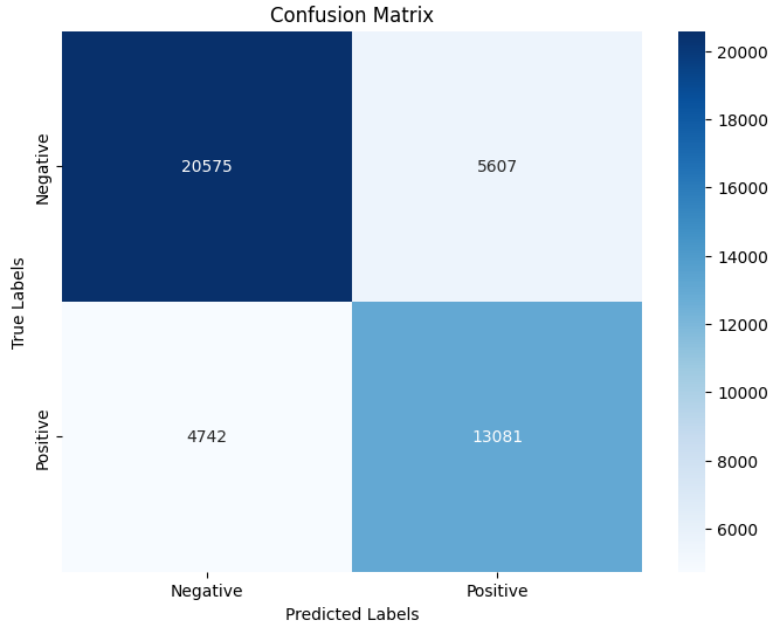

Fig 3.4 Confusion Matric

3.5 Model 5: Neural Network

Initial Implementation

- Architecture:
    - Input Layer: 1024 neurons (flattened image).
    - Hidden Layers: 2 layers with 512 and 256 neurons.
    - Activation: ReLU for hidden layers, Sigmoid for the output layer.
    - Dropout: Added 20% dropout to reduce overfitting.
- Training:
    - Batch Size: 32
    - Epochs: 5
    - Optimizer: Adam
    - Learning Rate: 0.001
- Data Handling:
    - Used normalized and augmented training data.
    - Did not explicitly address class imbalance.

Performance:

Accuracy: 70%; Precision: 73%; Recall: 70%; F1-score: 71%

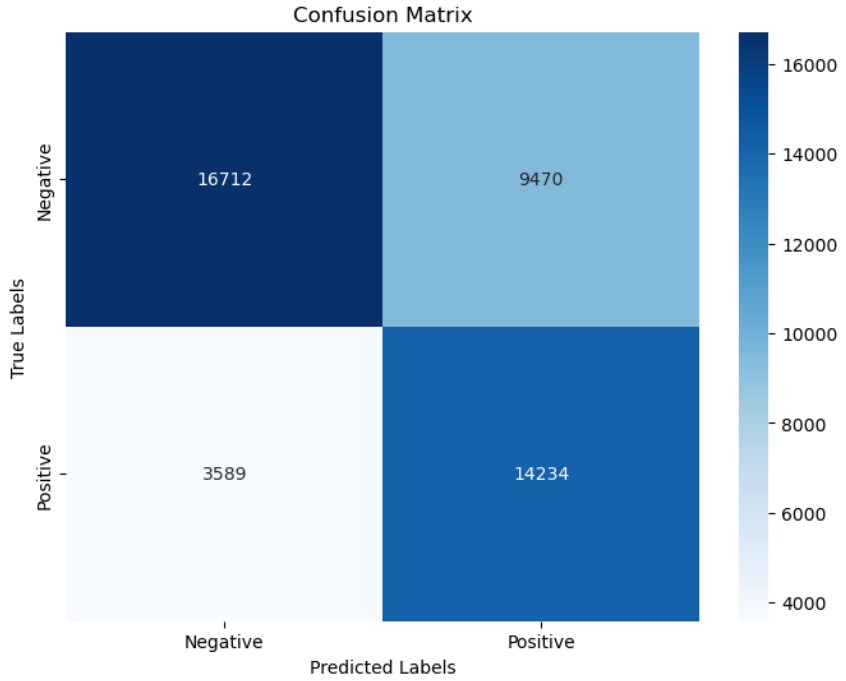

Fig 3.5 Confusion Matric

Improved Neural Network Model

In the second iteration, we made the following improvements to address key challenges and further enhance the model's performance:

1. Neural Network Architecture:
    - Increased Neurons: Added more neurons in each layer to improve the network's capacity to learn complex patterns.
    - Activation Functions: Replaced standard ReLU with LeakyReLU in the hidden layers for better gradient flow and stability, while keeping ReLU in the final hidden layer.
    - Dropout Layers: Incorporated dropout layers in each hidden layer to mitigate overfitting, especially on augmented data.
2. Hyperparameters:
    - Learning Rate: Adjusted to 0.0005 for more stable convergence.
    - Optimizer: Switched to AdamW to incorporate weight decay (L2 regularization) for better generalization.
    - Batch Size: Increased to 64 to balance computation time and gradient

updates.

- Epochs: Increased to 10 to allow the model more time to converge.

3. Class Imbalance:
   - Applied SMOTE to oversample minority classes, ensuring balanced training and improved recall for positive samples.
4. Data Augmentation:
   - Continued using normalized and augmented training data (e.g., random flips, resizing) to enhance generalization.

Performance:

Accuracy: 77%; Precision: 77%; Recall: 77%; F1-score: 77%

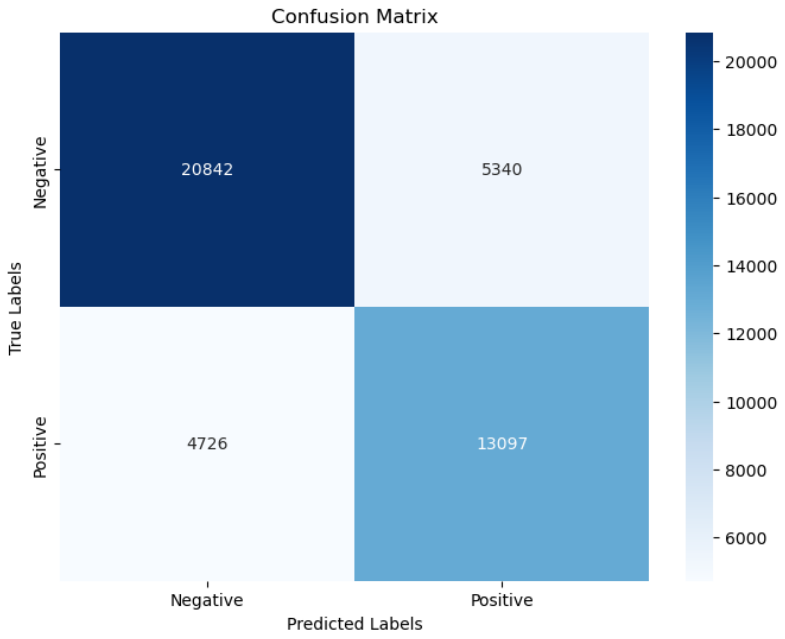

Fig 3.6 Confusion Matric

3.6 Model 6: XGBoost

- Key Features:
  - Gradient boosting framework for tree-based models.
  - Optimized for speed and accuracy.
- Hyperparameters:
  - Learning Rate: 0.1

- Max Depth: 6
- Subsample: 0.8
- Trees: 100

- Strengths: Robust to overfitting and handles class imbalance well.
- Limitations: Complexity in hyperparameter tuning.
- Performance:

  Accuracy: 81%; Precision: 81%; Recall: 81%; F1-score: 81%

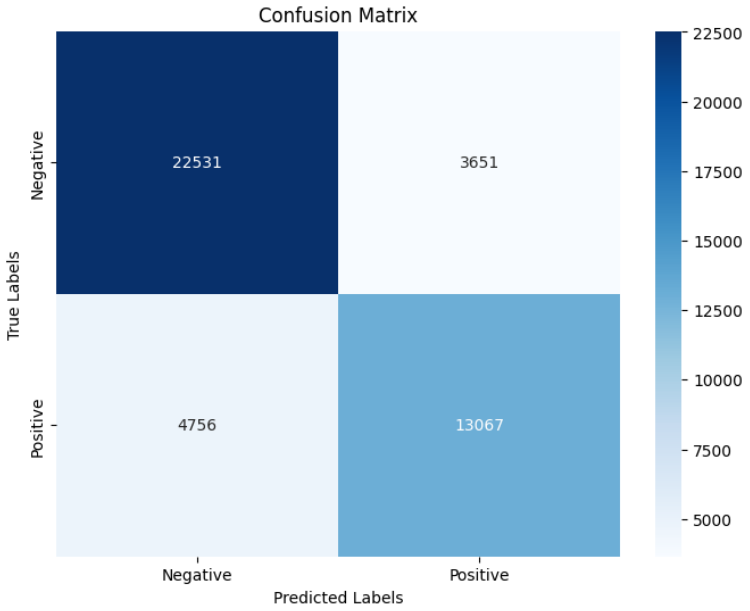

Fig 3.7 Confusion Matric

4. Models Results and Comparison

Table 4.1 Comparison of Model Performance

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Naive Bayes	0.73	0.75	0.73	0.73
Random Forest	0.75	0.76	0.75	0.74
Logistic Regression	0.63	0.62	0.63	0.62
SVM	0.76	0.77	0.76	0.77
Neural Network	0.77	0.77	0.77	0.77

Model	Accuracy	Precision	Recall	F1-score
XGBoost	0.81	0.81	0.81	0.81

- Best Model: XGBoost achieved the highest performance across all metrics.
- Neural Network and SVM showed competitive performance with slight differences.
- Logistic Regression underperformed due to its linear nature.

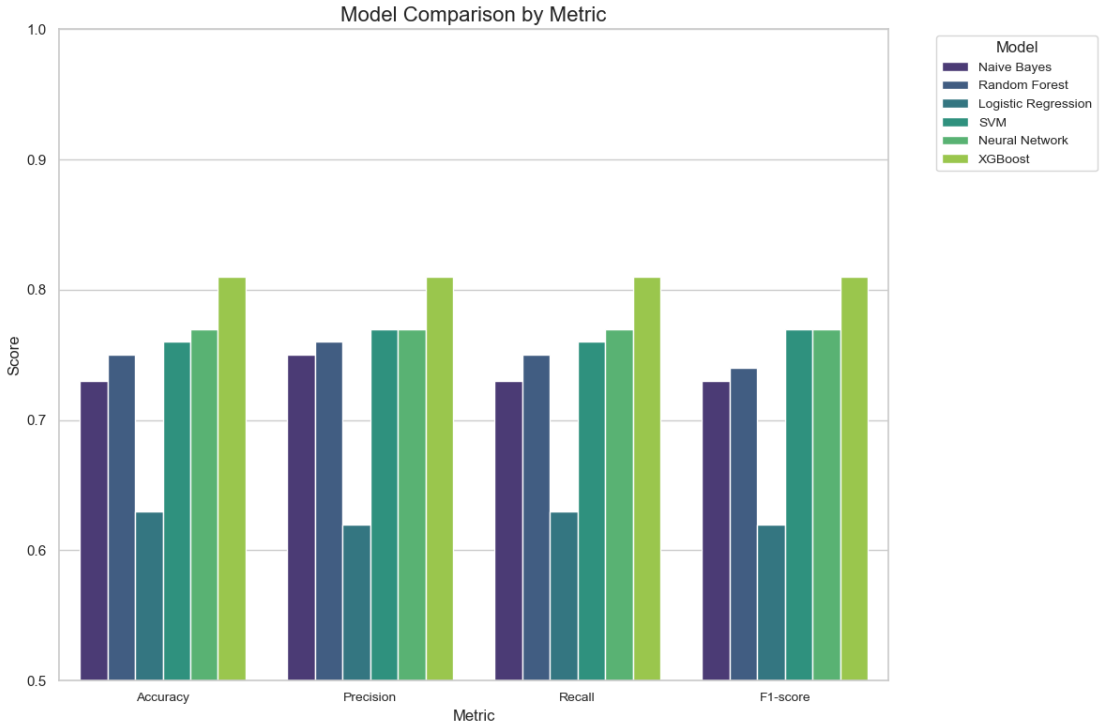

Fig 4.1 Model Comparison by metric

5. Future Improvements

Neural Network:

- Experiment with different loss functions (e.g., focal loss for imbalanced data).
- Adjust the learning rate dynamically with schedulers.
- Increase the number of epochs for longer training.

XGBoost:

- Fine-tune hyperparameters further (e.g., tree depth, subsampling rates).
- Add custom objective functions to better handle class imbalance.

6. Real-World Application

Cancer is one of the leading causes of death worldwide. Early detection of metastatic cancer through histopathologic analysis can significantly improve treatment outcomes. This project demonstrates how modern AI techniques, particularly neural networks and ensemble methods, can enhance the accuracy and speed of cancer detection. By leveraging technology, we aim to aid medical professionals in making faster and more reliable diagnoses, ultimately contributing to better patient care and outcomes.

This work highlights the potential of machine learning to revolutionize healthcare and make a meaningful impact on human lives.

7. Contribution of Team Members

Jing Wang:

Responsible for data preprocessing, all model training, combining works of other teammates, comparison, and report writing.

Vikash Bajiya:

Responsible for data preprocessing, model training (Logistic regression & SVM), report writing and presentation preparation.

Deviprasad Saka:

Responsible for data preprocessing, model training (Naive Bayes & Random Forest), report writing and presentation preparation.