CS640 Project Report

Kelvin Kuang
Boston University
Email: kkuang@bu.edu

December 12, 2024

Contents

1 Data Analysis

When examining the data, the first statistical analysis reveals significant class imbalance across the dataset. This imbalance presents potential challenges for predictive modeling and requires careful consideration in subsequent analytical approaches. The unequal distribution of class frequencies can introduce bias and potentially compromise the reliability of statistical inferences if not appropriately addressed. To quantify the extent of class imbalance, the proportion of instances within each class category was calculated. The observed disparities suggest the need for specialized techniques such as sampling, class weighting, or resampling methods to mitigate potential statistical artifacts and ensure robust analytical outcomes.

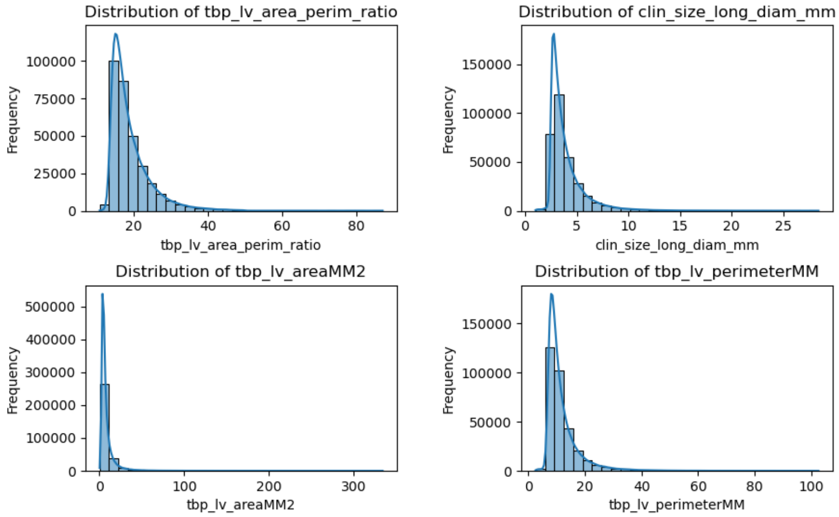

Figure 1: Distributions of 'tbp_lv_area_perim_ratio' (ratio between lesions perimeter and area), 'clin_size_long_diam_mm' (maximum diameter of the lesion in mm), 'tbp_lv_areaMM2' (area of lesion in mm^2), and 'tbp_lv_perimeterMM' (perimeter of lesion in mm).

When analyzing some of the numerical features within the dataset, we see that the features regarding the perimeter and area of the lesion are highly skewed (Figure 1). The distributions of the 'tbp_lv_areaMM2' and 'tbp_lv_perimeterMM' features display pronounced right-skewness, indicating that the majority of the data points have relatively low perimeter and area values, with a long tail towards higher values. This skewness suggests that the dataset may contain a large number of smaller lesions, with a smaller subset of larger lesions contributing to the extended tails of the distributions. This information is important to consider, as the presence of skewed data can impact the performance and interpretability of our models. To account for this skewness, it may be beneficial to apply data transformations, such as a logarithmic transformation, to normalize the distributions.

2 Methods

Since the data consists of both tabular and image data, a multi-modal approach was employed to effectively analyze and model the information. The tabular features, such as the measurements of the lesion perimeter and area, can be leveraged using traditional machine learning techniques. However, the image data will require specialized computer vision methods to extract relevant visual features and incorporate them into the analysis.

2.1 Methods for Tabular Data

Prior to classification, the tabular data underwent preprocessing to address skewness in numerical features such as perimeter and area. A logarithmic transformation was applied to normalize these values. Initially, the dataset contained approximately 40 features. Feature engineering expanded this to about 150 features, improving the dataset's predictive capacity (Table 1) [1].

For classification, the LightGBM gradient boosting model was selected due to its efficiency and suitability for large, high-dimensional datasets. LightGBM constructs sequential decision trees, with each tree minimizing the residual errors (gradients) of its predecessors. Compared to models like XGBoost, LightGBM is faster and more resource-efficient. The binary classifier was trained with the following parameters: metric: binary_logloss, boosting_type: gbdt, num_leaves: 31, and learning_rate: 0.05.

2.2 Methods for Image Data

For the image data, a transfer learning approach was implemented using EfficientNetV2-B2 as our backbone architecture. This model was chosen for its efficient balance of computational cost and performance. The network was pre-trained on ImageNet, providing it with robust feature extractors developed from exposure to millions of diverse images. To perform transfer learning, all the layers in the CNN were frozen except for the last convolutional and dense layer. The images were pre-processed by resizing them to 128 by 128 pixels, and normalized using the ImageNet means and standard deviations. During the fine-tuning process, data augmentations such as rotation, shear, zoom, width/height shift, and vertical/horizontal flip were used. The CNN was trained using the Adam optimizer and a cosine anealing learning rate scheduler with the following hyper-parameters: learning rate: 1e-4, batch size: 32.

2.3 Methods for Fusion Data

For fusing the tabular and image data, a intermediate fusion approach was taken. The predictions from the image model were added as an additional feature to the tabular data, and a LightGBM model was trained on this new tabular data. This specific fusion strategy was chosen based on empirical performance evaluations. When assessed independently, the tabular model demonstrated substantially higher accuracy compared to the image-based model. This performance disparity suggested that the features extracted from the images were less reliable predictors than the structured tabular data. Therefore, a more aggressive fusion approach (such as equally weighting both modalities or using complex neural architectures to combine them) could potentially degrade the overall model performance by giving too much weight to the less reliable image features. The fusion model was trained with the following parameters: metric: binary_logloss, boosting_type: gbdt, num_leaves: 31, and learning_rate: 0.05.

3 Experiments and Results

Due to the imbalance in the data which exhibited a distribution of 99.902134% for the negative class and 0.097866% for the positive class, the data was sampled to achieve a smaller imbalance of 67.120419% for the negative class and 32.879581% for the positive class. Stratified K-fold cross validation was also utilized to split the data into training and testing sets. Stratified K-fold cross validation is a variation of KFold that preserves the percentage of samples for each class in each fold. This approach makes sure that each class is adequately represented across all folds, allowing for a fair and accurate evaluation of the model's performance.

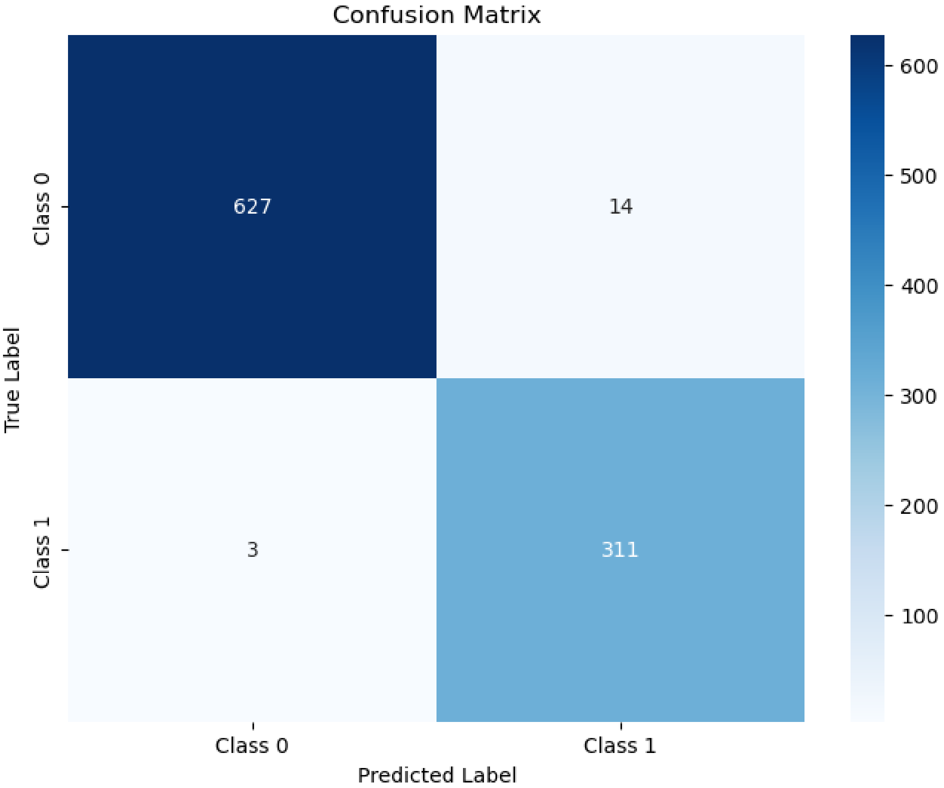

Figure 2: Confusion matrix for the tabular data model. For classification metrics: accuracy: 0.9822, F1: 0.9734, precision: 0.9569, recall: 0.9904.

For the tabular data, the overall cross validation accuracy achieved was 0.97738, with the best fold achieving an accuracy of 0.9822 (Figure 2).

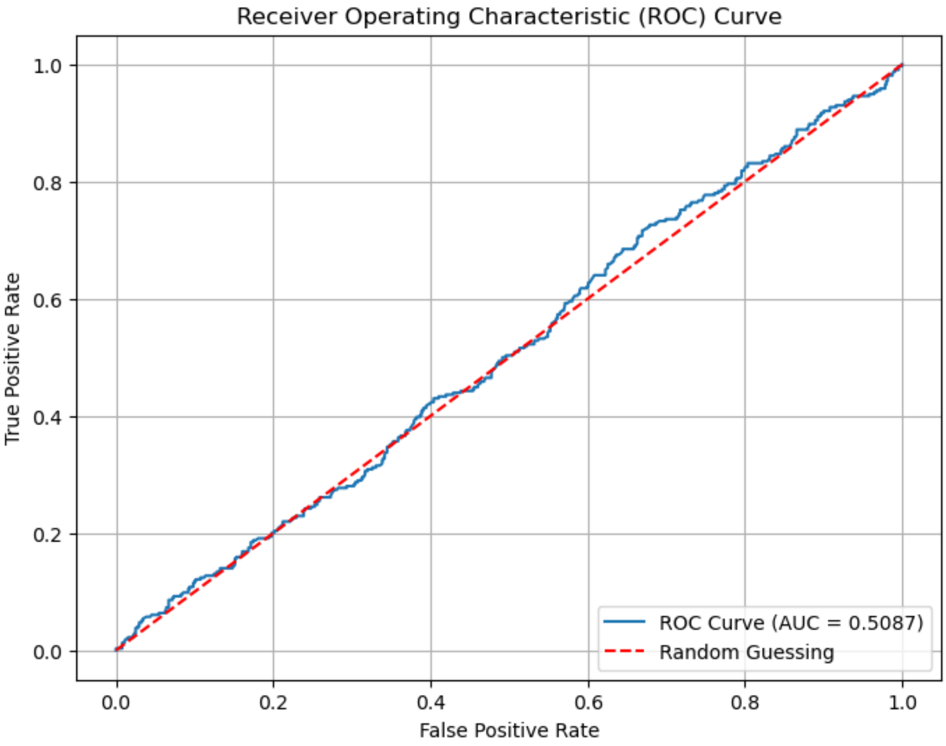

Figure 3: ROC for the image model. This model is about the same as random guessing.

For the image data, the overall cross validation accuracy was 0.53340, and the AUC was 0.49723. The best fold resulted in an accuracy of 0.67120 and an AUC of 0.50345 (Figure 3).

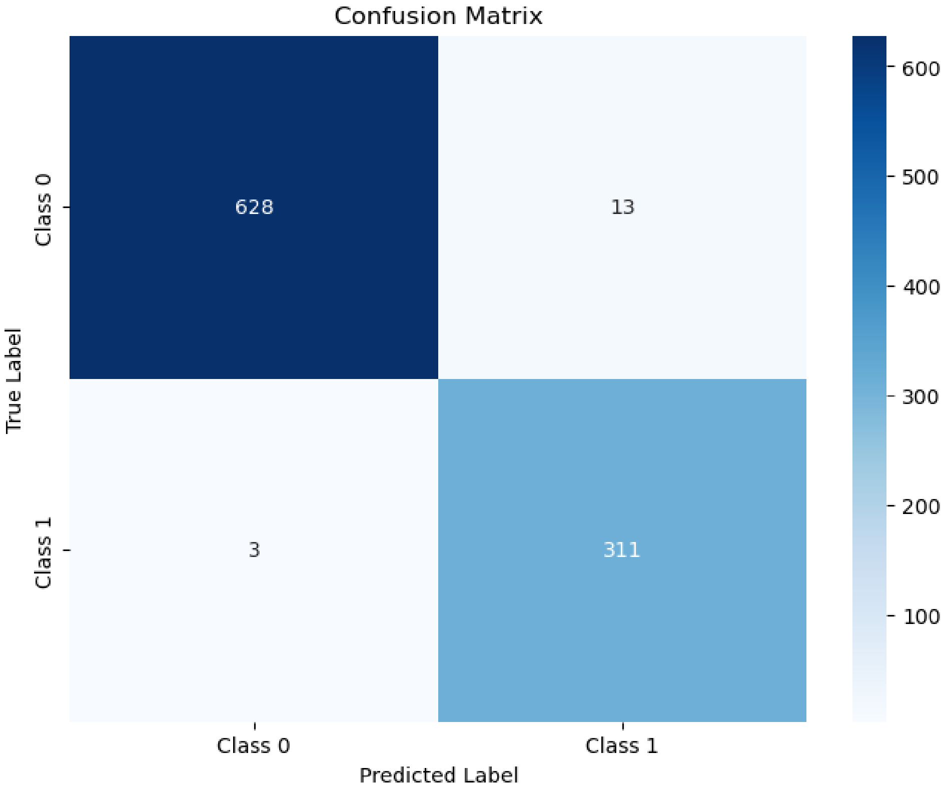

Figure 4: Confusion matrix for the fusion model. For classification metrics: accuracy: 0.9832, F1: 0.9749, precision: 0.9599, recall: 0.9904.

For the fusion model, the overall cross validation accuracy was 0.97654, and an average binary log loss of about 0.07. The best fold resulted in a binary log loss of 0.07473 and accuracy of 0.9832 (Figure 4).

4 Discussions

When comparing the three models, the tabular model far outperformed the image model, and was on par with the fusion model. During the training of the image model, there was severe overfitting, with the model tending the predict all images as the negative class. When overfitting prevention techniques were added, such as dropout and L2 regularization, the model would flip all its predictions to the positive class. This could have been due to the mild class imbalance prevalent in the data. When compared to models such as LightGBM for tabular data, CNNs are more susceptible to class imbalance issues, which explains the large disparity between the tabular and image model. The fusion model resulted in a minuscule improvement compared to the tabular model, indicating that the predictions from the image model had little to no importance in classification.

When analyzing the importance of features determined by the fusion model (Figure 5), the top 10 features tended to reflect some aspect of the color, diameter, or border of the lesion. When compared to characteristics of skin lesions that dermatologists use to diagnose melanomas, the features deemed important by the model fall in line with the same features that dermatologists use. By comparing the results of these models with dermatologists, the validity and robustness of the model can be assessed and validated, suggesting that the model is learning clinically relevant features rather than spurious correlations in the data.

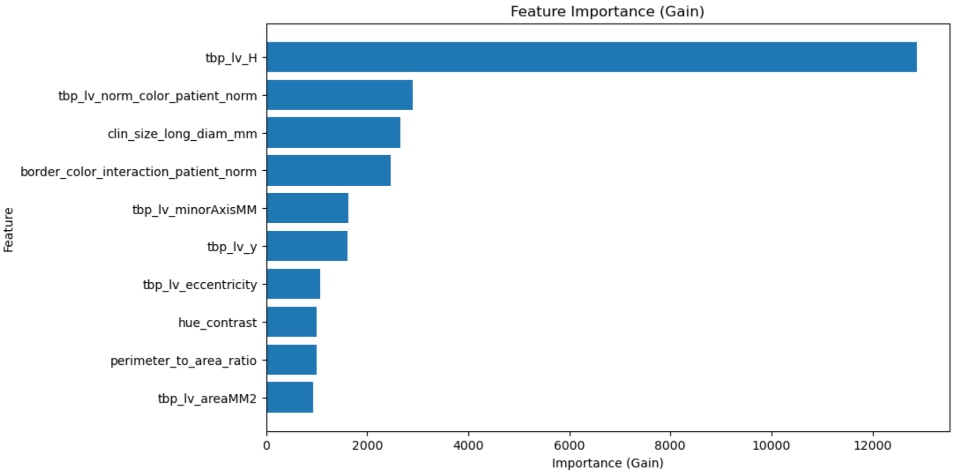

Figure 5: Feature importance determined by the fusion model.

5 Conclusion

For the test data, the fusion model was trained with the following parameters: metric: binary_logloss, boosting_type: gbdt, num_leaves: 31, and learning_rate: 0.05. One challenge with this data was the class imbalance. Benign lesions typically outnumbered malignant ones in the dataset, reflecting real-world prevalence but creating challenges for model training. This imbalance can lead to biased predictions favoring the majority class if not properly addressed through techniques like class weighting or balanced sampling. The tabular model significantly outperformed the image model primarily because the tabular data contained highly structured and standardized diagnostic features that were manually extracted by experienced dermatologists. These features, including precise measurements, standardized assessments of color variations, border irregularities, and specific morphological characteristics, represented decades of clinical knowledge distilled into quantifiable metrics. In contrast, the image model had to learn to extract relevant features from raw photographs. The structured nature of the tabular data, combined with its expert-driven feature engineering, provided a more reliable basis for classification compared to the variable image data.

6 Code Availability

The project notebook is available at `https://github.com/kkuang0/CS640_Final_Project`.

References

References

[1] Ilya Novoselskiy, ”ISIC-2024”. This is the first place submission for the ISIC-2024 challenge which provided the engineered features. https://github.com/ilyanovo/isic-2024

[2] ”ISIC2024: KerasCV Starter” . This is the starter notebook provided by Keras for the computer vision portion of this project. https://www.kaggle.com/code/awsaf49/isic2024-kerascv-starter

Engineered Features	
Feature Name	Description
lesion_size_ratio	Ratio of lesion minor axis to the clinical long diameter.
lesion_shape_index	Ratio of lesion area to the square of its perimeter.
perimeter_to_area_ratio	Ratio of lesion perimeter to area.
area_to_perimeter_ratio	Ratio of lesion area to perimeter.
hue_contrast	Absolute difference between lesion's internal and external hue.
luminance_contrast	Absolute difference between internal and external luminance.
lesion_color_difference	Euclidean distance in color space using delta L, A, B.
overall_color_difference	Average difference across color channels.
color_variance_ratio	Ratio of mean color standard deviation to external color standard deviation.
mean_hue_difference	Average hue difference between internal and external.
std_dev_contrast	Standard deviation of color channel differences.
border_complexity	Combined complexity from normalized border and 2-axis symmetry.
symmetry_border_consistency	Multiplicative interaction between symmetry and border.
consistency_symmetry_border	Ratio of symmetry-border product to their sum.
color_uniformity	Ratio of mean color standard deviation to maximum radial color standard deviation.
color_asymmetry_index	Product of 2-axis symmetry and radial color standard deviation.
color_shape_composite_index	Composite index combining color standard deviation, area-to-perimeter ratio, and symmetry.
position_distance_3d	Euclidean distance in 3D space using x, y, z coordinates.
lesion_orientation_3d	Angle of lesion position in 3D space.
volume_approximation_3d	Approximation of lesion volume as area multiplied by 3D position distance.
lesion_visibility_score	Combination of color normalization and delta LB normalization.
lesion_severity_index	Average of border normalization, color normalization, and eccentricity.
shape_complexity_index	Sum of border complexity and shape index.
comprehensive_lesion_index	Average of area-to-perimeter ratio, eccentricity, normalized color, and symmetry.
color_contrast_index	Summation of color channel differences and delta LB normalization.

Table 1: Engineered features [1].