IT1244 Project Report: Cancer Detection

Team 8: Kim Minjun, Jeon Jinkyung, Pham Ngoc Minh, Divyalakshmi Balasubramaniam

Introduction

Cancer remains a critical and urgent healthcare challenge, with progression and mortality outcomes highly dependent on timely diagnosis and therapeutic interventions. Current methods like biopsies, which are essential for definitive diagnosis, are both time-consuming and resource intensive (*Tests and Procedures Used to Diagnose Cancer - NCI*, 2015). Gene expression analysis can be instrumental in early cancer detection, as changes in gene expression serve as biomarkers that can indicate early-stage cancers (Alharbi & Vakanski, 2023).

In recent literature, various machine learning (ML) and deep learning (DL) models are widely used for cancer classification. Common ML approaches include Support Vector Machines (SVM), k-Nearest Neighbors (kNN), and Random Forest (RF) (Alharbi & Vakanski, 2023). For our cancer prediction task, we used Logistic Regression, SVM, and a voting classifier. Logistic Regression serves as a benchmark for its simplicity. SVM was chosen for its reported performance advantages in classification tasks (Huang et al., 2017). We also explored tree-based models by using a voting classifier composed of XGBoost, Random Forest, and Logistic Regression. Tree-based methods are known to be robust to outliers and handle irrelevant dependent variables (Cutler et al., 2009).

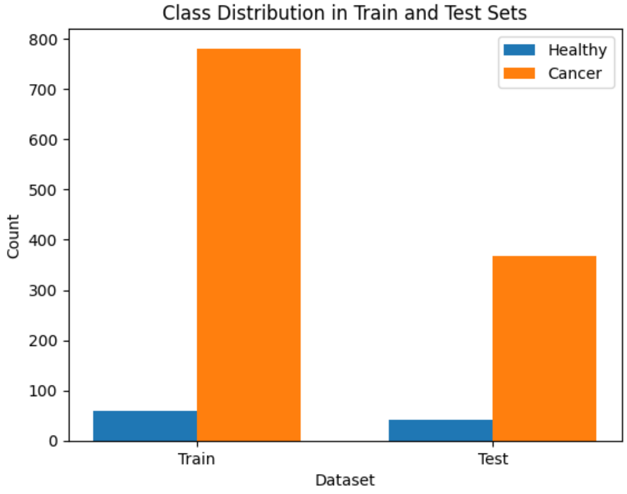
Gene expression datasets often present challenges with small sample sizes and high dimensionality, making most ML methods heavily reliant on feature-engineering techniques to reduce redundancy and select optimal features for classification. Thus, the quality of selected features significantly impacts their performance.

DL models, known for detecting unique gene expression patterns across cancer types, include multi-layer perceptron (MLP), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers (Alharbi & Vakanski, 2023). However, due to limited data, we restricted our approach to simpler models, specifically perceptron, as DL methods typically require large datasets for strong performance.

Dataset

The dataset contains two CSV files: “Train_Set” and “Test_Set.” Both files have 350 features (columns named length_51 to length_400) that represent normalized frequencies of DNA fragment lengths. The training set includes around 850 samples with an additional column (class_label) indicating whether a sample is healthy or cancerous. The test set contains around 400 samples for which the class label needs to be predicted.

One of the significant challenges was the class imbalance in the Train_Set, where there were 781 samples labeled as 'cancer' and only 60 samples labeled as 'healthy'. This large discrepancy causes the 'cancer' samples to dominate as the majority class, while the 'healthy' samples are underrepresented as the minority class. As a result, many machine learning algorithms prioritize the majority class, leading to a bias toward predicting 'cancer' more frequently and potentially overlooking the 'healthy' samples. This imbalance can cause the model to achieve deceptively high accuracy, as it predominantly predicts the majority class, but it may perform poorly on the minority class, which is often of higher importance in medical contexts. In such cases, false negatives (misclassifying healthy individuals as cancer patients) are particularly costly.



Methods

The analysis begins with the cancer dataset, which contains 350 features relevant to distinguishing between classes. To ensure consistency across variables, the data is standardized. Once standardized, exploratory data analysis (EDA), is conducted. Pearson’s R correlation heatmap between the features and pair plots of the least correlated features were plotted to visualize multicollinearity.

Class Balancing

Since the dataset exhibits class imbalance, it is important to address this issue to prevent biased models. Two techniques are applied for class balancing: NearMiss and SMOTE.

Near Miss under samples the points that are the closest to the minority sample to draw a more specific geometric border between the two classes. Near Miss was exclusively considered for SVM, a geometrical machine learning model, and it performed well for our SVM, and thus was used when running SVM models (Mqadi et al., 2021).

SMOTE combines over-sampling of the minority class and under-sampling of the majority class to enhance classifier performance. It was reported to be the most effective method, particularly for tree-based models (Han et al., 2024). To maintain reliable evaluation, only the training set was augmented using SMOTE, leaving the validation and test sets untouched. For the perceptron, logistic regression, and voting classifier, SMOTE was applied to the training data.

Support Vector Machine (not covered in IT1244)

Support Vector Machine (SVM) was chosen as a classification model for our cancer dataset due to its effectiveness in handling high-dimensional data and its robustness against issues that arise from non-linear boundaries, which were apparent in our dataset. SVM is particularly suitable for datasets like ours where the classes are not easily separable with simple linear boundaries. By leveraging a kernel trick, SVM can map the input features into higher-dimensional spaces, allowing it to construct an optimal separating hyperplane even in complex and non-linear cases (Dudzik et al., 2021). Overall, SVM's flexibility, adaptability to non-linear separations, and robustness to class imbalance made it a strong choice for our cancer dataset.

However, applying SVM to our raw cancer dataset revealed several limitations that affected its performance. The high dimensionality of DNA fragment variables increased computational demands, particularly with non-linear kernels. Additionally, the inherent class imbalance challenged the model's sensitivity, as SVM favored the majority class despite weight adjustments. Noise and outliers in fragment length data further impacted robustness, disrupting the decision boundary and necessitating extensive preprocessing to mitigate these effects.

Class balancing was first done with Near Miss as mentioned above. Then, two alternative methods for dimensionality reduction were considered: Principal Component Analysis (PCA) followed by Linear Discriminant Analysis (LDA), and just LDA. PCA followed by LDA is a more conventional method, as LDA does not run well on datasets with high dimension and collinear features, which are both removed by PCA to enhance the effectiveness of LDA. However, in practice, the data preprocessed only by LDA performed better, which could be the case if the multicollinearity does not hinder the process of LDA significantly. To validate our choice of LDA, we generated a 2D scatter plot of the first two LDA components. The plot "LDA Projection of Healthy vs. Cancer" in the appendix shows clear separation between cancer (blue) and healthy samples (red), confirming the suitability of LDA for our dataset.

Logistic Regression

Logistic regression is a statistical method for binary classification, where the outcome is categorical (e.g., 0 or 1). The sigmoid function $p(z) = \frac{1}{1+e^{-z}}$ is used to model the relationship between the input features and the probability that the output variable is 1. We chose logistic regression because our dataset is binary and, with regularization, it effectively handles large feature sets without overfitting.

In our model, we set *max_iter = 5000* to ensure convergence for the large dataset, *random_state = 42* for consistent results, *solver = 'saga'* for efficiency and compatibility with all penalties on large datasets, *penalty = 'l2'* to prevent overfitting, *C = 1* as a balanced regularization value, *tol = 1e-5* for higher solution accuracy through lower tolerance.

Voting Classifier (not covered in IT1244)

To improve classification performance, we implemented a Voting Classifier, an ensemble method combining three models: Logistic Regression, XGBoost, and Random Forest. This approach leverages each model's strengths, enhancing overall accuracy and robustness by using a soft voting strategy, where each model outputs a probability for each class (Nguyen et al., 2024). The final prediction is based on the averaged probabilities, which often improves performance in well-calibrated models.

The Voting Classifier was chosen for its ability to combine different perspectives on the data: Logistic Regression offers a linear view, XGBoost captures complex, non-linear relationships, and Random Forest reduces variance. Using soft voting, we benefit from the calibrated probability outputs of each model, enhancing the classifier's overall sensitivity and specificity. This approach provides a balanced model that captures both the linear and non-linear relationships in the data, resulting in improved predictive performance.

In the Voting Classifier method, we applied *voting='soft', weight= [1, 1, 2]* to give logistic regression additional influence due to its stronger standalone

Perceptron

Our initial goal was to implement an MLP; however, subsequent experiments with different numbers of hidden layers, neurons, and activation functions showed that a perceptron was best suited to the task.

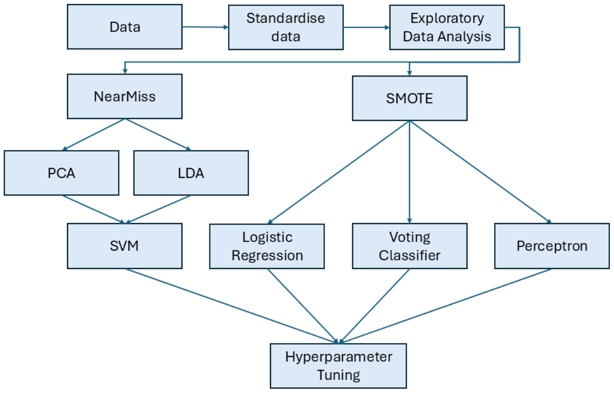
The model architecture consisted of an input layer that accepts a feature vector of size 350, followed by a single dense output layer with a sigmoid activation function. The absence of any additional dense layers implies that this model effectively operates as a logistic regression model, where the single output layer directly predicts the probability of the positive class. This design decision was influenced by the relatively small size of our dataset, which limits the capacity for more complex architectures without risking overfitting.

To better prepare the model for the validation dataset, SMOTE was applied exclusively to the training data, with 50% of the samples generated synthetically and the other 50% consisting of original data. This balance helps reduce overfitting to synthetic samples while enhancing the model's ability to handle real-world class imbalance.

For training, we compiled the model using binary cross-entropy as the loss function, which is standard for binary classification. We optimized the model using the Adam optimizer, known for its efficient handling of large datasets and adaptive learning rate adjustments. The model's performance was assessed using Precision, Recall, and the Area Under the Curve (AUC) as evaluation metrics, with AUC serving as our primary metric due to its ability to capture the model's discrimination capacity across different threshold settings.

The model was trained for 50 epochs with a batch size of 32, using the balanced training data and evaluated on a separate validation dataset (unbalanced).

The methods follow as the diagram below:



Results & Discussions

Each model was fine-tuned to ensure optimized performance in its own way. For SVM, the first optimization came with class balancing. Considering the robustness of SVM against imbalance data, a combination of Near Miss with other sampling methods like SMOTE or cluster centroids was considered. However, these did not perform well, and class balancing was done solely with Near Miss, at a disadvantage of reduced data size. Then, each parameter was tuned manually. Kernel parameter gamma was first tuned followed by complexity parameter C, as gamma affects the decision boundary, and C affects the margin complexity. Each set of NumPy arrays were iterated through to find the value that returns highest precision. The graph of this parameter tuning can be found in the appendix.

After hyperparameter tuning, recall and F1 scores for the minority class ('healthy') and AUC_ROC score improved in the Voting Classifier but not in Logistic Regression. This is due to Logistic Regression's regularization insensitivity, where minor changes in C don't significantly impact performance if the data is well-represented. As a simple linear model, Logistic Regression often reaches near-optimal results without extensive tuning, especially with linearly separable or SMOTE-unsampled data, which reduces class imbalance. Additionally, AUC-ROC, as an evaluation metric, can be stable against small changes, leading to seemingly unchanged results post-tuning.

During training, the perceptron was evaluated by tracking training and validation loss, precision, recall, and AUC. The training and validation curves had a similar general shape, but the validation curve was unstable and fluctuating. This was expected, as the validation data was not balanced, leading the model to underperform and produce unstable results on the minority class. We validated our choice of model complexity through manual hyperparameter tuning, testing only for additional dense layers. However, these experiments confirmed that the simpler architecture yielded more stable and generalizable performance (in the appendix). Other metrics were adjusted during model development, but extensive hyperparameter tuning was not considered necessary due to the data size and the tradeoff between performance and tuning complexity.

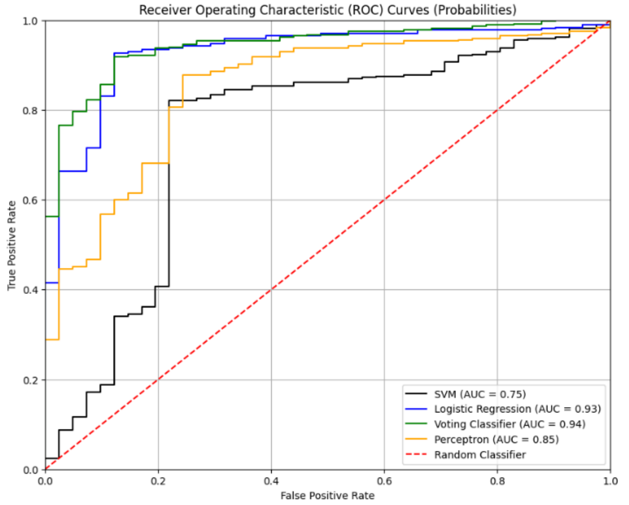
Performance Evaluation

We evaluated the performance of the tuned models using several metrics, including precision, recall, and AUC. Precision and recall were specifically chosen because they explicitly reflect the number of false positives and false negatives, whereas metrics like accuracy tend to be misleading in an imbalanced dataset if the number of positives is a majority class.

Model	**Class**	**Precision**	**Recall**
SVM	Healthy	0.32	0.78
	Cancer	0.97	0.82
LR	Healthy	0.52	0.88
	Cancer	0.99	0.91
Voting	Healthy	0.58	0.80

Classifier	Cancer	0.98	0.93
Perceptron	Healthy	0.29	0.78
	Cancer	0.97	0.79

Additionally, the AUC-ROC curves were plotted. Given our imbalanced dataset, AUC was used as the key metric to determine the best model, as it measures the model's ability to discriminate between positive and negative instances across all classification thresholds.



Logistic regression and the voting classifier performed best on the high-dimensional, correlated data. Simpler models like logistic regression often generalize better on smaller datasets, as they require less data to perform effectively. Although literature suggests that SVMs can handle high-dimensional predictors efficiently, our results showed that SVM underperformed in this context. A limitation of SVM is its lack of accuracy in estimating class membership probabilities (Zhang et al., 2007). Additionally, since SVM is not inherently a probabilistic model, it tends to be underrepresented in ROC-AUC comparisons with other models, which must be considered. More complex models like SVM and perceptron may struggle to capture patterns effectively without additional data. Perceptron also underperformed, likely because the dataset size was insufficient to fully train the model while further training would lead to overfitting. Overall, logistic regression and the voting classifier clearly outperformed SVM and perceptron with this dataset.

Human vs AI

Breast cancer screening programs have a notable miss rate, with studies indicating they can overlook between 15% and 35% of cancers due to detection errors or because the cancer is not visible to the radiologist (Freeman et al., 2021). This can lead to missed cancers that later present as interval cancers. Hofvind et al. reported the sensitivity of mammography with a 2-year follow-up at 74.9% and with a 1-year follow-up at 82.0% (Chubak et al., 2022).

In comparison, our best-performing model, the voting classifier, demonstrates significantly higher sensitivity, achieving 0.80 for healthy cases and 0.93 for cancer cases. These results are promising for the use of AI models as a screening tool. It is important to note that the model is not ready for deployment; it requires further training on more data and context regarding the dataset. Additionally, proper feature extraction must be implemented to reduce noise in the input. Importantly, the goal of our model is not to surpass human detection but to complement it as a first step screening tool, enhancing overall diagnostic capabilities.

Impacts

The societal impact of this project is multifaceted. Privacy is crucial, especially if the data used includes sensitive patient information, highlighting the need for stringent data security protocols to prevent unauthorized access. Fairness is also a concern, as models trained on biased datasets may inadvertently reinforce existing disparities, which could be harmful in contexts like healthcare where impartial treatment is critical. Interpretability is essential, as decisions made by the model need to be understandable and explainable, particularly if it is used in clinical decision-making. Ensuring that healthcare providers can comprehend the model's reasoning supports transparency and trust in its outputs.

Finally, while the model could potentially impact jobs by automating certain tasks, it also creates opportunities for healthcare professionals to shift their focus to predictive and personalized healthcare, enhancing overall patient care.

Recommendations for future improvements

To enhance the robustness and generalizability of our cancer detection model, we recommend increasing dataset size and diversity, improving validation methods, and addressing bias concerns. The current dataset's small size and class imbalance limit the model's generalization. Expanding the dataset through combining gene expression data or using data augmentation would help the model capture diverse patterns across various cancer stages, thereby strengthening its predictive power.

Currently, our model validation is based only on internal test data, which may not fully capture real-world cases. To address this, we recommend validating the model using external datasets and conducting longitudinal studies. This would allow us to assess its performance across various demographics, cancer subtypes, and stages, ensuring that the model is stable and reliable in real-world scenarios. By expanding validation efforts, we can gain a more complete understanding of the model's capabilities and limitations.

References

Alharbi, F., & Vakanski, A. (2023). Machine Learning Methods for Cancer Classification Using Gene Expression Data: A Review. *Bioengineering*, *10*(2), 173. https://doi.org/10.3390/bioengineering10020173

Chubak, J., Burnett-Hartman, A. N., Barlow, W. E., Corley, D. A., Croswell, J. M., Neslund-Dudas, C., Vachani, A., Silver, M. I., Tiro, J. A., & Kamineni, A. (2022). Estimating cancer screening sensitivity and specificity using healthcare utilization data: Defining the accuracy assessment interval. *Cancer Epidemiology, Biomarkers & Prevention: A Publication of the American Association for Cancer Research, Cosponsored by the American Society of Preventive Oncology*, *31*(8), 1517. https://doi.org/10.1158/1055-9965.EPI-22-0232

Cutler, A., Cutler, D. R., & Stevens, J. R. (2009). Tree-Based Methods. In X. Li & R. Xu (Eds.), *High-Dimensional Data Analysis in Cancer Research* (pp. 1–19). Springer. https://doi.org/10.1007/978-0-387-69765-9_5

Dudzik, W., Nalepa, J., & Kawulok, M. (2021). Evolving data-adaptive support vector machines for binary classification. *Knowledge-Based Systems*, *227*, 107221. https://doi.org/10.1016/j.knosys.2021.107221

Freeman, K., Geppert, J., Stinton, C., Todkill, D., Johnson, S., Clarke, A., & Taylor-Phillips, S. (2021). Use of artificial intelligence for image analysis in breast cancer screening programmes: Systematic review of test accuracy. *BMJ*, *374*, n1872. https://doi.org/10.1136/bmj.n1872

Han, Y., Wei, Z., & Huang, G. (2024). An imbalance data quality monitoring based on SMOTE-XGBOOST supported by edge computing. *Scientific Reports*, *14*(1), 10151. https://doi.org/10.1038/s41598-024-60600-x

Huang, M.-W., Chen, C.-W., Lin, W.-C., Ke, S.-W., & Tsai, C.-F. (2017). SVM and SVM Ensembles in Breast Cancer Prediction. *PLOS ONE*, *12*(1), e0161501. https://doi.org/10.1371/journal.pone.0161501

Mqadi, N. M., Naicker, N., & Adeliyi, T. (2021). Solving Misclassification of the Credit Card Imbalance Problem Using Near Miss. *Mathematical Problems in Engineering*, *2021*(1), 7194728. https://doi.org/10.1155/2021/7194728

Nguyen, M., Cao-Van, K., Minh, L. G., Bui, T. X., & Hong, S. (2024). *Hybrid Machine Learning Models Using Soft Voting Classifier for Financial Distress Prediction* (SSRN Scholarly Paper No. 4941751). Social Science Research Network. https://doi.org/10.2139/ssrn.4941751

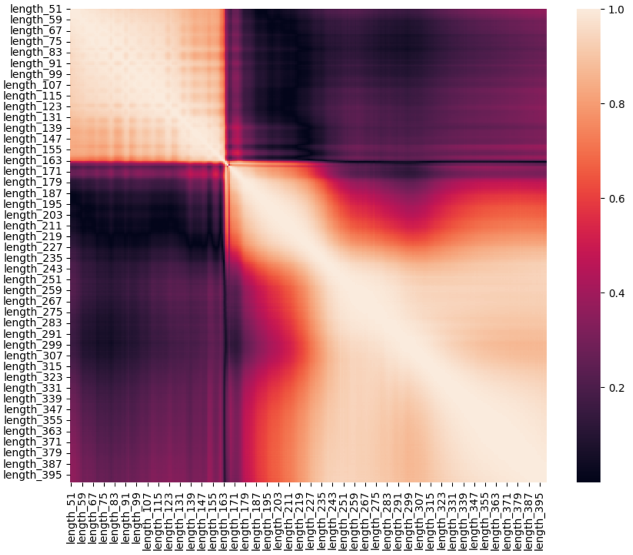
Tests and Procedures Used to Diagnose Cancer—NCI (nciglobal,ncienterprise). (2015, March 9). [cgvArticle]. https://www.cancer.gov/about-cancer/diagnosis-staging/diagnosis

Zhang, C., Fu, H., Jiang, Y., & Yu, T. (2007). High-dimensional pseudo-logistic regression and classification with applications to gene expression data. *Computational Statistics & Data Analysis*, *52*(1), 452–470. https://doi.org/10.1016/j.csda.2006.12.033

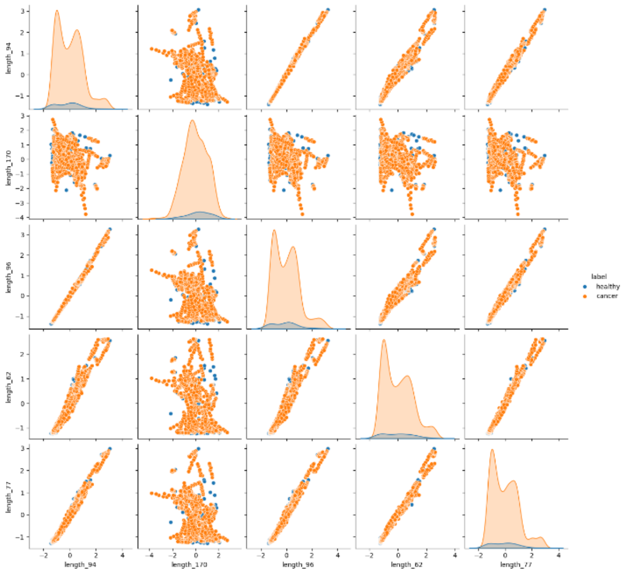
Appendix

EDA

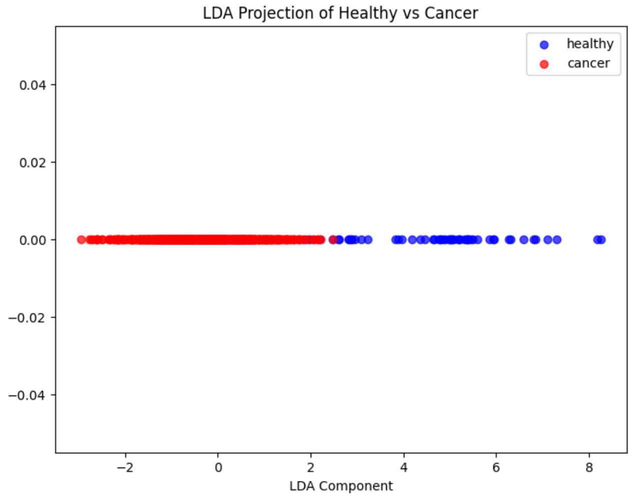
Pearson's R was calculated between the 350 features to determine if the features were correlated to one another. Majority of the features are multicollinear. Multicollinearity occurs when two or more predictor variables in a model are highly linearly correlated with each other. It can negatively impact models by increasing variance and making it difficult to determine the significance and effect of individual predictors.



To explore the dataset based on the least-correlated features, each feature's average correlation with others was calculated, and these values were sorted to identify the 50 features with the lowest average correlations. From this subset, 5 features were randomly selected for further analysis. The pair plot visualizes pairwise relationships between five random features (length_63, length_66, length_64, length_79, and length_74. Despite being selected for low correlation, the scatter plots reveal some linear relationships, suggesting residual collinearity among these features. This could result in overfitting for the ML and DL models.



LDA



SVM

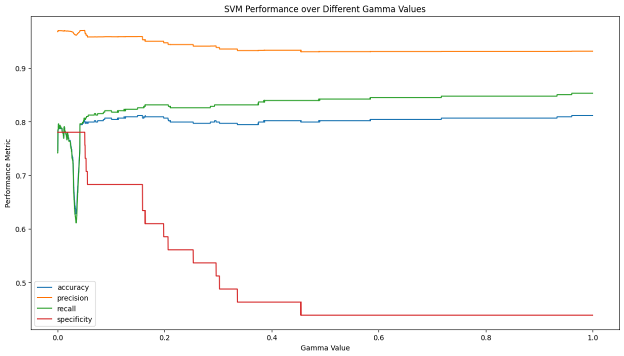


Figure of hyperparameter tuning – gamma

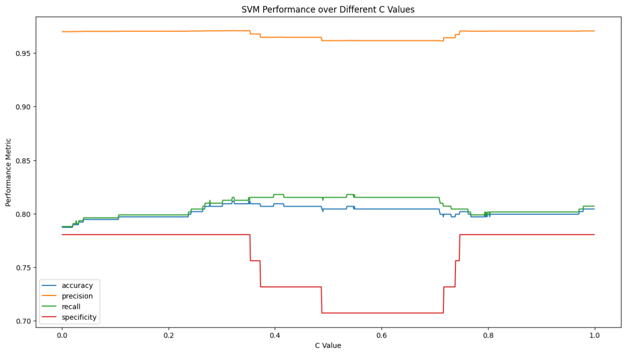


Figure of hyperparameter tuning – C

Voting Classifier

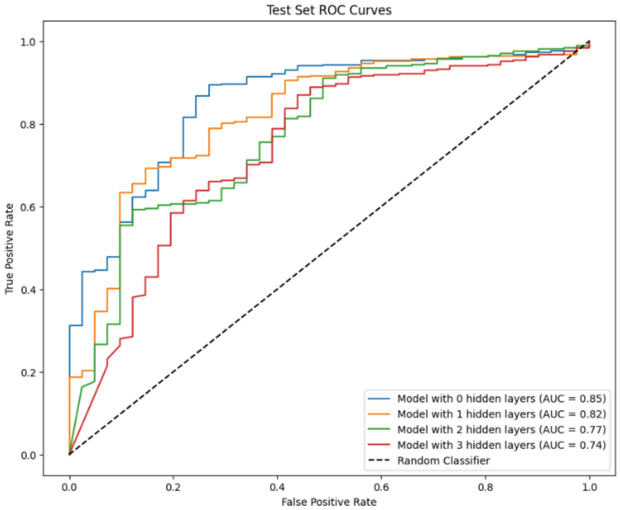
We used the XGBoost model because of its strong performance on complex datasets and it is adding a non-linear perspective to the ensemble. In the model, we used *booster='gbtree'* and *objective='binary:logistic'* to specify gradient boosting with logistic regression for binary classification, *eta=0.2* (learning rate) to balance the learning rate and number of estimators for optimal learning, *max_depth=5* and *min_child_weight=1* to control the tree's depth and splits to prevent overfitting, *subsample=0.8* and *colsample_bytree=0.8* to control the fraction of samples and features per tree for regularization, *scale_pos_weight=1* because we did balancing the class by SMOTE, eval_metric='auc' to select AUC metric for its relevance in imbalanced binary classification, *n_estimators=100* to specify the number of boosting rounds, *reg_lambda=1* and *reg_alpha=0* to add ridge (L2) regularization, with no L1 regularization.

We chose Random Forest because it brings robustness to the ensemble by averaging multiple decision trees, reducing variance. In the model, we set *n_estimators=200* to set a higher number of trees for stability in predictions, *max_depth=20* and *min_samples_split=10* to limit tree depth and enforce a minimum of 10 samples per split to reduce overfitting, *min_samples_leaf=5* to ensure a minimum of five samples in each leaf node for further regularization, *max_features='sqrt'* to use the square root of features for each split, balancing diversity in tree nodes, *bootstrap=True* to enable resampling for each tree to reduce overfitting, *n_jobs=-1* to utilize all processors for faster computation.

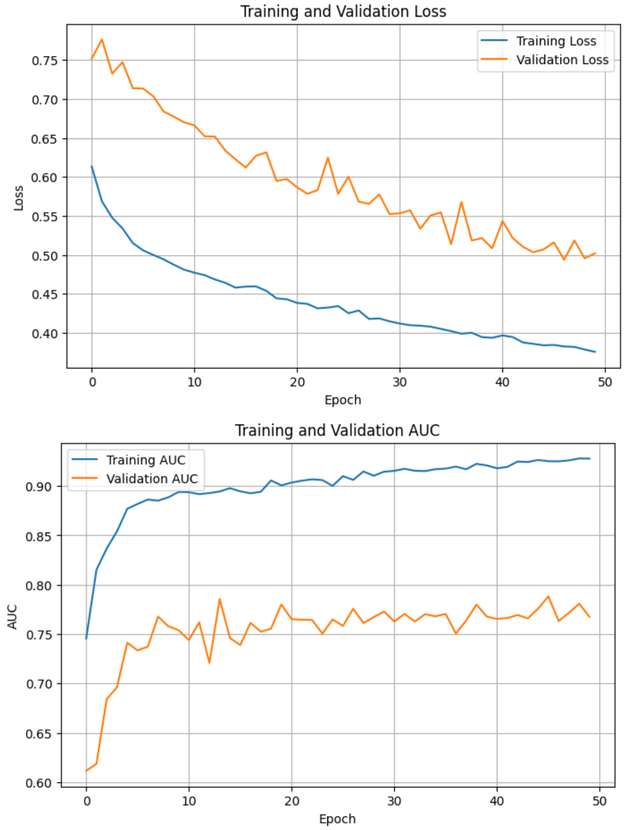
We used the above Logistic Regression model for this classifier.

In the Voting Classifier model, we applied *voting='soft', weight= [1, 1, 2]* to give logistic regression additional influence due to its stronger standalone performance.

Perceptron



Hyperparameter Tuning of dense layers



s

Model Training and Validation Curves