2023

DS 7333 | Quantifying the World

FINAL EXAM | COST OPTIMIZED CLASSIFICATION

JOEY HERNANDEZ | DANNY CHANG

Introduction

Background

In an increasingly data-driven business environment, the ability to accurately predict outcomes is paramount. The challenge often lies not just in the prediction itself but in the cost associated with the inaccuracies of such predictions. False predictions can be classified into two main categories: false positives and false negatives, each carrying its own implications and costs. A false positive occurs when the model incorrectly predicts a positive outcome, while a false negative is when the model fails to identify a positive outcome. The impact of these inaccuracies varies based on the nature of the business and the specific circumstances surrounding the predictive model's application.

Objective

The primary objective of our predictive modeling endeavor is to develop a binary classification model that can discern between 'Yes' (positive) and 'No' (negative) outcomes with the highest possible accuracy. The model will be applied to an anonymized dataset with undisclosed features, emphasizing the minimization of the associated cost of incorrect predictions. Given the cost structure, where a false positive is penalized five times more heavily than a false negative ($100 versus $40), the model must prioritize the reduction of false positives without significantly increasing the false negatives. The ideal model will strike a balance between sensitivity (true positive rate) and specificity (true negative rate) to achieve the lowest possible total cost to the business. Various modeling techniques will be explored and evaluated based on their performance against this cost function to ensure the most economically advantageous outcome.

Data Inspection

Target Distribution

The dataset at hand comprises 160,000 original observations, aimed at predicting a binary outcome—essentially a 'Yes' (1) or 'No' (0). The target variable distribution indicates a relatively balanced dataset with 95,803 instances of the 'No' class and 64,197 of the 'Yes' class. Such a distribution is beneficial as it does not heavily favor one class over the other, which can be a common issue in binary classification problems leading to a biased model.

Missing Values

Upon initial inspection, the dataset contains 1,608 missing values. The missing data represents 1% of the total observations, which is a relatively small proportion. There is no discernible pattern or correlation between the missing values and other variables in the dataset. Given the absence of a relationship and the comparatively low percentage of missing data, the decision has been made to proceed with the removal of these instances. This approach is deemed appropriate as it simplifies the modeling process without the risk of introducing bias or inaccuracies that could come with imputation methods.

Numerical Variables

The numerical features in the dataset exhibit a wide range of magnitudes, indicating the presence of variables measured on different scales. This observation suggests a need for normalization or standardization in the preprocessing phase to ensure that no single attribute dominates the model due to its scale.

Categorical Variables

The dataset includes several categorical variables which provide classificatory information:

- Country of origin with categories such as Asia, Europe, and America.
- Month of the year from January to December.
- Day of the week covering Monday to Friday.

These categorical variables are crucial for the model as they might hold significant predictive power. They will require appropriate encoding to transform them into a format that can be effectively utilized by the predictive algorithms.

Financial Values

Some columns represent financial figures initially in dollar terms, which have been converted to floating-point numbers to maintain consistency and to facilitate computation. Likewise, a percentage column has been adjusted by dividing by 100 and converting to a float. These transformations are essential for aligning all numerical variables on a similar scale and for preparing the data for analysis by various statistical and machine learning methods.

In summary, the initial data inspection has laid the groundwork for further preprocessing. The subsequent steps will involve handling missing values, transforming categorical variables, and normalizing numerical values to ensure the data is in an optimal form for modeling.

Figure 1: *Count plot illustrating the distribution of the classification target.*

Description*: The count plot displays the class distribution for our target variable, highlighting the difference between the two instances. This visualization aids in identifying class balance present in the dataset.*

Table 1: *Table detailing the distribution of target class.*

Distribution of Target Variable		
Class	**Class Counts**	**Class Percentages**
0 (No)	95803	59%
1 (Yes)	64197	41%

Description*:* The table presents the distribution of occurrences of the data *target* variable. It succinctly highlights the number of instances for each class and the corresponding percentages.

Modeling

Preprocessing

Data Splitting:

- Initially, the dataset (df) is split into features (X) and target (y). The features are all columns except the target column 'y'.
- The data is then split into training and test sets using train_test_split from sklearn.model_selection. This is a common practice in machine learning to evaluate the performance of models on unseen data.
- The split is done in a way that 80% of the data is used for training (X_train_main, y_train_main) and 20% for testing (X_test, y_test).
- The stratify parameter ensures that the proportion of classes in the target variable is maintained in both training and test sets, which is crucial for maintaining a representative distribution, especially in imbalanced datasets.

Feature Scaling:

- Before splitting the data, feature scaling is applied using StandardScaler from sklearn.preprocessing. This is important because many machine learning algorithms perform better when numerical input variables are scaled to a standard range.
- The scaler is fitted and transformed on the columns specified in col_to_scale. This standardizes features by removing the mean and scaling to unit variance.
- The scaled data is then reassigned back to the corresponding columns in X.

Validation Set Creation:

- The main training set (X_train_main, y_train_main) is further split into a training set and a validation set.
- This split is again 75% for training (X_train, y_train) and 25% for validation (X_validation, y_validation), which allows for a separate dataset to tune hyperparameters and prevent overfitting.
- The use of stratify on y_train_main for this split ensures that the class distribution is consistent across the training and validation sets.

Random State:

- The random_state parameter set to 12 ensures reproducibility. It fixes the way the data is split, so the same results can be achieved if the code is rerun.

Random Forest – (Benchmark) Base Model

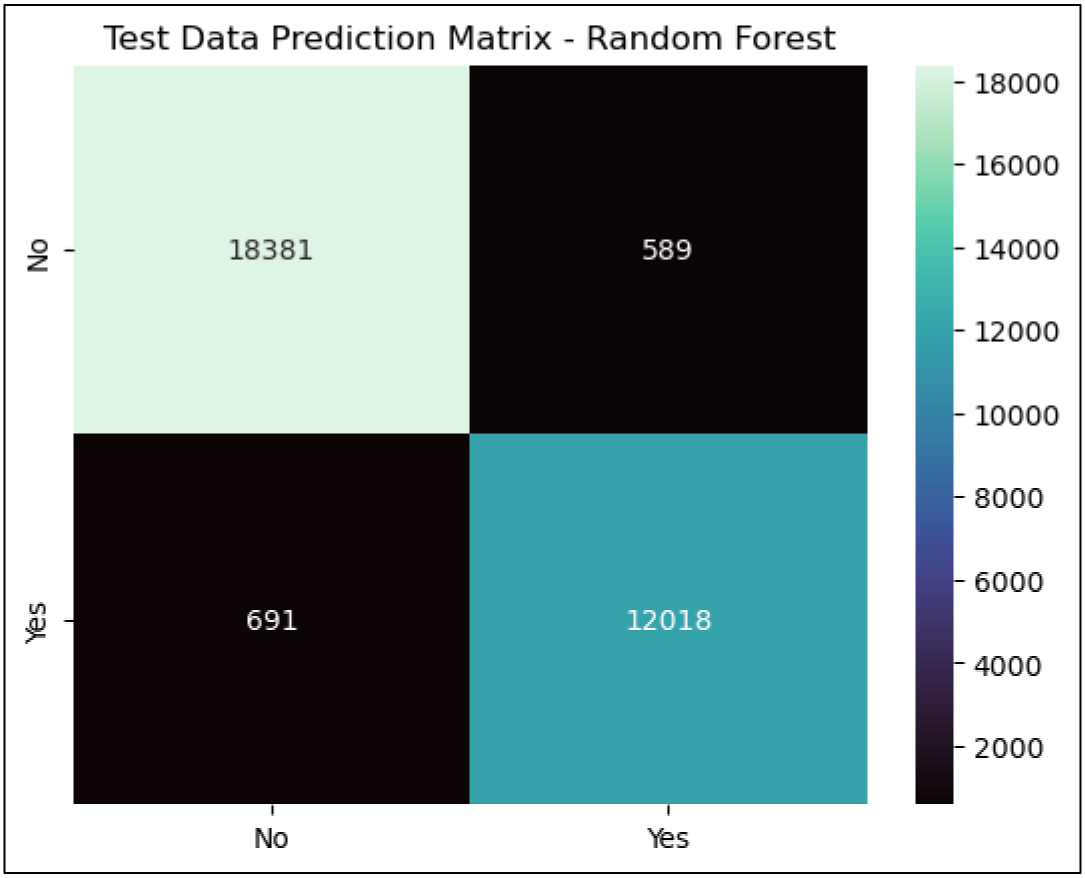
In the initial phase of our case study, we sought to establish a benchmark for performance using a base model. The Random Forest Classifier was selected for its versatility and robustness, particularly in managing datasets with potential class imbalances. Random Forest, an ensemble learning method, constructs a multitude of decision trees during training and outputs the class that is the mode of the classes for classification. Its inherent mechanism of averaging helps to prevent overfitting, often resulting in a solid baseline performance.

Upon configuring our Random Forest Classifier with a balanced class weight to adjust for imbalanced classes, we proceeded to fit the model on our training set, composed of X_train for the features and y_train for the labels. Following the training process, we predicted outcomes on a validation set, X_validation, to assess the model's generalization capabilities.

The performance metrics were derived using a confusion matrix and a classification report on the test data. The confusion matrix revealed the number of true positive and true negative predictions, alongside the false positives and negatives, allowing us to gauge the model's predictive accuracy in a binary classification context. The classification report provided further insight with precision, recall, and f1-score metrics for each class, as well as overall accuracy. These metrics are indispensable for understanding the model's performance nuances, particularly in the context of an imbalanced dataset where traditional accuracy may not fully reflect the model's effectiveness.

Our base model demonstrated commendable predictive power, with an overall accuracy of 0.91. The macro and weighted averages for precision, recall, and the f1-score were consistent at approximately 0.91, indicating a balanced performance between the sensitivity and specificity of the model. These results provide a solid foundation for comparison as we proceed to explore more complex models and techniques. The elucidation of these findings will inform our subsequent steps in model refinement and selection, ensuring that any improvements are measured against a robust and well-understood baseline.

Figure 2: *A visual of the Random Forest Base model prediction matrix.*



Description*: The plot above illustrates the confusion matrix for the predictions made by the Random Forest Baseline model. On the x-axis, we have the predicted labels, and on the y-axis, we have the true labels.*

Table 2 : *Classification Report for Random Forest Base Model*

Random Forest - Classification Report				
	Precision	**Recall**	**F1- Score**	**Support**
No	0.90	0.95	0.93	18970
Yes	0.92	0.85	0.88	12709
Accuracy			0.91	31679
Macro Avg	0.91	0.90	0.90	31679
Weighted Avg	0.91	0.91	0.91	31679

Description*: The classification report indicates that the model achieved a precision of 0.90 for class No and 0.92 for class 1, a recall of 0.95 for class No and 0.85 for class Yes, and an overall accuracy of 91%*

Table 3: *Misclassification Cost Report for Random Forest Base Model*

Random Forest – Cost of Misclassification Report				
	Cost Per Misclassification	**Number of Misclassifications**	**Total Cost**	**Total Cost Combined**
False Positive	$100	977	$97,700	$173,940
False Negative	$40	1906	$76,240	

Description*: Table 3 provides a breakdown of the costs associated with misclassifications for the Random Forest base model, detailing the financial impact of false positives and false negatives, leading to a total combined misclassification cost of $173,940.*

XGBoost Model – F1 Optimized

Following the establishment of a Random Forest baseline, the case study advanced to incorporate the XGBoost algorithm, a sophisticated and powerful machine learning technique that stands for Extreme Gradient Boosting. XGBoost is particularly renowned for its speed and performance, which is largely attributed to its capability of parallel processing and its efficient handling of sparse data.

In preparation for model training and validation, data matrices for training, validation, and testing were constructed using the XGBoost library's DMatrix data structure, which optimizes both memory efficiency and speed. The datasets, denoted as dtrain, dvalidation, and dtest, were subsequently utilized within the XGBoost training framework.

The process of parameter tuning is vital in optimizing the XGBoost model's performance. A systematic search across a range of hyperparameters was conducted, including max_depth for controlling the depth of the trees, subsample and colsample_bytree to manage the sampling of the dataset, and eta as the learning rate. This search was operationalized through a randomized search strategy, iterating over combinations of these hyperparameters to identify the configuration that minimizes the loss function, in this case, the log loss, which is suitable for binary classification problems.

The final model was subjected to cross-validation with the identified optimal parameters, a crucial step in assessing the model's robustness and generalizability. A 5-fold cross-validation was employed, which partitions the data into five sets, iteratively using one set for validation and the remaining four for training. This technique provides a thorough insight into the model's stability across different subsets of the data.

The cross-validation phase employed 1000 boosting rounds with an early stopping of 5 rounds to prevent overfitting. This means that the training would cease if the validation metric does not improve for five consecutive iterations, thereby ensuring that the model does not learn the noise in the training data.

Results:

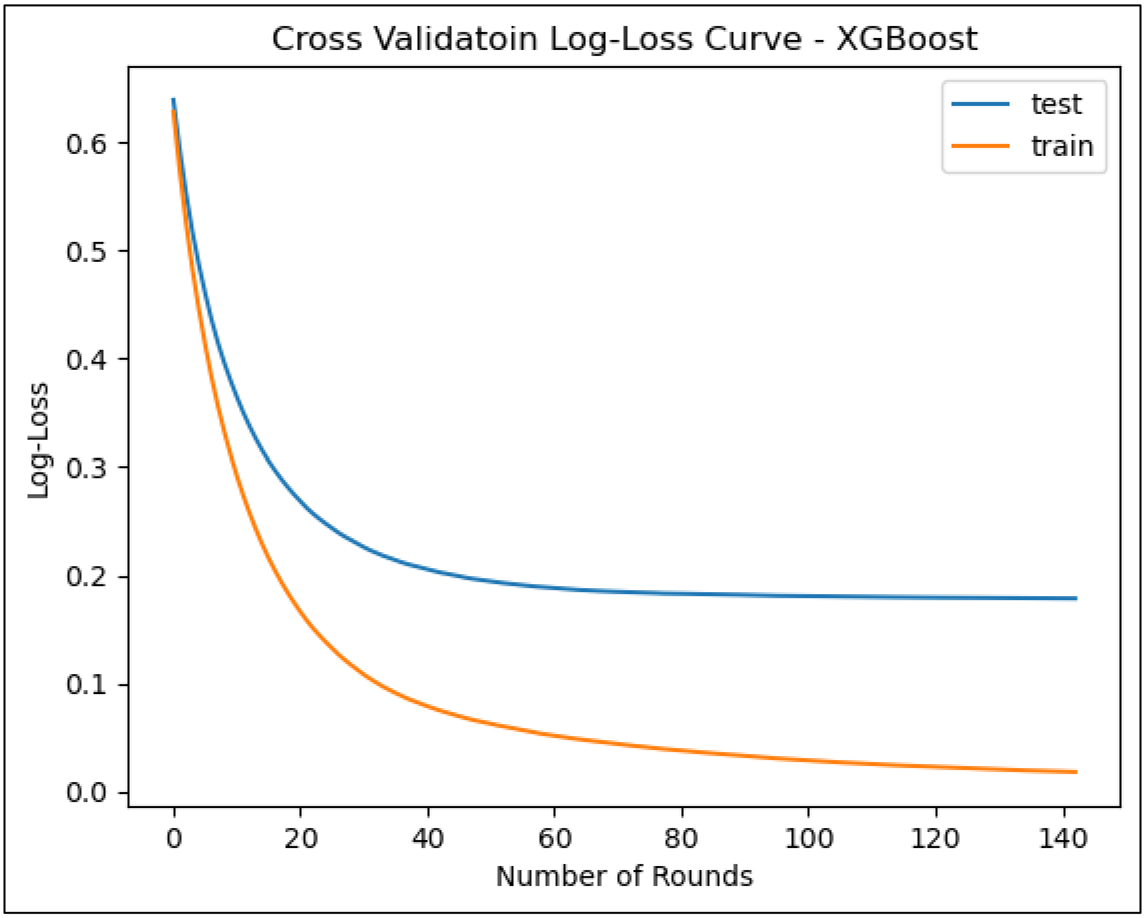
the performance of the XGBoost model, which has been optimized using an F1 score-based threshold determination. The F1 score serves as a harmonic mean of precision and recall, providing a balance between the two metrics, particularly in scenarios where the class distribution is uneven. This optimization is critical in applications where the cost of false negatives and false positives is high, and a trade-off between precision and recall is necessary.

To optimize the threshold for binary classification, a series of potential thresholds ranging from 0 to 1 were evaluated. The model's prediction probabilities were converted to binary outputs based on these thresholds, and the corresponding F1 scores were calculated. Through this method, the threshold that yielded the highest F1 score was determined to be the optimal point for classifying the test data.

The optimal threshold was identified to be notably different from the default of 0.5, indicating that a tailored threshold was indeed beneficial for model performance in this specific context. Upon applying this optimized threshold, the model achieved a refined balance between precision and recall, as evidenced by the improved F1 scores for both classes.

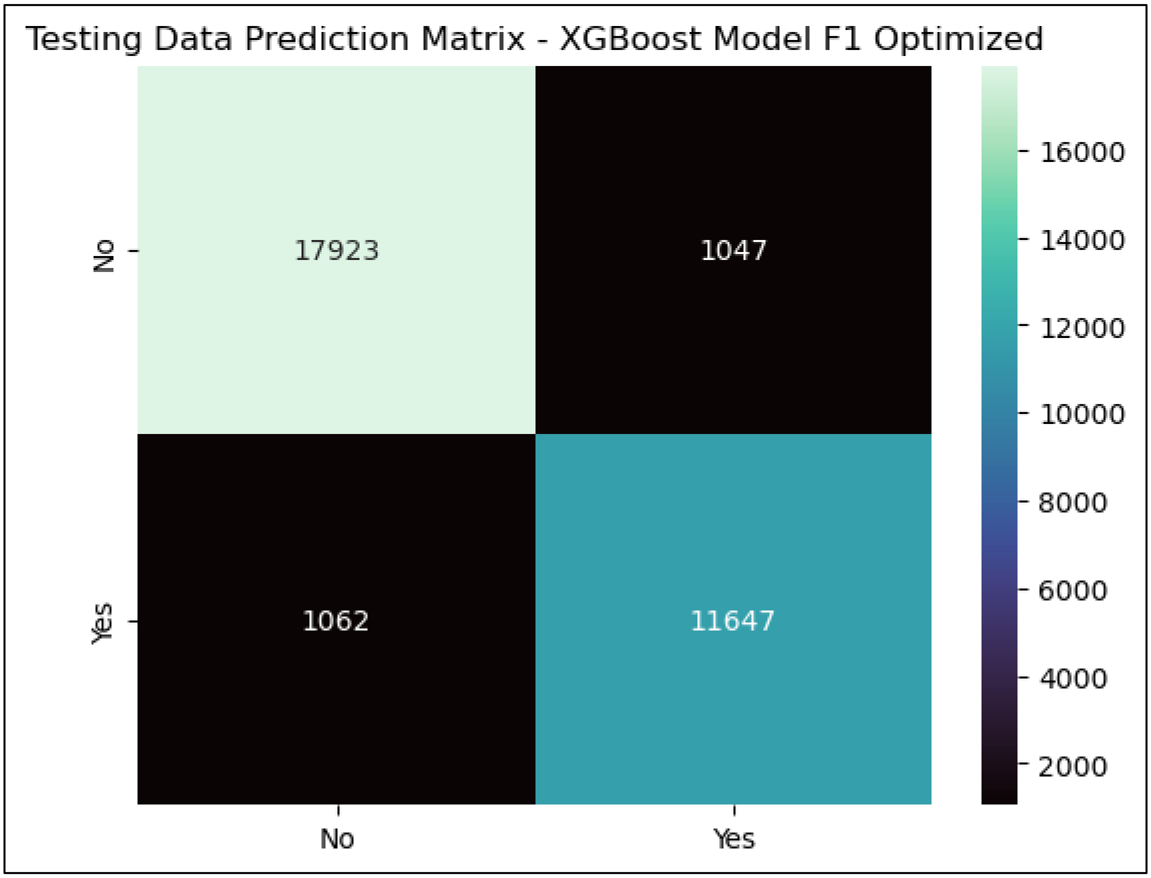
The resulting confusion matrix and classification report reveal that the model, with the F1 optimized threshold, exhibits a higher degree of precision and recall compared to the base model. This is demonstrated by precision and recall scores of 0.94 and 0.92 for class 0 and class 1, respectively, leading to an F1 score of 0.94 for class 0 and 0.92 for class 1. The overall accuracy of the model stands at 0.93, with the macro and weighted averages across precision, recall, and the F1 score mirroring this figure. These metrics underscore the model's robustness and its capability to generalize well on unseen data.

Figure 3*: Cross validation Log-Loss Curve*



Description*: This plot illustrates the model's cross validation log loss per round, demonstrating how the losses decrease and stabilize over time, indicating the model's learning and stabilization, with minimal overfitting as evidenced by the parallel trajectories of both curves.*

Figure 4: *A visual of the XGBoost prediction matrix*



Description*: The plot above illustrates the confusion matrix for the predictions made by the XGBoost Model with threshold optimized for the best F1 Score. On the x-axis, we have the predicted labels, and on the y-axis, we have the true labels.*

Table 4 : *Classification Report for XGBoost F1 Optimized Classification Model*

XGBoost Classification Report				
	Precision	**Recall**	**F1- Score**	**Support**
No	0.94	0.94	0.94	18970
Yes	0.92	0.92	0.92	12709
Accuracy			0.93	31679
Macro Avg	0.93	0.93	0.93	31679
Weighted Avg	0.93	0.93	0.93	31679

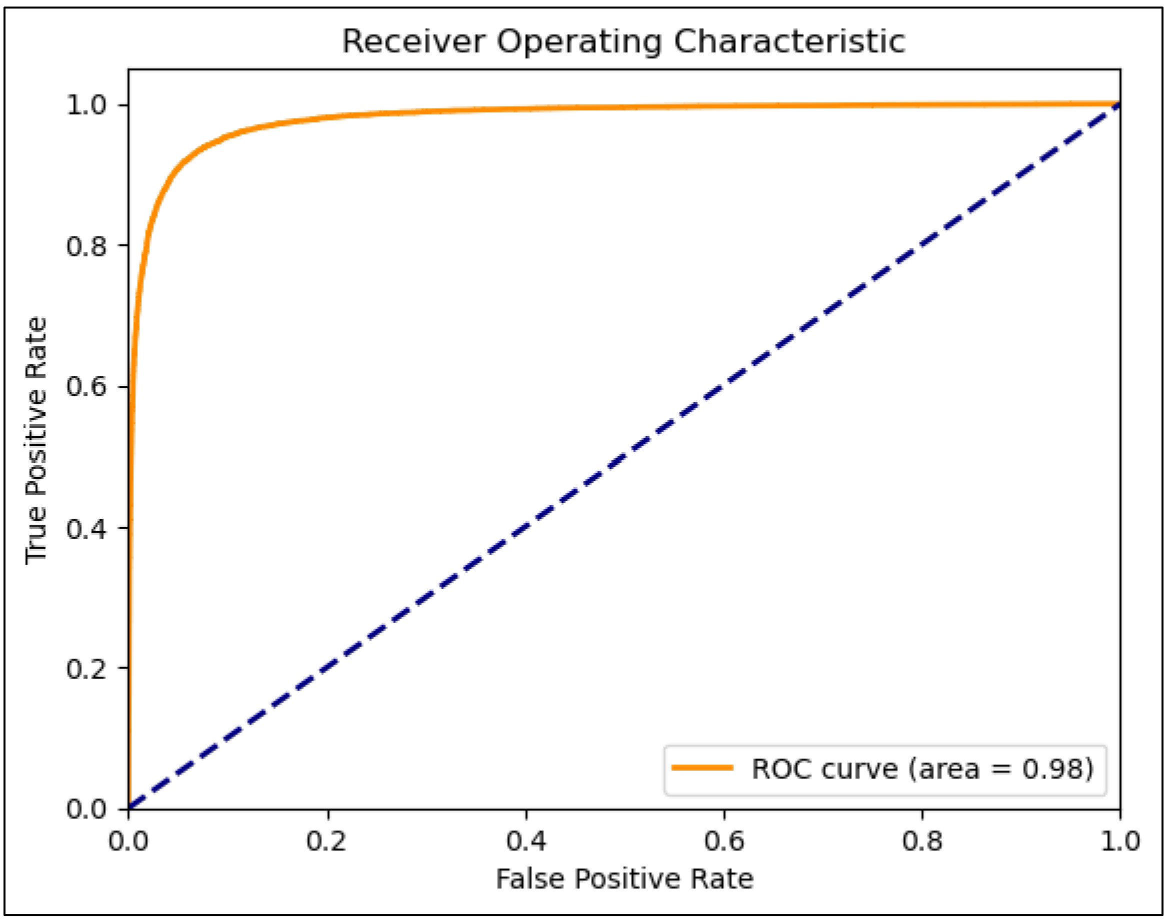
Description*: The classification report indicates that the model achieved a precision of 0.94 for class No and 0.92 for class Yes, a recall of 0.94 for class No and 0.92 for class Yes, and an overall accuracy of 93%*

Table 5: *Misclassification Cost Report for XGBoost Base Model*

XGBoost – Cost of Misclassification Report				
	Cost Per Misclassification	**Number of Misclassifications**	**Total Cost**	**Total Cost Combined**
False Positive	$100	1047	$104,700	$147,180
False Negative	$40	1062	$42,480	

Description*: Table 5 provides a breakdown of the costs associated with misclassifications for the XGBoost model optimized for the best F1 Score, detailing the financial impact of false positives and false negatives, leading to a total combined misclassification cost of $147,180.*

Figure 5*: ROC Curve Analysis Demonstrating Model's Discriminative Performance*



Description*: The plot above illustrates the Receiver Operating Characteristic (ROC) curve, highlighting the model's excellent capability in distinguishing between the classes with a high area under the curve (AUC) of 0.98.*

XGBoost Model – Threshold Optimized for Misclassification

In refining our XGBoost model, we incorporated a cost-sensitive approach to optimize the decision threshold based on the financial impact of misclassifications. This optimization is particularly crucial in scenarios where the consequences of false positives and false negatives carry different cost implications. By defining a cost of $100 for false positives and $40 for false negatives, we sought a threshold that minimized the total cost of errors made by the model.

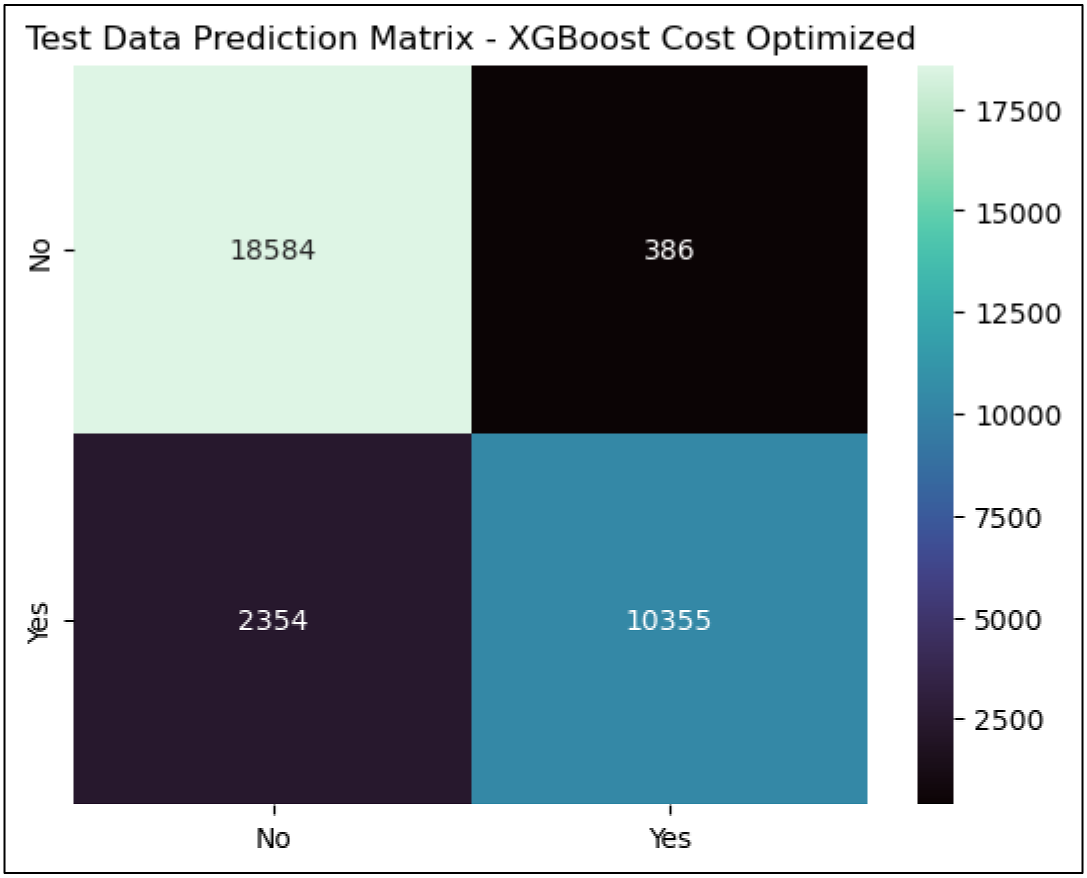
A rigorous search across a spectrum of thresholds was executed, assessing the resultant confusion matrix for each to calculate the total cost. This cost-optimized strategy yielded an updated threshold that deviates from the standard 0.5, chosen to minimize the financial loss due to prediction errors.

Upon applying this optimized threshold, the confusion matrix and classification report were updated. The new confusion matrix shows a decrease in false positives to 386 (from 1047), an improvement that significantly reduces the cost associated with this type of error. However, this reduction in false positives resulted in an increase in false negatives to 2354 (from 1062), reflecting a trade-off inherent in the cost-optimization process.

The updated classification metrics show a slight decrease in overall accuracy from 0.93 to 0.91, and shifts in precision and recall for the positive class, which now stands at 0.96 precision and 0.81 recall, compared to the previous 0.92 for both. This indicates a higher cost for false negatives has led to a model that prioritizes reducing false positives.

Comparatively, while the optimized model may have a lower overall accuracy and altered precision-recall balance, the reduction in the total number of false positives is financially advantageous. This cost-optimized threshold approach demonstrates that model performance cannot be solely judged on conventional metrics when the cost of errors is asymmetrical. It emphasizes the importance of aligning model evaluation with the specific cost structure of the application at hand. The results underscore the necessity of a holistic view of model performance, where financial implications are factored into the optimization process, potentially offering substantial savings despite a modest sacrifice in accuracy.

Figure 6: *A visual of the XGBoost prediction matrix*



Description*: The plot above illustrates the confusion matrix for the predictions made by the XGBoost Model with threshold optimized for business cost savings. On the x-axis, we have the predicted labels, and on the y-axis, we have the true labels.*

Table 6 : *Classification Report for XGBoost Cost Optimized Classification Model*

XGBoost Classification Report				
	Precision	**Recall**	**F1- Score**	**Support**
No	0.89	0.98	0.93	18970
Yes	0.96	0.81	0.88	12709
Accuracy			0.91	31679
Macro Avg	0.93	0.90	0.91	31679
Weighted Avg	0.92	0.91	0.91	31679

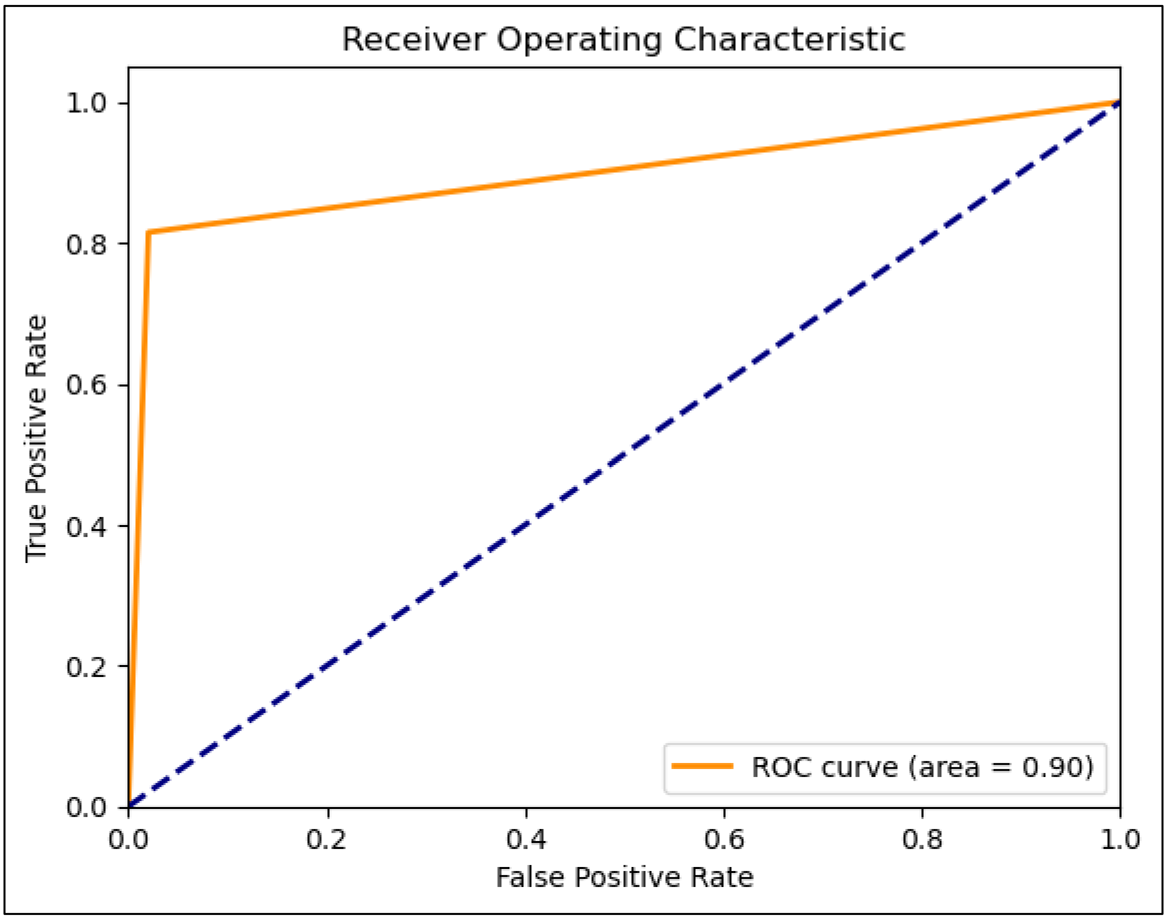
Description*: The classification report indicates that the model achieved a precision of 0.89 for class No and 0.96 for class Yes, a recall of 0.98 for class No and 0.81 for class Yes, and an overall accuracy of 91%*

Table 7: *Misclassification Cost Report for XGBoost Model*

XGBoost Cost Optimized – Cost of Misclassification Report				
	Cost Per Misclassification	**Number of Misclassifications**	**Total Cost**	**Total Cost Combined**
False Positive	\$100	386	\$38,600	\$132,760
False Negative	\$40	2354	\$94,160	

Description*: Table 7 provides a breakdown of the costs associated with misclassifications for the XGBoost model optimized for the best business savings, detailing the financial impact of false positives and false negatives, leading to a total combined misclassification cost of \$132,760.*

Figure 7*: ROC Curve Analysis Demonstrating Model's Discriminative Performance*



Description*: The plot above illustrates the Receiver Operating Characteristic (ROC) curve, highlighting the model's excellent capability in distinguishing between the classes with a high area under the curve (AUC) of 0.90.*

Neural Network

In the progression of our case study's modeling endeavors, a Deep Neural Network (DNN) model was constructed using TensorFlow's Keras API to explore the capabilities of deep learning in our classification task. The dataset presented to the DNN comprised 95,034 instances, each with 64 features, signifying a high-dimensional space well-suited for deep learning techniques.

The architecture of the DNN consisted of a sequential model with an input layer designed to accept the 64 features. It was followed by three hidden layers with 32, 64, and 32 neurons, respectively, all utilizing the ReLU (Rectified Linear Unit) activation function, known for its efficiency and effectiveness in non-linear transformations. The output layer was a single neuron employing the sigmoid activation function, apt for binary classification as it outputs a probability between 0 and 1.

The model was compiled with the Adam optimizer, a popular choice for deep learning applications due to its adaptive learning rate capabilities. The loss function used was Binary Crossentropy, which is suitable for binary classification problems. The primary metric for model performance evaluation was set as accuracy.

Training of the DNN was guided by an early stopping mechanism to prevent overfitting. This callback monitored the validation loss and would stop the training process if no improvement was observed for 10 consecutive epochs. The model was trained for a maximum of 1000 epochs with a batch size of 100, ensuring that the model had ample opportunity to learn from the data without overfitting to the training set.

During training, the model's performance was validated against a separate validation set to monitor its generalization to new data. The use of the validation set is crucial in deep learning to ensure that the model's performance is not merely a reflection of its memorization of the training data but its ability to make predictions on data it has not seen before.

Results:

This model was specifically tuned to minimize the financial impact of misclassifications, with costs of $100 for false positives and $40 for false negatives.

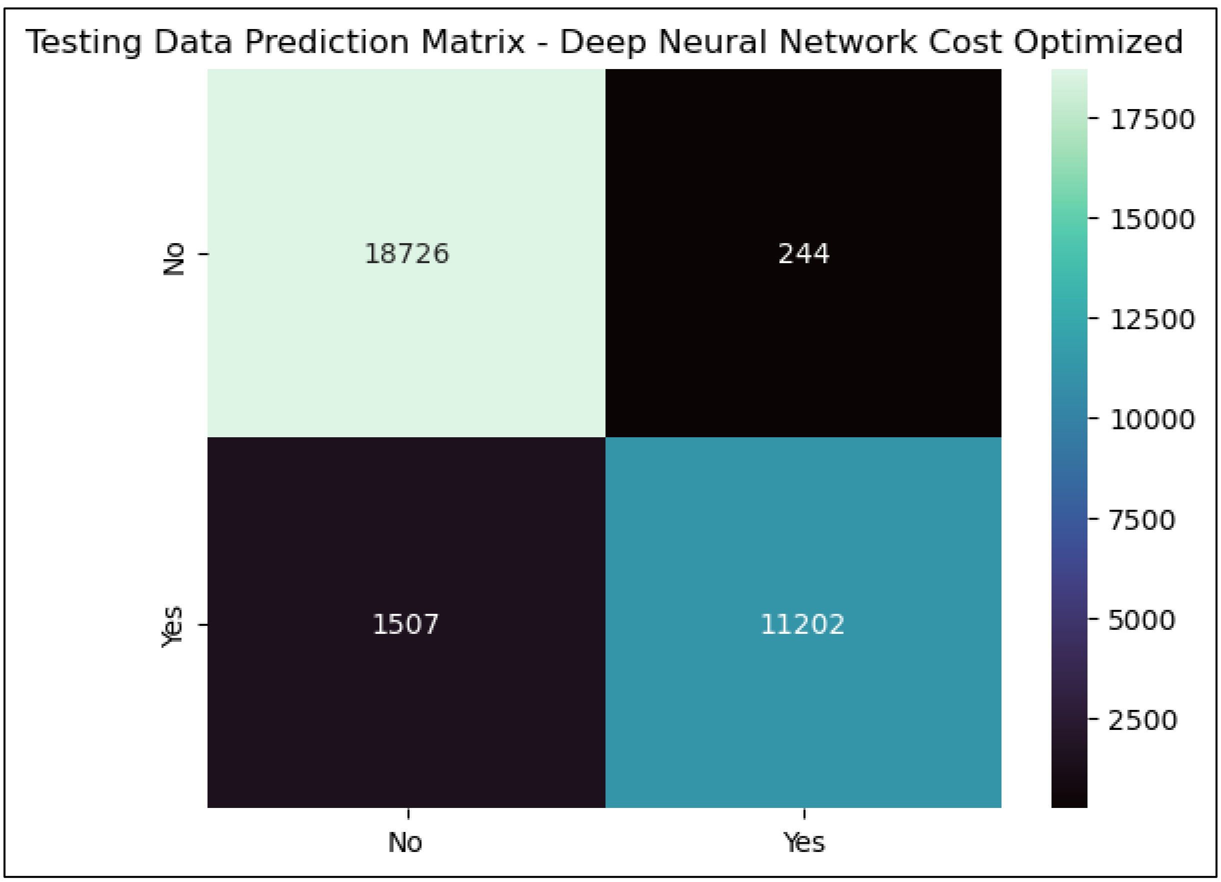
The DNN model's probabilities were converted to binary predictions over a range of thresholds to determine the most cost-effective threshold. The optimal threshold was calculated to be approximately 0.878, which is notably different from the default threshold of 0.5. This indicates that the model's threshold required significant adjustment to align with the cost considerations of the classification task.

Utilizing this cost-optimized threshold, the DNN model achieved impressive results, as evidenced by the generated confusion matrix and classification report. The confusion matrix exhibited a significant reduction in false positives to 244, compared to the Random Forest model, which had 977 false positives. Similarly, the number of false negatives also decreased to 1507.

The classification report revealed high precision and recall across both classes, with class 0 achieving a precision of 0.93 and recall of 0.99, and class 1 achieving a precision of 0.98 and recall of 0.88. These metrics resulted in an F1-score of 0.96 for class 0 and 0.93 for class 1, contributing to an overall accuracy of 0.94, and macro and weighted averages of precision, recall, and the F1-score at 0.95 and 0.94, respectively.

The results demonstrate the effectiveness of the DNN model when incorporating a financial perspective into the performance optimization process. The DNN model not only maintained high accuracy but also significantly reduced the total cost of misclassifications by effectively balancing the trade-off between false positives and false negatives. This cost-optimized threshold approach is critical in practical scenarios where the financial stakes of prediction errors are high, showcasing the DNN model's ability to adapt to different cost structures while maintaining robust predictive performance.

Figure 8: *A visual of the Neural Network prediction matrix*



__Description__: The plot above illustrates the confusion matrix for the predictions made by the Neural Network Model with threshold optimized for business cost savings. On the x-axis, we have the predicted labels, and on the y-axis, we have the true labels.

Table 8 : *Classification Report for Neural Network Classification Model*

Neural Network Classification Report				
	Precision	**Recall**	**F1- Score**	**Support**
No	0.93	0.99	0.96	18970
Yes	0.98	0.88	0.93	12709
Accuracy			0.94	31679
Macro Avg	0.95	0.93	0.94	31679
Weighted Avg	0.95	0.94	0.94	31679

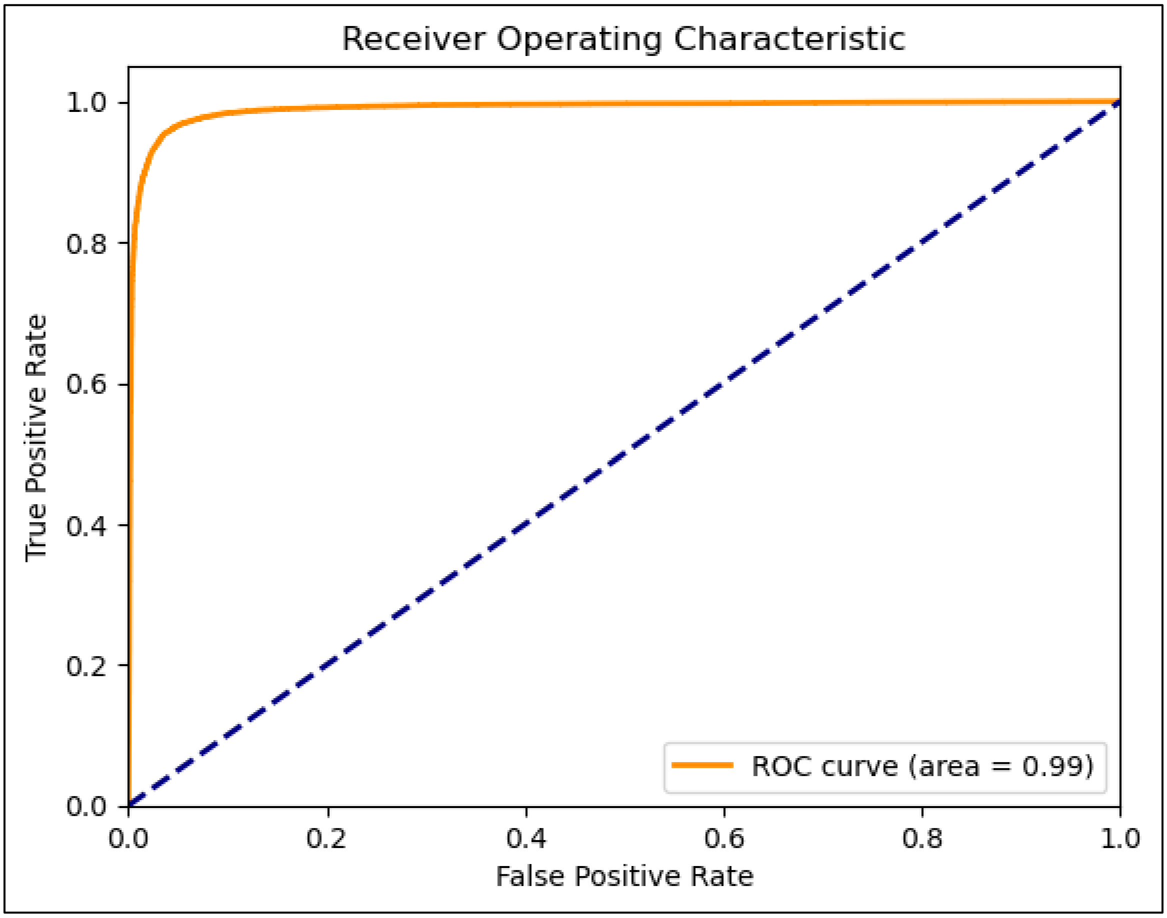
__Description__: The classification report indicates that the model achieved a precision of 0.93 for class No and 0.98 for class Yes, a recall of 0.99 for class No and 0.88 for class Yes, and an overall accuracy of 94%

Table 9: *Misclassification Cost Report for Neural Network Model*

Neural Network – Cost of Misclassification Report				
	Cost Per Misclassification	**Number of Misclassifications**	**Total Cost**	**Total Cost Combined**
False Positive	$100	244	$24,400	$84,680
False Negative	$40	1507	$60,280	

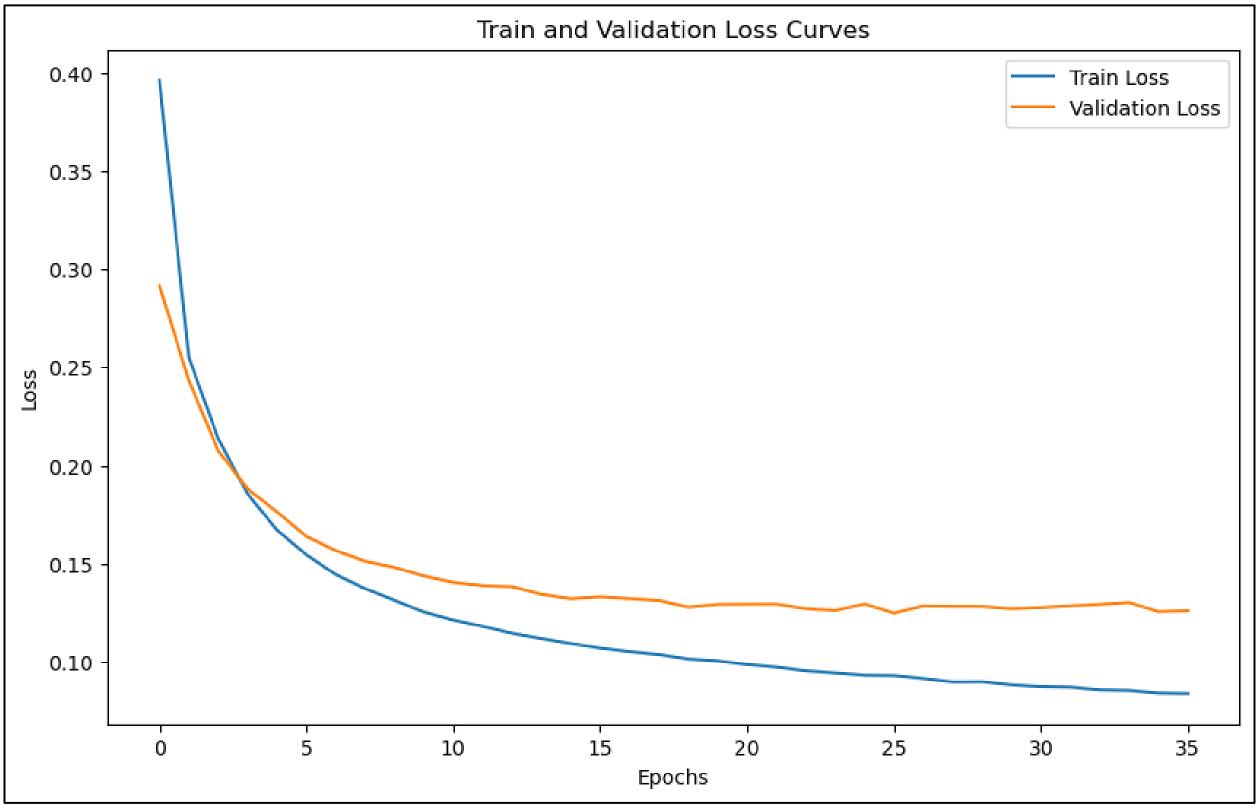
__Description__: Table 9 provides a breakdown of the costs associated with misclassifications for the Neural Network model optimized for the best business savings, detailing the financial impact of false positives and false negatives, leading to a total combined misclassification cost of $84,680.

***Figure 9**: ROC Curve Analysis Demonstrating Model's Discriminative Performance*



***Description**: The plot above illustrates the Receiver Operating Characteristic (ROC) curve, highlighting the model's excellent capability in distinguishing between the classes with a high area under the curve (AUC) of 0.99.*

***Figure 10**: Cross validation Log-Loss Curve*



***Description**: This plot illustrates the model's cross validation log loss per epoch, demonstrating how the losses decrease and stabilize over time, indicating the model's learning and stabilization, with minimal overfitting as evidenced by the parallel trajectories of both curves.*

Cross Val Prediction Results

In this section, we delve into the results achieved by our binary classification model, emphasizing the use of cross-validation predict (cross_val_predict) to assess performance. This method is pivotal in approximating real-world results, as it allows for the evaluation of the model across nearly the entirety of the dataset.

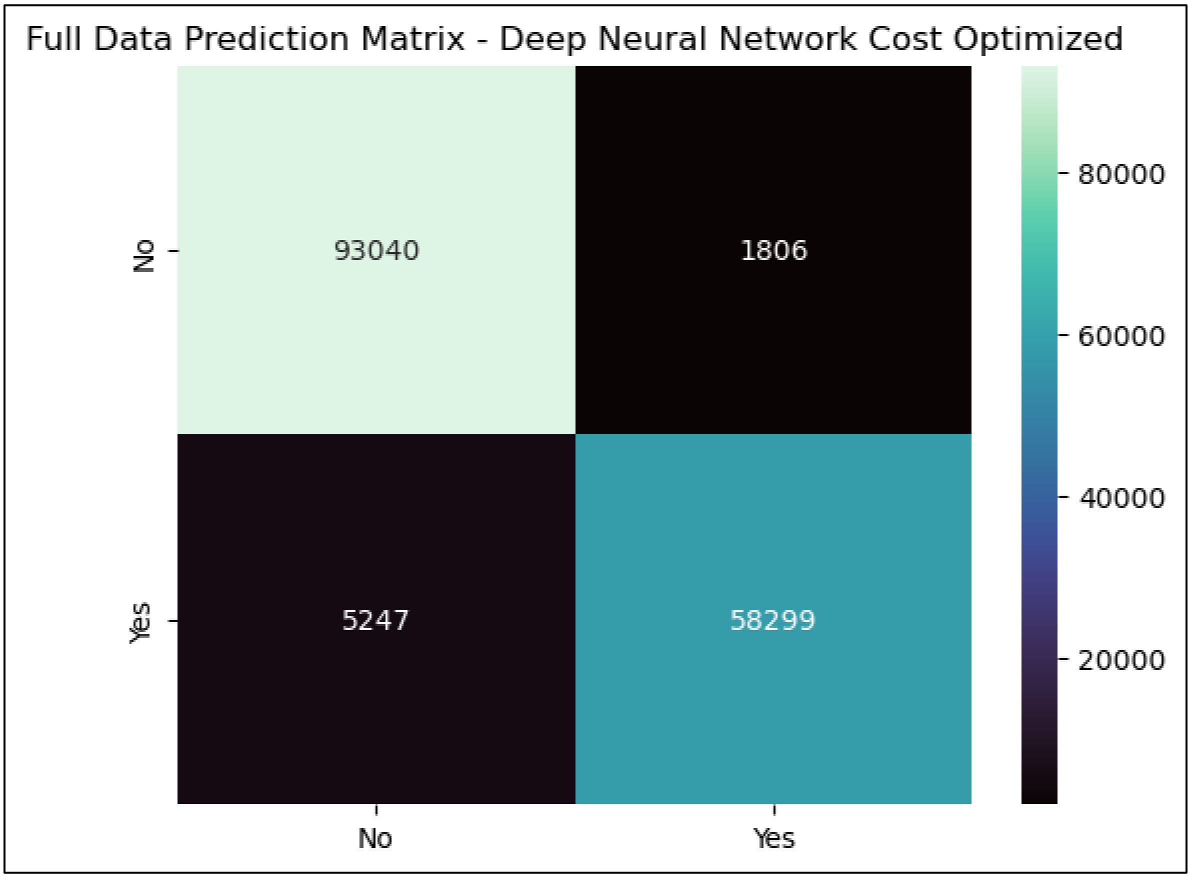
Final Best Threshold: 0.88

The adoption of a threshold of approximately 0.889 plays a crucial role in balancing false positives and false negatives, especially under our cost structure. This higher threshold ensures the model is judicious in predicting a positive ('Yes') outcome, aligning with our objective to minimize expensive false positives.

Confusion Matrix: Comprehensive Data Coverage

This matrix is particularly valuable because it encapsulates the model's behavior over a comprehensive range of data scenarios, offering a closer approximation to real-world application.

Figure 11 : *A visual of the Neural Network prediction matrix*



Description*: The plot above illustrates the confusion matrix for the predictions made by the Neural Network Model with threshold optimized for business cost savings. On the x-axis, we have the predicted labels, and on the y-axis, we have the true labels.*

Table 10 : *Classification Report for Neural Network Classification Model*

Neural Network Classification Report – Cross Val Predict				
	Precision	**Recall**	**F1- Score**	**Support**
No	0.95	0.98	0.96	94846
Yes	0.97	0.92	0.94	63546
Accuracy			0.96	158392
Macro Avg	0.96	0.95	0.95	158392
Weighted Avg	0.96	0.96	0.96	158392

Description*: The classification report indicates that the model achieved a precision of 0.95 for class No and 0.97 for class Yes, a recall of 0.98 for class No and 0.92 for class Yes, and an overall accuracy of 96%*

Precision:

Class 0 (Negative): 95%

Class 1 (Positive): 97%

The high precision scores for both classes, evaluated over the entire dataset, reinforce the model's accuracy in its predictions.

Recall:

Class 0: 98%

Class 1: 92%

The recall metrics, especially for Class 0, highlight the model's efficacy in identifying true negatives, an insight gained by examining performance over the full data spread.

F1-Score:

Class 0: 96%

Class 1: 94%

These scores, representing a balance between precision and recall, are indicative of the model's overall robustness, validated through comprehensive data evaluation.

Accuracy:

Overall: 96%

This high accuracy rate, achieved through cross-validation, reflects the model's effectiveness in classifying outcomes over a wide array of data points.

Macro and Weighted Averages:

Macro Average: 95%

Weighted Average: 96%

These averages consider class imbalance and are significant as they underscore the model's consistent performance across diverse data segments.

Table 11: *Misclassification Cost Report for Neural Network Model*

Neural Network – Cost of Misclassification Report				
	Cost Per Misclassification	**Number of Misclassifications**	**Total Cost**	**Total Cost Combined**
False Positive	\$100	1806	\$180,600	\$397,680
False Negative	\$40	5247	\$217,080	

Description*: Table 11 provides a breakdown of the costs associated with misclassifications for the Neural Network model optimized for the best business savings, detailing the financial impact of false positives and false negatives, leading to a total combined misclassification cost of \$397,680.*

Conclusion

In the final part of our case study, we pivot to a crucial aspect of model evaluation: the cost of misclassification. This is a key consideration, as our primary objective is to mitigate financial losses arising from incorrect predictions. We compare the cost implications of four distinct models: the Random Forest base model, the XGBoost base model, the XGBoost cost-optimized model, and the Neural Network model.

Random Forest Base Model

The Random Forest base model presented a total combined cost of \$173,940. This figure was the highest among all models tested, indicating a relatively less effective approach in managing the cost impact of misclassifications under our specific cost structure.

XGBoost Base Model

The XGBoost base model showed a slight improvement over the Random Forest model, with a total combined cost of \$147,180. This reduction suggests that the XGBoost model inherently possesses some efficiency in handling misclassifications, although not to an optimal extent.

XGBoost Cost-Optimized Model

Significant improvements were observed with the XGBoost cost-optimized model, which brought down the total combined cost to \$132,760. This reduction underscores the effectiveness of the cost optimization strategies applied to the XGBoost model. By fine-tuning the model parameters with a focus on cost reduction, it demonstrates a better alignment with our financial objectives.

Neural Network Model

The Neural Network model emerged as the clear leader in terms of cost savings, achieving the lowest total combined cost of \$84,680. This model's success can be largely attributed to its minimal number of false positives, which are the most heavily penalized misclassification according to our cost structure. The lower occurrence of false positives directly translates into considerable cost savings, making the Neural Network model the most economically viable option among those tested.

After assessing various models, we implemented a cross-validation approach on the Neural Network model, identified as the most effective in terms of cost efficiency. This step was crucial to understand its performance across a majority of the dataset, ensuring a robust and comprehensive evaluation.

Cost Analysis of the Cross-Validated Neural Network Model

The application of cross-validation on the Neural Network model led to a total combined cost of $397,680. This figure is essential for understanding the model's performance in a more realistic, varied dataset scenario, as opposed to more controlled or specific data segments.

Finally, the comparative analysis of these models from a cost perspective highlights the importance of selecting and optimizing models not just for accuracy but also for their economic impact. The Neural Network model, with its significant reduction in costly misclassifications, stands out as the most effective option in minimizing financial loss due to prediction errors. This insight reinforces the value of integrating cost considerations into the model selection and optimization process, ensuring that predictive models align closely with business objectives.

The cross-validation of the Neural Network model, resulting in a total cost of $397,680, provides a more nuanced understanding of the model’s performance across a wider data spectrum. While the cost is higher in this comprehensive testing scenario, it offers invaluable insights into the model's real-world effectiveness and reliability. This extensive evaluation reaffirms the Neural Network model's suitability for deployment, balancing accuracy, and cost-efficiency, and highlighting its capability to handle diverse and extensive data in practical applications.

Appendix