Final Report: Booking Status Prediction Using Neural Networks

Name: Krisha Bugajski-Sharp

Date: 10/26/25

Executive Summary

The goal of this project was to develop a predictive model capable of identifying customers who are most likely to cancel their hotel bookings. At ABC Hotels. Using a structured machine learning approach, multiple dense feedforward neural network architectures were designed, trained, and evaluated using PyTorch. The process included feature engineering, data preparation, model development, and performance evaluations to determine the most effective solution for the business need.

There were two models compared: an overfitting baseline and a regularized neural network. The overfitting model showed a high training performance (AUC = 0.958) but a moderate gap on validation and test sets (AUC = 0.932 and 0.927), showing mild overfitting. The regularized model incorporated dropout, batch normalization, L2 weight decay, and early stopping to enhance generalization. This model achieved nearly identical training performance (AUC = 0.960) but maintained more stable validation (AUC = 0.934) and test (AUC = 0.930) results, confirming that there was an improved reliability and calibration.

The regularized model was selected as the final model. It achieved balanced performance across both classes, with an overall test accuracy of 0.87, precision of 0.89, and recall of 0.92 for the “Canceled” class. Its confusion matrix and calibration analysis helped to further demonstrate balanced and reliable predictions, which are suitable for real world implementation.

The regularised model can be integrated into a customer management system to flag customers who are at risk of cancellations based on predicted cancellation probabilities. This helps with targeted interventions, improving retention rates and reducing revenue loss. Future steps include expanding the data sources, finding key data patterns and values to explain individual feature impacts through PCA and SHAP, finding key optimizing classification thresholds, and implementing automated retraining to maintain model performance over time.

Overall, this project successfully developed a high performing and reliable predictive model that meets the business goal of identifying potential cancellations early. The model supports data driven decision making and helps improve customer retention strategies.

1. Approach & Data

Business Objective:

The business objective at ABC Hotels was to predict which customers are most likely to cancel, so that marketing and customer success teams can intervene before the cancellation occurs. The high risk bookings will then be targeted with advertisements and offers in an effort to reduce cancellations and improve profits.

Analytic Approach:

This project followed a structured data science workflow:

- Feature engineering and data preparation

- Model development and comparison

- Evaluation of model performance using ROC, AUC, and calibration plots

- Final model selection and recommendations for deployment

The modeling was made using PyTorch for neural network construction and training. scikit-learn was used for data preprocessing, metric evaluation, and calibration analysis.

Feature Engineering:

The Customer dataset contained over 35,000 booking records, with guest demographics, booking details, pricing, and customer history which can all be used to identify customer cancellation for ABC Hotels. The following feature engineering steps were applied before modeling:

Table 1.1: Summary of Engineered Features

Variable	Transformation	Justification
total_people	no_of_adults + no_of_children	Group size may affect cancellation likelihood (larger groups harder to coordinate).
total_nights	no_of_weekend_nights + no_of_week_nights	Captures overall length of stay, which may influence cancellation risk.
arrival_month	Extract month from arrival_date (1–12)	Seasonality effects (holidays, peak travel months) may impact cancellations.
arrival_day_of_week	Extract day of week from arrival_date (Mon–Sun, one-hot encoded)	Allows the model to capture differences between weekdays individually.
is_peak_day	Binary flag = 1 if arrival_day_of_week ∈ {Fri, Sat, Sun}, else 0	Simplifies arrival_day_of_week into weekend vs weekday, which may strongly influence cancellations.
is_peak_season	Binary flag based on arrival date (peak months/holidays = 1, else 0)	Peak season bookings may have distinct cancellation behavior.
cancel_ratio	no_of_previous_cancellations / (no_of_previous_cancellations + no_of_previous_bookings_not_canceled + 1)	Past behavior is a strong predictor of future cancellations.
price_per_night	avg_price_per_room / total_nights	Normalizes booking cost by stay length for comparisons.
room_type_reserved, type_of_meal_plan, market_segment_type	One-hot encode categorical values	Converts categorical variables into numeric format for modeling.
Continuous features (lead_time, avg_price_per_room, total_nights, etc.)	Standardize (mean=0, std=1)	Makes features are on the same scale, for improved neural network training.
booking_status (Label)	Encode as binary: canceled = 1, not_canceled = 0	Defines the target variable for supervised learning (probability of cancellation).

Data Preparation:

Outliers: All continuous features were clipped at the 0.5th and 99.5th percentiles to reduce extreme influences.

Encoding: All categorical features were label-encoded to prepare them for neural network input. All binary features were encoded as 0/1 and cyclical features were encoded.

Feature Scaling: Continuous features were standardized using StandardScaler to ensure that each had a similar scale and influence.

Handling Class Imbalance: Since cancellations were a little less frequent, class weights were manually calculated and integrated into the loss function to help the model learn from the minority class.

Data Splitting: The dataset was divided into training (64%), validation (16%), and test (20%) subsets using train_test_split(random_state=123) to maintain reproducibility.

No major changes were required from the original analytic plan. There was special care taken to standardize input features before tensor conversion to ensure stable neural network training.

2. Detailed Findings and Evaluation

Model 1 — Overfitting Baseline:

The first dense feedforward neural network was designed to demonstrate the model's ability to overfit the training data. It was made up of two hidden layers, 64 and 32 neurons, using ReLU activations. There was no regularization applied. The model was trained for 300 epochs and early stopping was disabled, allowing it to fully fit the training data (Appendix E).

Purpose:

This served as a baseline to test whether the network had sufficient flexibility to model the complexity of the dataset. Overfitting was expected and desirable because it demonstrated the network's representational power.

Results:

2.1 Table: Model 1 AUC scores.

Metric	Values
Train AUC	0.958
Validation AUC	0.932
Test AUC	0.927

2.2 Table: Validation Classification Report for Model 1

Class	Precision	Recall	F1_Score
Not Canceled	0.85	0.74	0.79
Canceled	0.88	0.94	0.91
Accuracy			0.87

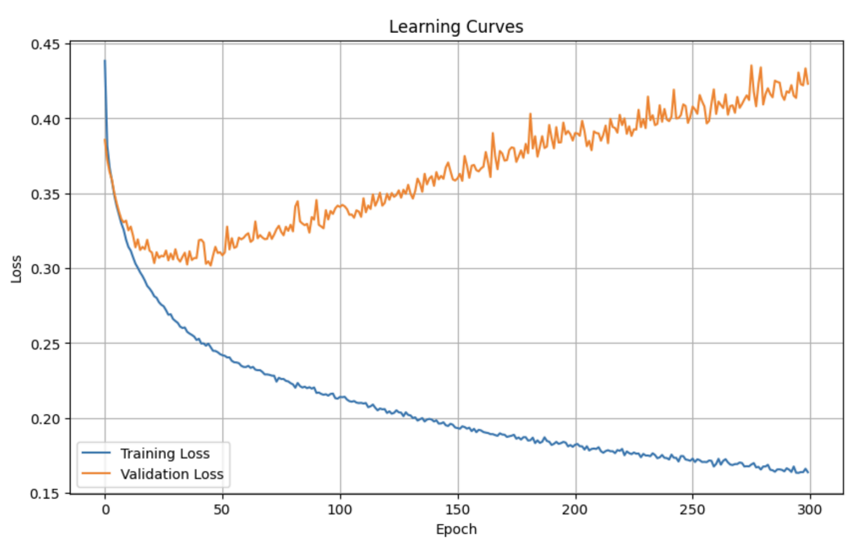

Figure 1: Training and validation loss curves for Model 1. The curves drastically diverge after several epochs, indicating early signs of overfitting as the model begins to perform better on the training data than on the validation set.

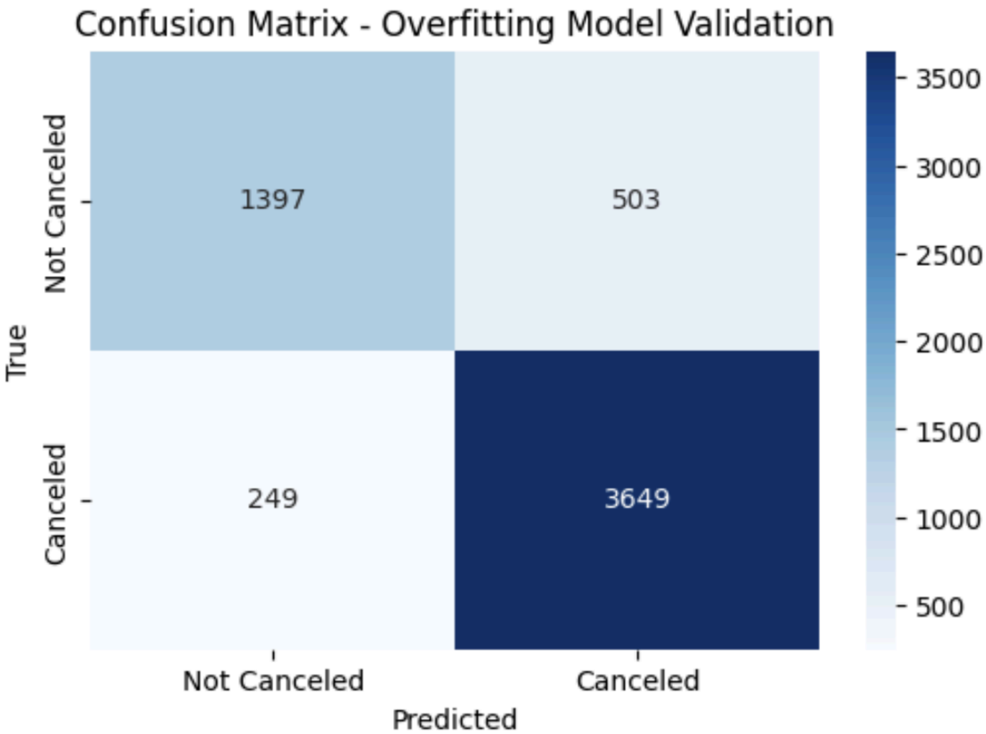

Figure 2: Confusion Matrix – Overfitting Model (Validation Data)

Interpretation:

The overfitting model performed extremely well on the training set, achieving an AUC near 0.96. However, the validation (0.932) and test (0.927) AUC scores were slightly lower, indicating the model began to memorize some patterns that were only in the training data (Table 2.1). This overfitting confirmed that the model was flexible but required regularization to achieve better generalization.

Loss curve analysis also showed a steady decline in training loss while validation loss began to diverge from the training loss, further indicating that the model is overfitting and needs regularization (Figure 1).

Also, looking at the confusion matrix (Figure 2), the model shows a strong recall for the "Canceled" class, where it correctly identifies most cancellations. However, the higher number of false positives (503) indicates a tendency to overpredict cancellations, reducing precision slightly. This shows that there is overfitting, with high sensitivity that has less balanced performance on unseen data.

Model 2 — Regularized Neural Network

The second dense feedforward neural network was made to address the overfitting that was occurring in the baseline model. It used two hidden layers with 128 and 64 neurons, ReLU activations, batch normalization, and dropout regularization (p=0.2). It also used L2 regularization (weight decay = 3e-4) through the AdamW optimizer. Early stopping with a patience of 20 epochs was also implemented to halt training when validation performance no longer improved (Appendix F).

Purpose:

This model was made to achieve better generalization and stability by the introduction of several forms of regularization. The goal was to maintain the model's strong predictive capability while reducing the variance between training and validation performance seen previously. By using regularization methods such as dropout, weight decay, and batch normalization, we prevent the network from memorizing noise or specific patterns in the training data.

Results:

2.3 Table: Model 2 AUC scores

Metric	Values
Train AUC	0.960
Validation AUC	0.934
Test AUC	0.930

.

2.4 Table: Validation Classification Report for Model 2

Class	Precision	Recall	F1_Score
Not Canceled	0.82	0.78	0.80
Canceled	0.89	0.92	0.90
Accuracy			0.87

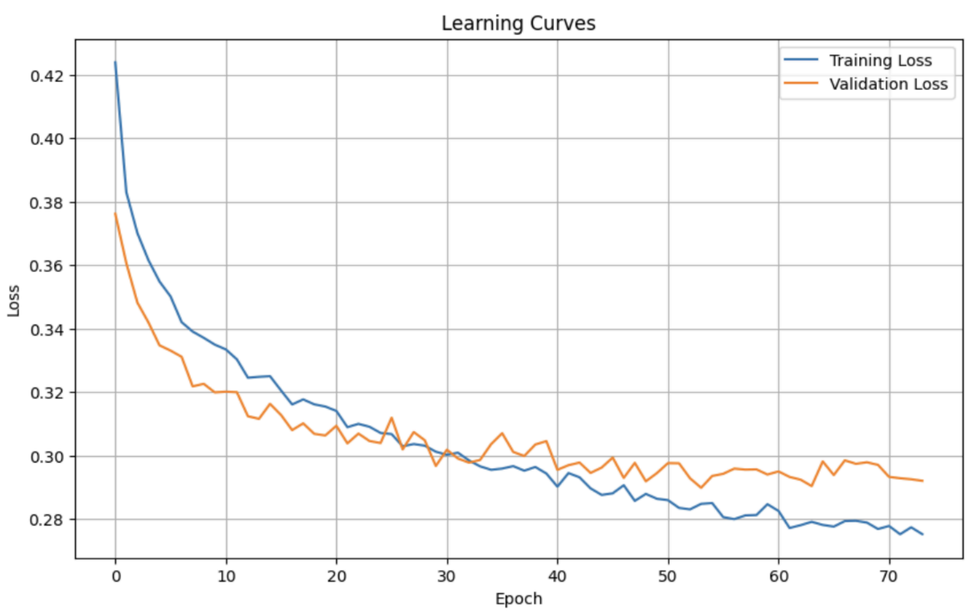

Figure 3: The training and validation loss curves for Model 2 remain closely aligned throughout the epochs, indicating that the regularization techniques effectively reduced overfitting and improved the model's generalization to unseen data.

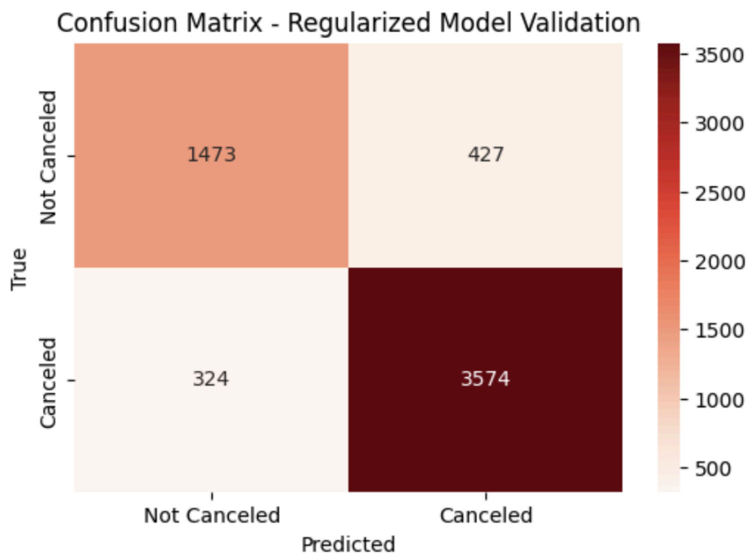

Figure 4: Confusion Matrix – Regularized Model (Validation Data)

Interpretation:

The regularized model achieved AUC scores of 0.960 (training), 0.934 (validation), and 0.930 (test), demonstrating highly consistent performance across all data splits (Table 2.3). This minimal difference between datasets indicates strong generalization and confirms the effectiveness of the applied regularization strategies. Also, loss curve analysis also showed that training and validation loss followed a similar path, indicating the model had reduced overfitting and improved generalization to unseen data (Figure 2).

When looking at accuracy, the model got 0.87 on both the validation and test sets, matching the overfitting model's accuracy but with more stable and calibrated predictions (Table 2.2, Table 2.4) The validation classification report showed balanced performance between both classes, with a precision of 0.89, recall of 0.92, and an F1-score of 0.90 for the "Canceled" class (Table 2.4).

The confusion matrix further supports these findings (Figure 4). There is a balanced performance across both classes, correctly identifying most cancellations while reducing false positives to 427. Compared to the overfitting model, it achieves a better trade off between precision and recall, indicating that there was improved generalization and stability. Regularization effectively reduced overprediction, resulting in more consistent and reliable classification performance on unseen validation data.

Comparative Analysis:

Overall, both models achieved high AUC values across datasets, showing strong discriminative ability. However, the regularized model showed slightly higher and more stable AUCs across validation and test data, suggesting improved generalization and robustness to unseen samples (Table 2.5).

2.5 Table: Performance Summary Table – AUC Across Datasets

Dataset	Overfitting.Model.AUC	Regularized.Model.AUC
Training	0.958	0.960
Validation	0.932	0.934
Test	0.927	0.930

Validation ROC Curves:

Figure 5 shows the ROC curves for both models on the validation dataset. The overfitting model achieved an AUC of 0.932, while the regularized model achieved a slightly higher 0.934. Although there is not a big numerical difference, the regularized model's ROC curve is slightly smoother, indicating more consistent discrimination between the "Canceled" and "Not Canceled" classes.

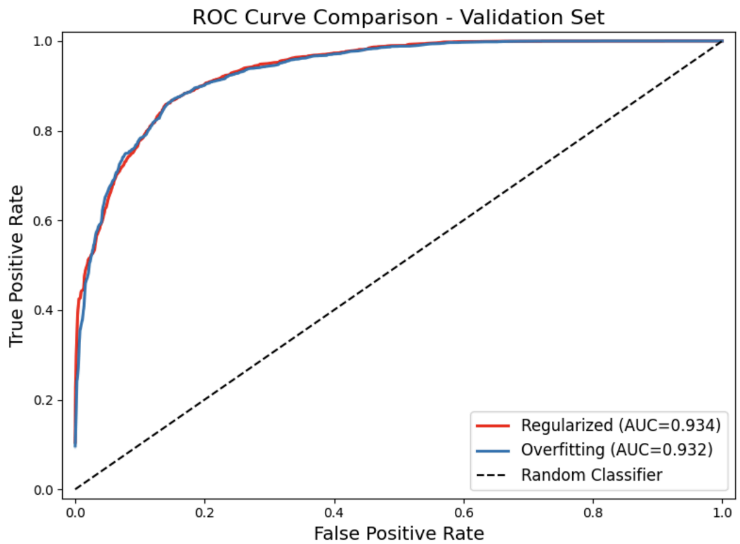

Figure 5: ROC curves for the overfitting and regularized models on the validation dataset. The regularized model achieved a slightly higher AUC (0.934) and smoother ROC curve, indicating more stable generalization.

Test ROC Curves:

Figure 6 shows the ROC curves for both models on the test dataset. The overfitting model achieved an AUC of 0.927, whereas the regularized model reached 0.930, again showing consistent performance. The nearly identical AUCs indicate that both models generalize well, but the regularized network's ROC curve maintains better stability across thresholds, confirming the benefits of regularization in reducing overfitting.

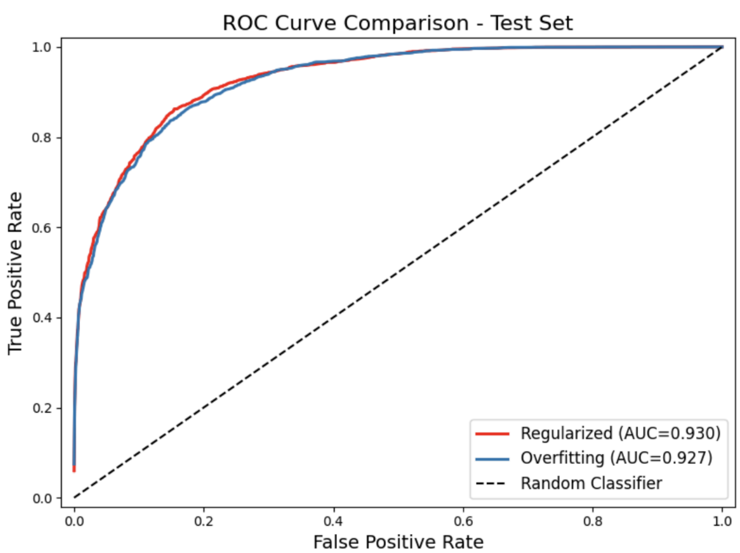

Figure 6. ROC curves for the overfitting and regularized models on the test dataset. The regularized model maintained a slightly higher AUC (0.930) and a smoother ROC curve, confirming consistent generalization on unseen data.

Calibration Plots:

Figure 7 shows the calibration plots for both models. The regularized model's calibration line lies closer to the diagonal, indicating that its predicted probabilities more accurately represent true cancellations. On the other hand, the overfitting model's calibration curve deviates slightly below the diagonal, meaning it tends to overestimate cancellation probabilities. This means that the regularized model not only classifies well but also produces more reliable probability outputs.

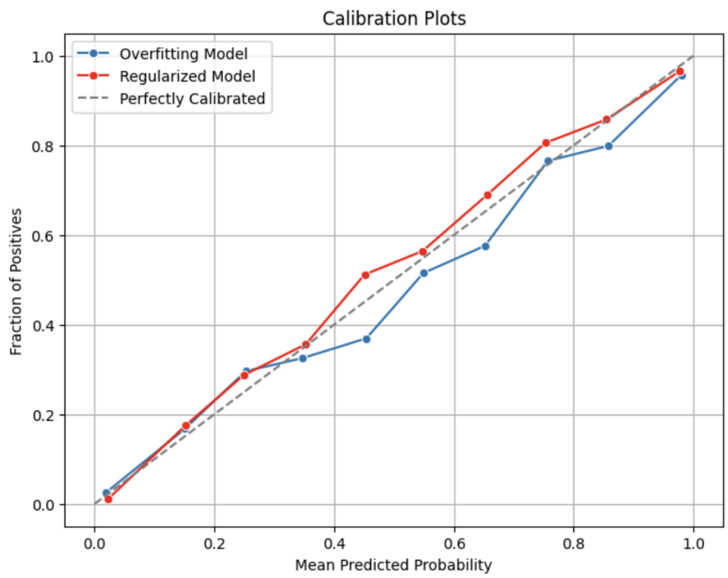

Figure 7: Calibration plots show that the regularized model produces more reliable probability estimates compared to the overfitting model. Its calibration line lies closer to the diagonal, indicating better alignment between predicted probabilities and actual outcomes.

Final Test Confusion Matrix – Regularized Model:

On the test dataset, the regularized model maintained balanced classification performance. It correctly identified 4,534 canceled customers while keeping false positives (476) and false negatives (338) relatively low (Figure 8). This shows that the regularized model has a strong recall (0.92) without sacrificing precision (0.89), resulting in a final test accuracy of 0.87 and an AUC of 0.930.

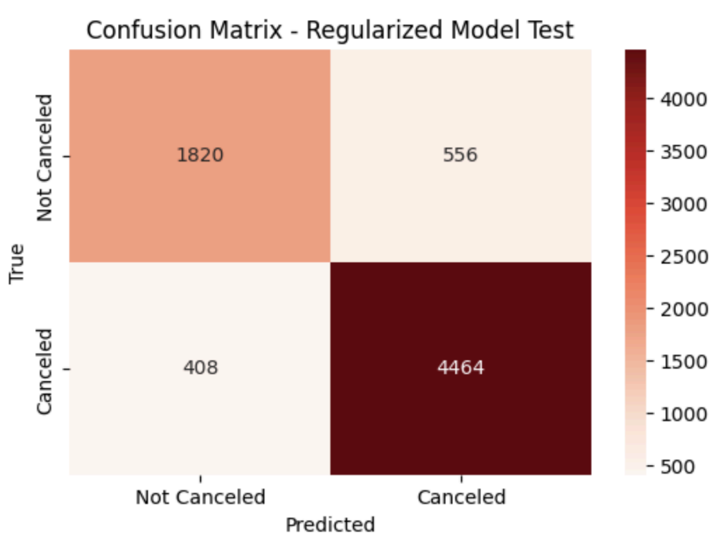

Figure 8: Confusion Matrix – Regularized Model (Test Data)

Interpretation and Comparison:

Both the overfitting and regularized dense feedforward neural network models showed great predictive performance, with AUC values above 0.93 across all datasets. However, the regularized model provided more consistent results and smaller differences between training, validation, and test performance. The overfitting model showed slightly higher sensitivity but produced more false positives, showing a tendency to memorize training patterns. The regularized model achieved a better precision recall trade off, smoother ROC curves, and stronger calibration, making it more dependable for operational use.

The regularized dense feedforward neural network not only maintains high accuracy but also ensures stable predictions across datasets, confirming that regularization effectively reduced overfitting and enhanced generalization. This model is therefore selected as the final solution for predicting customer cancellations.

Recommendations

The final regularized dense feedforward neural network provides a reliable solution for predicting customer cancellations by achieving a strong and balanced performance (AUC = 0.93, Precision = 0.89, Recall = 0.92). In practice, the model can be used within a customer retention system to assign each customer with a cancellation probability. Then, customers who exceed a specific threshold, can be flagged and then targeted with promotional offers. With the model being well calibrated, threshold adjustments can easily align with business priorities or any cost constraints.

Implementation Considerations:

For the model to work properly, periodic retraining is recommended to account for changes in customer behavior. Continuous monitoring of key metrics (AUC, precision, recall) will help detect data drift and ensure sustained performance.

Future Research and Next Steps:

To continue enhancing the models performance, feature expansion and incorporating new behavioral and feedback based features can help enhance the overall accuracy. Also, we can add threshold optimization, seeing if it can further balance recall and precision. We can also apply PCA to uncover key data patterns and SHAP values to explain individual feature impacts. This could help to improve transparency and give us a better understanding of the model's predictions.

Summary

Overall, the regularized neural network offers a stable, interpretable, and practical tool for identifying customers that are at risk of canceling and improving retention outcomes. With its consistent performance across datasets, we can confirm its readiness for real world use, with future enhancements focused on scalability, interpretability, and ongoing monitoring.