Customer Churn Prediction Report

Introduction:

This report presents an analysis and predictive modeling approach for customer churn prediction based on a large dataset. The goal is to develop a model that can predict whether a customer is likely to churn or not. The report covers data preprocessing, feature engineering, and model selection.

Data Preprocessing:

The dataset was loaded from an Excel file and explored to check for any missing values.

Initial data exploration showed that there were no missing values, indicating a clean dataset.

Data outliers were analyzed using box plots, and it was observed that there were no significant outliers in the dataset.

Feature Engineering:

A new feature, "total_bill_amount," was created by multiplying "Monthly_Bill" and "Subscription_Length_Months." This feature aims to capture the total billing amount over the subscription period.

Age groups were created to categorize customers into age brackets ("0-29," "30-39," "40-49," "50-59," and "60+"). This categorical feature, "Age_Group," was used for further analysis.

Data Visualization and Insights:

Visualizations were generated to understand the distribution of features and relationships between features and churn.

Age and gender distribution was explored, and it was observed that gender was evenly distributed.

Age distribution showed some peaks around ages 50 and 62.

The relationship between churn and features like age, total bill amount, and location was visualized to gain insights.

Model Selection and Training:

The dataset was prepared for modeling by encoding categorical variables using one-hot encoding and scaling the features using StandardScaler.

Two machine learning models were considered: Random Forest and Logistic Regression.

Hyperparameter tuning was performed using Grid Search for both models.

For deep learning, a feedforward neural network using TensorFlow/Keras was created.

Model Evaluation:

Model performance metrics such as accuracy, precision, recall, and F1-score were calculated to evaluate the models.

Grid Search helped identify the best hyperparameters for the Random Forest model.

The Logistic Regression model was evaluated for different regularization strengths.

The deep learning model was trained and tested using TensorFlow/Keras.

Model Performance Metrics

Logistic Regression Metrics:

Accuracy: 0.50255: This is the ratio of correctly predicted instances to the total instances. In this case, the model predicts the correct outcome approximately 50.26% of the time.

Precision: 0.5047: Precision measures the ratio of true positive predictions to all positive predictions made by the model. In this case, about 50.47% of the positive predictions were correct.

Recall: 0.3320: Recall, also known as Sensitivity or True Positive Rate, measures the ratio of true positives to all actual positives in the dataset. About 33.20% of actual positive cases were correctly identified.

F1 Score: 0.4006: The F1 Score is the harmonic mean of precision and recall. It provides a balance between precision and recall. In this case, the F1 score is 0.4006.

Random Forest Metrics:

Accuracy: 0.4998: The Random Forest model achieves an accuracy of approximately 49.98% on the test dataset.

Precision: 0.4940: The precision of the Random Forest model is 49.40%, indicating that 49.40% of its positive predictions are correct.

Recall: 0.4370: The recall of the Random Forest model is 43.70%, suggesting that it correctly identifies about 43.70% of the actual positive cases.

F1 Score: 0.4638: The F1 score of the Random Forest model is 0.4638, which is the harmonic mean of precision and recall.

Confusion Matrix: The confusion matrix shows the distribution of true positive, true negative, false positive, and false negative predictions. In this case, there are 8,506 true positives, 6,646 true negatives, 8,359 false positives, and 6,489 false negatives.

Conclusion:

Both models have relatively similar performance, with accuracy close to 50%. The Random Forest model appears to have slightly better precision, recall, and F1 score compared to Logistic Regression.

Keep in mind that model performance can vary depending on the dataset, and it's essential to consider other factors such as the business context and the specific problem you are trying to solve when evaluating these results. Additionally, you can further fine-tune your models or explore other algorithms to potentially improve performance.