Concordia University

Assignment 1
Group 74
(This document: Theoretical & Business Reports + Figures)

Ryan Li 40214839
Tuan Anh Pham 40213926
Mustafa Sameem 40190889
Antoine Cantin 40211205

SOEN 471
Big Data Analytics
Dr. Reza Mirsalari

SECTION 1

Why are decision trees useful in customer churn prediction?

Customer Churn using Decision Trees can be a very good choice for interpretability in a business environment. For stakeholders or management; it provides a high-level intuitive interpretation of the model used to predict churn. Such actors could be shown the decision trees with each branch being a question and each node being a label/category all in natural language. This would allow feedback and requirements engineering coming from both a business perspective and a technical perspective. The simple conditional "if else"-like nature of decision trees' paths make it also adequate to make proper business rules or heuristics that go hand in hand with the actual algorithm used to fulfill the churning classifiers or even regression models.

Decision trees are very good for customer support and feedback; and to make it such that business actors in the streaming service can easily make surveys or contact customers with the right questions since after all a decision tree flows along a path of questions (be it in natural language or not). Inversely, surveys and customer support emails with questions can provide themselves good indicators of what type of questions to include, cross-validate or test within the decision tree models; making for an efficient business cyclic model to follow to ensure customer subscriptions' retention. It is known that model-complexity and high variance are cause for instability and overfitting in decision tree models, but with this business cyclic model; pruning (Regularization for DTs) strategies could be used to counter such issues e.g. a survey feedback could show that in contrast to what dataset indicated, it would seem that a certain type of attribute or question has a actually a pretty universal response from customers; therefore the training or test datasets could be expanded with diversified survey results, and pruning to remove low-question-importance branches could be done in response.

The tree's modeling will also allow for versatility, flexibility and scalability. It will be easiest to start with questions in natural language that are yes or no with binary classification; but in the future trees offer also an intuitive migration towards categorical labels (multiclass classification) where instead of predicting "Churning: Yes or No", we could for example ordinal labels such as "Churning Risk: High, Medium, Low". Decision trees could even be turned into a regression problem where we could predict continuous numerical values based on stochastic probability models between features such as "Estimated time left before churn".

This would allow the business to adapt to evolving needs. With multiclass classification of Churning Risk ordinals, a company could take contextually-informed decisions to try to improve their chance of returning customers e.g. High Churn risks customers could be offered premiums, discounts, trials etc… Lower Churn risk customers could be suggested for more strategic and specific improvements such as personal recommendations accuracy if they have poor-click rate or better promotional campaigns to churners. The splitting done recursively while training the tree to partition it into groups at different tree depths is where the "answer" to the question(s) will be picked. In trees, depending on technical requirements and the nature of the data; we have different options e.g. Gini index/impurity can be faster to compute and adequate when we use mode statistics or frequencies along side confusion matrix; since it indicates how often a certain feature value will have low attribute purity; Otherwise Entropy / Information Gain focuses more on the uncertainty of the data and how much each split (each new questions) will add or remove stochastic variables and "randomness" in the outcomes of the current subtree path.

What business actions can be taken based on decision tree predictions?

With dataset growth/diversification (growing customer base), post-pruning, hyperparameter tuning, regularization techniques and cross-validation, decision trees could provide opportunity to the streaming service to flag potential churn-risk customer groups or individuals and engage in proactive support interactions early before problem leads to churning. The split segments/groups/partitions feature weights flowing in tree will make for intuitive insights allowing for companies to set their priorities in terms of customer support and technical requirements of their infrastructure; optimizing their resource allocation strategies. The high-level design, outputs, questions etc.. of the decision tree could allow stakeholders and developers alike to work towards better customer retention strategies; Marketing team could adjust their advertisement strategy based on demographics, Managers could propose thing such as loyalty rewards programs, better competitive pricing tiers, or propose personalized relevant user experience; Developers could use the growing data, classification metrics and statistics of the decision tree to improve their recommendation system or content library UI; Network engineers could extrapolate from the outcomes some data or metadata that could help with live streaming networking modules and hardware (for better streaming quality/bitrate/network volatility resistance). Lasty, it is known that decision trees are easy to set up due to their interpretability for business, but also because they do not need some common pre-processing steps like feature scaling and data normalization. This means that streaming services could use several data sources; whether it be their streaming service itself, customer support or surveys; and pick varied adequate questions and attributes for churn-prediction that are evolving at the same rate as the business model of Streamflex.

Concrete Example of Business Actions:

Targeted Retention Campaigns: If the model concludes that high resolution time correlates with churn to some degree,it could indicate inadequate customer support or certain other services. The business should consider improving it by training their support staff or automating support processes whenever possible if it's a common issue. The model of decision trees makes it such that using the model outcomes the employees could follow the decision paths and know what patterns and questions matter most when reviewing customer service issues that were previously analyzed by the tree.

Content Strategy Adjustments and Advertising: If the model indicates that the preferred content type as a major factor that affects churning or other churning-inducing factors, it suggests that the current content options may not be fully aligned with customers interests or market sentiment. Trends evolve quickly; Decision trees could be used to generate ML adequate questions for surveys to extend the dataset and obtain always up-to-date label predictions. The business should analyze customer feedback, surveys or directly feature attributes to optimize the content preference distribution and how it correlates to customer retention. The easy interpretability of decision trees will allow stakeholders and developers alike to contribute to designing loyalty and recommendation engines that fit both high risk and lower risk churners.

Billing Optimization: If the model shows that payment issues are a key factor in predicting churn, it suggests that the billing process may be problematic for customers. The business can offer multiple and flexible payment options that align with their paying abilities but also depending on their account tier; premium users expect higher service to not churn.

Personalized Experience and Programs: If the model finds that the subscription length is impacting churn, it implies that customers with shorter subscription periods may be more prone to leaving as they see no incentives to stay longer. To counteract this, the business can offer rebates or loyalty reward programs for customers who opt for longer subscriptions. Even if this incurs some cost short-term, streaming business will benefit most from long-term subscriptions.

SECTION 2

Task 1

To display summary statistics, we used the pandas DataFrame method df.describe() to obtain the following results:

	count	mean	std	min	25%	50%	75%	max
CustomerID	1000.0	500.5	288.82	1.0	250.75	500.5	750.25	1000.0
Age	1000.0	43.82	14.99	18.0	31.0	44.0	56.0	69.0
Subscription_Length_Months	1000.0	18.22	10.18	1.0	9.0	18.0	27.0	35.0
Watch_Time_Hours	1000.0	100.79	56.48	5.04	50.38	100.23	150.45	199.94
Number_of_Logins	1000.0	50.39	28.22	1.0	26.0	51.0	75.0	99.0
Payment_Issues	1000.0	0.15	0.36	0.0	0.0	0.0	0.0	1.0
Number_of_Complaints	1000.0	4.55	2.92	0.0	2.0	5.0	7.0	9.0
Resolution_Time_Days	1000.0	15.27	8.23	1.0	9.0	15.0	22.0	29.0
Churn	1000.0	0.26	0.44	0.0	0.0	0.0	1.0	1.0

Figure 1: Description of Statistics

To identify missing values, we used the df.isnull().sum() function on our pandas DataFrame, which indicated that there were no missing values. This finding is further confirmed by the df.describe() output, where the count for each parameter is exactly 1000, matching the 1000 rows in the customer_churn.csv file.

No missing values were found, so replacing them was unnecessary. However, if any missing data had been present, we could have used the df.fillna(df.mode().iloc[0], inplace=True) function to replace them with the mode, a common method for handling categorical columns. For numerical data, we would have used the df.fillna(df.median(numeric_only=True), inplace=True) function, which fills missing values with the median of the respective column.

Since most machine learning algorithms work with numerical values rather than categorical strings, we converted the categorical variables from the columns "Preferred_Content_Type," "Membership_Type," and "Payment_Method" using label encoding, which maps each unique categorical value to an integer.

For Figure 2, we created a grid of histograms to visualize the distribution of numeric columns in our DataFrame, excluding the redundant "CustomerID" column. Using matplotlib, we set up a subplot grid with three columns and a dynamic number of rows. Each histogram was generated with 15 bins and included appropriate titles and axis labels. Finally, we used plt.tight_layout() to ensure proper spacing before displaying the plots.

For Figure 3, we generated box plots for the numeric columns using the boxplot() function to visualize distributions and identify outliers. We adjusted the figure size, axis labels, and title for improved clarity.

For Figure 4, we created a heatmap to display pairwise correlations between the numeric features, where each cell shows the correlation coefficient based on color.

Preparing Task 2-4 Model evaluation metrics: focus on Recall for scoring and cross-validation

For our models a lot of experimenting was done with hyperparameter tuning, sampling weights, balancing class distributions, cross-validation strategies, test dataset scoring thresholds and more. But it is quite difficult to make a reliable model with a dataset of only 1000 rows, even if the data is diversified and high quality. In reality, more data and transfer learning would be needed.Nonetheless, some outcomes quantities of the churn binary classification matter more for Streamflex's purpose; here they are ranked in descending order of business importance:

1. **False Negatives:** How many churners were ignored. This is the highest business cost because losing subscribers on the long term is probably the largest financial strain for a streaming service.

2. **True Positives:** How many churners were detected by model. TP must be kept high because we want to service as many churners as possible, but its business risk is lesser than FN; this is because even if we overfit/bias and identify more churners than reality (high FP), if FPi s too high relative to TP than resources to assist churners will be reduced. But FN is more important since conversely, the TN incurs no risk or investment as long as FN is maximized.

3. **False Positives:** How many people that will stay subscribe but were predicted as churners. Even if FP is high/higher, if TP is relatively high the cost of FP will be worth the benefits of TP.

4. **True Negatives:** How many people will stay subscribed as predicted. Business cost of TN is negligible.

Since FN is most important and TP second most important, it is natural that recall will be our classification metrics/scoring of choice to optimize our training, cross-validation and other evaluation techniques on; since recall = tp/(tp+fn) (FN has greatest impact, followed by TP)

Task 2

Data Preparation And Resampling

The CustomerID and Churn were excluded from the features and the Churn was used as the target variable. The dataset was split into 80% training and 20% testing sets using a fixed random state (42) for reproducibility. We opted for 20% testing because the dataset was imbalanced and we needed more training of the minority class. To fight this imbalance, using adaptive synthetic sampling, we generated synthetic samples for the minority class.

Preprocessing

We tried different preprocessing solutions to best fit the dataset. The numerical features were standardized to the same scale for fairer training. Object features were one hot encoded to handle and distinguish categories for interpretability.

Hyperparameter Tuning And Model Training

For choosing the best hyperparameters for our model, we used GridSearchCV with hyperparameter grid (including settings for criterion, max_depth, min_samples_split, min_samples_leaf, class weight, and max features). Stratified 5-fold cross-validation, with each fold having the same class distribution, to evaluate all combinations and select the best model based on recall scoring.

Pruning

To avoid overfitting from cost complexity pruning was done over the model. This process removed branches that did not improve prediction and that overall increased complexity.

Results

Decision Tree Accuracy: 0.435 Decision Tree Precision: 0.217

Decision Tree Recall: 0.434 Decision Tree F1 Score: 0.289

TN=64 FP=83 FN=30 TP=23

With the grid search, it was found that the model's best settings were to use entropy for splitting, to have tree depth of 3, to need at least 2 samples to split a node, to allow a minimum of 2 samples per leaf, to use a balanced class weight, and to use the square root of feature count.

Task 3

Data Preparation And Sample Weight
Data split and features and target variables were the same as the decision tree. For the random forest, we assigned higher weight to the minority class to prevent imbalance issues. This is that during training, the model will not ignore the minority class.

Hyperparameter Tuning And Model Training
To find the best hyperparameters, we used GridSearchCV again with the hyperparameter grid (including settings for criterion, max_depth, min_samples_split, min_samples_leaf, number of estimators, and max features).

Cross Validation And Testing
Our stratified cross validation method will use 10 folds, with 9 for training and 1 for testing. To reduce biases, the data will be shuffled before splitting. Instead of using predict, we are using predict_proba to apply a custom threshold of 0.46. Threshold was lowered to better detect churn customers for higher recall. However, this threshold also comes with a trade-off of an increase of false positives.

Results

Random Forest Accuracy: 0.42 Random Forest Precision: 0.27
Random Forest Recall: 0.698 Random Forest F1 Score: 0.389
TN=47 FP=100 FN=16 TP=37

With the grid search, it was found that the model's best settings were to use entropy for splitting, to have tree depth of 3, to need at least 10 samples to split a node, to allow a minimum of 2 samples per leaf, to use 200 estimators, and to use the square root of feature count.

Comparison And Discussion
All performance metrics were improved except for accuracy. Accuracy moved from 43.5% to 42%. Although accuracy can be theoretically interesting, it is not important as this metric does not align with the business goals as explained in section 1. Precision increased from 21.7% to 27%. F1 score saw a raise from 28.9% to 39%. Recall with a significant increase from 43.4% to 69.8%. The random forest predicted lower false negatives and higher false positives, which are reflected by the increase in recall score and f1 score.

Our random forest performed better especially in the case of recall. Given that our dataset size was very small and had imbalance classes, the random forest was able to improve results. This can be due to many factors. Our dataset had 10 features, some of which may have been irrelevant and misleading to the churn rate. Decision tree branches split on features, but the forest diversely selects features per tree making it more robust and less susceptible to noisy features. A small dataset is more susceptible to overfitting due to its limited patterns, and high changes in prediction from small data change. A random forest uses multiple decision trees, averaging their outputs to get a better generalized pattern and avoids learning patterns too specifically. By using weighted subsets of data for each tree, it avoids being biased towards the majority class. Edge cases will impact the model disproportionately, creating deep branches and more overfitting and complexity. Post pruning is a technique to remove these unnecessary deep branches to better generalize a meaningful pattern.

Task 4

The most contributing factor to customer churn is watch time hours, followed by number of logins, subscription length, and resolution time days. Watch time hours with a large significance compared to the other 3 characteristics that mostly contribute the same amount.

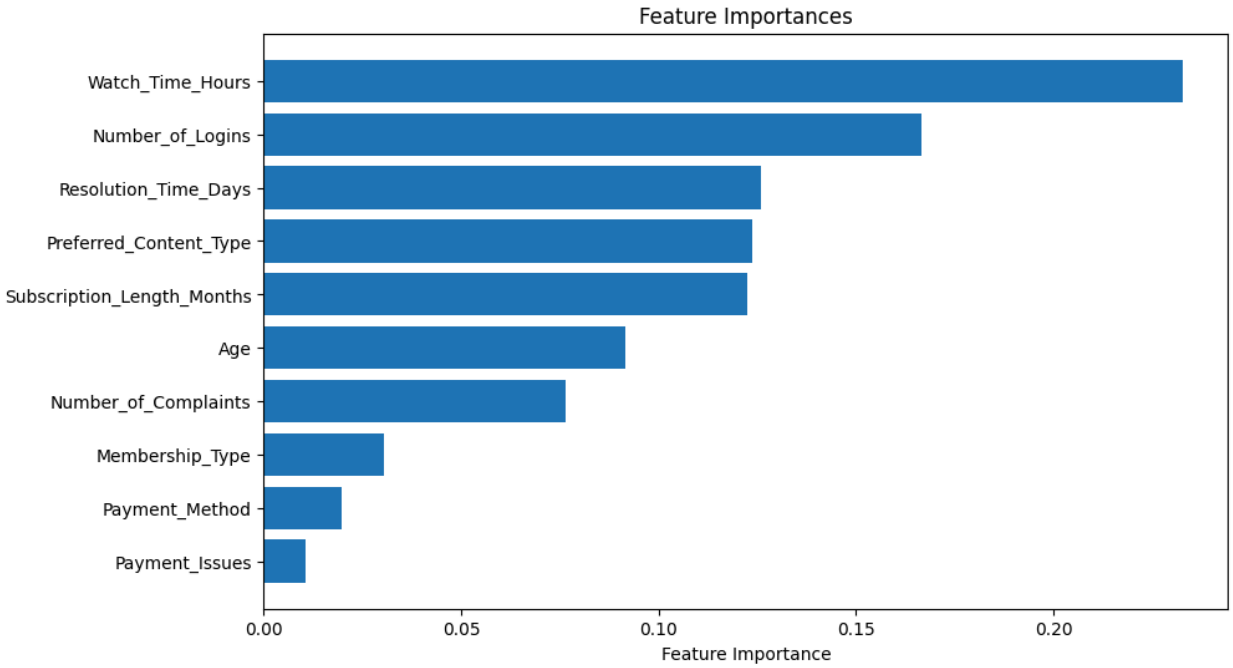

Figure 6: Feature Importance From Random Forest Model

Personalized Engagement And Binge Formats For Watch Time

Customers value being engaged and satisfied. AI powered personalized recommendation will catch a user's attention, drawing them back to continue using the service. Daily trending content, eye catching thumbnails, autoplay, and previews can keep the user's engagement. Releasing content in a binge format increases watch time.

Subscription Retention And Loyalty Programs For Subscription Length

Short subscription plans may correlate to a higher churn rate, therefore encouraging longer plans to customers is our goal. Incentive and loyalty programs such as exclusive offers and discounts and bundled value plans, can make longer plans more inviting. A loyalty system offering perks like early access to new content to long-term subscribers.

Login Gamification For Logins And Reminders

Creating login gamification like login streaks and watch streaks for rewards incentivises users to login daily. This increases user engagement by spending more time on the platform across multiple days raising exposure to trending content and personalized recommendations. Notifications, reminders, and personalized emails for recommendations and ongoing shows help to remind users to login.

APPENDIX

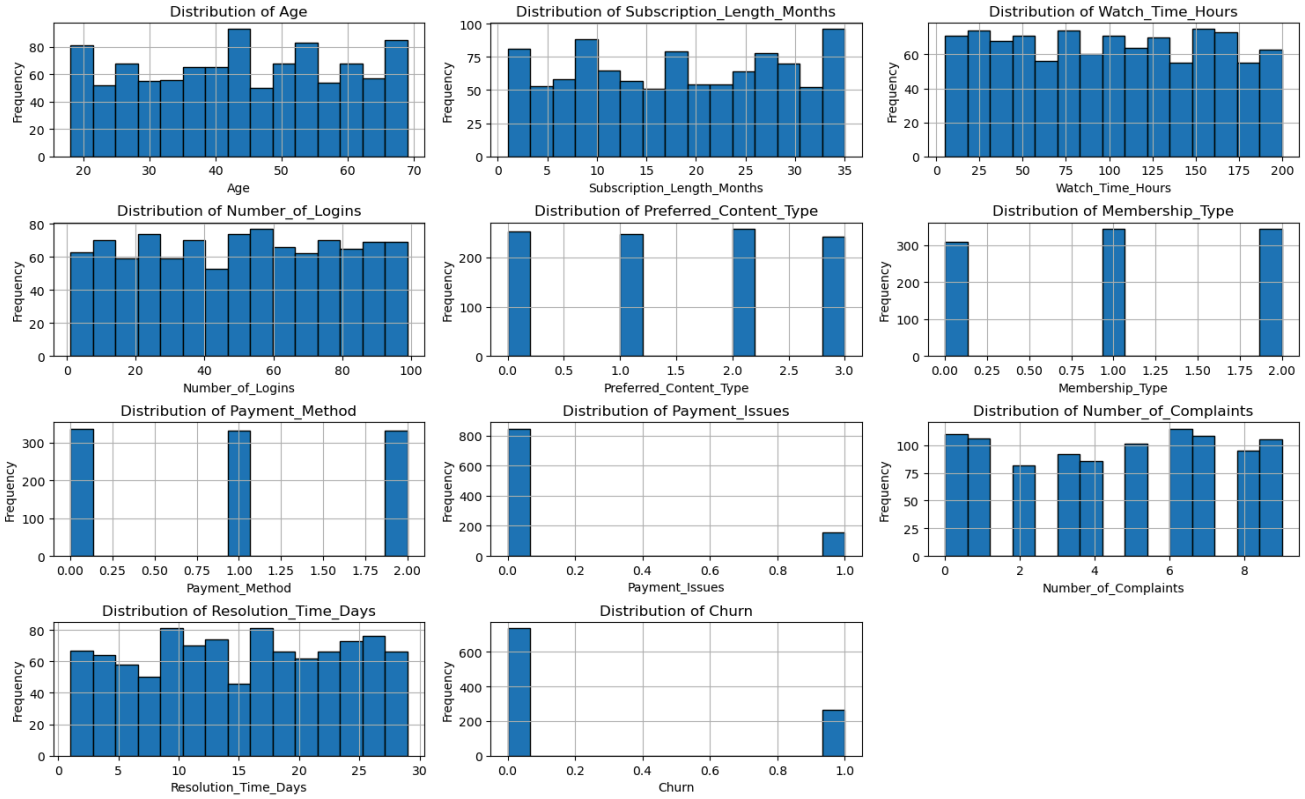

Figure 2: Histograms for Numeric Columns

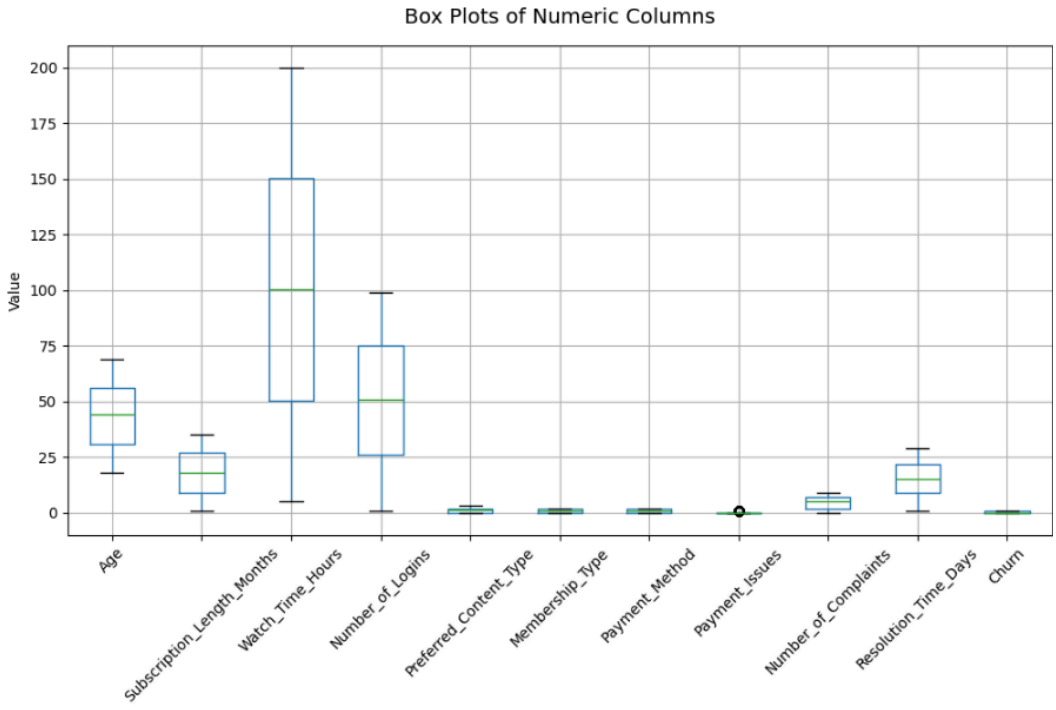

Figure 3: Box Plots of Numeric Columns

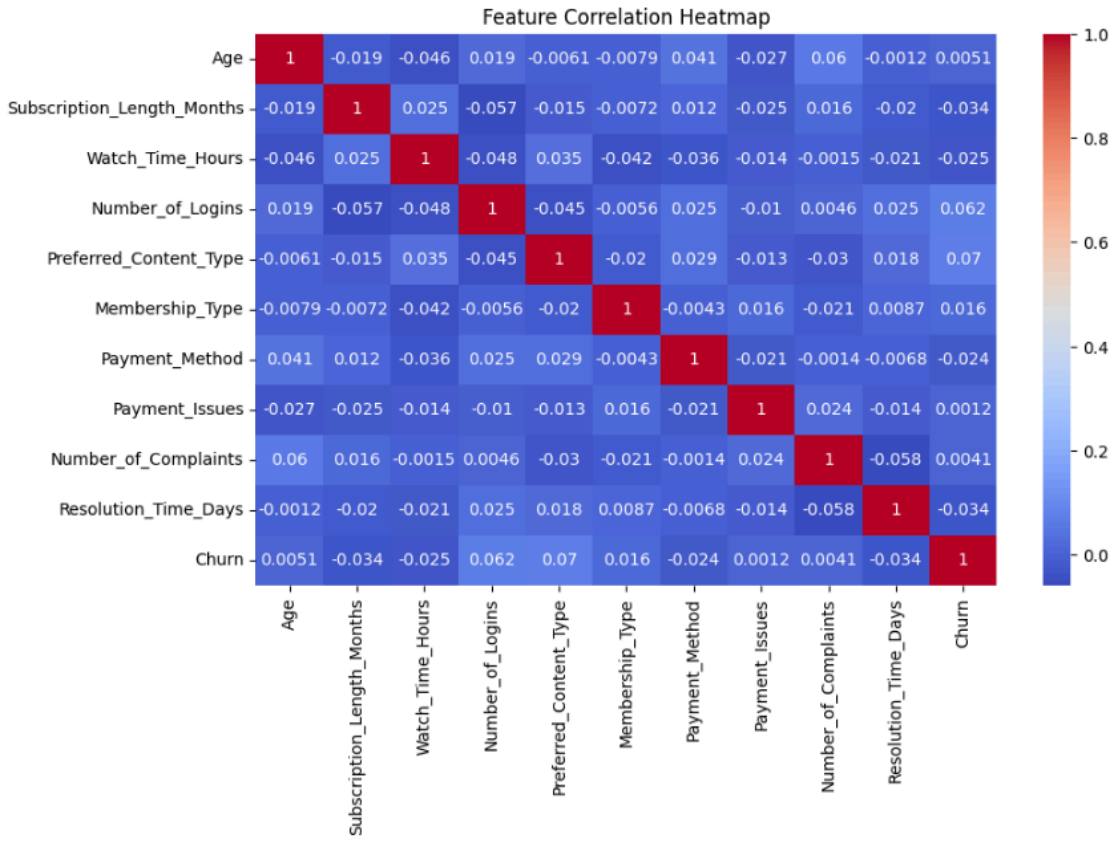

Figure 4: Feature Correlation Heatmap

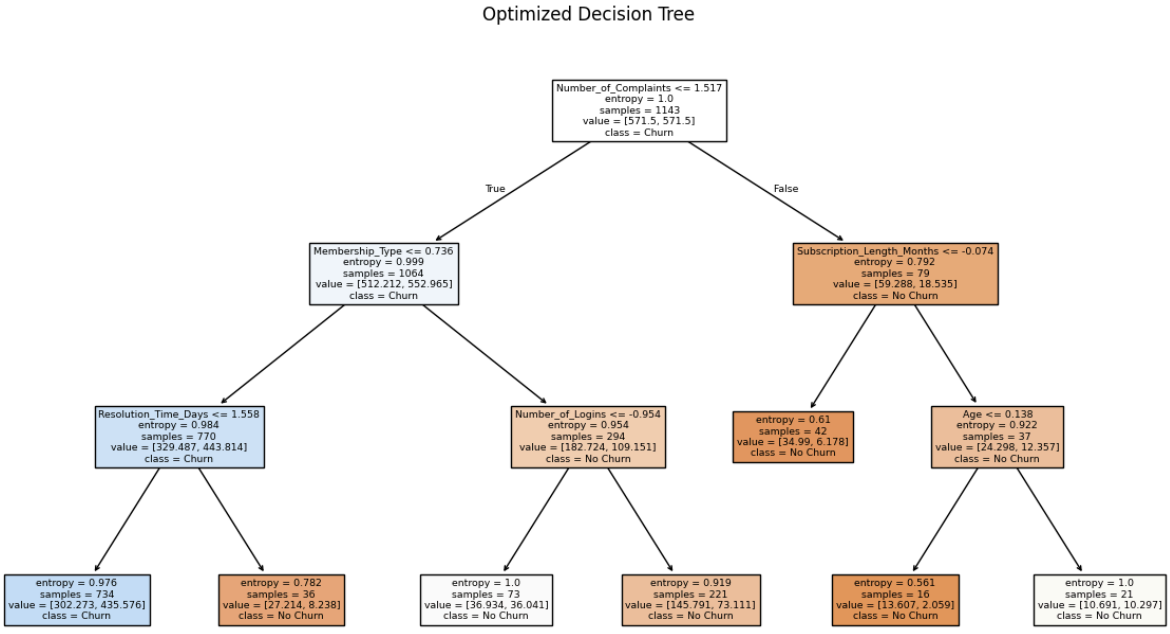

Figure 5: Decision Tree