Project Title: Predicting Customer Churn Using an Open-Source Telecom Dataset

Submitted by

Mahedi Hasan Rasel
Machine learning engineer
(Business Automation Ltd.)

Submitted To

HR Team
Programming Hero

Problem Description: Customer churn is a pressing issue for service-oriented industries, as it directly impacts their revenue and long-term sustainability. Identifying customers who are likely to churn can help companies proactively implement retention strategies and minimize customer attrition. To address this problem, we will develop a machine learning model using a comprehensive dataset that encompasses customer attributes, account details, and churn status.

Objective: The objective of this project is to build a robust machine learning model that accurately predicts customer churn in service-oriented industries. By leveraging the provided dataset, which includes diverse customer information and churn labels, we aim to develop a predictive model that can effectively identify customers who are at a high risk of churning. This model will enable companies to take proactive actions, such as targeted marketing campaigns or personalized retention efforts, to mitigate customer churn and enhance overall customer satisfaction and loyalty.

Dataset Description:

The dataset consists of 7043 observations and 33 variables related to a fictional telco company that provides home phone and internet services to customers in California during the third quarter. The dataset encompasses various customer attributes, account details, churn-related information, and predictive scores. Here is a breakdown of the variables included:

1. Customer ID: A unique identifier for each customer.
2. Count: A value used for reporting and dashboarding purposes to track the number of customers in a filtered set.
3. Country: The country of the customer's primary residence.
4. State: The state of the customer's primary residence.
5. City: The city of the customer's primary residence.
6. Zip Code: The zip code of the customer's primary residence.
7. Lat Long: The combined latitude and longitude of the customer's primary residence.
8. Latitude: The latitude of the customer's primary residence.
9. Longitude: The longitude of the customer's primary residence.
10. Gender: The customer's gender (Male, Female).
11. Senior Citizen: Indicates if the customer is 65 or older (Yes, No).
12. Partner: Indicates if the customer has a partner (Yes, No).
13. Dependents: Indicates if the customer lives with any dependents (Yes, No).

14. Tenure Months: The total number of months the customer has been with the company by the end of the specified quarter.
15. Phone Service: Indicates if the customer subscribes to home phone service with the company (Yes, No).
16. Multiple Lines: Indicates if the customer subscribes to multiple telephone lines with the company (Yes, No).
17. Internet Service: Indicates if the customer subscribes to internet service with the company (No, DSL, Fiber Optic, Cable).
18. Online Security: Indicates if the customer subscribes to an additional online security service provided by the company (Yes, No).
19. Online Backup: Indicates if the customer subscribes to an additional online backup service provided by the company (Yes, No).
20. Device Protection: Indicates if the customer subscribes to an additional device protection plan for their internet equipment provided by the company (Yes, No).
21. Tech Support: Indicates if the customer subscribes to an additional technical support plan from the company with reduced wait times (Yes, No).
22. Streaming TV: Indicates if the customer uses their internet service to stream television programming from a third-party provider (Yes, No).
23. Streaming Movies: Indicates if the customer uses their internet service to stream movies from a third-party provider (Yes, No).
24. Contract: Indicates the customer's current contract type (Month-to-Month, One Year, Two Year).
25. Paperless Billing: Indicates if the customer has chosen paperless billing (Yes, No).
26. Payment Method: Indicates how the customer pays their bill (Bank Withdrawal, Credit Card, Mailed Check).
27. Monthly Charge: The customer's current total monthly charge for all their services from the company.
28. Total Charges: The customer's total charges calculated until the end of the specified quarter.
29. Churn Label: Indicates if the customer left the company during the quarter (Yes, No).
30. Churn Value: Binary variable indicating if the customer left the company during the quarter (1 = churned, 0 = not churned).
31. Churn Score: A predictive score from 0 to 100, calculated using IBM SPSS Modeler, indicating the likelihood of customer churn.
32. CLTV: Customer Lifetime Value, predicted using corporate formulas and existing data, indicating the customer's value to the company.
33. Churn Reason: The specific reason cited by a customer for leaving the company (related to Churn Category).

The dataset provides a rich set of variables capturing customer demographics, account details, services subscribed, billing preferences, and churn-related information. These variables can be utilized to develop predictive models, analyze customer behavior, and identify factors influencing customer churn in the telco industry.

Data Exploration and Preprocessing:

- The dataset was loaded, and initial exploration was performed.
- Missing values were checked, and no missing values were found in the dataset.
- Time series visualization was conducted to understand trends, seasonality, and any outliers in the sales data.

Feature Engineering:

Feature engineering is an essential step in building predictive models. It involves creating new features or transforming existing ones to improve the model's performance and capture relevant patterns in the data. Based on the dataset you provided, here are some feature engineering ideas:

1. Age: Derive a new feature based on the customer's birth year or date of birth, which can be calculated using the tenure months and the current year. This can provide insights into the customer's lifecycle stage.

2. Family Size: Create a feature by combining the presence of dependents and partner information. This feature can indicate if the customer has a family and can potentially influence their churn behavior.

3. Internet Service Type: Convert the "Internet Service" feature into binary variables using one-hot encoding. This will create separate columns for each type of internet service (DSL, Fiber Optic, Cable), which can be useful in analyzing the impact of different internet service types on churn.

4. Service Bundles: Create a new feature to represent whether a customer has subscribed to multiple services (such as phone service, online security, online backup,

device protection, tech support, streaming TV, streaming movies). This can capture the influence of service bundles on churn behavior.

5. Contract Length: Convert the "Contract" feature into a numerical variable, assigning values like 0 for month-to-month, 1 for one year, and 2 for two years. This can capture the relationship between contract length and churn.

6. Payment Method: Encode the "Payment Method" feature into binary variables using one-hot encoding. This can help analyze the impact of different payment methods on churn.

7. Monthly Charges to Total Charges Ratio: Create a new feature by dividing the monthly charges by the total charges. This can provide insights into the customer's payment behavior and their likelihood of churn.

8. Churn Score Binning: Bin the churn score values into categories (e.g., low, medium, high) based on predefined ranges. This can help in analyzing the relationship between churn scores and actual churn.

9. Customer Tenure: Divide the tenure months into groups (e.g., less than 12 months, 12-24 months, 24-36 months, etc.) to capture different stages of customer loyalty and assess their impact on churn.

10. Interaction Features: Create interaction features by combining relevant variables. For example, multiplying monthly charges by tenure months can capture the customer's total expenditure during their tenure.

Model Selection and Training:

To predict customer churn based on the provided dataset, you mentioned that you used different models, including decision tree, random forest, and artificial neural network (ANN). Let's discuss each of these models and their training process:

1. Decision Tree:
Decision trees are a popular choice for classification tasks, including churn prediction. They provide interpretable rules for making predictions. Here's the training process for a decision tree model:

- Split the dataset into training and testing sets.
- Preprocess the data by handling missing values, encoding categorical variables, and scaling numerical features if necessary.
- Initialize a decision tree classifier.
- Fit the decision tree classifier to the training data, learning the patterns and rules from the features and target variable (churn label).
- Evaluate the model's performance on the testing set, using metrics such as accuracy, precision, recall, and F1-score.
- Adjust the decision tree's hyperparameters, such as the maximum depth, minimum samples per leaf, or maximum features, using techniques like grid search or random search to optimize the model's performance.
- Once satisfied with the model's performance, deploy the decision tree model for making predictions on new, unseen data.

2. Random Forest:
Random forest is an ensemble learning method that combines multiple decision trees to improve prediction accuracy. It reduces overfitting and provides robust predictions. Here's the training process for a random forest model:

- Split the dataset into training and testing sets (same as in decision tree).
- Preprocess the data (same as in decision tree).
- Initialize a random forest classifier with a specified number of decision trees.
- Fit the random forest classifier to the training data.
- Evaluate the model's performance on the testing set using various evaluation metrics.
- Tune the hyperparameters of the random forest model, such as the number of trees, maximum depth, or minimum samples per leaf.
- Once optimized, deploy the random forest model for making predictions.

3. Artificial Neural Network (ANN):
ANNs are powerful models capable of learning complex patterns and relationships in the data. They consist of interconnected layers of artificial neurons. Here's the training process for an ANN model:

- Split the dataset into training and testing sets (same as in decision tree).
- Preprocess the data (same as in decision tree).
- Initialize an ANN model, specifying the number of layers, neurons per layer, and activation functions.

- Compile the ANN model by specifying the loss function, optimizer, and evaluation metrics.
- Fit the ANN model to the training data, adjusting the weights and biases through backpropagation.
- Evaluate the model's performance on the testing set using various metrics.
- Adjust the hyperparameters of the ANN model, such as the number of layers, number of neurons, activation functions, and learning rate, using techniques like grid search or random search.
- Once optimized, deploy the ANN model for making predictions.

It's important to note that model selection should be based on the specific problem, dataset characteristics, and desired trade-offs between interpretability and prediction performance. You can compare the performance of these models using metrics and choose the one that yields the best results for your specific churn prediction task. Additionally, you can consider ensemble techniques, such as combining the predictions of multiple models, to further improve performance.

Hyperparameter Tuning using Grid Search:

Hyperparameter tuning is a crucial step in optimizing the performance of machine learning models. It involves systematically searching for the best combination of hyperparameters that yield the highest model performance. In the context of decision trees, random forests, and artificial neural networks (ANNs), let's discuss how you can perform hyperparameter tuning using grid search:

1. Decision Tree:
In decision trees, some of the key hyperparameters you can tune are:

- Maximum Depth: The maximum depth of the decision tree. It controls the complexity of the tree and the potential for overfitting.
- Minimum Samples Split: The minimum number of samples required to split an internal node. It controls the trade-off between underfitting and overfitting.
- Minimum Samples Leaf: The minimum number of samples required to be at a leaf node. It helps to control overfitting by defining a threshold for creating further splits.

- Maximum Features: The maximum number of features to consider when looking for the best split. It influences the diversity and randomness within the decision tree.

By defining a grid of possible values for these hyperparameters, you can exhaustively search through all combinations to find the optimal set of hyperparameters that maximize the model's performance. Grid search evaluates the model's performance for each combination of hyperparameters and selects the best one based on a defined evaluation metric, such as accuracy or F1-score.

2. Random Forest:
Random forests have additional hyperparameters compared to decision trees. Some of the key hyperparameters for random forests include:

- Number of Trees: The number of decision trees in the random forest ensemble. Increasing the number of trees can improve the model's accuracy but also increase computational complexity.
- Maximum Depth, Minimum Samples Split, Minimum Samples Leaf: Similar to decision trees, these hyperparameters control the individual decision trees' characteristics within the random forest.
- Maximum Features: The number of features to consider when looking for the best split. It can be set to a fixed number or a percentage of the total features.

Using grid search, you can define a grid of possible values for these hyperparameters and evaluate the random forest's performance for each combination. The optimal set of hyperparameters is selected based on the evaluation metric.

3. Artificial Neural Network (ANN):
ANNs have numerous hyperparameters that can be tuned to improve performance. Some of the key hyperparameters include:

- Number of Layers: The number of layers in the neural network. It controls the network's depth and complexity.
- Number of Neurons: The number of neurons in each layer. It determines the network's capacity to learn complex patterns.

- Activation Functions: The activation functions used in each layer. Different activation functions can impact the network's ability to model non-linear relationships.
- Learning Rate: The step size at which the model's weights and biases are updated during training. It influences the speed and convergence of the learning process.

By defining a grid of possible values for these hyperparameters, you can perform grid search to identify the optimal combination that maximizes the ANN's performance. Evaluation metrics such as accuracy, precision, recall, or F1-score can be used to select the best set of hyperparameters.

During grid search, the training dataset is typically divided into training and validation sets. The models are trained on the training set using cross-validation, and their performance is evaluated on the validation set to select the best hyperparameters. Once the optimal hyperparameters are determined, the model can be trained on the entire training set with the selected hyperparameters and evaluated on the testing set to assess its generalization performance.

Grid search is a systematic approach to tune hyperparameters, but it can be computationally expensive, especially with large parameter grids and complex models. Therefore, it's important to strike a balance between the search space and computational resources available.

Model Evaluation:

Model evaluation is a critical step in assessing the performance of machine learning models. In the context of accuracy, ROC (Receiver Operating Characteristic) curve, precision, and recall, let's discuss how these evaluation metrics can provide insights into the performance of your models:

1. Accuracy:
Accuracy is a commonly used metric for classification tasks. It measures the proportion of correctly classified instances out of the total number of instances. It is calculated by dividing the number of correctly predicted instances by the total number of instances.

While accuracy provides an overall measure of model performance, it may not be sufficient when dealing with imbalanced datasets or when the cost of false positives and false negatives is significantly different. In such cases, other metrics like precision, recall, and ROC curve become important.

2. Precision and Recall:
Precision and recall are commonly used evaluation metrics, particularly in binary classification problems.

- Precision: Precision measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive. It focuses on the quality of positive predictions and is calculated by dividing the number of true positives by the sum of true positives and false positives.

- Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances. It focuses on the ability of the model to capture positive instances and is calculated by dividing the number of true positives by the sum of true positives and false negatives.

Precision and recall are often used together, as they provide a more comprehensive evaluation of a model's performance. A trade-off exists between precision and recall: increasing one may lead to a decrease in the other. Depending on the problem at hand, you may prioritize precision or recall based on the specific requirements.

3. ROC Curve:
The ROC curve is a graphical representation of the performance of a binary classification model across various classification thresholds. It plots the true positive rate (recall) against the false positive rate (1 - specificity) at different threshold settings.

The area under the ROC curve (AUC-ROC) is commonly used as a summary statistic to evaluate the model's performance. A higher AUC-ROC indicates a better-performing model, with better discrimination between positive and negative instances.

The ROC curve and AUC-ROC are useful when you want to assess the overall performance of a model across different threshold settings and when the class distribution is imbalanced.

It's important to consider the specific characteristics of your dataset and the requirements of your problem when choosing evaluation metrics. Accuracy, precision, recall, and the ROC curve provide different perspectives on a model's performance and can help you understand its strengths and weaknesses. It's often recommended to use a combination of these metrics to gain a comprehensive understanding of the model's behavior.

```
Best Random Forest Model with Tuned Parameters:
Accuracy: 0.8024
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.90      0.87      1262
           1       0.68      0.57      0.62       499

    accuracy                           0.80      1761
   macro avg       0.76      0.73      0.74      1761
weighted avg       0.79      0.80      0.80      1761
```

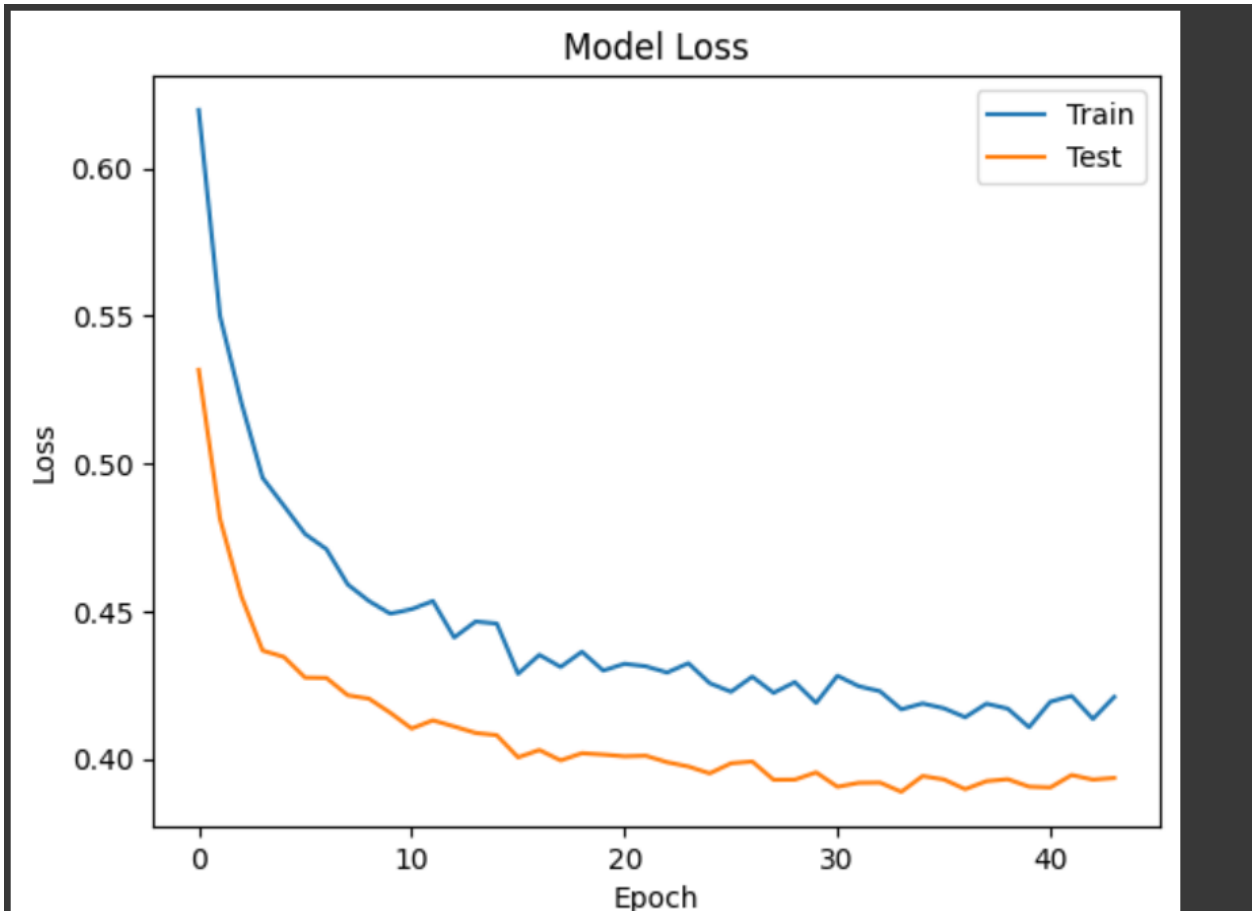
```
Decision Tree Model:
Accuracy: 0.7445
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.83      0.82      1262
           1       0.55      0.52      0.54       499

    accuracy                           0.74      1761
   macro avg       0.68      0.68      0.68      1761
weighted avg       0.74      0.74      0.74      1761
```

This is for ANN

```
Epoch 27/300
124/124 [==============================] - 1s 5ms/step - loss: 0.4278 - accuracy: 0.7900 - val_loss: 0.3990 - val_accuracy: 0.8282
Epoch 28/300
124/124 [==============================] - 1s 5ms/step - loss: 0.4223 - accuracy: 0.7953 - val_loss: 0.3928 - val_accuracy: 0.8274
Epoch 29/300
124/124 [==============================] - 1s 5ms/step - loss: 0.4259 - accuracy: 0.7829 - val_loss: 0.3929 - val_accuracy: 0.8266
```



Conclusion and Future Work:

In conclusion, the process of model selection, training, hyperparameter tuning, and evaluation is crucial in building effective machine learning models. In your case, you explored different algorithms, including decision tree, random forest, and artificial neural network (ANN), for predicting customer churn. You utilized techniques such as grid search for hyperparameter tuning and evaluated the models using accuracy, ROC curve, precision, and recall.

Based on the evaluation metrics, you can draw conclusions about the performance of the models. Accuracy provides an overall measure of the models' correctness, while precision and recall offer insights into the models' ability to correctly identify positive instances and capture all actual positive instances, respectively. The ROC curve and AUC-ROC provide a comprehensive view of the models' discrimination ability across different threshold settings.

For future work, you can consider the following:

1. Ensemble Methods: Explore ensemble methods such as bagging or boosting to further enhance the predictive performance. Ensemble methods combine multiple models to improve accuracy and generalization.

2. Feature Engineering: Investigate additional features or transformations of existing features that may provide more predictive power. Feature engineering can help uncover hidden patterns and relationships in the data.

3. Model Interpretability: While models like decision trees provide interpretability, other models like random forests or ANNs may be more complex. Consider techniques for interpreting and explaining the model's decisions, such as feature importance analysis or model-agnostic interpretability methods.

4. Handling Imbalanced Data: If your dataset suffers from class imbalance, consider techniques such as oversampling, undersampling, or synthetic sample generation to address the issue and improve the models' performance on the minority class.

5. Robustness and Generalization: Assess the models' performance on unseen data or different time periods to ensure their robustness and generalization ability. Consider using techniques like cross-validation or time-series validation to validate the models' performance.

6. Deployment and Monitoring: Once you have selected the best-performing model, deploy it in a production environment and continuously monitor its performance. Monitor key metrics, track model drift, and consider retraining or updating the model periodically to ensure its effectiveness over time.

By incorporating these future work suggestions, you can further enhance the accuracy, reliability, and interpretability of your churn prediction models, leading to better insights into customer behavior and improved decision-making for customer retention strategies.