Telecommunication Churn Prediction Using Multi-layered Supervised Learning

Machine learning Project : Supervised learning

April 6th, 2023

Abstract

This project aims to address the issue of customer churn of a telecom company in the US. Customer churn refers to the phenomenon of customers discontinuing their services with a company, and it is a significant concern for businesses in the telecom industry. The ultimate goal is to enable the company to take proactive measures, such as improving customer service, to retain at-risk customers and reduce overall churn rate. By leveraging machine learning techniques, we aim to build supervised learning models that can accurately classify customers as either churners or non-churners. The model will be trained on historical customer data, including various features such as account details, service usage patterns, and customer demographics. The prediction of churn can provide our company with actionable insights to allocate resources strategically, improve customer service, and implement targeted retention strategies.

Data pre-processing

The dataset for this project was downloaded from the Kaggle website by following the provided link here. To handle and manipulate the dataset efficiently, the Python pandas library was utilized, specifically utilizing the DataFrame object.

The first step in the data cleaning process involved inspecting the dataset for missing values and unique values. This was done to ensure data integrity and completeness. Null values, if any, could potentially hinder the analysis and modeling process. By identifying and handling missing values appropriately, we can ensure reliable results.

Additionally, analyzing unique values in the dataset provides insights into the categorical features and their cardinality. This helps in understanding the distribution of different categories within each feature, which can be useful during the exploratory data analysis phase.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   state                   3333 non-null   object
 1   account_length          3333 non-null   int64
 2   area_code               3333 non-null   int64
 3   phone_number            3333 non-null   object
 4   international_plan      3333 non-null   object
 5   voice_mail_plan         3333 non-null   object
 6   number_vmail_messages   3333 non-null   int64
 7   total_day_minutes       3333 non-null   float64
 8   total_day_calls         3333 non-null   int64
 9   total_day_charge        3333 non-null   float64
 10  total_eve_minutes       3333 non-null   float64
 11  total_eve_calls         3333 non-null   int64
 12  total_eve_charge        3333 non-null   float64
 13  total_night_minutes     3333 non-null   float64
 14  total_night_calls       3333 non-null   int64
 15  total_night_charge      3333 non-null   float64
 16  total_intl_minutes      3333 non-null   float64
 17  total_intl_calls        3333 non-null   int64
 18  total_intl_charge       3333 non-null   float64
 19  customer_service_calls  3333 non-null   int64
 20  churn                   3333 non-null   bool
dtypes: bool(1), float64(8), int64(8), object(4)
memory usage: 524.2+ KB
```

Fig. 1 : Insight on the raw data

Feature Engineering

After gaining initial insights into the dataset, the next step was to drop irrelevant features. One such feature identified and removed from the analysis was the phone number. In this context, the phone number is considered irrelevant because it does not provide any meaningful information for predicting customer churn. It serves as a unique identifier for each customer and does not contribute to understanding the underlying patterns or factors leading to churn.

By dropping the phone number feature, the focus shifted to the relevant attributes that are more likely to contribute to predicting churn. These relevant attributes may include account details, service usage patterns, and customer demographics. By excluding irrelevant features, the dataset becomes more streamlined and targeted, enabling more effective analysis and modeling.

To gain insights into the churn behavior across different area codes, we calculated the churn percentage for each area code and generated a bar plot to visualize the number of customers per area code, segmented by churn status.

Fig. 2 : Churn rate per area code

Upon analyzing the data, we observed that customer churn occurs at a relatively consistent rate of approximately 14-15% across all three area codes. The bar plot provides a clear visualization of the distribution of customers among the area codes and their respective churn status.

Interestingly, despite having the highest number of customers, area code 415 experiences churn at a similar rate to the other two area codes. This suggests that the area code itself does not significantly differentiate the likelihood of churn. Therefore, we have made the decision to drop the area code feature from our dataset as it does not provide meaningful insights regarding churn behavior.

```
1  df.drop('area_code', axis = 1, inplace = True)
✓ 0.0s
```

Fig. 3: Code snippet for dropping the area code feature.

Next, we examined the correlation between the minutes, charge, and calls columns to identify any relationships among these variables.

Fig. 4: Correlation Heat Map for the minutes, charge and calls features

Upon analyzing the data, we found that all the minutes and charge features exhibit a perfect correlation, with a correlation coefficient (r) of 1. This finding is expected since the charge is typically calculated based on the number of minutes used. The perfect correlation indicates that as the number of minutes increases, the corresponding charge also increases proportionally.

This correlation analysis highlights the strong relationship between minutes and charge, reinforcing the understanding that higher usage of minutes leads to higher charges for customers.

Feature Encoding

To ensure compatibility with machine learning models, we performed binary encoding on the "international_plan" and "voice_mail_plan" columns. This encoding technique transforms categorical variables into binary representations, where 1 indicates the presence of a plan and 0 indicates the absence of a plan.

Encoding categorical variables into numerical form is crucial because machine learning models primarily operate on numerical data. By converting the "international_plan" and "voice_mail_plan" columns into binary codes, we enable the models to process and analyze these features effectively.

Exploring the Target Variable: Customer Churn

To gain insights into customer churn within the dataset, we examined the number of churned customers and calculated the proportion of customers who churned. In our analysis, the target variable is represented as a binary category, where "True" indicates churned customers and "False" represents non-churned customers.

We generated a bar plot that distinguishes between the "No Churn" and "Churn" categories. This plot provides a clear visualization of the churn rate and helps us understand the distribution of churned and non-churned customers.

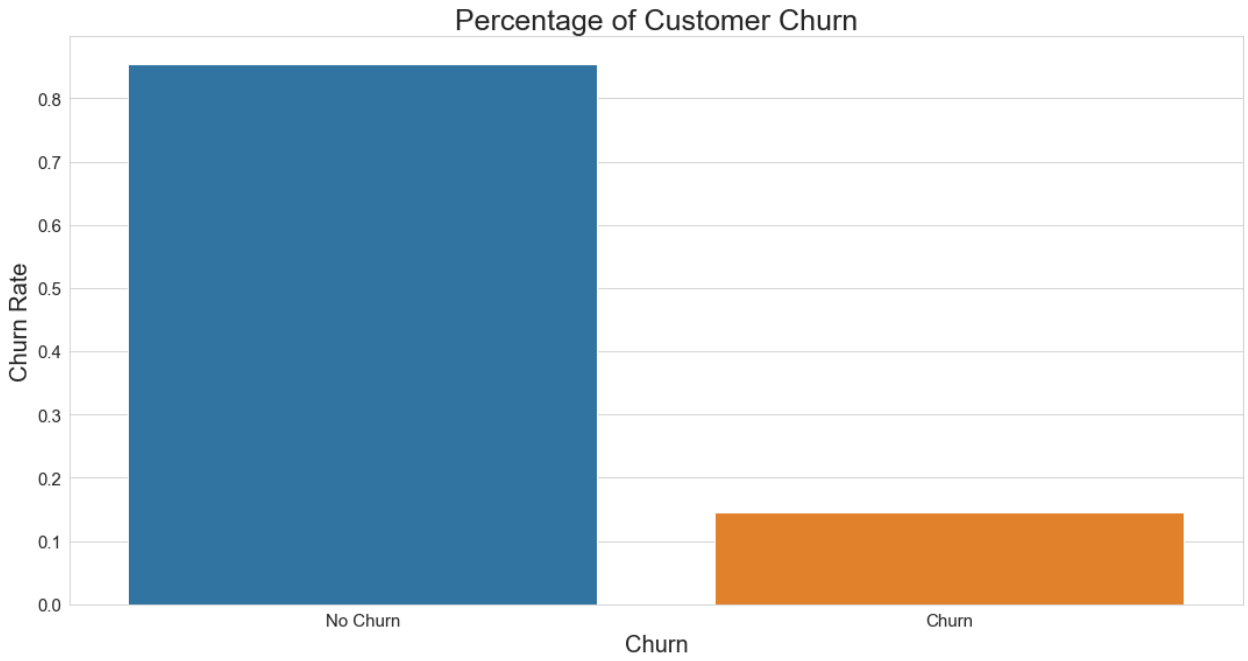

Fig. 5 : Percentage of churn in the whole dataset

The bar plot showcases the relative sizes of the "No Churn" and "Churn" categories, providing a visual representation of the churn rate and aiding in identifying potential patterns or trends. This exploration of the target variable sets the foundation for further analysis and modeling efforts to predict customer churn accurately.

Data Preparation for Model Creation

To prepare the data for model creation, we performed the following steps:

Create Features and Target Dataframes: We separated the dataset into two dataframes: one containing the features (X) and the other containing the target variable (y). This division allows us to train the machine learning model using the feature data and evaluate its performance based on the target variable.

Train-Test Split: We split the data into training and testing sets using a train-test split function. This step ensures that the model is trained on a portion of the data and evaluated on unseen data to assess its generalization capabilities.

One-Hot Encoding of 'state' Variable: As the 'state' variable is an important feature, we performed one-hot encoding on the 'state' column of the X_train dataframe. This encoding technique replaced the categorical 'state' column with a set of binary columns, where each unique state is represented by its own binary column. This transformation enables the machine learning algorithm to effectively process the categorical data and capture any state-specific patterns or correlations.

After performing one-hot encoding, the original 'state' column is replaced by the binary columns representing each unique state, creating a more suitable representation for modeling.

Correlation Analysis: Feature - Target Variable (Churn)

To understand the relationship between each feature and the target variable 'churn', we calculated the correlations using the DataFrame 'df'. The correlations were computed using the corr() function, and the resulting correlation values were sorted in descending order using the sort_values() method with the parameter ascending=False.

This analysis allows us to identify the features that exhibit a strong correlation with the target variable, providing insights into their potential predictive power in determining customer churn.

```
churn                     1.000000
international_plan        0.259852
customer_service_calls    0.208750
total_day_minutes         0.205151
total_day_charge          0.205151
total_eve_minutes         0.092796
total_eve_charge          0.092786
total_intl_charge         0.068259
total_intl_minutes        0.068239
total_night_charge        0.035496
total_night_minutes       0.035493
total_day_calls           0.018459
account_length            0.016541
total_eve_calls           0.009233
total_night_calls         0.006141
total_intl_calls         -0.052844
number_vmail_messages    -0.089728
voice_mail_plan          -0.102148
Name: churn, dtype: float64
```

Fig. 6 : Features’ correlation with target variable

Visualization of the First two Features

In order to gain further insights into the first two features of the dataset, we generated visualizations to explore their distribution and their relationships with the target variable, 'churn'.

We created a DataFrame called 'int_plan_churn' to examine the average churn rate for each category of the 'international_plan' feature. This DataFrame provides insights into the churn rates of customers based on their subscription to an international plan.

Next, we generated a bar plot to visualize the percentage of customer churn for international plan holders. The bar plot, created using the Seaborn library, depicts the churn rates for customers with and without an international plan.

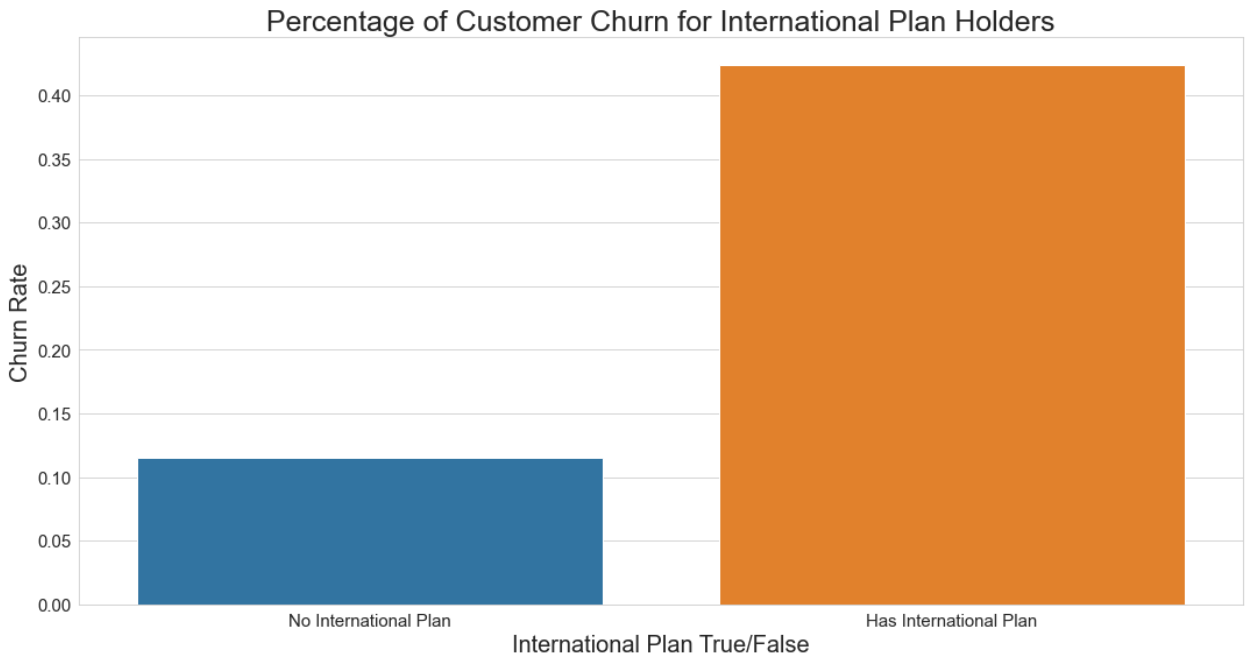

Fig. 7 : Churn rate according to International Plan

According to the analysis, it is evident that having an international plan significantly influences customer churn. Approximately 42% of customers with an international plan end up churning, highlighting the importance of this feature in predicting churn behavior.

Subsequently, we generated a bar plot to visualize the percentage of customer churn based on the number of customer service calls. This bar plot illustrates the churn rates for different categories of customer service calls.

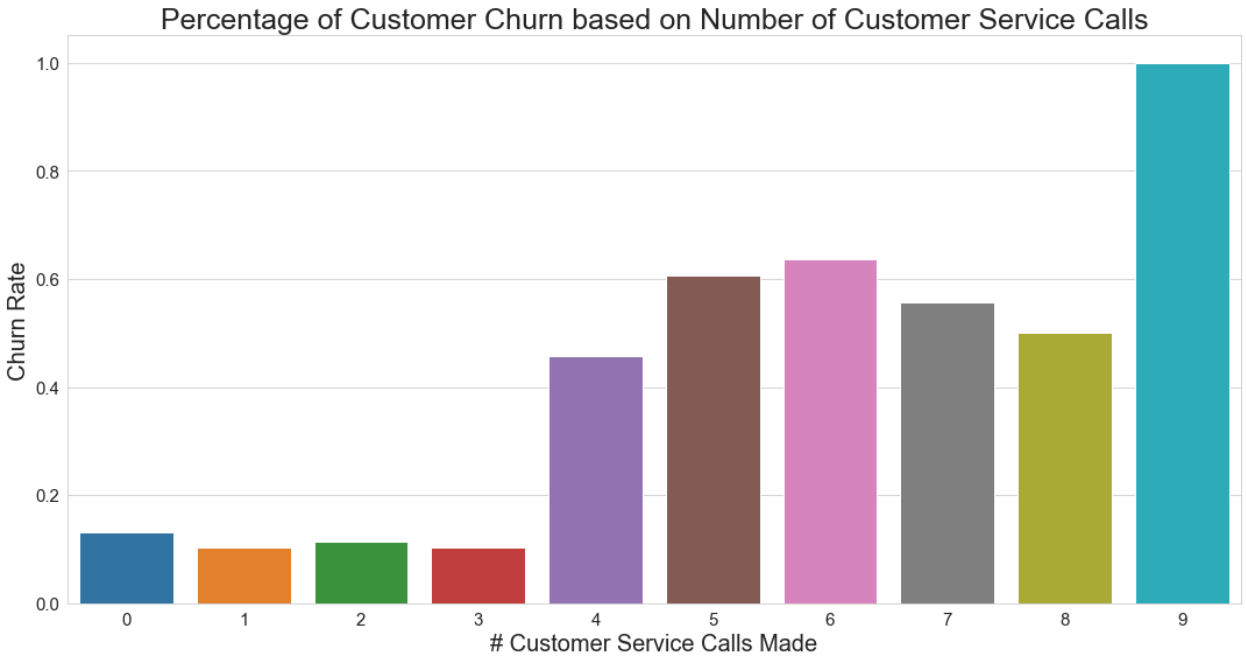

Fig. 8 : Churn rate according to International Plan

According to the analysis, a notable increase in the churn rate is observed for customers who make four or more calls to customer service. These customers exhibit a churn rate exceeding 40%. This finding highlights the significance of the number of customer service calls as an influential factor in customer churn.

Model Training: Gradient Boosting Classifier

For the model training phase, we utilized the Gradient Boosting Classifier algorithm to develop our churn prediction model.

Gradient Boosting Classifier is a multi-layered machine learning algorithm that falls under the ensemble learning family. It builds a strong predictive model by combining numerous weak prediction models, typically decision trees, in a layered fashion. The algorithm iteratively trains new models that specifically target the errors made by the preceding models, thereby

progressively enhancing the overall predictive capability. This layered approach allows Gradient Boosting Classifier to leverage the collective strength of multiple weak models, resulting in a highly accurate and robust ensemble model. Mathematically, Gradient Boosting Classifier can be understood as an additive model where each subsequent model is built to correct the errors made by the previous models. The algorithm starts with an initial prediction (often the mean value of the target variable) and then iteratively adds new models to the ensemble. Each new model is trained on the residuals (the difference between the target variable and the predictions of the current ensemble) of the previous models. By minimizing the residuals in each iteration, the subsequent models learn to capture the remaining patterns and errors in the data.

The training process of Gradient Boosting Classifier involves optimizing a loss function, typically using gradient descent. The loss function quantifies the discrepancy between the predicted and actual values. In each iteration, the algorithm calculates the gradients of the loss function with respect to the predictions of the current ensemble and trains a new model to minimize the loss. The learning rate determines the contribution of each new model to the ensemble, controlling the step size during the optimization process.

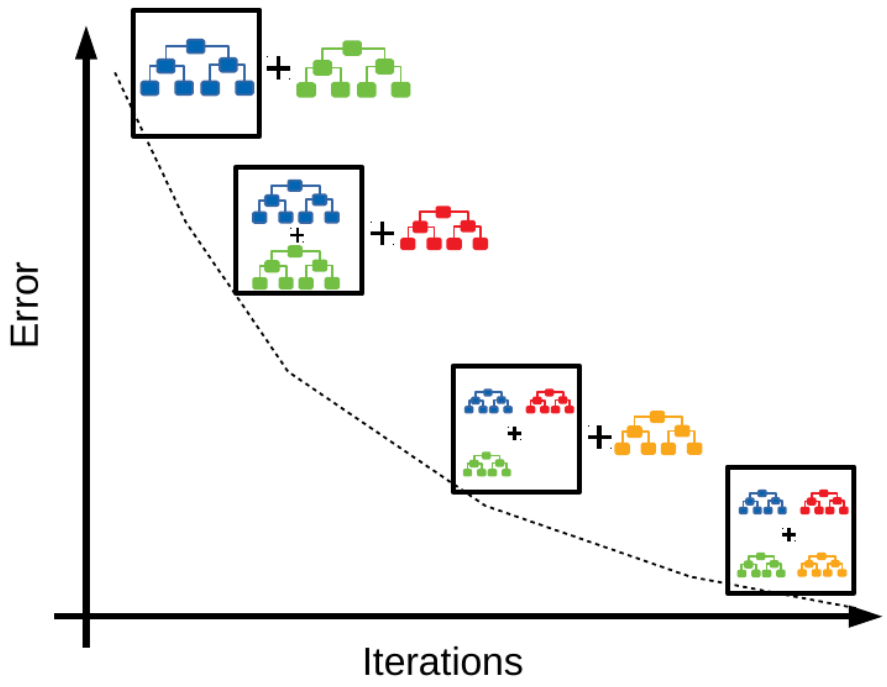

Fig. 9: Error minimization by layering more decision trees to the model.

The specific hyperparameters used for the Gradient Boosting Classifier model are as follows:

- Criterion: 'friedman_mse'
- Learning Rate: 0.01
- Loss Function: 'exponential'
- Maximum Depth: 3
- Minimum Samples Leaf: 0.13636363636363638
- Minimum Samples Split: 0.1

These hyperparameters were chosen based on prior experimentation and tuning to optimize the model's performance.

To ensure robustness and evaluate the model's effectiveness, we employed cross-validation using a KFold strategy. Cross-validation helps assess the model's generalization capabilities by splitting the data into multiple subsets and performing training and evaluation on different combinations of these subsets.

Metrics:

In this project, we will evaluate the performance of our predictive models using several metrics, namely accuracy score, recall score, and F1 score.

- *Accuracy score*: Accuracy measures the overall correctness of the model's predictions. It calculates the proportion of correctly classified instances (both churners and non-churners) over the total number of instances in the dataset. However, accuracy alone may not be sufficient in cases where the dataset is imbalanced, i.e., the number of churners and non-churners is significantly different.

- *Recall score*: Recall, also known as sensitivity or true positive rate, focuses on the model's ability to correctly identify churners. It calculates the proportion of actual churners that are correctly identified as churners by the model. A higher recall indicates that the model is effective in capturing churn cases, which is crucial to identify customers who are at risk of leaving.

- *F1 score*: The F1 score is the harmonic mean of precision and recall. It provides a balanced measure of both metrics and is particularly useful when the dataset is imbalanced. F1 score combines precision (the ability to correctly identify non-churners) and recall (the ability to correctly identify churners) into a single value. Maximizing the F1 score ensures a trade-off between correctly predicting both churners and non-churners, making it suitable for evaluating the overall performance of our models.

Eventually, the model reached an accuracy of 84.6% which is good for the company. In order to increase the performance and make it perfect, the company should give more historical data (the larger the dataset, the better the accuracy).

Real time test of our model

In order to assess the performance and accuracy of our machine learning model, we conducted a real-time test by selecting the first ten customers from our dataset. These customers have known churn status, which allows us to compare their actual churn values with the predictions made by our model. By evaluating the model's ability to correctly predict these churn values, we gain valuable insights into its effectiveness and suitability for deployment within our company.

The objective of this real-time test was twofold: first, to ascertain whether our model can accurately predict churn status for individual customers, and second, to evaluate the overall accuracy of our model's predictions based on our observations. By analyzing the results obtained from this test, we can make informed decisions regarding the deployment of our machine learning model in practical scenarios.

To conduct this evaluation, we carefully selected ten customers from our dataset, ensuring that we had access to their actual churn status. By comparing these known churn values with the churn predictions generated by our model, we were able to quantitatively assess its performance.

The ten customers were chosen to provide a representative sample from our dataset, allowing us to derive meaningful conclusions about the model's capabilities.

After executing the real-time test, we analyzed the results and found that our model accurately predicted 9 out of the 10 churn values. This impressive performance indicates a high level of accuracy and reliability in our model's predictions. The ability to correctly predict the churn status of customers is of paramount importance for our company, as it enables us to proactively identify and address potential customer attrition.

The outcome of this real-time test serves as compelling evidence for the effectiveness of our machine learning model. By achieving a 90% accuracy rate in predicting churn values, our model demonstrates its potential to significantly contribute to our company's decision-making processes and customer retention strategies. The reliable and precise predictions generated by our model equip us with valuable insights to take proactive measures aimed at mitigating churn risks and fostering long-term customer relationships.

Given the highly promising results of our real-time test, we have gained confidence in the accuracy and usability of our machine learning model. This assessment underscores the model's potential to be seamlessly integrated into our operational workflows, allowing us to harness its predictive capabilities for effective churn management. Leveraging the power of machine learning, our company can make data-driven decisions, allocate resources strategically, and optimize customer engagement initiatives.

Further investigations for company service improvement

1. Investigation into High Customer Service Calls:

- Further investigation should be conducted to analyze the characteristics and behaviors of customers who make multiple calls to customer service.

- Identify the root causes behind the need for numerous calls and explore potential issues or challenges faced by these customers.

2. Retention Efforts for International Plan Holders:

- Given the high churn rate of over 42% among international plan holders, it is crucial to focus on retention strategies for this customer segment.

- Conduct a comprehensive analysis of the factors contributing to churn among international plan holders, such as pricing, service quality, or competitive offerings.

3. Investigation into High Churn States:

- Perform an in-depth investigation of states with high churn rates to identify common trends and factors influencing customer attrition.

- Analyze market dynamics, competitive landscape, customer preferences, and regional demographics to understand the underlying causes of high churn.

4. Incentives for Customers with High Day Charges:

- Explore the reasons why customers with total day charges over $55 have a 100% churn rate and investigate potential areas of dissatisfaction.

- Design incentive programs to retain customers with high daily charges by offering additional value, perks, or benefits.

How to retain customers and make them avoid churn

1. **Improve Customer Service:**
    - Enhance customer service channels and response times to ensure prompt and efficient support.
    - Actively listen to customer feedback and address concerns to demonstrate responsiveness and care.
    - Provide personalized assistance and tailored solutions to meet individual customer needs.

2. **Offer Incentives for Loyalty:**
    - Implement loyalty programs that reward customers for their continued engagement and purchases.

- Provide exclusive offers, discounts, or rewards to long-term customers to encourage their loyalty.
- Regularly communicate with customers to highlight the benefits of staying with the company and the value they receive.

3. **Enhance Product and Service Quality:**
    - Continuously improve the quality of products and services based on customer feedback and market trends.
    - Conduct regular surveys and gather customer insights to identify areas for improvement.
    - Stay updated with industry advancements and provide innovative solutions that meet evolving customer demands.
4. **Personalize Customer Experience:**
    - Leverage customer data to personalize communication and offerings based on individual preferences.
    - Provide targeted recommendations and customized solutions to enhance customer satisfaction.
    - Anticipate customer needs and proactively offer relevant services or upgrades.
5. **Retention-focused Marketing:**
    - Implement targeted marketing campaigns to engage existing customers and reinforce their loyalty.
    - Use data analytics and segmentation techniques to identify at-risk customers and proactively address their concerns.
    - Communicate the unique value proposition of the company and highlight competitive advantages over rivals.
6. **Proactively Identify Churn Indicators:**
    - Utilize the churn prediction model to identify customers at risk of churning.
    - Focus resources and attention on customers who exhibit churn indicators, such as multiple customer service calls or international plan holders.
    - Implement proactive measures to address the specific needs and concerns of these customers and prevent them from churning.

References

Thomas J. F. & Guillaume L. & Nicolas h. (2022). Gradient Boosted Regression Trees. Retrieved: Jul 26th, 2023, from: *https://github.com/scikit-learn/scikit-learn/blob/364c77e047ca08a95862becf40a04fe9d4cd2c98/sklearn/ensemble/_gb.py*

Bengio Y. & David B. (2017). Churn in Telecom's dataset. Retrieved: Jul 26th, 2022, from: *https://www.kaggle.com/datasets/becksddf/churn-in-telecoms-dataset*

Matthieu K. & Dominic C. M.& Justin D. S. & Yves C. (2018). Exploratory Data Analysis, Chapter . Retrieved: Jul 26th, 2023, from: *https://www.researchgate.net/publication/308007227_Exploratory_Data_Analysis/link/57ff8fcc08ae32ca2f5d8022*

Shukla P. (2022). Top 10 Interview Questions on Gradient Boosting Algorithms. Retrieved: Jul 26th, 2023, from *: https://www.analyticsvidhya.com/blog/2022/11/top-10-interview-questions-on-gradient-boosting/*

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 29(5), 1189-1232. DOI: 10.1214/aos/1013203451.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). DOI: 10.1145/2939672.2939785.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.

Probst, P., Boulesteix, A. L., & Strobl, C. (2018). Tunability: Importance of Hyperparameters of Machine Learning Algorithms. Journal of Machine Learning Research, 18(87), 1-33.

Light, G. B. (2016). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems, 30, 3146-3154.

Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A. (2006). Ensemble Selection from Libraries of Models. Proceedings of the 21st International Conference on Machine Learning (ICML) (pp. 313-320). DOI: 10.1145/1143844.1143874.

Schapire, R. E. (1990). The Strength of Weak Learnability. Machine Learning, 5(2), 197-227. DOI: 10.1023/A:1022648809237.

Jang, S. H., Lee, S. H., & Lee, S. (2019). Early Stopping in Gradient Boosting Algorithm: A Study on Overfitting and Bias-Variance Trade-off. Applied Sciences, 9(17), 3493. DOI: 10.3390/app9173493.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive Logistic Regression: A Statistical View of Boosting. The Annals of Statistics, 28(2), 337-407. DOI: 10.1214/aos/1016218223.

Breiman, L. (1996). Bagging Predictors. Machine Learning, 24(2), 123-140. DOI: 10.1007/BF00058655.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques (4th ed.). Morgan Kaufmann.

Li, Y., & Wang, Y. (2020). A Survey on Gradient Boosting Decision Trees. arXiv preprint arXiv:2011.09601.

Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems, 30, 4765-4774.

Zhang, T. (2012). Ensemble Learning. In Encyclopedia of Machine Learning (pp. 314-317). Springer.

Bühlmann, P., & Hothorn, T. (2007). Boosting Algorithms: Regularization, Prediction and Model Fitting. Statistical Science, 22(4), 477-505. DOI: 10.1214/07-STS242.

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., & Feuston, B. P. (2003). Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. Journal of Chemical Information and Computer Sciences, 43(6), 1947-1958. DOI: 10.1021/ci034160g.

Tukey, J. W. (1977). Exploratory Data Analysis. Addison-Wesley.

Wickham, H., & Grolemund, G. (2017). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. O'Reilly Media.

VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. MIT Press.