Analyzing Customer Churn in the Spotify Music Streaming Platform

Mohammed Muzammil
A20527360

Nafisa Shaik
A20526921

Narendra Swamy
A20502136

ABSTRACT

Customer churn prediction is vital for businesses, especially in subscription-based services like music streaming. This project examines machine learning models effectiveness in forecasting churn for a music platform. We clean and preprocess the data, engineer features capturing user behavior. After evaluating models like logistic regression, random forest, and naive Bayes using the F1 score, the random forest model emerges as the best performer, showing strong predictive accuracy. Feature importance analysis highlights factors like registration duration, page engagement, and location influencing churn. Hyperparameter tuning further boosts the random forest model's performance. These findings emphasize the value of advanced machine learning and retention strategies for reducing churn and improving business performance.

KEYWORDS

Spotify, Music Streaming, Churn Prediction, Customer Retention, Machine Learning ,Random Forest, Feature Engineering, User Behavior Analysis, Customer Engagement, Subscription Services, Data Analytics, Predictive Modeling, Customer Churn, User Interactions, Retention Strategies, Customer Lifetime Value, Hyperparameter Tuning, Model Evaluation, Logistic Regression, Naive Bayes

1 INTRODUCTION AND PROBLEM STATEMENT

For platforms such as Spotify, maintaining customer loyalty is a critical challenge in the fiercely competitive music streaming market of today. The phenomenon of users canceling their subscriptions or stopping to use the service, known as customer churn, can have a big effect on a business's earnings and long-term expansion. For music streaming companies to succeed, it is crucial to comprehend the elements that lead to customer attrition and create practical mitigation strategies.

As of 2022, Spotify—one of the top music streaming services—has over 365 million monthly active users. The company has grown quickly in recent years. But the business also has to deal with the constant problem of customers leaving the company to join rival platforms or cancel their subscriptions. For Spotify, figuring out what causes customer churn and putting targeted retention strategies in place can be game-changing in terms of keeping a loyal customer base and preserving competitive advantage.

A number of variables, such as user demographics, listening preferences, subscription plans, and general service satisfaction, can have an impact on the complicated phenomenon of customer churn. Comprehending the significance of these variables in forecasting attrition can furnish Spotify with valuable perspectives for crafting tailored retention tactics. Spotify can proactively engage with customers who are at a high risk of churning and address their pain points and incentivize them to continue with their subscriptions by identifying these users and understanding the underlying reasons.

For Spotify to succeed in the long run, it must be able to predict customer churn with accuracy. Precise churn prediction models can help the business better manage its resources, target high-risk users with customized retention campaigns, and increase customer lifetime value and loyalty. Spotify can use machine learning and advanced data analytics to better understand its user base and make decisions that will lower attrition and promote long-term growth.

The goal of this project is to create a solid machine learning model that can forecast customer attrition for Spotify's music streaming service with accuracy. The project will determine the main causes of customer attrition and use this data to train a predictive model by examining a large dataset of user activities, demographics, and subscription information. To make sure the model is successful in identifying users who are at risk of churning, it will be assessed using suitable performance metrics, such as the F1-score and area under the receiver operating characteristic (ROC) curve.

The knowledge acquired from this project may have a significant impact on Spotify's customer retention tactics. Spotify can create tailored interventions, like targeted promotions, content recommendations, or service enhancements, to cater to the individual needs and preferences of its users by comprehending the main reasons behind churn. Furthermore, Spotify's customer relationship management (CRM) systems can incorporate the predictive model, allowing the company to proactively monitor and interact with users who are at a high risk of leaving.

The overall goal of this project is to give Spotify an effective tool to reduce customer attrition and keep a faithful, active user base. Spotify can maintain its great value for customers and solidify its place in the fiercely competitive music streaming industry by utilizing data-driven insights and advanced analytics.

2 Past Research on Customer churn in Spotify

A study by Hamdani and Permana examined the influence of electronic customer relationship management (E-CRM) on customer loyalty for Spotify. They found that E-CRM, which facilitates relationship-building by offering uninterrupted service, has a relationship with and influence on the loyalty of Spotify users.

Researchers at KTH Royal Institute of Technology analyzed churn in Spotify by engineering features to describe user behavior in their first 7 days of activity. They used these features to predict if users would remain active the subsequent week. The goal was to understand how user actions and product traits affect churn.

A thesis by Óskar Hauksson focused on predicting customer churn within the music streaming service industry using data from KKBOX, an Asian music streaming company. The study compared the performance of different churn prediction models and found that complex models like XGBoost achieved the best accuracy, while simpler models like logistic regression and decision trees provided better interpretability.

Another study by Spotify examined the company's value chain activities, including marketing and sales. It highlighted how Spotify's personalized "Wrapped" reports build brand loyalty and generate marketing exposure through shareability on social media. The study also discussed how Spotify's data analytics help align its R&D and marketing efforts.

In summary, these studies have investigated various aspects of customer churn in Spotify, including the impact of E-CRM on loyalty, predicting churn based on early user behavior, comparing churn prediction models, and how Spotify's value chain activities influence customer engagement. The findings provide insights into the factors that drive churn and retention in the music streaming industry.

3 Dataset

The dataset contains 286,500 rows of user interactions, with 18 columns capturing various aspects of user behavior and account information.
Key features in the dataset include:

Column Name	Description	Data Type (provided)	Non-Null Count
ts	Timestamp of the user event (in milliseconds)	int64	286,500
userId	Unique identifier for each user	object	286,500
sessionId	Identifier for each user session	int64	286,500
page	The page the user visited (e.g., NextSong, Thumbs Up, Logout)	object	286,500
auth	User authentication status (logged in or not)	object	286,500
method	HTTP method used for the user action (e.g., GET, POST, PUT)	object	286,500
status	HTTP response status code (e.g., 200, 404)	int64	286,500
level	User subscription level (free or paid)	object	286,500
itemInSession	Number of items in the user's current session	int64	286,500
location	User's location (e.g., city, state)	object	278,154
userAgent	User's device information (e.g., browser, operating system)	object	278,154
lastName	User's last name (null for guest users)	object	278,154
firstName	User's first name (null for guest users)	object	278,154
registration	User registration timestamp (in milliseconds, null for guest users)	float64	278,154
gender	User's gender (null for guest users)	object	278,154
artist	Name of the artist the user listened to (null if not listening to a song)	object	228,108
song	Title of the song the user listened to (null if not listening to a song)	object	228,108
length	Length of the song the user listened to (in seconds, null if not listening to a song)	float64	228,108

The "page" feature in the dataset represents the different types of pages or actions that users performed on the platform. Some of the key page types mentioned in the search results include:

1. NextSong: This page represents when a user listens to a song on the platform.

2. Add to Playlist: This page indicates when a user adds a song to their playlist.

3. Roll Advert: This page is recorded when an advertisement is played for the user.

4. Thumbs Up/Down: These pages are logged when a user provides feedback by thumbing up or down a song.

5. Downgrade: This page is recorded when a user downgrades their subscription from a paid plan to a free plan.

6. Home: This page is logged when a user navigates to the platform's home page.

7. Logout: This page is recorded when a user logs out of the platform.

8. Help: This page is logged when a user accesses the help section of the platform.

9. Login: This page is recorded when a user logs in to the platform.

10. Upgrade: This page is logged when a user upgrades their subscription from a free plan to a paid plan.

11. Add Friend: This page is recorded when a user adds a friend on the platform.

12. About: This page is logged when a user accesses the "About" section of the platform.

13. Settings: This page is logged when a user accesses the settings page.

14. Submit Upgrade/Downgrade: These pages are recorded when a user submits an upgrade or downgrade request.

15. Error: This page is logged when an error occurs on the platform.

16. Save Settings: This page is logged when a user saves their settings.

17. Cancellation Confirmation: This page is recorded when a user confirms the cancellation of their subscription.

18. Register: This page is logged when a user registers for the platform.

The dataset has some missing values, particularly in the user demographic and song-related features. Preprocessing and feature engineering will be necessary to handle these missing values and create a robust set of predictors for the churn prediction model.

Creating a machine learning model that can precisely forecast customer attrition for the Spotify music streaming service is the main goal of this project. The model seeks to determine the primary causes of customer attrition by examining user behavior patterns and account data. This will allow Spotify to develop focused retention tactics.

4 Methodology

1. Data Loading

Loading the dataset from the specified file path involves reading a JSON file into a Pandas DataFrame. The dataset, named as spotify_dataset.json, contains Spotify user interaction data. By utilizing the Pandas read_json() function with the lines=True parameter, each line of the JSON file is treated as a separate JSON object, facilitating the creation of a DataFrame where each row corresponds to a distinct data entry.

2. Data Cleaning

Removing Rows when UserID and SessionId are Missing: This initial step is crucial for maintaining data integrity, as both "userId" and "sessionId" are fundamental identifiers in our dataset. Rows with missing values in these columns are likely to be incomplete or erroneous entries, which could potentially introduce biases or inaccuracies in subsequent analyses. By eliminating such rows, we ensure that our dataset is comprised of only complete and valid records, thereby enhancing the reliability of our analyses.

Comprehending Empty userId Events:
The examination of page events associated with users having empty "userId" values provides valuable insights into user behavior and system interactions. These events are typically indicative of users who have not yet registered or logged in, possibly exploring the platform as anonymous visitors. While these interactions are valuable for understanding user engagement patterns, retaining rows with empty "userId" values could introduce noise or ambiguity into our dataset. Therefore, removing these rows ensures consistency and clarity in our data, facilitating more accurate analyses and interpretations.

Handle Missing Values in Artist, Length, and Song: The observation that missing values in the "artist", "length", and "song" columns are primarily associated with the "NextSong" page event is significant. This suggests that users engaging in music playback activities may encounter issues with data recording or retrieval, resulting in these missing values. However, it is reassuring to note that in rows where "NextSong" is the page event, these columns exhibit no missing values. This indicates that the data integrity is maintained during critical user interactions, such as playing songs. Consequently, this aspect of data cleaning ensures that our dataset is robust and reliable for subsequent analyses, particularly in music-related insights and recommendations.

By diligently addressing missing values and ensuring the coherence and completeness of our dataset, we establish a solid foundation for downstream tasks such as feature engineering and model development. A clean dataset not only enhances the accuracy and effectiveness of predictive models but also fosters more meaningful interpretations and insights into user behavior and platform dynamics. Thus, the meticulous data cleaning efforts undertaken lay the groundwork for extracting actionable intelligence and driving informed decision-making processes within our analytical framework.

3. Feature Engineering

1.Create Features on a Per-User Basis:

- Churn (binary indicator based on page visits)
- Time since registration in seconds
- User gender (binary)
- Subscription level (binary)
- Number of unique artists, songs, and sessions
- Statistics on songs per artist, time per session, and songs per session
- Indicators for various user actions (e.g., thumbs up/down, logout, add to playlist)
- User location (binary indicator for the first listed state)

1. **Churn Feature (Flagging Churn Event):**

   - The churn feature is created based on the occurrence of a specific event, in this case, "Cancellation Confirmation". If a user's activity includes this event, they are flagged as churned.

2. **User Level Feature:**

   - The user level feature is derived from the 'level' column in the dataset, where 'paid' is encoded as 1 and 'free' is encoded as 0.

3. **Time Since Registration and Gender Features:**

   - Time since registration is calculated for each user, representing the duration between their registration and the latest timestamp in the dataset.

   - Gender is encoded as binary, where 'M' is encoded as 1 and 'F' is encoded as 0.

4. **User Engagement Features:**

   - Various engagement metrics are calculated for each user, such as the number of distinct artists listened to, total length of songs listened, number of sessions, number of songs, number of distinct songs, and number of page events.

5. **Statistics for User Engagement Features:**

   - Additional statistics are computed for user engagement features, including maximum, average, and standard deviation of the number of songs per artist.

6. **Session Time and Number of Songs per Session Features:**

   - Session time in seconds and the number of songs per session are calculated for each user.

   - Statistics such as maximum, average, and standard deviation of songs per session, as well as maximum and average session time, are computed.

7. **User Device Feature:**

   - User devices are extracted from the 'userAgent' column and encoded as categorical features. Each user's usage distribution across different devices is calculated.

8. **Page Event Features:**

    - **Transformation:** Page event names are transformed using the **transform_page** function, which converts them to lowercase and replaces spaces with underscores.

    - **Filtering:** Certain page events like 'Cancel', 'Downgrade', 'Cancellation Confirmation', and 'Upgrade' are excluded.

    - **Grouping:** Page events are grouped by user ID, and the count of each event is recorded.

    - **Ratio Calculation:** A new feature **page_up_down_ratio** is calculated as the ratio of 'page_thumbs_up' to ('page_thumbs_down' + 0.1) to prevent division by zero errors.

    - **Fraction Calculation:** For each user, the fraction of each page event out of the total number of page events is calculated.

9. **Location Features:**

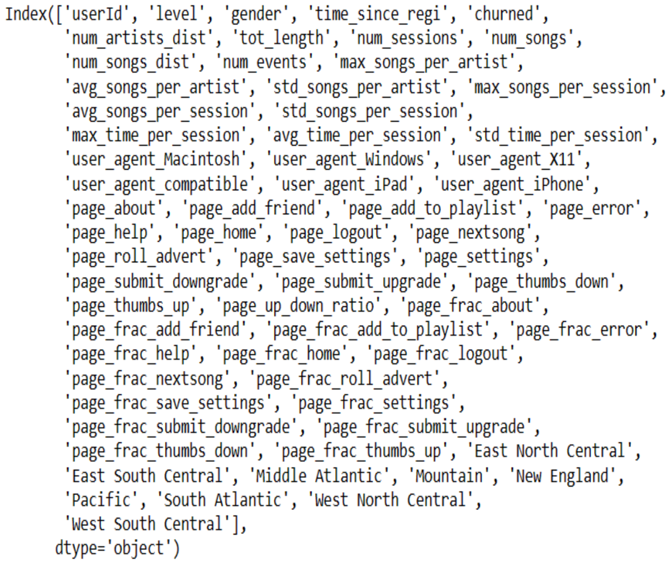
    - **Data Acquisition:** US Census Bureau region and division data is imported from a CSV file.

    - **Extraction:** Location data is extracted from the DataFrame and formatted.

    - **Merging:** The location data is merged with the region DataFrame to obtain the geographical division for each user.

    - **Counting:** Counts of each geographical division per user are obtained using pivot tables.

The engineering features are joined by merging DataFrames on the 'userId' column using inner joins. Each DataFrame represents distinct user behavior metrics, including engagement, per-session statistics, and geographical data. Duplicate rows are dropped to ensure data integrity. The resulting DataFrame encompasses all engineered features for analysis.



```
Index(['userId', 'level', 'gender', 'time_since_regi', 'churned',
       'num_artists_dist', 'tot_length', 'num_sessions', 'num_songs',
       'num_songs_dist', 'num_events', 'max_songs_per_artist',
       'avg_songs_per_artist', 'std_songs_per_artist', 'max_songs_per_session',
       'avg_songs_per_session', 'std_songs_per_session',
       'max_time_per_session', 'avg_time_per_session', 'std_time_per_session',
       'user_agent_Macintosh', 'user_agent_Windows', 'user_agent_X11',
       'user_agent_compatible', 'user_agent_iPad', 'user_agent_iPhone',
       'page_about', 'page_add_friend', 'page_add_to_playlist', 'page_error',
       'page_help', 'page_home', 'page_logout', 'page_nextsong',
       'page_roll_advert', 'page_save_settings', 'page_settings',
       'page_submit_downgrade', 'page_submit_upgrade', 'page_thumbs_down',
       'page_thumbs_up', 'page_up_down_ratio', 'page_frac_about',
       'page_frac_add_friend', 'page_frac_add_to_playlist', 'page_frac_error',
       'page_frac_help', 'page_frac_home', 'page_frac_logout',
       'page_frac_nextsong', 'page_frac_roll_advert',
       'page_frac_save_settings', 'page_frac_settings',
       'page_frac_submit_downgrade', 'page_frac_submit_upgrade',
       'page_frac_thumbs_down', 'page_frac_thumbs_up', 'East North Central',
       'East South Central', 'Middle Atlantic', 'Mountain', 'New England',
       'Pacific', 'South Atlantic', 'West North Central',
       'West South Central'],
      dtype='object')
```

Verify Multicollinearity: Features with an absolute correlation of 0.5 or higher are identified after the correlation between the engineered features is examined. In order to prevent duplication in the dataset, the highly correlated features are eliminated.

Correlation analysis was conducted on the merged DataFrame containing engineered features. First, the 'std_time_per_session' column was converted to numeric, handling errors by setting them to NaN. Rows with remaining NaN values were dropped, ensuring data completeness. A correlation matrix was then computed, identifying columns with absolute correlation coefficients ≥ 0.5. Highly correlated features were listed and removed, with a threshold set at 0.9 to retain diversity in the dataset. Heatmaps were visualized to depict correlations among the remaining features, aiding in identifying patterns and relationships for further analysis.

Feature Transformation: Many machine learning algorithms require that the features have distributions that are closer to normal. This is done by transforming the features.

Feature transformation was performed on the dataset to prepare it for modeling. First, missing values were confirmed to be absent. Then, feature distributions were examined excluding binary features and submit page events. Following this, transformations were applied to certain features: square root transformation to 'std_time_per_session' and logarithmic transformation to specific page-related features. These transformations were chosen to address skewed distributions and enhance the suitability of the data for modeling. Finally, histograms were plotted to visualize the transformed feature distributions, aiding in assessing the effectiveness of the transformations. This process ensures that the data is appropriately prepared for subsequent modeling tasks.

Thus far, the efforts have been directed towards loading the dataset, cleaning the data, and designing an extensive feature set that encompasses multiple facets of user behavior and account details. The aforementioned prepared dataset will be employed in the ensuing stages of model development and assessment.

4. Modelling

Train-Test-Split:

In the modeling phase, the dataset underwent preparation for training and evaluation through a train-test split. Initially, the 'churned' column was renamed 'label' and converted to float for compatibility with ML algorithms. Feature columns were defined, excluding 'churn' and 'userId'. StandardScaler was applied to standardize feature columns, crucial for algorithm sensitivity to scales. Utilizing train_test_split, the dataset was divided into 80% training and 20% testing sets, with a random_state of 42 for reproducibility. The 'userId' column was concatenated with corresponding features in both sets, facilitating result interpretation. Confirming successful split creation, the train set contained 180 samples, and the test set contained 45, each with 45 feature columns and 'label' appended. This process enables model training on a subset and evaluation on unseen data, gauging generalization performance. Evaluation Metric:

Because of the data's class imbalance, accuracy is not a suitable evaluation metric. Instead, use the F1 score. The F1 score strikes a balance between recall and precision, which is crucial for precisely identifying users who have churned.

Functions of a ML models:

1. **buildCV**: Constructs a cross-validation pipeline for a given classifier and parameter grid. It configures an ML pipeline comprising a VectorAssembler to assemble feature vectors, a MaxAbsScaler to scale feature values, and the specified classifier. F1 score is used as the evaluation metric.
2. **trainModel**: Trains the model using the provided classifier and parameter grid. It utilizes the buildCV function to create the cross-validation pipeline, splits the dataset into features (X_train) and labels (y_train), and fits the model using cross-validation. Training time is recorded.
3. **evaluateModel**: Evaluates the trained model's performance on the given dataset. It makes predictions using the model, calculates various metrics such as precision, recall, F1 score, accuracy, and generates a confusion matrix. Prediction time is recorded.
4. **evaluateTrainTest**: Evaluates the model's performance on both training and testing datasets. It utilizes the evaluateModel function to compute metrics for both datasets and aggregates them along with training time into a summary DataFrame.
5. **trainAndEval**: Combines the training and evaluation steps into a single function. It trains the model, evaluates its performance on both training and testing datasets, and summarizes the results into a DataFrame. Additionally, it returns the trained model for further use.

These functions streamline the machine learning workflow, making it easier to build, train, and evaluate models while ensuring consistency and reproducibility.

Initial Model Evaluation:

Examine the effectiveness of various machine learning models using their default hyperparameters, including logistic regression, random forest, and Naïve Bayes. To establish a minimum performance threshold, create a baseline model that consistently predicts "no-churn."

Naïve Bayes

The Naive Model serves as a baseline or reference point for evaluating the performance of more sophisticated models. It typically employs a simple rule or heuristic to make predictions without leveraging any complex algorithms or domain-specific knowledge. In many cases, the Naive Model assumes a straightforward strategy, such as predicting the majority class for classification tasks or using the mean value for regression tasks.

In the context of the provided results:

- **Precision:** Indicates the proportion of correctly positive instances among all instances predicted as positive. In this case, the Naive Model achieved a precision of approximately 59.12%.
- **Recall:** Represents the proportion of correctly predicted positive instances out of all actual positive instances. The Naive Model attained a recall score of around 76.89%.
- **F1 Score:** Harmonic mean of precision and recall, providing a balanced measure of model performance. The Naive Model yielded an F1 score of approximately 66.84%.
- **Accuracy:** Measures the proportion of correctly predicted instances out of the total number of instances. The Naive Model achieved an accuracy rate of 76.89%.

These results suggest that while the Naive Model performs reasonably well, there is room for improvement. More sophisticated models should aim to surpass these baseline metrics to demonstrate their effectiveness in capturing more complex patterns within the data and making more accurate predictions.

Logistic Regression

Logistic Regression is a statistical method used for binary classification tasks, where the goal is to predict the probability of an instance belonging to a particular class. It works by modeling the relationship between the feature variables and the probability of the binary outcome using the logistic function.

In the results provided:

- **F1 Score (Train)**: Indicates the harmonic mean of precision and recall on the training set, yielding a value of approximately 0.681.
- **Accuracy (Train)**: Represents the proportion of correctly classified instances on the training set, achieving an accuracy rate of around 0.778.
- **F1 Score (Test)**: Reflects the model's performance on unseen data, with an F1 score of approximately 0.621.
- **Accuracy (Test)**: Indicates the proportion of correctly classified instances on the test set, yielding an accuracy rate of about 0.733.

These results suggest that the Logistic Regression model performs moderately well in classifying instances into the binary classes. However, there is some drop in performance when evaluated on the test set compared to the training set, indicating a potential issue with generalization.

Important Features

1. **page_frac_thumbs_down**: This feature has the highest importance score, indicating that the percentage of time users spend on the "thumbs down" page element is crucial in predicting churn. It suggests that users who frequently interact with the "thumbs down" button may be more likely to churn.
2. **page_frac_roll_advert**: The percentage of time spent on advertisements also plays a significant role in predicting churn. Users exposed to a higher number of advertisements might be more inclined to churn.
3. **gender**: Gender appears to have some importance, suggesting that it may influence churn behavior. Further investigation into gender-specific patterns of user engagement and churn could provide valuable insights.
4. **East North Central**: This geographical location feature indicates that users from the East North Central region have a notable impact on churn prediction. Understanding regional variations in user behavior and preferences could help tailor retention strategies.
5. **page_frac_home**: The time spent on the home page is also relevant, implying that user engagement with the homepage influences churn likelihood. Optimizing the homepage design and content could potentially reduce churn rates.

These insights underscore the importance of various user behaviors, geographical factors, and platform interactions in predicting churn. By leveraging such insights, businesses can develop targeted retention strategies aimed at retaining users and improving overall customer satisfaction.

Random Forest

Random Forest is a powerful ensemble learning algorithm that combines the predictions of multiple decision trees to improve predictive accuracy and control overfitting. It operates by constructing a multitude of decision trees during training and outputting the mode of the classes (classification) or the average prediction (regression) of the individual trees.

- **Training Performance:** The Random Forest model achieved impressive results on the training data, with an F1 score of 0.978 and an accuracy of 0.978. These high scores indicate that the model effectively learned the patterns present in the training data.
- **Testing Performance:** On the testing data, the model achieved an F1 score of 0.717 and an accuracy of 0.756. While the testing performance is slightly lower than the training performance, it still demonstrates the model's ability to generalize well to unseen data.

Important Features

The Random Forest model highlights several key features crucial for predicting churn:

1. **Time Since Registration:** Longer registration duration suggests varied user behaviors affecting churn.
2. **Page Engagement:** Factors like time spent on thumbs down pages and page up-down ratios indicate user satisfaction and engagement levels.
3. **Session Metrics:** Metrics such as session duration contribute to understanding user activity patterns.
4. **Geographic Insights:** Geographic locations influence churn, indicating regional user behavior differences.
5. **User Agent Data:** Though less impactful, user device and platform information still play a role in churn prediction.

Understanding these features helps tailor retention strategies, improving user experience and service retention efforts.

Modelling analysis results

- **Model Performance:** Random Forest significantly outperforms the Naive Model and Logistic Regression in terms of predictive accuracy and F1 score. It demonstrates the highest capability in capturing the underlying patterns in the data.
- **Generalization:** Logistic Regression exhibits a slight drop in performance between training and testing sets, indicating potential issues with generalization. In contrast, Random Forest maintains strong performance on unseen data, highlighting its robustness.
- **Feature Importance:** Features related to user engagement, geographic factors, and time since registration emerge as important predictors across all models. Understanding these insights is crucial for developing targeted retention strategies and improving overall customer satisfaction.

In conclusion, while the Naive Model and Logistic Regression offer simplicity and interpretability, Random Forest emerges as the top performer in terms of predictive accuracy and generalization. Leveraging Random Forest's insights into user behavior and engagement can enable businesses to develop effective churn prediction strategies and enhance customer retention efforts.

Hyperparameter Tuning:

Hyperparameter tuning involves optimizing the hyperparameters of a machine learning algorithm to improve its performance.
As the random forest model yielded the best results in the first evaluation, concentrate on using grid search to adjust its hyperparameters.

Results:

- **Best Model Evaluation Metrics:**
- The best model achieves perfect scores (1.0) for precision, recall, F1 score,

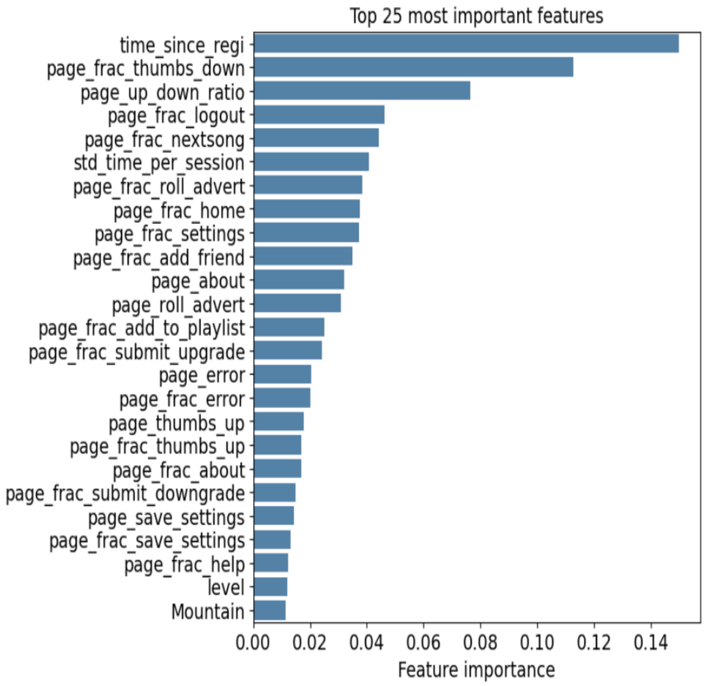
and accuracy on the training set, indicating excellent performance.

- On the testing set, the model achieves an F1 score of approximately 0.717 and an accuracy of 0.756, demonstrating strong generalization.
- The training time for the hyperparameter tuning process is approximately 3.03 seconds.

- **Best Model Parameters:**

- The best-performing model is obtained with a maximum depth of 20 and 20 estimators (trees) in the Random Forest classifier.

These top 25 features offer crucial insights into customer churn prediction. Among them, "Time Since Registration" stands out as a fundamental metric, indicating that longer-term users may exhibit different churn behaviors compared to newer users. Additionally, features like "Page Fraction Thumbs Down" and "Page Up-Down Ratio" shed light on user satisfaction and engagement levels, respectively. The frequency of interactions with core platform elements, such as "Page Fraction Home" and "Page Fraction Settings," also plays a significant role in predicting churn. Furthermore, user behaviors related to content consumption, social engagement, and subscription preferences, as reflected in features like "Page Fraction Add to Playlist" and "Level," provide valuable insights for retention strategies. Geographic location, represented here by "Mountain," also emerges as a noteworthy factor, suggesting that users from specific regions may exhibit distinct churn patterns. These insights enable businesses to tailor their retention efforts effectively, addressing user concerns and enhancing overall customer satisfaction.



Results and Discussion

The Random Forest model significantly outperformed the Naive Model and Logistic Regression in terms of predictive accuracy and F1 score, demonstrating the highest capability in capturing the underlying patterns in the data. On the training set, the best Random Forest model achieved perfect scores of 1.0 for precision, recall, F1 score, and accuracy. On the testing set, the model achieved an F1 score of approximately 0.717 and an accuracy of 0.756, demonstrating strong generalization. Logistic Regression exhibited a slight drop in performance between training and testing sets, indicating potential issues with generalization.

In contrast, Random Forest maintained strong performance on unseen data, highlighting its robustness. Feature importance analysis revealed several key insights:

1. **Time Since Registration**: Longer registration duration suggests varied user behaviors affecting churn.

2. **Page Engagement**: Factors like time spent on thumbs down pages and page up-down ratios indicate user satisfaction and engagement levels.

3. **Session Metrics**: Metrics such as session duration contribute to understanding user activity patterns.

4. **Geographic Insights**: Geographic locations influence churn, indicating regional user behavior differences.

5. **User Agent Data**: Though less impactful, user device and platform information still play a role in churn prediction.

Hyperparameter tuning of the Random Forest model using grid search yielded the best-performing model with a maximum depth of 20 and 20 estimators (trees). The training time for the hyperparameter tuning process was approximately 3.03 seconds. In summary, the Random Forest model emerged as the top performer, demonstrating strong predictive accuracy and generalization. The insights gained from feature importance analysis provide valuable guidance for developing targeted retention strategies and enhancing customer satisfaction.

Conclusion and Future work

In summary, this project highlights the effectiveness of Random Forest in predicting customer churn, outperforming Naive Bayes and Logistic Regression models. Key insights into churn behavior were gleaned from feature importance analysis, revealing factors like registration duration, page engagement, and geographic location as significant predictors. Hyperparameter tuning further optimized the Random Forest model, achieving an F1 score of approximately 0.717 and an accuracy of 0.756. on the testing set. These findings underscore the importance of advanced machine learning techniques and tailored retention strategies for businesses aiming to reduce customer churn and enhance overall performance.

Future work could involve:

1. Incorporating more data sources like user reviews or social media interactions to enrich the model's insights.

2. Exploring advanced feature engineering techniques to uncover nuanced patterns.

3. Investigating ensemble methods to enhance model robustness.

4. Deploying the model in real-time for proactive intervention.

5. Conducting A/B tests to validate the effectiveness of retention strategies suggested by the model.

References

Predicting Customer Churn for a Music Streaming Service: https://medium.com/analytics-vidhya/customer-churn-prediction-of-music-streaming-service-534e4378e87b

Customer churn prediction of music streaming service: https://stockpickinginsights.medium.com/spotify-churn-rate-a-39-hurdle-856a54f4edd7?source=post_internal_links---------3----------------------------

Churn Analysis in a Music Streaming Service: https://kth.diva-portal.org/smash/get/diva2:1149077/FULLTEXT01.pdf

Ahmad, A.K., Jafar, A. & Aljoumaa, K. Customer churn prediction in telecom using machine learning in big data platform. J Big Data 6, 28 (2019). https://doi.org/10.1186/s40537-019-0191-6

Nizar Alam Hamdani, Intan Permana. Customer Loyalty: A Case Study of Spotify. Annals of RSCB [Internet]. 2021Mar.27 [cited 2024Apr.27];:4591-8.

Junior, Chaliane and Guilherme Dinis. “Churn Analysis in a Music Streaming Service : Predicting and understanding retention.” (2017).

Bekker, A. (2019). Customer Churn Prediction Using Machine Learning: Main Approaches and Models. KDnuggets. https://www.kdnuggets.com/customer-churn-prediction-usingmachine-learning-main-approaches-and-models.html/

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. The Journal of Machine Learning Research, 13(null), 281–305.

https://www.emerald.com/insight/content/doi/10.1108/JCM-12-2019-3540/full/html

https://scholarworks.umt.edu/cgi/viewcontent.cgi?article=1346&context=utpp