Customer Churn Prediction with PySpark

Project notebook: Customer Churn Prediction Notebook

Group 23:

Daniel Joseph Eaton
Pranav Addipalli
Prateek Kumar Kumbar
Vinit Horakeri

Table of Contents

Executive Summary

The purpose of this project is to analyze web log data of music streaming services to predict customer churn using Pyspark and SparkML algorithms. In this streaming service, users can either listen to music for free or buy a subscription. Customer churn prevention is a hot and challenging problem in almost every product and service company. Predicting customer churn is a challenging and common problem that data scientists encounter these days. The ability to predict that a particular customer is at a high risk of churning, while there is still time to do something about it, represents a huge additional potential revenue source for every customer-facing business.If companies were able to utilize customer-usage data to find unique trends and accurately map them to indicate which customers may churn, it would be possible to incentivize customers to remain using their services giving them a loyal customer base which is key for a company's growth.

In order to identify users who are likely to churn, it's important to perform an exploratory analysis to glean insights from the data set and identify key variables of interest. The next process is to experiment with different model algorithms and then select the best model based on key evaluation metrics such as F1 Score and accuracy using Spark ML Library.

The expectation is that some of these features will reveal a substantial difference between customers that churn versus those that don't. This information is used to create useful features for a classification model for churn. We will experiment using models such as Logistic regression, random forest, gradient boosting, and decision trees to evaluate the problem.

The classification model is evaluated using standard metrics for binary output data - accuracy and F1-score. F1-score is given greater importance from an interpretation perspective due to the imbalanced nature of the output data Accuracy only works well when the dataset classes are balanced

The data we have is that of user events. Every interaction of every user with the application is given to us. This means every time a user goes to the Home page, listens to a song, thumbs up a song, etc. we have an event in the data corresponding to the same.

Introduction

1.1 Background

In practically every product and service organization, preventing customer churn is a hot and difficult issue. Companies would be able to retain a loyal client base, which is essential for business growth if they could use consumer usage data to identify distinctive trends and precisely map them to identify which customers may leave. Churn analysis frequently identifies trends that point to typical reasons why consumers quit you, like price sensitivity or poor product adoption. Additionally, it shows how users interact with your product over the course of its existence. These insights can help you enhance what at-risk consumers don't like while maximizing what devoted customers currently enjoy.

The ability to predict attrition rate is crucial because it enables music streaming companies to pinpoint problem areas in their customer service and make necessary improvements. Therefore, churn prediction aids in preventing people from canceling their membership. The likelihood of selling to a current customer is between 60 and 70 percent, claims ClickZ. However, only 20% to 25% of new prospects are likely to be sold to. Therefore, it is crucial to keep the current client.

This project covers the creation of a machine learning solution that will be able to predict customer churn. This solution will be realized with Apache Spark. We load large datasets into Spark and manipulate them using Spark Dataframes to engineer relevant features for predicting customer churn, then use the machine learning APIs within Spark ML to build and tune models and score the models using appropriate evaluation metrics.

1.2 Method Overview

Data: The data we have from the music streaming company is that of user events. Every interaction of every user with the application is given to us. This means every time a user goes to the Home page, listens to a song, thumbs up a song, etc. we have an event in the data corresponding to the same. This data is collected from Kaggle. We will use the subset for data exploration, feature engineering, and model selection on the local machine. It is more time and computationally-efficient to do all the modeling work in a small dataset than working on a full dataset. There are 18 features in both the full dataset and the subset of the dataset. The same number of values are missing from the columns firstName, gender, last name, registration, location, userAgent, and userId. Since we are unable to identify their userId, we will remove all of these null values. Since some users may simply log into the app without listening to any songs, we fill in any missing values in the artist, length, and song columns with None or 0. Before running EDA on the dataset, we define the churn label. Users who verified their subscription cancellation fall under the definition of the churn label.

Models: Due to the scope of the project and lack of computational resources, the analysis is not intended to be exhaustive, we only applied two classification machine learning models - Logistic Regression and Random Forest.

Tools: The programming is done in PySpark. Spark Dataframes, Pandas, NumPy Python libraries are utilized for EDA. SparkML is used for machine learning modeling. For data analysis and visualization, we use Numpy, Pandas, matplotlib, and seaborn.

2. Data Exploration

This dataset contains user-level web log data of a music streaming company. It includes 286500 rows and 18 columns, and there is no customer churn column (which will be created) The data dictionary is available in Appendix A.

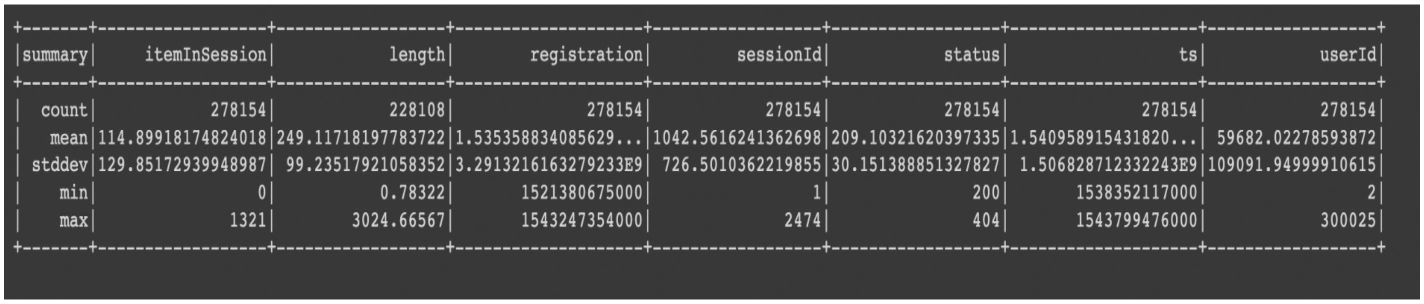
The dataset contains different columns such as artist, auth, firstName, gender, itemInSession, lastName, Length, Level, Location, Method, Page, Registration, sessionId, Song, Status, Ts, userAgent, userId.There are 11 Categorical variables and 7 numerical variables. A total of 10 columns have missing values out of which 7 columns have 8346 missing values each and 3 columns have 58392 missing values each.

The main findings from the exploratory analysis are as follows:

- 83% of paid customers are not churned compared to free tier customers and the female churn rate is less compared to male.
- The top features according to the Random Forest model are "Thumb_up, Thumb_down, and artist_code"

2.1 Numerical Variables Analysis

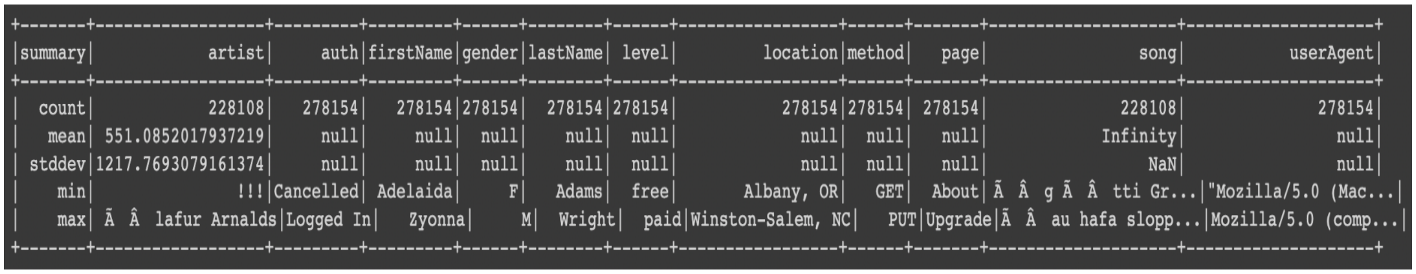
A summary of all the numeric variables is shown below. The mean length of songs listened to by users is 249.11 minutes with a standard deviation of 99.23. From the itenInSession column, we can infer that users stay for a longer time on the platform with a mean session time of 114.89.



```
+-------+------------------+------------------+--------------------+------------------+------------------+--------------------+------------------+
|summary|     itemInSession|            length|        registration|         sessionId|            status|                  ts|            userId|
+-------+------------------+------------------+--------------------+------------------+------------------+--------------------+------------------+
|  count|            278154|            228108|              278154|            278154|            278154|              278154|            278154|
|   mean|114.89918174824018|249.11718197783722|1.535358834085629...|1042.5616241362698|209.10321620397335|1.540958915431820...| 59682.02278593872|
| stddev|129.85172939948987| 99.23517921058352|3.2913216163279233E9| 726.5010362219855|30.151388851327827| 1.506828712332243E9|109091.94999910615|
|    min|                 0|           0.78322|       1521380675000|                 1|               200|       1538352117000|                 2|
|    max|              1321|        3024.66567|       1543247354000|              2474|               404|       1543799476000|            300025|
+-------+------------------+------------------+--------------------+------------------+------------------+--------------------+------------------+
```

2.2 Categorical Variables Analysis

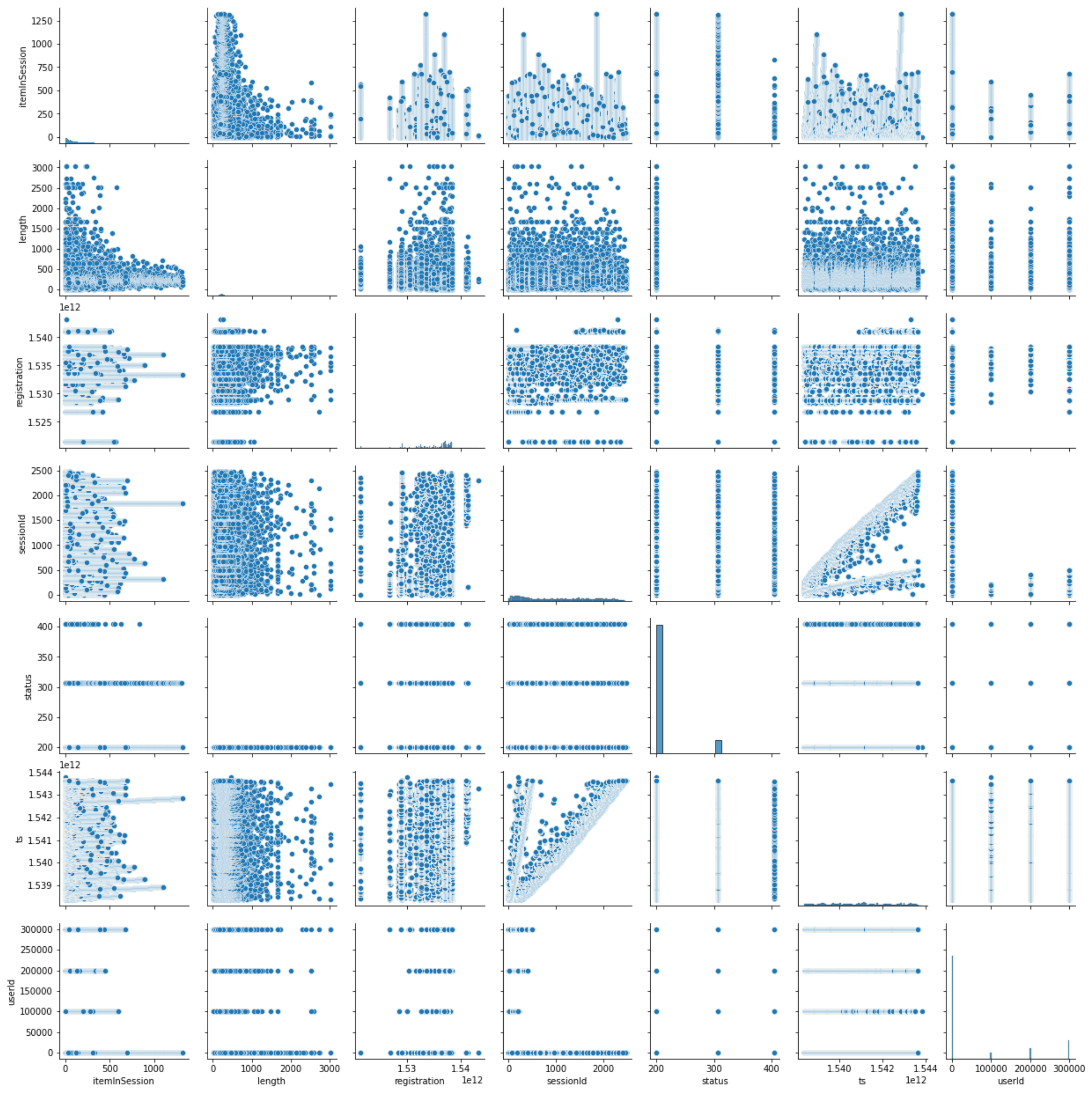
The below summary shows the description of categorical variables. The artist column summary tells us that there are 228108 artists for the same number of songs. The user's activity can be gauged from the page column with the least number of users visiting the about section of the application of music streaming service.



```
+-------+------------------+---------+---------+------+--------+------+-----------------+------+-------+--------------------+--------------------+
|summary|            artist|     auth|firstName|gender|lastName| level|         location|method|   page|                song|           userAgent|
+-------+------------------+---------+---------+------+--------+------+-----------------+------+-------+--------------------+--------------------+
|  count|            228108|   278154|   278154|278154|  278154|278154|           278154|278154| 278154|              228108|              278154|
|   mean| 551.0852017937219|     null|     null|  null|    null|  null|             null|  null|   null|            Infinity|                null|
| stddev|1217.7693079161374|     null|     null|  null|    null|  null|             null|  null|   null|                 NaN|                null|
|    min|               !!!|Cancelled| Adelaida|     F|   Adams|  free|       Albany, OR|   GET|  About|Ã  Â  g Ã  Â  tti Gr...|"Mozilla/5.0 (Mac...|
|    max| Ã  Â  lafur Arnalds|Logged In|   Zyonna|     M|  Wright|  paid|Winston-Salem, NC|   PUT|Upgrade|Ã  Â  au hafa slopp...|Mozilla/5.0 (comp...|
+-------+------------------+---------+---------+------+--------+------+-----------------+------+-------+--------------------+--------------------+
```

2.3 Pair Plot of Numeric Variables

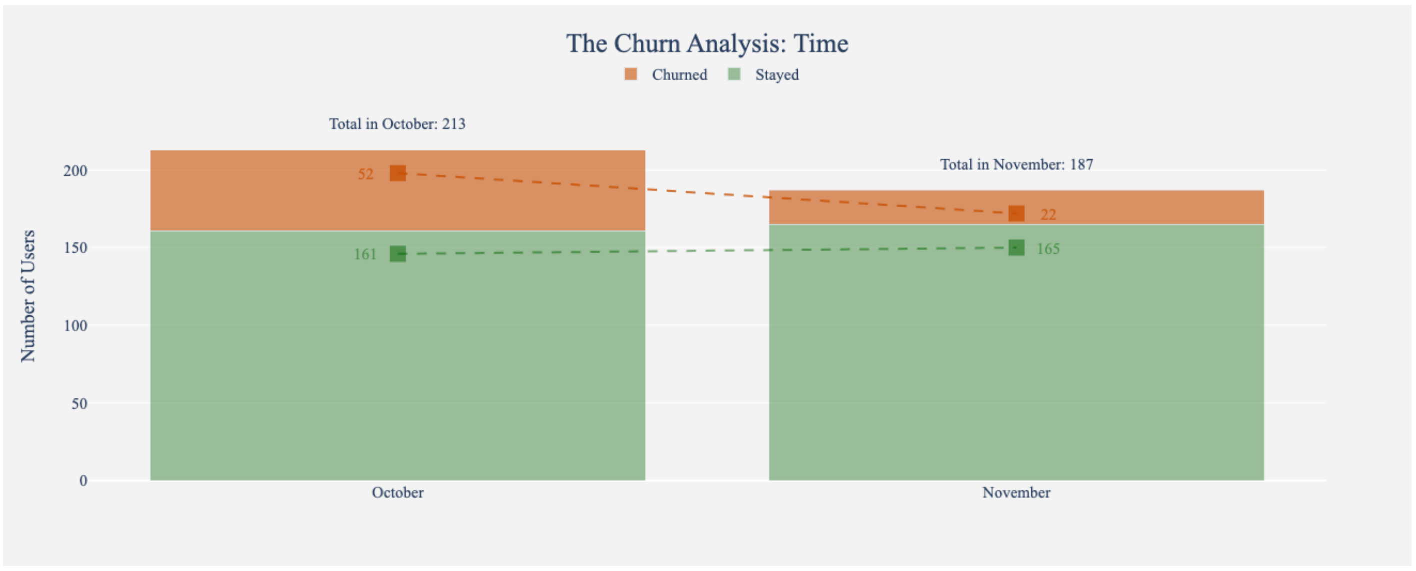
Pair Plots are a really simple way to visualize relationships between each variable. It produces a matrix of relationships between each variable to determine the relationship between variables. The following pair plot shows the relationship between all the numeric variables. There is a positive correlation between session and timestamp variables which is as expected. Other than that, we don't see any significant relationships among variables.



itemInSession
length
registration
sessionId
status
ts
userId
1e12

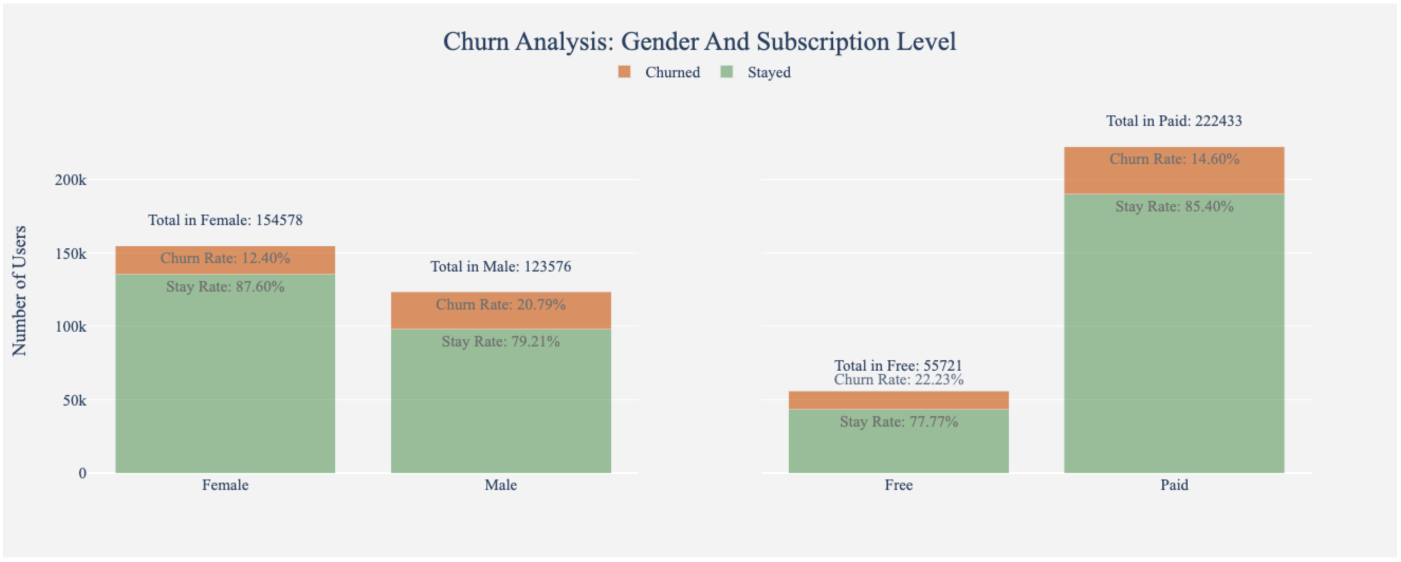
2.4 Churn Analysis: Time

The plot below shows the churn rate per month. The dataset contains the October and November months' activities of users. A total of 52 users churned out in the month of October compared to 22 in November showing a decreasing churn rate month on month.



2.5 Churn Analysis: Gender and Subscription Level

Gender and Subscription level churn analysis helps us in determining the churn rates across two different units of analysis i.e., Gender and Subscription. From the stacked plot shown below, we can say that the Female churn rate is less compared to males and the users with paid tiers are more likely to stay compared to free tiers which is as expected.

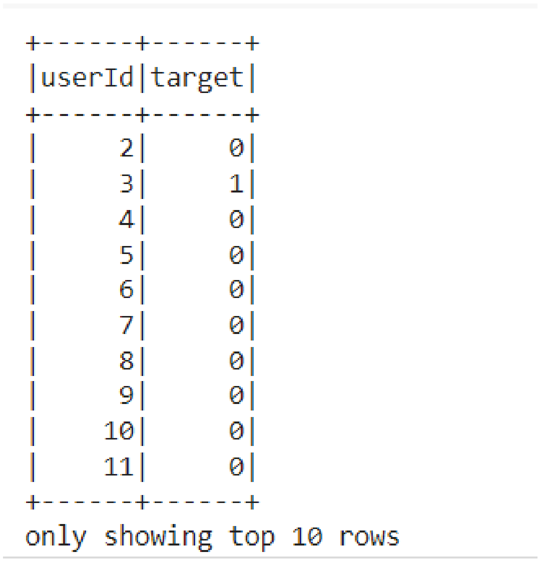


3. Feature Engineering

The dataset contains 18 columns and there is no churn column in it. Firstly, we created the Target variable (churn) and then the independent variables.

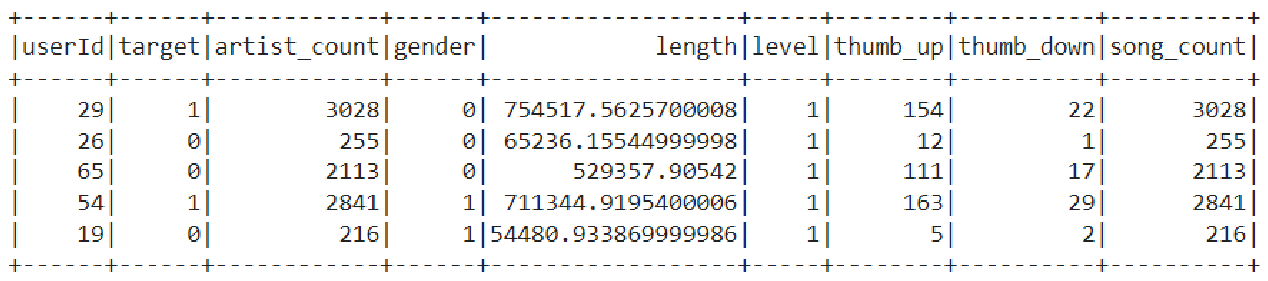
3.1 Target variable creation

The dataset has Page columns which contain the user's activity log, Page has values like 'Next Song', 'Play', 'Pause', and so on. If Page = 'Cancellation Confirmation' then target = 1 else the target will be 0.


```
+------+------+
|userId|target|
+------+------+
|     2|     0|
|     3|     1|
|     4|     0|
|     5|     0|
|     6|     0|
|     7|     0|
|     8|     0|
|     9|     0|
|    10|     0|
|    11|     0|
+------+------+
only showing top 10 rows
```

3.2 Other variables creation

- artist_count - total number of artists per userId
- Gender - user's gender
- length - total length of songs listened by the user in seconds
- thumb_up - total number of upvotes given by a user
- thumb_down - total number of downvotes given by a user
- level - level of user subscription (free, paid)
- songs_count - the number of songs played by a user



```
+------+------+------------+------+-----------------+-----+--------+----------+----------+
|userId|target|artist_count|gender|           length|level|thumb_up|thumb_down|song_count|
+------+------+------------+------+-----------------+-----+--------+----------+----------+
|    29|     1|        3028|     0| 754517.5625700008|    1|     154|        22|      3028|
|    26|     0|         255|     0| 65236.15544999998|    1|      12|         1|       255|
|    65|     0|        2113|     0|      529357.90542|    1|     111|        17|      2113|
|    54|     1|        2841|     1| 711344.9195400006|    1|     163|        29|      2841|
|    19|     0|         216|     1|54480.933869999986|    1|       5|         2|       216|
+------+------+------------+------+-----------------+-----+--------+----------+----------+
```

4. Modeling

4.1 Modeling Preparation

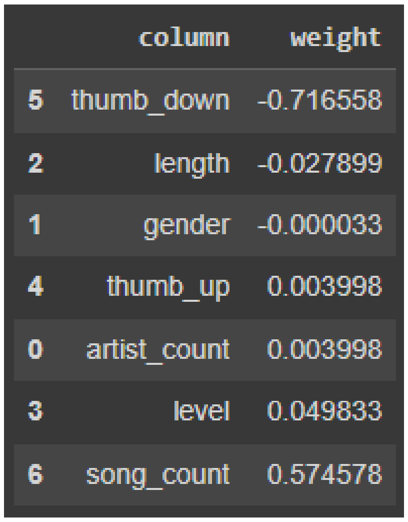
After data collection, data understanding, and data preparation, we next moved into modeling. The explanatory variables for modeling were: "artist_count", "length", "thumb_up", "thumb_down", "song_count", "gender", and "level." The target variable for modeling was "Target", which was a binary variable. The data was randomly split, with 60% of the data encompassing the training set, 30% of data encompassing the test set, and 10% of data encompassing the validation set.

4.2 Logistic Regression

After choosing the explanatory variables, target variable, and splitting the data set into training, test, and validation sets, we next moved into our first classification model: logistic regression.

We initially fit our logistic model using the training data set. After fitting the model with the training set, we next moved into evaluating our model. To evaluate the model, we used the testing set to see if model refinement was necessary. Evaluating the model with the test set provided an F1 score of 0.55. An F1 score of 0.55 reflects that the model is performing poorly and will need to be adjusted later.

4.3 Logistic Model Inference



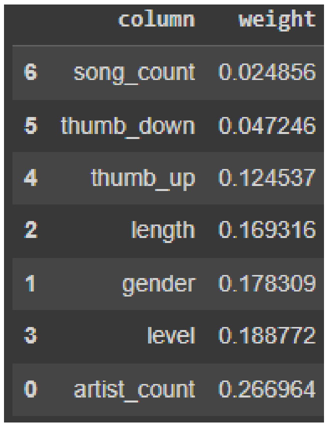
	column	weight
5	thumb_down	-0.716558
2	length	-0.027899
1	gender	-0.000033
4	thumb_up	0.003998
0	artist_count	0.003998
3	level	0.049833
6	song_count	0.574578

The image above displays the explanatory variables and their respective coefficient from the logistic regression model. With this table, one can determine the size and direction of the relationship between an explanatory variable and the target variable.

4.4 Random Forest

After training and testing our logistic model, we next moved into developing a Random Forest model. We first fit the model with our training data, and then evaluated our model using the test data. Using the test data to evaluate our model provided an F1 score of 0.62. With the F1 score of the Random Forest model being greater than the F1 score of the logistic regression model, we can conclude that the Random Forest model is currently outperforming the logistic regression model. However, the F1 score of the Random Forest model is still low, which will hopefully be increased through hyper parameter tuning.

4.5 Random Forest Inference



	column	weight
6	song_count	0.024856
5	thumb_down	0.047246
4	thumb_up	0.124537
2	length	0.169316
1	gender	0.178309
3	level	0.188772
0	artist_count	0.266964

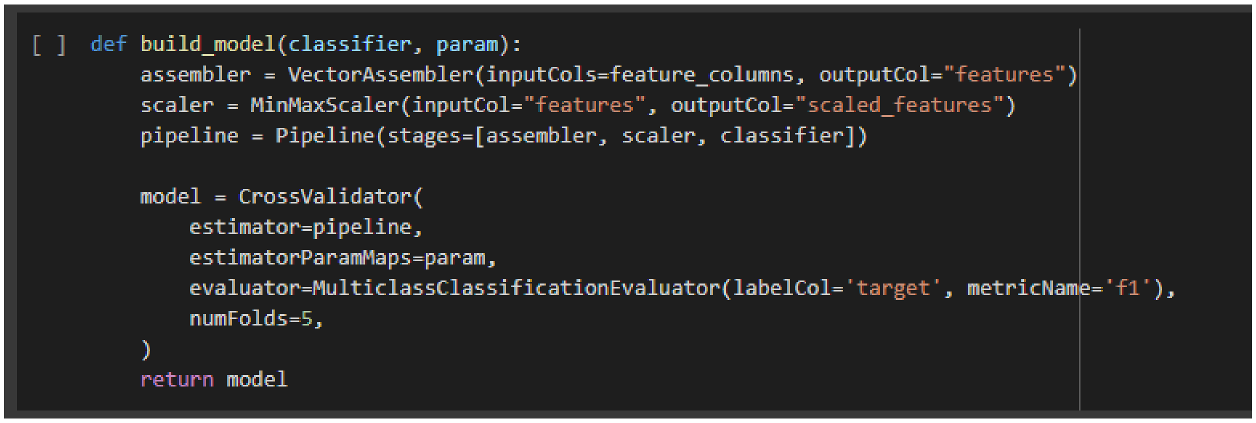
The image above displays the explanatory variables and their respective feature importance from the Random Forest model. This table is useful as it communicates which variables are most influential in the predictive model.

5. Hyperparameter Tuning

After initial modeling and prediction in the previous section, the f1-score or ROC-AUC scores for Random Forest and Logistic Regression models were calculated as 0.62 and 0.52 respectively. These scores are considerably low for
To improve f1-score we need to apply a few tuning methodologies like cross validation and grid search.

5.1 Cross-Validation

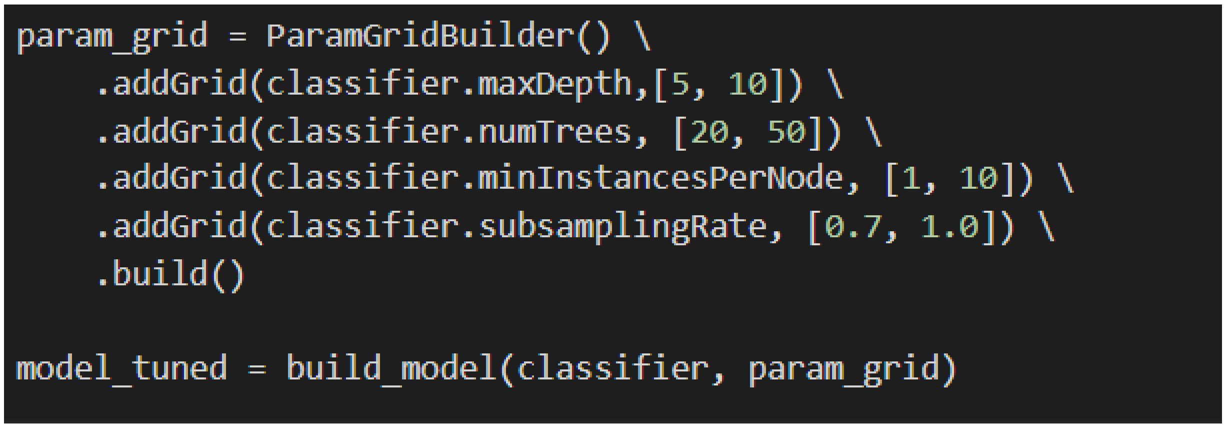
Cross-Validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The evaluation metrics show how well a model is performing. This procedure uses a single parameter called ‘k’ which divides a given sample data into k number of splits. Hence the name k-fold cross-validation.
For this project, we have used a 5-fold cross-validation method meaning, the sampled data will be split into 5 groups of data to be trained and fitted.


```
[ ]  def build_model(classifier, param):
         assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")
         scaler = MinMaxScaler(inputCol="features", outputCol="scaled_features")
         pipeline = Pipeline(stages=[assembler, scaler, classifier])

         model = CrossValidator(
             estimator=pipeline,
             estimatorParamMaps=param,
             evaluator=MulticlassClassificationEvaluator(labelCol='target', metricName='f1'),
             numFolds=5,
         )
         return model
```

5.2 Grid Search

Grid Search allows us to pass our specific estimator, our grid of parameters, and our chosen number of cross validation folds. Estimator allows us to select the specific model we're trying to run, in this case Random Forest. Grid parameters allow us to pass a grid of parameters we want to search.


```
param_grid = ParamGridBuilder() \
    .addGrid(classifier.maxDepth,[5, 10]) \
    .addGrid(classifier.numTrees, [20, 50]) \
    .addGrid(classifier.minInstancesPerNode, [1, 10]) \
    .addGrid(classifier.subsamplingRate, [0.7, 1.0]) \
    .build()

model_tuned = build_model(classifier, param_grid)
```

5.3 Evaluation Metrics

After building the cross-validation and grid search into the model, we train and fit the model and get the best model with highest f1-scores among all the models. Below are the improved f1-scores after implementing cross-validation with grid search.

Model	F1 score Before Tuning	F1 score After Tuning
Linear Regression	0.55	0.69
Random Forest	0.62	0.72

6. Conclusions

- During EDA, we explored data for insights and unwanted values. To analyze churn rate we visualized churn based on Time, Gender and Subscription levels.
- For inferences, user-data was analyzed to understand how different variables were behaving.
- Feature Engineering was performed on multiple features to create the best target variable for modeling.
- Built Linear Regression and Random Forest models to predict churn rate based on the features developed during feature engineering. To evaluate the model, f1 scores were calculated for both models.
- To optimize model performance, we built a cross-validation and grid search and tuned both models. To evaluate accuracy we calculated the f1 scores again and concluded that hyperparameter tuning improved the model accuracy.

Appendix

Data Dictionary:

artist: Artist name
auth: User authentication status
firstName: User first name
gender: Gender
itemInSession: Item count in a session
lastName: User last name
length: Length of song
level: User plan
location: User's location
method: HTTP method
page: Page name
registration: Registration timestamp
sessionId: Session ID
song: Song
status: HTTP status
ts: Event timestamp
userAgent: User's browser agent
userId: User ID (ex. 30)