Sparkify

Data Scientist Capstone Project

> *INTRODUCTION*

Sparkify is an audio listening platform in which we try to estimate whether users are likely to stay based on their activity logs. The dataset is mini json file ‘mini_sparkify_event_data.json’(128MB) containing 2,86,500 rows and 18 columns.

> *PROBLEM STATEMENT*

We have access to a JSON log of all actions performed by *Sparkify* users during a period of two months(October & November); our objective is to understand what behaviors can allow us to predict whether users will "*churn*" (i.e. unsubscribe from the service).

In order to achieve this, using 'Spark' framework we will extract the most relevant features from the log, train a machine learning classifier , fine tune it's parameters and with the help of accuracy and F1-score, determine the winning model.

METRICS

Accuracy of the various models, tuning parameters as necessary.

Since the churned users are a fairly small subset, I am using F1 score as the metric to optimize both precision and recall.

DATA PREPROCESSING AND EXPLORATION

Load the json file using spark.read.json.

Our dataset contains the following columns: artist, auth, firstName, gender, itemInSession, lastName, length, level, location, method, page, registration, sessionId, song, status, ts, userAgent, userId.

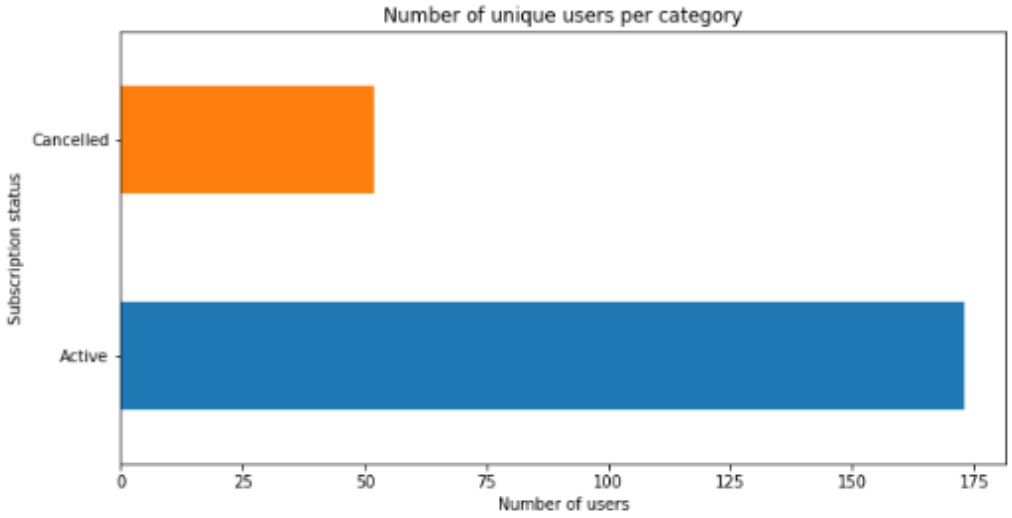
Remove the rows with empty userId & sessionId srrives at the following conclusion: When we are dealing with the rows of empty userId , he/she is a 'Guest' and trying to 'Register' or 'Submit Registration'. When we are dealing with the rows of empty sessionId , he/she has 'Logged Out' and trying to 'Login'. As we removed both the rows, the following rows associated with those values are removed.

Create a column `Churn` to use as the label for the model using the `Cancellation Confirmation` events on the page column .

We have to explore the behaviour of users who stayed v/s who churned based on level(paid/free), gender(female/male),page, ts.

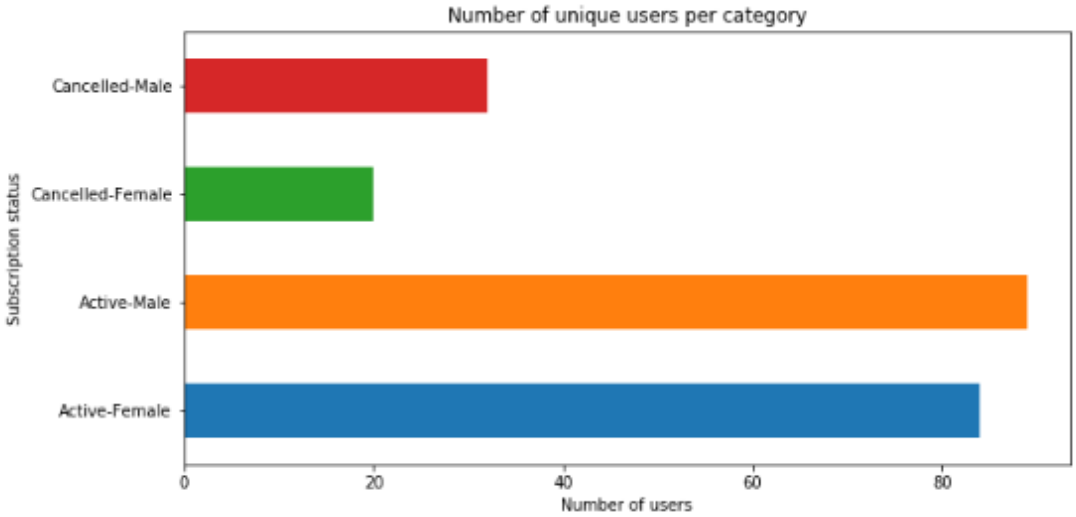
DATA VISUALISATION

Count of users who stayed v/s who cancelled



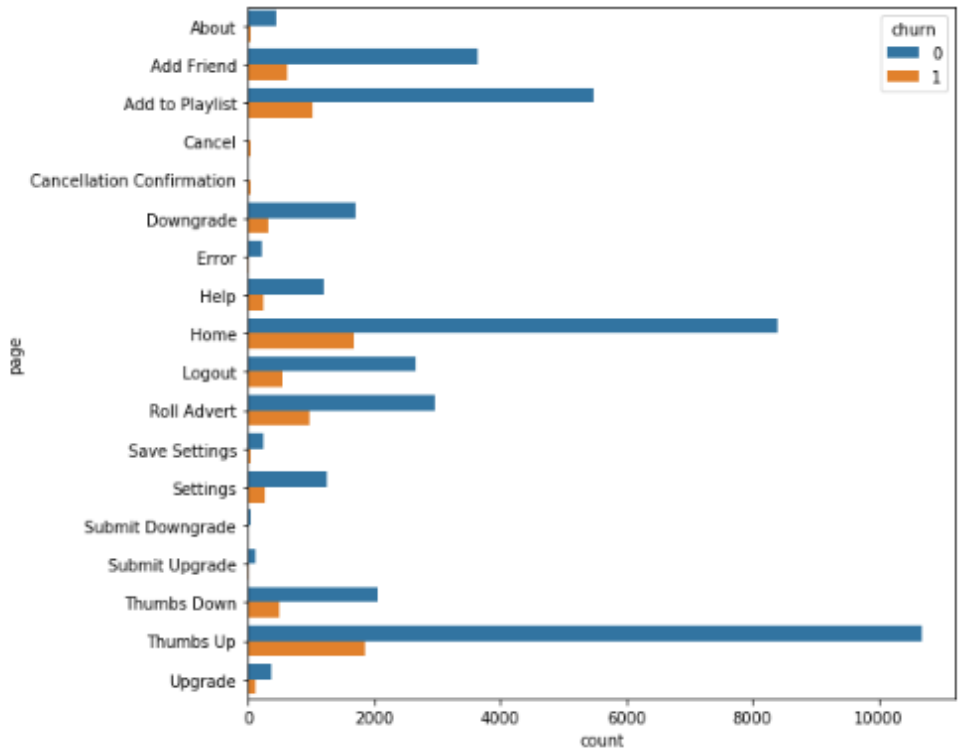
Conclusion: The number of unique users are more active than cancelled users.

2. Count of users based on Subscription status and gender



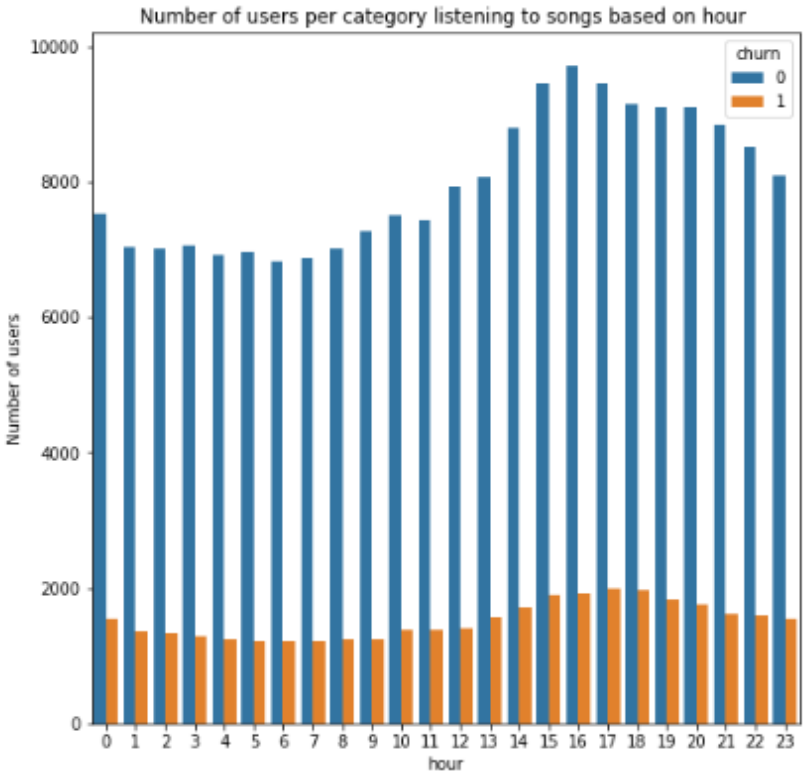
Conclusion: Males are slightly predominant than females in both active and cancelled subscriptions.

3. Count of users based on page requests labelled by their subscription status



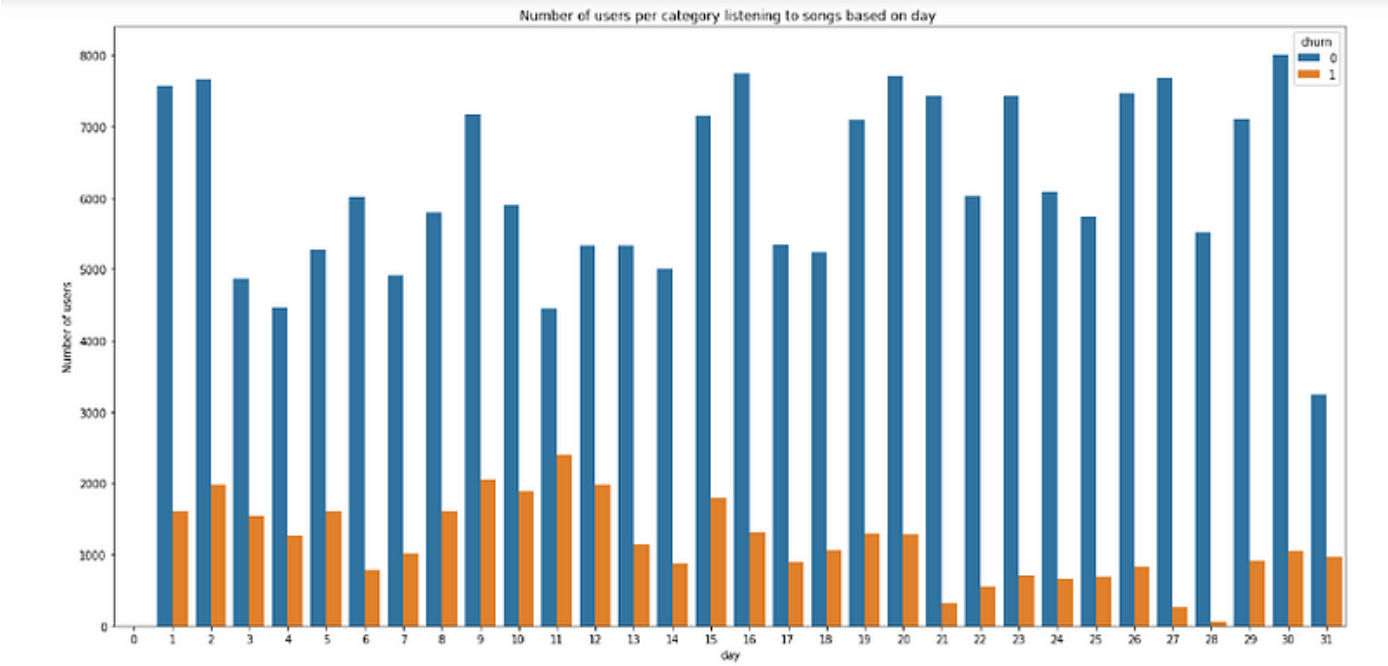
Conclusion: On the page list, the number of active users are way farther than cancelled users except for ‘Cancel’ and ‘Cancellation Confirmation’ pages. Out of all pages, the highest number of active users are visiting ‘Thumbs Up’,’Home’,’Add to playlist’,’Add to friend’ pages.

4. Count of users listening to songs based on hour of day labelled by their subscription status



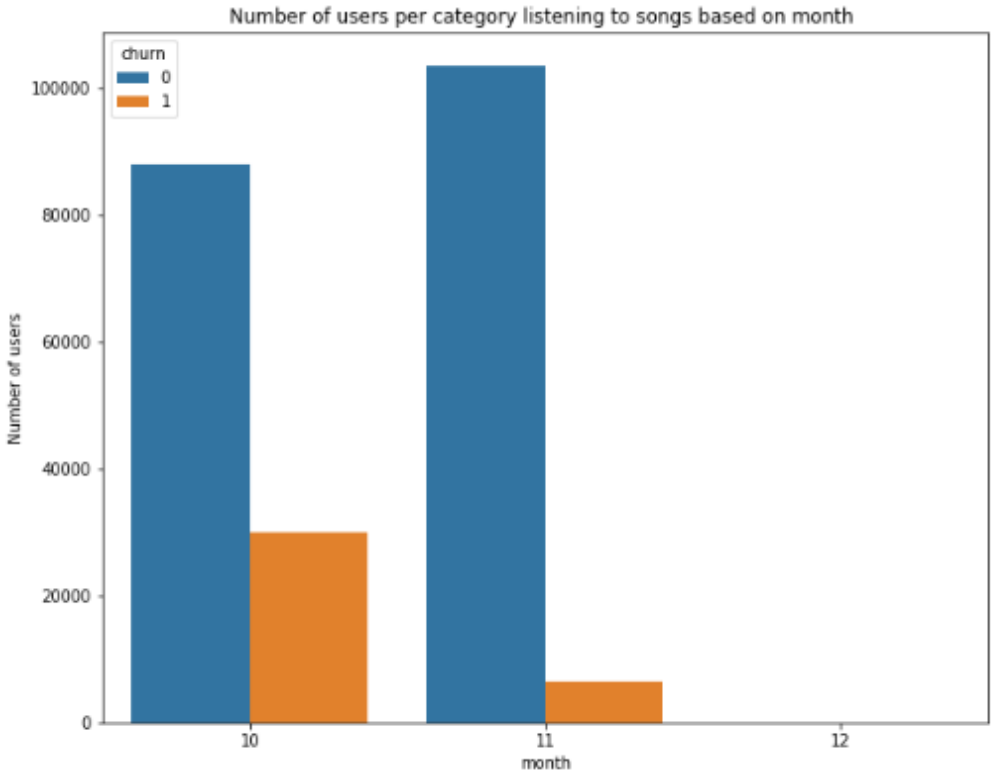
Conclusion: Most of the users are listening to the songs between 15th and 18th hours.

5. Count of users listening to songs based on day of month labelled by their subscription status



Conclusion: The no of users listening to the songs are increasing and decreasing , no particular trend is observed.

6. Count of users listening to songs based on month of year labelled by their subscription status



Conclusion: As the data is limited to 2 months,There is not much data to explore about the number of users listening to songs in a particular month of the year.

> *FEATURE ENGINEERING*

We will extract the following features to help create the model:

1) Total no of songs listened

2) Gender

3) Paid or Free users

4) Average songs played per session

5) Number of Thumbs up

6) Number of Thumbs down

7) Number of songs added to the playlist

8) Total number of friends

9) Churn(target)

Create an empty features list and merge all the above features(converting into numeric datatype)based on userId into a single, final dataframe which can be used for modelling.

> *IMPLEMENTATION AND MODEL EVALUATION*

Let's use vector Assembler and Standard Scaler to convert our numeric columns to vectors for ML modelling.

Split the final dataset into train and test data in the ratio of 0.8:0.2.

Since churn is a classification problem and I have a relatively small dataset, I decided to use f1-score as the main indicator since the number of users who stayed are in large numbers compared to those who cancelled and also accuracy to compare performance of different algorithms

Now, we have to define 2 functions:

For printing the performance metrics:Accuracy,F1score and training time.

For reporting the results of all ML models.

The ML classification models which are used here are:

Logistic Regression-This method is carried out on our 8 extracted and scaled features as featuresCol and our target variable 'churn' as our labelCol with max no of Iterations=10. We got F1score of 58% and accuracy of 70%.

Then I tried to fine-tune it's parameters by using **ParamGridBuilder()** while I train the model.

In this code, I tried to optimize the most important parameter of this algorithm.

regParam: Set the regularization parameter to [0.1,0.01,0.001]. Default is 0.0

Though the F1score and accuracy remained same as before, regParam of 0.1 is set to best fit for the data.

2. **Random Forest Classifier**-This method is carried out on our 8 extracted and scaled features as featuresCol and our target variable 'churn' as our labelCol .We got F1score of 70% and accuracy of 76.5%.

Then I tried to fine-tune it's parameters by
using **ParamGridBuilder()** while I train the model.

In this code, I tried to optimize 2most important parameters of this algorithm.

numTrees: Set the number of trees to train to [10,20].

maxDepth: Set Maximum depth of the tree(>= 0) to [10,20].(E.g., depth 0 means 1 leaf node; depth 1 means 1 internal node + 2 leaf nodes.')

We got F1score of 63% and accuracy of 70.6%.They actually decreased and this could be due to small size of dataset and large variation within,if one of the precision or recall is of low value then F1 score automatically reduces. Number of trees=10,Max depth=10 best fit the data.

3. **Support Vector Classifier**—This method is carried out on our 8 extracted and scaled features as featuresCol and our target variable 'churn' as our labelCol .We got F1score of 58.4% and accuracy of 70.6%.

Then I tried to fine-tune it's parameters by using **ParamGridBuilder()** while I train the model.

In this code, I tried to optimize the most important parameter of this algorithm.

maxIter: Set the number of iterations to [5,10].

Though the F1score and accuracy remained same as before, Max Iter=5 best fit the data.

CHALLENGES

Linear SVC model took a lot of time around 35min to train the data because the training time scales with O(num_samples² x num_features), so it was a challenge with so many features.

Due to the mini dataset used, hyperparameter tuning didn't make much difference in performance metrics. Making use of whole dataset(12GB) on aws cluster might resolve this issue.

The most challenging part of the project is creatively finding the features that will enable us to train a better model. I found 8 features but as we think through we can find many more features that will increase the accuracy of the model.

Finding the parameters that would increase the performance metrics as the parameter tuning. When I increase the number of alternatives for tuning parameters, the time to train the model increased so it became more difficult .

The following reports of performance metrics and training time for various ML models **without** hyperparameter tuning is as follows:

```
Logistic Regression
F1 Score: 0.584
Accuracy: 0.706
Total training time: 1.41 minutes

Random Forest Classifier
F1 Score: 0.703
Accuracy: 0.765
Total training time: 1.68 minutes

Support Vector Classifier
F1 Score: 0.584
Accuracy: 0.706
Total training time: 34.87 minutes
```

The following reports of performance metrics and training time for various ML models **with** hyperparameter tuning is as follows:

```
Logistic Regression
F1 Score: 0.584
Accuracy: 0.706
Total training time: 8.8 minutes

Random Forest Classifier
F1 Score: 0.629
Accuracy: 0.706
Total training time: 13.87 minutes

Support Vector Classifier
F1 Score: 0.584
Accuracy: 0.706
Total training time: 9.61 minutes
```

> *END-TO-END SUMMARY*

Loaded the json file using **‘Spark’** framework and removed the rows with empty userId & sessionId arrives at the following conclusion: When we are dealing with the rows of empty userId , he/she is a ‘Guest’ and trying to ‘Register’ or ‘Submit Registration’. When we are dealing with the rows of empty sessionId , he/she has ‘Logged Out’ and trying to ‘Login’. As we removed both the rows, the following rows associated with those values are removed.

Create a column Churn to use as the label for the model using the Cancellation Confirmation events on the page column and explored the behaviour of users who stayed v/s who churned based on level(paid/free), gender(female/male),page, ts.

Plotted some visualisations of the behavior of users and arrived at following conclusions:

a. The number of unique users are more active than cancelled users.

b. Males are slightly predominant than females in both active and cancelled subscriptions.

c. On the page list, the number of active users are way farther than cancelled users except for ‘Cancel’ and ‘Cancellation Confirmation’ pages. Out of all pages, the highest number of active users are visiting ‘Thumbs Up’,’Home’,’Add to playlist’,’Add to friend’ pages.

d. Most of the users are listening to the songs between 15th and 18th hours.

e. The no of users listening to the songs are increasing and decreasing , no particular trend is observed.

f. As the data is limited to 2 months,There is not much data to explore about the number of users listening to songs in a particular month of the year.

4. Performed feature engineering and extracted the following features and combined them into a single dataframe for performing the next step modelling:-

1) Total no of songs listened

2) Gender

3) Paid or Free users

4) Average songs played per session

5) Number of Thumbs up

6) Number of Thumbs down

7) Number of songs added to the playlist

8) Total number of friends

9) Churn(target)

5. Used vector Assembler and Standard Scaler to convert our numeric columns to vectors for ML modelling.Split the final dataset into train and test data in the ratio of 0.8:0.2.

6. Performance metrics:Accuracy,F1score and training time are chosen for reporting the results of all ML models.

7. The ML classification models which are used here are:

a. Logistic Regression

b. Random Forest Classifier

c. Support Vector Classifier

8.The following conclusions are drawn from training and testing different ML models:

a.Out of three ML models, Random Forest Classifier proves to be best model both in terms of F1score and accuracy.

b.Cross validation and hyperparameter tuning doesn't make much difference in output since the test dataset is a small subset of data.

c. With the tuning, parameters of Random Forest Classifier are best number of trees-10 and max depth of 10 best fit the model.

FUTURE IMPROVEMENT

In the future, the model can be improved if we have more data about the behavior of users on selecting songs , the genre then we could learn more about our users. Perhaps users refuse the service, not because of its price, but they simply do not find the songs they like. Discounts do not solve this problem.

Since, we worked on mini dataset, working on large dataset(12GB) might give some meaningful insights in our analysis on evaluating performance

metrics.