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*

1. Accuracy of the various models, tuning parameters as necessary.
2. 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*

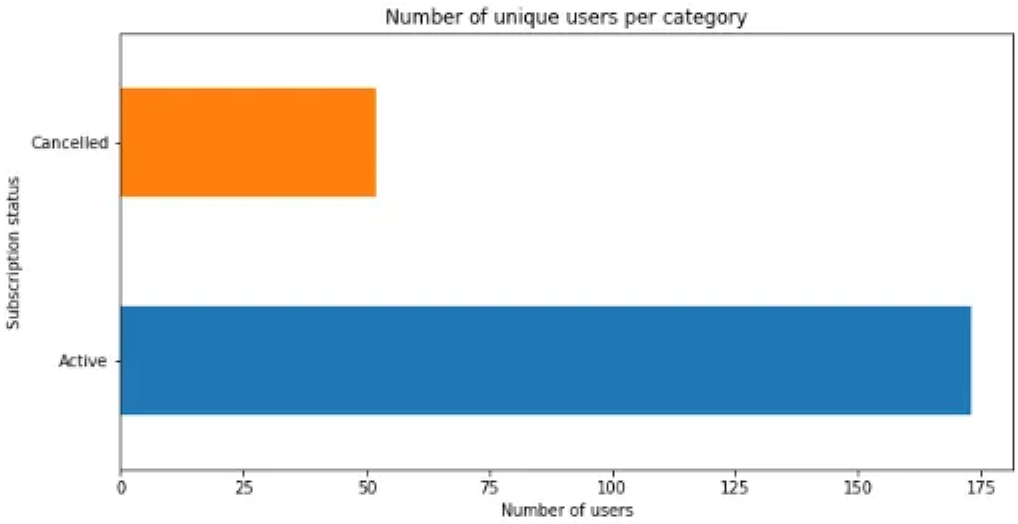
1. Load the json file using spark.read.json.
2. Our dataset contains the following columns: artist, auth, firstName, gender, itemInSession, lastName, length, level, location, method, page, registration, sessionId, song, status, ts, userAgent, userId.
3. Remove the rows with empty userId & sessionId srrives 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.

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

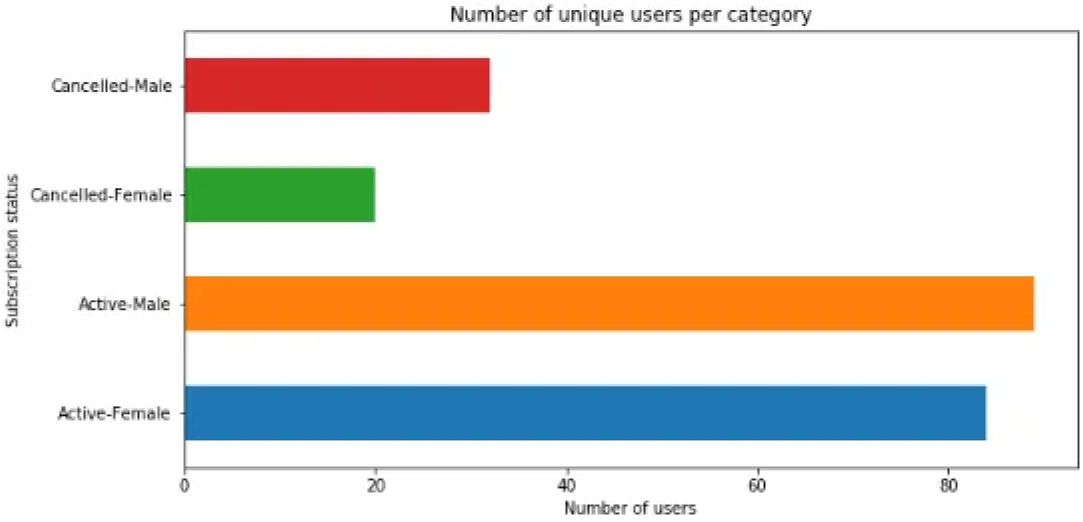
> *DATA VISUALISATION*

1. Count of users who stayed v/s who cancelled



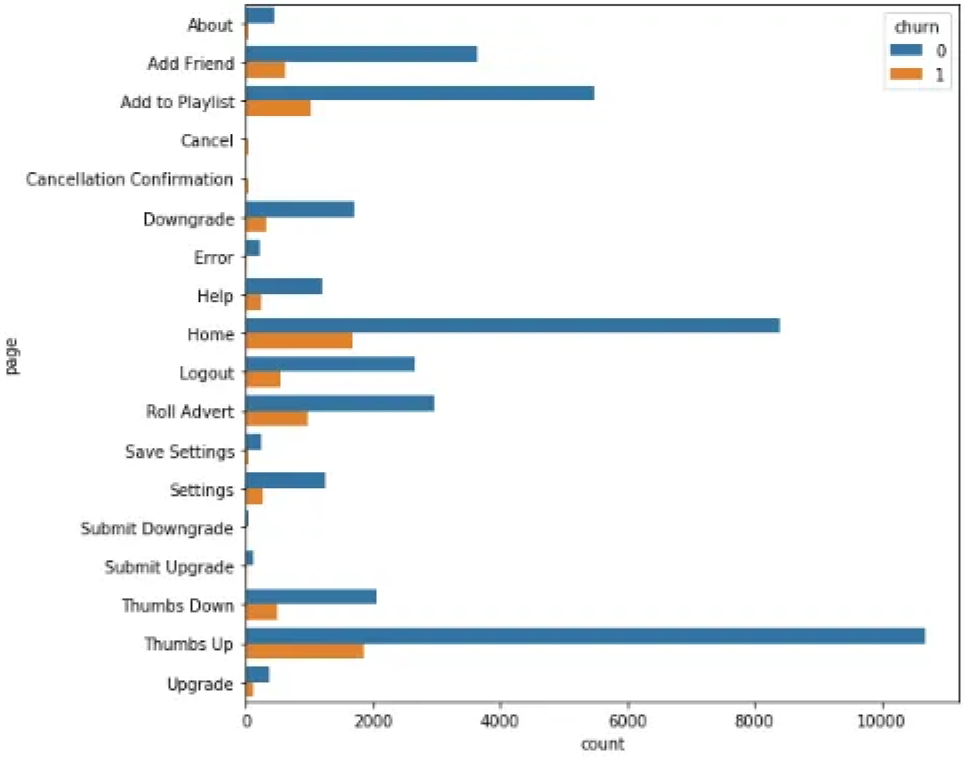
Conclusion: The number of unique users are more active than cancelled users.

2. Count of users based on Subscription status and gender



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

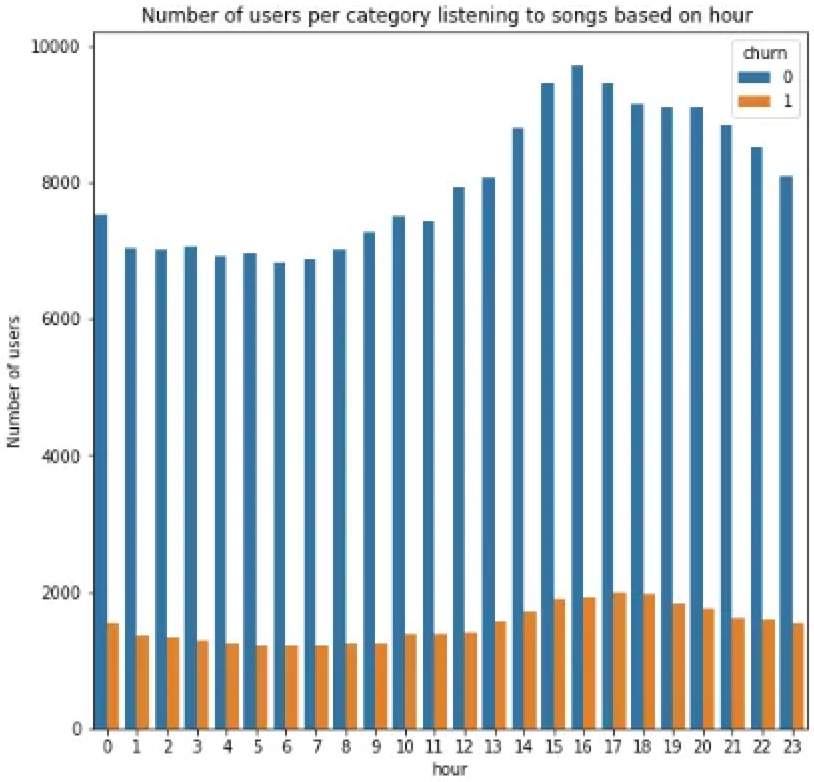
3. Count of users based on page requests labelled by their subscription status



Conclusion: On the page list, the number of active users are way farther than cancelled users except for 'Cancel' and 'Cancellation Confirmation' pages.

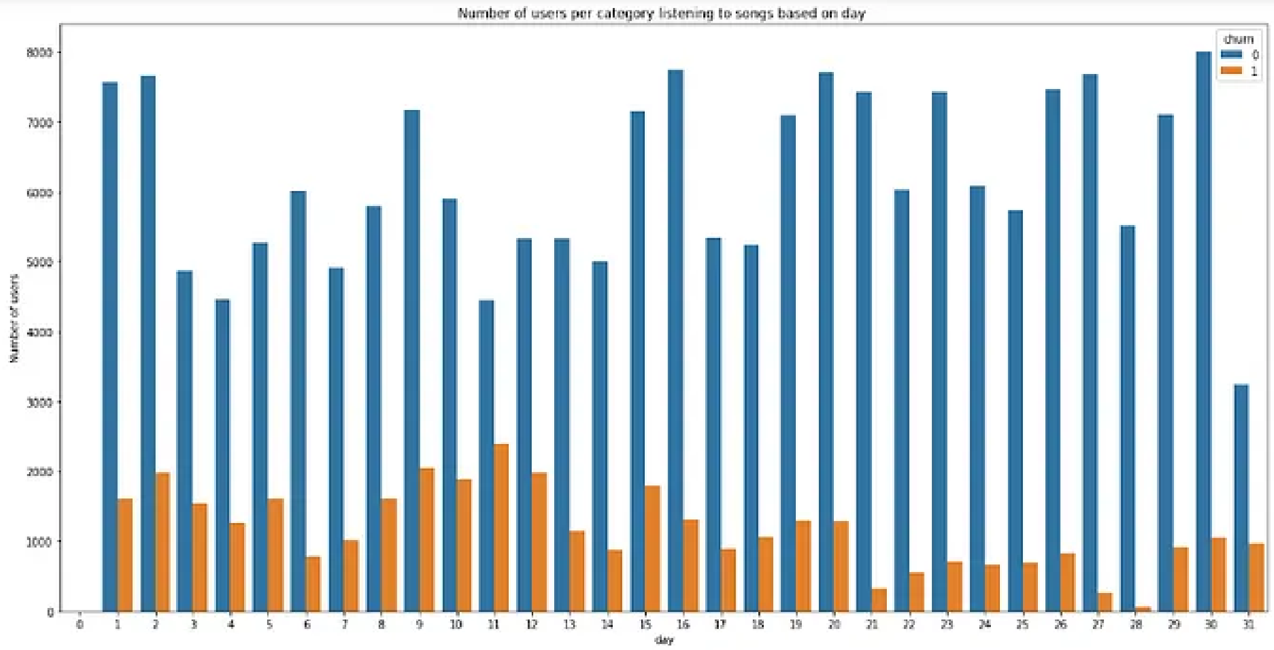
Out of all pages, the highest number of active users are visiting 'Thumbs Up','Home','Add to playlist','Add to friend' pages.

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



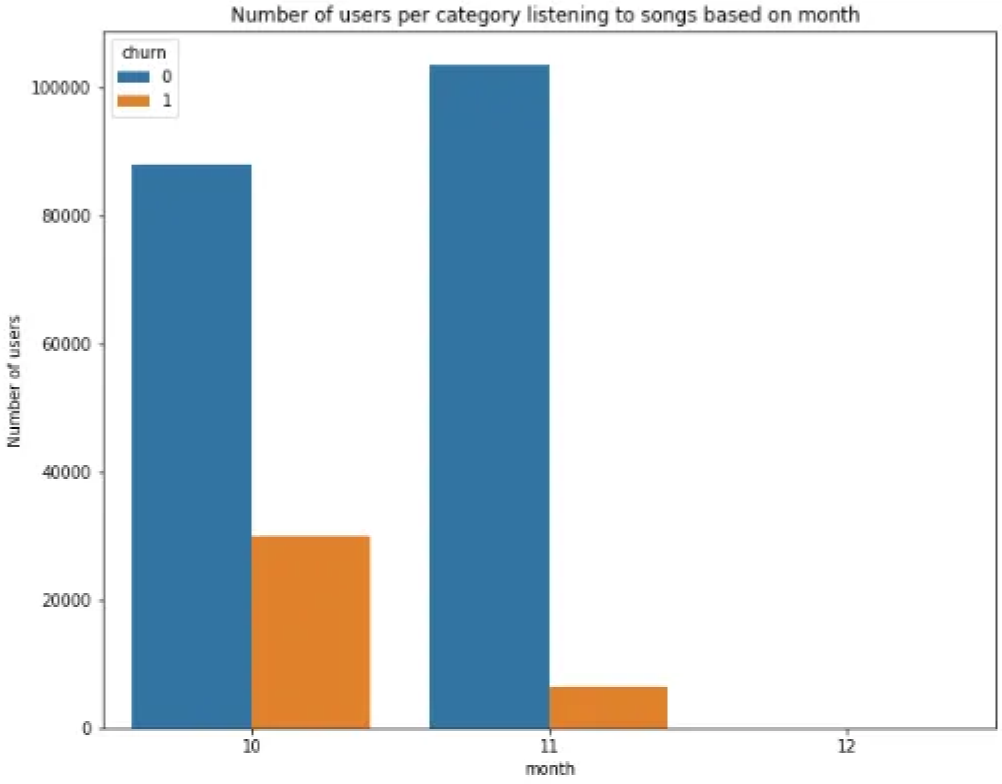
Conclusion: Most of the users are listening to the songs between 15th and 18th hours.

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.

> *MODELLING*

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.

Now, we have to define 2 functions:

1. For printing the performance metrics:Accuracy,F1score and training time.
2. For reporting the results of all ML models.

The ML classification models which are used here are:

1. Logistic Regression
2. Random Forest Classifier
3. Support Vector Classifier

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
```

> *CONCLUSION*

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

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

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