**Capstone Project Steven Greulich**

**Data Science Engineer Nanodegree February 6, 2019**

**Project Definition**

## **Project Overview**

Millions of users stream heir favorite songs through our service through the free tier that plays advertisements between songs, or using the premium subscription model that plays songs ad free, but pay a monthly fee. Users can upgrade, downgrade or cancel their service at any time. It is crucial that the users love the service. Every time a user interacts with a service (i.e. downgrading a service, playing songs, logging out, liking a song, hearing an ad, etc.) it generates data. All of this data contains the key insights to keeping the users happy and allowing the company to thrive. It is There is a fictional online music streaming site, Sparkify, in which we are interested in seeing if we can predict which users are at risk of cancelling their service. The goal is identify these users before they leave so they can hypothetically receive discounts and incentives to stay.

## **Problem Statement**

The overall goal of this project is to create a predictor system that can determine if a user is at risk of cancelling their subscription.

The tasks in order to do this is as follows:

* Download the “medium-sparkify-event-data.json” data
  + This data is located at the following file location: <https://s3.amazonaws.com/video.udacity-data.com/topher/2018/December/5c1d6681_medium-sparkify-event-data/medium-sparkify-event-data.json>)
* Preprocess the dataset to remove any missing values and to encode non-numerical values
* Train a classifier that can determine users that are at risk of cancelling their subscription
* Predict new users to determine the accuracy

Note: if the metrics of the predictions are low, performing model tuning as necessary.

## **Metrics**

Since this is a binary classifier (0 if the user is not at risk of cancelling their subscription and 1 if they are), we will be using the accuracy metric to determine how are model is doing on both the Training and Testing sets.

Accuracy = (True Positives + True Negatives) / Dataset Size

**Analysis**

## **Data Exploration**

Talk about at a high level the data (size, types of data etc.)

Talk about eh various fields (the types and what each represents

## **Data Visualization**

Show and talk about NaN

Show and talk about the various column types

**Methodology**

## **Data Preprocessing**

## **Implementation**

## **Refinement**

\*Make sure to show graphs of training and testing losses

**Results**

## Feature Engineering

Because of using Logistic Regression, I was able to get a training accuracy of around 99.99% and a testing accuracy of around 99.98%. This was due to the following transformations to the data:

String Indexer:

* Gender\_indexer (inputCol="gender")
* User\_indexer (inputCol="userAgent")
* Page\_indexer (inputCol="page")
* indexer (inputCol="Churn")

One Hot Encoding:

* Gender\_encoder (inputCol='Gender\_Index')
* User\_encoder (inputCol='User\_Index')
* Page\_encoder (inputCol='Page\_Index')

Vector Assembler:

* assembler (inputCols=["Gender\_Vec", "User\_Vec", "Page\_Vec", "status"])

## Acknowledgements

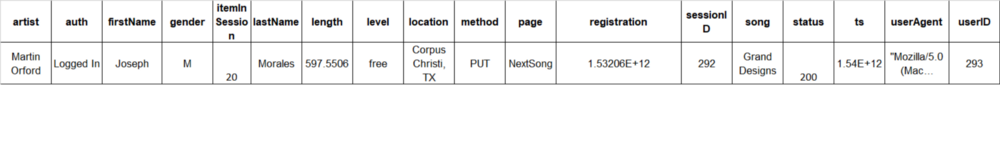
The data was taken from the following (<https://s3.amazonaws.com/video.udacity-data.com/topher/2018/December/5c1d6681_medium-sparkify-event-data/medium-sparkify-event-data.json>)

## Installation

The following was used for the environment setup:

* Python 3.5
* PySpark SQL
* PySpark ML

What I first decided to do was to load in the data set and look at all of the columns and the possible values that they may contain:



Example row of data

Note: What is not shown in the above is the predicted label, which is whether or not the user is an active member or cancelled their subscription.

When performing the feature engineering, I wanted to get user specific data removed that cannot be generalized to the overall population. I decided to keep the following columns: gender, userAgent, page, status, churn.

For the columns listed above, I used a mix of StringIndexer and OneHotEncoder to get these data columns ready to be inputting into the Machine Learning algorithm.

For the purposes of this project, I chose LogisticRegression as it is relatively simple to do and we were looking for a simple yes (1) or no (0) that the user might cancel their service.

I was actually quite shocked how well this model performed on both the training and the testing set.

For the training, it received a 99.99% accuracy whereas the testing set received a 99.98% accuracy.