

# Machine Learning Engineer Nanodegree

## Capstone Project

Matheus Luz

24 de Agosto de 2018

## I. Definição

### Project Overview

Este é um projeto que segue a competição da kaggle<sup>1</sup>, onde sua intenção é criar um Sistema de recomendação de músicas<sup>2</sup>. A competição forcene um dataset, no qual devemos utilizar para descobrir se o usuário irá ouvir uma determinada música novamente, dentro de um período de 1 mês.

### Problem Statement

O dataset possui uma variável 'target', que informará 1 se a música foi ouvida novamente em um período de um mês, e 0 caso contrário. Para realizar a previsão, possuímos uma série de informações, como: Nome do usuário, autor da música, data de lançamento da música, etc.

### Metrics

#### *ROC curve*

Na classificação binária, a predição de classe para cada instância é frequentemente feita com base em uma variável aleatória contínua  $X$ , que é uma "pontuação" calculada para a instância (por exemplo, o preditor linear na regressão logística). Dado um parâmetro de limite  $T$ , a instância é classificada como "positiva" se  $X > T$  e "negativa" de outra forma.  $X$  segue uma densidade de probabilidade  $f_1(x)$  se a instância realmente pertencer à classe "positivo" e  $f_0(x)$  se de outra forma. A taxa positiva verdadeira é dada por

$$(TPR(T) = \int_{-\infty}^{\infty} T f_1(x) dx)$$

e a taxa de falsos positivos é dada por

$$(FPR(T) = \int_{-\infty}^{\infty} T f_0(x) dx)$$

onde a curva ROC plota parametricamente  $TPR(T)$  versus  $FPR(T)$  com  $T$  como o parâmetro variável.

## II. Analysis

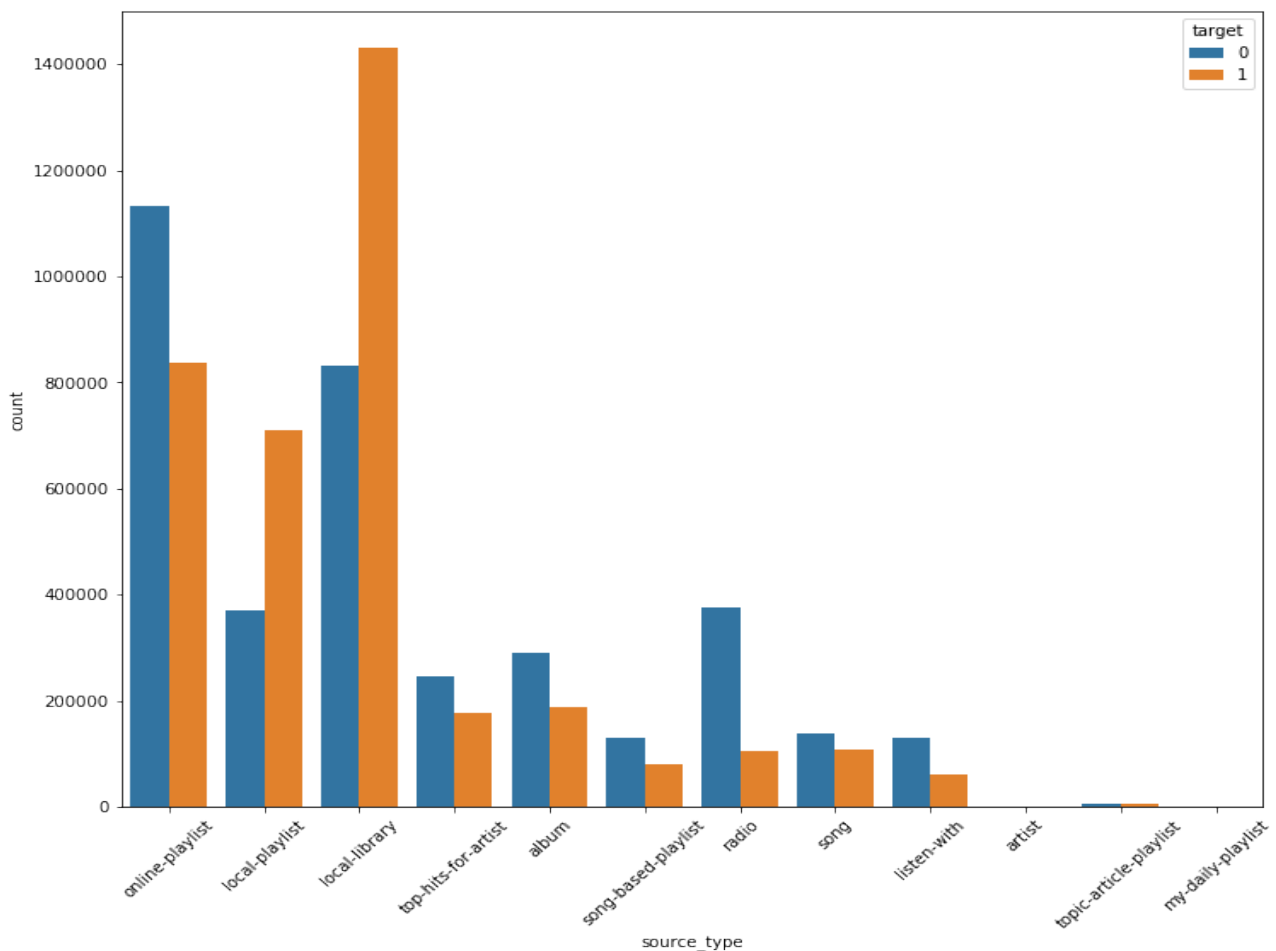
(approx. 2-4 pages)

### Data Exploration

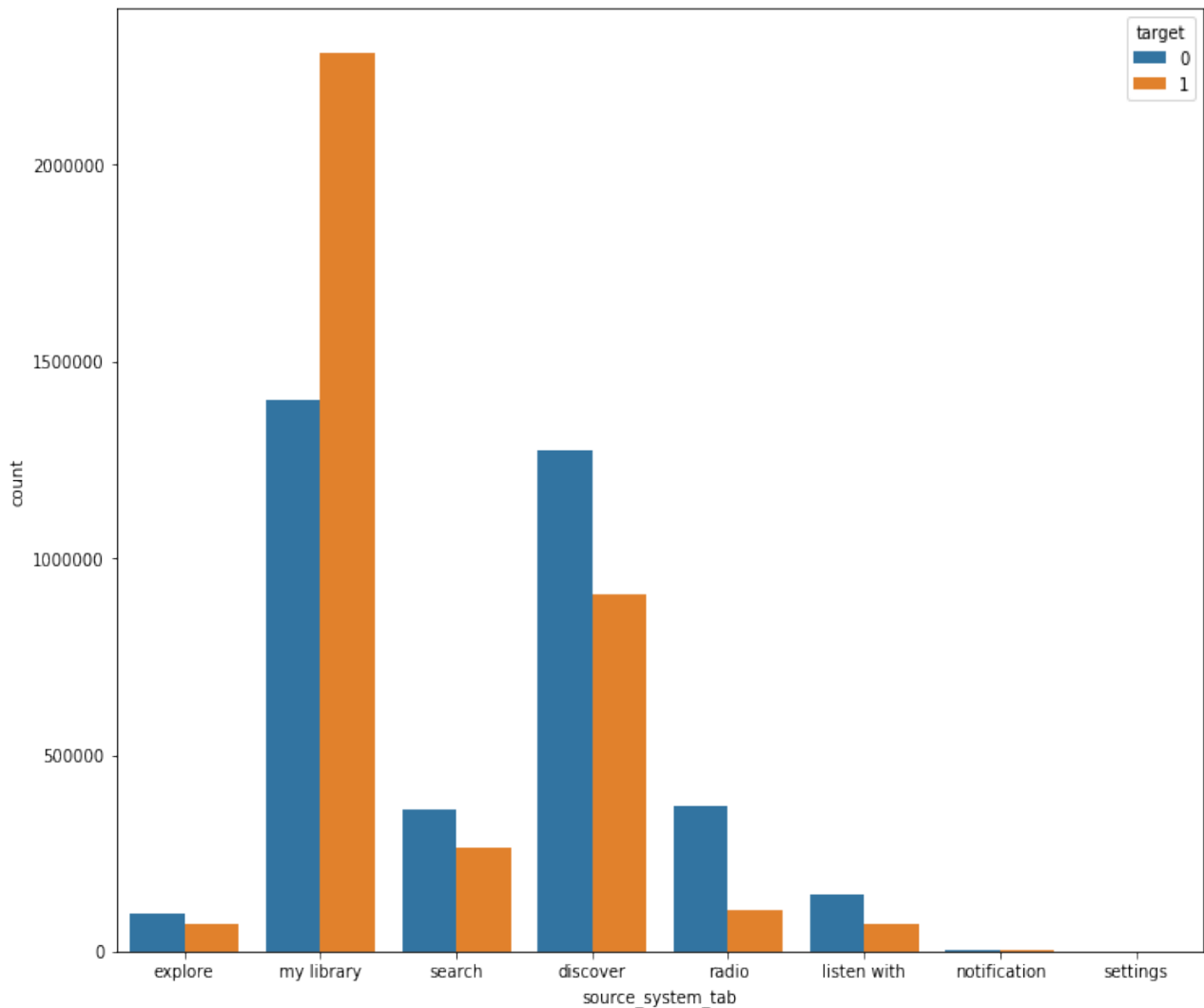
Analizando os dados, é possível observar que em alguns pontos as informações precisam ser corrigidas. Um exemplo disso é a coluna “BD” (Birthday), que possui 195 registros que possuem idade igual a 0. Este e outros tratamentos serão levados em consideração ao desenvolvermos a solução.

### Exploratory Visualization

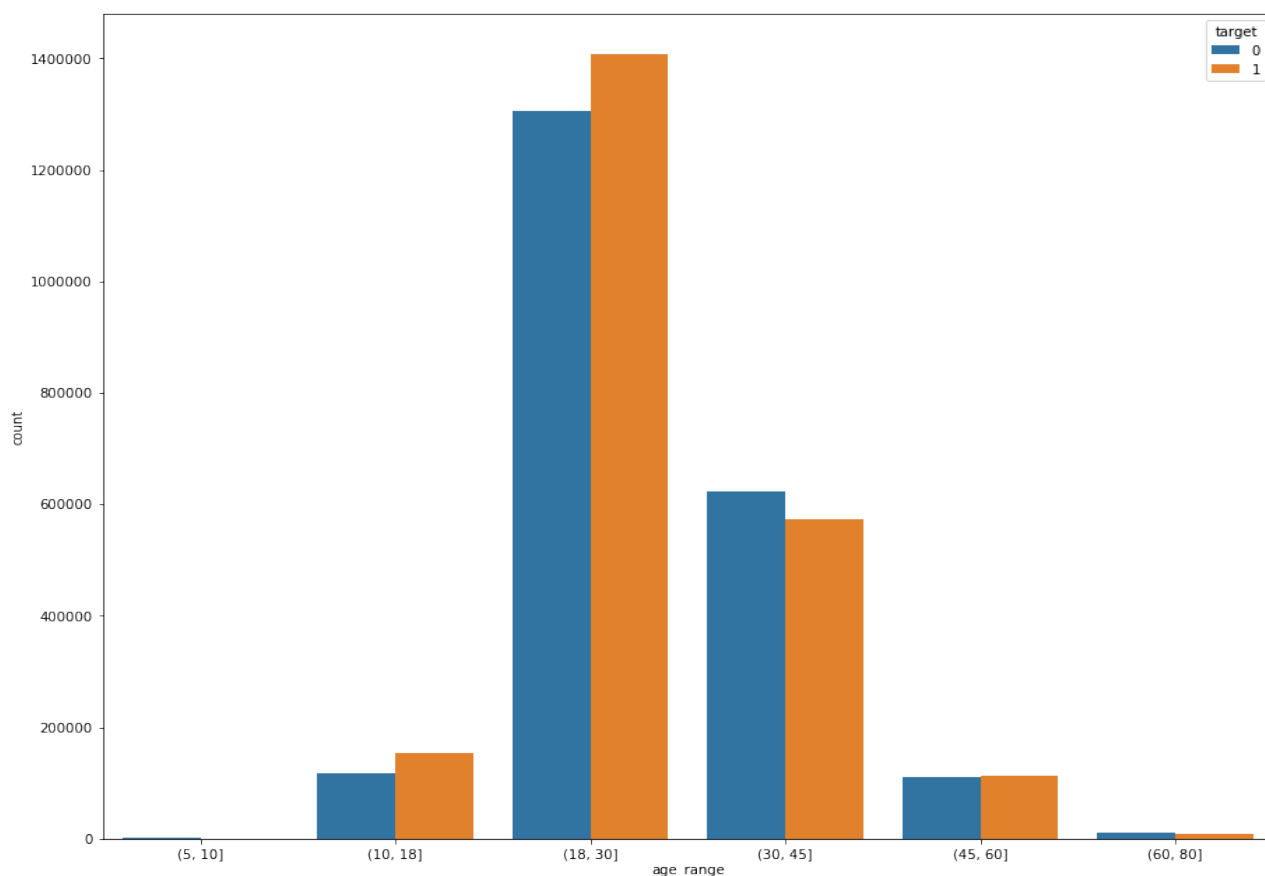
Utilizando a library Matplotlib podemos fazer a análise de algumas colunas, como a ‘source\_system\_tab’:



É possível então visualizar que o maior responsável pela execução das músicas pelo sistema é o ‘my library’, onde as mesmas são ouvidas novamente. Já na coluna ‘radio’ podemos ver que existe um número maior de músicas que apenas foram ouvidas uma vez (dentro de um mês).



Neste outro gráfico podemos observar a coluna ‘source type’, onde as categorias ‘local-library’ e ‘local-playlist’ possuem uma porcentagem maior de usuários que ouviram a mesma música novamente.



Com o age\_range, podemos observar que a maior parte do público possui idade entre 18 a 30 anos, seguido pelo público de 30 – 45 anos.

## Algorithms and Techniques

Será utilizado um algoritmo de Gradient Boosting para a solução do problema, com a library “Catboost” desenvolvida pela empresa Yandex. Utilizaremos hiperparâmetros como Number of trees para evitar overfitting e Learning rate para diminuir o gradient step.

## Benchmark

O modelo de Leaderboard público com score de 0.68310 será utilizado como modelo de benchmark. Será realizada uma tentativa para que nosso modelo possa se encontrar na metade para cima top 50%) na lista de Leaderboard.

## III. Methodology

*(approx. 3-5 pages)*

## Data Preprocessing

Durante nossa implementação utilizaremos uma série de tratamentos para desenvolver um dataset mais efetivo. Primeiramente, iremos unificar as colunas de songs.csv e members.csv para agrupar o conjunto de informações com o train.csv.

Também lidaremos com valores nulos, substituindo as categorias vazias com o valor “Unknown” em colunas como ‘gender’, ‘source\_system\_tab’, ‘source\_screen\_name’, ‘source\_type’, genre\_ids, etc.

## Implementation

Devido a grande quantidade de informações contidas no dataset, foi necessário realizar alguns ajustes para que o número de campos fosse o menor e mais eficiente possível. Com isso em mente, foram criados Campos de Count para armazenar e dar destaque em informações que eram contidas em mais de uma linha.

Foi separado uma parte de 60% dos dados para treinamento, 20% de validação e 20% de teste. Durante o treinamento do nosso modelo foi utilizamos o CatBoostClassifier, com 100 interações de treinamento.

## Refinement

As configurações utilizadas para melhorar o desempenho do CatboostClassifier foram as seguintes:

- *100 interações de treinamento* -
- learning rate configurado para 0.5
- profundidade máxima da árvore de decisão configurada para 12

## IV. Results

(approx. 2-3 pages)




















### Model Evaluation and Validation

Considerando o score obtido pela métrica estabelecida previamente, podemos dizer que o modelo consegue fazer um bom trabalho ao distinguir músicas que possuem uma maior probabilidade de serem ouvidas que outras. Ao utilizar um modelo de gradient boosting (Um dos modelos mais utilizados em competições da kaggle) nesses casos de classificação binária podemos contar com seu algoritmo robusto para obter uma solução do problema.

## Justification

Comparando com nosso modelo de benchmark anteriormente, podemos ver que o score ficou apenas um pouco abaixo. Como foram algoritmos de treino e tratamento de dados diferentes, é de se esperar que o resultado não seja o mesmo.

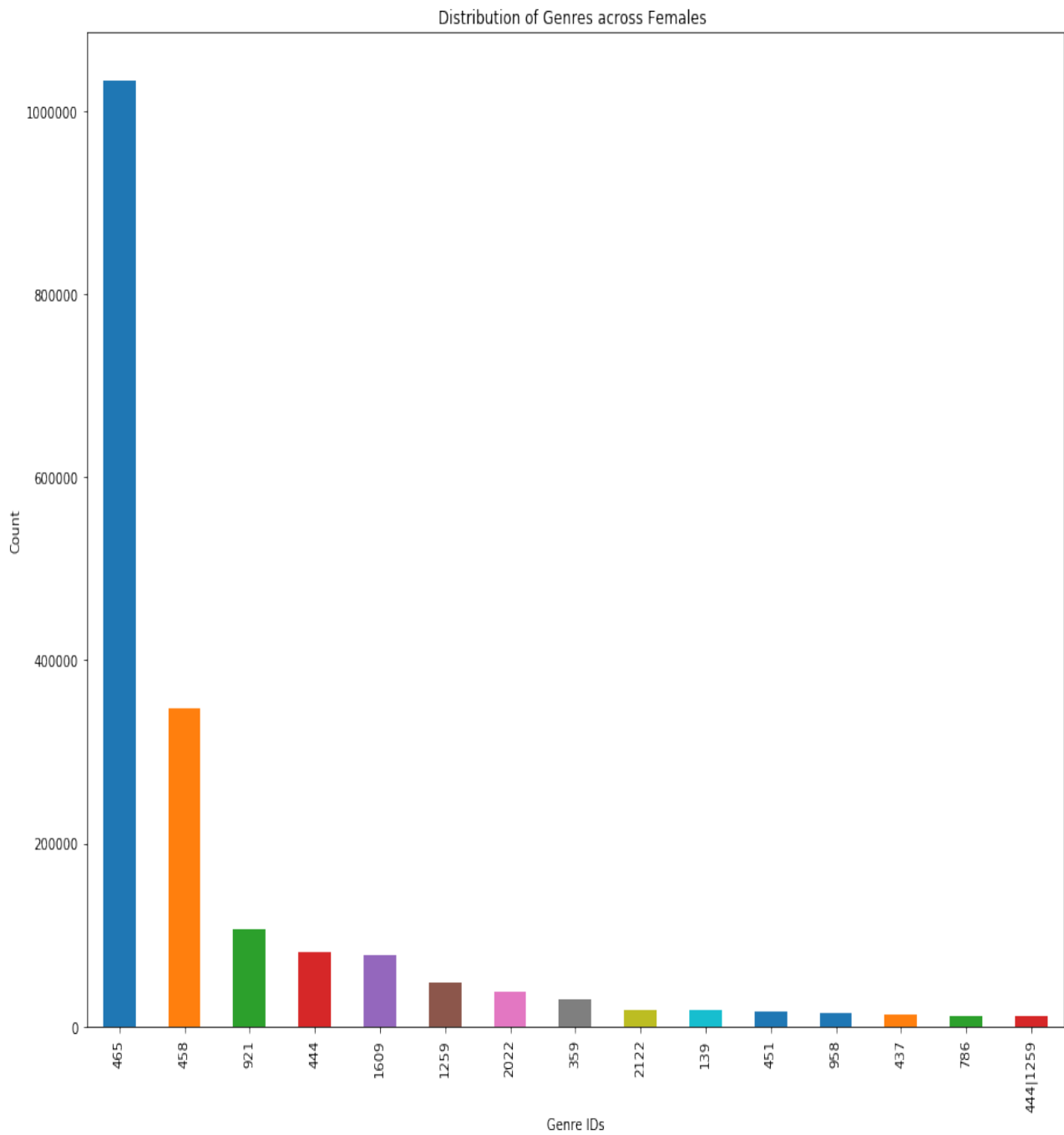
Porém, segundo o Leaderboard da página, o objetivo de obter um score acima dos 50% das submissões enviadas foi concluído.

Avatar	Score	Count	Time
	0.67362	10	8mo
   +3	0.67348	57	8mo
	0.67346	2	8mo
	0.67344	1	1y
	0.67338	21	1y
	0.67331	5	8mo
	0.67331	4	8mo
	0.67325	1	9mo
 	0.67325	14	10mo
	0.67323	6	1y
	0.67297	27	10mo
	0.67284	7	1y
	0.67284	10	10mo
	0.67282	8	10mo
	0.67282	22	8mo
	0.67282	5	10mo

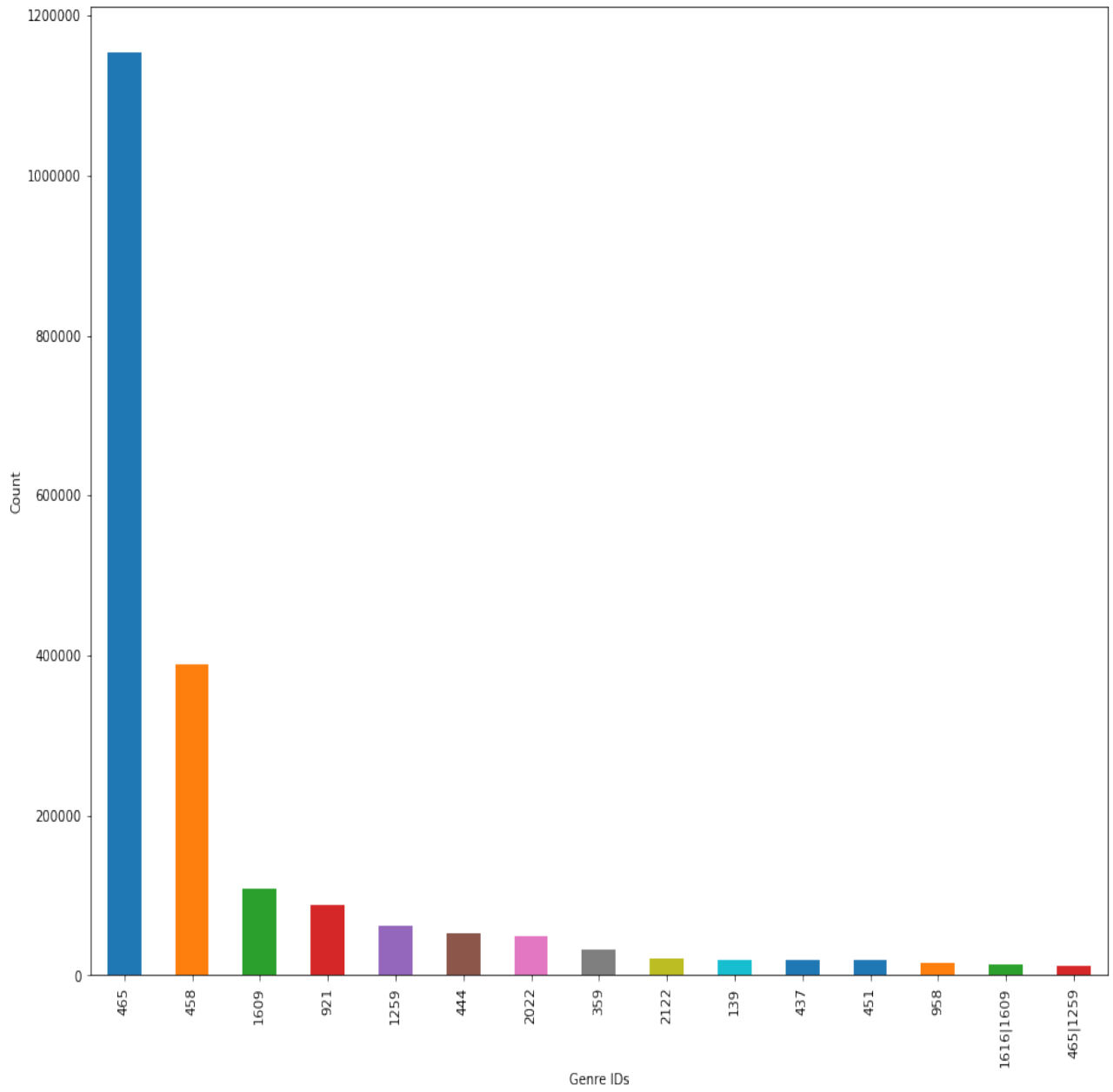
# Free-Form Visualization

Este foi um projeto interessante não apenas para prever preferências de músicas, mas também para analisar quais são as preferencias de um grupo específico.

Como exemplo, podemos analisar que apesar que o gosto musical em relação ao gênero da música pode ser diferente de homens para mulheres



Distribution of Genres across Males





## **Reflection**

Este foi um projeto interessante, pois eu nunca havia pensado em sistemas de recomendação trabalhando desta forma. Ao analisar a probabilidade de uma música ser ouvida novamente dentro de um tempo, podemos até, de certa forma, definir se uma nova música teria uma boa taxa de aceitação ou não.

Como o modelo trabalha com um grande número de categorias, é necessária a atualização dos procedimentos de treinamento caso uma nova categoria seja lançada em seu conjunto de dados, o levando a necessidade de constante evolução.

## **Improvement**

Como mencionado anteriormente, apesar do modelo atual possuir um bom resultado, ele acaba sendo frágil caso tenha que lidar com novas categorias, o levando a passar por um treinamento constante.

Como solução, seria possível generalizar os campos para que se mantenham apenas as categorias fixas (como gênero musical) para que o modelo possua uma melhor capacidade de lidar com dados novos (em troca de um pouco de precisão).

Fontes:

<https://www.kaggle.com/sidshady/basic-data-analysis-and-exploration>

[https://tech.yandex.com/catboost/doc/dg/concepts/python-reference\\_parameters-list-docpage/](https://tech.yandex.com/catboost/doc/dg/concepts/python-reference_parameters-list-docpage/)

<https://www.kaggle.com/asmitavikas/feature-engineered-0-68310>

<https://www.kaggle.com/sidshady/light-gbm-lb-0-65189>