

**UNIVERSIDADE ESTADUAL DE CAMPINAS**

**FACULDADE DE CIÊNCIAS APLICADAS**

**PO450 A – Introdução Aprendizado de Máquina e à Mineração de Dados**

Prof. Leonardo Tomazeli Duarte

Pré Projeto

**Criação de *playlists* de músicas de acordo com o gosto pessoal na plataforma *Spotify***

Autor:

Lucas Arruda Bonservizzi (182355) – lucasbbr98@gmail.com

Limeira, 19 de abril de 2019

**Índice**

1. **Introdução**
2. **Análise de dados**
3. **Considerações finais**
4. **Apêndice**

**Introdução**

Inteligência artificial (IA), o aprendizado de máquinas (Machine Learning) e aprendizagem profunda (Deep Learning) são temas populares utilizados por muitos como sinônimos para destacar uma série de avanços e promessas para os próximos anos. Apesar de essencialmente serem diferentes entre si, as três áreas visam tentar resolver problemas desafiadores através da interpretação de dados e vêm obtendo resultados expressivos, sendo responsáveis pela promoção de vantagens competitivas de algumas empresas. Alguns exemplos reais são o sistema de recomendação de filmes da Netflix e o sistema de customer relationship management (CRM) da Salesforce. Não obstante, também existem aplicações em âmbitos não empresariais que não visam lucros, como por exemplo a aplicação de robôs para jogar xadrez Alpha Zero, da Google, e Leela Zero, open source.

Mais especificamente sobre o aprendizado de máquinas, existem dois principais tipos de algoritmos de aprendizagem: a supervisionada e a não supervisionada. O primeiro caso é caracterizado quando se tem um conjunto de dados rotulados que já sabemos qual é a saída, geralmente são caracterizados em problemas de “regressão” ou “classificação”. Em um problema de regressão, ocorre a tentativa de mapear variáveis ​​de entrada para alguma função contínua ao passo que em um problema de classificação para categorias distintas. Já o segundo permite a abordagem de problemas com pouca ou nenhuma ideia da aparência dos resultados ou correlação das variáveis. Contudo, o feedback com base nos resultados da previsão é complexo, necessitando de maior atenção acerca da eficiência do algoritmo.

Dessa forma, o objetivo do trabalho é criar um algoritmo de recomendação de playlists de músicas na plataforma Spotify. Diversas opções ponderaram esta decisão. Em primeiro lugar, o Spotify é uma plataforma amplamente utilizada, com 87 milhões de assinantes. Sendo assim, a solução tem uma aplicação prática que pode ser utilizada por qualquer um no seu dia a dia. Em segundo lugar, a empresa tem uma boa presença na comunidade open source, e fornece uma interface de programação (API) gratuita com dados de todas as músicas disponibilizadas dentro da plataforma, facilitando a captura massiva de dados. Por último, além de música ser um tema do meu gosto pessoal, acredito que há grandes espaços para melhoria do algoritmo de recomendação já existente na plataforma, mais especificamente na criação de playlists diárias com base em outras que o próprio usuário construiu manualmente.

**Análise de dados**

Antes de demonstrar a escolha prévia de quais variáveis são fornecidas pelo Spotify e escolhidas, vale ressaltar algumas hipóteses que foram feitas para a criação do algoritmo. Em primeiro lugar, assumiu-se que as músicas podem ser categorizadas apenas como “Boas” e “Ruins”, para uma determinada pessoa. Como a base do algoritmo depende de listas de músicas, assumiu-se também que uma playlist contendo somente músicas “Boas” e outra de “Ruins” seriam fornecidas pelo usuário para o treinamento do modelo – o que é parcialmente verdade, visto que provavelmente poucos usuários possuem uma playlist somente de músicas “Ruins”. Por último, os atributos de cada música fornecidos pelo próprio Spotify foram considerados como corretos.

Tendo em vista as hipóteses mencionadas, uma análise prévia de dados foi feita utilizando a linguagem interpretada Python. A lista de músicas “Boas” continha um total de 316 músicas ao passo que a de “Ruins” continha 132, ou seja, um total de 448. Uma análise gráfica foi feita e disponibilizada no apêndice.

Para cada música, 10 atributos foram coletados:

* Tempo ou Batidas por minuto (bpm):
* Dança (0-1): Uma combinação de ritmo, força da batida e regularidade que tenta descrever se uma música é boa para dançar, onde 1 representa a melhor chance.
* Duração (ms): a duração da música em milissegundos.
* Volume (dB): O volume geral da música em decibéis entre -60 e 0.
* Falancia: Detecta a presença de palavras faladas em uma música. Quanto mais falada, mais perto de 1 é a nota. Valores acima de 0.66 indicam músicas feitas somente de falas. Valores entre 0.33 e 0.66 representam músicas que contem instrumentos e falas, como por exemplo raps e hip hops. Valores abaixo de 0.33 representam músicas mais cantadas.
* Valência (0-1): Uma medida entre 0 e 1 para descrever a positividade em uma música. Quanto maior a nota, maior a valência e maior a positividade, como por exemplo músicas felizes e alegres.
* Energia (0-1): Energia é uma medida de 0 a 1 e representa a atividade e intensidade de uma música. Geralmente, músicas energéticas são rápidas, altas e barulhentas, como por exemplo o gênero de rock metal. Alguns outros atributos são utilizados como entropia geral, faixa dinâmica, timbre e onset.
* Acústica (0-1): Medida de confiança de que uma música pode ser considerada acústica, onde 1 representa alta confiança.
* Instrumental: Mede se uma música contem ou não voz. Sons de “Ooh” e “aah” são tratados como instrumentos neste caso e palavras ditas como vozes. Para notas acima de 0.5, a música pode ser considerada sem voz, mas quanto mais perto de 1, maior a confiança da música de fato não conter nenhuma voz.
* Notas musicais: Nota musical geral da música, onde 0 = C, 1 = C♯/D♭, 2 = D, e assim por diante. O valor pode ser -1, caso a nota geral da música não foi identificada.

**Considerações finais**

Tendo em vista todos os fatos mencionados, pode-se concluir que o objetivo do projeto é criar um algoritmo de recomendação de listas de músicas na plataforma Spotify. O algoritmo será supervisionado, uma vez que duas playlists serão fornecidas na entrada, sendo uma de músicas “Boas” e outra de músicas “Ruins” e poderá utilizar técnicas de regressão e/ou classificação.

**Apêndice: análise de dados prévia**

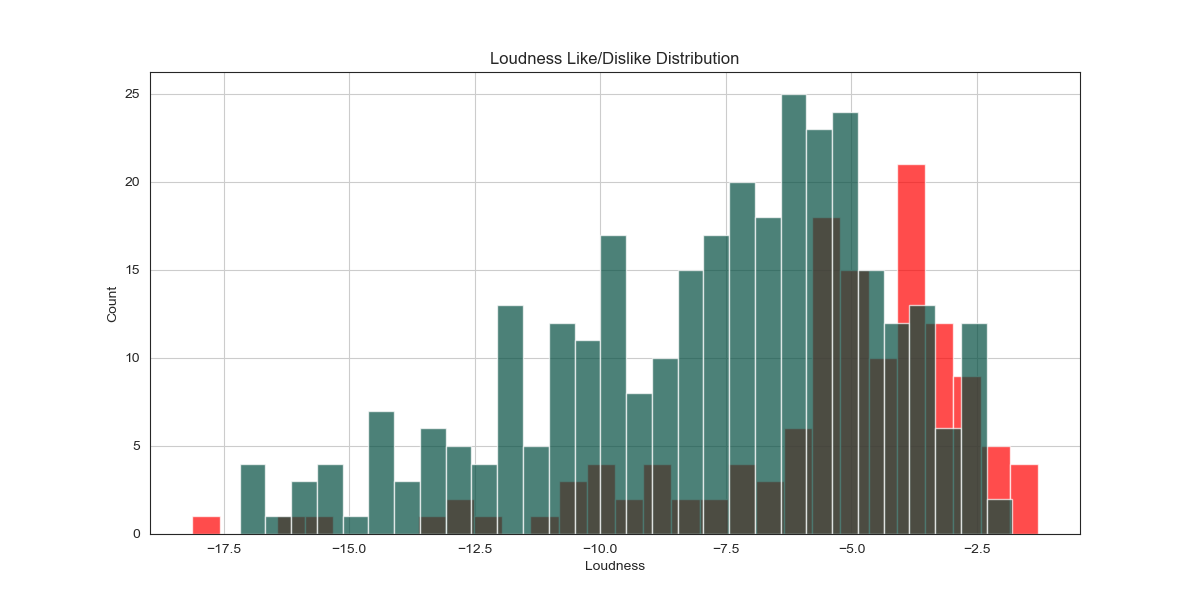
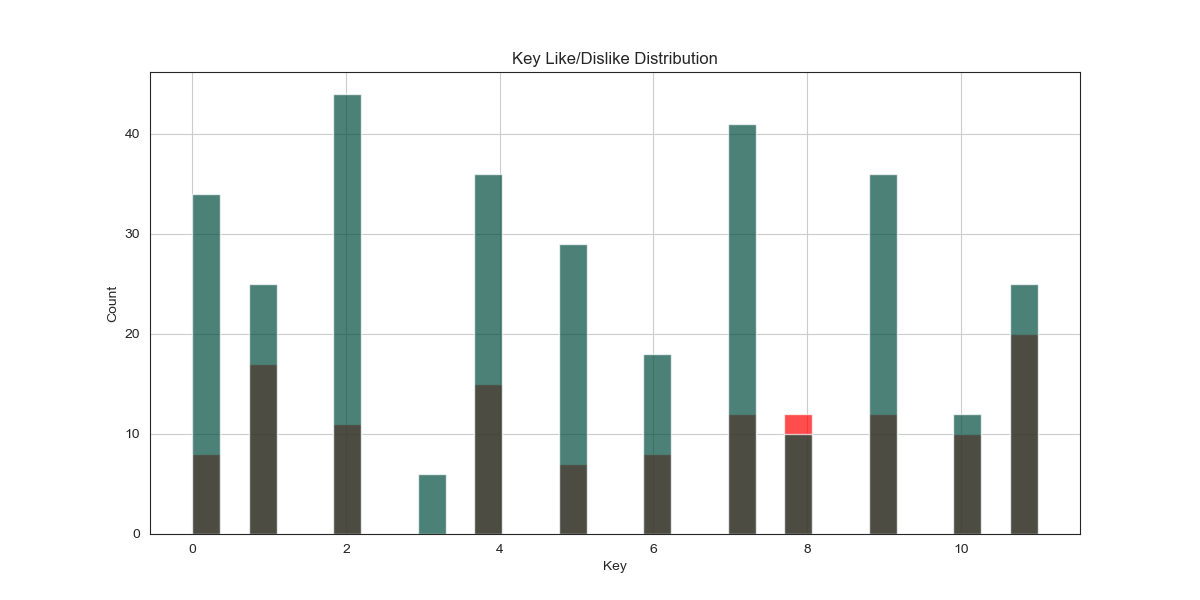
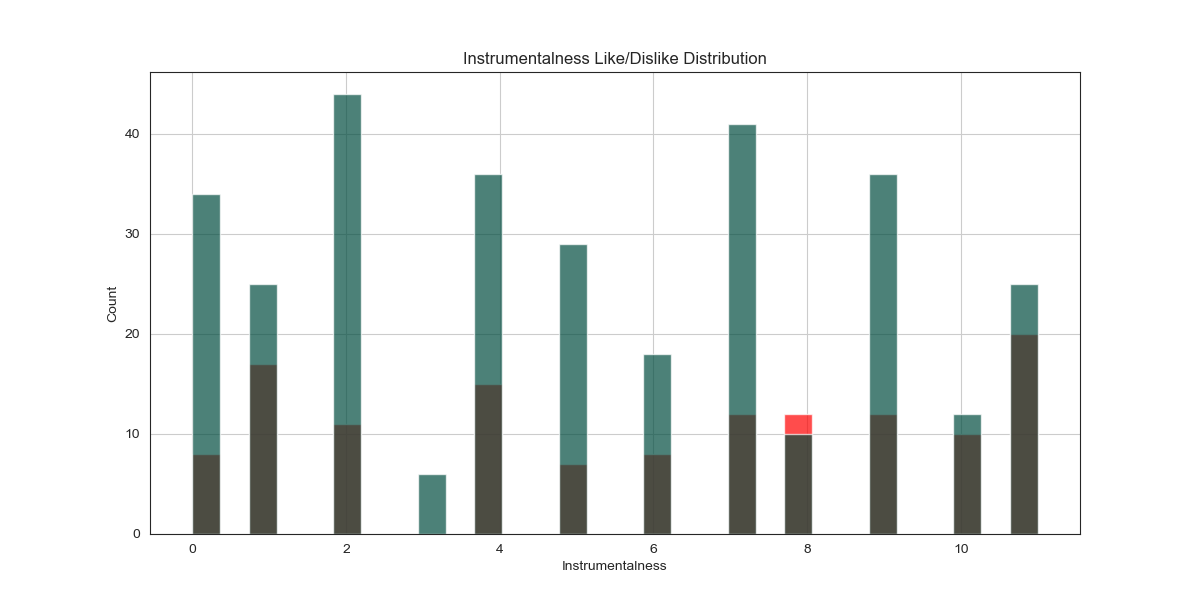
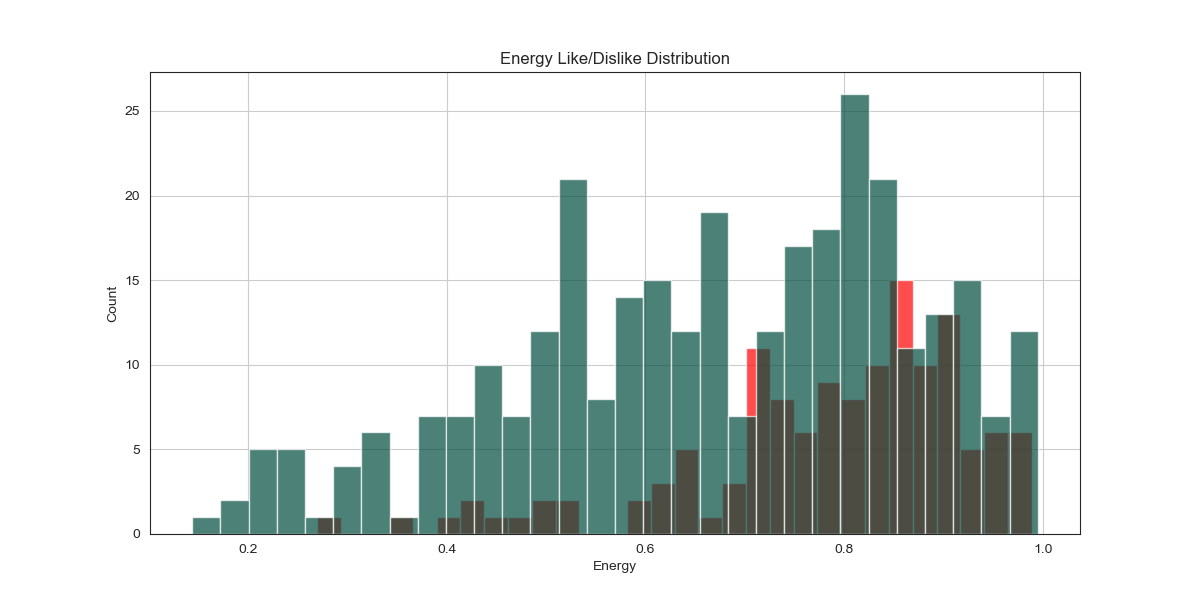
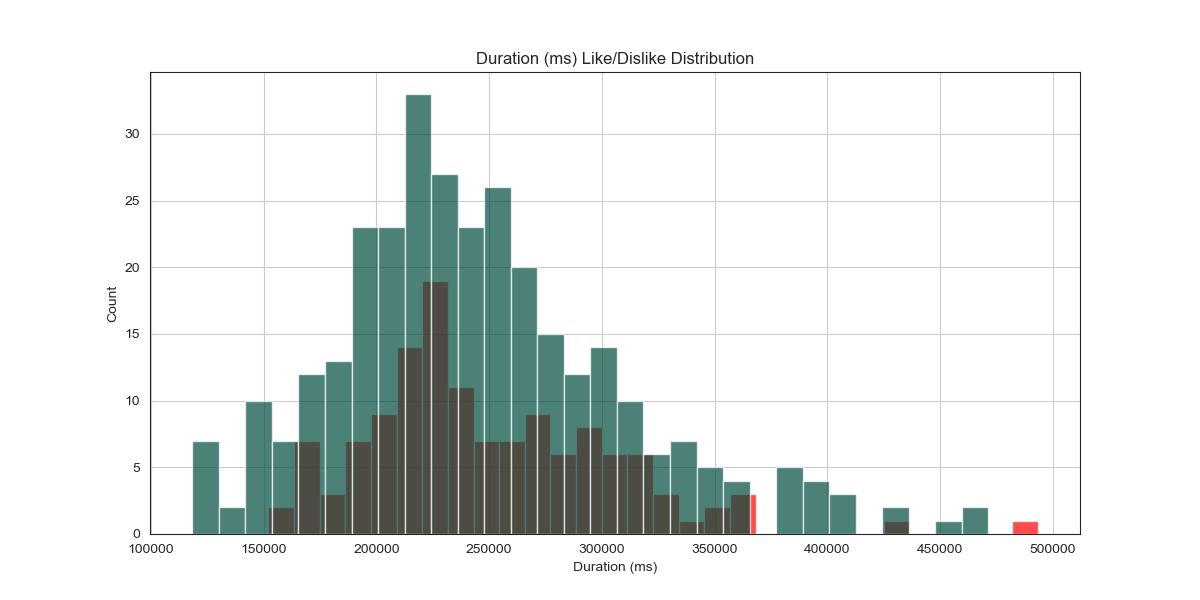
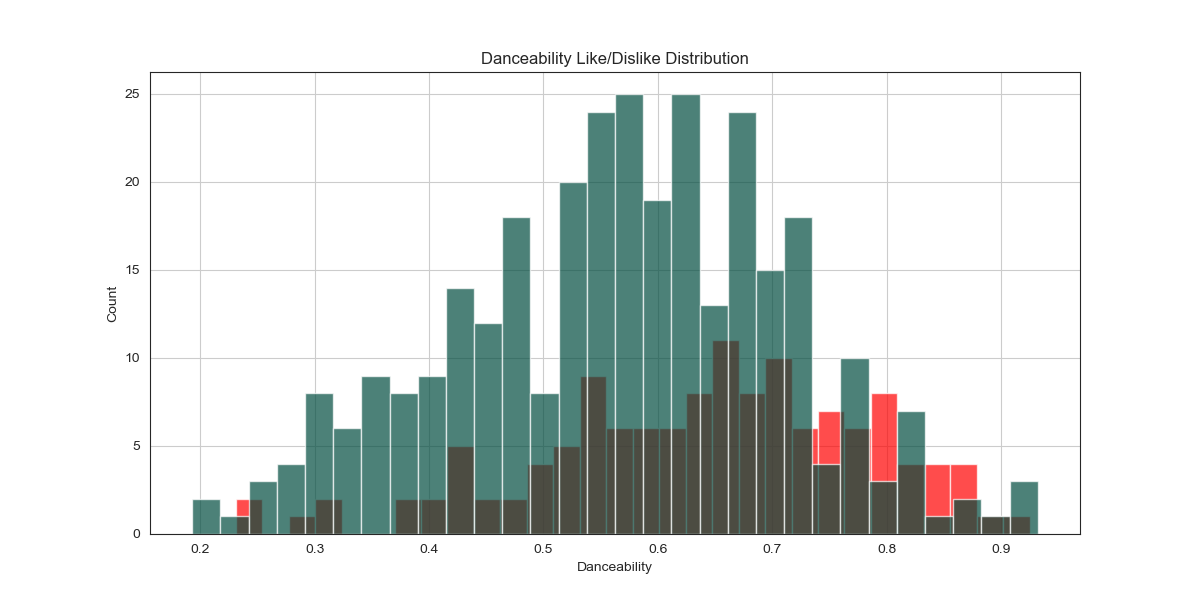
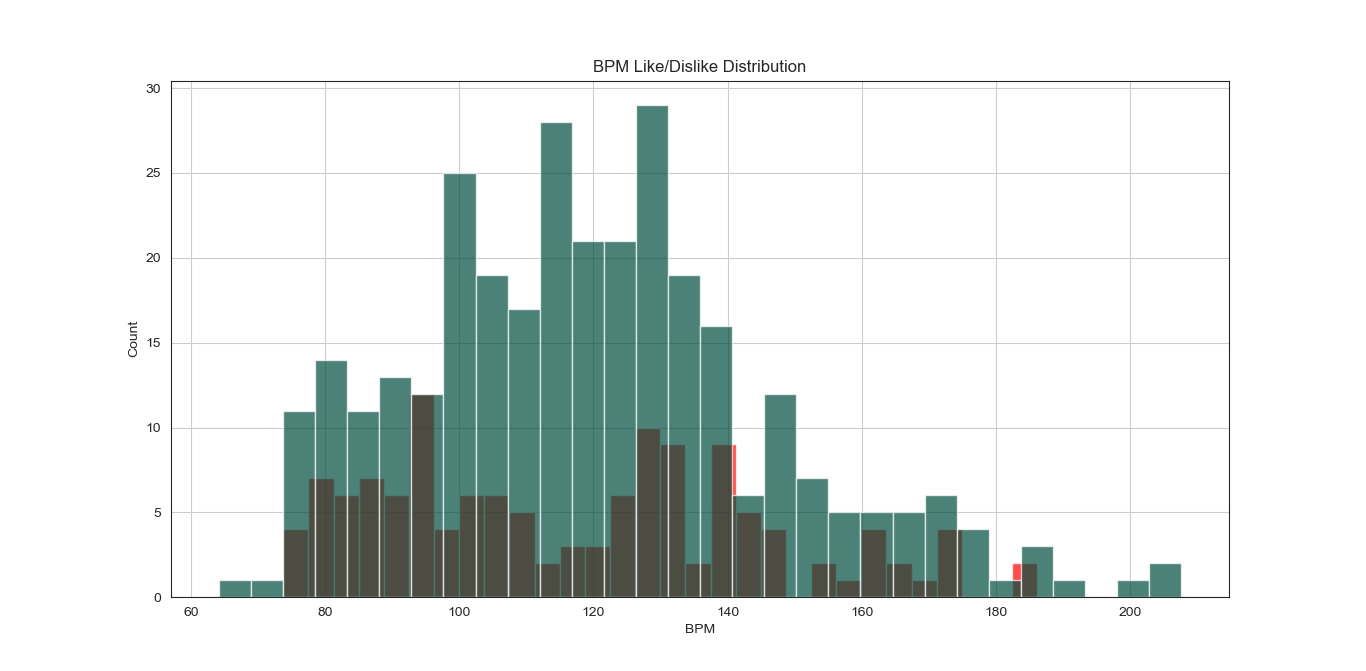
GitHub:

Legenda:

Verde: Playlist de músicas “Boas”

Vermelho: Playlist de músicas “Ruins”

Uma imagem contendo captura de tela

Descrição gerada automaticamenteUma imagem contendo texto

Descrição gerada automaticamente