Prediction on Streams of Spotify's Worldwide Daily Most Listened Songs

1. Introduction

Streaming is a technology used to deliver contents to computers and mobile devices over the internet, which is ubiquitous today. It transmits data- usually audio and video, as a continuous flow, which allows the recipients to begin to watch or listen almost immediately. Spotify, as an audio streaming platform, provides users millions of songs to enjoy without having to download first, which makes it so popular. Whenever we listen to a song for over 30 seconds on Spotify, it counts as one stream for the song. The number of streams that a song has gotten thus determines the ranking of the song and the proceed that its creators can receive. Just imagine, if we could predict steams for songs based on some other information that we could access ahead of time, like their name, their audio feature, and the date to be released, we would be able to help their creators learn in advance whether their songs would get popular. Creators could thus accordingly decide whether to make further changes to their songs, such as renaming songs, tweaking some audio features, or even changing the release dates. As a result, songs would have a greater chance to get popular and their creators could thus receive more proceeds. Besides, Spotify would be able to attract more users because of more popular songs coming up there. To turn imagination to reality, this project builds different machine learning models to predict streams for Spotify's songs.

2. Dataset Details

- Original Dataset

The original dataset consists of the daily 200 most listened songs in 53 countries from the first day of 2017 to the last day of 2018 by Spotify users. Below are descriptions of this dataset.

	Original Dataset
Name	data.csv
Shape	3441197 rows, 7 columns
Features	Position, Track Name, Artist, URL, Data, Region, (Streams)

- Extra Dataset

Although we have a giant original dataset, which contains more than 3 million rows of data, we might still not be able to build a good model with it, because, first of all, we will have to drop the Position column, since we are trying to predict songs' steams based on the information that we can access in advance, but Position is the ranking of songs, which won't be known until streams are available. Secondly, some features like URL actually have nothing to do with streams. Every different song has a unique url and it has no impact on the song's streams. Thus, having a giant dataset with few useful features is a big challenge for this project. To overcome the challenge, my solution is to expend more features for the original dataset by using URL to fetch audio features from Spotify Web API for all the tracks(songs).

An extra dataset which consists of audio features of all the tracks in the original dataset is thus obtained. Below are descriptions of the extra dataset.

Extra Dataset	
Name	audio_features.csv
Shape	3441197 rows, 18 columns
Feature	danceability, energy, key, loudness, mode, spechiness, acousticness, instrumentalness, liveness, valence, tempo, type, id, uri, track_href, analysis_url, duration_ms, time_signature

- Ultimate Dataset

All models in this project are built based on an ultimate dataset, which is obtained by merging the original dataset and the extra dataset. Below are descriptions of the ultimate dataset.

Ultimate Dataset	
Shape	3441197 rows, 25 columns
Feature	Position, Track Name, Artist, URL, Data, Region, (Streams), danceability, energy, key, loudness, mode, spechiness, acousticness, instrumentalness, liveness, valence, tempo, type, id, uri, track_href, analysis_url, duration_ms, time_signature

3. Methodology

Both **random forests** and **deep neural networks** are used respectively to built different models for the prediction of streams in this project. The random forests model is also used to calculate features' importance scores. The dataset is then updated by dropping those columns with little importance. The updated dataset is what finally fed to train the ultimate random forests model and the ultimate deep neural networks model.

4. Results

R-Squared and **Mean Squared Error** are the two measurements used to evaluate models.

Predicting streams is a regression problem. R-Squared & Mean Squared Error are just two straight forward measurements to tell us how good our model has solve the regression problem:
R-Squared measures how well observed outcomes are replicated by the model based on the proportion of total variation of outcomes explained by the model. It ranges from 0 to 1. Values closer to 1 are better; Mean Squared Error measures the average squared difference between the estimated values and the actual values. It is always non-negative. Values closer to 0 are better.

Results obtained:

	Random Forests Model	Neural Networks Model
R2 on Training Set	0.924591050522174	0.000594766740934061
MES on Training Set	2712325125.63189	35946819888.448900
R2 on Test Set	0.8692892867567440	0.0010583078515803400
MSE on Test Set	5351882934.220660	40901153867.4813