Dua Pria Ganteng Indonesia

Gardyan Priangga Akbar & Vincentius Gabriel Tandra

October 2021

1 Problem Introduction and Hypothesis

First we must introduce the problem that we are trying to solve. Our project will attempt to solve a problem in a video game known as The Elder Scrolls V: Skyrim Special Edition (SSE). Within SSE, there are optional community-created modifications to the game known as mods. These mods are published on a website where they can then be downloaded by other players and be used within the game. Nexus Mods is one of the largest and most popular modding website. In this website, mods are rated by points known as endorsements. Our project attempts to predict the characteristics of a mod that allow it to obtain the most amount of endorsements.

We will test with the hypothesis that the inclusion of adult content within certain categories of mods will allow it to gain more endorsements.

2 Dataset

Nexus Mods houses a wide variety of mods for various games. There are over 40,900 mods currently available for SSE alone. Therefore, we will be using this as the primary source for our dataset. We also believe this amount would be sufficient enough for us to analyze and create a predictive model from.

3 Data preparation and processing

To collect and build our dataset, we used the API endpoint provided by Nexus Mods to get the data of the mods (`https://app.swaggerhub.com/apis-docs/NexusMods/nexus-mods_public_api_params_in_form_data/1.0#/Mods/get_v1_games_game_domain_name_mods_id.json`. The attributes returned by the response body can be seen in Table 1.

Table 1: Attributes returned by the API endpoint and their descriptions

Attribute	Description
name	Name of the mod.
summary	A short summary of the mod given by the author. Usually 3-4 sentences long.
description	A full description of the mod. It explains the mod in full detail. Can contain a wide variety of information, such as images, videos, tables, and many more.
picture_url	URL of the image used for the mod preview.
uid	The mod's UID stored in the database.
game_id	ID of the game associated with the mod. For SSE, the value is 1704.
allow_rating	Boolean value, whether a mod can be endorsed.
domain_name	Name of the game associated with the game. In the case of SSE, it would be 'skyrimspecialedition'.
category_id	The ID of the category that describes what kind of mod it is (i.e. armor, weapons, new world). The ID for a mod with 'Miscellaneous' category would be 28, for example.
version	the latest version of the mod.
endorsement_count	The amount of endorsements the mod currently has.
created_timestamp	Timestamp of when the mod page was created.
created_time	Time when the mod page was created.
updated_timestamp	Timestamp of when the mod was last updated.
updated_time	Time of when the mod was last updated.
author	The name (or username) of the person that initially created the mod.
uploaded_by	The username of the person that uploaded the mod.
uploaded_users_profile_url	URL to the user profile of the person that uplaoded the mod.
contains_adult_content	Boolean value, determines whether a mod contains adult content.
status	The current status of the mod. There are 4 possible values • published • not_published • wastebinned • removed
available	Boolean value, determines whether a mod is available for public viewing.
users	Further information regarding the mod uploader.
endorsement	Further information regarding endorsement.

Next is the data cleaning and preprocessing. Before feeding our dataset to the predictor model for training, we need to handle NaN values as well as removing unnecessary attributes. This was done using Pandas library in Python. The list of attributes we decided to use for training are in Table 2. Additionally, we filtered out mods whose *allow_rating* and *status* attributes are *false* and not *published* respectively. Mods older than six months are also excluded, and data inside the *created_by* attribute are converted into a string and have the year removed. This is because they would not be a good representation of the dataset and may skew the result of our predictive model. At the end of the cleaning and filtering, we have around 30,000 rows of data to work with.

Attribute	NaN Handling Operation and Conditions
name	Discard row.
summary	Convert to empty string.
description	Convert to empty string.
category_id	Set value as 28 (Miscellaneous).
endorsement_count	Discard row.
created_time	Only accept mods older than 6 months (Discard the rest).
author	Convert to empty string.
uploaded_by	Convert to empty string.
contains_adult_content	Flag as false.

Table 2: Attributes to be used for prediction model training and how to treat their NaN values

4 Model and techniques

In order to realize our predictive model, we have decided to build our architecture based on ensemble architecture. Because certain architectures specializes in different formats of data, we decided to create four models tasked on analyzing different features from our dataset. The output for these four layers would then be fed into one last model tasked on determining the best output. Three of these models would be Recurrent Neural Networks (RNNs) tasked on analyzing *name*, *summary*, and *description* features individually. One decision tree model would be used to analyze the remaining features in tabular data format. The output model would have three hidden layers. All of these models would be trained for regression problem as to allow our predictive model to output a value that represents the projected amount of endorsement or likes to a mod for SSE.