Data Science Fundamentals

(CS 890ES)

Instructor: Dr. Alireza Manashty
Department of Computer Science
University of Regina
Winter 2020

User Guide for Analysts
on

"Games Description Analysis & Keyword Suggestions"

Submitted by:

Name	ID	Email	Role
Sai Keerthi Mettu	200416252	smj102@uregina.ca	BI Analyst Data Engineer Data Scientist Communicator
Somi Deepthi Nalamalpu	200412879	sny899@uregina.ca	

Table of Contents

USER GUIDE FOR ANALYSTS

INTRODUCTION:

Game description analysis and keyword suggestion, as the name portrays gives the users a bunch of appropriate and game-specific keywords that can enhance a game's credibility in terms of its draft and relevancy. Keywords tend to form a sentence and thereby conveying a reader with some information only by the existence of those phrases. When applying the same mechanism on game description, we can observe the descriptive sentences given below a game conveys a lot to its users in terms of its wording, organization, and relevancy. So working from a game developer's perspective, analyzing those keywords can interpret and help the gamer to understand the game and its features.

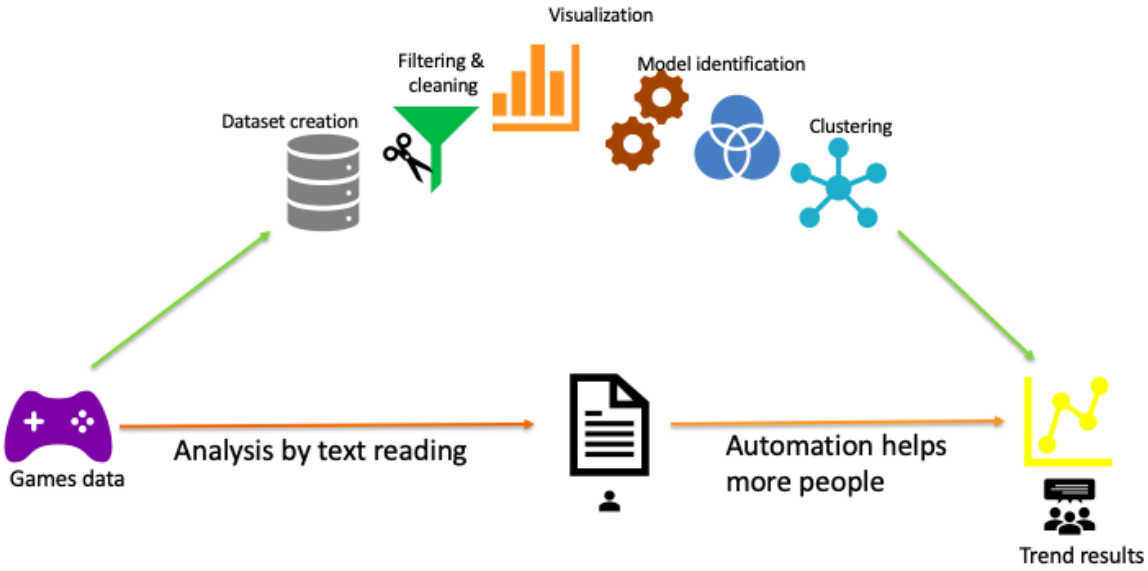
Game description can be a draft that tells what the game is about. Adding certain phrases help the user get a broad and accurate view of the game even before installing the game. So it is necessary to have relevant words available in it that seek user interest. If a game theme stating to be adventurous has puzzle word in its description won't be appropriate enough and may even confuse the gamer. So to avoid such instances, the model developed suggests a bunch of phrases that are appropriate to the game theme and can be embedded in the game description. We also looked into the fact that the description already possesses certain words that are relevant to the game. To be careful, and help the developer, we are also providing them with a percentage rank for those unique keywords that are in the description already, which shows, which phrases need to be kept and which can be changed by their percentage value.

PROBLEM STATEMENT:

Many factors may have contributed to the probability of a game being successful and reachable to a wide number of users.

There is a need to analyze the game description which intends to suggest proficient keywords that can be added to the existing or new description content to improve its readability.
The main idea of the project is to provide keywords/phrases to developers that constitute the idea/theme of the game which will tell them what the game is about in the description itself.

PROJECT DEVELOPMENT: High level diagram



DATA PREPARATION:

To attain all the features especially the game description, it was necessary to create our own dataset from a website. Play store is one of the many websites that provide an immense number of games and its feature aspects listed. So at first we scraped the data from different subcategories in the games section in play store and obtained a batch of games with different genre specifications. Using the batch file, we extracted more archived games using the game ID's and prepared our dataset into 2 CSV file.

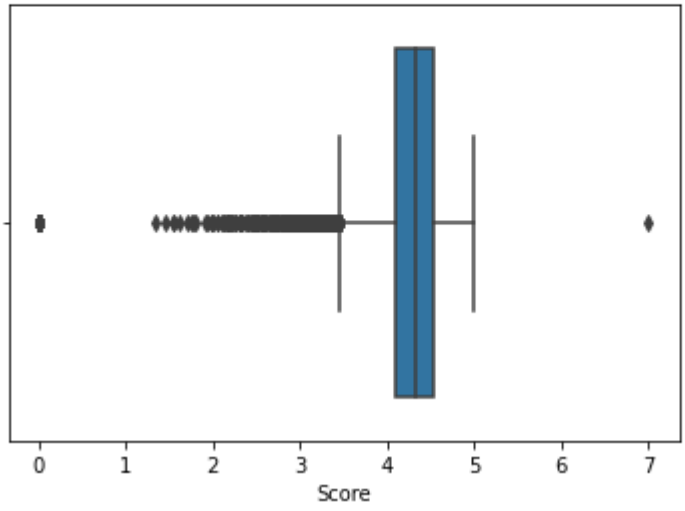
	Titlle	AppID	URL	Description	Summary	Installs	MinInstalls	Score	Rating	Reviews	Price	Free
0	Sniper 3D: Fun Offline Gun Shooting Games Free	com.fungames.sniper3d	https://play.google.com/store/apps/details?id=...	Call the best shooter, the guns are ready to a...	Fun cool free action shooting! The best online...	10,00,00,000	100000000	4.485727	12104202.0	4189270.0	0.0	True
1	Soul Knight	com.ChillyRoom.DungeonShooter	https://play.google.com/store/apps/details?id=...	"In a time of gun and sword, the magical stone...	Explore the dungeon, collect crazy weapons, do...	1,00,00,000	10000000	4.471023	874292.0	404858.0	0.0	True
2	Last Day on Earth: Survival	zombie.survival.craft.z	https://play.google.com/store/apps/details?id=...	The survival shooter Last Day on Earth is set ...	Survive in the zombie world	5,00,00,000	50000000	4.308150	3579297.0	1818068.0	0.0	True
3	Mobile Legends: Bang Bang	com.mobile.legends	https://play.google.com/store/apps/details?id=...	Join your friends in a brand new 5v5 MOBA show...	A thrilling 5v5 MOBA, now featuring a 99-playe...	10,00,00,000	100000000	4.385017	16983255.0	8428438.0	0.0	True

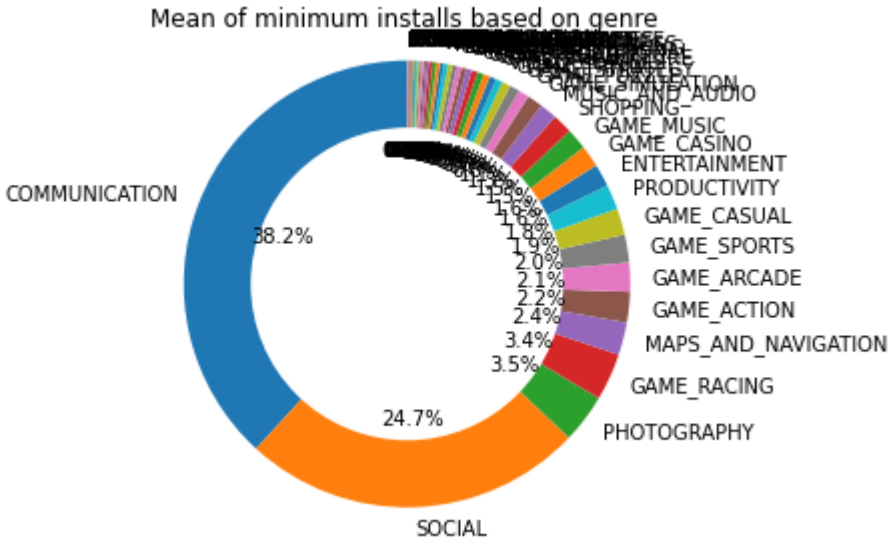
Once the data is extracted, Google Collaboratory helped us in doing the rest of the data science operations to solve the problem statement.

Data visualization is done on the most important features of the dataset that concern with the problem and to find the relationship between those features graphs and plots were designed and analyzed.

```
[18] #checking the anamolies in Score column

     sb.boxplot(x=data['Score'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f699017b160>
```





Various text analysis techniques like stopword, punctuations, emoji, emoticons, rarest words in the description along with most frequently used words were removed to clean the data. Lemmatizing is done to attain a root word from a similar context/ meaning words in a description. The lemmatization helps in cleaning the description further and narrowing the text data to its root words without missing the words with a different context. While stemming also performs the same operation but doesn't quite consider the context and meaning of words when grouped into a single root word and even provides a root word that is not available in the English dictionary.

TF-IDF IMPLEMENTATION:

The TF_IDF (Term Frequency-Inverse Document Frequency) vectorization is applied to the cleaned data to obtain important keywords and extract them as features. The TF-IDF vector performs its operation in 2 stages. The first step is the TF- where a term it creates a document term matrix that has a row per text message and column for a single unique term. The cells tend to represent weighting to identify how important a word is in an individual text, in this case, each game description. The next part of the formula works on how frequently a word occurs in all the texts, as of all the game descriptions. It divides no. game description in a log to the number of game descriptions this word appears. If a word is very frequent in a text then it is TF, and but infrequently then it is IDF. In general, the vectorizer helps to model out the important but seldom used words. The obtained keywords, as mentioned above were passed to the machine learning model to train the model with respect to the data.

```
# analyzer=clean_description
tfidf_vect=TfidfVectorizer()
tf_counts=tfidf_vect.fit_transform(data['Cleaned'])
print(tf_counts.shape)
print(tfidf_vect.vocabulary_.keys())
```

```
(15133, 26350)
dict_keys(['welcome', 'world', 'sniper', 'assassin', 'hold', 'back', 'achieve', 'perfect', 'shot', 'target', 'play', 'amazing'
```

Also, it is necessary to know which phrases in the existing game description and relevant and which are not relevant with respect to the game context. To provide the users with a bit of clarification and additional information, the importance of certain appropriate features were listed with a percentage value to get the difference. The keywords that are most likely to be kept in the description without removing them were indicated with good percentages and the least percentage values listed informs the user that they are not relevant enough to be kept in the description. This operation is performed by using the count vector and TF-IDF transformation on the cleaned text. The above transformation gives a TF-IDF score which is implemented on the game description and also on a package of game description, Genre Id, rating score and game title to observe the stats of these two implementations. When a user provides his game title, ratings, genre id the second implementation would be helpful as it gives the TF-IDF score by considering these features along with the game description. This method only provides the importance of certain phrases that need to be kept in the existing description and words that can be taken out.

MODEL BUILDING:

As the problem we are addressing is intending text data and obtaining a bunch of keywords that would be relevant to the game context, it ensures that clustering will help in forming a set of words into a cluster. Those clusters obtained can be tested by giving an unseen game description and checking in which cluster the description falls into. Once the cluster is obtained the words in it specify their relevancy with the game context. K-Means clustering algorithm is considered as it facilitates many iterations and also as it an unsupervised learning

technique the hidden patterns in the data can be checked. The TF-IDF vectorized data is fitted in the model to obtain eligible clusters and its predictions.

```
#give the description that has to be predicted
test_description='Escape to the world of farming, friends and fun! Go on farm adventures to collect rare goods and craft new recipes. Raise animals a
```

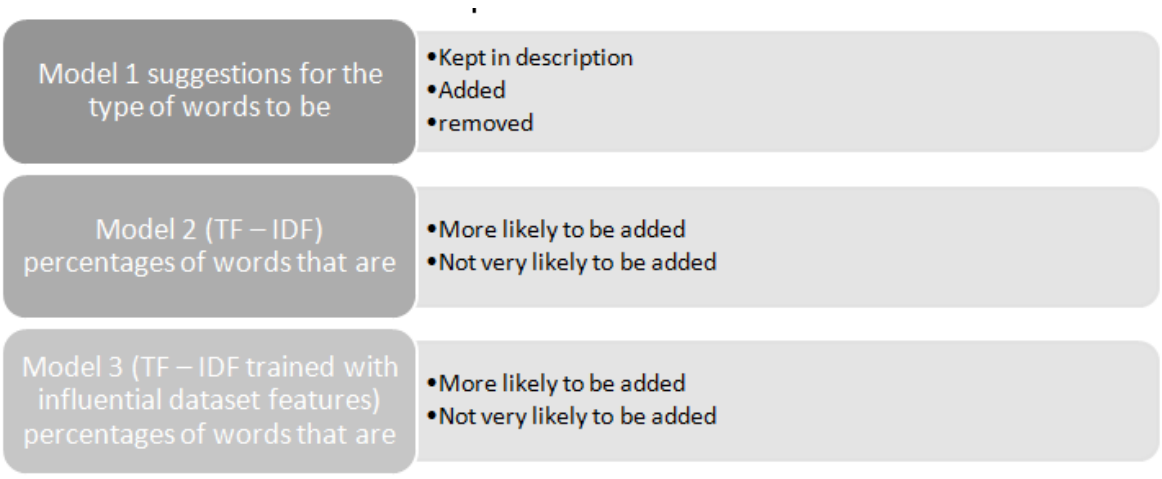
Result for the above given test game description is displayed below,

```
all_suggestions(test_description)

Predicted Cluster :  [6]

Words that are must to be kept in description :  ['game', 'play', 'free', 'fun', 'new']

Words to be Added (Suggestion from Clustering model) : ['learn', 'learning', 'music', 'color', 'time', 'use', 'help', 'make', 'application',

Words not to be Added (Suggestion from Clustering model) : ['zombie', 'shooting', 'sniper', 'dead', 'game', 'survival', 'shooter', 'gun', 'a

TF-IDF score determinations from trained Game Descriptions
Keywords in the given description that are more likely interested to keep:

friends:  46.6
farm:  40.7
farming:  32.5
anytime:  26.2
animals:  26.2
co:  25.0
adventures:  22.4
anonymous:  19.9
raise:  13.9
play:  13.9
connected:  13.8
trade:  13.3

Keywords in the given description that are not very likely interested to keep:
```

OPERATIONALIZE AND COMMUNICATION:

As the training data is applied and the model is developed, it is necessary to test the model with unseen data. So when provided a new game description, the model suggests the unique keywords that can be added to the description. The keywords obtained will be appropriate to the theme and context of the game. We also extracted the words that don't quite go along with the given game description. That is the words that should be added or applied in the game description. The following suggestions can help the developed to modify his/her game description with appropriate and relevant phrases and also help in getting suggesting those words that don't cooperate with the game context. It can enhance the descriptive content of the game making it readable and reachable to more number of users.

RESULTS:



The users get a package of keywords that can be added to his/her description and keywords that shouldn't be embedded in their description suggested by the model developed. Based on the existing description given, the user can check which phrases are most likely to be kept and which to be replaced by the TF-IDF score. This can ensure that user keeps the phrases that are important and relevant to their game and replace the less and unimportant words with the keywords suggested by the model also considering not to add those irrelevant words when working on modifying/ updating their game description.

```
TF-IDF score determinations from trained Game Descriptions
Keywords in the given description that are more likely interested to keep:

friends:  46.6
farm:  40.7
farming:  32.5
anytime:  26.2
animals:  26.2
co:  25.0
adventures:  22.4
anonymous:  19.9
raise:  13.9
play:  13.9
connected:  13.8
trade:  13.3

Keywords in the given description that are not very likely interested to keep:

even:  7.5
mode:  7.7
collect:  7.7
go:  7.9
join:  8.1
share:  9.1
popular:  10.1
anywhere:  10.2
```

```
TF-IDF score determinations from trained Game Descriptions+Genre+Title+Score/Rating
Keywords in the given description that are more likely interested to keep based on the Genre and Score Ratings:

farm:  38.6
friends:  36.6
farming:  30.7
op:  26.6
internet:  24.8
anytime:  24.8
recipes:  23.7
co:  23.3
animals:  21.2
adventures:  19.6
anonymous:  18.8
raise:  13.2
play:  13.1
connected:  13.1
trade:  12.6

Keywords in the given description that are not very likely interested to keep based on the Genre and Score Ratings:
even:  7.1
collect:  7.3
mode:  7.3
go:  7.5
join:  7.7
share:  8.7
popular:  9.6
anywhere:  9.7
```

The images above give the results of TF-IDF score for the test game description provided above.

LIMITATIONS AND FUTURE ADVANCEMENTS:

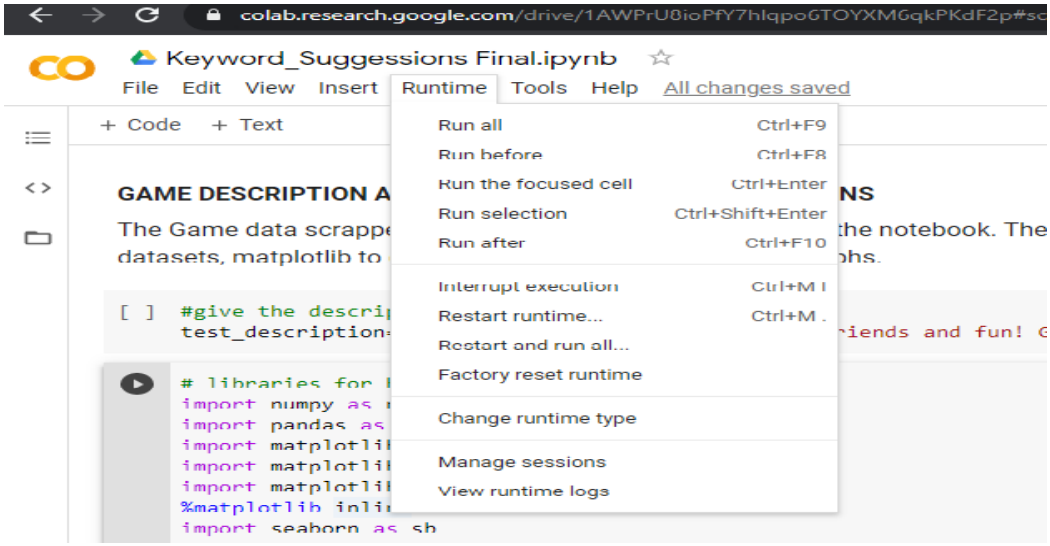
At present, the project is limited to local runtime executions, Use needs to download the data and store it in Google drive and import the Keyword suggestions notebook to Google Collaboratory to execute the model and obtain its results. Once the data and notebook are gathered users can host it on their colab and as all the import functionality of data from Google drive and execution methodology is listed in the notebook, it will be easy for the user to get the prediction for his own game description.

Future advancements can help in obtaining the predictions to a given game description on a web server instead of having to download the data and execute the colab notebook. We are trying to host the model developed on the web to make it functional as a global website and get the results on just providing the input without any hassle.

STEPS TO BE PERFORMED:

- User needs to download theGaming.csv and BatchFile1.csv and store it their Google Drive.
- The Keyword suggestions.ipynb available in the GitHub can be opened with Colab (Google Collaboratory).
- The user needs to sign in his Google account to get the runtime and working environment in Google colab or create an account if it doesn't exist and then connect to a local runtime.

- Once the Keyword suggestions.ipynb is imported on to your colab environment allot the runtime and check if the drive path is correct. The path specified should be the same as the one where the dataset is tore. Make sure it is accurate.
- There exist a text description variable where user can give their game description within the inverted commas. The variable takes the user input and provides it to the model as test data.
- Then the user can execute the code by opting for "Run ALL" at runtime to attain the suggestions ignoring the remaining methodology or can run each cell to view and analyze the output of each technique and operation performed on the data.



Git Hub Link to access the project files:

https://github.com/KeerthiMettu/Games-Analysis

https://github.com/somideepthi/Game-description-analysis-and-keyword-suggestion

Access links for our 2 datasets from Google drive:

1. BatchFile1.csv: https://drive.google.com/open?id=18cRi8kB54mGyih2ufc1pPr6hHvxPDbJT
2. originalgaming.csv:
3. https://drive.google.com/file/d/10-E7sLMq1P2E8Ubv0IrlCugyXrsyfoHw/view