STEAM®

Practical exercise 3

In this third practice session we will work with clustering. For this practice session we will use a game dataset. The provided dataset is a group of distinct tables, each one containing distinct information about the games on the steam platform and stored in a distinct csv file.

This dataset contains data from video games published on the Steam distribution platform. Each dataset table provides detailed information for each product on the platform, this data was collected using the Steam API and from SteamSpy.

Description on the provided data:

"steam.csv" data set, has basic game data

- appid: Unique identifier for each product on Steam
- name: Name of the game
- release_date: The date the game was issued on the platform
- english: Boolean indicating if the game is in English
- developer: game developer name
- publisher: game publisher name
- required_age: minimum recommended age to play the game
- categories: Main categories of the game
- genres: Main genres of the game
- achievements: Number of achievements in the game
- positive_ratings: Number of positive ratings
- negative_ratings: Number of negative ratings
- average_playtime: Time the users played the game in average
- median_playtime: Time the users played the game in median
- owners: Number of owners of the game in the steam platform
- price: Price of the game in the steam platform

"steamspy_tag_data.csv" data set, collects information from the SteamSpy site related with game user defined tags/features.

- appid: Unique identifier for each product on Steam
- For each tag or feature (i.e. "sports", "cars", "shooter", etc.) on Steam, a natural number is provided that indicates how relevant the tag is to describe the product. These tags are provided by the users not the developers.

“steam_support_info.csv” data set, collects information related with game customer support.

- steam_appid: Unique identifier “appid” for each product on Steam
- website: Website of the developer.
- support_url: URL provided for support.
- support_email: e-mail provided for support.

“steam_requirements_data.csv” data set, collects information related with game hardware and soft requirements.

- steam_appid: Unique identifier “appid” for each product on Steam
- pc_requirements: Windows users’ requirements to play the game
- mac_requirements: Mac users’ requirements to play the game

- linux_requirements: Linux users' requirements to play the game
- minimum: Minimum required hardware to play the game
- recommended: Recommended required hardware to play the game

"steam_media_data.csv" data set, collects information related with game steam site page visualization.

- steam_appid: Unique identifier "appid" for each product on Steam
- header_image: url of the header image
- screenshots: url of game play or publicity screenshots
- background: url of the image to be displayed on the page background
- movies: url of game play or publicity videos

"steam_description_data.csv" data set, collects information related with game descriptions to be shown in distinct places.

- steam_appid: Unique identifier "appid" for each product on Steam detailed_description: Text describing the game in a detailed version.
- about_the_game: Text description of the game characteristics
- short_description: Short text describing the game

Follow next steps / questions and answer them in your Moodle submission. Please submit the python code you have used.

1) Load the dataset in **"steam.csv".** We want to perform a clustering using k-means, scikit-learn only implements Euclidean distance so only accepts numeric attributes, transform all the attributes to numeric. Preprocess the data in other to convert categoric and non-numeric attributes to numeric. (1 point)

For "release_date" you can use the following code.

```
data['release_date']=pd.to_datetime(data['release_date']).astype('int64')
```

2) Fill the NaN values in the dataset. Use A KNN algorithm. Was a good decision to fill these values using KNN or is better to give them a fix value or just delete them? What will happen in each case? (3 points)

3) Clustering needs data to be scaled. If not variables with higher variability will be favoured. Scale the data. (1 point)

4) Fit a k-means clustering with 2 clusters and perform the prediction. Retrieve the cluster centers (cluster_centers_ attribute on your fitted k-means). Try to understand what kind of information each cluster has. (2 points)

5) Set yourself a clustering goal. Do you see any not useful variable? delete it. Try other cluster numbers, preprocessings, variable selections, add weights add information in other files... (do what you want to achieve a goal). Try to explain what are you expecting to achieve, what are you doing and what you achieve. (3 points)

Submission. As results will change from one execution to the other due to randomness. You have 14 days to finish and upload the task to Moodle. This practice can be done in pairs or individually. Please submit the python code, commented.