Projet de data visualisation

Rebecca Leygonie
19000002

Explications de la préparation des données

Données Apple Store

Lien de téléchargement des données : https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps

7197 lignes, 16 colonnes

Explication des variables :

Id : App ID
track_name : nom de l'application
size_bytes : taille (en Bytes)
currency : monnaie
price : prix
rating_count_tot : total des notes (pour toutes les versions de l'application)
rating_count_ver : total des notes (pour la version actuelle de l'app)
user_rating : note moyenne donnée par les utilisateurs (pour toutes les versions de l'application)
user_rating_ver : note moyenne donnée par les utilisateur (pour la version actuelle de l'app)
ver : version de l'application
cont_rating : content rating
prime_genre : catégorie
sup_devices.num : nombre de supports sur lesquels l'application est téléchargeable/utilisable
ipadSc_urls.num : nombre de screenshots de démonstration présents sur la page de téléchargement de l'application.
lang.num : nombre de langues proposées par l'applications
vpp_lic : vpp device based licensing enabled

1. Suppression des variables inutiles

- Id -> ne représente aucune information
- Currency -> contient seulement la valeur « USD »
- vpp_lic -> contient toujours la même valeur
- rating count ver
- user rating ver

2. Renommage de variable

sup_devices.num -> Nb_dispositifs_de_support
iPadSc_urls.num -> Nb_captures d'écran
lang_num -> nb Lang supportés

3. Création de variables

Pour que les valeurs de la variable ***rating*** soient interprétables, il faut exclure le cas ou la note d'une application est 0 car l'application n'a jamais été notée. Pour cela, nous créons une variable ***True_rating*** qui prend en compte ce cas. C'est avec cette variable que nous travaillerons pas la suite.
Nous remarquons d'ailleurs qu'avec cette nouvelle variable, aucune note est à 0. Ce qui signifie que les applications dont les notes étaient 0 sont seulement celles qui n'ont jamais été noté.
En réalité, sur Apple Store, il est impossible de mettre une note inférieure à 1 à une application.

```
True_rating = IF [user_rating] != 0
THEN [user_rating]
ELSEIF [rating_count_tot] != 0 THEN [user_rating]
ELSE NULL
END
```

Dans certains cas, il est plus pratique, visuellement parlant, de voir les notes par classe de notes. Nous créons donc une variable ***Rating classe*** qui regroupe les notes en classe de note allant de 0,5 à 5 avec un pas de 0,5.

Objectif de l'analyse des données :

- Analyser la variation des notes moyennes des applications en fonction du prix
- Analyser la variation des notes moyennes des applications en fonction de la catégorie
- Analyser la variation du prix des applications en fonction de la catégorie
- Analyser la variation de la taille des applications par catégorie
- Analyser les variations du nombre de langues proposées par une application en fonction de la catégorie à laquelle elle appartient
- Comparer les notes moyennes en fonction du nombre de langues proposées et en fonction du nombre de captures d'écrans
- Analyser la variation des notes moyennes en fonction du nombre de dispositifs supportés

Données Google PlayStore :

Lien de téléchargement des données : https://www.kaggle.com/lava18/google-play-store-apps

10 840 lignes, 13 colonnes

Explication des variables :

App : nom de l'application
Category : catégorie
Rating : moyenne des notes données par les utilisateurs
Reviews : nombre de commentaires donnés par les utilisateurs
Size : taille
Installs : nombre de fois ou l'application a été téléchargé/installé
Type : Payant ou gratuit
Price : prix
Content_rating : age du public visé par l'application (Children/ Mature 21+ /Adult)
Genres : une application peut appartenir à plusieurs genres.
Last_updated : date du dernier téléchargement de l'application
Current_ver : version actuelle de l'application
Android_ver : version requise d'Android pour pouvoir télécharger l'app

1. Création de variables

Création de la variable ***Rating_float*** car les notes sont écrites avec un « . » donc tableau ne l'interprète pas comme un float. De plus, nous mettons les notes sur la même échelle que celles d'Apple Store pour pouvoir comparer les résultats de nos analyses.

```
IF FLOAT([Rating])/0.5=int(FLOAT([Rating])/0.5)
THEN FLOAT([Rating])
ELSEIF  FLOAT([Rating])<0 then int(FLOAT([Rating])/0.5)*0.5
ELSE
int(FLOAT([Rating])/0.5+1)*0.5
END
```

Création de la variable ***Install_numeric*** car les valeurs de installs sont de la forme « String » : « 10 0000+ »

```
FLOAT(REPLACE([Installs],'+',''))
```

Même chose pour price.

price_numeric :

```
FLOAT(IF STARTSWITH([Price],'$')
THEN REPLACE([Price],'$','')
ELSE [Price]
END)
```

Rating_float classe : classe de note (0,5 ; 1; 1,5 ; …)

2 . Suppression de valeur aberrante

Même après avoir normalisé les notes, il y a une note à 19. On supprime cette ligne.

3. Changement de type des variables

Changement du type de ***Last_update*** en Date

Objectif de l'analyse des données :

- Analyser les notes moyennes des applications en fonction de la catégorie à laquelle elles appartiennent.
- Analyser les notes moyennes des applications en fonction du nombre d'installations qu'elles ont à leur actif
- Analyser l'évolution du nombre d'avis et de commentaires laissés par les utilisateurs au fur et à mesure des années
- Analyser la variation du prix moyen d'une application en fonction de la tranche d'âge du public qu'elle vise
- Analyser la variation de la note moyenne d'une application en fonction de la tranche d'âge du public qu'elle vise

Jointure des deux tables

La table Google Playstore contient des applications qui n'existent pas dans Apple Store. En effet, n'importe quelle application peut être mise sur Google Playstore alors que pour ajouter une application sur Apple Store, il faut passer par des étapes de validations de la part d'Apple.

Afin d'analyser la jointure de ces deux tables, il est préférable d'en créer deux :

1. googleplaystore JOIN (externe) Apple Store

Dont toutes les lignes des deux tables sont gardées et la jointure se fait sur les variables ***App*** pour Google Play Store et ***trackname*** pour Apple store.

-> 17 710 lignes

Objectif de l'analyse des données :

- Comparaison des notes moyennes par catégorie et par support de téléchargement (Google PlayStore, Apple Store)
- Comparer la variation des notes moyennes des applications de Google PlayStore et Apple Store
- Comparer l'évolution des notes moyennes des applications en fonction de leur dernier téléchargement
- Comparer les notes moyennes des applications en fonction du public qu'elles visent

2. googleplaystore JOIN (interne) Apple Store

Dont seulement les lignes dont ***App*** == ***trackname*** sont conservées

Suppression de valeurs non interprétables :

Dans la colonne « Review » : on conserve les valeurs non-null uniquement
Dans la colonne Rating : exclut Nan

-> 553 lignes

Objectif de l'analyse des données :

- Comparer la variation des notes moyennes des applications de Google PlayStore et Apple Store en fonction de leur catégorie
- Analyser les variations de la note moyenne des applications Apple Store (qui sont aussi dans Google PlayStore) en fonction du nombre moyen d'installations
- Comparer, entre Google PlayStore et Apple Store, la variation du prix en fonction de la catégorie a laquelle une application appartient.
- Comparer, entre Google PlayStore et Apple Store, la variation de la note moyenne des applications en fonction du prix