Dédicaces

Je dédie ce modeste travail à : Mes chers parents, Je mets entre vos mains le fruit de votre amour, de votre tendresse, de vos sacrifices, et vos encouragements, tout au long de mes études.

Ma chère mère, Saliha toutes les expressions ne peuvent exprimer mes sentiments d'amour et de respect vers vous. Vous êtes pour moi la meilleure des mères, qui a consacré sa vie aux bonheurs et réussite de ses enfants. Vous m'avez donné la vie et vous avez guidé tous mes pas, vous êtes toujours à mes côtés pour me motiver, me pousser à devenir ce que je suis aujourd'hui. En témoignage de vos fatigues, vos efforts et vos sacrifices, je vous dédie ce travail pour avouer ma reconnaissance et ma profonde estime. Espérant être votre source de bonheur et de fierté pour ce jour attendu. Puisse Dieu vous accorder santé et longue vie.

À ma sœur Ichraf et mon frère Ahmed, je ne peux exprimer à travers ces lignes tous mes sentiments d'amour envers vous. Que l'amour et la fraternité nous unissent à jamais.

À tous mes amis qui ont toujours fait confiance en moi et en mes capacités à faire mieux. Je ne pourrais être que reconnaissante envers tout ce que vous avez fait pour moi.

Randa maghraoui

Remerciements

Au terme de mon projet de fin d’études, j’aimerais exprimer mes sentiments de gratitude envers toutes les personnes qui par leur présence, leur soutien, leur disponibilité et leurs conseils, j’ai pu accomplir ce projet.

J’adresse aussi mes profonds et sincères remerciements à mon enseignant et encadrant Dr. Faouzi Mhamdi qui a accepté de diriger ce travail, pour ses conseils précieux, son encouragement continu, sa haute disponibilité, sa patience énorme et le temps précieux qu’elle m’a réservé tout au long de la réalisation de mon projet.

Je tiens d’autre part à remercier les membres du jury pour vouloir j'accorde de leur temps précieux pour commenter, discuter et juger notre travail.

Enfin je ne peux pas achever ce mémoire sans exprimer ma gratitude à tous les professeurs d’ISLAIB pour leur soutien et leur assistance tout au long de mon étude universitaire.

Sommaire

Liste des figures

Liste des tables

Introduction générale

Google Play Store est englouti avec quelques milliers nouvelles applications régulièrement avec un nombre de designers travaillant librement ou au contraire dans un groupe pour les faire réussir, avec l'énorme défi partout dans le monde. Depuis la plupart des Play Store les applications sont gratuites, le modèle de revenu est très obscur et inaccessible en ce qui concerne la façon dont l'application est téléchargée, les publicités et les adhésions contribuent à la réalisation d'une candidature.

De cette façon, la prospérité d'une application est normalement dictée par la quantité d'installation de l'application et du client appréciations qu'il a acquises au cours de sa vie au lieu d'un revenu est créé.

Les évaluations des applications sont des commentaires fournis volontairement par les utilisateurs et fonction importants critères d'évaluation des applications. Cependant, ces évaluations peuvent souvent être biaisées en raison de votes insuffisants ou manquants.

Aditionellement, des différences significatives sont observées entre les notes numériques et les avis des utilisateurs. Le projet actuel, vise à prédire les évaluations d'Applications Google Play Store en utilisant des algorithmes d'apprentissage automatique. Nous avons essayé d'effectuer une analyse de données et une prédiction dans l'ensemble de données de l'application Google Play Store que j'ai collecté de la plate forme *Kaggle*. En utilisant des algorithmes d'apprentissage automatique, j'ai essayé pour découvrir les relations entre les différents attributs présents dans mon ensemble de données, par exemple quelle application est gratuite ou payante, environ les avis des utilisateurs et la notation de l'application.

Les développeurs mobiles ne trouvent pas des moyens pour prédire le résultat de leur application, en termes de notes d'utilisateurs sur le Google Play store, avant même le lancement de leur propre application. De plus, les utilisateurs ne sont pas toujours formels lors de la rédaction des critiques.

Les expressions de sentiment dans les avis des utilisateurs varient diversement. Cette diversité peut être des mots simples comme génial, digne, etc. Ainsi que dans le cas de phrases avec plusieurs mots comme perdre du temps, super à utiliser, etc.

Le premier objectif de cette étude est d'avoir des informations exploitables peuvent être tirées pour que les développeurs travaillent et capturent le marché Androïde.

Maintenant on va présenter notre solution. Sur la base de nos modèles de prédiction, nous sommes en mesure de comprendre cela pour les développeurs mobiles anxieux. Il leur suffirait de saisir le type de leur application (gratuite ou payante), le prix de leur application (si payée), le genre de leur application (Art & Design, Créativité, etc.), Evaluation du contenu (Tout le monde, Ados, etc.), taille de leur application (en octets) via notre tableau de bord d'entrée et prédiction de sentiment sur Google Play Store.

Ce travail est organisé en 3 chapitres :

Dans le premier chapitre, on va détailler l'apprentissage automatique et ses différents types. Cela permettra de définir par la suite les différents algorithmes existants et leurs usages, ainsi que les différents types d'évaluations.

Le deuxième chapitre introduira deux grandes parties, la première partie consiste à prédire le *rating* et introduire la méthodologie suivante :

- Acquisition de données
- Prétraitement de données (nettoyage, encodage, ...)
- Traitement : nous appliquerons des algorithmes de régression (régression linéaire, support vector régression, arbre de discision de régression)
- Evaluations des modèles : nous choisirons les métriques de régression (*Mae*, *MSE et* R^2*score*).

Dans la deuxième partie on focalisera sur la prédiction de sentiment sur Google Play Store, nous avons tout d'abord passerons par le nettoyage de données, ainsi que les algorithmes de prédiction. Ensuite on évaluera notre modèle par des mesures calculées à partir de la matrice de confusion tel que rappel, précision et score f1.

Le troisième chapitre présentera les différentes bibliothèques utilisées ainsi que l'intégration du Framework *flask et dash*, enfin nous présenterons notre interface avec la description de chaqu'une.

Enfin nous présenterons la conclusion et les perspectives de développement de ce travail.

Chapitre 1 : État de l'art

Introduction

Dans ce chapitre on va voire les types d'apprentissage automatique et les différents algorithmes et enfin on va expliquer quelque méthode d'évaluation.

1. Apprentissage Automatique

L'apprentissage automatique consiste à laisser l'ordinateur apprendre quel calcul effectuer, plutôt que de lui donner ce calcul (c'est-à-dire le programmer de façon explicite) [1].

1.1 Apprentissage Supervisé

Dans le cas de l'apprentissage supervisé, les données utilisées pour l'entraînement sont déjà « étiquetées » par conséquent, le modèle d'apprentissage automatique sait déjà ce qu'il doit chercher (motif, élément, …) dans ces données. À la fin de l'apprentissage, le modèle ainsi entraîné sera capable de retrouver les mêmes éléments sur des données non étiquetées.

Parmi les algorithmes supervisés, on distingue les algorithmes de classification (prédictions non numériques) et les algorithmes de régression (prédictions numériques). En fonction du problème à résoudre, on utilisera l'un de ces deux archétypes [2].

1.2 Apprentissage non supervisé

L'apprentissage non supervisé, au contraire, consiste à entraîner le modèle sur des données sans étiquette. La machine parcourt les données sans aucun indice, et tente d'y découvrir des motifs ou des tendances récurrentes. Cette approche est couramment utilisée dans certains domaines, comme le cyber sécurité.

Parmi les modèles non supervisés, on distingue les algorithmes de clustering (pour trouver des groupes d'objets similaires), d'association (pour trouver des liens entre des objets) et de réduction dimensionnelle (pour choisir ou extraire des caractéristiques)[2].

1.3 Apprentissage par renforcement

Une troisième approche est celle de l'apprentissage par renforcement**.** Dans ce cas de figure, l'algorithme apprend en essayant encore et encore d'atteindre un objectif précis. Il pourra essayer toutes sortes de techniques pour y parvenir. Le modèle est récompensé s'il s'approche du but, ou pénalisé s'il échoue [3].

2. Algorithmes d’apprentissage supervisé et leurs applications

Il existe plusieurs applications que nous utilisons ces algorithmes dans notre projet.

2.1 Régression linéaire

La régression linéaire est une modélisation linéaire qui permet d'établir des estimations dans le futur à partir d'informations provenant du passé. Dans ce modèle de régression linéaire, on a plusieurs variables dont une est explicative et les autres sont à expliquées. Cet outil est utilisé pour les analyses techniques boursières mais aussi pour la gestion de budgets. Elle est souvent calculée avec la méthode des moindres carrés qui permet de réduire les erreurs en ajoutant de l’information [4].

2.2 Support Vector Régression

Le temps de trajet est une mesure fondamentale du transport. Une prévision précise du temps de trajet est également cruciale pour le développement de systèmes de transport intelligents et de systèmes d'information avancés pour les voyageurs. Nous appliquons la Régression Vectorielle de Support (SVR) pour la prédiction du temps de trajet et comparons ses résultats à d'autres méthodes en utilisant des données de trafic routier réel. Étant donné que les machines vectorielles de support ont une plus grande capacité de généralisation et garantissent des minima globaux pour des données d'entraînement données, on pense que la SVR fonctionnera bien pour l'analyse des séries chronologiques. Par rapport à d'autres prédicteurs de base, nos résultats montrent que le prédicateur SVR peut réduire de manière significative à la fois les erreurs moyennes relatives et les erreurs quadratiques moyennes des temps de trajet prévus [5].

2.3 Régresseur d'arbre de décision

En informatique, l’apprentissage par arbre de décision utilise un arbre de décision (en tant que modèle prédictif) pour passer des observations relatives à un élément (représenté dans les branches) à des conclusions sur la valeur cible de l'élément (représentée dans les feuilles).

C'est l'une des approches de modélisation prédictive utilisées dans les statistiques, l'exploration de données et l’apprentissage automatique. Les modèles d'arborescence dans lesquels la variable cible peut prendre un ensemble discret de valeurs sont appelés des arbres de classification. Dans ces arborescences, les feuilles représentent les étiquettes de classe et les branches représentent les conjonctions d'entités menant à ces étiquettes de classe. Les

arbres de décision où la variable cible peut prendre des valeurs continues (généralement des nombres réels) sont appelés des arbres de régression [6].

2.4 Régression Logistique

La régression logistique effectue la classification binaire en utilisant une fonction sigmoïde comme hypothèse, qui est donnée par:

$$y = \sigma(z) = 1/1 + e^{-z} = 1/1 + \exp(-z)(eq.1)$$

[0,1] qui est exactement ce que nous voulons pour une probabilité. Parce que c'est presque linéaire autour 0 mais s'aplatit vers les extrémités, il a tendance à écraser les valeurs aberrantes vers 0 ou 1. Et il est différenciable, ce qui, comme nous le verrons dans la section 5.8, sera pratique pour l'apprentissage.

Nous y sommes presque. Si nous appliquons le sigmoïde à la somme des caractéristiques pondérées, nous obtenons un nombre compris entre 0 et 1. Pour en faire une probabilité, il suffit de faire sûr que les deux cas, p (y = 1) et p (y = 0), totalisent 1. Nous pouvons le faire comme suit:

$$p(y = 1) = \sigma(w.x + b) = \frac{1}{1 + \exp\big(-(w.x + b)\big)}$$

$$p(y = 0) = 1 - \sigma(w.x + b) = 1 - \frac{1}{1+\exp(-(w.x+b))} = \frac{\exp(-(w.x+b))}{1+\exp(-w.x+b))}(eq.2)$$

Nous avons maintenant un algorithme qui, étant donné une instance x, calcule la probabilité P (y = 1 | x). Comment prenons-nous une décision ? Pour une instance de test x, on dit oui si la probabilité P (y = 1 | x) est supérieure à 0,5, et aucune autre. Nous appelons 0 ,5 la décision frontière [7]:

$$\hat{y} = \{1\ if\ P(y = 1)|x) > 0.5|0\ otherwise\}$$

2.5 Classifieur Naïve Bayes

L'algorithme Multinomial *Naïve Bayes* est utile dans le cas où les fonctionnalités. Ont des valeurs discrètes en raison de sa simplicité et de sa facilité de mise en œuvre. Cet algorithme est basé sur une forte hypothèse est conditionnellement indépendant étant donné, qui est aussi connu sous le nom d'hypothèse *Naïve Bayes* (NB) [8].

Après ajustement des paramètres, la prédiction sur un nouvel échantillon avec les attributs x peut être obtenue comme suit:

$$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x)} = \frac{(\sum_{i=1}^{n} p(xi|y=1))p(y=1)}{(\sum_{i=1}^{n} p(xi|y=1))p(y=1)+(\sum_{i=1}^{n} p(xi|y=0)p(y=0)} \text{(eq.3)}$$

2.6 Support vector machine (SVM)

«Support Vector Machine» (SVM) est un algorithme d'apprentissage automatique supervisé qui peut être utilisé à la fois pour les défis de classification ou de régression. Cependant, il est principalement utilisé dans les problèmes de classification. Dans l'algorithme SVM, nous traçons chaque élément de donné comme un point dans un espace à n dimensions (où n'est le nombre d'entités dont vous disposez), la valeur de chaque entité étant la valeur d'une coordonnée particulière. Ensuite, nous effectuons la classification en trouvant l'hyper-plan qui différencie très bien les deux classes (regardez l'instantané ci-dessous) [9].

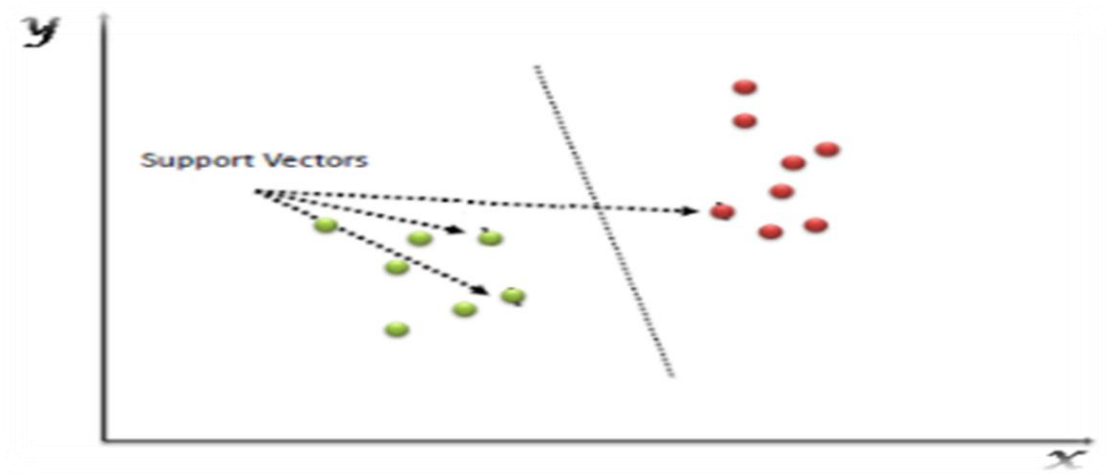


Figure 1: SVM graphe

3. les mesures d'évaluation du modèle

3.1 Importance de l'évaluation du modèle

Être capable de mesurer correctement les performances d'un modèle d'apprentissage automatique est une compétence essentielle pour chaque praticien de l'apprentissage automatique. Afin d'évaluer les performances du modèle, nous utilisons des métriques d'évaluation.

Selon le type de problème que nous voulons résoudre, nous pouvons effectuer une classification (où une variable catégorielle est prédite) ou une régression (où un nombre

réel est prédit) afin de le résoudre. Heureusement, la bibliothèque *scikit-learn* nous permet de créer facilement des régressions, sans avoir à gérer la théorie mathématique sous-jacente.

3.2 Choix de la métrique de la performance

3.2.1 Pour les problèmes de régression

Les métriques de régression sont différentes des métriques de classification car nous prédisons une quantité continue. En outre, la régression a généralement des besoins d'évaluation plus simples que la classification.

Les mesures fondamentales utilisées pour évaluer le modèle de régression sont présentées ci-dessous.

- **Erreur absolue moyenne**

L'erreur absolue moyenne (*MAE*) est l'une des mesures les plus courantes utilisées pour calculer l'erreur de prédiction du modèle. L'erreur de prédiction d'une seule ligne de données est:$PredictionError = ActualValue - PredictedValue$

Nous devons calculer les erreurs de prédiction pour chaque ligne de données, obtenir leur valeur absolue, puis trouver la moyenne de toutes les erreurs de prédiction absolues.

MAE est donnée par la formule suivante:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |yi - \hat{y}i| \text{(eq.4)}$$

Où y_i représente la valeur prédite de $\hat{y}_i$ [10].

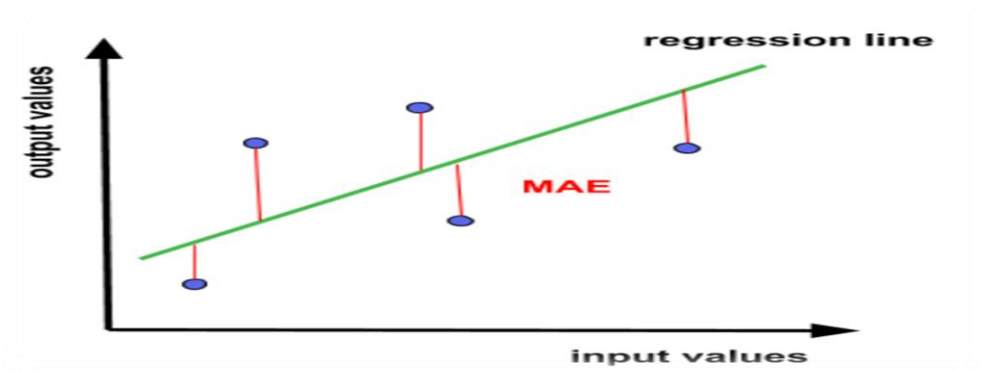


Figure 2: MAE graphe

Le graphique ci-dessus représente les résidus les différences entre les valeurs prédites (ligne de régression) et les valeurs de sortie. *MAE* utilise la valeur absolue des résidus, il ne peut pas donc indiquer si le modèle est sous-performant ou surperformant. Chaque résidu contribue linéairement à l'erreur totale parce que nous additionnons les résidus individuels. Pour cette raison, un petit *MAE* suggère que le modèle est excellent pour la prédiction. De même, un grand *MAE* suggère que votre modèle peut avoir du mal à bien généraliser. Un *MAE* de 0 signifie que notre modèle produit des prédictions parfaites, mais il est peu probable que cela se produise dans des scénarios réels.

- **Erreur quadratique moyenne**

L'erreur quadratique moyenne (*MSE*) prend la différence quadratique moyenne entre la cible et les valeurs prédites. Cette valeur est largement utilisée pour de nombreux problèmes de régression et des erreurs plus importantes ont des contributions au carré proportionnellement plus importantes à l'erreur moyenne.

MSE est donné par la formule suivante:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(yi - \hat{y}i)^2 \text{(eq.5)}$$

Où y_i représente la valeur prédite de $\hat{y}_i$ [10].

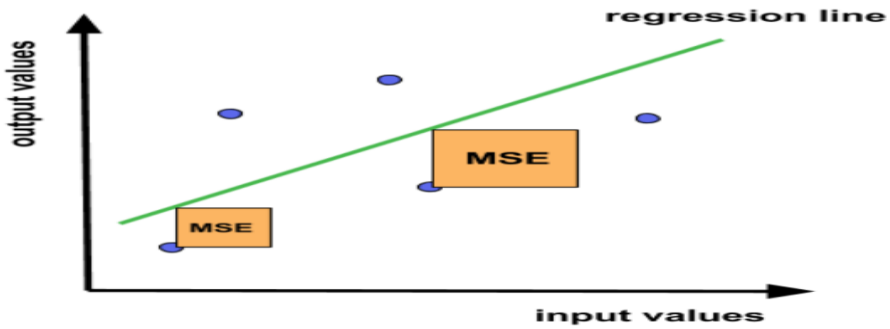


Figure 3: MSE graphe

MSE sera presque toujours plus grand que *MAE* parce que dans *MAE* les résidus contribuent linéairement à l'erreur totale, tandis que dans *MSE* l'erreur croît quadratiquement avec chaque résidu. C'est pourquoi MSE est utilisé pour déterminer dans quelle mesure le modèle ajuste les données car il pénalise fortement les valeurs aberrantes lourdes.

- **Le coefficient de détermination (score R^2)**

Le score R^2 détermine dans quelle mesure les prédictions de régression se rapprochent des points de données réels.

La valeur de R^2 est calculée avec la formule suivante:

$$R^2 = 1 - \sum_{1=1}^{N}(yi - \hat{y})^2 / \sum_{1=1}^{N}(yi - \hat{yi})^2 \text{(eq.6)}$$

Où $\hat{y}_i$ représente la valeur prédite de y i et $\bar{y}$ est la moyenne des données observées qui est calculée comme :$\overline{y} = \sum_{i=1}^{N} yi$

R^2 peut prendre des valeurs de 0 à 1. Une valeur de 1 indique que les prédictions de régression correspondent parfaitement aux données [10].

- **Conseils d'utilisation des métriques de régression**
  - Nous devons toujours nous assurer que la métrique d'évaluation que nous choisissons pour un problème de régression pénalise les erreurs d'une manière qui reflète les conséquences de ces erreurs pour les besoins commerciaux, organisationnels ou utilisateurs de notre application.
  - S'il y a des valeurs aberrantes dans les données, elles peuvent avoir une influence indésirable sur les scores globaux R^2 ou *MSE*. *MAE* est robuste à la présence de valeurs aberrantes car il utilise la valeur absolue. Par conséquent, nous pouvons utiliser le score *MAE* s'il est important pour nous d'ignorer les valeurs aberrantes.
  - *MAE* est la meilleure métrique lorsque nous voulons faire une distinction entre différents modèles car elle ne reflète pas de gros résidus.
  - Si nous voulons nous assurer que notre modèle prend davantage en compte les valeurs aberrantes, nous devons utiliser les métriques *MSE* [10].

3.2.2 Pour les problèmes de classification

L'évaluation d'un problème de classification se base sur une matrice deconfusion, qui met en regard des données prédites et des données observées.

- **Matrice de confusion**

Une matrice de confusion est une *matrice * n* utilisée pour évaluer les performances du modèle de classification. Pour la classification binaire la matrice de confusion est une matrice 2 * 2.Si la classe cible est 3, cela signifie que la matrice de confusion est une matrice 3 * 3 et ainsi de suite [11].

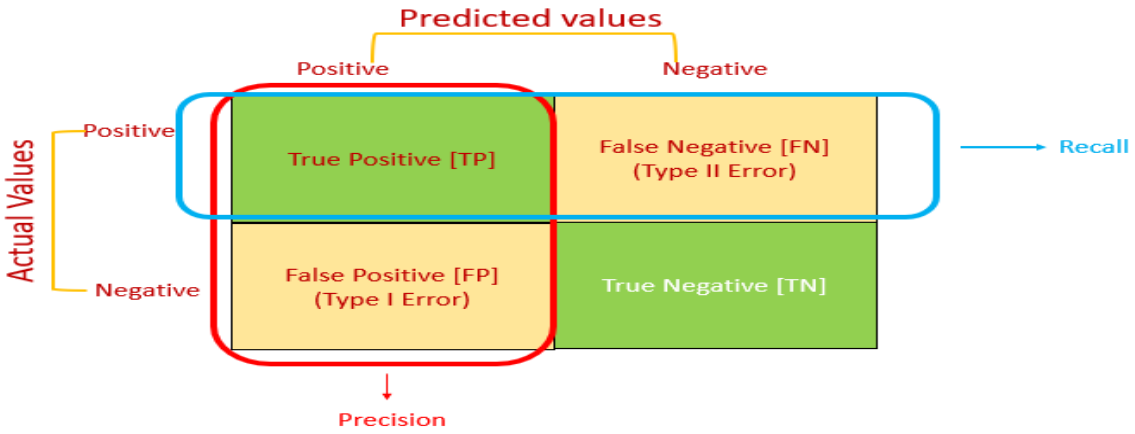


Figure 4: Matrice de confusion

- **Terminologies utilisées dans la Matrice de Confusion**
  - Vrai positif → Classe positive qui est prédite comme positive.
  - Vrai Négatif → Classe négative qui est prédite comme négative.
  - Faux positif → Classe négative qui est prévue comme positive.
  - Faux négatif → Classe positive qui est prédite comme négative.
- **Mesures calculées à partir de la matrice de confusion**

Rappel

Le rappel est une mesure du nombre de points positifs que votre modèle est capable de rappeler à partir des données. Sur tous les enregistrements positifs, combien d'enregistrements sont prédits correctement.

$$Recall = \frac{TP}{TP+FN} \text{(eq.7)}$$

Précision

La précision est le rapport entre les prévisions positives correctes et les prévisions positives totales. Sur tous les positifs prédits, combien sont réellement positifs.

$$Precision = \frac{TP}{TP+FP} \text{(eq.8)}$$

Score F1

Le score F1 est une moyenne harmonique de précision et de rappel.

$$F1 = \frac{2*Precision\ *Recall}{Precision\ +Recall}\text{(eq.9)}$$

Conclusion

Nous avons vu les types d'apprentissage automatique et on a expliqué quelques algorithmes ainsi que les méthodes d'évaluation. Dans le chapitre suivant, nous allons étudier les prévisions de notation numérique et prédiction de sentiment sur Google Play Store.

Chapitre 2 : Prédiction de rating et Prédiction de sentiment sur Google Play Store

Introduction

Dans ce chapitre, on va voir la première partie la prédiction de *rating* avec leur différente étape et la deuxième partie concerne la prédiction de sentiment sur Google Play store.

1. Prédiction de rating de Google Play Store

Nous avons utilisé une architecture pour mener la prédiction de rating, comme illustré dans la figure suivante.

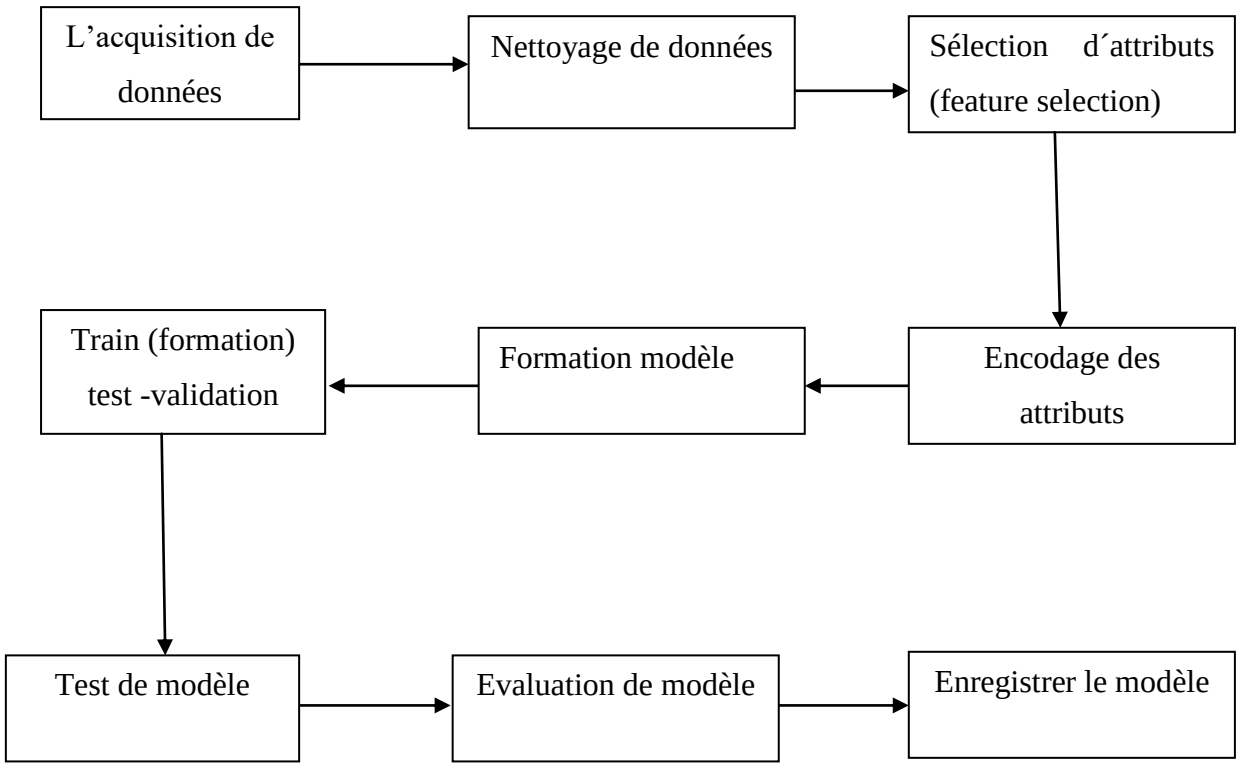


Figure 5: Architecture de la prédiction rating

1.1 Ensemble de données Google Play Store

L'ensemble de données se compose de l'application Google Play Store est extrait de *Kaggle*, qui est le plus grand du monde communauté pour que les scientifiques des données explorent, analysent et partagent ces données.

Cet ensemble de données est pourWebscratchedinformationof10kPlay Stockez des applications pour analyser le marché d'Android, la description des données et les visualisations dans l'annexe.

Tableau 1: Base de données des applications de Google Play store

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

1.2 Prétraitement

1.2.1 Sélection des données

Nous avons Supprimé des lignes avec des données manquantes Il y a 9360 lignes de données après la suppression des lignes avec des données manquantes.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             9360 non-null   object 
 1   Category        9360 non-null   object 
 2   Rating          9360 non-null   float64
 3   Reviews         9360 non-null   object 
 4   Size            9360 non-null   object 
 5   Installs        9360 non-null   object 
 6   Type            9360 non-null   object 
 7   Price           9360 non-null   object 
 8   Content Rating  9360 non-null   object 
 9   Genres          9360 non-null   object 
 10  Last Updated    9360 non-null   object 
 11  Current Ver     9360 non-null   object 
 12  Android Ver     9360 non-null   object 
dtypes: float64(1), object(12)
memory usage: 1023.8+ KB
```

Figure 6 : Résultat de suppressions des lignes avec des données manquantes

1.2.2 Nettoyage des données et Transformer la taille en un format uniforme

Les données sont actuellement encore sous la forme "Mo" et "Ko".C'est-à-dire 1000 Mo ou 100 Ko, Les données sous forme de Mo sont multipliées par 1000000 et Ko par 1000 pour obtenir la taille en octets.

Il existe également des données avec «Varie selon l'appareil». Pour ces données, il est imputé avec la fonction pandas *fillna* avec la méthode *ffill*, qui impute les données avec les données de la ligne précédente.

12.3 Codage des données

Les données actuelles sont désormais «gratuites» ou «payantes» Gratuit est converti en 0 et Payé est converti en 1, Transformation du libellé Content Rating en content rating pour une manipulation plus facile, Il y a une seule ligne de données avec les données "Non classé", nous supprimons donc ces données car elles ne peuvent pas être utilisées. Ensuite nous avons Transformez le prix en valeur flottante : Les données actuelles qui sont actuellement sous la forme "$" ou "0" soit 4,99 $ Nous supprimons le "$" et convertissons les données en un type de données flottant.

Tableau 2: Résultat après le nettoyage

	Category	Rating	Size	Type	Price	content_rating	Genres
0	ART_AND_DESIGN	4.1	19000000.0	0	0.0	Everyone	Art & Design
1	ART_AND_DESIGN	3.9	14000000.0	0	0.0	Everyone	Art & Design;Pretend Play
2	ART_AND_DESIGN	4.7	8700000.0	0	0.0	Everyone	Art & Design
3	ART_AND_DESIGN	4.5	25000000.0	0	0.0	Teen	Art & Design
4	ART_AND_DESIGN	4.3	2800000.0	0	0.0	Everyone	Art & Design;Creativity

1.2.4 Sélection d´attributs

Nous avons supprimé certaines fonctionnalités qui n'affectent probablement pas la prédiction, telles que "Dernière mise à jour", "Version actuelle", "Androïde Ver", "App", "Installations" et "Avis". En effet, lorsque nous prédisons la note, nous ne pouvons pas saisir les installations ou les critiques de notre application avant de l'avoir réellement sur le Google Play Store.

1.2.5 Encodage des fonctionnalités

Étant donné que les modèles n'acceptent pas les types de données chaîne, nous devons convertir nos types de données chaîne dans un format acceptable. Il y a 3 fonctionnalités que nous encodons dans cette section qui sont "Catégorie", "content rating" et "Genres". Ces fonctionnalités sont des données catégoriques et pour les convertir en données numériques, il existe plusieurs méthodes telles que l'encodage d'étiquettes et l'encodage à chaud.

Nous utilisons un *encodage à chaud* sur l'encodage d'étiquettes pour nos fonctionnalités, car avec l'encodage d'étiquettes, l'algorithme d'apprentissage automatique peut «mal interpréter» la valeur numérique des données encodées de sorte que l'une soit plus significative que l'autre, même si ce n'est peut-être pas le cas.

Un encodage à chaud sépare chaque donnée unique de notre fonctionnalité dans une colonne distincte. Si les données correspondent à cette colonne, alors sa valeur est un, sinon elle vaut 0.

Pour "Genres", car il existe plusieurs données avec plusieurs genres tels que "Art &Design, Pretend Play", nous les encodons de manière à ce que Art & Design et Pretend Play aient la valeur 1.

Tableau 3: Matrice de transformation après l'encodage du genre

	Category	Rating	Size	Type	Price	content_rating	Action	Action & Adventure	Adventure	Arcade	Art & Design	Auto & Vehicles	Beauty	Board	Books & Reference	Brain Games	E
0	ART_AND_DESIGN	4.1	19000000.0	0	0.0	Everyone	0	0	0	0	1	0	0	0	0	0	
1	ART_AND_DESIGN	3.9	14000000.0	0	0.0	Everyone	0	0	0	0	1	0	0	0	0	0	
2	ART_AND_DESIGN	4.7	8700000.0	0	0.0	Everyone	0	0	0	0	1	0	0	0	0	0	
3	ART_AND_DESIGN	4.5	25000000.0	0	0.0	Teen	0	0	0	0	1	0	0	0	0	0	
4	ART_AND_DESIGN	4.3	2800000.0	0	0.0	Everyone	0	0	0	0	1	0	0	0	0	0	

Pour "Catégorie" et "content rating", nous pouvons simplement utiliser la fonction pandas *get_dummies* pour séparer chaque donnée unique dans une colonne.

Tableau 4: Matrice de transformation après l'encodage de la catégorie

	Rating	Size	Type	Price	Action	Action & Adventure	Adventure	Arcade	Art & Design	Auto & Vehicles	Beauty	Board	Books & Reference	Brain Games	Business	Card	Casino	Casual	C
0	4.1	19000000.0	0	0.0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
1	3.9	14000000.0	0	0.0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
2	4.7	8700000.0	0	0.0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
3	4.5	25000000.0	0	0.0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
4	4.3	2800000.0	0	0.0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	

1.2.6 Création, test et validation

Ici, nous définissons quelles colonnes sont des fonctionnalités dans X et des étiquettes dans Y. L'étiquette est Rating et les attributs sont le reste des colonnes.

1.2.7 La validation croisée

La qualité de l'ajustement ou de la prévision est calclculée pour chacun des jeux de données à partir de la métrique p (retenue).en pratique, on prend souvent 60% des données pour m (entrainement) ,20% pour m (validations) et 20% pour m (test) [12].

2. Modèle de classification

2.1 Régression linéaire

Nous avons divisé les données en données de formation, de test et de validation avec un ratio de 3/5, 1/5 et 1/5. Nous entrons ensuite les données d'apprentissage dans l'algorithme de régression linéaire pour obtenir le modèle et enregistrons le modèle d'apprentissage dans pickle. Ensuite, nous testons le modèle avec nos données de test.

```
[4.03958041 4.05551751 4.10558132 ... 4.34870678 4.06948995 4.2584719 ]
```

Figure 7: Résultat du modèle de régression linéaire

2.2 Support Vector Régression

Encore une fois, nous avons divisé les données en données d'entraînement, de test et de validation avec un ratio de 3/5, 1/5 et 1/5. Nous entrons ensuite les données d'entraînement dans l'algorithme de régression vectorielle de support pour obtenir le modèle et enregistrons le

modèle d'apprentissage dans pickle. Ensuite, nous testons le modèle avec nos données de test.

```
[4.26868067 4.3021018  4.29919264 ... 4.30580883 4.30318399 4.28490001]
```

Figure 8: Résultat du modèle de support vector régression

2.3 Arbre de décision de Régression

Encore une fois, nous avons divisé les données en données de formation, de test et de validation avec un ratio de 3/5, 1/5 et 1/5. Nous entrons ensuite les données d'apprentissage dans l'algorithme du régresseur d'arbre de décision pour obtenir le modèle, et enregistrons le modèle d'apprentissage dans pickle. Ensuite, nous testons le modèle avec nos données de test.

```
[3.95 3.9  3.   ... 4.6  4.39 4.65]
```

Figure 9: Résultat du modèle de régresseur d'arbre décision

3. évaluation de modèle

Nous évaluons le modèle de régression linéaire, support vecteur régression et arbre de discision de régression en utilisant l'erreur absolue moyenne, l'erreur quadratique moyenne et le score R^2 du résultat de la prédiction à l'aide des données de test et des données réelles des données de validation. Nous obtenons le résultat comme indiqué ci-dessous.

- **Modèle de régression linéaire**

```
MAE:  0.3491071751602544
MSE:  0.2506395138988662
R2 Score:  0.03445027542677148
```

Figure 10: Résultat d'évaluation du modèle de régression linéaire

- **Support vector régression**

```
MAE:  0.33899561081265994
MSE:  0.2664225839032345
R2 Score:  -0.02635154571698828
```

Figure 11: Résultat d'évaluation du modèle de support vector régression

- **Arbre de discision de régression**

```
MAE:  0.40933616434017084
MSE:  0.3895493437290457
R2 Score:  -0.500678227843321
```

Figure 12: Résultat d'évaluation du modèle d'arbre discision régression

Le R^2 score est négative donc on ne peut pas interpréter les résultats des algorithmes donc nous avons besoin d'utiliser les métriques *MSE* car il est Sensible aux valeurs aberrantes, punit davantage les erreurs plus importantes Et d'après le *MSE* le model le plus performant est l'arbre de discision régression.

4. Prédiction de sentiment sur Google Play Store

Nous avons utilisé une architecture pour mener l'analyse des sentiments des avis, comme illustré dans cette figure. Tout d'abord, nous avons illustré et alimenté les données dans le nettoyage et le prétraitement des données. Ensuite, Nous avons supprimé les mots vides et certains mots non pertinents des données originales, puis, la vectorisation des techniques est appliquées pour transformer du texte en entité matrice. Enfin, nous avons appliqué trois algorithmes différents pour former et tester la matrice de caractéristiques.

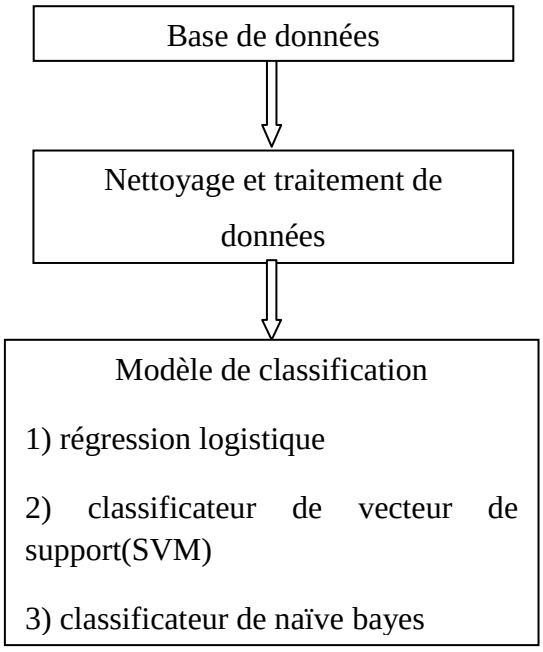


Figure 13: Architecture de prédiction de sentiment de Google Play Store

4.1 Base de données

La base de données est tirée de Kaggle aussi.

Tableau 5: La base de données de sentiment de Google Play Store

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	10 Best Foods for You	I like eat delicious food That s I m cooking ...	Positive	1.00	0.533333
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.288462
3	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.875000
4	10 Best Foods for You	Best idea us	Positive	1.00	0.300000
5	10 Best Foods for You	Best way	Positive	1.00	0.300000

On à des applications et leur avis, La fonction de sentiment de tex blob renvoie deux propriétés, la polarité et la subjectivité.

La polarité est flottante qui se situe dans la plage de [-1,1] où 1 signifie une déclaration positive et -1 signifie une déclaration négative.

Les phrases subjectives font généralement référence à une opinion personnelle, à une émotion ou à un jugement, tandis que l'objectif se réfère à des informations factuelles. La subjectivité est également un flottant qui se situe dans la plage de [0,1].

4.2 Nettoyage et traitement des données

4.2.1 Nettoyage des données

Suppression des ponctuations, des nombres et des caractères spéciaux et des mots courts.

4.2.2 La tokenisation

C'est à ce moment que le texte est décomposé en unités plus petites pour travailler.

```
0    [like, delicious, food, That, cooking, food, m...
1    [This, help, eating, healthy, exercise, regula...
2    [Works, great, especially, going, grocery, store]
3                                          [Best, idea]
4                                                [Best]
Name: Translated_Review, dtype: object
```

Figure 14: La tokenisation du texte

4.2.3 La lemmatisation

C'est à ce moment que les mots sont réduits à leur forme racine pour être traités.

```
0        [like, delicious, food, That, cooking, food, m...
1        [This, help, eating, healthy, exercise, regula...
2        [Works, great, especially, going, grocery, store]
3                                                [Best, idea]
4                                                      [Best]
                                ...
37422    [Most, older, many, agent, much, owner, posted...
37423    [photo, posted, portal, load, purpose, sure, s...
37424    [Dumb, wanted, post, property, rent, give, opt...
37425    [property, business, link, happy, performance,...
37426    [Useless, searched, flat, kondapur, Hyderabad,...
Name: Translated_Review, Length: 37427, dtype: object
```

Figure 15: La lemmatisation du texte

4.2.4 Conversion d'étiquettes en nombres

```
0
1
2
3
4
       ..
37422
37423
37424
37425
37426
Name: Translated_Review, Length: 37427, dtype: object
```

Figure 16: Résultat de la conversion d'étiquettes en nombres

4.2.5 Nuages de mots

Les nuages de mots sont devenus une méthode de visualisation simple et attrayante pour le texte. Ils sont utilisés dans divers contextes comme un moyen de fournir une vue d'ensemble en distillant le texte jusqu'aux mots qui apparaissent avec la fréquence la plus élevée. En règle générale, cela se fait de manière statique sous forme de résumé de texte pur. Nous pensons, cependant, qu'il existe un plus grand potentiel pour ce paradigme de visualisation simple mais puissant dans l'analyse de texte. Dans ce travail, nous explorons l'utilité des nuages de mots pour les tâches générales d'analyse de texte. Nous avons développé un système prototypique appelé Word Cloud Explorer qui repose entièrement sur les nuages de mots comme méthode de visualisation. Il les équipe d'un traitement avancé du langage naturel, de techniques d'interaction sophistiquées et d'informations contextuelles.

Nous montrons comment cette approche peut être utilisée efficacement pour résoudre des tâches d'analyse de texte et l'évaluer dans une étude utilisateur qualitative. : [13]

- Nuage des mots pour tous les avis

Figure 17: Nuage des mots pout tous les avis

- Nuage des mots pour les avis positive

Figure 18: Nuage des mots pour les avis positif

- Nuage des mots pour les avis négatif

Figure 19: Nuage des mots pour les avis négatif

4.2.6 Vérifier la distribution des sentiments

```
Positive    23998
Negative     8271
Neutral      5158
Name: Sentiment, dtype: int64
```

Figure 20: Résultat de la distribution de sentiment

Ici nous avons observé que le pourcentage des avis positives est la partie la plus grande de 64.2% les avis négatives 22.1% et les avis neutres de 13.8%.

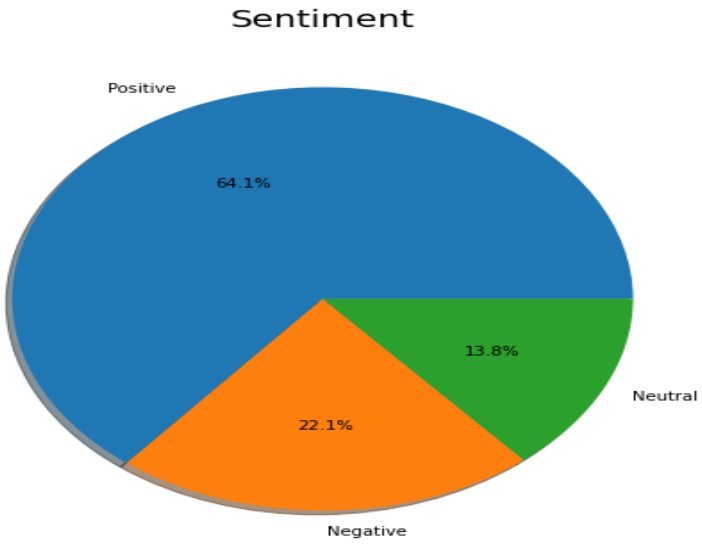


Figure 21: Pourcentage des avis total

5. Modèle de classification

Nous avons met en œuvre trois modèles de classification pour analyser le sentiment du contexte, y compris régression logistique, machine à vecteurs de support et Naïve Classificateur Bayes.

5.1 Test et validation

Nous avons définissons quelles colonnes sont des fonctionnalités dans X et des étiquettes dans Y. L'étiquette est sentiment et les fonctionnalités sont les revues traduites.

5.2 Pipeline

Pipeline peut être utilisé pour enchaîner plusieurs estimateurs en un seul. Ceci est utile car il y a souvent une séquence fixe d'étapes dans le traitement des données, par exemple la

sélection de caractéristiques, la normalisation et la classification. Pipeline sert à plusieurs fins ici:

- **Commodité et encapsulation**
  - Il vous suffit d'appeler fit et de prédire une fois sur vos données pour ajuster toute une séquence d'estimateurs.
- **Sélection des paramètres communs**
  - Vous pouvez effectuer une recherche de grille sur les paramètres de tous les estimateurs du pipeline à la fois.
- **Sécurité**
  - Les pipelines permettent d'éviter les fuites de statistiques de vos données de test vers le modèle entraîné lors de la validation croisée, en garantissant que les mêmes échantillons sont utilisés pour entraîner les transformateurs et les prédicteurs.
  - Tous les estimateurs d'un pipeline, à l'exception du dernier, doivent être des transformateurs (c'est-à-dire doivent avoir une méthode de transformation). Le dernier estimateur peut être de n'importe quel type (transformateur, classificateur, etc.)[14].

5.3 TF-IDF Vectorizer

Convertissez une collection de documents bruts en une matrice de fonctionnalités TF-IDF.

TF-IDF signifie terme-fréquence tandis que TF-IDF signifie terme-fréquence multiplié par l'inverse de la fréquence du document. Il s'agit d'un schéma de pondération des termes courant dans la recherche d'informations, qui a également trouvé une bonne utilisation dans la classification des documents [15].

Dans cette partie nous avons utilisé des algorithmes qui sont expliqués dans l'état de l'art.

```
Logistic Regression

 Accuracy Score:-  0.8795974708344465

 Confusion Matrix:-
 [[1897  105  523]
 [  50 1138  360]
 [ 163  151 6842]]

 Classification Report:-
              precision    recall  f1-score   support

    Negative       0.90      0.75      0.82      2525
     Neutral       0.82      0.74      0.77      1548
    Positive       0.89      0.96      0.92      7156

    accuracy                           0.88     11229
   macro avg       0.87      0.81      0.84     11229
weighted avg       0.88      0.88      0.88     11229
```

Figure 22: Résultat d'évaluation de la modèle logistique régression

```
Naive Bayes

 Accuracy Score:-  0.6846558019414017

 Confusion Matrix:-
 [[ 499    0 2026]
 [  13   43 1492]
 [   8    2 7146]]

 Classification Report:-
              precision    recall  f1-score   support

    Negative       0.96      0.20      0.33      2525
     Neutral       0.96      0.03      0.05      1548
    Positive       0.67      1.00      0.80      7156

    accuracy                           0.68     11229
   macro avg       0.86      0.41      0.39     11229
weighted avg       0.77      0.68      0.59     11229
```

Figure 23:Résultat d'évaluation du modèle naïve bayes

```
Support Vector Classifier

Accuracy Score:-  0.8908184165998754

Confusion Matrix:-
[[2015  115  395]
[  66 1230  252]
[ 217  181 6758]]

Classification Report:-
              precision    recall  f1-score   support

    Negative       0.88      0.80      0.84      2525
     Neutral       0.81      0.79      0.80      1548
    Positive       0.91      0.94      0.93      7156

    accuracy                           0.89     11229
   macro avg       0.87      0.85      0.85     11229
weighted avg       0.89      0.89      0.89     11229
```

Figure 24: Résultat d'évaluation du modèle support vector classifier

Ces estimations nous fournissent un outil pour évaluer l'exactitude des prévisions positives, négatives et neutre. le classificateur *SVM* contribuer à une meilleure précision de prédiction environ 89% (le plus élevé de tous les modèles).

Conclusion

La prédiction des données est une tache dont l'objectif est de prédire rating et Prédiction de sentiment d'examen sur Google Play Store Dans ce chapitre, nous avons vu les grands concepts de prédiction, les principales techniques. Nous avons également présenté les étapes de prétraitement. Nous sommes passés ensuite, à l'étape de traitement tel que les algorithmes de prédiction qu'on a appliqué, et Enfin, on a terminé avec les l'évaluation de model.

Dans le prochain chapitre, nous allons aborder les outils et les différentes bibliothèques utiliser avec des captures d'écrans de l'application.

Chapitre 3 : expérimentation et résultat

Introduction

Ce chapitre constitue le dernier volet du rapport ayant pour objectif d'exposer le travail achevé. Pour ce faire, nous allons présenter dans un premier temps l'environnement matériel et logiciel supportant notre application. Par la suite, nous présentons la plateforme de développement et les choix technologiques. Ensuite, nous allons passer en revue les différentes tâches réalisées à travers quelques interfaces homme- machine et un chronogramme récapitulatif qui décrit toutes étapes de mise en œuvre de notre système.

1 .Environnements de travail

Tout au long de la réalisation de notre application, nous avons utilisé des matériels et des logiciels bien particuliers.

1.1 Environnement matériel

Le développement de l'environnement matériel est caractérisé par :

- Système d'exploitation : Windows 10 Professionnel.
- CPU : Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz
- Mémoire : 4,00 Go

1.2 Environnement logiciel

L'environnement logiciel consiste les composants suivants :

Google Colab : Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique [16].

Visual Studio : Code dispose d'un éditeur de code à haute productivité qui, lorsqu'il est combiné avec des services de langage de programmation, vous donne la puissance d'un IDE et la vitesse d'un éditeur de texte [17].

Python : est un langage de programmation informatique le plus populaire pour le traitement Big Data, l'exécution de calculs mathématiques ou le Machine Learning. De manière générale, il s'agit du langage de prédilection pour la Data Science[20].

2. Implémentation et imprime écran

2.1 Les Bibliothèques utilisé

Nltk : Le Natural Language Toolkit (NLTK) est une plate-forme utilisée pour créer des programmes Python qui fonctionnent avec des données de langage humain pour une application dans le traitement statistique du langage naturel (NLP). Il contient des bibliothèques de traitement de texte pour la tokenisation, l'analyse, la classification, le radicalisme, le balisage et le raisonnement sémantique. Il comprend également des démonstrations graphiques et des exemples d'ensembles de données, ainsi qu'un livre de recettes et un livre qui explique les principes sous-jacents aux tâches de traitement du langage sous-jacentes prises en charge par NLTK [18].

Bokeh : est une bibliothèque Python permettant de créer des visualisations interactives pour les navigateurs Web modernes. Il vous aide à créer de superbes graphiques, allant de simples tracés à des tableaux de bord complexes avec des ensembles de données en continu. Avec Bokeh, vous pouvez créer des visualisations basées sur JavaScript sans écrire vous-même de JavaScript [19].

Pandas : Pandas est une librairie python qui permet de manipuler facilement des données à analyser :

- manipuler des tableaux de données avec des étiquettes de variables (colonnes) et d'individus (lignes).
- ces tableaux sont appelés Data Frames, similaires aux data frames sous R.
- on peut facilement lire et écrire ces data frames à partir ou vers un fichier tabulé.
- on peut facile tracer des graphes à partir de ces Data Frames grâce à Matplotlib [20].

Matplotlib : est une bibliothèque de visualisation étonnante en Python pour les tracés 2D de tableaux. Matplotlib est une bibliothèque de visualisation de données multiplateforme construite sur des tableaux Numpy et conçue pour fonctionner avec la pile SciPy plus large. Il a été introduit par John Hunter en 2002 [21].

Numpy : est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet de tableau multidimensionnel, divers objets dérivés (tels que des tableaux et des matrices masqués) et un assortiment de routines pour des opérations rapides sur des tableaux, y compris des opérations mathématiques, logiques, de

forme, de tri, de sélection, d'E / S, transformées de Fourier discrètes, algèbre linéaire de base, opérations statistiques de base, simulation aléatoire et bien plus encore [22].

Pickle : est utilisé pour la sérialisation et la désérialisation des structures d'objets Python, également appelées marshalling ou aplatissement. La sérialisation fait référence au processus de conversion d'un objet en mémoire en un flux d'octets qui peut être stocké sur disque ou envoyé sur un réseau. Plus tard, ce flux de caractères peut ensuite être récupéré et désérialisé en un objet Python [23].

Ngrok : Un serveur web local à internet.

TCP RÉSERVÉS ADRESSES DOCS : ngrok http -hostname=*.example.com 8080

DOMAINE NGROK.IO : curl, http://localhost:4040/api/tunnels [24].

2.2 Les technologies web utilisés

***Flask* :** est un Framework Web. Cela signifie que *Flask* vous fournit des outils, des bibliothèques et des technologies qui vous permettent de créer une application Web [25].

Dash (**Dynamic Adaptive Streaming over HTTP**) : est un Framework Python productif pour la création d'applications d'analyse Web.

Écrit au-dessus de *Flask, Plotly.js* et *React.js*, Dash est idéal pour créer des applications de visualisation de données avec des interfaces utilisateur hautement personnalisées en Python pur. Il est particulièrement adapté à quiconque travaille avec des données en Python [26].

Dash est livré avec des composants suralimentés pour des interfaces utilisateur interactives. Un ensemble de composants de base, rédigé et maintenu par l'équipe *Dash*, est disponible dans la bibliothèque *dash-core-components*.

Dash est livré avec un composant Graph qui rend les graphiques avec *plotly.js* . *Plotly.js* convient parfaitement à *Dash*: il est déclaratif, open source, rapide et prend en charge une gamme complète de graphiques scientifiques, financiers et commerciaux. Plotly.js

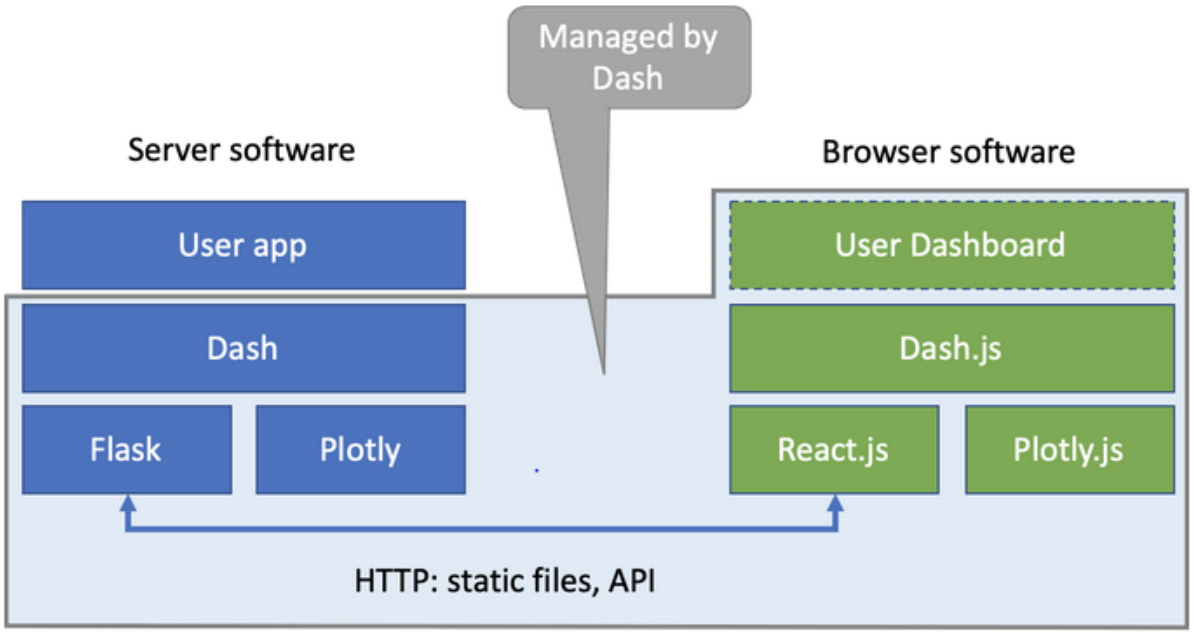


Figure 25: Architecture de Dash

Le paquet *Plotly* Python a trois modules principaux qui sont donnés ci-dessous :

- Intrigue. Plotly
- Plotly. Graph_objs
- plotly.tools

Le **module plotly. Plotly** contient des fonctions qui nécessitent une réponse des serveurs de *Plotly*. Les fonctions de ce module sont une interface entre votre machine locale et Plotly.

Le **module plotly. Graph_objs** est le **module** le plus important qui contient toutes les définitions de classe pour les objets qui composent les tracés que vous voyez. Les objets graphiques suivants sont définis [27] :

- Chiffre
- Données
- layout

Différentes traces graphiques telles que **Scatter, Box, Histogramme,** etc.

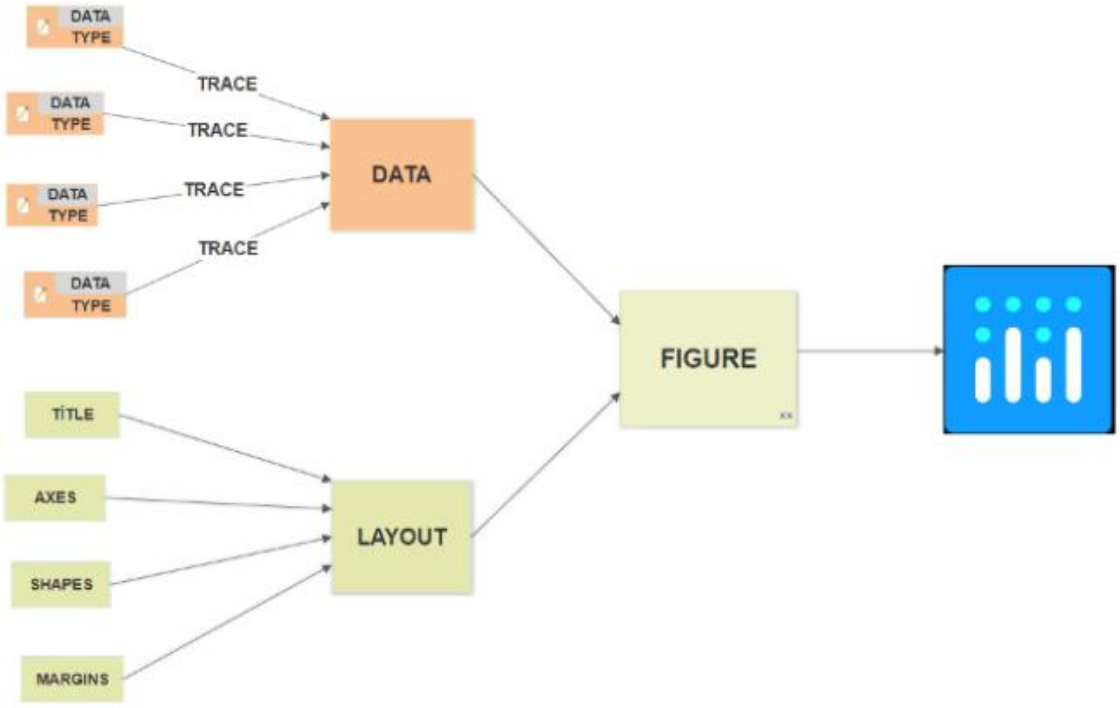


Figure 26: Structure de Plotly

3. Flask intégration dash

Dash a un petit secret: il est arrivé ici avec un peu d'aide de *Flask*. En fait, Dash étend *Flask*: chaque fois que nous créons une application Dash, nous créons en fait une application Flask avec des cloches et des sifflets supplémentaires. Cela semble raisonnable, et peut-être même excitant

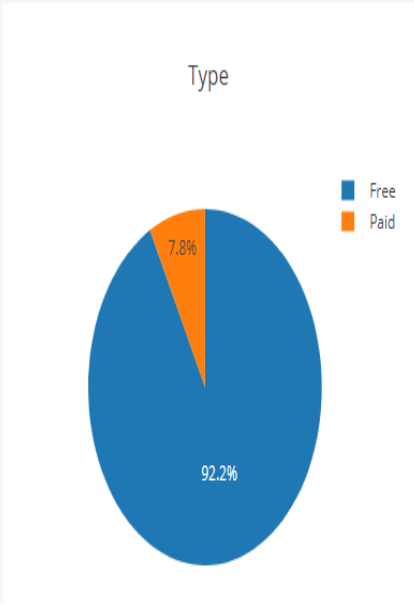
Au moment où *Dash* est initialisé avec app = *Dash* (__name__), il lance une application *Flask* sur laquelle se greffer. Rétrospectivement, cela ne devrait pas être surprenant car la syntaxe pour démarrer une application *Dash* est exactement la même que pour démarrer une application *Flask* [28].

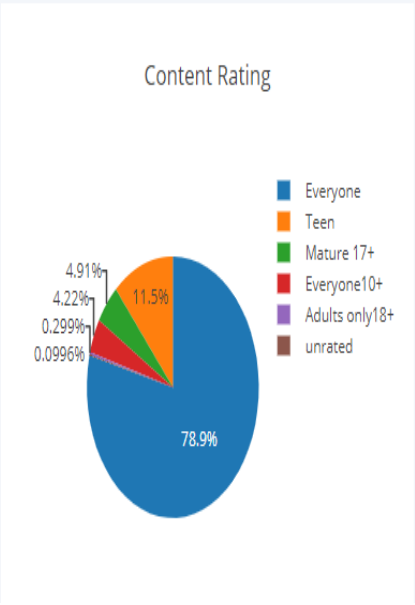
4. Exemple de résultats

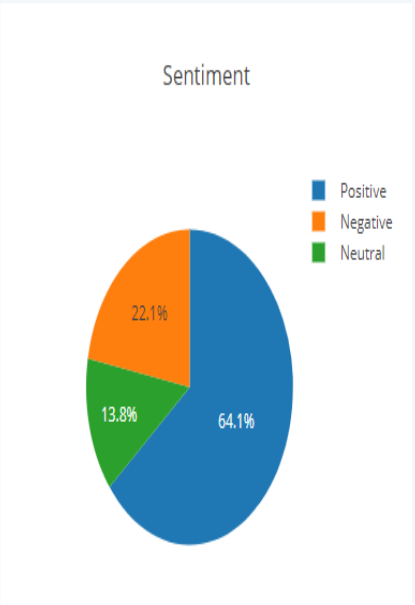
Dans cette partie, nous présentons à travers un enchaînement de captures d'écran, un scénario d'exécution donnant un aperçu général sur les fonctionnalités de notre système. La figure 32 montre la page d'accueil tel que toutes les informations de Google Play store.

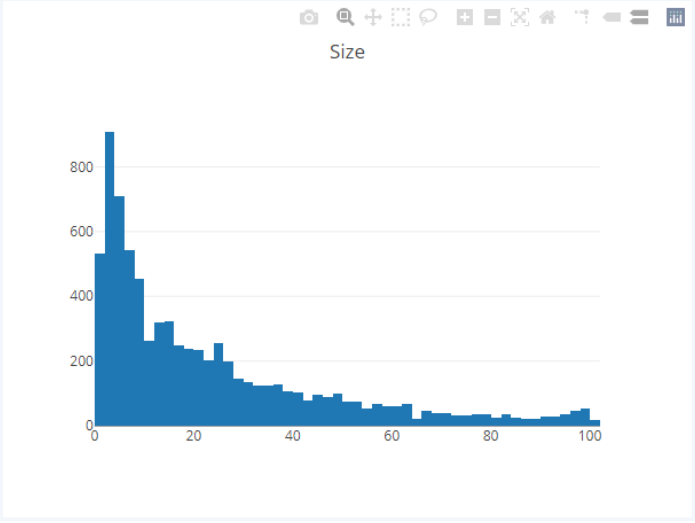
Google Play Store App Dashboard

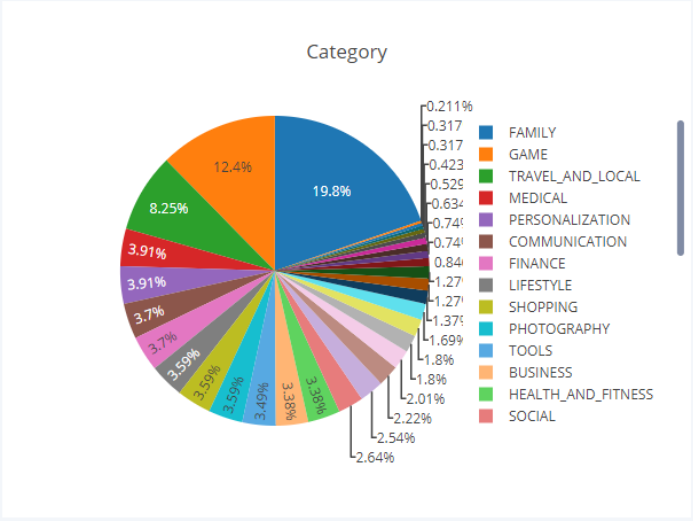
Google Play Store App | Rating | Predection Rating

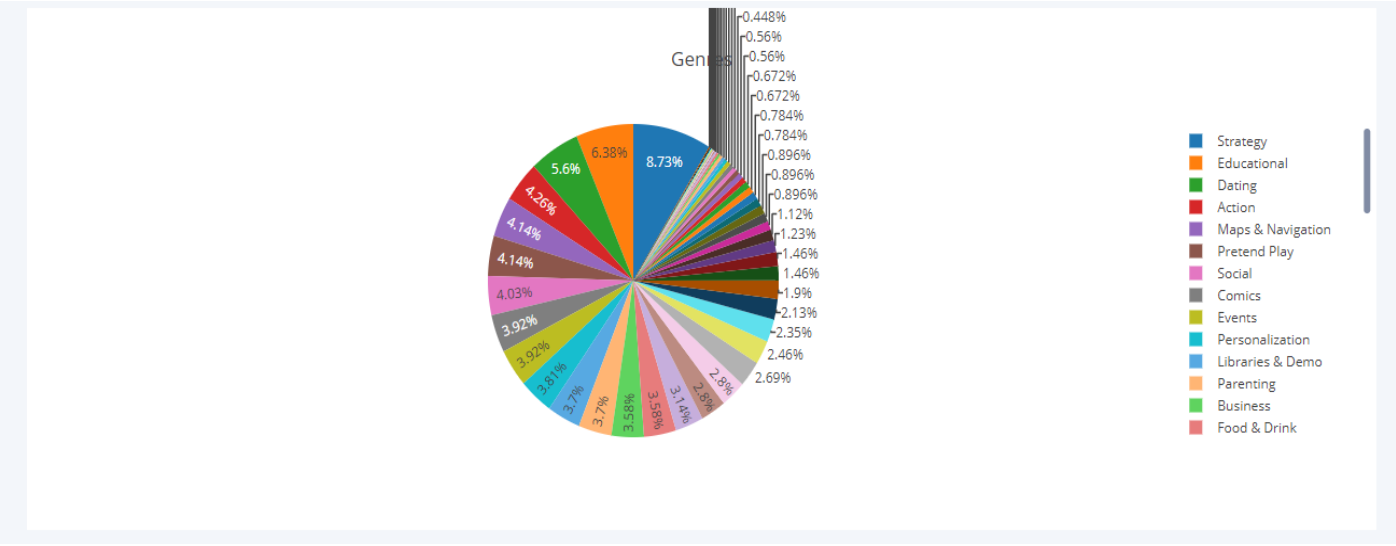












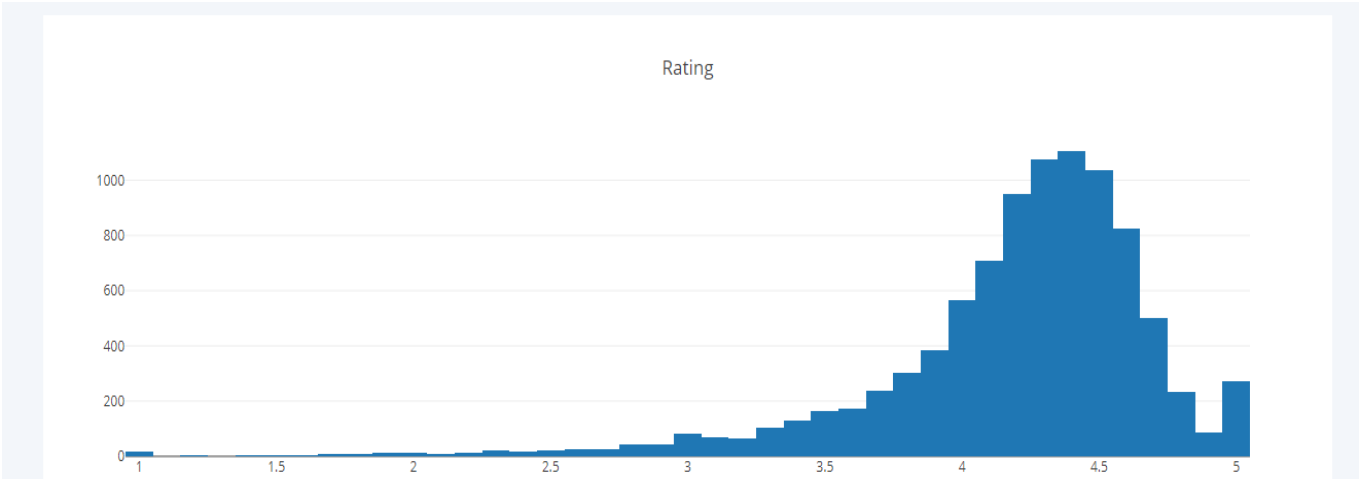
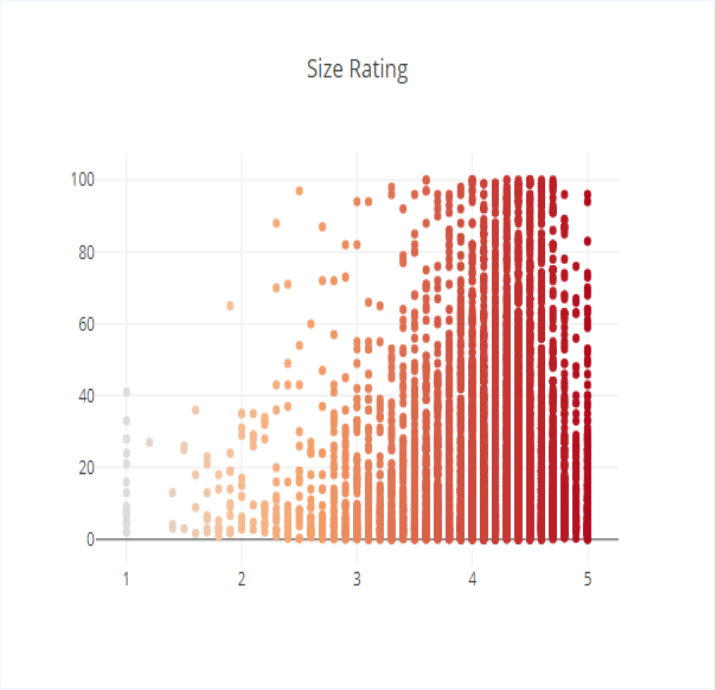
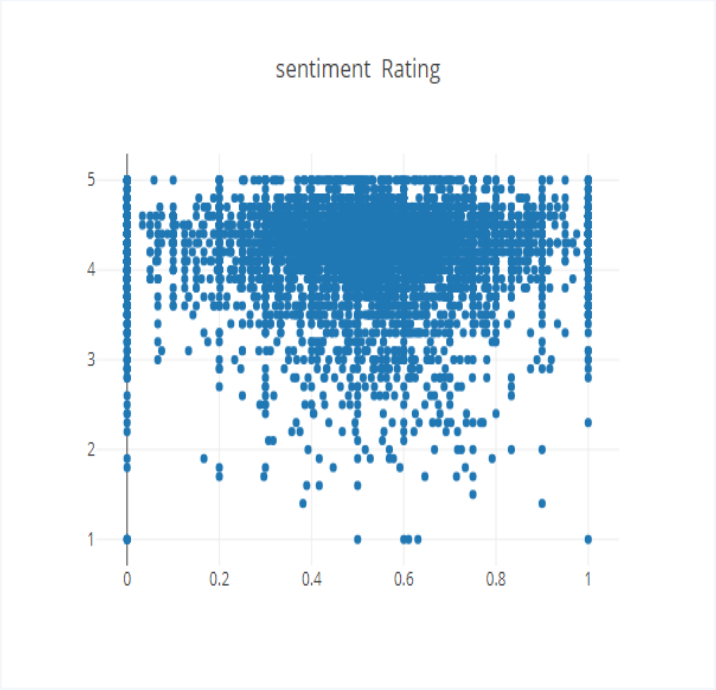
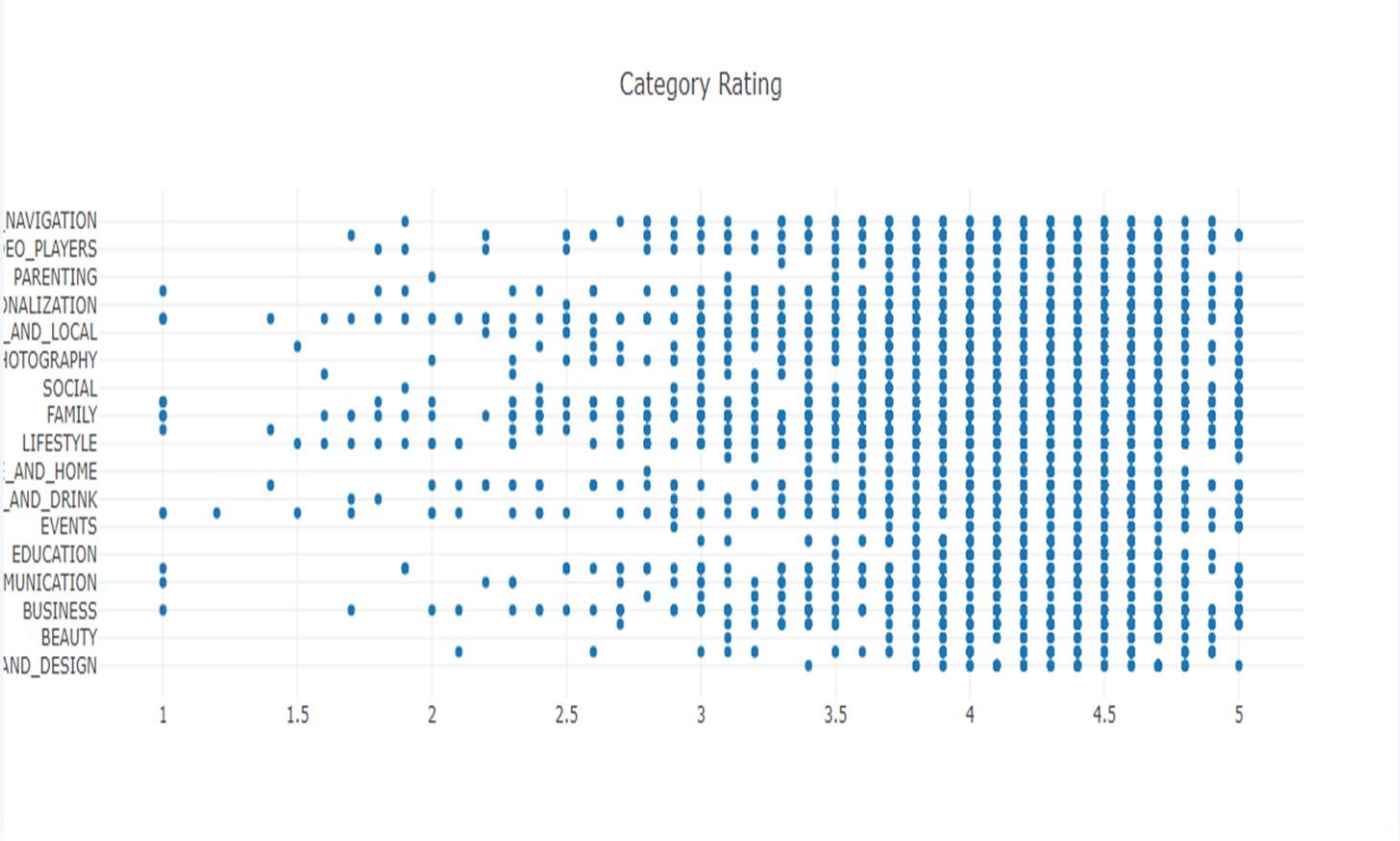


Figure 27: Dashboard de Google Play Store

En choisissant le menu « Rating» de notre application, l'utilisateur peut savoir les informations sur le rating concernant (size, sentiment, catégories, genre).


Size Rating
100
80
60
40
20
0
1
2
3
4
5


sentiment Rating
5
4
3
2
1
0
0.2
0.4
0.6
0.8
1


Category Rating
NAVIGATION
EO_PLAYERS
PARENTING
NALIZATION
_AND_LOCAL
OTOGRAPHY
SOCIAL
FAMILY
LIFESTYLE
_AND_HOME
_AND_DRINK
EVENTS
EDUCATION
MUNICATION
BUSINESS
BEAUTY
ND_DESIGN
1
1.5
2
2.5
3
3.5
4
4.5
5

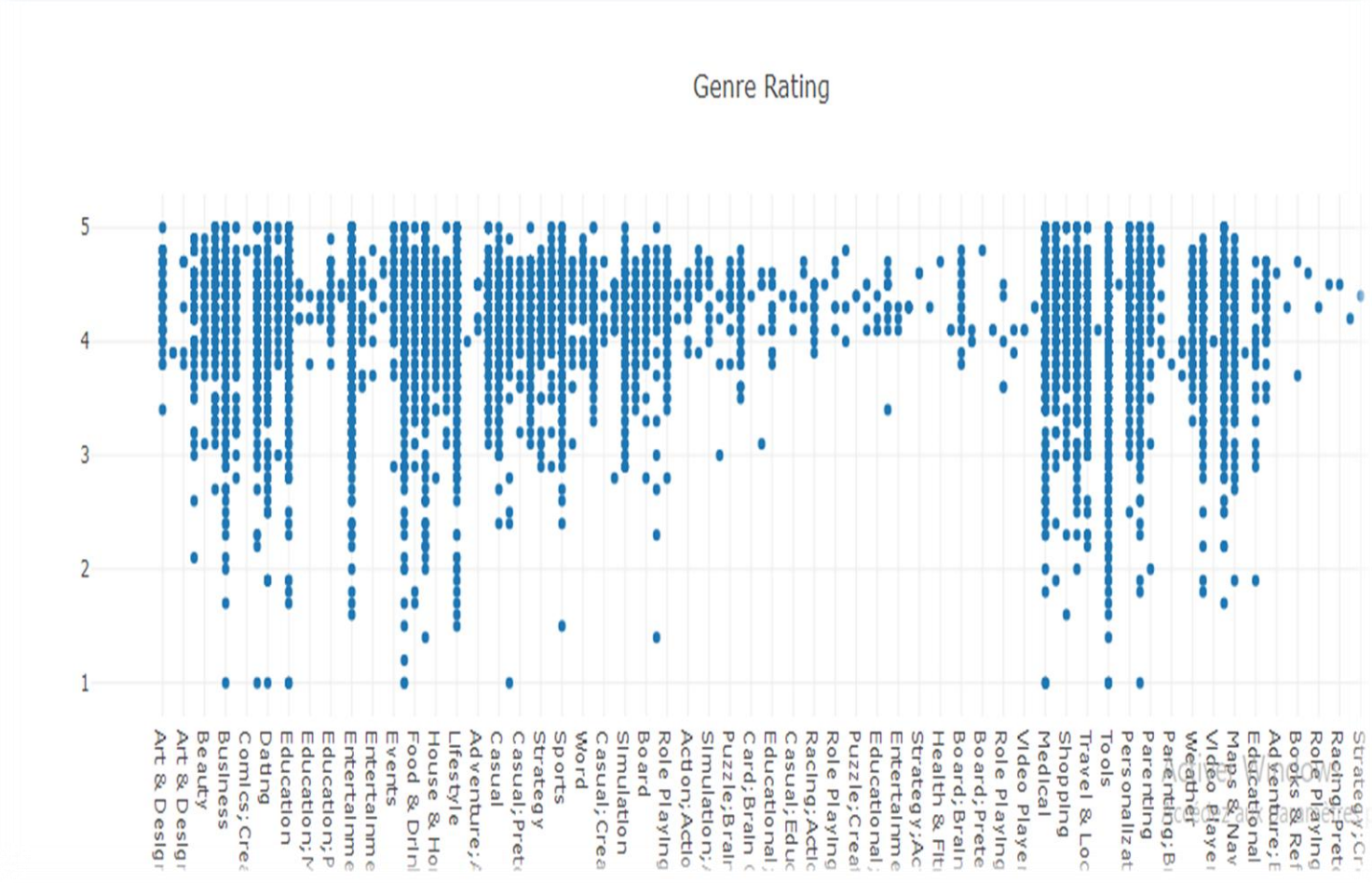


Figure 28: Visualisations en fonction de rating

En choisissant le menu «prédiction Rating» de notre application, le développeur remplir le formulaire (content rating, le type gratuit ou payant, genre, size, catégorie) et le modèle va prédire le rating.

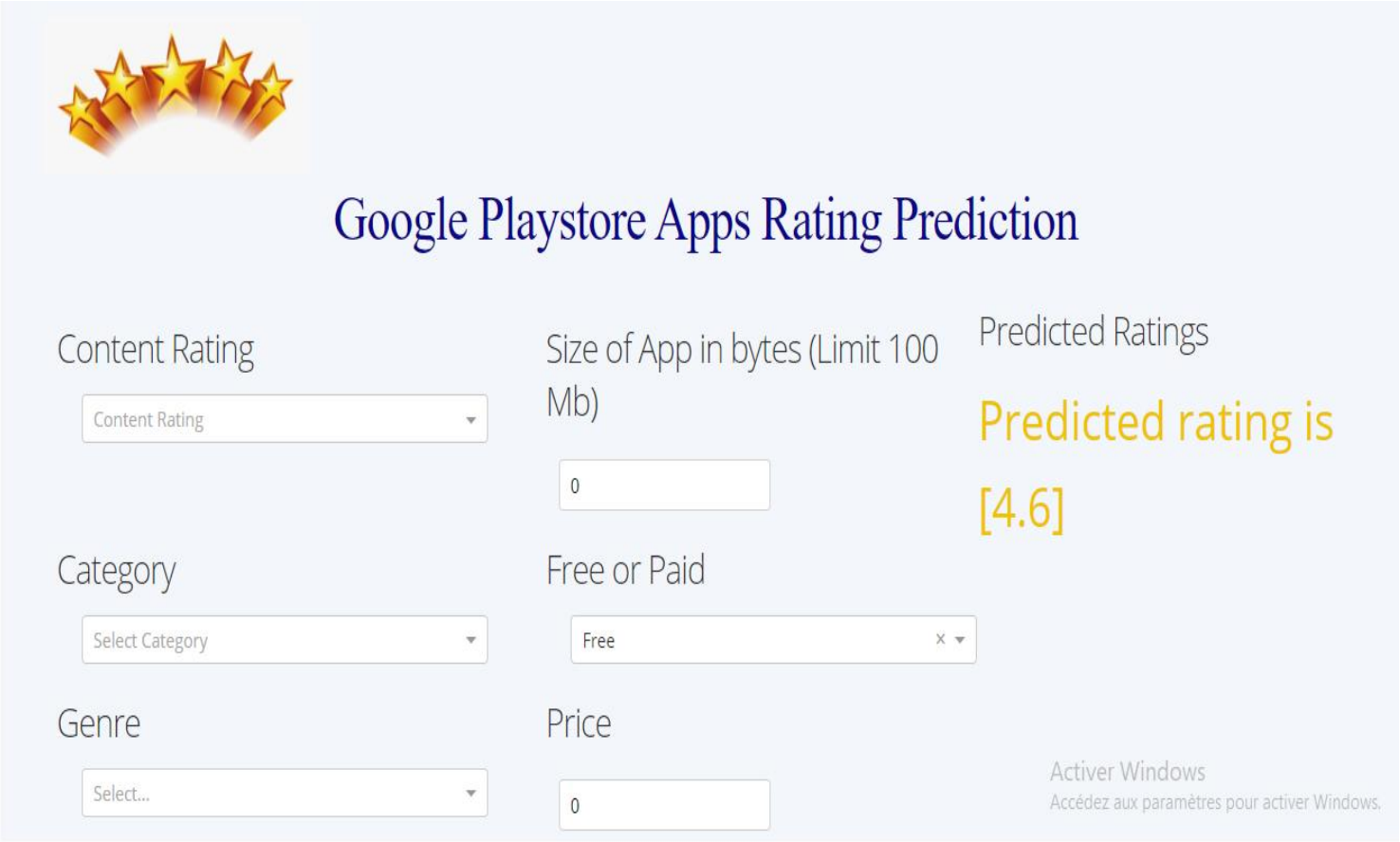


Figure 29: Formulaire de prédiction de rating

Conclusion

Dans ce chapitre, nous nous sommes m'intéresse à la réalisation des fonctionnalités de notre solution. Nous avons décelé cette réalisation à travers un ensemble d'interfaces accompagnées de description et interprétation. Nous clôturons notre travail par une conclusion qui résume notre travail et avise ses futures perspectives.

Conclusion générale et perspective

Les données des applications du *Google Play Store* sont volumineuses ce qui conduit les entreprises créatrices d'applications vers le succès. Les développeurs peuvent travailler et capturer le marché Androïde après l'exploitation des informations enregistrées dans *Google Play Store*.

Dans ce mémoire, nous avons commencé par la présentation des types d'apprentissages automatique avec des exemples d'algorithmes utilisés et les métriques de performances. Ensuite, nous avons collecté les données de *Kaggle* et passé par la phase de prétraitement des données des deux parties (La prédiction rating et la prédiction sentiment de *Google Play Store*). Après, nous avons passé à la phase de traitement (fouille de données), nous avons appliqué les algorithmes d'apprentissage et évalué chaque algorithme pour choisir le meilleur. Enfin nous avons développé une application de prédiction de *Google Play Store* ainsi son évaluation.

En effet ce projet nous a permis de comprendre et maitriser les algorithmes d'apprentissage ainsi leurs performances à l'aide du concept de Machine Learning étudiés durant les deux années du master « Data Science » à l'ISLAIB.

Finalement, notre travail ne s'arrête pas à ce niveau, nous allons continuer à travailler sur la prédiction avec la technique de *DeepLearning* et l'évaluation des données en temps réel.

Bibliographie

[1] https://fr.scribd.com/document/455692323/Apprendre-le-ML-en-une-semaine-pdf

[2] https://www.lebigdata.fr/machine-learning-et-big-data

[3] https://fr.scribd.com/document/455692323/Apprendre-le-ML-en-une-semaine-pdf

[4] https://www.andlil.com/definition-de-regression-lineaire-132481.html

[5] https://ieeexplore.ieee.org/abstract/document/1364002

[6] https://www.24pm.com/117-definitions/327-apprentissage-par-arbre-de-decision

[7] https://web.stanford.edu/~jurafsky/slp3/5.pdf

[8]https://web.iitd.ac.in/~sumeet/cs229-notes2.pdf

[9]https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[10] https://www.datacourses.com/evaluation-of-regression-models-in-scikit-learn-846/

[11] https://ichi.pro/fr/matrice-de-confusion-clairement-expliquee-17516543954375

[12] écric biernat et michel lutz, Data Science : Fondamentaux et étude de cas machine Learning avec python et r, éditions eyrolles ,2019.

[13] https://ieeexplore.ieee.org/document/6758829

[14] https://scikit-learn.org/stable/modules/compose.html

[15]https://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html

[16] https://ledatascientist.com/google-colab-le-guide-ultime/

[17] https://code.visualstudio.com/docs/editor/editingevolved

[18] https://www.techopedia.com/definition/30343/natural-language-toolkit-nltk

[19] https://docs.bokeh.org/en/latest/index.html

[20] http://www.python-simple.com/python-pandas/panda-intro.php

[21] https://www.geeksforgeeks.org/python-introduction-matplotlib/

[22] https://numpy.org/doc/stable/user/whatisnumpy.html

[23] https://www.datacamp.com/community/tutorials/pickle-python-tutorial

[24] https://ngrok.com/whatsnew

[25] https://pymbook.readthedocs.io/en/latest/flask.html

[26] https://dash.plotly.com/introduction

[27] https://www.tutorialspoint.com/plotly/plotly_package_structure.htm

[28] https://hackersandslackers.com/plotly-dash-with-flask/

Annexe : Prétraitement et visualisations de données

1. Description des données

	Rating	Reviews	Size	Price
count	9360.000000	9.360000e+03	7723.000000	9360.000000
mean	4.191838	5.143767e+05	22.970456	0.961279
std	0.515263	3.145023e+06	23.449629	15.821640
min	1.000000	1.000000e+00	0.008500	0.000000
25%	4.000000	1.867500e+02	5.300000	0.000000
50%	4.300000	5.955000e+03	14.000000	0.000000
75%	4.500000	8.162750e+04	33.000000	0.000000
max	5.000000	7.815831e+07	100.000000	400.000000

Figure 30: Description de données

2. L'analyse exploratoire des données

2.1. Gratuit vs payant

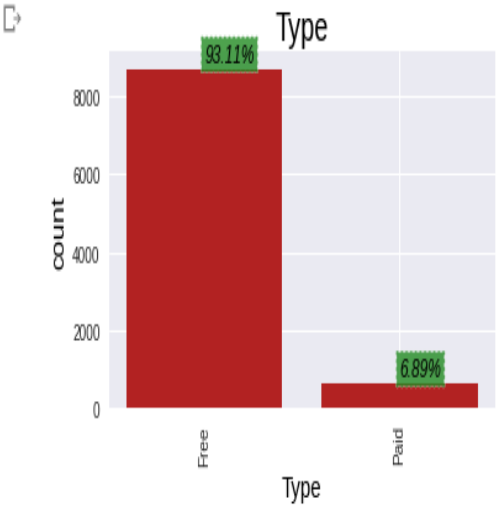
Ici, nous pouvons voir que 93,11% des applications sont gratuites et 6,89% les applications sont payées sur Google Play Store, nous pouvons donc dire que les applications sont gratuites sur Google Play Store

2.2. Nombre d'installations

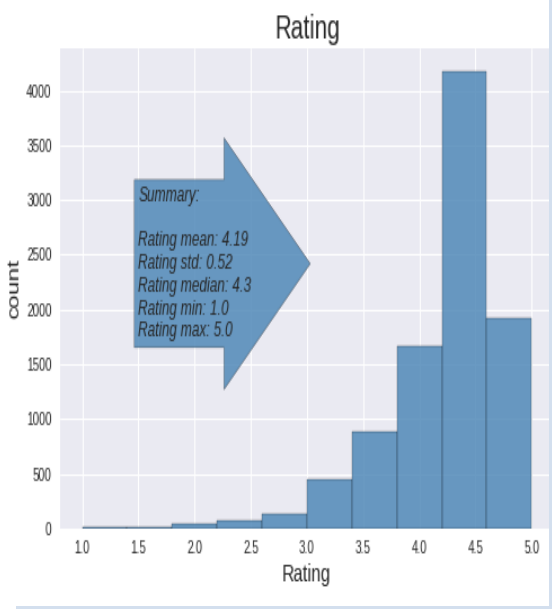
Des millions, puis 10 millions, très moins d'applications traversent le500M et rêve installer 1B. Certaines applications comme Instagram, YouTube, Facebook, WhatsApp, etc. traversent le rêve installez 1B.

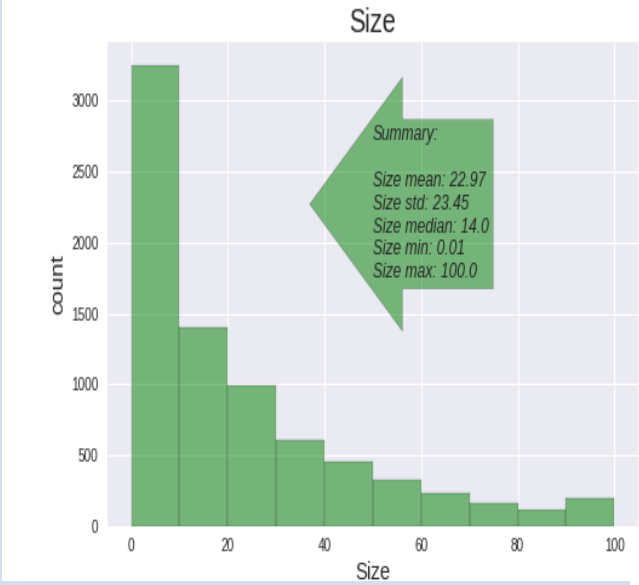
2.3 Catégorie

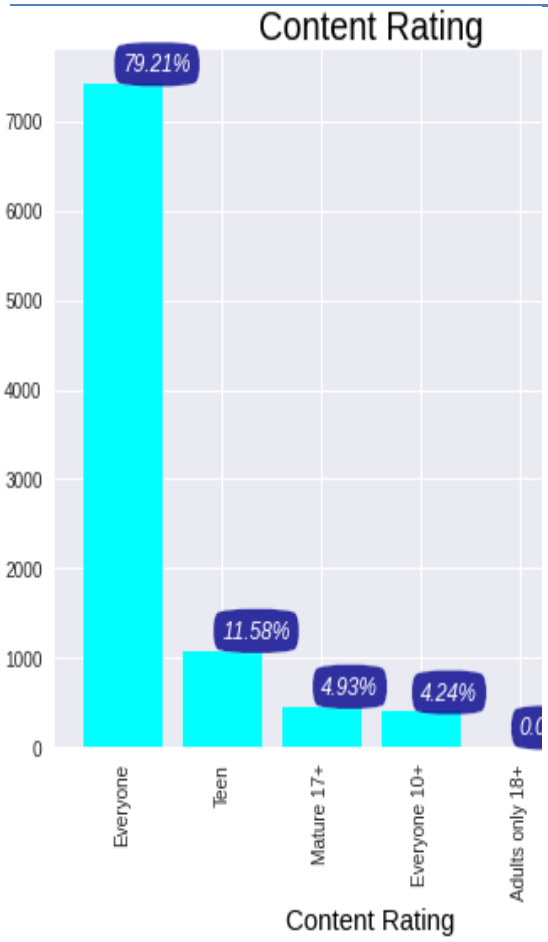
La catégorie Famille, jeux et outils a obtenu le plus haut d'application utilisé

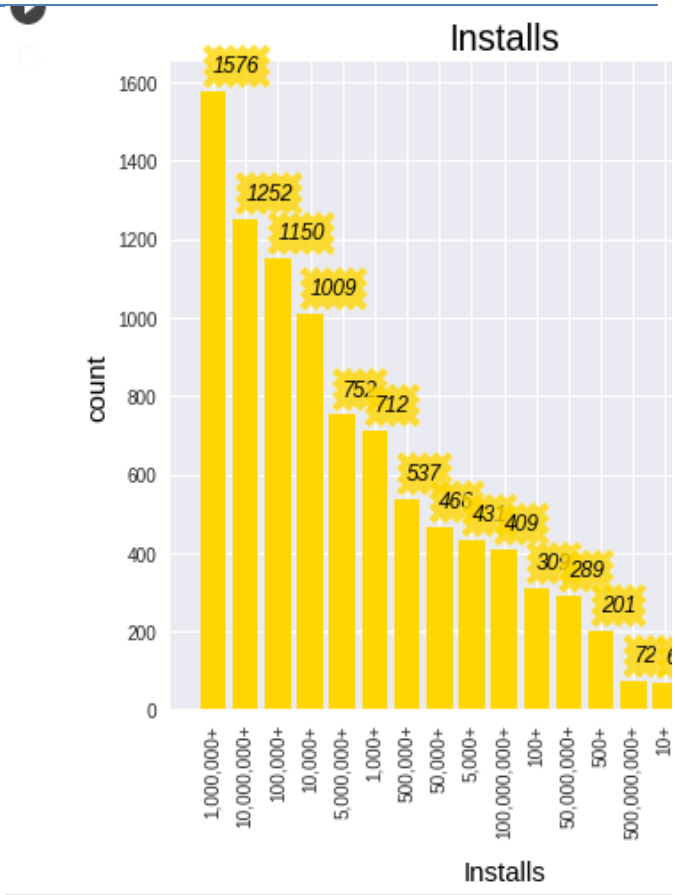

Type
93.11%
6.89%
count
Free
Paid
Type

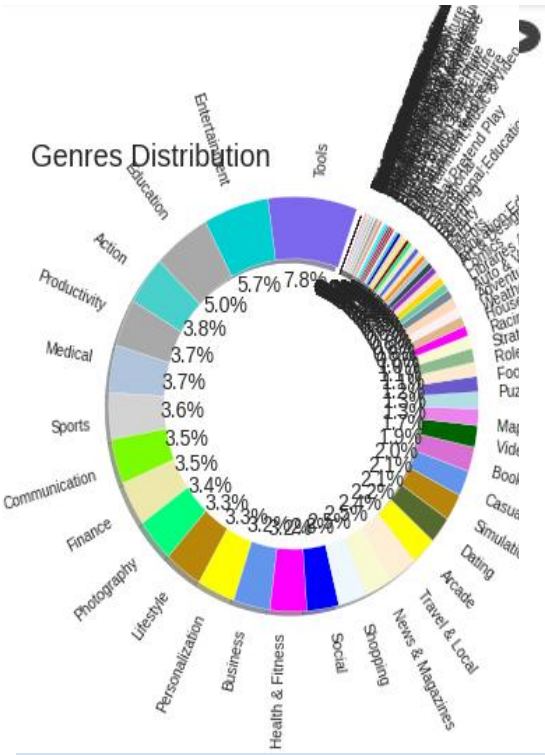

Price
Summary:
Price mean: 0.96
Price std: 15.82
Price median: 0.0
Price min: 0.0
Price max: 400.0
count

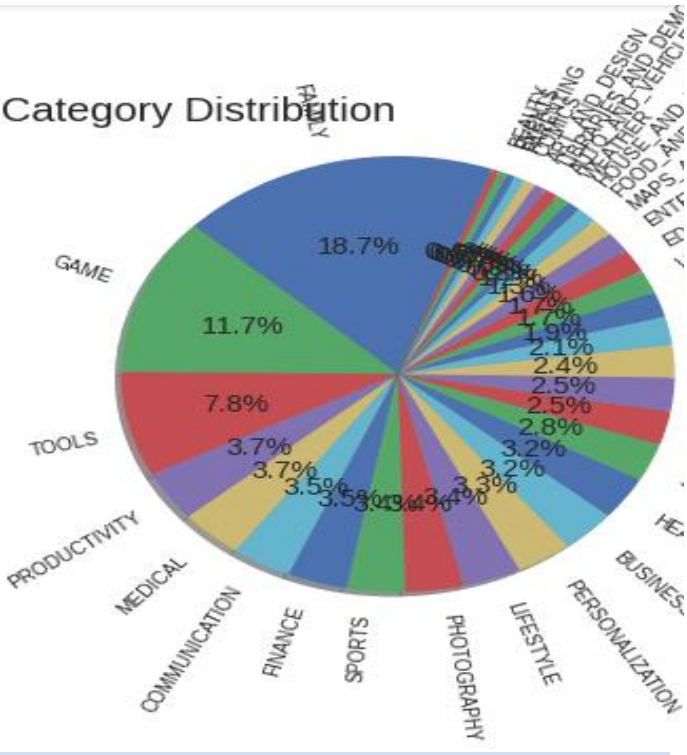

Rating
Summary:
Rating mean: 4.19
Rating std: 0.52
Rating median: 4.3
Rating min: 1.0
Rating max: 5.0
count
Rating


Size
Summary:
Size mean: 22.97
Size std: 23.45
Size median: 14.0
Size min: 0.01
Size max: 100.0
count
Size


Content Rating
79.21%
11.58%
4.93%
4.24%
0.0
7000
6000
5000
4000
3000
2000
1000
0
Everyone
Teen
Mature 17+
Everyone 10+
Adults only 18+
Content Rating


Installs
1576
1252
1150
1009
752
712
537
466
43
409
30
289
201
72
1600
1400
1200
1000
800
600
400
200
0
count
1,000,000+
10,000,000+
100,000+
10,000+
5,000,000+
1,000+
500,000+
50,000+
5,000+
100,000,000+
100+
50,000,000+
500+
500,000,000+
10+
Installs


Genres Distribution
Entertainment
Tools
Education
Action
Productivity
Medical
Sports
Communication
Finance
Photography
Lifestyle
Personalization
Business
Health & Fitness
Social
Shopping
News & Magazines
Travel & Local
Arcade
Dating
5.7%
7.8%
5.0%
3.8%
3.7%
3.7%
3.6%
3.5%
3.5%
3.4%
3.3%


Category Distribution
GAME
TOOLS
PRODUCTIVITY
MEDICAL
COMMUNICATION
FINANCE
SPORTS
PHOTOGRAPHY
LIFESTYLE
PERSONALIZATION
18.7%
11.7%
7.8%
3.7%
3.7%
3.5%
2.8%
2.5%
2.5%
2.4%
2.1%
3.2%
3.2%

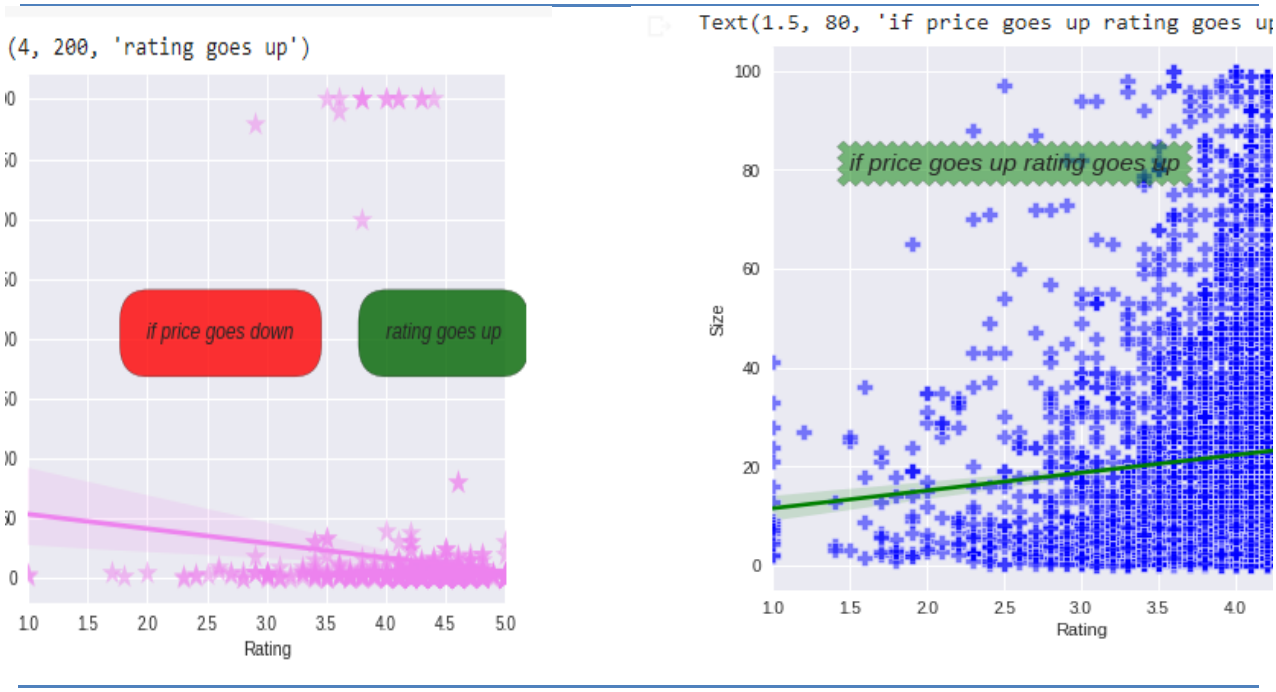


Figure 31: Visualisations de données

3. Matrice de corrélation

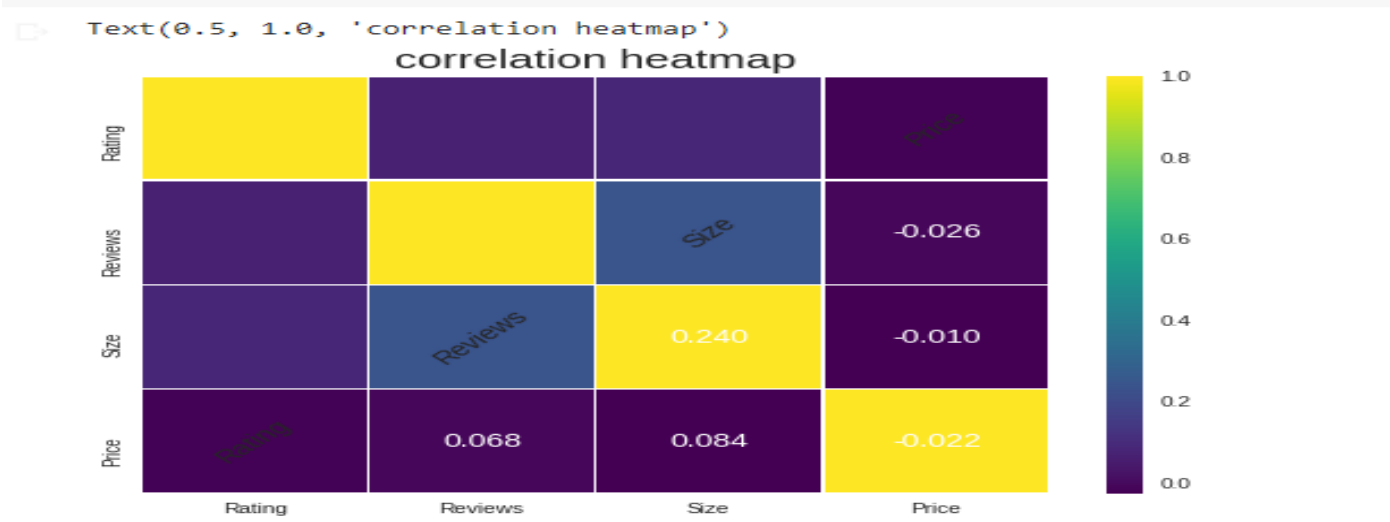


Figure 32: Matrice de corrélation