**Classification de personnages dans la trilogie originale Star Wars :**

**1. Introduction :**

La saga Star Wars est constituée de 3 trilogies, qui font intervenir de nombreux personnages, que l'on peut distinguer selon leur espèce, leur genre, ou encore leur appartenance à un groupe. La 1ère trilogie est constituée des épisodes 4, 5 et 6 pendant que la 2ème trilogie est constituée des épisodes 1, 2 et 3 qui racontent des événements antérieurs à la 1ère trilogie. Après l’achat des droits d’auteur de Star Wars par Walt Disney Company en 2012, le 7ème épisode est sorti en 2015, soit le premier épisode de la 3ème trilogie. Pour chacun des 7 épisodes que nous avons étudiés, nous avons pour données les dialogues sous forme de scripts complets, c’est-à-dire composés de dialogues et de didascalies et des caractéristiques concernant les personnages présents. Notre base de données contient un fichier par film, pour apporter plus d’informations au lecteur car ces fichiers sont différents des films et contiennent naturellement plus de détails afin de donner au lecteur l’intégralité des éléments nécessaires dont il a besoin, notamment des descriptions qui donnent plus de sens aux dialogues.

L'objectif de ce travail est de mettre en évidence les principaux indicateurs (âge, genre, champs lexicaux utilisés) qui influencent le nombre d'intervention et le temps de parole des personnages. Pour ce faire, nous créons une matrice de design des 6 premiers épisodes (afin de faire la prédiction sur le 7ème épisode) dont les colonnes sont : les paroles prononcées par chaque personnage, le pourcentage de paroles prononcées par chaque personnage, le pourcentage de mots prononcés par chaque personnage, le genre de chaque personnage et le nombre de fois où chaque personnage a prononcé les 5 mots les plus prononcés.

On pourra en déduire si la répartition du temps de parole est équilibrée ou pas. Pour cela, on s’est basé sur le nombre de mots prononcés par chaque personnage, cela signifie qu’un nombre de mots plus élevé correspond à un temps de parole plus long. Enfin, on essayera de répondre à la question suivante : d'après l'analyse des données (c’est-à-dire les données textuelles des 6 premiers épisodes), aurait-on pu prédire que le personnage principal de la dernière trilogie aurait été une femme ? Dans cette optique, on a importé les données brutes et fait le pré-traitement de ces données afin d’avoir une matrice de design sur laquelle on peut appliquer notre modèle. Ensuite on a procédé au traitement et à la visualisation des données en analysant les données textuelles de dialogues pour créer des indicateurs statistiques qui influencent le nombre d’intervention et le temps de parole des personnages. La dernière étape de ce projet consiste à utiliser une méthode d’apprentissage statistique et d’appliquer le modèle choisi à nos données afin de prédire si le personnage principal de la dernière trilogie aurait été une femme. Dans cette optique, nous avons choisi d’utiliser la régression logistique.

**2. Modélisation et exposition de la problématique :**

Dans un problème d'apprentissage supervisé, nous somme en générale en présence d'une une matrice de conception, c'est-à-dire, une matrice de données dont les lignes représentent les individus statistiques, et dont les colonnes se composent de variables explicatives X1,X2,...Xn, et d'une variable expliquée Y , ces variable peuvent être quantitatives ou qualitatives. Dans ce type d'apprentissage, les valeurs de la variable Y sont connu pour chaque observations, on dit que les observations sont étiquetées. L'objectif de l'apprentissage supervisé est de construire une fonction de prédiction à partir de ces observations étiquetées, c'est-à-dire, d'expliquer la variable Y par les variables (X1,X2,...Xn) an de pouvoir prédire, pour une observation non étiquetée, la valeur de la variable Y correspondante. Lorsque la variable Y est qualitative, on considère que ce problème de prédiction est un problème de classification.

Ce projet présente un problème de classification (la classification du genre des personnages de la saga "STAR WARS"), cependant, nous ne somme pas en présence de données exploitables automatiquement (sous forme d'une matrice de conception), mais en présence de données brute sous forme des scripts de la saga "STAR WARS". Ceci étant dit, une grosse partie de ce projet va être consacré au pré-traitement de ces données afin de les rendre exploitables automatiquement.

**3. Pré-traitement des données :**

Dans ce chapitre, nous allons procéder au pré-traitemment des données. Nous disposons d'un jeu de données sous forme de texte (les scripts de la saga), nous allons donc transformer ce jeu de données en données exploitables automatiquement, cela veut dire que nous allons les transformer en dataframe (sous forme d’une matrice).

**3.1 Importation des données :**

Afin d’avoir une idée plus claire sur les données brutes qu’on a eu initialement comme base de données, on se permet de mettre un petit extrait de l’épisode 4, dans la Figure 1 :

Une image contenant capture d’écran

Description générée automatiquement

Ces données ne contiennent pas seulement que les noms des personnages et leurs dialogues mais aussi des didascalies et des descriptions par rapport au décor, au lieu, au temps et aux caractéristiques des personnages. Elles n’étaient donc pas automatiquement exploitables. Notre base de de données n’est pas un dataframe, c’est-à-dire qu’elle n’est pas structurée comme une matrice dont les lignes correspondent aux individus et les colonnes aux différentes variables.

La première étape de ce projet sera de prendre en main les données (brutes) des scripts de la saga, et d'opérer les premiers traitements permettant de les transformer en données exploitables automatiquement.

Les 7 fichiers de films qu’on a nettoyés (dont on détaillera la démarche de nettoyage des données à la partie 3.2) ont tous le format CSV, c’est-à-dire un format de texte simple qui est utilisé dans de nombreux contextes lorsque de grandes quantités de données doivent être fusionnées sans être directement connectées les unes aux autres.

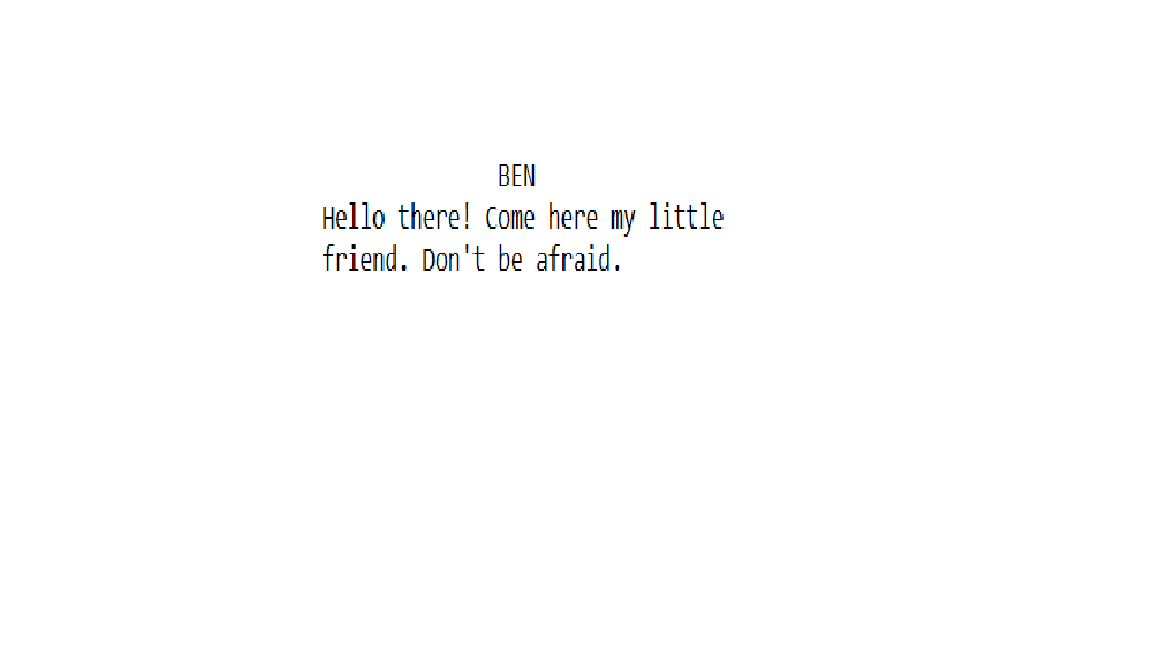
On a fait le pré-traitement uniquement sur l’épisode 4 pour ensuite appliquer les fonctions utilisées, lors de cette étape, sur les autres épisodes. On s'intéresse uniquement aux personnages et à leurs dialogues, donc on a d'abord fait le tri afin d'obtenir des données structurées faciles à exploiter, c’est-à-dire sous forme d’un dataframe auquel on peut appliquer une méthode d’apprentissage statistique.

Afin de vous expliquer notre démarche, nous avons pris un petit extrait de l’épisode 4, dans la Figure ci-dessous :

Une image contenant capture d’écran

Description générée automatiquement

À partir de ces scripts complets, on souhaite extraire dans un premier temps uniquement les noms de personnages et les dialogues. Dans un deuxième temps, on séparera ensuite les personnages de leurs dialogues, pour former une matrice à deux colonnes. Pour ce faire, on va créer ce qu'on appelera un motif (pattern) qui nous permet de faire une recherche d'expression régulière, ce motif permet d'extraire la suite des couples : (noms de personnages, dialogues prononcées). On a tout d’abord constaté que le texte n’est pas aligné de la même façon, on a ainsi remarqué que les discours des personnages sont indentés de 25 espaces suivis d'un espace. Afin de faire le tri grâce à ce critère, on a utilisé Python comme language informatique. Dans un premier temps, on a téléchargé le package re (regular expression) qui est utilisé pour manipuler des chaines de caractères. Ce package nous donne accès à la fonction re.compiler qui permet de faire la compilation des chaines de caractères. Les expressions régulières sont compilées en objets motifs, qui contiennent différentes fonctions pour effectuer différentes tâches qui peuvent inclure la recherche, la correspondance, le remplacement ou les substitutions dans les chaines. Cette fonction prend pour entrée des groupes de chaines de caractères. On a ainsi défini 3 groupes dans la fonction re.compiler pour créer un motif régulier. Le 1er groupe correspond à 25 espaces (qui précèdent les discours des personnages), le 2ème groupe correspond à un ensemble de mots et/ou de chiffres quelconque et le 3ème groupe correspond à un seul espace (qui suit les discours des personnages). On a ensuite utilisé la fonction finditer du package re afin de récupérer uniquement le 2ème groupe qui correspond à toute phrase précédée par 25 espaces et suivie par un espace, à savoir les noms de personnages et leurs dialogues. On rappelle que la fonction finditer est utilisée pour trouver le motif défini par la classe re dans les chaines de caractères ainsi que l’emplacement des chaines de caractères correspondantes qui est l’index des chaines de caractères. Cette fonction itère en fait les chaines correspondantes et retourne les index ou les emplacements de la chaine. Ainsi on récupère uniquement les noms de personnages et leurs dialogues. On obtient un résultat de cette forme-là, dans la Figure ci-dessous :

 On a remarqué que le nom de personnage 'BEN' est caractérisé par le fait qu'il est précédé par 14 espaces, (d’ailleurs c’est le cas pour tous les noms de personnages), on a donc utilisé le même concept que précédemment afin de séparer les noms de personnages de leurs dialogues, cela veut dire qu’on a créé une expression régulière pour récupérer les noms de personnages grâce aux fonctions re.compiler et finditer. On a ensuite listé les noms de personnages dans une liste. Tous les éléments qui ne sont pas dans cette liste correspondent aux dialogues des personnages, ainsi on a récupéré les dialogues en utilisant la fonction join qui permet la concaténation des chaines de caractères dans une liste. On rappelle que le pré-traitement des données est appliqué initialement uniquement sur l’épisode 4.

Pour cela, on a créé une fonction qui permet de faire ces manipulations. Ainsi on a obtenu un dataframe qui contient les noms de personnages sur la 1ère colonne et leurs dialogues respectifs sur la 2ème colonne, comme on peut le voir dans l'exemple Figure 2. Cette fonction va par la suite être appliquée aux autres fichiers films correspondant aux films 1,2,3,5 et 6 en modifiant les paramètres permettant d’identifier les noms de personnages et leurs dialogues pour chaque fichier. On tient à préciser que les personnages n’apparaissent pas une seule fois, ils apparaissent sur la matrice selon leurs dialogues, par exemple si un personnage intervient au 5ème au 11ème dialogues il va apparaitre sur la 5ème et la 11ème lignes de notre matrice, ainsi de suite.

Afin de manipuler des dataframes, on a utilisé la librairie Pandas qui fait partie des librairies de base pour la science des données en Python. Pandas fournit des structures de données puissantes et simples à utiliser, ainsi que les moyens d'opérer rapidement des opérations sur ces structures. Elle permet d’importer des données dans un dataframe, modifier ce dataframe afin de pouvoir le visualiser et en retirer de la valeur. Pandas a permis de résoudre certaines difficultés rencontrées lors de l’utilisation de Numpy, par exemple le chargement à partir d’un fichier CSV exigeait que le contenu de chaque colonne soit une chaine de caractères et cela posait problème quand il y avait une colonne contenant des données numériques. Il s’agit d’un outil très pratique pour visualiser et analyser les données d’un dataframe.

Afin d’avoir une idée plus claire sur le dataframe obtenu pour l’épisode 4, on joint ci-dessous un extrait de ce dataframe, il s’agit d’un tableau composé de 1002 lignes et 2 colonnes :

Une image contenant capture d’écran

Description générée automatiquement