Classification de personnages dans la trilogie originale Star Wars

Yassine SAYD, Abdesstar SAHBANE

Université Montpellier II

30/05/2020

Table des matières

Introduction

L'objectif de ce projet est de mettre en évidence les principaux indicateurs statistiques (genre, champs lexicaux utilisés) qui influencent le nombre d'intervention et le temps de parole des personnages. On essayera de répondre à la question suivante : d'après l'analyse des données textuelles des 6 premiers épisodes, aurait-on pu prédire que le personnage principal du 7ème épisode aurait été une femme ? On cherche donc à prédire une modalité, ici le genre, ce qui constitue un problème de classification supervisé.

Pré-traitemment des données

Présentation des données

```
Artoo moves even tighter into the shadows as the slight
swishing sound that frightened off the Sandpeople grows even
closer, until a shabby old desert-rat-of-a-man appears and
leans over Luke. His ancient leathery face, cracked and
weathered by exotic climates is set off by dark, penetrating
eyes and a scraggly white beard. Ben Kenobi squints his eyes
as he scrutinizes the unconscious farm boy. Artoo makes a
slight sound and Ben turns and looks right at him.

                        BEN
              Hello there! Come here my little
              friend. Don't be afraid.

Artoo waddles over to were Luke lies crumpled in a heap and
begins to whistle and beep his concern. Ben puts his hand on
Luke's forehead and he begins to come around.
```

Extrait du script de l'épisode 4 de la saga Star Wars.

Pré-traitement des données

Importation des données

- Expressions régulières.
- Le package RE :
  re.compile
  finditer
- Application à nos données.

Pré-traitement des données

Importation des données

```
pattern = re.compile(r'(\s{25})(.+)(\s)')
matches = pattern.finditer(text_to_search)
for match in matches :
    print(match.group(2))
```

```
                    BEN
Hello there! Come here my little
friend. Don't be afraid.
```

Extrait du code appliqué à l'épisode 4.

Pré-traitement des données

Importation des données

	Name	parole
0	THREEPIO	Did you hear that? They've shut down the main ...
1	THREEPIO	We're doomed!
2	THREEPIO	There'll be no escape for the Princess this time.
3	THREEPIO	What's that?
4	THREEPIO	I should have known better than to trust the l...
...	...	...
997	LUKE	Oh, no!
998	THREEPIO	Oh, my! Artoo! Can you hear me? Say something!...
999	TECHNICIAN	We'll get to work on him right away.
1000	THREEPIO	You must repair him! Sir, if any of my circuit...
1001	LUKE	He'll be all right.

Extrait du dataframe obtenu pour l'épisode 4.

Pré-traitement des données

Nettoyage des données

- Le package Nltk.
- Tokenisation.
- Lemmatisation.

Pré-traitement des données

Tokenisation

```
## On va définir notre chaine de caractère
s = "Did you hear that? They've shut down the main reactor. We'll be destroyed for sure. This is madness!"

## Nous allons définir l'expression régulière à utiliser pour diviser la chaine de caractère
tokenizer = RegexpTokenizer(r'\w+') ## r'\w+' est l'expression régulière désignant un alphanumérique

## Nous allons appliquer la méthode tokenize() sur notre chaine de caractère
words = tokenizer.tokenize(s)

## affichage des résultats
print(s)
print(words)
```

```
Did you hear that? They've shut down the main reactor. We'll be destroyed for sure. This is madness!
['Did', 'you', 'hear', 'that', 'They', 've', 'shut', 'down', 'the', 'main', 'reactor', 'We', 'll', 'be', 'destroyed', 'for',
'sure', 'This', 'is', 'madness']
```

Exemple de Tokenisation d'une phrase.

```
['did', 'you', 'hear', 'that', 'they', 've', 'shut', 'down', 'the', 'main', 'reactor', 'we', 'll', 'be', 'destroyed', 'for',
'sure', 'this', 'is', 'madness']
['hear', 'shut', 'main', 'reactor', 'destroyed', 'sure', 'madness']
```

Exemple de suppression des mots vides.

Pré-traitement des données

Lemmatisation

```
## Importation du package
from nltk.stem import WordNetLemmatizer

## list des mots
list_mots = ['move', 'moves']

## affectation du lemmatiseur
lemmatizer = WordNetLemmatizer()

## liste vide qui contiendra les lemmes
list_lemmes = []

## stocker le lemme de chaque mots de la liste dans la liste list_lemmes
for i in range (len(list_mots)):
        list_lemmes.append(lemmatizer.lemmatize(list_mots[i]))

## affichage des résultats
print(list_mots)
print(list_lemmes)
```

```
['move', 'moves']
['move', 'move']
```

Exemple de Lemmatisation des mots.

Matrice de design

Indicateurs statistiques

- Nombre de paroles prononcées par chaque personnage.
- Nombre de mots prononcés par chaque personnage.
- Les 5 mots les plus prononcés.

Matrice de design

Variables explicatives

- Poucentage de paroles prononcées par chaque personnage.
- Poucentage de mots prononcés par chaque personnage.
- Le nombre de fois où chaque personnage a prononcé les 5 mots les plus prononcés.

Matrice de design

Matrice de design avec les 5 mots les plus prononcés

	perc_parole	perc_mots	master	jedi	going	get	right	gender
0	0.120813	0.100742	71.0	44.0	33.0	19.0	25.0	0
1	0.086766	0.070883	7.0	6.0	30.0	25.0	32.0	0
2	0.084020	0.074269	42.0	18.0	17.0	16.0	19.0	0
3	0.081457	0.073548	0.0	1.0	34.0	45.0	24.0	0
4	0.069559	0.074701	4.0	19.0	16.0	6.0	9.0	1
5	0.065166	0.072108	32.0	3.0	24.0	11.0	10.0	0
6	0.042651	0.070199	13.0	42.0	7.0	5.0	6.0	0
7	0.038623	0.028382	0.0	0.0	8.0	13.0	4.0	1
8	0.036610	0.036162	1.0	9.0	3.0	10.0	4.0	0
9	0.030752	0.035370	11.0	13.0	0.0	0.0	2.0	0

Matrice de design

Matrice de design avec les 5 mots les plus importants

	perc_parole	perc_mots	master	jedi	ship	force	power	gender
0	0.206853	0.172392	0.0	0.0	1.0	1.0	1.0	0
1	0.184010	0.145288	1.0	1.0	4.0	1.0	1.0	1
2	0.173858	0.180901	0.0	3.0	3.0	2.0	2.0	0
3	0.077411	0.090451	0.0	0.0	1.0	0.0	0.0	0
4	0.074873	0.076899	0.0	0.0	1.0	2.0	2.0	0
5	0.032995	0.046328	0.0	0.0	1.0	0.0	0.0	0
6	0.030457	0.030570	0.0	1.0	0.0	1.0	0.0	1
7	0.024112	0.040340	0.0	1.0	0.0	1.0	0.0	1
8	0.017766	0.019855	1.0	1.0	0.0	0.0	0.0	0
9	0.017766	0.039395	3.0	0.0	0.0	0.0	1.0	0

Traitement des données

Notons : $X = (X_1, X_2, \ldots, X_p)$.

La régression logistique

- $\mathbb{P}(Y = 1 \mid X) = f(X_1, X_2, \ldots, X_p) + \epsilon = \ ?$

La régression logistique se base sur l'estimation du rapport suivant :

$$\frac{\mathbb{P}(Y = 1 \mid X)}{1 - \mathbb{P}(Y = 1 \mid X)} \tag{1}$$

Traitement des données

Le modèle logistique

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right). \tag{2}$$

La régression logistique introduit l'hypothèse fondamentale suivante :

$$\text{logit}\left(\mathbb{P}(Y=1 \mid X)\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n. \tag{3}$$

Par conséquent :

$$\ln\left(\frac{\mathbb{P}(Y=1 \mid X)}{1-\mathbb{P}(Y=1 \mid X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n. \tag{4}$$

Traitement des données

Estimation des coefficients

On obtient donc la formule suivante :

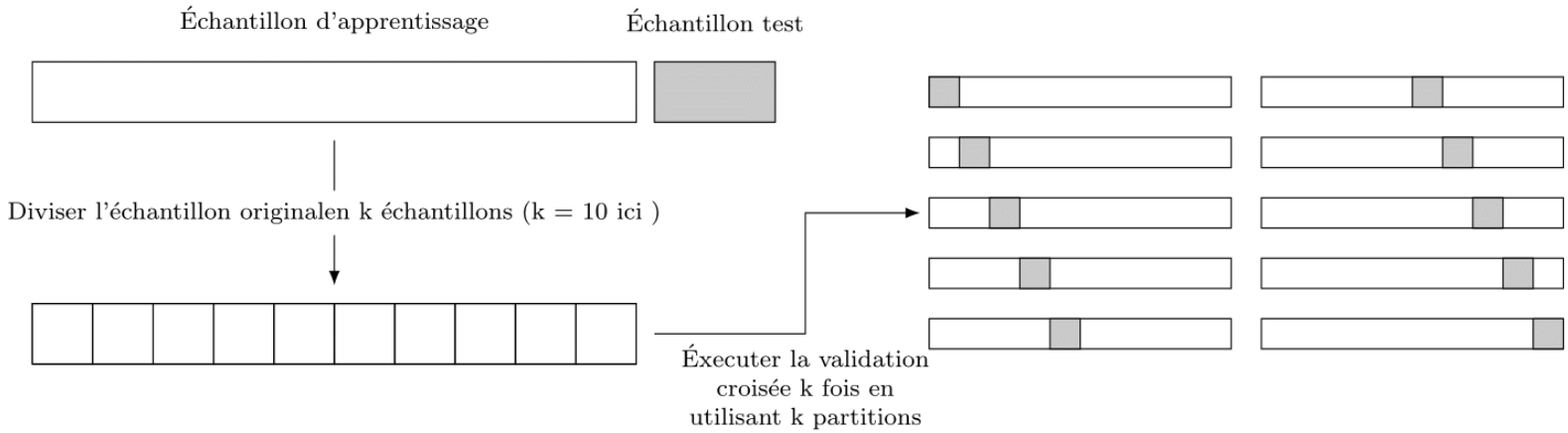
$$\mathbb{P}(Y = 1 \mid X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 - e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}. \tag{5}$$

Les coeffcients $\beta_0, \beta_1, \ldots, \beta_p$ sont inconnus et doivent être estimés en se basant sur les données d'apprentissage. Ils sont estimés grâce à l'estimateur du maximum de vraisemblance.

Prédiction

Validation croisée

- On divise l'échantillon original en k sous-échantillons.
- Chaque sous-échantillon sert successivement d'échantillon test, le rest sert d'échantillon d'apprentissage .



Prédiction

Insuffisance des données

- Le nombre des personnages féminines (11) présentes dans l'ensemble des individus ne représente que 15% du total des individus.
- Nos données sont donc déséquilibré.
- "Stratified k-Fold cross validation" : faire en sorte de garder le même pourcentage de chaque classe dans tous les sous-échantillons.

Prédiction

Résultats de la prédiction

```
Confusion Matrix:
[[11  0]
 [ 2  0]]
Confusion Matrix:
[[9 2]
 [1 0]]
Confusion Matrix:
[[10  0]
 [ 2  0]]
Confusion Matrix:
[[10  0]
 [ 1  1]]
Confusion Matrix:
[[9 1]
 [1 1]]
Confusion Matrix:
[[9 1]
 [2 0]]
0.6842105263157895
```

Conclusion

Conclusion

- Nos données sont insuffisantes.
- Un modèle de classification est plus performant quand les données sont équilibrées et quand le nombre d'observations est élevé.
- Notre modèle n'est pas performant donc pas de prédiction.