

Machine Learning



Rajmagan BALAKICHENIN
M2 ISE

Sommaire

INTRODUCTION

PHASE D'ENTRAÎNEMENT ET VALIDATION

TEST

CONCLUSION

INTRODUCTION

Comment prédire les émotions positive et négative ?

Prédire les sentiments négative et positive sur les satisfactions des clients, pour les compagnies aériennes sur Tweeter.

Technologies utilisées :
Langage Python3, Bibliothèque Keras

Etape :
Entraînement
Validation
Test

ENTRAINNEMENT

Corpus pour l'apprentissage → 12 591 lignes
80% du corpus

1	airline_sentiment	text
2	1	@VirginAmerica What @dhepburn said.
3	1	@VirginAmerica plus you've added commercials to the experience... tacky.
4	1	@VirginAmerica I didn't today... Must mean I need to take another trip!
5	0	@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces & they have little recou
6	0	@VirginAmerica and it's a really big bad thing about it
7	0	@VirginAmerica seriously would pay \$30 a flight for seats that didn't have this playing. it's really the only bad thing about flying VA
8	1	@VirginAmerica yes, nearly every time I fly VX this ĀĀ□Ā□ear wormĀĀ□Ā□ wonĀĀ□Ā□t go away :)
9	1	@VirginAmerica Really missed a prime opportunity for Men Without Hats parody, there. https://t.co/mWpG7grEZP
10	1	@virginamerica Well, I didn'tĀĀ□Ā but NOW I DO! :-D
11	1	@VirginAmerica it was amazing, and arrived an hour early. You're too good to me.
12	1	@VirginAmerica did you know that suicide is the second leading cause of death among teens 10-14
13	1	@VirginAmerica I <

Le corpus est constitué de 2 colonnes :
airline_sentiment : 0 sentiment négative – 1 sentiment non négative
Text : avis client

ENTRAINEMENT

Tokenization pour diviser mot par mot les phrases

Créer le modèle puis les couches

```
#Diviser le jeu de donnée 80% pour l'entrainement et 20% pour valider
X_train, X_test, Y_train, Y_test = train_test_split(encoded_docsX, Y2, test_size=0.20,
random_state=7)

nbre_ligne = len(tokenizer.word_index) + 1

# create model
model = Sequential()
model.add(Dense(1000, input_dim=nbre_ligne, activation='relu'))
model.add(Dense(500, activation='relu'))
model.add(Dense(250, activation='relu'))
model.add(Dense(2, activation='sigmoid'))
```

VALIDATION

20% du corpus pour validation
3 147 lignes

Prédire les sentiments du reste du corpus
0 ← sentiment négatif
1 ← sentiment positif

Utiliser la descente du gradient pour pour ajuster notre modèle :

```
#Evaluer le model  
loss, accuracy = model.evaluate(X_test, Y_test, verbose=False)  
print("Loss: ", loss, " Accuracy: ", accuracy)
```

TEST

Pour réaliser des tests :

Utiliser un autre corpus (1 153 lignes)

Prédire émotions négatives ou positives

Prédiction en temps réel

Données entrées par l'utilisateur

```
Entrer qqche:  
very rude with their customer  
[0]
```

```
Entrer qqche:  
good service !!  
[1]
```

CONCLUSION

Fait :

Réalisation du modèle

Réalisation des prédictions pour les tests

A améliorer :

Améliorer la fiabilité du programme

Améliorer la qualité du corpus