

## Master Spécialisé Ingénierie de Données et Développement Logiciel



# RAPPORT DU PROJET

Analyse du sentiment des revues des compagnies d'assurance

Réalisé par:

Tarik OUHAMMOU  
Abdelkhaleq ELHADDAD

Encadre par :

Pr. Mahmoudi Abdelhak

ANNEE ACADEMIQUE : 2020/2021

## Sommaire:

1. INTRODUCTION .....	3
2. ANALYSE DES SENTIMENTS .....	4
2.1. DEFINITION .....	4
2.2. EXEMPLE .....	4
3. IMPLEMENTATION.....	5
3.1. OUTILS .....	5
3.1.1. COLAB : .....	5
3.2. TEXTBLOB .....	5
3.3. RandomForest .....	5
3.4. CODE.....	6
Install est l'importation des bibliothèques.....	6
Supprimer la colonne sans nom .....	6
Vérifier les valeurs NAN.....	6
Convertir les données en minuscules.....	6
Tokenisation des mots et suppression de la ponctuation.....	6
Ajouter une nouvelle colonne à nos données.....	6
ensemble de mots vides par défaut de Spacy et supprimer les mots de négation .....	7
Supprimer les mots vides .....	7
Ajouter une nouvelle colonne : commentaire après le prétraitement .....	7
Analyse des sentiments avec TextBlob.....	7
Ajouter une colonne : sentiment.....	7
Built Machine learning Module (RandomForest) .....	7
Division des données en ensembles d'apprentissage et de test.....	8
Training the Model .....	8
Faire des prédictions et évaluer le modèle .....	8
4. Conclusion .....	9

## 1. INTRODUCTION



Des études récentes montrent que de nombreuses compagnies d'assurance automobile, habitation et vie ont enduré plusieurs années de croissance contrainte. Cependant, il existe de multiples opportunités pour ces compagnies d'acquérir un avantage concurrentiel par une meilleure compréhension des attitudes des clients envers leurs services. Dans cette optique, les sentiments des gens pourraient fournir aux compagnies d'assurance des informations précieuses. Par exemple, l'analyse des sentiments peut les aider à réagir avec des stratégies de marketing et de relations publiques appropriées.

Dans le contexte du secteur de l'assurance, aucune étude n'a utilisé l'analyse des sentiments pour identifier les attitudes des consommateurs envers les services d'assurance.

Le premier objectif de notre projet est d'effectuer une analyse des sentiments des consommateurs afin de mieux comprendre les attitudes et les émotions globales des consommateurs envers différents types de services d'assurance.

Le présent rapport résume notre période de réalisation du ce programme pour l'analyse des sentiments (via des données CSV)

## 2. ANALYSE DES SENTIMENTS

### 2.1. DEFINITION

Analyse des sentiments (aussi appelé opinion mining) : consiste à déterminer si l'attitude d'une personne à propos d'un sujet ou du produit est positive, négative ou neutre, aussi de savoir le ressenti général d'une communauté.

Ce procédé apparaît au début des années 2000 et connaît un succès grandissant dû à l'abondance de données provenant de réseaux sociaux, notamment celles fournies par **Twitter**.

### 2.2. EXEMPLE



Neutre

Python ou le R n'ont pas d'importance pour moi, je maîtrise les deux.



Python est magnifique ! j'aime la simplicité.



Positive



Négative

J'aime Python, mais je déteste le R.



## 3. IMPLEMENTATION

### 3.1. OUTILS

#### 3.1.1. 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. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. Cool, n'est-ce pas ? Avant de présenter ce magnifique service, nous rappellerons ce qu'est un Jupyter Notebook.

### 3.2. TEXTBLOB

TextBlob : est une bibliothèque Python pour le traitement de données textuelles. Elle fournit une API simple pour accéder à ses méthodes et effectuer des tâches de traitement du langage naturel (NLP) telles que le balisage de parties de discours, l'extraction de phrases nominatives, l'analyse des sentiments, la classification, la traduction, ...

### 3.3. RandomForest

RandomForest est une méthode d'apprentissage d'ensemble pour la classification (et la régression) qui opère en construisant une multitude d'arbres de décision au moment de l'apprentissage et en produisant la classe qui est le mode des classes sorties par des arbres individuels

### 3.4. CODE

#### Install est l'importation des bibliothèques

```
!pip install textblob_fr

import pandas as pd
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize
import re
import spacy
from spacy.lang.fr.stop_words import STOP_WORDS
import string
from textblob import Blobber
from textblob_fr import PatternTagger, PatternAnalyzer
tb = Blobber(pos_tagger=PatternTagger(), analyzer=PatternAnalyzer())
import plotly.graph_objects as go
import plotly.express as px
```

```
data = pd.read_csv('/content/Comments.csv')
```

#### Supprimer la colonne sans nom

```
data = data.drop(['Unnamed: 0'], axis=1)
```

#### Vérifier les valeurs NAN

```
NAN = [(c, data[c].isna().mean()*100) for c in data]
NAN = pd.DataFrame(NAN, columns=["column_name", "percentage"])
NAN.sort_values("percentage", ascending=False)
```

#### Convertir les données en minuscules

```
data["Comment"] = data["Comment"].str.lower()
```

#### Tokenisation des mots et suppression de la ponctuation

```
AComment=[]
for comment in data["Comment"].apply(str):
    Word_Tok = []
    for word in re.sub("\W", " ", comment).split():
        Word_Tok.append(word)
    AComment.append(Word_Tok)
```

#### Ajouter une nouvelle colonne à nos données

```
data["Word_Tok"] = AComment
```

## ensemble de mots vides par défaut de Spacy et supprimer les mots de négation

```
stop_words=set(STOP_WORDS)

deselect_stop_words = ['n\\'', 'ne', 'pas', 'plus', 'personne', 'aucun', 'ni', 'aucune', 'rien']
for w in deselect_stop_words:
    if w in stop_words:
        stop_words.remove(w)
    else:
        continue
```

## Supprimer les mots vides

```
AllfilteredComment=[]
for comment in data["Word_Tok"]:
    filteredComment = [w for w in comment if not ((w in stop_words) or (len(w) == 1))]
    AllfilteredComment.append(' '.join(filteredComment))
```

## Ajouter une nouvelle colonne : commentaire après le prétraitement

```
data["CommentAferPreproc"]=AllfilteredComment
```

## Analyse des sentiments avec TextBlob

```
senti_list = []
Labels = []
for i in data["CommentAferPreproc"]:
    vs = tb(i).sentiment[0]
    if (vs > 0):
        senti_list.append('Positive')
        Labels.append(1)
    elif (vs < 0):
        senti_list.append('Negative')
        Labels.append(-1)
    else:
        senti_list.append('Neutral')
        Labels.append(0)
```

## Ajouter une colonne : sentiment

```
data["sentiment"]=senti_list
```

## Built Machine learning Module (RandomForest)

```
from sklearn.feature_extraction.text import TfidfVectorizer
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

vectorizer = TfidfVectorizer (max_features=2500, min_df=0.2, max_df=0.8, stop_words=stopwords.words('french'))
processed_features = vectorizer.fit_transform(data["CommentAferPreproc"]).toarray()
()
```

## Division des données en ensembles d'apprentissage et de test

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(processed_features, Labels, test_size=0.2, random_state=0)
```

## Training the Model

```
from sklearn.ensemble import RandomForestClassifier
```

```
text_classifier = RandomForestClassifier(n_estimators=200, random_state=0)  
text_classifier.fit(X_train, y_train)
```

## Faire des prédictions et évaluer le modèle

```
predictions = text_classifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score  
  
print(confusion_matrix(y_test, predictions))  
print(classification_report(y_test, predictions))  
print(accuracy_score(y_test, predictions))
```

```
[[ 17   0  710]  
 [   2   0  313]  
 [  18   0 1141]]  
      precision    recall  f1-score   support  
  
   -1         0.46      0.02      0.04         727  
    0         0.00      0.00      0.00         315  
    1         0.53      0.98      0.69        1159  
  
 accuracy          0.53         2201  
 macro avg          0.33         0.24         2201  
weighted avg          0.43         0.53         2201
```

```
0.526124488868696
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: UndefinedMetricWarning:
```

```
Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
```



## 4. Conclusion

L'analyse des sentiments a permis de mieux comprendre l'attitude des consommateurs à l'égard de différents services d'assurance. Nous avons constaté que les consommateurs avaient des attitudes plus positives envers les services d'assurance.

Cependant, depuis lors, ils ont eu des attitudes plus négatives envers les services d'assurance automobile, habitation et vie. Cela pourrait être l'une des raisons pour lesquelles les compagnies d'assurance ont enduré plusieurs années de croissance contrainte. Nous avons également extrait les émotions des consommateurs pour chaque type d'assurance et avons constaté que les consommateurs ont généralement des émotions plus négatives envers les services d'assurance.

De plus, nous avons trouvé les mots les plus fréquents dans les commentaires des consommateurs qui façonnent leurs émotions pour différents types de services d'assurance. Les résultats ont également montré que le sentiment des avis des consommateurs peut prédire avec précision les évaluations des avis.