INPT
المعهد الوطني للبريد والمواصلات
ⵜⴰⵙⴷⴰⵡⵜ ⵏ ⵜⵣⵔⴼⵜ ⴰⵏⴰⵎⵓⵔ ⵏ ⵜⵙⵡⵉⵍⵜ
Institut National des Postes et Télécommunications

MACHINE LEARNING

Twitter sentiment analysis

Analyze the sentiment of the tweets

Realized by :
ISBAINE Mohamed & **LABRIJI** Saad

Supervised by :
FISSAA TARIK

TABLE OF CONTENTS

Introduction & problem statement

01

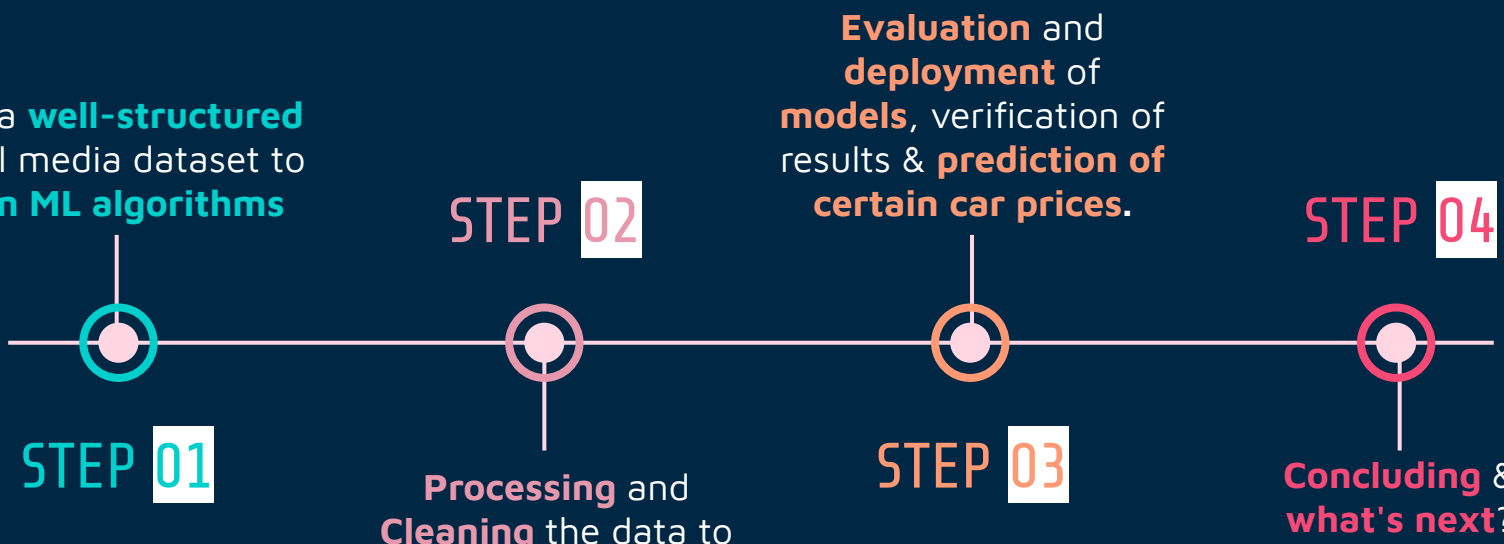
Introduction & problem statement

In social networks, **twitter** for example, is the ideal place for anyone to post their ideas and thoughts, but t**o avoid the spread of hateful tweets**, it is absolutely necessary to have an **automated system capable of detecting the nature of the feelings behind a text**. To solve this kind of problem, we thought my teammate and I that we could use the modest knowledge we acquired in data science courses this year, **to develop an Automated Machine Learning Sentiment Analysis Model in order to compute the customer perception**.

Divide Long-Term Goals into Smaller Tasks

02

Divide Long-Term Goals into Smaller Tasks



Technologies & tools used in this project

03

Technologies & tools used in this project ?

```
import warnings
warnings.filterwarnings('ignore')

# Importing necessary libraries and functions :
import pandas as pd
import numpy as np
from math import sqrt
import time

# Text processing libraries :
!pip install gensim
import gensim
import re # Regular Expression library
import string
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.stem.porter import PorterStemmer
from gensim.parsing.preprocessing import remove_stopwords
from nltk.tokenize import word_tokenize # Tokenizaion
from spacy.lang.en import English
from spacy.lang.en.stop_words import STOP_WORDS

# Plotting libraries :
import seaborn as sns
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# sklearn :
import sklearn
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import BernoulliNB
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_curve, auc
```

The most important ones...

Python

Seaborn

Scikit-learn

WHAT IS Seaborn?

Seaborn is a **Python data visualization library** based on **matplotlib**. It provides a **high-level interface** for drawing attractive and informative statistical **graphics**.

What is the difference between matplotlib and seaborn?

Seaborn vs matplotlib : **seaborn** utilises **fascinating themes**, while **matplotlib** used for **making basic graphs**. **Seaborn** contains a few plots and patterns for data visualisation, while in matplotlib, datasets are visualised with the assistance of lines, scatter plots, pie charts, histograms, bar-graphs, etc.

WHAT IS Scikit-learn?

Scikit-learn is an indispensable part of the Python **machine learning toolkit** at JPMorgan. It is very widely used across **all parts of the bank for classification**, **predictive analytics**, and very many other **machine learning tasks**.

What is scikit-learn vs TensorFlow?

- The **Tensorflow** is a library for constructing Neural Networks.
- The **scikit-learn** contains ready to use algorithms.
- The **TF** can work with a variety of data types: tabular, text, images, audio.
- The **scikit-learn** is intended to work with tabular data.

Preprocessing the data

04

Preprocessing the data :

The various steps involved in the Machine Learning Pipeline are :

- 1️⃣ `Import Necessary Dependencies`.
- 2️⃣ `Read and Load the Dataset`.
- 3️⃣ `Exploratory Data Analysis`.
- 4️⃣ `Data Visualization of Target Variables`.
- 5️⃣ `Data Preprocessing`.
- 6️⃣ `Data Visualization after Preprocessing`.

Choosing the machine learning models adapted to the dataset

05

Choosing the machine learning models adapted to the dataset

- 7️⃣ Splitting our data into Train and Test Subset .
- 8️⃣ Word Embedding and Transforming Dataset using TF-IDF Vectorizer .
- 9️⃣ Function for Model Evaluation .
- 🔟 Model Building .

Conclusion

06

Results after performing machine learning

After evaluating all models, we can conclude the following details :

Model Id	Model Name	Accuracy	F1-score (class 0)	F1-score (class 1)	AUC Score	Trainig execution time in seconds	Testing execution time in seconds	Nature of dataset used for training	Nature of dataset used for testing
1	Bernoulli Naive Bayes (BNB)	73%	73%	74%	73%	0.44	0.77	full dataset	full dataset
2	Support Vector Machine (SVM)	74%	73%	75%	74%	34.38	0.63	full dataset	full dataset
3	Logistic Regression (LR)	74%	74%	75%	74%	36.13	0.67	full dataset	full dataset
4	K-nearest neighbors (KNN)	60%	NaN	NaN	61%	0.16	NaN	10% of the dataset	10% of the dataset
5	Decision Tree (DT)	69%	69%	69%	69%	31.84	0.76	10% of the dataset	full dataset

- `Execution time` : When it comes to comparing the running time of models, `Bernoulli Naive Bayes` performs faster with a good accuracy score.
- `Accuracy` : When it comes to model accuracy, `logistic regression` & `Suppoort Vector Machine` performs better than most of the other models, with an accuracy of **74%**.
- `F1-score` : The F1 Scores for **class 0** and **class 1** are :

  - For **class 0** (negative tweets) :
    `accuracy : DT (=0.69) < BNB (=0.73) = SVM (=0.73) < LR (=0.74)`
  - For **class 1** (positive tweets) :
    `accuracy : DT (=0.69) < BNB (=0.74) < SVM (=0.73) = LR (=0.75)`

- `AUC Score` :

  `AUC score : KNN (=0.61) < BNB (=0.63) < DT (=0.69) < SVM (=0.74) = LR (=0.74)`

- We therefore conclude that `logistic regression` & `Bernoulli Naive Bayes` & `Suppoort Vector Machine` are the **best model** for the above dataset.

- In our problem statement, `logistic regression` follows `Occam's razor principle` which defines that for a particular problem statement, if the data has no assumptions, then the simplest model works best. Since our **dataset has no assumptions** and **logistic regression is a simple model**, so the concept holds true for the dataset mentioned above.(although it took much longer to run than the fastest model).

THANKS !

Realized by :

ISBAINE Mohamed & **LABRIJI** Saad

Supervised by :

FISSAA Tarik