ML-Task repport

Oracle Labs

**Manar Tayouga(PGX- DS-T1711)**

[Presentation 2](#_Toc26133486)

[1. Feature extraction : 2](#_Toc26133487)

[1.a Importing the libraries : 2](#_Toc26133488)

[1-b importing and exploring the data 2](#_Toc26133489)

[1-c Data Pre-processing 2](#_Toc26133490)

[2. Model training 3](#_Toc26133491)

[3. Performance 3](#_Toc26133492)

[4. Task tools 3](#_Toc26133493)

[5. Conclusion 3](#_Toc26133494)

# **Presentation**

In this task report, we will go through the steps of building a machine learning model that classify the tweets either Politics or Sports.

1. Feature extraction :

## 1.a Importing the libraries :

Our first step will be importing the libraries which we will need for reading the data, processing the tweets and for implementing and training our model:

import pandas as pd

import string

from sklearn.feature\_extraction.text import CountVectorizer

import re, nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

nltk.download('punkt')

nltk.download('stopwords')

from  nltk.stem import SnowballStemmer

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import GaussianNB

## 1-b importing and exploring the data

The data provided is divided in two CSV file (train.csv and test.csv), we import the data using the Pandas library , then we explore it to know how many rows and columns our data contains, and if it contains null values, also to discover the type of our data frame.

## 1-c Data Pre-processing

The data pre-processing is a primary step for building a strong model that can predict labels for our data in the file “test.csv” .

The Data pre-processing steps for the “TweetText “column are as follows:

* **Converting text to lower**
* **Removing links, users names , punctuation marks, additional white spaces and hashtags**
* **Tokenizing the text**
* **Stemming**
* **Removing English stopwords**
* **Using word2vec model, in order to convert words into vectors**

1. Model training

I’ve tried different Machine learning algorithm to train my train dataset such as: **XGBoost Classifier, SVM, Random Forest Classifier, KNN classifier,Logistic Regression, Naïve Bayes**.

After testing them all with different hyper parameter, I opted for the SVM classifier (SVC) algorithm because it gave me the highest score.

1. Performance

In order to build my model, I followed the steps bellow:

* **Importing the libraries needed**
* **Importing train and the test data**
* **Pre-processing the data**
* **Fitting the model**
* **Making prediction of the testing data**

I’ve tried different models, with different hyperparameter but svm.SVC algorithm of sickit learn helps me to get the highest score with the help of word2vec model .

1. Task tools

* Python (using google colab)

1. Conclusion

I think that word embedding techniques will help to make the score higher , therefore I believe that the use of word2vec could help to make a better model, and indeed after applying the word2vec model my score became 92.848%.