Telecom Project

CUSTOMER SATISFACTION

Prepared by:

BEN JEDOU Wejdene
FERCHICHI SARRA
KHCHINI Rawia
ZAOUALI Mayssa
BACCOUCHE Fedi
KHANFIR Yessine
BEN NACEF Med Yassine

Academic Year
2020-2021

TABLE OF CONTENTS

I. Introduction

The telecom operators field nowadays is a highly competitive ground for innovation and creative thinking. Wanting to outshine the next competitor is the number one priority when it comes to dealing with customers. And what is a better way to visualize the customers' content other than a well-put dashboard englobing all the analytical charts and statistics that the client would need direct access to?

Thereby, we present our project, entitled "Customers Satisfaction" that will be the only analytical tool needed to display customers' opinions and ratings in the form of measurable figures and charts per service, per time span and territory.

1. Project Goals

Our primary purpose during this project is the evaluation of the quality of Orange Telecommunication services and its competition in the market , to do so our final output is a business intelligence solution which is basically a dashboard system that displays statistical results related to key performance indicators (KPI)
In order to achieve our plan , we have listed the following goals which are equally important :

- Our BI model needs to be clear and easy to manipulate to our non-technical professional clients .

- In order to support decision making we have to build a coherent and flexible system that can be easily updated from different types of data sources .
- Our dashboard needs to reflect real-time statistics in an efficient way and that is by making the analytics completed as quickly as possible .
- Our solution needs to enhance the decision by working with different sources of data in order to make our dashboard trustworthy and genuine .

2. Project Challenges

In our project, we have encountered many obstacles and they were mainly related to our data manipulation. Some of which are:

- ➔ Lack of technical knowledge regarding some of the tools used.
- ➔ Unstructured data file containing customers' answers.
- ➔ Working with a variety of data (flat files, scraped data, voice messages etc..)
- ➔ Lack of business knowledge concerning Orange Customers' satisfaction.
- ➔ Sentiment analysis on the Tunisian Dialect.

3. IBM Master Plan

As the use of Data Science in the business context is constantly growing, the need of setting a business-oriented strategy is becoming a very important step in order to limit the huge number of existing algorithms and therefore, understand and answer the client's needs.

Why Choose IBM Master Plan?

IBM Master plan methodology is based on a problem-solving logic with the primary objective of facilitating the understanding of the client's needs.

It is necessary in order to limit the number of possibilities and target a one, clear goal. So from problem to approach, it is schemed as follows :

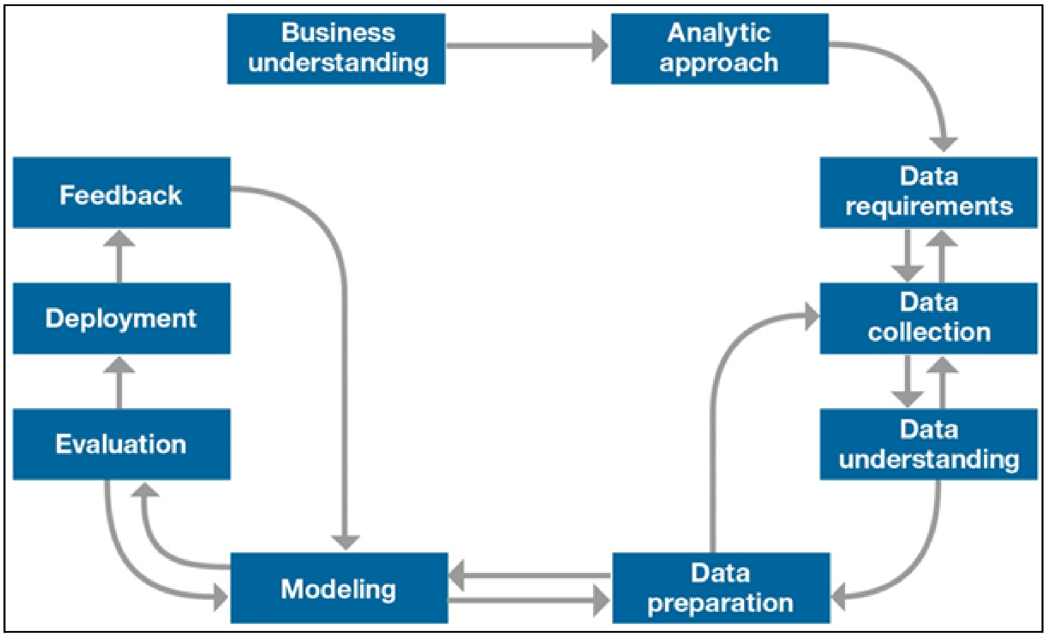


image 1: IBM Master Plan Process

Business understanding:

Business Understanding is placed at the beginning of the methodology because getting clarity allows you to determine which data will be used to answer the core question.

Analytic approach:

The analytic approach phase helps limit the algorithms that will be used later (predictive model, clusters, descriptive model, classification model etc...)

Data requirements:

The necessary data content, formats and sources for initial data collection. In the data collection step,we extract data from various sources and then group them in appropriate databases.

Data collection:

In the initial data collection stage, data scientists identify and gather the available data resources (structured, unstructured, semi-structured) relevant to the problem domain.

Data understanding:

After the original data collection, data scientists typically use descriptive statistics and visualization techniques to understand the data content and assess data quality.

Data preparation:

Data must be prepared using many operations such as addressing missing or invalid values and removing duplicates. This step generally takes almost 90% of the overall project time.

Modeling:

The modeling step includes two types that depend on the business understanding that we dealt with at first: Descriptive and predictive. These models are based on the analytic approach that was taken.

Evaluation:

Evaluating the accuracy of a model is an essential part of the project. It’s the step in which we check if the model we have already generated answers the initial request or not.

Deployment & Feedback:

Finally, once valid, the model will be deployed and a feedback phase will be launched in order to reevaluate it from a customer point of view.

II.BUSINESS AND DATA COMPREHENSION

Working with IBM Master plan, our first step is Business Understanding which is in other words setting clear measurable results we will be working on in order to keep our focus sharp and produce the right answer for the right question.

1. Business Objectives

The business objectives that our project will be built upon is the answer to our client's needs and preferences. Our objective is to present an online dashboard composed with a series of advanced analytical tools such as charts with filter options, a map that displays the satisfaction percentage per region to keep track of the ebb and flow of customers content as well as a time span calendar through which the client will have a clear view on his monthly growth facilitating in the process the task of strategic decision-making in critical times. The dashboard will also be equipped with an innovative yet important flow of data deriving from the social media (Facebook, Twitter, Linkedin, Web Forms, etc..) which will later go through our emotional intelligent models to predict the general level of content throughout the reviews all the while being able to keep an eye on the company's competitors services' reviews via a specific chart revolving around the second competitor in the Telecom Operators' domain. All the previous analytical features of the dashboard will be apt to be updated periodically depending on the client's preferences giving him an up-to-date visualization of the company's overall progress.

2. Data Science Objectives

The primary focus of data science is mainly divided into the following areas : Data extraction, pre-processing, analysing and inferring information to draw insights and conclusions .
As mentioned above in our business objectives, in order to be able to assist Telecommunication companies in making efficient decisions according to their position in the market in terms of customer satisfaction , we had to set our data science objectives towards Natural Language Processing field which is strongly related to the nature of the customer satisfaction business domain .
So in this project , we have defined our data science objectives below :

- Data source understanding
- External data collection/scrapping
- Data cleaning/preprocessing
- Creation of models using Deep Learning /Transfer Learning

3. Business Intelligence Objectives

Our project's final output is the implementation of a BI solution to transform our internal and external data into actionable insights that boosts the strategic and tactical ability of our end-user.
To enhance the efficiency of our final output , we have used data science as mentioned above to labelize our external data using Sentiment Analysis Techniques ,and as a result we were able to add multi-source data to our BI tools .
The goal of Business Intelligence is to analyze data sets and present analytical findings in reports, summaries, dashboards, graphs, charts and maps to provide users with detailed intelligence about the state of the business.
And our business intelligence objective are defined below :

- Data source understanding
- Data cleaning/transforming/loading
- Dashboard creating
- Deployment of BI solution

Business and Data comprehension is an essential step for a solid project, so before moving any further we had to build a strong foundation for our project by mentioning our business , data science and BI objectives . And in the next chapter , we will start the data preparation phase .

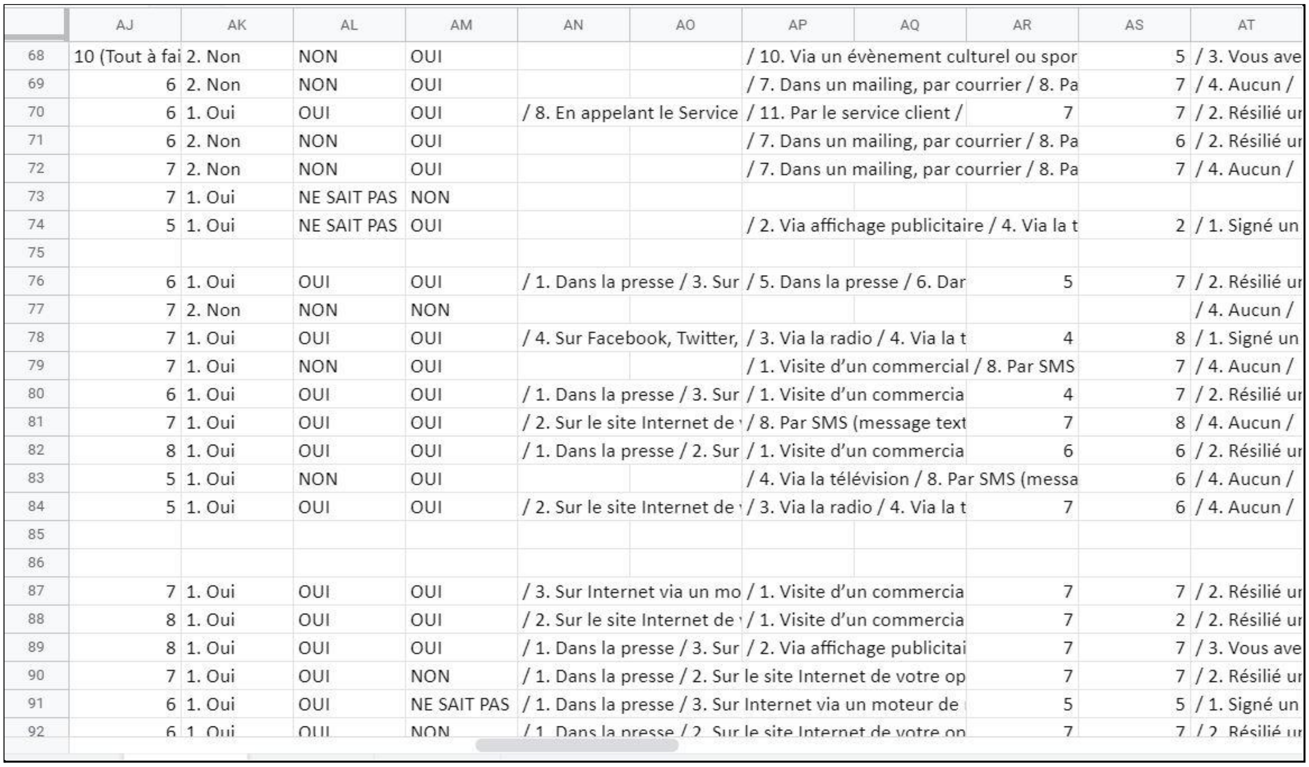
III. DATA PREPARATION

1. Internal Data

Our data source consist of a collection of survey answers:

Document Type	Size	Queries	Customers
Microsoft Excel	801K	168	1688

We encountered multiple obstacles dealing with the data file due to a malfunction of the Excel document and we had to go through a series of data manipulation in order to fix the unaligned data. Image 2 shows the raw data before any ultering.



	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT
68	10 (Tout à fai	2. Non	NON	OUI			/ 10. Via un évènement culturel ou spor			5	/ 3. Vous ave
69	6	2. Non	NON	OUI			/ 7. Dans un mailing, par courrier / 8. Pa			7	/ 4. Aucun /
70	6	1. Oui	OUI	OUI	/ 8. En appelant le Service		/ 11. Par le service client /		7	7	/ 2. Résilié ur
71	6	2. Non	NON	OUI			/ 7. Dans un mailing, par courrier / 8. Pa			6	/ 2. Résilié ur
72	7	2. Non	NON	OUI			/ 7. Dans un mailing, par courrier / 8. Pa			7	/ 4. Aucun /
73	7	1. Oui	NE SAIT PAS	NON							
74	5	1. Oui	NE SAIT PAS	OUI			/ 2. Via affichage publicitaire / 4. Via la t			2	/ 1. Signé un
75											
76	6	1. Oui	OUI	OUI	/ 1. Dans la presse / 3. Sur		/ 5. Dans la presse / 6. Dar		5	7	/ 2. Résilié ur
77	7	2. Non	NON	NON							/ 4. Aucun /
78	7	1. Oui	OUI	OUI	/ 4. Sur Facebook, Twitter,		/ 3. Via la radio / 4. Via la t		4	8	/ 1. Signé un
79	7	1. Oui	NON	OUI			/ 1. Visite d’un commercial / 8. Par SMS			7	/ 4. Aucun /
80	6	1. Oui	OUI	OUI	/ 1. Dans la presse / 3. Sur		/ 1. Visite d’un commercia		4	7	/ 2. Résilié ur
81	7	1. Oui	OUI	OUI	/ 2. Sur le site Internet de		/ 8. Par SMS (message text		7	8	/ 4. Aucun /
82	8	1. Oui	OUI	OUI	/ 1. Dans la presse / 2. Sur		/ 1. Visite d’un commercia		6	6	/ 2. Résilié ur
83	5	1. Oui	NON	OUI			/ 4. Via la télévision / 8. Par SMS (messa			6	/ 4. Aucun /
84	5	1. Oui	OUI	OUI	/ 2. Sur le site Internet de		/ 3. Via la radio / 4. Via la t		7	6	/ 4. Aucun /
85											
86											
87	7	1. Oui	OUI	OUI	/ 3. Sur Internet via un mo		/ 1. Visite d’un commercia		7	7	/ 2. Résilié ur
88	8	1. Oui	OUI	OUI	/ 2. Sur le site Internet de		/ 1. Visite d’un commercia		7	2	/ 2. Résilié ur
89	8	1. Oui	OUI	OUI	/ 1. Dans la presse / 3. Sur		/ 2. Via affichage publicitai		7	7	/ 3. Vous ave
90	7	1. Oui	OUI	NON	/ 1. Dans la presse / 2. Sur le site Internet de votre op				7	7	/ 2. Résilié ur
91	6	1. Oui	OUI	NE SAIT PAS	/ 1. Dans la presse / 3. Sur Internet via un moteur de				5	5	/ 1. Signé un
92	6	1. Oui	OUI	NON	/ 1. Dans la presse / 2. Sur le site Internet de votre op				7	7	/ 2. Résilié ur

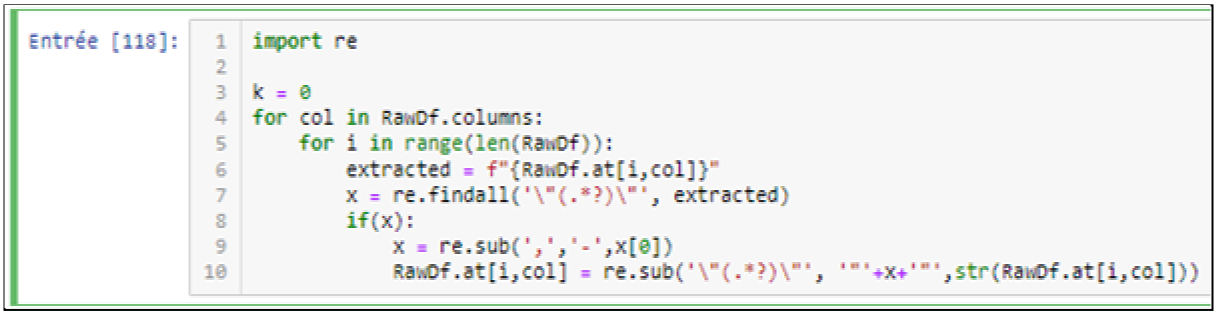
image 2 : Excel File Raw Data

We resorted to using two primary tools to accomplish our desired output which are Jupyter Notebook and Notepad++.
We will explain below the process we went through in order to rearrange our data file so that it can be convenient for further procedures.

1.1 Eliminating the wrong separators [“,” and “.,”]:

First of all, we have created more than one algorithm seeing that the shift of the Excel columns could not be resolved with a single regular expression.

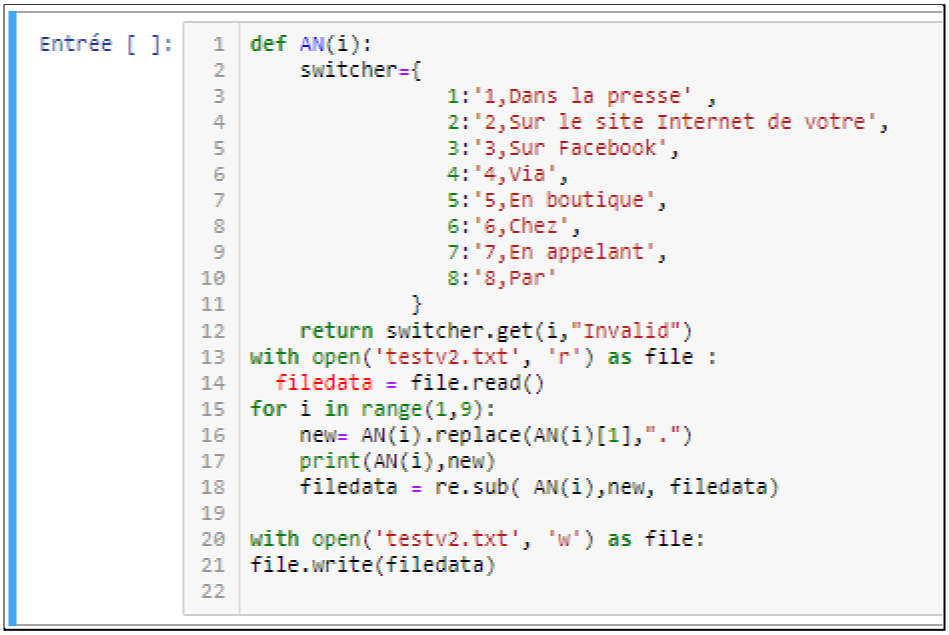
The screenshot below shows a snippet of the code on Jupyter Notebook:



Entrée [118]:

```python
import re

k = 0
for col in RawDf.columns:
    for i in range(len(RawDf)):
        extracted = f"{RawDf.at[i,col]}"
        x = re.findall('\"(.*?)\"', extracted)
        if(x):
            x = re.sub(',','-',x[0])
            RawDf.at[i,col] = re.sub('\"(.*?)\"', '"'+x+'"',str(RawDf.at[i,col]))
```

image 3 : Python regular expression code 1

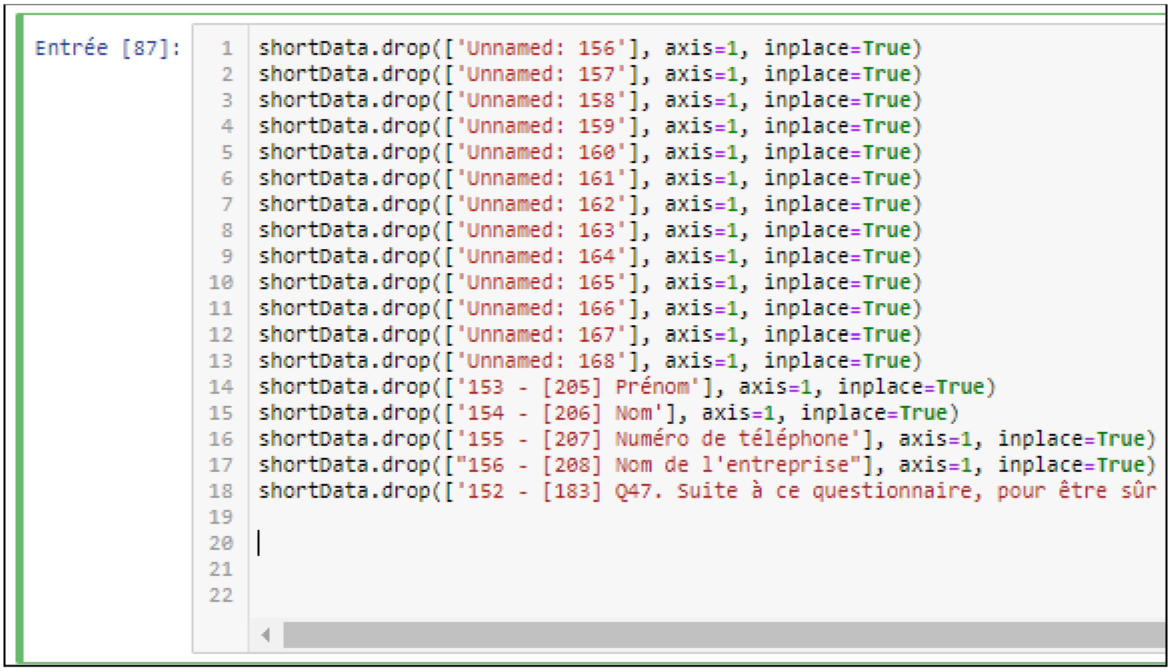


```
Entrée [ ]:
def AN(i):
    switcher={
                1:'1,Dans la presse' ,
                2:'2,Sur le site Internet de votre',
                3:'3,Sur Facebook',
                4:'4,Via',
                5:'5,En boutique',
                6:'6,Chez',
                7:'7,En appelant',
                8:'8,Par'
            }
    return switcher.get(i,"Invalid")
with open('testv2.txt', 'r') as file :
  filedata = file.read()
for i in range(1,9):
    new= AN(i).replace(AN(i)[1],".")
    print(AN(i),new)
    filedata = re.sub( AN(i),new, filedata)

with open('testv2.txt', 'w') as file:
file.write(filedata)
```

image 4 : Python regular expression code 2

1.2 Eliminating the unnecessary columns:

After rearranging the data in its right order, we noticed that surface of multiple futile empty columns. In addition, we concluded that the following list of columns have no added value to our following procedures. So we have resorted to eliminate them in the preparation stage.

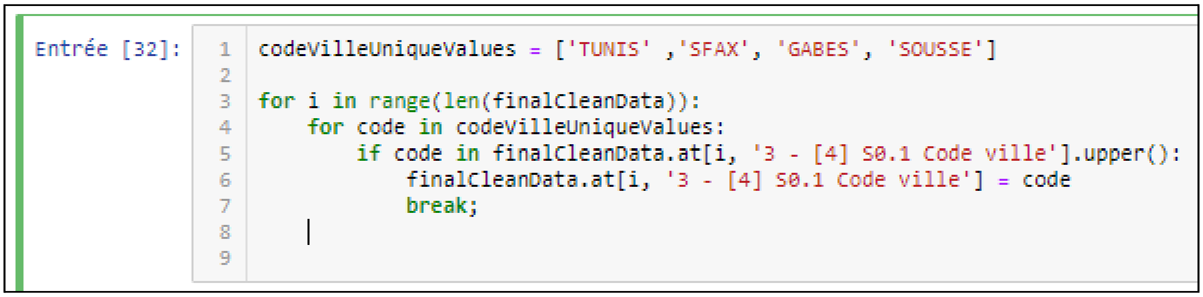


```
Entrée [87]:
shortData.drop(['Unnamed: 156'], axis=1, inplace=True)
shortData.drop(['Unnamed: 157'], axis=1, inplace=True)
shortData.drop(['Unnamed: 158'], axis=1, inplace=True)
shortData.drop(['Unnamed: 159'], axis=1, inplace=True)
shortData.drop(['Unnamed: 160'], axis=1, inplace=True)
shortData.drop(['Unnamed: 161'], axis=1, inplace=True)
shortData.drop(['Unnamed: 162'], axis=1, inplace=True)
shortData.drop(['Unnamed: 163'], axis=1, inplace=True)
shortData.drop(['Unnamed: 164'], axis=1, inplace=True)
shortData.drop(['Unnamed: 165'], axis=1, inplace=True)
shortData.drop(['Unnamed: 166'], axis=1, inplace=True)
shortData.drop(['Unnamed: 167'], axis=1, inplace=True)
shortData.drop(['Unnamed: 168'], axis=1, inplace=True)
shortData.drop(['153 - [205] Prénom'], axis=1, inplace=True)
shortData.drop(['154 - [206] Nom'], axis=1, inplace=True)
shortData.drop(['155 - [207] Numéro de téléphone'], axis=1, inplace=True)
shortData.drop(["156 - [208] Nom de l'entreprise"], axis=1, inplace=True)
shortData.drop(['152 - [183] Q47. Suite à ce questionnaire, pour être sûr
```

image 5 : elimination of unnecessary columns

1.3 Data Type Consistency rectification:

Here we noticed that the same entries for “Code Ville” are typed in different ways, and the data type for the “Date de passation du questionnaire” column are not unified (String and DateTime) .
So we unified the data type for “Date de passation du questionnaire” using “NotePad ++” and we used a small “Python” code on “Code Ville” as shown below.



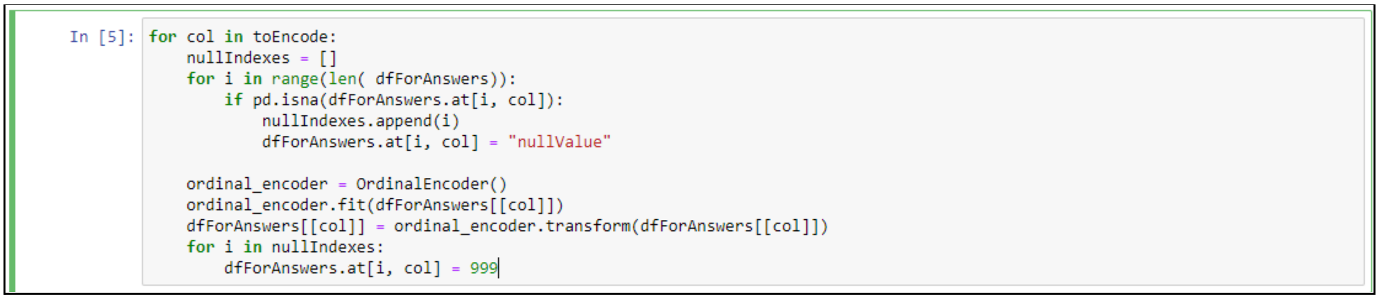
```
Entrée [32]:
codeVilleUniqueValues = ['TUNIS' ,'SFAX', 'GABES', 'SOUSSE']

for i in range(len(finalCleanData)):
    for code in codeVilleUniqueValues:
        if code in finalCleanData.at[i, '3 - [4] S0.1 Code ville'].upper():
            finalCleanData.at[i, '3 - [4] S0.1 Code ville'] = code
            break;
```

image 6 : unifying “Code Ville” Data entries

1.4 Data Encoding:

- **Ordinal Encoding**

In short, we stored all the columns on which we want to apply the “OrdinalEcoder” in an array names “toEncode” which we will loop through and encode each element of it.

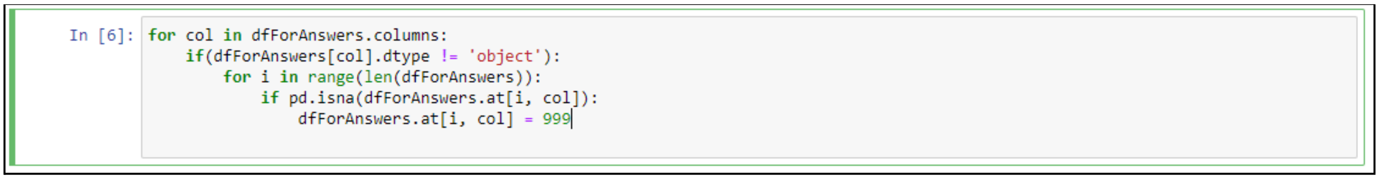


```
In [5]: for col in toEncode:
            nullIndexes = []
            for i in range(len( dfForAnswers)):
                if pd.isna(dfForAnswers.at[i, col]):
                    nullIndexes.append(i)
                    dfForAnswers.at[i, col] = "nullValue"

            ordinal_encoder = OrdinalEncoder()
            ordinal_encoder.fit(dfForAnswers[[col]])
            dfForAnswers[[col]] = ordinal_encoder.transform(dfForAnswers[[col]])
            for i in nullIndexes:
                dfForAnswers.at[i, col] = 999
```

image 7 : Ordinal Encoder

- **Customized Encoding For the NaN value**

It is necessary to eliminate all null values before supplying the data warehouse, so we used this piece of code to replace nulls with “999” .

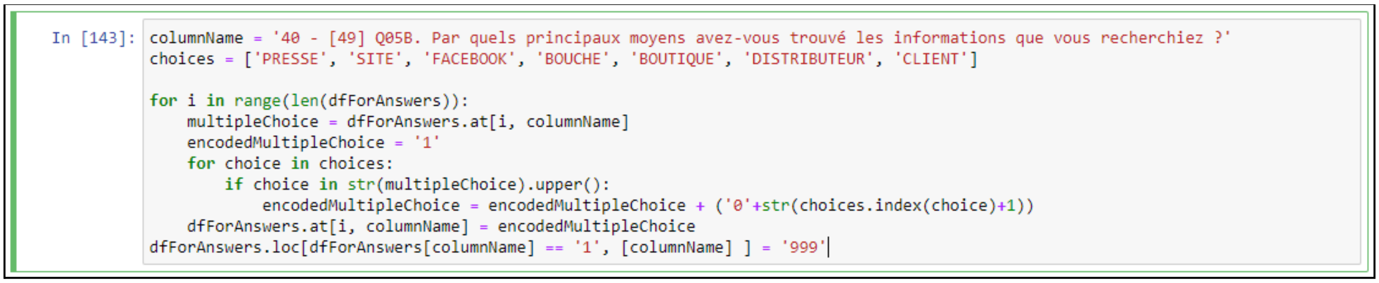


```
In [6]: for col in dfForAnswers.columns:
            if(dfForAnswers[col].dtype != 'object'):
                for i in range(len(dfForAnswers)):
                    if pd.isna(dfForAnswers.at[i, col]):
                        dfForAnswers.at[i, col] = 999
```

image 8 : Encoding the Nan Values

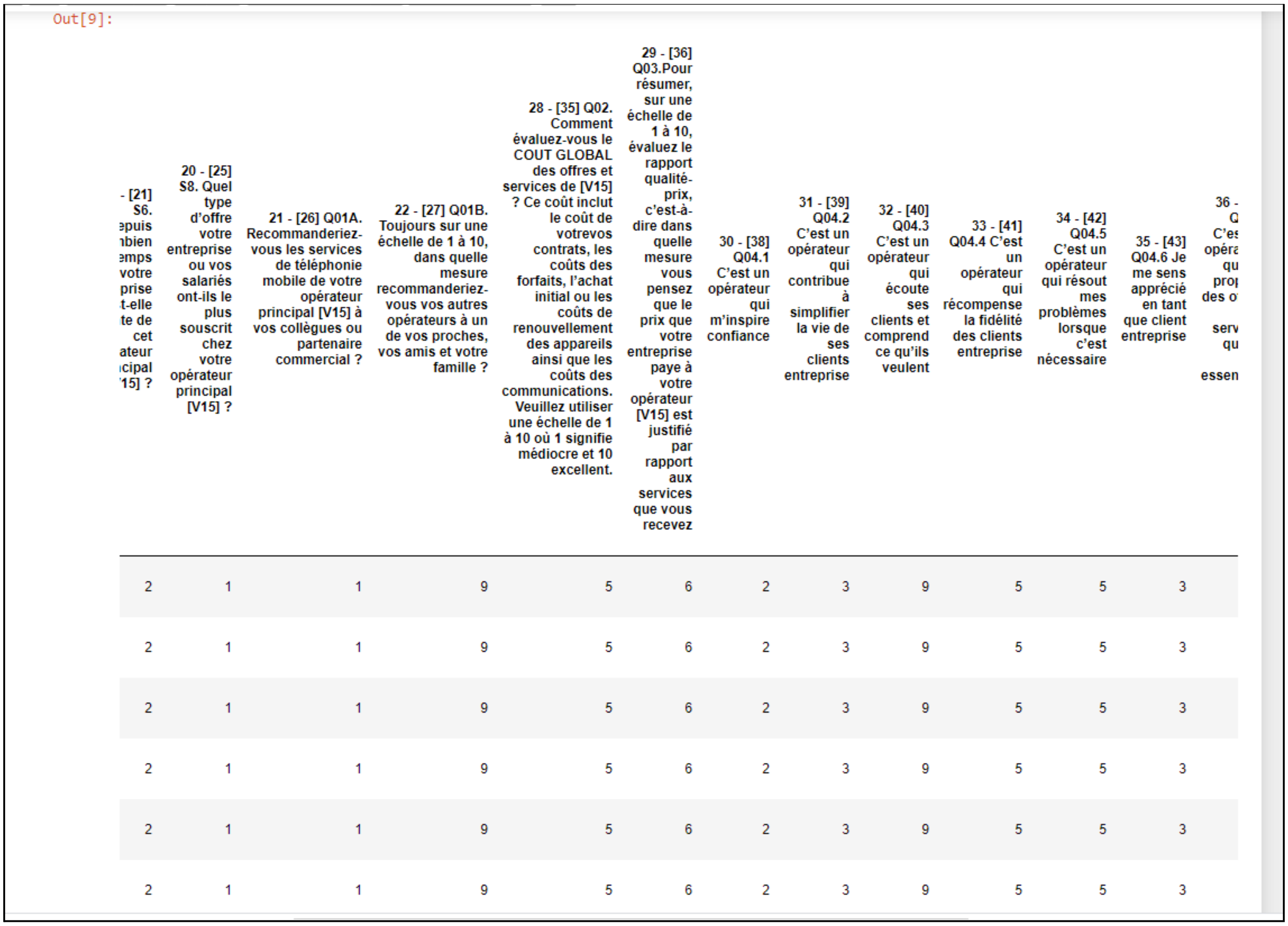
- **Customized Encoding (Multiple choices)**

At this stage, and as there is no optimal pre-defined encoding algorithm that deals with multiple choice columns, we have elaborated our own logic to transform them.
The encoded result of each cell is a numerical value that reflects its meaning, to be more accurate, it is the concatenation of choices' indexes separated by 0s.



```
In [143]: columnName = '40 - [49] Q05B. Par quels principaux moyens avez-vous trouvé les informations que vous recherchiez ?'
          choices = ['PRESSE', 'SITE', 'FACEBOOK', 'BOUCHE', 'BOUTIQUE', 'DISTRIBUTEUR', 'CLIENT']

          for i in range(len(dfForAnswers)):
              multipleChoice = dfForAnswers.at[i, columnName]
              encodedMultipleChoice = '1'
              for choice in choices:
                  if choice in str(multipleChoice).upper():
                      encodedMultipleChoice = encodedMultipleChoice + ('0'+str(choices.index(choice)+1))
              dfForAnswers.at[i, columnName] = encodedMultipleChoice
          dfForAnswers.loc[dfForAnswers[columnName] == '1', [columnName] ] = '999'
```

image 9 : Encoding the multiple choices



```
Out[9]:
```

- [21] S6. epuis nbien emps votre prise t-elle te de cet ateur icipal '15] ?	20 - [25] S8. Quel type d'offre votre entreprise ou vos salariés ont-ils le plus souscrit chez votre opérateur principal [V15] ?	21 - [26] Q01A. Recommanderiez-vous les services de téléphonie mobile de votre opérateur principal [V15] à vos collègues ou partenaire commercial ?	22 - [27] Q01B. Toujours sur une échelle de 1 à 10, dans quelle mesure recommanderiez-vous vos autres opérateurs à un de vos proches, vos amis et votre famille ?	28 - [35] Q02. Comment évaluez-vous le COUT GLOBAL des offres et services de [V15] ? Ce coût inclut le coût de votrevos contrats, les coûts des forfaits, l'achat initial ou les coûts de renouvellement des appareils ainsi que les coûts des communications. Veuillez utiliser une échelle de 1 à 10 où 1 signifie médiocre et 10 excellent.	29 - [36] Q03.Pour résumer, sur une échelle de 1 à 10, évaluez le rapport qualité-prix, c'est-à-dire dans quelle mesure vous pensez que le prix que votre entreprise paye à votre opérateur [V15] est justifié par rapport aux services que vous recevez	30 - [38] Q04.1 C'est un opérateur qui m'inspire confiance	31 - [39] Q04.2 C'est un opérateur qui contribue à simplifier la vie de ses clients entreprise	32 - [40] Q04.3 C'est un opérateur qui écoute ses clients et comprend ce qu'ils veulent	33 - [41] Q04.4 C'est un opérateur qui récompense la fidélité des clients entreprise	34 - [42] Q04.5 C'est un opérateur qui résout mes problèmes lorsque c'est nécessaire	35 - [43] Q04.6 Je me sens apprécié en tant que client entreprise	36 - Q C'es opéra qu prop des o serv qu essen
2	1	1	9	5	6	2	3	9	5	5	3	
2	1	1	9	5	6	2	3	9	5	5	3	
2	1	1	9	5	6	2	3	9	5	5	3	
2	1	1	9	5	6	2	3	9	5	5	3	
2	1	1	9	5	6	2	3	9	5	5	3	
2	1	1	9	5	6	2	3	9	5	5	3	

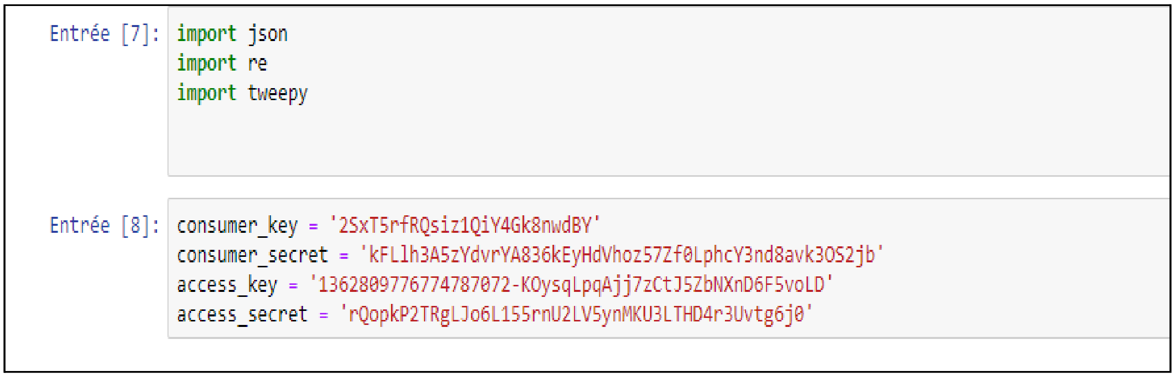
image 10 : Final data output

Our final output in image 10 is the last step in the preparation process of the internal data file. Now we have all the answers stored in numeric format and we will be able to exploit it on the next stage.

2. External Data Collecting

In this part , we extracted data by scrapping different sites like Twitter , LinkedIn and Orange forum site and we sent a survey by mail to collect information related to the customer satisfaction context .

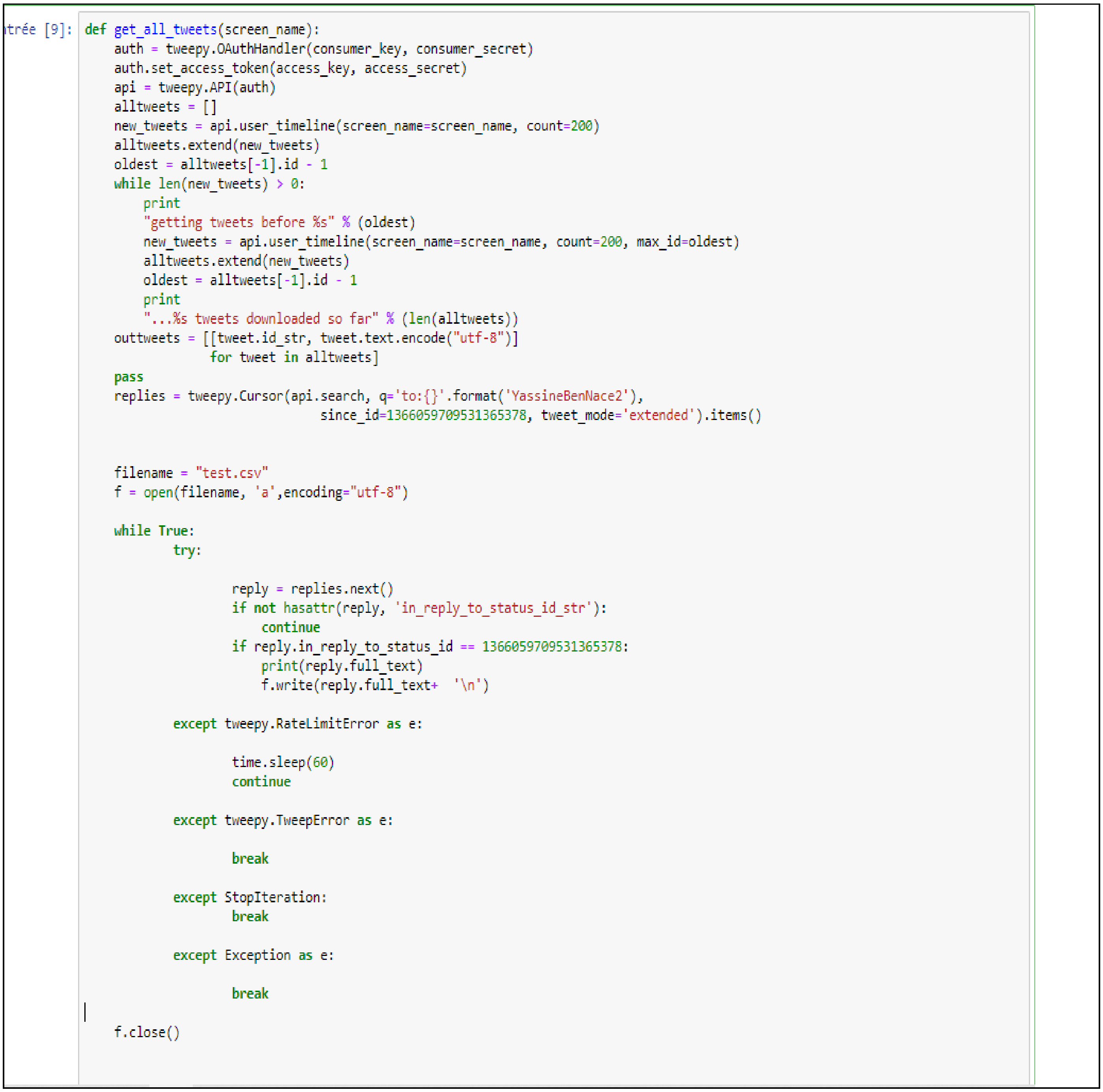
2.1 Collecting data from twitter



```
Entrée [7]: import json
            import re
            import tweepy

Entrée [8]: consumer_key = '2SxT5rfRQsiz1QiY4Gk8nwdBY'
            consumer_secret = 'kFLlh3A5zYdvrYA836kEyHdVhoz57Zf0LphcY3nd8avk3OS2jb'
            access_key = '1362809776774787072-KOysqLpqAjj7zCtJ5ZbNXnD6F5voLD'
            access_secret = 'rQopkP2TRgLJo6L155rnU2LV5ynMKU3LTHD4r3Uvtg6j0'
```

image 11 : Twitter credentials

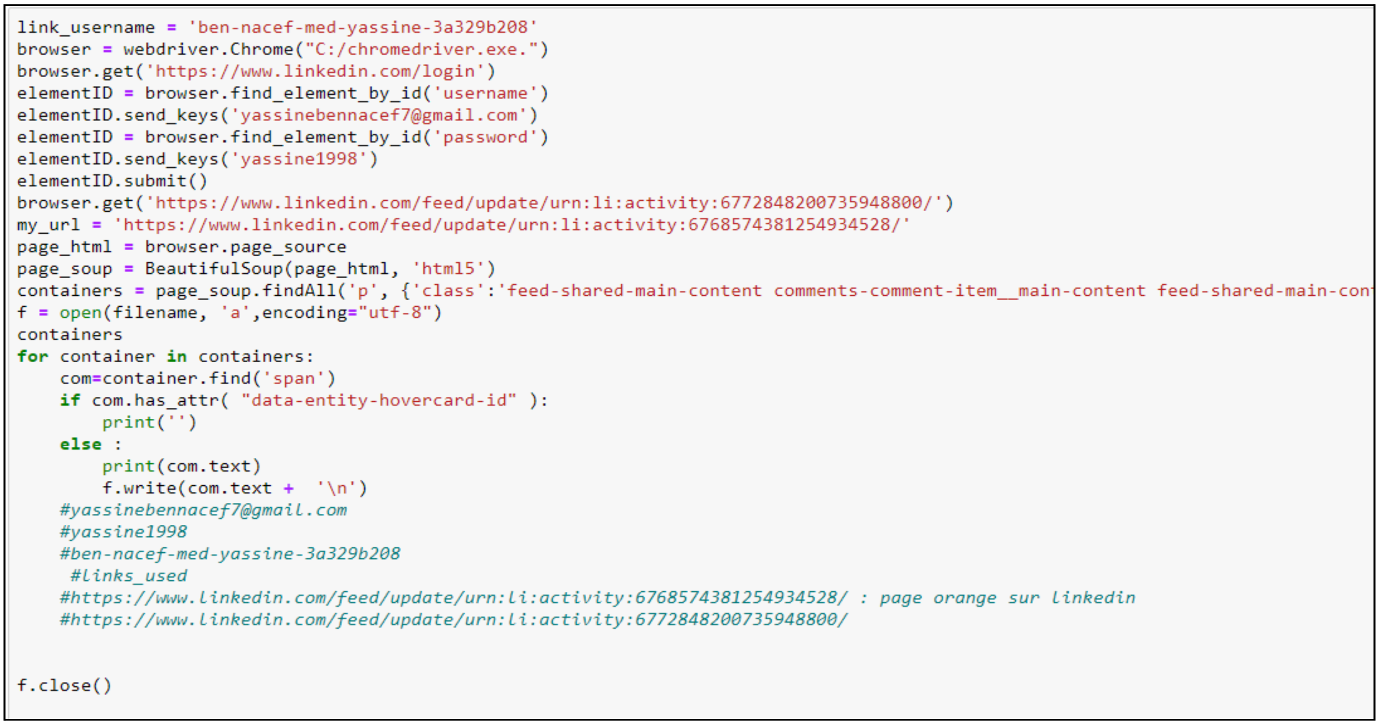
From Twitter we scrapped retweets related to a post about Orange customer satisfaction and we stocked the comments in csv file called test that will be used later as an input source.



```
def get_all_tweets(screen_name):
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_key, access_secret)
    api = tweepy.API(auth)
    alltweets = []
    new_tweets = api.user_timeline(screen_name=screen_name, count=200)
    alltweets.extend(new_tweets)
    oldest = alltweets[-1].id - 1
    while len(new_tweets) > 0:
        print
        "getting tweets before %s" % (oldest)
        new_tweets = api.user_timeline(screen_name=screen_name, count=200, max_id=oldest)
        alltweets.extend(new_tweets)
        oldest = alltweets[-1].id - 1
        print
        "...%s tweets downloaded so far" % (len(alltweets))
    outtweets = [[tweet.id_str, tweet.text.encode("utf-8")]
                 for tweet in alltweets]
    pass
    replies = tweepy.Cursor(api.search, q='to:{}'.format('YassineBenNace2'),
                                since_id=1366059709531365378, tweet_mode='extended').items()


    filename = "test.csv"
    f = open(filename, 'a',encoding="utf-8")

    while True:
            try:

                    reply = replies.next()
                    if not hasattr(reply, 'in_reply_to_status_id_str'):
                        continue
                    if reply.in_reply_to_status_id == 1366059709531365378:
                        print(reply.full_text)
                        f.write(reply.full_text+  '\n')

            except tweepy.RateLimitError as e:

                    time.sleep(60)
                    continue

            except tweepy.TweepError as e:

                    break

            except StopIteration:
                    break

            except Exception as e:

                    break

    f.close()
```

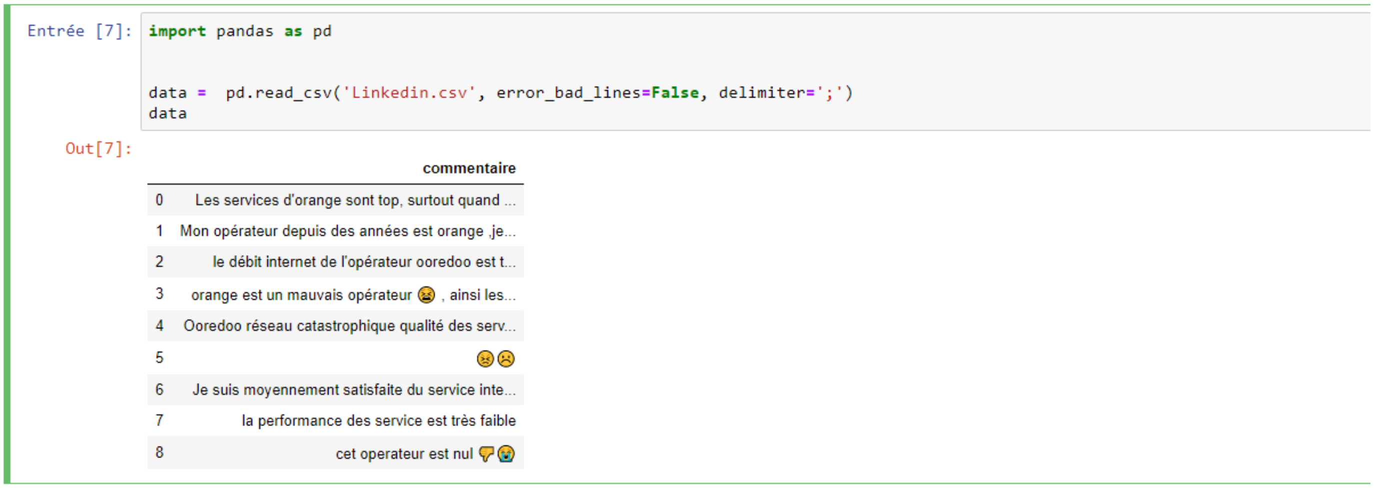
image 12 : Using Twitter API tweepy for scrapping

2.2 Collecting data from LinkedIn



```
link_username = 'ben-nacef-med-yassine-3a329b208'
browser = webdriver.Chrome("C:/chromedriver.exe.")
browser.get('https://www.linkedin.com/login')
elementID = browser.find_element_by_id('username')
elementID.send_keys('yassinebennacef7@gmail.com')
elementID = browser.find_element_by_id('password')
elementID.send_keys('yassine1998')
elementID.submit()
browser.get('https://www.linkedin.com/feed/update/urn:li:activity:6772848200735948800/')
my_url = 'https://www.linkedin.com/feed/update/urn:li:activity:6768574381254934528/'
page_html = browser.page_source
page_soup = BeautifulSoup(page_html, 'html5')
containers = page_soup.findAll('p', {'class':'feed-shared-main-content comments-comment-item__main-content feed-shared-main-con
f = open(filename, 'a',encoding="utf-8")
containers
for container in containers:
    com=container.find('span')
    if com.has_attr( "data-entity-hovercard-id" ):
        print('')
    else :
        print(com.text)
        f.write(com.text +  '\n')
    #yassinebennacef7@gmail.com
    #yassine1998
    #ben-nacef-med-yassine-3a329b208
     #links_used
    #https://www.linkedin.com/feed/update/urn:li:activity:6768574381254934528/ : page orange sur linkedin
    #https://www.linkedin.com/feed/update/urn:li:activity:6772848200735948800/


f.close()
```

image 13 : Scraping LinkedIn posts using Chromedriver and BeautifulSoup 1



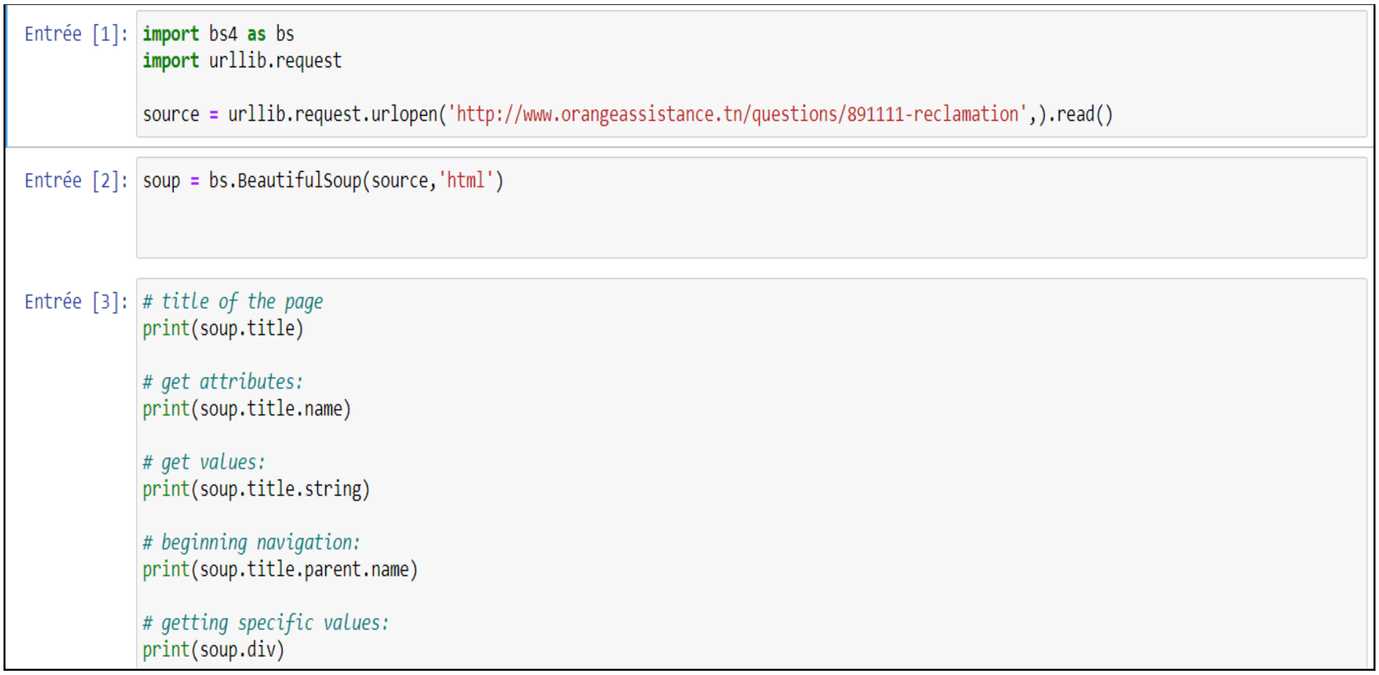
```
Entrée [7]: import pandas as pd


            data =  pd.read_csv('Linkedin.csv', error_bad_lines=False, delimiter=';')
            data
```

Out[7]:

	commentaire
0	Les services d'orange sont top, surtout quand ...
1	Mon opérateur depuis des années est orange ,je...
2	le débit internet de l'opérateur ooredoo est t...
3	orange est un mauvais opérateur 😫 , ainsi les...
4	Ooredoo réseau catastrophique qualité des serv...
5	😣☹
6	Je suis moyennement satisfaite du service inte...
7	la performance des service est très faible
8	cet operateur est nul 👎😭

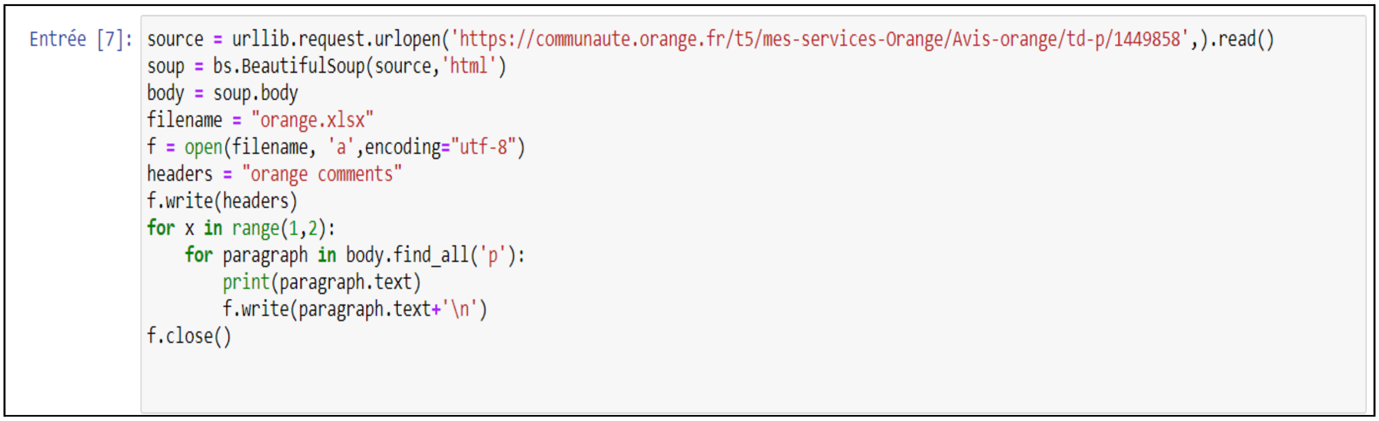
image 14 : Scraping LinkedIn posts using Chromedriver and BeautifulSoup 2

2.3 Collecting data from Orange community forum



```
Entrée [1]: import bs4 as bs
            import urllib.request

            source = urllib.request.urlopen('http://www.orangeassistance.tn/questions/891111-reclamation',).read()

Entrée [2]: soup = bs.BeautifulSoup(source,'html')

Entrée [3]: # title of the page
            print(soup.title)

            # get attributes:
            print(soup.title.name)

            # get values:
            print(soup.title.string)

            # beginning navigation:
            print(soup.title.parent.name)

            # getting specific values:
            print(soup.div)
```

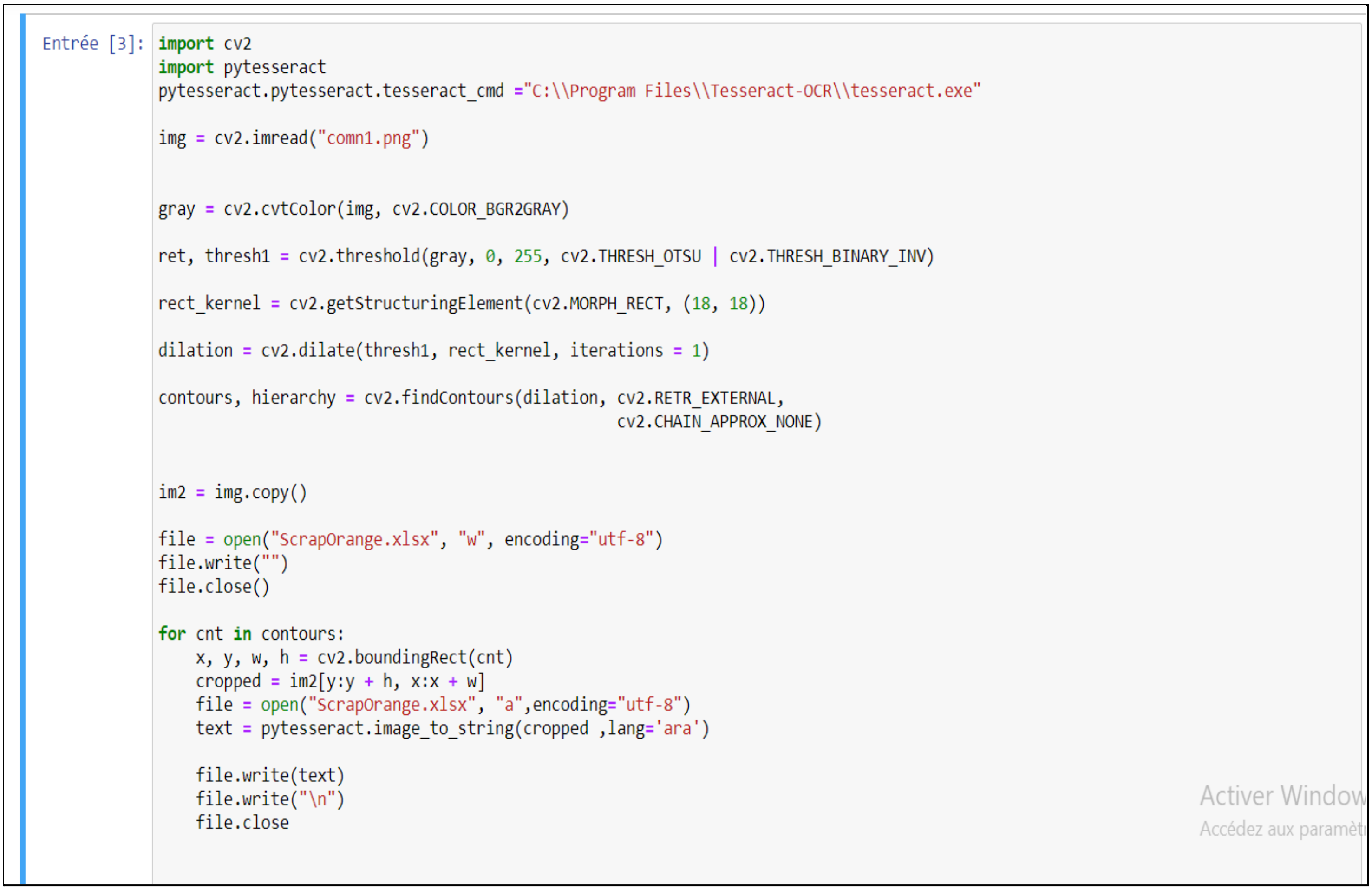
image 15 : Scraping LinkedIn posts using Chromedriver and BeautifulSoup 1



```
Entrée [7]: source = urllib.request.urlopen('https://communaute.orange.fr/t5/mes-services-Orange/Avis-orange/td-p/1449858',).read()
            soup = bs.BeautifulSoup(source,'html')
            body = soup.body
            filename = "orange.xlsx"
            f = open(filename, 'a',encoding="utf-8")
            headers = "orange comments"
            f.write(headers)
            for x in range(1,2):
                for paragraph in body.find_all('p'):
                    print(paragraph.text)
                    f.write(paragraph.text+'\n')
            f.close()
```

image 16: Using BeautifulSoup library for comment scrapping

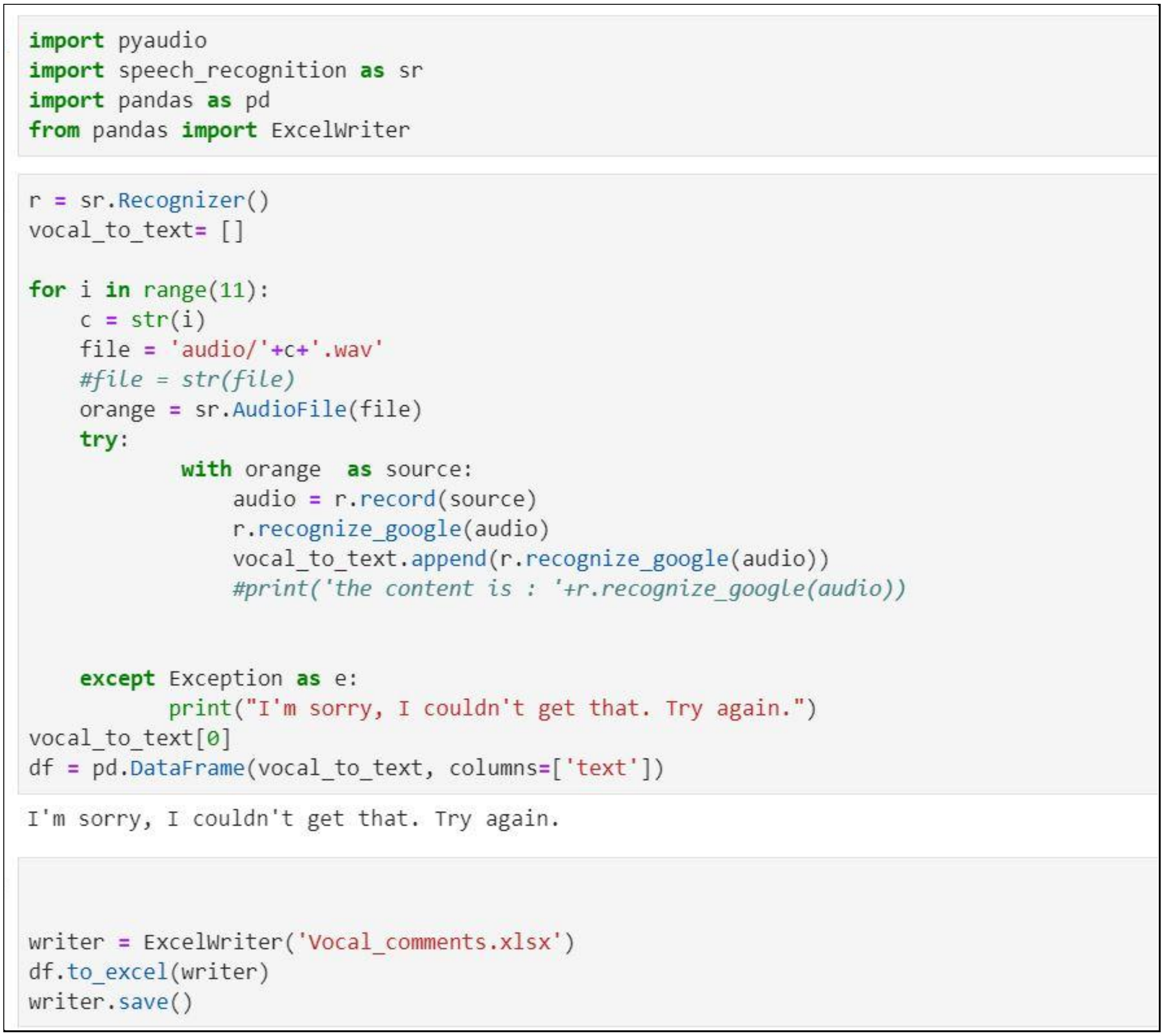
2.4 Collecting data from facebook using Optical Character Recognition (OCR)



```
import cv2
import pytesseract
pytesseract.pytesseract.tesseract_cmd ="C:\\Program Files\\Tesseract-OCR\\tesseract.exe"

img = cv2.imread("comn1.png")


gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

ret, thresh1 = cv2.threshold(gray, 0, 255, cv2.THRESH_OTSU | cv2.THRESH_BINARY_INV)

rect_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (18, 18))

dilation = cv2.dilate(thresh1, rect_kernel, iterations = 1)

contours, hierarchy = cv2.findContours(dilation, cv2.RETR_EXTERNAL,
                                                 cv2.CHAIN_APPROX_NONE)


im2 = img.copy()

file = open("ScrapOrange.xlsx", "w", encoding="utf-8")
file.write("")
file.close()

for cnt in contours:
    x, y, w, h = cv2.boundingRect(cnt)
    cropped = im2[y:y + h, x:x + w]
    file = open("ScrapOrange.xlsx", "a",encoding="utf-8")
    text = pytesseract.image_to_string(cropped ,lang='ara')

    file.write(text)
    file.write("\n")
    file.close
```

image 17 :Using Pytesseract to detect text in a facebook post

2.5 Collecting audio data using speech recognition



```
import pyaudio
import speech_recognition as sr
import pandas as pd
from pandas import ExcelWriter
```

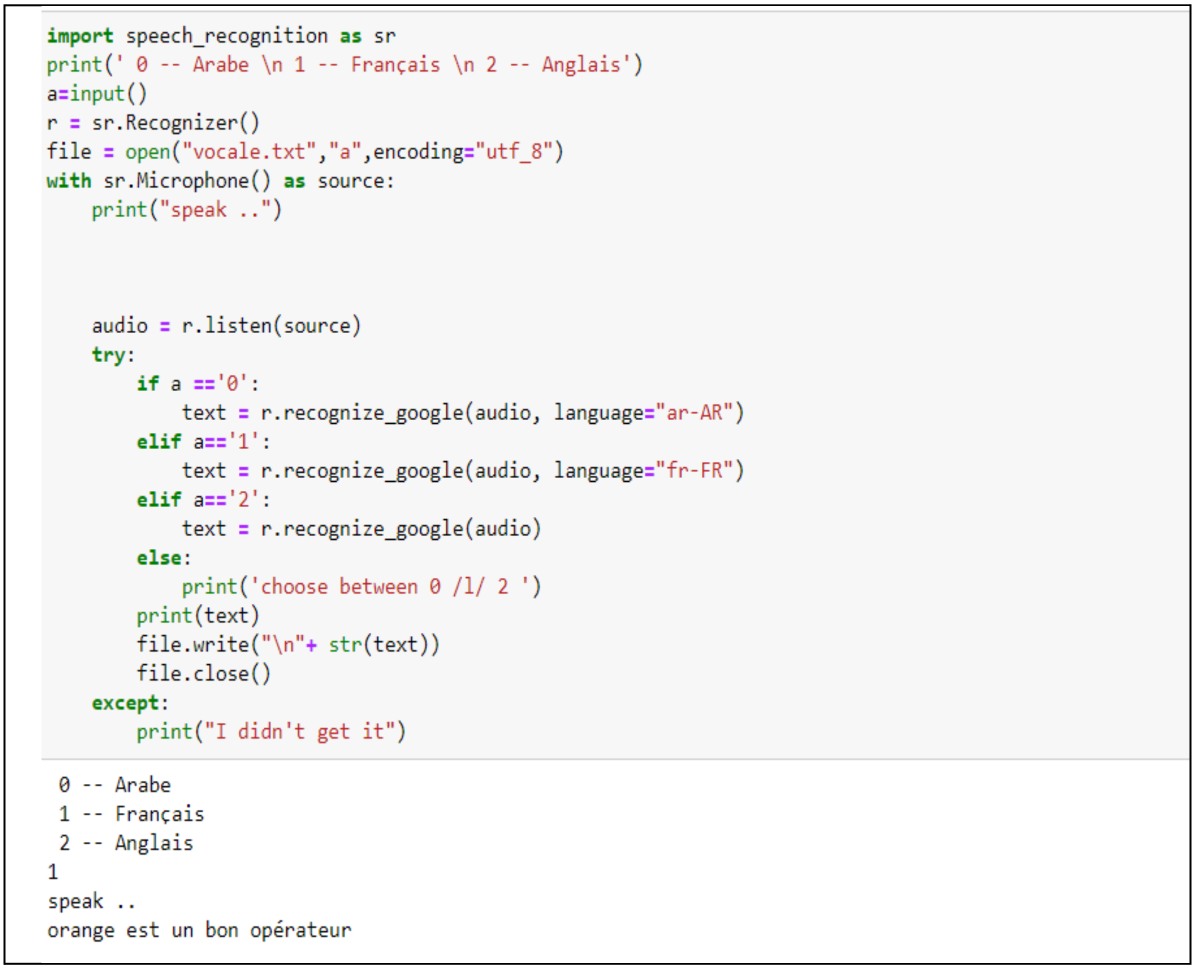
```
r = sr.Recognizer()
vocal_to_text= []

for i in range(11):
    c = str(i)
    file = 'audio/'+c+'.wav'
    #file = str(file)
    orange = sr.AudioFile(file)
    try:
            with orange  as source:
                audio = r.record(source)
                r.recognize_google(audio)
                vocal_to_text.append(r.recognize_google(audio))
                #print('the content is : '+r.recognize_google(audio))


    except Exception as e:
           print("I'm sorry, I couldn't get that. Try again.")
vocal_to_text[0]
df = pd.DataFrame(vocal_to_text, columns=['text'])
```

```
I'm sorry, I couldn't get that. Try again.
```

```


writer = ExcelWriter('Vocal_comments.xlsx')
df.to_excel(writer)
writer.save()
```

image 18 : Reading and processing audio files

As mentioned above , we collected vocal data recorded on an audio file and we used speech recognition to transform our audio input into text which is exported to an Excel file containing different comments .



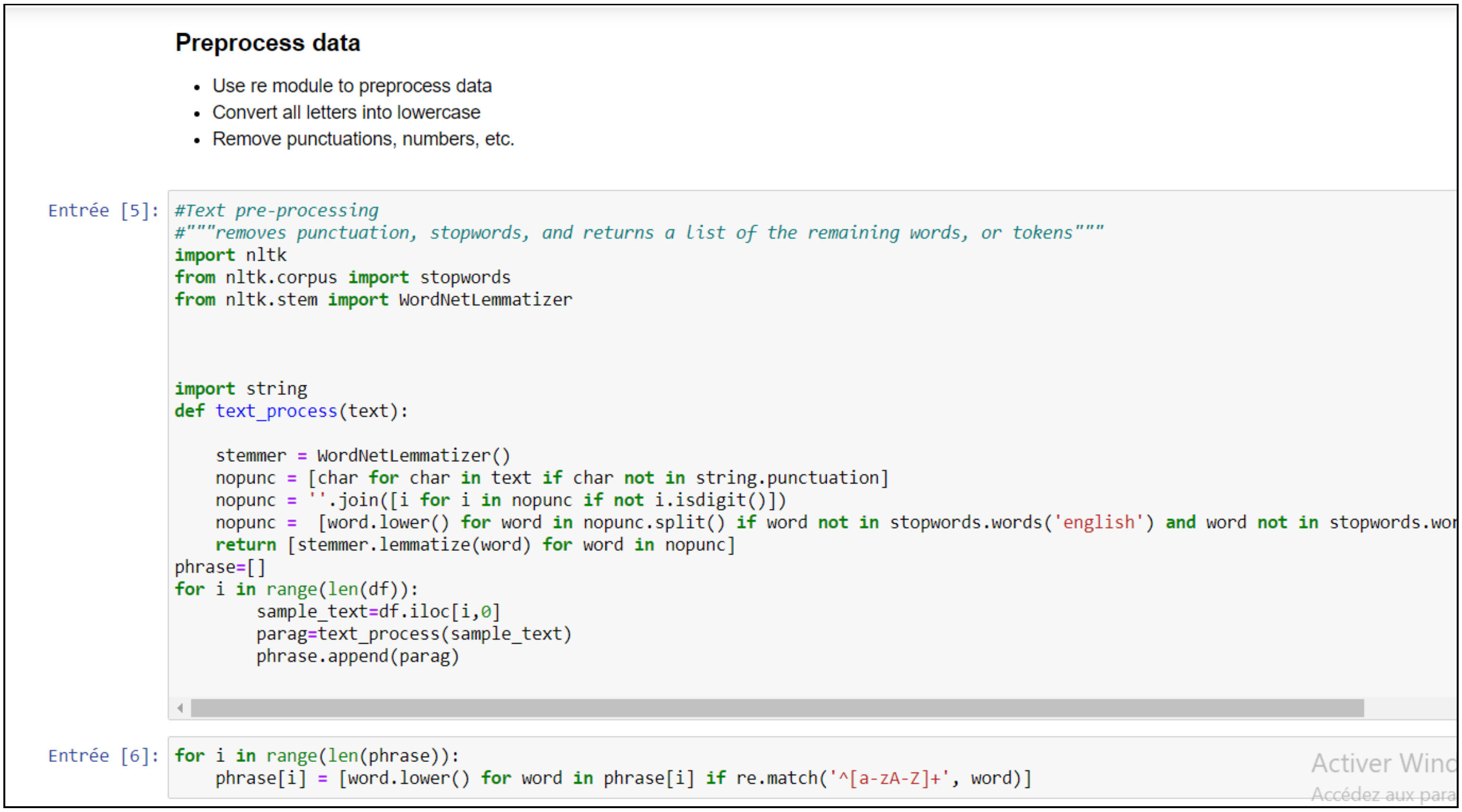
```
import speech_recognition as sr
print(' 0 -- Arabe \n 1 -- Français \n 2 -- Anglais')
a=input()
r = sr.Recognizer()
file = open("vocale.txt","a",encoding="utf_8")
with sr.Microphone() as source:
    print("speak ..")



    audio = r.listen(source)
    try:
        if a =='0':
            text = r.recognize_google(audio, language="ar-AR")
        elif a=='1':
            text = r.recognize_google(audio, language="fr-FR")
        elif a=='2':
            text = r.recognize_google(audio)
        else:
            print('choose between 0 /1/ 2 ')
        print(text)
        file.write("\n"+ str(text))
        file.close()
    except:
        print("I didn't get it")
```

```
 0 -- Arabe
 1 -- Français
 2 -- Anglais
1
speak ..
orange est un bon opérateur
```

image 19: Reading and processing data directly from the microphone

3. External Data Preparation using Natural Language Processing (NLP)

The data that we have collected through different forms and methods will be used within the Natural Language Processing context since it is basically comments and feedback .
Now , it is very important to prepare and preprocess our data before creating our sentiment analysis model .



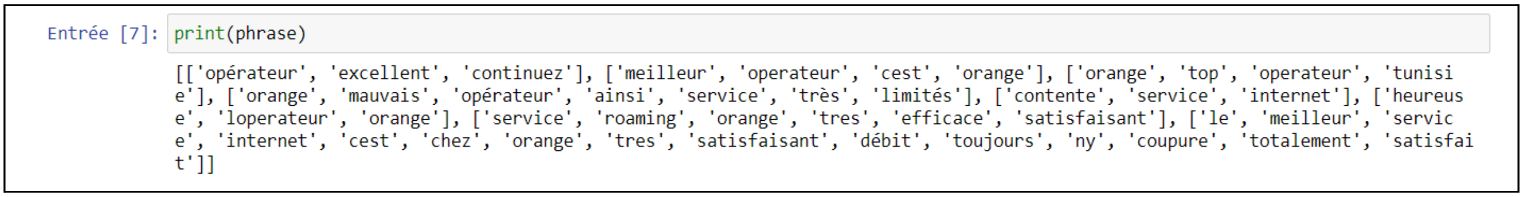
Preprocess data

- Use re module to preprocess data
- Convert all letters into lowercase
- Remove punctuations, numbers, etc.

```
Entrée [5]: #Text pre-processing
            #"""removes punctuation, stopwords, and returns a list of the remaining words, or tokens"""
            import nltk
            from nltk.corpus import stopwords
            from nltk.stem import WordNetLemmatizer



            import string
            def text_process(text):

                stemmer = WordNetLemmatizer()
                nopunc = [char for char in text if char not in string.punctuation]
                nopunc = ''.join([i for i in nopunc if not i.isdigit()])
                nopunc =  [word.lower() for word in nopunc.split() if word not in stopwords.words('english') and word not in stopwords.wor
                return [stemmer.lemmatize(word) for word in nopunc]
            phrase=[]
            for i in range(len(df)):
                    sample_text=df.iloc[i,0]
                    parag=text_process(sample_text)
                    phrase.append(parag)
```

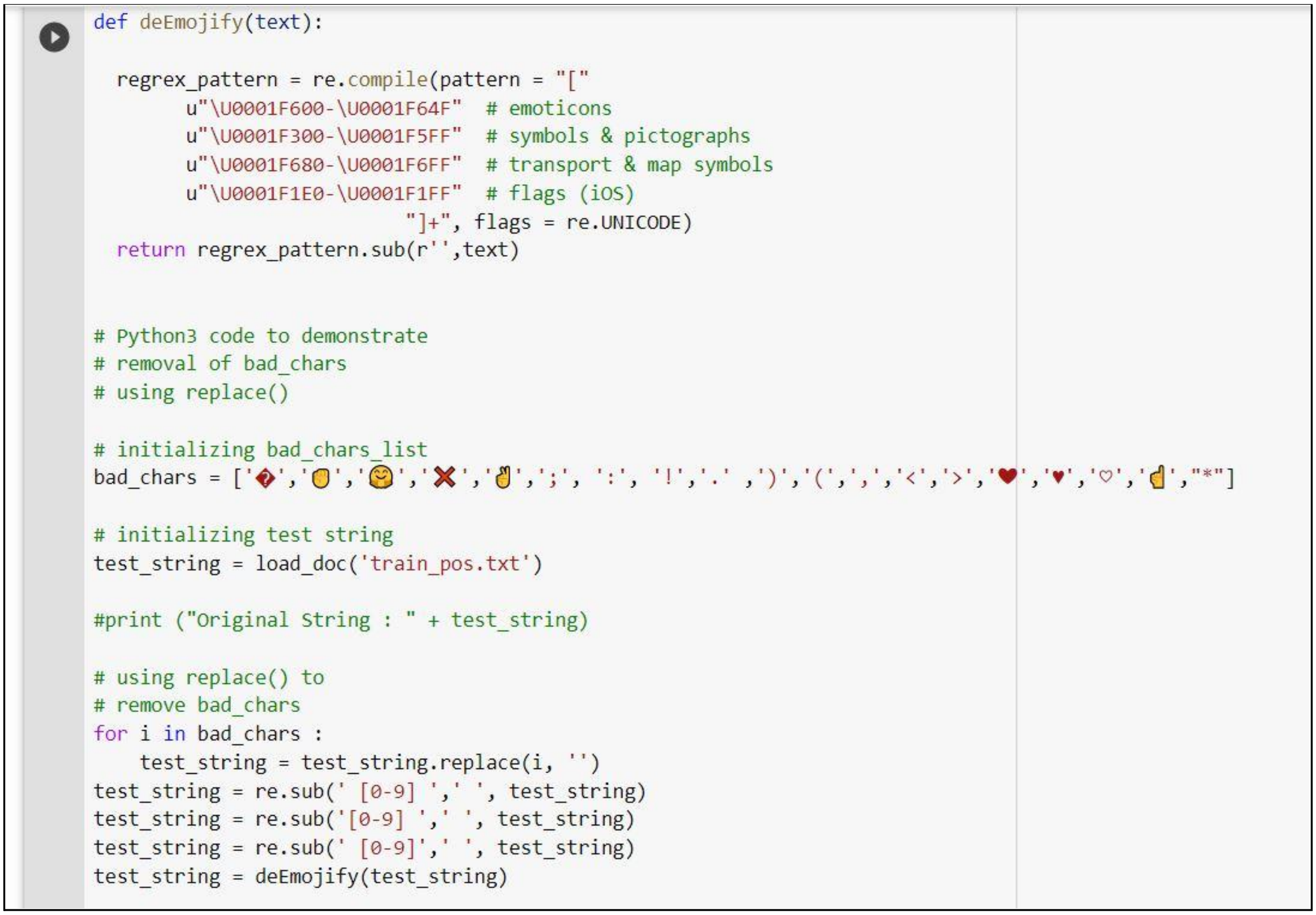
```
Entrée [6]: for i in range(len(phrase)):
                phrase[i] = [word.lower() for word in phrase[i] if re.match('^[a-zA-Z]+', word)]
```

image 20 : Using NLTK to preprocess text 1



```
Entrée [7]: print(phrase)

            [['opérateur', 'excellent', 'continuez'], ['meilleur', 'operateur', 'cest', 'orange'], ['orange', 'top', 'operateur', 'tunisi
            e'], ['orange', 'mauvais', 'opérateur', 'ainsi', 'service', 'très', 'limités'], ['contente', 'service', 'internet'], ['heureus
            e', 'loperateur', 'orange'], ['service', 'roaming', 'orange', 'tres', 'efficace', 'satisfaisant'], ['le', 'meilleur', 'servic
            e', 'internet', 'cest', 'chez', 'orange', 'tres', 'satisfaisant', 'débit', 'toujours', 'ny', 'coupure', 'totalement', 'satisfai
            t']]
```

image 21 : Using NLTK to preprocess text 2



```
def deEmojify(text):

  regrex_pattern = re.compile(pattern = "["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                           "]+", flags = re.UNICODE)
  return regrex_pattern.sub(r'',text)


# Python3 code to demonstrate
# removal of bad_chars
# using replace()

# initializing bad_chars_list
bad_chars = ['�','✊','🤗','❌','✌','；', '：', '！','.' ,')','(',',','<','>','❤','♥','♡','☝',"*"]

# initializing test string
test_string = load_doc('train_pos.txt')

#print ("Original String : " + test_string)

# using replace() to
# remove bad_chars
for i in bad_chars :
    test_string = test_string.replace(i, '')
test_string = re.sub(' [0-9] ',' ', test_string)
test_string = re.sub('[0-9] ',' ', test_string)
test_string = re.sub(' [0-9]',' ', test_string)
test_string = deEmojify(test_string)
```

image 22 : Punctuation emojis and numbers elimination function

As mentioned above, we eliminated unnecessary punctuation , emojis and digits in the comments then we normalized the text by transforming every character to the lower case.
And the most essential part is using NLTK library to :

- eliminate stop words that are not useful for any NLP model
- Using Stemming and Lemmatization to reduce inflection in words to their root and that is by mapping a group of words to the same stem .

Given the fact that our external data comes in different languages, this part is valid for french, english and arabic texts but specifically for the Tunisian dialect we decided to work different methods that will be explained in the next chapter.

IV. DATA MODELING AND EVALUATION

1. Data Modeling

1.1 Business Intelligence Modeling

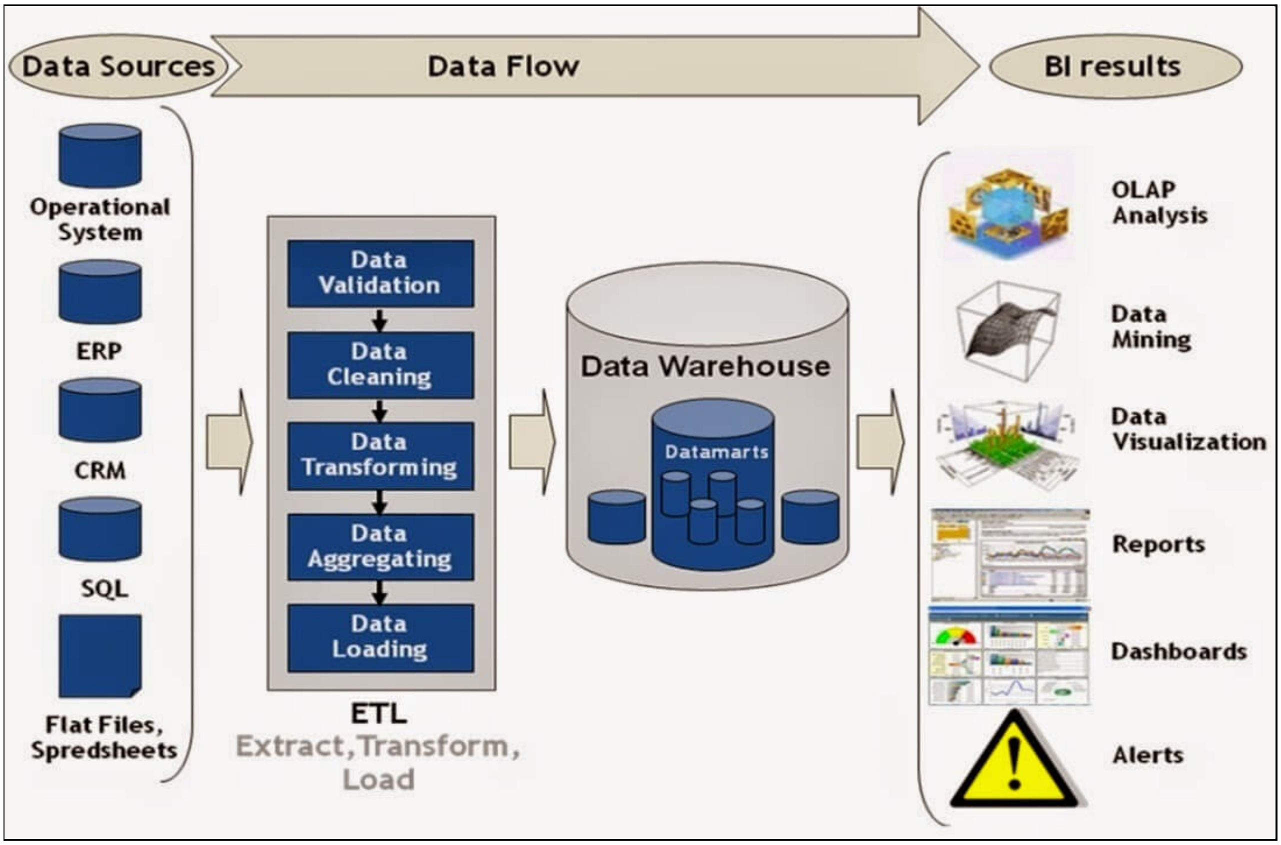


image 23 : Business Intelligence Modeling Process

1.1.1 SQL Server Management Studio (SSMS)

SQL Server Management Studio is the multilingual Microsoft SQL Server database management tool and allows interaction between the SQL code required to manipulate databases.

image 24 : SSMS Logo

A star schema, or “star” data model, is a multidimensional structure storing atomic or aggregated data, typically in data warehouses or datamarts.

A snowflake model is a model for which each dimension is represented with multiple tables. It is therefore more standardized (less redundant) than a star model.

Star and snowflake schemas are the most popular multidimensional data models used for a DataWarehouse. ... A schema is used to describe the entire database in a logical way. Likewise, the data warehouse requires a schema for its maintenance.

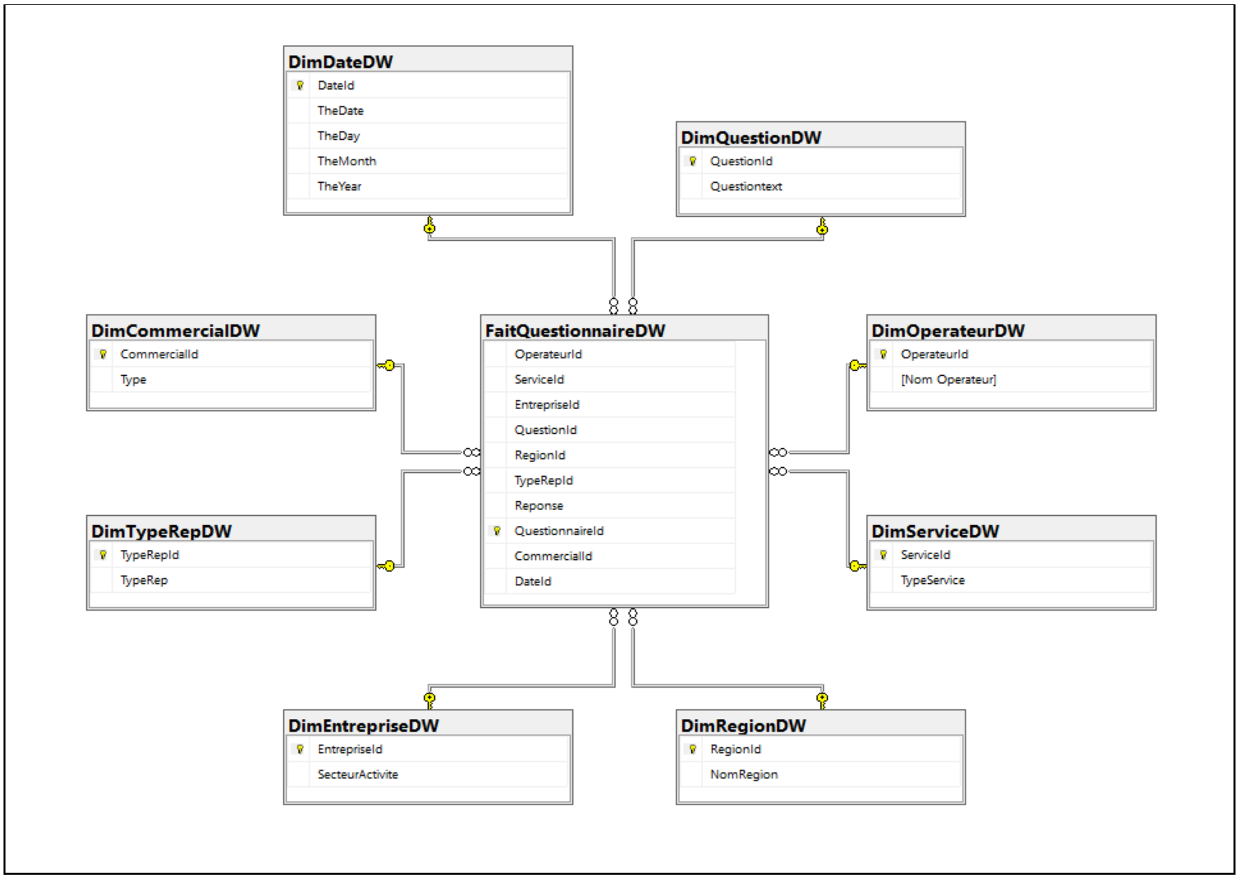


image 25 : Star Schema Model

We opted for the star model, in fact we have a table made up of a questionnaire and 8 dimensions:

DimEntrepriseDW ,DimRegionDW ,DimServiceDW ,DimTypeRepDW, DimQuestionDW DimDateDW, DimOperateurDW, DimCommercialDW

1.1.2 SQL Server Integration Studio (SSIS)

SQL Server Integration Services is a component of Microsoft SQL Server database software that can be used to perform a wide variety of data migration tasks. SSIS is a platform for data integration and workflow applications.

image 26 : SSIS Model

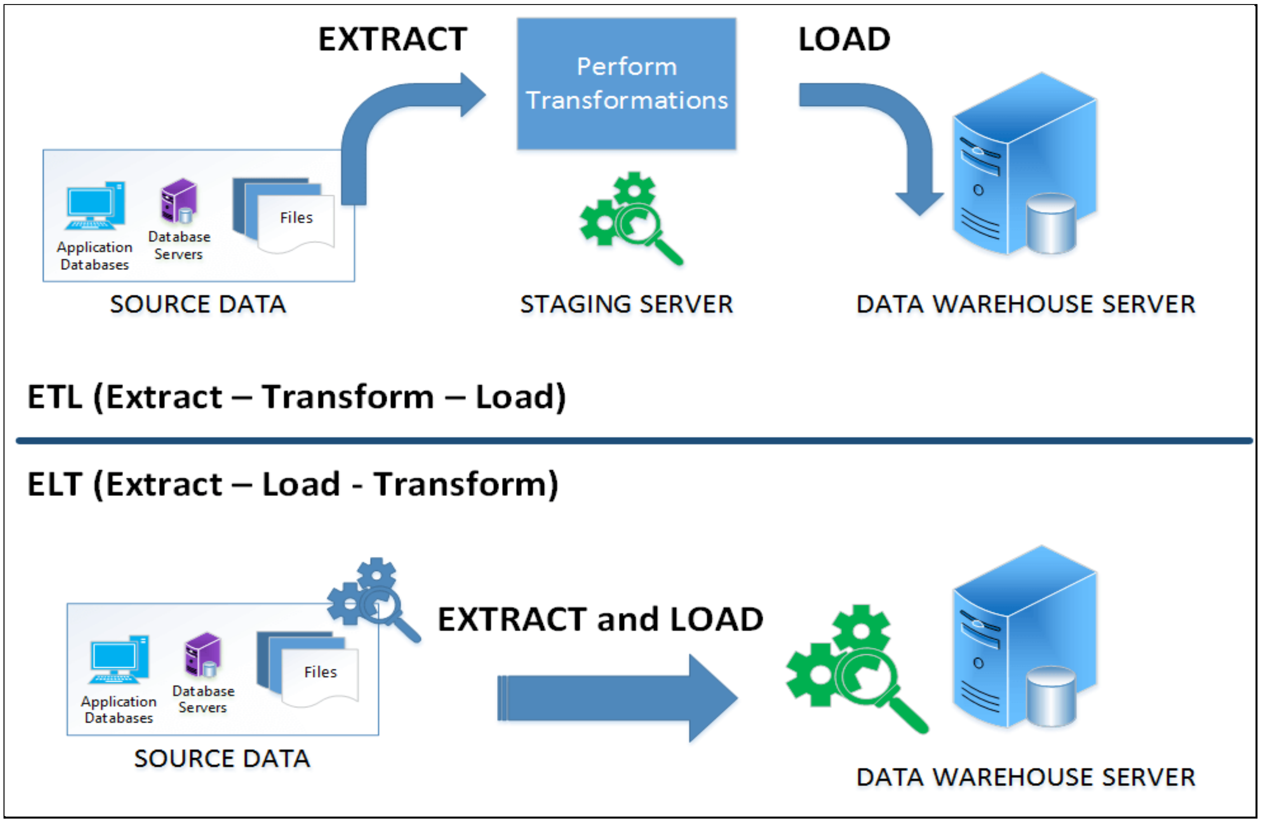


image 27 : ETL Process

ETL (Extract Transform Load) software allows you to extract raw data from a database, then restructure it, and finally load it into a Data Warehouse. This software has been around for a long time, but has evolved a lot to meet the new needs of the rise of Cloud, SaaS (Software as a Service) and Big Data.

EL (Extract, Load) is a data integration process that allows data to be transferred from a source system (production databases), and sent to the staging area.

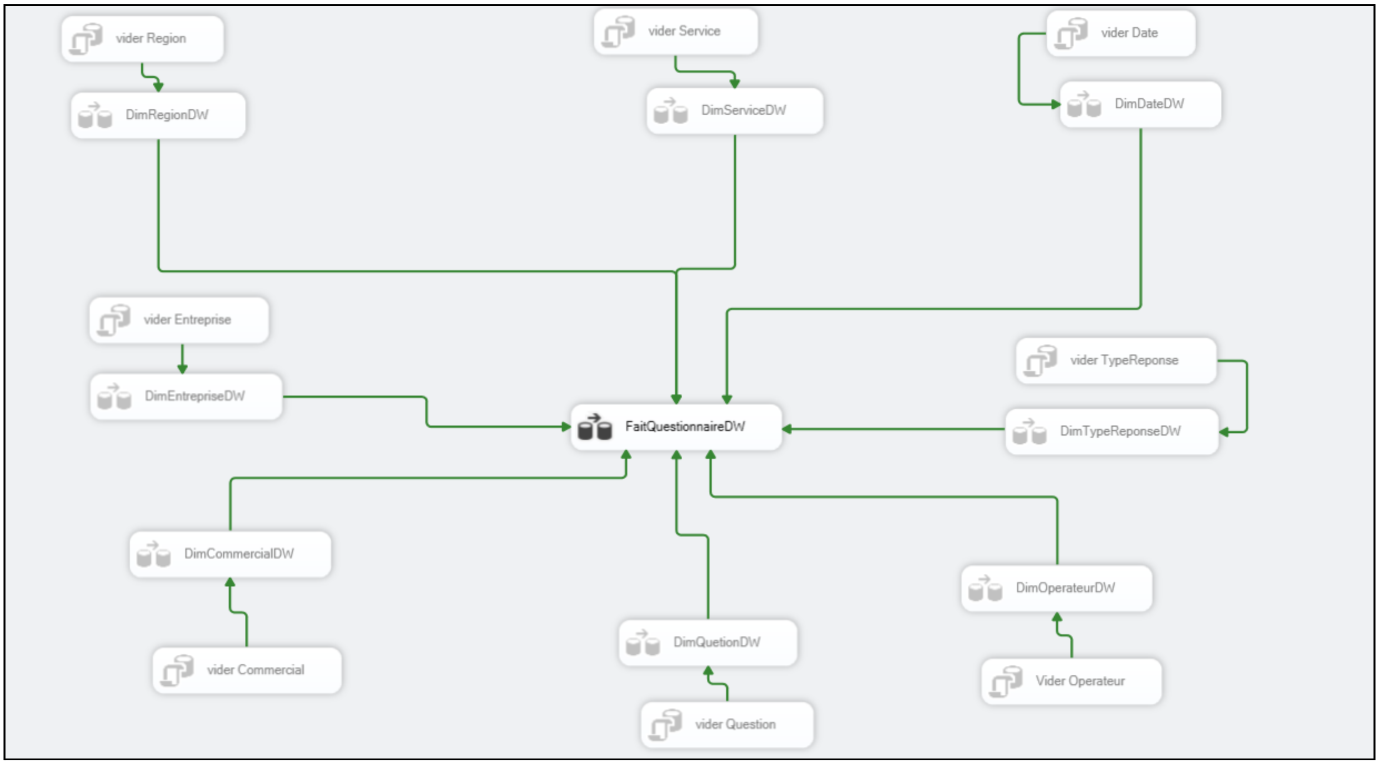


image 28 : SSIS Model

1.2 Data Science Modeling

1.2.1 Transfer Learning

The Transformers architecture as shown in image 23 is based solely on attention mechanisms. Unlike RNNs, Transformers do not require that the sequential data be processed in order, they use parallelization that helps reduce the training time.

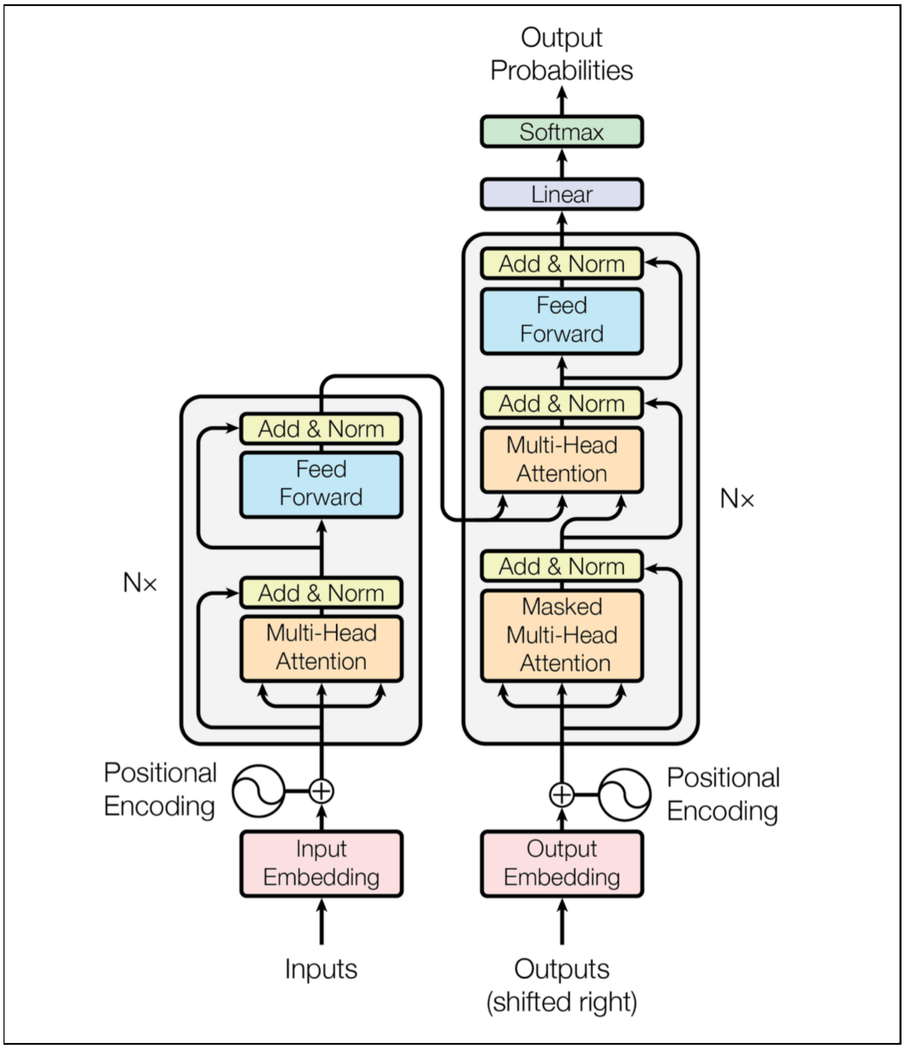


image 29 : The Transformer - model architecture

To understand Transformers we have to understand what attention mechanism means first.

A simple way to put it is that Attention mechanism is a powerful mechanism developed to enhance the performance of the Encoder-Decoder architecture on neural network-based machine translation tasks. It allows output to focus on input while producing output.

1.2.2 BERT Model

What is Bert ?

Bert is A Bi-Directional Encoder Representations for Transformers which is a natural language processing framework based on Transformers.

BERT uses a Multi-head Attention module which runs through an attention mechanism several times in parallel to focus on several distinct aspects of tokens. The independent attention outputs are then aggregated and linearly transformed into the expected dimension using Softmax function as shown in this figure. (image 24)

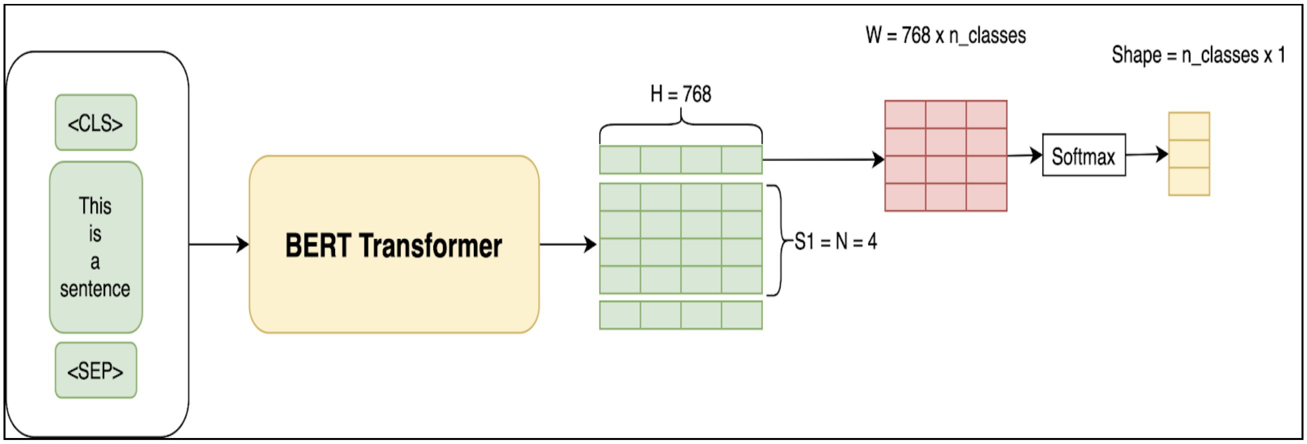


image 30 : BERT Transformer

Bert has two main pre-trained strategies :

- **The Mask Language Modeling strategy** which in short is replacing 80% of time the predicted word with a mask token and 20% of time with random words and that is because the lower layers leak information and allow a token to see itself in next layers
- **The Next sentence prediction NSP** which is the process of input going through the following stages:
  - ➔ Input Token embedding
  - ➔ Segment embedding
  - ➔ Position Embedding

As shown in the figure the sentence starts with the CLS token that stands for binary classification and ends with the SEP separator token.

All of the previous is the Input that goes into BERT transformer layers and is fed to the multihead attention to be processed later through query-Key-Value matrices and a softmax activation function will finally result in our model output.

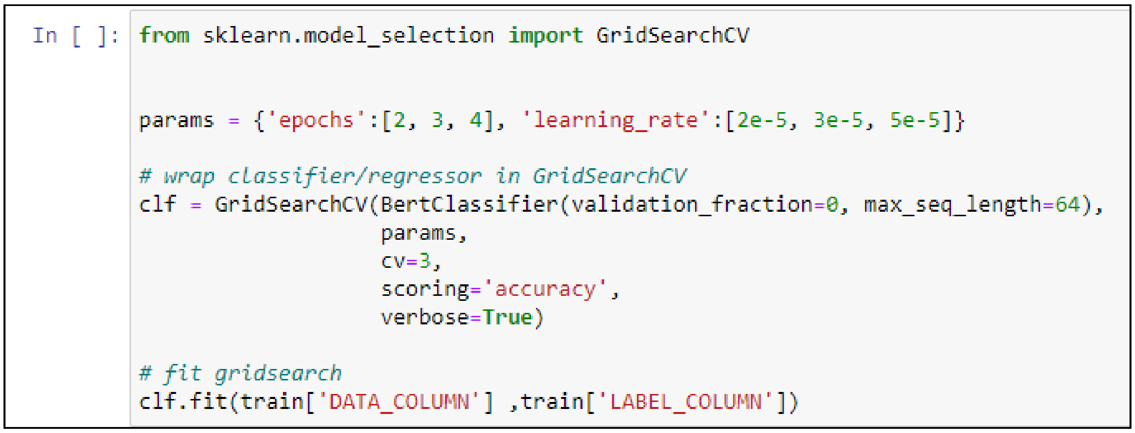
Why BERT ?

As we mentioned before, BERT ha shown a revolutionary groundbreaking technique in the world of natural language processing and here are the primary reasons why we should be including it in our project:

- ➔ It’s bidirectional
- ➔ It combines Mask Language Model (MLM) and Next Sentence Prediction (NSP).
- ➔ So far, it’s the best method in NLP to understand context-heavy texts

1.2.3 Hyperparameters Selection

It is important to make sure we’ve chosen the best combination of hyperparameters to make sure we obtain the most accurate predictions, for that we used the well known GridSearchCV for hyperparameters selection.

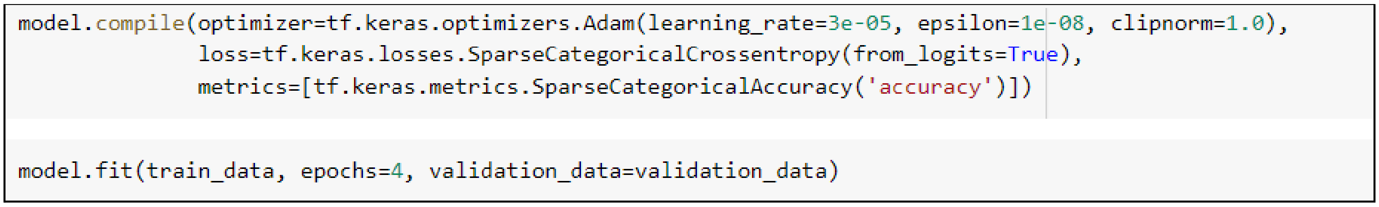


```
In [ ]: from sklearn.model_selection import GridSearchCV


        params = {'epochs':[2, 3, 4], 'learning_rate':[2e-5, 3e-5, 5e-5]}

        # wrap classifier/regressor in GridSearchCV
        clf = GridSearchCV(BertClassifier(validation_fraction=0, max_seq_length=64),
                           params,
                           cv=3,
                           scoring='accuracy',
                           verbose=True)

        # fit gridsearch
        clf.fit(train['DATA_COLUMN'] ,train['LABEL_COLUMN'])
```

image 31 : GridSearch Parameters Selection

1.2.4 Model Fine-Tuning

We will use BERT specifically for sentiment analysis classification by adding a classification layer on top of the transformer output .
So we will use Adam as our optimizer, Categorical_Crossentropy as our loss function, and SparseCategoricalAccuracy as our accuracy metric and then we will fine-tune our model according to the output of grid_search_cv which will set the number of epochs that are optimal for tuning our model .
In our case 4 epochs is convenient to train our data as mentioned below :



```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=3e-05, epsilon=1e-08, clipnorm=1.0),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=[tf.keras.metrics.SparseCategoricalAccuracy('accuracy')])
```

```
model.fit(train_data, epochs=4, validation_data=validation_data)
```

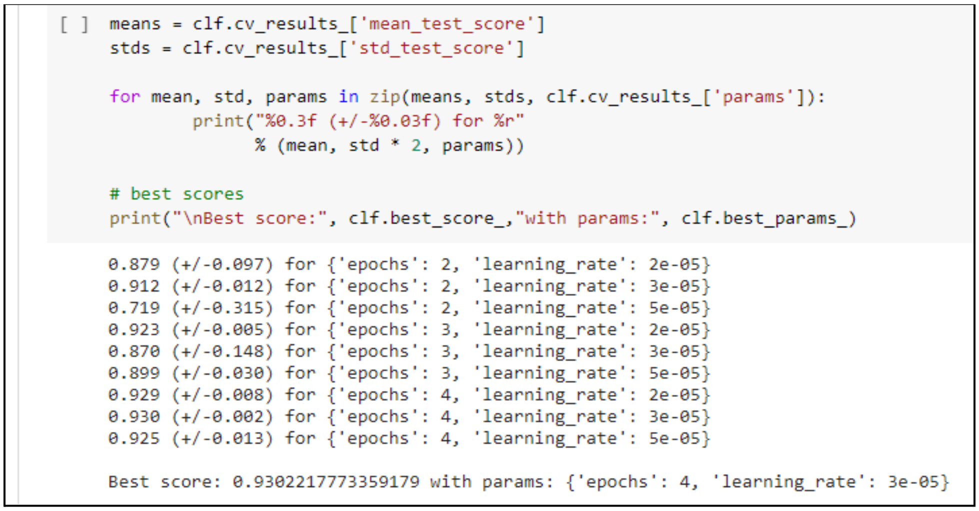
image 32 : Model Fine-Tuning

2. Model Evaluation

The purpose of this part is to quantify the “goodness” of our models, and that is through evaluation methods that will provide us with metrics which will serve as an insight into our models’ performance.

2.1 BERT Model Evaluation For Tunisian Dialect

GridSearchCV output:

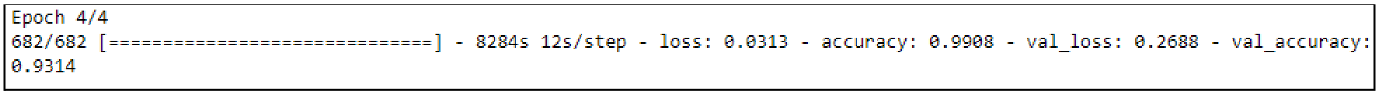


```
[ ] means = clf.cv_results_['mean_test_score']
    stds = clf.cv_results_['std_test_score']

    for mean, std, params in zip(means, stds, clf.cv_results_['params']):
            print("%0.3f (+/-%0.03f) for %r"
                  % (mean, std * 2, params))

    # best scores
    print("\nBest score:", clf.best_score_,"with params:", clf.best_params_)
```

```
0.879 (+/-0.097) for {'epochs': 2, 'learning_rate': 2e-05}
0.912 (+/-0.012) for {'epochs': 2, 'learning_rate': 3e-05}
0.719 (+/-0.315) for {'epochs': 2, 'learning_rate': 5e-05}
0.923 (+/-0.005) for {'epochs': 3, 'learning_rate': 2e-05}
0.870 (+/-0.148) for {'epochs': 3, 'learning_rate': 3e-05}
0.899 (+/-0.030) for {'epochs': 3, 'learning_rate': 5e-05}
0.929 (+/-0.008) for {'epochs': 4, 'learning_rate': 2e-05}
0.930 (+/-0.002) for {'epochs': 4, 'learning_rate': 3e-05}
0.925 (+/-0.013) for {'epochs': 4, 'learning_rate': 5e-05}

Best score: 0.9302217773359179 with params: {'epochs': 4, 'learning_rate': 3e-05}
```

image 33 : GridSearchCV Output

Model accuracy:



```
Epoch 4/4
682/682 [==============================] - 8284s 12s/step - loss: 0.0313 - accuracy: 0.9908 - val_loss: 0.2688 - val_accuracy:
0.9314
```

image 34 : Model Accuracy

Confusion matrix:

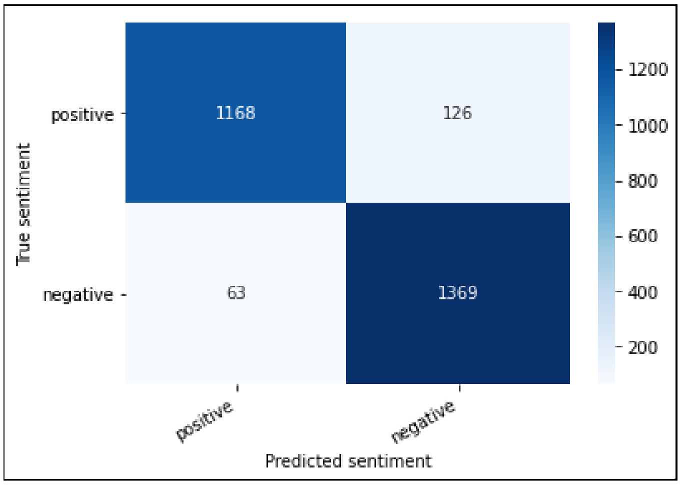
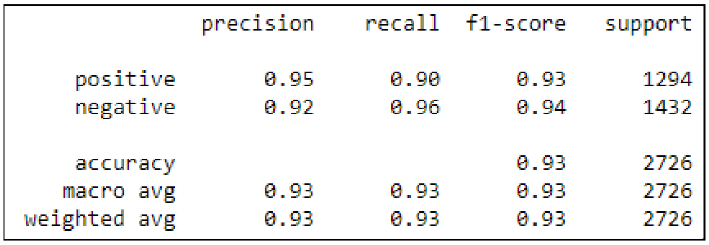


image 35 : Confusion Matrix

Evaluation Report:

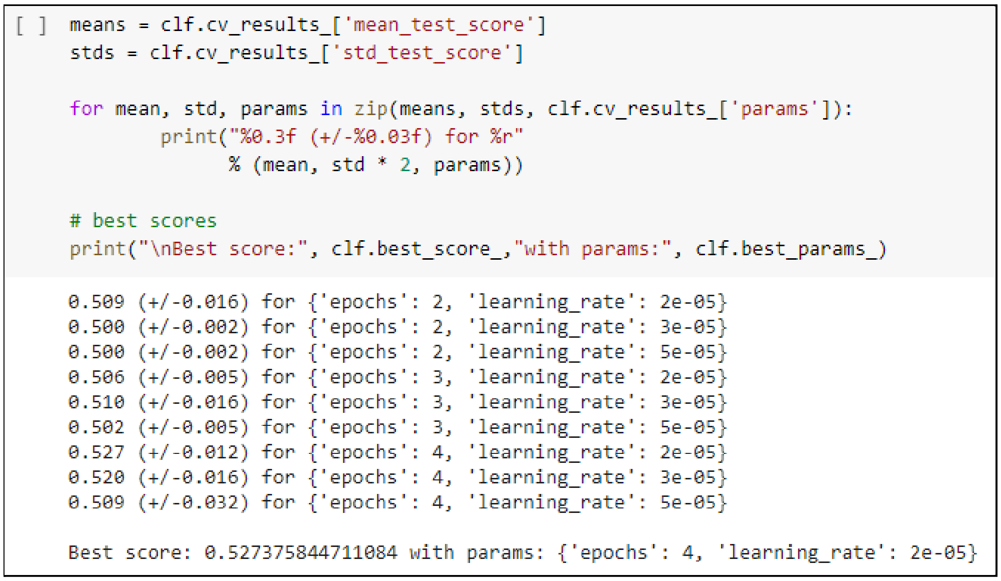


```
              precision    recall  f1-score   support

    positive       0.95      0.90      0.93      1294
    negative       0.92      0.96      0.94      1432

    accuracy                           0.93      2726
   macro avg       0.93      0.93      0.93      2726
weighted avg       0.93      0.93      0.93      2726
```

image 35 : Evaluation Report

2.2 BERT Model Evaluation For English

GridSearchCV output:

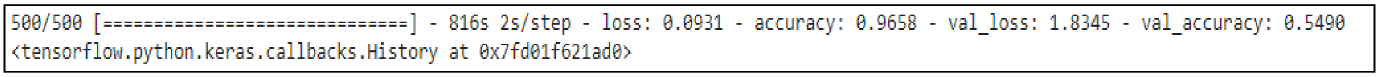


```
[ ]  means = clf.cv_results_['mean_test_score']
     stds = clf.cv_results_['std_test_score']

     for mean, std, params in zip(means, stds, clf.cv_results_['params']):
             print("%0.3f (+/-%0.03f) for %r"
                   % (mean, std * 2, params))

     # best scores
     print("\nBest score:", clf.best_score_,"with params:", clf.best_params_)
```

```
0.509 (+/-0.016) for {'epochs': 2, 'learning_rate': 2e-05}
0.500 (+/-0.002) for {'epochs': 2, 'learning_rate': 3e-05}
0.500 (+/-0.002) for {'epochs': 2, 'learning_rate': 5e-05}
0.506 (+/-0.005) for {'epochs': 3, 'learning_rate': 2e-05}
0.510 (+/-0.016) for {'epochs': 3, 'learning_rate': 3e-05}
0.502 (+/-0.005) for {'epochs': 3, 'learning_rate': 5e-05}
0.527 (+/-0.012) for {'epochs': 4, 'learning_rate': 2e-05}
0.520 (+/-0.016) for {'epochs': 4, 'learning_rate': 3e-05}
0.509 (+/-0.032) for {'epochs': 4, 'learning_rate': 5e-05}

Best score: 0.527375844711084 with params: {'epochs': 4, 'learning_rate': 2e-05}
```

image 36 : GridSearchCV Output

Model accuracy:



```
500/500 [==============================] - 816s 2s/step - loss: 0.0931 - accuracy: 0.9658 - val_loss: 1.8345 - val_accuracy: 0.5490
<tensorflow.python.keras.callbacks.History at 0x7fd01f621ad0>
```

image 37 : Model Accuracy

Confusion matrix:

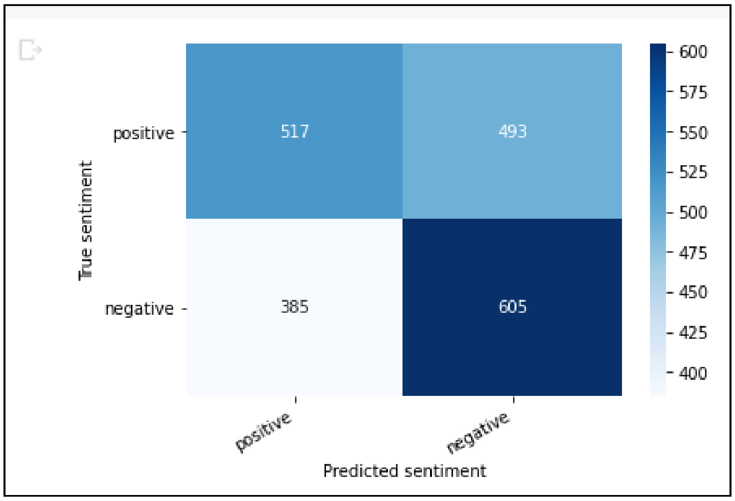
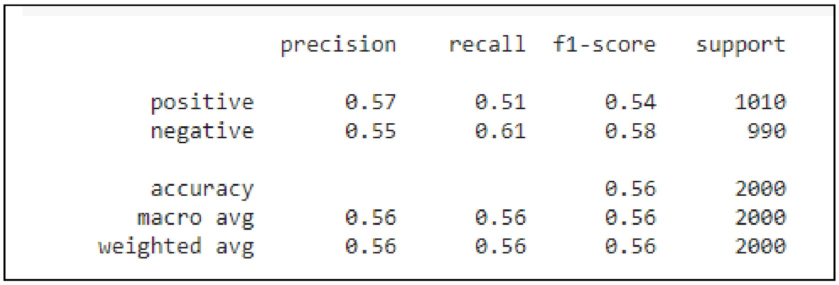


image 38 : Confusion Matrix

Evaluation Report:

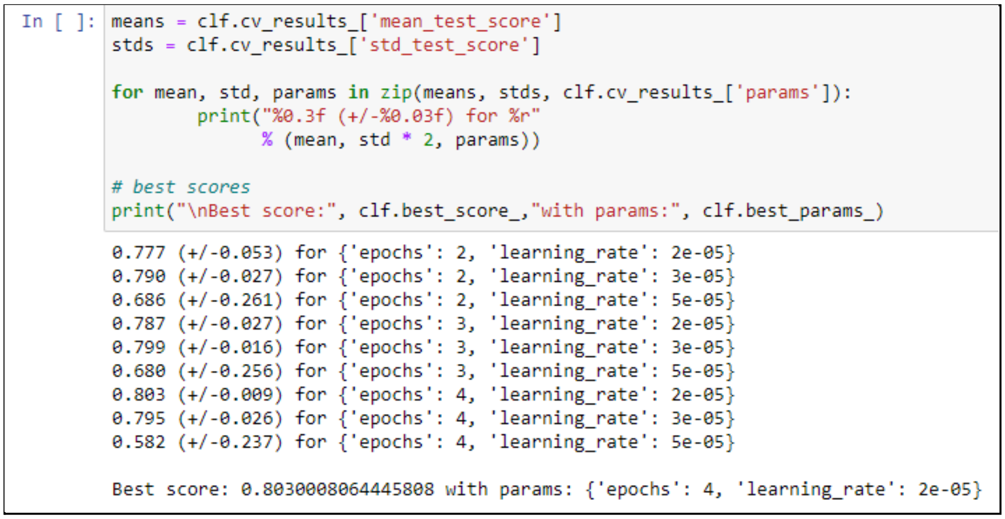


```
              precision    recall  f1-score   support

    positive       0.57      0.51      0.54      1010
    negative       0.55      0.61      0.58       990

    accuracy                           0.56      2000
   macro avg       0.56      0.56      0.56      2000
weighted avg       0.56      0.56      0.56      2000
```

image 39 : Evaluation Report

2.3 BERT Model Evaluation For French

GridSearchCV output:

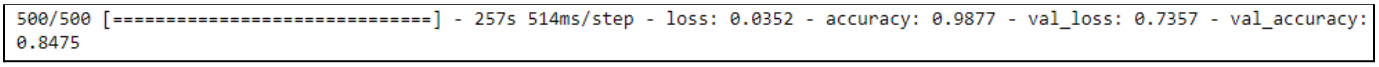


```
In [ ]: means = clf.cv_results_['mean_test_score']
        stds = clf.cv_results_['std_test_score']

        for mean, std, params in zip(means, stds, clf.cv_results_['params']):
                print("%0.3f (+/-%0.03f) for %r"
                      % (mean, std * 2, params))

        # best scores
        print("\nBest score:", clf.best_score_,"with params:", clf.best_params_)
```

```
0.777 (+/-0.053) for {'epochs': 2, 'learning_rate': 2e-05}
0.790 (+/-0.027) for {'epochs': 2, 'learning_rate': 3e-05}
0.686 (+/-0.261) for {'epochs': 2, 'learning_rate': 5e-05}
0.787 (+/-0.027) for {'epochs': 3, 'learning_rate': 2e-05}
0.799 (+/-0.016) for {'epochs': 3, 'learning_rate': 3e-05}
0.680 (+/-0.256) for {'epochs': 3, 'learning_rate': 5e-05}
0.803 (+/-0.009) for {'epochs': 4, 'learning_rate': 2e-05}
0.795 (+/-0.026) for {'epochs': 4, 'learning_rate': 3e-05}
0.582 (+/-0.237) for {'epochs': 4, 'learning_rate': 5e-05}

Best score: 0.8030008064445808 with params: {'epochs': 4, 'learning_rate': 2e-05}
```

image 40 : GridSearchCV output

Model accuracy:



```
500/500 [==============================] - 257s 514ms/step - loss: 0.0352 - accuracy: 0.9877 - val_loss: 0.7357 - val_accuracy:
0.8475
```

image 41 : Model Accuracy

Confusion matrix:

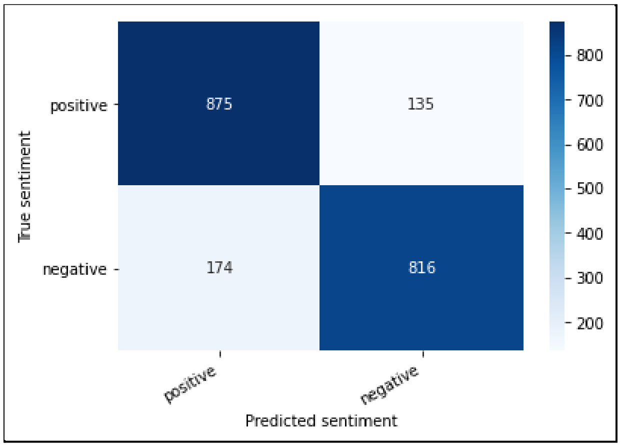
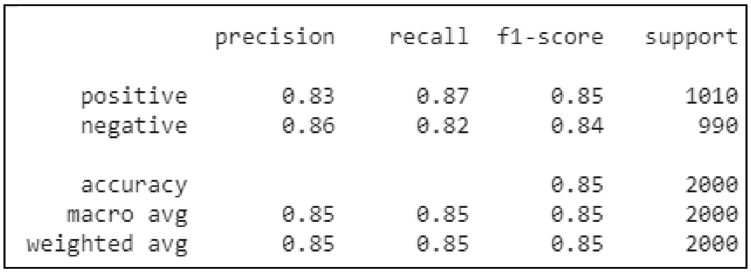


image 42 : Confusion Matrix

Evaluation Report:



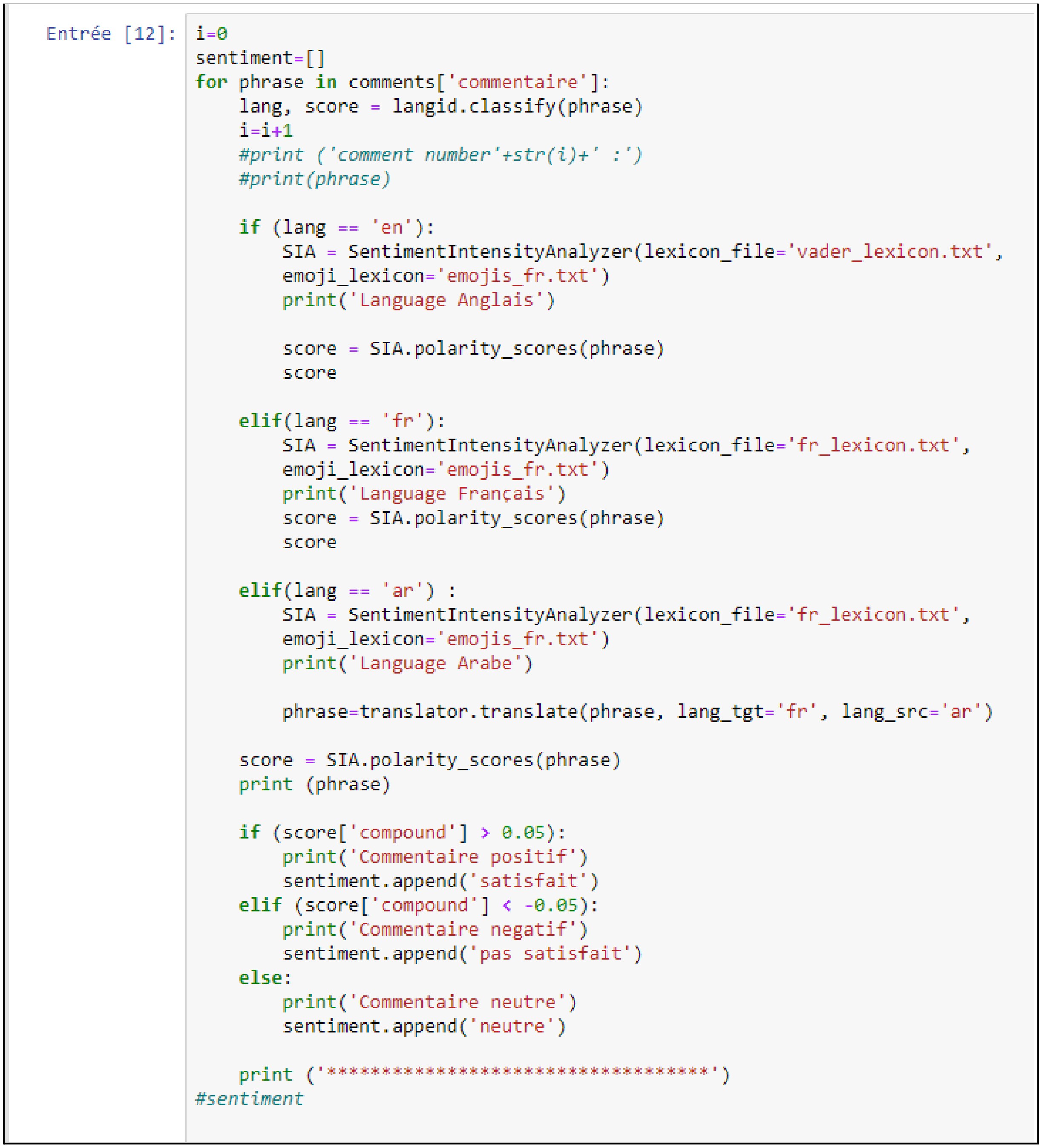
```
              precision    recall  f1-score   support

    positive       0.83      0.87      0.85      1010
    negative       0.86      0.82      0.84       990

    accuracy                           0.85      2000
   macro avg       0.85      0.85      0.85      2000
weighted avg       0.85      0.85      0.85      2000
```

image 43 : Evaluation Report

3. VADER Model

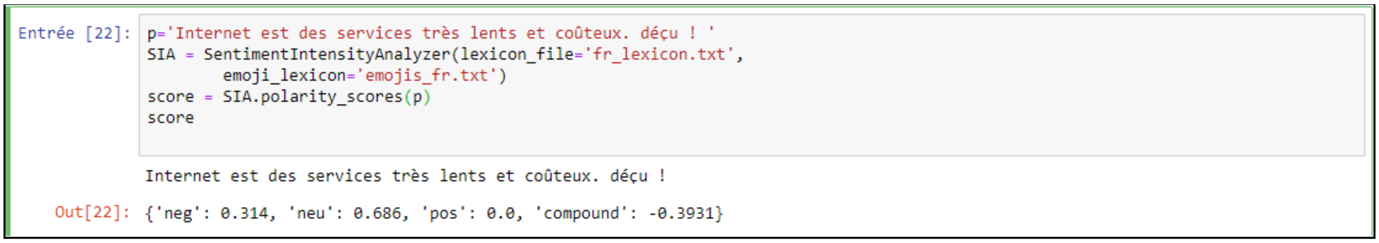
VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media which is basically the source of our data .
In order to diversify our NLP models we decided to use the framework VADER for the various reasons, the most important one are the following :

- It does not require any training data
- It can very well understand the sentiment of a text containing emoticons, slangs, conjunctions, capital words, punctuations and much more
- VADER works with various languages like english , french , arabic



```
Entrée [12]: i=0
             sentiment=[]
             for phrase in comments['commentaire']:
                 lang, score = langid.classify(phrase)
                 i=i+1
                 #print ('comment number'+str(i)+' :')
                 #print(phrase)

                 if (lang == 'en'):
                     SIA = SentimentIntensityAnalyzer(lexicon_file='vader_lexicon.txt',
                     emoji_lexicon='emojis_fr.txt')
                     print('Language Anglais')

                     score = SIA.polarity_scores(phrase)
                     score

                 elif(lang == 'fr'):
                     SIA = SentimentIntensityAnalyzer(lexicon_file='fr_lexicon.txt',
                     emoji_lexicon='emojis_fr.txt')
                     print('Language Français')
                     score = SIA.polarity_scores(phrase)
                     score

                 elif(lang == 'ar') :
                     SIA = SentimentIntensityAnalyzer(lexicon_file='fr_lexicon.txt',
                     emoji_lexicon='emojis_fr.txt')
                     print('Language Arabe')

                     phrase=translator.translate(phrase, lang_tgt='fr', lang_src='ar')

                 score = SIA.polarity_scores(phrase)
                 print (phrase)

                 if (score['compound'] > 0.05):
                     print('Commentaire positif')
                     sentiment.append('satisfait')
                 elif (score['compound'] < -0.05):
                     print('Commentaire negatif')
                     sentiment.append('pas satisfait')
                 else:
                     print('Commentaire neutre')
                     sentiment.append('neutre')

                 print ('************************************')
             #sentiment
```

image 44 : Vader Framework implementation



```
Entrée [22]: p='Internet est des services très lents et coûteux. déçu ! '
             SIA = SentimentIntensityAnalyzer(lexicon_file='fr_lexicon.txt',
                     emoji_lexicon='emojis_fr.txt')
             score = SIA.polarity_scores(p)
             score

             Internet est des services très lents et coûteux. déçu !

    Out[22]: {'neg': 0.314, 'neu': 0.686, 'pos': 0.0, 'compound': -0.3931}
```

image 45: Vader Framework Output

DATA VISUALIZATION

1. PowerBi :

Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. Your data may be an Excel spreadsheet, or a collection of cloud-based and on-premises hybrid data warehouses. Power BI lets you easily connect to your data sources, visualize and discover what's important, and share that with anyone or everyone you want.

image 46 : PowerBI Logo

➔ **To actively sum up things, PowerBi allows the creation of personalized and interactive data visualizations with an interface simple enough for end users to understand their own reports and dashboards.**

2. EasyDash Dashboard :

Our product is a dashboard named EasyDash, containing a collection of graphs and illustrations allowing the provider Orange to visualize the satisfaction rate by analyzing the data of their clients and representing it into coherent insights.

It contains four different pages, each one has specific goals to represent alongside to a menu on the left panel to navigate between the different parts of the product:

- Orange Satisfaction and their competitor
- Satisfaction by Date
- Social media analysis
- Interaction

2.1 Orange Satisfaction and their competitor:

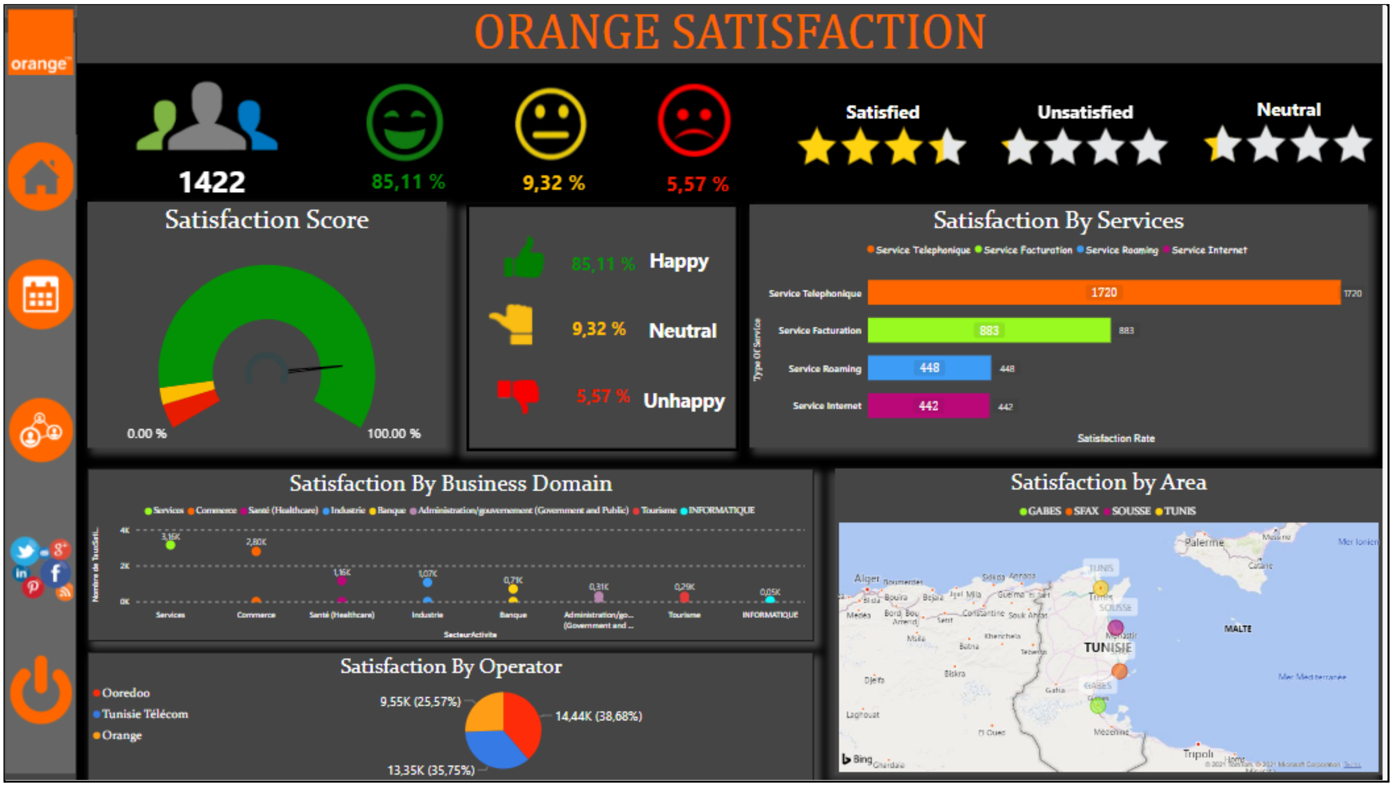
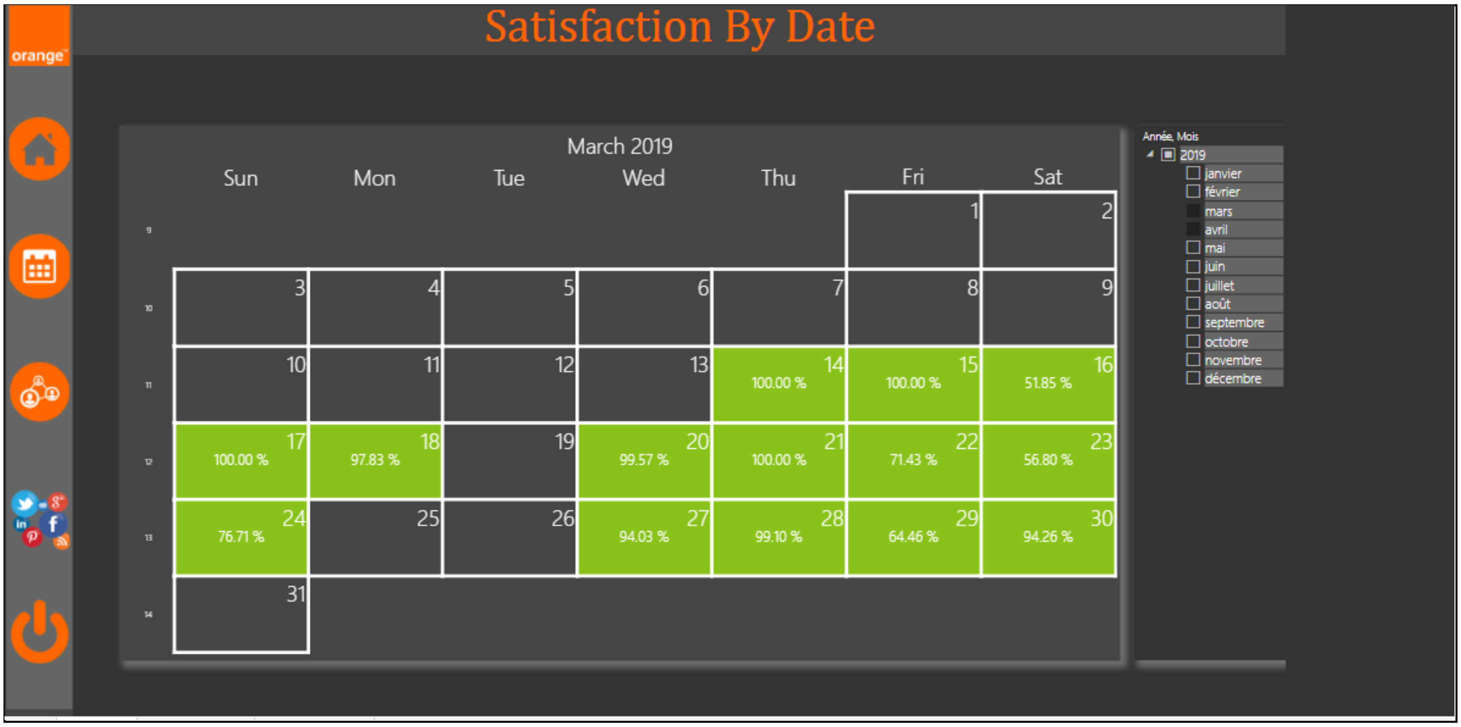


image 47 : EasyDash Prime Interface

➔ This page illustrates the internal data stored in our data WareHouse. It presents the satisfaction rate by the following axis :
Provider's services
Clients' Business Domain
Geographical Area
Telecommunication Operator

➔ In General among **1422** responses to the questionnaire we have :
85.11 % of Satisfied Clients
9.32% of neutral Clients
5.57% of unsatisfied Clients

2.2 Satisfaction By Date:



Satisfaction By Date

March 2019

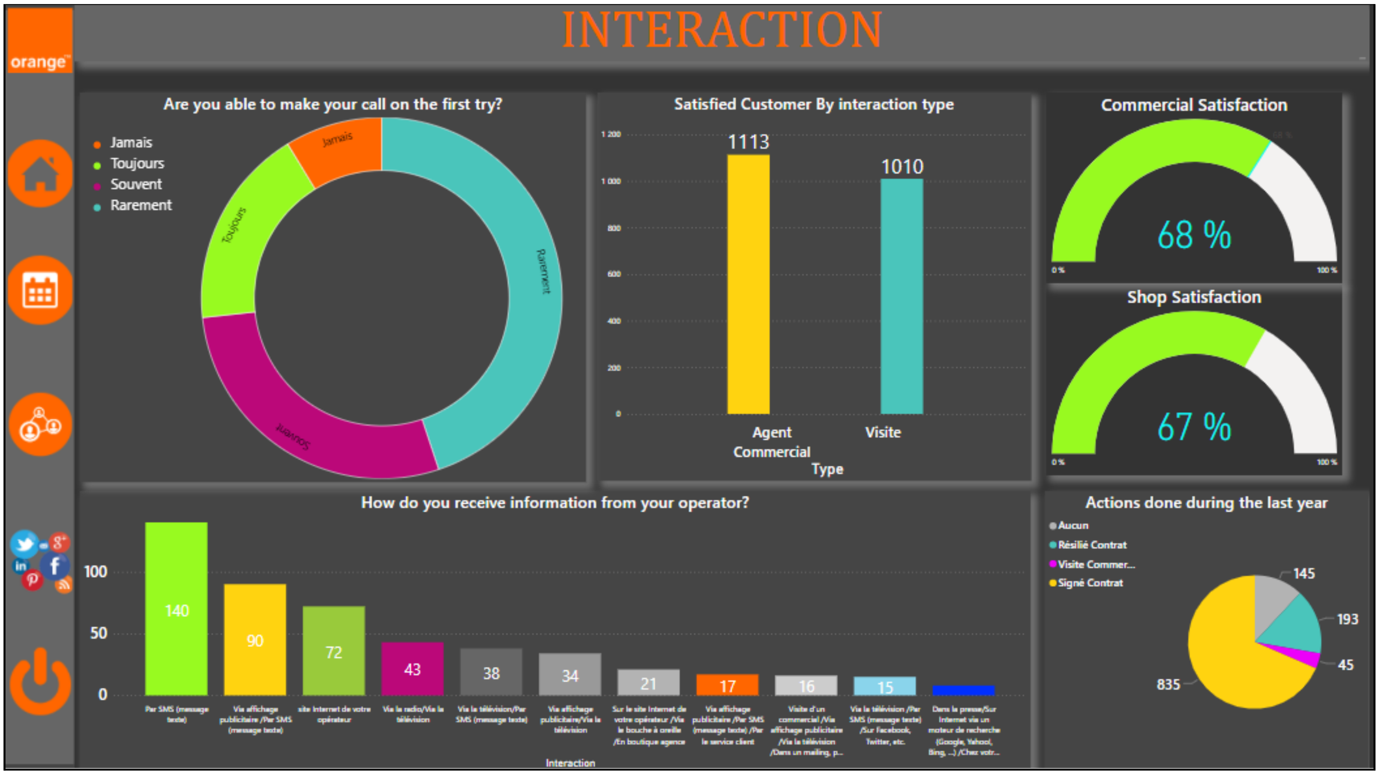
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
9						1	2
10	3	4	5	6	7	8	9
11	10	11	12	13	14 100.00 %	15 100.00 %	16 51.85 %
12	17 100.00 %	18 97.83 %	19	20 99.57 %	21 100.00 %	22 71.43 %	23 56.80 %
13	24 76.71 %	25	26	27 94.03 %	28 99.10 %	29 64.46 %	30 94.26 %
14	31						

image 48 : Calendar visualization for the clients Satisfaction Interface

- ➔ As for the second component of the Dashboard, We represented the data relative to the date axis in the form of a calendar where every cell is a single date containing the rate of the satisfaction associated to that day.
- ➔ This will provide a clear visibility of the variations of the clients' satisfaction in function of the date axis.

Confirming the results observed in the first component (Orange Satisfaction), we can see that most of the clients are satisfied with variant rates (ex: 100% on the 14th of March 2019 and 56.80% on the 23rd of the same month) .

2.3 Interaction:



INTERACTION
orange
Are you able to make your call on the first try?
Jamais
Toujours
Souvent
Rarement
Satisfied Customer By interaction type
1 200
1 000
800
600
400
200
0
1113
1010
Agent Commercial
Visite
Type
Commercial Satisfaction
68 %
0 %
100 %
Shop Satisfaction
67 %
0 %
100 %
How do you receive information from your operator?
100
50
0
140
90
72
43
38
34
21
17
16
15
Par SMS (message texte)
Via affichage publicitaire /Par SMS (message texte)
site Internet de votre opérateur
Via la radio/Via la télévision
Via la télévision/Par SMS (message texte)
Via affichage publicitaire/Via la télévision
Sur le site Internet de votre opérateur /Via le bouche à oreille /En boutique agence
Via affichage publicitaire /Par SMS (message texte) /Par le service client
Visite d'un commercial /Via affichage publicitaire /Via la télévision /Dans un mailing, p...
Via la télévision /Par SMS (message texte) /Sur Facebook, Twitter, etc.
Dans la presse/Sur Internet via un moteur de recherche (Google, Yahoo!, Bing, ...) /Chez votr...
Interaction
Actions done during the last year
Aucun
Résilié Contrat
Visite Commer...
Signé Contrat
145
193
45
835

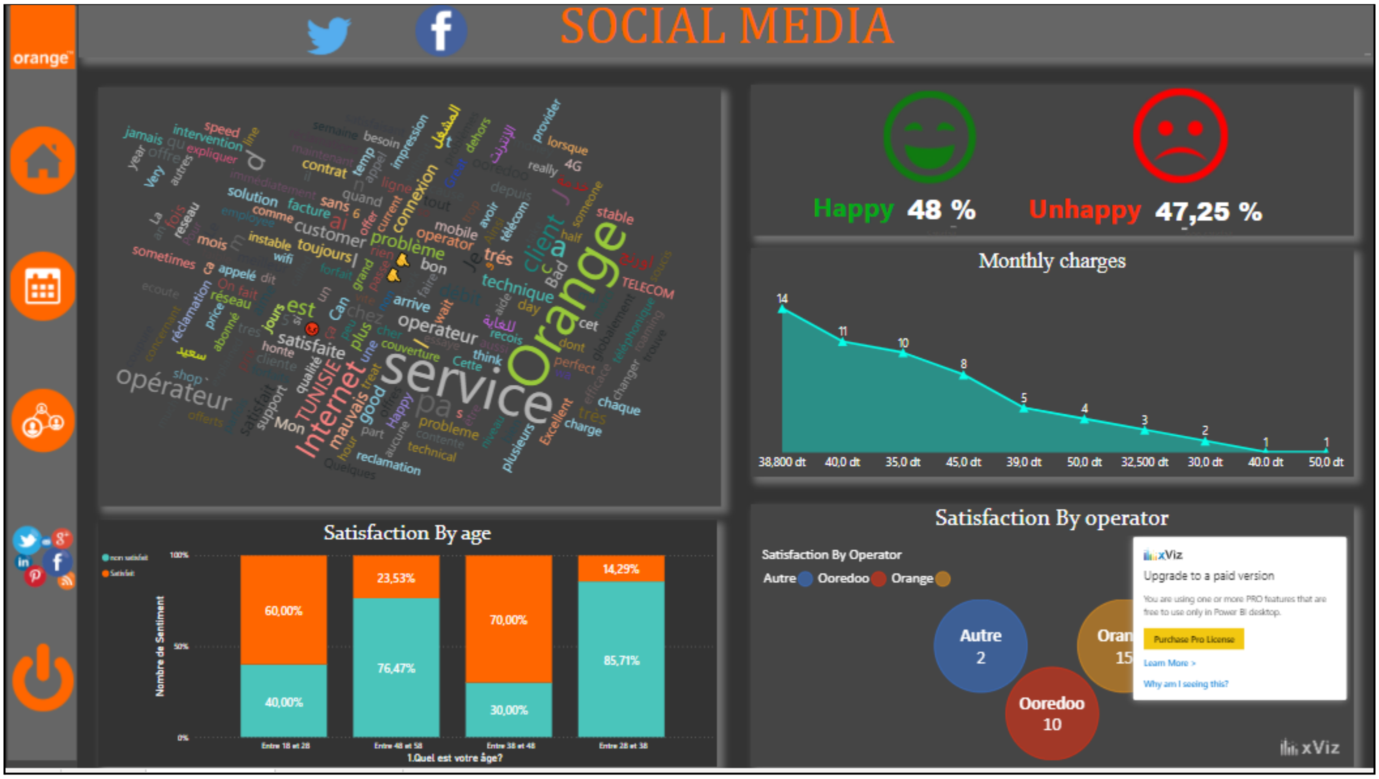
image 49 : Interaction's Evaluation Interface

- ➔ This part treats the information regarding the evaluation of the interaction between the provider Orange and his customers as well as the key components of every interactivity such as the Actions done during the past year, the information spread strategies of
  Orange and their efficiency and the effectiveness of the customers' service.

➔ This analysis will help improve the quality of the professional relationship between the provider Orange and his customers.

Observing the results of this page, It's obvious that the company needs to work on the quality of the interaction, scoring a 68% satisfaction rate for the Commercial and 67% for the Shop as well as a big part of the clients rarely succeed to make their call on their first try.

2.4 Social Media:



Upgrade to a paid version

You are using one or more PRO features that are free to use only in Power BI desktop.

image 50 : Social Media Analysis Interface

➔ From this page we can visualize our external data collected From social networks as shown above we used **Word Cloud** as a tool to visualize the most used words by customers. In addition to that, We tried to add other axes as **the Age** and **Monthly Charges.**

➔ The analysis resulted in :

48% of satisfied customers
47.25% of unsatisfied customers

This component is a very important part of the product because it treats the spontaneous reaction of customers resulting in different results than the classic methods.

CONCLUSION AND PERSPECTIVES

EasyDash has demonstrated its great value in analysing the customers' both internal and external data through graphs, charts, maps and illustrations. Each component of the dashboard offers our clients a different angle through which they can visualize a certain aspect of their data. We have built EasyDash going through a number of major steps following IBM Master plan methodology as we previously mentioned. We started off cleaning our internal data, preparing it to populate the Business Intelligence infrastructure that aims to significantly enhance our client's decision making and analysis of different KPIs (Key Performance Indicators).

Bringing this project to reach its prime objective has provided us with the opportunity to work thoroughly within not only the Data Science field but also within the Business Intelligence's. And just like any project, EasyDash dashboard still opens a multitude of perspectives that will be interesting to add in the future as an upgrade:

➢ EasyDash can be applied to any business that focuses on customer satisfaction even outside of the telecommunication sector which is currently our target market

➢ EasyDash can be deployed as a mobile application so that our clients are always a click away from consulting their position in the market