5222: Feature Engineering ICE#1

Rubric

1. Complete and proper Github Submission (10%)
2. Complete and proper submission to Canvas (5%)
3. Source Code (50%)
4. Explaining the answers (30%)
5. Commenting, formatting, and visualizing your code properly and timely submission (5%)

Please goto https://towardsdatascience.com/text-classification-in-python-dd95d264c802 and follow the article.

This article is for text classification using python.

Their Github is available at https://github.com/miguelfzafra/Latest-News-Classifier/tree/master/0.%20Latest%20News%20Classifier.

Follow their step 00,01,02,03 and 04. Use the same workbook to execute your code

```
!pip install altair vega_datasets notebook vega
import pandas as pd
import matplotlib.pyplot as plt
import pickle
import seaborn as sns
sns.set_style("whitegrid")
import altair as alt
alt.renderers.enable("notebook")

# Code for hiding seaborn warnings
import warnings
warnings.filterwarnings("ignore")
```

```
Requirement already satisfied: notebook in /usr/local/lib/python3.7/dist-packages (5.
Requirement already satisfied: vega in /usr/local/lib/python3.7/dist-packages (3.6.0)
Requirement already satisfied: toolz in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: entrypoints in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: pandas>=0.18 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /usr/

Requirement already satisfied: importlib-resources>=1.4.0 in /usr/local/lib/python3.7
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-pa
```

```
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: ipykernel in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: nbformat in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: terminado>=0.8.1 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: nbconvert in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: jupyter-client>=5.2.0 in /usr/local/lib/python3.7/dist
Requirement already satisfied: tornado>=4 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: jupyter-core>=4.4.0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: traitlets>=4.2.1 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: jupyter<2.0.0,>=1.0.0 in /usr/local/lib/python3.7/dist
Requirement already satisfied: qtconsole in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: jupyter-console in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: ipython>=5.0.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/lib/python3
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: jedi>=0.10 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: backcall in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.7/
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/lib/python3.7/
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: bleach in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: defusedxml in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: testpath in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: webencodings in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: qtpy>=2.0.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/d
```

00,01-Data Creation,02-Exploratory Data Analysis

```
df_path=""
df_path2 = df_path + 'News_dataset.csv'
df = pd.read_csv(df_path2, sep=';')
df.head()
```

	File_Name	Content	Category	Complete_Filename
0	001.txt	Ad sales boost Time Warner profit\r\n\r\nQuart...	business	001.txt-business
1	002.txt	Dollar gains on Greenspan speech\r\n\r\nThe do...	business	002.txt-business
2	003.txt	Yukos unit buyer faces loan claim\r\n\r\nThe o...	business	003.txt-business
3	004.txt	High fuel prices hit BA's profits\r\n\r\nBriti...	business	004.txt-business
4	005.txt	Pernod takeover talk lifts Domecq\r\n\r\nShare...	business	005.txt-business

Number of articles in each category

```
bars = alt.Chart(df).mark_bar(size=50).encode(
    x=alt.X("Category"),
    y=alt.Y("count():Q", axis=alt.Axis(title='Number of articles')),
    tooltip=[alt.Tooltip('count()', title='Number of articles'), 'Category'],
    color='Category'

)

text = bars.mark_text(
    align='center',
    baseline='bottom',
).encode(
    text='count()'
)

(bars + text).interactive().properties(
    height=300,
    width=700,
    title = "Number of articles in each category",
)
```

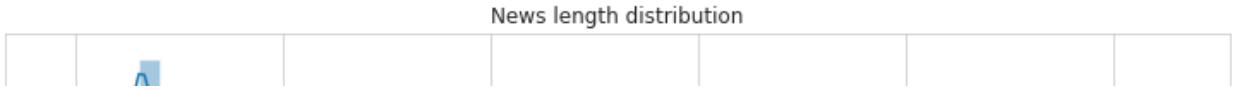
% of articles in each category

```
df['id'] = 1
df2 = pd.DataFrame(df.groupby('Category').count()['id']).reset_index()

bars = alt.Chart(df2).mark_bar(size=50).encode(
    x=alt.X('Category'),
    y=alt.Y('PercentOfTotal:Q', axis=alt.Axis(format='.0%', title='% of Articles')),
    color='Category'
).transform_window(
```

```
        TotalArticles='sum(id)',
        frame=[None, None]
    ).transform_calculate(
        PercentOfTotal="datum.id / datum.TotalArticles"
    )

    text = bars.mark_text(
        align='center',
        baseline='bottom',
        #dx=5  # Nudges text to right so it doesn't appear on top of the bar
    ).encode(
        text=alt.Text('PercentOfTotal:Q', format='.1%')
    )

    (bars + text).interactive().properties(
        height=300,
        width=700,
        title = "% of articles in each category",
    )
```

News length by category

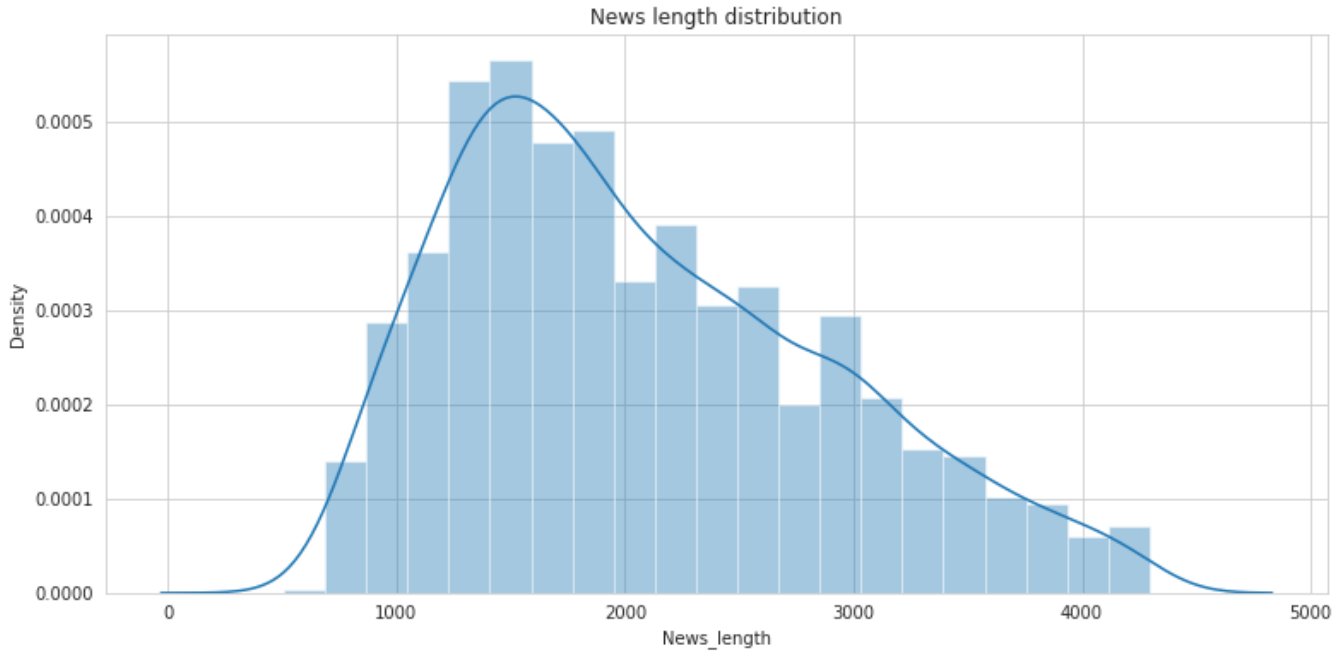
```
df['News_length'] = df['Content'].str.len()

plt.figure(figsize=(12.8,6))
sns.distplot(df['News_length']).set_title('News length distribution');
```




```
df['News_length'].describe()
```

```
count     2225.000000
mean      2274.363596
std       1370.782663
min        506.000000
25%       1454.000000
50%       1978.000000
75%       2814.000000
max      25596.000000
Name: News_length, dtype: float64
```

```
quantile_95 = df['News_length'].quantile(0.95)
df_95 = df[df['News_length'] < quantile_95]
plt.figure(figsize=(12.8,6))
sns.distplot(df_95['News_length']).set_title('News length distribution');
```



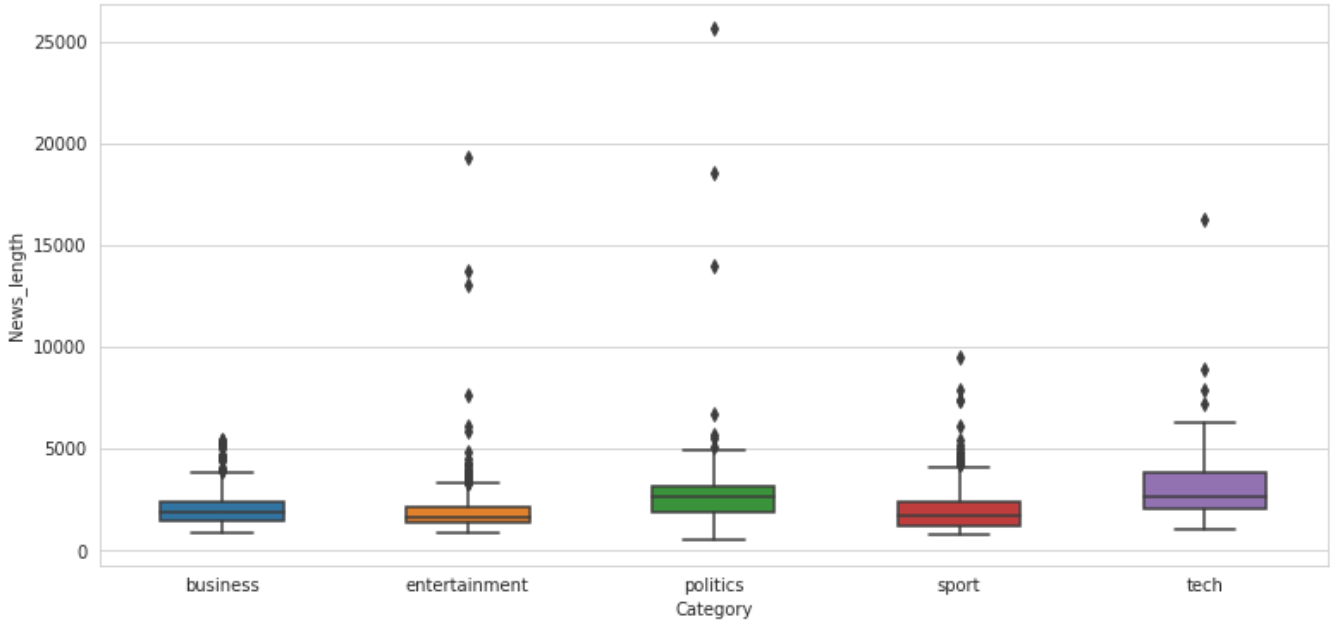

```
df_more10k = df[df['News_length'] > 10000]
len(df_more10k)
```

```
7
```

```
df_more10k['Content'].iloc[0]
```

```
'Scissor Sisters triumph at Brits\r\n\r\nUS band Scissor Sisters led the winners at the
UK music industry\'s Brit Awards, walking off with three prizes. The flamboyant act sco
red a hat-trick in the international categories, winning the best group, best album and
best newcomer awards. Glasgow group Franz Ferdinand won two prizes, as did Keane and Jo
ss Stone, who was voted best urban act by digital TV viewers. Robbie Williams\' Angels
was named the best song of the past 25 years. Scissor Sisters frontwoman Ana Matronic c
```

```
plt.figure(figsize=(12.8,6))
sns.boxplot(data=df, x='Category', y='News_length', width=.5);
```




```
plt.figure(figsize=(12.8,6))
sns.boxplot(data=df_95, x='Category', y='News_length');
```


```
with open('News_dataset.pickle', 'wb') as output:
    pickle.dump(df, output)
```

03 - Feature Engineering

```
import pickle
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import chi2
import numpy as np


#path_df = "/home/lnc/0. Latest News Classifier/02. Exploratory Data Analysis/News_dataset.pi
path_df = "News_dataset.pickle"

with open(path_df, 'rb') as data:
    df = pickle.load(data)

df.head()
```

	File_Name	Content	Category	Complete_Filename	id	News_length
0	001.txt	Ad sales boost Time Warner profit\r\n\r\nQuart...	business	001.txt-business	1	2569
1	002.txt	Dollar gains on Greenspan speech\r\n\r\nThe do...	business	002.txt-business	1	2257
2	003.txt	Yukos unit buyer faces loan claim\r\n\r\nThe o...	business	003.txt-business	1	1557
		High fuel prices hit BA's				

```
df.loc[1]['Content']
```

```
'Dollar gains on Greenspan speech\r\n\r\nThe dollar has hit its highest level against t
he euro in almost three months after the Federal Reserve head said the US trade deficit
is set to stabilise.\r\n\r\nAnd Alan Greenspan highlighted the US government\'s willing
ness to curb spending and rising household savings as factors which may help to reduce
it. In late trading in New York, the dollar reached $1.2871 against the euro, from $1.2
974 on Thursday. Market concerns about the deficit has hit the greenback in recent mont
hs. On Friday, Federal Reserve chairman Mr Greenspan\'s speech in London ahead of the m
eeting of G7 finance ministers sent the dollar higher after it had earlier tumbled on t
```

Text cleaning and preparation

Special character cleaning

```
# \r and \n
df['Content_Parsed_1'] = df['Content'].str.replace("\r", " ")
df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace("\n", " ")
df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace("    ", " ")

text = "Mr Greenspan\'s"
print(text)

# " when quoting text
df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace('"', '')
```

```
     Mr Greenspan's
```

Upcase/downcase

```
# Lowercasing the text
df['Content_Parsed_2'] = df['Content_Parsed_1'].str.lower()
```

Punctuation signs

```
punctuation_signs = list("?:!.,;")
df['Content_Parsed_3'] = df['Content_Parsed_2']

for punct_sign in punctuation_signs:
    df['Content_Parsed_3'] = df['Content_Parsed_3'].str.replace(punct_sign, '')
```

Possessive pronouns

```
df['Content_Parsed_4'] = df['Content_Parsed_3'].str.replace("'s", "")
```

Stemming and Lemmatization

```
# Downloading punkt and wordnet from NLTK
nltk.download('punkt')
```

```
print("------------------------------------------------------------")
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
# Saving the lemmatizer into an object
wordnet_lemmatizer = WordNetLemmatizer()

nltk.download('omw-1.4')

nrows = len(df)
lemmatized_text_list = []

for row in range(0, nrows):

    # Create an empty list containing lemmatized words
    lemmatized_list = []

    # Save the text and its words into an object
    text = df.loc[row]['Content_Parsed_4']
    text_words = text.split(" ")

    # Iterate through every word to lemmatize
    for word in text_words:
        lemmatized_list.append(wordnet_lemmatizer.lemmatize(word, pos="v"))

    # Join the list
    lemmatized_text = " ".join(lemmatized_list)

    # Append to the list containing the texts
    lemmatized_text_list.append(lemmatized_text)

df['Content_Parsed_5'] = lemmatized_text_list
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

Stop words

```
# Downloading the stop words list
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
```

```
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
# Loading the stop words in english
stop_words = list(stopwords.words('english'))
stop_words[0:10]
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're"]
```

```
example = "me eating a meal"
word = "me"

# The regular expression is:
regex = r"\b" + word + r"\b"  # we need to build it like that to work properly

re.sub(regex, "StopWord", example)
```

```
'StopWord eating a meal'
```

```
df['Content_Parsed_6'] = df['Content_Parsed_5']

for stop_word in stop_words:

    regex_stopword = r"\b" + stop_word + r"\b"
    df['Content_Parsed_6'] = df['Content_Parsed_6'].str.replace(regex_stopword, '')
```

Results of parsing

```
df.loc[5]['Content']
```

```
'Japan narrowly escapes recession\r\n\r\nJapan\'s economy teetered on the brink of a te
chnical recession in the three months to September, figures show.\r\n\r\nRevised figure
s indicated growth of just 0.1% - and a similar-sized contraction in the previous quart
er. On an annual basis, the data suggests annual growth of just 0.2%, suggesting a much
more hesitant recovery than had previously been thought. A common technical definition
of a recession is two successive quarters of negative growth.\r\n\r\nThe government was
keen to play down the worrying implications of the data. "I maintain the view that Japa
n\'s economy remains in a minor adjustment phase in an upward climb, and we will monito
```

```
df.loc[5]['Content_Parsed_1']
```

```
'Japan narrowly escapes recession Japan's economy teetered on the brink of a technical
recession in the three months to September, figures show. Revised figures indicated gro
wth of just 0.1% - and a similar-sized contraction in the previous quarter. On an annua
l basis, the data suggests annual growth of just 0.2%, suggesting a much more hesitant
recovery than had previously been thought. A common technical definition of a recession
is two successive quarters of negative growth. The government was keen to play down the
worrying implications of the data. I maintain the view that Japan's economy remains in
a minor adjustment phase in an upward climb, and we will monitor developments carefull
```

```
df.loc[5]['Content_Parsed_2']
```

```
'japan narrowly escapes recession japan's economy teetered on the brink of a technical
recession in the three months to september, figures show. revised figures indicated gro
wth of just 0.1% - and a similar-sized contraction in the previous quarter. on an annua
l basis, the data suggests annual growth of just 0.2%, suggesting a much more hesitant
recovery than had previously been thought. a common technical definition of a recession
is two successive quarters of negative growth. the government was keen to play down the
worrying implications of the data. i maintain the view that japan's economy remains in
a minor adjustment phase in an upward climb, and we will monitor developments carefull
```

```
df.loc[5]['Content_Parsed_3']
```

```
'japan narrowly escapes recession japan's economy teetered on the brink of a technical
recession in the three months to september figures show revised figures indicated growt
h of just 01% - and a similar-sized contraction in the previous quarter on an annual ba
sis the data suggests annual growth of just 02% suggesting a much more hesitant recover
y than had previously been thought a common technical definition of a recession is two
successive quarters of negative growth the government was keen to play down the worryin
g implications of the data i maintain the view that japan's economy remains in a minor
adjustment phase in an upward climb and we will monitor developments carefully said eco
```

```
df.loc[5]['Content_Parsed_4']
```

```
'japan narrowly escapes recession japan economy teetered on the brink of a technical re
cession in the three months to september figures show revised figures indicated growth
of just 01% - and a similar-sized contraction in the previous quarter on an annual basi
s the data suggests annual growth of just 02% suggesting a much more hesitant recovery
than had previously been thought a common technical definition of a recession is two su
ccessive quarters of negative growth the government was keen to play down the worrying
implications of the data i maintain the view that japan economy remains in a minor adju
stment phase in an upward climb and we will monitor developments carefully said economy
```

```
df.loc[5]['Content_Parsed_5']
```

```
'japan narrowly escape recession japan economy teeter on the brink of a technical reces
sion in the three months to september figure show revise figure indicate growth of just
01% - and a similar-sized contraction in the previous quarter on an annual basis the da
ta suggest annual growth of just 02% suggest a much more hesitant recovery than have pr
eviously be think a common technical definition of a recession be two successive quarte
r of negative growth the government be keen to play down the worry implications of the
data i maintain the view that japan economy remain in a minor adjustment phase in an up
ward climb and we will monitor developments carefully say economy minister heizo takena
```

```
df.loc[5]['Content_Parsed_6']
```

```
'japan narrowly escape recession japan economy teeter   brink   technical recession   t
hree months  september figure show revise figure indicate growth   01% -   similar-size
d contraction   previous quarter   annual basis  data suggest annual growth   02% sugge
st  much  hesitant recovery   previously  think  common technical definition   recessio
n  two successive quarter  negative growth  government  keen  play   worry implications
data  maintain  view  japan economy remain   minor adjustment phase   upward climb    m
onitor developments carefully say economy minister heizo takenaka    face   strengthen
```

```
df.head(1)
```

	File_Name	Content	Category	Complete_Filename	id	News_length	Content_Pa
0	001.txt	Ad sales boost Time Warner profit\r\n\r\nQuart...	business	001.txt-business	1	2569	Ad sale Time Warn Quart

```
list_columns = ["File_Name", "Category", "Complete_Filename", "Content", "Content_Parsed_6"]
df = df[list_columns]

df = df.rename(columns={'Content_Parsed_6': 'Content_Parsed'})

df.head()
```

	File_Name	Category	Complete_Filename	Content	Content_Parsed
0	001.txt	business	001.txt-business	Ad sales boost Time Warner profit\r\n\r\nQuart...	ad sales boost time warner profit quarterly pr...
1	002.txt	business	002.txt-business	Dollar gains on Greenspan speech\r\n\r\nThe do...	dollar gain greenspan speech dollar hit hi...
2	003.txt	business	003.txt-business	Yukos unit buyer faces	yukos unit buyer face loan claim owners

Label Coding

```
category_codes = {
    'business': 0,
    'entertainment': 1,
    'politics': 2,
    'sport': 3,
    'tech': 4
}

# Category mapping
df['Category_Code'] = df['Category']
df = df.replace({'Category_Code':category_codes})

df.head()
```

	File_Name	Category	Complete_Filename	Content	Content_Parsed	Category
0	001.txt	business	001.txt-business	Ad sales boost Time Warner profit\r\n\r\nQuart...	ad sales boost time warner profit quarterly pr...	
				Dollar gains on	dollar gain	

Train - test split

```
X_train, X_test, y_train, y_test = train_test_split(df['Content_Parsed'],
                                                    df['Category_Code'],
                                                    test_size=0.15,
                                                    random_state=8)
```

Text Representation

```
# Parameter election
ngram_range = (1,2)
min_df = 10
max_df = 1.
max_features = 300


tfidf = TfidfVectorizer(encoding='utf-8',
                        ngram_range=ngram_range,
                        stop_words=None,
                        lowercase=False,
                        max_df=max_df,
                        min_df=min_df,
                        max_features=max_features,
                        norm='l2',
                        sublinear_tf=True)

features_train = tfidf.fit_transform(X_train).toarray()
labels_train = y_train
print(features_train.shape)

features_test = tfidf.transform(X_test).toarray()
labels_test = y_test
print(features_test.shape)
```

```
(1891, 300)
(334, 300)
```

```
from sklearn.feature_selection import chi2
```

```
import numpy as np

for Product, category_id in sorted(category_codes.items()):
    features_chi2 = chi2(features_train, labels_train == category_id)
    indices = np.argsort(features_chi2[0])
    feature_names = np.array(tfidf.get_feature_names())[indices]
    unigrams = [v for v in feature_names if len(v.split(' ')) == 1]
    bigrams = [v for v in feature_names if len(v.split(' ')) == 2]
    print("# '{}' category:".format(Product))
    print("  . Most correlated unigrams:\n. {}".format('\n. '.join(unigrams[-5:])))
    print("  . Most correlated bigrams:\n. {}".format('\n. '.join(bigrams[-2:])))
    print("")
```

```
 # 'business' category:
   . Most correlated unigrams:
. market
. price
. economy
. growth
. bank
   . Most correlated bigrams:
. last year
. year old

 # 'entertainment' category:
   . Most correlated unigrams:
. tv
. music
. star
. award
. film
   . Most correlated bigrams:
. mr blair
. prime minister

 # 'politics' category:
   . Most correlated unigrams:
. minister
. blair
. party
. election
. labour
   . Most correlated bigrams:
. prime minister
. mr blair

 # 'sport' category:
   . Most correlated unigrams:
. win
. side
. game
. team
. match
   . Most correlated bigrams:
. say mr
```

```
      . year old

      # 'tech' category:
        . Most correlated unigrams:
      . digital
      . technology
      . computer
      . software
      . users
        . Most correlated bigrams:
      . year old
      . say mr
```

```
bigrams
```

```
['tell bbc', 'last year', 'prime minister', 'mr blair', 'year old', 'say mr']
```

```
# X_train
with open('Pickles/X_train.pickle', 'wb') as output:
    pickle.dump(X_train, output)

# X_test
with open('Pickles/X_test.pickle', 'wb') as output:
    pickle.dump(X_test, output)

# y_train
with open('Pickles/y_train.pickle', 'wb') as output:
    pickle.dump(y_train, output)

# y_test
with open('Pickles/y_test.pickle', 'wb') as output:
    pickle.dump(y_test, output)

# df
with open('Pickles/df.pickle', 'wb') as output:
    pickle.dump(df, output)

# features_train
with open('Pickles/features_train.pickle', 'wb') as output:
    pickle.dump(features_train, output)

# labels_train
with open('Pickles/labels_train.pickle', 'wb') as output:
    pickle.dump(labels_train, output)

# features_test
with open('Pickles/features_test.pickle', 'wb') as output:
    pickle.dump(features_test, output)

# labels_test
```

```
with open('Pickles/labels_test.pickle', 'wb') as output:
    pickle.dump(labels_test, output)

# TF-IDF object
with open('Pickles/tfidf.pickle', 'wb') as output:
    pickle.dump(tfidf, output)
```

04. Model Training

```
import pickle
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from pprint import pprint
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.model_selection import ShuffleSplit
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd


# Dataframe
path_df = "Pickles/df.pickle"
with open(path_df, 'rb') as data:
    df = pickle.load(data)

# features_train
path_features_train = "Pickles/features_train.pickle"
with open(path_features_train, 'rb') as data:
    features_train = pickle.load(data)

# labels_train
path_labels_train = "Pickles/labels_train.pickle"
with open(path_labels_train, 'rb') as data:
    labels_train = pickle.load(data)

# features_test
path_features_test = "Pickles/features_test.pickle"
with open(path_features_test, 'rb') as data:
    features_test = pickle.load(data)

# labels_test
path_labels_test = "Pickles/labels_test.pickle"
with open(path_labels_test, 'rb') as data:
    labels_test = pickle.load(data)
```

```
print(features_train.shape)
print(features_test.shape)
```

```
(1891, 300)
(334, 300)
```

Random Forest

```
rf_0 = RandomForestClassifier(random_state = 8)

print('Parameters currently in use:\n')
pprint(rf_0.get_params())
```

```
Parameters currently in use:

{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': 8,
 'verbose': 0,
 'warm_start': False}
```

```
# n_estimators
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 1000, num = 5)]

# max_features
max_features = ['auto', 'sqrt']

# max_depth
max_depth = [int(x) for x in np.linspace(20, 100, num = 5)]
max_depth.append(None)

# min_samples_split
min_samples_split = [2, 5, 10]

# min_samples_leaf
```

```
min_samples_leaf = [1, 2, 4]

# bootstrap
bootstrap = [True, False]

# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}

pprint(random_grid)
```

```
     {'bootstrap': [True, False],
      'max_depth': [20, 40, 60, 80, 100, None],
      'max_features': ['auto', 'sqrt'],
      'min_samples_leaf': [1, 2, 4],
      'min_samples_split': [2, 5, 10],
      'n_estimators': [200, 400, 600, 800, 1000]}
```

```
# First create the base model to tune
rfc = RandomForestClassifier(random_state=8)

# Definition of the random search
random_search = RandomizedSearchCV(estimator=rfc,
                                   param_distributions=random_grid,
                                   n_iter=50,
                                   scoring='accuracy',
                                   cv=3,
                                   verbose=1,
                                   random_state=8)

# Fit the random search model
random_search.fit(features_train, labels_train)
```

```
     Fitting 3 folds for each of 50 candidates, totalling 150 fits
     RandomizedSearchCV(cv=3, estimator=RandomForestClassifier(random_state=8),
                        n_iter=50,
                        param_distributions={'bootstrap': [True, False],
                                             'max_depth': [20, 40, 60, 80, 100,
                                                           None],
                                             'max_features': ['auto', 'sqrt'],
                                             'min_samples_leaf': [1, 2, 4],
                                             'min_samples_split': [2, 5, 10],
                                             'n_estimators': [200, 400, 600, 800,
                                                              1000]},
                        random_state=8, scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Random Search are:")
print(random_search.best_params_)
```

```
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(random_search.best_score_)
```

```
The best hyperparameters from Random Search are:
{'n_estimators': 600, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_features': 's

The mean accuracy of a model with these hyperparameters is:
0.9434181068095491
```

```
# Create the parameter grid based on the results of random search
bootstrap = [False]
max_depth = [30, 40, 50]
max_features = ['sqrt']
min_samples_leaf = [1, 2, 4]
min_samples_split = [5, 10, 15]
n_estimators = [800]

param_grid = {
    'bootstrap': bootstrap,
    'max_depth': max_depth,
    'max_features': max_features,
    'min_samples_leaf': min_samples_leaf,
    'min_samples_split': min_samples_split,
    'n_estimators': n_estimators
}

# Create a base model
rfc = RandomForestClassifier(random_state=8)

# Manually create the splits in CV in order to be able to fix a random_state (GridSearchCV do
cv_sets = ShuffleSplit(n_splits = 3, test_size = .33, random_state = 8)

# Instantiate the grid search model
grid_search = GridSearchCV(estimator=rfc,
                           param_grid=param_grid,
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

# Fit the grid search to the data
grid_search.fit(features_train, labels_train)
```

```
Fitting 3 folds for each of 27 candidates, totalling 81 fits
GridSearchCV(cv=ShuffleSplit(n_splits=3, random_state=8, test_size=0.33,
train_size=None),
             estimator=RandomForestClassifier(random_state=8),
             param_grid={'bootstrap': [False], 'max_depth': [30, 40, 50],
                         'max_features': ['sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [5, 10, 15],
```

```
                               'n_estimators': [800]},
                   scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Grid Search are:")
print(grid_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(grid_search.best_score_)
```

```
The best hyperparameters from Grid Search are:
{'bootstrap': False, 'max_depth': 40, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'mi

The mean accuracy of a model with these hyperparameters is:
0.9450666666666668
```

```
best_rfc = grid_search.best_estimator_
best_rfc
```

```
RandomForestClassifier(bootstrap=False, max_depth=40, max_features='sqrt',
                       min_samples_split=5, n_estimators=800, random_state=8)
```

```
best_rfc.fit(features_train, labels_train)
```

```
RandomForestClassifier(bootstrap=False, max_depth=40, max_features='sqrt',
                       min_samples_split=5, n_estimators=800, random_state=8)
```

```
rfc_pred = best_rfc.predict(features_test)
```

```
# Training accuracy
print("The training accuracy is: ")
print(accuracy_score(labels_train, best_rfc.predict(features_train)))
```

```
The training accuracy is:
1.0
```

```
# Test accuracy
print("The test accuracy is: ")
print(accuracy_score(labels_test, rfc_pred))
```

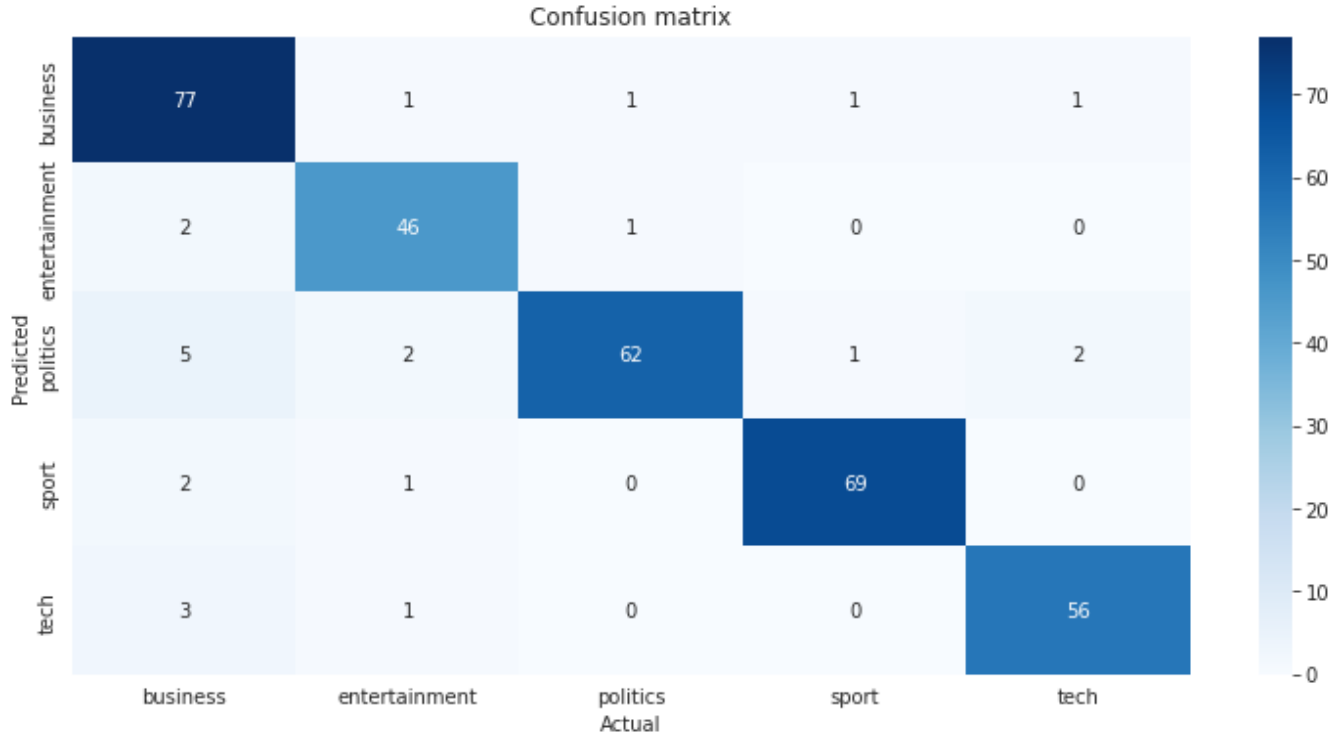
```
The test accuracy is:
0.9281437125748503
```

```
# Classification report
print("Classification report")
print(classification_report(labels_test,rfc_pred))
```

```
Classification report
```

	precision	recall	f1-score	support
0	0.87	0.95	0.91	81
1	0.90	0.94	0.92	49
2	0.97	0.86	0.91	72
3	0.97	0.96	0.97	72
4	0.95	0.93	0.94	60
accuracy			0.93	334
macro avg	0.93	0.93	0.93	334
weighted avg	0.93	0.93	0.93	334

```
aux_df = df[['Category', 'Category_Code']].drop_duplicates().sort_values('Category_Code')
conf_matrix = confusion_matrix(labels_test, rfc_pred)
plt.figure(figsize=(12.8,6))
sns.heatmap(conf_matrix,
            annot=True,
            xticklabels=aux_df['Category'].values,
            yticklabels=aux_df['Category'].values,
            cmap="Blues")
plt.ylabel('Predicted')
plt.xlabel('Actual')
plt.title('Confusion matrix')
plt.show()
```




```
base_model = RandomForestClassifier(random_state = 8)
base_model.fit(features_train, labels_train)
accuracy_score(labels_test, base_model.predict(features_test))
```

```
0.9281437125748503
```

```
best_rfc.fit(features_train, labels_train)
accuracy_score(labels_test, best_rfc.predict(features_test))
```

```
0.9281437125748503
```

```
d = {
    'Model': 'Random Forest',
    'Training Set Accuracy': accuracy_score(labels_train, best_rfc.predict(features_train)),
    'Test Set Accuracy': accuracy_score(labels_test, rfc_pred)
}

df_models_rfc = pd.DataFrame(d, index=[0])
```

```
df_models_rfc
```

	Model	Training Set Accuracy	Test Set Accuracy
0	Random Forest	1.0	0.928144

Support Vector Machine

```
from sklearn import svm
svc_0 =svm.SVC(random_state=8)

print('Parameters currently in use:\n')
pprint(svc_0.get_params())
```

```
Parameters currently in use:

{'C': 1.0,
 'break_ties': False,
 'cache_size': 200,
 'class_weight': None,
 'coef0': 0.0,
 'decision_function_shape': 'ovr',
 'degree': 3,
 'gamma': 'scale',
 'kernel': 'rbf',
 'max_iter': -1,
 'probability': False,
 'random_state': 8,
 'shrinking': True,
 'tol': 0.001,
 'verbose': False}
```

```
# C
C = [.0001, .001, .01]

# gamma
gamma = [.0001, .001, .01, .1, 1, 10, 100]

# degree
degree = [1, 2, 3, 4, 5]

# kernel
kernel = ['linear', 'rbf', 'poly']

# probability
probability = [True]

# Create the random grid
random_grid = {'C': C,
               'kernel': kernel,
               'gamma': gamma,
               'degree': degree,
               'probability': probability
              }

pprint(random_grid)
```

```
     {'C': [0.0001, 0.001, 0.01],
      'degree': [1, 2, 3, 4, 5],
      'gamma': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100],
      'kernel': ['linear', 'rbf', 'poly'],
      'probability': [True]}
```

```
# First create the base model to tune
svc = svm.SVC(random_state=8)

# Definition of the random search
random_search = RandomizedSearchCV(estimator=svc,
                                   param_distributions=random_grid,
                                   n_iter=50,
                                   scoring='accuracy',
                                   cv=3,
                                   verbose=1,
                                   random_state=8)

# Fit the random search model
random_search.fit(features_train, labels_train)
```

```
     Fitting 3 folds for each of 50 candidates, totalling 150 fits
     RandomizedSearchCV(cv=3, estimator=SVC(random_state=8), n_iter=50,
                        param_distributions={'C': [0.0001, 0.001, 0.01],
                                             'degree': [1, 2, 3, 4, 5],
                                             'gamma': [0.0001, 0.001, 0.01, 0.1, 1,
```

```
                                                  10, 100],
                                        'kernel': ['linear', 'rbf', 'poly'],
                                        'probability': [True]},
                           random_state=8, scoring='accuracy', verbose=1)

print("The best hyperparameters from Random Search are:")
print(random_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(random_search.best_score_)
```

```
The best hyperparameters from Random Search are:
{'probability': True, 'kernel': 'poly', 'gamma': 10, 'degree': 4, 'C': 0.01}

The mean accuracy of a model with these hyperparameters is:
0.9217358857612424
```

```
# Create the parameter grid based on the results of random search
C = [.0001, .001, .01, .1]
degree = [3, 4, 5]
gamma = [1, 10, 100]
probability = [True]

param_grid = [
  {'C': C, 'kernel':['linear'], 'probability':probability},
  {'C': C, 'kernel':['poly'], 'degree':degree, 'probability':probability},
  {'C': C, 'kernel':['rbf'], 'gamma':gamma, 'probability':probability}
]

# Create a base model
svc = svm.SVC(random_state=8)

# Manually create the splits in CV in order to be able to fix a random_state (GridSearchCV do
cv_sets = ShuffleSplit(n_splits = 3, test_size = .33, random_state = 8)

# Instantiate the grid search model
grid_search = GridSearchCV(estimator=svc,
                           param_grid=param_grid,
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

# Fit the grid search to the data
grid_search.fit(features_train, labels_train)
```

```
Fitting 3 folds for each of 28 candidates, totalling 84 fits
GridSearchCV(cv=ShuffleSplit(n_splits=3, random_state=8, test_size=0.33,
train_size=None),
             estimator=SVC(random_state=8),
             param_grid=[{'C': [0.0001, 0.001, 0.01, 0.1], 'kernel': ['linear'],
                          'probability': [True]},
                         {'C': [0.0001, 0.001, 0.01, 0.1], 'degree': [3, 4, 5],
```

```
                                 'kernel': ['poly'], 'probability': [True]},
                                {'C': [0.0001, 0.001, 0.01, 0.1],
                                 'gamma': [1, 10, 100], 'kernel': ['rbf'],
                                 'probability': [True]}],
                    scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Grid Search are:")
print(grid_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(grid_search.best_score_)
```

```
The best hyperparameters from Grid Search are:
{'C': 0.1, 'kernel': 'linear', 'probability': True}

The mean accuracy of a model with these hyperparameters is:
0.9498666666666665
```

```
best_svc = grid_search.best_estimator_
best_svc
```

```
SVC(C=0.1, kernel='linear', probability=True, random_state=8)
```

```
best_svc.fit(features_train, labels_train)
```

```
SVC(C=0.1, kernel='linear', probability=True, random_state=8)
```

```
svc_pred = best_svc.predict(features_test)
```

```
# Training accuracy
print("The training accuracy is: ")
print(accuracy_score(labels_train, best_svc.predict(features_train)))
```

```
The training accuracy is:
0.9592808038075092
```

```
# Test accuracy
print("The test accuracy is: ")
print(accuracy_score(labels_test, svc_pred))
```

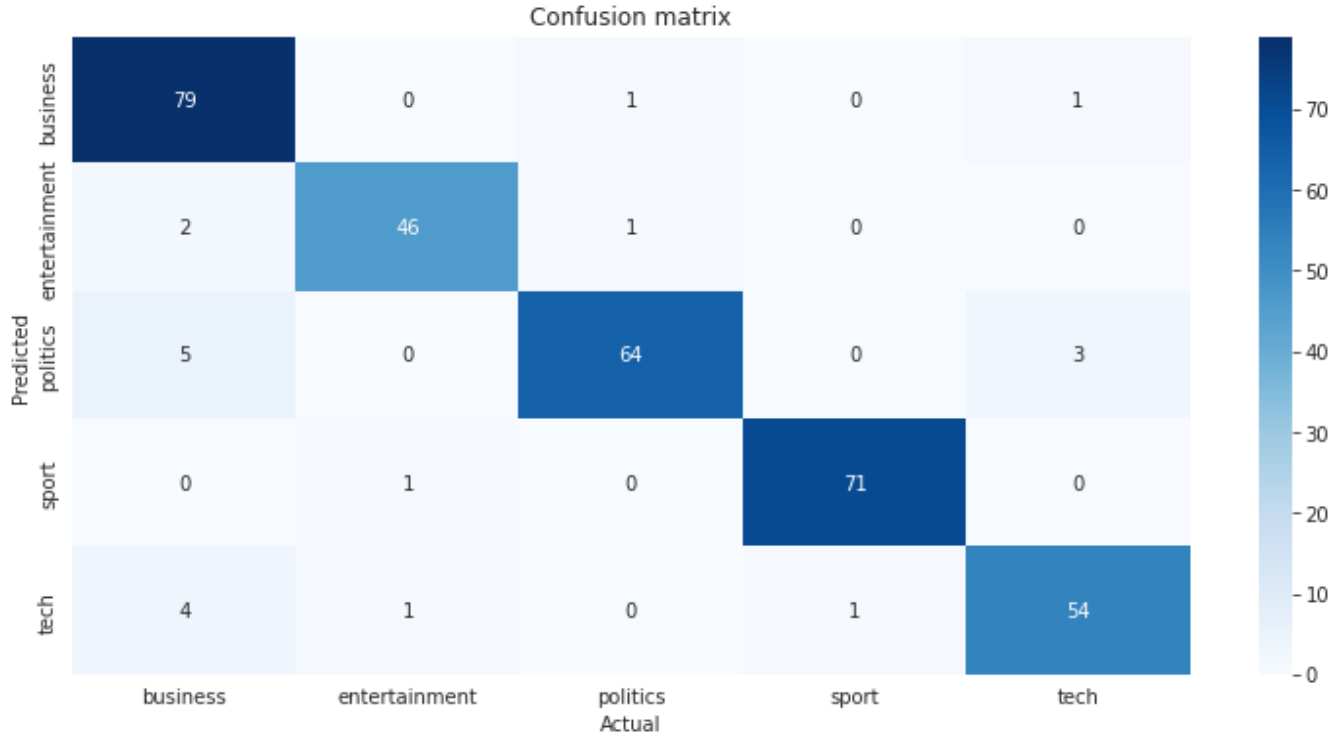
```
The test accuracy is:
0.9401197604790419
```

```
# Classification report
print("Classification report")
print(classification_report(labels_test,svc_pred))
```

```
Classification report
```

```
                precision    recall  f1-score   support

             0       0.88      0.98      0.92        81
             1       0.96      0.94      0.95        49
             2       0.97      0.89      0.93        72
             3       0.99      0.99      0.99        72
             4       0.93      0.90      0.92        60

      accuracy                           0.94       334
     macro avg       0.94      0.94      0.94       334
  weighted avg       0.94      0.94      0.94       334
```

```
aux_df = df[['Category', 'Category_Code']].drop_duplicates().sort_values('Category_Code')
conf_matrix = confusion_matrix(labels_test, svc_pred)
plt.figure(figsize=(12.8,6))
sns.heatmap(conf_matrix,
            annot=True,
            xticklabels=aux_df['Category'].values,
            yticklabels=aux_df['Category'].values,
            cmap="Blues")
plt.ylabel('Predicted')
plt.xlabel('Actual')
plt.title('Confusion matrix')
plt.show()
```




```
base_model = svm.SVC(random_state = 8)
base_model.fit(features_train, labels_train)
accuracy_score(labels_test, base_model.predict(features_test))
```

```
0.9550898203592815
```

```
best_svc.fit(features_train, labels_train)
accuracy_score(labels_test, best_svc.predict(features_test))
```

```
0.9401197604790419
```

```
d = {
    'Model': 'SVM',
    'Training Set Accuracy': accuracy_score(labels_train, best_svc.predict(features_train)),
    'Test Set Accuracy': accuracy_score(labels_test, svc_pred)
}

df_models_svc = pd.DataFrame(d, index=[0])
df_models_svc
```

	Model	Training Set Accuracy	Test Set Accuracy
0	SVM	0.959281	0.94012

```
with open('Models/best_svc.pickle', 'wb') as output:
    pickle.dump(best_svc, output)

with open('Models/df_models_svc.pickle', 'wb') as output:
    pickle.dump(df_models_svc, output)
```

KNN

```
from sklearn.neighbors import KNeighborsClassifier
knnc_0 =KNeighborsClassifier()

print('Parameters currently in use:\n')
pprint(knnc_0.get_params())
```

```
Parameters currently in use:

{'algorithm': 'auto',
 'leaf_size': 30,
 'metric': 'minkowski',
 'metric_params': None,
 'n_jobs': None,
 'n_neighbors': 5,
 'p': 2,
 'weights': 'uniform'}
```

```
# Create the parameter grid
```

```
n_neighbors = [int(x) for x in np.linspace(start = 1, stop = 500, num = 100)]

param_grid = {'n_neighbors': n_neighbors}

# Create a base model
knnc = KNeighborsClassifier()

# Manually create the splits in CV in order to be able to fix a random_state (GridSearchCV do
cv_sets = ShuffleSplit(n_splits = 3, test_size = .33, random_state = 8)

# Instantiate the grid search model
grid_search = GridSearchCV(estimator=knnc,
                           param_grid=param_grid,
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

# Fit the grid search to the data
grid_search.fit(features_train, labels_train)
```

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
GridSearchCV(cv=ShuffleSplit(n_splits=3, random_state=8, test_size=0.33,
train_size=None),
             estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': [1, 6, 11, 16, 21, 26, 31, 36, 41, 46,
                                         51, 56, 61, 66, 71, 76, 81, 86, 91, 96,
                                         101, 106, 111, 116, 121, 127, 132, 137,
                                         142, 147, ...]},
             scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Grid Search are:")
print(grid_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(grid_search.best_score_)
```

```
The best hyperparameters from Grid Search are:
{'n_neighbors': 6}

The mean accuracy of a model with these hyperparameters is:
0.9477333333333333
```

```
n_neighbors = [1,2,3,4,5,6,7,8,9,10,11]
param_grid = {'n_neighbors': n_neighbors}

knnc = KNeighborsClassifier()
cv_sets = ShuffleSplit(n_splits = 3, test_size = .33, random_state = 8)

grid_search = GridSearchCV(estimator=knnc,
                           param_grid=param_grid,
                           scoring='accuracy',
```

```
                            cv=cv_sets,
                            verbose=1)

grid_search.fit(features_train, labels_train)
```

```
Fitting 3 folds for each of 11 candidates, totalling 33 fits
GridSearchCV(cv=ShuffleSplit(n_splits=3, random_state=8, test_size=0.33,
train_size=None),
             estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]},
             scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Grid Search are:")
print(grid_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(grid_search.best_score_)
```

```
The best hyperparameters from Grid Search are:
{'n_neighbors': 6}

The mean accuracy of a model with these hyperparameters is:
0.9477333333333333
```

```
best_knnc = grid_search.best_estimator_
best_knnc
```

```
KNeighborsClassifier(n_neighbors=6)
```

```
best_knnc.fit(features_train, labels_train)
```

```
KNeighborsClassifier(n_neighbors=6)
```

```
knnc_pred = best_knnc.predict(features_test)
```

```
# Training accuracy
print("The training accuracy is: ")
print(accuracy_score(labels_train, best_knnc.predict(features_train)))
```

```
The training accuracy is:
0.9598096245372819
```

```
# Test accuracy
print("The test accuracy is: ")
print(accuracy_score(labels_test, knnc_pred))
```

```
The test accuracy is:
0.9281437125748503
```

```
# Classification report
print("Classification report")
print(classification_report(labels_test,knnc_pred))
```

```
     Classification report
                   precision    recall  f1-score   support

                0       0.91      0.95      0.93        81
                1       0.93      0.88      0.91        49
                2       0.97      0.92      0.94        72
                3       0.97      0.96      0.97        72
                4       0.86      0.92      0.89        60

         accuracy                           0.93       334
        macro avg       0.93      0.92      0.93       334
     weighted avg       0.93      0.93      0.93       334
```

```
aux_df = df[['Category', 'Category_Code']].drop_duplicates().sort_values('Category_Code')
conf_matrix = confusion_matrix(labels_test, knnc_pred)
plt.figure(figsize=(12.8,6))
sns.heatmap(conf_matrix,
            annot=True,
            xticklabels=aux_df['Category'].values,
            yticklabels=aux_df['Category'].values,
            cmap="Blues")
plt.ylabel('Predicted')
plt.xlabel('Actual')
plt.title('Confusion matrix')
plt.show()
```

```
base_model = KNeighborsClassifier()
base_model.fit(features_train, labels_train)
accuracy_score(labels_test, base_model.predict(features_test))
```

```
0.9341317365269461
```

```
best_knnc.fit(features_train, labels_train)
accuracy_score(labels_test, best_knnc.predict(features_test))
```

```
0.9281437125748503
```

```
d = {
    'Model': 'KNN',
    'Training Set Accuracy': accuracy_score(labels_train, best_knnc.predict(features_train))
    'Test Set Accuracy': accuracy_score(labels_test, knnc_pred)
}

df_models_knnc = pd.DataFrame(d, index=[0])


df_models_knnc
```

	Model	Training Set Accuracy	Test Set Accuracy
0	KNN	0.95981	0.928144

```
with open('Models/best_knnc.pickle', 'wb') as output:
    pickle.dump(best_knnc, output)

with open('Models/df_models_knnc.pickle', 'wb') as output:
    pickle.dump(df_models_knnc, output)
```

Model Selection

```
path_pickles = "Models/"

list_pickles = [
    "df_models_knnc.pickle",
    "df_models_rfc.pickle",
    "df_models_svc.pickle"
]

df_summary = pd.DataFrame()

for pickle_ in list_pickles:
```

```
    path = path_pickles + pickle_

    with open(path, 'rb') as data:
        df = pickle.load(data)

    df_summary = df_summary.append(df)

df_summary = df_summary.reset_index().drop('index', axis=1)
df_summary
```

	Model	Training Set Accuracy	Test Set Accuracy
0	KNN	0.959810	0.928144
1	Random Forest	1.000000	0.928144
2	SVM	0.959281	0.940120

```
df_summary.sort_values('Test Set Accuracy', ascending=False)
```

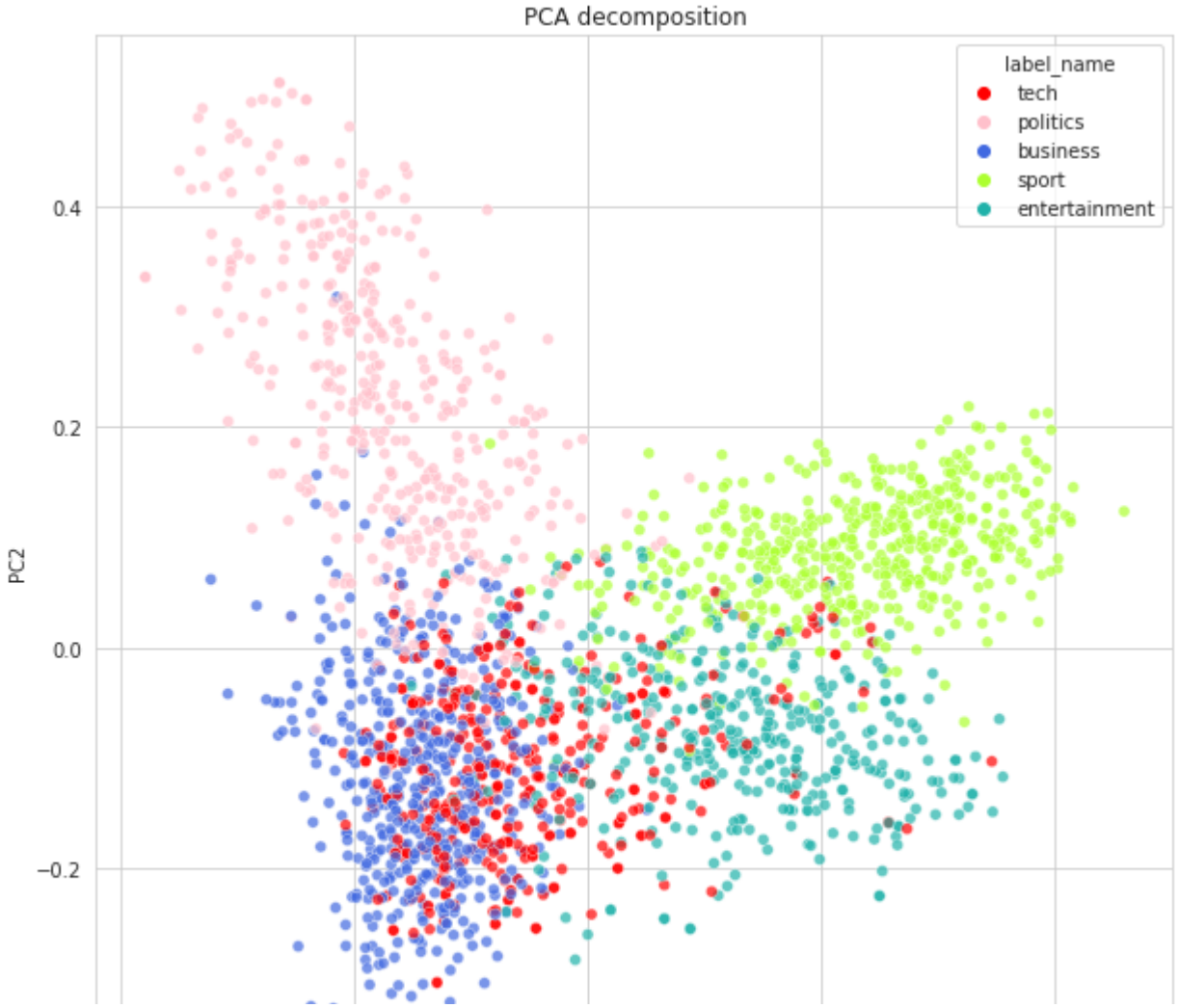
	Model	Training Set Accuracy	Test Set Accuracy
2	SVM	0.959281	0.940120
0	KNN	0.959810	0.928144
1	Random Forest	1.000000	0.928144

```
features = np.concatenate((features_train,features_test), axis=0)
labels = np.concatenate((labels_train,labels_test), axis=0)
print(features.shape)
print(labels.shape)
```

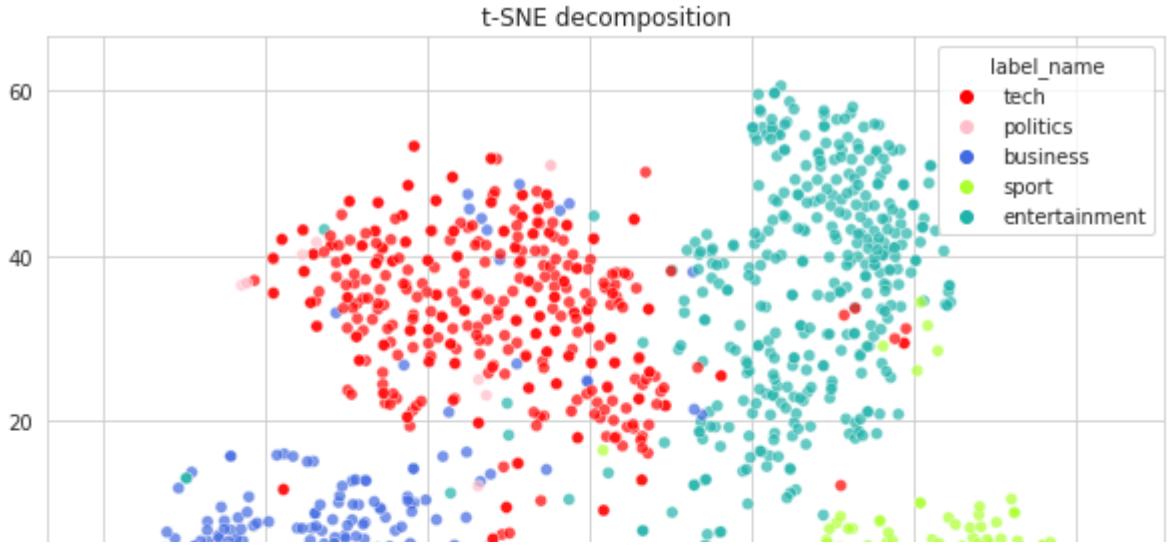
```
(2225, 300)
(2225,)
```

```
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE

def plot_dim_red(model, features, labels, n_components=2):

    # Creation of the model
    if (model == 'PCA'):
        mod = PCA(n_components=n_components)
        title = "PCA decomposition"  # for the plot

    elif (model == 'TSNE'):
        mod = TSNE(n_components=2)
        title = "t-SNE decomposition"
```

```
    else:
        return "Error"

    # Fit and transform the features
    principal_components = mod.fit_transform(features)

    # Put them into a dataframe
    df_features = pd.DataFrame(data=principal_components,
                    columns=['PC1', 'PC2'])

    # Now we have to paste each row's label and its meaning
    # Convert labels array to df
    df_labels = pd.DataFrame(data=labels,
                             columns=['label'])

    df_full = pd.concat([df_features, df_labels], axis=1)
    df_full['label'] = df_full['label'].astype(str)

    # Get labels name
    category_names = {
        "0": 'business',
        "1": 'entertainment',
        "2": 'politics',
        "3": 'sport',
        "4": 'tech'
    }

    # And map labels
    df_full['label_name'] = df_full['label']
    df_full = df_full.replace({'label_name':category_names})

    # Plot
    plt.figure(figsize=(10,10))
    sns.scatterplot(x='PC1',
                    y='PC2',
                    hue="label_name",
                    data=df_full,
                    palette=["red", "pink", "royalblue", "greenyellow", "lightseagreen"],
                    alpha=.7).set_title(title);


plot_dim_red("PCA",
             features=features,
             labels=labels,
             n_components=2)
```




```
plot_dim_red("TSNE",
             features=features,
             labels=labels,
             n_components=2)
```




```
# Dataframe
path_df = "Pickles/df.pickle"
with open(path_df, 'rb') as data:
    df = pickle.load(data)

# SVM Model
path_model = "Models/best_svc.pickle"
with open(path_model, 'rb') as data:
    svc_model = pickle.load(data)

# Category mapping dictionary
category_codes = {
    'business': 0,
    'entertainment': 1,
    'politics': 2,
    'sport': 3,
    'tech': 4
}

category_names = {
    0: 'business',
    1: 'entertainment',
    2: 'politics',
    3: 'sport',
    4: 'tech'
}


predictions = svc_model.predict(features_test)
# Indexes of the test set
index_X_test = X_test.index

print(index_X_test)


# We get them from the original df
```

```
df_test = df.loc[index_X_test]

# Add the predictions
df_test['Prediction'] = predictions

# Clean columns
df_test = df_test[['Content', 'Category', 'Category_Code', 'Prediction']]

# Decode
df_test['Category_Predicted'] = df_test['Prediction']
df_test = df_test.replace({'Category_Predicted':category_names})

# Clean columns again
df_test = df_test[['Content', 'Category', 'Category_Predicted']]
df_test.head()
```

```
Int64Index([1691, 1103,  477,  197,  475,  162,  887,  307, 1336, 1679,
            ...
            1567, 2130, 1216, 1135,  359,  393, 1746,  444, 2215,  733],
           dtype='int64', length=334)
```

	Content	Category	Category_Predicted
1691	Ireland call up uncapped Campbell\r\n\r\nUlste...	sport	sport
1103	Gurkhas to help tsunami victims\r\n\r\nBritain...	politics	business
477	Egypt and Israel seal trade deal\r\n\r\nIn a s...	business	business
197	Cairn shares up on new oil find\r\n\r\nShares ...	business	business
475	Saudi NCCI's shares soar\r\n\r\nShares in Saud...	business	business

```
condition = (df_test['Category'] != df_test['Category_Predicted'])

df_misclassified = df_test[condition]

df_misclassified.head(3)
```

	Content	Category	Category_Predicted
1103	Gurkhas to help tsunami victims\r\n\r\nBritain...	politics	business
1880	Half-Life 2 sweeps Bafta awards\r\n\r\nPC firs...	tech	entertainment
2137	Junk e-mails on relentless rise\r\n\r\nSpam tr...	tech	business

```
def output_article(row_article):
    print('Actual Category: %s' %(row_article['Category']))
    print('Predicted Category: %s' %(row_article['Category_Predicted']))
    print('----------------------------------------')
    print('Text: ')
    print('%s' %(row_article['Content']))
```

```
import random

random.seed(8)
list_samples = random.sample(list(df_misclassified.index), 3)
list_samples
```

```
[956, 1339, 1205]
```

```
output_article(df_misclassified.loc[list_samples[0]])
```

```
Actual Category: politics
Predicted Category: tech
-----------------------------------------
Text:
Assembly ballot papers 'missing'

Hundreds of ballot papers for the regional assembly referendum in the North East have "c

Royal Mail says it is investigating the situation, which has meant about 300 homes in Co

A spokeswoman for Royal Mail said: "We are investigating a problem with the delivery rou

The Darlington Council spokesman added: "Initially we had complaints from a couple of re
```

```
output_article(df_misclassified.loc[list_samples[1]])
```

```
Actual Category: sport
Predicted Category: entertainment
-----------------------------------------
Text:
Holmes feted with further honour

Double Olympic champion Kelly Holmes has been voted European Athletics (EAA) woman athle

The Briton, made a dame in the New Year Honours List for taking 800m and 1,500m gold, wo
```

```
output_article(df_misclassified.loc[list_samples[2]])
```

```
Actual Category: politics
Predicted Category: tech
-----------------------------------------
Text:
MPs issued with Blackberry threat

MPs will be thrown out of the Commons if they use Blackberries in the chamber Speaker Mi

The Â£200 handheld computers can be used as a phone, pager or to send e-mails. The devic

The use of electronic devices in the Commons chamber has long been frowned on. The sound
```

```
path_models = "Models/"

# SVM
path_svm = path_models + 'best_svc.pickle'
with open(path_svm, 'rb') as data:
    svc_model = pickle.load(data)

path_tfidf = "Pickles/tfidf.pickle"
with open(path_tfidf, 'rb') as data:
    tfidf = pickle.load(data)

category_codes = {
    'business': 0,
    'entertainment': 1,
    'politics': 2,
    'sport': 3,
    'tech': 4
}


punctuation_signs = list("?:!.,;")
stop_words = list(stopwords.words('english'))

def create_features_from_text(text):

    # Dataframe creation
    lemmatized_text_list = []
    df = pd.DataFrame(columns=['Content'])
    df.loc[0] = text
    df['Content_Parsed_1'] = df['Content'].str.replace("\r", " ")
    df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace("\n", " ")
    df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace("    ", " ")
    df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace('"', '')
    df['Content_Parsed_2'] = df['Content_Parsed_1'].str.lower()
    df['Content_Parsed_3'] = df['Content_Parsed_2']
    for punct_sign in punctuation_signs:
        df['Content_Parsed_3'] = df['Content_Parsed_3'].str.replace(punct_sign, '')
    df['Content_Parsed_4'] = df['Content_Parsed_3'].str.replace("'s", "")
    wordnet_lemmatizer = WordNetLemmatizer()
    lemmatized_list = []
    text = df.loc[0]['Content_Parsed_4']
    text_words = text.split(" ")
    for word in text_words:
        lemmatized_list.append(wordnet_lemmatizer.lemmatize(word, pos="v"))
    lemmatized_text = " ".join(lemmatized_list)
    lemmatized_text_list.append(lemmatized_text)
    df['Content_Parsed_5'] = lemmatized_text_list
    df['Content_Parsed_6'] = df['Content_Parsed_5']
    for stop_word in stop_words:
```

```
            regex_stopword = r"\b" + stop_word + r"\b"
            df['Content_Parsed_6'] = df['Content_Parsed_6'].str.replace(regex_stopword, '')
        df = df['Content_Parsed_6']
        df = df.rename({'Content_Parsed_6': 'Content_Parsed'})

        # TF-IDF
        features = tfidf.transform(df).toarray()

        return features


    def get_category_name(category_id):
        for category, id_ in category_codes.items():
            if id_ == category_id:
                return category


    def predict_from_text(text):

        # Predict using the input model
        prediction_svc = svc_model.predict(create_features_from_text(text))[0]
        prediction_svc_proba = svc_model.predict_proba(create_features_from_text(text))[0]

        # Return result
        category_svc = get_category_name(prediction_svc)

        print("The predicted category using the SVM model is %s." %(category_svc) )
        print("The conditional probability is: %a" %(prediction_svc_proba.max()*100))


    text = """

    The center-right party Ciudadanos closed a deal on Wednesday with the support of the conserva

    Talks in Andalusia have been ongoing since regional polls were held on December 2. The PSOE,

    The move would see the Socialist Party lose power in the region for the first time in 36 year

    On Thursday, Marta Bosquet of Ciudadanos was voted in as the new speaker of the Andalusian pa

    The speaker’s role in the parliament is key for the calling of an investiture vote and for th

    Officially, the talks as to the make up of a future government have yet to start, but in real

    The speaker’s role in the parliament is key for the calling of an investiture vote and for th

    The PP, which was ousted from power by the PSOE in the national government in June, is keen t

    Wednesday was a day of intense talks among the parties in a bid to find a solution that would

    The PSOE, meanwhile, argues that having won the elections with a seven-seat lead over the PP
```

```
"""

predict_from_text(text)
```

```
     The predicted category using the SVM model is politics.
     The conditional probability is: 93.21339369980114
```

```
# Politics

text = """Disputes have already broken out within the new political alliance that is working

Just hours after the far-right Vox agreed to support the Popular Party (PP)’s candidate to he

These early clashes suggest it could be difficult to export the model to other parts of Spain

The PP and the liberal Ciudadanos have reached their own governing agreement in the wake of a

Ciudadanos has refused point-blank to meet with Vox representatives, but the PP has struck it

On Friday morning, Juan Marín of Ciudadanos said that there are no plans for a separate famil

The reform party has insisted that the Vox-PP deal does not affect them at all, and Ciudadano

 Vox national leader Santiago Abascal (c) and Andalusian leader Francisco Serrano (r).
Vox national leader Santiago Abascal (c) and Andalusian leader Francisco Serrano (r). REUTERS
But Vox insists on a family department, and said it will expect loyalty from the PP on this i

These early clashes suggest it could be difficult to export the model to other parts of Spain

The PP is anxious to win back power in regions like Valencia, the Balearic Islands, Castilla-

Parliamentary debate
The PSOE has already digested the fact that it is losing its hold on Spain’s most populated r

The Socialists will not be putting forward a candidate, now that the PP nominee has enough su

The sum of the PP, Ciudadanos and Vox votes is four above the 55 required for a majority. The

"""

predict_from_text(text)
```

```
     The predicted category using the SVM model is politics.
     The conditional probability is: 99.43575050943763
```

```
# Entertainment
```

```
text = """
Cádiz is in style: it has just been included in The New York Times’ list of 52 Places to Go i

The journalist Andrew Ferren, who wrote about Cádiz for The New York Times’ list, lives in Sp

“Despite the fact that Cádiz was historically a major maritime link between America and Europ

Culinary delights
 Aponiente restaurant in El Puerto de Santa María.
Aponiente restaurant in El Puerto de Santa María.
Suggestions include the new Western-style gastrobar Saja River, recently opened on Santa Elen

To these suggestions, EL VIAJERO adds several of its own, including Restaurante Café Royalty,

Jerez de la Frontera and its wineries
 Bodegas Lustau, en Jerez de la Frontera (Cádiz).ampliar foto
Bodegas Lustau, en Jerez de la Frontera (Cádiz). NEIL FARRIN GETTY IMAGES
Around 36 km to the north of Cádiz lies Jerez de la Frontera, known for the fortified wines k

The NMAC Montenmedio Foundation
 Vejer de la Frontera.ampliar foto
Vejer de la Frontera. GETTY IMAGES
The NMAC Montenmedio Foundation of contemporary art sits between Barbate and Vejer de la Fron

EL VIAJERO expands on Ferren’s recommendations with a few of its own:

1.The Cádiz Carnival
 The Cádiz carnival.ampliar foto
The Cádiz carnival.
An unique and fun festival that takes place from February 28 to March 10. In fact it is so un

2. Barrio del Pópulo
 The Pópulo neighborhood.ampliar foto
The Pópulo neighborhood. RAQUEL M. CARBONELL GETTY
This is the oldest neighborhood in Cádiz and features an old Roman theater, the old cathedral

3. Cádiz à la Havana
 Cathedral square in Cádiz.ampliar foto
Cathedral square in Cádiz. RAQUEL M. CARBONELL GETTY
Stroll from the colonial-style Mina Square, with its ficus and palm trees, to the Provincial

4. A wealth of history
 Baelo Claudia Roman site in Tarifa (Cádiz).ampliar foto
Baelo Claudia Roman site in Tarifa (Cádiz). KEN WELSH GETTY
Standing on the frontier between two continents, the province of Cádiz has a long and action-

5. Sanlúcar de Barrameda
 Summer beach horse races in Sanlúcar de Barrameda.ampliar foto
Summer beach horse races in Sanlúcar de Barrameda. JUAN CARLOS TORO
Famous for its summer horse racing on the beach as well as for its wineries, this coastal tow
```

```
6. Coast and mountains
 Olvera, a white village in Cádiz.ampliar foto
Olvera, a white village in Cádiz. RUDI SEBASTIAN GETTY
Cádiz has miles of windswept beaches that make it a perfect haunt for surfers of various desc

7. The flamenco route
Located in San Fernando, the Peña Flamenca Camarón de la Isla, named after the famous singer,

8. Conil de la Frontera
 The beach in Conil de la Frontera.ampliar foto
The beach in Conil de la Frontera. GETTY IMAGES
There are three national parks that stretch along Cádiz's Atlantic coast – La Breña, Los Alco

9. Surfing in Tarifa
In the inlets of Los Lances and Valdevaqueros in Tarifa, wind and kitesurfers can skid across

10. The white villages
Nineteen districts in the Cádiz mountains take you through a string of white villages – Alcal

"""

predict_from_text(text)
```

```
     The predicted category using the SVM model is entertainment.
     The conditional probability is: 99.31167341445837
```

```
# Business

text = """
Vodafone España has informed representatives of its employees that it is putting a collective

"In the current market climate, demand for services continues to grow exponentially, but this

Vodafone added that the current expectations of clients, "who demand an agile, simple and imm

As such, the company continued, it is looking to "reverse the negative trend of the business,

The operator says that it is sure it can reach a deal with labor unions so that the measures

Vodafone has suffered a great deal in the trade war that was sparked by its rivals Movistar a

In the first three quarters of 2018, Vodafone has lost 361,000 cellphone lines (70,000 of whi

The operator executed a similar collective dismissal plan (known in Spanish as an ERE) in 201

Before the acquisition of ONO, Vodafone also executed an ERE in 2013. On that occasion, the c

"""

predict_from_text(text)
```

```
The predicted category using the SVM model is business.
The conditional probability is: 93.0600852065347
```

```
# Tech

text = """
Elon Musk told the world in late 2017 that Tesla was taking its automotive know-how and apply

The German automaker also committed to manufacturing the truck this summer, with deliveries s


While there are a few Tesla Semi prototypes on the road now, and a dozen or so big name compa

DAIMLER FIRST SHOWED OFF A PROTOTYPE IN 2015
This has left the door wide open for companies like Daimler, the parent company of Mercedes-B

The new Cascadia is not much more advanced than the prototype was in 2015. In fact, the techn


The Freightliner Inspiration Truck at the event in 2015.
But the new Cascadia is doing this with a limited set of sensors. There’s a forward-facing ca

This helps keep costs down, but means the technology is more in line with what you’d find pow

DAIMLER’S TRUCK HAS MORE IN COMMON WITH NISSAN’S PROPILOT SYSTEM THAN TESLA’S AUTOPILOT
Keeping with a theme of less is more, there’s also no camera-based monitoring system in the t

A sensor in the steering column measures resistance applied to the steering wheel. If the dri


The new Cascadia is a far cry from a fully autonomous truck, but based on my brief ride, Daim

A Daimler representative also told me that, while lane centering is on, the driver can even c

RELATED

This is what it’s like to ride in Daimler’s self-driving semi truck
Daimler promised some other modern technologies are coming the new Cascadia, though none of i

The Cascadia won’t be as stuffed with tech as the Tesla Semi, nor is it as sleek. But it will
"""

predict_from_text(text)
```

```
The predicted category using the SVM model is tech.
The conditional probability is: 98.17546586390013
```

```
# Sports
```

```
text = """
Spain has agreed to host the soccer final of the Copa Libertadores between Argentina teams Ri

The final in Madrid is a punch in the soul to all fans of soccer in Argentina

ONLINE SPORTS DAILY OLE

The final was set to take place in Argentina but was suspended twice after fans turned violen

In view of the insecurity, the South American Football Confederation (Conmebol), which organi

 Embedded video

Sebastián Lisiecki
@sebalisiecki
 Así fue la llegada de Boca al Monumental. Pésimo la seguridad q los mete entre toda la gente

575
7:23 PM - Nov 24, 2018
637 people are talking about this
Twitter Ads info and privacy
This was how Boca arrived at Monumental stadium. The security that got between the all people

This is the first time a Copa Libertadores game has been played outside the Americas since th

But the feeling in Argentina has been less optimistic. The national newspaper La Nación wrote

Security risk
In a message on Twitter, Sánchez promised that "security forces have extensive experience of

River and Boca have a long-standing rivalry fueled largely by the class divide between the te

Scheduling issues
The final will take place on Sunday, December 9, on the final day of a three-day national hol

 Conmebol president Alejandro Domínguez on Tuesday.
Conmebol president Alejandro Domínguez on Tuesday.
Many details about the game have yet to be revealed, including how tickets will be sold, what

Conmebol and soccer club representatives began considering destinations for the match on Tues



"""

predict_from_text(text)
```

```
     The predicted category using the SVM model is sport.
     The conditional probability is: 75.68806067700831
```

```
# Weather

text = """
A polar air mass that entered the Iberian peninsula on Wednesday has already caused sharp dro

“An episode of intense cold” is forecast for Friday, when the mercury will continue to plumme

Elsewhere, weather stations have recorded -8.2ºC in La Molina (Girona), at an elevation of 1,

 Almería has rolled out vehicles to deal with wintry road conditions.
Almería has rolled out vehicles to deal with wintry road conditions. DIPUTACIÓN DE ALMERÍA EU
Aemet spokesman Rubén del Campo said that the cold spell is not out of the ordinary for a mon

Temperatures have already dipped between six and eight degrees in a matter of hours in some p

Temperatures on Friday and Saturday will be “very cold, with lows of five to 10 degrees below

No snow
However, little to no snow is expected “not for lack of cold, but for lack of precipitation,

Alerts are in place in Almería, Granada, Jaén, Aragón, Cantabria, Castilla-La Mancha, Castill

On Saturday, the orange warnings will extend to Córdoba, Salamanca, Valladolid, Galicia and L


"""

predict_from_text(text)

     The predicted category using the SVM model is business.
     The conditional probability is: 62.95086242483375


# Health

text = """
The obesity epidemic has been on the rise for years, with cases nearly tripling since 1975, a

An investigation by the Mar de Barcelona hospital has found that 80% of men and 55% of women

Being overweight can mean a higher risk of suffering a number of diseases, including diabetes

The study, published in the Spanish Cardiology Magazine, points out that this epidemic will m

The issue, the experts state, is not an esthetic one, but rather a question of health. Being

Researchers at the Barcelona hospital revised all of the scientific literature published in S

There are currently 25 million people with excess weight, three million more than a decade ag

DR ALBERT GODAY, AUTHOR OF THE STUDY
```

```
"There are currently 25 million people with excess weight, three million more than a decade a

"In men, excess weight is more usual up to the age of 50," explains Goday. "From 50 onward, o

The experts argue that any weight loss, no matter how small, reduces the risk of contracting

"""

predict_from_text(text)
```

```
     The predicted category using the SVM model is tech.
     The conditional probability is: 40.79994044584591
```

```
# Animal abuse

text = """
Spain's animal rights party PACMA posted a 38-second video on Twitter on Friday showing a man

"Hunters shut what appears to be a fox in a cage and let it out only to pepper it with bullet

 Video insertado

PACMA
✓
@PartidoPACMA
 Cazadores enjaulan a lo que parece ser un zorro y lo liberan solo para acribillarlo a tiros.

En realidad, son peligrosos psicópatas con escopeta y permiso de amas. #YoNoDisparo

4.188
10:43 - 4 ene. 2019
7.443 personas están hablando de esto
Información y privacidad de Twitter Ads
At the start of the video, a man teases the caged animal with a stick. When the cage door is

The release of the video, which has had 255,000 views, coincided with the launch of PACMA's c

As it notes on its website, PACMA is the only political group that opposes hunting, and it is

No animal should die under fire. We will fight tirelessly until hunting becomes a crime

PACMA

The animal rights group is preparing a report to send to the regional government of Galicia a

Last month, a Spanish hunter who was filmed while he chased and tortured a fox was identified

And in November, animal rights groups and political parties reacted with indignation over a v

"""
```

```
predict_from_text(text)
```

```
     The predicted category using the SVM model is entertainment.
     The conditional probability is: 50.955623421406294
```

Part 2

After successfully implementing their code. Try to gather data from an online URL related to autonomous cars (your choice but a long article) Use all techniques covered in the above code on the dataset that you have just created.

00,01-Data Creation,02-Exploratory Data Analysis

```
df_path = ""
df_path2 = df_path + 'New_dataset.csv'
df = pd.read_csv(df_path2)
df.head()
```

	category	text
0	tech	tv future in the hands of viewers with home th...
1	business	worldcom boss left books alone former worldc...
2	sport	tigers wary of farrell gamble leicester say ...
3	sport	yeading face newcastle in fa cup premiership s...
4	entertainment	ocean s twelve raids box office ocean s twelve...

Number of articles in each category

```
bars = alt.Chart(df).mark_bar(size=50).encode(
    x=alt.X("category"),
    y=alt.Y("count():Q", axis=alt.Axis(title='Number of articles')),
    tooltip=[alt.Tooltip('count()', title='Number of articles'), 'category'],
    color='category'

)

text = bars.mark_text(
    align='center',
    baseline='bottom',
```

```
).encode(
    text='count()'
)

(bars + text).interactive().properties(
    height=300,
    width=700,
    title = "Number of articles in each category",
)
```

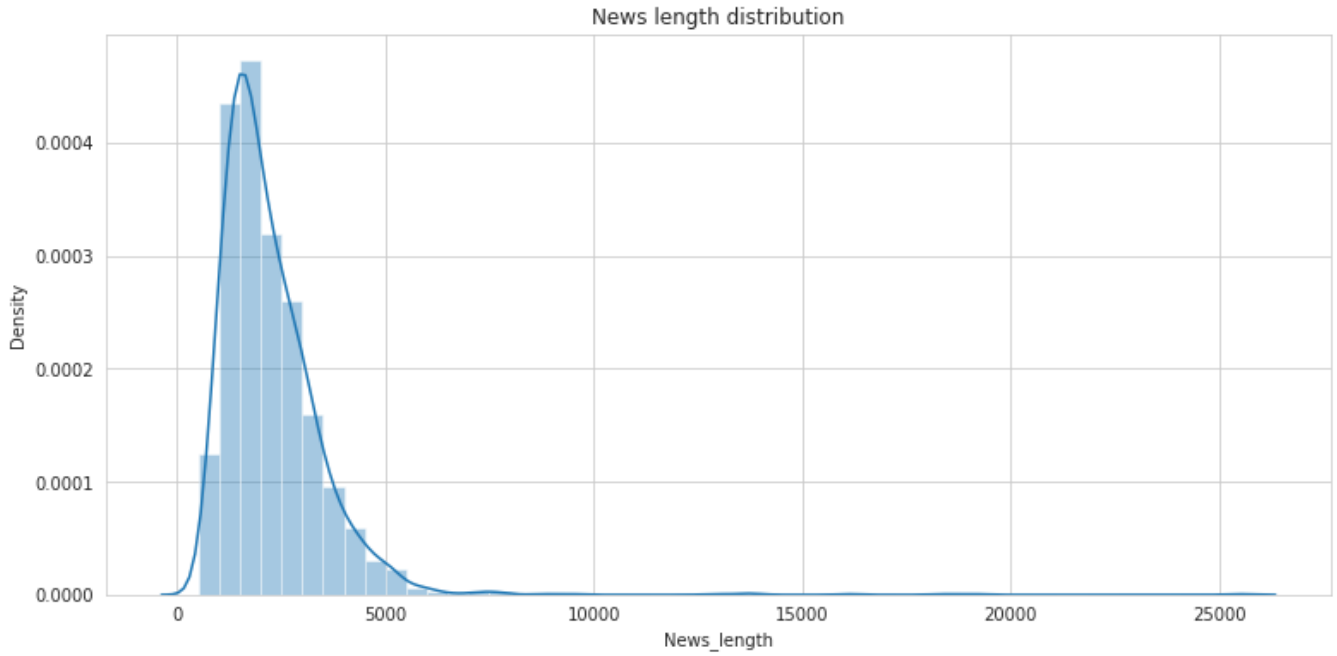
% of articles in each category

```
df['id'] = 1
df2 = pd.DataFrame(df.groupby('category').count()['id']).reset_index()

bars = alt.Chart(df2).mark_bar(size=50).encode(
    x=alt.X('category'),
    y=alt.Y('PercentOfTotal:Q', axis=alt.Axis(format='.0%', title='% of Articles')),
    color='category'
).transform_window(
    TotalArticles='sum(id)',
    frame=[None, None]
).transform_calculate(
    PercentOfTotal="datum.id / datum.TotalArticles"
)

text = bars.mark_text(
    align='center',
    baseline='bottom',
    #dx=5  # Nudges text to right so it doesn't appear on top of the bar
).encode(
    text=alt.Text('PercentOfTotal:Q', format='.1%')
)

(bars + text).interactive().properties(
    height=300,
    width=700,
    title = "% of articles in each category",
)
```

News length by category

```
df['News length'] = df['text'].str.len()
```

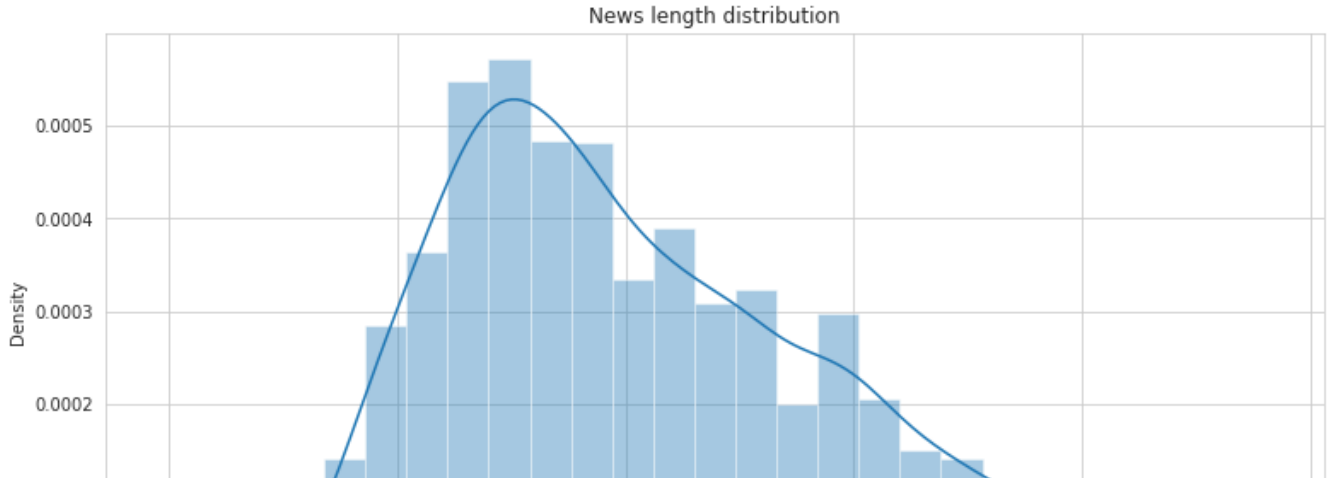
```
plt.figure(figsize=(12.8,6))
sns.distplot(df['News_length']).set_title('News length distribution');
```




```
df['News_length'].describe()
```

```
count     2225.00000
mean      2262.93618
std       1364.10253
min        501.00000
25%       1446.00000
50%       1965.00000
75%       2802.00000
max      25483.00000
Name: News_length, dtype: float64
```

```
quantile_95 = df['News_length'].quantile(0.95)
df_95 = df[df['News_length'] < quantile_95]
plt.figure(figsize=(12.8,6))
sns.distplot(df_95['News_length']).set_title('News length distribution');
```




```
df_more10k = df[df['News_length'] > 10000]
len(df_more10k)
```

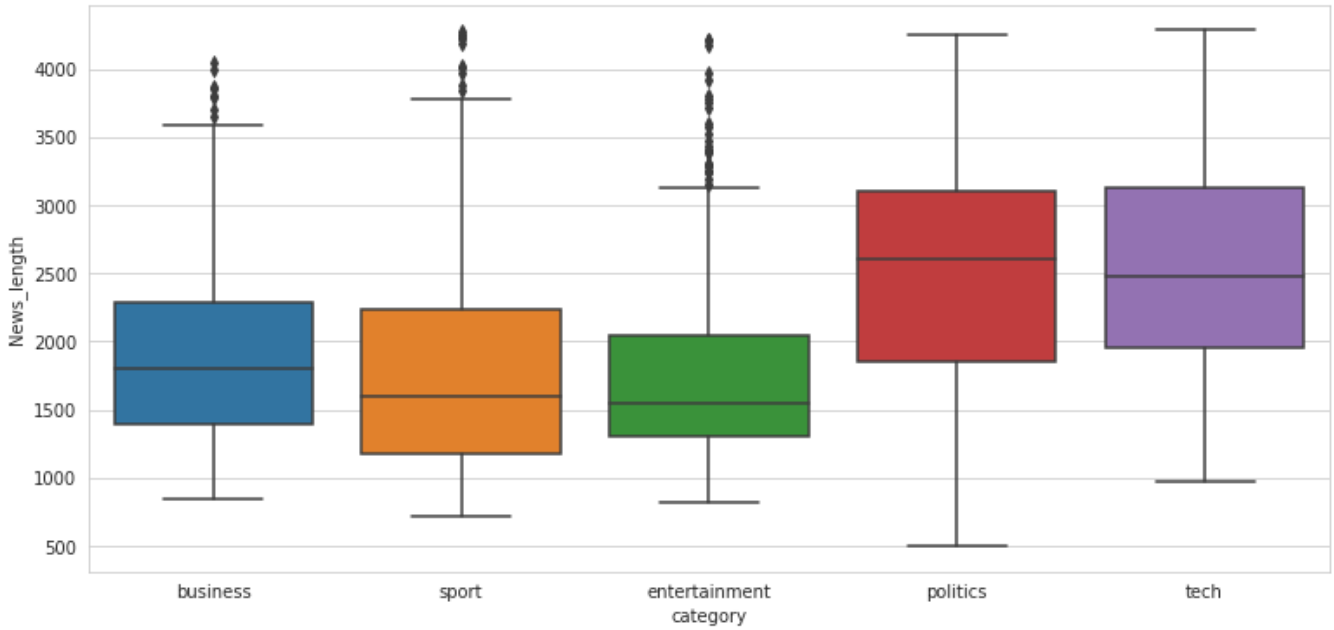
```
7
```

```
df_more10k['text'].iloc[0]
```

```
'terror powers expose  tyranny  the lord chancellor has defended government plans to in
troduce control orders to keep foreign and british terrorist suspects under house arres
t  where there isn t enough evidence to put them on trial.  lord falconer insists that
the proposals do not equate to a police state and strike a balance between protecting t
he public against the threat of terrorism and upholding civil liberties. but thriller w
riter frederick forsyth tells bbc news of his personal response to the move.    there i
s a mortal danger aimed at the heart of britain. or so says home secretary charles clar
ke. my reaction  so what  it is not that i am cynical or just do not care. i care about
```

```
plt.figure(figsize=(12.8,6))
sns.boxplot(data=df, x='category', y='News_length', width=.5);
```

```
plt.figure(figsize=(12.8,6))
sns.boxplot(data=df_95, x='category', y='News_length');
```




```
with open('Part2_dataset.pickle', 'wb') as output:
    pickle.dump(df, output)
```

03

```
path_df = "Part2_dataset.pickle"

with open(path_df, 'rb') as data:
    df = pickle.load(data)

df.head()
```

	category	text	id	News_length
0	tech	tv future in the hands of viewers with home th...	1	4333
1	business	worldcom boss left books alone former worldc...	1	1842
2	sport	tigers wary of farrell gamble leicester say ...	1	1342
3	sport	yeading face newcastle in fa cup premiership s...	1	2176
4	entertainment	ocean s twelve raids box office ocean s twelve...	1	1579

```
df.loc[1]['text']
```

```
'worldcom boss  left books alone  former worldcom boss bernie ebbers  who is accused of overseeing an $11bn (£5.8bn) fraud  never made accounting decisions  a witness has told jurors.  david myers made the comments under questioning by defence lawyers who have been arguing that mr ebbers was not responsible for worldcom s problems. the phone company collapsed in 2002 and prosecutors claim that losses were hidden to protect the firm s shares. mr myers has already pleaded guilty to fraud and is assisting prosecutors.  on monday  defence lawyer reid weingarten tried to distance his client from the allegations. during cross examination  he asked mr myers if he ever knew mr ebbers  make an accou
```

Text Cleaning

Special Character Cleaning

```
# \r and \n
df['Content_Parsed_1'] = df['text'].str.replace("\r", " ")
df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace("\n", " ")
df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace("    ", " ")
# " when quoting text
df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace('"', '')
text = "Mr Greenspan\'s"
text
```

```
'Mr Greenspan's'
```

Upcase/downcase

```
# Lowercasing the text
df['Content_Parsed_2'] = df['Content_Parsed_1'].str.lower()
```

Punctuation signs

```
punctuation_signs = list("?:!.,;")
df['Content_Parsed_3'] = df['Content_Parsed_2']

for punct_sign in punctuation_signs:
    df['Content_Parsed_3'] = df['Content_Parsed_3'].str.replace(punct_sign, '')
```

Possessive Pronouns

```
df['Content_Parsed_4'] = df['Content_Parsed_3'].str.replace("'s", "")
```

Stemming and Lemmatization

```
# Saving the lemmatizer into an object
wordnet_lemmatizer = WordNetLemmatizer()

nrows = len(df)
lemmatized_text_list = []

for row in range(0, nrows):

    # Create an empty list containing lemmatized words
    lemmatized_list = []

    # Save the text and its words into an object
    text = df.loc[row]['Content_Parsed_4']
    text_words = text.split(" ")

    # Iterate through every word to lemmatize
    for word in text_words:
        lemmatized_list.append(wordnet_lemmatizer.lemmatize(word, pos="v"))

    # Join the list
    lemmatized_text = " ".join(lemmatized_list)

    # Append to the list containing the texts
    lemmatized_text_list.append(lemmatized_text)

df['Content_Parsed_5'] = lemmatized_text_list
```

Stop words

```
# Loading the stop words in english
stop_words = list(stopwords.words('english'))
stop_words[0:10]
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're"]
```

```
example = "me eating a meal"
word = "me"

# The regular expression is:
regex = r"\b" + word + r"\b"  # we need to build it like that to work properly

re.sub(regex, "StopWord", example)
```

```
'StopWord eating a meal'
```

```
df['Content_Parsed_6'] = df['Content_Parsed_5']

for stop_word in stop_words:

    regex_stopword = r"\b" + stop_word + r"\b"
    df['Content_Parsed_6'] = df['Content_Parsed_6'].str.replace(regex_stopword, '')
```

Results of parsing

```
df.loc[5]['text']
```

```
'howard hits back at mongrel jibe michael howard has said a claim by peter hain that th
e tory leader is acting like an  attack mongrel  shows labour is  rattled  by the oppos
ition.  in an upbeat speech to his party s spring conference in brighton  he said labou
r s campaigning tactics proved the tories were hitting home. mr hain made the claim abo
ut tory tactics in the anti-terror bill debate.  something tells me that someone  somew
here out there is just a little bit rattled   mr howard said. mr hain  leader of the co
mmons  told bbc radio four s today programme that mr howard s stance on the government
s anti-terrorism legislation was putting the country at risk. he then accused the tory
```

```
df.loc[5]['Content_Parsed_1']
```

```
'howard hits back at mongrel jibe michael howard has said a claim by peter hain that th
e tory leader is acting like an  attack mongrel  shows labour is  rattled  by the oppos
ition.  in an upbeat speech to his party s spring conference in brighton  he said labou
r s campaigning tactics proved the tories were hitting home. mr hain made the claim abo
ut tory tactics in the anti-terror bill debate.  something tells me that someone  somew
here out there is just a little bit rattled   mr howard said. mr hain  leader of the co
mmons  told bbc radio four s today programme that mr howard s stance on the government
s anti-terrorism legislation was putting the country at risk. he then accused the tory
```

```
df.loc[5]['Content_Parsed_2']
```

```
'howard hits back at mongrel jibe michael howard has said a claim by peter hain that th
e tory leader is acting like an  attack mongrel  shows labour is  rattled  by the oppos
ition.  in an upbeat speech to his party s spring conference in brighton  he said labou
r s campaigning tactics proved the tories were hitting home. mr hain made the claim abo
ut tory tactics in the anti-terror bill debate.  something tells me that someone  somew
here out there is just a little bit rattled   mr howard said. mr hain  leader of the co
mmons  told bbc radio four s today programme that mr howard s stance on the government
s anti-terrorism legislation was putting the country at risk. he then accused the tory
```

```
df.loc[5]['Content_Parsed_3']
```

```
'howard hits back at mongrel jibe michael howard has said a claim by peter hain that th
e tory leader is acting like an  attack mongrel  shows labour is  rattled  by the oppos
```

```
df.loc[5]['Content_Parsed_4']
```

```
'howard hits back at mongrel jibe michael howard has said a claim by peter hain that th
e tory leader is acting like an  attack mongrel  shows labour is  rattled  by the oppos
ition  in an upbeat speech to his party s spring conference in brighton  he said labour
s campaigning tactics proved the tories were hitting home mr hain made the claim about
tory tactics in the anti-terror bill debate  something tells me that someone  somewhere
out there is just a little bit rattled   mr howard said mr hain  leader of the commons
told bbc radio four s today programme that mr howard s stance on the government s anti-
terrorism legislation was putting the country at risk he then accused the tory leader o
```

```
df.loc[5]['Content_Parsed_5']
```

```
'howard hit back at mongrel jibe michael howard have say a claim by peter hain that the
tory leader be act like an  attack mongrel  show labour be  rattle  by the opposition
in an upbeat speech to his party s spring conference in brighton  he say labour s campa
ign tactics prove the tories be hit home mr hain make the claim about tory tactics in t
he anti-terror bill debate  something tell me that someone  somewhere out there be just
a little bite rattle   mr howard say mr hain  leader of the commons  tell bbc radio fou
r s today programme that mr howard s stance on the government s anti-terrorism legislat
ion be put the country at risk he then accuse the tory leader of behave like an  attack
```

```
df.loc[5]['Content_Parsed_6']
```

```
'howard hit back  mongrel jibe michael howard  say  claim  peter hain   tory leader  ac
t like   attack mongrel  show labour   rattle   opposition   upbeat speech   party  s
pring conference  brighton   say labour  campaign tactics prove  tories  hit home mr ha
in make  claim  tory tactics   anti-terror bill debate  something tell   someone  somew
here      little bite rattle   mr howard say mr hain  leader   commons  tell bbc radio
four  today programme  mr howard  stance   government  anti-terrorism legislation  put
country  risk   accuse  tory leader  behave like   attack mongrel    play opposition  o
pposition sake   mr howard tell  party  labour would   anything  say anything  claim an
```

```
df.head(1)
```

	category	text	id	News_length	Content_Parsed_1	Content_Parsed_2	Content_Parse
0	tech	tv future in the hands of viewers with home	1	4333	tv future in the hands of viewers with home th...	tv future in the hands of viewers with home th...	tv future in hands of view with home

```
list_columns = ["category", "text", "Content_Parsed_6"]
df = df[list_columns]

df = df.rename(columns={'Content_Parsed_6': 'Content_Parsed'})
```

```
df.head()
```

	category	text	Content_Parsed
0	tech	tv future in the hands of viewers with home th...	tv future hand viewers home theatre system...
1	business	worldcom boss left books alone former worldc...	worldcom boss leave book alone former worldc...
2	sport	tigers wary of farrell gamble leicester say ...	tigers wary farrell gamble leicester say ...

Label Coding

```
category_codes = {
    'business': 0,
    'entertainment': 1,
    'politics': 2,
    'sport': 3,
    'tech': 4
}

# Category mapping
df['Category_Code'] = df['category']
df = df.replace({'Category_Code':category_codes})

df.head()
```

	category	text	Content_Parsed	Category_Code
0	tech	tv future in the hands of viewers with home th...	tv future hand viewers home theatre system...	4
1	business	worldcom boss left books alone former worldc...	worldcom boss leave book alone former worldc...	0
2	sport	tigers wary of farrell gamble leicester say ...	tigers wary farrell gamble leicester say ...	3
		veading face newcastle in fa	veading face newcastle fa cup	

Train Test Split

```
X_train, X_test, y_train, y_test = train_test_split(df['Content_Parsed'],
                                                    df['Category_Code'],
                                                    test_size=0.15,
                                                    random_state=8)
```

Text Representation

```
# Parameter election
ngram_range = (1,2)
min_df = 10
max_df = 1.
max_features = 300


tfidf = TfidfVectorizer(encoding='utf-8',
                        ngram_range=ngram_range,
                        stop_words=None,
                        lowercase=False,
                        max_df=max_df,
                        min_df=min_df,
                        max_features=max_features,
                        norm='l2',
                        sublinear_tf=True)

features_train = tfidf.fit_transform(X_train).toarray()
labels_train = y_train
print(features_train.shape)

features_test = tfidf.transform(X_test).toarray()
labels_test = y_test
print(features_test.shape)
```

```
     (1891, 300)
     (334, 300)
```

```
for Product, category_id in sorted(category_codes.items()):
    features_chi2 = chi2(features_train, labels_train == category_id)
    indices = np.argsort(features_chi2[0])
    feature_names = np.array(tfidf.get_feature_names())[indices]
    unigrams = [v for v in feature_names if len(v.split(' ')) == 1]
    bigrams = [v for v in feature_names if len(v.split(' ')) == 2]
    print("# '{}' category:".format(Product))
    print("  . Most correlated unigrams:\n. {}".format('\n. '.join(unigrams[-5:])))
    print("  . Most correlated bigrams:\n. {}".format('\n. '.join(bigrams[-2:])))
    print("")
```

```
     # 'business' category:
       . Most correlated unigrams:
     . market
     . price
     . economy
     . growth
     . bank
       . Most correlated bigrams:
```

```
      . last year
      . year old

      # 'entertainment' category:
        . Most correlated unigrams:
      . tv
      . music
      . award
      . star
      . film
        . Most correlated bigrams:
      . mr blair
      . prime minister

      # 'politics' category:
        . Most correlated unigrams:
      . minister
      . blair
      . election
      . party
      . labour
        . Most correlated bigrams:
      . prime minister
      . mr blair

      # 'sport' category:
        . Most correlated unigrams:
      . game
      . win
      . team
      . cup
      . match
        . Most correlated bigrams:
      . say mr
      . year old

      # 'tech' category:
        . Most correlated unigrams:
      . digital
      . computer
      . technology
      . software
      . users
        . Most correlated bigrams:
      . year old
      . say mr
```

```
bigrams
```

```
['tell bbc', 'last year', 'mr blair', 'prime minister', 'year old', 'say mr']
```

```
# X_train
with open('NewPickles/X_train.pickle', 'wb') as output:
    pickle.dump(X_train, output)
```

```
# X_test
with open('NewPickles/X_test.pickle', 'wb') as output:
    pickle.dump(X_test, output)

# y_train
with open('NewPickles/y_train.pickle', 'wb') as output:
    pickle.dump(y_train, output)

# y_test
with open('NewPickles/y_test.pickle', 'wb') as output:
    pickle.dump(y_test, output)

# df
with open('NewPickles/df.pickle', 'wb') as output:
    pickle.dump(df, output)

# features_train
with open('NewPickles/features_train.pickle', 'wb') as output:
    pickle.dump(features_train, output)

# labels_train
with open('NewPickles/labels_train.pickle', 'wb') as output:
    pickle.dump(labels_train, output)

# features_test
with open('NewPickles/features_test.pickle', 'wb') as output:
    pickle.dump(features_test, output)

# labels_test
with open('NewPickles/labels_test.pickle', 'wb') as output:
    pickle.dump(labels_test, output)

# TF-IDF object
with open('NewPickles/tfidf.pickle', 'wb') as output:
    pickle.dump(tfidf, output)
```

04

```
# Dataframe
path_df = "NewPickles/df.pickle"
with open(path_df, 'rb') as data:
    df = pickle.load(data)

# features_train
path_features_train = "NewPickles/features_train.pickle"
with open(path_features_train, 'rb') as data:
    features_train = pickle.load(data)
```

```
# labels_train
path_labels_train = "NewPickles/labels_train.pickle"
with open(path_labels_train, 'rb') as data:
    labels_train = pickle.load(data)

# features_test
path_features_test = "NewPickles/features_test.pickle"
with open(path_features_test, 'rb') as data:
    features_test = pickle.load(data)

# labels_test
path_labels_test = "NewPickles/labels_test.pickle"
with open(path_labels_test, 'rb') as data:
    labels_test = pickle.load(data)

print(features_train.shape)
print(features_test.shape)
```

```
(1891, 300)
(334, 300)
```

Random Forest

```
rf_0 = RandomForestClassifier(random_state = 8)

print('Parameters currently in use:\n')
pprint(rf_0.get_params())
```

```
Parameters currently in use:

{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': 8,
 'verbose': 0,
 'warm_start': False}
```

```
# n_estimators
```

```
# n_estimators
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 1000, num = 5)]

# max_features
max_features = ['auto', 'sqrt']

# max_depth
max_depth = [int(x) for x in np.linspace(20, 100, num = 5)]
max_depth.append(None)

# min_samples_split
min_samples_split = [2, 5, 10]

# min_samples_leaf
min_samples_leaf = [1, 2, 4]

# bootstrap
bootstrap = [True, False]

# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}

pprint(random_grid)
```

```
     {'bootstrap': [True, False],
      'max_depth': [20, 40, 60, 80, 100, None],
      'max_features': ['auto', 'sqrt'],
      'min_samples_leaf': [1, 2, 4],
      'min_samples_split': [2, 5, 10],
      'n_estimators': [200, 400, 600, 800, 1000]}
```

```
# First create the base model to tune
rfc = RandomForestClassifier(random_state=8)

# Definition of the random search
random_search = RandomizedSearchCV(estimator=rfc,
                                   param_distributions=random_grid,
                                   n_iter=50,
                                   scoring='accuracy',
                                   cv=3,
                                   verbose=1,
                                   random_state=8)

# Fit the random search model
random_search.fit(features_train, labels_train)
```

```
     Fitting 3 folds for each of 50 candidates, totalling 150 fits
```

```
RandomizedSearchCV(cv=3, estimator=RandomForestClassifier(random_state=8),
                   n_iter=50,
                   param_distributions={'bootstrap': [True, False],
                                        'max_depth': [20, 40, 60, 80, 100,
                                                      None],
                                        'max_features': ['auto', 'sqrt'],
                                        'min_samples_leaf': [1, 2, 4],
                                        'min_samples_split': [2, 5, 10],
                                        'n_estimators': [200, 400, 600, 800,
                                                         1000]},
                   random_state=8, scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Random Search are:")
print(random_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(random_search.best_score_)
```

```
The best hyperparameters from Random Search are:
{'n_estimators': 400, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_features': 'sc

The mean accuracy of a model with these hyperparameters is:
0.9423632597959063
```

```
# Create the parameter grid based on the results of random search
bootstrap = [False]
max_depth = [30, 40, 50]
max_features = ['sqrt']
min_samples_leaf = [1, 2, 4]
min_samples_split = [5, 10, 15]
n_estimators = [800]

param_grid = {
    'bootstrap': bootstrap,
    'max_depth': max_depth,
    'max_features': max_features,
    'min_samples_leaf': min_samples_leaf,
    'min_samples_split': min_samples_split,
    'n_estimators': n_estimators
}

# Create a base model
rfc = RandomForestClassifier(random_state=8)

# Manually create the splits in CV in order to be able to fix a random_state (GridSearchCV do
cv_sets = ShuffleSplit(n_splits = 3, test_size = .33, random_state = 8)

# Instantiate the grid search model
grid_search = GridSearchCV(estimator=rfc,
                           param_grid=param_grid,
```

```
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

# Fit the grid search to the data
grid_search.fit(features_train, labels_train)
```

```
Fitting 3 folds for each of 27 candidates, totalling 81 fits
GridSearchCV(cv=ShuffleSplit(n_splits=3, random_state=8, test_size=0.33,
train_size=None),
             estimator=RandomForestClassifier(random_state=8),
             param_grid={'bootstrap': [False], 'max_depth': [30, 40, 50],
                         'max_features': ['sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [5, 10, 15],
                         'n_estimators': [800]},
             scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Grid Search are:")
print(grid_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(grid_search.best_score_)
```

```
The best hyperparameters from Grid Search are:
{'bootstrap': False, 'max_depth': 30, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'mi

The mean accuracy of a model with these hyperparameters is:
0.9402666666666667
```

```
best_rfc = grid_search.best_estimator_
```

```
best_rfc
```

```
RandomForestClassifier(bootstrap=False, max_depth=30, max_features='sqrt',
                       min_samples_split=10, n_estimators=800, random_state=8)
```

```
best_rfc.fit(features_train, labels_train)
```

```
RandomForestClassifier(bootstrap=False, max_depth=30, max_features='sqrt',
                       min_samples_split=10, n_estimators=800, random_state=8)
```

```
rfc_pred = best_rfc.predict(features_test)
```

```
# Training accuracy
print("The training accuracy is: ")
print(accuracy_score(labels_train, best_rfc.predict(features_train)))
```

```
The training accuracy is:
1.0
```

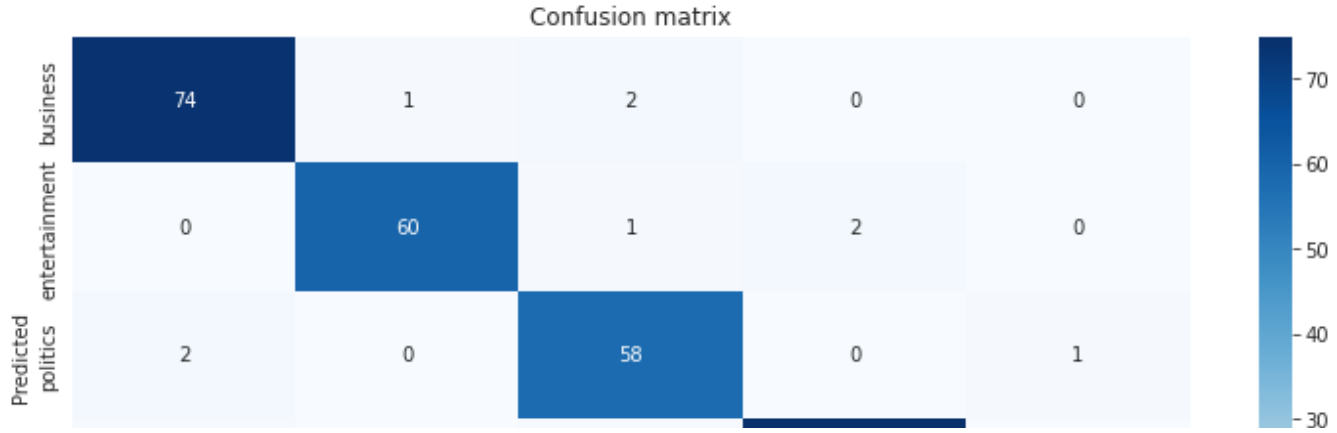
```
# Test accuracy
print("The test accuracy is: ")
print(accuracy_score(labels_test, rfc_pred))
```

```
The test accuracy is:
0.9610778443113772
```

```
# Classification report
print("Classification report")
print(classification_report(labels_test,rfc_pred))
```

```
Classification report
              precision    recall  f1-score   support

           0       0.95      0.96      0.95        77
           1       0.97      0.95      0.96        63
           2       0.94      0.95      0.94        61
           3       0.97      0.97      0.97        77
           4       0.98      0.96      0.97        56

    accuracy                           0.96       334
   macro avg       0.96      0.96      0.96       334
weighted avg       0.96      0.96      0.96       334
```

```
aux_df = df[['category', 'Category_Code']].drop_duplicates().sort_values('Category_Code')
conf_matrix = confusion_matrix(labels_test, rfc_pred)
plt.figure(figsize=(12.8,6))
sns.heatmap(conf_matrix,
            annot=True,
            xticklabels=aux_df['category'].values,
            yticklabels=aux_df['category'].values,
            cmap="Blues")
plt.ylabel('Predicted')
plt.xlabel('Actual')
plt.title('Confusion matrix')
plt.show()
```




```
base_model = RandomForestClassifier(random_state = 8)
base_model.fit(features_train, labels_train)
accuracy_score(labels_test, base_model.predict(features_test))
```

```
0.9550898203592815
```

```
best_rfc.fit(features_train, labels_train)
accuracy_score(labels_test, best_rfc.predict(features_test))
```

```
0.9610778443113772
```

```
d = {
    'Model': 'Random Forest',
    'Training Set Accuracy': accuracy_score(labels_train, best_rfc.predict(features_train)),
    'Test Set Accuracy': accuracy_score(labels_test, rfc_pred)
}

df_models_rfc = pd.DataFrame(d, index=[0])


df_models_rfc
```

	Model	Training Set Accuracy	Test Set Accuracy
0	Random Forest	1.0	0.961078

```
with open('NewModels/best_rfc.pickle', 'wb') as output:
    pickle.dump(best_rfc, output)

with open('NewModels/df_models_rfc.pickle', 'wb') as output:
    pickle.dump(df_models_rfc, output)
```

SVM

```
svc_0 =svm.SVC(random_state=8)

print('Parameters currently in use:\n')
pprint(svc_0.get_params())
```

```
     Parameters currently in use:

     {'C': 1.0,
      'break_ties': False,
      'cache_size': 200,
      'class_weight': None,
      'coef0': 0.0,
      'decision_function_shape': 'ovr',
      'degree': 3,
      'gamma': 'scale',
      'kernel': 'rbf',
      'max_iter': -1,
      'probability': False,
      'random_state': 8,
      'shrinking': True,
      'tol': 0.001,
      'verbose': False}
```

```
# C
C = [.0001, .001, .01]

# gamma
gamma = [.0001, .001, .01, .1, 1, 10, 100]

# degree
degree = [1, 2, 3, 4, 5]

# kernel
kernel = ['linear', 'rbf', 'poly']

# probability
probability = [True]

# Create the random grid
random_grid = {'C': C,
               'kernel': kernel,
               'gamma': gamma,
               'degree': degree,
               'probability': probability
              }

pprint(random_grid)
```

```
     {'C': [0.0001, 0.001, 0.01],
      'degree': [1, 2, 3, 4, 5],
      'gamma': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100],
```

```
        'kernel': ['linear', 'rbf', 'poly'],
        'probability': [True]}

# First create the base model to tune
svc = svm.SVC(random_state=8)

# Definition of the random search
random_search = RandomizedSearchCV(estimator=svc,
                                   param_distributions=random_grid,
                                   n_iter=50,
                                   scoring='accuracy',
                                   cv=3,
                                   verbose=1,
                                   random_state=8)

# Fit the random search model
random_search.fit(features_train, labels_train)
```

```
Fitting 3 folds for each of 50 candidates, totalling 150 fits
RandomizedSearchCV(cv=3, estimator=SVC(random_state=8), n_iter=50,
                   param_distributions={'C': [0.0001, 0.001, 0.01],
                                        'degree': [1, 2, 3, 4, 5],
                                        'gamma': [0.0001, 0.001, 0.01, 0.1, 1,
                                                  10, 100],
                                        'kernel': ['linear', 'rbf', 'poly'],
                                        'probability': [True]},
                   random_state=8, scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Random Search are:")
print(random_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(random_search.best_score_)
```

```
The best hyperparameters from Random Search are:
{'probability': True, 'kernel': 'poly', 'gamma': 10, 'degree': 4, 'C': 0.01}

The mean accuracy of a model with these hyperparameters is:
0.9217434323614989
```

```
# Create the parameter grid based on the results of random search
C = [.0001, .001, .01, .1]
degree = [3, 4, 5]
gamma = [1, 10, 100]
probability = [True]

param_grid = [
  {'C': C, 'kernel':['linear'], 'probability':probability},
  {'C': C, 'kernel':['poly'], 'degree':degree, 'probability':probability},
  {'C': C, 'kernel':['rbf'], 'gamma':gamma, 'probability':probability}
]
```

```
# Create a base model
svc = svm.SVC(random_state=8)

# Manually create the splits in CV in order to be able to fix a random_state (GridSearchCV do
cv_sets = ShuffleSplit(n_splits = 3, test_size = .33, random_state = 8)

# Instantiate the grid search model
grid_search = GridSearchCV(estimator=svc,
                           param_grid=param_grid,
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

# Fit the grid search to the data
grid_search.fit(features_train, labels_train)
```

```
Fitting 3 folds for each of 28 candidates, totalling 84 fits
GridSearchCV(cv=ShuffleSplit(n_splits=3, random_state=8, test_size=0.33,
train_size=None),
             estimator=SVC(random_state=8),
             param_grid=[{'C': [0.0001, 0.001, 0.01, 0.1], 'kernel': ['linear'],
                          'probability': [True]},
                         {'C': [0.0001, 0.001, 0.01, 0.1], 'degree': [3, 4, 5],
                          'kernel': ['poly'], 'probability': [True]},
                         {'C': [0.0001, 0.001, 0.01, 0.1],
                          'gamma': [1, 10, 100], 'kernel': ['rbf'],
                          'probability': [True]}],
             scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Grid Search are:")
print(grid_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(grid_search.best_score_)
```

```
The best hyperparameters from Grid Search are:
{'C': 0.1, 'kernel': 'linear', 'probability': True}

The mean accuracy of a model with these hyperparameters is:
0.9413333333333332
```

```
best_svc = grid_search.best_estimator_
best_svc
```

```
SVC(C=0.1, kernel='linear', probability=True, random_state=8)
```

```
best_svc.fit(features_train, labels_train)
```

```
SVC(C=0.1, kernel='linear', probability=True, random_state=8)
```

```
svc_pred = best_svc.predict(features_test)

# Training accuracy
print("The training accuracy is: ")
print(accuracy_score(labels_train, best_svc.predict(features_train)))
```

```
The training accuracy is:
0.958223162347964
```

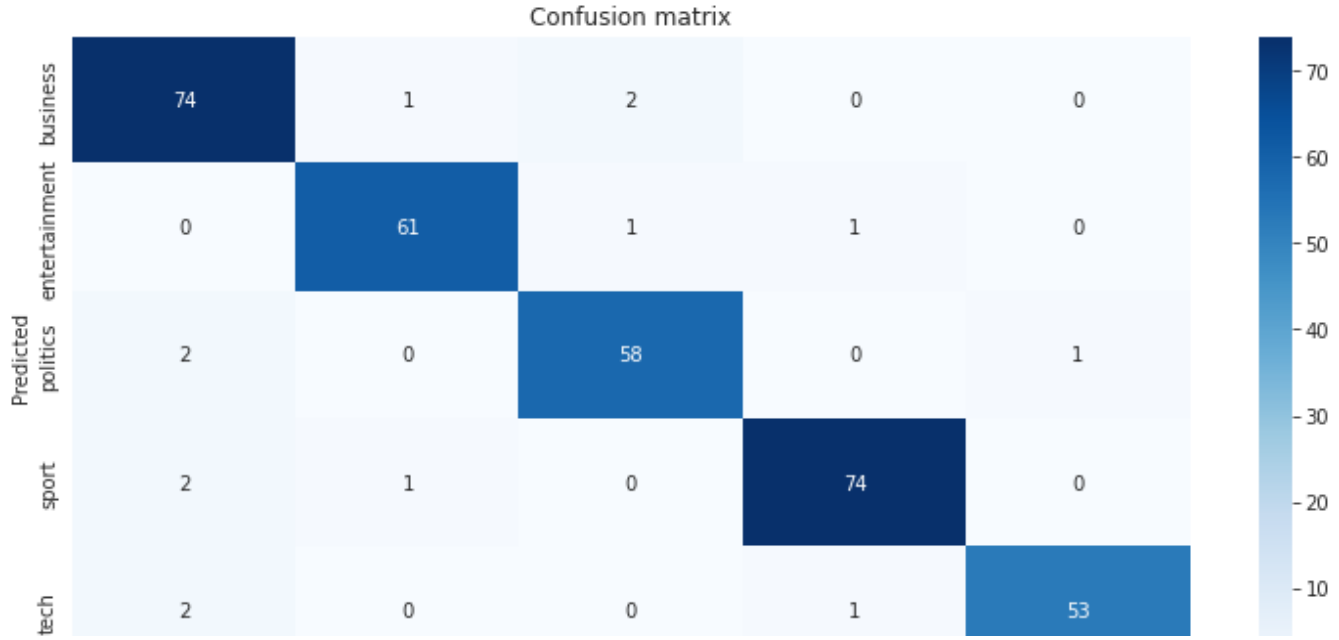
```
# Test accuracy
print("The test accuracy is: ")
print(accuracy_score(labels_test, svc_pred))
```

```
The test accuracy is:
0.9580838323353293
```

```
# Classification report
print("Classification report")
print(classification_report(labels_test,svc_pred))
```

```
Classification report
              precision    recall  f1-score   support

           0       0.93      0.96      0.94        77
           1       0.97      0.97      0.97        63
           2       0.95      0.95      0.95        61
           3       0.97      0.96      0.97        77
           4       0.98      0.95      0.96        56

    accuracy                           0.96       334
   macro avg       0.96      0.96      0.96       334
weighted avg       0.96      0.96      0.96       334
```

```
aux_df = df[['category', 'Category_Code']].drop_duplicates().sort_values('Category_Code')
conf_matrix = confusion_matrix(labels_test, svc_pred)
plt.figure(figsize=(12.8,6))
sns.heatmap(conf_matrix,
            annot=True,
            xticklabels=aux_df['category'].values,
            yticklabels=aux_df['category'].values,
            cmap="Blues")
plt.ylabel('Predicted')
plt.xlabel('Actual')
plt.title('Confusion matrix')
plt.show()
```




```
base_model = svm.SVC(random_state = 8)
base_model.fit(features_train, labels_train)
accuracy_score(labels_test, base_model.predict(features_test))
```

```
0.9730538922155688
```

```
best_svc.fit(features_train, labels_train)
accuracy_score(labels_test, best_svc.predict(features_test))
```

```
0.9580838323353293
```

```
d = {
    'Model': 'SVM',
    'Training Set Accuracy': accuracy_score(labels_train, best_svc.predict(features_train)),
    'Test Set Accuracy': accuracy_score(labels_test, svc_pred)
}

df_models_svc = pd.DataFrame(d, index=[0])
df_models_svc
```

	Model	Training Set Accuracy	Test Set Accuracy
0	SVM	0.958223	0.958084

```
with open('NewModels/best_svc.pickle', 'wb') as output:
    pickle.dump(best_svc, output)

with open('NewModels/df_models_svc.pickle', 'wb') as output:
    pickle.dump(df_models_svc, output)
```

KNN

```
knnc_0 =KNeighborsClassifier()

print('Parameters currently in use:\n')
pprint(knnc_0.get_params())
```

```
     Parameters currently in use:

     {'algorithm': 'auto',
      'leaf_size': 30,
      'metric': 'minkowski',
      'metric_params': None,
      'n_jobs': None,
      'n_neighbors': 5,
      'p': 2,
      'weights': 'uniform'}
```

```
# Create the parameter grid
n_neighbors = [int(x) for x in np.linspace(start = 1, stop = 500, num = 100)]

param_grid = {'n_neighbors': n_neighbors}

# Create a base model
knnc = KNeighborsClassifier()

# Manually create the splits in CV in order to be able to fix a random_state (GridSearchCV do
cv_sets = ShuffleSplit(n_splits = 3, test_size = .33, random_state = 8)

# Instantiate the grid search model
grid_search = GridSearchCV(estimator=knnc,
                           param_grid=param_grid,
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

# Fit the grid search to the data
grid_search.fit(features_train, labels_train)
```

```
     Fitting 3 folds for each of 100 candidates, totalling 300 fits
     GridSearchCV(cv=ShuffleSplit(n_splits=3, random_state=8, test_size=0.33,
     train_size=None),
                  estimator=KNeighborsClassifier(),
                  param_grid={'n_neighbors': [1, 6, 11, 16, 21, 26, 31, 36, 41, 46,
                                              51, 56, 61, 66, 71, 76, 81, 86, 91, 96,
                                              101, 106, 111, 116, 121, 127, 132, 137,
                                              142, 147, ...]},
                  scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Grid Search are:")
print(grid_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(grid_search.best_score_)
```

```
The best hyperparameters from Grid Search are:
{'n_neighbors': 11}

The mean accuracy of a model with these hyperparameters is:
0.9418666666666667
```

```
n_neighbors = [1,2,3,4,5,6,7,8,9,10,11]
param_grid = {'n_neighbors': n_neighbors}

knnc = KNeighborsClassifier()
cv_sets = ShuffleSplit(n_splits = 3, test_size = .33, random_state = 8)

grid_search = GridSearchCV(estimator=knnc,
                           param_grid=param_grid,
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

grid_search.fit(features_train, labels_train)
```

```
Fitting 3 folds for each of 11 candidates, totalling 33 fits
GridSearchCV(cv=ShuffleSplit(n_splits=3, random_state=8, test_size=0.33,
train_size=None),
             estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]},
             scoring='accuracy', verbose=1)
```

```
print("The best hyperparameters from Grid Search are:")
print(grid_search.best_params_)
print("")
print("The mean accuracy of a model with these hyperparameters is:")
print(grid_search.best_score_)
```

```
The best hyperparameters from Grid Search are:
{'n_neighbors': 10}

The mean accuracy of a model with these hyperparameters is:
0.9461333333333334
```

```
best_knnc = grid_search.best_estimator_
best_knnc
```

```
KNeighborsClassifier(n_neighbors=10)
```

```
best_knnc.fit(features_train, labels_train)
```

```
KNeighborsClassifier(n_neighbors=10)
```

```
knnc_pred = best_knnc.predict(features_test)
```

```
# Training accuracy
print("The training accuracy is: ")
print(accuracy_score(labels_train, best_knnc.predict(features_train)))
```

```
The training accuracy is:
0.952934955050238
```

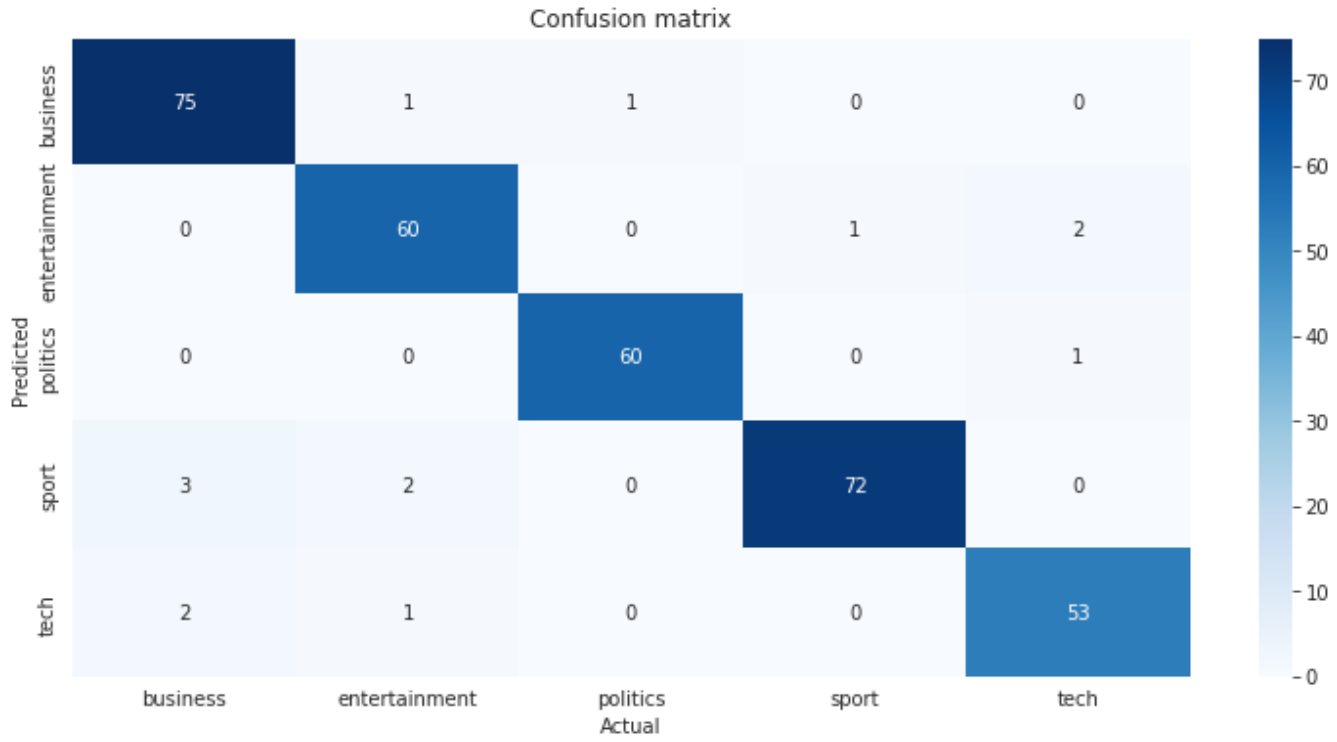
```
# Test accuracy
print("The test accuracy is: ")
print(accuracy_score(labels_test, knnc_pred))
```

```
The test accuracy is:
0.9580838323353293
```

```
# Classification report
print("Classification report")
print(classification_report(labels_test,knnc_pred))
```

```
Classification report
              precision    recall  f1-score   support

           0       0.94      0.97      0.96        77
           1       0.94      0.95      0.94        63
           2       0.98      0.98      0.98        61
           3       0.99      0.94      0.96        77
           4       0.95      0.95      0.95        56

    accuracy                           0.96       334
   macro avg       0.96      0.96      0.96       334
weighted avg       0.96      0.96      0.96       334
```

```
aux_df = df[['category', 'Category_Code']].drop_duplicates().sort_values('Category_Code')
conf_matrix = confusion_matrix(labels_test, knnc_pred)
plt.figure(figsize=(12.8,6))
sns.heatmap(conf_matrix,
            annot=True,
            xticklabels=aux_df['category'].values,
            yticklabels=aux_df['category'].values,
            cmap="Blues")
plt.ylabel('Predicted')
plt.xlabel('Actual')
plt.title('Confusion matrix')
plt.show()
```




```
base_model = KNeighborsClassifier()
base_model.fit(features_train, labels_train)
accuracy_score(labels_test, base_model.predict(features_test))
```

```
0.9491017964071856
```

```
best_knnc.fit(features_train, labels_train)
accuracy_score(labels_test, best_knnc.predict(features_test))
```

```
0.9580838323353293
```

```
d = {
    'Model': 'KNN',
    'Training Set Accuracy': accuracy_score(labels_train, best_knnc.predict(features_train))
    'Test Set Accuracy': accuracy_score(labels_test, knnc_pred)
}

df_models_knnc = pd.DataFrame(d, index=[0])
```

```
df_models_knnc
```

	Model	Training Set Accuracy	Test Set Accuracy
0	KNN	0.952935	0.958084

```
with open('NewModels/best_knnc.pickle', 'wb') as output:
```

```
    pickle.dump(best_knnc, output)

with open('NewModels/df_models_knnc.pickle', 'wb') as output:
    pickle.dump(df_models_knnc, output)
```

Model Selection

```
path_pickles = "NewModels/"

list_pickles = [
    "df_models_knnc.pickle",
    "df_models_rfc.pickle",
    "df_models_svc.pickle"
]

df_summary = pd.DataFrame()

for pickle_ in list_pickles:

    path = path_pickles + pickle_

    with open(path, 'rb') as data:
        df = pickle.load(data)

    df_summary = df_summary.append(df)

df_summary = df_summary.reset_index().drop('index', axis=1)
df_summary
```

	Model	Training Set Accuracy	Test Set Accuracy
0	KNN	0.952935	0.958084
1	Random Forest	1.000000	0.961078
2	SVM	0.958223	0.958084

```
df_summary.sort_values('Test Set Accuracy', ascending=False)
```

	Model	Training Set Accuracy	Test Set Accuracy
1	Random Forest	1.000000	0.961078
0	KNN	0.952935	0.958084
2	SVM	0.958223	0.958084

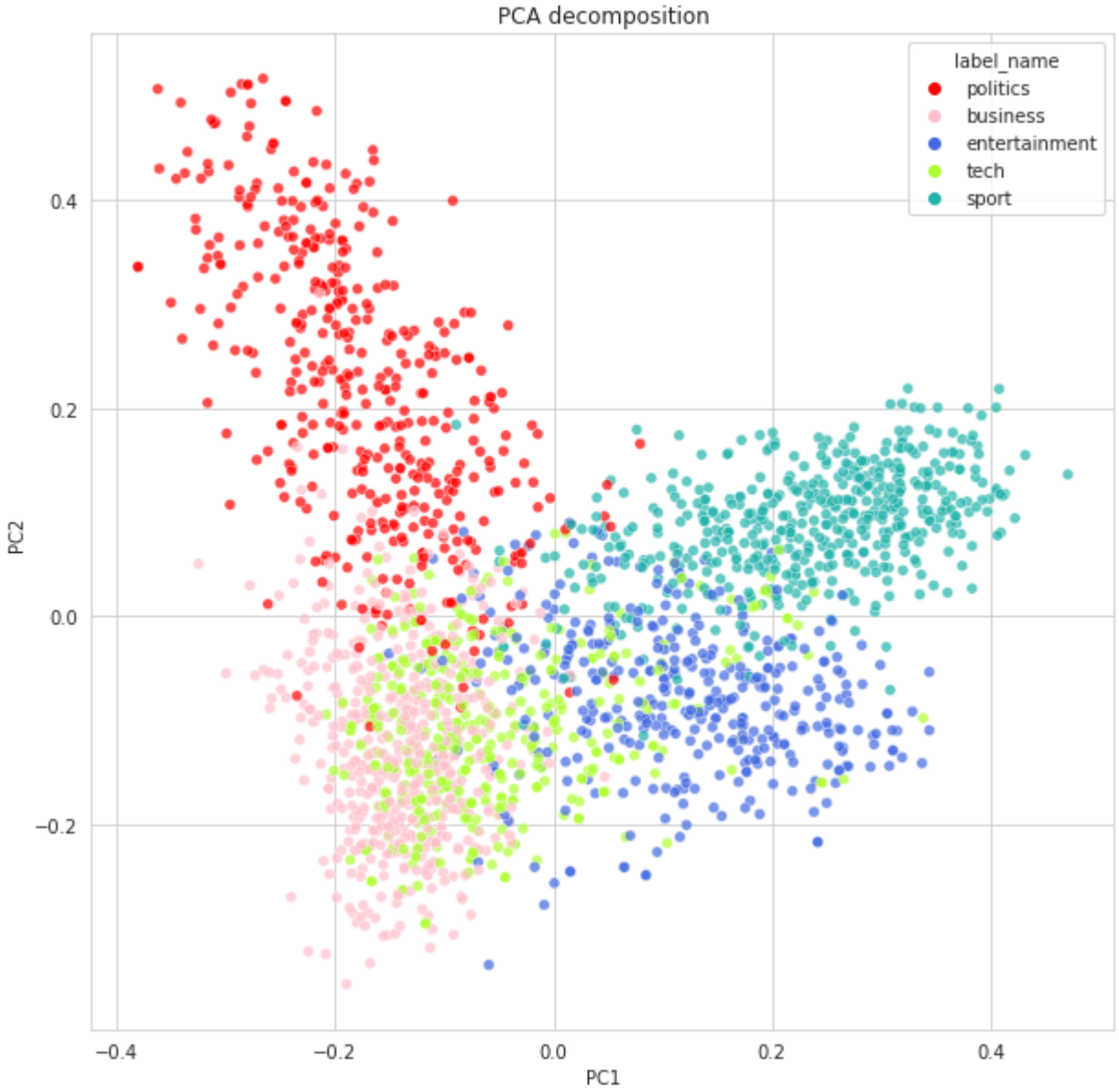
```
features = np.concatenate((features_train,features_test), axis=0)
```

```
labels = np.concatenate((labels_train,labels_test), axis=0)
print(features.shape)
print(labels.shape)
```

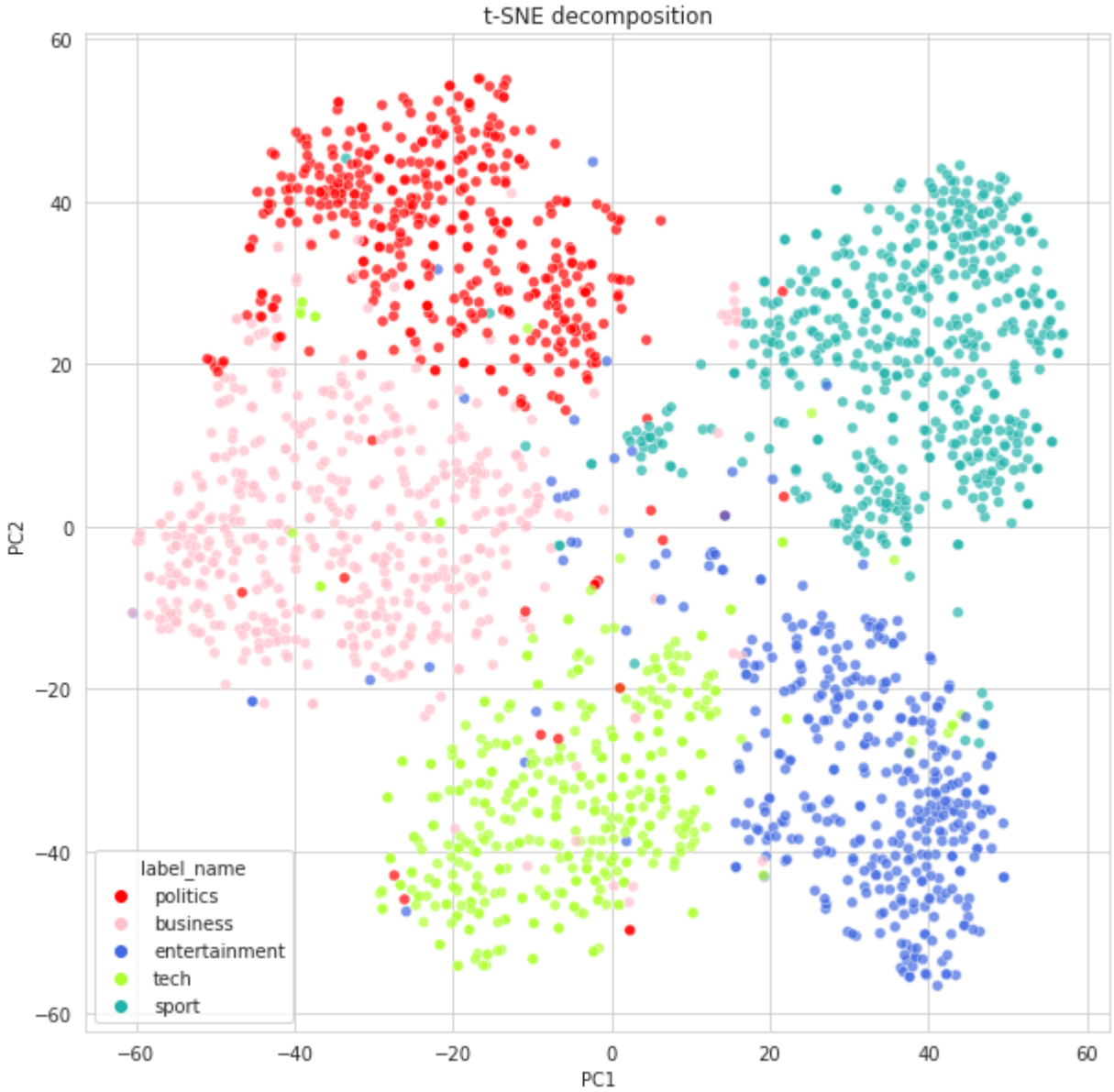
```
(2225, 300)
(2225,)
```

```
def plot_dim_red(model, features, labels, n_components=2):

    # Creation of the model
    if (model == 'PCA'):
        mod = PCA(n_components=n_components)
        title = "PCA decomposition"  # for the plot

    elif (model == 'TSNE'):
        mod = TSNE(n_components=2)
        title = "t-SNE decomposition"

    else:
        return "Error"

    # Fit and transform the features
    principal_components = mod.fit_transform(features)

    # Put them into a dataframe
    df_features = pd.DataFrame(data=principal_components,
                     columns=['PC1', 'PC2'])

    # Now we have to paste each row's label and its meaning
    # Convert labels array to df
    df_labels = pd.DataFrame(data=labels,
                             columns=['label'])

    df_full = pd.concat([df_features, df_labels], axis=1)
    df_full['label'] = df_full['label'].astype(str)

    # Get labels name
    category_names = {
        "0": 'business',
        "1": 'entertainment',
        "2": 'politics',
        "3": 'sport',
        "4": 'tech'
    }

    # And map labels
    df_full['label_name'] = df_full['label']
    df_full = df_full.replace({'label_name':category_names})

    # Plot
    plt.figure(figsize=(10,10))
    sns.scatterplot(x='PC1'
```

```
    sns.scatterplot(x="PC1",
                    y='PC2',
                    hue="label_name",
                    data=df_full,
                    palette=["red", "pink", "royalblue", "greenyellow", "lightseagreen"],
                    alpha=.7).set_title(title);


plot_dim_red("PCA",
             features=features,
             labels=labels,
             n_components=2)
```




```
plot_dim_red("TSNE",
             features=features,
             labels=labels,
             n_components=2)
```




```
# Dataframe
path_df = "NewPickles/df.pickle"
with open(path_df, 'rb') as data:
    df = pickle.load(data)

# SVM Model
path_model = "NewModels/best_knnc.pickle"
with open(path_model, 'rb') as data:
    knnc_model = pickle.load(data)

# Category mapping dictionary
category_codes = {
    'business': 0,
    'entertainment': 1,
    'politics': 2,
    'sport': 3,
    'tech': 4
}
```

```
category_names = {
    0: 'business',
    1: 'entertainment',
    2: 'politics',
    3: 'sport',
    4: 'tech'
}


predictions = knnc_model.predict(features_test)
# Indexes of the test set
index_X_test = X_test.index

print(index_X_test)


# We get them from the original df
df_test = df.loc[index_X_test]

# Add the predictions
df_test['Prediction'] = predictions

# Clean columns
df_test = df_test[['text', 'category', 'Category_Code', 'Prediction']]

# Decode
df_test['Category_Predicted'] = df_test['Prediction']
df_test = df_test.replace({'Category_Predicted':category_names})

# Clean columns again
df_test = df_test[['text', 'category', 'Category_Predicted']]
df_test.head()
```

```
Int64Index([1691, 1103,  477,  197,  475,  162,  887,  307, 1336, 1679,
            ...
            1567, 2130, 1216, 1135,  359,  393, 1746,  444, 2215,  733],
           dtype='int64', length=334)
```

	text	category	Category_Predicted
1691	moya sidesteps davis cup in 2005 carlos moya h...	sport	sport
1103	poll idols face first hurdles vote for me - i...	politics	politics
477	britons fed up with net service a survey condu...	tech	tech
197	lib dems predict best ever poll the lib dems...	politics	politics
475	prince crowned top music earner prince earne...	entertainment	entertainment

```
condition = (df_test['category'] != df_test['Category_Predicted'])

df_misclassified = df_test[condition]
```

```
df_misclassified.head(3)
```

	text	category	Category_Predicted
1144	mcdonald s to sponsor mtv show mcdonald s the...	business	entertainment
1565	ferdinand casts doubt over glazer rio ferdinan...	sport	business
535	pc ownership to double by 2010 the number of...	tech	business

```
def output_article(row_article):
    print('Actual Category: %s' %(row_article['category']))
    print('Predicted Category: %s' %(row_article['Category_Predicted']))
    print('-----------------------------------------')
    print('Text: ')
    print('%s' %(row_article['text']))


random.seed(8)
list_samples = random.sample(list(df_misclassified.index), 3)
list_samples
```

```
[1820, 1191, 1809]
```

```
output_article(df_misclassified.loc[list_samples[0]])
```

```
Actual Category: entertainment
Predicted Category: tech
-----------------------------------------
Text:
johnny and denise lose passport johnny vaughan and denise van outen s saturday night ent
```

```
output_article(df_misclassified.loc[list_samples[1]])
```

```
Actual Category: sport
Predicted Category: business
-----------------------------------------
Text:
jones files lawsuit against conte marion jones has filed a lawsuit for defamation agains
```

```
output_article(df_misclassified.loc[list_samples[2]])
```

```
Actual Category: business
Predicted Category: politics
-----------------------------------------
Text:
golden rule  intact  says ex-aide chancellor gordon brown will meet his golden economic
```

```
path_models = "NewModels/"

# SVM
path_svm = path_models + 'best_knnc.pickle'
with open(path_svm, 'rb') as data:
    knnc_model = pickle.load(data)

path_tfidf = "NewPickles/tfidf.pickle"
with open(path_tfidf, 'rb') as data:
    tfidf = pickle.load(data)

category_codes = {
    'business': 0,
    'entertainment': 1,
    'politics': 2,
    'sport': 3,
    'tech': 4
}


punctuation_signs = list("?:!.,;")
stop_words = list(stopwords.words('english'))

def create_features_from_text(text):

    # Dataframe creation
    lemmatized_text_list = []
    df = pd.DataFrame(columns=['text'])
    df.loc[0] = text
    df['Content_Parsed_1'] = df['text'].str.replace("\r", " ")
    df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace("\n", " ")
    df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace("    ", " ")
    df['Content_Parsed_1'] = df['Content_Parsed_1'].str.replace('"', '')
    df['Content_Parsed_2'] = df['Content_Parsed_1'].str.lower()
    df['Content_Parsed_3'] = df['Content_Parsed_2']
    for punct_sign in punctuation_signs:
        df['Content_Parsed_3'] = df['Content_Parsed_3'].str.replace(punct_sign, '')
    df['Content_Parsed_4'] = df['Content_Parsed_3'].str.replace("'s", "")
    wordnet_lemmatizer = WordNetLemmatizer()
    lemmatized_list = []
    text = df.loc[0]['Content_Parsed_4']
    text_words = text.split(" ")
    for word in text_words:
        lemmatized_list.append(wordnet_lemmatizer.lemmatize(word, pos="v"))
    lemmatized_text = " ".join(lemmatized_list)
    lemmatized_text_list.append(lemmatized_text)
    df['Content_Parsed_5'] = lemmatized_text_list
    df['Content_Parsed_6'] = df['Content_Parsed_5']
    for stop_word in stop_words:
        regex_stopword = r"\b" + stop_word + r"\b"
        df['Content_Parsed_6'] = df['Content_Parsed_6'].str.replace(regex_stopword, '')
```

```
        df = df['Content_Parsed_6']
        df = df.rename({'Content_Parsed_6': 'Content_Parsed'})

        # TF-IDF
        features = tfidf.transform(df).toarray()

        return features


    def get_category_name(category_id):
        for category, id_ in category_codes.items():
            if id_ == category_id:
                return category


    def predict_from_text(text):

        # Predict using the input model
        prediction_svc = svc_model.predict(create_features_from_text(text))[0]
        prediction_svc_proba = svc_model.predict_proba(create_features_from_text(text))[0]

        # Return result
        category_svc = get_category_name(prediction_svc)

        print("The predicted category using the SVM model is %s." %(category_svc) )
        print("The conditional probability is: %a" %(prediction_svc_proba.max()*100))


    text = """ The center-right party Ciudadanos closed a deal on Wednesday with the support of t

    The move would see the Socialist Party lose power in the region for the first time in 36 year

    On Thursday, Marta Bosquet of Ciudadanos was voted in as the new speaker of the Andalusian pa

    The speaker’s role in the parliament is key for the calling of an investiture vote and for th

    Officially, the talks as to the make up of a future government have yet to start, but in real

    The speaker’s role in the parliament is key for the calling of an investiture vote and for th

    The PP, which was ousted from power by the PSOE in the national government in June, is keen t

    Wednesday was a day of intense talks among the parties in a bid to find a solution that would

    The PSOE, meanwhile, argues that having won the elections with a seven-seat lead over the PP



    """

    predict_from_text(text)
```

```
The predicted category using the SVM model is business.
The conditional probability is: 55.09999028372485
```

```
# Politics

text = """Disputes have already broken out within the new political alliance that is working

Just hours after the far-right Vox agreed to support the Popular Party (PP)’s candidate to he

These early clashes suggest it could be difficult to export the model to other parts of Spain

The PP and the liberal Ciudadanos have reached their own governing agreement in the wake of a

Ciudadanos has refused point-blank to meet with Vox representatives, but the PP has struck it

On Friday morning, Juan Marín of Ciudadanos said that there are no plans for a separate famil

The reform party has insisted that the Vox-PP deal does not affect them at all, and Ciudadano

 Vox national leader Santiago Abascal (c) and Andalusian leader Francisco Serrano (r).
Vox national leader Santiago Abascal (c) and Andalusian leader Francisco Serrano (r). REUTERS
But Vox insists on a family department, and said it will expect loyalty from the PP on this i

These early clashes suggest it could be difficult to export the model to other parts of Spain

The PP is anxious to win back power in regions like Valencia, the Balearic Islands, Castilla-

Parliamentary debate
The PSOE has already digested the fact that it is losing its hold on Spain’s most populated r

The Socialists will not be putting forward a candidate, now that the PP nominee has enough su

The sum of the PP, Ciudadanos and Vox votes is four above the 55 required for a majority. The

"""

predict_from_text(text)
```

```
The predicted category using the SVM model is politics.
The conditional probability is: 66.28189176348307
```

```
# Entertainment

text = """
Cádiz is in style: it has just been included in The New York Times’ list of 52 Places to Go i

The journalist Andrew Ferren, who wrote about Cádiz for The New York Times’ list, lives in Sp

“Despite the fact that Cádiz was historically a major maritime link between America and Europ

Culinary delights
```

Aponiente restaurant in El Puerto de Santa María.
Aponiente restaurant in El Puerto de Santa María.
Suggestions include the new Western-style gastrobar Saja River, recently opened on Santa Elen

To these suggestions, EL VIAJERO adds several of its own, including Restaurante Café Royalty,

Jerez de la Frontera and its wineries
Bodegas Lustau, en Jerez de la Frontera (Cádiz).ampliar foto
Bodegas Lustau, en Jerez de la Frontera (Cádiz). NEIL FARRIN GETTY IMAGES
Around 36 km to the north of Cádiz lies Jerez de la Frontera, known for the fortified wines k

The NMAC Montenmedio Foundation
Vejer de la Frontera.ampliar foto
Vejer de la Frontera. GETTY IMAGES
The NMAC Montenmedio Foundation of contemporary art sits between Barbate and Vejer de la Fron

EL VIAJERO expands on Ferren’s recommendations with a few of its own:

1.The Cádiz Carnival
The Cádiz carnival.ampliar foto
The Cádiz carnival.
An unique and fun festival that takes place from February 28 to March 10. In fact it is so un

2. Barrio del Pópulo
The Pópulo neighborhood.ampliar foto
The Pópulo neighborhood. RAQUEL M. CARBONELL GETTY
This is the oldest neighborhood in Cádiz and features an old Roman theater, the old cathedral

3. Cádiz à la Havana
Cathedral square in Cádiz.ampliar foto
Cathedral square in Cádiz. RAQUEL M. CARBONELL GETTY
Stroll from the colonial-style Mina Square, with its ficus and palm trees, to the Provincial

4. A wealth of history
Baelo Claudia Roman site in Tarifa (Cádiz).ampliar foto
Baelo Claudia Roman site in Tarifa (Cádiz). KEN WELSH GETTY
Standing on the frontier between two continents, the province of Cádiz has a long and action-

5. Sanlúcar de Barrameda
Summer beach horse races in Sanlúcar de Barrameda.ampliar foto
Summer beach horse races in Sanlúcar de Barrameda. JUAN CARLOS TORO
Famous for its summer horse racing on the beach as well as for its wineries, this coastal tow

6. Coast and mountains
Olvera, a white village in Cádiz.ampliar foto
Olvera, a white village in Cádiz. RUDI SEBASTIAN GETTY
Cádiz has miles of windswept beaches that make it a perfect haunt for surfers of various desc

7. The flamenco route
Located in San Fernando, the Peña Flamenca Camarón de la Isla, named after the famous singer,

```
8. Conil de la Frontera
 The beach in Conil de la Frontera.ampliar foto
The beach in Conil de la Frontera. GETTY IMAGES
There are three national parks that stretch along Cádiz’s Atlantic coast – La Breña, Los Alco

9. Surfing in Tarifa
In the inlets of Los Lances and Valdevaqueros in Tarifa, wind and kitesurfers can skid across

10. The white villages
Nineteen districts in the Cádiz mountains take you through a string of white villages – Alcal

"""

predict_from_text(text)
```

```
     The predicted category using the SVM model is politics.
     The conditional probability is: 51.108910898771256
```

```
# Business

text = """
Vodafone España has informed representatives of its employees that it is putting a collective

“In the current market climate, demand for services continues to grow exponentially, but this

Vodafone added that the current expectations of clients, “who demand an agile, simple and imm

As such, the company continued, it is looking to “reverse the negative trend of the business,

The operator says that it is sure it can reach a deal with labor unions so that the measures

Vodafone has suffered a great deal in the trade war that was sparked by its rivals Movistar a

In the first three quarters of 2018, Vodafone has lost 361,000 cellphone lines (70,000 of whi

The operator executed a similar collective dismissal plan (known in Spanish as an ERE) in 201

Before the acquisition of ONO, Vodafone also executed an ERE in 2013. On that occasion, the c

"""

predict_from_text(text)
```

```
     The predicted category using the SVM model is business.
     The conditional probability is: 47.16858388488912
```

```
# Tech

text = """
Elon Musk told the world in late 2017 that Tesla was taking its automotive know-how and apply
```

```
The German automaker also committed to manufacturing the truck this summer, with deliveries s


While there are a few Tesla Semi prototypes on the road now, and a dozen or so big name compa

DAIMLER FIRST SHOWED OFF A PROTOTYPE IN 2015
This has left the door wide open for companies like Daimler, the parent company of Mercedes-B

The new Cascadia is not much more advanced than the prototype was in 2015. In fact, the techn


The Freightliner Inspiration Truck at the event in 2015.
But the new Cascadia is doing this with a limited set of sensors. There’s a forward-facing ca

This helps keep costs down, but means the technology is more in line with what you’d find pow

DAIMLER’S TRUCK HAS MORE IN COMMON WITH NISSAN’S PROPILOT SYSTEM THAN TESLA’S AUTOPILOT
Keeping with a theme of less is more, there’s also no camera-based monitoring system in the t

A sensor in the steering column measures resistance applied to the steering wheel. If the dri



The new Cascadia is a far cry from a fully autonomous truck, but based on my brief ride, Daim

A Daimler representative also told me that, while lane centering is on, the driver can even c

RELATED

This is what it’s like to ride in Daimler’s self-driving semi truck
Daimler promised some other modern technologies are coming the new Cascadia, though none of i

The Cascadia won’t be as stuffed with tech as the Tesla Semi, nor is it as sleek. But it will
"""

predict_from_text(text)

     The predicted category using the SVM model is business.
     The conditional probability is: 67.73745032912375


# Sports

text = """
Spain has agreed to host the soccer final of the Copa Libertadores between Argentina teams Ri

The final in Madrid is a punch in the soul to all fans of soccer in Argentina

ONLINE SPORTS DAILY OLE

The final was set to take place in Argentina but was suspended twice after fans turned violen
```

```
In view of the insecurity, the South American Football Confederation (Conmebol), which organi

 Embedded video

Sebastián Lisiecki
@sebalisiecki
 Así fue la llegada de Boca al Monumental. Pésimo la seguridad q los mete entre toda la gente

575
7:23 PM - Nov 24, 2018
637 people are talking about this
Twitter Ads info and privacy
This was how Boca arrived at Monumental stadium. The security that got between the all people

This is the first time a Copa Libertadores game has been played outside the Americas since th

But the feeling in Argentina has been less optimistic. The national newspaper La Nación wrote

Security risk
In a message on Twitter, Sánchez promised that "security forces have extensive experience of

River and Boca have a long-standing rivalry fueled largely by the class divide between the te

Scheduling issues
The final will take place on Sunday, December 9, on the final day of a three-day national hol

 Conmebol president Alejandro Domínguez on Tuesday.
Conmebol president Alejandro Domínguez on Tuesday.
Many details about the game have yet to be revealed, including how tickets will be sold, what

Conmebol and soccer club representatives began considering destinations for the match on Tues



"""

predict_from_text(text)

     The predicted category using the SVM model is tech.
     The conditional probability is: 59.62510649552267


# Weather

text = """
A polar air mass that entered the Iberian peninsula on Wednesday has already caused sharp dro

“An episode of intense cold” is forecast for Friday, when the mercury will continue to plumme

Elsewhere, weather stations have recorded -8.2ºC in La Molina (Girona), at an elevation of 1,
```

```
 Almería has rolled out vehicles to deal with wintry road conditions.
Almería has rolled out vehicles to deal with wintry road conditions. DIPUTACIÓN DE ALMERÍA EU
Aemet spokesman Rubén del Campo said that the cold spell is not out of the ordinary for a mon

Temperatures have already dipped between six and eight degrees in a matter of hours in some p

Temperatures on Friday and Saturday will be “very cold, with lows of five to 10 degrees below

No snow
However, little to no snow is expected “not for lack of cold, but for lack of precipitation,

Alerts are in place in Almería, Granada, Jaén, Aragón, Cantabria, Castilla-La Mancha, Castill

On Saturday, the orange warnings will extend to Córdoba, Salamanca, Valladolid, Galicia and L


"""

predict_from_text(text)

     The predicted category using the SVM model is entertainment.
     The conditional probability is: 71.28419601310195


# Health

text = """
The obesity epidemic has been on the rise for years, with cases nearly tripling since 1975, a

An investigation by the Mar de Barcelona hospital has found that 80% of men and 55% of women

Being overweight can mean a higher risk of suffering a number of diseases, including diabetes

The study, published in the Spanish Cardiology Magazine, points out that this epidemic will m

The issue, the experts state, is not an esthetic one, but rather a question of health. Being

Researchers at the Barcelona hospital revised all of the scientific literature published in S

There are currently 25 million people with excess weight, three million more than a decade ag

DR ALBERT GODAY, AUTHOR OF THE STUDY

“There are currently 25 million people with excess weight, three million more than a decade a

“In men, excess weight is more usual up to the age of 50,” explains Goday. “From 50 onward, o

The experts argue that any weight loss, no matter how small, reduces the risk of contracting

"""

predict_from_text(text)
```

```
     The predicted category using the SVM model is business.
     The conditional probability is: 78.04540552025212


# Animal abuse

text = """
Spain’s animal rights party PACMA posted a 38-second video on Twitter on Friday showing a man

“Hunters shut what appears to be a fox in a cage and let it out only to pepper it with bullet

 Video insertado

PACMA
✓
@PartidoPACMA
 Cazadores enjaulan a lo que parece ser un zorro y lo liberan solo para acribillarlo a tiros.

En realidad, son peligrosos psicópatas con escopeta y permiso de amas. #YoNoDisparo

4.188
10:43 - 4 ene. 2019
7.443 personas están hablando de esto
Información y privacidad de Twitter Ads
At the start of the video, a man teases the caged animal with a stick. When the cage door is

The release of the video, which has had 255,000 views, coincided with the launch of PACMA’s c

As it notes on its website, PACMA is the only political group that opposes hunting, and it is

No animal should die under fire. We will fight tirelessly until hunting becomes a crime

PACMA

The animal rights group is preparing a report to send to the regional government of Galicia a

Last month, a Spanish hunter who was filmed while he chased and tortured a fox was identified

And in November, animal rights groups and political parties reacted with indignation over a v

"""

predict_from_text(text)

     The predicted category using the SVM model is business.
     The conditional probability is: 64.06761565559422
```

✓ 0s completed at 5:38 PM