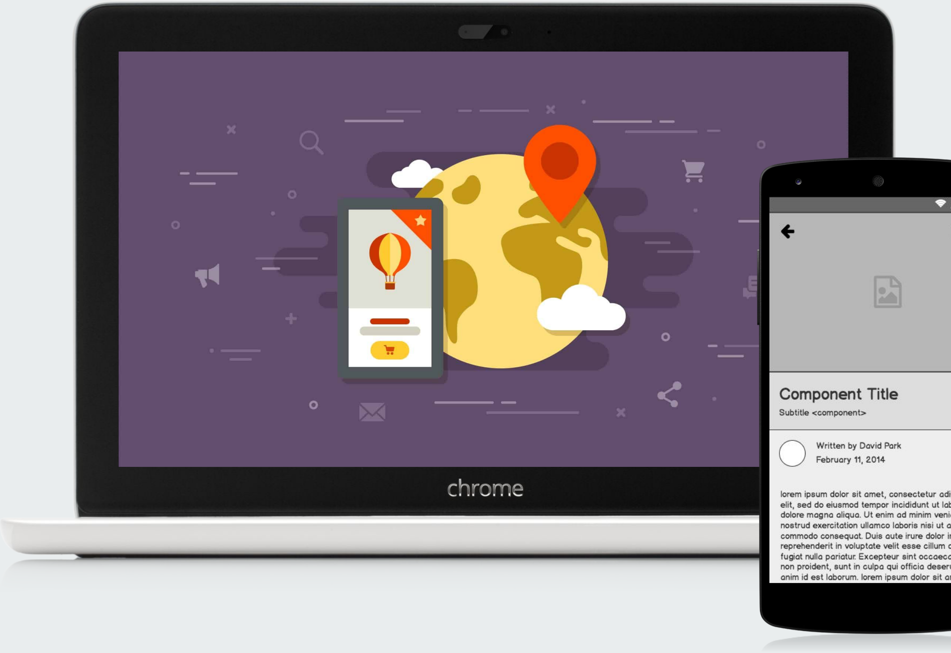
Negative Review Detection

Shashank G S (PES2201800706)
Ajitesh Nair (PES2201800681)
G Ghana Gokul (PES2201800077)



Outline

The Problem

The Literature Survey

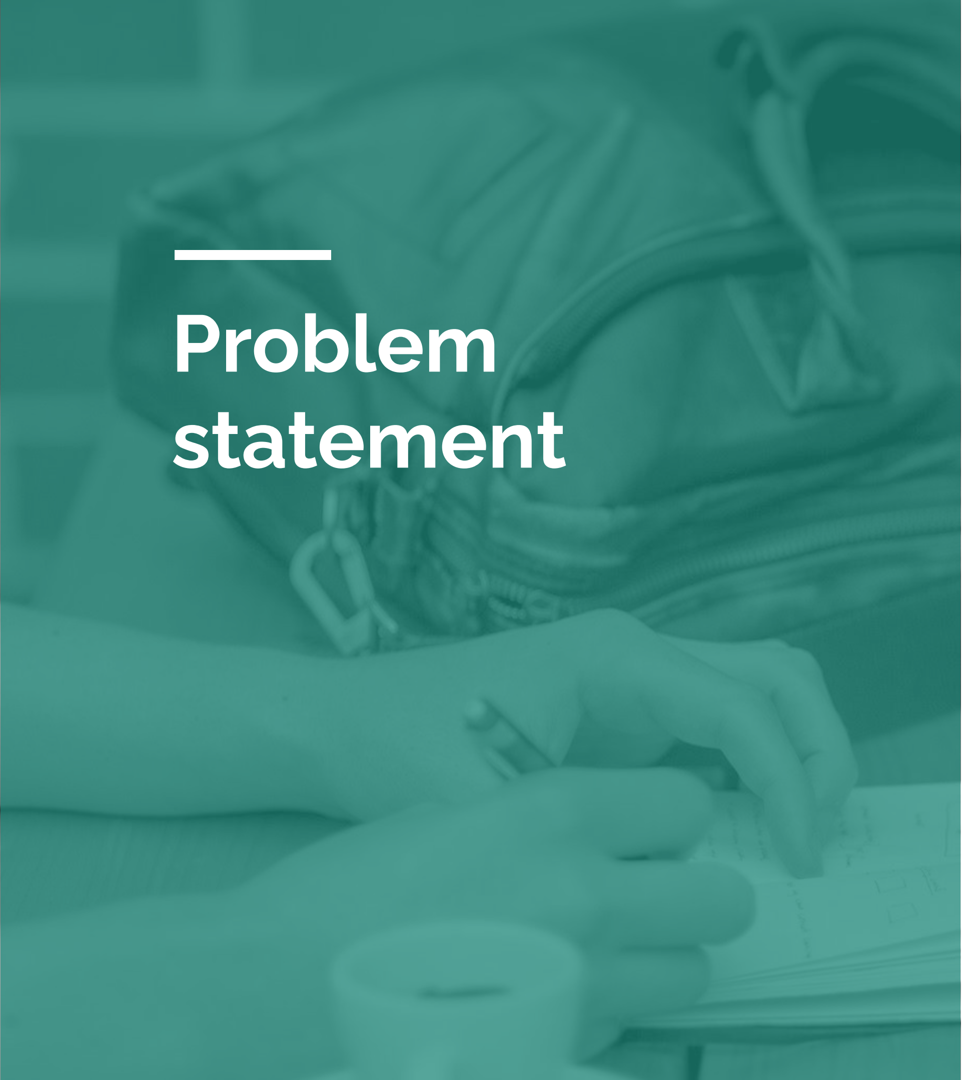
Solution Proposal

Steps Followed

Next Steps

The Problem

Sentiment analysis is part of the Natural Language Processing (NLP) techniques that consists in extracting emotions related to some raw texts. This is usually used on social media posts and customer reviews in order to automatically understand if some users are positive or negative and why. The goal of this study is to show how sentiment analysis can be performed using python and detect the negative reviews given by the user.



Problem statement

Negative Review Detection On Hotel Reviews Dataset.

We used some 515k hotel reviews data present on kaggle. Each observation consists in one customer review for one hotel. Each customer review is composed of a textual feedback of the customer’s experience at the hotel and an overall rating.

What customers do today

Purpose of choosing this problem statement.

Negative reviews succeed in chasing away customers from your **business** to your competitors. Research shows that one **negative review** drives away 22% of prospects, around 30 customers.So using this we can determine the faults and also check what customers are expecting and meet their requirements.

Literature Survey

Literature Survey

01

Deep Convolutional Neural Networks for Twitter Sentiment Analysis

ZHAO JIANQIANG1,2 , GUI XIAOLIN1,2, AND ZHANG XUEJUN2,3

Merits :

- Comparing to the deep convolutional neural network and the baseline method for Twitter sentiment classification algorithm, the results indicate that their method has obvious advantages for the given Datasets.

Limitation :

- Approach concatenates the pre-trained word embeddings feature generated using the GloVe word sentiment polarity features based sentiment lexicon.

In this paper, they introduce a word embeddings method obtained by unsupervised learning based on large twitter corpora.

The goal of Twitter sentiment classification is to automatically determine whether a tweet's sentiment polarity is negative or positive.

A model called GloVe-CNN is presented which implements the binary task of classifying the tweet into negative or positive sentiment categories.

The experimental results clearly indicate that the GloVe-DCNN model can obtain a good performance of the sentiment classification.

Literature Survey

02

Emotion Detection from Text via Ensemble Classification Using Word Embeddings

Jonathan Herzig, Michal Shmueli-Scheuer, David Konopnicki
(IBM Research - Haifa)

Merits :

- Ensemble methods outperforms other baseline models.

Limitation :

- Time taken for training and prediction is high because all the members of the ensemble have to be individually trained.
- Even number of classifiers may lead to a tie in the predicted class.

Ensembles tend to achieve better results when there is a significant diversity among the classifiers. Thus, they have utilized classifiers that are based on different document representations, to form an ensemble model.

Each classifier makes use of SVM for classification. Each classifier makes use of one of the two kernels, Linear and RBF. Document representations used are:

-BOW Classifier
-Word Embedding-Based Classifier
    -CBOW (Continuous Bag of Words)
    -TF-IDF weights
    -Classifier weights (CLASS)

The ensemble classifier showed an average relative improvement of 11.6% in the accuracy of classification.

Literature Survey

03

A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction

FARKHUND IQBAL , JAHANZEB MAQBOOL HASHMI, BENJAMIN C. M. FUNG (Senior Member, IEEE), PATRICK C. K. HUNG)

Merits :

- Reduces size of feature vectors. Hence can be used for large datasets.
- Increased accuracy.

Limitation :

- 70% of execution time spent on reducing size of feature vectors.

The authors of this paper aims to reduce the size of the feature vectors by using Genetic Algorithms since the size of the feature vectors grows with the size of the dataset.

Several classifiers like J48, NB, PART, SMO, KNN, JRip were used to test the improvement using GA optimized feature vector.

This approach resulted in 36% - 42% reduced feature size. NB classifier had accuracy of about 80% while using the GA based optimal feature selection on Twitter and reviews dataset. GA based feature reduction showed up to 15.4% increased accuracy over PCA and up to 40.2% increased accuracy over LSA.

Literature Survey

04

Opinion mining and sentiment analysis on online customer review

SANTOSH KUMAR,JHARNA MAJUNDAR

Merits :

- Comparison of SentiwordNet with Naive bayes,logistic regression.

Limitation :

- Only three classification models are applied.

In this paper, they focus on amazon product review mining.

The goal is to determine whether a sentiment is positive or negative.

Three classification models are applied on features of each review after the removal of stop words.

The experimental results clearly indicate that the Naive Bayes classifier performs better than other two.

Literature Survey

05

Classification of sentimental reviews using machine learning techniques

Merits :

- Use of vectorization techniques to convert textual data to numerical data.
- Cross validation technique is applied to choose the training and testing the data.

Limitation :

- Classifiers limited to SVM and Naive Bayes.

In this paper they have done classification of sentiment reviews using machine learning techniques on labeled polarity movie dataset,which consist of 1000 positive,negative reviews.

Methodology

1.Preparing of dataset

2.Cleaning reviews,removing stop words

3.The vectorization of features by CountVectorizer and TF-IDF.

4.The numeric vectors given as input to the classifiers.

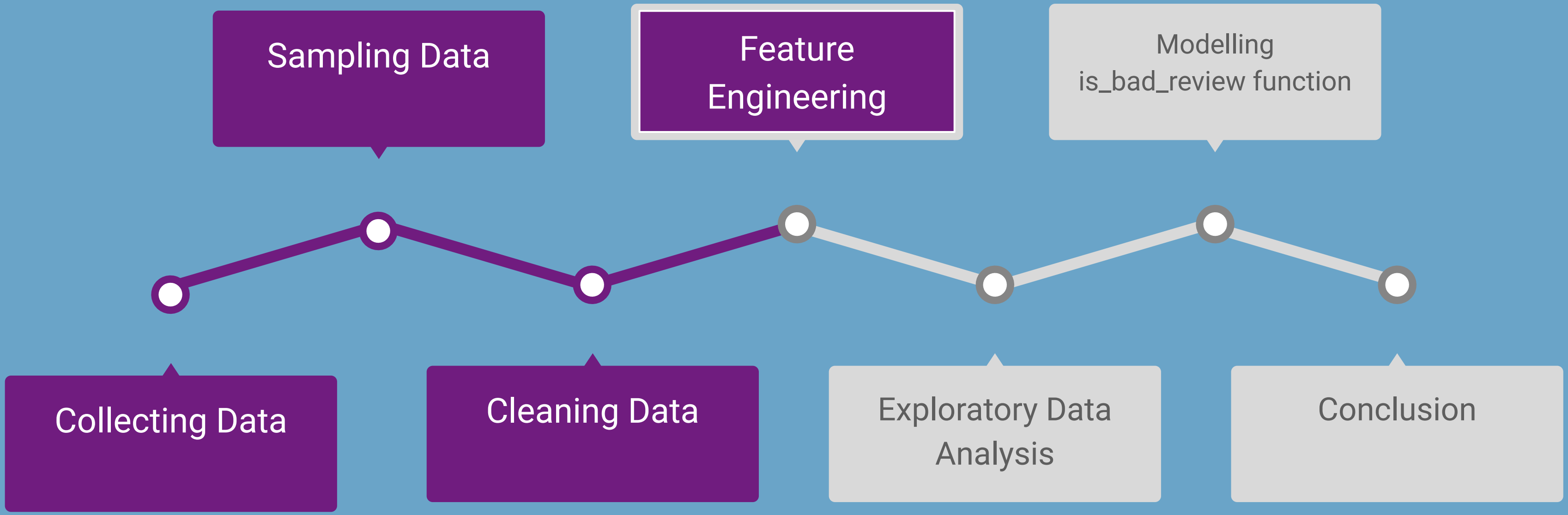
5.Confusion matrix is generated for each classifier.

6.Performance metrics were calculated using above confusion matrix,they considered a average metrics of 10-folds.

- The Goal is to compare results with Other literature results.
- SVM classifier gave best accuracy.

Project Description

Solution Proposal



Collecting Data

Sampling Data

Cleaning Data

Feature Engineering

Exploratory Data Analysis

Modelling is_bad_review function

Conclusion

Solution description

For each textual review, we want to predict if it corresponds to a good review (the customer is happy) or to a bad one (the customer is not satisfied). The reviews overall ratings can range from 2.5/10 to 10/10. In order to simplify the problem we will split those into two categories:

- bad reviews have overall ratings < 5
- good reviews have overall ratings >= 5

The challenge here is to be able to predict this information using only the raw textual data from the review.

Steps Followed

Load data

We firstly started by loading data.Each textual data is splitted into positive and negative parts.We grouped them together.

Sampling

We sample the data to reduce the computation

Cleaning

This step contains various method to clean the text data using various operations.

Featuring

We first start by adding sentiment analysis features.

Extracting

This step consist of extracting the vector representation for each review.

Next Steps

EXPLORATORY DATA ANALYSIS

1 TO HAVE A BETTER UNDERSTANDING OF OUR DATA LET'S EXPLORE IT A LITTLE.

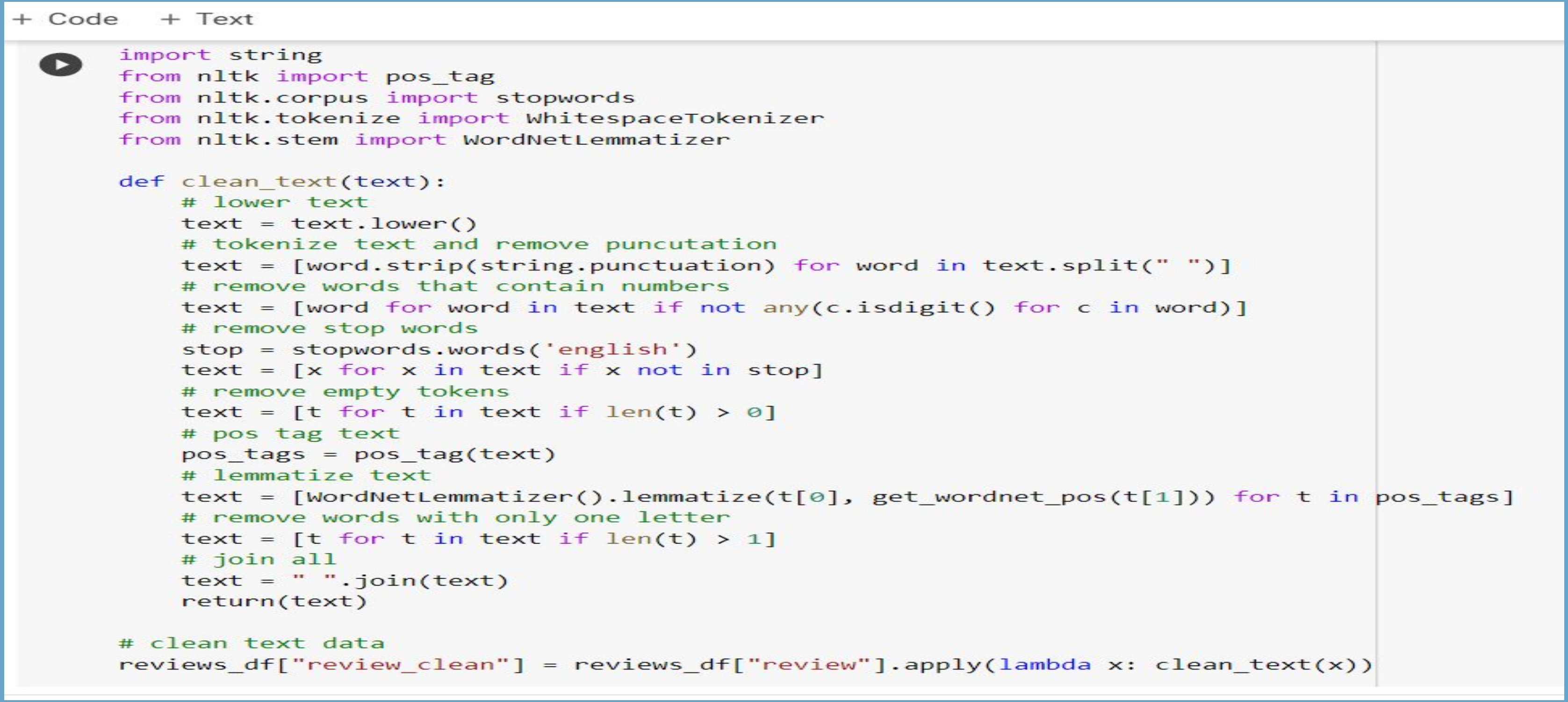
MODELLING THE FUNCTION

2 SPLIT THE DATA AS TEST AND TRAIN AND PERFORM THE CLASSIFICATION OPERATION.

CONCLUSION

3 GET THE ACCURACY AND COMPARE WITH THE DATA AND CONCLUDE.

Cleaning the data

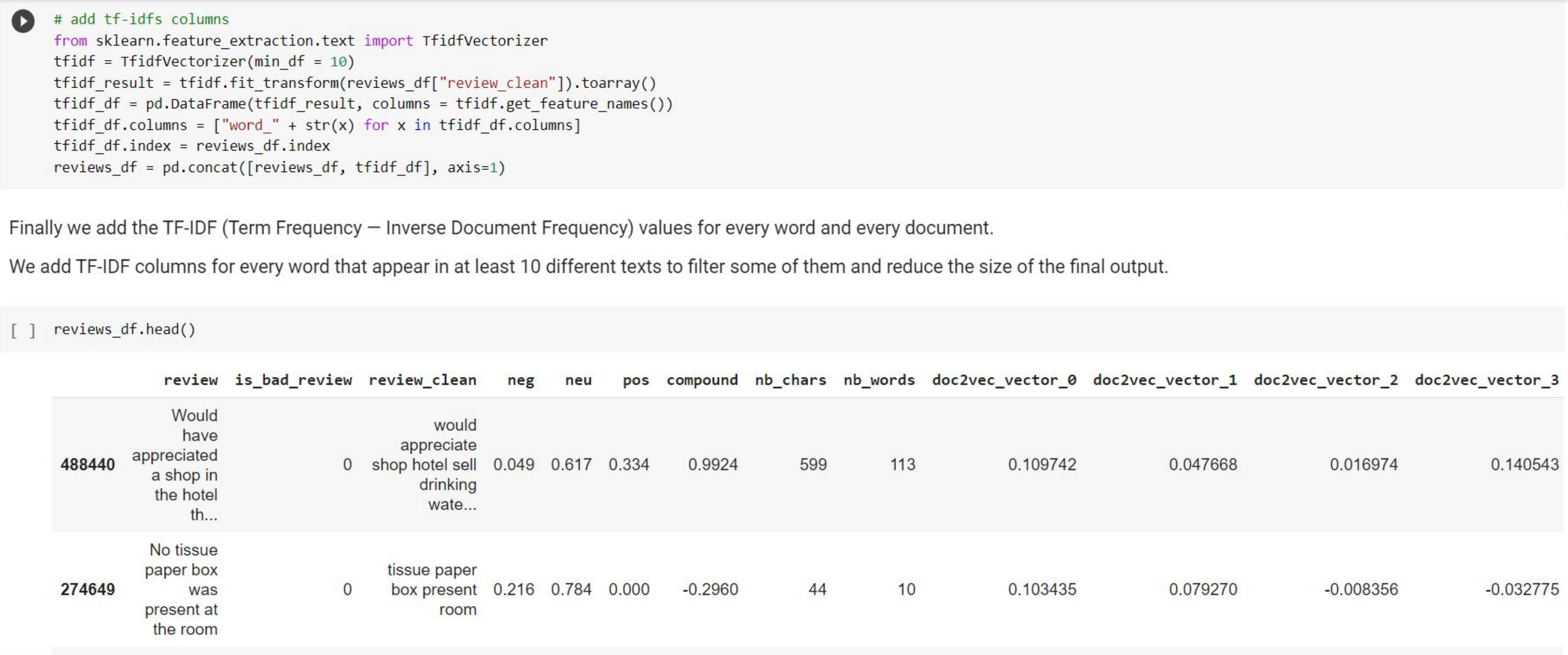


```
import string
from nltk import pos_tag
from nltk.corpus import stopwords
from nltk.tokenize import WhitespaceTokenizer
from nltk.stem import WordNetLemmatizer

def clean_text(text):
    # lower text
    text = text.lower()
    # tokenize text and remove puncutation
    text = [word.strip(string.punctuation) for word in text.split(" ")]
    # remove words that contain numbers
    text = [word for word in text if not any(c.isdigit() for c in word)]
    # remove stop words
    stop = stopwords.words('english')
    text = [x for x in text if x not in stop]
    # remove empty tokens
    text = [t for t in text if len(t) > 0]
    # pos tag text
    pos_tags = pos_tag(text)
    # lemmatize text
    text = [WordNetLemmatizer().lemmatize(t[0], get_wordnet_pos(t[1])) for t in pos_tags]
    # remove words with only one letter
    text = [t for t in text if len(t) > 1]
    # join all
    text = " ".join(text)
    return(text)

# clean text data
reviews_df["review_clean"] = reviews_df["review"].apply(lambda x: clean_text(x))
```

After finding tf-idf



```python
# add tf-idfs columns
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(min_df = 10)
tfidf_result = tfidf.fit_transform(reviews_df["review_clean"]).toarray()
tfidf_df = pd.DataFrame(tfidf_result, columns = tfidf.get_feature_names())
tfidf_df.columns = ["word_" + str(x) for x in tfidf_df.columns]
tfidf_df.index = reviews_df.index
reviews_df = pd.concat([reviews_df, tfidf_df], axis=1)
```

Finally we add the TF-IDF (Term Frequency — Inverse Document Frequency) values for every word and every document.

We add TF-IDF columns for every word that appear in at least 10 different texts to filter some of them and reduce the size of the final output.

```
[ ]  reviews_df.head()
```

	review	is_bad_review	review_clean	neg	neu	pos	compound	nb_chars	nb_words	doc2vec_vector_0	doc2vec_vector_1	doc2vec_vector_2	doc2vec_vector_3
488440	Would have appreciated a shop in the hotel th...	0	would appreciate shop hotel sell drinking wate...	0.049	0.617	0.334	0.9924	599	113	0.109742	0.047668	0.016974	0.140543
274649	No tissue paper box was present at the room	0	tissue paper box present room	0.216	0.784	0.000	-0.2960	44	10	0.103435	0.079270	-0.008356	-0.032775

Questions?

References

Deep Convolutional Neural Networks for Twitter Sentiment Analysis

ZHAO JIANQIANG1,2 , GUI XIAOLIN1,2, AND ZHANG XUEJUN2,3

Emotion Detection from Text via Ensemble Classification Using Word Embeddings

Jonathan Herzig, Michal Shmueli-Scheuer, David Konopnicki (IBM Research - Haifa)

A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction

FARKHUND IQBAL , JAHANZEB MAQBOOL HASHMI, BENJAMIN C. M. FUNG (Senior Member, IEEE), PATRICK C. K. HUNG

THANK YOU