Project Name: Text-Classification and ML-Model-Interpretation with Spacy-Eli5-Sklearn

Table of Contents

Demo

```
In [7]: stopwords
Out[7]: ['have',
         'already',
         'latter',
         'above',
         'we',
         'most',
         'as',
         'namely',
         'it',
         'during',
         'further',
         'n't',
         'and',
         'everywhere',
         'hereafter',
         'throughout',
         'will',
         'up',
         'whence',
         'upon'
```

```
In [26]: X_train
Out[26]: 2226    Verizon's bills, however, are difficult to und...
         1736    The opening sequence of this gem is a classic,...
         1018    This if the first movie I've given a 10 to in ...
         891     Pricing is a bit of a concern at Mellow Mushroom.
         2206                 We would recommend these to others.
                                      ...
         1638    End of Days is one of the worst big-budget act...
         1095                          I hate movies like that.
         1130                    There were too many close ups.
         1294    I really hope the team behind this movie makes...
         860     Talk about great customer service of course we...
         Name: Message, Length: 2196, dtype: object
```

```
In [27]: X_train.shape
Out[27]: (2196,)
```

In [35]:
```
# Prediction Results
# 1 = Positive review
# 0 = Negative review
for (sample,pred) in zip(X_test,sample_prediction):
    print(sample,"Prediction=>",pred)
```

```
Great pork sandwich. Prediction=> 1
It is a true classic.   Prediction=> 0
It's close to my house, it's low-key, non-fancy, affordable prices, good food. Prediction=> 1
Audio Quality is poor, very poor. Prediction=> 0
We loved the biscuits!!! Prediction=> 1
I don't have very many words to say about this place, but it does everything pretty well. Prediction=> 0
Was not happy. Prediction=> 1
The headsets are easy to use and everyone loves them. Prediction=> 1
I miss it and wish they had one in Philadelphia! Prediction=> 0
Still it's quite interesting and entertaining to follow.   Prediction=> 1
All three broke within two months of use. Prediction=> 0
Oh yeah, and the storyline was pathetic too.   Prediction=> 0
IT'S REALLY EASY. Prediction=> 1
Every element of this story was so over the top, excessively phony and contrived that it was painful to sit through.   Predic
tion=> 0
The food was outstanding and the prices were very reasonable. Prediction=> 1
I am so tired of clichés that is just lazy writing, and here they come in thick and fast.   Prediction=> 1
Ordered an appetizer and took 40 minutes and then the pizza another 10 minutes. Prediction=> 0
I would highly recommend this. Prediction=> 1
I'm still trying to get over how bad it was.   Prediction=> 0
```

In [43]:
```
exp.show_prediction()
```

Out[43]: **y=1** (probability **0.840**, score **1.655**) top features

Contribution?	Feature
+1.879	great
+0.210	great pork
+0.150	pork sandwich
-0.157	<BIAS>
-0.184	pork
-0.241	sandwich

In [44]:
```
exp.show_prediction(target_names=target_names)
```

Out[44]: **y=Positive** (probability **0.840**, score **1.655**) top features

Contribution?	Feature
+1.879	great
+0.210	great pork
+0.150	pork sandwich
-0.157	<BIAS>
-0.184	pork
-0.241	sandwich

Overview

This is diving into Text Classification with Interpretation of ML Models Concept.

Text classification also known as text tagging or text categorization is the process of categorizing text into organized groups. Text classification algorithms are at the heart of a variety of software systems that process text data at scale. Email software uses text classification to determine whether incoming mail is sent to the inbox or filtered into the spam folder. Discussion forums use text classification to determine whether comments should be flagged as inappropriate.

This repository contains the code for Text Classification using python's various libraries.

It used Numpy, Pandas, sklearn, spacy, string and eli5 libraries.

These libraries help to perform individually one particular functionality.

Numpy is used for working with arrays. It stands for Numerical Python.

Pandas objects rely heavily on Numpy objects.

Sklearn has 100 to 200 models.

spaCy uses the latest and best algorithms, its performance is usually good as compared to NLTK. In word tokenization and POS-tagging spaCy performs better.
ELI5 is a Python package which helps to debug machine learning classifiers and explain their predictions.
String module contains a number of functions to process standard Python strings.
The purpose of creating this repository is to gain insights into working of Classification of Text Data.
These python libraries raised knowledge in discovering these libraries with practical use of it.
It leads to growth in my ML repository.
This above few screenshots will help you to understand flow of output.

Motivation

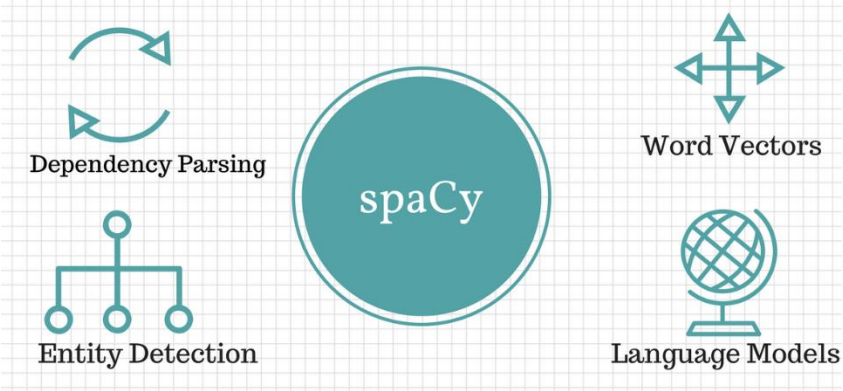
The reason behind building this is, it understands large volume of text. Till now I have made almost all projects that deals with tabular data and numbers and few categorical variables. However, in real world, there is huge text data that lies because that is what humans understands and is communication medium. Hence, working with text data becomes a major aspect of Data Science. The goal of text classification is to automatically classify the text documents into one or more defined categories be it Auto tagging of customer queries for business and clients, Understanding audience sentiment from social media. ELI5 library you can relate simply as humans or when we were kids, we cannot understand complex things easily and therefore when mentor explains those complex things through simple, we not only understand it but also be able to retain it longer. For every company, decisions are essential part and therefore, model interpretation is done exactly to understand model and to fabricate decision making policies better. This is to enable fairness, accountability and transparency which will give humans enough confidence to use these models in real-world problems which a lot of impact to business and society. Hence, I continue to gain knowledge while practicing the same and spread literary wings in tech-heaven.

Technical Aspect

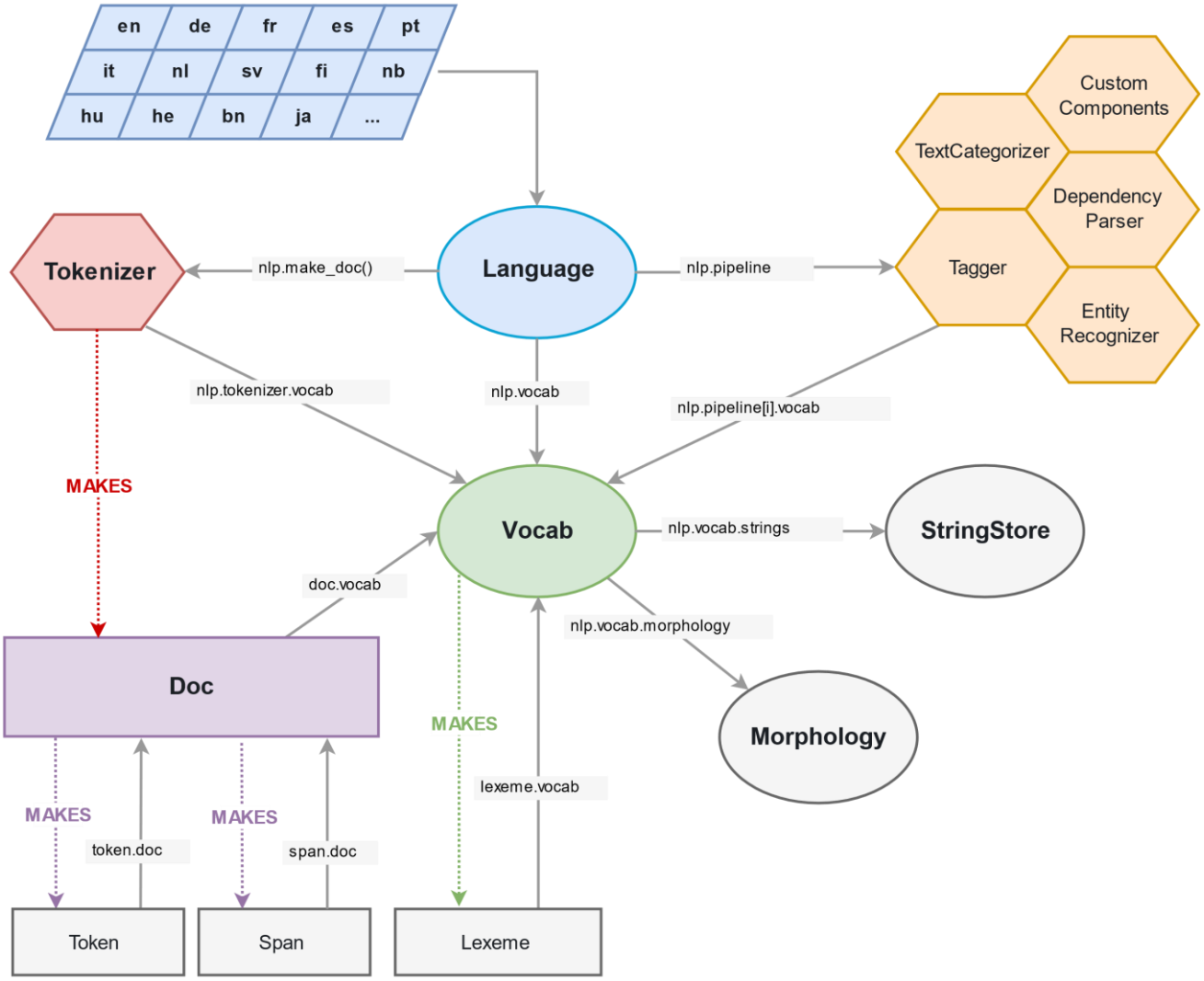
Numpy contains a multi-dimensional array and matrix data structures. It works with the numerical data. Numpy is faster because is densely packed in memory due to its homogeneous type. It also frees the memory faster.
Pandas module mainly works with the tabular data. It contains DataFrame and Series. Pandas is 18 to 20 times slower than Numpy. Pandas is seriously a game changer when it comes to cleaning, transforming, manipulating and analyzing data.
Sklearn is known as scikit learn. It provides many ML libraries and algorithms for it. It provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

To load a model, use spacy.load with the model's shortcut link, package name or a path to the data directory: import spacy nlp = spacy. load("en_core_web_sm") - load model package

"en_core_web_sm" nlp = spacy. spaCy is designed specifically for production use and helps you build applications that process and “understand” large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning. The models have been designed and implemented from scratch specifically for spaCy, to give you an unmatched balance of speed, size and accuracy. Algorithm that spaCy uses is, has its own deep learning library called thinc used under the hood for different NLP models. for most (if not all) tasks, spaCy uses a deep neural network based on CNN with a few tweaks. En_core_web_sm is, English multi-task CNN trained on OntoNotes. Assigns context-specific token vectors, POS tags, dependency parse and named entities.



Spacy Architecture:



Tokenization is the process of breaking a document down into words, punctuation marks, numeric digits, etc.
Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even if the stem itself is not a valid word in the Language. Purpose of Stemming is, usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Need Stemming because, Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. That additional information retrieved is why stemming is integral to search queries and information retrieval. When a new word is found, it can present new research opportunities.
Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is similar to stemming but it brings context to the words. So, it links words with similar meaning to one word.
Stemming and Lemmatization are different because, stemming technique only looks at the form of the word whereas lemmatization technique looks at the meaning of the word. It means after applying lemmatization, we will always get a valid word.

Stop words are a set of commonly used words in a language that they carry very little useful information.

ELI5 means Explain Like I am 5.

There are two main ways to look at a classification or a regression model:

1. inspect model parameters and try to figure out how the model works globally;
2. inspect an individual prediction of a model, try to figure out why the model makes the decision it makes.

For (1) ELI5 provides `eli5.show_weights()` function;

For (2) it provides `eli5.show_prediction()` function.

ELI5 aims to handle not only simple cases, but even for simple cases having a unified API for inspection has a value:

- you can call a ready-made function from ELI5 and get a nicely formatted result immediately;
- formatting code can be reused between machine learning frameworks;
- 'drill down' code like feature filtering or text highlighting can be reused.

Python strings are "immutable" which means they cannot be changed after they are created.
Need to train_test_split - Using the same dataset for both training and testing leaves room for miscalculations, thus increases the chances of inaccurate predictions.
The train_test_split function allows you to break a dataset with ease while pursuing an ideal model. Also, keep in mind that your model should not be overfitting or underfitting.

The purpose of the pipeline is to assemble several steps that can be cross-validated together while setting different parameters. Sklearn.pipeline make your workflow much easier to read and understand. They enforce the implementation and order of steps in your project.

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. The advantages of support vector machines

are: Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples.

SVC is a nonparametric clustering algorithm that does not make any assumption on the number or shape of the clusters in the data. In our experience it works best for low-dimensional data, so if your data is high-dimensional, a pre-processing step. import the SVM module and create support vector classifier object by passing argument kernel as the linear kernel in SVC() function. scikit-learn implements SVC, NuSVC and LinearSVC which are classes capable of performing multi-class classification on a dataset. They are just different implementations of the same algorithm. The SVM module (SVC, NuSVC, etc) is a wrapper around the libsvm library and supports different kernels while LinearSVC is based on liblinear and only supports a linear kernel.

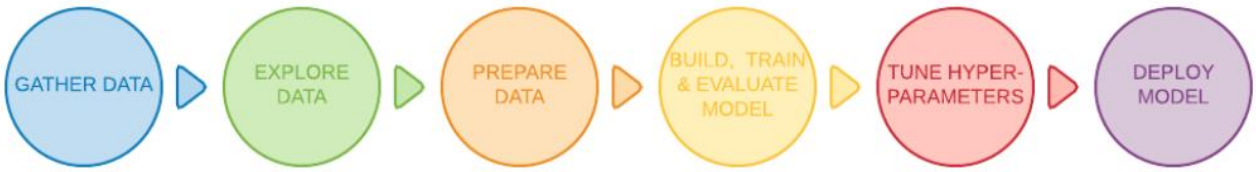
Transformers in Python can be used to clean, reduce, expand or generate features. The fit method learns parameters from a training set and the transform method applies transformations to unseen data.

Scikit-learn's CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

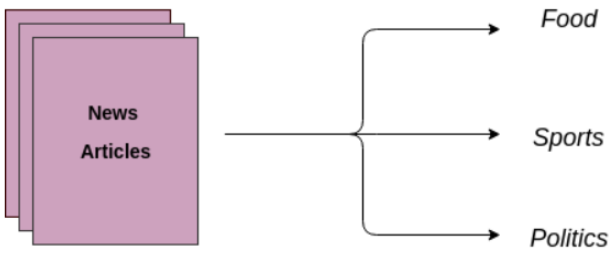
TfidfVectorizer - Transforms text to feature vectors that can be used as input to estimator. vocabulary_ Is a dictionary that converts each token (word) to feature index in the matrix, each unique token gets a feature index. In each vector the numbers (weights) represent features tf-idf score.

The only difference is that the TfidfVectorizer() returns floats while the CountVectorizer() returns ints. And that's to be expected – as explained in the documentation quoted above, TfidfVectorizer() assigns a score while CountVectorizer() counts.

Text Classification Workflow:



Text Classification Example:



Installation

Using intel core i5 9th generation with NVIDIA GFORECE GTX1650.
Windows 10 Environment Used.
Already Installed Anaconda Navigator for Python 3.x
The Code is written in Python 3.8.
If you don't have Python installed then please install Anaconda Navigator from its official site.
If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python* -m pip install --*upgrade pip and press Enter*.

Run/How to Use/Steps

Keep your internet connection on while running or accessing files and throughout too.
Follow this when you want to perform from scratch.

Open Anaconda Prompt, Perform the following steps:

cd <PATH>
pip install numpy
pip install pandas
pip install spacy
pip install en_core_web_sm
pip install sklearn
pip install eli5

You can also create requirement.txt file as, pip freeze > requirements.txt

run files.

Follow this when you want to just perform on local machine.
Download ZIP File.
Right-Click on ZIP file in download section and select Extract file option, which will unzip file.
Move unzip folder to desired folder/location be it D drive or desktop etc.
Open Anaconda Prompt, write cd <PATH> and press Enter.
eg: cd C:\Users\Monica\Desktop\Projects\Python Projects 1\ allin1\ Project_5_TextClassification_and_ModelInterpretation

In Anconda Prompt, pip install -r requirements.txt to install all packages.
Open in Jupyter Notebook, <filename>.ipynb
That is,

Open in Jupyter Notebook,
1)Text_Classification_and_ML_Model_Interpretation_with_Spacy_Eli5_Sklearn.ipynb
This takes sentimentdataset.csv file as input dataset.

Please be careful with spellings or numbers while typing filename and easier is just copy filename and then run it to avoid any silly errors.
Note: cd <PATH>
[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to cd one space <path> and press enter, then you can access all files of that folder] [cd means change directory]

Directory Tree/Structure of Project

Folder: allin1 > Project_5_TextClassification_and_ModelInterpretation
1)Text_Classification_and_ML_Model_Interpretation_with_Spacy_Eli5_Sklearn.ipynb

To Do/Future Scope

Can try with other libraries.

Technologies Used/System Requirements/Tech Stack

Credits

https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/

https://realpython.com/python-keras-text-classification/