Data Science Lab: Process and methods

Politecnico di Torino

Project report
Student ID: s277284

Exam session: Winter 2020

1. Data exploration

The purpose of the project is to define the sentiment of the reviews extracted from the TripAdvisor website. Sentiment analysis is the subfield of Natural Language Processing (NLP) and based on retrieving information from opinions.

Beginning of the process, the distribution of the dataset analyzed.
The dataset contains 27854 reviews each review corresponds to a label that is either positive or negative.

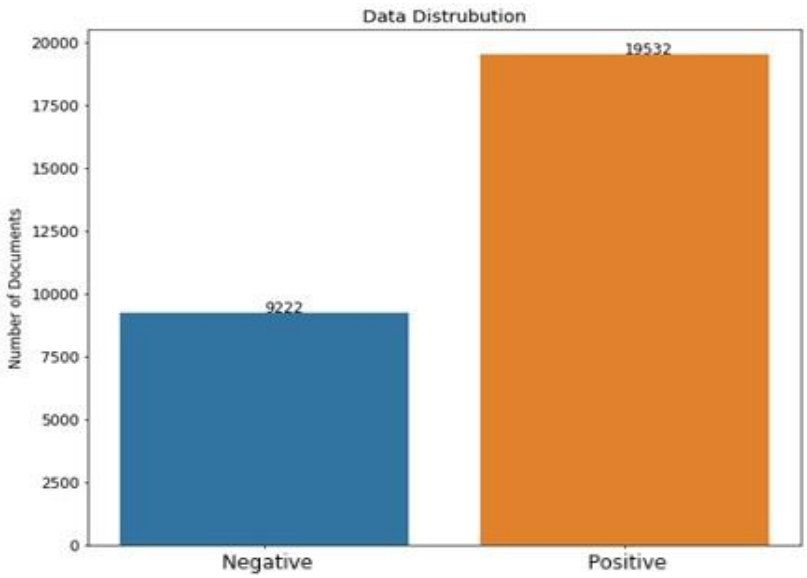

Figure 1 Distrubution of reviews

The given dataset is unbalanced. In such cases, prediction models tend to ignore the minority class and bias towards the majority class [1]. Biased models can lead to serious problems in case of medical diagnosis or fraud detection analysis. Consideration of accuracy metric possibly misleading, a trivial model that predicts all the reviews as positive will reach %67 accuracy. Precision, Recall, F1 scores, and ROC

curves can be useful for evaluating the model. There are several ways to deal with unbalanced datasets such as using additional data sources, considering different evaluation metrics, resampling techniques and using penalized algorithms.

The solution approach proposed for an unbalanced dataset problem is addressed by selecting an appropriate algorithm that adjusts the class weights by considering the proportionality of the classes.

Data does not contain any null values.

2. Preprocessing

Data preprocessing is a crucial part of the whole pipeline, Data must be cleaned before the analysis to get rid of disturbance and reduce the dimensions(vocabulary) to make it more manageable. Spacy is used for the preprocessing purposes since Natural Language Tool Kit (nltk) which is a common package often used for NLP does not support lemmatization for the Italian language.

Data cleaning steps performed in the project;

Tokenization

Tokenization is splitting the long strings into smaller pieces(tokens). Spacy tokenizer used for segmentation of the reviews into words, punctuations and special characters.

Removal of special characters & Case Normalization

Punctuations, numbers, unnecessary spaces, symbols are removed as they do not provide any information for the analysis

All capital letters converted into lowercase, in order not to treat the same words as they are different.

The Italian language has accented letters in its alphabet. The classic approach treats those letters as a special character. Unidecode package has used to treat those letters in the same way.

Removal of stop words

Stop words are the common words that has high frequency on a corpus. They generally do not help to catch the context, but sentiment analysis is much more sensitive to stop words removal since the actual meaning of the text can be lost by removing the stop words, especially for negations.

Removing negations may manipulate the actual meaning of the reviews dramatically.

Spacy's stop word list contains approximately 600 words. Instead of using that default list, another small stopword list [2] is used and negations removed from the list.

Lemmatization & Stemming

There are two general methodologies to access the root form of a word. Stemming and lemmatization. Both methods can be used for same purpose: freeing each word from suffixes and converting it into a common root so that they will be treated in the same way.

Lemmatization: Lemmatization is a technique to capture the morphological basis of the words.

Tokens collapsed into single token called as lemma using Spacy

3. Algorithm choice

Algorithm for Feature Extraction

Bag of Words approach

Feature extraction is done by implementing the TF-IDF technique. Numeric representations are calculated by multiplying TF and IDF of the words.

Scikit-learn's TfidfVectorizer class is used to obtain TF-IDF representation.

Frequent words that are specific in one class but not in other class can be useful for discriminating the reviews.

Including n-grams can boost the model and help to catch the meaning of consecutive word groups. A word phrase “not good” can be detected by a bigram while unigram yield “not” and “good” as two independent vocabulary items and disturb the actual opinion in a review.

Dimensionality Reduction

Singular Value Decomposition used for dimensionality reduction to improve efficiency. Nevertheless, the explained variance ratio was very low with a large number of principal components, algorithm choice requires further considerations.

Estimator Selection

Properties of Data

- Data is slightly unbalanced (1:2)
- Spare matrix contains approximately 60.000 columns(dimensions) for (bigrams = (1,2)).
- Number of instances < number of features

Support Vector Machines finds an optimal hyperplane that can be used for classifying classes. “Their ability to learn can be independent of the dimensionality of the feature space” [3], Sklearn’s LinearSVC algorithm can be appropriate for classification since it implements the One-vs-All strategy and problem has 2 classes. Moreover, LinearSVC use liblinear solver which converges faster than libsvm[4].

The regularization parameter (i.e., C) determines the effect of misclassifications on the objective function and can be useful for preventing the overfitting risk.

SVM assigns the same cost to the objective function for both positive and negative misclassifications. One C value for both classes can provide a hyperplane that skewed through to minority class. Different Error Cost proposed [5] to overcome skewness by using 2 different C values for penalizing the misclassification for each class independently.

Alternatively, Logistic regression can be used for a binary classification problem. Classifier multiples weights extracted by TF-IDF with input features and sum up before adding bias to weighted features. Weighted features passed through Sigmoid function to generate a probability. Logistic Regression uses the difference between estimated

probability and true label for updating weights and its cost function. The regularization parameter is used to have a generalized model [6].

Sigmoid function takes a real value as input and maps it to a range of 0-1.

4. Tuning and validation

Model validation is important for measuring the performance and reliability of the model.

Steps;

- Dataset split into training and test partition with proportionality of %70 (20127) and %30 (8627) respectively. Sklearn's "train_test_split" class used for the partitioning with "stratify" parameter in order to preserve to proportionality of the classes.

- The train partition is used to perform the gridsearch for tuning the parameters, with cross validation.

Building a pipeline for gridsearch;

1. Using TD-IDF to get the features matrix

Parameters of tfidfVectorizer specified in params_grid variable.

- max_df
- min_df
- ngram_range

2. Parameters of the estimators

Specified parameter for LinearSVC and Logistic Regression in params_grid dictionary.

- C (Regularization Parameter)
- Class_weight parameter is specified as "balanced" to prevent skewness.

- 5-fold cross validation has performed with different configurations of hyperparameters to see which parameter configuration generalize across these folds by maximizing the f1_weighted score

- Model is evaluated with unseen test partition that split at the beginning of the process to test the best configuration found in the training process. This is done to make sure that model is generalizable and not just learned the data available.

- Model trained with all the available data source to make predictions for the evaluation set.

Evaluation scores of LinearSVC on Test set;

Best Parameters found in gridsearch :

- C :0.9,
- class_weight : 'balanced',
- max_df: 0.5,
- min_df: 5,
- ngram_range: (1, 2)

	Positive	Negative
Positive	5696	164
Negative	158	2609

Figure 2 LinerSVC confusion matrix

	Precision	Recall	F1-Score	Support
Negative Class	0.9409	0.9429	0.9419	2767
Positive Class	0.9730	0.9720	0.9725	5860
Accuracy			0.9627	8627
Macro Av.	0.9569	0.9575	0.9572	8627
Weighted Av.	0.9627	0.9627	0.9627	8627

Table 1 LinearSVC Classification report

Evaluation scores of LogisticRegression on Test set;

Best Parameters found in gridsearch :

- C :100,
- class_weight : 'balanced',
- max_df: 0.8,
- min_df: 5,
- ngram_range: (1, 2)

	Positive	Negative
Positive	5697	163
Negative	143	2624

Figure 3 LogisticRegression confusion matrix

	Precision	Recall	F1-Score	Support
Negative Class	0.9415	0.9483	0.9449	2767
Positive Class	0.9755	0.9722	0.9738	5860
Accuracy			0.9645	8627
Macro Ave.	0.9585	0.9603	0.9594	8627
Weighted Ave	0.9646	0.9645	0.9646	8627

Figure 4 LogisticRegression Classification report

Both models provided high precision and recall scores which means they can handle with both classes.

Most important words that helped to discriminate the classes are visualized with word clouds. Bigrams are useful, without using bigrams it was not possible to catch the consecutive words like “non consigliare” and “non tornare”.

Figure 5 Wordclouds

5. References

1. Batuwita, Rukshan, and Vasile Palade. "Class imbalance learning methods for support vector machines." (2013).

2. https://www.ranks.nl/stopwords/italian

3. Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." *European conference on machine learning*. Springer, Berlin, Heidelberg, 1998.

4. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *Journal of machine learning research* 12.Oct (2011): 2825-2830.

5. Veropoulos, Konstantinos, Colin Campbell, and Nello Cristianini. "Controlling the sensitivity of support vector machines." *Proceedings of the international joint conference on AI*. Vol. 55. 1999.

6. Jurafsky, Daniel, and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River N.J: Prentice Hall, 2009. Print.