Advanced Analytics for Business

Assignment 3: *Text Minin*

xxxxxx

Table of Contents

[1 Data Collection 2](#_Toc9442127)

[2 Predictive model implementation 2](#_Toc9442128)

[2.1 Multiclass classification 2](#_Toc9442129)

[*Pre-procesisng pipeline* 2](#_Toc9442130)

[*Logisitic regression* 2](#_Toc9442131)

# Data Collection

We used Spark Streaming and the notebook “spark\_streaming\_example\_saving.py.ipynb” to build up an history of movie review on a local drive. We implemented a Python script to read the reviews saved in the different folders created by Spark and consolidate them I one single JSON file. We

We bult up a data set of 11 573 movei reviews. As depicted below this data set is extremely imbalanced.

|  |  |  |
| --- | --- | --- |
| Class | # of samples | % |
| Class 1 (= 1 star reviews) | 145 | 1.2% |
| Class 2 (= 2 star reviews) | 170 | 1.5% |
| Class 3 (= 3 star reviews) | 346 | 3% |
| Class 4 (= 4 star reviews) | 1 529 | 13% |
| Class 5 (= 5 star reviews) | 9 383 | 81% |
| Total | 11 573 | 100 |

# Predictive model implementation

## Multiclass classification

### Pre-procesisng pipeline

We started with the multiclass classification model, i.e. a model that assign a review from 1 to 5 to the movie reviews. We implemented a pipeline with the followings steps:

* Lemmatization: this operation was not implemnetd in our final implementation as we relied on
* Tokenization
* Stop word removal
* Count Vectorization
* TF-IDF transformation

### Logisitic regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Threshold | !. LR Rnd SPlit | Same as 1 + stratified spit | Same as 2 + downsampling | Same as 3  + up & down sampling |
| Accuracy | 82.1% | 84.3% | 43.8% | 48.7% |
| Class 1 Acc. | 27.6% | 32.2% | 50.0% | 65.4% |
| Class 2 Acc. | 20.5% | 30.0% | 37.0% | 25.0% |
| Class 3 Acc. | 33.8% | 52.6% | 50.0% | 38.8% |
| Class 4 Acc. | 57.0% | 24.9% | 33.3% | 54.5% |
| Class 5 Acc. | 90.5% | 99.8% | 47.8% | 48.4% |
| Mean Class Acc. | **45.7%** | **47.9%** | **43.6%** | **46.4%** |

!. Logictics regression , random split 80% training 20% tetsing

2. same but with elastic net regularization

3. with stratified split bease on on

4. all class 2 to 5 donwsample to level of class 1

5. downsampling / upsampling in order to get 2 000 examples per calss

Remarks:

Better way with SMOTE Sampling