Anomaly Detection Challenge

Challenge 2: Review Anomaly Detection

Mustika Rizki Fitriyanti (03667399)

Ishmeet Kaur (03677735)

November 2016

1 Introduction

The aim of this challenge is to use supervised anomaly detection in order to detect fake reviews in Yelp dataset. The dataset consisted of training data with the dataset of the reviewers and the hotels. The binary classification of the fake reviews by analyzing different feature patterns was the goal of the challenge.

2 Data Preprocessing and Analysis

The dataset consisted of 2908 entries with 9 features with one review containing no content. This dataset when combined with the reviewer and the hotel dataset based on hotel ID and the reviewer ID contained 33 features with 5 missing data entries of the reviewer . But the test dataset when combined with reviewer and the hotel data consisted of 1 missing data entry of the hotel and 4 missing data entries of the reviewer.

2.1 Reading the dataset

The reading of the dataset was a challenge as it involved several separators(;) inside the review content. The dataset was read by pandas library by removing the special characters using regular expression.

2.2 Replacement of the Missing Values

The missing values could not be ignored as the hotel ID of the data consisting of no review content had 22 entries in the test set. So, none of the missing values in the test and the training dataset were ignored. Initially, all the missing values were replaced by a null value but later in order to increase the accuracy, we tried replacing the missing values by mean, median , and mode of the respective column in the dataset in both the training and the test dataset.

3 Feature Engineering

There were several related features in the data and after reading the paper provided, we tried to engineer features in a similar direction:

3.1 Bag of Words

The Bag of Words model learns a vocabulary from all of the documents, then models each document by counting the number of times each word appears. First, the tokenization of the dataset was done which removes all the grammar and the stop words from the text and retrieves important vocabulary. Stemming of the words was also done. The bag of Words model was used to model the feature review Count.

3.2 Tf-Idf(Term-Frequency Inverse Document-Frequency)

The TF-IDF value for a token increases proportionally to the frequency of the word in the document but is normalized by the frequency of the word in the corpus. This essentially reduces importance for words that appear a lot generally, as opposed to appearing a lot within a particular document. The dense matrix of the Tf-Idf matrix was compared with the test set.

3.3 N-Grams

N-Grams representation of the text is the set of n contiguous words appearing together in the document as a feature which is known to increase accuracy. We tried to implement unigram, bigram and trigrams as features of the training set.

3.4 Maximum Number of Reviews

As a fake reviewer notably writes many reviews per day as compared to non fake reviewer, we found the maximum number of reviews by dividing the reviewCount feature of the reviewer by the number of days (date in the train data – YelpJoinDate in the reviewer dataset). We did not set any threshold as the fake reviewers in the training set did not have notably high reviews per day.

3.5 Percentage of Positive Reviews

The percentage of positive reviews is obtained by dividing the number of 4 and 5 star ratings by the reviewcount of the reviewer. Also, the rating scale divided by the reviewCount was also tried to improve accuracy .

3.6 Review Length

The review length of the reviewes review content was used as a feature.

3.7 Absolute and Expected Rating Deviation

The absolute rating deviation was obtained by the absolute value of the difference of the rating of the reviewer and the hotel. The expected rating deviation was obtained by the standard deviation of all the ratings of a reviewer.

3.8 Maximum Content Similarity(MCS)

The Maximum Content Similarity of the reviewers was obtained by the maximum value of the cosine distance of the reviews of all the reviewers. All these three features when used gave a very low (51 percent accuracy in the private score).

3.9 Mean Rating of reviewer by location of the hotel

The mean rating of the reviewer was taken grouped by the location of the hotel as the hotels in one location had similar ratings .
The first three text mining features gave an accuracy of 51 percent and the other behavioral features provided an accuracy of 82.3 percent (Kaggle private score). In order to further increase the accuracy, we tried the following methods:

- Dimensionality Reduction by PCA :

  The best *ncomponents* for the PCA of all the features providing the maximum accuracy (that is providing features with high variance) was found(which was 1)

- Feature Selection:

  1. Optimal Number of Features:

  The optimal number of features were found to be 8 using Random Forest with a stratified cross validation of 10, 5 using stratified cross validation of 5 .

  2. Feature Importance and Ranking:

  The feature importance was found by Recursive Feature Elimination and Univariate Selection using the following models Logistic Regression Random Forest Extra Trees Classifiers Principal Component Analysis

  Using all these methods the rank of the first 8 features found out was used in Training Set in order to train the model.

4 Normalization

All the features were normalized and the textual features like Accepts Credit cards, Wifi and Price Range were mapped to numerical categories.

5 Models

We tried fitting our dataset to the following models:

- Naive Bayes Classifier
- Decision Tree Classifier
- K- nearest Neighbors
- Random Forest
- Logistic Regression
- AdaboostClassifier
- Support Vector Machine

All these models were used for fitting the training data with a stratified cross validation of 5.

6 Observations and Results

The following observations were made during feature engineering:

- Using ngrams ,tf-Idf and bag of Words as features with other numerical features gave drastically low accuraccy (50 percent - kaggle public score)
- Using all the numerical features except the text mining features (ngrams,and tf-idf) resulted in an accuracy of 79 percent(Kaggle public score) and 0.82313 in the private score.
- Feature selection and feature importance increased the cross validation score to 90.8 and the kaggle public score to 0.81763
- The best number of trees for random forest using only the selcted features from the original dataset was 166.
- The original features when combined with the behavioral analysis features gave a low cross validation score.
- Knn with the number of nearest neighbors as 15 and 5 features gave an accuracy of 0.82172 in the private score.
- Adaboost and Random Forest gave a similr accuracy using 8 optimal features from the dataset.

7 Summary

By trying out different models and looking at an private score, the best results were using random forest (with n=150) with the behavioural fetaures.

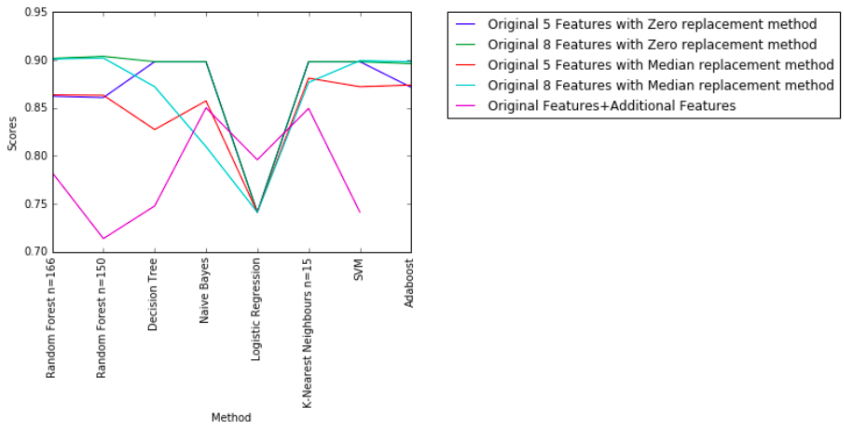

Figure 1. The accuracy score from different feature engineering method

Here are the top result of private and public score that we got on Kaggle:

Method	Public Score	Private Score
all original + behavioural features with Random Forest (k=150)	0.79288	0.82313
5 Features median with KNN (k=15)	0.80821	0.82172
8 Features zero with Random Forest(with n=166)	0.82066	0.81004