Project-Proposal-DataVizLab

Cynthia (Mengyuan) Li

2023-02-10

Sentiment analysis and Machine Learning with Scikit-learn

This project is based on the Yelp competition[https://www.drivendata.org/competitions/5/keeping-it-fresh-predict-restaurant-inspections/page/17/], which was based on restaurants in the City of Boston. This project will try to apply similar analysis to the City of Atlanta. Sentiment analysis refers to analyzing an opinion or feelings about something using data like text or images, regarding almost anything. Sentiment analysis helps companies in their decision-making process.

scikit-learn[https://scikit-learn.org/stable/] is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Problem Definition

Fulton County [https://www.fultoncountyga.gov/inside-fulton-county/fulton-county-departments/board-of-health/environmental-health/restaurant-inspection] regularly inspects every restaurant to monitor and improve food safety and public health. As in most cities, health inspections are generally random, which can increase time spent on spot checks at clean restaurants that have been following the rules closely — and missed opportunities to improve health and hygiene at places with more pressing food safety issues.

Analysis Targets

Using the Fulton County public health inspection on the scores and grades for food services, which can be thought of as a score out of 100. [http://ga.healthinspections.us/georgia/search.cfm?county=Fulton]

Sample of dataset

All data set can be accessed and downloaded from Yelp Open Data Set [https://www.yelp.com/dataset/download].

```
# business - yelp_academic_dataset_business.json
{
    'type': 'business',
    'business_id': (business id),
    'name': (business name),
    'neighborhoods': [(hood names)],
    'full_address': (localized address),
    'city': (city),
    'state': (state),
    'latitude': latitude,
    'longitude': longitude,
    'stars': (star rating, rounded to half-stars),
    'review_count': review count,
```

```
    'categories': [(localized category names)]
    'open': True / False (corresponds to closed, not business hours),
    'hours': {
        (day_of_week): {
            'open': (HH:MM),
            'close': (HH:MM)
        },
        ...
    },
    'attributes': {
        (attribute_name): (attribute_value),
        ...
    },
}
# reviews - yelp_academic_dataset_review.json
{
    'type': 'review',
    'business_id': (business id),
    'user_id': (user id),
    'stars': (star rating, rounded to half-stars),
    'text': (review text),
    'date': (date, formatted like '2012-03-14'),
    'votes': {(vote type): (count)},
}
# users - yelp_academic_dataset_user.json
{
    'type': 'user',
    'user_id': (user id),
    'name': (first name),
    'review_count': (review count),
    'average_stars': (floating point average, like 4.31),
    'votes': {(vote type): (count)},
    'friends': [(friend user_ids)],
    'elite': [(years_elite)],
    'yelping_since': (date, formatted like '2012-03'),
    'compliments': {
        (compliment_type): (num_compliments_of_this_type),
        ...
    },
    'fans': (num_fans),
}

# check-ins - yelp_academic_dataset_checkin.json
{
    'type': 'checkin',
    'business_id': (business id),
    'checkin_info': {
        '0-0': (number of checkins from 00:00 to 01:00 on all Sundays),
        '1-0': (number of checkins from 01:00 to 02:00 on all Sundays),
        ...
        '14-4': (number of checkins from 14:00 to 15:00 on all Thursdays),
        ...
        '23-6': (number of checkins from 23:00 to 00:00 on all Saturdays)
```

```
    }, # if there was no checkin for a hour-day block it will not be in the dict
}

# tips - yelp_academic_dataset_tip.json
{
    'type': 'tip',
    'text': (tip text),
    'business_id': (business id),
    'user_id': (user id),
    'date': (date, formatted like '2012-03-14'),
    'likes': (count),
}
```

Tools

The approach of this project will be based on the implementation of random forests in Python with ML packages including:

- pandas
- NumPy
- scipy
- scikit-learn

Approach Overview

Importing the Required Libraries and the Dataset

The first step as always is to import the required libraries and the dataset

Data Analysis

We will start exploring the dataset a bit to see if we can find any trends.

Data Cleaning

Yelp review can contain many slang words and punctuation marks. We need to clean our data before they can be used for training the machine learning model. However, before cleaning the reviews, we also need to divide our dataset into feature and label sets.

Representing Text in Numeric Form

Statistical algorithms use mathematics to train machine learning models. However, mathematics only work with numbers. To make statistical algorithms work with text, we first have to convert text to numbers. To do so, three main approaches exist i.e. Bag of Words, TF-IDF and Word2Vec.

Bag of Words

Bag of words scheme is the simplest way of converting text to numbers.

TF-IDF

In the bag of words approach, each word has the same weight. The idea behind the TF-IDF approach is that the words that occur less in all the documents and more in individual document contribute more towards classification.

TF-IDF using the Scikit-Learn Library

Python's Scikit-Learn library contains the TfidfVectorizer class that can be used to convert text features into TF-IDF feature vectors.

Dividing Data into Training and Test Sets

In the previous section, we will have the data converted into the numeric form. As the last step before we train our algorithms, we need to divide our data into training and testing sets. The training set will be used to train the algorithm while the test set will be used to evaluate the performance of the machine learning model.

Training the Model

Once data is split into training and test set, machine learning algorithms can be used to learn from the training data. Any machine learning algorithm could be used here. The Random Forest algorithm will likely be chosen, owing to its ability to act upon non-normalized data.

Making Predictions and Evaluating the Model

Once the model has been trained, the last step is to make predictions on the model. To do so, we need to call the predict method on the object of the RandomForestClassifier class that we used for training.

Conclusion

The sentiment analysis is one of the most commonly performed NLP tasks as it helps determine overall public opinion about a certain topic. In this project a few foundamental concepts will be touched and utilized that can benefit a wide audience: - Preprocess text - Vectorize text input easily - Work with the tf.data API and build performant Datasets - Build Transformers from scratch with TensorFlow/Keras and KerasNLP - the official horizontal addition to Keras for building state-of-the-art NLP models - Build hybrid architectures where the output of one network is encoded for another