Analysis Quality of Meals Restaurants By ML

The focus of this project was Supervised learning (Classification Algorithms) and Unsupervised Learning (K-Means Clustering) and Natural Language Processing.

Introduction

In this kernel, we are going to analysis the Arabic language based restaurant reviews for the sentiment analysis. There are two types of sentiments which are 1 = positive and -1 = negative. The data contains the positive reviews text and negative reviews text for the binary classification.

Problem description:

1- How to deal correctly with Arabic texts.

2- To build a system to classify the sentiments in arabic text based restaurant reviews using machine learning.

Datasets description:

RES1.csv, The dataset is a collection of Arabic texts, which covers Dialectal Arabic (DA). Dataset of restaurant reviews scrapped from qaym.com, 8364 reviews. The dataset consists of four columns and 8364 rows. It has two classes (Negative and Positive)

Column name	description
Polarity	which is a string value has two classes (-1: Negative & 1:Positive) indicating the sentiment around the review
Text	is the review plain text of a restaurant in Arabic
Restaurant_id	the restaurant ID on the website.
User_id	the user ID on the website

Source of Data::

https://github.com/hadyelsahar/large-arabic-sentiment-analysis-resouces/tree/master/datasets

Phase 1

Natural Language Processing (NLP): Arabic text preprocessing

Tools:

- Nltk
- arabic_reshaper
- preprocessing
- PIL
- bidi.algorithm
- arabic_reshaper
- WordCloud

Phase 2

Supervised learning (Classification Algorithms).

Classification Problems : with unBalanced Datasets

Sandard Features: before running our models we need to transform our text to numbers, by multiple approaches .

1. Count Vectorizer.

is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. This is helpful when we have multiple such texts, and we wish to convert each word in each text into vectors.

https://www.geeksforgeeks.org/using-countvectorizer-to-extracting-features-from-text/

2.TF–IDF.
TF-IDF stands for “Term Frequency Inverse Document Frequency” it uses a slightly more complicated approach which will penalize common words that occur in multiple documents.

Classifiers :
Naïve bayes : MultinomialNB, GaussianNB and BernoulliNB.
Linear model: LogisticRegression and SGDClassifier.
SVM: Linear SVM, and LinearSVC.

Evolution measures:

After training the model, we will apply the evaluation measures to check that how the model is getting predictions. We will use the following evaluation measures to evaluate the performance of the model:

- Accuracy
- Precision
- Recall
- F1 Measure

Tools:

- CountVectorizer
- TfidfVectorizer
- imblearn.over_sampling
- naive_bayes
- linear_model

Phase 3

Unsupervised Learning (K-Means Clustering)

Sandard Features : TFIDF Vectorizer, Count Vectorizer.

Clustering :MiniBatchKMeans

Tools:

- cluster
- matplotlib.cm
-

General Tools

- Pandas
- Numpy
- matplotlib.pyplot

Technical Approach:

We are using python language in the implementations and Jupyter Notebook that support the machine learning and data science projects.