Capstone Project Proposal. Machine Learning Engineering Nanodegree

Sentiment Analysis: Unlocked Mobile Phones Dataset
Silvia Onofrei
18 December 2018

Domain Background

The ubiquity of the social media in everyday life led to the creation of numerous forums, blogs, review sites, etc. The vast amounts of opinionated data available online propelled the research in the field of sentiment analysis which is a subfield of Natural Language Processing. Natural Language Processing (NLP) is a branch of Artificial Intelligence whose subject is to analyze and process the human language. Its origins can be related to Alan Turing's article Computing Machinery and Intelligence, published in 1950, which proposed a criterion of intelligence, now known as the Turing test. A brief description of the NLP field can be found in Wikipedia: Natural Language Processing, for a practical introduction using Python see the book. A list of references is given in the blog. Sentiment analysis investigates people's opinions and attitudes towards products, services, organizations, issues, events, topics (see Bing Liu's page for a comprehensive treatment).

In the present day, it is quite common for a customer to research online a product or a service before making a purchase. The reviews posted by other customers play an essential role, they are often more relevant than the opinions of professional reviewers. On the other side, as the online commerce becomes prevalent, more businesses follow up wih their customers via online channels to request product reviews. As an Amazon customer I have noticed a sharp increase in such requests during the past months. Given the amount of data that is available from these reviews, it is desirable to develop automated methods to determine whether a certain review is favorable or not, or whether the review is relevant to other customers (an interesting read on sentiment analysis).

The questions that arise in NLP are interesting, challenging, in actuality and originate in various areas such as news, finance, business, internet, to mention only a few. My main interest in studying Machine Learning resides in Deep Learning, in particular in Artificial Neural Networks which are extensively used in NLP.

Problem Statement

Given a large set of text documents, perform sentiment analysis using deep learning techniques and supervised machine learning methods (when labels are available). The outcome should provide a binary classification of each document, with 0 indicating a negative sentiment and 1 indicating a positive sentiment.

Datasets and Inputs

The data used in this project is from the Kaggle public dataset repository: Amazon Reviews: Unlocked Mobile Phones. An initial analysis of the data can be accessed on KDnuggets.

The dataset consists of more than 400,000 reviews from Amazon's unlocked mobile phone category, extracted by PromptCloud in December 2016. It has 6 features: Product Name, Brand Name, Price, Rating, Reviews, Review Votes.

The Unlocked Mobile Phones dataset is a substantial dataset which will allow for reasonable predictions and thorough investigation. The *Reviews* column will be used for the NLP and in particular for the sentiment analysis, the results will be corroborated with the *Rating* feature. Further analysis could include the correlations between the sentiment of the review and the price tag and/or the brand name.

From an initial evaluation of the data I observe that the data is highly imbalanced (with more than half of the reviews receiving a 5 star rating) and this aspect has to be taken into account when splitting the data into training and testing sets and also in choosing the evaluation metrics; see this J. Brownlee's blog and this Analytics Vidhya blog for practices on how to work with imbalanced datasets.

Solution Statement

The primary goal of this analysis is to automatically determine whether the review expresses an overall positive or negative opinion about the product. In order to achieve this goal, I plan to employ Convolutional Neural Networks (see Kim's article) in conjunction with the advanced text processing techniques such as Tf-Idf, word embedings and the SentimentAnalyzer module from NLTK. I will also use a Support Vector Machine (SVM) classifier as a benchmark model.

Benchmark Model

For the benchmark model I will use a Support Vector Machine (SVM) classifier with linear kernel and 10-fold cross validation. Such classifier has been found to perform well in document sentiment analysis, see Vohra, Teraiya and Shinde, Radhod.

Support Vector Machines (SVM) is a class of supervised machine learning models used for both classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, a SVM classifier assigns one of the two categories to the testing data. The examples are represented as points in space that are separated by certain hyperplanes. To avoid overfitting, I will use k-fold cross-validation (with k = 10), which means that the training set is split into k subsets, k-1 subsets are used for training and the remaining subset is used for testing. The performance metric reported by k-fold cross-validation is then the average of the values computed in the loop.

After processing the review texts, each review is assigned a vector by the Tf-Idf scoring model. The *Rating* which takes values between 1 and 5, are binned together so that 1 and 2 ratings are relabelled as 0 (negative review), and 4 and 5 are relabelled as 1 (positive review). The remaining reviews with neutral rating (of 3) will be ignored.

Evaluation Metrics

I will start by plotting the confusion matrices. Although not an evaluation metric, it is an intuitive way to represent the performance of the model. Next, I will compute the F_1 scores for both models (see Performance metrics in ML). Note that since the data is imbalanced, the accuracy is not an appropriate evaluation metric. I am aiming for high precision and thus high F_1 scores.

I also plan to plot the receiver operating characteristic (ROC) curve. This illustrates the performance of a binary classifier as its discrimination threshold is varied. It is created by plotting the sensitivity(true positive rate) versus the false positive rate. Finally, I will evaluate the ROCAUC score; this is the area under the ROC curve and it represents how much the model is capable of distinguishing between the classes. Higher the ROAUC, better the model's predictions (see Understanding AUC-ROC Curve).

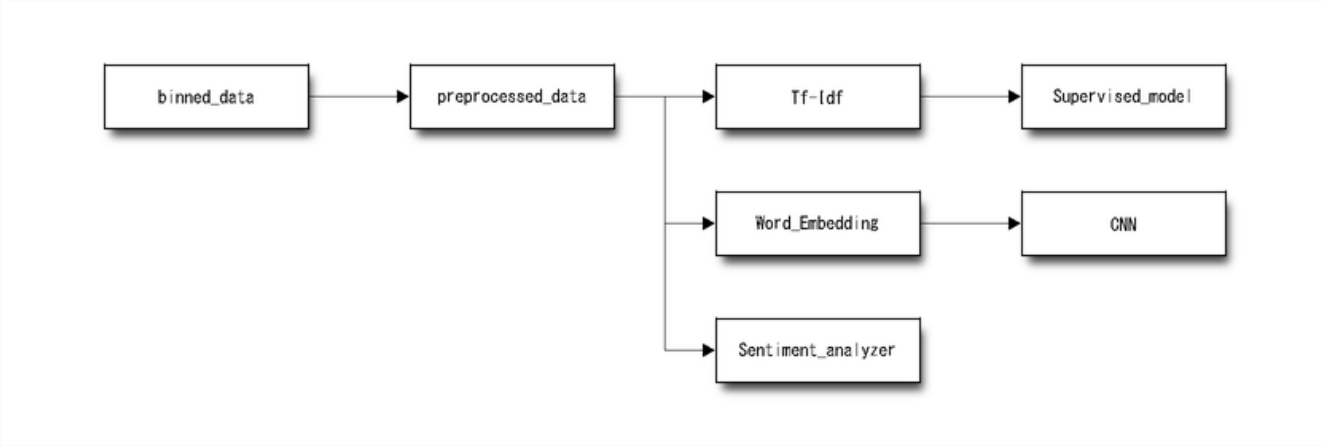
Project Design

The work flow for the project consists of 5 main parts which are described below.

1. Data Preparation which contains the following steps:

- *data upload*: data is available as .csv file, that will be uploaded as a Pandas dataframe;
- *investigate data*: a preliminary investigation will be performed, details on the size and the object types of the features and also the number of missing or NaN entries will be determined;
- *clean data*: I handle the missing data, I change the object type of the features if necessary;
- *analyze data*: I will perform a visual analysis of the data, in order to determine if there are any dependencies to be taken into account among the available features;
- *split data*: the data will be shuffled and split into a training set (80% of data) and a test set (20% of data), using StratifiedShuffleSplit();
- *data binning*: the ratings 1 and 2 are grouped together (negative or 0) and the ratings 4 and 5 are binned as 1 (positive), the neutral ratings are removed.

2. Text Processing



Each review text will be processed according to the scheme. The architecture of this part follows the work flow given in

Ultimate guide to deal with text data, and it consists of the following steps:

- Text Pre-Processing:
  - Convert all letters to lower case
  - Remove punctuation and special characters
  - Remove rare words
  - Tokenization
  - Remove stopwords
  - Stemming and/or Lemmatization
- Advanced Text Processing. Feature Extraction:
  - **Tf-Idf** In this case each review is represented as a vector of dimension given by the entire text corpus, and whose components are given by the Tf-Idf measure of each word. This stands for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection of documents (see Wikipedia). I will use TfidfVectorizer. to convert the reviews into a matrix of Tf-Idf features (see Text processing).
  - **Word Embeddings** This step will be done using Word2vec, a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus (see Word2Vec.).
  - **SentimentAnalyzer** Is a tool to implement and facilitate Sentiment Analysis tasks using NLTK features and classifiers.

3. Create the Benchmark Model

Create and train a classifier on the processed Tf-Idf vectors. I intend to use a SVM classifier, with linear kernel combined with 10-fold cross validation. For labeling I will use the positive(1) vs. negative(0) information provided by the binned star ratings.

4. Create the Convolutional Neural Network

Some research on sentiment analysis for document classification indicates that one can get good results with a single layer CNN (see Kim's article). I will start with a simple architecture of the form (inspired from J. Brownlee's blog):

I also intend to experiment with other CNN architectures, which involve multiple convolutional layers.

5. Compute the Evaluation Metrics

The evaluation metrics are computed for the two models: the benchmark (SVM) model and for the deep learning (CNN) model. They are compared among themselves and against the results used via the SentimentAnalyzer toolkit from NLTK.

Software

This project is performed on a Jupyter notebook, in Python 3 with the following dependencies:

- Numpy and Pandas
- scikit-learn
- matplotlib
- Seaborn
- Keras
- TensorFlow
- NLTK