The Occidental Computer Science Comprehensive Project: Tutorial Report

Odelia Putterman
putterman@oxy.edu
Occidental College

Abstract

This report documents the tutorial completed as part of my Occidental College Computer Science Comprehensive Project: Predicting Cryptocurrency Prices for Stock Trading Using Machine Learning. This report has four components: methods, evaluation, discussion, and software documentation. For each component, we include a section. This report walks us through the tutorial followed, Stock Market Sentiment Analysis Using Python & Machine Learning (Youtube tutorial), reviewing the key concepts and learned information relevant to the comprehensive project.

1 Methods

This section is a walk-through of the methods implored in this tutorial in a numbered list. We detail the steps below.

1. Install *vaderSentiment* with pip to get access to sentiment analysis tools.
2. Load necessary libraries for this tutorial. These are:
    - pandas;
    - numpy;
    - textblob.TextBlob;
    - re;
    - vaderSentiment.SentimentIntensityAnalyzer;
    - sklearn.model_selection.train_test_split;
    - sklearn.metrics.accuracy_score;
    - sklearn.metrics.classification_report; and
    - sklearn.discriminant_analysis.LinearDiscriminantAnalysis.
3. Load the data. In this tutorial, we used two data sets: *Dow_Jones_Industrial_Average_News.csv* and *Dow_Jones_Industrial_Average_Stock.csv*, all originating from the Dow Jones data and downloaded from Kaggle.
4. Next, we merged the two data sets on the 'Date' field to create one consolidated file.
5. To prepare the data for sentiment analysis, we grabbed the *news* data set inputs, consolidated them into one joint string for each date, and cleaned the data by removing certain unnecessary string patterns, storing this processed data in a new list and adding it to our consolidated csv.
6. Next, we created functions to get the subjectivity and polarity of text using **TextBlob(text).sentiment.subjectivity** and **TextBlob(text).sentiment.polarity**, where **text** is replaced with the actual text to analyze.
7. We used these newly created functions to create a subjectivity and polarity column, with inputs as the subjectivity and polarity outputs from these functions on each line of 'Combined_News' (the cleaned news string).
8. Next, we made a function to get the sentiment scores called **getSIA**. Using this function, we got the sentiment scores for each date, creating a new column in the combined csv for each of: 'combined', 'negative', 'neutral', and 'positive', where each is a value between -1 and 1.
9. We cleaned our combined csv to keep only the necessary columns: 'Open', 'High', 'Low', 'Volume', 'Subjectivity', 'Polarity', 'Compound', 'Negative', 'Neutral', 'Positive', and 'Label'.
10. We split this cleaned combined csv into two data frames, one for the 'input' data (everything but the 'Label' column) and one for the target data (the 'Label' columns).
11. Using **train_test_split**, we split the data into 80 percent and 20 percent to train and test the data.
12. Using the train data, we trained a **LinearDiscriminantAnalysis()** model.
13. And, finally, we got the model predictions for the test data and compared between these predictions and the labeled data to get the precision, recall, f1-score, and support.

2 Evaluation

This tutorial evaluated its outputted results by comparing the predicted stock prices produced from the trained model with the actual stock prices from that same period. This found to produce a precision rate over 80%. Other measures besides precision were accounted for as well, includ-

ing recall, f1-score, and support, all showing very promising results.

3 Discussion

This tutorial walked me through how to predict stock prices (whether they will increase or decrease) with sentiment analysis based on top news headlines. The whole process was so seamless and straight-forward it boosted my confidence in my ability to carry-out this project tremendously. While it showed the ease with which sentiment analysis and model building can be done using pre-built libraries, it did not cover the complexity involved with sourcing data, as these data sets were simply provided in a form which was easy to work with. That said, data processing was covered, which is essential to the success of the model.

4 Software Documentation

See attached **stock_market_tutorial.pdf** file for the code from this tutorial.