Stock Market Prediction Applying Daily News

Jing Xiang

January 2020

Contents

Capstone Project 2: Final Report

Chapter 1 Problem statement

Market prediction is an interesting topic to individuals and investment companies, and there are dozens of related competitions and datasets on Kaggle. To practice NLP, I'm going to use the past ten years' daily news headlines from Reddit to predict Dow Jones Industrial Average (DJIA) up and down through binary classification.

This capstone project contains three parts of data science studies: natural language processing, feature engineering, and training and prediction. The preprocess of NLP will include lower case, removing stop words, token, stemming and lemmatization; and after preprocess, sentiment analysis will be applied to each news headline. Feature engineering section will select the best features, including news feature and time series features, to train the machine learning model and predict market.

The public and free dataset used in this study is here:
https://www.kaggle.com/aaron7sun/stocknews

The preliminary plan of this project:

1. Download the past ten years' news headlines on Reddit and Dow Jones Industrial Average (DJIA) market datasets from Kaggle.
2. Data cleaning: filter out the news during weekends when market is closed.
3. Run NLP on the news: preprocess including lower case, removing stop words, token, stemming and lemmatization, and sentiment analyze after preprocessing.
4. Select the best features to train the model and predict.
5. Test features on different machine learning models.

Chapter 2 Dataset cleaning and natural language processing

1. Dataset cleaning

There are two channels of data provided in this dataset:

- News data: Historical news headlines from Reddit World News Channel (/r/worldnews). They are ranked by Reddit users' votes, and only the top 25 headlines are considered for a single date. (Range: 2008-06-08 to 2016-07-01)

- Stock data: Dow Jones Industrial Average (DJIA) from Yahoo Finance is used to "prove the concept". (Range: 2008-08-08 to 2016-07-01)

To clean the datasets, I convert the date columns to date time format in both table, filter out the news headlines in weekends to reduced news data size before NLP, and calculate the binary label (1 and 0) of stock data:

- "1" when DJIA adjust close value rose
- "0" otherwise

2. NLP on the news

The news data is preprocessed by lowering case, removing stop words in English, tokenization, stemming and lemmatization. After preprocess, sentiment analysis is applied to each news headline.

The NLP libraries used are:

- nltk.corpus.stopwords
- nltk.stem.WordNetLemmatizer
- nltk.stem.PorterStemmer
- nltk.tokenize.RegexpTokenizer
- nltk.sentiment.vader.SentimentIntensityAnalyzer

From the sentiment analysis, the negative, neutral, positive and compound scores are computed for each news headline, which will be used to train machine learning models.

Chapter 3 Feature engineering

After data cleaning and preprocessing, I calculate the average negative, neutral, positive and compound news scores each day, and based on the machine learning model test results, only top 5 headlines are used to get the average scores.

The past seven days' scores are used as features to train and test models, and only past 4 days scores are chosen after testing.

Since the dataset across 10 years range, and the stock market is related to seasons, the year, month, day and day of week are used as features to train and test models.

Chapter 4 Train and test models

From feature engineering, I get 8 features (the past 4 days scores, year, month, day, and day of week) to train and test model to predict the market label.

Instead of randomly split the 10 years dataset for training and testing, I use the first 8 years data to train the models, and the last 2 years data to test the models.

I have tested 16 different classification machine learning models, including K nearest neighbors, support vector machine, Gaussian process, decision tree, random forest, Ada boost, gradient boost and so on. Ada boost classifier gives the best test score, which is 0.581.

Future Work

- Test news topic modeling, and include the topics in model features
- Apply grid search CV in models