Raport postępów (9.04.2018)

Środowisko do pozyskiwania i agregacji informacji o kursach kryptowalut i informacji z nimi związanych oraz predykowania przyszłych kursów

Rafał Grabiański, Patryk Konior

Wykonana praca

- prosty panel administracyjny pozwalający wybierać spośród obsługiwanych kryptowalut
- perzystencja w bazie danych
- pierwsza wersja klasyfikatora Naive Bayes do analizy sentymentu

Analiza Sentymentu

W celu lepszej wizualizacji stanu rynkowego wspieranych przez aplikację pozycji kryptowalutowych, do aplikacji dołączyliśmy moduł analizy sentymentu przychodzących tweetów, dotyczących kryptowalut.

Moduł korzysta z implementacji naiwnej sieci Bayes'a z biblioteki NLTK. Poniżej przedstawiamy budowę naszego klasyfikatora.

Przetwarzanie danych wejściowych

Zanim poddamy pozyskane tweety zadaniom uczenia modelu lub też już klasyfikacji, wykonujemy na nich proste przetwarzanie składające się z kilku etapów:

Etapy wstępnego przetwarzania tweetów:

1. Tokenizacja
2. Zamiana hashtagów na zwykłe słowa
3. Usuwanie odniesień do użytkowników
4. Usuwanie adresów URL
5. Standaryzacja emotikon
6. Usuwanie znaków przestankowych, stopwords i normalizacja wielkości liter
7. Stemming

Za wszystkie te zadania odpowiedzialna jest klasa *TweetProcessor()*, która składa się z dwóch metod:

tokenize_tweet(tweet_str) - kroki 1-6

Pobiera string reprezentujący Tweet, zwraca listę znormalizowanych tokenów.

stem_tweet(tokens) - krok 7

Pobiera listę tokenów, zwraca listę tokenów z wykonaniem stemmingu. W pierwszej wersji algorytmu korzystamy ze stemmingu dostarczonego w implementacji *SnowballStemmer* modułu *NLTK*.

Uczenie klasyfikatora

Dane do uczenia

Zdecydowaliśmy się wykorzystać publicznie dostępne dane do klasyfikacji wraz z ręcznie wykonaną klasyfikacją dla tweetów pobranych w związku z domeną naszej aplikacji.

Zewnętrzne dane treningowe (Stanford Twitter Corpus) to dane pobrane i sklasyfikowane automatycznie na podstawie zawartych w treści emotikon. Każda krotka zbioru zawiera tekst tweet'a, ocenę pozytywności 0-4 oraz inne dodatkowe dane.
http://help.sentiment140.com/for-students/

Wewnętrzne dane treningowe zostały pobrane przez nas używając Twitter API. Dotyczyły one wybranych przez nas kryptowalut i zostały ręcznie sklasyfikowane do trzech grup (negatywne, neutralne, pozytywne).

Feature extraction

Do pierwszej ewaluacji naszego algorytmu wybraliśmy prostą metodę budowania featerów na bazie tweetów. Składa się ona z następujących etapów:

- **zebranie zbioru występujących słów w danych treningowych**
  Jeżeli dane słowo po normalizacji występuje przynajmniej w jednym tweecie, jest dodawane do zbioru.
- **budowa wektora wystąpień dla każdego tweet'u**
  Dla każdego tweetu budowany jest wektor składający się ze wszystkich słów

słownika o wartościach logicznych oznaczających czy dane słowo występuje w tweecie

Początkowo uznaliśmy, że najłatwiej będzie budować klasyfikator jedynie w oparciu o unigramy. W dalszej części planowane jest rozszerzenie sposobu feature extraction.

Uczenie

Na tak zbudowanych featurach uruchamiana jest implementacja klasyfikatora z biblioteki nltk: *NaiveBayesClassifier()*.

Skuteczność klasyfikatora testowana jest na osobnym pliku dostarczanym wraz ze zbiorem Stanford Twitter.

Plany na najbliższą iterację

- dokładne testy ilościowe klasyfikatora
- poprawienie preprocessingu: dodanie negation handling, lepszy stemming, usuwanie powtarzanych znaków (typu: yeaaaah -> yeah)
- ograniczenie słownika o słowa występujące więcej niż 1 raz (threshold do ustalenia)
- wyświetlanie bieżących cen