Final Project: Real-Time Bitcoin Price Predictions with News Sentiment Analysis

ID2221 Data-Intensive Computing

Group 2: Ya Ting Hu & Zhen Tian

Problem description

Cryptocurrency prices are known to behave differently compared to traditional currencies. Therefore it is extra difficult to determine what leads to the volatility of bitcoin prices. It is a challenge to correctly predict the future bitcoin prices. One way to do so is to collect data from news articles regarding cryptocurrency to see if there are significant effects on the price of bitcoin.

The problem statement of our project is thus to analyse whether and how real-time bitcoin prices are affected by cryptocurrency news. The goal is to predict real-time bitcoin prices with natural language processing, specifically sentiment analysis, on news about cryptocurrency.

Tools

- Data pre-processing: Jupyter Notebook and Sentiment Analysis using NLTK
- Data storage: distributed messaging systems using Kafka and databases using Cassandra
- Data processing: Streaming Data using Spark Streaming and Machine Learning using MLlib
- Development: Python, Scala (Pyspark)

Data

The data is obtained from the following two APIs:

1) Coindesk [1] to retrieve the price of bitcoin in USD.
2) Newsapi [2] to retrieve cryptocurrency news related to bitcoin, where all non-English news, as well as links and images, were filtered out.

To measure the sentiment of each news article VADER (ValenceAware Dictionary and sEntiment Reasoner) Sentiment Intensity Analyzer of NLTK is used [3]. When given a text corpus, VADER outputs three scores for each sentiment, that is, positive, negative, and neutral. A fourth compound score is computed by summing the valence scores of each word in the lexicon and then normalized to be between -1 and +1. The value -1 indicates extreme negative, +1 extreme positive, and 0 neutral. It is a normalized, weighted composite score and is used for the sentiment score for the extracted cryptocurrency news.

The incoming data stream from Kafka Producer to Kafka Consumer consists of:

1) Columns “date”, “price”, and “type”. The price column indicates the bitcoin price and the date column the corresponding date. The type column consists of the string “bitcoin” to indicate the type.
2) Columns “title”, “date”, “sentiment”, and “type”. The title represents the title of the news article, the date indicates when this article is published, the sentiment column consists of the average compound score of all news articles per day, and the type consists of the string “news”.

The final data consists of one column with the bitcoin price and one column with the compound score, both for a specific date.

Methodology

First, we retrieve the latest bitcoin prices with the Coindesk API and recent news articles regarding bitcoin with the Newsapi API. The data of both APIs is processed and merged based on the date. During the evaluation phase, the data is split into training (70%) and test data (30%). A linear regression, decision tree regression, and gradient-boosted tree regression models are trained on the training data with the compound score as the independent variable and bitcoin price as the dependent variable. Model performance is then evaluated on the test data. After the evaluation phase, we predict the real-time bitcoin price based on the full dataset in Cassandra, and the prediction of the bitcoin prices is then returned.

Results and Discussion

Based on a test run executed on October the 23rd, the results of the evaluation phase of our trained models are as follows:

```
----------------Evaluating Models-----------------
Root Mean Squared Error (RMSE) on Linear Regression = 2011.9
R Squared (R2) on Linear Regression = -0.024025
Root Mean Squared Error (RMSE) on Decision Tree Regression = 2291.25
Root Mean Squared Error (RMSE) on Gradient-Boosted Trees Regression = 2554.01
```

Both the root mean squared error and the R squared indicate that the models fit the data very poorly. To improve model performance, adding more training data to Cassandra should help. This is limited due to time constraints of the project and the set amount of data that we can extract from the APIs. Also involving more features besides only sentiment score can help in improving model performance. This is left for future work.

The prediction of the incoming news by the different models is depicted below:

```
--------Output prediction of incoming news--------
Coefficients: [-538.0875666559227]
Intercept: 62760.29111879713
-----------Linear Regression prediciton-----------
+----------+-----------------+
|      date|       prediction|
+----------+-----------------+
|2021-10-22|62760.29111879713|
+----------+-----------------+

-------Decision Tree Regression prediciton--------
+----------+-----------------+
|      date|       prediction|
+----------+-----------------+
|2021-10-22|62861.38059347827|
+----------+-----------------+

---Gradient-Boosted Trees Regression prediciton---
+----------+----------------+
|      date|      prediction|
+----------+----------------+
|2021-10-22|62860.6895681555|
+----------+----------------+
```

How to run the code

Requirements: Java 8 SDK, Python 3, Apache Kafka, Apache Spark, Apache Cassandra, Python library, Spark library

Steps:

1. Start ZooKeeper server
   zookeeper-server-start.sh $KAFKA_HOME/config/zookeeper.properties
2. Start Kafka server
   kafka-server-start.sh $KAFKA_HOME/config/server.properties
3. Create running topic called bitcoin for producer
   kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic bitcoin
4. Create running topic called news for producer
   kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic news
5. Start Cassandra
   cassandra -f
6. Run consumer.ipynb
7. Run generator.ipynb (pre-processing and producer)

References

1. Coindesk: https://api.coindesk.com/v1/bpi/currentprice.json
2. Newsapi: https://newsapi.org/v2/everything
3. VADER: https://www.nltk.org/_modules/nltk/sentiment/vader.html