## 一.项目背景

比特币是目前上线时间最长且最被人所熟悉的加密式虚拟货币。它由中本聪于 2009 年发布。比特币作为一种分散式的数字交易媒介,是由公共地分布式分类帐(即区块链)中进行校验和存储,而不是像现实货币一样有可信的机构或中央机构所发行且保存。与市场上的任何货币一样,比特币交易及其相关的金融工具在很短时间内就被公众所接受。

## 二. 问题描述

我计划,通过研究和模拟比特币交易,从中分析出具有参考意义的交易规则和现象。

## 三.输入数据

数据来源: https://www.kaggle.com/mczielinski/bitcoin-historical-data

整个数据集有 218MB, 其中包含 7 个子集

- 1) bitstampUSD\_1-min\_data\_2012-01-01\_to\_2017-05-31.csv 从 2012-01-01 至 2017-05-31 每秒通过交易平台 bitstamp 所发生的比特币交易(美元为单位)
- 2) btceUSD\_1-min\_data\_2012-01-01\_to\_2017-05-31.csv 从 2012-01-01 至 2017-05-31 每秒通过交易平台 btce 所发生的比特币交易(美元为单位)
- 3) btcnCNY\_1-min\_data\_2012-01-01\_to\_2017-05-31.csv 从 2012-01-01 至 2017-05-31 每秒通过交易平台 btcn 所发生的比特币交易(人民币为单位)
- 4) coinbaseUSD\_1-min\_data\_2014-12-01\_to\_2017-05-31.csv 从 2014-12-01 至 2017-05-31 每秒通过交易平台 coinbase 所发生的比特币交易(美元为单位)
- 5) coincheckJPY\_1-min\_data\_2014-10-31\_to\_2017-05-31.csv 从 2014-10-31 至 2017-05-31 每秒通过交易平台 coincheck 所发生的比特币交易 (日元为单位)
- 6) krakenEUR\_1-min\_data\_2014-01-08\_to\_2017-05-31.csv 从 2014-01-08 至 2017-05-31 每秒通过交易平台 kraken 所发生的比特币交易(欧元为单位)
- 7) krakenUSD\_1-min\_data\_2014-01-07\_to\_2017-05-31.csv 从 2014-01-07 至 2017-05-31 每秒通过交易平台 kraken 所发生的比特币交易(美元为单位)

#### 其中, 每个数据集均以以下格式呈现:

1) Timestamp:交易时间戳

Open: 开盘价
High: 最高价

4) Low: 最低价

5) Volume\_(BTC): 交易数量

6) Volume\_(Currency): 交易金额 7) Weighted Price: 权重金额

## 四.解决办法

通过区分交易的 Open, High, Low 和 Close 价格, 预测未知的 Volume\_(BTC)交易数量,从而制定有效的交易策略。

#### 五.基准模型

通过已有的评估模型 <a href="https://www.kaggle.com/fedecape/predict-btc-price-movement">https://www.kaggle.com/fedecape/predict-btc-price-movement</a> 使用我的模型与之预测结果进行比较

#### 六. 评估指标

我计划使用与基准模型相同的评估参数 precision, recall, f1-score 及 support 来对我的模型进行验证

# 七.设计大纲

经过初步分析所获取的数据, 我得出对各个数据子集的公有特征:每个子集分别代表对应的交易平台的相关币种在一段特定时间内的交易数据, 且各个子集内的特征列均相同, 我计划通过子集 btcnCNY\_1-min\_data\_2012-01-01\_to\_2017-05-31.csv 预测在平台btcn 的交易数据, 步骤包含:

- 1) 数据清洗
  - 经过分析,数据集中包含了大量的 NaN 数据,此类数据并不包含在分析预测范围内,所以将去除这些数据
- 2) 数据图示

按照目标特征 Open, High, Low 和 Close, 分别对其进行图示化, 分析其走向趋势区分训练数据集和测试数据集

使用 train\_test\_split 区分训练数据集和测试数据集

- 3) 使用模型对训练数据集进行训练 使用 sklearn 中的相关算法训练数据集
- 4) 测试模型

使用测试数据集对模型进行测试

- 5) 获取评估参数
  - 计算得出 precision, recall, f1-score 及 support
- 6) 得出结论

根据模型预测结果对比基准模型得出结论