Big Data and Machine Learning With Applications to Economics and Finance

Short-Term Price Prediction Using Tick-Level Data and Microstructural Features

Mikhail Mironov, Maxim Shibanov

September 29, 2024

Abstract

We want to study cryptocurrency market, namely trading activity on Binance which is the biggest in terms of trading volume exchange. In this work we are attempting to predict prices for cryptocurrencies N seconds ahead using tick level data collected from the exchange. We will create various microstructure features to extract the most information from the data. Then, we aim to construct multiple regression/ranking type models that will be able to yield superior returns.

Introduction

In this project, we're focusing specifically on Binance - the largest exchange in terms of trading volume. Our main goal is to predict short-term price movements for various cryptocurrencies, looking N seconds ahead. To do this, we'll be using high-frequency tick-level data collected directly from the exchange, which gives us detailed information about each trade and order.

We plan to build out a set of microstructure features from this data, which will help us capture useful patterns that can give us an ability to predicting prices. Once we've extracted these features, we'll use them to create different different models as regression, ranking, and classification models—aimed at making accurate price predictions and eliminate probability of loss.

The bigger picture here is to see if we can use these predictions to generate positive excess returns trading returns. By carefully crafting our models and fine-tuning them, we hope to find strategies that not only predict prices effectively but also lead to improved trading performance. In the end, we're going to create portfolio according to the trading strategy based on our model ensemble that will yield excess returns.

Section 2

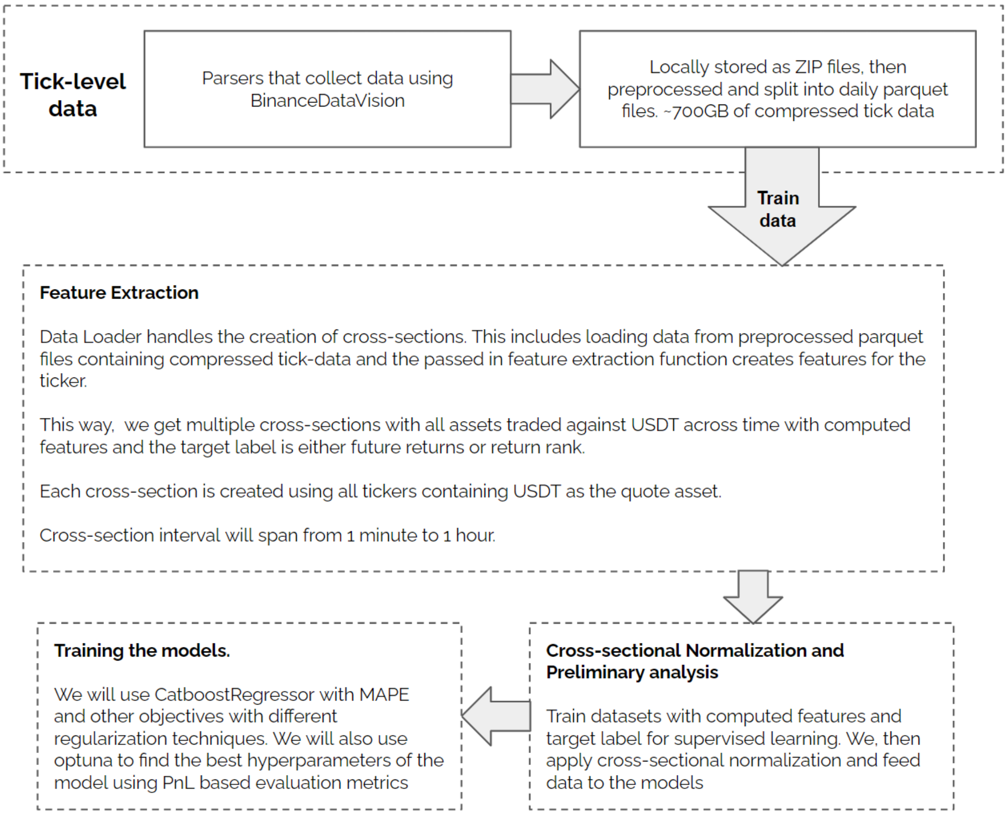
Our problem will be either a regression or ranking problem. In the regression we will be predicting N seconds asset returns using features computed from collected tick level data. We will use RMSE or MAPE or some other suitable regression loss function, we will try and test multiple of them and choose the one that helps to maximize returns of our strategy. As evaluation metrics we will most likely use profit-loss based metrics or some other metric measuring rewards to risk like Sharpe ratio or any other.

But this task can also be stated as a ranking problem where we aim to rank assets correctly based on their N seconds future returns. In this case we will be using loss functions like NDCG loss which in Catboost is approximated by differentiable YetiRank or LambdaMART.

Our problem is obviously falling into the group of supervised learning task where we have a label either a return that we want to predict or correct rankings of the assets based on their future returns. In our task we require high quality of predictions, therefore we would like to use more complex, blackbox models that sacrifice some of the tractability in favor of higher predictive power.

Section 3

We will collect tick level data from Binance, namely we will use BinanceDataVision website which has historical data stored as compressed zip files. We already have all the code to parse all of the data from this website. We also have partially finished pipelines to handle this data, namely, unzipping, preprocessing files, computing features and applying various transformations. We will have 700GB of compressed data collected from Binance exchange, since we would like to ideally deploy the model on Binances it makes perfect sense to use data from the same exchange.



Parsers that collect data using BinanceDataVision

Locally stored as ZIP files, then preprocessed and split into daily parquet files. ~700GB of compressed tick data

Data Loader handles the creation of cross-sections. This includes loading data from preprocessed parquet files containing compressed tick-data and the passed in feature extraction function creates features for the ticker.

This way, we get multiple cross-sections with all assets traded against USDT across time with computed features and the target label is either future returns or return rank.

Each cross-section is created using all tickers containing USDT as the quote asset.

Cross-section interval will span from 1 minute to 1 hour.

Train datasets with computed features and target label for supervised learning. We, then apply cross-sectional normalization and feed data to the models

We will use CatboostRegressor with MAPE and other objectives with different regularization techniques. We will also use optuna to find the best hyperparameters of the model using PnL based evaluation metrics

Figure 1: Data pipeline