Assignment 2

This assignment continues where assisgnment 1 ended. The goal is to improve the price forecasting model by adding another stream of information. For this, you should use some online source of data that can be automatically pulled at regular intervals.

However, it is quite unlikely that this new data set has the same length as the data set you used for assignment 1. For assignment 1 you had around 9 years of bitcoin price data, but what if the new data set you find only covers 2-3 years? How do you combine them? Are you going to throw out all bitcoin price data older than the start of your new data set in order to ensure that the two data sets have the same length?

Instead of throwing out data, you can train two models, one on each data set and use a simple (e.g. linear) model to combine the two forcasts to a single improved forecast.

Task

- Provide code that automatically pulls data from internet, processes it, and returns it in a format that can be used as input to an sklearn model.
- Use a simple model (regression or classification) to combine the predictions from your Assignment 1 model with the new data/forecasts.

Example sources

- Sentiment analysis on Bitcoin-related tweets
  - See FastAI text for a quick and simple implementation.
  - Contact me for twitter keys
- Google trends: Bitcoin
  - `pytrends` provides a Python API
- Crypto Fear & Greed Index
  - Use Python `requests` and the API
- Other cryptocurrency prices: Coinmarketcap
- Technical indicators from CoinGecko

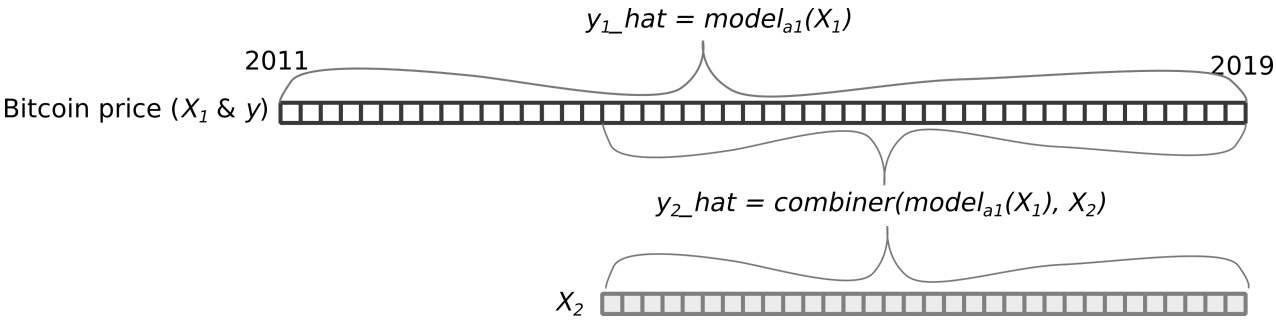
Submission & test

Once submitted, your model will be tested against previously unseen data. Your goal is to beat the **naive forecast** on the test set.

The submission should be a `PY` file and the trained model weights. The `PY` file should have three functions: `preprocess()`, `model()` and `combiner()`.

Functions

- `preprocess()`
  - **Description**: loads, downloads and perprocesses the two data sets.
  - **Arguments**: a list of names of the data (`csv`) files (the list can have length 1).
  - **Returns**:
    - `X1` the bitcoin price as input to `model()` (see below).
    - `X2`: the additional data, pulled from the internet.
    - `y`: the bitcoin price as targets, i.e. the true bitcoin price (to be used for training and evaluation).
- `model()`
  - **Description**: loads the model weights from Assignment 1 ($model_{a1}$) and returns predictions, `y1_hat`, given inputs `X1`.
  - **Arguments**:
    - `filename_a1`, the file name of the weights for $model_{a1}$
    - `X1`, the input to $model_{a1}$ (returned by `preprocess()`).
  - **Returns**:
    - `y1_hat`: a numpy array with bitcoin price predictions.
- `combiner()`
  - **Description**: loads the weights for the combining model and returns the final forecast. Note that `y1_hat` and `X2` have to be temporally aligned.
  - **Arguments**:
    - `y1_hat`: the output of $model_{a1}$
    - `X2` - the new data
    - `filename_comb` - the filename of the weights.
  - **Returns**: `y2_hat`, a numpy array with bitcoin price predictions (hopefully better than `y1_hat`), it will be evaluated against `y`.



The assignment is due at 2:30 pm on March 9.