Stock Indicator

Vincent Scala and Ozzie Martin

Stock Indicator Program

Using historical data we will make predictions on how the market will perform day-to-day.

Historical data can elude to how the market will perform in the future, for example, if the market has recently been bullish we may come to the conclusion that it will continue to perform this way.

It is important to note that there is no perfect model and due to the use of computers in the stock market basic models may underperform the market.

Stock Indicator Program

We will be exploring the basic moving average indicator and how it performs against the market.

We will discuss

1. Program Structure
2. Data Parsing
3. Indicator Computations
4. Output and Observations

Program Structure

- Stock Data
  - Constructor with stock symbol
  - API key setter
  - Populate Data method
  - Get a StockDataEntry by day
- Stock Indicators
  - Constructor with stock symbol
  - Get nth day average
- Stock Model (not yet implemented)
  - Create and populate a model based on some running average
  - Comparison between a running average model and actual market performance

Data Parsing: API Call

API calls used a c++ version of Libcurl and the Alpha Vantage API

1. Curl is initialized with the address of the Alpha Vantage API
2. Headers are set based on the users API key
3. Data is read into a read buffer

Data is received in the CSV format

```
void initializeCurl() {
    easyhandle = curl_easy_init();
    curl_global_init(CURL_GLOBAL_ALL);
    curl_easy_setopt(easyhandle, CURLOPT_CUSTOMREQUEST, "GET");
    curl_easy_setopt(easyhandle, CURLOPT_URL, url.c_str());
}

void setHeaders() {
    string keyheader = "x-rapidapi-key: " + key;
    headers = curl_slist_append(headers, keyheader.c_str());
    headers = curl_slist_append(headers, "x-rapidapi-host: alpha-vantage.p.rapidapi.com");
    curl_easy_setopt(easyhandle, CURLOPT_HTTPHEADER, headers);
}

string readData() {
    string readBuffer;
    curl_easy_setopt(easyhandle, CURLOPT_WRITEFUNCTION, WriteCallback);
    curl_easy_setopt(easyhandle, CURLOPT_WRITEDATA, &readBuffer);
    curl_easy_perform(easyhandle);
    return readBuffer;
}
```

Data Parsing

1. Data is received from Alpha Vantage as a CSV format
2. Text is split and stored in StockDataEntry structure
3. Entries are stored in a vector

```
struct StockDataEntry {
    string data;
    float open;
    float high;
    float low;
    float close;
    float adjusted_close;
    float volume;
    float dividend;
    float split_coefficient;
    StockDataEntry() {}
    StockDataEntry(string data, float open, float high, float low, float close,
                   float adjusted_close, float volume, float dividend, float split_coefficient) {
        this->data = data; this->open = open; this->high = high; this->low = low;
        this->close = close; this->adjusted_close = adjusted_close; this->volume = volume;
        this->dividend = dividend; this->split_coefficient = split_coefficient;
    }
};
```

Structure used to store a data entry

Indicators

MA (Moving Average) Indicator, these indicators will compute the average of the last n-days and use this to determine whether or not to stay in the market.

- 10 day moving average
  - Used for short time horizons and volatile markets
- 100 day moving average
  - Used for moderate time horizons
- 200 day moving average
  - Used for long time horizons and stable markets

Indicator Computation

N-Day Average is computed by selecting a time frame (some date **start** and the number of days before it **n**).

The algorithm used adds up the dates from start to start+n and divides it by the count of dates.

(Another implementation using the deque data structure may be better when computing moving averages in succession)

```
float getNDayAverage(int n, int start = 0) {
    float sum = 0;
    int i;
    for (i = start; i < n+start; i++) {
        if (i >= datev.size()) { i--; break; }
        sum += data[datev[i]].close;
    }
    return sum / (i+1-start);
}
```

Observations

Generally this model of stock prediction using Moving Averages performs poorly in comparison to other models and even the natural growth of the market, however it is a good starting point for making more powerful models.

A natural progression of this project would be to construct a model using the computed moving averages, or furthermore the use of a volatility index such as VIX to determine the period of the moving average rather than having it fixed.