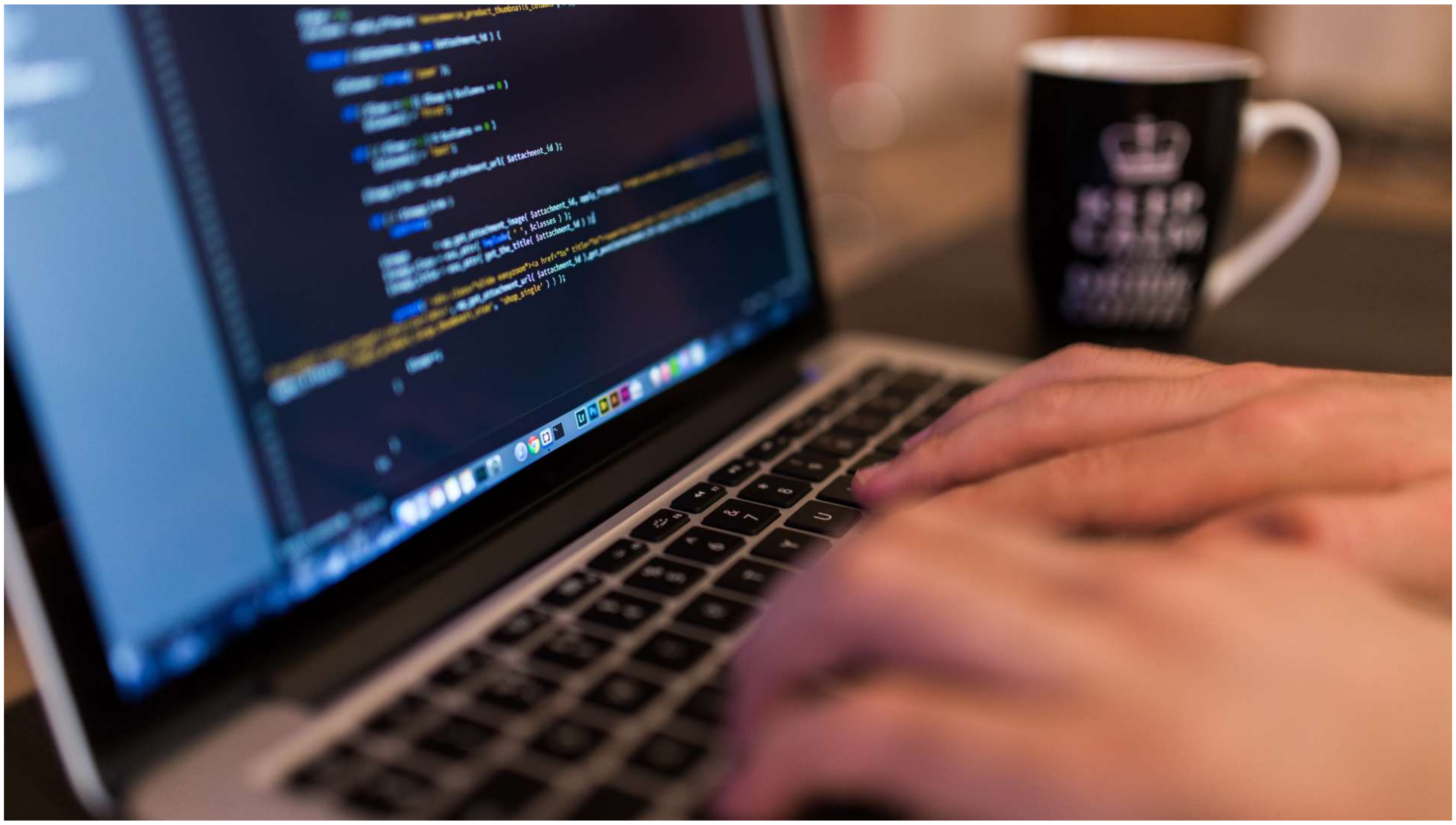
Python for Trading — How I Used Classification Algorithms to Enhance My Winning Rate By 22.4% Easily

A walk-through of a misknown machine learning algorithm used by Netflix, to increase the robustness of my technical analysis and trade like a high-frequency trader.

Sajid Lhessani Follow

Sep 1 · 11 min read ★



Due to the length of the article and the content's density, I chose to divide this article into two parts. The second part will be released in 5 days. In this first part, we will go through the two first steps needed to develop this algorithm. Enjoy!

. . .

This article will be a bit more advanced mathematically speaking than usual. If you are new in this field, at the end of this article, on top of getting the full explanation, code and details, you will be able to get a general overview of how the classification algorithms are working.

During the last few months, I published a series of articles either on Medium or through hedge-fund letters, about some algorithms that I am using in my trading arsenal.

Today the algorithm will be more sophisticated. However, if you start with coding or you are eager to know more about this field, that can be an exciting challenge to solve.

If you follow it step by step, you will be stronger in algorithmic trading and data science.

At the end of this article, you will have a video tutorial (where you will have further explanations) published on Youtube, and a link to download the Python script associated.

Before to start

So before to start …

If you want to go further and take a look at some further examples, you can go through the articles below:

Python: I have tested a Trading Mathematical Technic in RealTime.

I have tested in real-time the implementation coded with Python of a famous mathematical technics to predict market…

medium.com

How I Tripled My Return on Bitcoin Using Mathematics, Algorithms, and Python

Using the Golden Line algorithm, I predicted Bitcoin, Ripple, Ethereum & Dogecoin price evolutions. Full Python code…

levelup.gitconnected.com

Both articles cover some interesting and robust algorithms, and you can have a live trade example on how to predict a market movement easily. The first one is based on the US Market (SP500 + random shares) and the second on cryptocurrencies.

The added value of the algorithm presented today is not to generate a new way of predicting the market but to **secure** and **magnify** the current technical analysis.

. . .

Classification algorithm and SVM

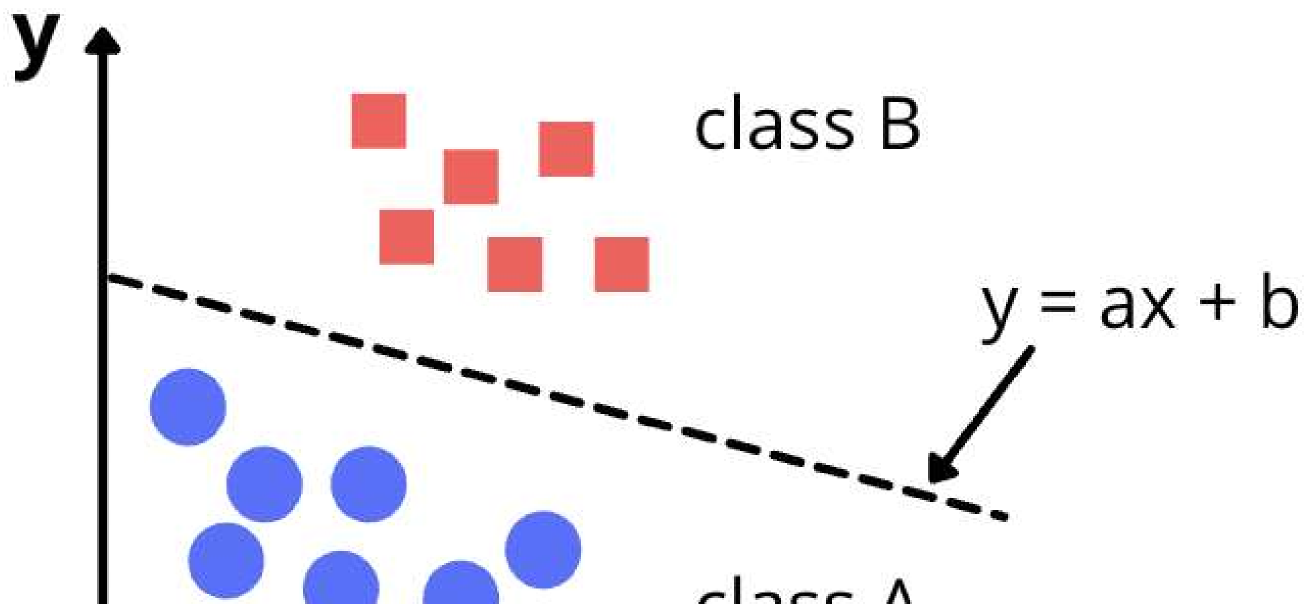
The algorithm which will be covered in this article is the SVM called "Support Vector Machine". SVM combines statistical technics invented by Vladimir Vapnik, Alexeï Tchervonenkis in the '90s, belonging to the family of classification's algorithms.

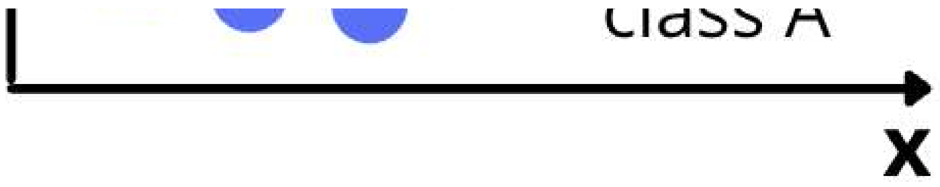
Before implementing, let review a bit of theory about classification and SVM …

What is a classification algorithm?

A classification is an approach of machine learning, which consist of allocating a label to each data point. This involves taking input and running it into a classification technique or a classifier to map it into a discrete class or category.

Below is a short example:





Above is an example of a Binary classification

Here is an example of a primary binary classification. In red are located the high performers, and in blue the low performers. The classification drew a line called **a hyperplane** (mathematically speaking: a function) to separate the 2 clusters.

The goal of a classification algorithm is to detect a defined number of clusters that have the same pattern to predict an output.

In our daily lives, classification algorithms are used in various fields, and the most common are **medical diagnosis**, **spam**, **fraud detection**, **handwriting recognition**, **customer segmentation**, **risk assessment** etc.

. . .

Now, let's ask ourselves how to use these mathematical approaches to help our trading decision?

If you take a look at the scientific literature, five main approaches are commonly used to apply classification to real life:

- K-nearest neighbours algorithm (KNN),
- Random forests using decision trees,
- **Support Vector Machine (SVM),**
- Artificial neural networks (ANN),
- Naive Bayes classification.

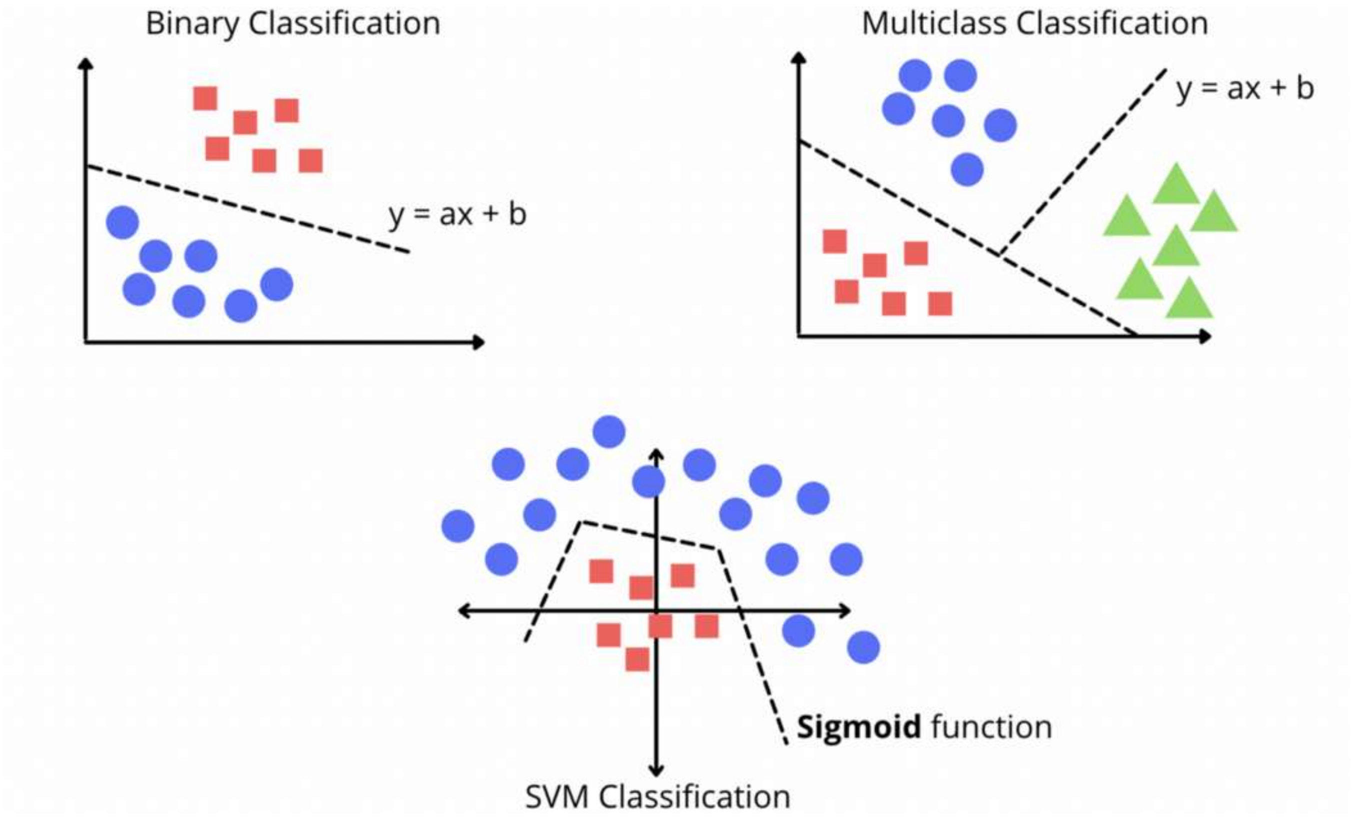
The approach which will be pertinent to our case is the **SVM**. According to the literature, this approach is getting more and more used in the field of trading. According to google scholar, citations have increased by more than 125% since last year.

Before implementing our approach, let's look at how this approach is working and the main difference with the classification mentioned above.

Difference SVM vs Binary/Multiclass classification

The polynomial degree is the significant difference between the algorithm we will use (SVM algorithm) and the usual classification technics.

Let's introduce the difference through a graph:



In the third graph above, a linear boundary will not separate the negative outcomes from the positive ones. In this case, we will have to use a nonlinear decision boundary for making the classification.

The SVM classification will work on optimising the distance with the boundary by using a third-degree, a quadratic or even a **sigmoid** equation. Support Vector

Machine will choose an **optimal line that maximises the distance between the boundary and the nearest points** in either class.

That's why in the scholar world, the Support Vector Machine is sometimes called a Maximum Margin Classifier.

This example shows that we can build more complex decision boundaries by fitting complex parameters to the relatively simple sigmoid function. By using higher-order polynomial terms, we can get complex decision boundaries like the one shown in the plot above.

If you want more information, you can go over this link.

. . .

Now that we have covered the mathematical intuition behind it, let's start the test and see how that can help to predict the market.

> *If you want to code at the same pace, there is some configuration and packages that will need to have installed prior.*

Furthermore, if you are too impatient and want to know how to implement this strategy quickly, you can directly look at the recording at the end of this article. I am applying this strategy live, and you can look at the result by yourself :).

I recorded myself explaining the theory, code and gave extra tips on the way to develop it. You can apply it at home by yourself.

Coding

This part is made to help you step by step to develop this algorithm. If you are not interested in the code, you can jump to the results or the link to the trading robot.

Prerequisite:

First of all, if you want to track my progress and before starting, you will need to have installed on your machine a **Python 3** version and the following packages:

- **Pandas**

- **NumPy**
- **Yfinance**
- **Scikit-learn**
- **Ta-Lib**
- **Plotly** (*Not mandatory, but valuable for plotting*)

If any of these packages are not already installed, you can use the pip command, as shown below.

```
pip install yfinance
pip install plotly
pip install -U scikit-learn
```

And for Mac users:

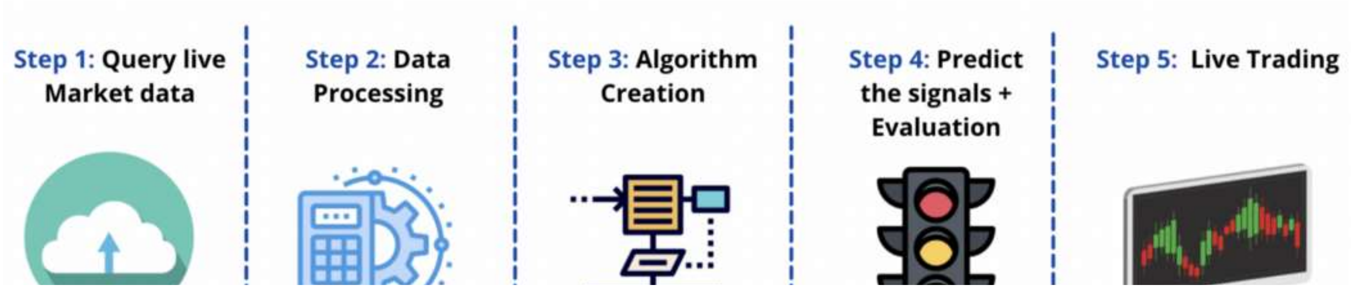
```
pip install yfinance
pip install plotly
pip install -U numpy scipy scikit-learn
```

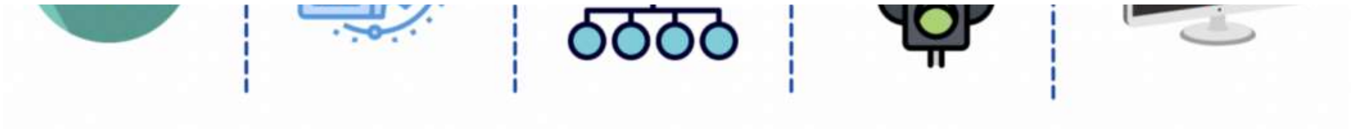
Once you have ensured that the following packages are installed, we can discuss our data pipeline and model.

Data pipeline and modelling:

Now that we have confirmed that the packages above are installed within our machine, we can define the data processing.

The data model will be divided into 5 distinct steps.


Step 1: Query live Market data
Step 2: Data Processing
Step 3: Algorithm Creation
Step 4: Predict the signals + Evaluation
Step 5: Live Trading



First of all, we will query live cryptocurrency's data using **Yahoo Finance API**.

Second, define a period of time, create new columns for our calculated fields, and update these values each second.

Third, draw this graph live, and check if our signals are accurate.

During this article, I will not go too much into details regarding the code and API; you can find more information on how to get live market data reading the article below:

Python: How to Get Live Market Data (Less Than 0.1-Second Lag).

This article is going to be a bit special. I am going to test the latest release from Yahoo Finance API for Python…

towardsdatascience.com

Now that we have a clear plan for our model, we can start coding!

> **If you already have experience with Python, you can jump to the second step. The first step will consist of importing packages and input data.**

Step 1: Import required packages and market data.

The first step will consist of importing the necessary packages and our initial entry data.

You will start by importing your packages previously installed by using the following lines of code:

Once we are set up, and no errors are raised, let's pursue the next step.

Now that libraries are imported, we can now import our initial entry data.

Import market data

Now that the different packages needed have been uploaded. We are going to use **Tesla**to test our hypothesis.

So, we will call live market data using ***Yahoo Finance API***.

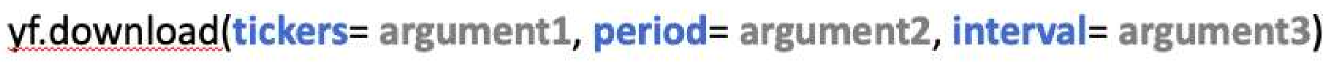
For your information, Yahoo Finance API will need 3 mandatory arguments (*more information in the article above*) in this order:

- Tickers *(1)*
- *Start date* + *End date* or Period *(2)*
- Interval *(3)*

For our case, the ticker*(argument 1)* will be **TSLA**. Furthermore, we will choose for our test a period*(argument 2)* to get the latest data instead of defining a Start and End date. And we will set up an interval*(argument 3)* of 1 minute to stay in our optic to perform as a day trader. *(You can even choose a shorter period, but one minute is optimal)*

*As a quick reminder, Tesla's ticker is **TSLA**.*

To call your data, you will have to use the following structure:



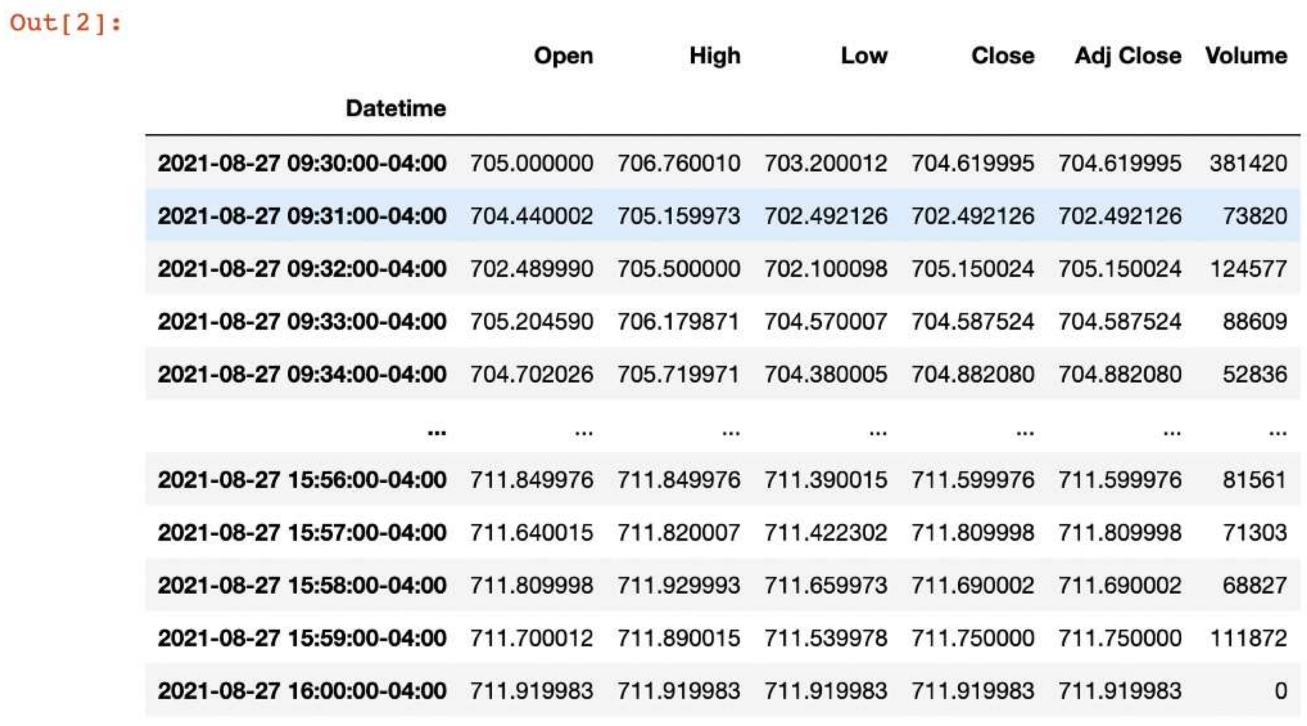
```
yf.download(tickers= argument1, period= argument2, interval= argument3)
```

The above described the structure to get market data that interest us.

Now that we have the structure let's run the code.

Line of code to download market data

Below is a sample of the output you should get:

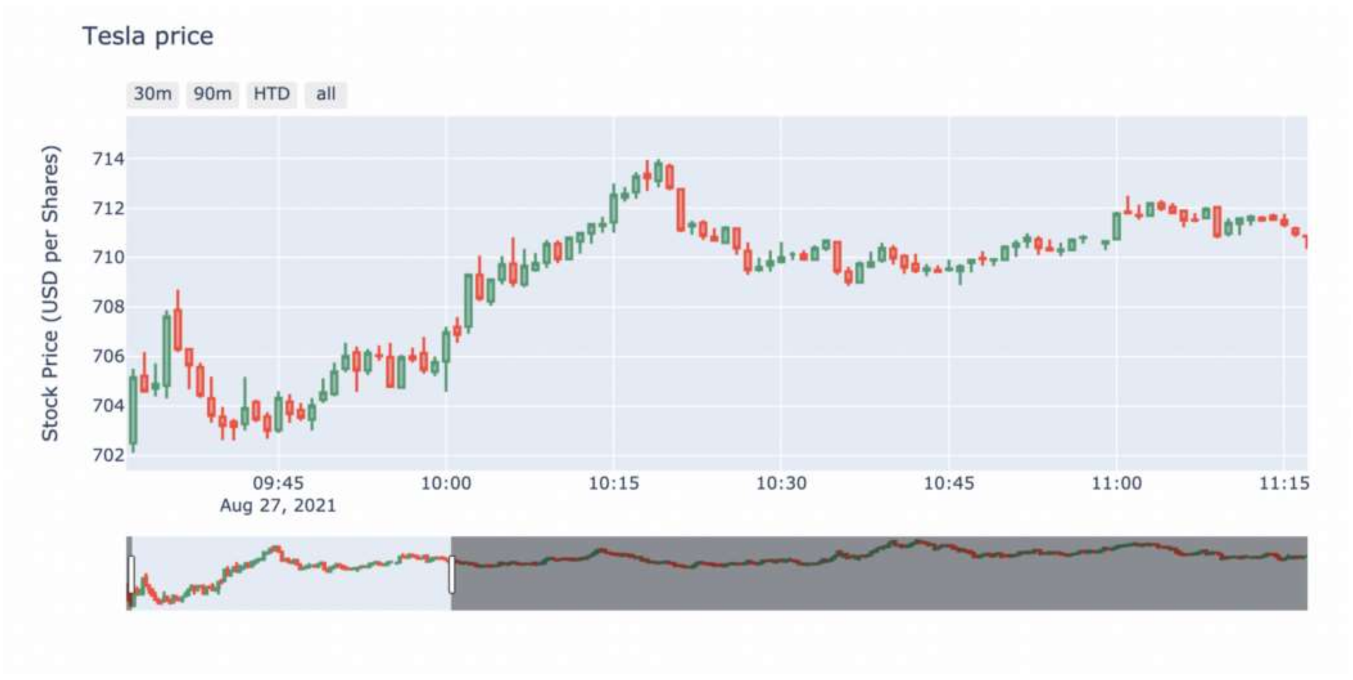


```
Out[2]:
                                 Open        High         Low       Close   Adj Close  Volume
                 Datetime
2021-08-27 09:30:00-04:00  705.000000  706.760010  703.200012  704.619995  704.619995  381420
2021-08-27 09:31:00-04:00  704.440002  705.159973  702.492126  702.492126  702.492126   73820
2021-08-27 09:32:00-04:00  702.489990  705.500000  702.100098  705.150024  705.150024  124577
2021-08-27 09:33:00-04:00  705.204590  706.179871  704.570007  704.587524  704.587524   88609
2021-08-27 09:34:00-04:00  704.702026  705.719971  704.380005  704.882080  704.882080   52836
                      ...         ...         ...         ...         ...         ...     ...
2021-08-27 15:56:00-04:00  711.849976  711.849976  711.390015  711.599976  711.599976   81561
2021-08-27 15:57:00-04:00  711.640015  711.820007  711.422302  711.809998  711.809998   71303
2021-08-27 15:58:00-04:00  711.809998  711.929993  711.659973  711.690002  711.690002   68827
2021-08-27 15:59:00-04:00  711.700012  711.890015  711.539978  711.750000  711.750000  111872
2021-08-27 16:00:00-04:00  711.919983  711.919983  711.919983  711.919983  711.919983       0
```

390 rows × 6 columns

Output sample

Now that we have downloaded and stored our data, we can take a quick look at how the data looks like on a graph and start the data processing:

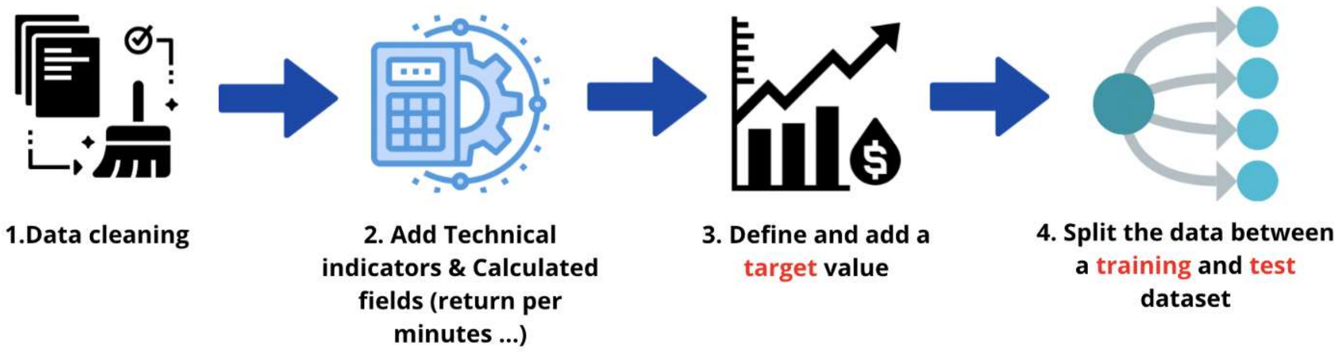


So everything looks good; we can process to the second step :).

. . .

Step 2: Data processing

The data processing part will work in 4 steps:


1.Data cleaning
2. Add Technical indicators & Calculated fields (return per minutes ...)
3. Define and add a target value
4. Split the data between a training and test dataset

First, we will need to clean our data and remove zeros and NA values. Then, index our values and add market indicators. Third, set up a target value (we will talk about the concept of “target value” in the second article) to evaluate our model. Fourth, divide our dataset between a training set and a test set.

1. Data cleaning

Before starting developing our algorithm, a bit of data preparation is required. Indeed if you want your algorithm to perform well, the quality of data is crucial. And unfortunately, it can sometimes appear a little bug in Yahoo Finance tables when you trade live.

The data provided by the API are clean overall, but one variable can cause ***issues***.

The column that will need cleaning is the **volume traded**.

Indeed, because the period is short, at some point in our dataset, Yahoo Finance is showing a volume traded of 0. If we keep it as it is, it might generate outliers which will corrupt the algorithm.

If you want to fix this issue, here we go :

The command above will drop the line for zeros volume traded

Index values and add extra variables

Now that we have a clean dataset, it is time to give your algorithm extra variables to help him understand the market.

We will add calculated fields such as technical indicators and other influential variables such as the return per minute, correlation coefficient, moving averages, previous minutes information …

> *Within the section below, I kept only the most interesting and influencal variables after going through a sensitivity test. In this first part only the RSI will be covered. The other variable will be discussed in the second article.*

The technical indicator that we will combine with our SVM algorithm is the RSI.

For people who are not familiar with trading, the RSI (Relative Strength Index) is a well-known trading indicator based on momentum strategy.

It will analyse the market variation and give the relative strength index (RSI). Here is the mathematics behind it:

$$RSI = 1 - \frac{100}{1 + \mathit{Relative\ strength}\ (RS)}$$

$$RS(n) = \frac{\mathit{Average\ Gain}\ (n)}{\mathit{Average\ Loss}\ (n)}$$

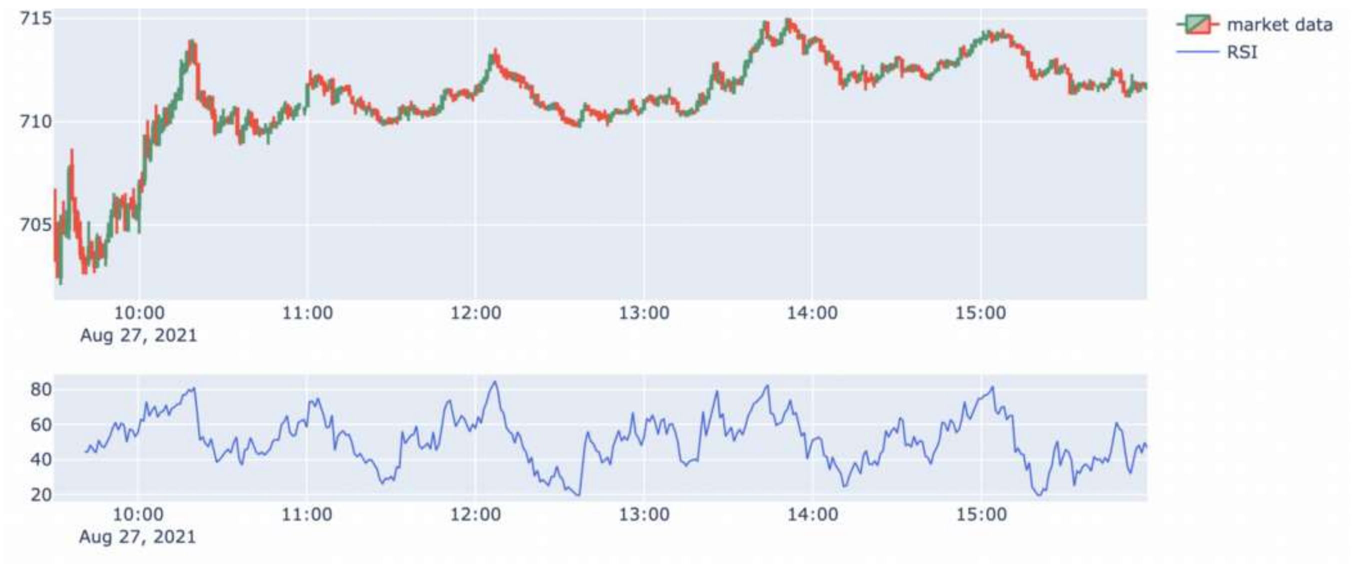
n represents the number period of time.

However, today, no need to develop mathematical calculations manually. We are going to take a shortcut by using the Ta-Lib library. The only factors that we will need to define are the **period of time and the number of periods of time**.

The period of time that we will choose for our day trading optic **is 1 minute**, and the **number of periods is 14**.

So to do so, we execute the line below, and here we go:

Let’s visualise it! (Full visualisation code in the video tutorial below)



All good.

It is interesting to see the pattern between buy and sell of the trading indicator.

On top of using the RSI, we will add extra variables to make the algorithm more robust and less influenced by outliers.

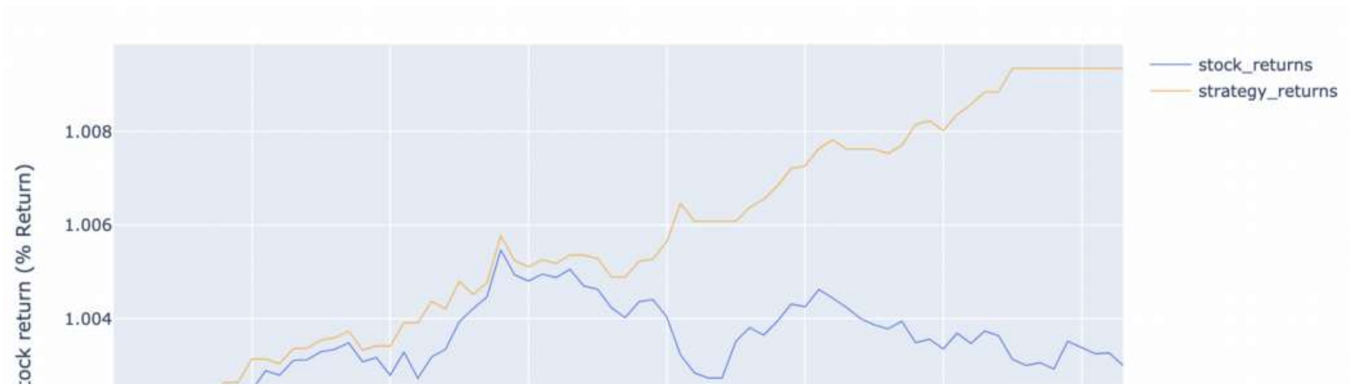
These variables will be discussed in the second part. The second part will be published on **Wednesday, 7th September 2021.** So, think to subscribe on **Medium** or **Youtube** to get the update.

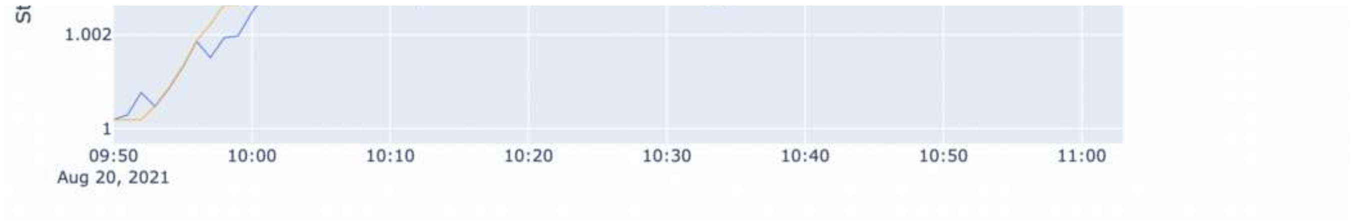
I love you all. I hope you like this first part; the second part will be even more exciting.

In the next article …

In the second part, we will develop our machine learning and run our AI robot for trading!

Below, you can get a sample of the final results:





You can access the full video tutorial below, with further explanations if you want to develop it at home:

Tutorial Video

Conclusion

At the end of this first article, we have covered the following point:

- The intuition behind classification algorithm
- How to import market data
- A quick view of the RSI
- Some of the most influential factors to a market movement

The next article will be released in the next few days; I did not want to make it too heavy to read, just think to subscribe to don't miss it. I hope you enjoy this article, see you soon.

Happy coding,

Sajid

. . .

To go further:

Recommended content

If you want to go further in this field, I can recommend both courses below:

Machine Learning and Deep Learning for Financial Market

Algorithmic trading for everyone

Trading with Machine Learning Regression

Need a one to one lesson

If you need a specific 1 to 1 session to start with algorithmic trading and data science:

Sajid - London,Greater London : Data Scientist working in Banking and Capital Market. Youtube...

The added value that I propose to my student is based on real-life projects. The goal is to come from beginner to…

www.superprof.co.uk

Follow my automated trading robot.

Here is a sample of my trading bot to get daily live signals on the crypto market:

Cryptop

Link - https://t.me/joinchat/IIQwifpza3BjZTE0

t.me

Complete Python code — Part 1/2