Famulus - building a stock predictor

Author: Vedang Joshi

Published: May 9, 2021 · 4 min read

About a year back, my friends and I participated in a hackathon and received the following problem statement.

> Stock Market Trend Prediction using an Automated News Analysis

I won't narrate the story about the happenings of the hackathon, but I would surely say 2 things:

- NMIMS is the most luxurious college in Indore.
- We didn't win this hackathon, but we did secure 3rd place, unfortunately, there were prizes only for the top 2 :(

So, here is what we built.

Famulus

> Famulus means an assistant working for a scholar.

Famulus is an easy-to-use web application, where the user enters a brand name (company), our web app scraps news related to it, analyses it and predicts the change in the stock prize in an interval of 5, 7, 15, and 30 days. Displaying 3 things on the UI - Change in the stock price, predicted graph and the 3 point news summary.

How does it work ?

The tech stack used here is

- Python: machine learning stuff
- Flask
- React

Machine learning model

Various components are working together to give the final output.

1. Sentiment analyzer - This was an easy task, we took the very popular Twitter dataset and trained a model which outputs the sentiment score of a text, -1 to 0 being negative and 0 to 1 being positive. We tested it on a small sample of news headlines and it worked quite well.
2. We downloaded this Historic Indian news dataset and filtered out all business news. Then put this through our sentiment analyzer and got a sentiment score for each news headline. The final dataset we obtained here was
   ```
   { date, news_headline, sentiment_score }
   ```
3. Then we downloaded Historic Sensex data from BSE (Bombay stock exchange) website and merged it with the dataset created above with the index being 'date', so finally, the dataset looked like this
   ```
   { date, (sensex_high + sensex_low)/2, sentiment_score }
   ```
4. Now, comes an interesting part, We have two 2 values - sensex_avg and sentiment_score, to map both of them we can simply use linear regression, but do we need to predict these? NO. This is something we already know. What we need here is to use today's sentiment_score to predict the sensex_avg 5 days from now. We need to shift the sensex_avg column accordingly. So, the dataset now looks like
   ```
   { date, sentiment_score, sensex_avg_date+5, sensex_avg_date+7,
   sensex_avg_date+15, sensex_avg_date+30 }
   ```
5. Most of the work is done. We used linear regression to map sentiment_score to sensex_avg_date+5, and similarly others. We create 4 linear regression models here.

Flask

Now, we use beautifulSoup to scrap the current news, put this news through the sentiment analyzer and obtain a sentiment score. This score is feed to the model and we get price predictions for 5, 7, 15, 30 days. Nice :)

Then we went on to create a summarizer, this takes in a few articles and outputs a 3 point summary for it.

Now, was the time to make all this deliverable. We create a Flask API exposing the prediction value, to plot the graph, we used Altair (Vega) which lets us send a JSON response that can be converted to a graph on the frontend.

Hackathon ends

All the work was done in the hackathon, there were a few patchworks and some things didn't work, but I loved this project. I returned home and it took me about a week to remove all the patchwork and fix things, I wrote test cases and also added a WebSocket endpoint using FastAPI. Here is the link to the repository.

- Backend
- Frontend

Note

Neither I nor my teammates are machine learning experts, so this approach has flaws, If you can provide a better approach, my inbox is always open, and as it always is

Thanks for reading!

Machine Learning   Hackthon Project   Python   Github