Major reasons why machine learning fails in stock prediction : Part — 02

Ved Prakash · Follow

6 min read · Feb 17, 2024

This is continuation of previous article where I have discussed some of the reason why machine learning fails to predict stock price. Here is the link of the previous article : https://ved933409.medium.com/major-reasons-why-machine-learning-fails-in-stock-prediction-part-01-479834eb891d. Please make sure to read the article I shared earlier before proceeding to read this one.

Continuing from the previous article where I was discussing the mistake made by

market. The primary objective of an information-driven bar is to gather information when either buyers or sellers become more active than the other. This indicates that more informed traders are entering the market, and we may be able to make decisions before the market reaches its equilibrium level.

In this section, we will look into different indices of information arrival.

1. **Tick Imbalance bar**

According to the author, " The idea behind tick imbalance bars (TIBs) is to sample bars whenever tick imbalances exceed our expectations". Tick imbalance will give you an idea if price is moving a one direction. If the expected imbalance of movement increases a threshold then we sample the bar.

Consider a sequence of ticks $\{(p_t, v_t)\}_{t=1,\ldots,T}$ where p_t is the price associated with tick t and v_t is the volume associated with tick t. The so-called tick rule defines a sequence $\{b_t\}_{t=1,\ldots,T}$ where

$$b_t = \begin{cases} b_{t-1} & \text{if } \Delta p_t = 0 \\ \dfrac{|\Delta p_t|}{\Delta p_t} & \text{if } \Delta p_t \neq 0 \end{cases}$$

step 2 : Define tick imbalance at time t

we define tick imbalance at time t as

$$\theta_T = \sum_{t=1}^{T} b_t$$

Step 3 : Compute expected value of tick imbalance

expected value of tick imbalance can be computed as

$$E_0(\theta_T) = E_0(T)\left(P(b_t = 1) - P(b_t = -1)\right)$$

where $E_0(T)$ = Expected size of the tick bar

$P(b_t = 1)$ = unconditional Prob. that tick is buy

$P(b_t = -1)$ = unconditional Prob. that tick is sell

$$E_0(\theta_T) = E_0(T)\left(2P(b_t = 1) - 1\right)$$

we can estimate E_0 [T] as an exponentially weighted moving average of T values from prior bars, and (2P[b_t = 1] − 1) as an exponentially weighted moving average

Third, we define a tick imbalance bar (TIB) as a T^*-contiguous subset of ticks such that the following condition is met:

$$T^* = \arg\min_T \left\{ \left|\theta_T\right| \geq \mathrm{E}_0\left[T\right] \left| 2\mathrm{P}\left[b_t = 1\right] - 1 \right| \right\}$$

Python code for generating the tick imbalance

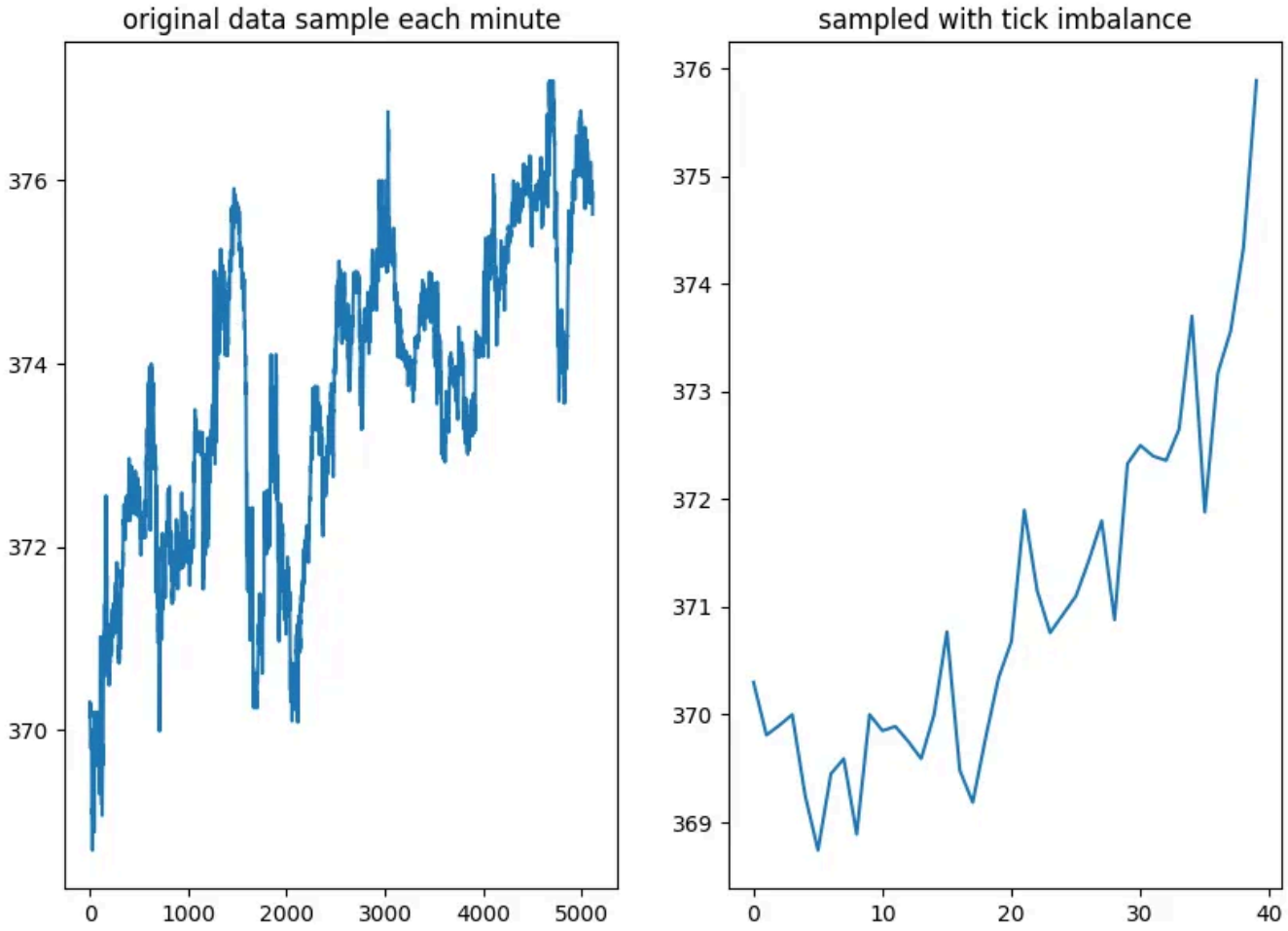
```
## code to create tick imbalance bar
weighted_sum_T = 7
alpha = 0.9
weighted_sum_prob = 0.3
data["delta_p"] = data['open'].pct_change()
imbalance = 2
b_t = np.zeros(len(data["delta_p"]))
b_t[0]=1
indx = []
T=0

for i in range(1,len(data["delta_p"])):
    if data["delta_p"][i] == 0:
        b_t[i]=b_t[i-1]
    else:
        b_t[i] = np.abs(data["delta_p"][i])/data["delta_p"][i]
    T+=1
    if (abs(b_t.sum())>=imbalance):
        indx.append(i)
        weighted_sum_T = alpha*T + (1-alpha)*weighted_sum_T  ### E(T)
        weighted_sum_prob = alpha*sum(x>0 for x in b_t)/(T) + (1 - alpha) * wei
        imbalance = weighted_sum_T * abs(2*weighted_sum_prob-1)

        T = 0
        b_t = np.zeros(len(data["delta_p"]))



final_data=pd.DataFrame()
open =[]
close=[]
volume=[]
for i in range(len(indx)-1):
  ind = (indx[i],indx[i+1])
  temp_data = data[ind[0]:ind[1]]
```



when sampled with Tick imbalance bar we sampled more frequently when market is moving one sided

TIBs are produced more frequently under the presence of informed trading (asymmetric information that triggers one-side trading).

We have many other sampling strategy based on the information coming in to market like

- Volume/Dollar Imbalance Bars : Sampled more frequently when volume or dollar (traded amount value) imbalance is more than our than our exception. this is similar to the above strategy only difference is now we are looking the

the market.

2. Tick Run bar

When a large trader will sweep the order book, use iceberg orders, or slice a parent order into multiple children, all of which leave a trace of runs in the {b_t}t = 1,..., T sequence. To take advantage of the patterns left by large traders, we monitor the sequence of overall buy volume and take samples when that sequence diverges from our expectations.

Steps to create a tick run bar

Step 1 : Define the length of current run as

$$\theta_T = \max\left\{\sum_{t|b_t=1}^{T} b_t, -\sum_{t|b_t=-1}^{T} b_t\right\}$$

Step 2 : Compute the expected value of imbalance at the beginning of the bar

$$\mathrm{E}_0[\theta_T] = \mathrm{E}_0[T]\max\{\mathrm{P}[b_t = 1], 1 - \mathrm{P}[b_t = 1]\}$$

To estimate the value of E_0 [T] and P[b_t = 1], we use an exponential moving average of T values from previous bars and b_t values from previous bars, respectively. This is similar to the process we followed while creating a tick imbalance bar.

step 3 : sample the bar if imbalance will become greater than expectation

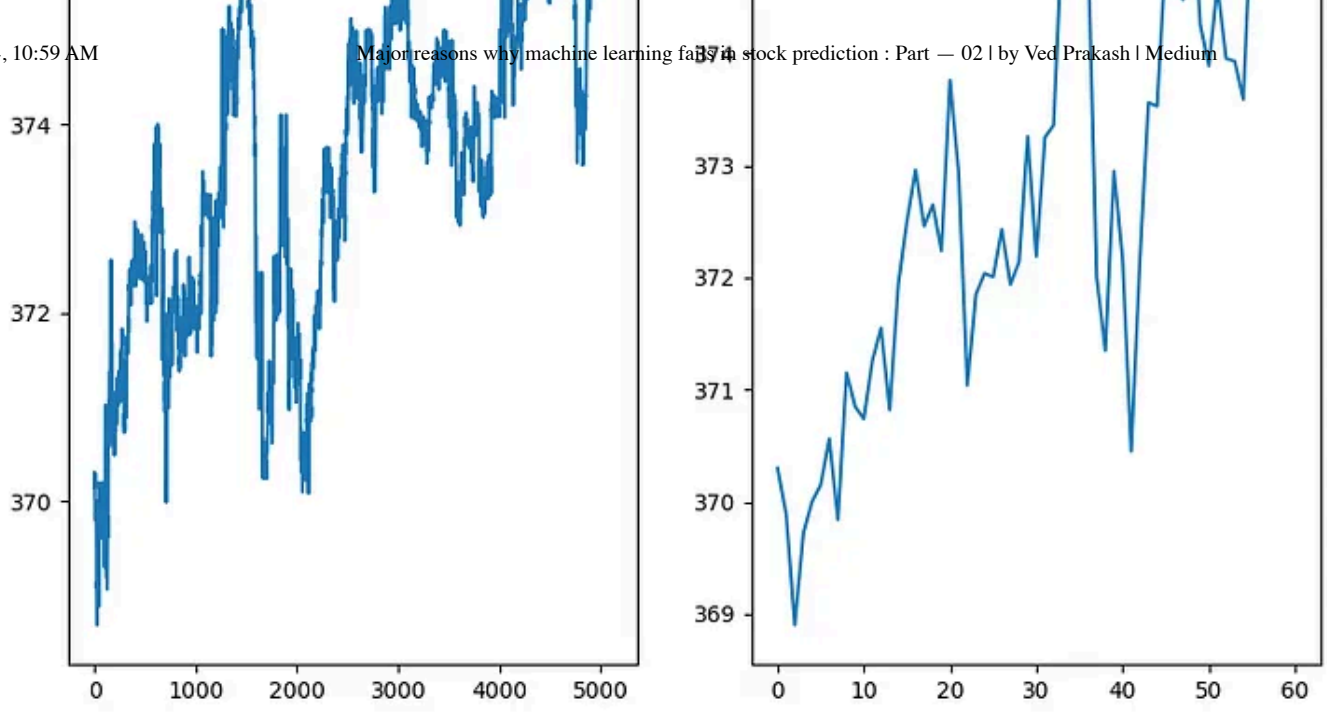
```
weighted_sum_T = 7
alpha = 0.9
weighted_sum_prob = 0.3
data["delta_p"] = data['open'].pct_change()
imbalance = 2
b_t = np.zeros(len(data["delta_p"]))
b_t[0]=1
indx = []
T=0

for i in range(1,len(data["delta_p"])):
    if data["delta_p"][i] == 0:
        b_t[i]=b_t[i-1]
    else:
        b_t[i] = np.abs(data["delta_p"][i])/data["delta_p"][i]
    T+=1
    theta = max(sum(x>0 for x in b_t), -sum(x<0 for x in b_t))
    if (theta >=imbalance):
        indx.append(i)
        weighted_sum_T = alpha*T + (1-alpha)*weighted_sum_T
        weighted_sum_prob = alpha*sum(x>0 for x in b_t)/(T) + (1 - alpha) * wei
        imbalance = weighted_sum_T * max(1-weighted_sum_prob, weighted_sum_prob

        T = 0
        b_t = np.zeros(len(data["delta_p"]))

final_data=pd.DataFrame()
open =[]
close=[]
volume=[]
for i in range(len(indx)-1):
  ind = (indx[i],indx[i+1])
  temp_data = data[ind[0]:ind[1]]
  open.append(temp_data['open'].iloc[0])
  close.append(temp_data['close'].iloc[-1])
  volume.append(temp_data['volume'].sum())

final_data['open']=open
final_data['close']=close
final_data['volume']=volume
```



tick is sampled when there is a evidence of presence of large trader in market

In the book, there are different types of run bars that are discussed, such as volume/dollar run bars. These are similar to tick run bars, but the difference is that we sample the bar when the volumes or dollars traded by one side exceed our expectation for a bar.

3. Reason 03 : working in silos

Portfolio managers used to work in silos because if they will work together then they may end up influencing each other and this lead to less diversify portfolio. If 50 portfolio managers will work in a silo then they will come up with 50 different theories without influencing other theories for investment and this will lead to achieving a more diversify portfolio. But this type of approach usually backfires in case of quant portfolio managers because if we have 50 quant portfolio managers frantically searching for investment opportunities separately then they may end up eventually settling for a false positive that will look great in overfit backtest or a model with high academic support but low Sharpe ratio. So identifying new strategies requires a large team working together. Also quant firm is not very open in sharing the strategy due to secrecy reasons and this may lead to not having an

predict the stock market. Don't forget to share your thoughts and give a thumbs up if you enjoyed the read.

Reference:

1. Advance in finance machine learning by Marcos López de Prado
2. https://www.youtube.com/watch?v=BRUlSm4gdQ4&t=586s

Finance | Machine Learning | Stock Market | AI | Quant

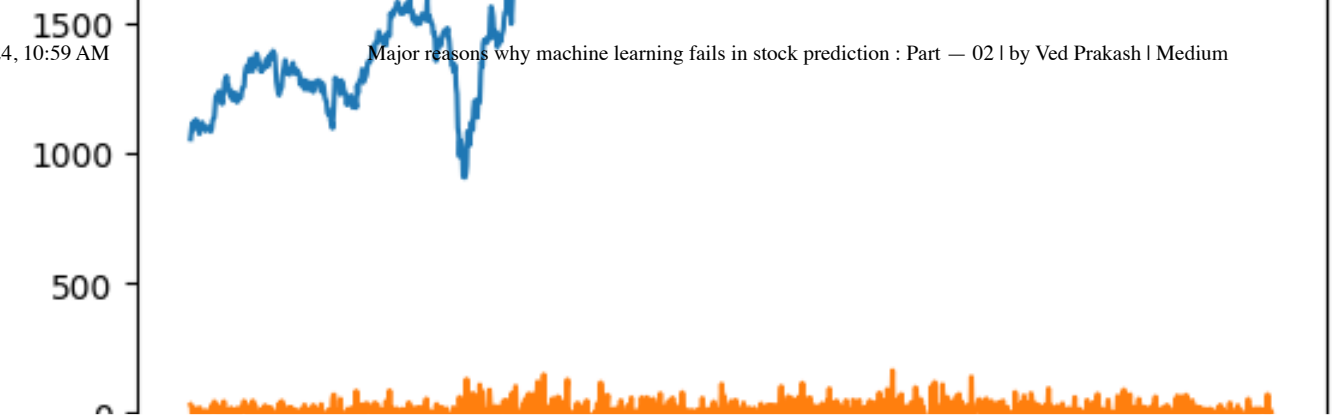
Follow

Written by Ved Prakash

508 Followers

Data Scientist at Michelin

More from Ved Prakash



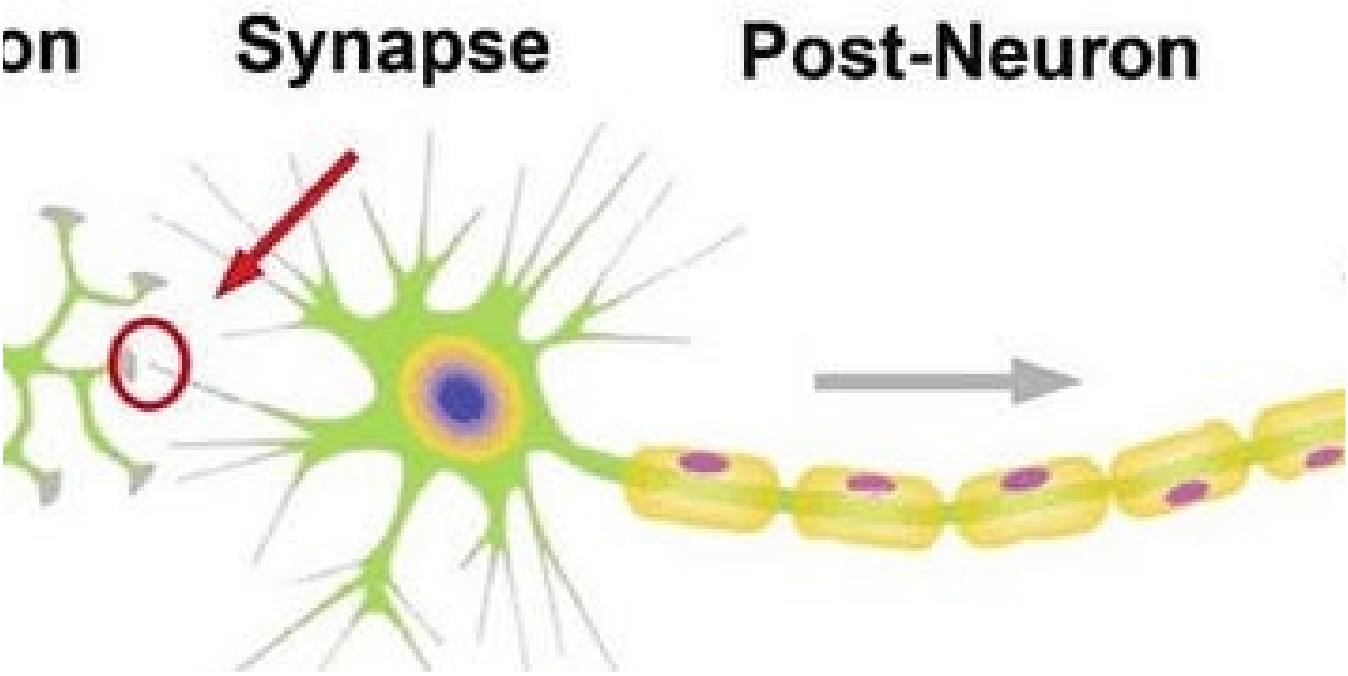
Ved Prakash

major reasons why machine learning fails in stock prediction: part -01

in this series of blogs i am going to discuss about the reasons why machine learning fails in predicting the stock prices or in general why...

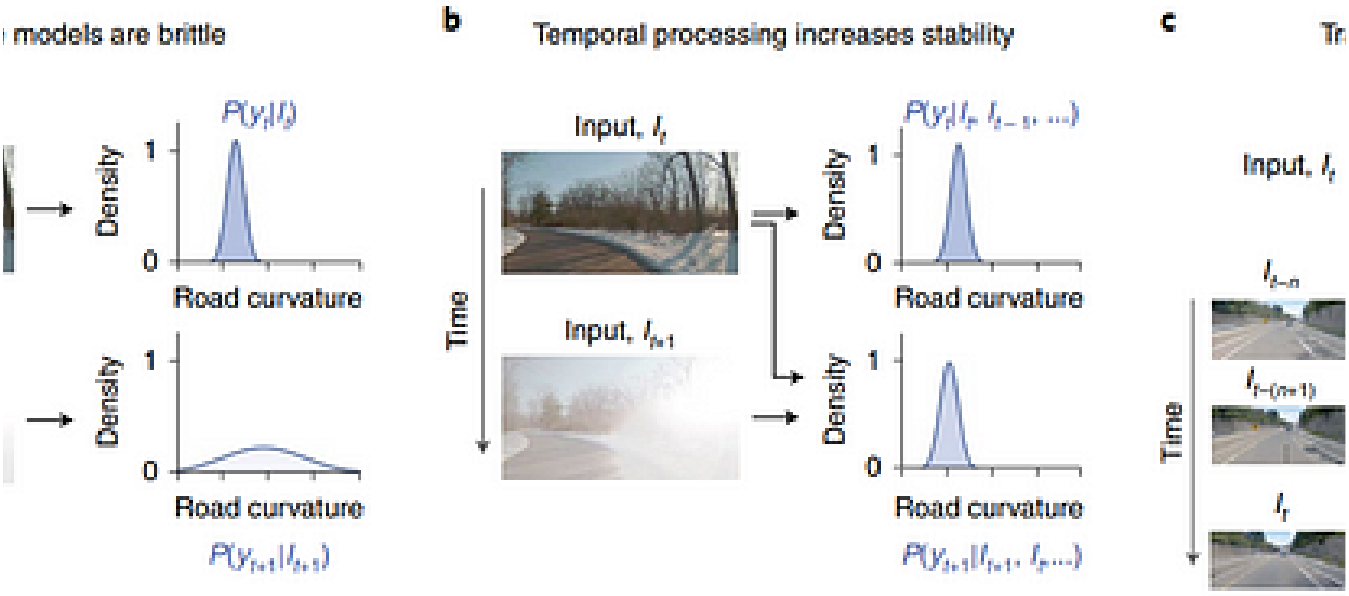
6 min read · Jan 10, 2024

1.1K 25



Ved Prakash

Liquid Neural Network : A adaptive way to train ML model



Ved Prakash

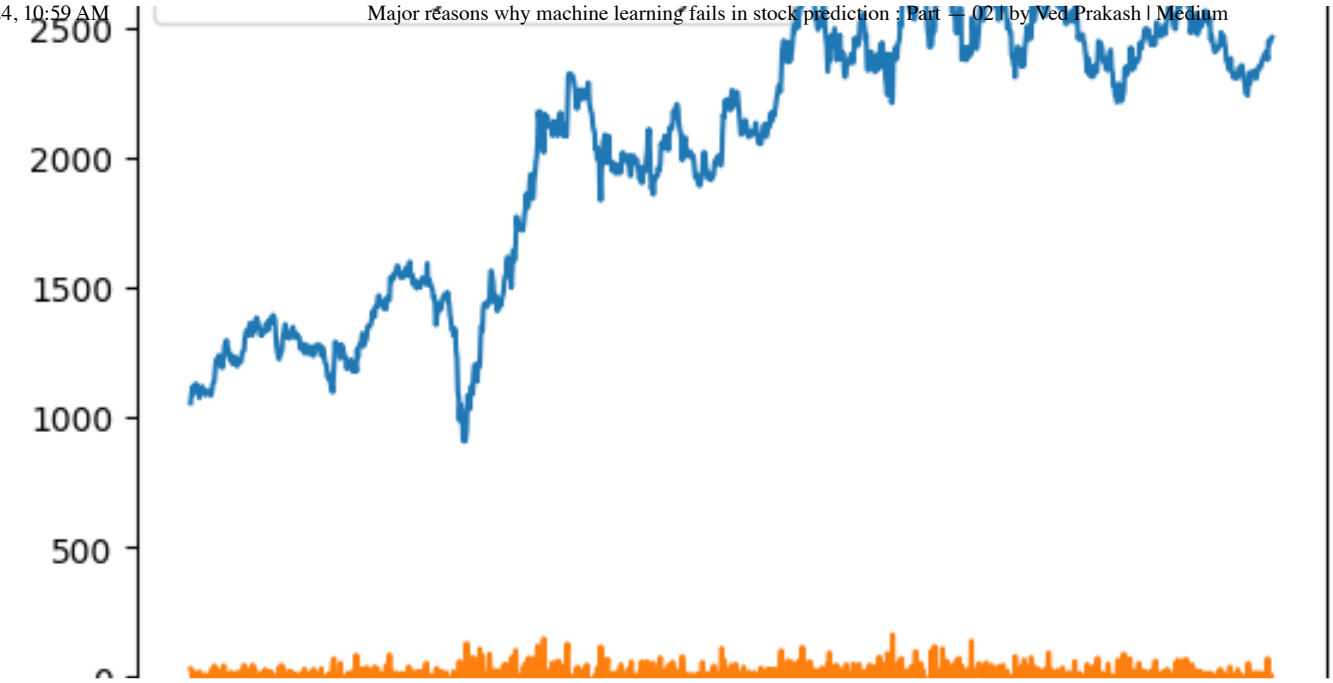
Neural Circuit Policy : training a autonomous vehicles using models inspired by nervous system.

This is my 2nd article in series of articles where I will review the different research paper from the AI, machine learning, deep learning...

9 min read · Sep 16, 2023

58 2

See all from Ved Prakash



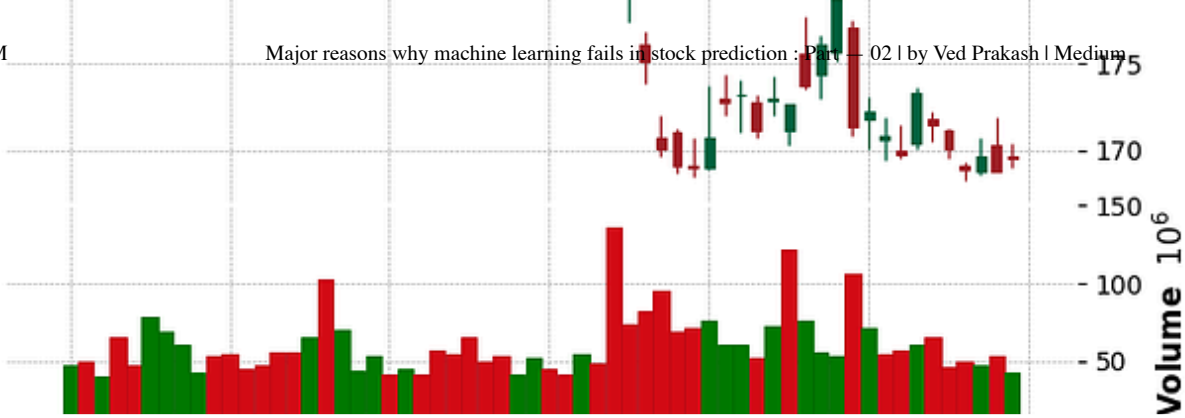
Ved Prakash

major reasons why machine learning fails in stock prediction: part -01

in this series of blogs i am going to discuss about the reasons why machine learning fails in predicting the stock prices or in general why...

6 min read · Jan 10, 2024

1.1K 25



Dr. Ernesto Lee

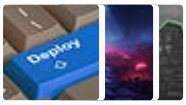
Advanced Stock Pattern Prediction using LSTM with the Attention Mechanism in TensorFlow: A step by...

Introduction

15 min read · Apr 8, 2024

894 18

Lists



Predictive Modeling w/ Python
20 stories · 1316 saves

The New Chatbots: ChatGPT, Bard, and Beyond
12 stories · 412 saves

Practical Guides to Machine Learning
10 stories · 1580 saves

Natural Language Processing
1535 stories · 1071 saves

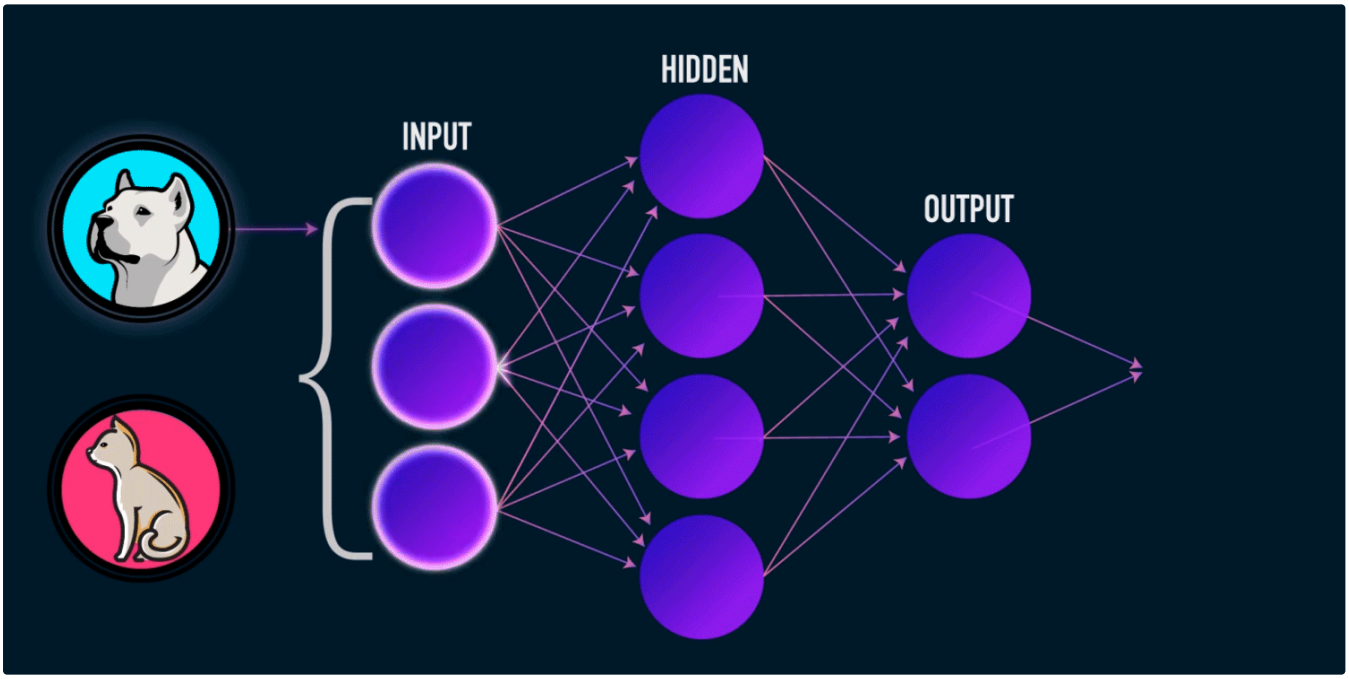
Quant Prep

My tips on how to be become a Quant (if I were to start again)

In this article, I will be sharing tips and the list of resources I'd use if I had to start over with becoming a Quant again.

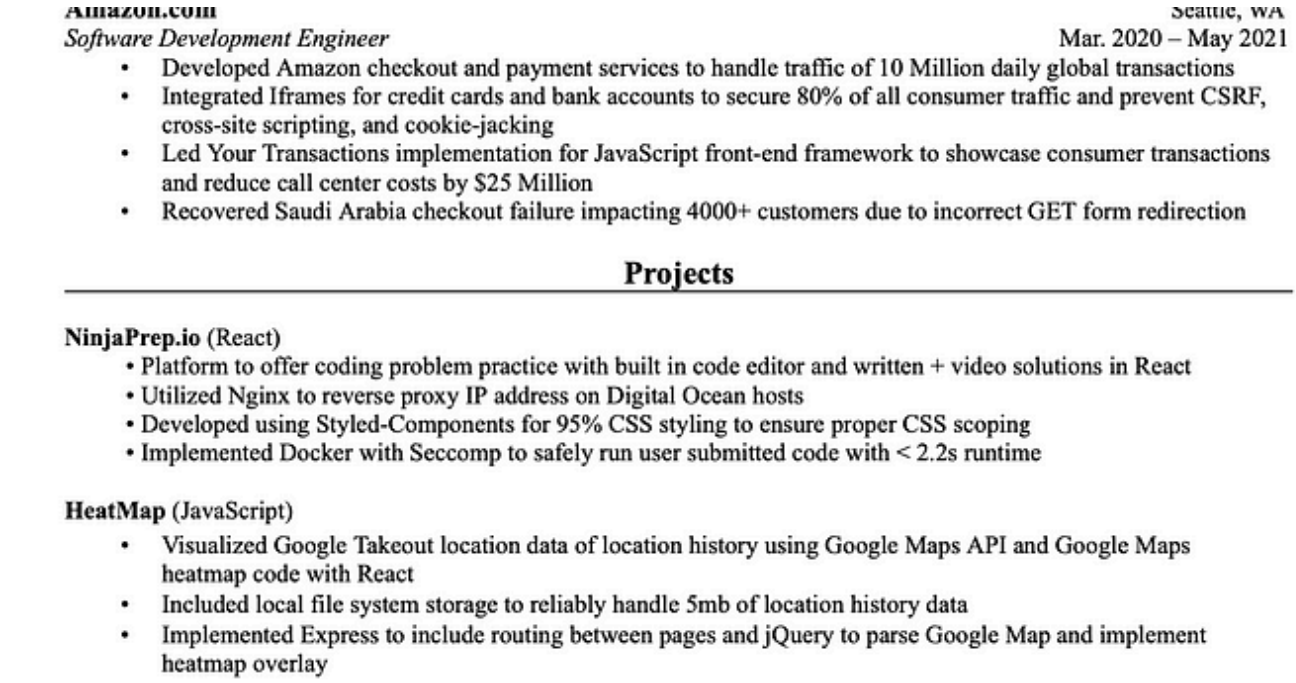
6 min read · Jan 28, 2024

503 7



Quant Club, IIT Kharagpur

STOCK BUY-SELL-HOLD PREDICTION USING CNN



Amazon.com Seattle, WA
Software Development Engineer Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by $25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects

NinjaPrep.io (React)
- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)
- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay

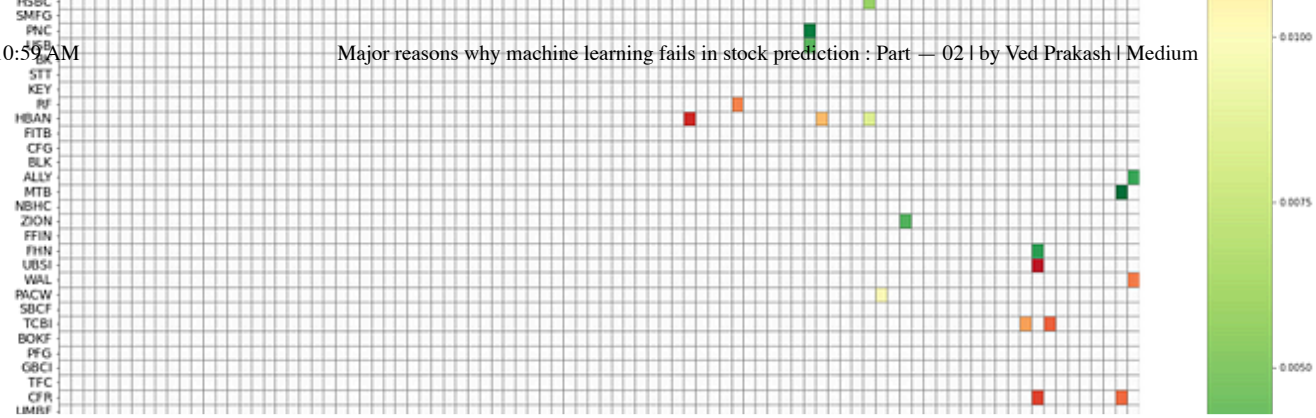
Alexander Nguyen in Level Up Coding

The resume that got a software engineer a $300,000 job at Google.

1-page. Well-formatted.

✦ · 4 min read · Jun 1, 2024

8.2K 94



Ayrat Murtazin in DataDrivenInvestor

Citadel's Strategy Anyone Can Use—Pairs-Trading

Pairs trading is a sophisticated strategy often employed by quantitative traders to trade a portfolio, rather than focusing solely on...

· 12 min read · Mar 6, 2024

402 10

See more recommendations