Stock Market - Buy-Sell Recognition

Group 2 | Carol Ho | Serena Chan | Thomas Wong

Agenda

01 **Background**
Objective | User Interface Demo | Workflow Snapshot

02 **Data Preparation**
Data Collection | Graph Plotting | Labeling

03 **Training & Testing**
Yolov5 Training | Sensitivity Review | Back-testing

04 **Recommendation & Deployment**
Program recommendation | Streamlit

05 **Conclusion**
Issue | Further Improvement

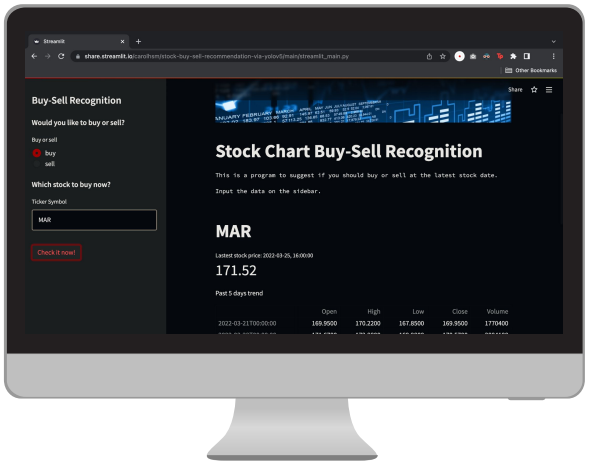
Project Background

Objective

Difficult for amateur traders to identify candlestick patterns from stock charts.
Utilize image recognition to identify patterns.

Our Aim

Build a program to find optimal buying and selling opportunities on-the-day for stocks of user preference.

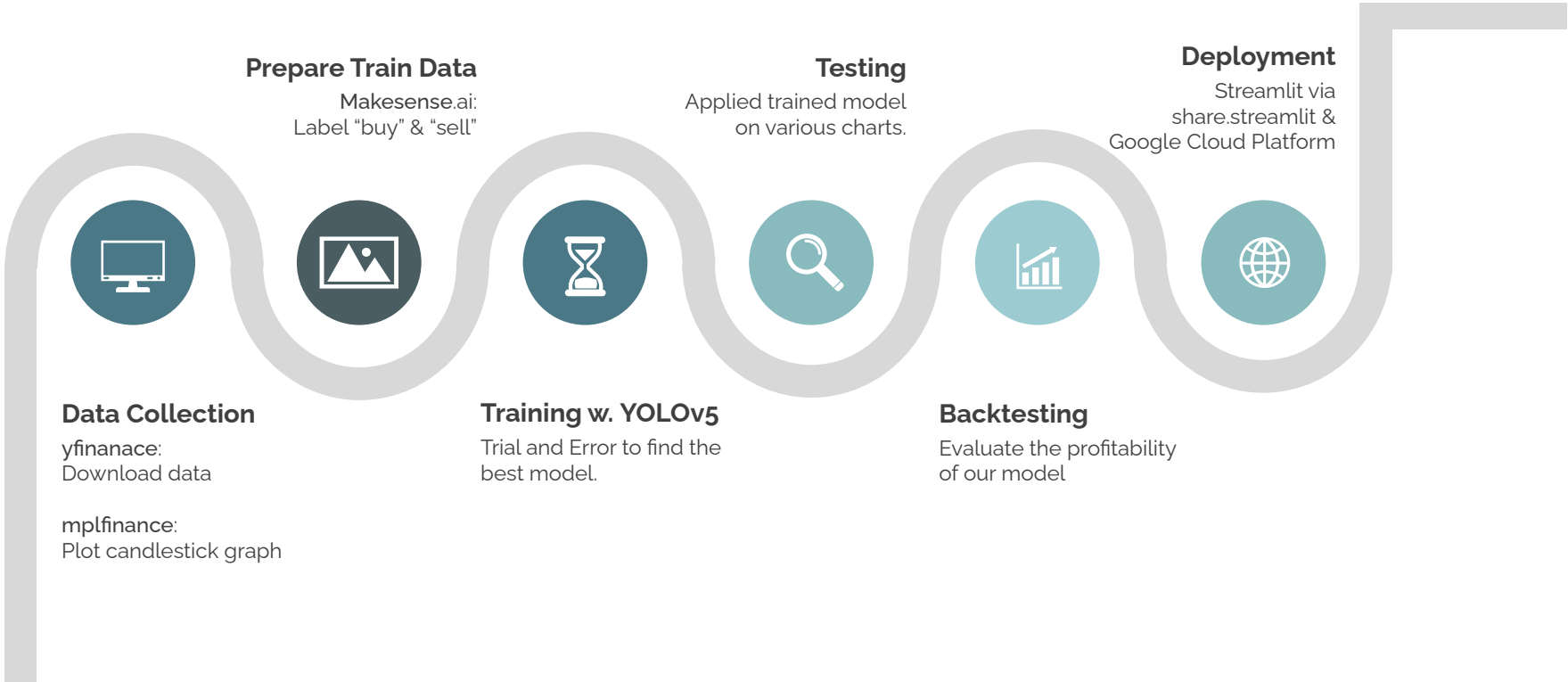


https://share.streamlit.io/carolhsm/stock-buy-sell-recommendation-via-yolov5/main/streamlit_main.py

Programs / Libraries

Programs: makesense.ai, streamlit cloud, google cloud
Libraries: yfinance, mplfinance, streamlit
Pre-trained Model: Yolov5

Workflow



Data Collection

yfinanace:
Download data

mplfinance:
Plot candlestick graph

Prepare Train Data

Makesense.ai:
Label “buy” & “sell”

Training w. YOLOv5

Trial and Error to find the best model.

Testing

Applied trained model on various charts.

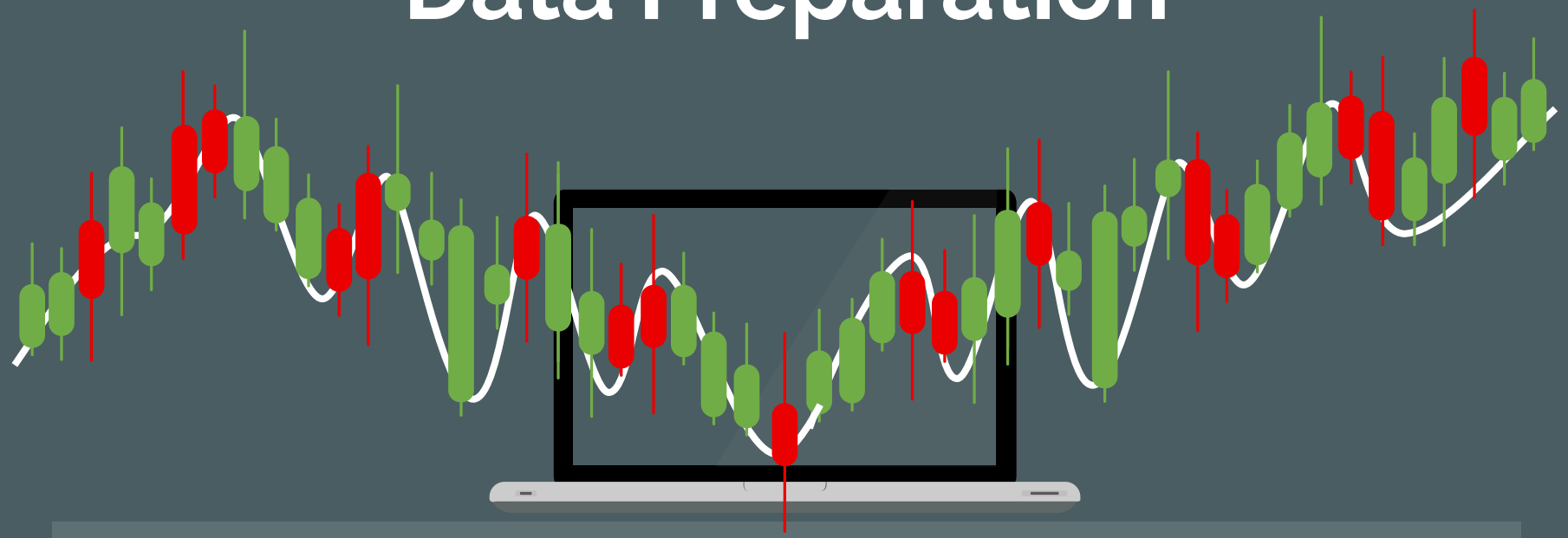
Backtesting

Evaluate the profitability of our model

Deployment

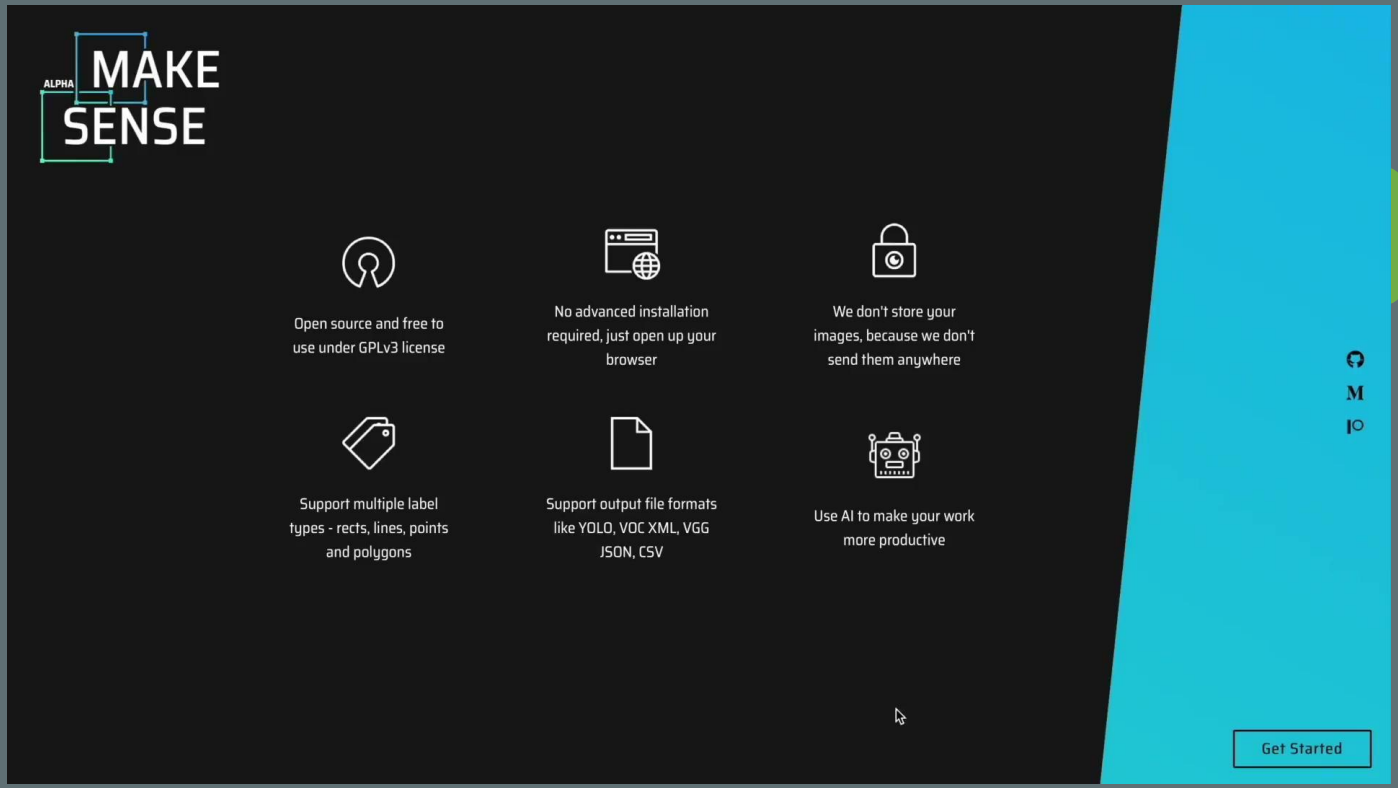
Streamlit via share.streamlit & Google Cloud Platform

Data Preparation



- Downloaded **Nasdaq top 50** stocks **10yrs** data with the use of **yfinance library**
- Plotted **100-days candlestick charts** of grey background with **mplfinance module**
- Labeled the **buy and sell area** with the help of **makesense.ai**

Image Labeling Demo



Exported Annotations

Training and Testing

Trained **665 images** with the use of **Yolov5**

- 🚨 GPU and time is the issue - we tried various ways including Google Colab/Kaggle/Desktop
- 💡 Noticed the importance of training data consistency and validation data choice (mAP 0.3 vs. 0.9)

Applied trained weight to detect other **new graphs** to review its **sensitivity**

- 🚨 Discovered the program rarely bound the edge - go against our app function
- 💡 Added 5 fake candlesticks to the right of same price of last data point (previous slide)

Tried to apply the trained weight to graphs of **different time frame** (e.g. 50/150/200 days)

- 🚨 Sensitivity is less performing on other time frames
- 💡 Sticked with 100-day chart in the deployment

Backtesting

BUY

SELL

Backtesting

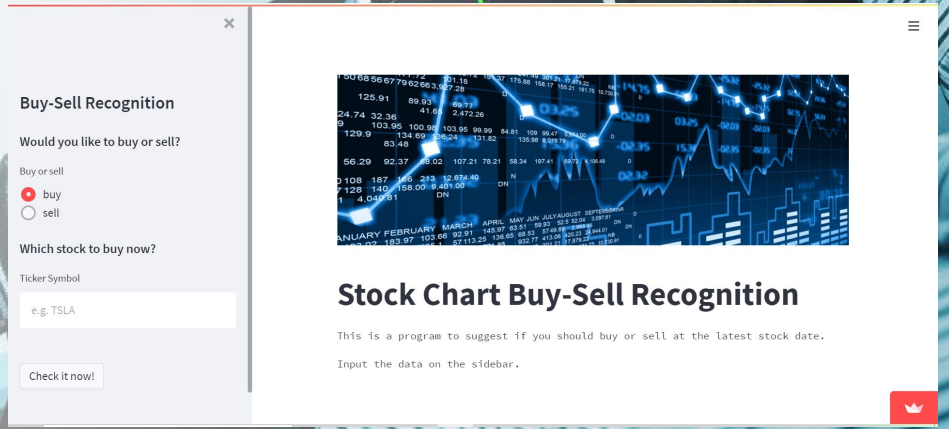
- A total of 6 stocks outside of the training samples were used for backtesting
- Starting capital: 10,000 USD
- Look for the 1st buy day; sell all the stocks after 1 year
- Buy as much as we can when suggested to buy; sell all stocks when suggested to sell

Test result

Stock	Profit/Loss percentage
PYPL	+21.05%
LRCX	+23.88%
MAR	+89%
COST	+17.11%
ETSY	-8.32%
CMCSA	+30.17%

Streamlit Deployment

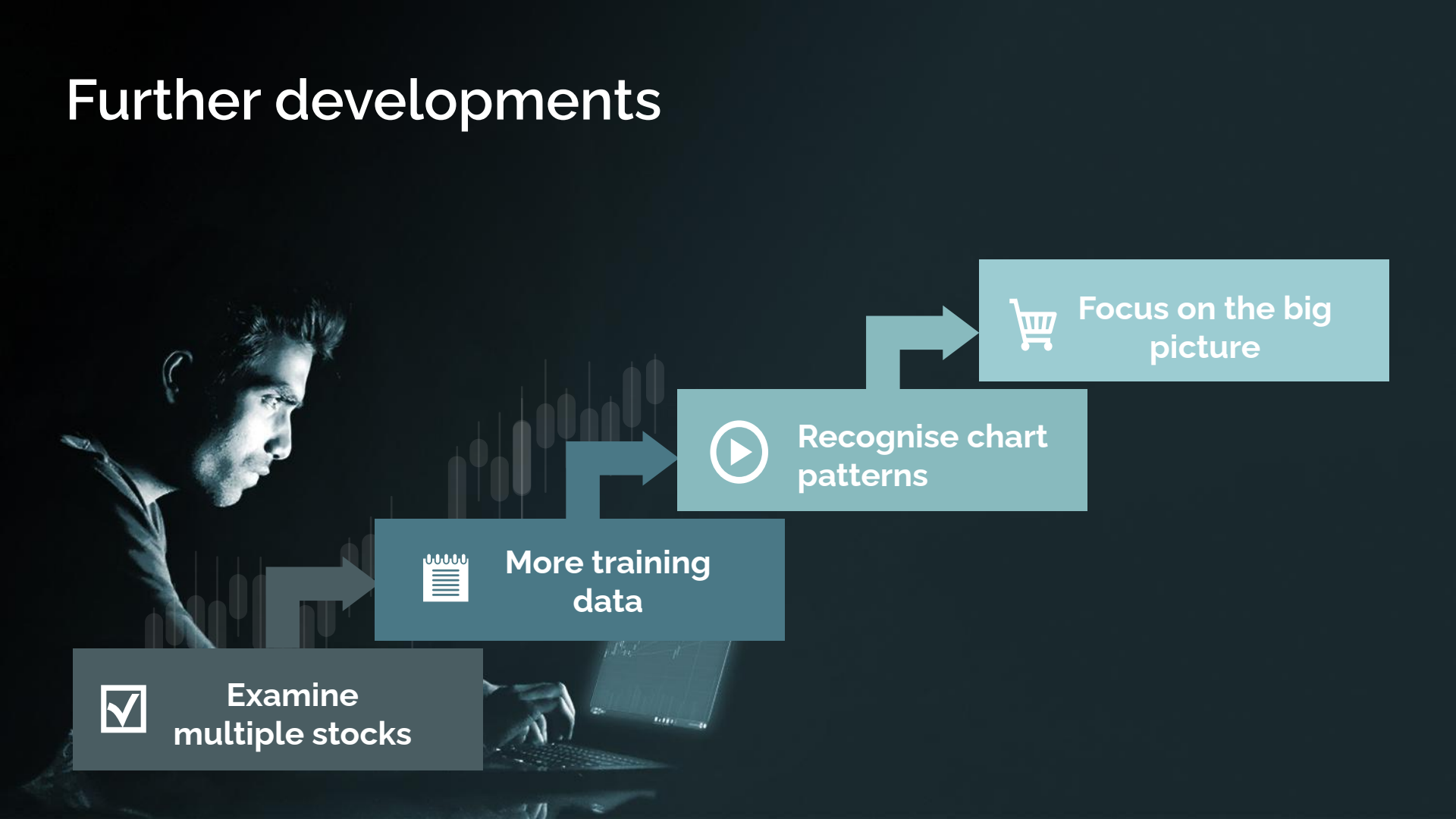
- Users can choose their action: buy or sell
- If uses choose to sell: stop loss check
- No need to look for the ‘box’



Streamlit Deployment

- The program is uploaded to Streamlit Cloud and Google Cloud
- Easy access
- Can ignore the modules installation process

Further developments




Thank You
Any Qs?