TeamTBA Presentation

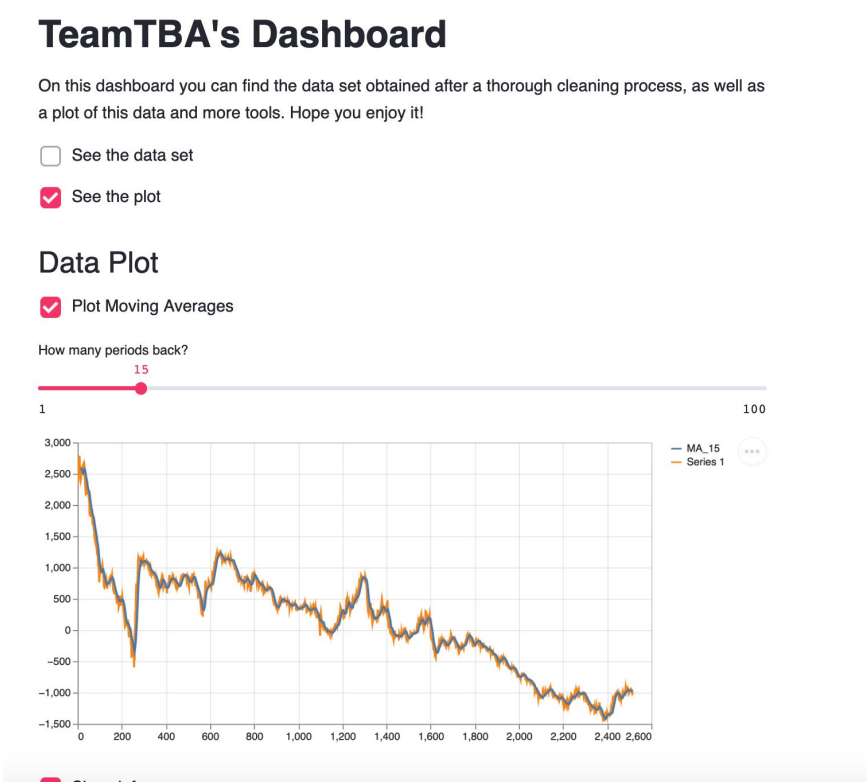
Chris Chia, Farid Abou Jaoude, Sanjit Neelam

Data Cleaning Challenge

- Clip values exceeding the 0.5% and 99.5% quantiles respectively
- Use `interpolate` pandas

Data Visualisation Challenge

- Link: https://teamtba.herokuapp.com/
- Using Streamlit



Machine Learning Strategy Challenge

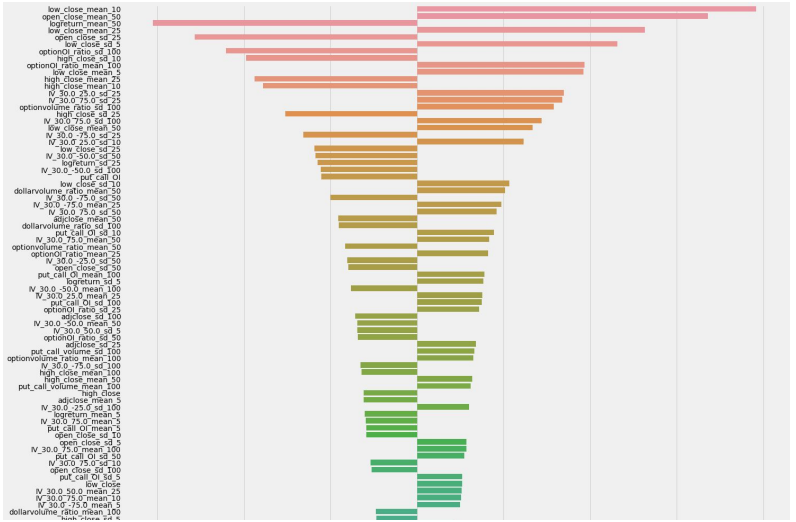
Link:
https://colab.research.google.com/drive/1Fgs_EqtefHmesxacSVlE_ie3UE8vlTYR?usp=sharing

- **Features**: Trade_time (transformed to categorical), Market (transformed to categorical), Trade_duration (hours), Previous Day Open Interest (for a relevant liquid futures contract), Volume (for a relevant liquid futures contract)
- **Approach**: Train four models, one for each asset type. Then generate predictions using appropriate model
- **Model**: lightgbm
- **Signal:** Predict portfolio weight, then map to {1, 0, -1} as signals based on a threshold

Prediction Challenge

Link:
https://colab.research.google.com/drive/1wLFhY1rpSsYA3MTuxMKOSw6Qc61IG52O?usp=sharing

- Train Neural Network to directly optimise for Sharpe (minimise Batch Negative Sharpe)
- When only 1 hidden unit and hidden layer, effectively just optimisation / regression and weights are interpretable
- Could be extended to include hidden layers with non-linearities -> Representation Learning / Autoencoders



ESG challenge

Link:https://colab.research.google.com/drive/1ljxBlfiNtftPh4vlZCPpc5r5eGFJztK4?usp=sharing

- First filter all assets to select those with ESG score in E, S, G > 45. The question is under what circumstances. We simply take this to be the mean ESG score
- **Model**: Train a neural network to predict the appropriate *weights,* with softmax activation. The alternative would be akin to regression - predict returns using a single layer with 1 output unit, (in this case the NN weights would be the asset weights) but it is difficult to impose constraints on the model weights
- Use Mean-Squared Error against S&P500 as a loss function

***Quite hard to actually implement so just submitted 1 / N strategy**

Long-Short Sector Strategy Challenge

Link: https://colab.research.google.com/drive/1E0-vi71KTtICUFkHEvwGiWO8YobBOPGZ?usp=sharing

- **Task**: Come up with a long-short strategy from returns data- but multi asset
- Could be viewed as a form of constrained portfolio optimisation
- **Model**: Train a Neural Network (with just 1 layer, so in fact multi-output logistic regression) to predict weights
- Use softmax activation to ensure weights sum to 0
- Minimise negative of cumulative returns - 0.5 * CVaR as a loss function
- Not the actual loss function but CVaR could be a proxy for max drawdown

$$\frac{e^{-wx}}{1+e^{-wx}} - \frac{1}{N_{assets}}$$

$$\mathbb{E}[R_t] - \frac{1}{2}\mathrm{CVaR}_\alpha[R_t]$$

Low Latency Challenge

- **Link:** https://github.com/Faridabj/TeamTBA
- **Iteration 1:** Logistic Regression using scikit-learn.
- **Iteration 2:**Since the public leaderboard suggests it has greater than 50% accuracy...Extract weights and cache them as a numpy array within the file itself. We directly calculate logistic function, and then take prediction above 0.5
- **Iteration 3:** For a threshold of 0.5, equivalent to a Linear Probability Model. So just need to calculate (WX+b) > 0

$$\sigma(x) = \frac{1}{1 + e^{-(\mathbf{XW}+\mathbf{b})}} \qquad \hat{y} = \sigma(x) > 0.5$$

$$(\mathbf{XW} + \mathbf{b}) > 0$$