

## Roy Ng

Compilation of School and Side Projects

Resume on my web page @ <https://banila48.github.io/Bani/>

### ₿ [Crypto Price Alert Bot](#)

Crypto Bot that will send real time alerts if a price of a coin is above or below target level. You can also retrieve current price charts of coins. Bot is hosted on Amazon EC2 Free Tier.

- **Context + Inspiration:** I got into the crypto trading scene in 2020 and due to the volatile price movement of alt coins, I have to monitor my TradingView account often. TradingView had 2 big pain points:
  - Real time charts were unoptimized for mobile viewing
  - Can only set 1 alert for free account users
- This bot **completely resolve** both issues as the interface is on Telegram. Users are able to set unlimited alerts with a small catch, the bot will respond slower
- Check it out live [here](#)
- *Challenges: Due to rate limitation from the API I used, the bot will have an occasional downtime of 1-2 days every month*
- **Technologies used:** Python

### [McDeliveroo](#)

This project aims to simulate the interactions between online food order in Singapore and the capacity of food delivery service. The simulation demonstrates a food order's journey from receiving order, processing it and finally delivering it.

- **Context + Inspiration:** Many times customers were frustrated over the long waiting time of their food orders. In order to better depict real life scenarios, we created a simulation that allows a user to modify different parameters to gain insights on the entire process  
  
From a company standpoint, this allows them to **discover whether one's infrastructure and logistics** would be able to meet his expected average delivery time and average number of deliveries done hourly for each of one's riders
- *Challenges: Debugging javascript was extremely problematic. I realised the final outputs of the simulation will vary depending the weightage assigned to the parameters. Finally, the path that was used to compute travel distance by rider is not realistic as it could be significantly shorter or longer due to weather and traffic conditions*
- **Technologies used:** Vanilla Javascript

### [Weather Sentiments Predictor](#)

I was involved in developing a RNN model to predict the sentiments of weather tweets.

- **Context + Inspiration:** Sentiment analysis has become more popular in recent years especially in the political scene where comments are scraped online to give a preliminary forecast of the polling results. This project aim to expose us to NLP by forecasting the weather using twitter tweets
- The Long Short-term Memory (LSTM) model offered only a slightly better performance over Support Vector Machine (SVM) and a fully optimized Random Forest (RF) model. This could be improved if we used a Random Search approach for the regularization (dropout) Accuracy:
  - LSTM - 87.4%
  - SVM - 83.7%
  - Fully optimized RF - 85.6%

- *Challenges: The algorithm has to be built in R which proved to be difficult as I was unfamiliar with developing a deep learning model in that language*
- **Technologies used:** R

## [Netflix Recommender](#)

Our team was tasked to build a Netflix Recommendation System using restricted Boltzmann machine (RBM). As this was a competition we had to pit our prediction scores with other teams as well as a Linear Regression model.

- **Context + Inspiration:** Recommendation systems are becoming more popular from Youtube to Spotify and Online Shopping. We are provided with a dataset that contains information about user's opinions of movies in the form of a rating. The task lies in finding what users might rate for movies they have not yet watched
- The plain RBM model severely underperformed against the Linear Regression model. We subsequently improved the accuracy by adding a few extensions. The below are the list of extensions I contributed to:
  - Random Searching
  - Momentum
  - Biases
  - L1 Regularisation
  - Adaptive Learning Rate
  - Mini-Batches
- *Challenges: It was a bottomless pit, a lot of the resources I found did not prove helpful as it did not translate to an improvement in prediction. This is mainly due to poor implementation. If there were more time, I will try using a Bayesian optimisation approach as well as consider rounding off values to the nearest integer to simulate a user's pick*
- **Technologies used:** Vanilla Python