

FFXIV MARKETBOARD ASSISTANT

Technologies used for the application:

- Pandas
- Plotly
- Tensorflow
- Python
- HTML
- Dash

I will be using two API's:

- <https://xivapi.com/>
- <https://universalis.app/api/v2/>

Pandas is going to be holding the data and cleaning up the different information coming in and feeding it to Tensorflow and Plotly for the AI training and the graphing. I have built my own Neural Network with Keras and will be handling training the AI outside the actual program to allow for better performance when doing predictions and graphing. The language I am using is Python but also some HTML as it needs formatting on the front end. I am using Dash as my technology which melds well with Plotly and Python.

As I was going through the documentation for the universalis API I found that they stored historical information for market board sales at <https://docs.universalis.app/#schema-universalis.application.views.v1.extra.stats.recentlyupdateditemsview> (*Universalis Documentation*, n.d.) which allowed me to circumvent the need for an SQL database to be stored locally. This makes pulling the price data quite easy and allows for better results as some items take a while to sell because of the price they are or the demand that they have.

I found a small issue where the name of the item was not shown in the calls to the Universalis API so I had to work another way of pulling the data. I found that the XIVAPI linked item name with item number by doing a few queries provided through <https://v2.xivapi.com/api/docs#tag/sheets/get/sheet/{sheet}> (*Boilmaster Documentation*, n.d.) this allowed access to not only name but all the features if I wanted to build the page out with more options.

Working with the AI has been an adventure as it has caused me problems with performance and with predictions. I need to do fine tuning and try and get the training and development of the AI done outside the running of the program to keep it updated. I have found that I might need to do some tweaking to the Keras network, I need to add at least one more layer to add complexity so I can improve on the predictions that I get during the item queries.