**README:**

**Welcome to my Git Hub page!**

I would like to showcase 2 Python projects which I initiated, in order to learn more about the language during my free time, outside of work.

**Data Reporting Automation Project**

The first project showcases a completely automated data reporting process. It starts with extracting the excel and/or CSV file from a centralised server location, perform data cleaning, manipulation, analysis and grouping, and transforming it into a readable dataframe.

Next, I export this dataframe out to an Excel spreadsheet, and perform the necessary edits to make the spreadsheet look more professional. This will include aesthetic changes such as font, borders, width, colours etc.

I then set up a function that will be able to send out this spreadsheet attachment with an appropriate email subject and body text, for management to view it.

Lastly, I set up an automatic scheduler that will perform all the above-mentioned functions by the last working day of every calendar month.

As long as management does not change their business requirements or management information request, this script can run indefinitely and will continue churning out the monthly MI reports to management without having me to do anything.

**Logistic Regression Machine Learning Project (Donald Trump’s Tweets about Tariffs)**

The second project showcases a basic machine learning methodology which I learnt during my course in the National University of Singapore, Python for Data Science.

For this project, I came up with an original idea to find out the relationship between Donald Trump tweeting about anything to do with ‘Tariffs’, be it slapping more tariffs on China/Mexico or withdrawing tariffs due to successful negotiations.

I use the Twitter API, Tweepy, to pull all relevant tweets from Donald Trump over the last year. I have also manually looked through the files and added the following values –

1. If a tweet suggests a tariff is incoming, we give it a value of +10 under ‘tariff\_impact’
2. If a tweet suggests that Trump will withdraw tariffs, we give it a value of -10 under ‘tariff\_impact’
3. If a tweet is ambiguous or does not mention any confirmation of tariffs, a value of 0.

I admit that this part could have been done better if I had incorporated Natural Language Processing to analyse Trump’s tweets, and let the script AI determine if Trump was really threatening to impose more tariffs or release them, based on his words. This will be a good idea for me to set up my third side project!

Moving on, I pull data from the Straits Times Index (STI) within the same timeframe as Trump’s tweets. I then prepare both the data tables, merge it, and prepare it again in order to get ready for the Logistic Regression portion.

Again, I feel that there could be better methods instead of using Logistic Regression for time series and stock prediction data (such as K Neural Networks, or Random Forests). I am aiming to be a better data analyst/scientist, and will continue to explore advanced machine learning techniques to make my projects more accurate and viable for real-world reports.

You may find my full .py scipts in my GitHub repository. I still have much more to learn, but I hope you enjoy viewing how I learn Python by doing these fun projects!

**Titanic Kaggle Machine Learning – Decision Tree for Survivor Rates**

I was very intrigued by a Kaggle exercise. Kaggle has provided datasets on pseudo travelers who were onboard a hypothetical titanic, split by their human demographics, cabin room and location during the iceberg incident.

The goal is to make use of a “Decision Tree” Machine Learning method to accurately predict the survival rate of all passengers onboard the Titanic based on their location at the time of the crash.

By taking advantages of the lesson I’ve learned about ML and Decision Trees (being one of the easier modules of ML), I managed to create a DecisionTree() class that takes into account all the necessary definitions and functions to calculate the survivor rate and put it into a neat dataframe.

The Decision Tree involves methods such as ‘pre-pruning’, to eliminate any scenarios of overfitting and noisy data corrupting the tree results. This is a simple ML model that I learnt online, and I would like to showcase my basic knowledge of machine learning through this example.