Central European University

MSc in Finance 2024

Turkan Aliyeva

Capstone Project Summary

Python-powered Broker Report Analysis: Automated Insights and Report Generation

Contents

Introduction

Before going into the details about how technical part of my project works, firstly, I would like to emphasize how the analysis of broker reports is very important for better understanding the market trends in the brokerage industry, and also to make decisions based on the analysis done. The mentioned reports are regularly published by exchanged-listed brokers, containing essential data such as financial performance, assets under management, trading activities, and etc. They hold a significance for the project sponsor, which specializes in finding brokers for the clients, helping them offer informed recommendations to the clients. So, the company can provide up-to-date recommendations, identifying the potential trends and shifts in brokerage market, evaluating and comparing the brokers through the analysis of their financial and also operational datas.

However, there are some challenges with the current method of analysis which is traditional mannual approach. Most importantly, one of them is that, the traditional method is extremely time-consuming which can lead them not to react to the changes in the market immediately, and also to provide out-dated information to the customers considering the time-consuming nature of all the processes like data collection, analysis and getting insights. Additionally, the another challenge is that, manual process is very prone to human error which can later lead to the inaccuracies in the analysis.

Objectives

Considering all above, I was aiming to develop a Python script for automatically analyzing exchange-listed brokers' reports and generating report with conclusions as a main objective of this project. This script created a system which would automatically extract and analyze the data from broker reports, getting insights and generating the report with required conclusions including some visualizations and tables, reducing the time and effort required for all this process. So, this automated system would enable real-time monitoring of market trends and providing timely insights and also enhance the accuracy and consistency of analysis by eliminating human errors.

Data

The original data for this project was stored in Excel file provided by the project sponsor. I also used the brokers quarterly reports taken from their websites (for 3 companies as sample). The Excel file contained various financial metrics, most importantly, revenue, net profit, the number of new accounts opened, total customers, trades executed, and total assets under management, each being stored in a different sheet with their names accordingly. All the sheets contained a table with the broker company names and the proper figures for the relevant metric from 2020 Q1 till 2024 Q1, even though for some metrics, there was some yearly data before 2020 as well. I changed all the tables to a structured format, so that I could easily start to work on that.

The process

Throughout the project, I used Jupyter Notebook as a primary development environment for Python because of its more versatility, convenience, flexibility, and efficiency for handling different type of tasks which were involved in the project.

My project had two parts: where the first part was to develop a Python script for data collection and storing, second part was developing a script for the analysis, getting insights, and finally, generating the report based on the insights.

I started to the project with the review of Excel data provided by the project sponsor, which included quarterly metrics such as total assets, revenue, net profit, total customers, number of new accounts opened, and the number of executed trades. Then, after analyzing the structures of the PDFs (which is important to identify patterns to find the required datas), I developed a Python script that followed the patterns in PDFs (brokers' financial and operational reports) using regular expressions with the function that I defined for both extracting the text from PDF and also searcing the patterns which were stored in a dictionary, where the keys were the company names (consistent with the column titles in Excel). So, that the user can simply enter the path (where the broker report to be scraped locates) into the code to extract the text, which will be used to collect the essential financial data later.

I selected three companies as a sample to develop a script after identifying the patterns for extracting key financial metrics such as total assets, total customers, revenue, and net profit. The structure of the codes allows the clients to add more company names and metrics in the future. The function that I defined includes 2 arguments as input, where the first one, takes the text extracted with the help of another function to extract all the text from PDFs, the second argument takes the dictionary of patterns in which I stored the patterns for each metric and for each company accordingly, so I could make sure that the function could give the result in a structured way to be able to store it properly. At the end, for sure, testing part is very important. Since I stored the results in a way that, I could compare them with the actual data which was manually collected and also reviewed carefully. I wrote a code which helped me to ensure that, there are no differences and the function works properly.

The extracted data was read using pandas function "pd.read_excel" into a dictionary of DataFrames and every sheet was stored in Excel in separate DataFrames and separate variables with the names of sheets accordingly. This step was very crucial for further cleaning, manipulation, and preparation for analysis. First I cleaned the data with the defined function for stripping any leading or trailing spaces from the titles of the DataFrames and also for replacing 'NA' values with pandas "pd.NA".

After data cleaning, now this is the step for data preparation. I created new dataframes for further analysis and creating plots, such as revenue per customer, average account size, or the dataframe where we can analyze the brokerage market between US vs EU during the recent year for which the yearly report was already available on their website.

Then, I generated visualizations using the matplotlib library, mostly focusing on the line plots, bar charts to provide insights into the broker companies yearly and quarterly performances, as well as to visualize the regional comparison of the metrics over a period of time. Those visuals helped me to identify the growth patterns, seasonal variations, and compare the brokers performances during the last quarter.

Subsequently, using docx library, I developed a code to generate a report storing in ".docx" file using dynamic variables to synthesize all these insights I got with the previous steps. This document contains a written analysis of the loaded data, with the mentioned metrics which are presented as heading and subsections, and paragraphs containing visuals and tables along with

explanations. Looking at this report can give us the information about the key findings during the analysis in brokerage industry and also provide stakeholders with a clear understanding of market dynamics.

At the end, I imported "dox2pdf" library to convert the ".docx" file to PDF format which offers more benefits such as more professional presentation as PDF files preserve the formatting, and layouts of the original ".docx" file and enhanced accessibility since PDF is globally accepted format which also can be accessed on almost any device without any additional softwares.

Conclusion

As a main outcome, this project has successfully created a system as an automated solution for streamlining the company's data analysis process and also generation of the report. By automating all the steps starting from the scraping the data from uploaded PDFs, and most importantly, generation of the report with conclusions after the analysis of financial data from broker reports, which helped to achieve significant time and effort savings.

So, the system solved the problems mentioned above providing accurate, and real-time insights into market trends and broker performance. These insights encapsulates a thorough trend analysis, incorporating insightful regional comparisons between US and EU brokers. It delves into yearly and quarterly changes, providing averages such as average account size and trades per customer. With simply entering PDF path into the code and a run button, the analysis report is ready, streamlining the entire process for the users.

Knowledges I learned

In the course of this project, I learned many important lessons and skills that which I will be able to use in my future projects and during my career life. To make the financial data analysis process automated, I had to do research about Python libraries like re, PyMuPDF for text extraction from PDF and scraping the text through regular expressions and pandas for manipulating/analyzing data. Even though they took some time to learn them, however it made the process much easier for me to carry out. For example, regular expressions helped me a lot to identify the financial data for the key metrics from PDF documents with unstructured format of which, the results were then stored in an excel file using openpyxl library for preparing them further for data analysis. Matplotlib library also granted me a lot of experience which helped me to identify the trends and growth patterns or broker performances in the industry. Additionally, I used the .docx library to create a document file which showed me the benefits of using dynamic variables and I got a great deal of expertise on that and also the "docx2pdf" library to convert the .docx file to PDF which provided more professional and accessible format for presenting the results.

Besides of the Python library research, there were a lot of challenges I experienced that helped me to improve my Python skills. One of the challenges was finding consistent patterns from the PDF reports with an unstructured format which made it more challenging. Another challenge was debugging errors which required a lot of time and problem-solving skills. At the beginning, I encountered some unexpected issues. So, I learned how to identify the bugs and fix them, enhancing my understanding of Python libraries that I mentioned above. Since as we know that, every learning process brings some challenging at the beginning which is an opportunity for growth, higlighting the areas where our skills may be lacking, so that we can

focus on those areas more to strengthen. This happened to me as well. With all these challenges that I mentioned, I improved my technical skills in Python as well as more problem-solving mindset which equipped me to be more successful throughout my professional life.