10.18686/neet.v2i2.3931

The Integration of a Python-Based Corporate Finance Course Teaching Tool

Pu Kan[1*], Jiajun Lu[2], Hongyu Zhou[3]

University of Sanya, Sanya 572022

Abstract: This paper introduces the development process of a financial analysis tool based on Python and discusses how to apply it to the teaching practice of corporate finance courses. The joint participation of students and teachers in the development of this practical project enhances the students' computer operation skills and also provides them with a complete financial analysis practical project. This paper focuses on analyzing the problems encountered by students during the development process and provides corresponding solutions.

Keywords: Python; Discounted Cash Flow Valuation Model; Saxo Fintech Business School USY

1. Introduction

Within the swift evolution of financial technology, the application of programming within the finance sector has gained paramount importance. Python, distinguished by its efficiency and accessibility, is extensively utilized across financial domains owing to its formidable capabilities in data manipulation and analysis. The advent of fin-tech has rendered proficiency in coding and data scrutiny indispensable for finance professionals. The deployment and utilization of tools based on Python, such as the Discounted Cash Flow (DCF) models, equip students and practitioners with the necessary skills to navigate this changing landscape, thereby bolstering their competitive edge in the fintech arena.

The significance of corporate finance management in the contemporary commercial milieu cannot be overstated. Its relevance extends to aspects such as organizational capital structure, investment strategy, and the pursuit of maximum shareholder value, all of which directly influence a corporation's competitive standing and potential for enduring success. In response to the accelerated pace of fin-tech advancements, numerous entities and financial bodies have adopted innovative technological solutions to refine their financial oversight and decision-making frameworks, thus securing a vantage position in robust market competition. For instance, JP-Morgan Chase leveraged machine learning algorithms for the automated evaluation of client financial data, facilitating tailored investment counsel and fiscal planning services. Through analytical examination of extensive datasets, the institution is equipped to forecast market movements with heightened accuracy, delivering opportune investment propositions to its clientele.

2. Literature Review

The collaborative initiative between IBM and Maersk, the TradeLens platform, harnessed blockchain technology to enhance supply chain transparency, diminish expenses, and expedite the distribution of goods[1]. This system's capability for real-time documentation and monitoring of trade data presents a secure and efficient alternative for supply chain finance. Similarly, Square, a leading financial services firm, employs artificial intelligence alongside machine learning techniques for the scrutiny of data, aiming to identify and mitigate fraudulent activities promptly[2]. This approach enables the system to detect irregularities swiftly, safeguarding both consumers and merchants. Another notable example is BlackRock's Aladdin platform, an all-encompassing investment management framework that integrated sophisticated data analysis and risk assessment models to deliver exhaustive asset management services. This digital platform empowers BlackRock to furnish clients with precise market insights and strategic investment advice[3].

These examples underscore the impact of technological innovation within the realm of corporate finance, enhancing the caliber of decision-making, streamlining financial administration procedures, and augmenting risk management efficacy. They offer invaluable lessons and benchmarks for the Python-based financial analysis tool's development and pedagogical application, underscoring the criticality of weaving technological instruction into the fabric of financial education. By embedding such cutting-edge practical examples into the curriculum, educators can ignite students' passion for learning and cultivate their innovative capabilities, laying a robust foundation for their prospective endeavors in the fin-tech domination.

3. Teaching Reform Practice - Development of a Python-based DCF Model

Traditional Discounted Cash Flow (DCF) models often rely on spreadsheet software for calculation and analysis. While this approach is intuitive in operation, it is prone to errors when dealing with complex financial models, especially when updates or modifications are required[4]. The development of DCF model tools using Python can automate this process, reducing human errors and enhancing the accuracy and efficiency of calculations. As an advanced programming language, Python boasts exceptional flexibility and extensibility. It has powerful data processing capabilities, particularly excelling in handling large volumes of data. When conducting analysis with the DCF model, which may be necessary to process vast amounts of financial data from various sources. Python can easily facilitate the collection, cleaning, integration, and analysis of data, providing accurate input for the DCF model.

Developing a Python-based DCF model tool allows for customization and expansion according to the specific needs of different businesses and industries, such as incorporating financial models that consider the impact of changes in tax policies and risk assessment modules. Combining Python programming with financial analysis models not only enhances the programming skills of students and financial professionals but also deepens their understanding and application of financial theories. Through the actual development and use of Python-based DCF model tools, students can gain a more intuitive understanding of the principles and application scenarios of the DCF model, improving their ability to solve real financial problems[5].

In response to practical needs, the progress of teaching practice is as follows:

1. Preparation: Initially, students determine the functionalities to be implemented through group discussions. Subsequently, after three weeks of study, some students are capable of basic user interface design and learning Excel data processing.

2. Design: The teacher guides the students to spend two months learning how to decompose data processing and academic APIs into independently functioning modules, with outstanding students capable of designing financial functions on their own.

3. Coding Implementation: In this phase, the teacher instructs the students on how to streamline the codes of various functions and integrate them into a software package, ultimately designing a visual interface to meet basic user experience requirements.

4. Testing and Feedback: Functionality of the software is verified through testing to ensure it meets the requirements, with students learning how to write test cases and correct identified issues.

4. Course Evaluation

Currently, the teacher leading the sophomore students in the Financial Technology major has completed a feature-rich financial analysis tool. This tool is capable of loading data from Excel files, calculating free cash flow for the next five years, applying the DCF (Discounted Cash Flow) model to calculate the total value of a company, and ultimately determining the price per share. The program encompasses the complete process of data loading, processing, analysis, and result presentation, making it an excellent practical project. The specific functionalities implemented are as follows:

Setting up the GUI: The app initializes a window and sets basic window properties. It defines three buttons for loading Excel files, calculating free cash flow for the next five years, and applying the DCF model for company valuation, respectively. A canvas is defined for displaying charts in subsequent steps.

Loading and Processing Excel Files: Using the load_file method and the file-dialog function, a dialog box pops up for users to select an Excel file. The application automatically calls the process_file method to handle the file. During the processing, regular expressions are used to filter out key financial indicator data and years for subsequent calculations, and the plot_data method is called to draw charts.

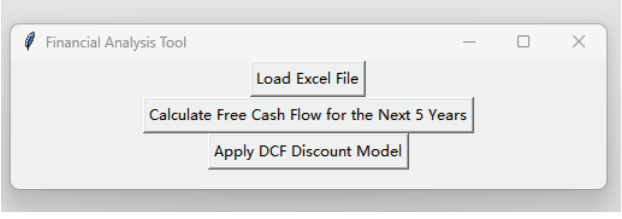

Figure 1: Financial Analysis Tool Options

Forecasting Free Cash Flow: A dialog box is created to allow users to input the expected annual growth rate. Then, based on the free cash flow of the last year and the growth rate entered by the user, the free cash flow for the next five years is calculated.

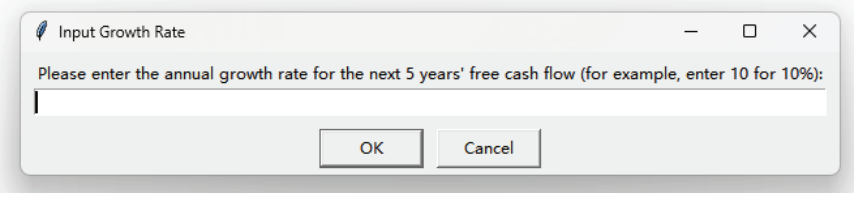

Figure 2: Dialog Box (Prompting the User to Enter the Expected Annual Growth Rate)

```
Forecast of free cash flow for the next 5 years and corresponding years::
2024: [181326.033, 190392.33465, 199911.9513825, 209907.54895162504, 220402.9263992063]
2025: [181326.033, 190392.33465, 199911.9513825, 209907.54895162504, 220402.9263992063]
2026: [181326.033, 190392.33465, 199911.9513825, 209907.54895162504, 220402.9263992063]
2027: [181326.033, 190392.33465, 199911.9513825, 209907.54895162504, 220402.9263992063]
2028: [181326.033, 190392.33465, 199911.9513825, 209907.54895162504, 220402.9263992063]
```

Figure 3: Terminal Output Results of Free Cash Flow for the Next 5 Years

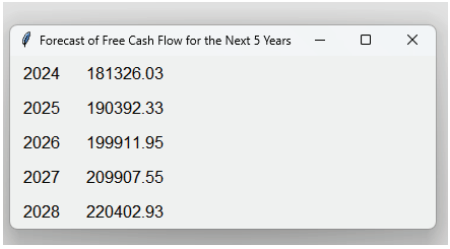

Figure 4: Output Results of Free Cash Flow for the Next 5 Years

Applying the DCF Discount Model to Calculate Company Value: Through a series of dialog boxes, data for variables such as the Weighted Average Cost of Capital (WACC) and perpetual growth rate input by users are collected. Subsequently, the total value of the company, equity value, and price per share under different scenarios are calculated. As shown in Table 6, the enterprise value analysis interface displays the calculated results for the theoretical value of the company, net debt, and predicted stock price.

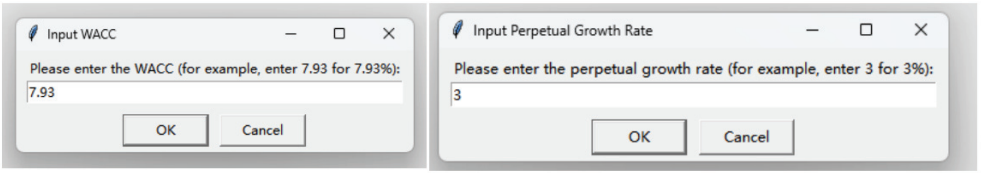

Figure 5: Dialog Box (Prompting the User to Enter Historical WACC and Perpetual Growth Rate)

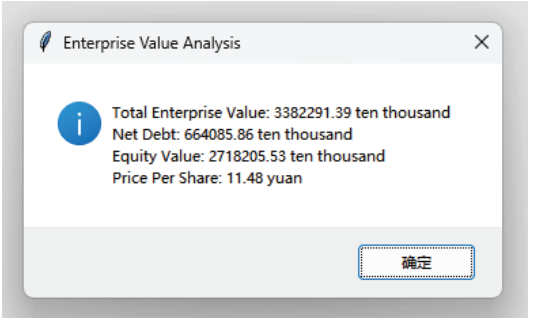

Figure 6: Output Results of Enterprise Value Analysis

Displaying Free Cash Flow Forecasts: This application uses plot method to draw charts for cash flow from operating activities, capital expenditures, and free cash flow. Additionally, which displays forecast-ed free cash flow for the next five years in a new window.

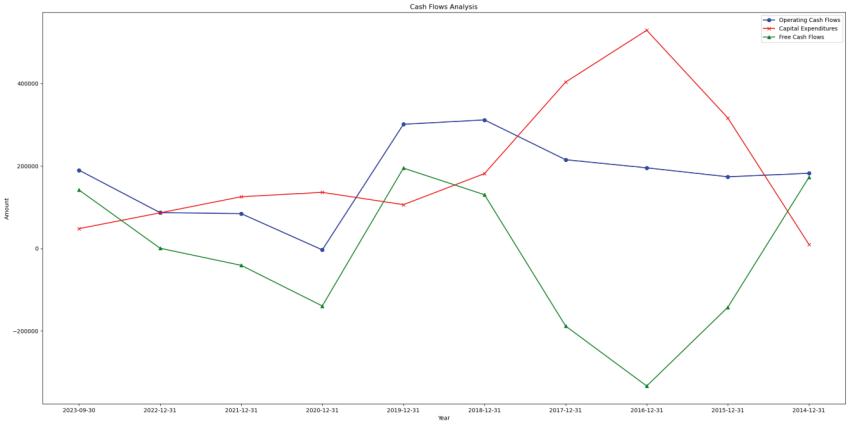

Figure 7: Historical OCF, Capital Expenditure, Free Cash Flow

The entire program is a practical financial analysis tool that combines data processing, financial computation, and a graphical user interface, making it suitable for scenarios that require extracting data from Excel files for financial analysis. Testing has revealed that the program is capable of fetching and calculating data from the financial statements of publicly traded companies in the WIND database. Upon examination, it has been determined that the program can produce accurate results under varying degrees of variable influence. However, there are currently several shortcomings: firstly, the Excel file formats of some publicly traded companies' balance sheets are problematic, requiring manual adjustments before data can be extracted; when calculating the Weighted Average Cost of Capital (WACC), the differences in national tax policies are not taken into account, leading to variations in the results; some function codes are excessively large, and plans are in place to break down these large functions into smaller parts for easier reading and learning. Due to time constraints, only a basic visual interface has been provided for this application. Future developments will focus on enhancing the interface and modular design to enable the use of more financial analysis features.

As the teacher in this development process, I found that the vast majority of students did not have experience developing Python applications. After they mastered the relevant functions and syntax, the students did not know how to independently design new functions. In the later stage of development, two students performed excellently and cooperated to write the code for the visualization interface, data plotting and company valuation modules. Additionally, students provided feedback that their learning time was relatively limited. Indeed, their course schedule was very tight. To address this issue, I believe there are a few ways to improve learning quality, such as increasing lab hours and credits.

More specifically, during the design phase, most students struggled with how to decompose the data processing and analysis APIs into independently functioning functions. While they understood the basic Python syntax and could write simple functions, object-oriented design was still a challenge. To help students gain more hands-on experience, I proposed allocating an additional lab session each week for them to work on modularizing and testing individual functions under guidance. I also encouraged peer learning by having higher performing students assist others. Given the tight course schedule, allocating more time for practice was not feasible within the semester. As an alternative, I proposed awarding extra credits for optional assignments or an extended project, to motivate self-directed learning outside of class hours. This allowed students with more interest or aptitude to further enhance their skills. The extra guidance and learning opportunities helped most students overcome initial difficulties in independent function design. By the coding implementation phase, they were able to systematically integrate functions into the software package with minimal issues.

References

[1] Zhou, Yang. (2024). Application of Python Software in Experimental Teaching of Financial Courses. Electronics Technology, 53(01), 81-83.

[2] Xie, Meifen. (2023). Personal Financial Data Acquisition and Analysis Based on Python. Digital Technology and Application. Digital Technology and Application, 41(05), 118-122.

[3] Luo, Menghua, Wang, Ke. (2022). Application and Research of Financial Data Visualization [J]. Fujian Computer. Fujian Computer, 38(07), 38-41.

[4] Zhang, B. S. (2013). Research on the Reform of Finance Major Courses in Universities—Based on the Perspective of Social Demands. China Market, 30, 186-188.

[5] Alessandro G, Blanka H. (2022). Mathematical Modeling and Computation in Finance: With Exercises and Python and Matlab Computer Codes. Quantitative Finance, 22(11), 1971-1972.

Acknowledgements: This research is supported by the Second Batch of Course Assessment Reform Pilot Project□ Corporate Finance□ of Sanya University.