

# Research on the Application of Python Big Data in Financial Analysis

Qiao Yan<sup>1</sup>

<sup>1</sup> Henan Weijia Automobile Trading Group Co., LTD., Henan, Zhengzhou, China

Correspondence: Qiao Yan, Henan Weijia Automobile Trading Group Co., LTD., Henan, Zhengzhou, China.

Received: May 5, 2025    Accepted: June 11, 2025    Online Published: June 16, 2025

## Abstract

With the rapid development of big data technology and the widespread use of the Python programming language in data science, financial analysis is gradually shifting from traditional manual calculations and analysis to automated and intelligent approaches. This paper explores the application of Python in financial analysis within a big data environment, examining its specific uses in data collection, processing, analysis, and the automation of financial reporting. First, it introduces the advantages of Python and big data technology, along with their integration methods. Next, it outlines the basic concepts and common techniques of financial analysis. Then, it delves into Python's applications in financial analysis, including data preprocessing, financial statement analysis, predictive modeling, and risk management. Through case studies, the paper demonstrates how Python enhances the efficiency and accuracy of financial analysis for businesses. Finally, it summarizes the current state of Python's application in financial analysis, addressing challenges such as data quality issues and the integration of artificial intelligence, while also exploring future trends. This study provides practical guidance for professionals in the financial sector and offers insights for future research.

**Keywords:** python, financial analysis, data preprocessing, financial statements, predictive analysis

## 1. Introduction

As big data technology advances rapidly and Python gains prominence in data science, financial analysis is evolving from traditional manual computation to automated and intelligent methodologies. Python, with its concise syntax and robust data analysis libraries, has emerged as a vital tool in financial analysis. The adoption of big data technology enables businesses to handle vast amounts of data, uncover patterns, and make precise predictions. This paper investigates the application of Python and big data technology in financial analysis, focusing on practical implementations in data collection, preprocessing, automated reporting, and predictive modeling. Through real-world examples, it illustrates how Python improves the efficiency and accuracy of financial analysis, supporting better decision-making for enterprises.

## 2. Overview of Python and Big Data Technology

In today's data-driven business landscape, the integration of Python and big data technology offers fresh approaches and perspectives for financial analysis. Python, an efficient and beginner-friendly programming language, has become a cornerstone in finance and data analysis thanks to its clear syntax and rich ecosystem of scientific computing tools[1]. Leveraging powerful third-party libraries such as Pandas, NumPy, and Matplotlib, Python excels at data cleaning, statistical analysis, and visualization, providing flexible and diverse technical support for tasks like financial statement analysis, trend forecasting, and risk assessment. Meanwhile, the rapid evolution of big data technology allows companies to collect, store, and process massive volumes of financial data at lower costs and higher speeds, laying a solid foundation for real-time monitoring and precise decision-making[2].

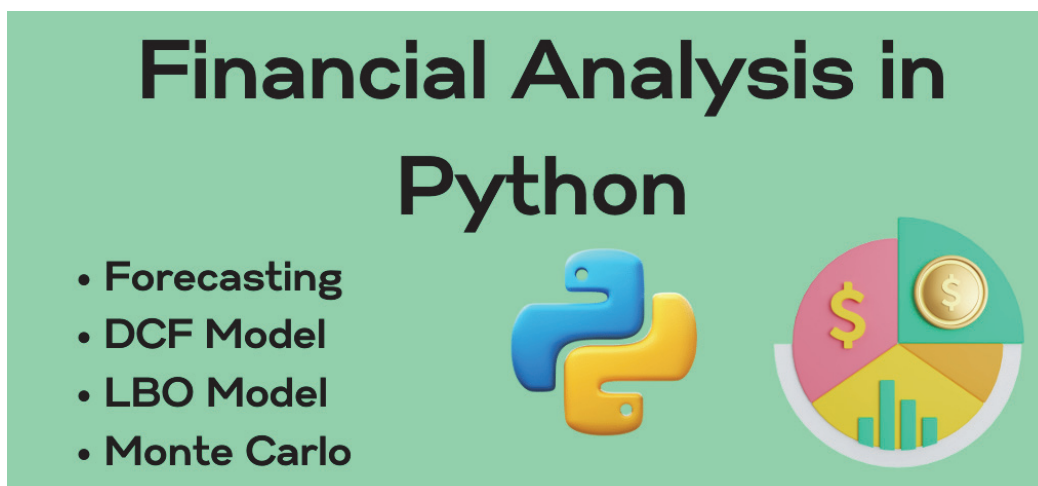


Figure 1. Core Application Scenarios of Python in Financial Analysis

As shown in Figure 1, Python is applied in various key areas of financial analysis, such as Forecasting (predictive analysis), DCF Model (discounted cash flow model), LBO Model (leveraged buyout model), and Monte Carlo (Monte Carlo simulation). Predictive analysis uses historical data and statistical models to provide insights into future financial trends and market shifts. The discounted cash flow model helps evaluate the intrinsic value of a company or project, widely used in investment and merger decisions. The leveraged buyout model assesses the feasibility and risks of acquiring a target company under external financing conditions. Monte Carlo simulation, by generating and analyzing numerous random scenarios, enables businesses to comprehensively evaluate the uncertainties and potential risks of investment projects. These models rely heavily on processing and analyzing large datasets, and the combination of Python and big data technology provides a robust technical backbone for their implementation. In practice, big data platforms like Hadoop and Spark integrate seamlessly with Python, enabling businesses to efficiently process financial data in distributed environments and overcome the performance limitations of traditional database systems when handling massive datasets. Furthermore, the deep integration of Python and big data has spurred the development of financial analysis methods based on machine learning and deep learning[3]. Companies can now uncover hidden patterns in vast datasets, monitor financial conditions in real time, and accurately predict market changes. As a result, Python has become an indispensable tool in financial analysis and big data technology. To deploy these tools effectively, businesses need not only proficiency in Python programming and data analysis but also an understanding of big data platform architectures and operations. This knowledge equips them to make more accurate and efficient financial decisions in an ever-changing business environment.

### 3. Basic Concepts and Methods of Financial Analysis

#### 3.1 Definition and Objectives of Financial Analysis

Financial analysis typically refers to the systematic organization, comparison, and study of a company's financial statements—such as the balance sheet, income statement, and cash flow statement—along with other relevant data. This process aims to uncover the outcomes of business operations, assess the current financial condition, and predict future development trends. Its primary objectives include:

- Evaluating Business Performance:** By analyzing metrics like profit margins and return on assets, it assesses a company's profitability and operational efficiency[4].
- Measuring Solvency and Liquidity Risks:** Indicators such as debt ratios and current ratios are used to evaluate a company's ability to repay debts, helping to identify potential liquidity risks in a timely manner.
- Supporting Decision-Making and Planning:** It provides a basis for decisions by management, investors, and creditors, such as investment strategies, financing options, and internal management improvements.
- Forecasting Future Development:** Based on historical data and market trends, it predicts a company's future operational and financial performance to support more informed strategic planning[5].

#### 3.2 Common Methods and Tools in Financial Analysis

In practice, financial professionals often combine quantitative and qualitative approaches to evaluate a company's financial health and risk levels from multiple perspectives. Quantitative methods primarily rely on traditional techniques like ratio analysis, trend analysis, and structural analysis, while qualitative methods focus on factors

such as industry context, regulatory environment, and management capabilities. Beyond these conventional approaches, recent advancements in data technology have led to the growing adoption of tools based on Python and big data platforms, offering more efficient and intelligent support for financial analysis[6].

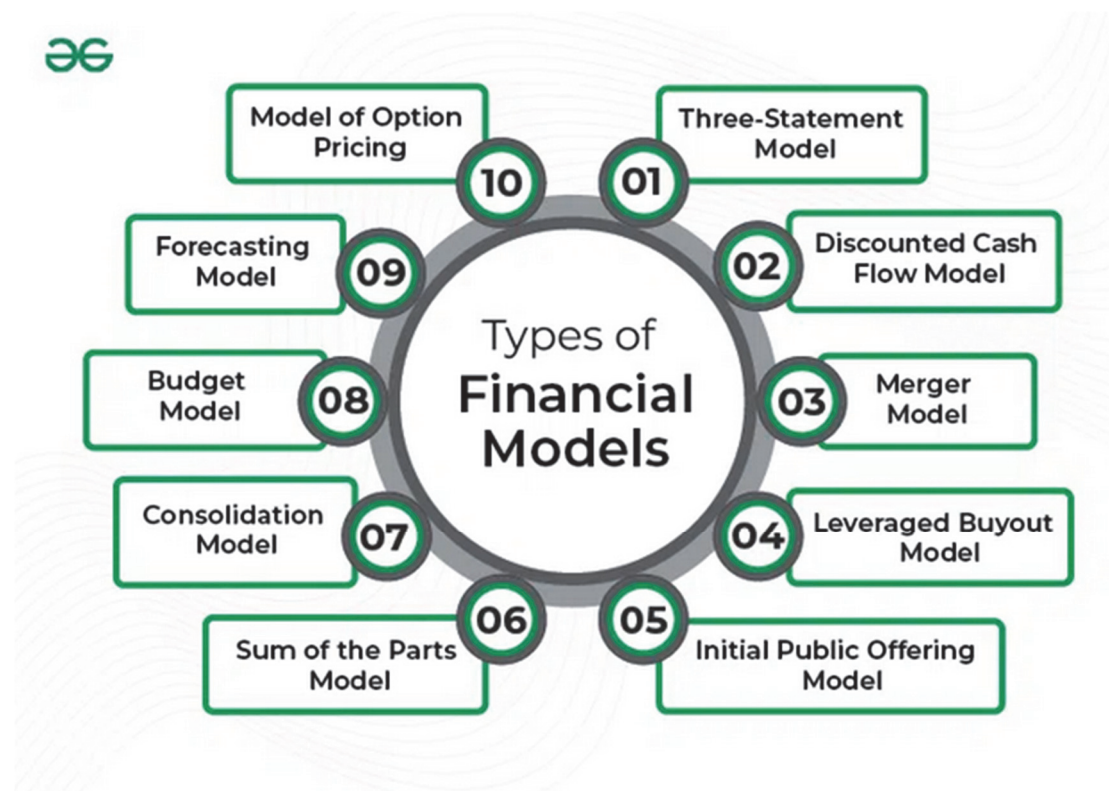


Figure 2. Common Model Types in Financial Analysis

As illustrated in Figure 2, the range of models used in financial analysis is diverse, including the Three-Statement Model, Discounted Cash Flow (DCF) Model, Consolidation Model, Budget Model, Forecasting Model, and Model of Option Pricing, among others. Different models serve distinct analytical purposes: for instance, the Leveraged Buyout (LBO) Model is commonly used to study the feasibility of acquisition deals; the Initial Public Offering (IPO) Model provides valuation support for companies going public; and the “Sum of the Parts” Model evaluates individual business segments separately before aggregating them into an overall valuation. These models cater to specific business needs and industry characteristics, enabling companies or investors to more accurately measure value, assess risks, and develop strategies[7].

### 3.3 Characteristics and Processing Needs of Financial Data

Financial data typically exhibits several notable characteristics: **Accuracy and Compliance:** Since financial data directly impacts the reliability of company reports and investor confidence, it demands a high level of accuracy and adherence to regulatory standards. **Timeliness and Continuity:** Financial data requires regular or even real-time updates to meet the needs of internal management and external disclosure. **Multidimensionality and Multiple Sources:** Financial data originates not only from various internal business units but may also require integration with external economic data, industry insights, and other sources for comprehensive analysis. **Security and Confidentiality:** As financial data often involves sensitive business information, stringent security and confidentiality measures are essential to prevent leaks or misuse. Given these characteristics, financial analysis requires efficient data management systems paired with appropriate tools for data cleaning, visualization, and modeling to ensure the accuracy and timeliness of results[8]. With the increasing application of Python and big data technology, companies now have access to a broader range of technical solutions for processing, storing, and analyzing financial data. These advancements enable faster extraction of valuable insights, providing robust support for decision-making.

## 4. Applications of Python Big Data Technology in Financial Analysis

### 4.1 Data Collection and Processing Workflow

As shown in Figure 3, a complete financial data analysis process typically involves the following key steps: setting objectives, data acquisition, data cleansing, data analysis, and presenting results.

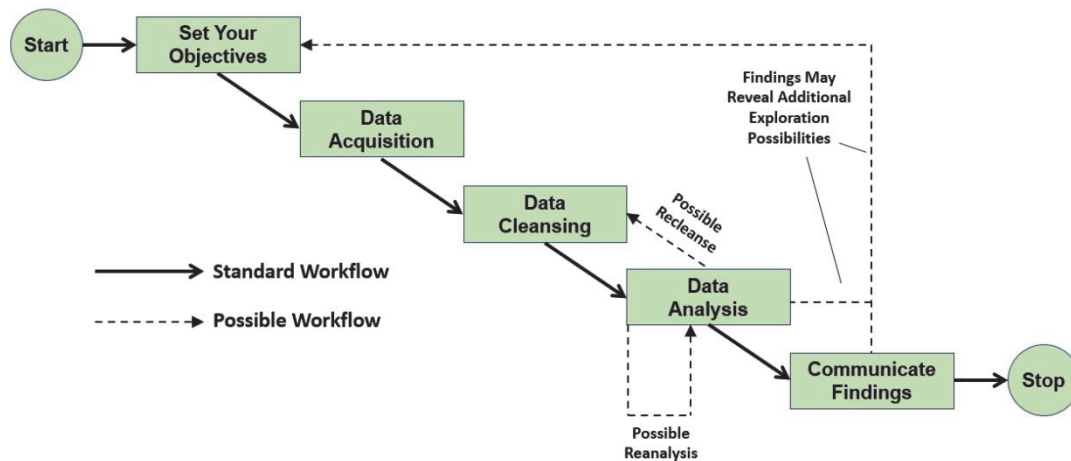


Figure 3. Schematic Diagram of the Financial Data Analysis Workflow

**Set Your Objectives:** Before collecting data, it's essential to define the goals and scope of the analysis, such as evaluating a company's profitability, forecasting future cash flows, or issuing risk alerts. Clear objectives help determine data sources and technical approaches. **Data Acquisition:** With the support of Python and big data technology, the methods for obtaining financial data have become more diverse. Internal data can be extracted in bulk via database connections, APIs, or distributed file systems like HDFS. External data—such as industry reports, macroeconomic indicators, or market trends—can be gathered in real time using web scraping tools or third-party data interfaces. **Data Cleansing:** Raw data often contains duplicates, missing values, inconsistencies, or noise. Python offers powerful libraries like Pandas and NumPy to perform cleansing tasks, including handling missing values, detecting outliers, converting formats, and merging datasets[9]. A high-quality cleansing process significantly improves the accuracy of subsequent analyses. **Data Analysis:** After cleansing, Python's libraries for statistical analysis, visualization, and machine learning enable multidimensional exploration and modeling of financial data. For example, regression analysis can predict revenue trends, while clustering algorithms can identify distinct customer or business segments. For large datasets or scenarios requiring distributed processing, big data platforms like Spark can be integrated for parallel computing. **Communicate Findings:** The final results need to be presented in an intuitive, accessible way to decision-makers or stakeholders. Beyond standard charts and reports, results can be combined with visualization tools or BI platforms to create dynamic dashboards for real-time monitoring of key financial metrics. Feedback from the results may prompt further iterations or refinements of the data or models, fostering a cycle of continuous improvement. In this workflow, Python and big data technology complement each other: Python simplifies scripting and offers flexible access to various analysis libraries, while big data platforms ensure high concurrency and availability in large-scale data scenarios. By effectively integrating the two, businesses can streamline data collection and processing, laying a solid foundation for risk management, predictive analysis, and decision support[10].

### 4.2 Financial Statement Analysis and Automation

Financial statement analysis is a core component of financial management, providing insights into a company's financial health, operational performance, and cash flows to help decision-makers understand its status and potential risks. Traditionally, processing and analyzing financial statements was a tedious and time-consuming task, with manual calculations and report generation being both inefficient and prone to errors. However, the advent of Python and big data technology has greatly optimized this process. Automated tools and algorithms allow businesses to collect, process, analyze, and present data in a fraction of the time, significantly boosting efficiency and accuracy.



Figure 4. Schematic Diagram of the Financial Statement Analysis Process

As depicted in Figure 4, the financial statement analysis process includes six key steps: List Down the Purpose of Analysis: Before starting, it's crucial to define the purpose and direction of the analysis. This involves identifying specific questions, goals, and discussions with stakeholders to ensure the analysis delivers valuable insights for decision-making.

**Collect Relevant Data:** Gathering the necessary financial data is the foundation of analysis. By integrating data from various sources (e.g., financial statements, market data, industry reports), completeness and accuracy are ensured. Python libraries like 'requests' and 'BeautifulSoup' automate external data collection, while internal data can be extracted and stored in real time using SQL databases or big data platforms like Hadoop and Spark.

**Process Data:** This stage involves cleaning and consolidating raw data, which may contain missing values, outliers, or inconsistent formats. Python's Pandas and NumPy libraries offer robust tools for automating data filling, deduplication, and standardization, ensuring high-quality data for analysis.

**Analyze & Interpret Data:** After processing, the data is analyzed using methods such as ratio analysis, trend analysis, or regression analysis to uncover insights and potential issues within the financial statements. Python libraries like scikit-learn and statsmodels enable efficient statistical modeling, while visualization tools (e.g., Matplotlib, Seaborn) present results clearly.

**Develop Conclusions & Recommendations:** In-depth analysis yields conclusions about the company's current financial status, operational efficiency, and future trends, along with recommendations for improvement. These insights guide more informed decision-making.

**Review & Revise Recommendations:** Financial statement analysis is an iterative process. Results may reveal new issues or opportunities, necessitating a review and revision of the analysis to ensure accuracy and relevance. The application of Python and big data technology in this process—particularly in data collection, cleansing, and analysis—dramatically enhances efficiency and precision. By building automated financial analysis systems, companies can obtain real-time key metrics from financial statements, generate reports, charts, and forecasts automatically, and reduce manual

intervention and errors. Furthermore, integrating machine learning and AI technologies elevates the intelligence of financial statement analysis, enabling faster and more strategic decisions in dynamic market conditions.

#### 4.3 Data Mining and Predictive Algorithm Analysis

With the rise of big data technology, data mining and predictive analysis have become indispensable in financial analysis. Data mining helps businesses uncover hidden patterns and trends in vast financial datasets, while predictive algorithms enable effective forecasting of future financial conditions. Together, these techniques enhance the scientific rigor and accuracy of financial decisions. Common algorithms include regression analysis, time series analysis, and clustering, with three widely used examples detailed below:

##### 1. Linear Regression Model

Linear regression is a staple of predictive analysis in finance, particularly for forecasting metrics like sales or profits. It establishes a linear relationship between a dependent variable and one or more independent variables as shown in Formula 1:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon \quad (1)$$

where  $Y$  is the predicted variable,  $X_1, X_2, \dots, X_n$  are independent variables,  $\beta_0$  is the constant,  $\beta_1, \beta_2, \dots, \beta_n$  are coefficients, and  $\epsilon$  is the error term. Using the least squares method (OLS), coefficients are estimated to predict future financial data.

##### 2. Time Series Analysis

Time series analysis forecasts future trends based on historical data, often applied to predict seasonal fluctuations or trends in financial metrics. A popular model is the Autoregressive Integrated Moving Average (ARIMA), with the formula 2:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} + \epsilon_t \quad (2)$$

where  $Y_t$  is the predicted value at time  $t$ ,  $c$  is a constant,  $\phi_1, \phi_2, \dots, \phi_p$  and  $\theta_1, \theta_2, \dots, \theta_q$  are autoregressive and moving average coefficients, and  $\epsilon_t$  is the error term. ARIMA helps businesses forecast financial trends and volatility for long-term planning.

##### 3. K-means Clustering Algorithm

K-means, an unsupervised learning method, is widely used for customer segmentation and market analysis. In finance, it groups customers or assets based on behavior or credit history, aiding in tailored strategies. The objective function is as shown in Formula 3:

$$J = \sum_{i=1}^k \sum_{j=1}^n \|X_j^{(i)} - \mu_i\|^2 \quad (3)$$

where  $J$  measures the sum of squared distances between data points and cluster centroids,  $k$  is the number of clusters,  $n$  is the number of data points,  $X_j^{(i)}$  is a data point, and  $\mu_i$  is the centroid. By minimizing  $J$ , K-means reveals distinct financial behavior patterns. These algorithms play a critical role in financial analysis. Regression quantifies relationships between variables and outcomes, time series predicts trends, and K-means enables segmentation. Leveraging Python and its libraries (e.g., scikit-learn, statsmodels, Pandas), these methods streamline data processing and analysis, elevating the intelligence of financial decision-making.

## 5. Case Study

### 5.1 Case Background and Data Overview

This case study focuses on "SmartTech Electronics," a well-known electronics manufacturing company facing challenges such as rapid product updates and complex supply chains amid fierce market competition. To enhance decision-making precision, SmartTech Electronics integrated internal and external data to build a real-time updated data warehouse, providing robust support for financial analysis. Primary Data Sources: The data spans quarterly financial reports from 2018 to 2023, capturing key metrics like operating revenue, net profit, and cash flow. A sample data Table 1 is presented below:

Table 1. Internal Financial Statement Data

| Year/Quarter | Operating Revenue<br>(10,000 CNY) | Net Profit (10,000<br>CNY) | Cash Flow (10,000<br>CNY) |
|--------------|-----------------------------------|----------------------------|---------------------------|
| 2018 Q1      | 500                               | 50                         | 80                        |
| 2018 Q2      | 520                               | 55                         | 85                        |
| 2019 Q1      | 600                               | 65                         | 90                        |
| 2019 Q2      | 630                               | 70                         | 95                        |
| 2020 Q1      | 700                               | 80                         | 100                       |
| 2020 Q2      | 720                               | 82                         | 105                       |
| 2021 Q1      | 800                               | 90                         | 110                       |
| 2021 Q2      | 830                               | 95                         | 115                       |
| 2022 Q1      | 900                               | 100                        | 120                       |
| 2022 Q2      | 930                               | 105                        | 125                       |
| 2023 Q1      | 1000                              | 110                        | 130                       |

Additional data includes sales by product line, market share, and inventory turnover rates, alongside supply chain data such as raw material costs and inventory levels. A sample is shown below Table 2:

Table 2. Operational and Sales Data

| Product Line | Sales (10,000 CNY) | Market Share (%) | Average Inventory Turnover |
|--------------|--------------------|------------------|----------------------------|
| Product A    | 300                | 40               | 6                          |
| Product B    | 250                | 35               | 5                          |
| Product C    | 150                | 25               | 4                          |

**Data Collection and Processing:** Using Python scripts and big data platforms, the company automated data collection from internal systems and external markets via APIs. Pandas was employed for data cleansing tasks (e.g., filling missing values, detecting outliers, and format conversion) to ensure data quality and timeliness. Processed data was stored in a distributed database, forming a solid foundation for automated financial reporting, predictive analysis, and risk alerts. This case demonstrates how Python and big data technology can integrate multi-source data to create an efficient financial analysis platform, enabling automated data collection, cleansing, and processing to support real-time, precise decision-making. It offers valuable insights and a practical model for other businesses.

### 5.2 Python Implementation Process in Financial Analysis

In the SmartTech Electronics case, Python served as the core tool for data processing and analysis, with the implementation divided into four stages: data collection, data cleansing and preprocessing, data analysis and modeling, and result visualization. Each stage leveraged Python's rich third-party libraries and automated scripts to create a seamless workflow from data input to decision support.

#### 1) Data Collection and Integration

Using Python libraries like 'requests' and 'BeautifulSoup' for web scraping, and database connectors like 'SQLAlchemy', the company automated the collection of internal system data (e.g., financial reports, sales records, supply chain data) and external market data. Internal data was extracted periodically via scheduled tasks (e.g., cron jobs or scripts), while external data came from public APIs and data providers. All collected data was initially stored in local or cloud-based temporary databases for further processing.

```
BEGIN
```

```
    // Fetch external data from an API
```

```
    SET url = "http://api.example.com/financial_data"
```

```
    SEND HTTP GET request to url
```

```
STORE response in external_data
CONVERT external_data to a structured table format

// Connect to internal database and extract data
SET database_connection = CREATE connection to "mysql://user:password@host/db"
EXECUTE query "SELECT * FROM financial_reports" on database_connection
STORE result in internal_data
END
```

## 2) Data Cleansing and Preprocessing

Raw data often contained missing values, duplicates, or inconsistent formats. Using Pandas and NumPy, the cleansing process addressed these issues through null value imputation, outlier detection, and format standardization. Dataframe operations automated the handling of inconsistent formats, unifying all data into a standard structure. For time series data, date fields were normalized to facilitate subsequent analysis.

```
BEGIN
// Clean internal data
REMOVE duplicate rows from internal_data
FILL missing values in internal_data using forward-fill method
CONVERT 'report_date' column in internal_data to datetime format

// Merge internal and external data
SET merged_data = JOIN internal_data and external_data on 'common_key' with inner join
END
```

## 3) Data Analysis and Model Building

After cleansing, the core financial analysis involved extracting insights using statistical methods and machine learning algorithms. Linear regression and time series forecasting (e.g., ARIMA) were implemented with `statsmodels` and `scikit-learn`. Linear regression analyzed correlations between financial metrics, while ARIMA predicted future trends. K-means clustering was used to segment customers or products, uncovering hidden business patterns. Cross-validation and parameter tuning ensured model robustness and accuracy.

```
BEGIN
// Linear Regression Model
SET X = SELECT columns 'ad_spend' and 'market_growth' from merged_data
SET y = SELECT column 'sales' from merged_data
ADD constant term to X
FIT linear regression model with X as predictors and y as target
PRINT model summary (coefficients, statistics)

// K-means Clustering
INITIALIZE K-means with 3 clusters and fixed random seed
SET features = SELECT columns 'sales' and 'profit' from merged_data
RUN K-means clustering on features
ASSIGN cluster labels to merged_data in new column 'cluster'
END
```

#### 4) Result Visualization and Report Generation

Visualization tools like Matplotlib, Seaborn, and Plotly transformed analysis results into charts, dashboards, and dynamic reports, including trend lines, scatter plots, and clustering visuals. Automated scripts generated and delivered financial reports to management via email or internal systems, enabling real-time monitoring and feedback.

```
BEGIN
```

```
// Visualize sales trend
```

```
CREATE new figure with size (10, 6)
```

```
PLOT line graph with 'report_date' on x-axis and 'sales' on y-axis from merged_data
```

```
SET title to "Sales Trend"
```

```
SET x-axis label to "Date"
```

```
SET y-axis label to "Sales"
```

```
DISPLAY plot
```

```
END
```

This Python-driven process automated the entire workflow from data collection to visualization. SmartTech Electronics not only improved data processing efficiency but also leveraged accurate predictive models to support decision-making and reduce financial risks. The system highlights the immense potential of Python and big data technology in real-world financial analysis, offering strong support for dynamic decision-making and strategic adjustments.

#### 5.3 Case Analysis Results and Discussion

After integrating, cleansing, and modeling SmartTech Electronics' data, this study conducted an in-depth analysis of historical financial data, future forecasts, and customer segmentation. The results, achieved through Python and big data techniques, are presented below with detailed discussions supported by data tables.

##### 1) Historical Financial Data Analysis

Based on quarterly financial data from 2018 to 2023, we analyzed key metrics like operating revenue, net profit, and cash flow. Figure 5 shows quarterly averages and year-over-year growth rates, providing a clear view of the company's financial performance.

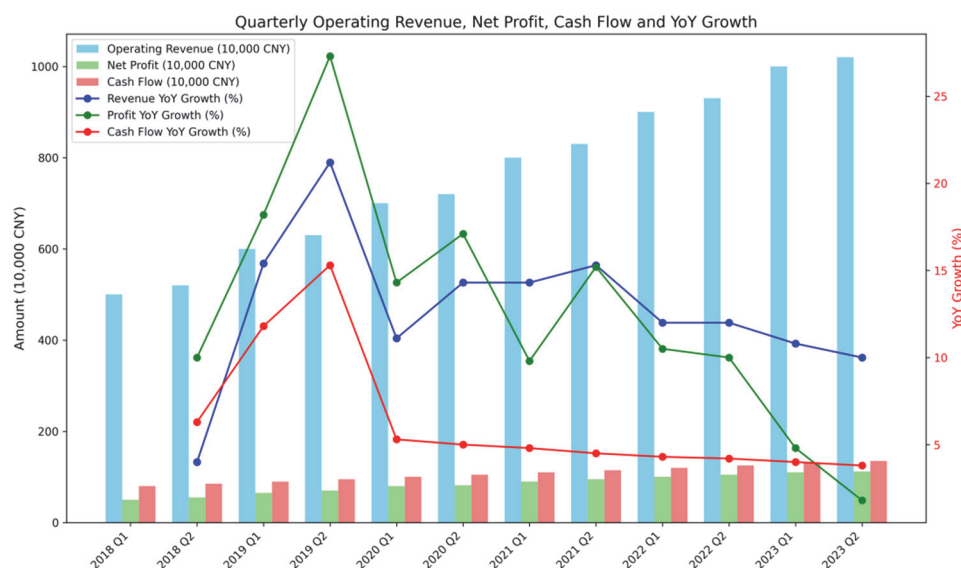


Figure 5. Quarterly Financial Performance and year-to-year growth rate of smart Electronics 2018-2023

From Figure 5, SmartTech Electronics exhibited overall growth from 2018 to 2023, with higher growth rates in net profit and revenue in the early years. Recent years show a stabilization of growth, likely due to intensified market competition and strategic adjustments. Cash flow grew steadily, reflecting strong financial stability and risk resilience.

## 2) Predictive Model Results

An ARIMA model was used to forecast future quarterly operating revenue, achieving a high goodness-of-fit with historical data. Figure 6 compares actual and predicted values, including forecast errors.



Figure 6. Comparison of ARIMA model forecast operating revenue and actual value

Discussion: Figure 6 shows the ARIMA model performs well in fitting historical data and short-term forecasting, with absolute and relative errors below 1.5%. This reliability supports SmartTech Electronics in anticipating revenue trends for strategic planning and resource allocation.

## 3) Customer Segmentation via Clustering

K-means clustering was applied to customer data to understand market and customer characteristics. Figure 7 summarizes the key features of each cluster, including average sales, profit, and customer count.



Figure 7. Characteristics of customer groups for K-means clustering analysis

The analysis identified three customer groups: Cluster 0 (moderate sales, low profit) likely represents price-sensitive customers; Cluster 1 (high sales and profit) is the core customer base; Cluster 2 (low sales and profit) shows development potential. Tailored strategies—e.g., higher credit for core customers and increased marketing for potential ones—can optimize customer and revenue structures. Overall, this case demonstrates the successful application of SmartTech Electronics' Python-based financial analysis system across multiple dimensions. Historical data analysis revealed performance trends, ARIMA forecasts provided future insights, and K-means clustering enhanced customer targeting. These results validate Python's capabilities in data processing and modeling, offering a solid foundation for scientific and refined financial strategies. As data volumes grow and algorithms improve, such technologies will yield deeper insights, helping businesses navigate competition and uncertainty.

## 6. Conclusion

Through the SmartTech Electronics case, this paper illustrates the practical application of Python and big data technology in financial analysis. By integrating data collection, cleansing, analysis, and predictive modeling, the company efficiently processed vast datasets, achieving automated financial reporting, trend forecasting, and customer segmentation. The results highlight the significant advantages of Python and its libraries (e.g., Pandas, NumPy, scikit-learn) in enhancing analysis efficiency, accuracy, and decision support. Looking ahead, as technology evolves, Python and big data will play an increasingly vital role in financial analysis, empowering businesses to tackle complex market dynamics and boost competitiveness.

## References

- [1] Goh, C., Kusnadi, Y., & Pan, G. (2022). The effectiveness of using Python programming approach in teaching financial analytics. *Advances in Economics and Management Research*, 5(1), 1.
- [2] Liang, Y., Liu, Y., Chen, C., & Jiang, Z. (2020). Financial big data analysis and early warning platform: A case study. *IEEE Access*, 8, 36515–36526. <https://doi.org/10.1109/ACCESS.2020.2974897>
- [3] Leung, M.-F., Wang, J., & Li, D. (2023). A portfolio recommendation system based on machine learning and big data analytics. *Data Science in Finance and Economics*, 3(2), 152–165. <https://doi.org/10.3934/DSFE.2023009>
- [4] Du, P., & Shen, Y. (2023). Analysis of international stock data based on Python. *Advances in Economics and Management Research*, 5(1), 146.
- [5] Li, Y. (2022). Python data analysis and attribute information extraction method based on intelligent decision

- system. *Mobile Information Systems*, 2022, 2495166. <https://doi.org/10.1155/2022/2495166>
- [6] Bose, S., Dey, S. K., & Bhattacharjee, S. (2023). Big data, data analytics and artificial intelligence in accounting: An overview. In *Handbook of big data research methods* (pp. 32–51). Edward Elgar Publishing. <https://doi.org/10.4337/9781800888555.00009>
- [7] McBride, K., & Philippou, C. (2022). “Big results require big ambitions”: Big data, data analytics and accounting in masters courses. *Accounting Research Journal*, 35(1), 71–100. <https://doi.org/10.1108/ARJ-04-2021-0132>
- [8] Siano, F., & Wysocki, P. (2021). Transfer learning and textual analysis of accounting disclosures: Applying big data methods to small(er) datasets. *Accounting Horizons*, 35(3), 217–244. <https://doi.org/10.2308/HORIZONS-19-111>
- [9] Salijeni, G., Samsonova-Taddei, A., & Turley, S. (2021). Understanding how big data technologies reconfigure the nature and organization of financial statement audits: A sociomaterial analysis. *European Accounting Review*, 30(3), 531–555. <https://doi.org/10.1080/09638180.2021.1883700>
- [10] Zhang, Z., Ye, F., Ma, Y., Li, Q., & Huang, H. (2023). Analysis of large data sets in a physical chemistry laboratory NMR experiment using Python. *Journal of Chemical Education*, 100(10), 4109–4113. <https://doi.org/10.1021/acs.jchemed.3c00476>

### Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).