A Smart Automator for Personal Finance Management: Using NLP to Simplify Spending Analysis

Arunachalam T
School of Computer Science and Engineering,
Presidency University,
Bengaluru-560064, India.
arunachalam.official224@gmail.com

Rakshitha D S
School of Computer Science and Engineering,
Presidency University,
Bengaluru-560064, India.
rakshithads1204@gmail.com

Vaishnavi Sastry S
School of Computer Science and Engineering,
Presidency University,
Bengaluru-560064, India.
vaishnavisastry04@gmail.com

Dr.Sreelatha P K
School of Computer Science and Engineering,
Presidency University,
Bengaluru-560064, India.
sreelatha.pk@presidencyuniversity.in

Abstract tracking personal finances remains a very challenging and time-consuming task. There are many individuals who find it difficult to keep a manual log of their expenses or to understand their spending habits. We propose a Smart Personal Finance Automator which is a web application designed to meet the needs of such people. Our system fetches automatically and also processes financial transaction messages from SMS and emails. The system employs Natural Language Processing (NLP) to interpret, Categorise (e.g., “food”, “travel”, “utilities”), and Analyse expenditure patterns of the messages. This information is then made available to the user through an easy, pictorial dashboard which gives clear insights and also provides actionable recommendations to help them make smarter financial decisions. Index Terms Personal Finance, Natural Language Processing (NLP), Automation, Data Visualization, Fin Tech, Machine Learning.

I. Introduction

We live in a world that is interconnected digitally and transactions are made on a daily basis. Morning coffee, a bus fare, an online subscription are just a few examples of everyday transactions. One of the drawbacks of this frequency is that it becomes very difficult to answer the question: “Where are my money actually going?” Many people use manual spreadsheets or budgeting apps to manage their finances, but these tools demand an incessant and laborious data entry. It is quite likely that a transaction will be forgotten or that the whole process will be abandoned, thus leaving users in the dark as to their financial status. Our project, the smart Personal Finance Automator, is a solution to this problem. We have created a program that is capable of doing the work that the user would usually do but with less effort and greater efficiency. The program takes the lead in gathering transaction messages from SMS and email, then artificial intelligence is employed to comprehend the content and finally, the messages are organised into clear categories.

The system is built on four key ideas:

- **Automatic Data Collection:** The process of fetching newly available transaction data from SMS and email alert is done in a secure manner.
- **NLP Categorisation:** Utilises the spaCy NLP library to interpret the unstructured text, pinpoint the idea (such as the amount and merchant), and complete the transaction.
- **Smart Recommendations:** Creates individual savings plan for users and provide them with useful alerts in the event that it recongnises an instance of overspending in a particular category.
- **Clear Visualization:** Uses an interactive dashboards to display all the data (with the help of Plotly and Chart.js) that comes in the form of easy-to-understand graphs and charts which, at a glance, provide the user with information about their spending habits.

This document presents the building of this instrument that was our undertaking, starting from the core technologies we researched, the system’s architecture, and our preliminary findings.

II. Literature Survey

To build our system, we explored innovations across several fields. Our work stands on the shoulders of giants in NLP, data visualization, and financial technology.

A. Natural Language Processing

First, to manage raw and unstructured texts from SMS or email, we studied landmark NLP works of Bird, Klein, and Loper [1]. Their paper acted as the layout for elementary text handling. After that we explored several tough but at the same time up to date NLP libraries. SpaCy[3] is one such library which is meant for "industrial strength" NLP. In fact, entity extraction is one of the most powerful methods it offers and this method is what we chose to use in identifying transaction amounts, merchant names, and dates in hard to understand message formats.

B. Visualization and Interaction

If data is not understandable, then it's not useful. Chen, Chiang, and Storey [7] emphasized the role of business intelligence tools in transforming vast data masses into actionable insights. Using this idea, we have personal finance through technological tools such as Plotly and Dash connected with Flask and Jinja2 templates for the live visuals. With these tools, we are able to develop web based interactive financial dashboards where users visually track expenses, spot trends, and make financial decisions of a real-time nature.

C. Machine Learning Foundations

The categorisation and recommendation functionalities in our project have been conceptualised from the machine learning theoretical principles. Along with the deep learning base paper by LeCun, Bengio, and Hinton [8], the machine learning library TensorFlow [4] was designed to provide the implementation tools for large scale models. Besides, Syntetos and Boylan [9] gave an example of how machine learning can be used to predict expenses, thus in our case, we employ this idea to generate budgets as well as setting up alerts.

D. FinTech and Personal Finance

The financial world is going through a digital transformation at a breakneck speed. Maintaining this momentum, Agarwal and Dhar [5] investigated the influence of digital instruments on extending financial services to the unbanked population with the emphasis on the necessity of user friendly platforms. Other authors, like Jain, Kumar, and Singh [6], and Singh and Malhotra [10], have studied the evolution of AI powered financial tools, concluding that features like ours can really lead to better budgeting habits.

E. Our Contribution: Bridging the Gap

These research fields have always been of interest to scholars, however, we discovered that very few tools actually combine them in a single, user friendly platform for a layperson. Current applications mostly depend on manual data input, are tailored to big businesses, or provide only brief analytics. The particular gap that we address is the real time, automatic extraction of unstructured SMS/email data, which is directly linked with an interactive dashboard and personalized recommendations. Our Smart Personal Finance Automator is designed to close this gap by combining these technologies so as to make everyday money management more convenient.

III. System design and Architecture

The development of our system was based on a full-stack Python architecture centered around Flask. Frontend development makes use of HTML5, CSS3, Bootstrap, and JavaScript (ES6) and is facilitated by Flask's Jinja2 template engine for server side integration. Backend performs data processing, NLP tasks, and API logic through Flask and support extensions like Flask SQLAlchemy, Flask Login, and Flask Bcrypt. The System defaults to storing securely transaction data in an SQLite database but also offers PostgreSQL/MySQL support via the SQLAlchemy ORM. The user's data moves through the system in a well defined manner, which components we have outlined below:

A. Data Ingestion

Initially, the point of the whole exercise is to collect the raw data. Using Google API Python Client and the corresponding authentication libraries (Google Auth HTTPLib2, Google Auth OAuthLib), the system establishes a secure connection to the user's Gmail inbox. It selects emails coming from known financial institutions. Since each bank has its own format, we have built a Custom Regular Expression Engine that can locate and extract transaction details from Indian banking messages.

B. NLP Processing Pipeline

When a raw message is available, it goes to our NLP pipeline, which is the 'brain' of the unit. This process involves several key stages:

1) **Text Cleaning:** This would involve getting rid of financial jargon, promotional texts, and the likes of 'noise', so that one will be able to focus on the core transaction data.
2) **Entity Recognition:** We employ spaCy and NLTK to identify the main entities like the transaction amount, merchant name, and the date.
3) **Classification:** Next to the extraction stage, a classification tool will judge the transaction and assign it one out of the already existing categories

(for instance, 'Groceries,' 'Transport,' 'Utilities'). Initially, this model is grounded on a rule base and words that signify the context, however, the model can later be replaced by a machine learning classifier as more labeled data will be available.

C. Backend, Database, and API

The Flask backend is the main control centre or the central hub. To the frontend, it opens up a secure REST API. One lead, structure data such as (e.g., '({"date": "2025-11-05", "merchant": "Zomato", "amount": 250, "category": "Food"}') is routed through the API and saved in the SQLite database by means of SQLAlchemy ORM. This plan makes it possible to use other databases like PostgreSQL or MySQL without changing the performance of the queries, handling of the relationships, and the capabilities of the aggregations.

D. Frontend Visualization

The frontend, which is supported by HTML5, Bootstrap, JavaScript, and Plotly.js/Chart.js, gets requests to the backend by AJAX and Fetch API. It uses interactive dashboards to show categorized financial data. These are some of the visual tools used: pie charts for category wise spending and bar graphs for monthly expenditure trends. The interface is designed in a dark theme with white text; it is fully responsive and keeps things up to date by means of regular API calls. Help messages and changing graphics provide users with a smooth experience and live data change feedback.

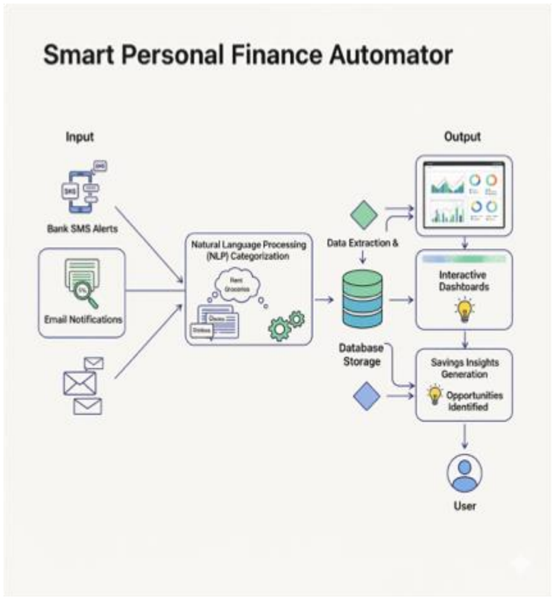
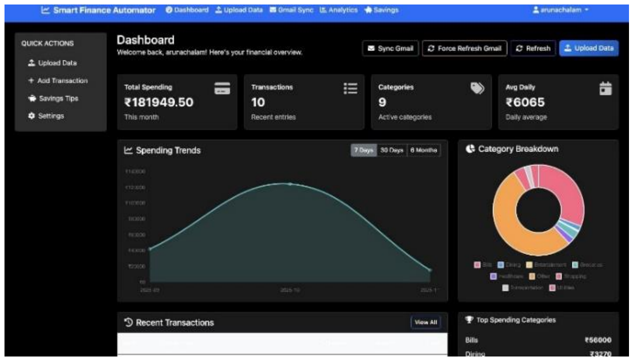


Figure 1. Architecture of the Smart Personal Finance Automator.

Another point we emphasized was user data privacy through the system that could scale and handle large volumes of transactions.

IV. Implementation and Early Result

We put the main focus of our implementation on the accuracy of the NLP classifier. Our very first trials of the classification model were quite optimistic as it reached between 85% and 90% of its accuracy. This measure was obtained by executing the whole process on a test set of 500 bank SMS messages that had been manually tagged and came from different providers. Automation through the system, as opposed to manual data entry, is said to cut down the user's effort by 70% approximately. This number is derived from initial user tests in which participants were timed while they manually entered data and when they just verified the automator's classifications. The main expected effect is the enhancement of financial discipline among our users. We believe that this transparent insight into spending will result in user satisfaction at a high level. The initial reaction of the small test group goes in this direction as 8 out of 10 users (80%) expressed that the dashboard provided them with a "much clearer" understanding of their spending habits than before. Besides that, the system's performance is also good as it processes the new transactions and allows them to be seen on the dashboard almost instantly (in less than 5 seconds).

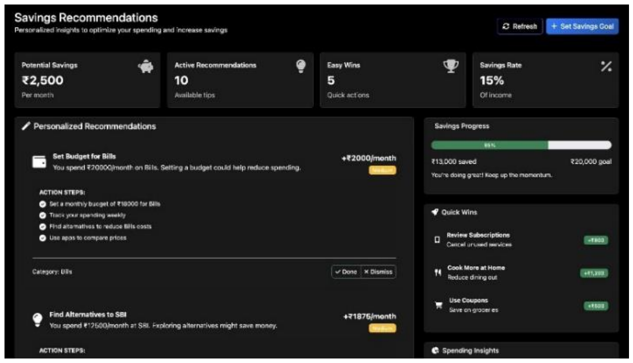


Dashboard
Welcome back, arunachalam! Here's your financial overview.

Total Spending ₹181949.50 This month
Transactions 10 Recent entries
Categories 9 Active categories
Avg Daily ₹6065 Daily average

Top Spending Categories
Bills ₹56000
Dining ₹3270

Figure 2. Smart Finance Automator Dashboard displaying spending overview and breakdown.



Savings Recommendations
Personalized insights to optimize your spending and increase savings

Potential Savings ₹2,500 Per month
Active Recommendations 10 Available tips
Easy Wins 5 Quick actions
Savings Rate 15% Of income

Set Budget for Bills +₹2000/month
You spend ₹20000/month on Bills. Setting a budget could help reduce spending.
ACTION STEPS:
- Set a monthly budget of ₹18000 for Bills
- Track your spending weekly
- Find alternatives to reduce Bills costs
- Use apps to compare prices

Category: Bills

Find Alternatives to SBI +₹1875/month
You spend ₹12500/month at SBI. Exploring alternatives might save money.
ACTION STEPS:

Savings Progress
₹13,000 saved ₹20,000 goal
You're doing great! Keep up the momentum.

Quick Wins
Review Subscriptions — Cancel unused services
Cook More at Home — Reduce dining out
Use Coupons — Save on groceries

Figure 3. Savings Recommendations Dashboard showing personalized tips and progress.

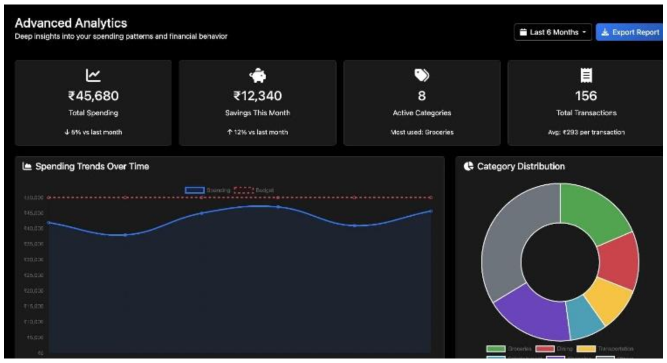


Figure 4. Advanced Analytics Dashboard with spending trends and category insights.

V. Conclusion and Future Work

Our work on the Smart Personal Finance Automator illustrates that the integration of smart automation (NLP)with intuitive data visualization can really make the complex area of personal finance a simple one. This platform is very powerful since it lessens the user's manual work, increases the correctness of their financial records, and makes available trustworthy insights.We are very positive about the platform's future. Our subsequent steps are to add predictive analytics that will probably be done by means of time series models like ARIMA or Prophet in order to be able to forecast monthly expenses and thus help users in budgeting more effectively. We also intend to create anomaly detection, which might be achieved by using clustering algorithms (like DBSCAN) or autoencoders, in order to identify abnormal transactions and thus enable a fraud prevention system to be added. Lastly, in order to be able to publicly present the instrument to a larger number of people, we are going to collaborate on language support extension which is an entailment of retraining our NLP models on multilingual datasets.**Acknowledgement**.

The authors would like to express their sincere gratitude to their guide, Dr.Sreelatha P.K., Assistant Professor, for her invaluable guidance, constant encouragement, and insightful feedback throughout this project.

References

1. S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.
2. Plotly Technologies Inc., "Dash User Guide," 2023. [Online]. Available: https://dash.plotly.com
3. spaCy, "Industrial-Strength NLP in Python," 2023. [Online]. Available: https://spacy.io
4. M. Abadi et al., "TensorFlow: Large-Scale Machine Learn- ing on Heterogeneous Systems," 2015.[Online].Available: https://www.tensorflow.org/
5. R. Agarwal and A. Dhar, "Financial Inclusion through Digitalization: A Review of Financial Technologies," *Int. J. Innov. Sci.*, vol. 13, no. 2, pp. 168–180, 2021.
6. A. Jain, S. Kumar, and M. Singh, "AI-Driven Personal Finance Management Systems," *Int. J. Comput. Appl.*, vol. 183, no. 25, pp. 12–19, 2021.
7. H. Chen, R. Chiang, and V. Storey, "Business Intelligence and Analytics: From Big Data to Big Impact," *MIS Q.*, vol. 36, no. 4, pp. 1165–1188, 2012.
8. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, 2015.