Nigerian Journal of Technology (NIJOTECH)
Vol. 39, No. 2, April 2020, pp. **553 - 561**
Copyright© Faculty of Engineering, University of Nigeria, Nsukka,
Print ISSN: 0331-8443, Electronic ISSN: 2467-8821
www.nijotech.com
http://dx.doi.org/10.4314/njt.v39i2.26

DEPLOYING DATA MINING TECHNIQUES TO GAIN DEEPER INSIGHT INTO NIGERIAN CUSTOMERS' FINANCIAL ACTIVITIES

A. M. Nwohiri [1,]* and F. T. Sonubi[2]
[1,] DEPARTMENT OF COMPUTER SCIENCES, UNIVERSITY OF LAGOS, AKOKA YABA, LAGOS STATE, NIGERIA
[2,] COWRYWISE, 5C REVEREND OGUNBIYI STREET, IKEJA, LAGOS, LAGOS STATE, NIGERIA
E-mail addresses: [1] *anwohiri@unilag.edu.ng,* [2] *feyi@cowrywise.com*

ABSTRACT

Presently, Nigerian banks issue account statements in a tabular flat form. These statements mainly show basic logs of credit and debit transactions. They do not offer a deeper insight into the pure nature of transactions. Moreover, they lack rich mine-able data, and rather contain basic data tables that do not provide enough insights into customers' monthly/weekly/yearly expenses and earnings. In today's fast-paced digital world, where information processing methods are rapidly changing, customers need not just a basic table of transactions but deeper analysis and detail report of their finances. This paper aims at identifying and addressing these problems by deploying data mining techniques and practices in building an application that helps customers gain a deeper insight and understanding of their spending and earnings over a particular period. Some of the techniques used are classification, statistical analysis, visualization, report generation and summarization.

Keywords: *Data mining, API, Anomaly Detection, GTBank, CBN, Bank statements, Nigeria*

1. INTRODUCTION

Ever since digitization was made mandatory by the Central Bank of Nigeria for the financial activities of Nigeria-based banks [1-4], bank statements are now issued in the form of data tables in pdf spreadsheets. Such statements only tabulate transaction details, which usually include transaction dates, reference, amount and any other data the bank deems relevant to a customer. Tech-savvy customers are able to find their way around, opening spreadsheets and going through logs of their transactions to see how their finances have been in a particular period. However, the above approach comes along with poor user experience in terms of how statements are used by customers who might not have the time to go through all their transactions to find some needed information.

This paper seeks to provide answers to the following questions:

- Are there better ways bank statements can summarize and present financial data?
- How can data mining of existing statements help customers find more meaning in their transactions?
- Can data mining techniques improve overall accessibility and comprehensibility of how bank statements are presented to customers?
- In what ways can data in bank statements be mined to bring out relevant financial data?
- Why will the existing method of presenting statements pose a user experience problem to customers who are already used to them?

Data mining is the process of finding anomalies, patterns and correlations within large data sets to predict outcomes [5, 6]. With data mining, you analyze concealed data patterns based on different views for the purpose of classifying it into useful information. This information is assembled in shared areas, such as data warehouses, for systematized analysis, data mining algorithms, facilitating decision

* Corresponding author, tel: +234-814-546-5855

making to reduce costs and boost income. Data mining is known as data discovery and knowledge discovery [7]. Its methods are of two major kinds: supervised and unsupervised. Both embrace features that can find various concealed patterns in large data sets.

Unsupervised data mining focuses not on pre-established attributes. It does not predict a target value but instead tries to reveal hidden structure and identify any relationship among data [8]. Cluster analysis is the most common unsupervised learning method [9]. It is applied for exploration-based analysis to uncover hidden patterns in data.

Supervised learning methods are suitable when there is a particular target value you would like to predict about your data. It uses a set of labelled training examples, including predictors and results, and the examples are used for execution [10].

The current approach used by banks in presenting statements to customers lacks summarization, classification and visualization of a customer's transactions to show the most relevant data and insights to how a customer spends and earns periodically.

According to [11], data mining involves six common classes of tasks:

- Anomaly Detection: The process of identifying curious unusual data records. In the context of this research, this can come in the form of a sudden spike in the earnings of a bank customer over a short period of time.
- Summarization: The process of providing a denser representation of the data set.
- Classification: Looks for new patterns that might result in a change in how the data is organized, for example, classifying grocery transactions and according them a "Grocery" label.
- Clustering: Here, we find and visually document groups of facts that were previously unknown to a customer.
- Regression: Involves finding a relation which models the data with minimum error.
- Association rule learning: searches for patterns inherent in variables. For example: the bank can use a customer's average income and balance to suggest investment options to the customer.

In addressing the above questions, this paper aims at providing a concise, classified and summarized view of a user's monthly transactions. It discovers hidden patterns in data, plugs leakages in how customers spend on a monthly basis, creates visual context by helping customers understand the significance of their data which might go undetected when data is presented in a simple text format, and helps customers predict and make better financial decisions.

2. RELATED RESEARCH

Thanks to its prediction and classification potentials, data mining has been deployed to lubricate the auditing and financial reporting process, generate comprehensive reports, visualize data and facilitate credit risk estimation.

Application of data mining techniques on financial data has been widely used for bankruptcy prediction. Corporate bankruptcy causes economic damages for businesses, management, investors, creditors and customers.

Edward Altman [12] was the first to use financial statements for bankruptcy prediction. He claims that corporate failure is a long-term process and that financial statements should be able to warn of any imminent bankruptcy. In his work, he applied multiple discriminant analysis techniques to develop a bankruptcy prediction model.

Since the work of Altman, many researchers have developed alternative models that are based on statistical techniques. In [13], the author employed logistic regression to predict company failure. The work used the logit model and US firms to develop an estimate of the probability of failure for each firm. The result shows that the proposed method has obvious advantages in predicting corporate financial stress.

The Zmijewski Score [14] – a bankruptcy model used to predict a firm's bankruptcy in two years – used probit analysis.

Researchers have attempted to forecast corporate failure via four different techniques [15]. Two of the methods used statistical techniques, while the other two methods were based on machine learning. Additionally, a hybrid algorithm was proposed. The sample included data about 1133 UK companies. A total of 690 non-failed and 106 failed companies were used as the training set, where 289 non-failed and 48 failed companies were used as the testing set.

Two models for identifying falsified financial statements from publicly available data were developed in [16]. In the first model, input variables featured nine financial ratios, while in the second model, z-score was included as input variable to factor in the relationship between financial distress and financial statement manipulation.

In commercial lending, risk assessment is an attempt to quantify the risk of loss to the lender when deciding to lend. Data mining techniques helps to distinguish borrowers who repay loans promptly from those who don't. It also helps to predict when the borrower is at fault and whether a particular customer is worth giving a loan, etc. Some group of researchers performed credit rating analysis by using support vector machines (SVMs), a machine learning technique [17]. Two data sets were used - one containing 74 Korean firms and the other containing 265 US firms. For both data sets, 5 rating categories were defined. Two models for Korean data set and two models for US data set, each one having a different input vector. SVMs and a back-propagation neural networks were used to predict credit rating. SVMs performed better in three of the four models. The study also aimed at interpreting the neural network. The Garson method was used to measure the relative importance of input values.

Neural networks are popular and support vector machines (SVMs) have been applied. A three-layer, feed-forward Radial Basis Function (RBF) neural network was used in [18] with only two training passes needed to produce a fraud score in every two hours for new credit card transactions.

A group of scientists from Israel [19] presented a two-stage rules-based fraud detection system that involved generating rules using a modified C4.5 algorithm. After generating the rules, they were sorted based on accuracy of customer level rules, and selected based on coverage of fraud of customer rules and difference between behavioural level rules. It was applied to a telecommunications subscription fraud. Boosted C5.0 algorithm was applied on tax declarations of companies [20]. A variant of C4.5 was applied for customs fraud detection in [21].

Popular supervised algorithms such as neural networks, Bayesian networks, and decision trees have been used in combination or sequentially to improve mining results. Authors in [22] utilized naive Bayes, C4.5, CART, and RIPPER algorithms as base classifiers. They also examined bridging incompatible data sets from different companies and pruning base classifiers. Results indicate high cost savings and better efficiency on credit card transactions.

3. METHODOLOGY & SYSTEM DESIGN

The Agile methodology was adopted as the software development approach because building the software was incremental and iterative in nature. The development timeline was divided into four short time boxes, as listed below.

1. Data preprocessing module: Data contained in bank statements is scraped and parsed. In this phase, the classes to be used are also chosen.
2. Data representation module: After undergoing the data preprocessing module, data is normalized and represented in the database as models for classification and categorization.
3. Data categorization module: This phase takes in the models and uses a decision tree algorithm of "IF-THEN" rules to categorize a customer's expenses.
4. Presentation module: It presents the categorized data to the customer using attractive charts depicting the customer's expenses.

The requirements were gathered into functional and non-functional requirements. Functional requirements are the main features and requirements that generally make up the system, including input and output. The system allows users to upload their account statements. The supported document types allowed are *.xls* and *.csv.* All statements and transactions will be represented using a unique Universal Unique Identifier (UUID) hash, a unique 128-bit number used to identify information in computer systems. Every transaction must be associated to a statement in the database as a where the *statement_id* is a foreign key to a statement. The system must take in a transaction and categorize it to a class, e.g. Airtime, ATM Withdrawal, Online/POS, Income. The system must show the user a total of all their income and expenses in one statement. The system must be developed in the form of an API pattern making it easy for other programmers to use. The system must show users intuitive charts of their categorized transactions.

Non-functional requirements are features that are not explicitly visible in the development application, but are expected to enhance the user experience of the application. Availability and reliability are the non-functional requirements of the application.

The software packages used for development of the system were.

- Python, Django, Graphene, Numpy
- GraphQL, SQLite
- VueJS, ChartJS

Figure 1 shows the context diagram, defining and clarifying the boundaries of the software system. It shows information flow between the system and external entities.

The system is divided into three main parts, the user facing interface to upload a statement, the statement parsing and formatting API where data is prepared for analysis and finally, the analyzer API that categorizes statements and presents them to users.

The user is responsible for providing the dataset in *.xls* for analysis. The statement parsing API is responsible for scraping the file, cleaning up and data and preparing it for saving to the database for analysis. The analysis engine takes in the parsed data and creates models for it on the database, classifying it using a decision tree using selected variables in the dataset.

The architectural pattern adopted is a component/service-oriented architecture (Figure 2). Using a modular approach makes it easy to separate concerns of tasks. This makes sure each module adheres to the *Single Responsibility* principle in software engineering [23, 24].

This service-oriented architecture implements the following procedure:

1. The user uploads the statement file (PDF/XLS)
2. The browser processes the statement and sends it to the backend server for parsing
3. The backend parses and processes data in the statement classifying each expense record into spending categories using the remarks of each transaction
4. The backend sends back summarised and processed JavaScript Object Notation data of the entire statement to the browser(client)
5. The browser then illustrates and presents the data to the user in a simpler form using charts and graphs

The main elements used in the system architecture were the Statement Parsing/Preprocessing Module, the Database and the Categorization and Classifying Module.

The statement parsing module retrieves the statement file from the user, gathers required data from it after parsing, then prepares it for insertion into the database. All necessary dataset models are stored in the database. The dataset is then analyzed by the classification module. The classification module uses a decision tree approach to classify and assign a label to transactions on the database based on set rules.

The application was created using the Model–View–Controller (MVC) design pattern illustrated in Figure 3.

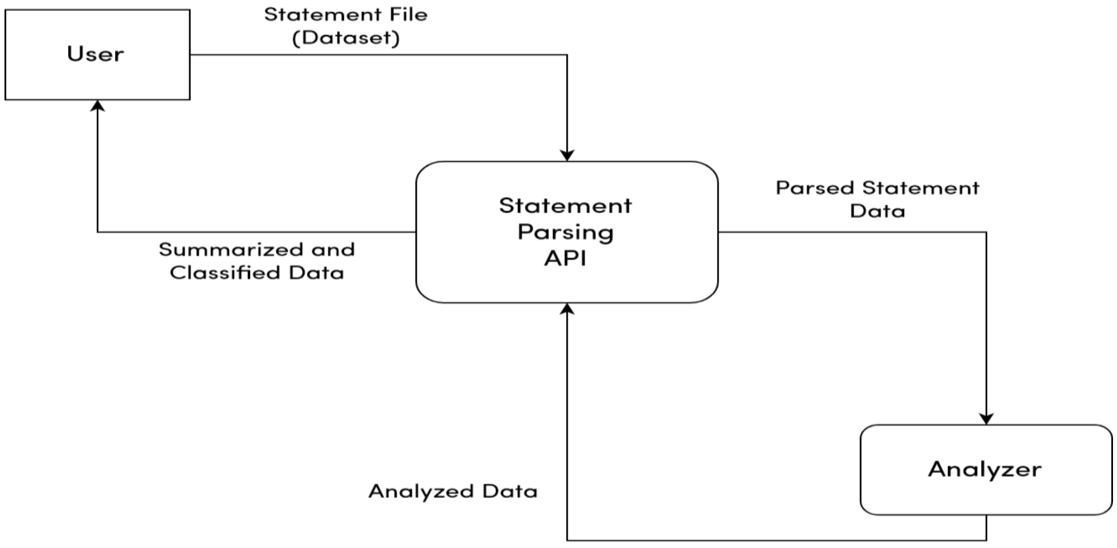


Figure 1: System Context Diagram

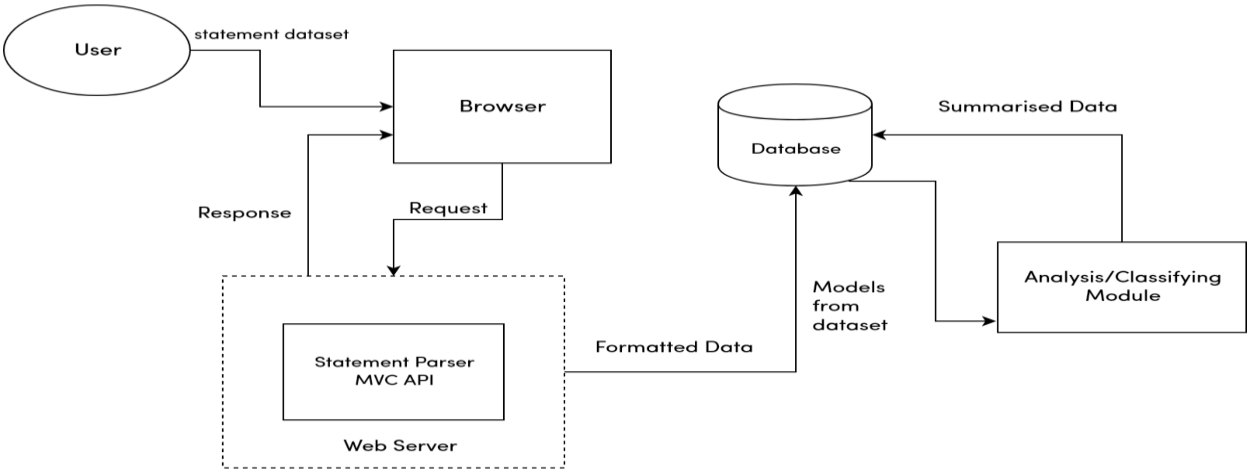


Figure 2: Component-based/service-oriented architecture

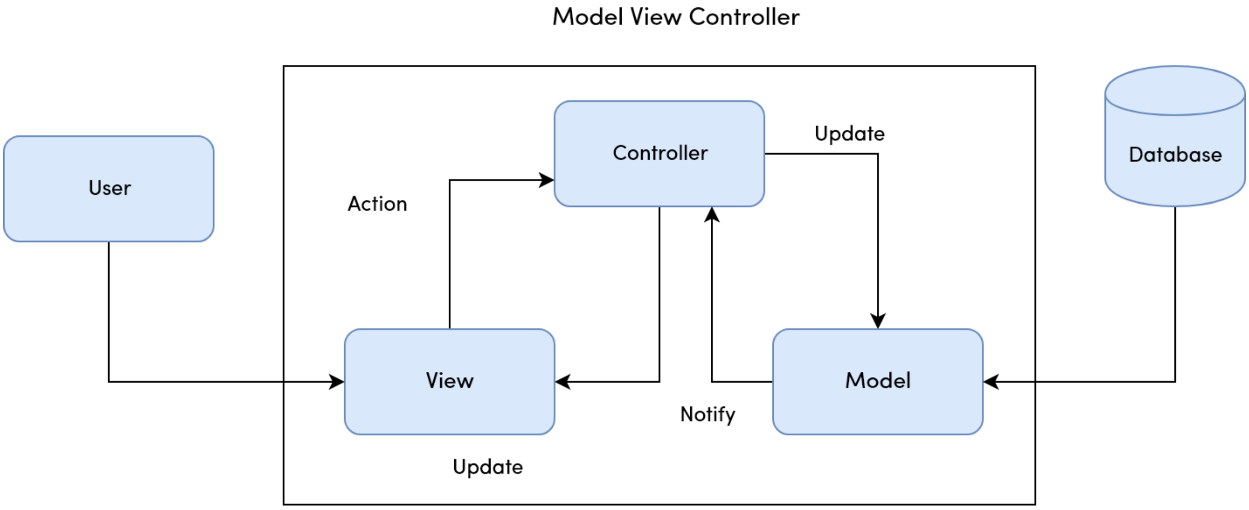


Figure 3: MVC design pattern

3.1 Design process

The initial dataset (bank statement in .xls format) file is read in by a FileReader which parses and makes sure the file is well formatted before it's sent to the parser by verifying its file type. The file is read in and sent to the statement parser which extracts and scrapes out necessary data. This is done on the frontend right on the browser. The reason is to avoid loading the server with too much work and allow it focus on classification alone. The parsed data from the statement file is then sent to the server represented as JavaScript Object Notation (JSON) saving the users' session to a browser cookie with a unique hash gotten from the server. The server receives the data and represents the data as models, passes the data through the data classification module illustrated with a decision tree in Figure 4. Once each transaction in the statement has been successfully classified and categorized, the classified transactions are then sent back to the server.

The decision tree in Fig. 4 is also used in getting the sum of the user's total income by using the rules above. In this case, a *total_sum* variable adds up all transactions that pass the check of being credit transactions and vice-versa for debit transactions.

3.2 Implementation techniques

The techniques used for implementation are explained in Table 1.

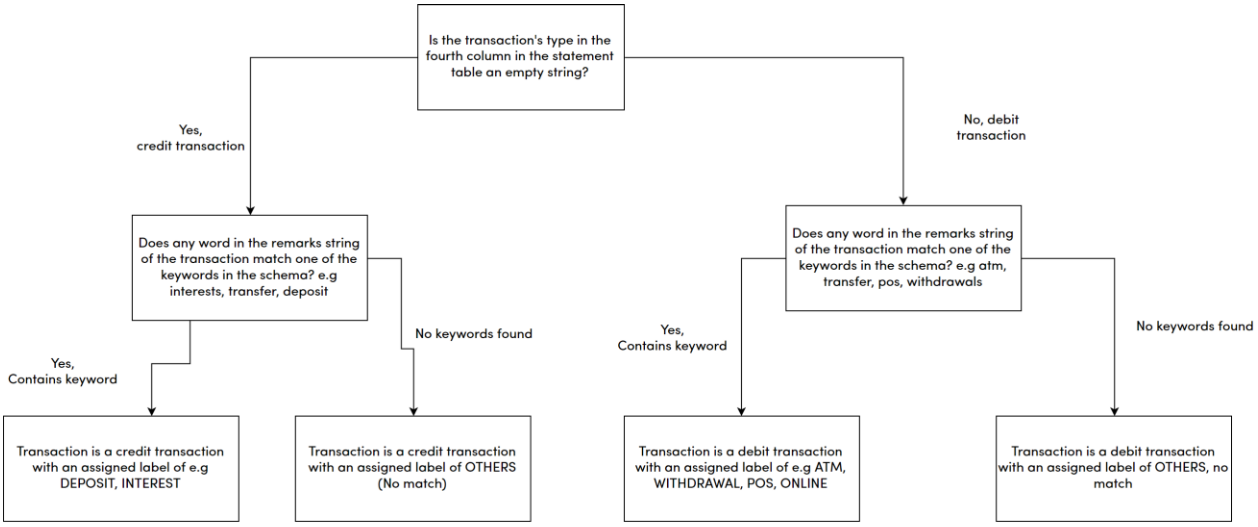

Is the transaction's type in the fourth column in the statement table an empty string?
Yes, credit transaction
No, debit transaction
Does any word in the remarks string of the transaction match one of the keywords in the schema? e.g interests, transfer, deposit
Does any word in the remarks string of the transaction match one of the keywords in the schema? e.g atm, transfer, pos, withdrawals
Yes, Contains keyword
No keywords found
Yes, Contains keyword
No keywords found
Transaction is a credit transaction with an assigned label of e.g DEPOSIT, INTEREST
Transaction is a credit transaction with an assigned label of OTHERS (No match)
Transaction is a debit transaction with an assigned label of e.g ATM, WITHDRAWAL, POS, ONLINE
Transaction is a debit transaction with an assigned label of OTHERS, no match

Figure 4: Data classification module

Table 1: Implementation Techniques

Technique	**Action**
Normalization	The text contained in the statement is prepared for processing. This is done by scraping and tokenizing according to the presence of specific variables in the statement file.
Trimming	Whitespace and undesired characters in the dataset are removed. This is to circumvent outliers and errors on the server.
Date Parsing	In most bank statements, transactions also have dates which are only represented in text. This must be parsed to well-represented *DateTime* object to be sent to the server. This step is done using a *dateparser* python module that takes in date and returns its datetime equivalent.
Classification	This is done using the decision tree in Fig. 4 and simply resenting the rules using IF-THEN clauses in the program.

There are several classes in the application as shown in Fig. 5.

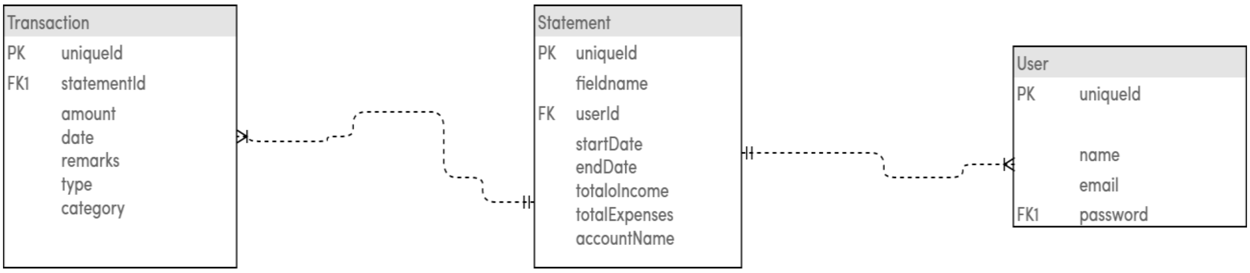


Figure 5: UML diagram

4. SYSTEM IMPLEMENTATION

Implementation featured three stages - coding, testing and deployment.

In coding, abstractions performed during analysis and design were physically implemented using a suitable programming language. The programming languages used were:

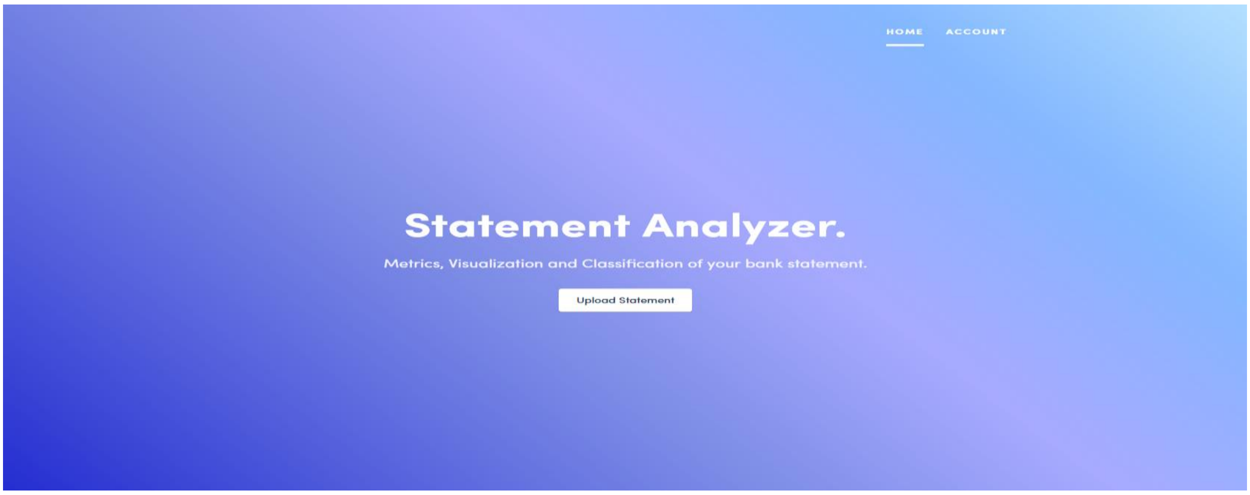
1. Python (used to develop the server-side backend of the application, which implements the classification and parsing modules).
2. JavaScript (used on the client side for handling event-driven actions in the browser).
3. HTML/CSS (used for writing markup and styling the user interface of the application).
4. Django (used to implement models and server-side logic).

SQLite was the database used because of its better performance, portability and reliability.

The heroku SAAS service was used to deploy the application. It provides fast deployment tools and version control.

5. RESULTS

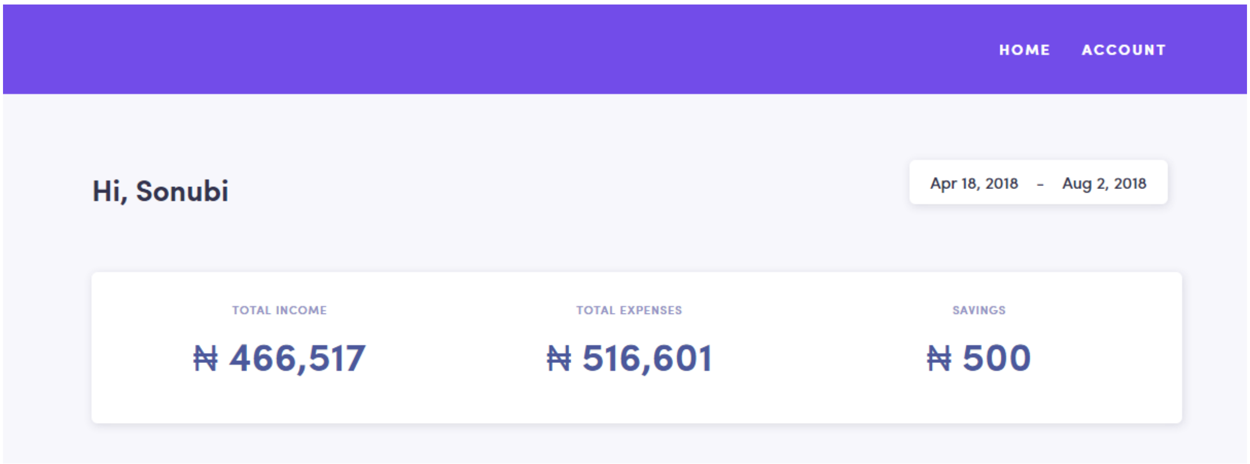
The built program uses data mining methods to classify, run statistical analysis on bank statements, visualise, generate and summarise reports for bank users in Nigeria. This enables users to understand the nature and essence of their financial transactions better, faster and clearer. The user interfaces (screenshots) of the app are depicted in Figures 6a and b. Fig 6a is the first interface the user encounters when he opens the app and wants to upload a file. Fig 6b shows the interface of the analyser API that categorizes statements and presents them to users. User's transaction history is summarized in Fig. 6c (a line graph illustrating the user's transaction history for the last 30 days in the statement) and visualized in Fig. 6d. (a chart showing the categorized transactions).



Statement Analyzer.

Metrics, Visualization and Classification of your bank statement.

Figure 6a: User Interface (statement analyzer)



Hi, Sonubi

Apr 18, 2018 - Aug 2, 2018

TOTAL INCOME ₦ 466,517

TOTAL EXPENSES ₦ 516,601

SAVINGS ₦ 500

Figure 6b: User Interface (classification)

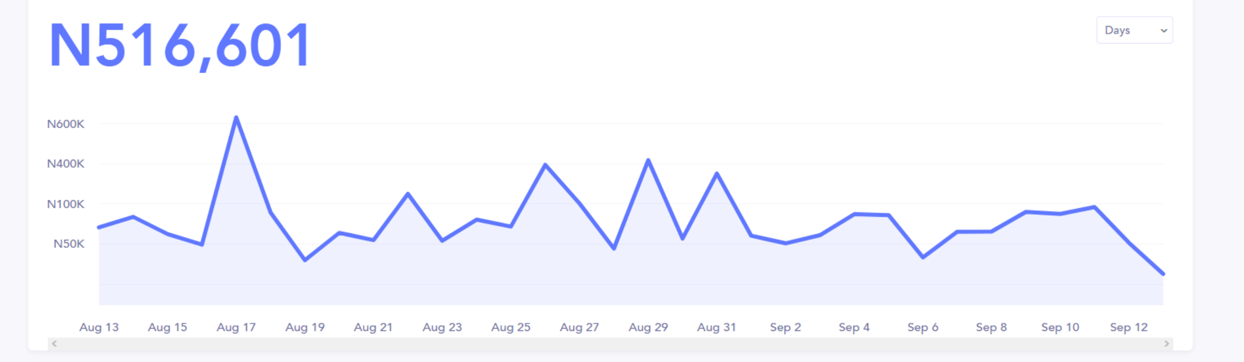


Figure 6c: User Interface (summarisation)

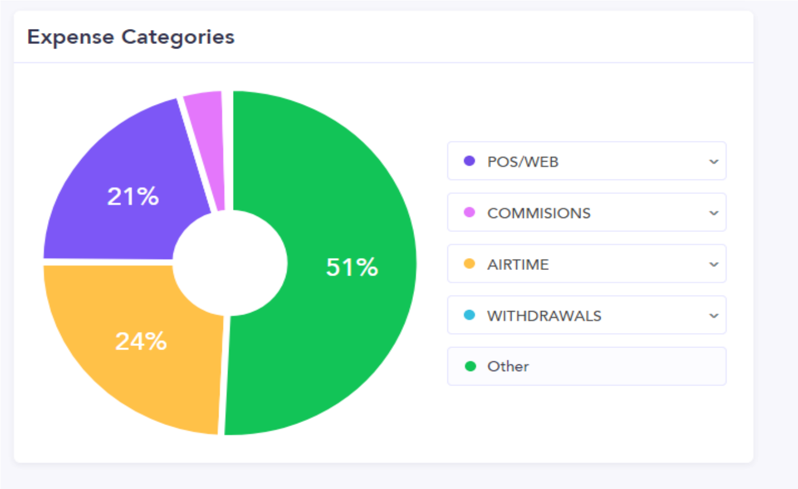


Figure 6d: User Interface (visualization)

6. CONSTRAINTS

Out of all the bank statements investigated, GTBank was found to have the most feasible dataset and file formats. The bank also provides support for exporting account statements in different file formats from its online banking platform.

GTBank's data source was hence used for this study. Other banks either had their statements in obscure file formats, or a formatting that is very intuitive and predictable.

The authors for now were unable to handle bank statements in *.pdf* format and opted for a spreadsheet *.xls* file type instead. They intend to continue research on possible ways of processing this file format.

7. RECOMMENDATIONS & CONCLUSION

The Nigerian financial industry needs to create easily accessible and open banking APIs for knowledgeable and skilled developers to access and create products from them. Banks need to deploy data mining extensively in their processes, considering that banks have huge financial data on millions of customers. This avoids the volatility of having to rely on the structure of a file to present financial data to customers.

There is currently an *Open Banking Nigeria API* in the works in Nigeria right now [25]. It enables financial institutions in Nigeria to design API endpoints that can then be accessed by API users to develop mobile and web apps for their customers. However, this opportunity is accessible only to banks and few financial firms for now.

8. REFERENCES

[1] Central Bank of Nigeria. "Publications" www.cbn.gov.ng, Accessed on October 3, 2019.

[2] Central Bank of Nigeria. "Nigeria Payments System Vision 2020" www.cbn.gov.ng/icps2013/papers/NIGERIA_PAYMENTS_SYSTEM_VISION_2020%5Bv2%5D.pdf, Accessed on October 4, 2019.

[3] Gates, B, and Gates, M. "Digitizing Government Payments in Nigeria" https://docs.gatesfoundation.org/documents/digitizing%20government%20payments%20in%20nigeria.pdf, Accessed on October 4, 2019.

[4] Central Bank of Nigeria. "Central Bank of Nigeria: Policy Measures" https://www.cbn.gov.ng/MonetaryPolicy/policy.asp, Accessed on October 3, 2019.

[5] ACM SIGKDD. "Data Mining Curriculum: A Proposal" https://www.kdd.org/curriculum/index.html, Accessed on September 10, 2019.

[6] SAS team. "Data Mining: What it is and why it matters" https://www.sas.com/en_us/insights/analytics/data-mining.html, Accessed on September 10, 2019.

[7] Techopedia. "Data Mining" https://www.techopedia.com/definition/1181/data-mining, Accessed on September 10, 2019.

[8] Kotu, V. and Deshpande B. *Data Science: Concepts and Practice* (Second Edition), Morgan Kaufmann, Burlington, Massachusetts, 2018.

[9] Han, J., Kamber, M. and Pei, J. *Data Mining. Concepts and Techniques* (3rd Edition), Morgan Kauffman Publishers, Waltham, MA, USA, 2012.

[10] Cawley, K. "When To Use Supervised And Unsupervised Data Mining" https://cloudtweaks.

com/2014/09/supervised-unsupervised-data-mining/, Accessed on October 11, 2019.

[11] Kesavaraj, G. and Sukumaran, S. "A study on classification techniques in data mining", *Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Tiruchengode, Tamil Nadu, India, July 4-6, 2013, pp.1-7.

[12] Altman, E. I. "Financials Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy", *The Journal of Finance*, Vol. 23, Number 4, 1968, pp 589-609.

[13] Ohlson, J.A. "Financial Ratios and the Probabilistic Prediction of Bankruptcy", *Journal of Accounting Research*, Vol. 18, Number 1, 1980, pp 109-131.

[14] Zmijewski, M. E. "Methodological issues related to the estimation of financial distress prediction models", *Journal of Accounting Research*, Vol. 22, 1984, pp 59-82.

[15] Lin, F.Y. and McClean, S. "A data mining approach to the prediction of corporate failure", *Knowledge-Based Systems*, Vol. 14., 2001, pp 189-195.

[16] Spathis, C. "Detecting false financial statements using published data: some evidence from Greece", *Managerial Auditing Journal*, Vol. 17, Number 4, 2002, pp 179-191.

[17] Huang, Z., Chen, H., Hsu, C.J., Chen, W.H. and Wu, S. "Credit Rating Analysis with Support Vector Machines and Neural Networks: a Market Comparative Study", *Decision Support Systems*, Vol. 37, Number 4, 2004, pp 543-558.

[18] Ghosh, S. and Douglas, L. R.. "Credit card fraud detection with a neural-network." Proceedings of the *Twenty-Seventh Hawaii International Conference on System Sciences*, Wailea, Hawaii, USA, January 4-7, 1994, pp. 621-630.

[19] Rosset, S., Murad, U., Neumann, E., Idan, Y. & Pinkas, G. "Discovery of Fraud Rules for Telecommunications - Challenges and Solutions", Proceedings of the *Fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, California, USA, August 15-18, 1999, pp. 409-413.

[20] Bonchi, F., Giannotti, F., Mainetto, G., Pedreschi, D. "Using Data Mining Techniques in Fiscal Fraud Detection", DataWarehousing and Knowledge Discovery. Proceedings of the *First International Conference, DaWaK'99*, Florence, Italy, August 30 – September 1, 1999, pp.369-376.

[21] Shao, H., Zhao, H. & Chang, G. (2002). "Applying Data Mining to Detect Fraud Behavior in Customs Declaration", Proceedings of *1st International Conference on Machine Learning and Cybernetics*, Beijing, China, November 4-5, 2002, pp. 1241-1244.

[22] Chan, P., Fan, W., Prodromidis, A. and Stolfo, S. "Distributed Data Mining in Credit Card Fraud Detection." *IEEE Intelligent Systems*, Vol. 14, Number 6, 1999, pp 67-74.

[23] Martin, R.C. "The Clean Code Blog" https://blog.cleancoder.com/uncle-bob/2014/05/08/SingleReponsibilityPrinciple.html, Accessed on September 20, 2019.

[24] SOLID. "The Single Responsibility Principle" https://code.tutsplus.com/tutorials/solid-part-1-the-single-responsibility-principle--net-36074, Accessed on September 20, 2019

[25] Open Banking Nigeria API. "Open Banking Nigeria" www.openbanking.ng, Accessed on October 22, 2019.