Integrating Apache Hadoop and Apache Spark for Comprehensive Data Analysis

Ashwin Muthuraman
Columbian College of Arts and Science
George Washington University
Washington D.C, USA
ashwin.muthuraman@gwu.edu

Bhoomika Nanjaraja
Columbian College of Arts and Science
George Washington University
Washington D.C, USA
bhoomika.najaraja@gwu.edu

Smitha Patil
Columbian College of Arts and Science
George Washington University
Washington D.C, USA
smitha.patil@gwu.edu

Kashyap Nimmagadda
Columbian College of Arts and Science
George Washington University
Washington D.C, USA
venkataanjanakashyap.nimmagadda@gwu.edu

Sanjayram Raja Srinivsan
Columbian College of Arts and Science
George Washington University
Washington D.C, USA
sanjayram.rajasrinivasan@gwu.edu

***Abstract*— This project demonstrates the effective integration of Apache Hadoop and Apache Spark for comprehensive data analysis, utilizing a large dataset from an Indian bank featuring over 1 million transactions from more than 800,000 customers. By storing data on Hadoop and processing it with Spark, we achieved efficient data handling and insightful analysis, leading to robust customer segmentation. Our methodology involved custom schema definitions for data integrity, exploratory data analysis (EDA) for initial insights, and advanced clustering techniques for segmentation. The project outcomes include a detailed system setup, a guide on integration, and visualizations representing diverse customer behaviors. This work underscores the potential of combining these technologies to enhance data-driven decision-making in banking.**

Keywords*— Apache Hadoop, Apache Spark, Data Integration, Customer Segmentation, Big Data Analytics, Financial Transactions, Cluster Analysis, EDA, RFM Analysis, K-Means Clustering

I. INTRODUCTION

In the contemporary data-centric landscape, the ability to process large volumes of data efficiently is crucial for deriving actionable insights, especially in sectors like banking where customer data is voluminous and varied. This project explores the integration of two pivotal Big Data technologies—Apache Hadoop and Apache Spark—to manage and analyze a substantial dataset from an Indian bank effectively. The dataset comprises over 1 million transactions and extensive demographic details from more than 800,000 customers, offering fertile ground for segmentation and behavior analysis.

The primary objective of this project is to illustrate the synergistic use of Hadoop for robust data storage and Spark for dynamic data processing. This approach not only streamlines data workflows but also enhances the analytical capabilities of financial institutions. By detailing the installation, configuration, and integration of Hadoop and Spark, this paper serves as a practical guide for similar big data endeavors. The deliverables include a configured system that performs detailed customer segmentation analyses, characterized by data-driven marketing strategies and enhanced customer service optimizations.

Through the course of this project, we engage in a rigorous Exploratory Data Analysis (EDA) to understand the underlying patterns within the data. We address challenges such as schema consistency, data quality, and transformation, and deploy advanced analytics like RFM (Recency, Frequency, Monetary) metrics calculation and K-Means clustering to discern distinct customer segments based on transactional behaviors. The insights derived from these analyses highlight the critical interplay between customer characteristics and their banking activities, offering a nuanced view of consumer bases that can drive targeted marketing and service provisions.

II. LITERATURE REVIEW

The advent of big data technologies has revolutionized many industries, with banking being one of the most impactful. Banks today generate a colossal amount of data, and the ability to efficiently process and extract meaningful insights from this data is crucial. Technologies such as

Apache Hadoop and Apache Spark have emerged as leaders in this space, providing robust frameworks for handling large-scale data across distributed environments. This literature review explores the integration of these technologies in a banking context, demonstrating their application in a customer segmentation project.

A. Apache Hadoop and HDFS

Apache Hadoop is foundational in the realm of big data, known for its robust handling of large data sets across distributed systems. Its core component, the Hadoop Distributed File System (HDFS), offers high throughput access to application data and is designed to span large clusters of commodity servers. By storing data across an extended network of machines, it ensures data availability and redundancy, which are critical in banking environments where data loss can have severe implications. Hadoop's framework is highly fault-tolerant, a key feature that allows financial institutions to maintain continuity even in the face of hardware failure.

B. Apache Spark

While Hadoop excels at data storage and batch processing, Apache Spark is lauded for its superior data processing capabilities. Spark facilitates in-memory computing, which significantly speeds up data processing tasks by minimizing disk I/O and network transfers. In customer segmentation, where iterative algorithms are common, Spark's fast processing speed allows for real-time analytics and machine learning, making it an invaluable tool for dynamic data analysis tasks in banking.

C. PySpark and PySpark ML

PySpark, the Python API for Spark, brings high-level abstraction and ease of use to Spark's powerful processing capabilities. For data scientists working within the banking sector, PySpark provides a gateway to implement machine learning algorithms on large datasets seamlessly. PySpark ML, a library within PySpark, further simplifies the application of machine learning models by offering pre-built algorithms and data transformation capabilities. This is crucial for conducting sophisticated analytics such as customer segmentation, where understanding customer behavior patterns can lead to targeted marketing and improved customer service.

D. Data Visualization with Matplotlib & Seaborn

Understanding data through visualization is a critical step in any data analysis process, especially in banking where decision-making needs to be supported by clear and interpretable insights. Matplotlib and Seaborn enhance the visualization capabilities within the Python ecosystem. While Matplotlib provides a wide range of basic plotting tools, Seaborn builds on these with a focus on statistical models and provides a more aesthetically pleasing and concise syntax. This combination is particularly effective for visualizing the results of customer segmentation, facilitating better storytelling through data.

E. NumPy & Pandas

Handling and transforming data in preparation for analysis is made efficient with libraries like NumPy and Pandas. NumPy provides support for large, multi-dimensional arrays and matrices which are often required in data analytics, along with a large collection of high-level mathematical functions to operate on these arrays. Pandas, on the other hand, offers data structures and operations for manipulating numerical tables and time series. This library is particularly useful in the banking sector for time-series analysis and for structured data operations that are commonplace in customer demographic analyses.

III. RESEARCH METHODOLOGY

This section outlines the systematic approach used to analyze a comprehensive dataset from Kaggle, aimed at segmenting bank customers based on their transactional data and demographics. The methodology is structured into key phases: data collection, preprocessing, exploratory data analysis (EDA), feature engineering, and clustering. Each phase plays a critical role in ensuring the data's readiness and the accuracy of our insights, culminating in effective customer segmentation through advanced clustering techniques. This streamlined approach aligns with best practices in data science, ensuring a robust analysis.

A. Data Collection

The dataset in this project consists of over 1 million transactions from upwards of 800,000 customers of an Indian bank. Featuring granular data on customer age, location, gender, account balance, and detailed transactional information, it sets the foundation for an in-depth customer segmentation analysis. This dataset was sourced from Kaggle, ensuring a rich and varied sample for robust analytical outcomes.

TransactionID	CustomerID	CustomerDOB	CustGender	CustLocation	CustAccountBalance	TransactionDate	TransactionTime	TransactionAmount (INR)
T1	C5841053	10/1/94	F	JAMSHEDPUR	17819.05	2/8/16	143207	25.0
T2	C2142763	4/4/57	M	JHAJJAR	2270.69	2/8/16	141858	27999.0
T3	C4417068	26/11/96	F	MUMBAI	17874.44	2/8/16	142712	459.0
T4	C5342380	14/9/73	F	MUMBAI	866503.21	2/8/16	142714	2060.0
T5	C9031234	24/3/88	F	NAVI MUMBAI	6714.43	2/8/16	181156	1762.5
T6	C1536588	8/10/72	F	ITANAGAR	53609.2	2/8/16	173940	676.0
T7	C7126560	26/1/92	F	MUMBAI	973.46	2/8/16	173806	566.0
T8	C1220223	27/1/82	M	MUMBAI	95075.54	2/8/16	170537	148.0
T9	C8536061	19/4/88	F	GURGAON	14906.96	2/8/16	192825	833.0
T10	C6638934	22/6/84	M	MUMBAI	4279.22	2/8/16	192446	289.11
T11	C5430833	22/7/82	M	MOHALI	48429.49	2/8/16	204133	259.0
T12	C6939838	7/7/88	M	GUNTUR	14613.46	2/8/16	205108	202.0
T13	C6339347	13/6/78	M	AHMEDABAD	32274.78	2/8/16	203834	12300.0
T14	C8327851	5/1/92	F	THANE	59950.44	1/8/16	84706	50.0
T15	C7917151	24/3/78	M	PUNE	10100.84	1/8/16	82253	338.0
T16	C8334633	10/7/68	F	NEW DELHI	1283.12	1/8/16	125725	250.0
T17	C1376215	1/1/1800	M	MUMBAI	77495.15	1/8/16	124727	1423.11
T18	C8967349	16/7/89	M	MUMBAI	2177.85	1/8/16	124734	54.0
T19	C3732016	11/1/91	M	MUMBAI	32816.17	1/8/16	122135	315.0
T20	C8999019	24/6/85	M	PUNE	10643.5	1/8/16	152821	945.0

Fig. 1. Dataset

B. Data Preprocessing

1) Custom Schema Definition: Before loading the dataset into our analytical environment, we defined a custom schema. This critical step sets explicit data types for each field, ensuring proper handling of missing or malformed data, which is particularly important for date fields and numerical values that require specific formats for accurate analysis.

2) Data Loading: With the schema in place, we proceeded to load the dataset into a Data Frame, leveraging the capabilities of our processing tools to maintain consistency in data formatting. Special attention was given to date fields by applying a legacy time parser policy, ensuring that all temporal data adhered to a uniform standard.

```
#Custom Schema for hundred thousand records
custom_schema_full = StructType([
    StructField("TransactionID", StringType(), True),
    StructField("CustomerID", StringType(), True),
    StructField("CustomerDOB", DateType(), True),
    StructField("CustGender", StringType(), True),
    StructField("CustLocation", StringType(), True),
    StructField("CustAccountBalance", DoubleType(), True),
    StructField("TransactionDate", DateType(), True),
    StructField("TransactionTime", IntegerType(), True),
    StructField("TransactionAmount (INR)", DoubleType(), True),
])

df_full = spark.read.format("csv") \
    .option("header", "true") \
    .option("dateFormat", "d/M/yy") \
    .schema(custom_schema_full) \
    .load("hdfs://localhost:9000/project/full_data.csv")
```

Fig. 2. Custom Schema & Loading

3) Column Optimization: Once loaded, we optimized the dataset by renaming columns. This was not merely an aesthetic adjustment but a strategic move to enhance clarity and simplify data access throughout the subsequent stages of analysis.

4) Data Scrubbing: The dataset was thoroughly scrubbed to ensure data quality. This involved the removal of invalid or null entries and the enforcement of logical criteria, such as ensuring that transaction times and amounts were positive values. This step was crucial to maintain the integrity of the dataset for accurate and meaningful analysis.

C. Exploratory Data Analysis (EDA)

1) Top Customer Locations by Transaction Volume (Location Analysis): In this analysis, we visualized the distribution of transactions among the top 20 customer locations. The bar chart indicates that Mumbai, Bangalore, and New Delhi are the leading cities in terms of transaction volume, suggesting a higher concentration of banking activity in these metropolitan areas. These insights can guide localized marketing strategies and service delivery optimizations.

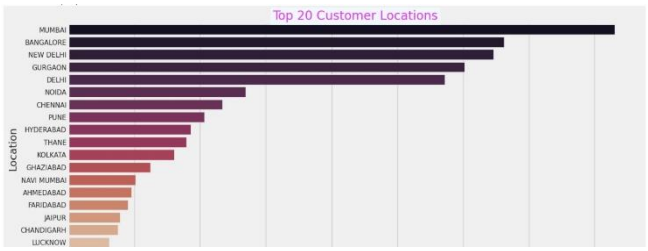

Fig. 3. Top Customer Locations by Transaction Volume

2) Frequency Analysis: The bar chart illustrates the number of transactions per customer within the dataset. It shows a dominant number of customers with a single transaction, while a substantially smaller group appears to engage in three transactions. The precipitous drop in counts from one transaction to two and beyond suggests a trend where most customers interact with the bank on an infrequent basis. This informs targeted engagement strategies to increase transaction frequency among customers.

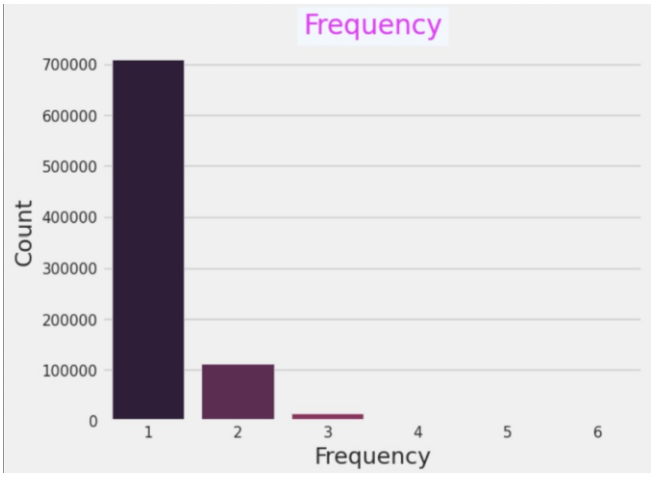

Fig. 4. Number of transactions per customer

3) Missing Values Analysis: In this chart, we see a visualization of missing data across three key customer attributes: gender, location, and account balance. The account balance column has the highest number of missing values, followed by gender and location. This indicates potential issues in data collection or entry, which could affect analyses that depend on these variables. Strategies to address these missing values might include data imputation, the use of median values for account balances, or exclusion of the affected records from certain analyses.

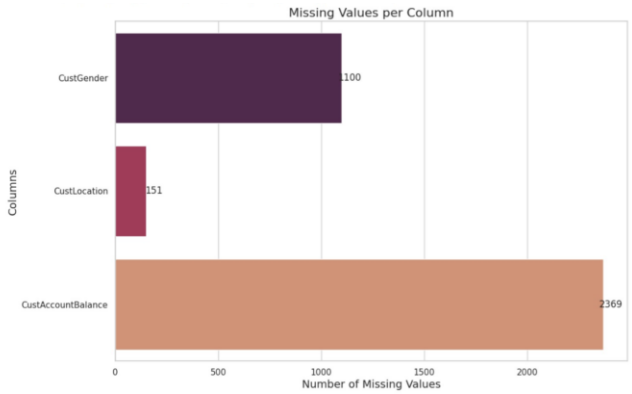

Fig. 5. Missing Values per Column

4) Correlation Analysis: The heatmap presents a visual correlation matrix assessing the relationship between several key variables: Frequency, CustAccountBalance, TransactionAmount (INR), CustomerAge, and Recency.

1) Frequency & Recency (0.78): There's a strong positive correlation between how frequently customers transact and the recency of their activity. This suggests that customers who transact more often tend to do so more recently, which could be indicative of regular banking habits or engagement with bank services.

2) Account Balance & Transaction Amount (0.12): There is a slight positive correlation indicating that customers with higher account balances tend to make larger transactions, though the relationship is not strong.

3) Customer Age & Recency (-0.13): There's a weak negative correlation here, hinting that younger customers may engage in transactions more recently compared to older customers, possibly reflecting more active financial behavior or different banking needs among younger demographics.

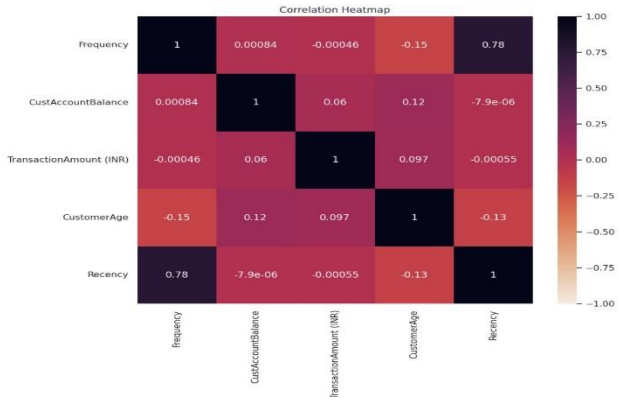

Fig. 6. Heat Map

5) Distribution of Customer Age: The histogram for "Distribution of Customer Age" suggests a younger customer base, with the largest frequency of customers falling in the 20-30 age bracket. This indicates the bank's demographic skew towards younger individuals, potentially reflecting the bank's product appeal to this age group or demographic trends in the population served.

6) Customer Gender Distribution: The pie chart for "Customer Gender" shows that approximately 27.7% of the customers are female, while the remaining 72.3% are male. This significant difference implies gender-specific banking behavior or could highlight an opportunity for the bank to increase engagement with female customers.

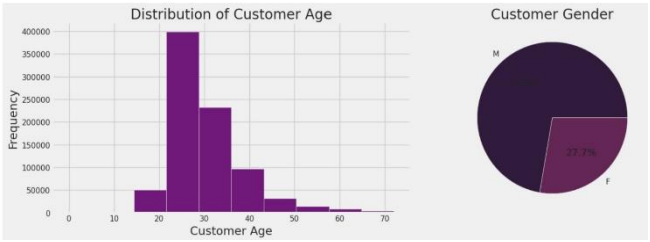

Fig. 6. Age & Gender distribution

7) Relationship Between Transaction Amount and Customer Account Balance: The scatter plot "Transaction Amount (INR) and CustAccountBalance" with overlaid frequency and recency points suggests that while there are customers with a range of account balances and transaction amounts, most transactions are clustered at the lower end of the spectrum. The size of the points representing frequency indicates that transactions are more frequent with smaller amounts. Similarly, the color gradation representing recency implies that more recent transactions tend to involve smaller amounts, which indicate regular small-scale banking activity or a predominance of low-value transactions within the customer base.

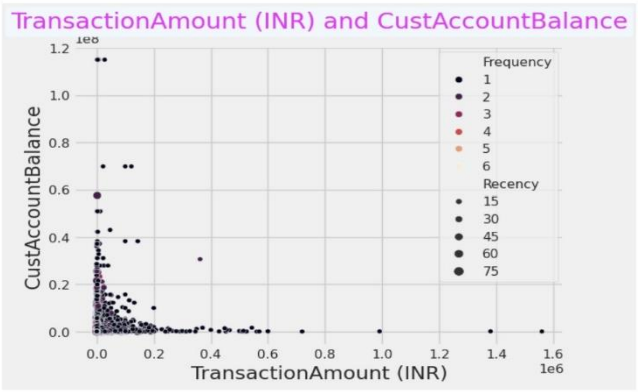

Fig. 7. Transaction Amount vs. Customer Account Balance

D. Feature Engineering

Feature engineering serves as an instrumental phase where raw data is transformed into informative features, enhancing the performance of machine learning models. This section details the refinement of RFM (Recency, Frequency, Monetary) metrics, crucial for our analysis of customer value based on transaction history.

1) RFM Metrics: The RFM model is a well-established approach for evaluating customer behavior by examining three critical dimensions: Recency (R) — the time since the last transaction, Frequency (F) — the number of transactions over a period, and Monetary (M) — the total spends of a customer.

2) Normalization of the Recency Metric: Prior to normalization, the Recency metric exhibited a right-skewed distribution. This skewness could potentially bias prediction models, especially those that presuppose feature normality.

1) Data Transformation Technique: To mitigate this skewness, a Box-Cox transformation was applied to the Recency metric. This transformation is designed to stabilize variance and promote the normality of the distribution, as evidenced by the symmetry in the post-transformation histogram and the alignment with the red line in the probability plot, indicative of normal distribution conformity.

2) Post-normalization Distribution: Following the Box-Cox transformation, the Recency feature achieved a more normalized distribution, making it a more potent predictor in clustering algorithms. The transformation's impact was critical for the model's precision and the integrity of the customer segmentation results.

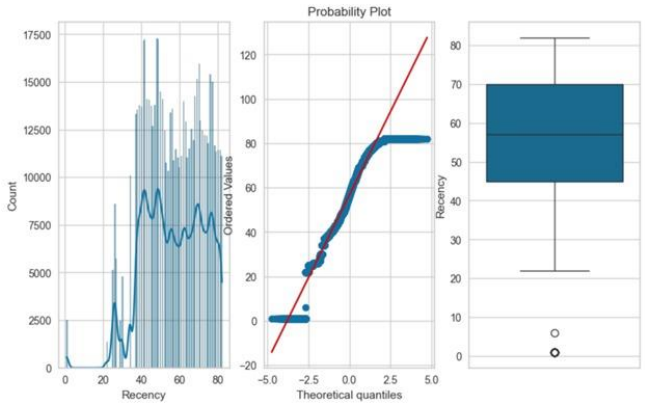

Fig. 8. Before Normalization (Recency)

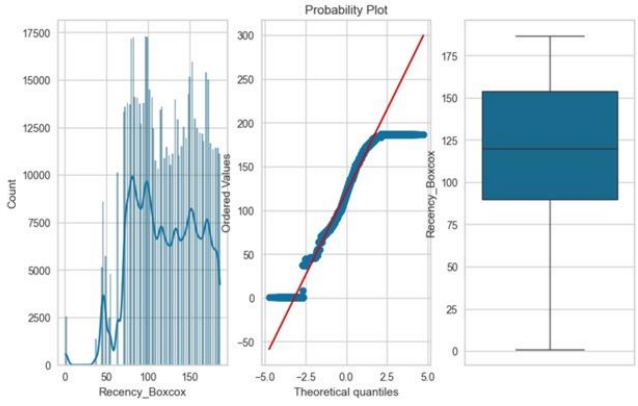

Fig. 9. After Normalization (Recency)

3) Frequency Value Distribution Prior to Normalization: Before normalization: The Frequency distribution exhibits a significant right skew, indicating a concentration of customers with a low number of transactions. The associated probability plot deviates from the theoretical quantiles, confirming the non-normality of the distribution.

1) Transformation for Normalization: A logarithmic transformation is applied to the Frequency metric to address non-normality, a common approach for skewed distributions. The transformation aims to diminish the influence of outlier values and reduce skewness.

2) Post-normalization Distribution: Following normalization, the transformed Frequency metric demonstrates a distribution that approaches normality. The histogram shows a decrease in skewness, and the probability plot aligns more closely with the theoretical quantiles, suggesting improved symmetry. The boxplot indicates a more uniform spread, underscoring the effectiveness of the logarithmic transformation.

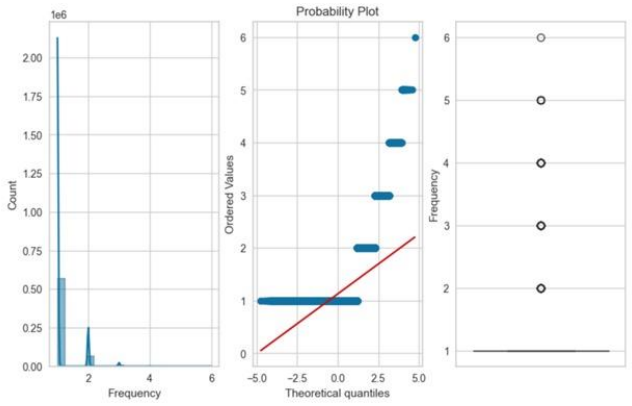

Fig. 10. Before Normalization (Frequency)

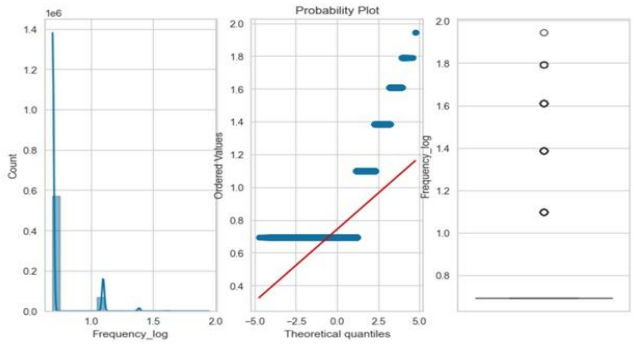

Fig. 11. After Normalization (Frequency)

4) Monetary Value Distribution Prior to Normalization: The initial examination of the Monetary value distribution reveals a pronounced right skew, indicating that most customers engage in lower-value transactions, with a few outliers conducting significantly higher-value transactions. The probability plot diverges substantially from the red line representing the theoretical quantiles, further emphasizing the non-normality of the original distribution.

1) Transformation for Normalization: A logarithmic transformation is applied to Monetary values, a conventional technique to correct positive skewness in financial data. This transformation compresses the range of the data, diminishing the effect of extreme values.

2) After applying the logarithmic transformation, the distribution of Monetary values exhibits a considerable reduction in skewness, as depicted in the histogram. The probability plot's alignment with the theoretical quantiles dramatically improves, reflecting the transformation's effectiveness. The boxplot confirms the diminishing effect on extreme outliers, presenting a more compact interquartile range.

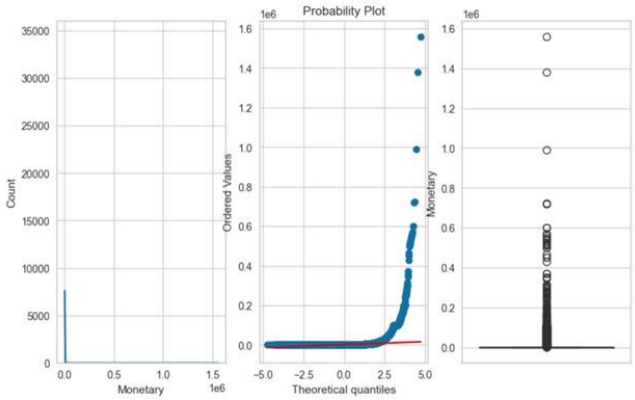

Fig. 12. Before Normalization (Monetary)

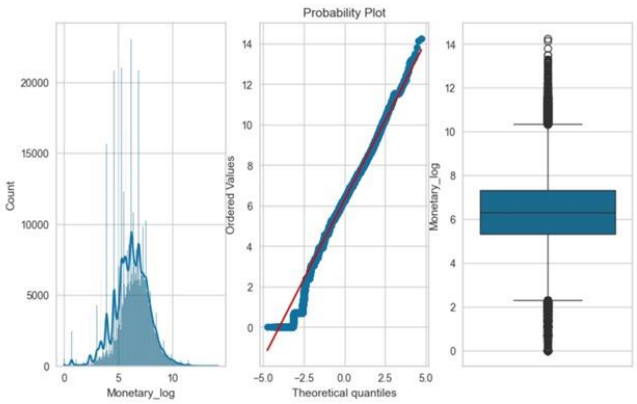

Fig. 13. After Normalization (Monetary)

IV. RESULTS

A. Determination of Optimal Cluster Number: The Elbow Method graph provided a visualization to determine the optimal number of clusters. The distortion score—representing the sum of squared distances from each point to its assigned center—demonstrates a clear elbow at k=4. This indicates that increasing the number of clusters beyond four results in diminishing returns regarding variance explained, thus we selected four as the optimal cluster count for our K-Means algorithm.

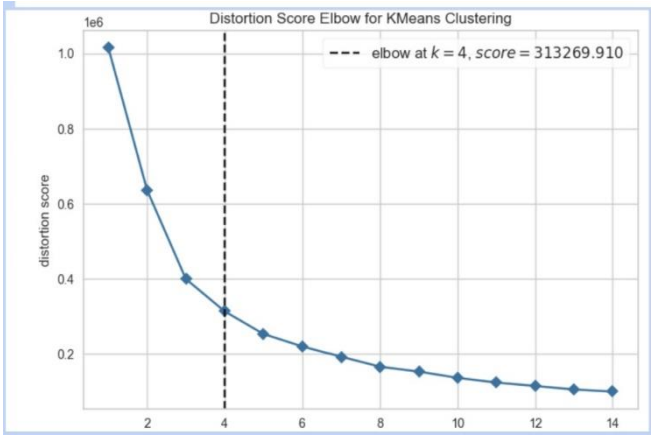

Fig. 14. Elbow Method graph

B. Clustering Results and Interpretation: The customer segmentation graph illustrates the distribution of customers in the four identified clusters, plotted against Recency and Monetary logarithmically transformed metrics. Each cluster reveals different customer behaviors.

1) Cluster 1: Characterized by recent interactions and lower monetary values, possibly indicating new or casual customers.

2) Cluster 2: Exhibiting less recent activity and moderate monetary values, suggestive of occasionally active customers.

3) Cluster 3: Shows infrequent, but high-value transactions, which might correspond to high-value, yet less engaged customers.

4) Cluster 4: Comprises customers with older transactions and lower monetary contributions, which might be dormant accounts or customers with decreased engagement.

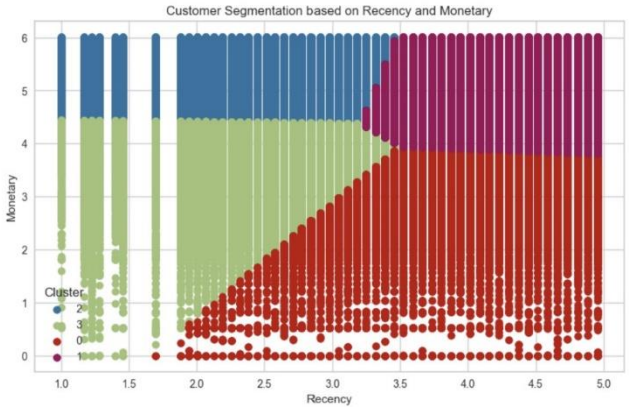

Fig. 15. Customer segmentation graph

C. Summary Statistics of Customer Clusters: Post clustering, summary statistics were compiled to characterize each cluster, providing a quantitative overview of the segments derived from our K-Means analysis. Each cluster is evaluated on average Recency, Monetary value, and Frequency of transactions, alongside the count of customers belonging to each segment.

1) Cluster 1: With an average Recency of 71.20 days and a Monetary value of 1830.88, along with the largest customer count of 183,670, this cluster represents regular customers with moderate transaction values.

2) Cluster 3: Exhibits a lower average Recency of 44.99 days and a higher Monetary value of 162.71. With 112,860 customers, this group consists of recently active customers with lower transaction values.

3) Cluster 2: Shows an average Recency similar to Cluster 3 at 45.68 days but with the highest Monetary value of 1066.04. The 138,889 customers in this cluster represent a segment of premium customers with high transaction amounts but moderate activity.

4) Cluster 0: Presents an average Recency of 68.49 days and the lowest Monetary value of 68.75, encompassing 72,107 customers. This indicates infrequent customers with minimal transaction values.

Cluster	avg(Recency)	avg(Monetary)	avg(Frequency)	count(Cluster)
1	71.19960254804813	830.8760374040388	1.0	183670
3	44.9878344852029	162.7025667198298	1.0	112860
2	45.67677065858347	1066.036610458711	1.0	138889
0	68.49409904724922	68.7550032590456	1.0	72107

Fig. 16. Summary Statistics of Customer Clusters

V. CONCLUSION AND FUTURE WORK

This project has effectively leveraged Hadoop and Spark to segment bank customers, yielding valuable behavioral insights through RFM analysis and K-Means clustering. Our findings provide a springboard for targeted banking strategies.

We plan to introduce real-time transaction processing and deploy advanced predictive models, enhancing the system's responsiveness and analytical depth to meet evolving data and business requirements.

VI. REFERENCES

[1] Apache Spark Documentation - https://spark.apache.org/documentation.html

[2] Apache Hadoop Documentation - https://hadoop.apache.org/docs/current

[3] PySpark Documentation – https://spark.apache.org/docs/latest/api/python/index.html

[4] RFM Analysis for Customer Segmentation – https://towardsdatascience.com/implementing-customer-segmentation-using-rfm-analysis-with-pyspark-3aed363f1d53

[5] K-Means Clustering with PySpark – https://medium.com/data-and-beyond/customer-segmentation-using-k-means-clustering-with-pyspark-unveiling-insights-for-business-8c729f110fab

[6] PySpark MLib library – https://spark.apache.org/docs/1.6.1/api/python/pyspark.ml.html