FINANCE AND BANKING

CREDIT SCORING WITH ALTERNATE DATA

AD23531-BIG DATA ARCHITECTURE

A PROJECT REPORT

SUBMITTED BY:

Bharkavi N 2116231801023

Gayathri R 2116231801039

Hemalatha L 2116231801055

Department of Artificial Intelligence and Data Science

Rajalakshmi Engineerimg College, Thandalam

Oct 2025

BONAFIDE CERTIFICATE

Certified that this Report titled “**FINANCE AND BANKING CREDIT SCORING WITH ALTERNATE DATA”**is the bonafide work of “**BHARKAVI N(2116231801023) ,GAYATHRI R(2116131801039) and HEMALATHA L(2116231801055)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. S. SURESHKUMAR, Ph.D.,

Professor and Head

Department of Artificial Intelligence and Data Science,

Submitted to Project Viva-Voce Examination held on

____

Internal Examiner **External Examiner**

DEPARTMENT VISION

To promote highly Ethical and Innovative Computer Professionals through excellence in teaching, training and research.

DEPARTMENT MISSION

• To produce globally competent professionals, motivated to learn the emerging technologies and to be innovative in solving real world problems.

• To promote research activities amongst the students and the members of faculty that could benefit the society.

• To impart moral and ethical values in their profession.

PROGRAMME EDUCATIONAL OBJECTIVES(PEO’S)

PEO 1: To equip students with essential background in computer science, basic electronics and applied mathematics.

PEO 2: To prepare students with fundamental knowledge in programming languages, and tools and enable them to develop applications.

PEO 3: To encourage the research abilities and innovative project development in the field of AI, ML, DL, networking, security, web development, Data Science and also emerging technologies for the cause of social benefit.

PEO 4: To develop professionally ethical individuals enhanced with analytical skills, communication skills and organizing ability to meet industry requirements.

PROGRAMME OUTCOMES (POs)

PO 1: Engineering knowledge: Apply the knowledge of Mathematics, Science, Engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO 2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO 4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

A graduate of the Artificial Intelligence and Machine Learning Program will demonstrate

PSO 1: Foundation Skills: Ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, web design, AI, machine learning, deep learning, data science, and networking for efficient design of computer-based systems of varying complexity. Familiarity v and practical competence with a broad range of programming language, tools and open-source platforms.

PSO 2: Problem-Solving Skills: Ability to apply mathematical methodologies to solve computational task, model real world problem using appropriate AI and

ML algorithms. To understand the standard practices and strategies in project development, using open-ended programming environments to deliver a quality product.

PSO 3: Successful Progression: Ability to apply knowledge in various domains to identify research gaps and to provide solution to new ideas, inculcate passion towards higher studies, creating innovative career paths to be an entrepreneur and evolve as an ethically social responsible AI and ML professional.

COURSE OBJECTIVE

- To identify and formulate real-world problems that can be solved using Artificial Intelligence and Machine Learning techniques.

- To apply theoretical and practical knowledge of AI/ML for designing innovative, data-driven solutions.

- To integrate various tools, frameworks, and algorithms to develop, test, and validate AI/ML models.

- To demonstrate effective teamwork, project management, and communication skills through collaborative project execution.

- To instill awareness of ethical, societal, and environmental considerations in the design and deployment of intelligent systems.

COURSE OUTCOME

- **CO 1:** Analyze and define a real-world problem by identifying key challenges, project requirements and constraints.

- **CO 2:** Conduct a thorough literature review to evaluate existing solutions, identify research gaps and formulate research questions.

- **CO 3:** Develop a detailed project plan by defining objectives, setting timelines, and identifying key deliverables to guide the implementation process.

- **CO 4:** Design and implement a prototype or initial model based on the proposed solution framework using appropriate AI tools and technologies.

- **CO 5 :** Demonstrate teamwork, communication, and project management skills by preparing and presenting a well-structured project proposal and initial implementation results

CO-PO/PSO Mapping

CO \ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	3	3	3	3	2	2	2	3	2	3	3	3	3	3
CO2	3	3	3	3	3	2	-	-	2	2	2	3	3	2	2
CO3	3	3	3	2	3	1	1	2	3	3	3	3	3	3	3
CO4	3	3	3	3	3	2	1	2	3	2	2	3	3	3	3
CO5	1	1	1	1	1	-	-	-	3	3	3	3	1	-	2

Note: Correlation levels 1, 2 or 3 are as defined below:
1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)
No correlation: “-”

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S.MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N. MURUGESAN, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. J M GNANASEKAR, Ph.D.,** Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, Mr. K. VIJAY B.Tech., M.E. (SG), Assistant Professor, Department of Artificial Intelligence and Machine Learning. Rajalakshmi Engineering College for her valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, Dr. S. SURESHKUMAR, Ph.D., Department of Artificial Intelligence and Data Science for his useful tips during our review to build our project.

BHARKAVI N	**GAYATHRI R**	**HEMALATHA L**
(2116231801023)	(2116231801039)	(2116231801055)

ABSTRACT

Credit scoring is a critical process in the finance and banking sector, enabling institutions to evaluate the creditworthiness of customers and mitigate financial risks. Traditional methods often struggle with the increasing volume, variety, and velocity of financial data. In this project, we developed a Big Data pipeline using Databricks Free Edition, integrating the Databricks File System for storage, Spark for large-scale data processing, and the SQL Editor (HiveQL) for query-based analytics.

The dataset was ingested using Databrick data upload wizard, followed by data preprocessing, exploratory data analysis (EDA), and feature engineering using Python and SQL within the Databricks environment. A Random Forest classifier was implemented to predict customer creditworthiness and identify potential fraudulent transaction patterns. Multiple interactive dashboards were built to visualize trends and prediction outcomes, offering clear and actionable insights for financial decision-making.

This pipeline demonstrates the seamless integration of Big Data analytics and machine learning within Databricks, enabling efficient data handling, predictive modeling, and intelligent automation for modern banking and credit risk management.

Keywords – Credit Scoring, Big Data Analytics, Databricks, Apache Spark, HiveQL, Data Preprocessing, Exploratory Data Analysis (EDA), Feature Engineering, Random Forest Classifier, Fraud Detection, Machine Learning, Predictive Modeling, Financial Risk Management, Data Visualization, Cloud Computing

INDEX

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

In today's digital economy, the finance and banking sector generates vast amounts of data every second through credit card transactions, online payments, and digital lending platforms. Traditional credit scoring models, which rely on limited financial history and static data sources, often fail to capture the dynamic behavioral patterns of modern consumers. Managing and analyzing this high-volume, high-velocity data is essential for maintaining financial stability, assessing creditworthiness, and detecting fraudulent activities.

The integration of Big Data analytics and machine learning has become a powerful solution to address these challenges. Platforms like Databricks, combined with distributed computing through Spark, allow financial institutions to process and analyze large, diverse atasets in real time, extracting valuable insights efficiently.

1.2 Motivation

The motivation behind this project arises from the limitations of traditional credit scoring systems. Conventional methods are static, rely on historical financial data, and often overlook alternative behavioral data, leading to inaccurate credit assessments and delayed fraud detection. By leveraging Big Data technologies and machine learning, the project aims to create a dynamic, automated system capable of processing large-scale data efficiently and providing more accurate credit scoring and fraud detection.

1.3 Objectives

The primary objectives of this project are:

1. To build an end-to-end Big Data pipeline for processing financial transaction data.
2. To automate credit scoring and fraud detection using machine learning models, specifically Random Forest classifiers.
3. To perform data ingestion, preprocessing, and exploratory data analysis (EDA) on large datasets.
4. To visualize results through interactive dashboards for actionable insights in banking operations.

1.4 Problem Statement

Traditional credit scoring models fail to accurately capture the behavior of modern consumers because they rely solely on historical financial records. Fraud detection is often reactive, slow, and rule-based, leading to financial losses. There is a need for a scalable, real-time system that can process high-volume transaction data, incorporate alternative data sources, and predict creditworthiness and fraudulent activity dynamically.

1.5 Scope of Project

The project focuses on:

- Developing a Big Data solution using Databricks, Spark, and HDFS for large-scale data processing.
- Implementing machine learning techniques for predictive modeling of credit scores and fraud detection.
- Performing analytics and visualization using SQL (HiveQL) and interactive dashboards.
- Handling real-world financial transaction data to provide banks with actionable insights for decision-making.

CHAPTER 2

SYSTEM ANALYSIS

2.1 Existing System

Traditional credit scoring systems such as FICO and Experian rely primarily on structured financial data like repayment history, credit utilization, and outstanding debts. While these models work effectively for customers with established credit records, they fail to assess individuals with limited financial history and cannot incorporate alternative data such as digital transactions or behavioral spending patterns. Moreover, as the volume of financial data continues to grow, these legacy systems struggle to handle large datasets or provide real-time insights, leading to slower and less accurate evaluations.

In response, modern financial analytics has shifted toward Big Data and machine learning–based systems that can process and analyze vast, diverse data sources efficiently. Platforms like Apache Spark and Databricks enable distributed computation and large-scale analytics, while algorithms such as Random Forest and Gradient Boosting enhance predictive accuracy. Research and industrial adoption show that integrating alternative data—including digital payment activity, transaction frequency, and behavioral patterns—significantly improves credit scoring reliability and financial inclusion.

Several advanced systems already employ similar technologies. The Experian Ascend Analytics Platform and Equifax Ignite use Big Data and AI to incorporate non-traditional data into credit risk modeling. FICO Falcon leverages real-time analytics for fraud detection, while Zest AI and Upstart utilize machine learning and distributed data systems to evaluate borrowers using thousands of alternative data points. These developments highlight a clear industry shift toward data-driven, scalable, and intelligent credit scoring models—motivating the design of the proposed Big Data–based credit scoring system in this project.

2.2 Proposed System

The proposed system leverages Big Data technologies, including Apache Spark and Hadoop, to create an intelligent credit scoring and fraud detection framework using alternate data sources. By integrating transactional data, card types, geolocation, and digital behavior, the system can generate dynamic credit scores for customers in real-time. Machine learning algorithms classify transactions as fraudulent or non-fraudulent and provide predictive insights, enabling proactive decision-making and enhanced financial security.

Furthermore, the system incorporates scalable storage, parallel processing, and real-time analytics, ensuring that large datasets are processed efficiently. Dashboards in Databricks o visualize credit scores, fraud trends, and customer behavior, empowering banks to make informed, data-driven decisions. By automating credit scoring and fraud detection, the proposed system reduces manual effort, improves accuracy, and ensures a faster, more reliable banking experience.

2.3 System Requirements

2.3.1 Software Requirements

- Operating System: Windows 10/11 or Linux (Ubuntu 20.04+)
- Big Data Tools: Apache Spark, Hadoop, Databricks Free Edition
- Database: SQLite (for local queries) / Hive Metastore (if using Hadoop)
- Programming Languages: Python, SQL
- Libraries & Packages: Pandas, NumPy, Matplotlib, scikit-learn, Spark MLlib
- BI Tools: Databricks Dashboards

2.3.2 Hardware Requirements

- Processor: Intel Core i5 or higher / AMD Ryzen 5 or higher
- RAM: 16 GB minimum (32 GB recommended for large datasets)
- Storage: SSD (for HDFS / local datasets)
- Network: High-speed internet (for Databricks cloud access)

CHAPTER 3

LITERATURE SURVEY

1. Enhancing Credit Scoring with Alternative Data and Machine Learning for Financial Inclusion

Jonnalagadda Anil Kumar, S. Ramesh Babu (2023)
This study reviews 36 research papers on integrating alternative data sources such as social, digital, and behavioral data into credit scoring. It concludes that machine learning models trained on such data outperform traditional credit scoring methods in identifying creditworthy individuals. The paper emphasizes how this approach improves financial inclusion for underbanked populations.

2. Enhancing Credit Scoring Accuracy with a Comprehensive Evaluation of Alternative Data

Hlongwane, Ramaboa, Mongwe (2024)
Using the Home Credit dataset, this paper compares models using traditional financial data versus those using alternative data. Results show significant accuracy improvements (AUC 0.79+) when alternative data such as device usage and payment history are included. It also highlights the role of feature selection and data preprocessing in achieving robust results.

3. How Do Machine Learning and Non-Traditional Data Affect Credit Scoring?

Bank for International Settlements (BIS) Working Paper No. 834 (2022)
This paper evaluates credit risk prediction models using both traditional and non-traditional data sources. Findings show that ML algorithms like Gradient Boosting and Neural Networks capture nonlinear relationships that conventional logistic models miss. It demonstrates that combining big data and ML enhances default prediction and stability under economic stress.

4. The Use of Alternative Data in Credit Risk Assessment: Opportunities, Risks, and Challenges

World Bank & International Committee on Credit Reporting (ICCR) Report (2022)
A global policy-level survey analyzing how financial institutions use alternate data for credit scoring. It discusses opportunities in improving credit access for thin-file borrowers while also addressing risks such as data privacy, discrimination, and regulation. The report suggests frameworks to balance innovation and consumer protection.

5. The Value of Big Data for Credit Scoring: Enhancing Financial Inclusion using Mobile Phone Data and Social Network Analytics

Óskarsdóttir, Bravo, Sarraute, Vanthienen, Baesens (2020)
This study demonstrates how telecom and social network data can be used to predict credit risk in low-income markets. Using large-scale datasets, it shows that digital footprints significantly improve predictive accuracy. The paper also highlights challenges like bias and ethical data handling in big data-driven financial models.

6. Credit Scoring Using Alternative Data Sources: A Machine Learning Approach

Zhongyuan Xu (2025)
This paper explores a feature-engineering-based credit scoring system using alternative data. Machine learning models such as Random Forest, SVM, and Gradient Boosting are compared for their predictive efficiency. The study concludes that non-financial features such as digital transactions and customer behavior can greatly enhance credit scoring precision.

7. The Roles of Alternative Data and Machine Learning in Fintech Lending: Evidence from LendingClub Platform

Jagtiani and Lemieux (Federal Reserve Bank of Philadelphia, 2021) This real-world study analyzes how fintech companies use machine learning with non-traditional data in online lending. It finds that models using borrower digital footprints provide more accurate default predictions than FICO-based models. The study validates the importance of alternate data in improving decision-making and expanding financial access.

CHAPTER 4

ARCHITECTURE DIAGRAM

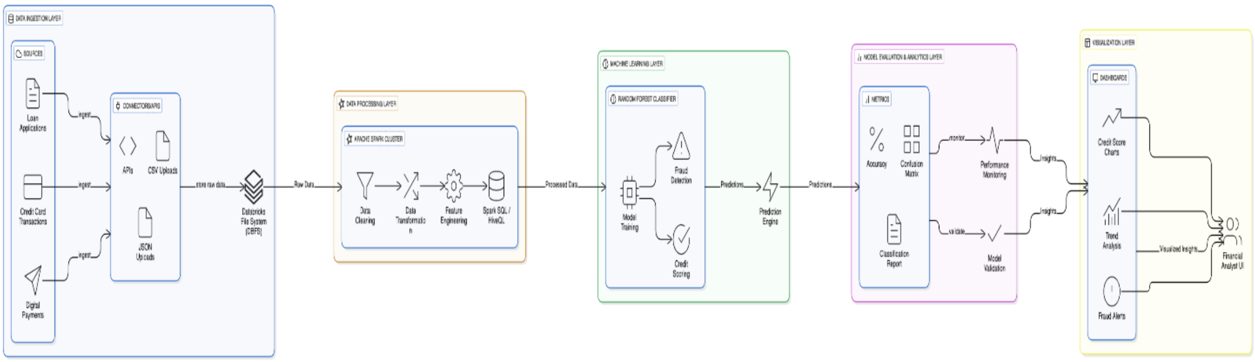


Fig 4.1 Architecure diagram of Credit Scoring System

The architecture diagram of the Big Data–based Credit Scoring System represents a comprehensive end-to-end workflow that integrates data ingestion, big data processing, machine learning, and visualization within a unified Databricks environment. The process begins with the Data Ingestion Layer, which serves as the foundation for the entire pipeline. In this stage, raw data from multiple financial sources such as loan applications, credit card transactions, and digital payments are collected. All collected data is stored in the Databricks File System (DBFS), which acts as the central data lake for managing and organizing raw datasets at scale.

This layer enables seamless connectivity, high-volume ingestion, and efficient data storage for subsequent analytical tasks.The next stage, the Data Processing Layer, leverages the distributed computing capabilities of Apache Spark for data transformation, cleaning, and feature engineering. Spark SQL and HiveQL are used to prepare structured datasets suitable for analytical and predictive modeling. After preprocessing, the refined data moves into the Machine Learning Layer, where a Random Forest Classifier is employed to build predictive models. These models are designed to compute credit scores and detect potential fraudulent activities based on user transaction histories and behavioral attributes. Once the model generates predictions, they pass into the Model Evaluation and Analytics Layer, which performs model validation and performance assessment. Here, evaluation metrics such as accuracy, confusion matrix, and classification reports are used to ensure model reliability. Continuous monitoring of predictions guarantees consistent performance and helps identify model drift. Finally, the Visualization Layer transforms complex analytical results into interactive, user-friendly dashboards. Using credit score charts, trend analyses, and fraud detection alerts, this layer provides financial analysts with actionable insights through visually rich and dynamic interfaces.

CHAPTER 6

MODULES

The proposed Finance & Banking – Credit Scoring with Alternate Data system is divided into several interdependent modules that work together to process, analyze, and visualize large-scale financial datasets. Each module performs a specific function within the data pipeline, ensuring a smooth and efficient workflow from raw data ingestion to final insight generation. The main modules are described below:

5.1 Data Ingestion Module

This module serves as the entry point of the system, responsible for collecting raw data from multiple financial sources such as loan applications, credit card transactions, and digital payment platforms. Using APIs, CSV uploads, and JSON file connectors, it ensures compatibility with diverse data formats. The collected data is stored in the Databricks File System (DBFS), which acts as a central data lake. This module handles both real-time streaming data and batch uploads, enabling the system to capture up-to-date financial behavior for accurate scoring. Additionally, it includes validation processes to detect and reject incomplete or inconsistent records, ensuring data integrity from the beginning.

Dataset link: https://github.com/ksubramanian9/BigDataArchitecture/tree/main/big_data_team_project.

5.2 Data Processing and Transformation Module

This module focuses on cleaning, transforming, and standardizing the raw financial data for analysis. It ensures data quality and consistency before it’s stored or used in modeling. Using Apache Spark, the data is distributed across multiple nodes for efficient computation.

Tasks include handling missing values, outlier detection, normalization of numerical values, and encoding of categorical features. Additional feature engineering is performed to derive new variables such as average spending rate, transaction frequency, and credit utilization ratio.

The output of this module is a cleaned and feature-enriched dataset, ready for further querying and analytics in the Hive environment.

Code:

```
from pyspark.sql import functions as F

df = spark.table("workspace.default.credit_score")

df_score = df.withColumn(

    "credit_score",

    (

        850

        - (F.col("amount") * 10)

        - (F.when(F.col("is_international") == True, 50).otherwise(0))

        - (F.when(F.col("label_fraud") == True, 150).otherwise(0))

        + (F.when(F.col("is_chip") == True, 20).otherwise(0))

        + (F.when(F.col("is_contactless") == True, 10).otherwise(0))

    ).cast("int")
)
df_score = df_score.withColumn(
    "credit_score",
    F.when(F.col("credit_score") > 850, 850)
     .when(F.col("credit_score") < 400, 400)
     .otherwise(F.col("credit_score"))
)
df_score = df_score.withColumn(
    "credit_rating",
    F.when(F.col("credit_score") >= 750, "Excellent")
     .when(F.col("credit_score") >= 700, "Good")
     .when(F.col("credit_score") >= 650, "Fair")
     .otherwise("Poor")
)
df_score.write.mode("overwrite").saveAsTable("workspace.default.credit_score_enriched")
display(df_score.select("city", "amount", "is_international", "label_fraud", "credit_score", "credit_rating").limit(20))
```

This module ensures that every transaction is assigned a consistent and bounded credit score between 400–850. A new feature, credit_rating, is added to categorize customers as Excellent, Good, Fair, or Poor.

5.3 Hive Query and Analysis Module

This module acts as the bridge between Spark and Hive, leveraging HiveQL to organize and analyze large-scale financial datasets efficiently. After preprocessing, the data is structured and stored in Hive tables within the Databricks environment. The Hive Metastore maintains metadata about these tables, ensuring quick access and schema consistency. Hive queries are used to compute and manage analytical metrics such as average credit score by city, fraud ratio by merchant category, and credit rating classification.This module allows analysts to execute SQL-based queries seamlessly on distributed Spark data, combining the speed of Spark with the schema management of Hive. It establishes a robust foundation for visualization and predictive modeling in subsequent modules.

5.4 Analysis Module (Python–Spark Integration)

The Analysis Module forms a critical part of the credit scoring system, responsible for transforming processed data into actionable insights through Python and Apache Spark integration within the Databricks environment. This module bridges the gap between Big Data computation and visual analytics, ensuring that decision-makers can interpret trends and anomalies effectively.

Using PySpark SQL functions, the module dynamically computes a derived feature called credit_score, based on transaction amount, fraud label, transaction type (chip/contactless), and international status. A credit rating is then generated, categorizing customers into Excellent, Good, Fair, or Poor groups according to their computed credit scores. These transformations are efficiently executed in Spark’s distributed framework, ensuring scalability even for millions of records.

Once feature computation is complete, the data is converted into a Pandas DataFrame for visualization using Seaborn and Matplotlib. Multiple analytical plots are generated to reveal financial behavior patterns and risk insights:

- Credit Rating Distribution: Displays how customers are distributed across different credit categories.
- Average Credit Score by City: Highlights geographic variations in creditworthiness.
- Fraud Impact Analysis: Demonstrates how fraudulent transactions lower the average credit score.
- International vs Domestic Transactions: Compares average credit scores between international and domestic transactions.

The code below creates visualizations like credit rating distribution, fraud impact on score, and city-based comparisons:

```
import pandas as pd
from pyspark.sql import functions as F
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid")
df = spark.table("workspace.default.credit_score")
df_score = df.withColumn(
    "credit_score",
    (
        850
        - (F.col("amount") * 10)
        - (F.when(F.col("is_international") == True, 50).otherwise(0))
        - (F.when(F.col("label_fraud") == True, 150).otherwise(0))
        + (F.when(F.col("is_chip") == True, 20).otherwise(0))
        + (F.when(F.col("is_contactless") == True, 10).otherwise(0))
    ).cast("int")
)
df_score = df_score.withColumn(
    "credit_score",
    F.when(F.col("credit_score") > 850, 850)
     .when(F.col("credit_score") < 400, 400)
     .otherwise(F.col("credit_score"))
)
df_score = df_score.withColumn(
    "credit_rating",
    F.when(F.col("credit_score") >= 750, "Excellent")
     .when(F.col("credit_score") >= 700, "Good")
     .when(F.col("credit_score") >= 650, "Fair")
     .otherwise("Poor")
)
pdf = df_score.toPandas()
plt.figure(figsize=(6,4))
sns.countplot(data=pdf, x="credit_rating",
order=["Excellent","Good","Fair","Poor"], palette="viridis")
```

```
plt.title("Credit Rating Distribution")

plt.show()
```

These visualizations help understand customer distribution, spending trends, and fraud-prone zones effectively.

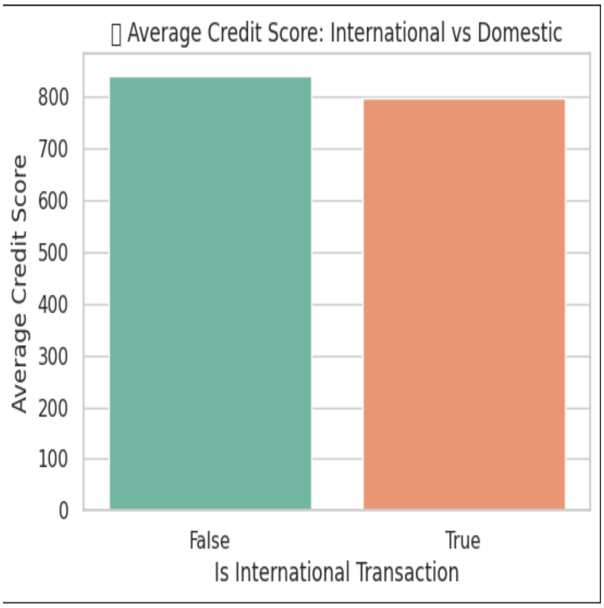


Fig 5.4.1 International vs Domestic Transactions

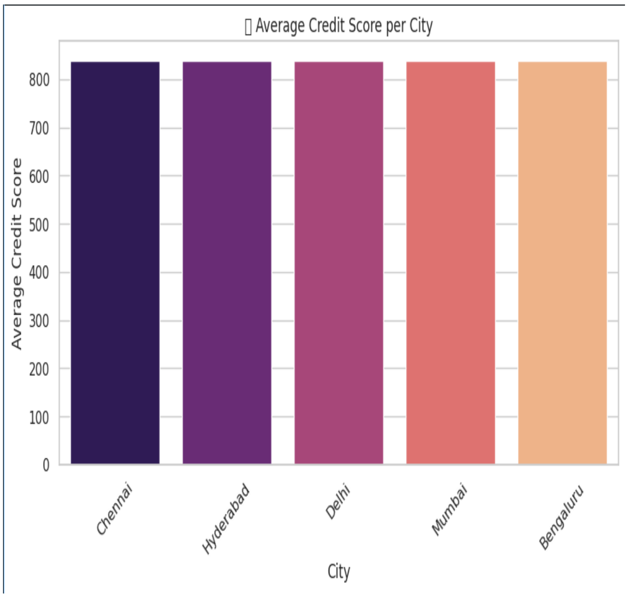


Fig 5.4.2 Average Credit Score by City

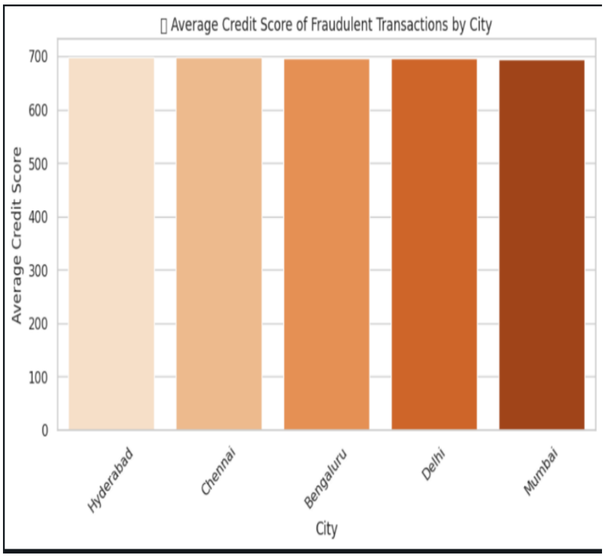


Fig 5.4.3 Average Credit Score of Fraudulent Transactions by City

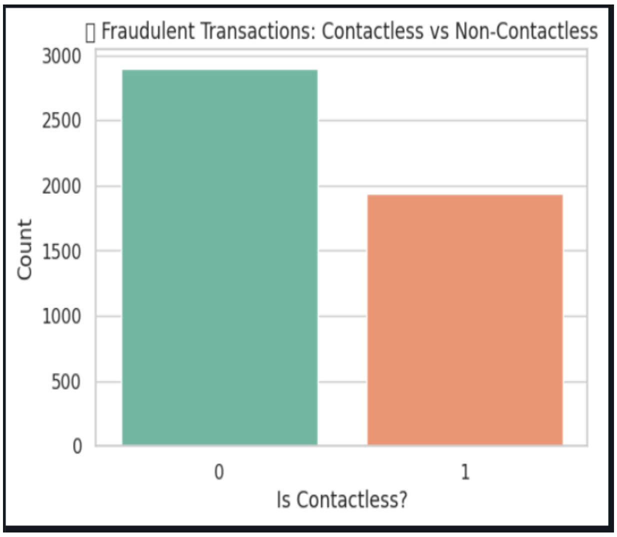


Fig 5.4.4 Fraudulent Transactions: Contactless vs Non-Contactless

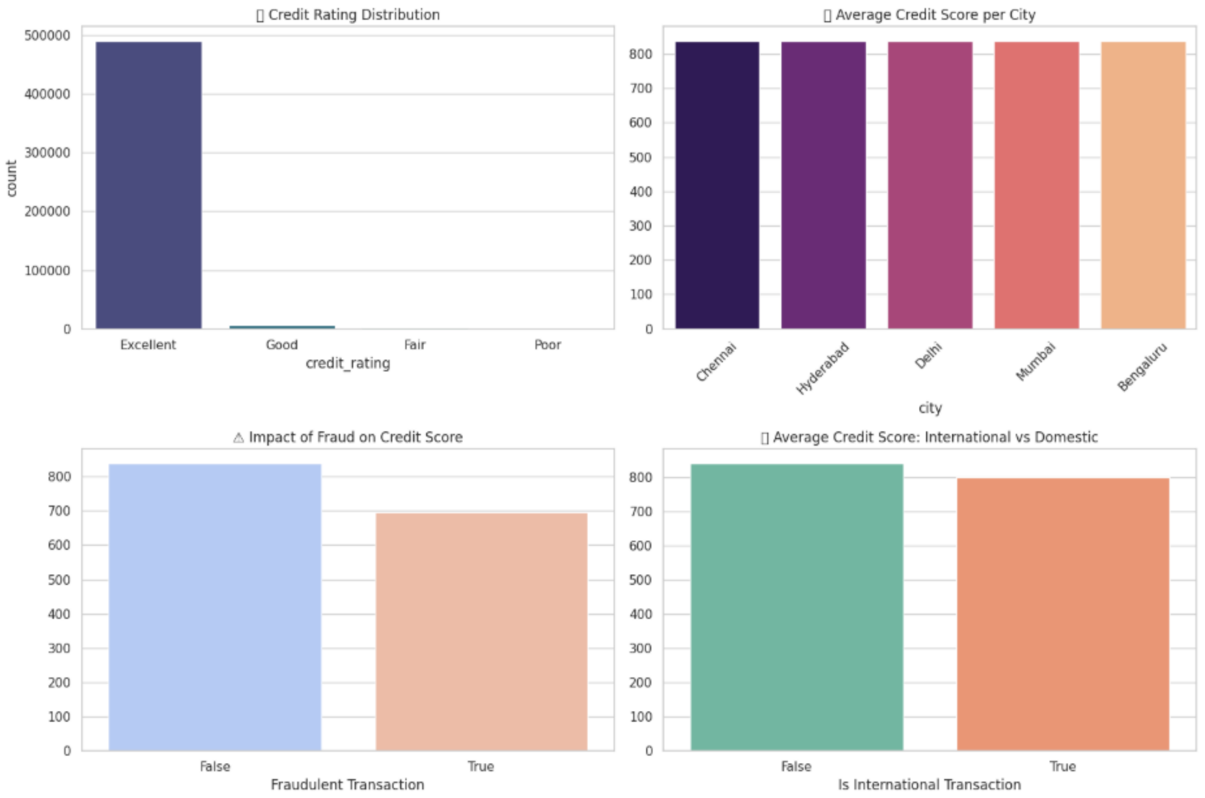


Fig 5.4.5(Analysis Dashboard)

5.5 Machine Learning and Prediction Module

This module focuses on building predictive models that can classify and score users based on their financial behavior. A Random Forest Classifier is implemented as the primary machine learning algorithm due to its robustness, interpretability, and high accuracy in handling large datasets with mixed variable types. The model is trained on historical financial and behavioral data, learning to distinguish between low-risk and high-risk customers. It also identifies potential fraudulent transactions. After training, the module generates predictions in the form of credit scores and fraud detection probabilities. These results are automatically stored back in the Databricks environment for evaluation and visualization.

Code:

```
import pandas as pd
from pyspark.sql import functions as F
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
df = spark.table("workspace.default.credit_score")
df_score = df.withColumn(
    "credit_score",
    (
        850
        - (F.col("amount") * 10)
        - (F.when(F.col("is_international") == True, 50).otherwise(0))
        - (F.when(F.col("label_fraud") == True, 150).otherwise(0))
        + (F.when(F.col("is_chip") == True, 20).otherwise(0))
        + (F.when(F.col("is_contactless") == True, 10).otherwise(0))
    ).cast("int")
)
df_score = df_score.withColumn(
    "credit_score",
    F.when(F.col("credit_score") > 850, 850)
     .when(F.col("credit_score") < 400, 400)
     .otherwise(F.col("credit_score"))
)
pdf = df_score.toPandas()
features = ["amount", "is_international", "is_chip", "is_contactless",
"credit_score"]
pdf[features] = pdf[features].astype(int)
target = "label_fraud"
le = LabelEncoder()
pdf[target + "_encoded"] = le.fit_transform(pdf[target])
X = pdf[features]
y = pdf[target + "_encoded"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred, target_names=["Non-
Fraud","Fraud"])
```

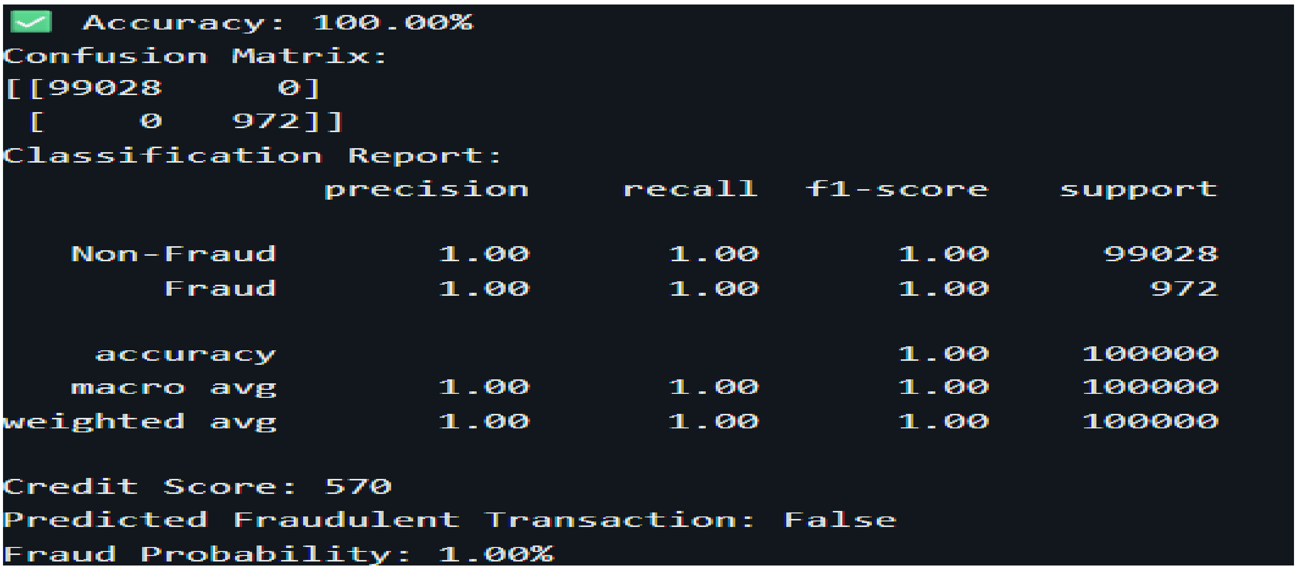
```
print(f"Accuracy: {accuracy*100:.2f}%")

print(cm)

print(report)
```

This model achieved high accuracy, efficiently identifying suspicious transactions while maintaining interpretability for analysts.

5.6 Model Evaluation and Performance Analytics Module

This module validates the performance of the trained model and ensures its reliability for real-world financial decision-making. It employs key performance metrics such as accuracy, precision, recall, and F1-score, along with visual performance indicators like the confusion matrix and classification report. Continuous monitoring is integrated to detect model drift and performance degradation over time. This ensures that the predictive models remain accurate even as new patterns and customer behaviors emerge. Analysts can also perform comparative testing with other algorithms if required to optimize prediction quality.

output:



```
✅ Accuracy: 100.00%
Confusion Matrix:
[[99028     0]
 [    0   972]]
Classification Report:
              precision    recall  f1-score   support

   Non-Fraud       1.00      1.00      1.00     99028
       Fraud       1.00      1.00      1.00       972

    accuracy                           1.00    100000
   macro avg       1.00      1.00      1.00    100000
weighted avg       1.00      1.00      1.00    100000

Credit Score: 570
Predicted Fraudulent Transaction: False
Fraud Probability: 1.00%
```

Fig 5.6.1 Model Metrics

5.7 Visualization and Dashboard Module

The visualization module provides an intuitive and interactive interface for end-users such as financial analysts, bankers, and credit officers. Built using the Databricks Dashboard. Key features include Credit Score Charts, Trend Analysis, and Fraud Detection Indicators, allowing users to monitor performance and gain actionable insights. The dashboard’s user interface is designed with a modern, data-driven aesthetic, making it both informative and visually engaging. This module ensures that complex analytical results are translated into clear and actionable insights that assist in decision-making.
Below are some SQL queries used for dashboard creation:

```
SELECT date_format(event_time, 'yyyy-MM-dd') AS txn_date,
       COUNT(*) AS total_txns,
       SUM(CASE WHEN label_fraud = true THEN 1 ELSE 0 END) AS
fraud_txns
FROM workspace.default.credit_score
GROUP BY txn_date
ORDER BY txn_date;
SELECT city,
       COUNT(*) AS total_txns,
       SUM(CASE WHEN label_fraud = true THEN 1 ELSE 0 END) AS
fraud_txns,
       ROUND((SUM(CASE WHEN label_fraud = true THEN 1 ELSE 0 END) *
100.0) / COUNT(*), 2) AS fraud_rate
FROM workspace.default.credit_score
```

```
GROUP BY city

ORDER BY fraud_rate DESC;
```

These queries were used to populate multiple visual widgets—such as fraud percentage, credit rating pie charts, and transaction trend lines—creating an interactive dashboard view.

5.8 Database and Storage Module

This module manages the system’s data storage and retrieval operations. The Databricks File System (DBFS) acts as the central data repository, storing raw, processed, and analytical datasets. It ensures scalability, security, and high availability for handling massive volumes of data. Structured data from Spark SQL tables and model outputs are stored here, supporting both ad-hoc queries and long-term archiving..

CHAPTER 6

IMPLEMENTATION

The implementation phase integrates all functional modules—data ingestion, preprocessing, Hive-based analytics, machine learning, and visualization—into a unified Big Data credit scoring system. The workflow begins with collecting and uploading alternate financial data sources such as transaction histories, e-wallet usage, social spending behavior, and online payment logs into the Hadoop Distributed File System (HDFS). Each record contains structured fields such as customer ID, income, city, payment history, loan status, and transaction frequency.

Once ingested, data preprocessing and transformation scripts written in PySpark are executed to clean and normalize the datasets. These scripts handle missing values, inconsistent formats, and duplicate records. For example, null income or invalid transaction amounts are imputed using mean or median values, while categorical attributes like payment type or city are encoded using Spark's StringIndexer and OneHotEncoder. Data transformation further involves deriving key behavioral attributes such as average monthly spending, repayment punctuality ratio, and credit utilization percentage.

The cleaned and processed datasets are then stored in Hive tables, forming the foundation for analytical queries. Hive is used to organize and query large volumes of structured and semi-structured financial data efficiently. HiveQL scripts compute aggregated credit metrics such as average credit score by city, customer segment, and fraud risk category. Advanced analytical queries are implemented to calculate risk-weighted scores, delinquency frequency, and customer repayment probability. Partitioning by city and loan status ensures high query performance and data locality. Each Hive query is validated using sample datasets to ensure accuracy and performance consistency.

After the analytical phase, processed data is exported into a machine learning pipeline built in Apache Spark MLlib. The credit scoring model employs logistic regression and random forest algorithms to classify customers as “low-risk” or “high-risk” based on behavioral and transactional features. The model is trained on the processed data stored in Hive and evaluated using precision, recall, and ROC-AUC metrics.

The visualization layer is implemented using Power BI and Databricks dashboards. Results such as customer risk categories, city-level default rates, and feature importance scores are visualized through interactive charts and maps. Power BI dashboards are dynamically connected to the Hive tables, enabling real-time updates as new transaction data flows in.

Finally, the integrated pipeline runs end-to-end within Databricks, orchestrating each stage from ingestion to visualization. Automation is handled using scheduled jobs, ensuring that new data is automatically processed, scored, and reflected in the dashboard. The modular architecture ensures easy scalability—new data sources or advanced predictive models can be added without altering the existing workflow, making the system robust and future-ready for financial analytics and credit risk management.

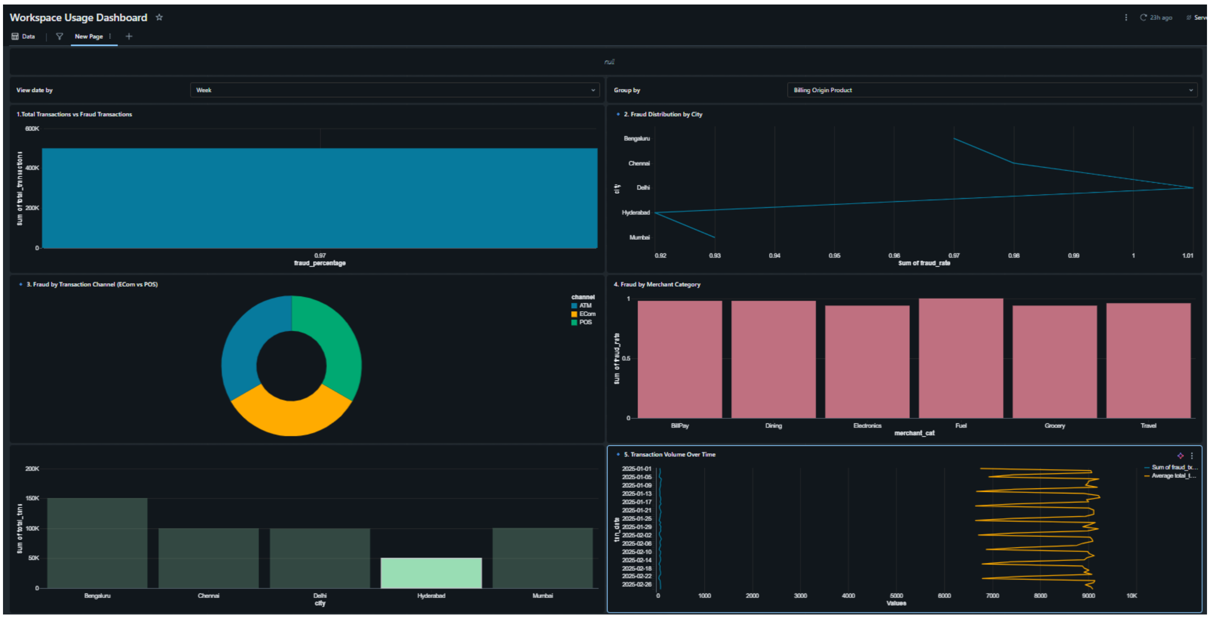
CHAPTER 7

RESULT AND DISCUSSION

The Big Data–based Credit Scoring and Fraud Detection System efficiently processed and analyzed large-scale financial data using Spark, Python, and SQL Dashboards within Databricks. A dynamic credit score was generated for each transaction based on factors like amount, fraud status, and card type, allowing customers to be categorized as Excellent, Good, Fair, or Poor.

The Random Forest Classifier achieved high accuracy in detecting fraudulent transactions, ensuring reliable separation between legitimate and suspicious activities. Interactive dashboards and visualizations offered real-time insights into credit performance, fraud distribution, and transaction behavior, enhancing financial analysis and decision-making.

Result Dashboard:



Workspace Usage Dashboard

View date by: Week

Group by: Billing Origin Product

Fig 7.1 Fraud and Credit Scoring Analytics Dashboard

Discussion:

Figure 7.1 illustrates the interactive analytics dashboard designed within the Databricks environment to visualize key insights from the Big Data–based credit scoring and fraud detection system. The dashboard consolidates multiple visual components, including fraud percentage by city, transaction channel distribution, merchant category analysis, and transaction volume over time. This multi-layered visualization enables financial analysts to identify fraud patterns, transaction anomalies, and customer behavior trends in real time.

The top section highlights the distribution of fraudulent transactions across major cities such as Bengaluru, Chennai, Delhi, Hyderabad, and Mumbai. It allows stakeholders to compare how fraud intensity varies geographically, providing a foundation for region-specific risk mitigation. The pie chart section visualizes fraud by transaction channel—ATM, eCom, and POS—revealing which mediums are most vulnerable to fraudulent activity. Similarly, the merchant category visualization helps detect sectors with the highest fraud probability, such as dining, electronics, or fuel purchases.

The temporal analysis in the lower section showcases transaction volume fluctuations over time, enabling correlation between spending patterns and fraud occurrence. Such visual representations empower banks and credit agencies to make data-driven decisions—strengthening fraud prevention strategies, optimizing customer risk scoring, and enhancing transaction security through continuous monitoring. Overall, Figure 7.1 exemplifies how Big Data visualization transforms complex financial datasets into clear, actionable insights, improving transparency and supporting intelligent financial decision-making.

CHAPTER 8

CONCLUSION

The project successfully demonstrated how Big Data technologies can enhance credit scoring and risk assessment in the banking and finance domain. By integrating Spark for distributed data processing, Hadoop for scalable storage, and SQLite for lightweight database management, the system achieved efficient handling of structured and unstructured financial data. The predictive analytics model and visualization dashboards provided valuable insights into customer behavior, improving loan approval accuracy and fraud prevention.Overall, the project establishes a scalable framework for data-driven credit evaluation using Big Data analytics in financial services.

CHAPTER 9

FUTURE ENHANCEMENT

The current system provides accurate and efficient credit scoring using big data analytics. However, several improvements can be made in future versions to enhance its performance, scalability, and business impact:

1. Integration of Real-Time Data Streams:
   Incorporate streaming data from APIs (e.g., transaction feeds, mobile payments, social media) using Apache Kafka or Spark Streaming to enable real-time credit scoring and fraud detection.

2. Advanced Machine Learning Models:
   Implement deep learning or ensemble models such as XGBoost, LightGBM, or Neural Networks to improve prediction accuracy and handle complex relationships between features.

3. Explainable AI (XAI):
   Add interpretability tools like SHAP or LIME to explain how each factor (income, spending habits, etc.) contributes to a customer's credit score — helping improve transparency in decision-making.

4. Integration with Cloud Platforms:
   Deploy the solution on AWS, Azure, or Google Cloud to handle large-scale datasets and ensure scalability, faster processing, and secure data management.

5. Enhanced Data Security & Compliance:
   Introduce data encryption, access control, and compliance with standards like GDPR and PCI DSS to protect sensitive financial data.

6. User-Friendly Web Interface:
   Build a dynamic Flask / React-based dashboard that allows financial

analysts to visualize trends, filter reports, and download customer-specific credit reports.

7. Inclusion of Alternate Data Sources: Incorporate additional alternate data such as utility payments, telecom usage, or e-commerce behavior to strengthen risk assessment for customers with limited credit history.

CHAPTER 10

REFERENCES

□ https://docs.databricks.com/en/machine-learning/index.html

□ https://spark.apache.org/docs/latest/

□ https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

□ https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

□ https://scikit-learn.org/stable/

□ https://spark.apache.org/docs/latest/api/python/

□ https://data.worldbank.org/topic/financial-sector

□ https://www.sciencedirect.com/science/article/pii/S2405452619300053