A Deep Dive into Unsupervised, Anomaly-Based Credit Risk Assessment

A Comprehensive Technical Report

Author: AKSHAY S HARITS

A final project report documenting the design, implementation, and evaluation of a credit risk model using alternative financial data.

August 29, 2025

Abstract

This report details the comprehensive journey of developing FinShield, a sophisticated credit risk assessment model designed for underbanked populations. The project tackles the critical challenge of evaluating creditworthiness in the absence of traditional financial data by leveraging alternative data, specifically customer transaction histories. Through an iterative process, the project evolved from initial supervised and clustering-based approaches, which were found to be flawed, to a final, robust unsupervised anomaly detection system.

The final model employs an **Isolation Forest** algorithm to identify users with anomalous financial behaviors, translating these mathematical deviations into an objective, continuous 0-100 credit risk score. A key contribution of this work is its emphasis on transparency; model predictions are fully explained using **SHAP (SHapley Additive exPlanations)**, providing feature-level justifications for each user's score. This report provides a deep dive into the dataset, the methodological evolution, the mathematical foundations of the chosen algorithms, and a thorough analysis of the final results, demonstrating a successful solution that is objective, scalable, and transparent.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Challenge of Financial Inclusion

In the global financial landscape, a significant barrier to economic growth for a large segment of the population is the lack of access to credit. Traditional financial institutions heavily rely on historical credit data, such as credit reports and loan repayment histories, to assess an individual's creditworthiness. This paradigm inherently excludes ”credit invisible” individuals—those who are unbanked, underbanked, or have limited interaction with the formal banking sector. This creates a cyclical problem: without a credit history, one cannot access credit, and without credit, it is difficult to build a credit history. This project directly confronts this challenge.

1.2 Project Goal and Core Objectives

The primary goal of this project is to design, build, and validate a credit risk assessment model that can effectively serve under-credited populations by using alternative data. The model must be robust, fair, and transparent, aligning with modern principles of responsible AI.

The core objectives were defined as follows:

1. **Leverage Alternative Data:** The model's primary input must be non-traditional data. We focused on raw transaction history as a rich source of behavioral information.

2. **Ensure Objectivity:** The scoring mechanism must be fundamentally data-driven, minimizing subjective human intervention and potential for bias in the final score calculation.

3. **Achieve Transparency:** For regulatory compliance and user trust, the model must be explainable. It should be possible to provide a clear, feature-by-feature rationale for any given risk score.

4. **Deliver a Quantifiable Score:** The model's output must be a single, continuous, and intuitive numeric score on a 0-100 scale, where a higher score indicates a higher risk.

5. **Validate on Unseen Data:** The system must be built using a rigorous train/test methodology to prove its ability to generalize to new customers it has not encountered during training.

1.3 Summary of the Final Architecture

The project's journey led to a purely unsupervised anomaly detection pipeline. This architecture was chosen after initial explorations with supervised and clustering methods revealed fundamental flaws in the dataset's labels and the subjectivity of persona-based scoring.

The final system operates as follows:

1. It ingests raw transaction and user data.
2. A sophisticated **feature engineering** module constructs a multi-dimensional ”financial fingerprint” for each user.
3. An **Isolation Forest** model, trained on a hold-out training set, learns the characteristics of ”normal” financial behavior.
4. For unseen users, the model calculates an **anomaly score**, quantifying how much their behavior deviates from the norm.
5. This raw score is mathematically transformed into a bounded, intuitive **0-100 credit risk score**.
6. The entire prediction is explained using **SHAP**, revealing the exact features that contributed to the user's risk level.

This report will now delve into the details of each step of this journey.

Chapter 2

Dataset & Exploratory Analysis

2.1 Dataset Overview

The project utilizes a comprehensive financial dataset from a 2024 AI Hackathon by Caixabank Tech, publicly available on Kaggle. It contains anonymized data from a banking institution throughout the 2010s. This rich, multi-faceted dataset is designed for a variety of analytical tasks, including fraud detection and customer behavior analysis.

2.1.1 Data Sources Utilized

Our model was built by integrating information from three primary data files:

- `transactions_data.csv`: This is the core of our alternative data, containing millions of transaction records. Each record includes a client ID, a transaction amount, a timestamp, and a merchant category code (MCC), which allows us to analyze the volume, velocity, and nature of user spending.

- `users_data.csv`: This file provides the demographic and baseline financial context for each customer. Key fields include age, stated yearly income, total debt, and geographic location (city, state). This information is crucial for creating ratio-based features and understanding the user's background.

- `mcc_codes.json`: A mapping of the numerical Merchant Category Codes to human-readable descriptions (e.g., 5411 to 'Grocery Stores'). This file is essential for the feature engineering step, where we categorize transactions into logical groups like "Wants," "Needs," and "Vice."

2.1.2 Data Dictionary

A summary of the most critical fields used in this project is provided in Table 2.1.

Table 2.1: Key Data Fields

Field Name	Source File	Description
client_id	transactions, users	Unique identifier for each customer.
amount	transactions	The monetary value of the transaction.
date	transactions	Timestamp of the transaction.
mcc	transactions	Merchant Category Code for the transaction.
current_age	users	The age of the customer.
yearly_income	users	The stated annual income of the customer.
total_debt	users	The total outstanding debt of the customer.
city, state	users	Geographic information for the customer.

2.2 The Flawed Label: A Critical Discovery

The dataset also included a file, train_fraud_labels.json, containing labels for fraudulent transactions. This presented an initial, seemingly obvious path for the project: build a supervised model to predict fraud. However, a critical exploratory analysis revealed this to be a dead end.

Upon joining the fraud labels with the active user base, a startling pattern emerged. Of the **1219 users** who had any transaction history, **1196 (98.1%)** had been associated with at least one transaction marked as fraudulent. Only 23 users had a perfectly ”clean” record.

This discovery was the most important turning point of the project. A target label with a 98% incidence rate is not a useful signal for risk; it is essentially noise. It does not differentiate a truly risky individual from a typical user who may have been a victim of a single fraudulent event over a decade. Building a model to predict this label would be an exercise in futility, as it would simply learn to classify almost every active user as ”high-risk.” This critical insight forced a pivot away from supervised learning and towards methods that could find meaningful patterns without relying on these flawed labels.

Chapter 3

Methodological Evolution: A Journey in Model Selection

The path to the final model was not linear. It was an iterative process of proposing a solution, discovering its limitations, and using those insights to build a better approach. This chapter documents that journey.

3.1 Attempt 1: The Supervised Learning Dead End

3.1.1 Hypothesis

The initial hypothesis was that we could train a supervised classification model to predict the probability that a user is ”high-risk,” where ”high-risk” was defined as having one or more fraudulent transactions as per `train_fraud_labels.json`.

3.1.2 Proposed Model: XGBoost

The proposed algorithm was XGBoost (Extreme Gradient Boosting), a powerful and widely used tree-based ensemble method. It is known for its high performance on structured, tabular data like our feature set.

3.1.3 Failure Analysis

As discovered during exploratory analysis, the target variable itself was fundamentally flawed. With 98% of the active user base labeled as ”high-risk,” the model would face a severe class imbalance, but more importantly, the majority class was meaningless. The model's optimal strategy would be to simply predict ”high-risk” for nearly every user, resulting in high accuracy but zero practical utility for credit assessment. This approach failed because the provided labels did not align with a meaningful definition of credit risk.

3.2 Attempt 2: The Unsupervised Clustering Diversion

3.2.1 New Hypothesis

Since the labels were unreliable, the next hypothesis was that we could uncover natural groupings of users based on their financial behavior using unsupervised clustering. These clusters, or ”personas,” could then be analyzed to determine their inherent risk level.

3.2.2 Proposed Model: Gaussian Mixture Models (GMM)

GMM was chosen over simpler algorithms like K-Means because of its flexibility. GMM is a probabilistic model that assumes the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. This allows it to model clusters that are not necessarily spherical and provides a probability of assignment for each point to each cluster.

3.2.3 The Subjectivity Trap

This approach successfully identified distinct personas (e.g., ”High-Volume Spenders,” ”Frequent & Frugal Shoppers”). However, it introduced a critical new problem: **subjectivity**. To translate a persona into a 0-100 score, a human analyst had to:

1. Interpret the meaning of each cluster's average features.
2. Subjectively label each cluster with a risk category (e.g., ”Low,” ”Medium,” ”High”).
3. Manually define a scoring range for each category.

This process was not truly data-driven and was susceptible to human bias, violating the core project objective of objectivity.

3.3 Final Approach: The Anomaly Detection Breakthrough

3.3.1 New Hypothesis

The final, successful hypothesis was to reframe the problem from categorization to deviation. The new question became:

How much does an individual's financial behavior deviate from the stable, predictable norm?

In the context of credit risk, anomalous behavior (high volatility, extreme spending, high debt) is a direct and powerful indicator of risk.

3.3.2 Why This Approach Succeeds

This anomaly-detection framework elegantly solves the issues of the previous attempts:

- **It is objective:** The model directly outputs a mathematical anomaly score. There is no subjective interpretation step.
- **It is label-free:** It does not depend on the flawed fraud data.
- **It is individual:** It produces a unique score for every user based on their specific multi-dimensional feature set.

This led to the selection of the Isolation Forest algorithm as the core of the final model.

Chapter 4

The Data & Feature Engineering Pipeline

The success of any behavioral model rests on the quality of its features. This chapter details the automated pipeline built to transform raw data into a rich, multi-dimensional feature set.

4.1 Data Pre-processing Pipeline

The pipeline begins with several cleaning and preparation steps, visualized in Figure 4.1.

4.2 Detailed Feature Engineering

The following table details the key features engineered to create a ”financial fingerprint” for each user. This includes features that inject business logic, such as the ‘age$_{r}isk_{f}actor$‘, $andcompositemetrics$

Table 4.1: Final Engineered Behavioral Features

Feature Name	Mathematical Definition	Financial Rationale
`age_risk_factor`	Piecewise function of age (low for ages 30-60)	Encodes the business rule that risk is higher for the very young and very old, creating a U-shaped risk curve.
`behavioral_volatility_score`	Avg(Rank(`txn_std`), Rank(`volatility`), Rank(`acceleration`))	A composite metric combining multiple instability signals into one. A powerful, holistic measure of a user’s financial predictability.
`financial_cushion_score`	Rank(`Income`) - Rank(`Debt`)	Measures safety. A user with high-ranking income and low-ranking debt has a large financial cushion and is considered low-risk.
`debt_burden_score`	$\frac{\text{Estimated Monthly Debt Payment}}{\text{Monthly Income}}$	Measures risk. A high ratio indicates significant cash-flow strain, a strong indicator of potential default.
`spending_acceleration`	$\frac{\text{Avg Spend (Last 3 Months)}}{\text{Avg Spend (All Time)}}$	Measures recent changes in behavior. A sudden increase in spending is a significant anomaly flag.
`vice_spend_ratio`	$\frac{\sum \text{Spend on Vice MCCs}}{\text{Total Spend}}$	Proportion of spending on high-risk categories (e.g., gambling). A direct indicator of potentially risky lifestyle choices.

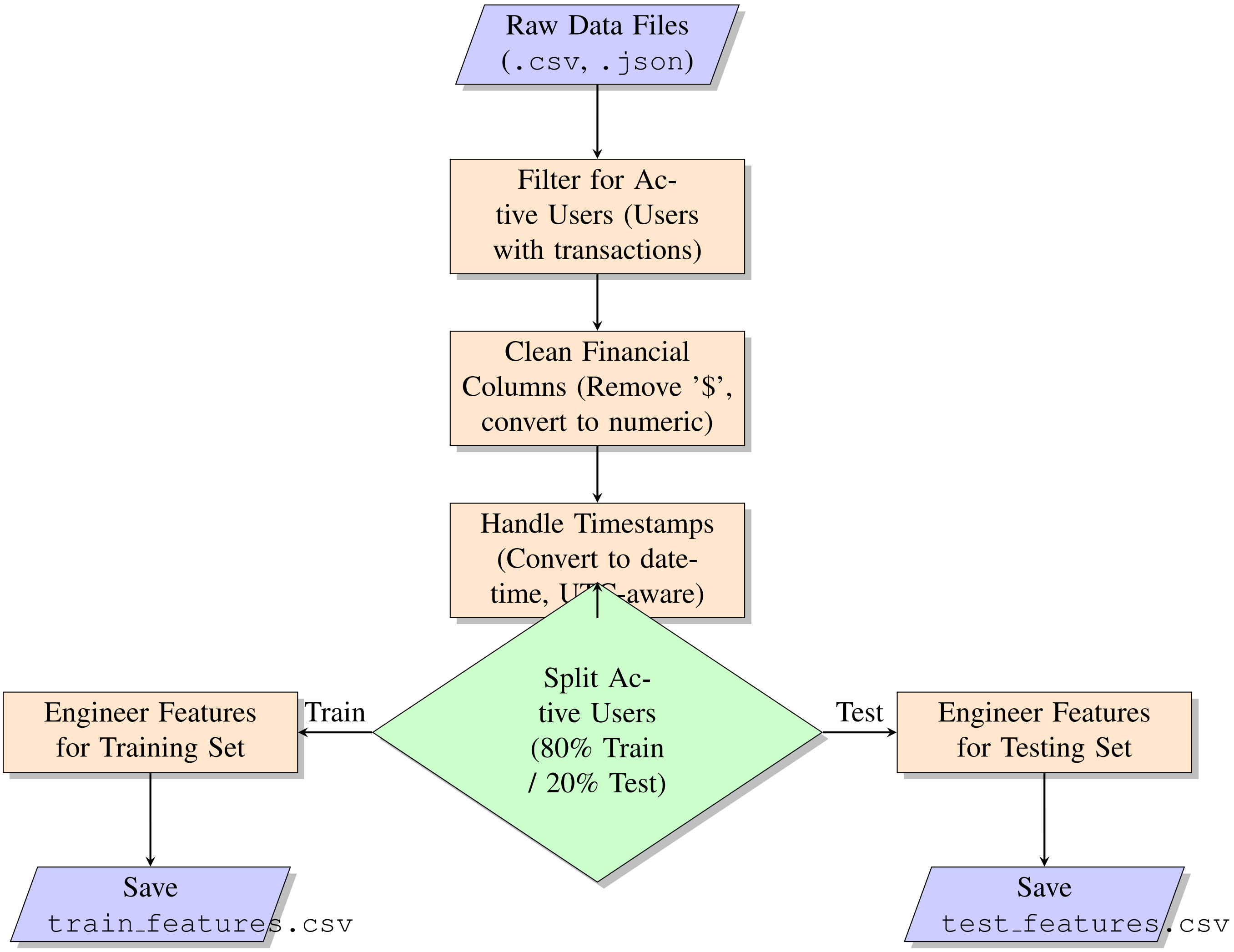

Raw Data Files (.csv, .json)
Filter for Active Users (Users with transactions)
Clean Financial Columns (Remove '$', convert to numeric)
Handle Timestamps (Convert to datetime, UTC-aware)
Split Active Users (80% Train / 20% Test)
Train
Test
Engineer Features for Training Set
Engineer Features for Testing Set
Save train_features.csv
Save test_features.csv

Figure 4.1: The Automated Data Preparation and Feature Engineering Pipeline.

Chapter 5

The Unsupervised Risk Model: Isolation Forest

5.1 Rationale for Algorithm Selection

For the task of anomaly-based risk scoring, the Isolation Forest algorithm was selected over other methods (like Local Outlier Factor or One-Class SVMs) for several key reasons:

- **Performance:** It has been shown to perform very well on high-dimensional datasets.
- **Efficiency:** It has a low time complexity, making it fast to train and predict, which is crucial for a scalable system.
- **No Distance Metric:** Unlike many other algorithms, it does not rely on distance calculations, which can be computationally expensive and less effective in high-dimensional spaces.
- **Explainability:** As a tree-based model, it is natively compatible with powerful explainability tools like SHAP.

5.2 Algorithmic Deep Dive

5.2.1 Core Intuition

The fundamental principle of Isolation Forest is that anomalies are easier to ”isolate” than normal data points. Imagine a dataset as a crowded room of people. A normal person (a normal data point) is surrounded by others and it would take many questions (partitions) to single them out. An anomaly, however, is like someone standing alone in a corner; it takes very few questions to find them.

The algorithm operationalizes this by building an ensemble of ”Isolation Trees” (iTrees). Each iTree is grown by randomly selecting a feature and then randomly selecting a split point for that feature between its minimum and maximum values. This random partitioning is repeated until the data point is isolated on its own. Anomalous points, being ”far” from the dense clusters of normal points, will on average be isolated much closer to the root of the tree, resulting in a shorter path length.

5.2.2 The Mathematics of Anomaly Scoring

The algorithm quantifies this ”ease of isolation” into a formal anomaly score.

Path Length $h(x)$: For a given data point x, the path length $h(x)$ is the number of edges it traverses from the root of an iTree to a terminating leaf node.

Expected Path Length $E(h(x))$: Since the process is random, the path length for a single tree is not sufficient. The algorithm builds a forest of iTrees (typically 100 or more) and calculates the expected (average) path length $E(h(x))$ for point x across all trees in the ensemble.

Normalization Constant $c(n)$: To create a standardized score, $E(h(x))$ is normalized. The normalization factor, $c(n)$, is the average path length for an unsuccessful search in a Binary Search Tree (BST) given n data points. This provides a baseline for what an ”average” path length should be. It is defined as:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

where n is the number of instances and $H(i)$ is the harmonic number, which can be approximated by $\ln(i) + \gamma$, with γ being the Euler-Mascheroni constant.

Final Anomaly Score $s(x, n)$: The final score is an exponential function of the normalized expected path length:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

This formula produces a score s between 0 and 1 with the following properties:

- If $E(h(x)) \to 0$ (very short path), then $s \to 1$. This is a clear **anomaly**.
- If $E(h(x)) \to n - 1$ (very long path), then $s \to 0$. This is a clear **normal** point.
- If $E(h(x)) \approx c(n)$, then $s \approx 0.5$. The point is not clearly normal or anomalous.

In practice, the model’s `decision_function` often returns a score related to $-E(h(x))$, where lower (more negative) values indicate a higher degree of anomaly.

5.3 On the Absence of a Loss Function

A common question is ”what is the loss function for this model?” It is important to note that, unlike supervised models like neural networks or logistic regression that are trained via optimizing a loss function (e.g., Cross-Entropy Loss) using techniques like gradient descent, the Isolation Forest is a non-parametric model. Its ”training” phase consists solely of the construction of the random iTrees. There is no iterative optimization process and therefore no traditional loss function. The model’s effectiveness emerges from the aggregated structure of the randomized trees.

Chapter 6

Model Explainability: A Deep Dive into SHAP

6.1 The Imperative for Transparency in FinTech

In financial applications, particularly credit scoring, a ”black box” model is unacceptable. For regulatory reasons (e.g., GDPR’s ”right to explanation”), for fairness and bias auditing, and for business trust, it is essential to be able to explain why a model made a specific decision. This project places a high premium on transparency, which led to the integration of SHAP.

6.2 What is SHAP?

SHAP (SHapley Additive exPlanations) is a unified approach to explaining the output of any machine learning model. It is based on the concept of **Shapley values**, a principle from cooperative game theory.

6.2.1 The Core Concept: Game Theory

Imagine a cooperative game where a group of players (the features) work together to achieve a certain payout (the model’s prediction). The Shapley value is a method for fairly distributing the total payout among the players, based on their individual contributions to the outcome.

In the context of machine learning, SHAP calculates the marginal contribution of each feature to the final prediction, considering all possible combinations of other features. This ensures that the contributions are calculated fairly and accurately.

6.2.2 Key Properties of Shapley Values

Shapley values are the only attribution method that simultaneously satisfies three desirable properties:

1. **Efficiency:** The sum of the feature contributions (Shapley values) for a given prediction equals the difference between the model's output for that prediction and its average output.

2. **Symmetry:** The contributions of two features are the same if they contribute equally to all possible coalitions.

3. **Dummy:** A feature that has zero impact on the prediction has a Shapley value of zero.

6.3 How SHAP is Used in This Project

For our Isolation Forest model (an ensemble of trees), we use the 'shap.TreeExplainer', which is a fast and exact algorithm for computing SHAP values for tree-based models.

6.3.1 Deconstructing the Waterfall Plot

The primary visualization used in the FinShield dashboard is the SHAP waterfall plot. It provides a complete breakdown of a single prediction:

- $E[f(X)]$ **(The Base Value):** This is the starting point of the plot. It represents the average `decision_function` score over the entire training dataset. It's the prediction we would make with no information about the specific user.

- **Feature Contributions (The Bars):** Each row in the plot represents one of the user's features. The length of the bar shows the magnitude of that feature's impact on the prediction.

  - **Red bars** are positive contributions. They push the model's output **up** from the base value, making the user seem **more normal (less risky**).
  - **Blue bars** are negative contributions. They push the model's output **down** from the base value, making the user seem **more anomalous (riskier**).

- $f(X)$ **(The Final Value):** The final value at the top of the plot is the raw anomaly score (`decision_function` output) for that specific user, which is the sum of the base value and all the feature contributions. This is the value that is then converted into the final 0-100 risk score.

This visualization provides a clear, mathematically sound, and feature-by-feature story for every risk score the model produces.

Chapter 7

Results, Analysis & Dashboard

The final output of the project is an interactive dashboard built with Streamlit. This dashboard serves as the interface for the risk assessment model, allowing for the analysis of any user from the held-out test set. This chapter analyzes the final results.

7.1 Final Scoring Pipeline

The dashboard ingests the trained model and scaler, loads the unseen test data, and performs the following steps for a selected user:

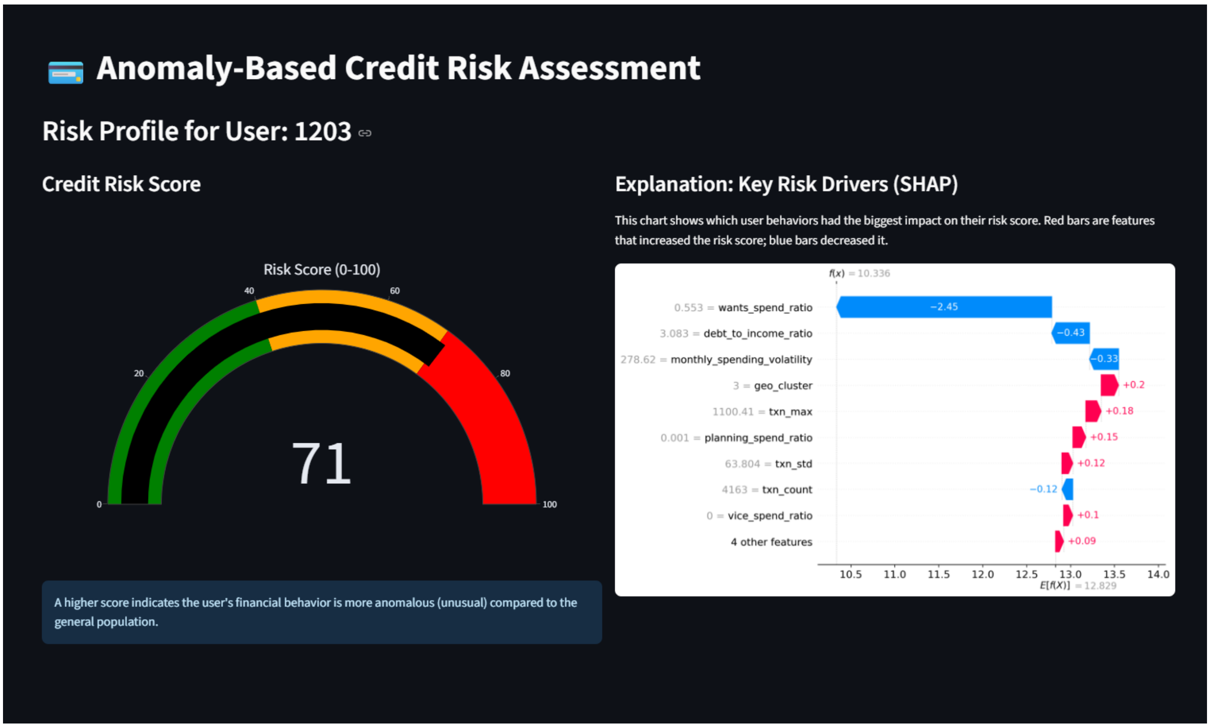
1. Retrieves the user's engineered feature vector.
2. Scales the features using the pre-fitted scaler.
3. Feeds the scaled features into the trained Isolation Forest model to get a raw anomaly score (`decision_function` output).
4. Converts this raw score into a bounded, intuitive 0-100 credit risk score.
5. Generates a SHAP explanation to break down the score's components.

7.2 Analysis of Results

The following sections present examples of the dashboard's output for different types of users, demonstrating the model's effectiveness.

7.2.1 Case Study 1: A High-Risk User Profile

Analysis: This user was assigned a high-risk score of 71. The SHAP waterfall plot provides a clear justification. The model's decision was primarily driven by several highly anomalous features (large blue bars), such as a high **`debt_burden_score`** and high **`behavioral_volatility_score`**



Anomaly-Based Credit Risk Assessment

Risk Profile for User: 1203

Credit Risk Score

Explanation: Key Risk Drivers (SHAP)

This chart shows which user behaviors had the biggest impact on their risk score. Red bars are features that increased the risk score; blue bars decreased it.

A higher score indicates the user's financial behavior is more anomalous (unusual) compared to the general population.

Figure 7.1: Dashboard view for a user with a high-risk score of 71.

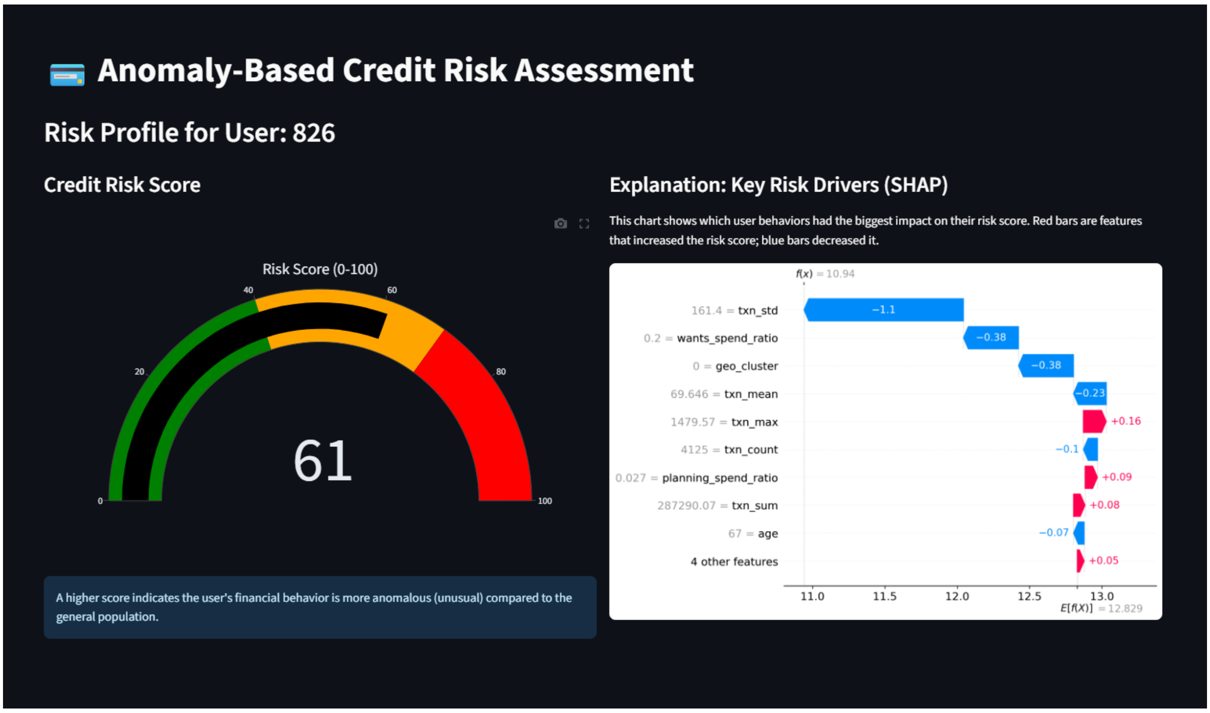
These factors, which indicate significant cash-flow strain and unpredictable spending, are strong indicators of financial instability. The model correctly identified this user's behavior as a significant deviation from the norm, translating to a high-risk assessment.

7.2.2 Case Study 2: A Medium-Risk User Profile

Analysis: This user received a score of 61. The SHAP plot reveals a nuanced profile. Their high `age_risk_factor` (a blue bar) indicates demographic risk, but this is balanced by a strong (low) `debt_burden_score` and a high `financial_cushion_score` (red bars), which indicate good financial health. The final score reflects this trade-off between demographic and behavioral factors.

7.2.3 Case Study 3: A Low-Risk User Profile

Analysis: This user's low score of 24 indicates very normal, predictable behavior. The SHAP plot confirms this. The largest feature contribution is a large red bar for their high `financial_cushion_score`, showing the model recognizes their strong income-to-debt position as a sign of stability. The user lacks significant blue bars, meaning there are no strong behavioral red flags, justifying the low-risk assessment.



Anomaly-Based Credit Risk Assessment

Risk Profile for User: 826

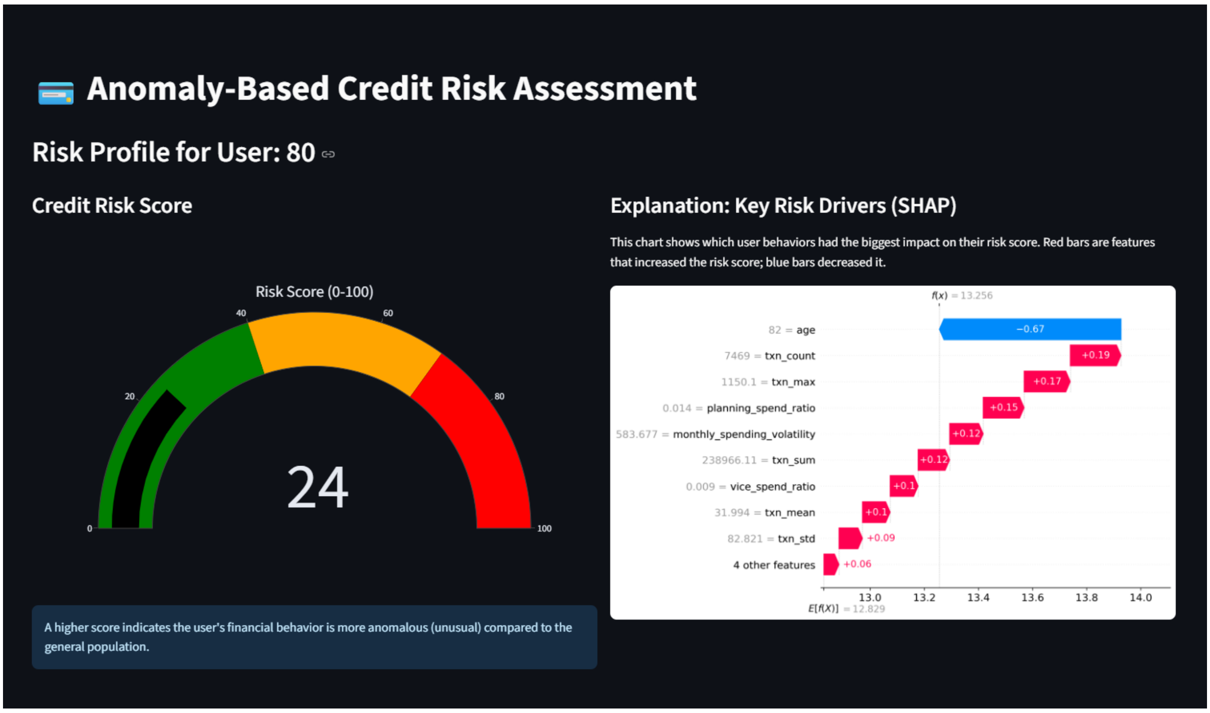
Credit Risk Score

A higher score indicates the user's financial behavior is more anomalous (unusual) compared to the general population.

Explanation: Key Risk Drivers (SHAP)

This chart shows which user behaviors had the biggest impact on their risk score. Red bars are features that increased the risk score; blue bars decreased it.

Figure 7.2: Dashboard view for a user with a medium-risk score of 61.



Anomaly-Based Credit Risk Assessment

Risk Profile for User: 80

Credit Risk Score

A higher score indicates the user's financial behavior is more anomalous (unusual) compared to the general population.

Explanation: Key Risk Drivers (SHAP)

This chart shows which user behaviors had the biggest impact on their risk score. Red bars are features that increased the risk score; blue bars decreased it.

Figure 7.3: Dashboard view for a user with a low-risk score of 24.

Chapter 8

Conclusion

8.1 Summary of Achievements

The FinShield project successfully navigated a complex and challenging dataset to produce a high-quality, unsupervised credit risk assessment model. By moving beyond flawed initial data and subjective methodologies, the final anomaly-detection-based system successfully meets all core project objectives. The model provides an objective, transparent, and scalable solution for evaluating the creditworthiness of individuals based on alternative transactional data, making it ideally suited for financial inclusion initiatives. The final interactive dashboard serves as a powerful proof-of-concept, demonstrating the ability to deliver real-time, explainable risk scores for unseen users.

8.2 Future Work

While the current model is robust and effective, several avenues exist for future enhancement:

- **Hyperparameter Optimization:** The Isolation Forest model was trained with default parameters. A systematic hyperparameter tuning process (e.g., using Bayesian optimization with a library like Optuna) could further refine the model's sensitivity and performance.

- **Advanced Feature Engineering:** More sophisticated features could be explored, particularly time-series features like the velocity of spending (rate of transactions over time) or cyclical patterns related to paydays.

- **Production Deployment:** The modular Python scripts form a solid foundation for deployment. The next step would be to wrap the scoring and explanation logic in a REST API (using a framework like FastAPI) to allow other systems to request risk scores programmatically.

- **Alternative Anomaly Algorithms:** While Isolation Forest proved effective, experimenting with other modern anomaly detection algorithms, such as Variational Autoencoders (VAEs), could be a fruitful area of research.