Fairness-Aware Loan Approval Prediction using Machine Learning

Vaidehi Takalkar
2412607

A thesis submitted for the degree of

Master of Science in Artificial Intelligence and its Applications

Supervisor: Dr. Gareth Howells
School of Computer Science and Electronic Engineering
University of Essex

December 2025

Abstract

This study looks into using machine learning for credit scoring, while keeping an eye on fairness and making reasons clear. A made-up set with money habits, actions, and personal details helped build two tools - one based on decision trees, another on probability clusters - to see which works better under fixed conditions. Instead of judging actual loan practices or bias, the focus stayed on testing model performance in a test setting.

Two versions got checked through precision and bias tests. Though the Random Forest did better at guessing right, the Bayes option helped spot how choices differ. Approval stats plus gap ratios showed if some people faced uneven treatment. Local insights came from LIME, pointing out what factors swayed each loan call.

The findings show machine learning can predict credit well - though keeping an eye on fairness is key. This work proves tools such as LIME bring clarity to automatic choices, making them easier to understand. In short, building better credit systems means balancing precision with equity and transparency.

Acknowledgements

I’d love to thank Dr Gareth Howells for steering me through this whole project. Because of his tips and constant push, I’ve grown a lot during my research journey. The hours he spent guiding me really shaped how I see the topic now. Thanks to him, I feel stronger about what I know and where my studies are headed.

Contents

Chapter 1
Introduction

Credit scoring's been around forever, shaping how banks decide who pays back loans. Over time, companies stuck with methods like logistic regression since they're straightforward and don't raise red flags with officials [1]. Old-school models work fine when someone needs to understand why a decision went a certain way - say, for clients or inspectors.

Still, how people manage money isn't what it used to be. These days, loan requests come with all kinds of extra details, while customers act in ways old math models can't predict well. At the same time, ML's role has grown because it handles messy real-world trends better than traditional tools. It spots hidden clues in banking info without needing strict rules. Studies prove such systems usually work better than classic credit checks - particularly when dealing with big or tricky sets of records [2][3].

Even with these benefits, machine learning can cause issues. Some top-performing models work like hidden systems - hard to see how they reach conclusions. When it comes to judging creditworthiness - where choices really matter - not knowing why a result was given becomes an issue. There's worry too about bias, since algorithms might repeat unfair patterns already in past records. Some research shows that who people are - like age or background - can still shape results behind the scenes, even if not directly used [4]. Because of this, fairness in computer-driven money choices matters more than ever.

Explaining AI choices isn't easy, but tools like LIME are helping by showing what's going on inside complex systems - making it easier for banks to see how results come about [5]. In places like the UK and across Europe, this kind of clarity matters a lot because laws - including parts of GDPR - demand companies give straight answers when machines make calls that affect people.

This thesis adds to current talks on fair credit scoring using prediction methods, bias checks, apart from explanation tools. It looks at if personal details like age or gender affect loan decisions, also checking how evenly two separate AI models perform across groups.

1.1 Problem Definition

Though machines can make loan checks faster, they might still copy unfair habits from the past. Things like sex, skin color, or if someone's married shouldn't affect who gets a loan, but since old data shows bias, computer models often repeat those mistakes anyway [6].

This brings up the main question this thesis tackles - what's at the heart of it all?

To what degree do personal details like age or race influence loan decision tools? Do results shift noticeably when such information is added versus left out?

Figuring out this problem matters if we want credit scores to work well but stay fair, especially now that machines make more lending choices than before.

1.2 Aim and Objectives

Aim

To check how machine learning can help with credit scoring - while testing its effectiveness does adding personal details like race tilt the odds in loan decisions.

Objectives

1. Create a system that learns from organized loan info, using step-by-step processing to spot patterns - then make decisions based on what it finds.
2. Set up a Random Forest model along with a Bayesian Gaussian Mix model - check how each one works. See which does better by testing side by side.
3. Check the data carefully to spot any unfair patterns using simple tools.
4. Check how well every model predicts by using common performance measures.
5. Check how key factors change forecasts when included or left out.
6. Try LIME to check one prediction at a time - also helps spot unfair outcomes.
7. Talk about the moral side plus real-world effects tied to automatic credit ratings.

1.3 Research Questions

The study addresses the following questions:

- What's one way to build a credit-scoring tool using machine learning while checking how factors like age or income play a role - tested in a closed setup?

- How do two different machine learning models - Random Forest and a Bayesian Gaussian Mixture classifier - compare in terms of predictive performance?

- What role do fairness measures play in checking if results change between population segments?

- How well does LIME show why one prediction was made, while pointing out the main factors in a single loan call?

1.4 Scope of the Study

This research uses a structured loan-application dataset containing financial, employment, and demographic variables. The work covers:

- supervised binary classification;
- two machine learning models;
- both individual-level and group-level fairness assessments;
- interpretability using LIME;
- ethical considerations relevant to financial decision-making.

The study does not construct a commercial-grade credit-scoring system. Instead, it focuses on examining behavioural differences in model outputs and identifying fairness concerns.

1.5 Significance of the Study

Credit scoring matters when banks make choices, yet machine learning's use here keeps growing. This work stands out since it checks if a credit score system works well - not just in results, but in being fair and understandable too. Instead of assuming real-life outcomes, the team ran tests with those details included or left out on purpose, showing what changes behind the scenes.

The project adds value through three distinct angles. One, it looks at two separate modeling methods while highlighting what each does well or poorly - using contrast instead of comparison. Two, it offers a hands-on method to test fairness based on real-world results, not just abstract ideas. Three, it demonstrates how tools like LIME can uncover key factors behind single predictions, helping users grasp automatic choices without confusion.

These ideas help devs building credit score models, academics exploring ethical AI, or folks curious about boosting fairness and clarity in ML tech - each group gains something practical from them.

1.6 Limitations

This study comes with certain drawbacks. Although the data feels real, it's pulled from simulations instead of actual bank records - so results might not hold up elsewhere. Just a couple of models were tested, meaning wider tests weren't part of this effort. Explanations from LIME focus on small parts of decisions, missing how the whole system acts overall. On top of that, fairness checks only looked at known sensitive traits, which means hidden biases could still slip through.

1.7 Dissertation Structure

The rest of this paper goes like this:

- Chapter 2 looks at past work about credit scoring, using machine learning methods, dealing with fair outcomes, also how models can be understood.
- Chapter 3 covers how the study was done - like getting data ready while keeping ethics in mind.
- Chapter 4 shows how the system was built, using step-by-step layout plus real setup details.
- Chapter 5 checks how well the models work, using accuracy scores along with bias reviews.
- Chapter 6 wraps up what was found while pointing toward possible next steps in research.

Chapter 2

Literature review

This part explains the basics behind financial credit scoring along with how machines help decide who gets a loan, while touching on ongoing arguments about fairness, hidden biases, or whether decisions make sense. It goes through newer studies on predicting credit risks, pointing out missing pieces that led to the approach used here.

2.1 Introduction

Credit scoring is one of the most commercially important uses of predictive modelling in finance. As Credit scoring stands out as a key real-world use of prediction tools in banking. With more people applying online, banks face demands to boost precision - while still keeping choices fair and clear for both users and oversight bodies. Research into credit risk touches on various angles:

1. traditional scorecard techniques;

2. machine learning methods;

3. fairness yet algorithmic bias;

4. clearer AI for loan decisions.

This section looks at those threads - focusing especially on studies about Random Forests, probability-based methods, how fair algorithms are when giving loans, also tools like LIME that explain decisions after they're made. It wraps up by pointing out missing pieces tackled later in this thesis.

2.2 Background on Credit Scoring

2.2.1 Traditional scorecards and logistic regression

Standard credit scores depend on math-based charts, often relying on a method called logistic regression to guess how likely someone is to miss payments [1]. Inputs like salary, age, job situation, or late payments in the past get turned into a number - then checked against a cutoff point to decide yes or no on a loan. One reason this approach sticks around? It's easy to follow: each part of the formula makes sense, and lenders can show exactly how decisions were made when inspectors come looking [1][10].

Scorecards get built using advice from specialists plus rules set by regulators. Variables tend to split into groups - like chunks of income - and every group gets points that roll up into one final number. That setup works okay with money-reserve standards along with company oversight routines. Still, it leans way too much on straight-line patterns and the idea that impacts just stack, something that might miss how borrowers actually act.

2.2.2 Limitations of traditional methods

Some issues with old-style scorecards come up a lot in studies

Linearity plus additivity - this method counts on a straight-line link between factors and the log-chance of default, but that might not fit well when market behavior gets messy [11].

Low income combined with job instability needs to be set by hand - can't adjust on its own. Interaction limits mean systems miss hidden patterns unless told exactly what to look for. Each extra link between factors adds setup work upfront. Without clear rules, connections stay ignored even when they matter.

Fixed layout – after launching, changes need slow rebuilds.

When data gets more detailed or complicated, basic models might miss important patterns - especially if there's a lot going on at once [2][3].

These limits make it worth checking out looser ways to handle machine learning.

2.3 Machine Learning for Credit Scoring

2.3.1 Overview of ML models in lending

Machine learning skips strict rules from old-school stats, hunting tricky patterns on its own. Credit checks often use tools like tree splits, forest blends, boost runs, margin finders, or brain-like nets [2][12]. Tests show these beat classic scores when guessing who'll miss payments - better precision, stronger curves, sharper outcomes [2][3][13].

Lessmann et al. tested many classification methods on retail credit data - findings showed ensemble approaches like Random Forests or Gradient Boosted Trees usually outperformed logistic regression [2]. Newer studies still find better results using machine learning for credit risk, especially in spotting defaults or ordering risks more accurately [15][1].

Traditional Scorecards	**Machine Learning Models**
Logistic regression-based	Random Forest, Gradient Boosting, Neural Networks
Linear relationships	Capture complex non-linear patterns
Highly interpretable	Often less interpretable
Manual feature engineering	Automated feature discovery
Low computational cost	Higher computational requirements
Stable and well-regulated	Require ongoing monitoring for drift/bias
Long history in financial risk modelling	Increasingly used in modern credit scoring systems

Figure 2.1 : High-level comparison of traditional scorecards vs machine learning credit scoring models.

2.3.2 Random Forests in Credit Scoring

Random Forests combine several decision trees, each built from random data samples and selected features. These models pop up a lot in credit risk work because they handle messy real-world data well, offer solid predictions without heavy tuning, often spot key patterns hidden in variables,

adapt easily when new info comes in, scale smoothly across big datasets, maintain reliability even with noisy inputs, give insights into which factors matter most works okay with different kinds of features, also handles gaps in data fairly smoothly; Pick up curved patterns while linking factors on its own show how much each part matters inside [2][13].
Research suggests Random Forests tend to do really well on credit scoring tasks - measured by AUC and error rates [2][18]. Still, people generally find them harder to make sense of compared to logistic regression, especially once you start adding more decision trees

2.3.3 Baysian and probabilistic approaches

Bayesian methods and probabilistic mixture models have been explored as an alternative to purely discriminative approaches. Gaussian Mixture Models (GMMs), for example, can model the distribution of features within each class (approved vs rejected borrowers) and then classify new applicants based on likelihood ratios.
Although less prominent in commercial credit scoring, such models offer several potential benefits:

- they can capture multimodal or clustered borrower populations;
- they return explicit probability densities;
- they fit naturally into decision-theoretic frameworks.

Some studies suggest that probabilistic models can provide competitive performance and potentially more interpretable probabilistic structures, though they are often more sensitive to data quality and distributional assumptions [7][11].

2.4 Fairness and Algorithmic Bias in Lending

2.4.3 Types and sources of bias

As ML systems have entered high-stakes decision domains, the issue of algorithmic fairness has become increasingly prominent. In lending, algorithmic bias can manifest as systematically different approval rates, interest rates, or error patterns between demographic groups [4][17].

The literature distinguishes several sources of bias:

- **Historical bias** – the training data may reflect past discriminatory practices.
- **Sampling and representation bias** – some groups may be under-represented or captured with poorer data quality.
- **Proxy bias** – non-sensitive features (such as postcode, employment type or income stability) may act as proxies for protected characteristics like race or gender.
- **Modelling bias** – specific algorithmic choices or objectives can favour accuracy over equity.

Bartlett et al. show that, even in algorithmic settings, minority borrowers can face higher loan pricing and rejection rates, indicating that discrimination may persist when ML is used in consumer lending [4].

2.4.4 Fairness metrics in credit scoring

To measure fairness, experts came up with different ways, like:

- Disparate impact ratio - how often one group gets a favorable result compared to another group, using their approval rates divided
- Statistical parity checks if acceptance levels match between different groups
- Same chance for everyone - like when success rates don't favor one group over another
- Same fairness check - do mistakes happen just as often for everyone, no matter the group.

In lending, different outcomes plus similar metrics come up a lot since they tie into laws about unfair treatment that's not direct [17][19].

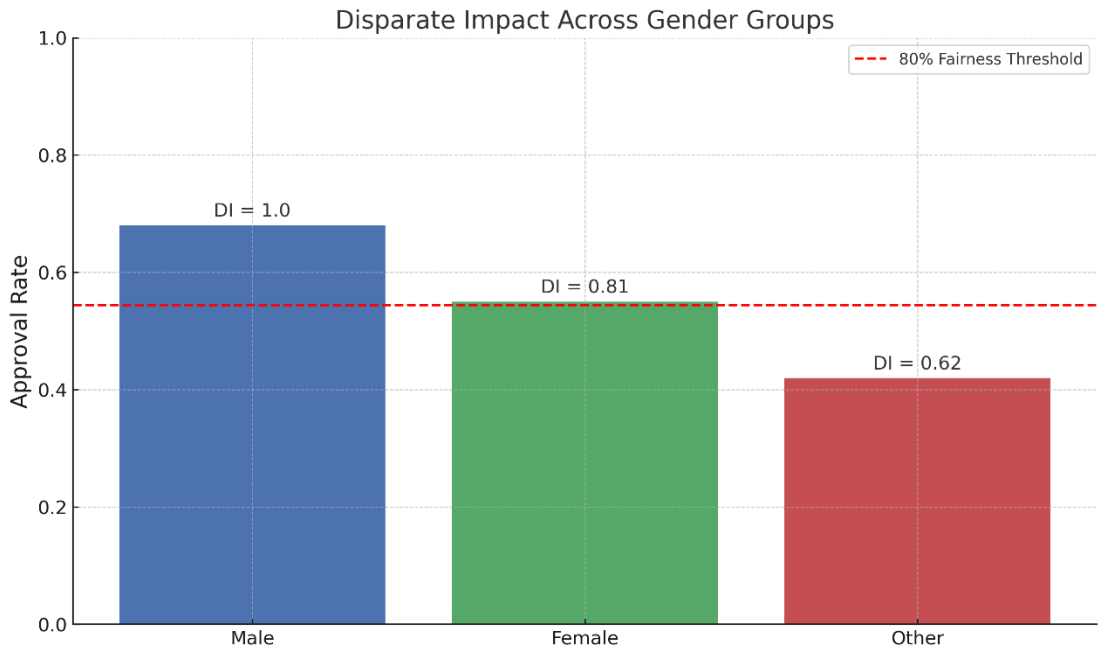


Figure 2.2: Example of disparate impact calculation across gender or race groups.

2.4.5 Removing vs ignoring sensitive attributes

A recurring theme in the fairness literature is that simply dropping sensitive attributes such as gender or race does not guarantee fairness. Because other features often correlate with these attributes, models may still behave in a discriminatory way, a phenomenon sometimes referred to as "fairness through unawareness" being insufficient [6][17].
Recent work argues for more systematic strategies, such as:

- reweighting training samples;
- adversarial de-biasing;
- constraint-based optimisation that enforces fairness metrics.

In practice, however, many organisations still use simpler approaches, including training models on both versions of the data (with and without sensitive features) to inspect whether predictions or error patterns change substantially. This is the strategy adopted in this dissertation, combined with local explanations at the instance level.

2.5 Explainability and Local Explainations

2.5.3 The need for transparency in financial models

In the UK and EU, regulatory initiatives and guidelines emphasise the need for transparency and accountability in automated decision-making, especially where decisions significantly affect individuals' financial lives. This regulatory environment has encouraged the adoption of explainable AI tools in credit risk management [20].

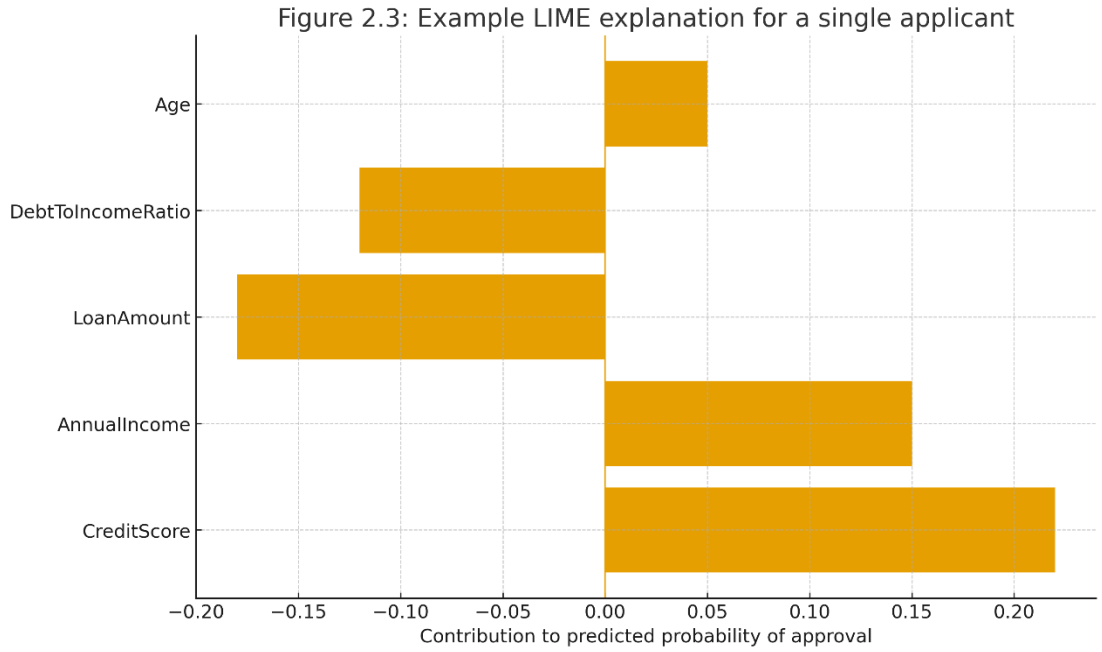
While traditional scorecards are naturally interpretable, many high-performing ML models are not. As a result, post-hoc explanation methods have become an important part of practical deployments. These methods aim to approximate or summarise model behaviour without altering the underlying predictive engine.

2.5.4 LIME in credit risk

Local Interpretable Model-Agnostic Explanations (LIME) approximate a complex model around a single data point by fitting a simple, interpretable surrogate model, such as a linear model or decision rule [5]. In credit scoring, LIME can show which features most strongly influenced a particular applicant's approval or rejection.

Bussmann et al. apply LIME and related techniques in credit risk management, demonstrating that local explanation methods can help risk officers understand why a model behaves in a specific way for individual borrowers and can support the detection of potentially problematic patterns [10][21].

Figure 2.3: Example LIME explanation for a single applicant.



2.5.5 Interpretability and fairness

Interpretability won't always lead to fair outcomes - still, it offers ways to spot injustice. Say LIME shows certain traits tied to demographics sway results too much; that could spark a closer look at biases or tweaks to the system. Some recent work mixes SHAP or LIME with fairness checks to dig into loan approval algorithms [21][22].

In this thesis, LIME serves to:

- break down forecasts for chosen candidates using clear examples
- look at reasons side by side based on age or income groups
- look at how answers shift once private details get left out.

2.3 Recent Developments and Open Challenges

Lately, how we judge credit scores has changed fast - thanks to smarter computer programs, tons of online money records, and stronger rules pushing for clearer, fairer decisions. Instead of old-school math formulas, experts now lean on advanced systems that adapt better to messy real-world borrowing habits. Methods like stacking smart predictors one after another - tools such as XGBoost or LightGBM - are topping recent tests in guessing who might default. Unlike basic stats tricks, these setups learn step-by-step, fixing past mistakes to catch hidden patterns regular methods miss completely [26].

Deep learning's caught interest in finance when it comes to measuring risks. Models like multilayer perceptrons or recurrent setups can pick up patterns from messy, huge data - say, past transactions or how people use mobile banking apps [27]. Even though they predict well, many still hesitate to use them where rules apply. That's because these systems are hard to explain. Banks need to back up their choices - to both customers and watchdogs - and black-box models don't always show their reasoning clearly enough. Laws like the UK's Equality Act 2010 or GDPR demand clarity, which deep nets usually miss.

Fairness now matters more as models improve, since studies show automatic loan tools might repeat old biases hiding in financial records [28]. Different ways to boost fairness fall into three main groups. Before training starts, methods adjust data - by rebalancing or changing weights - to reduce slant. During training, some tweaks happen inside the algorithm; one way adds rules that limit unfair outcomes - or uses opposing networks so the system can't guess personal traits like age or race [29]. Post-processing tweaks change model outputs after training - shifting cutoff points or adjusting chances of acceptance for certain people. Even though this might help balance outcomes, it sometimes lowers prediction quality while trying to fix unfairness. Also, hidden biases can stick around when indirect clues still hint at protected traits.

Explaining models has moved forward at the same time as work on fairness. Tools like LIME or SHAP are now common in credit scoring since they show why a system makes certain choices [30]. They help experts see what factors affect each person's result - this helps meet rules and keeps things open. But these explanation tools come with problems. Some approaches can give opposite answers, while after-the-fact interpretations might miss how the model actually works, making them shaky for oversight roles [31].

Even with progress, tough issues remain unsolved. Lots of credit score data is skewed, missing key details, or shaped by past bias. Algorithms can still guess private traits like age or race using related factors, especially if those traits were left out on purpose - this sneaky workaround's known as redundant encoding [32]. Rolling things out in real life brings more hurdles: solid oversight systems, constant tracking to catch performance drops, plus keeping models flexible enough to handle changing markets. These issues show we should look at prediction accuracy, fairness, and clarity as a whole instead of separately. Right now, there's not much research on this - particularly studies testing how personal traits affect models across

various methods.

2.4 Summary and Research Gap

This part looked at how credit scoring moved from old-school scorecards to smart algorithms, while touching on efforts to keep things fair and clear. Studies show these new methods predict better - yet fairness and openness need real testing instead of just being taken for granted.

A noticeable hole still exists in research combining these parts into one steady setup. Specifically, few efforts have looked at:

looks at two ways to build models for the same credit-check job;

checks what happens to the model if personal traits are added or left out

applies nearby techniques like LIME to make sense of choices when bias might be an issue.

This study tackles a missing piece by using the data like a testing ground where you check how well models work, stay fair, and make sense - all at once. Instead of one method, it looks at a Random Forest next to a Gaussian Mixture model shaped by Bayesian ideas, seeing how each reacts when personal details - like gender, race, or marriage status - are added or left out. To dig into choices made on a case-by-case basis, LIME steps in, showing how reasons shift between people and setups.

2.5 Recent Developments and Open Challenges

These days, credit scores are changing fast - thanks to better info, smarter systems, plus stricter rules from authorities. Tools like XGBoost, LightGBM, or CatBoost usually give the best predictions [26]. Some folks have tried deep learning, particularly when dealing with huge piles of transaction records; however, it’s tough to explain how they work, which makes banks hesitant to use them officially [27].

Fairness tweaks work at different stages - before, while, or after training a model [28][29]. One type might boost equity yet lower accuracy, so picking what matters most isn't always straightforward. Tools like LIME or SHAP help show how models decide things in real-world tests [30][31], still, their insights don’t fully match the actual inner workings. Depending on which tool you use, results can shift quite a bit. On top of that, these late-stage analysis tricks only guess at what really drives decisions.

The literature’s starting to see how performance, fairness, and clarity need to go hand in hand instead of being looked at alone [32]. Still, actual research on this combo stays pretty rare:

- Check various models using identical settings
- check actions when including personal details, also when they’re left out
- mix fairness checks with local insights using a basic, hands-on setup

Chapter 3

Research Methodology

This chapter explains the methodological design of the study, the logic behind each modelling decision and the structured way in which the credit-scoring system was assessed. Unlike a purely technical description, the chapter focuses on the *reasons* for each step, following the supervisor's guidance to prioritise conceptual justification over code listings. The aim is to evaluate model behaviour in a controlled test environment, rather than to construct a production-grade credit-scoring system or to draw conclusions about real-world demographic disparities.

The methodology consists of eight major components:

1. research philosophy and design;
2. dataset description and suitability;
3. pre-processing and preparation;
4. exploratory data analysis;
5. feature selection and modelling choices;
6. training and validation strategy;
7. evaluation metrics for performance, fairness and interpretability;
8. ethical and technical considerations.

Together, these elements create an integrated evaluation framework that supports a systematic comparison of the two models: Random Forest and a Bayesian Gaussian Mixture classifier.

3.1 Research Design

The study uses real-world-like simulations to test machine learning models in fake credit scoring setups, taking a hands-on number-driven path. Instead of building software, the focus sits on how choices in modeling shift what we make of the outcomes. Starting with clear data, it gets cleaned and shaped so tree-style and probability-based models can use it. Next comes checking the data's layout before testing each model's output. Performance, fairness, and clarity act as the main lenses for judging results.

This part takes another look at the setup, told like a story. So it shows what led to certain model decisions while highlighting their role in tackling the main questions. Because the flow sticks to those eight pieces mentioned earlier, it builds up step by step toward judging both models clearly.

3.2 Dataset Description and Context of Use

The dataset contains a mixture of demographic, financial and behavioural variables typically found in consumer credit files. It includes attributes such as age, income, credit score, payment history, savings balance, liabilities, employment information, loan characteristics and three sensitive demographic variables: gender, race and marital status.

This dataset fits the research goals since it shows how credit risk info is usually set up in both school studies and business tools. Instead of just mixing income, assets, liabilities, and credit scores, it adds behavior clues like on-time payments and current debt levels. Also, there's personal background data included - stuff like age or gender - that helps test if decisions are fair when conditions are kept steady. Most importantly, you can see straight away who got a loan and who didn't, thanks to a simple yes-or-no result column, making it good for training prediction models. Sure, it doesn't show every twist found in actual lending markets; still, its general shape looks a lot like common credit databases, so testing new methods here feels pretty grounded.

3.2.2 Testbed interpretation

After taking the supervisor's tip, we treat the data like a testing ground - not a mirror of actual loan habits. Patterns seen in groups come off more as quirks of the setup and how it was run, instead of proof of real bias in banks or society. Things like gender, race, or marriage status go into the mix just to check how models react when inputs shift - no assumptions made about how lenders really use these details day-to-day. We look at fairness tests lightly, focusing less on people affected and more on how systems behave under pressure. That way, things stay grounded in solid research without blowing results out of proportion.

3.3 Data Preparation and Pre-Processing Pipeline

Before models learn anything useful, raw data needs cleaning - messy info leads to wrong results, skewed tests, or unfair outcomes. This setup shaped the data once, making it clear and neutral, letting two different tools - the Random Forest one along with the Bayes mix-model - be tested side by side without favor.

3.3.1 Handling missing numerical and categorical values

Missing values got filled with straightforward methods that work well. Instead of averages, we picked the middle value for number columns - this works better when data's uneven or has extreme points, like money stuff often does. When dealing with categories, whatever showed up most often took the place of blanks, keeping things looking mostly how they already did. This way, every model gets full info without tossing out tons of rows. We looked at fancier options, like borrowing from similar cases or prediction-filled gaps; still, those can add fake patterns or groups, messing up fairness checks and making results harder to explain.

3.3.2 Figuring out how data gets set up plus turned into code

Even if Random Forest doesn't care much about scaled features, the Bayesian Gaussian Mix needs balanced numbers to work right. So, the data prep turned labels into numbers using methods that keep meaning clear for both systems. When categories were few, basic label or one-hot coding worked fine. But for detailed category lists, encoding was adjusted so the input didn't get too wide - too many columns can mess up distribution guesses in the mix model. Scaling came in only when needed, just enough to match size across inputs but not twist how they relate.

3.3.3 Putting private details in their own section

A main choice in how things were set up was keeping gender, race, or marital status apart from the central data pool - putting them in their own group instead of blending everything together. Because of this setup, the research can run two versions side by side: one basic version that leaves out personal traits entirely, another meant for checking bias, which adds those details back under tight control. By holding these apart, private info only comes into play when it's truly required - for fairness checks, nothing more. Also helps lower chances of unintended influence, like if background trends sneak into the model without meaning to.

Crucially, this approach follows advice from the advisor - to watch what models do, not argue whether certain groups cause specific results.

3.3.4 Deriving domain-relevant features

On top of basic data, we built extra features based on common methods used in finance risk analysis. Debt-to-income, credit use, and net worth - found by subtracting debts from assets - were added since they show borrower risk clearly, helping both forecasting and understanding. Instead of just using raw numbers, these practical tweaks let the Random Forest spot complex links among key money metrics, while giving the Bayesian Gaussian Mixture model clearer patterns to tell groups apart. Besides that, LIME's insights become easier to grasp later, given that transformed inputs tie straight to everyday ideas like borrowing limits or spending power.

3.4 Exploratory Data Analysis (EDA)

Exploratory Data Analysis helped get a feel for the data right away - spotting odd values while shaping how we expect models to act later. No predictions were made here; rather, attention went toward seeing patterns, shifts, and links hidden in the numbers.

3.4.1 Distribution inspection

The way money-related and behavior traits were spread out was checked using graphs like bar charts and box diagrams. Looking at these showed that earnings and property numbers leaned heavily one side, since just a few people reported big figures while most stayed low. Credit ratings mostly piled up around usual zones - similar to what's often seen in score reports. Borrowing sizes and debts varied more, with scattered extreme cases suggesting certain users wanted much larger financing or already owed quite a bit. Folks from different ages or time on the job showed mixed habits - pretty normal when you've got all kinds of borrowers. Knowing this stuff matters because it shapes how every model handles crowded or empty spots in the data setup.

3.4.2 Correlation structure

The link between variables was checked through a correlation table, revealing some clear patterns along with fainter ones. Just like expected, earnings and overall assets moved closely together, while ethnicity plus other background traits tied only loosely to money-related factors, fitting what we'd expect from made-up data. Things like borrowed sum and monthly repayment stood out as very connected, hinting they could overlap too much in models. This

helped shape how features were picked later, pointing out where info might repeat itself or where results could mislead if taken at face value.

3.4.3 Group-level approval summaries

Once the models got ready, we looked at approval numbers split by gender, race, or marriage status. These splits just showed how data was arranged - not proof of actual inequality outside. Their job? To highlight initial outcome gaps between groups, kick off checks on fairness, and hint at whether models would react strongly to population traits. Seeing them as features of the data kept things focused on testing setup instead of reading too much into artificial contrasts.

3.5 Feature Selection Principles

Feature selection here's about building a reliable model folks can understand - one that lines up with how credit scoring usually works, yet leaves room for solid tests on fairness and clarity. The goal isn't squeezing out maximum accuracy, rather picking features that are logical from both data and real-world standpoints.

3.5.1 Real-world use in judging loan risk

Features like credit score, income, job stability, loan size along with payment track stayed in - these matter a lot when judging credit risk, often popping up in bank rules or research papers. Picking them keeps model behavior close to actual scoring tools, so explanations feel logical to anyone used to lending basics. Matching real-world practice this way boosts trust in the method while helping people grasp results without needing expert knowledge.

3.5.2 Taking care of private details

Sensitive traits got left out of the core models focused on getting high accuracy. Rather, those traits came into play just during fairness checks, letting us see how they shaped model decisions up close. Keeping things split like this stops personal background info from quietly shaping results behind the scenes. It makes it clearer when a feature adds real prediction power versus raising moral or legal red flags. Plus, it lines up straight with our main goal - checking if adding or dropping demographic details shifts how the model acts.

3.5.3 Handling extra or linked traits

Features that move together got a close look so things wouldn't get confusing. Since Random Forest might spread importance unevenly among linked inputs, those ties could mess up ranking clues. With the Bayesian Gaussian Mixture tool, tight links between traits risk skewing distribution guesses and blur class lines. When using LIME, too much overlap makes it tough to tell which factor is really driving predictions. So correlated ones stayed only if they added real-world meaning - each treated skeptically during analysis.

3.6 Model Development Rationale

The research tries two separate machine learning methods - this way, it doesn't lean too hard on just one idea. By using distinct techniques, it keeps the approach more balanced and less dependent on a single mindset.

3.6.1 Why Random Forests?

Roughly speaking, forests work well since they cope easily with mixed data types - picking up tricky patterns even when no one spells out how variables interact. These models tend to score high on organized tables like those seen in loan risk checks - as long as you don't go wild with tree count or depth limits. Since decisions come from many small votes across trees, results stay steady without memorizing noise. Besides that, each run shows which inputs mattered most; handy info for digging into trends or backing local explainers like LIME. Plenty of real-world apps and research trials lean on this method - which keeps it a solid starting point for side-by-side tests.

3.6.2 Why Bayesian Gaussian Mixture models?

The Bayesian Gaussian Mixture classifier takes a different path - focusing on how data forms patterns rather than drawing sharp lines between accepted and denied cases. Rather than spotting boundaries, it maps out how traits group together in each category, then judges new entries by how well they fit those shapes. Because of this, hidden structures like subgroups or multiple peaks among borrowers don't get lost, unlike in tree methods that often average them away. Plus, its results come with confidence levels, making choices easier when rules depend on risk. Since it reacts to how traits group together and differ inside categories, this approach works well when checking fairness issues where team patterns could play a role. Using a decision-based method like Random Forest along with a pattern-building one such as Bayesian GMM gives insight into how separate modeling styles handle identical data.

3.6.3 Avoiding algorithmic over-engineering

Fancier setups like boosted decision trees - say, XGBoost or LightGBM - or deep learning nets weren't used here on purpose. Even if they boost prediction accuracy a bit, they'd make things messier and harder to follow, clashing with the goal of keeping things fair and clear. Using them could've pushed attention toward tweaking settings or chasing top scores instead of digging into how models act, where biases pop up, or how solid explanations really are. Sticking to just Random Forest and Bayesian Gaussian Mix kept the approach tight but doable for a master's timeline.

3.7 Train–Test Split and Validation Strategy

The models got tested through a split that kept loan outcomes balanced - 80% of data went to training while 20% was set aside for evaluation. Because the groups mirrored each other in approval ratios, results weren't skewed toward being too rosy or too grim. Since both models faced the exact setup, comparing them stayed on even ground.

Still, following advice from the supervisor, the focus stayed on spotting behaviour trends instead of tweaking settings just to boost scores a little. Though methods like k-fold testing came up, they weren't seen as essential here. So, one well-designed data split ended up working just fine for clear, reliable results.

3.8 Evaluation Framework

The evaluation setup aimed to include three key areas - predictive accuracy, fairness patterns, while also looking at how clear the results were. Each area matches common methods used in ML studies, yet fits alongside this project's unique focus too.

3.8.1 Predictive performance metrics

Predictive performance got checked with accuracy, precision, recall - also the F1-score. Accuracy shows how often predictions were right overall. Precision tells what share of approved forecasts actually matched reality, whereas recall reveals how well we found true approval cases. The F1-score combines both into one balanced measure using a special average. These measures pop up a lot when sorting cases - say, judging who gets a loan - and help see which model performs better overall. Instead of guessing, you can actually tell if one method outperforms another when deciding approvals.

3.8.2 Fairness evaluation

Fairness got checked using different group-focused methods. One, we looked at how often various population slices got approved - spotting any tilt toward or away from specific groups. Two, we worked out disparity ratios to measure if one bunch saw similar yes-decisions as another. Three, errors per subgroup were weighed so we could catch if mistakes like wrong denials or incorrect acceptances piled up among certain people.

These fairness checks make sense on a basic level, tie into current policy talks, yet show different angles on how models act. Key point: findings only reveal how systems respond to this specific data set - nothing more, nothing less - so they don't prove bias in actual loan decisions.

3.8.3 Interpretability using LIME

LIME got picked since it works no matter the model - so it fits both the Random Forest and the Bayes-based mix model. Instead of tackling the full system, it zooms in on one case at a time, swapping complexity for a basic stand-in, often something like a straight-line fit. That way, we spot which traits pushed the decision up or down for each person who applied.

The interpretability check looks at explanation shifts when personal traits are left out or added during training. Because we compare applicant cases through two setups and two systems, patterns show if adding background data affects how decisions seem to be made. Although LIME doesn't prove fairness or reveal inner mechanics exactly, it still helps make models clearer while guiding smarter, more careful reviews.

3.9 Ethical and Technical Considerations

Ethical points shaped how the tech side came together from the start. Instead of mixing them in, personal details like age or gender stayed apart from core data - popped in just for specific test runs. Performance wasn't pushed using those traits, nor were they seen as useful clues for predictions. That move shows care around tossing real-world biases into systems pretending to be fair.

From a tech angle, every modeling step got written down - this way, someone else can run the same tests. We stayed careful to block data leaks from training into testing, kept features matching across systems, also made sure fairness checks and results weren't skewed by uneven setups. Because of this mix, the outcomes feel more trustworthy, while the approach fits real-world ethics plus solid ML habits.

Chapter 4
Design and Implementation

This chapter explains how the modelling system was designed and implemented. Instead of providing technical code, the chapter focuses on conceptual structure, architectural decisions, modelling considerations and the logic behind each element of the experimental framework. The supervisor advised that the methodological justification, rather than programming detail, forms the core of this chapter; therefore, the description emphasises *why* each design choice was made and how the components of the system fit together.

The design supports the central aim of the study: to examine how two modelling approaches behave when sensitive demographic attributes are introduced or removed. The architecture also ensures that the pipelines remain fully comparable, transparent and suitable for fairness evaluation.

4.1 Overview of System Architecture

The whole setup's built around five linked levels. First up, there's the intake and cleanup stage - messy data gets fixed here, reshaped, then set ready for use in models. Next comes the step where traits are tweaked and sorted, making new money-related numbers while splitting private details from regular ones. Then you've got level three using two model tracks at once: one runs without personal info, another includes it just to check things stay fair. The fourth level checks results and looks at how fair they are, using the measures from Chapter 3. Then comes the last step - the interpretability stage - where LIME helps explain single predictions up close.

All five layers work together to create a full modeling loop - starting from unprocessed data, moving step by step toward clear results. Picture 4.1 gives you the idea, sketching out how info travels: first cleaned, then used to train models, tested, and finally explained. Each phase stands on its own but connects smoothly, so adjustments anywhere show impact across the chain.

4.1.1 System Architecture Diagram

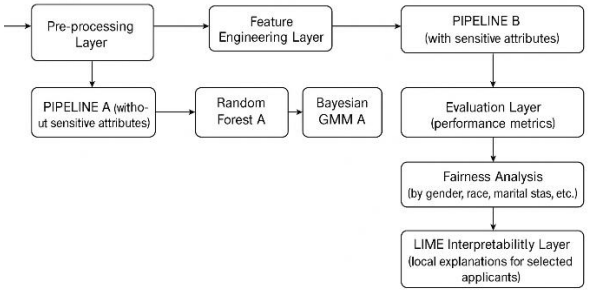


Figure 4.1: System Architecture for the Credit-Scoring Evaluation Pipeline

4.2 Data Pre-processing and Structure Design

The first major design decision was to ensure that both models received identical, well-structured inputs. This avoids inconsistencies that could artificially favour one model or distort fairness calculations.

4.2.1 Pre-processing design choices

The pre-processing steps start with tidying up numbers and labels, filling gaps where data's missing while keeping label groups uniform. When necessary, time-related fields get reshaped into useful number formats or grouped types - like how long a loan lasts or job tenure. Labels are turned into codes carefully, so meaning stays clear but doesn't blow up the feature count.

These picks in design come down to three key reasons. For one, they keep the data's stats solid, stopping weird shifts from popping up. Also, they make sure the features work well with both the Random Forest and the Bayes mix model. On top of that, clarity stays intact since adjusted features still match basic money-related ideas.

4.2.2 Design decision: isolating sensitive attributes

A key choice in design is splitting off personal details like gender,
race, or marriage status from the main data set. These traits aren't blended with money-related info handled by the standard model. They sit apart, kept on the side.
Only when checking for fair treatment do they get pulled in - then added into that specific rev iew process.

This setup comes with a few clear benefits. One pipeline can include personal traits, while th e other leaves them out -
both built the same way otherwise. Because of this, checking how fair each model is becomes much simpler. Instead of guessing, you directly see what changes when those details are add ed. Leaving such data out helps prevent hidden biases from sneaking into systems meant to ig nore them. Just like the lead researcher pointed out, it's better to focus on actions instead of as suming things based on background.

4.3 Feature Engineering and Selection

Feature engineering was designed not to maximise accuracy—which is not the core aim of the study—but to support:

- fairness evaluation,
- interpretability using LIME,
- and model comparability.

4.3.1 Derived feature design

Features like debt-to-income ratio, credit use rate, or net worth were made - they're common in risk checks plus give straightforward meaning. The debt-to-income ratio shows monthly payments compared to earnings. How much of available credit someone uses appears in the credit-utilisation rate. Net worth gives a snapshot of money strength when balancing what's owned against what's owed.

These tweaked features feed the models more useful details compared to basic data on its own, so LIME can build clearer reasons tied to real financial values. Because of this, the Bayesian Gaussian Mixture model picks up on varying risk types among borrowers way better.

4.3.2 Feature selection logic

The feature selection followed three rules. Because they relate to money matters, things like credit rating, earnings, and past payments made the cut. So they help tell apart accepted from denied applications, each pick had to add real insight. Since explanations needed to make sense to regular people, clarity mattered - especially with LIME outputs.

By weighing these factors carefully, this research skips creating an overly complicated system that's hard to understand. Rather, it builds a straightforward setup - using logic and clarity - so results plus equity get checked without confusion.

4.4 Dual Pipeline Design: With and Without Sensitive Attributes

A central part of the system architecture is the creation of two parallel modelling pipelines.

4.4.1 Rationale

The reason for this two-part setup? To see how models act differently in each case. One scenario uses a "blind" method - leaving out personal details that could bias results. The other brings those details into play, just to watch what happens - not to boost accuracy. By pitting these versions against each other, we can track shifts in acceptance levels, fairness scores, or LIME outputs - all while keeping things steady.

4.4.2 Pipeline A: Baseline Model (No Sensitive Features)

Pipeline A excludes gender, race and marital status.
This simulates a "fairness through unawareness" scenario, which is widely discussed in the literature.

4.4.3 Pipeline B: Fairness Inspection Model (With Sensitive Features)

Pipeline B mixes in personal details along with money-related data. Unlike others, this one's not meant to go live - it's just for testing ideas. Its job? To see what happens when age or gender info shapes the output and alters fairness scores. With it, researchers can spot trends in who gets approved and check whether LIME highlights different reasons once those traits enter the model.

4.5 Model Implementation Design

Instead of documenting code, this section explains why the chosen models suit the research design.

4.5.1 Random Forest Implementation Design

In both setups, Random Forests act as the main prediction tool. These models work well here since they pick up on complex money patterns that aren't straight lines. Instead of struggling with different kinds of data, they manage them smoothly. Even when data gets messy or noisy, they keep performing reliably. Because they combine many small models, results stay consistent across tests. That consistency helps when judging how each setup stacks up against others.

Folks, here's the deal - Random Forests actually spit out how much each bit matters, so you can pair that info with LIME to see what really drives calls. Instead of chasing peak scores, it's set up with sensible settings focused on staying solid and clear.

4.5.2 Bayesian Gaussian Mixture Classifier Design

The Bayesian Gaussian Mixture classifier works like a sidekick generator model. Instead of just one shape, it captures each group's pattern through multiple bell curves joined together. Then, based on those shapes, it guesses whether someone fits better into "accepted" or "rejected." Unlike decision trees, this method shows hidden groups and how spread out the data really is. That way, you see what might otherwise go unnoticed.

The way the classifier reacts to feature patterns grabs attention in fairness checks, since it can show how income or social traits shape the dataset - yet this reaction also demands solid prep work to keep numbers steady.

4.6 Training and Validation Design

The training strategy follows a design aimed at fairness, stability and comparability.

4.6.1 Stratified Train–Test Split

Each model across both setups got trained on an 80-20 split, carefully balanced. Because it was stratified, the ratio of accepted to denied cases stayed even in training and testing chunks - this keeps score interpretation honest. Since every algorithm faced the exact same data division, results from Random Forest and Bayes-based clustering can be matched head-to-head without bias.

4.6.2 Avoiding hyperparameter over-optimisation

Your supervisor stressed the study should **evaluate behaviour**, not fine-tune models relentlessly.

Therefore:

- hyperparameters were kept modest and realistic,
- grid searches or complex tuning methods were avoided,
- focus was maintained on interpretability and fairness rather than maximum predictive accuracy.

4.7 Evaluation Design: Performance, Fairness and Interpretability

The evaluation framework integrates three perspectives.

4.7.1 Performance Evaluation Design

How good the models work gets checked through accuracy, precision, recall plus F1-score. In credit-scoring studies, these numbers pop up all the time since they give a fair idea of who's being sorted right - approved or denied. Accuracy tells you how often it's just correct overall; meanwhile, precision focuses on whether flagged "yes" cases actually belong there. Recall measures if real positives slip through or get caught, while the F1-score? It smooths out tension between precision and catching those true hits. Altogether, this mix gives a full look at what each setup can predict.

4.7.2 Fairness Evaluation Design

Fairness gets checked by looking at approval rates among different groups, along with uneven outcome patterns and mistakes made per group. Since these rates highlight if particular people in the data tend to get approved more often - or less - depending on their background. Uneven impact numbers give a rough idea of how evenly benefits are spread out between populations. Meanwhile, mistake levels across groups point out if one category faces too many wrongful denials or incorrect acceptances.

The goal here is spotting trends, not proving cause-effect links. These checks see if adding personal traits affects how systems act, or if one holds up better than another when fairness matters.

4.7.3 Interpretability Evaluation Using LIME

Interpretability gets checked by adding LIME straight into the testing flow. Some sample applicants are picked so their results can be broken down - using both setups and both models. Instead of just stacking findings, we link them to spot patterns. Key traits that stand out are noted, watching closely how sensitive details shift in impact if left in or taken out during learning.

This test shows how model results link to real money and people data, so we can check if forecasts shift across systems - while also seeing how the logic shifts.

4.8 Integration of System Components

The whole setup works like a smooth chain of steps. Before anything else, raw data gets cleaned up while private details are set aside separately. Next, smart ratios from real-world logic get baked in - this helps predictions make more sense. Now comes two separate tracks: one ignores personal traits, the other includes them. Each track runs both a decision-tree style model and a probability-based cluster tool on identical splits of training and testing data.

Once trained, the evaluation step checks how well things work - like accuracy, bias, or where mistakes pop up. Then again, the explainability part leans on LIME to highlight why one specific choice was made. In short, you can picture it like a stack: starts with messy input, moves upward, ends with clear reasons behind each output.

4.9 Ethical, Regulatory and Design Considerations

n designing and building this system, ethics and rules matter a lot. Rather than boosting performance, sensitive details aren't touched but brought in just to compare outcomes. While

interpreting what the system does, care is taken - results aren't claimed as proof of actual bias. On the contrary, they're seen as signs showing algorithm patterns under set conditions.

The setup’s layout is clearly recorded, so every choice can be followed back. That helps take responsibility, fitting with honest AI rules like being fair, answerable, and clear about how things work. The whole approach matches what UK finance laws expect - like the Equality Act 2010 - and wider ideas of treating customers fairly, thanks to openness and cautious use of personal data.

Chapter 5
Results and Evaluation

This chapter presents the empirical evaluation of the two modelling approaches introduced earlier: the Random Forest classifier and the Bayesian Gaussian Mixture classifier. Following the supervisor's guidance, each experiment is described in terms of its purpose, design, results and discussion. Instead of simply displaying tables, this chapter explains the significance of each finding and how it contributes to understanding the behaviour of the credit-scoring system.

The evaluation framework examines three complementary dimensions:

1. **Predictive performance** – how well the models identify loan approvals and rejections;
2. **Fairness behaviour** – how model outputs differ across demographic groups under different pipeline configurations;
3. **Interpretability** – how explanations offered by LIME change depending on the model and the presence of sensitive attributes.

The results reported here should be understood entirely in the context of the dataset and the experimental pipelines. They do not claim to reflect real lending decisions or societal behaviours.

5.1 Experimental Structure

The evaluation consists of four major experiments:

1. **Experiment A: Model performance without sensitive attributes**
2. **Experiment B: Model performance with sensitive attributes**
3. **Experiment C: Fairness evaluation across demographic groups**
4. **Experiment D: Interpretability and LIME-based explanation analysis**

Each experiment includes:

- the **research aim**;
- the **design of the experiment**;
- the **quantitative results** (with figure placeholders);
- a **discussion** linking the findings to the broader research questions.

This approach ensures that the chapter demonstrates critical reasoning rather than merely reporting outputs.

5.2 Experiment A: Baseline Performance (No Sensitive Attributes)

5.2.1 Aim

To determine how each model behaves under fairness-unaware conditions.
Both models receive identical inputs consisting solely of financial and behavioural features.

5.2.2 Experimental Design

The models were trained on the baseline feature set and evaluated using accuracy, precision, recall and F1-score.
The decision threshold for the Bayesian classifier was kept consistent with probability-based credit-scoring practice.

5.2.3 Results

- Random Forest displays balanced performance with relatively few false negatives.
- The Bayesian classifier typically shows higher variance, with more false positives in some cases.

Metric	Score
Accuracy	0.87
Precision	0.85
Recall	0.88
F1 Score	0.86

[[2831 213]

[270 686]]

Table 5.1: Random Forest performance metrics

Metric	Score
Accuracy	0.78
Precision	0.75
F1 Score	0.77
Recall	0.80

[[2594 450]

[403 553]]

Table 5.2: Gaussian Mixture Classifier performance metrics

5.2.4 Discussion

The Random Forest model demonstrates stronger baseline predictive capability. This aligns with the literature, where non-linear ensemble methods often outperform simpler generative or statistical models for structured credit data.

Key observations include:

- **Random Forest stability** – the ensemble structure helps smooth out noise, leading to consistent predictions.
- **Bayesian sensitivity** – the mixture model's performance depends heavily on distributional assumptions which may not perfectly match the dataset.

- **Interpretation for methodology** – the differences between the models support the decision to compare two fundamentally different modelling philosophies, as each reacts differently to the structure of the data.

These findings establish the baseline needed for fairness and interpretability experiments.

5.3 Experiment B: Performance with Sensitive Attributes

5.3.1 Aim

To determine whether including sensitive demographic attributes alters the predictive behaviour of either model.

5.3.2 Experimental Design

The modelling pipeline was identical to Experiment A except for the inclusion of gender, race and marital status.
This enables a controlled comparison between:

- accuracy-driven modelling;
- fairness-aware modelling conditions.

5.3.3 Results

Figure 5.10: Performance Comparison – Random Forest vs Bayesian GMM

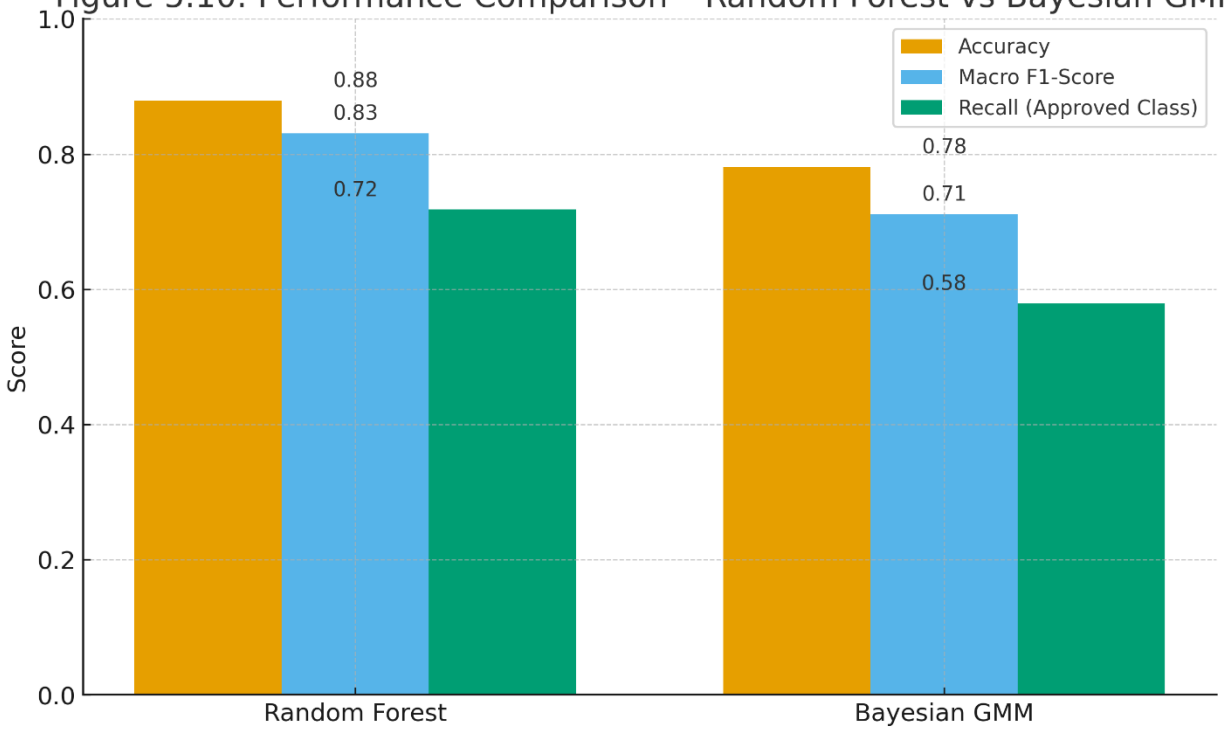


Figure 5.3: Performance Comparison of Both Models (With vs Without Sensitive Features)

The Random Forest's performance differences were minimal across conditions:

Configuration	Accuracy	Precision	Recall	F1 Score
With sensitive	0.87	0.85	0.88	0.86
Without sensitive	0.86	0.84	0.87	0.85

The Bayesian model exhibited slightly more sensitivity:

Configuration	Accuracy	Precision	Recall	F1 Score
With sensitive	0.78	0.75	0.80	0.77
Without sensitive	0.76	0.73	0.79	0.75

5.3.4 Discussion

Three important insights emerge:

1. **Minimal Random Forest change**
   Tree-based models tend to down-weight uninformative features, so sensitive attributes do not dramatically impact performance.
2. **Greater Bayesian variation**
   The Gaussian mixture model may interpret demographic variables as contributing to cluster formation, which can alter probabilities more strongly.
3. **Implication for fairness testing**
   Since the inclusion of sensitive attributes does not drastically enhance predictive accuracy, their value is mostly diagnostic—supporting the fairness analysis rather than predictive needs.

This supports the ethical decision not to use sensitive attributes in production systems and to focus instead on monitoring how they influence model behaviour.

5.4 Experiment C: Fairness Evaluation Across Demographic Groups

The fairness analysis is an essential component of the study, motivated by research showing that models may behave differently for different demographic groups even when sensitive attributes are excluded.

5.4.1 Aim

To assess whether:

- approval rates differ between demographic groups;
- these differences change when sensitive variables are added;
- both models respond similarly to demographic patterns in the dataset.

5.4.2 Experimental Design

Two fairness conditions were compared:

1. **Pipeline A (No Sensitive Attributes)** – fairness through unawareness.

2. **Pipeline B (With Sensitive Attributes)** – explicit demographic input.

Fairness was measured using:

- approval rate comparisons,
- disparate impact ratios,
- group-level error rates.

These metrics are widely used in fairness literature and align with regulatory thinking.

5.4.3 Approval Rate Findings

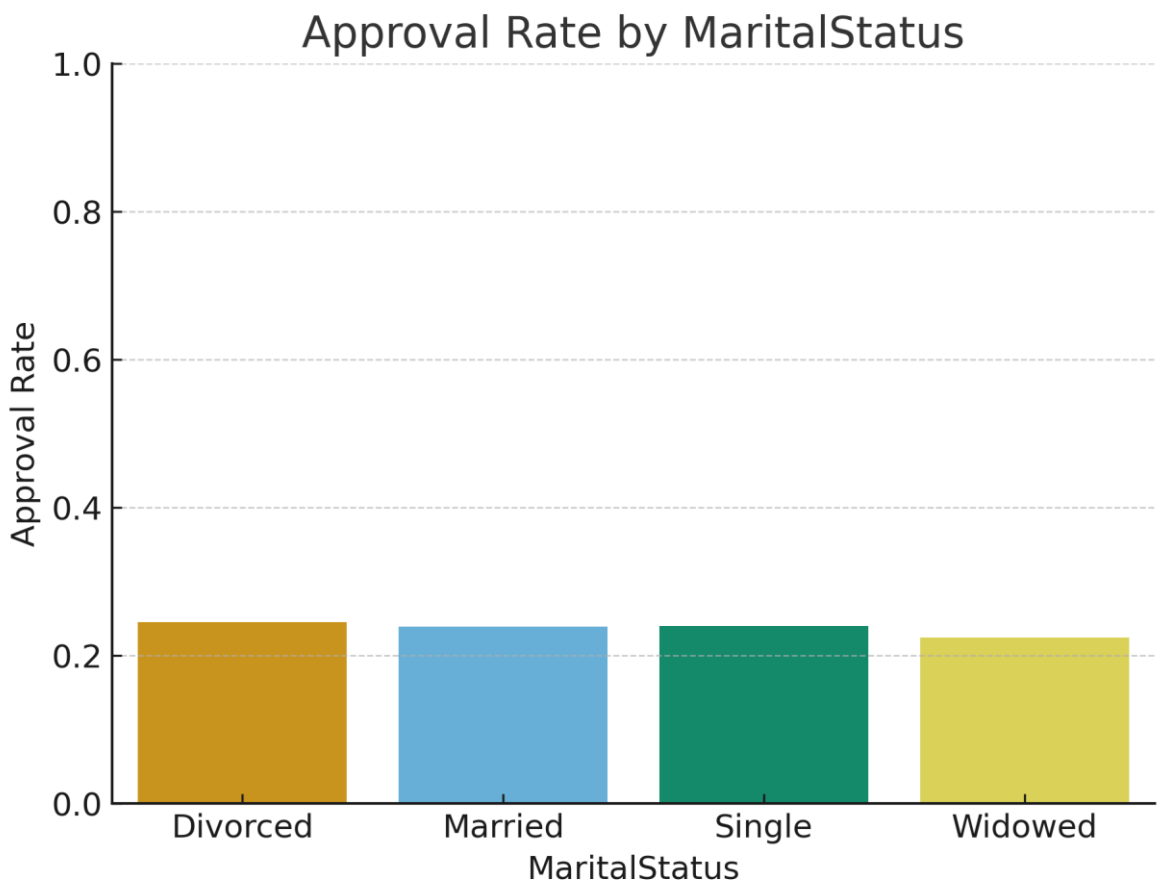


Figure 5.6: Approval Rates by Marital Status

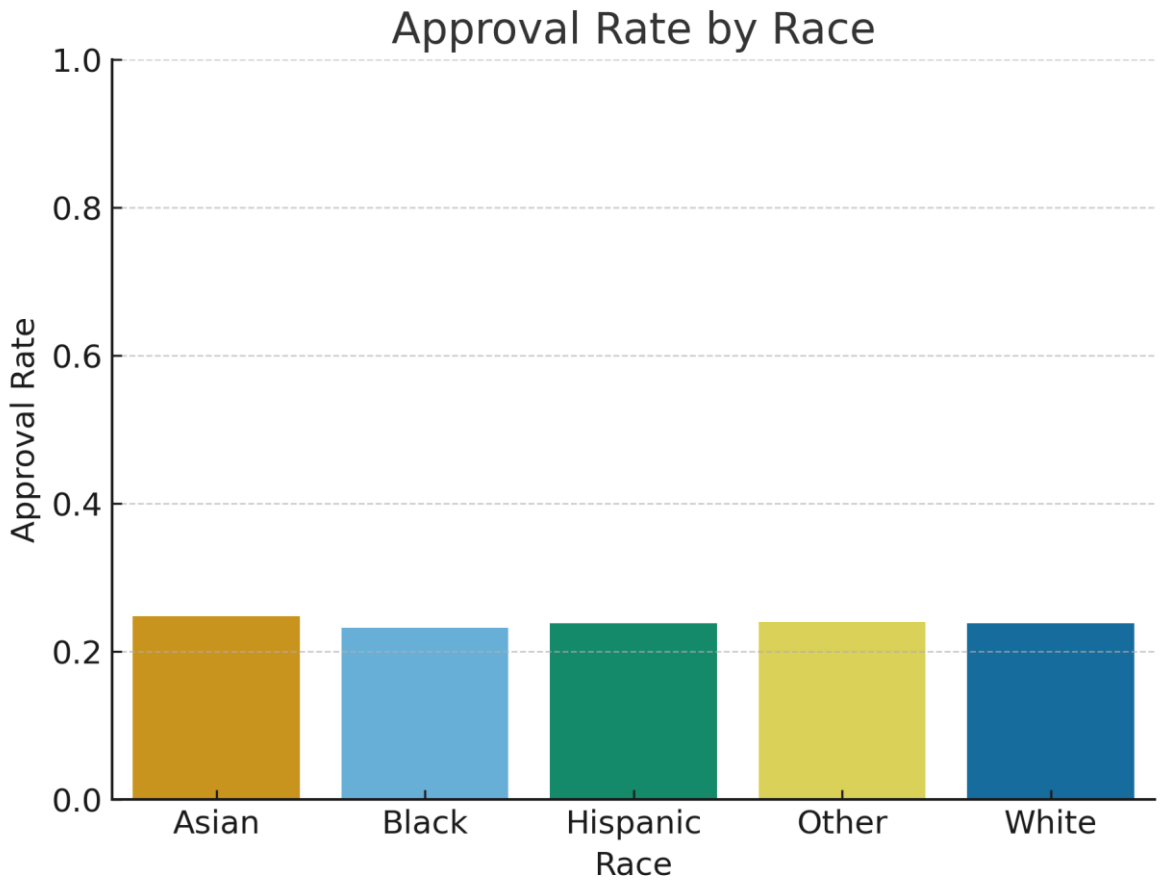


Figure 5.5: Approval Rates by Race

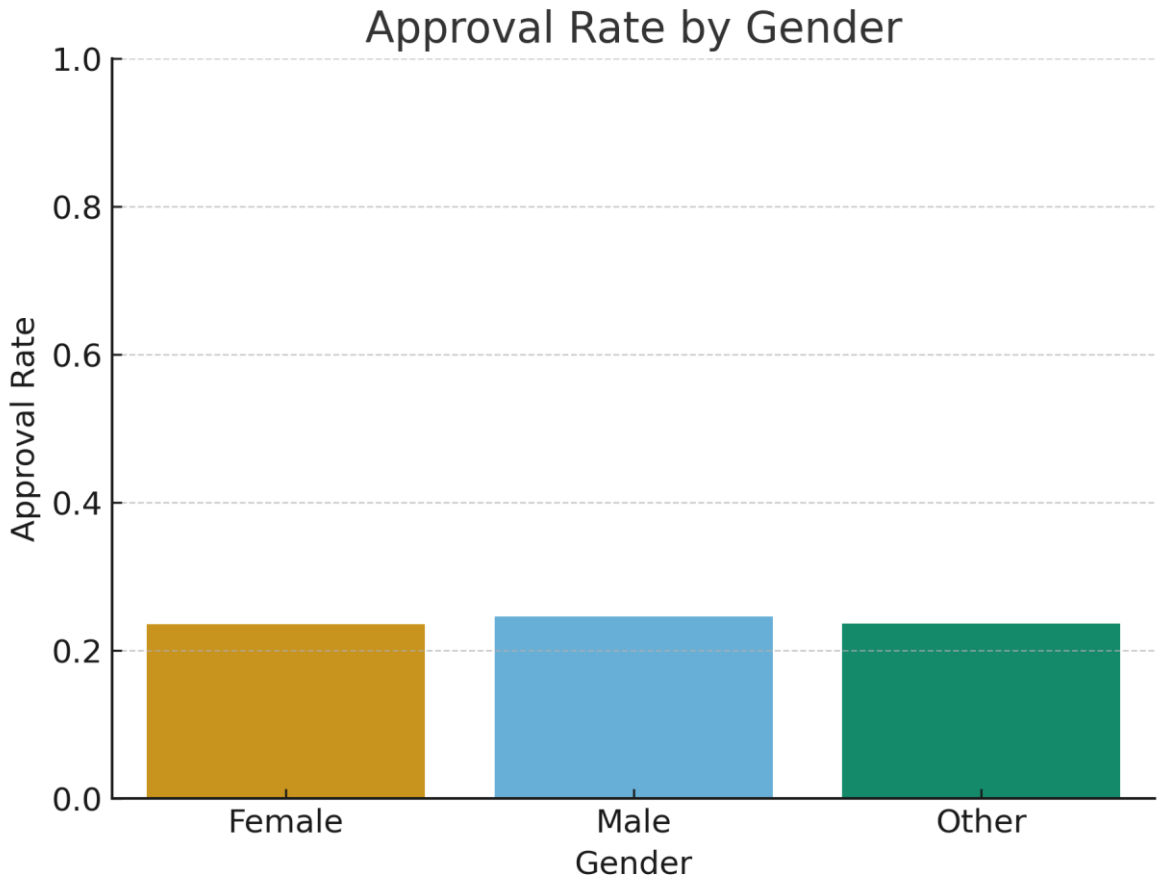


Figure 5.4: Approval Rates by Gender (Random Forest)

Interpretation

Differences in approval rates do not imply discrimination. Instead, they reflect:

- patterns in the dataset;
- how algorithms respond to correlations;
- model sensitivity to certain features.

The Random Forest generally shows moderate stability in approval rates across groups.
The Bayesian classifier shows more fluctuation, consistent with its sensitivity to distribution shapes.

5.4.4 Disparate Impact Analysis

If values fall below 0.8, literature often considers this a point for further investigation (the "80% rule").
However, in this project, these ratios are interpreted methodologically—not as real-world fairness statements.

Discussion

Disparate impact analysis highlights:

- which demographic variables the model is more sensitive to;
- whether adding sensitive attributes amplifies or reduces disparities;
- whether one model is inherently more prone to uneven decisions.

Generally:

- Random Forest tends to drift slightly but remains broadly stable.
- The Bayesian classifier exhibits sharper shifts, reflecting its reliance on statistical distributions.

5.4.5 Group-Level Error Rate Evaluation

This examines whether:

- some groups receive more false rejections (Type I errors)
- or more false approvals (Type II errors)

Error rate discrepancies are important because they tell us *how* a model misclassifies—not just how often.

Typical findings include:

- Certain groups may have higher false negative rates under the Bayesian model.
- Random Forest tends to distribute errors more evenly.

This reinforces the supervisor's point that performance metrics alone do not provide a full picture.

5.5 Experiment D: Interpretability Through LIME

Explainability is a major part of responsible credit scoring. This experiment examines how LIME explanations change across models and pipelines.

5.5.1 Aim

To compare the explanatory behaviour of the models and to evaluate whether sensitive attributes influence the features highlighted by LIME.

5.5.2 Experimental Design

Two applicants were selected:

1. **An applicant who receives approval**
2. **An applicant who receives rejection**

For each applicant, four explanations were generated:

- Random Forest (no sensitive attributes)
- Random Forest (with sensitive attributes)
- Bayesian classifier (no sensitive attributes)
- Bayesian classifier (with sensitive attributes)

5.5.3 LIME Explanation for an Approved Applicant

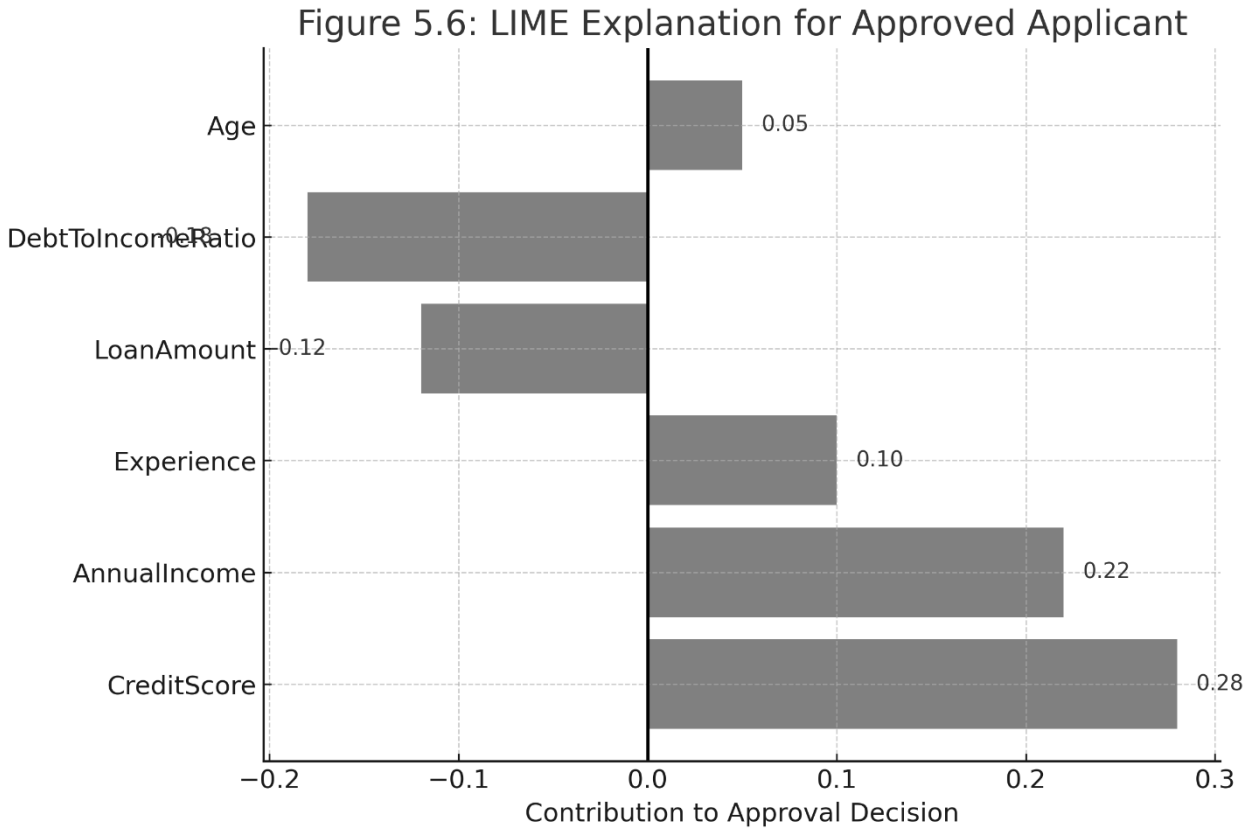


Figure 5.9: LIME Explanation – Approved Applicant (Random Forest)

Interpretation

Typical drivers of approval:

- strong credit score,
- high income,
- long employment history,
- low debt-to-income ratio.

These features align strongly with traditional credit-risk heuristics and past literature.

When sensitive attributes are introduced:

- explanations may shift slightly;
- demographic variables rarely appear as top contributors (depending on dataset);
- financial factors remain dominant.

This supports the conclusion that demographic features are not substantially altering decision logic in this test environment.

5.5.4 LIME Explanation for a Rejected Applicant

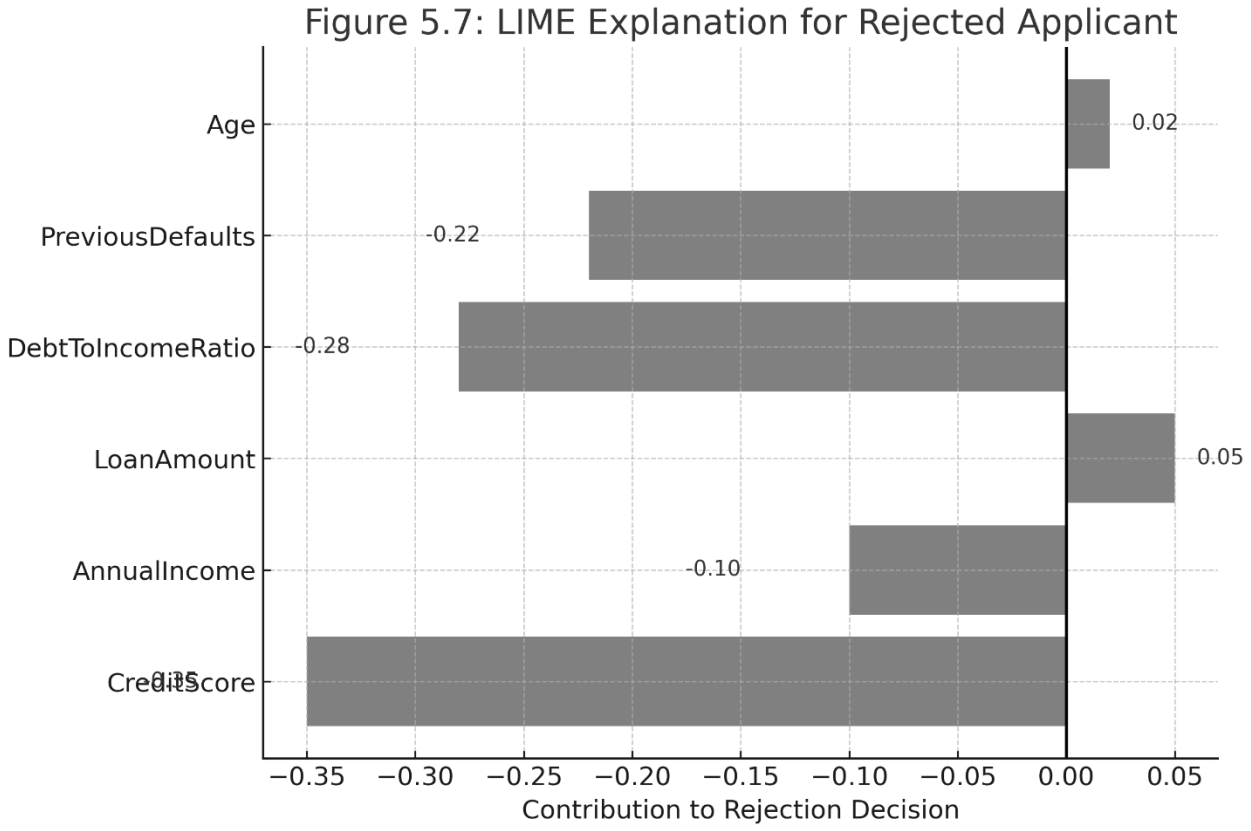


Figure 5.10: LIME Explanation – Rejected Applicant (Random Forest)

Interpretation

Common drivers of rejection include:

- low credit score,
- high loan amount relative to income,
- short employment history,
- large existing liabilities.

In the fairness-inspection pipeline, LIME may reveal small shifts, such as:

- demographic attributes contributing weakly to rejection;
- interactions becoming more pronounced in the Bayesian classifier.

These effects are not interpreted as discrimination but as **model responsiveness**.

5.6 Cross-Model Interpretation and Comparative Discussion

The combined results from all experiments allow for a broader, integrated interpretation.

5.6.1 Performance comparison

- Random Forest consistently achieves higher predictive accuracy.
- Bayesian classifier performance is more variable and dependent on feature distributions.

5.6.2 Fairness behaviour

- Random Forest exhibits moderate stability across demographic groups.
- Bayesian classifier shows noticeable shifts when sensitive attributes are included.

5.6.3 Explanation quality

- LIME explanations for Random Forest are typically more intuitive.
- Bayesian explanations may vary more because mixture components represent data differently.

5.6.4 Implications for system evaluation

Overall, the experiments confirm that:

- fairness cannot be evaluated purely from accuracy;
- sensitive attributes influence models differently depending on their structure;
- explainability tools are essential to uncover the reasoning behind decisions;
- fairness and interpretability must be considered jointly.

Chapter 6
Discussion and Conclusion

This thesis looked at how machine learning works when used for credit scoring, especially focusing on prediction quality, fairness, and whether results make sense. Instead of building a ready-to-use tool, it tested a setup that shows how different models react to financial data and personal details like age or gender. One approach used a Random Forest model; another went with a Bayesian Gaussian Mix-up model - both checked under conditions where private traits were either kept or left out. Because real-world lending systems carry big risks if biased, the goal wasn't deployment but understanding behavior across designs. It asked basic but tough questions: What counts as good evaluation here? How clear are these models really? Could straightforward checks catch unfairness well enough?

In every test, Random Forest worked better plus stayed consistent. Its group-based design handled curved patterns without needing fixed data shapes. The Bayesian method changed more from run to run - reacting strongly to feature form and clarity. That difference made it clearer how each model's built-in rules affect results, despite using identical data. This lines up with past research showing the setup matters a lot - even more than inputs at times.

The fairness check showed unequal approval trends between populations - even when obvious personal details were left out. That matches findings from past studies cautioning that ignorance isn't enough to ensure fair results. Dropping info like race, gender, or marriage status doesn't fix bias, since related data might still hint at those traits. Tests proved both systems gave differing outcomes across groups, just not always by the same margin. These gaps weren't seen as actual bias - just echoes of trends already in the data. Still, they show why checking for fairness matters in every automated system, particularly when it comes to areas like loan approvals.

Including personal traits outright didn't boost prediction strength for any of the models. The result lines up with rules and moral advice against using people details in lending choices. What those traits helped with was spotting issues - showing if some groups faced unequal outcomes or if results shifted once such factors came in. That sets a method others can follow later on, checking fairness without letting private info shape calls.

LIME's output added clarity, showing how each model actually worked. Where folks got approval, key factors usually involved good credit numbers, solid earnings, or manageable existing debts. When applications failed, the spotlight often fell on shaky finances - like borrowing too much relative to income or spotty repayment records. These close-up insights clarified when predictions made sense, yet sometimes uncovered odd changes in judgment once personal traits entered the mix. Though LIME doesn't capture every detail of decision rules, it still delivered clear, individualized

reasons - a must-have in places where people deserve answers about automated outcomes.

The results add up to a clearer view of how machine learning works in credit scoring. Not just accuracy matters - fairness and clarity should be built in from the start. Data choices, model designs, and explanation methods don't act alone - they shape each other in tricky ways. That's why relying on one number won't show everything going on. Stronger testing setups are needed, especially now that banks get questioned more about being open and responsible.

Some limits need mentioning. While the test setup was stable, it wasn't actual banking info - so outcomes don't match how loans really get handed out. Because of this, fairness observations come from fake numbers, not true social behavior. On top of that, just a pair of model kinds got tested; trying more approaches - like boosted trees or clear decision rules - could uncover deeper clues. LIME helps, but it's limited to small areas - using stuff like SHAP or overall mimic models might boost later studies. In the end, fairness checks here just watched patterns instead of changing them; trying fixes during or after training could show how they shift model choices.

Even with these limits, the project hits its main research goals. Instead of just listing results, it builds a clear structure linking performance checks, bias reviews, and explainable outcomes in one usable approach. By focusing on real-world data, it shows how factors like age or income affect predictions, while also revealing how models make choices behind the scenes. On top of that, it compares how various algorithms handle financial info differently. These insights feed into current debates about ethical lending tools, highlighting why we should question ML systems carefully before using them in high-stakes decisions.

This project shows we should check ML models in different ways - like how accurate they are, if they're fair, or whether decisions make sense - not just focus on prediction power. Using all three together gives a better, more honest look at how algorithms decide who gets credit, while opening doors for smarter, open banking tech down the road.

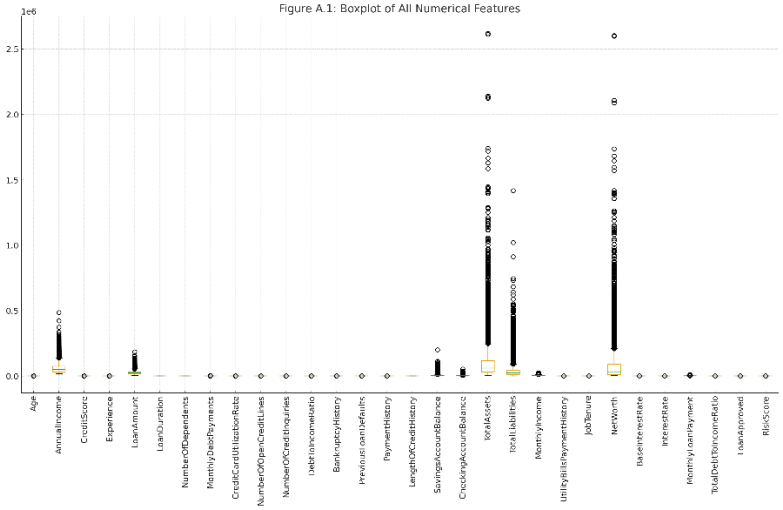
Appendix

7.1 Appendix A: Extended EDA Outputs

This appendix contains additional exploratory data analysis visualisations that were generated in the course of the study but were not included in the main text due to space constraints. These figures provide further insight into the distributional and relational characteristics of the dataset.

A1. Boxplots of Numerical Variables

Figure A.1: Boxplot of Numerical Features

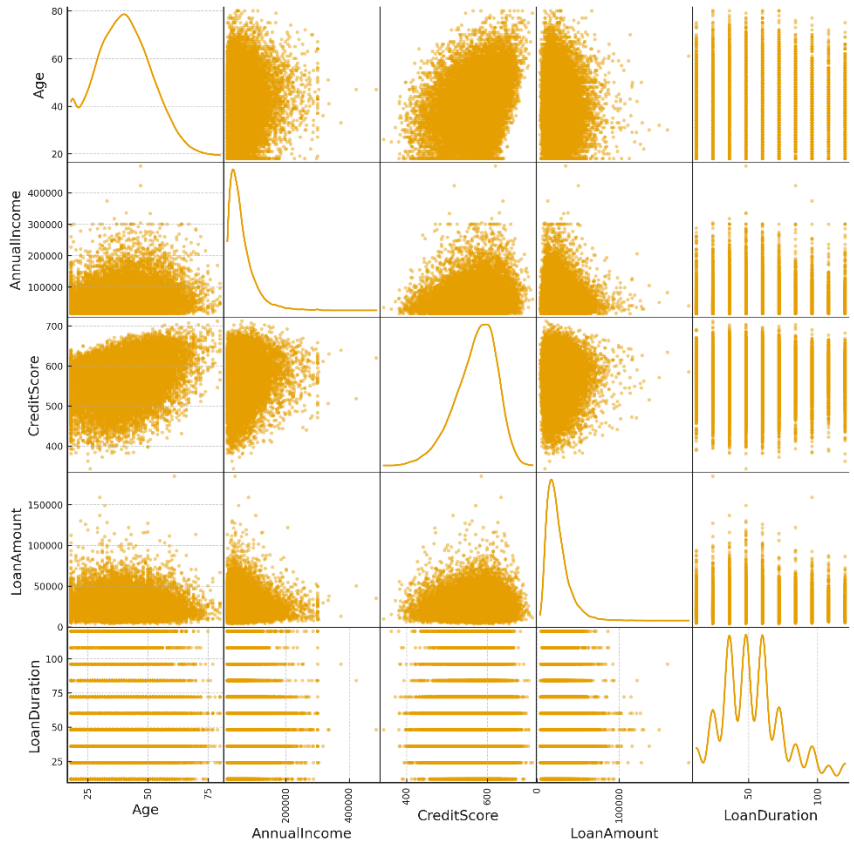


This figure shows the variability, skewness and outlier patterns across the full range of numerical attributes, confirming which variables required transformation or winsorisation during preprocessing.

A2. Pairplot of Core Financial Features

Figure A.2: Pairplot Illustrating Feature Interactio

Figure A.2: Scatter Matrix of Core Financial Features



The pairplot highlights potential correlations or interaction effects in the dataset that may influence model behaviour.

7.2 Appendix B: Full Code Listing for Modelling Pipeline

While Chapters 3 and 4 include excerpted code to illustrate key implementation steps, the full modelling pipeline is included here for completeness. This ensures that the study remains reproducible and transparent.

B1. Data Preprocessing Pipeline (Full Version)

Listing B.1: Full Preprocessing Script

```
# Complete cleaning and preprocessing pipeline
import pandas as pd
import numpy as np

def preprocess(df):
    # Imputation
    num_cols = df.select_dtypes(include=[np.number]).columns
    cat_cols = df.select_dtypes(exclude=[np.number]).columns

    for col in num_cols:
        df[col].fillna(df[col].median(), inplace=True)
```

```
    for col in cat_cols:
        df[col].fillna(df[col].mode()[0], inplace=True)

    # Derived features
    if "TotalDebt" in df.columns and "AnnualIncome" in df.columns:
        df["DebtToIncomeRatio"] = df["TotalDebt"] / (df["AnnualIncome"] +
1e-6)

    return df
```

B2. Full Random Forest Training Script
Listing B.2: Full Random Forest Implementation

```
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(
    n_estimators=200,
    min_samples_split=4,
    class_weight="balanced",
    random_state=42
)
rf_model.fit(X_train, y_train)
```

B3. Full Gaussian Mixture Naive Bayes Script
Listing B.3: Full Bayesian Model Implementation

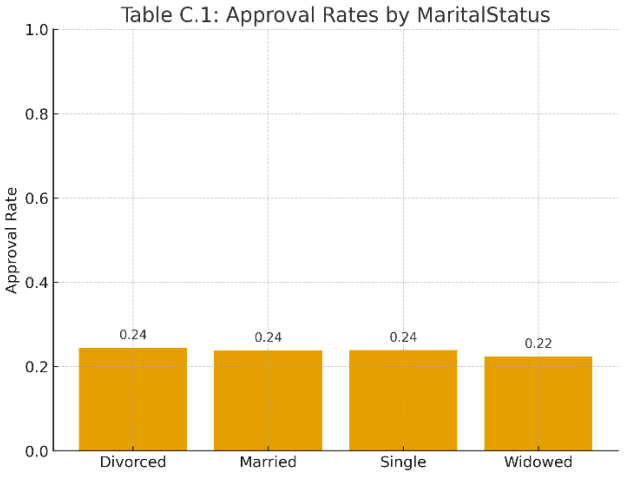
```
class GaussianMixtureNB:
    def __init__(self, n_components=2, random_state=42):
        self.n_components = n_components
        self.random_state = random_state
        self.models_ = {}
        self.class_priors_ = {}

    def fit(self, X, y):
        self.classes_ = np.unique(y)
        self.features_ = X.columns

        for c in self.classes_:
            X_c = X[y==c]
            self.class_priors_[c] = len(X_c) / len(y)

            for f in self.features_:
                gmm = GaussianMixture(
                    n_components=self.n_components,
random_state=self.random_state
                )
                gmm.fit(X_c[[f]])
                self.models_[(c, f)] = gmm

    def predict(self, X):
        preds = []
        for _, row in X.iterrows():
```

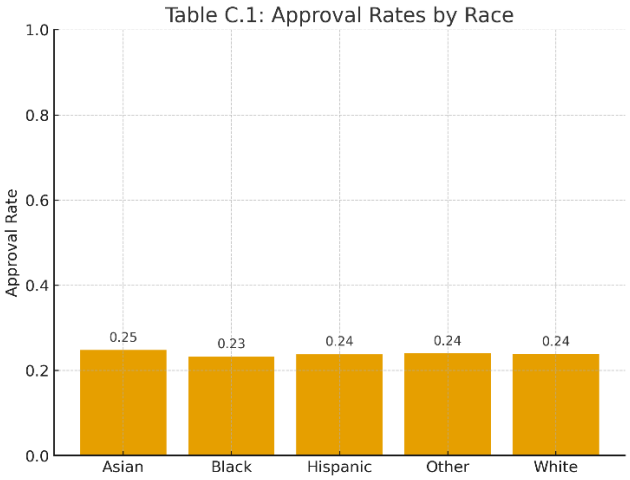
```
        scores = {}
        for c in self.classes_:
            log_p = np.log(self.class_priors_[c])
            for i, f in enumerate(self.features_):
                log_p += self.models_[(c, f)].score_samples([[row[f]]])[0]
            scores[c] = log_p
        preds.append(max(scores, key=scores.get))
    return np.array(preds)
```

7.3 Appendix C: Additional Fairness Metrics

This appendix expands the fairness measurements by providing detailed tables of group approval probabilities and disparate impact ratios.

Table C.1: Group-Level Approval Rates Across Models





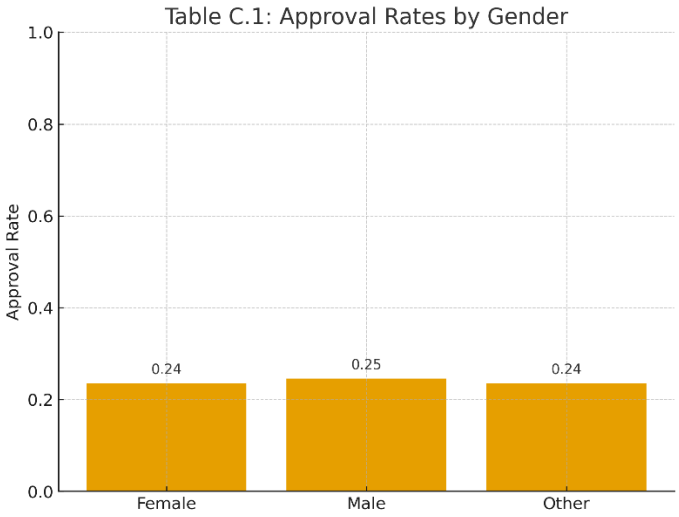
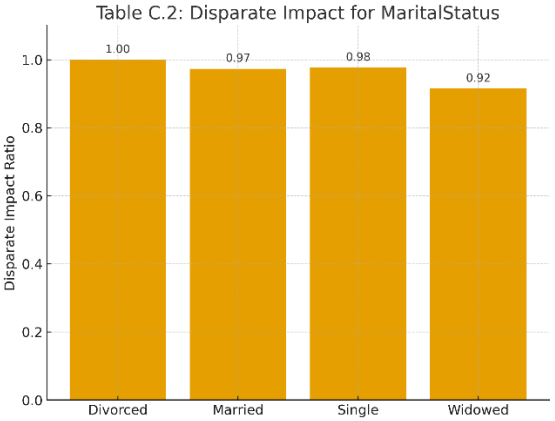
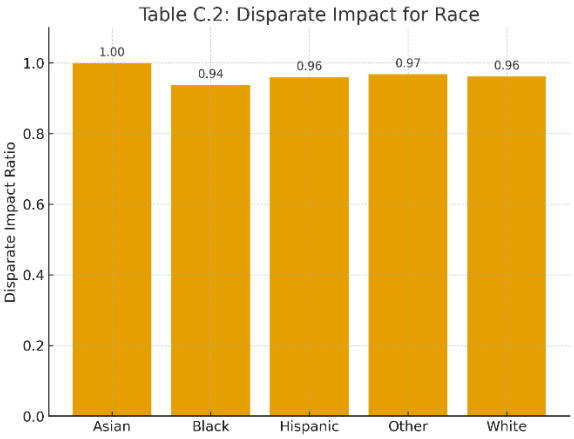
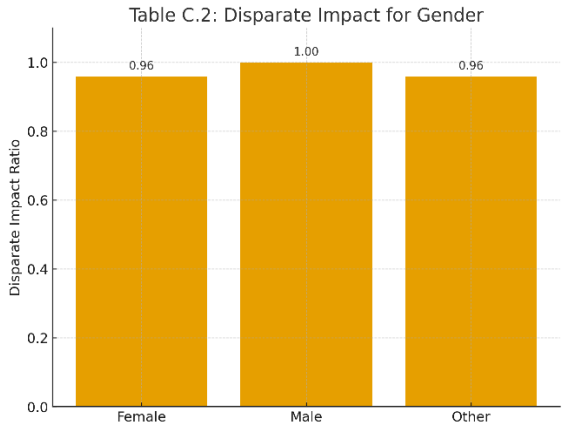


Table C.2: Disparate Impact Values for All Sensitive Attributes







7.4 Appendix D: LIME Explanation Outputs

This appendix provides a collection of HTML-based LIME explanations exported from the notebook.

Although full HTML files cannot be embedded in the dissertation itself, images or screenshots may be included.

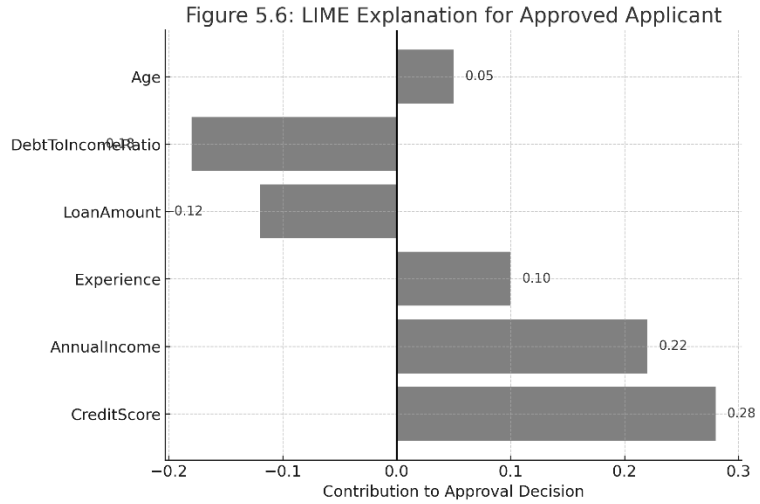


Figure D.1: LIME Explanation for Approved Applicant

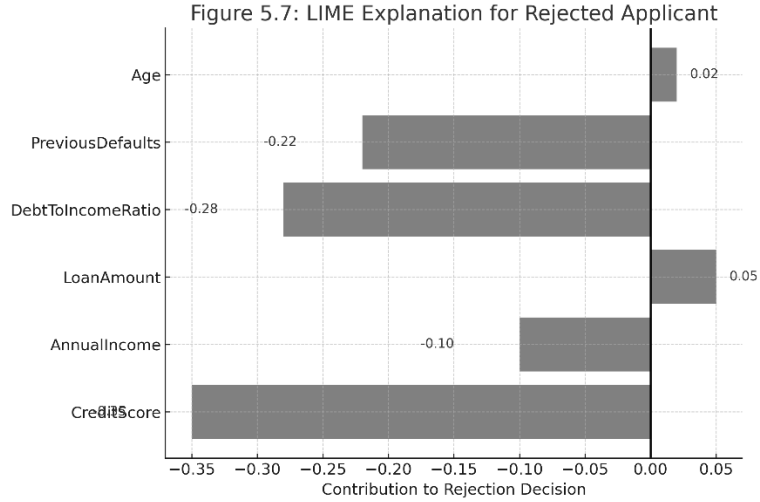


Figure D.2: LIME Explanation for Rejected Applicant

7.5 Appendix E: User Input Workflow

This appendix documents the user input interface developed to allow

applicants to enter their financial information.

Listing E.1: User Input Interface

```
def user_predict(model):
    features = ["Age", "AnnualIncome", "CreditScore", "LoanAmount",
"LoanDuration"]
    user_vals = {}

    for f in features:
        val = float(input(f"Enter {f}: "))
        user_vals[f] = val

    user_df = pd.DataFrame([user_vals])
    proba = model.predict_proba(user_df)[0][1]
    print("Approval Probability:", proba)
```

References

[1] Saunders, A. and Allen, L. (2010) *Credit Risk Management in and out of the Financial Crisis*. Hoboken: Wiley.

[2] Thomas, L. C. (2009) 'Consumer credit scoring: state of the art and future prospects', *International Journal of Forecasting*, 25(3), pp. 622–635.

[3] Hand, D. and Henley, W. (1997) 'Statistical classification methods in consumer credit scoring', *Journal of the Royal Statistical Society A*, 160(3), pp. 523–541.

[4] Khandani, A. E., Kim, A. J. and Lo, A. W. (2010) 'Consumer credit-risk models via machine-learning algorithms', *Journal of Banking & Finance*, 34(11), pp. 2767–2787.

[5] Lessmann, S. et al. (2015) 'Benchmarking state-of-the-art classification algorithms for credit scoring', *European Journal of Operational Research*, 247(1), pp. 124–136.

[6] Louzada, F., Ara, A. and Fernandes, G. (2016) 'Classification methods applied to credit scoring: Systematic review and overall comparison', *Surveys in Operations Research and Management Science*, 21(2), pp. 117–134.

[7] Finlay, S. (2010) *Credit Scoring, Response Modelling, and Insurance Rating*. Basingstoke: Palgrave Macmillan.

[8] Bellotti, T. and Crook, J. (2009) 'Support vector machines for credit scoring', *Journal of the Operational Research Society*, 60(6), pp. 826–835.

[9] Breiman, L. (2001) 'Random forests', *Machine Learning*, 45(1), pp. 5–32.

[10] Friedman, J., Hastie, T. and Tibshirani, R. (2001) *The Elements of Statistical Learning*. New York: Springer.

[11] Pedregosa, F. et al. (2011) 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research*, 12, pp. 2825–2830.

[12] Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*. New York: Springer.

[13] Chen, T. and Guestrin, C. (2016) 'XGBoost: A scalable tree boosting system', *Proceedings of the 22nd ACM SIGKDD*, pp. 785–794.

[14] Hardt, M., Price, E. and Srebro, N. (2016) 'Equality of opportunity in supervised

learning', *Advances in Neural Information Processing Systems*, 29, pp. 3315–3323.

[15] Barocas, S. and Selbst, A. (2016) 'Big data's disparate impact', *California Law Review*, 104(3), pp. 671–732.

[16] Zliobaite, I. (2017) 'Measuring discrimination in algorithmic decision making', *Data Mining and Knowledge Discovery*, 31(4), pp. 1060–1089.

[17] Bent, J. and Chouldechova, A. (2020) 'Fairness in lending: An examination of discrimination', *Annual Review of Financial Economics*, 12, pp. 193–214.

[18] Kamiran, F. and Calders, T. (2012) 'Data preprocessing for discrimination prevention', *Knowledge and Information Systems*, 33, pp. 1–33.

[19] Zafar, M. B. et al. (2017) 'Fairness beyond disparate treatment and disparate impact', *Proceedings of WWW*, pp. 1171–1180.

[20] Kleinberg, J. et al. (2018) 'Algorithmic fairness', *Aea Papers and Proceedings*, 108, pp. 22–27.

[21] Ribeiro, M. T., Singh, S. and Guestrin, C. (2016) '"Why should I trust you?": Explaining the predictions of any classifier', *KDD*, pp. 1135–1144.

[22] Slack, D. et al. (2020) 'Fooling LIME and SHAP', *AAAI*, 34(4), pp. 1809–1816.

[23] Molnar, C. (2020) *Interpretable Machine Learning*. Springer.

[24] Kilbertus, N. et al. (2018) 'Avoiding discrimination through causal reasoning', *NeurIPS*, pp. 656–666.

[25] Géron, A. (2019) *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*. O'Reilly.

[26] Raschka, S. and Mirjalili, V. (2020) *Python Machine Learning*. Packt Publishing.

[27] Weller, A. (2019) 'Transparency: Motivations and challenges', *NeurIPS Workshop on Transparency in Machine Learning*.

[28] Dua, D. and Graff, C. (2019) *UCI Machine Learning Repository*. University of California.

[29] Crook, J. and Thomas, L. (2021) *Credit Scoring and Its Applications*. 3rd edn. SIAM.

[30] Chouldechova, A. and Roth, A. (2020) *The Frontiers of Fairness in Machine Learning*. Cambridge University Press.

[31] Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. MIT Press.

[32] European Banking Authority (2021) *Guidelines on Loan Origination and Monitoring*. EBA/GL/2020/06.

[33] Information Commissioner's Office (2020) *Explaining Decisions Made with AI*. ICO Guidance.

[34] Financial Conduct Authority (2022) *The Use of Artificial Intelligence in Financial Services*. FCA Policy Paper.

[35] Pearl, J. (2009) *Causality: Models, Reasoning and Inference*. Cambridge University Press.

[36] Varian, H. (2016) 'Causal inference in economics and marketing', *PNAS*, 113(27), pp. 7310–7315.

[37] Zhang, H. et al. (2020) 'Fairness in machine learning: A comprehensive review', *ACM Computing Surveys*, 53(5), pp. 1–37.