Credit Score Classification Model

# Introduction

## The Problem

Credit scores play a fundamental role in the global financial system. They are used by banks, credit card companies, and other lenders to assess the creditworthiness of individuals and businesses. This classification helps determine whether to approve loans, the amount to be loaned, and the interest rate. As such, a credit score acts as a gatekeeper to access to capital—a critical factor in driving economic participation, entrepreneurship, and innovation.

## Why It Matters

Credit scores play a foundational role in the economy. They directly influence decisions about who gets access to financing and at what cost. This affects not just individuals, but businesses and entire industries. Capital is what fuels innovation, productivity, and progress — it allows people to scale ideas, build infrastructure, and weather economic downturns. But without access to credit, even the best ideas can’t get off the ground. That’s why accurate, timely credit scoring matters so much.

Finance and credit are basically the lubrication that keeps the engine of capitalism running. Whether it's launching a new manufacturing process or building digital products like data science platforms, these efforts almost always require capital that the individual or business doesn’t initially have. Being able to correctly identify trustworthy borrowers means lenders can confidently distribute capital where it’s most useful. That helps grow the economy and generates returns for both sides — the lender and the borrower.

I also believe there’s a strong connection between someone’s financial behavior and their broader personal traits. For example, if someone is earning more than the median salary in their field, it probably means they’re excelling — and that discipline or ambition might extend into how they manage their money too. That’s why I created a feature that compares a person’s income to the median in their occupation, and that turned out to be one of the most important predictors in the model.

Ultimately, if banks and lenders can improve how they predict credit scores, it leads to a much more efficient system. Money can be sent to where it's most productive, rates can be priced more accurately, and fewer people fall through the cracks simply because of poor classification. This kind of optimization benefits individuals, businesses, and the global economy as a whole.

## Stakeholder Pitch

To gain stakeholder support, the problem can be framed as follows: 'We are creating a machine learning classification model that increases accuracy in predicting creditworthiness. This allows lenders to better assess risk, optimize loan offers, reduce defaults, and extend credit to underserved segments—creating a win-win for financial institutions and borrowers alike.'

# Data Source

The dataset used in this project was sourced from Kaggle, a well-known platform for hosting data science competitions and datasets. It originally contained a wide range of demographic, employment, and financial variables, including annual income, monthly balances, occupation, spending behaviors, and a categorical label representing the individual’s credit score (classified as “Poor,” “Standard,” or “Good”). While the dataset provided a valuable starting point, it required extensive cleaning and transformation. This included correcting data entry errors, handling missing values, converting categorical fields into numerical labels, and engineering new features to enhance model performance.

# Milestones Summary

## Exploratory Data Analysis (EDA)

During the EDA phase, I investigated the structure and distribution of various fields. Histograms and boxplots revealed significant outliers in income and spending-related fields. A strong class imbalance was observed, necessitating balanced sampling or weight adjustments. Many values required normalization and cleaning. For example, income fields contained non-numeric characters, and some categorical fields were inconsistently labeled.

## Data Preparation

Key preprocessing steps included dropping identifiers (like ID and Customer\_ID), cleaning numeric fields by removing non-integer characters, converting strings to numeric types, and using Label Encoding on categorical variables. Additionally, domain-informed feature engineering played a critical role in enhancing model performance. Notably, the following new features were created:

- Savings Rate: monthly balance divided by monthly income

- Industry Income Advantage: income minus the median income for that individual's occupation

- Boolean indicators for high-income occupations (e.g., doctor, accountant)

These features captured nuanced behaviors and traits indicative of creditworthiness, and several were among the top-ranked in feature importance scores during modeling.

# Model Building and Evaluation

For this project, I used a **Random Forest Classifier** to predict credit scores. Random forests are a strong choice for classification tasks because they are robust to overfitting, handle missing values well, and offer built-in feature importance. However, to get the most out of the model, I needed to tune its hyperparameters carefully — and that’s where **Optuna** came in.

Optuna is a modern, efficient hyperparameter optimization framework that uses a technique called **sequential sampling**. Unlike grid search or even randomized search, Optuna is smarter about how it explores the parameter space. It learns from previous trials to decide where to search next, making it far more computationally efficient and effective — especially with models like random forests that take a long time to train.

I chose **7 trials** for optimization because of the long compute time required to train each random forest. Even with a relatively low number of trials, Optuna was able to identify meaningful improvements in model performance thanks to its intelligent search strategy. By focusing on maximizing the **weighted F1-score** using 3-fold cross-validation, I ensured the model was balanced across different classes and robust against overfitting to any particular subset of the data.

The hyperparameters I tuned included:

* n\_estimators: Number of trees in the forest
* max\_depth: Maximum depth of each tree
* min\_samples\_split and min\_samples\_leaf: Control tree complexity and help prevent overfitting
* max\_features: Whether to consider a subset of features at each split (sqrt or log2)
* class\_weight: Whether to balance class weights for imbalanced datasets

The best parameters were chosen by Optuna based on the **average weighted F1-score across 3-fold cross-validation**, rather than a single train-test split. This cross-validation strategy made the evaluation more robust by ensuring that the model performed consistently well across different subsets of the data. It helped avoid the risk of overfitting to one specific fold and added an important layer of **generalizability** to the model.

After 7 optimization trials, Optuna identified the following best hyperparameters:

* n\_estimators: **437** — A relatively high number of trees, which helps reduce variance and improve stability.
* max\_depth: **27** — Deep trees allow the model to capture complex interactions, and this depth suggests the data benefited from such complexity.
* min\_samples\_split: **6** — This prevents the model from splitting nodes too aggressively, which helps prevent overfitting.
* min\_samples\_leaf: **2** — Ensures that no leaf node represents too small a fraction of the data, also aiding generalization.
* max\_features: **'log2'** — This reduces the number of features considered at each split, which encourages diversity among the trees and lowers correlation.
* class\_weight: **'balanced'** — Important for addressing any class imbalance in the dataset, ensuring minority classes weren’t ignored.

By using these parameters, the final Random Forest model achieved a well-balanced combination of depth, regularization, and feature diversity. It was able to capture important relationships in the data without overfitting and handled class imbalance effectively. Overall, the Optuna tuning process enabled the construction of a **high-performing, interpretable, and scalable** classification model tailored specifically to the credit score prediction task.

# Conclusion

## Insights from Analysis

One of the most interesting takeaways from the feature importance ranking was how heavily the model relied on **Outstanding Debt**. It was by far the most important predictor of credit score, with significantly more weight than any other feature. While this initially surprised me, it makes sense upon reflection — high levels of outstanding debt signal potential financial overextension or mismanagement, both of which are critical red flags when assessing someone's creditworthiness.

Other top features included **Interest Rate, Delay from Due Date**, and **Changed Credit Limit**, all of which are tightly tied to real-time credit behavior rather than demographic or identity-based metrics. Features like **Credit Mix, Number of Delayed Payments, and Number of Credit Cards** also made the top 10, reinforcing the idea that a person’s current financial habits and history are more telling than static background traits.

What really stood out, though, was what didn’t show up: **Income** and **Age**. Neither made the top 10 most important features. This was unexpected, since conventional thinking often assumes higher income and older age (implying more financial experience) would lead to better credit scores. Their absence suggests that how someone manages their finances is far more predictive than how much they earn or how old they are. It also emphasizes the value of behavioral data over demographic assumptions — a reminder that creditworthiness is about actions, not just attributes.

## Is the Model Ready for Deployment?

While the model is suitable for pilot deployment, further validation is recommended. Its current performance suggests it could be deployed in a decision support role, augmenting traditional underwriting models. Additional tuning, validation on live or out-of-sample data, and fairness audits are necessary steps before full deployment.

## Recommendations

- Incorporate more domain expertise during feature engineering  
- Use SHAP or other explainability methods to interpret predictions for stakeholders  
- Validate model performance across demographic slices to ensure fairness and equity  
- Explore advanced models like LightGBM or XGBoost for further performance improvements

## Future Challenges and Opportunities

Remaining challenges include data quality issues—particularly those stemming from manual entry—and model explainability. Lenders often require transparent reasoning for decisions, especially under regulatory scrutiny. Bias mitigation must be carefully handled to prevent disparate impact on protected groups. Future iterations could include ensemble models, synthetic data augmentation, or integration with real-time financial behavior data to improve responsiveness and accuracy.