# Credit Risk Modelling and FICO Score Bucketing

This report explains the credit‑risk notebooks. The objective is to predict loan defaults, estimate expected losses and assign borrowers into risk buckets using FICO scores. The explanations below translate the Python code into plain language for non‑programmers.

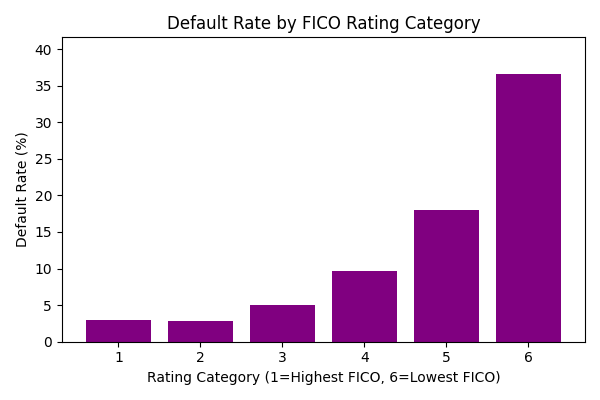
## Loan portfolio overview

The data set contains 10,000 loans with fields such as FICO score, income, loan amount outstanding, total debt outstanding, years employed and number of credit lines. Each record has a binary label indicating whether the loan defaulted (1) or not (0). Basic commands like .head(), .info() and .describe() show sample rows, data types and summary statistics. Overall, the average FICO score is 637.6 and about 18.51% of loans have defaulted.

Borrowers are grouped into six rating categories based on their FICO score. Table 2 summarises the mapping and observed default rates in the data set.

|  |  |  |
| --- | --- | --- |
| Rating | FICO range | Default rate (%) |
| 6 | 300–599 | 36.6 |
| 5 | 600–649 | 17.94 |
| 4 | 650–699 | 9.61 |
| 3 | 700–749 | 4.98 |
| 2 | 750–799 | 2.8 |
| 1 | 800–850 | 2.94 |

Figure 3 plots the empirical default rate by rating; higher‑risk borrowers (rating 6) default much more often than top‑rated borrowers (rating 1).



## Feature engineering and exploratory analysis

New variables are created to capture a borrower’s leverage: ratio\_income\_totdebt divides annual income by total debt outstanding and ratio\_income\_loan\_amt divides income by the loan balance. Histograms and scatter plots (created with matplotlib and seaborn) show how defaulting borrowers tend to have lower income relative to debt and lower FICO scores. A correlation heatmap highlights that FICO score is negatively correlated with default status, whereas higher outstanding debt is positively correlated.

## Logistic regression model

To predict the probability of default, the notebook defines features X by dropping the default column and uses Y = Loans\_df["default"] as the target. The data is split into training and test sets using train\_test\_split(). A logistic regression model is fitted via LogisticRegression().fit(X\_train, Y\_train). This algorithm estimates coefficients that map features to a probability between 0 and 1 using the sigmoid function.

* Key steps include:
* predict\_proba(X\_test) returns a two‑column array containing the probability of non‑default and default for each observation.
* A threshold of 60 % is applied so that probabilities above 0.60 are classified as defaults.
* Confusion matrices and classification reports from sklearn.metrics reveal model performance. With all features, accuracy is approximately 98.22999999999999% and the AUC (area under the ROC curve) is 0.998. The confusion matrix [[4857, 33], [73, 1037]] shows relatively few false positives and false negatives.

To compute financial impact, the code multiplies the predicted probability of default by the loss given default (LGD), defined as 90 % of the outstanding loan balance when the recovery rate is 10 %. Summing the expected losses over all loans yields about $7.42 million.

## XGBoost model

The notebook also trains a gradient‑boosted decision‑tree model using XGBoost. After specifying an XGBClassifier, the model fits the data and predicts default probabilities. The same evaluation steps as above are followed: probabilities are thresholded at 60 %, the confusion matrix is computed and the ROC curve is plotted. The XGBoost model achieves higher accuracy of about 99.63% and an AUC close to 1.0. Its confusion matrix [[2423, 2], [9, 566]] indicates very few misclassifications. Expected loss is similar to the logistic model (≈$7.42 million) because PD estimates are close, but XGBoost provides greater confidence in high‑risk cases.

## Rating‑only model

A simpler model uses only the FICO‑derived rating as a predictor. The code reshapes the Rating column into a 2‑D array, splits the data and trains another logistic regression. Since the rating carries less information than the full feature set, the model’s accuracy drops to about 81.39999999999999% and its AUC is 0.704. At a 60 % threshold it predicts all loans as non‑defaults, producing the confusion matrix [[5698, 0], [1302, 0]]. Nevertheless, the rating remains a useful high‑level indicator of risk: Figure 3 shows default rates climbing steeply from rating 1 to rating 6.

## Key insights

The FICO score is the dominant predictor of default, while income‑to‑debt ratios and outstanding balances provide additional nuance. Logistic regression is simple and interpretable but can under‑perform on complex relationships. XGBoost captures non‑linear interactions and gives slightly better classification accuracy. Calculating expected loss combines PD, LGD and exposure to quantify how much capital the lender should set aside. Finally, rating buckets translate continuous FICO scores into intuitive risk categories for portfolio monitoring and communication with non‑technical stakeholders.