**Loan Policy Exercise: Choosing a Probability of Default (PD) Cutoff**

**Exercise:**

This exercise is a way of assessing how accurate and precise a model predicts. This is important for knowing when banks should lend or not.

Below is a confusion-matrix summary table at three policy cutoffs (5%, 8%, 12%). A lower cutoff is stricter (more rejections for loans). Use these to evaluate trade-offs using the Excel worksheet.

Run the Loan\_ML code provided, changing the pd\_cutoff to 0.08 and 0.12 (cut off is currently set to 0.05).

Fill in the empty cells with the output Confusion Matrix for each code run and assess the precision and accuracy of each model. **5% cut off is prefilled for you.**

**Confusion Matrix Summary**

| **PD Cutoff** | **TP** | **FP** | **TN** | **FN** | **Precision (rejects)** | **Recall (defaults caught)** | **Approval Rate** | **Rejection Rate** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5.00% | 483 | 2016.0 | 1.0 | 0.0 | 0.193 | 1.0 | 0.000 | 1.000 |
| 8.00% |  |  |  |  |  |  |  |  |
| 12.00% |  |  |  |  |  |  |  |  |

**Review Appendix 1 below for guidance to populate this table and help interpret the code outputs.**

**Task**

1. Download the files and run the code
2. Inspect the code outputs for each **Cutoff** to populate table above
3. Compare the three policies using Appendix 1 for guidance and recommend a cutoff. Explain choice.

Tip: You can also explore different PD cutoffs at e.g. 7% or 10%, to find best solution by re-running the notebook further.

**Appendix 1: Breakdown of each output**

Here’s a full breakdown of your **cutoff = 0.05 results**, explained step by step for the following output:

A screenshot of a computer code

AI-generated content may be incorrect.

**1. ROC-AUC: 0.653**

* **What it measures:** Ability of the model to distinguish between good borrowers and defaulters across *all thresholds*.
* **Interpretation:** 0.653 > 0.5, so it’s better than random, but still modest discriminatory power.

**2. PR-AUC: 0.309**

* **What it measures:** Trade-off between precision and recall, useful in imbalanced data.
* **Interpretation:** 0.309 shows the model can’t maintain high precision and recall together — it tends to sacrifice one for the other.

**3. Confusion Matrix (cutoff = 0.05) example**

[[ 1 2016]

[ 0 483]]

* Format:
* [[TN FP]
* [FN TP]]
  + **TN = 1** → 1 good borrower correctly approved.
  + **FP = 2016** → 2016 good borrowers wrongly rejected.
  + **FN = 0** → No defaulters wrongly approved.
  + **TP = 483** → All 483 defaulters correctly rejected.

**Interpretation:** The model rejects nearly everyone. It catches *all defaults*, but also wrongly rejects nearly every good customer.

**4. Classification Report**

**Class 0 = Good borrowers**

* **Precision = 1.000** → Of those *predicted good*, 100% really were good (but there was only 1 case!).
* **Recall = 0.000** → It almost never approves good customers (caught 1 out of 2017).
* **F1 = 0.001** → Essentially useless for good borrowers.
* **Support = 2017** → Number of good borrowers.

**Class 1 = Defaulters**

* **Precision = 0.193** → Only 19.3% of rejected applicants were truly defaulters; the rest were good borrowers wrongly rejected.
* **Recall = 1.000** → Every default was caught.
* **F1 = 0.324** → Better than class 0, but still low.
* **Support = 483** → Number of default borrowers.

**5. Accuracy = 0.194**

* Only ~19% of predictions are correct overall.
* This happens because the model *rejects almost everyone*, so it only gets defaults right, not the majority good customers.

**6. Macro Average**

* **Precision = 0.597, Recall = 0.500, F1 = 0.162**
* Average across both classes (treats them equally).
* Shows imbalance: defaults are well-detected, good borrowers are not.

**7. Weighted Average**

* **Precision = 0.844, Recall = 0.194, F1 = 0.063**
* Weighted by class size (good borrowers dominate).
* Looks high for precision because almost all rejections are predicted as default, but recall is terrible (most good customers lost).
* F1 is very low — confirming the model is not useful in practice at this threshold.

✅ **Overall takeaway at cutoff 0.05:**

* The model is **overly strict** → rejects almost everyone.
* It **catches 100% of defaulters** but at the cost of **rejecting 99.95% of good customers**.
* Business-wise: disastrous — customers leave, revenue collapses, even though defaults are avoided.