Static ML Project Audit Report

Adversarial examiner for machine learning notebooks

File analysed: credit_card_fraud_detection.ipynb
Audit type: Static notebook analysis with optional AI examiner notes
Generated by: ModelHawk

Executive Summary

The notebook shows strong static audit signals. The detected project profile is Classification with Tabular data. ModelHawk detected positive evidence for evaluation quality, reproducibility and portfolio readiness. No major static red flags were detected, although the workflow should still be reviewed because static analysis cannot prove methodological correctness.

Important limitation. This report is based on static notebook analysis. It does not execute the notebook and does not prove the absence of data leakage, methodological issues or implementation errors. Instead, it highlights technical signals, likely risks and improvement opportunities detected from the notebook structure, code and markdown.

Report Scope and Methodology

Component	Description
Notebook parser	Reads code cells, markdown cells and notebook-level structure from the uploaded .ipynb file.
Static audit rules	Detects ML workflow signals such as splitting, metrics, reproducibility, resampling, pipelines, limitations and business context.
Heuristic scores	Converts detected signals into review dimensions for leakage risk, metric quality, reproducibility and portfolio readiness.
AI examiner	Optionally converts the deterministic audit result into clearer technical review notes using a configured AI provider.
PDF report	Compiles the structured audit into a professional LaTeX report suitable for portfolio review and technical discussion.

Table 1: ModelHawk report generation pipeline.

Detected Project Profile

Field	Detected value
Primary task	Classification
Secondary tasks	None detected
Data modality	Tabular
Problem traits	Imbalanced Data, Fraud Detection, Threshold Or Cost Sensitive, Deep Learning, Anomaly Or Rare Event Context
Applied audit packs	Universal, Classification, Tabular, Imbalanced Data, Deep Learning, Portfolio And Reporting
Confidence	High - 84/100

Table 2: Detected project profile and applied audit packs.

Notebook Structure

Metric	Value
Total cells	196
Code cells	80
Markdown cells	116

Table 3: Notebook structure summary.

Audit Scores

Dimension	Score	Interpretation
Leakage risk	8/100	Lower is better. This reflects static warning signals.
Metric quality	92/100	Higher suggests stronger metric coverage.
Reproducibility	90/100	Higher suggests better reproducibility signals.
Portfolio readiness	92/100	Higher suggests stronger presentation and documentation.

Table 4: ModelHawk static audit scores.

Score interpretation. Scores are heuristic static-analysis scores based on detected notebook signals. They are not statistical probabilities and do not guarantee correctness.

Audit Findings

No major static red flags detected

Severity: Low

The notebook passed the first static checks. ModelHawk detected several positive signals, although a deeper audit would require executing the workflow and testing robustness.

Good Practices Detected

The following positive signals were detected in the notebook. These do not prove correctness, but they indicate stronger ML project structure and documentation.

- Clear train/test split signal detected.
- Stratified split detected, useful for imbalanced classification.
- Pipeline or ColumnTransformer detected, which can reduce preprocessing leakage risk.

- Cross-validation or advanced validation signal detected.
- Random seed or random_state detected.
- PR-AUC or precision-recall evaluation detected.
- ROC-AUC evaluation detected.
- Precision, recall or F1-style evaluation detected.
- Threshold or probability-based decision logic detected.
- Hyperparameter search detected.
- Baseline comparison signal detected.
- Accuracy limitation appears to be acknowledged and supported by stronger metrics.
- Resampling/leakage awareness detected in the notebook explanation.
- Deep learning diagnostics such as early stopping or learning curves detected.
- Limitations or future work section detected.
- Business/cost context detected.
- Substantial markdown explanation detected.

Risks to Verify

These are not necessarily errors. They are review points that should be checked carefully before submission, publication or technical discussion.

- Verify that SMOTE/resampling is applied only to the training data in every experiment and cross-validation workflow.
- Accuracy is present, but stronger metrics and explanation were detected. Verify that the final conclusion prioritises minority-class performance.
- Metric coverage looks strong statically, but verify that the metrics are calculated on held-out data only.

Priority Fixes

- No urgent priority fixes were returned by the current audit.

Recommendations

1. State the project type, target variable and evaluation objective clearly.
2. Make the final metric choice explicit and explain which metric should drive model selection.
3. Strengthen the final discussion by clearly comparing the selected model against the baseline.
4. Keep the resampling/leakage explanation close to the validation methodology.
5. Make dependency versions and rerun instructions easy to find.

6. Connect the limitations section more directly to deployment risk and future validation.
7. Strengthen the business/cost interpretation of false positives and false negatives.

Suggested Notebook Improvements

The following markdown suggestions are designed to improve the explanation and defensibility of the notebook.

Final metric justification

Where: Before the final model selection or conclusion

Although accuracy is reported, the final model should be selected using metrics that better reflect minority-class performance. In this project, precision, recall, F1-score and PR-AUC provide a more informative view of fraud detection performance than accuracy alone.

AI Examiner Notes

AI technical review. These notes were generated by a configured AI provider using the deterministic ModelHawk audit as evidence. They should be treated as explanatory support, not as proof that the notebook is correct.

Executive summary

ModelHawk identified this notebook as a Classification project using Tabular data. The detected traits include Imbalanced Data, Fraud Detection, Threshold Or Cost Sensitive, Deep Learning and Anomaly Or Rare Event Context. The static audit produced a leakage risk score of 8/100, metric quality of 92/100, reproducibility of 90/100 and portfolio readiness of 92/100. No major static red flags were detected, but this remains a static review rather than proof of methodological correctness.

Technical assessment

The notebook appears to be structured as a Classification workflow over Tabular data. ModelHawk detected 196 total cells, including 80 code cells and 116 markdown cells, suggesting a mix of implementation and explanation. Positive static signals include Clear train/test split signal detected, Stratified split detected, useful for imbalanced classification, Pipeline or ColumnTransformer detected, which can reduce preprocessing leakage risk, Cross-validation or advanced validation signal detected, Random seed or random_state detected, PR-AUC or precision-recall evaluation detected, ROC-AUC evaluation detected and Precision, recall or F1-style evaluation detected. Because the detected profile involves fraud detection or class imbalance, the evaluation strategy should focus on minority-class behaviour rather than headline accuracy alone. Threshold or probability-based decision logic was detected, so the final threshold choice should be explicitly justified. SMOTE or resampling signals were detected, which makes it important to verify that resampling is applied only inside the training workflow. Deep learning signals were detected, so validation curves, early stopping and overfitting checks should be clearly explained if neural models are part of the comparison.

Risk interpretation

The risks identified by ModelHawk are review points, not confirmed errors. The leakage-risk score is low (8/100). The most important checks are: Verify that SMOTE/resampling is applied only to the training data in every experiment and cross-validation workflow, Accuracy is present, but stronger metrics and explanation were detected. Verify that the final conclusion prioritises minority-class performance and Metric coverage looks strong statically, but verify that the metrics are calculated on held-out data only. These points matter because static analysis can detect likely workflow signals but cannot confirm exactly how the notebook behaves when executed.

Portfolio verdict

As a portfolio project, this notebook looks promising. It presents a clear Classification use case with Tabular data and strong static signals around evaluation, reproducibility and explanation. The leakage-risk score is low (8/100), but the remaining review points should still be verified manually. The main improvement is to make the final model-selection logic, threshold choice, resampling safety and deployment limitations even easier to defend.

AI-Improved Recommendations

1. Make it explicit that final model selection should be driven by minority-class performance rather than accuracy alone.
2. Explain the practical trade-off between catching more fraudulent transactions and reducing false alerts for legitimate users.
3. Justify the selected decision threshold and explain how it changes the balance between precision and recall.
4. Keep the SMOTE/resampling explanation close to the validation methodology so the reader can verify that resampling is applied only inside the training workflow.
5. Strengthen the final discussion by comparing the selected model directly against the baseline.
6. Clarify whether neural networks are part of the final solution or used mainly as a comparison against classical machine learning models.
7. State the project type, target variable and evaluation objective clearly.
8. Make dependency versions and rerun instructions easy to find.
9. Connect the limitations section more directly to deployment risk, data drift and future validation.

AI-Suggested Markdown Cells

Final metric justification

Where: Before the final model selection or conclusion

Although accuracy is reported, the final model should be selected using metrics that better reflect minority-class performance. In this project, precision, recall, F1-score and PR-AUC provide a more informative view of fraud detection performance than accuracy alone.

Fraud detection cost trade-off

Where: Before the final model selection or conclusion

In fraud detection, false positives and false negatives have different practical consequences. A false negative may allow a fraudulent transaction to pass undetected, while a false positive may inconvenience a legitimate customer or create additional review workload. For this reason, the final threshold and model choice should be linked to the desired balance between fraud capture and operational cost.

Resampling and leakage control

Where: Near the validation methodology section

Any resampling method such as SMOTE should be applied only to the training data within the validation workflow. Applying resampling before the train/test split can leak information into the test set and make performance estimates overly optimistic. The notebook should make clear where resampling is applied and how the held-out test set remains untouched.

Decision threshold rationale

Where: Near the evaluation or final model section

The default probability threshold may not be optimal for an imbalanced fraud detection problem. The selected threshold should be justified using precision, recall, F1-score, PR-AUC or a cost-sensitive objective, depending on whether the project prioritises catching more fraud cases or reducing false alerts.

Role of neural network models

Where: Near the model comparison section

If neural networks are included, their role should be clearly explained. They may be candidate final models or benchmarks against classical machine learning approaches. The final selection should be justified using validation performance, minority-class metrics, overfitting behaviour and interpretability considerations.

AI-Assisted Viva Answer Guidance

Q1. Why is accuracy not enough for this project?

Accuracy is not enough because the detected project profile involves imbalanced classification and fraud detection. In that context, a model can achieve high overall accuracy while still missing many minority-class cases. Precision, recall, F1-score and PR-AUC are more informative because they focus on the fraud class and on the precision-recall trade-off.

Q2. How did you make sure resampling did not create leakage?

I would verify that resampling is applied only to the training data and ideally inside the validation workflow. If SMOTE or any similar method is applied before the train/test split, synthetic information can leak into the test set and make the results overly optimistic. The held-out test set should remain untouched until final evaluation.

Q3. How did you choose your decision threshold?

ModelHawk detected threshold or probability-based decision logic, so I would explain the threshold as a deliberate modelling choice rather than using the default 0.5 blindly. In fraud detection, lowering the threshold can improve recall and catch more fraud cases, but it can also increase false positives. The chosen threshold should therefore be justified using precision, recall, F1-score, PR-AUC or an explicit cost trade-off.

Q4. Which metric would you prioritise if precision and recall move in opposite directions?

If precision and recall move in opposite directions, the priority depends on the practical cost of each error. For fraud detection, recall is often important because missing fraud can be costly, but precision also matters because too many false alerts can create operational burden and affect legitimate users. The final choice should be linked to the business or risk objective.

Q5. How would you test whether this model generalises to new data?

Generalisation should be tested using held-out data, cross-validation where appropriate and, ideally, more recent or external data. For fraud detection, it is also important to consider data drift because fraud patterns can change over time. Strong validation should check that the model is not only fitting the current dataset but remains reliable under changing conditions.

Q6. What are the main limitations of this project?

The main limitations to discuss are that notebook-level validation does not prove real-world reliability, fraud patterns may change over time, the selected threshold may depend on operational costs, and additional validation would be needed before any deployment decision. The notebook should connect these limitations to future work such as external validation, monitoring and data-drift checks.

AI Layer Limitations

- This AI examiner layer uses the static ModelHawk audit as evidence and does not execute the notebook.
- The configured AI provider can improve wording and interpretation, but it cannot prove that the methodology is correct.
- The audit can detect code and markdown signals, but manual review is still needed to verify data leakage, metric calculation and validation design.
- The generated text should be treated as technical guidance, not as a guarantee of model quality or deployment readiness.

Viva and Interview Preparation

These questions are designed to help the author defend modelling choices, evaluation strategy and project limitations in a technical discussion.

1. Why is accuracy not enough for this project?
2. How did you make sure resampling did not create leakage?
3. How did you choose your decision threshold?

4. Which metric would you prioritise if precision and recall move in opposite directions?
5. How would you test whether this model generalises to new data?
6. What are the main limitations of this project?

Detected Technical Signals

The following static signals were detected from the notebook code and markdown.

- Acknowledges Accuracy Limitation
- Discusses Resampling Safety
- Has Markdown Explanation
- Mentions Business Context
- Mentions Conclusion
- Mentions Imbalance
- Mentions Limitations
- Uses Accuracy
- Uses Anomaly Signals
- Uses Baseline
- Uses Classification Model
- Uses Classification Report
- Uses Confusion Matrix
- Uses Cross Validation
- Uses Deep Learning Signals
- Uses Early Stopping
- Uses Hyperparameter Search
- Uses Learning Curves
- Uses Multiple Models
- Uses Pipeline
- Uses Prauc
- Uses Precision Recall F1
- Uses Random State
- Uses Requirements Or Dependencies
- Uses Roc Auc
- Uses Smote
- Uses Stratify

- Uses Threshold Tuning
- Uses Train Test Split
- Uses Train Validation Split

Final Note

Generated by ModelHawk. This automatically generated report should be interpreted as technical guidance, not as a final guarantee of correctness. A complete review would require executing the notebook, validating the data pipeline, checking runtime outputs and reviewing the modelling decisions in context.