Mini Project 01 — Titanic Classification

Dataset Source: Kaggle Titanic Dataset

www.kaggle.com

Titanic Dataset

Titanic Survival Prediction Dataset

The sinking of the Titanic is one of the most infamous shipwrecks in history.

On April 15, 1912, during her maiden voyage, the widely considered “unsinkable” RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren’t enough lifeboats for everyone on board, resulting in the death of 1502 out of 2224 passengers and crew.

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

In this challenge, we ask you to build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (ie name, age, gender, socio-economic class, etc).

Section A — Problem Definition & Data Preparation

Project Objectives

1 Target Selection

Choose a meaningful **classification target** from the Titanic dataset and clearly state the prediction objective in one concise line.

2 Data Preprocessing Pipeline

- Drop obvious identifiers and free-text columns
- Implement simple imputation strategies
- Encode categorical variables appropriately
- Create stratified train/test split (80/20)

3 Expected Output

Clean, processed datasets ready for modelling: **X_train, X_test, y_train, y_test**

Deliverable: A_minimal_prep_and_split.ipynb

Marks Allocated: 10 marks

Section B — From-Scratch Implementations

NumPy-Only Implementation Challenge

Demonstrate deep understanding by implementing core machine learning algorithms from scratch using only NumPy. This section tests both theoretical knowledge and practical coding skills.

Logistic Regression Variants

- Standard Logistic Regression (no regularisation)
- L1 Regularised Logistic Regression
- L2 Regularised Logistic Regression

Ensemble Methods

- **Simple Bagging:** Configurable levels and estimators
- **Simple Boosting:** AdaBoost-style with decision stumps

API Requirements

- fit(X, y)
- predict_proba(X)
- predict(X, threshold=0.5)

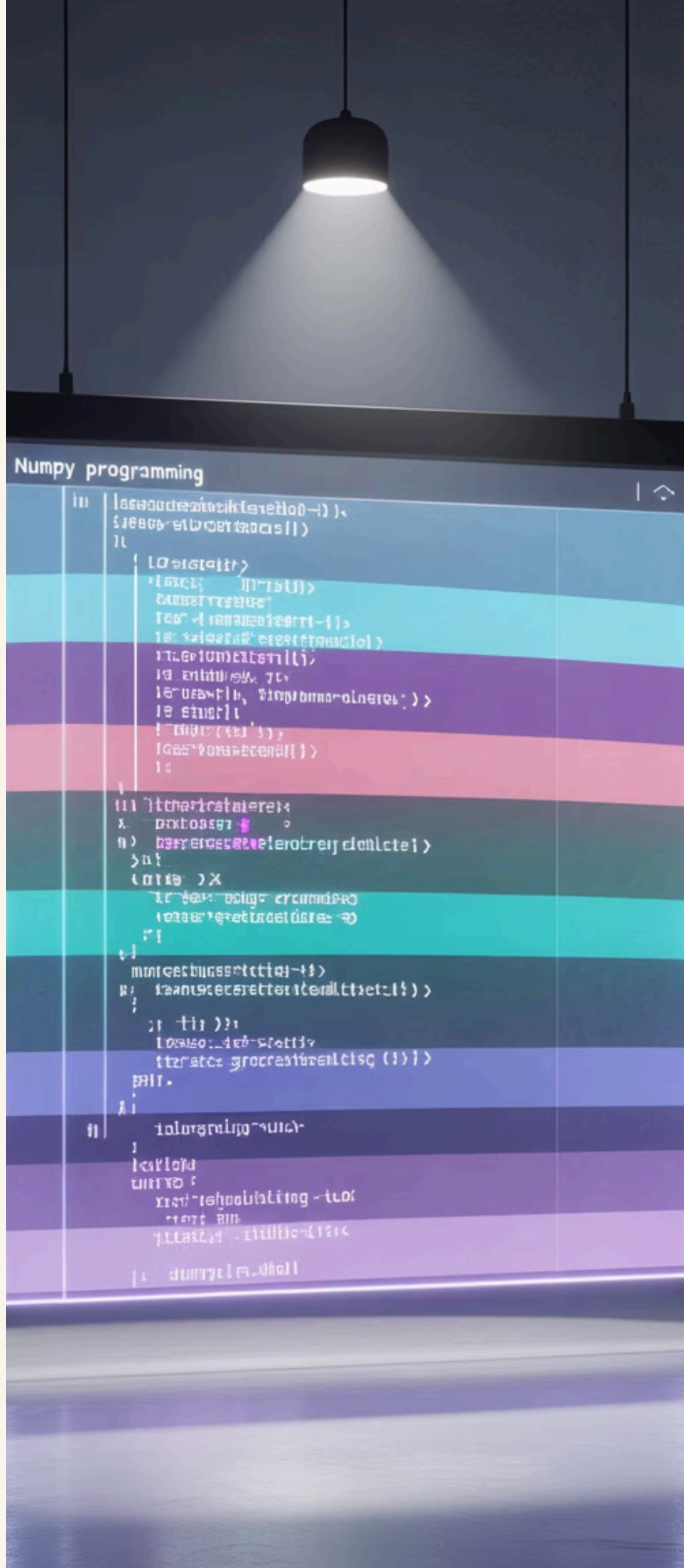
Evaluation Metrics

Implement comprehensive evaluation using **⩾4 metrics** including:

- Confusion matrix visualisation
- ROC or Precision-Recall curve
- Additional performance metrics

⚠ **Deliverable:** B_from_scratch_models.ipynb

Marks: 45 marks (highest weighted section)



Section C — Library Implementations

Logistic Regression

Implement using scikit-learn with three regularisation variants: **none, L1, and L2.**

Bagging Approach

Utilise **RandomForestClassifier** for ensemble bagging methodology.

Boosting Methods

Implement **at least 3 different** boosting classifiers.

Comparative Analysis Framework

This section focuses on leveraging established machine learning libraries to implement the same algorithms as Section B, enabling direct performance comparisons between from-scratch and library implementations.

- Use identical evaluation metrics (⩾4 metrics)
- Apply same train/test split for fair comparison
- Include confusion matrix and curve visualisation
- Generate comparison table against Section B results

Deliverable: C_library_models.ipynb

Marks: 35 marks

Section D — Analysis & Reporting

Comprehensive Project Analysis

Synthesise findings from both implementation approaches to provide meaningful insights into algorithm performance and behaviour.

01

Performance Conclusion

Identify which approaches performed best and provide detailed analysis of why certain methods excelled, including regularisation effects and ensemble method comparisons.

02

Technical Report (~1000 words)

Model-centric write-up covering:

- Minimal preprocessing summary
- Key results comparison table
- Analysis of logistic regression variants
- Bagging vs boosting performance insights

03

Future Recommendations

Propose **2-3 concrete next steps** for model improvement and further research directions.

Deliverable: D_report.pdf

Marks: 10 marks

Project Rules & Constraints

Implementation Requirements

- **Notebooks only** — No separate .py files permitted
- **NumPy restriction** — Section B must use NumPy exclusively
- **No sklearn** metrics or models in from-scratch section

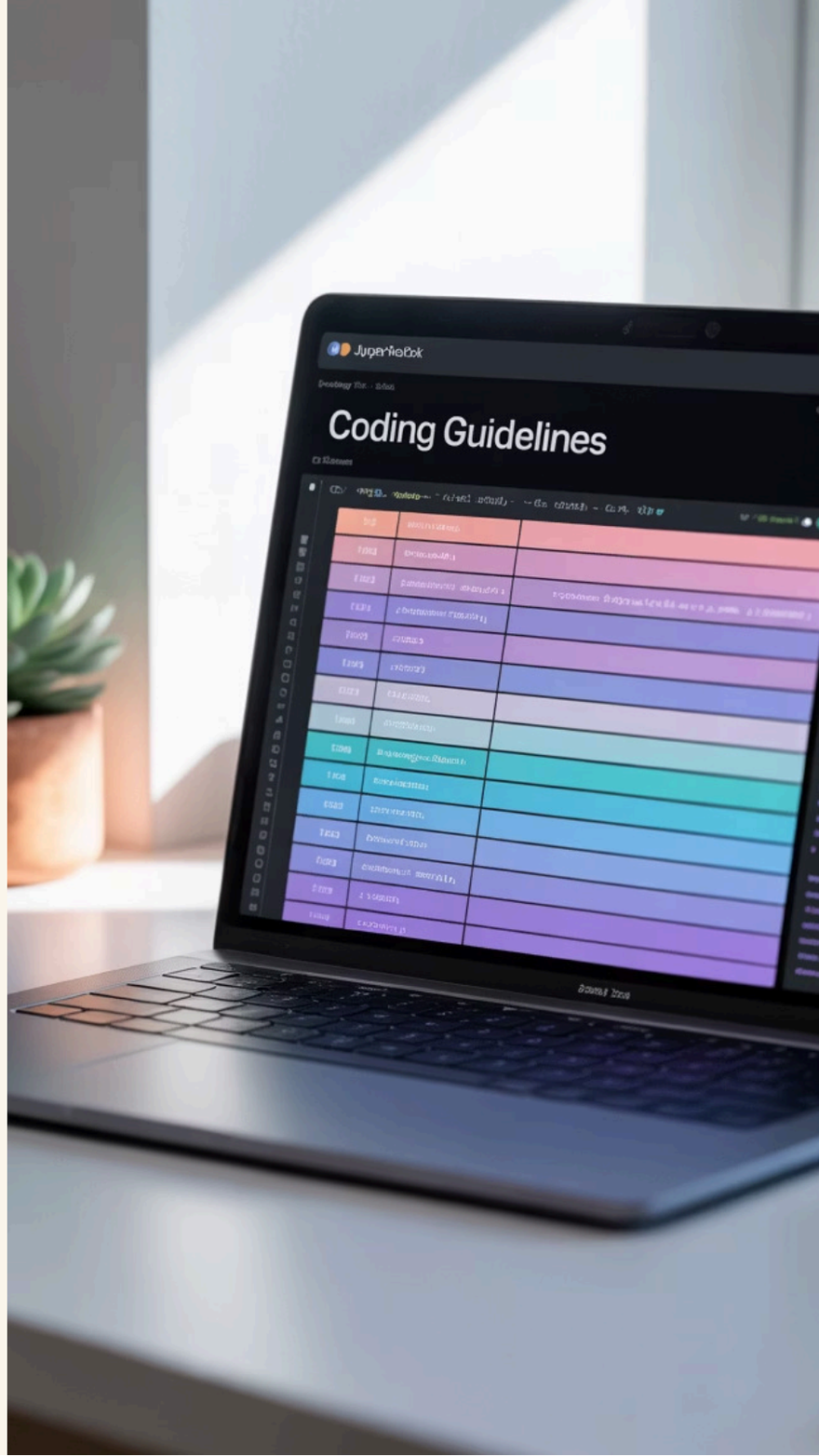
Reproducibility Standards

- Fix a **global seed** (e.g., 42) across all sections
- Use **identical train/test split** throughout project
- Ensure consistent random state for fair comparisons

Performance Expectations

- Keep runtime reasonable (⩽5 minutes on CPU)
- Optimise code for efficiency without sacrificing clarity
- Document any performance considerations

> "Consistency in methodology enables meaningful comparison between implementation approaches and ensures reproducible results."



Submission Requirements

Required Deliverables

- Core Notebooks
  - A_minimal_prep_and_split.ipynb
  - B_from_scratch_models.ipynb
  - C_library_models.ipynb
- Documentation
  - D_report.pdf
  - README.md (execution instructions, package versions)

Submission Format

Compress all files into a single ZIP archive with the naming convention:

'<First Name> <Last Name> <Student ID>.zip'

Example: 'Isuru Alagiyawanna 0001.zip'

Quality Checklist

- All notebooks execute without errors
- Code is well-commented and readable
- Results are clearly presented with appropriate visualisations
- README provides clear setup and execution instructions
- Package versions are documented for reproducibility

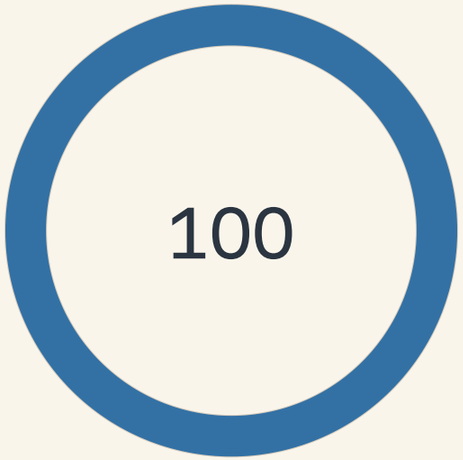
Assessment Breakdown

10	45	35	10
Section A	Section B	Section C	Section D
Minimal preprocessing and correct stratified split implementation	From-scratch implementations: LR variants, bagging, boosting, and code clarity	Library implementations: LR variants, RandomForest, 3+ boosters, fair comparison	Clear, model-focused conclusion and comprehensive report

Success Criteria

Excellence in this project requires demonstrating both theoretical understanding through from-scratch implementations and practical skills using established libraries. The highest marks are awarded for:

- **Mathematical accuracy** in NumPy implementations
- **Code quality** and documentation standards
- **Comprehensive evaluation** with multiple metrics
- **Insightful analysis** comparing different approaches
- **Professional presentation** of results and conclusions



Total Marks

Complete project assessment

Key to Success: Balance theoretical depth with practical implementation skills whilst maintaining clear, reproducible methodology throughout.