OWNML MACHINE LEARNING CANVAS

Designed for: MLOps Designed by: Team 16 Date: Oct 2025 Iteration: 1

PREDICTION TASK

- What is the type of task?

Binary classification - predicting wether a student will achieve high performance (Excellent/Very Good) or lower performance (Good/Average) on the entrance examination

- Which entity are predictions made on?

Individual students taking the entrance examination.

- What are the possible outcomes to predict?

High performance (Excellent/Very Good) or Lower Performance (Good/Average)

- When are outcomes observed?

After the entrance examination is compolicated and graded.

DECISIONS

- How are predictions turned into actionable recommendations or decisions for the end-user? (Mention parameters of the process / application for this.)

For Educational Institutions:
  - Identify at risk students early for targeted interventions.
  - Allocate resources (coaching/tutoring) to students predicted as "Lower Performance"

For students:
  - Receive personalized study recommendations based on weak areas
  - Get early warning (e.g. 3-6 months) before exams to adjust strategy

Key Parameters
  - Prediction confidence threshold: 70%
  - Model provides explainability (which factors drive the prediction)

VALUE PROPOSITION

- Who is the end beneficiary, and what specific pain points are addressed?

Students
Educational Institutions/Coaching Centers
Admissions Officers

- How will the ML solution integrate with their workflow, and through which user interfaces?

The project could be integrated on:

  - Web dashboard (for institutions)

Batch upload student data (csv); view predictions, risk scores, and class-level analytics; generate intervention reports.
  - Student portal (web/mobil app)

Students input their academic profile; receive instant performance predictions with confidence score; get personalized study recommendations
  - API integration

Automated data sync from enrollment databases
  - Report Generation

Downloadable PDF reports for counselors with student predictions and recommended actions.

DATA COLLECTION

- Initial Data Source

Single dataset extract from UCI Machine Learning Repository.
Contains 666 pre-labeled student records with complete features and outcomes.
All data is collected and labeled; no additional data collection required.

- What strategies are in place to update data continuously while controlling cost and maintaining freshness?

For this project phase:
  - No continuous update. We are working with a static dataset for model development and evaluation.
  - Using DVC to track different versions of the dataset for model development and evaluation
  - Single batch approach: train/validate/test split
  - Future consideration: could implement quarterly batch updates

DATA SOURCES

Where can we get data on entities and observed outcomes? (Mention internal and external database tables or API methods.)

- External data source

UCI Machine Learning Repository
Format: CSV file with 666 records and 12 features.
Access method: Direct download from repository

- Data structure

Single csv file
Contains all required features and target variable (Performance)
No API calls needed

- For production scenario (future consideration)

Internal institutional databases; educational board APIs; student survey platforms for self reported data.

IMPACT SIMULATION

- What are the cost/gain values for (in)correct decisions?

Correct predictions have +1 value. FP (predict high, actually lower) cost -2 because at-risk students miss needed support. FN (predict lower, actually high) cost -1 for missed opportunities but students still succeed. Priority: minimize FP to catch struggling students.

- Which data is used to simulate pre-deployment impact?

20% test set (~130 students) for final evaluation. 5-fold cross-validation on training data. Confusion matrix analysis at different thresholds. Test edge cases like mixed academic signals. Metrics: accuracy, precision, recall, F1-score, ROC-AUC, and cost-weighted accuracy.

- What are the criteria for deployment?

Accuracy ≥75%, Recall ≥70% (catch at-risk students), F1-score ≥0.72, ROC-AUC ≥0.75. Must beat baseline (55.1%), provide interpretable features, and pass fairness audit.

- Are there fairness constraints?

Yes. Gender: performance within ±5% for male/female. Caste: equal true positive rates across all categories. Remove biased features if needed. Goal: help all students equitably.

MAKING PREDICTIONS

- Are predictions made in batch or in real time?

Batch predictions made on test set after model training for this academic project.

- How frequently?

One-time prediction on test set for model evaluation

- How much time is available for this (including featurization and decisions)?

There is no strict latency requirements for this use case - prediction don't need to be instantaneous.

- Which computational resources are used?

Same local development machine (laptop/desktop). No GPU needed; CPU based interference. RAM and storage is minimal.

BUILDING MODELS

- How many models are needed in production?

Single binary classification model is sufficient for this project. Will evaluate multiple algorithms (Logistic Regression, Random Forest, XGBoost, SVM) and select best. Final deployment - 1 production model.

- When should they be updated?

For this project: NO updates need.
How much time is available for this (including featurization and analysis)?
Aprox. ~2 weeks from start to final deliverable of phase 1.
Training time per model: <5 minutes (small dataset - 666 records)

- Which computation resources are used?

Local machine (laptop). CPU based training.
Typical specs: 8GB RAM.
Cloud resources are not required. Storage: <1MG for dataset, <10MG for all project files.

FEATURES

- Feature Representations
  - Ordinal Encoded Features (ordered categories)

Class_X_Percentage, Class_XII_Percentage, time (study hours)
  - One-Hot Encoded Features (nominal categories)

Gender, Caste, coaching, Class_ten_education, twelve_education, medium, Father_occupation, Mother_occupation

- Data Transformations
  - Handling Duplicates

Remove 44 duplicate rows
  - Target Variable Binarization

High Performance: Excellent+Vg → Class 1
Lower Performance: Good+Average → Class0
  - Feature Scaling

StandardScaler applied to ordinal features, One-Hot encoded features remain binary
  - Rare Category Grouping

Combine FIVE and SEVEN study hours into "4+" category (only 2 students)
  - Class Imbalance Handling

No aggregation needed - each row is already at student level (entity grain matches prediction grain).

MONITORING

- Which metrics and KPIs are used to track the ML solution's impact once deployed, both for end-users and for the business?

Model Performance Metrics: Accuracy, Precision, Recall, F1-Score, ROC-AUC.
Business Impact KPIs: Student improvement rate after interventions, resource allocation efficiency, prediction accuracy by demographic group, user adoption rate.
Fairness Metrics: Performance parity across gender and caste groups, disparate impact ratio (>0.8).

- How often should they be reviewed?

Ideally in a real life scenario, Technical metrics monthly, business KPIs quarterly, fairness audit semi-annually.