A Data-Driven Multimodal Analysis of How Attendance and Behavioral Factors Influence Academic Success in Higher Education

Course: MSDS696_S70 – Data Science Practicum 2
Author: Balarama Raju Saripalli
Email: bsaripalli@regis.edu
GitHub: https://github.com/varma1234/MSDS696-S70-Practicum-Project-2

Table of Contents

Project Overview

This project examines how attendance patterns, homework completion, classroom engagement, and resilience-related behavioral measures influence academic success. By merging multiple data sources attendance logs, homework records, class participation scores, assessment performance, and mental-health/resilience measures into a single analytical dataset (~12,156 students), the work aims to identify early risk markers and provide actionable guidance for educators to design targeted interventions.

Problem Definition and Motivation

The project frames student success prediction as a supervised classification problem: predict target_pass (binary: pass vs. not pass) using multimodal features that capture attendance behavior, homework completion, engagement, and resilience. Motivation includes: earlier identification of at-risk students, data-driven resource allocation, and evidence-based intervention design. Unlike single-source studies, this project fuses academic, behavioral, and resilience signals to improve prediction and interpretability.

Data Description and Collection Effort

Datasets used (stored in Google Drive under Practicum_Project_2/Data):

- Data/CleanData/ — cleaned component files (attendance_cleaned.csv, homework_cleaned.csv, performance_cleaned.csv, mental_health_cleaned.csv, students_cleaned.csv, mendeley_raw_cleaned.csv)

- Data/Final/ — merged master files (student_master.csv, student_master_updated.csv, student_master_updated_1.csv)
- Data/RawData/ — original sources (SEL dataset, Mendeley student dataset, mental health/resilience dataset, student performance & attendance dataset)

Master dataset summary (final cleaned file used for modeling):

- File loaded: /content/drive/My Drive/Practicum_Project_2/Data/Final/student_master_updated_1.csv
- Raw dataframe (post-merge) shape: (12156, 31) → after dropping all-NaN/time-id fields and IDs: (12156, 16) used for modeling
- Typical features: grade_level, age, attendance_rate, hw_done_rate, avg_score, attendance_pct, homework_completion, class_participation, engagement_index, stress_level, coping_score, sleep_hours, resilience_score, attendance_cat, engage_x_resilience
- Target: target_pass (binary).

Collection effort notes: multiple datasets were joined by student identifier; several columns were engineered (attendance categories, engagement × resilience), and noisy or empty sensor-like fields were removed.

Data Collection & Cleaning (pipeline summary)

1. Identify & load raw files from Data/RawData and Data/CleanData.
2. Drop ID/text columns & all-NaN channels (e.g., communication-text features that had no usable data).
3. Generate target: If raw avg_score was scaled, target was created via median split; where raw pass threshold existed, that was used instead. (Final dataset uses target_pass.)
4. Imputation & normalization: numeric fields imputed (median) and scaled (StandardScaler) as needed during modeling pipelines; categorical fields one-hot encoded.
5. Save cleaned master to /content/drive/My Drive/Practicum_Project_2/outputs/data/clean_master.csv and record processing steps in README.md.

Analysis Approach and Methodology

High-level stages:

1.Preprocessing: drop useless columns, impute, scale, and one-hot encode necessary categoricals (e.g., grade_level).

2.Feature selection: preserve meaningful numeric features and derived aggregates; remove leakage (avg_score removal experiments performed).

3.Balance handling: SMOTE used on training set to handle class imbalance where appropriate.

4.Modeling: baseline and tuned tree models (Random Forest typically primary), logistic regression baseline, optional XGBoost experiments.

5.Explainability: feature importances (RF), permutation importance, SHAP TreeExplainer for best tree model.

6.Validation: stratified train/test split, cross-validation for hyperparameter tuning, ablation studies (e.g., remove age) and permutation importance to ensure robustness.

Evaluation metrics: Accuracy, F1-score, ROC-AUC, confusion matrix, precision/recall at class level.

EDA, Feature Engineering, Modeling & Evaluation, Analysis and Improvement : Exploratory Data Analysis (key findings1`)

- Target distribution: target_pass ≈ 60% pass vs 40% fail.

- Important patterns: attendance rate, engagement index, and homework completion show meaningful separation by target in boxplots/summary statistics.

- Correlations: low–moderate correlations found between attendance/engagement/resilience indicators; no single feature dominated after leakage removal.

- Missingness: several feature columns were fully NaN and dropped; remaining features have full non-null in cleaned master.

Feature Engineering

- Engineered features include attendance_cat (Low/High), engage_x_resilience (engagement × resilience), and normalized versions of raw academic scores.

- Categorical encoding: grade_level → one-hot (Grade 1–5).

- Leakage handling: experiments removed avg_score from predictors and re-evaluated model performance.

Modeling & Evaluation (selected results)

- Pipelines used: ColumnTransformer (numeric imputer + scaler, categorical imputer + one-hot) → SMOTE → Classifier (RandomForest).
- Model performance (examples):
  - RandomForest after leak-removal: Accuracy ≈ 0.504, F1 ≈ 0.556, ROC-AUC ≈ 0.496 (test set).
  - Logistic Regression baseline: Accuracy ≈ 0.507, F1 ≈ 0.551.
  - Earlier (leaky) runs produced inflated perfect metrics (1.0) which were diagnosed and corrected by removing leakage features.
- Feature importance: After rebuilding the preprocessor, RF importances consistently highlight age, grade_level dummies, engagement_index, and engage_x_resilience. Permutation importance confirmed relative ordering (negative permuted decreases indicate usefulness).

- Explainability: SHAP TreeExplainer applied on the top-performing tree model; key SHAP plots and force plots were produced for interpretation.

Analysis & Improvement Steps Taken

- Detected target leakage, removed offending features (notably scaled score duplicates), rebuilt pipeline and re-ran.
- Performed ablation studies (e.g., remove age) to test feature influence; age removal decreased performance slightly, indicating limited but non-dominant effect.
- Reconstructed a robust feature-name alignment pipeline to ensure feature importances and permutation importances map correctly to transformed columns.
- Saved artifacts: models, feature importance CSVs, permutation_importances_fixed.csv, confusion matrices, and plots in /outputs.

Key Tools and Libraries Used

- Python (pandas, numpy)
- scikit-learn (preprocessing, models, ColumnTransformer, RandomizedSearchCV)
- imbalanced-learn (SMOTE, imblearn.pipeline)
- xgboost (optional)
- shap (for explainability)
- matplotlib, seaborn (visualizations)
- joblib (model persistence)
- Google Colab environment & Google Drive for storage

Practicum Deliverables and Evaluation Rubric Alignment

This project targets each rubric item: problem sourcing, data cleaning complexity, rigorous EDA, model building & optimization, handling class imbalance, interpretability, and polished presentation. Specific deliverables include:

- Cleaned master dataset saved under /outputs/data/clean_master.csv
- Modeling notebook(s) and hyperparameter search artifacts (/outputs/models/)
- Visualizations: distribution plots, correlation heatmap, feature importances, ROC/PR curves (/outputs/charts/)
- README and notebook documentation for reproducibility.

Week-by-Week Timeline and Progress

Week	Activities & Status
1	Proposal, problem definition — Done
2	Dataset review and project plan — Done
3	Data sources finalized and folder structure — Done
4	Data cleaning, merging, initial EDA — Done
5	Final master dataset & feature engineering — Done

6 Baseline modeling, deeper EDA, clustering experiments — Done

7 Final evaluation, visualization polishing, dashboard prototyping — Done

8 Final report, presentation recording, repo packaging — Done

Roadblocks and Learnings

- Target leakage was detected (e.g., scaled avg_score duplicates) and required removing features and re-running models — a useful lesson in careful feature audits.
- Feature-name mismatch between transformed arrays and feature importance outputs required building robust feature-name mapping logic.
- Clustering sensitivity to scaling demanded multiple normalization strategies and qualitative checks.
- Compute time: permutation importance and SHAP were computationally heavy; sampling strategies were used for explainability stages.

Lessons learned: build reproducible preprocessing pipelines, validate for leakage early, and always align transformed features to importance outputs.

Online Presence and Documentation

- A professional GitHub repo will include: polished README, notebooks organized by stage (data cleaning, EDA, modeling, explainability), saved outputs under /outputs.
- Executive summary and visuals will be placed on the repo landing page for quick consumption by non-technical stakeholders.
- A small text file with the repo link will be uploaded to WorldClass by the Week 7 deadline.

Visualizations and Dashboard Summary

Outputs saved to /content/drive/My Drive/Practicum_Project_2/outputs/charts/ include:

- Distributions: age, attendance_rate, hw_done_rate, avg_score
- Boxplots by target_pass for key predictors
- Correlation heatmap and cluster visualizations
- RF importances and permutation importance bar charts
- Confusion matrix, ROC and PR curves

Project Structure (repo / drive layout)

Practicum_Project_2/

```
 ├– Data/
   ├– CleanData/
|    ├– Final/
|   └─ RawData/
 ├– notebooks/
|    ├– 01_Quick_peak_for_each_dataset.ipynb
|    ├– 02_data_cleaning_and_merge.ipynb
|    ├– 03_feature_engineering.ipynb
|   └─ 04_exploratory_analysis.ipynb
|   |_05_Modelling.ipynb
|   |_06_Final_Modeling.ipynb
 ├– outputs/
|    ├– charts/
|    ├– data/
|   └─ models/
 ├– README.md
└─ presentation/
└─ Online presence/
└─ Summary/
```

Quick Start Guide

Prerequisites: Python 3.8+, Google Colab or local Jupyter, required packages listed in requirements.txt.

Mount Drive in Colab:

```
from google.colab import drive

drive.mount('/content/drive')
```

Load and run main notebook:

Open notebooks/03_modeling.ipynb and run sequentially. Saved artifacts will appear under /content/drive/My Drive/Practicum_Project_2/outputs/.

Run inference example (after saving final_preprocessor.joblib and final_classifier.joblib):

```
from src.inference import load_models, predict_df

clf, preproc = load_models()
```

```
preds, proba = predict_df(df_new, preproc, clf)
```

Authors & Contact

Balarama Raju Saripalli

Email: bsaripalli@regis.edu

References

1.Kaggle Student Performance and Attendance Dataset https://www.kaggle.com/datasets/marvyaymanhalim/student-performance-and-attendance-dataset

2.Mendeley Student Dataset

https://data.mendeley.com/datasets/4mvgvtxc8s/1

3.CORE Social-Emotional Learning (SEL) Dataset https://dataqualitycampaign.org/wp-content/uploads/2018/03/DQC-CORE-CaseStudy-2018Mar22.pdf

4.Kaggle Student Mental Health & Resilience Dataset https://www.kaggle.com/datasets/ziya07/student-mental-health-and-resilience-dataset

5.Why scientists need to be better at data visualization (Betsy Mason 11.12.2019)

https://knowablemagazine.org/content/article/mind/2019/science-data-visualization

6.IBM white paper about the Foundational Methodology for Data Science.

file:///C:/Users/balar/Downloads/IBMOpenSource_FoundationalMethologyforDataScience%20(1).PDF

7.Essential tools for Data Science

https://www.youtube.com/watch?v=pAXeCpwKgYg

8.Big Data Management. (n.d.). Retrieved November 05, 2020, from

https://www.datamation.com/big-data/big-data-management.html

9.The Belmont Report. (2010, January 28). HHS.Gov.

https://www.hhs.gov/ohrp/regulations-and-policy/belmont-report/index.html

10.Descriptive and Inferential Statistics. (n.d.). Retrieved from

https://luminousmen.com/post/descriptive-and-inferential-statistics

11.Trochim, P. (n.d.). Descriptive Statistics. Retrieved from

https://conjointly.com/kb/descriptive-statistics/

12.What is Descriptive Statistics? - Data Science and Data Analytics. (2017, February 28). Retrieved from

https://www.cognixia.com/blog/what-is-descriptive-statistics

13.Gottfried, in press at Urban Education https://www.attendanceworks.org/wp-content/uploads/2017/09/AW-gottfried_chronic_peers-2.pdf

14.A Motivational Perspective

on Engagement and Disaffection

Conceptualization and Assessment

of Children’s Behavioral and

Emotional Participation in

Academic Activities in the Classroom

Ellen A. Skinner

Thomas A. Kindermann

Portland State University, Oregon

Carrie J. Furrer

NPC Research

https://qqseminar.weebly.com/uploads/1/5/2/4/152407464/skinner-et-al-2008-a-motivational-perspective-on-engagement-and-disaffection-conceptualization-and-assessment-of.pdf

15.Project drive path

https://drive.google.com/drive/folders/1p6k0R9_QJVPjRFnxLE4XmdHlA7EXplwA?usp=sharing

16.ChatGPT (OpenAI)

OpenAI. (2025). ChatGPT (version GPT-5.1) [Large language model]. https://chat.openai.com/

17.Perplexity AI

Perplexity AI. (2025). Perplexity AI conversational search engine [AI model]. https://www.perplexity.ai/

Note : Every reference i went through toroughly make how good to know about tecnical concepts and data sceince manadate priciples to know and can be used based on requirement to progress in each step of project and these references helped me to complete the end to end project and i used AI models for formating the documents , asking suggestions when i got stuck in middle , debugging code and learning the concepts regarding my project