BRIGHTPATH GRADE PREDICTOR

Guided Project

Institution: Belgium Campus

Course: MLG382

Group S:

- Jeremia Fourie (577881)
- Juan Oosthuizen (600161)
- Busisiwe Radebe (601255)
- Phumlani Ntuli (577529)

Submission Date: 22 April 2025, 23:59 PM

Table of Contents

Link to GitHub repository: https://github.com/jeremiafourie/BrightPath-Grade-Predictor

Link to Dash App hosted by render.com: https://brightpath-grade-predictor.onrender.com/performance_prediction

Problem Statement

BrightPath Academy faces challenges in identifying at-risk students early, understanding how extracurricular activities influence grades, and developing targeted support strategies. This project addresses these issues by building a predictive model for GradeClass and analyzing key factors affecting student outcomes.

Hypotheses

We hypothesize that:

- Students with higher StudyTimeWeekly are more likely to achieve better grades.
- Higher Absences correlate with lower grades.
- Participation in Extracurricular activities positively impacts grades.
- ParentalSupport levels significantly influence student performance.

These hypotheses will be explored and tested in the notebooks/eda.ipynb notebook through data visualizations and statistical analysis.

Preparing Data

In the Preparing Data phase, we first loaded the raw student dataset (2,392 records, 15 columns), validated that StudentID was unique for every row—confirming it served only as an identifier—and then dropped it to avoid introducing non-predictive noise. Next, we ran data.info() to verify there were no missing values and that all remaining features were numeric, and used data.describe() to see that age clustered tightly (15–18 yrs, median 16), StudyTimeWeekly varied widely (IQR ≈ 9.7 hrs, max ≈ 20 hrs), and absences were right-skewed (median 15 days, max 29). We also checked for duplicate rows (none found), applied IQR-based outlier detection to flag extreme study-time and absence cases for review, and computed skewness and kurtosis to highlight any features needing transformation or scaling.

Exploratory Data Analysis (EDA)

To understand our cleaned student dataset, we first split variables into three types—numerical (StudyTimeWeekly, Absences, GPA), categorical (Gender, Ethnicity, Tutoring, Extracurricular, Sports, Music, Volunteering) and ordinal (Age, ParentalEducation, ParentalSupport, GradeClass)—so that each group could be visualized with the most informative chart: histograms/KDEs for continuous measures, pie charts for binary flags, and countplots for ordered categories. This targeted approach reveals both the shape of each feature's distribution and its relationship to student performance.

Univariate Analysis

- StudyTimeWeekly: Most students study between 0–10 hrs/week; a long right tail out to ~20 hrs identifies a small "super-studier" subgroup whose habits may merit separate analysis or outlier treatment.
- Absences: Centered around 10–20 days with a tail up to 29, indicating a handful of chronic absentees; these extreme cases should be flagged for potential capping or deeper review.
- GPA: Peaks at ~1.5–2.0, but also shows a secondary spike at the maximum (4.0), suggesting ceiling effects or grade inflation among high achievers.
- Gender & Ethnicity: Gender is nearly balanced (51 % vs. 49 %), while Ethnicity is dominated by category 0 (50.5 %), with three smaller groups making up the rest—allowing us to keep all levels without severe sparsity.
- Participation Flags: Tutoring (~30 %), Extracurricular (~38 %), and Sports (~30 %) show healthy variation; Music (~20 %) and Volunteering (~16 %) are less common but still sufficiently represented to include as features.
- Age, ParentalEducation & ParentalSupport: Ages 15–18 are almost evenly split (slight peak at 15). ParentalEducation clusters at level 2 (≈ 940 records) and ParentalSupport at level 2–3 (≈ 700 each), indicating most students come from moderately educated, moderately supportive homes.
- GradeClass: Strongly skewed toward the top class (4) with ~1,210 students versus only ~110 in class 0, flagging a class-imbalance issue for modeling.

Bivariate Analysis

- Numeric vs. GradeClass (Boxplots): Both median StudyTimeWeekly and GPA rise steadily from lower to higher GradeClass, while median Absences fall—confirming our hypotheses that more study time and better attendance relate to higher grades.
- Categorical vs. GradeClass (Grouped Bars): The proportion of students in tutoring and extracurriculars grows with GradeClass, suggesting these supports are associated with better outcomes. Music and volunteering show smaller but consistent gains.
- Correlation Matrix: A very strong negative correlation between Absences and GPA (≈ –0.92) and a modest positive link between StudyTimeWeekly and GPA (≈ +0.18) highlight attendance as the single most powerful univariate GPA predictor, with study time also important but to a lesser degree. ParentalSupport shows a weaker yet meaningful positive correlation with GPA (≈ +0.19).
- Pairwise Scatter & KDE by GradeClass: Lower-class students cluster in the high-absence, low-GPA region, while upper-class students concentrate at low absences and high GPA. Overlap in StudyTime vs. GPA suggests some high-achievers study less, warranting deeper segmentation.

- Group-Level Summary: As GradeClass increases from 0 to 4, average study time and GPA both decline while absences climb dramatically—underscoring how worsening attendance and shrinking study habits drive lower performance.

```
Mean StudyTime, Absences & GPA by GradeClass:
             StudyTimeWeekly   Absences       GPA
GradeClass
0.0                11.854926   5.747664  3.102942
1.0                11.122335   5.312268  3.001673
2.0                10.106404   7.250639  2.659742
3.0                 9.757963  11.427536  2.215545
4.0                 9.184822  20.786953  1.208041
```

Preprocessing Data

In this phase, we transform our cleaned student dataset into its final, model-ready form by engineering new features and encoding all predictors as numeric.

- Target construction & leakage prevention: We first create the ordinal target variable GradeClass by applying our gpa_to_grade_class mapping to each student's continuous GPA score, assigning them to one of five performance bands. To avoid leakage, we then drop the original GPA column from the training set.
- Engagement score: To capture each student's overall non-academic involvement, we aggregate the five binary participation flags—Tutoring, Extracurricular, Sports, Music, and Volunteering—into a single Engagement feature by summing them row-wise.
- Family support index: We combine ParentalEducation and ParentalSupport into a single FamilySupport index by taking their product. This interaction term reflects the joint effect of parents' educational background and the level of support they provide.
- Categorical encoding: Finally, we convert any remaining nominal or categorical variables into numeric form via one-hot encoding, dropping the first level of each to prevent multicollinearity.

The end result is an "engineered" dataset of 2,392 students, with all predictors (including Engagement, FamilySupport, and the dummy indicators) in numeric form alongside our target GradeClass. We export this to engineered_data.csv for all downstream modeling steps.

Model Evaluation Metrics

To ensure a consistent and transparent comparison across all candidate models, we hold out 20 % of the data via a stratified train–test split (preserving the GradeClass distribution) and evaluate each model on the same test set using:

- Accuracy: the overall fraction of correctly classified students.
- Precision, Recall & F_1-score (per class and aggregated): to gauge how well each model balances false positives and false negatives, especially important given the skew toward higher GradeClasses.
- Confusion Matrix (both raw counts and normalized): to visualize which grade bands are most frequently confused, guiding targeted improvements.

All metrics are computed automatically by our evaluate_and_save helper, which logs model accuracy, prints the full classification report, and renders a side-by-side heatmap of the normalized and raw confusion matrix.

Model Building: Part 1 (Baseline Models)

We began with three classical classifiers to establish performance benchmarks on our 80/20 stratified split. A logistic regression (C=1.0, liblinear solver, one-vs-rest) achieved 76.2 % test accuracy, reliably identifying top-performing students but often confusing lower GradeClasses. Introducing a balanced random forest (200 trees, max depth 12) yielded a modest lift to 77.0 %, with more consistent recall across the mid-range classes. Finally, our baseline XGBoost (150 trees, learning rate 0.05, max depth 4, 80 % subsampling) further improved accuracy to 78.9 %, demonstrating stronger precision and recall especially in the highest and lowest grade bands. These results highlight that tree-based ensembles capture non-linear patterns missed by linear models, setting a solid foundation for more advanced approaches.

Model Building: Part 2 (Deep Learning)

To capture deeper interactions among our engineered features—such as the Engagement score and FamilySupport index—we developed a feed-forward neural network with two hidden layers (128 and 64 units) each followed by 30 % dropout, and a softmax output across the five GradeClasses. Compiled with the Adam optimizer and trained for 50 epochs (20 % validation split), the MLP attained 80.8 % accuracy on the held-out test set. Notably, it outperformed all baseline learners in recall for under-represented GradeClasses, indicating its capacity to disentangle complex, non-linear relationships that simpler models overlook. We saved the trained network (DeepLearningMLP.keras) for downstream deployment.

Model Building: Experimentation

To further refine predictive accuracy and address class imbalance, we conducted three targeted experiments using randomized hyperparameter search and class-balancing techniques. First, scaling inputs and switching to a multinomial logistic regression (lbfgs, max_iter 1000) propelled accuracy to 82.3 %, improving overall F_1 scores without sacrificing interpretability. Second, we applied SMOTE oversampling alongside a deeper random forest (n_estimators 200, max_depth 30, min_samples_split 10), which boosted minority-class recall but reduced overall accuracy to 74.1 %. Third, we fine-tuned XGBoost's parameters (subsample, colsample_bytree, learning_rate, depth), nudging its accuracy to 79.1 %. Together, these experiments underscore the trade-offs between raw accuracy and balanced class performance, guiding our final model selection toward the scaled logistic regression for its blend of simplicity, interpretability, and strong predictive power.

Model Deployment

We packaged our GradeClass predictor into an interactive Dash web application structured with four main pages: Overview, Performance Prediction, Model Comparison, and About. Upon startup, the app loads pre-trained model artifacts (both joblib and Keras formats) from the artifacts/ directory. Users can input student features, select a model, and instantly view the predicted grade class alongside confidence scores, all within a responsive Bootstrap-themed interface.

To make the application publicly accessible, we deployed it on Render.com. The full repository—including requirements.txt, Python modules, and the artifacts/ folder—was pushed to GitHub and linked to Render. We configured the build command as pip install -r requirements.txt and the start command to launch the Dash server with Gunicorn. With automatic deploys enabled on Git pushes, Render provides a stable, scalable endpoint for stakeholders to interact with the model live.