Predict Students' Dropout and Academic Success

Grupo 42

Alexandre Morais

Carlos Costa

Nuno Ramos

Problem Definition

- Develop and evaluate machine learning models capable of learning from labeled data to make accurate predictions regarding a specific target variable, focusing on supervised learning techniques in the context of classification problems.

- The dataset used is from "Early prediction of student’s performance in higher education: a case study" which looks to predict students' dropout and academic success, making use of 4424 instances and 36 different features.

- Our target variable is defined by three categories: “Dropout”, “Enrolled” and “Graduate”, reflecting the state of the students at the end of the normal duration of the course.

Related Work

- **Ensemble Learning Approach** by Shivam Singh. A project on GitHub applied ensemble methods (Voting Classifier combining RF, SVM, and k-NN) using the same dataset as us.
- **Comparative Study** by Hanah Kim. Compared Neural Networks vs. Random Forests, finding RF slightly better (92.05 % vs. 91.71 %) in predicting students' drop out in the U.S.

Methodology

- Some of the tools and libraries used are as follows:
  - Python
  - Anaconda
  - Jupyter Labs
  - NumPy
  - SciPy
  - Pandas
  - Scikit-Learn
  - MatPlotLib
  - Seaborn

Preprocessing

- Encoding of objects into numerical, so models could read it
- Removal of extreme outliers
- Feature Selection, by:
  - Removing highly correlated (redundant) numerical features
  - Removing low-variance features (for numerical columns)
  - Removing statistically insignificant features, by the Chi-Square Test of Independence
- SMOTE

Machine Learning Models

Decision Tree Classifier

- Rule-based algorithm splits data based on feature values to create a tree structure for prediction.
- Used with default parameters: Gini impurity for splitting, unlimited depth.
- **Pros:**
  - Highly interpretable and easy to visualize.
  - Handles both numerical and categorical data without preprocessing.
  - Fast training and inference time.
- **Cons:**
  - Prone to overfitting, especially without pruning or regularization.
- **Justification:** Used as a strong baseline model for comparison due to its simplicity and explainability.

Machine Learning Models

k-Nearest Neighbors (k-NN)

- An instance-based, non-parametric method that classifies a data point based on the majority class among its k closest neighbors.
- Implemented with k=5 using Euclidean distance.
- **Pros:**
  - Simple and intuitive algorithm.
  - No training phase — stores data and classifies at runtime.
- **Cons:**
  - Performance can degrade with high-dimensional or noisy data.
  - Sensitive to feature scaling and irrelevant features.
  - Computationally expensive during testing (especially with large datasets).
- **Justification:** Chosen for its simplicity and ability to capture local patterns in the data.

Machine Learning Models

Multilayer Perceptron (Neural Network)

- A feedforward artificial neural network trained using the MLPClassifier from scikit-learn.
- Used default settings with an ‘adam’ optimizer and hidden layer configuration.
- **Pros:**
  - Capable of modeling complex, non-linear relationships.
  - Learns high-level feature interactions automatically.
- **Cons:**
  - Requires more time and resources to train.
  - Less interpretable compared to tree-based models.
  - Sensitive to hyperparameter tuning and input scaling.
- **Justification:** Included for its power in learning deep patterns in the data. Despite longer training time, it performed strongly on accuracy and F1-score.

Machine Learning Models

Support Vector Machine (SVM)

- A supervised learning model that constructs an optimal hyperplane to separate classes with maximum margin.
- Implemented with an RBF (Radial Basis Function) kernel for handling non-linear decision boundaries.
- **Pros:**
  - Effective in high-dimensional spaces.
  - Robust and performs well with clear class separation.
- **Cons:**
  - Training time increases significantly with larger datasets.
  - Less interpretable than simpler models like Decision Trees.
- **Justification:** Chosen for its balance between accuracy and robustness. Demonstrated consistently high F1-scores, especially in complex classification boundaries.

Evaluation and comparison of models

Model	Accuracy	Precision	Recall	F1-Score	Training Time
Decision Tree	0.648190	0.597999	0.601829	0.598216	0.2863
k-NN	0.685520	0.639377	0.623935	0.626814	0.0426
Neural Network	0.723982	0.668335	0.662918	0.665243	107.2632
SVM	0.753394	0.707414	0.692459	0.697823	34.4578

Model	Testing Time
Decision Tree	0.0110
k-NN	0.0629
Neural Network	0.0680
SVM	1.4602