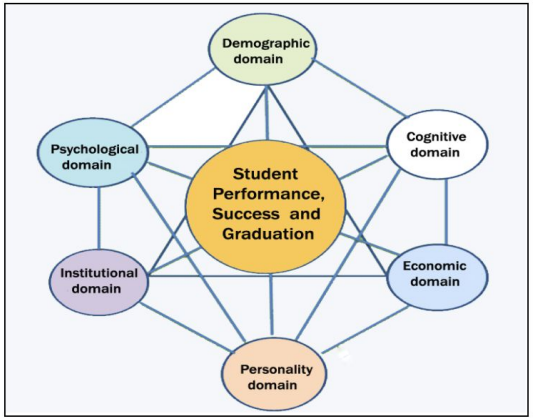
Prediction of Student's Performance and Academic Success

A Comparison of Machine Learning Models

Ananya Mantravadi, Nainisha Bhallamudi, Pavithra Velumani

Introduction & Problem Statement

- Academic success is essential for both individuals and society
- Academic success fosters social justice by providing equitable access to education
- Predict student academic performance in higher education institutions.
- Identify students at risk of failure or underperformance early on.
- Use machine learning techniques to build predictive models.
- Provide early identification to enable timely interventions and support for at-risk students.
- We focus on 3 main categories: Enrolled, Graduate and Dropout



Related Work

1. "Early Prediction of student's Performance in Higher Education: A Case Study" - Mónica V. Martins, Daniel Tolledo, Jorge Machado, Luís M. T. Baptista, and Valentim Realinho
   Used techniques of Boosting and compared various models

	Logistic Regression	Support Vector Machine	Decision Tree	Random Forest
F1-score Failure	0.63	0.53	0.63	0.66
F1-score Rel.Success	0.41	0.31	0.39	0.37
F1-score Success	0.69	0.71	0.75	0.82
Average F1-score	0.58	0.52	0.59	**0.62**
Accuracy	0.61	0.60	0.65	**0.72**

	Gradient Boosting	Extreme Gradient Boosting
F1-score Failure	0.68	0.68
F1-score Rel.Success	0.44	0.44
F1-score Sucess	0.81	0.83
Average F1-score	0.65	**0.65**
Accuracy	0.72	**0.73**

Disadvantages of previous methods

Class Imbalance: The dataset shows a significant imbalance between the target classes, with "Success" being the majority. Despite attempts to mitigate this using techniques like SMOTE and ADASYN, challenges persist due to highly imbalanced data.

Solution: Using Repeated Stratified K-Fold cross-validation

Feature Importance: The paper lacks insight into the importance of pre-enrollment features for predicting student success, hindering the development of effective interventions.

Solution: Use of Feature Engineering and Data Preprocessing

```
[ ]  # Check class distribution
     print(df['Target'].value_counts())

     Target
     Graduate    2209
     Dropout     1421
     Enrolled     794
     Name: count, dtype: int64
```

Our Method - Dataset

Predict Students' Dropout and Academic Success

Donated on 12/12/2021

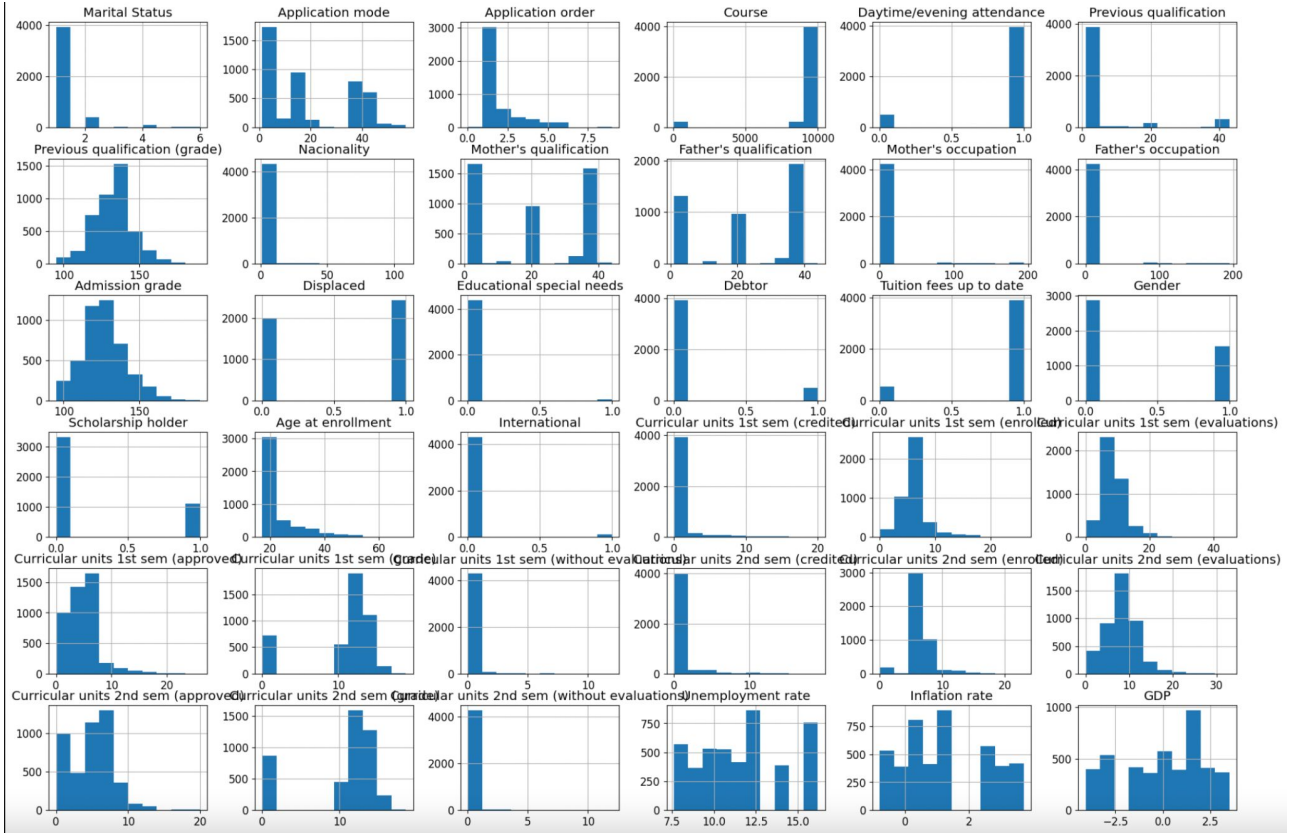
A dataset created from a higher education institution (acquired from several disjoint databases) related to students enrolled in different undergraduate degrees, such as agronomy, design, education, nursing, journalism, management, social service...

Dataset Characteristics	Subject Area	Associated Tasks
Tabular	Social Science	Classification
Feature Type	**# Instances**	**# Features**
Real, Categorical, Integer	4424	36

Target - Enrolled, Graduate, and Dropout

#	Column
0	Marital Status
1	Application mode
2	Application order
3	Course
4	Daytime/evening attendance
5	Previous qualification
6	Previous qualification (grade)
7	Nacionality
8	Mother's qualification
9	Father's qualification
10	Mother's occupation
11	Father's occupation
12	Admission grade
13	Displaced
14	Educational special needs
15	Debtor
16	Tuition fees up to date
17	Gender
18	Scholarship holder
19	Age at enrollment
20	International
21	Curricular units 1st sem (credited)
22	Curricular units 1st sem (enrolled)
23	Curricular units 1st sem (evaluations)
24	Curricular units 1st sem (approved)
25	Curricular units 1st sem (grade)
26	Curricular units 1st sem (without evaluations)
27	Curricular units 2nd sem (credited)
28	Curricular units 2nd sem (enrolled)
29	Curricular units 2nd sem (evaluations)
30	Curricular units 2nd sem (approved)
31	Curricular units 2nd sem (grade)
32	Curricular units 2nd sem (without evaluations)
33	Unemployment rate
34	Inflation rate
35	GDP
36	Target

Raw data histogram



Our Method - Feature Engineering

- **Importance of Occupation-Based Feature Engineering:** Each occupation is mapped to predefined categories like 'Management,' 'Technical Workers,' and 'Service Workers' to balance complexity reduction and essential information retention.

- **Feature Engineering for Qualification**: Feature engineering was performed to categorize parental qualifications into broader categories, facilitating a more concise yet informative representation of the data, which is crucial for subsequent analysis and modeling.

Before

0 - Student 1 - Representatives of the Legislative Power and Executive Bodies, Directors, Directors and Executive Managers 2 - Specialists in Intellectual and Scientific Activities 3 - Intermediate Level Technicians and Professions 4 - Administrative staff 5 - Personal Services, Security and Safety Workers and Sellers 6 - Farmers and Skilled Workers in Agriculture, Fisheries and Forestry 7 - Skilled Workers in Industry, Construction and Craftsmen 8 - Installation and Machine Operators and Assembly Workers 9 - Unskilled Workers 10 - Armed Forces Professions 90 - Other Situation 99 - (blank) 101 - Armed Forces Officers 102 - Armed Forces Sergeants 103 - Other Armed Forces personnel 112 - Directors of administrative and commercial services 114 - Hotel, catering, trade and other services directors 121 - Specialists in the physical sciences, mathematics,

After

```
occupation_categories = {
    'Student': [0],
    'Management': [1, 112, 114],
    'Intellectual_Workers': [2, 121, 122, 123, 124, 125],
    'Technical_Workers': [3, 131, 132, 134, 135],
    'Administrative_Workers': [4, 141, 143, 144],
    'Service_Workers': [5, 151, 152, 153, 154],
    'Agriculture_Fisheries': [6, 161, 163],
    'Skilled_Workers': [7, 171, 172, 173, 174, 175],
    'Machine_Operators': [8, 181, 182],
    'Unskilled_Workers': [9, 191, 192, 193, 194, 195],
    'Military': [10, 101, 102, 103],
    'Other_Situations': [90, 99]
}
```

Experimental Settings

Train Test Split:

- The dataset is divided into Training and Test sets with an 80% and 20% split.
- Purpose: Evaluate model performance on unseen data.

Repeated Stratified K-Fold Cross-Validation:

- Used due to class imbalance.
- 10 Folds: Each fold maintains class proportions.
- Repeated thrice and took the mean of them

Hyperparameter Tuning:

Technique: RandomizedSearchCV

- Initially attempted GridSearchCV, but due to its exhaustive nature it is time consuming.
- Randomized parameter search over specified hyperparameters.
- Balances thoroughness with computational efficiency.

Experimental Settings

Evaluation Metric Metric:

- Accuracy and F1-Macro (F1 score provides a balance between precision and recall).
- Macro averaging for class imbalance handling. It takes the F1 scores of each class and averages them, treating all classes equally.

Process Optimization:

Runtime Optimization:

- Changed runtime type from CPU to TPU in Google Colab.
- Significant reduction in processing time.

Parallel Processing:

- Utilized to expedite the hyperparameter tuning process.
- All available cores used for efficiency.

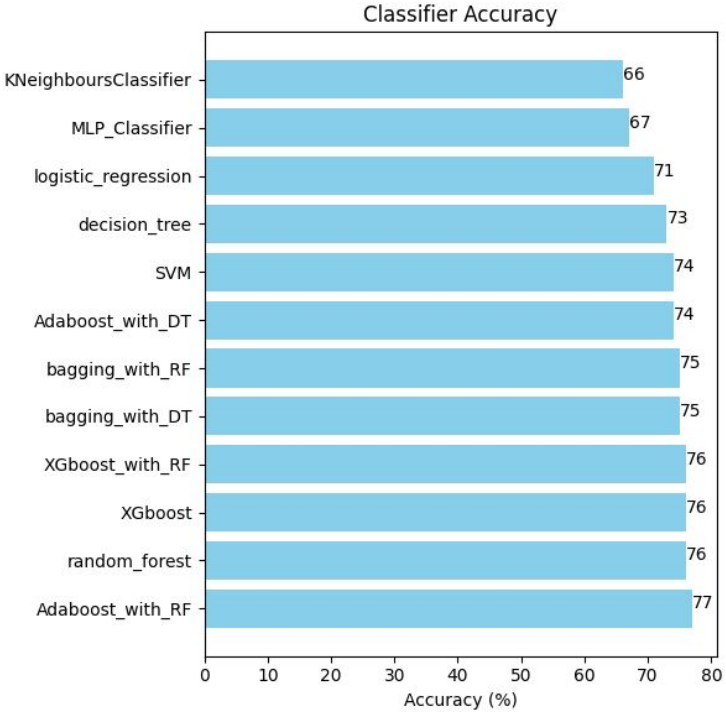
Models Explored

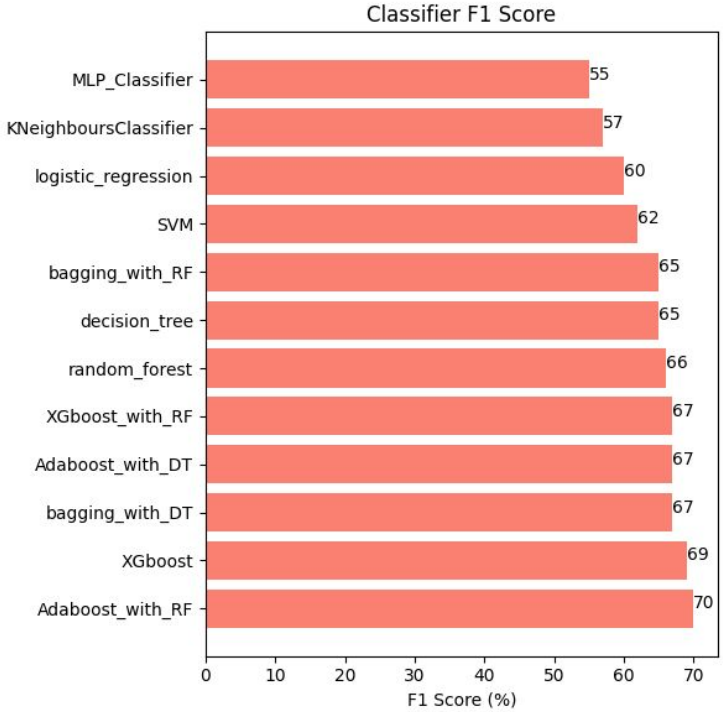
Experimented with a wide range of models as a robust method to establish baseline performance.

- Logistic Regression
- Random Forest
- Decision Tree
- Bagging with DT
- Bagging with RF
- Adaboost with DT
- Adaboost with RF
- K Nearest Neighbours
- SVM
- XGboost
- XGboost with RF
- MLP Classifier
- Voting

Baseline Model Results

Adaboost with RF (77% accuracy, 70% F1 score) and Random Forest (76% accuracy, 66% F1 score).





Final Results - Voting Classifier

Voting Ensemble:
In our experimentation, the voting ensemble outperformed stacking.
Used a soft voting ensemble with the following models:

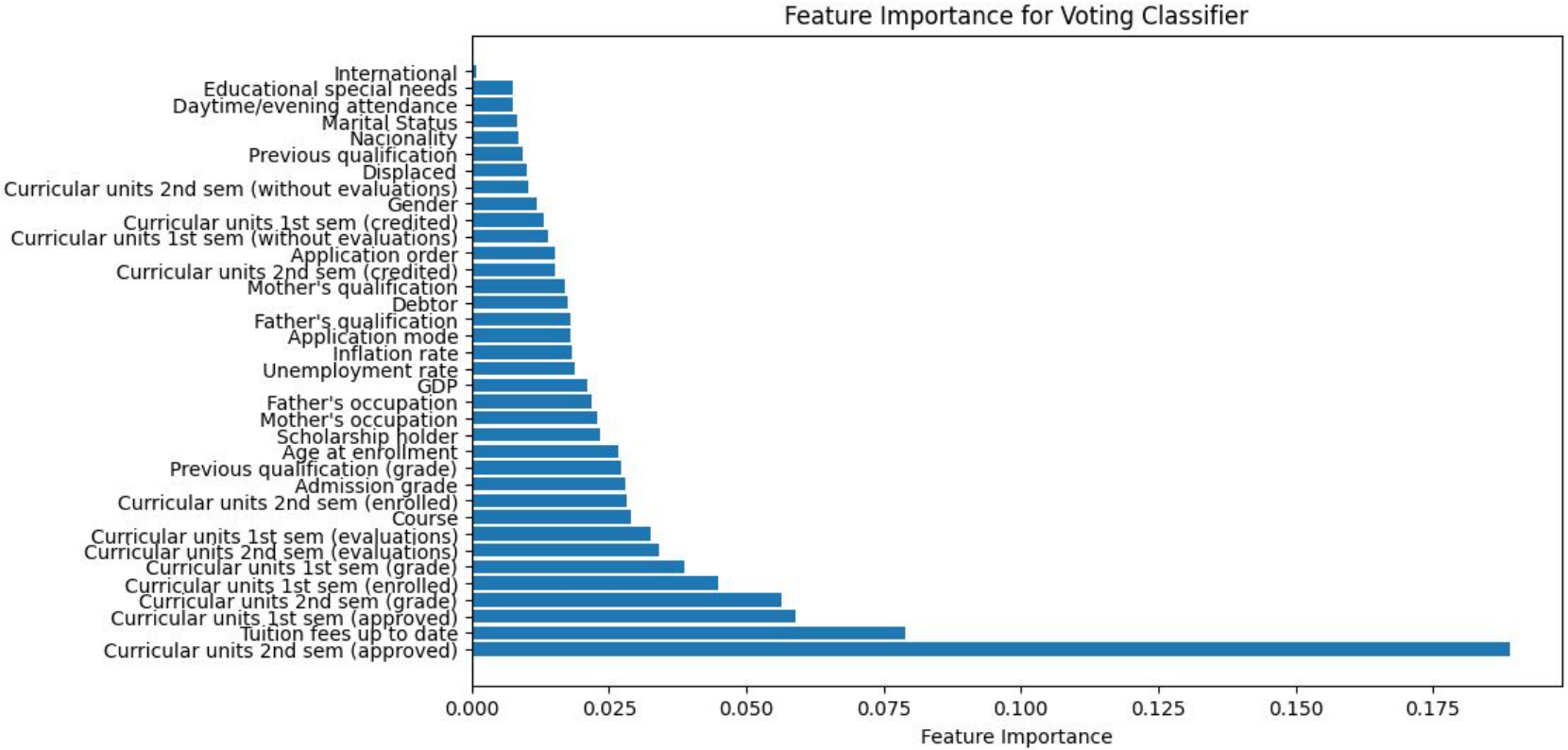
- Adaboost with Random Forest
- XGBoost

Overall Performance:

- Accuracy: 77.8%
- F1-Score: 71%

	Precision	Recall	F1-Score	Support
Dropout	0.85	0.76	0.80	316
Enrolled	0.56	0.40	0.47	151
Graduate	0.79	0.93	0.85	418
Accuracy			**0.78**	885
Macro avg	0.73	0.70	**0.71**	885
Weighted avg	0.77	0.78	**0.77**	885

Feature Importance



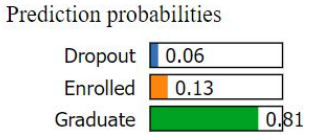
LIME (Local Interpretable Model-agnostic Explanations)

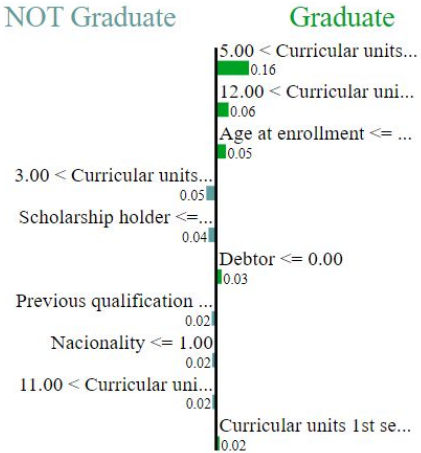
- Trade-off between accuracy and interpretability exists; complex algorithms offer higher accuracy but are black-box methods.
- Business settings often favor simpler, interpretable models despite lower accuracy.
- LIME can explain individual predictions of any classifier or regressor in a faithful way by computing a local surrogate model

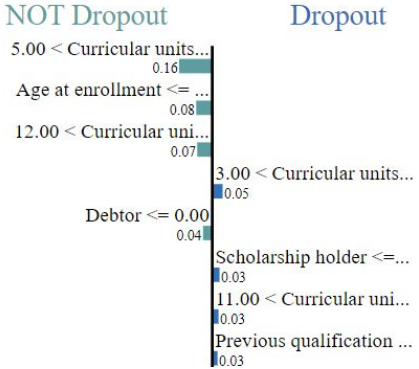
Example 1: Correct Prediction

True Label:
Graduate

Predicted:
Graduate





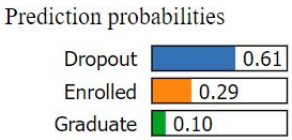


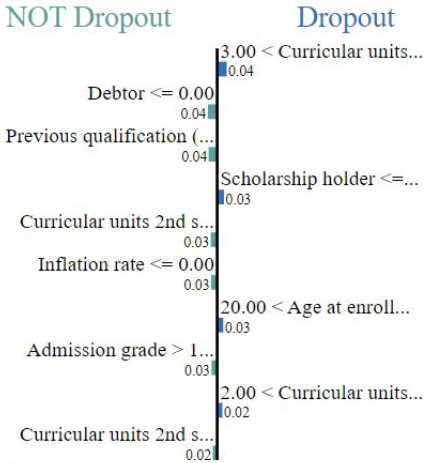
Feature	Value
Curricular units 2nd sem (approved)	6.00
Curricular units 2nd sem (grade)	12.14
Age at enrollment	18.00
Curricular units 1st sem (approved)	5.00
Scholarship holder	0.00
Debtor	0.00
Previous qualification (grade)	125.00
Nacionality	1.00
Curricular units 1st sem (grade)	11.57
Curricular units 1st sem (without evaluations)	0.00

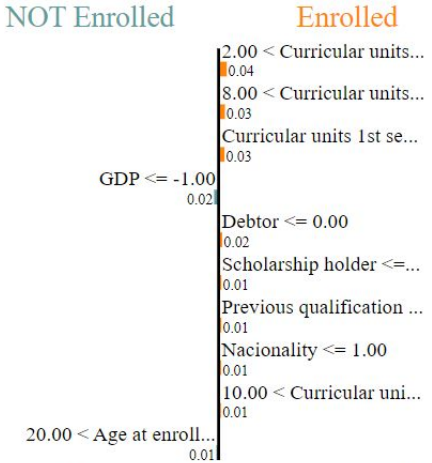
Example 2: Incorrect Prediction

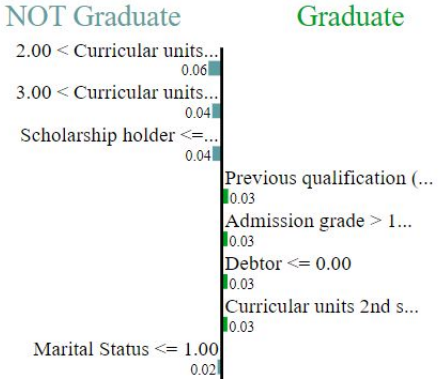
True Label:
Enrolled

Predicted:
Dropout









Feature	Value
Curricular units 1st sem (approved)	4.00
Debtor	0.00
Previous qualification (grade)	150.00
Scholarship holder	0.00
Curricular units 2nd sem (enrolled)	5.00
Inflation rate	-0.80
Age at enrollment	24.00
Admission grade	160.00
Curricular units 2nd sem (approved)	3.00
Curricular units 2nd sem (credited)	0.00

Conclusions

- Previous method
  - Accuracy: **73**, Avg F1: **65**
- Our method
  - Accuracy: **77.8**, Avg F1: **70.8**, Weighted F1: **77**
- Voting Classifier (Adaptive Boosting with Random Forests and eXtreme Gradient Boosting) combines the complementary benefits of bagging and boosting.
- Manual handcrafted feature engineering
- Interpretation efforts towards transparency and explainability
- Future Scope: Experiment with deep learning architectures

Appendix - LIME