Predicting Student Dropout and Academic Success

Submitted by:

Dilsha Singh D - 3122225001028
Kushal Varma G - 3122225001031
Hannah S - 3122225001032

Problem Statement

Educational institutions face challenges in identifying students at risk of dropping out or underperforming. Early intervention is crucial to improve student retention rates and academic success. This dataset contains student demographic details, academic records, and socio-economic factors. The objective is to develop a predictive model that classifies students into categories such as Dropout, Graduate and Enrolled based on these factors.

Domain and Challenges

Domain: Educational Data Science & Predictive Analytics

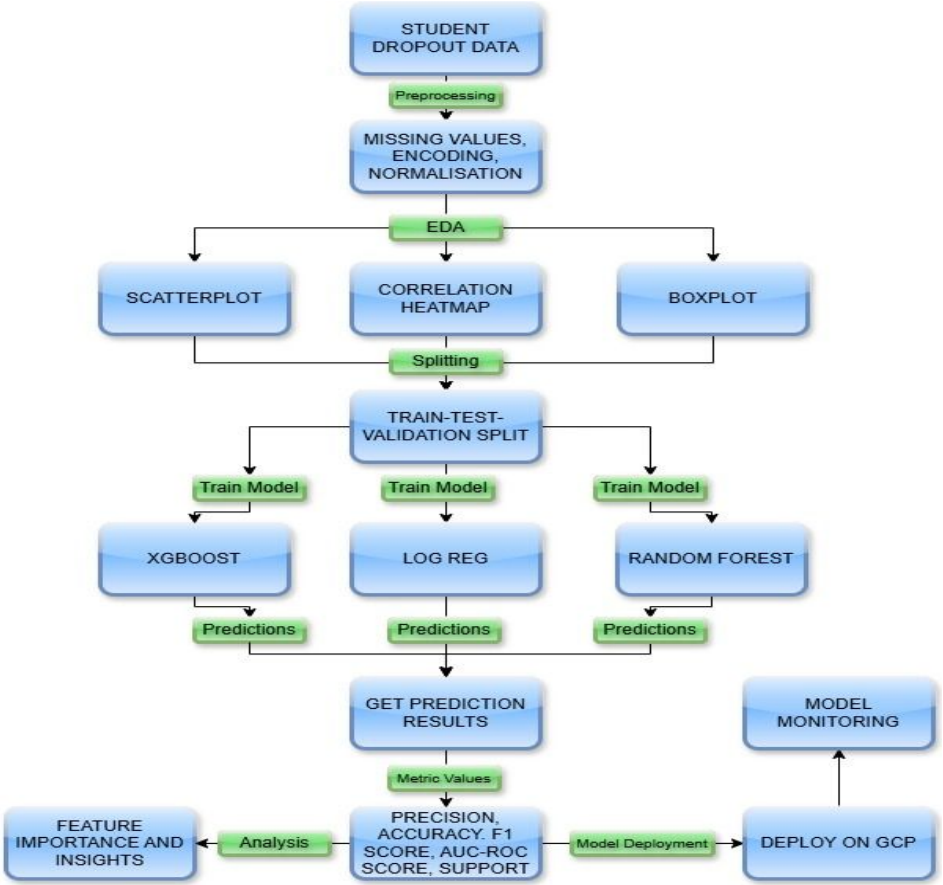
Challenges:

- High dropout rates affecting academic success
- Lack of real-time intervention mechanisms
- Data imbalance (fewer dropout cases compared to successful students)
- Feature selection for improving model accuracy
- Ethical considerations in using student data

Objective

- Develop an ML model to predict student dropout and academic success.
- Improve accuracy using dimensionality reduction and ensemble learning techniques.
- Provide insights for early interventions to reduce dropout rates.
- Optimize model performance through hyperparameter tuning.
- Predict the graduate and success rate
- Deploy the model in Cloud

Architecture Diagram



Dataset Description

DATASET NAME: Predict students' dropout and academic success dataset

Feature Summary:

The dataset consists of about 4424 rows and 37 columns and it covers a wide range of information including:

•**Demographics:** Gender, Age, Marital Status, Nationality, Displacement, Special Needs.
•**Academic Info:**
•Admission Grade, Previous Qualification & Grade
•Course Enrolled, Daytime/Evening Attendance
•1st & 2nd Semester: Enrolled, Approved, Grades, Evaluations
•**Socioeconomic:** Parents' Education & Occupation, Scholarship, Tuition Status
•**Contextual Factors:** Unemployment Rate, Inflation, GDP
•**Administrative:** Application Mode, Order, Debtor Status, International Student
Target Variable (Multi-class Classification)

- Dropout
- Graduate
- Enrolled

Steps in building ML Model

1. Data Collection

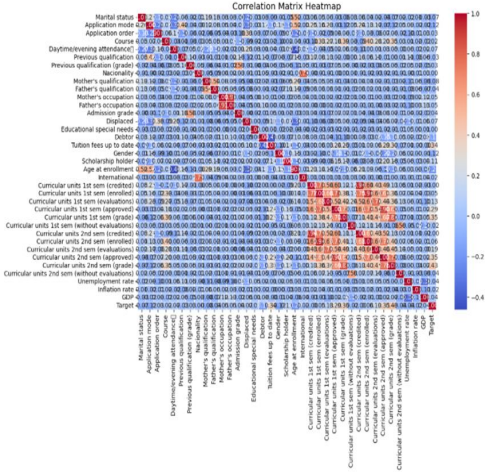
Data is retrieved from [Predict Students' Dropout and Academic Success - UCI Machine Learning Repository]

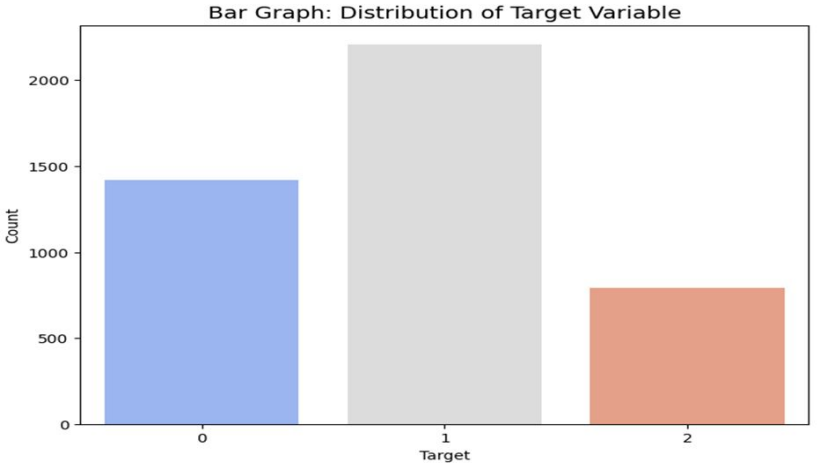
```
df=pd.read_csv("/content/drive/MyDrive/data.csv", sep=';')
df.head()
```

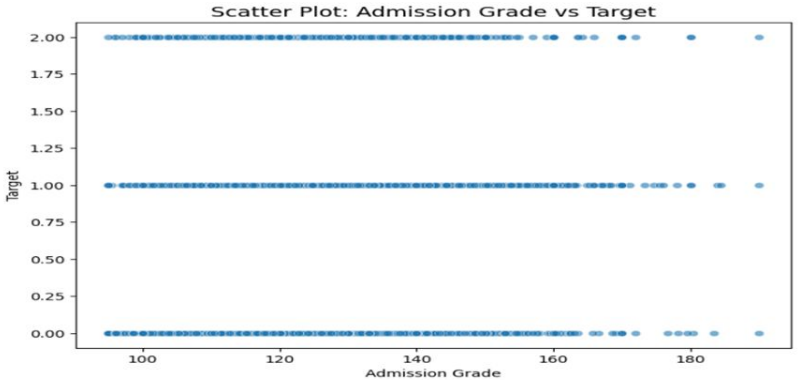
	Marital status	Application mode	Application order	Course	Daytime/evening attendance\t	Previous qualification	Previous qualification (grade)	Nacionality	Mother's qualification	Father's qualification	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)	Curricular units 2nd sem (without evaluations)	Unemployment rate	Inflation rate	GDP	Target
0	1	17	5	171	1	1	122.0	1	19	12	...	0	0	0	0	0.000000	0	10.8	1.4	1.74	Dropout
1	1	15	1	9254	1	1	160.0	1	1	3	...	0	6	6	6	13.666667	0	13.9	-0.3	0.79	Graduate
2	1	1	5	9070	1	1	122.0	1	37	37	...	0	6	0	0	0.000000	0	10.8	1.4	1.74	Dropout
3	1	17	2	9773	1	1	122.0	1	38	37	...	0	6	10	5	12.400000	0	9.4	-0.8	-3.12	Graduate
4	2	39	1	8014	0	1	100.0	1	37	38	...	0	6	6	6	13.000000	0	13.9	-0.3	0.79	Graduate

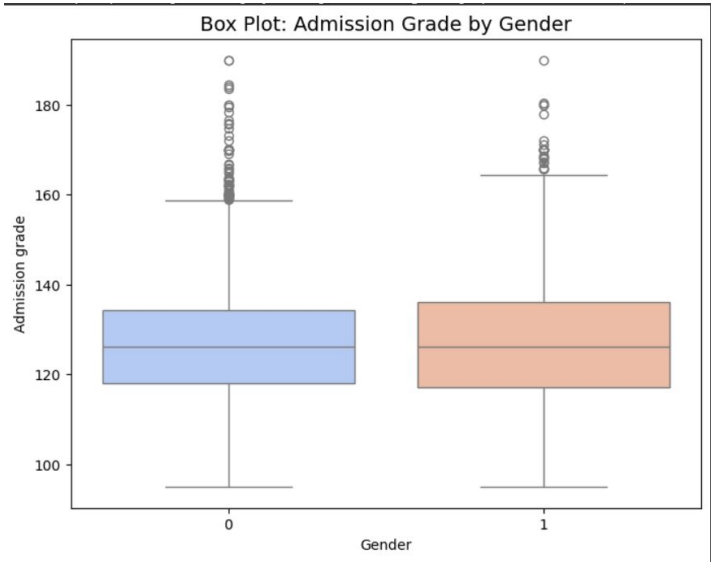
2. Exploratory Data Analysis:

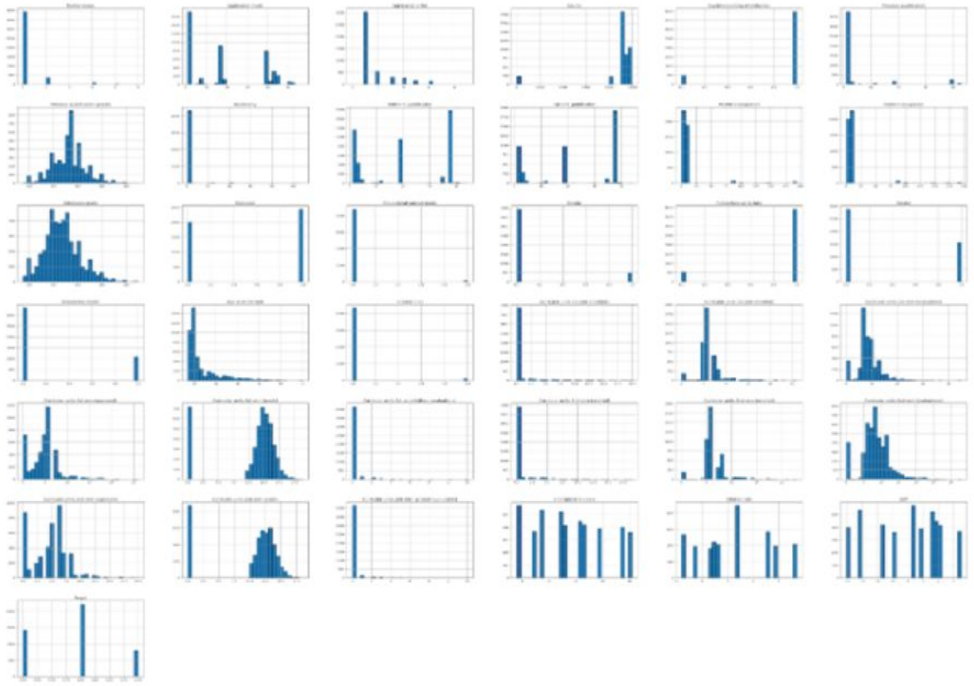
1. Basic Info & Descriptive Statistic
2. Histograms
3. Correlation Matrix Heatmap
4. Scatter Plot: Admission Grade vs Target
5. Box Plot: Admission Grade by Gender
6. Bar Graph: Target Variable Distribution








Box Plot: Admission Grade by Gender
Admission grade
180
160
140
120
100
0
1
Gender



3. Preprocessing

1. Handle missing values using ffill
2. Feature Scaling - StandardScaler
3. Handle class imbalance using SMOTE
4. Performed stratified train-test split

4. Model Building and training

We experimented with 3 models

- Random Forest
- XGBoost
- Logistic Regression

5. Model Evaluation

- Training accuracy
- Test accuracy
- ROC AUC
- Confusion Matrix
- Horizontal Bar Chart of Top 15 Absolute Coefficients

6. Optimization

- Dimensionality Reduction using PCA
- Hyperparameter optimization
- Randomized search

7. Cloud Deployment in GCP

Link to cloud project : ***Student Performance Prediction***

RESULTS PRIOR OPTIMIZATION

Dataset	Types of EDA task performed	Data Pre-processing task performed	Feature Selection Technique performed	Type of ML task to be performed (Supervised/ unsupervised)	Type of suitable ML algorithm to be performed.	List Performance Metric Values	Training Vs Test results
Predict students' dropout and academic success dataset	Histogram, Heatmap, Scatterplot, BoxPlot, bargraph	Handling Missing Values, Label Encoder, Normalization, Handling Class Imbalance	SelectK Best with ANOVA F-test (f_classif	Supervised Learning	Random Forest XGBoost Logistic Regression	Random Forest ACC: 0.7578 XGBoost ACC: 0.7859 Logistic Regression ACC: 0.660	Random Forest Train ACC: 0.8573 Test ACC: 0.7578 ROC AUC Score: 0.9076 XGBoost Train ACC: 0.0.9836 Test ACC: 0.7859 ROC AUC Score: 0.9171 Logistic Regression Train ACC: 0.669 Test ACC: 0.660 ROC AUC Score: 0.830

MODEL 1 – RANDOM FOREST

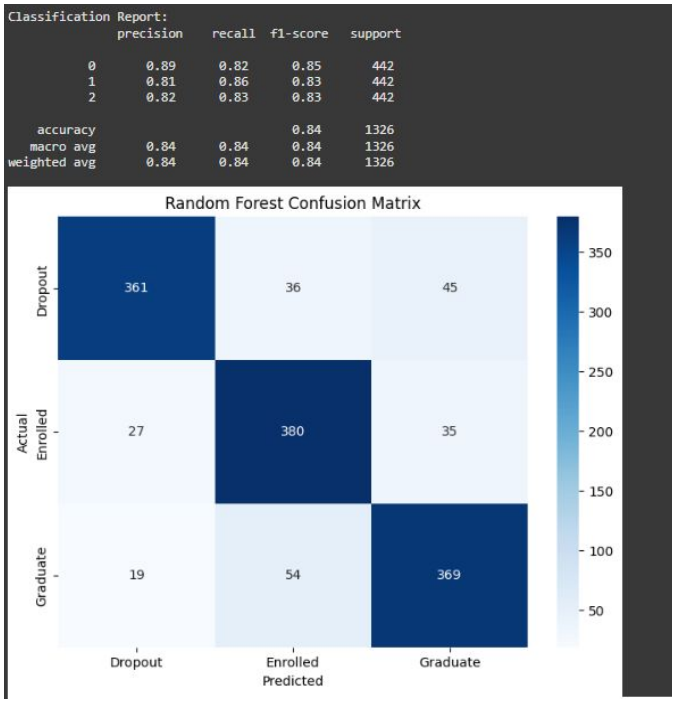
- Dimensionality Reduction – PCA with 95% variance.
- Hyperparameter Tuning using Randomized Search :
  - `'n_estimators': randint(100, 500),`
  - `'max_depth': [None, 10, 20, 30, 50],`
  - `'min_samples_split': randint(2, 20),`
  - `'min_samples_leaf': randint(1, 10),`
  - `'max_features': ['sqrt', 'log2', None],`
  - `'bootstrap': [True, False]`

```
Best Parameters: {'bootstrap': False, 'max_depth': 50,
'max_features': 'sqrt', 'min_samples_leaf': 1,
'min_samples_split': 11, 'n_estimators': 287}
```

- Performed 20 iterations with 3-fold CV
- Model Evaluation after optimization:

```
Train Accuracy: 1.0

Test Accuracy: 0.8371

ROC AUC: 0.9463
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.82      0.85       442
           1       0.81      0.86      0.83       442
           2       0.82      0.83      0.83       442

    accuracy                           0.84      1326
   macro avg       0.84      0.84      0.84      1326
weighted avg       0.84      0.84      0.84      1326
```



MODEL 2 – Logistic Regression

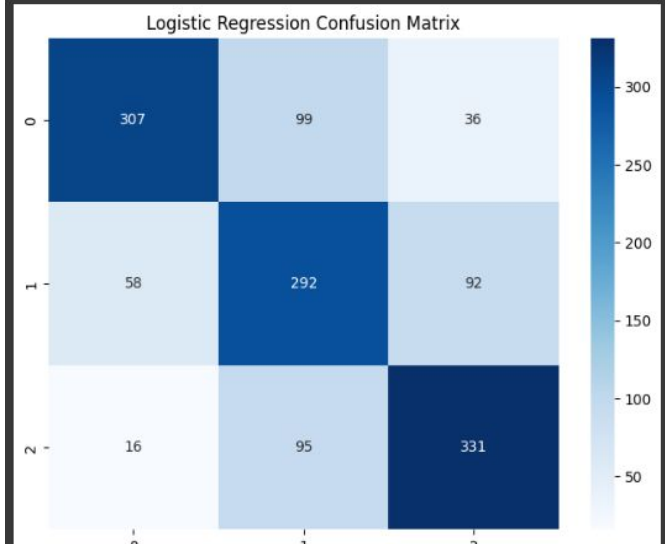
- Dimensionality Reduction – PCA with 20 components.
- Hyperparameter Tuning using Randomized Search :
  - `'penalty': ['l1', 'l2', 'elasticnet'],`
  - `'C': loguniform(1e-3, 100),`
  - `'solver': ['saga', 'lbfgs'],`
  - `'l1_ratio': [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]`

```
Best Hyperparameters: {'C': 1, 'penalty': 'l2', 'solver': 'saga'}
```

- Performed 20 iterations with 3-fold CV
- Model Evaluation after optimization:

```
Train Accuracy: 0.713

Test Accuracy: 0.701

ROC AUC: 0.857
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.69      0.75       442
           1       0.60      0.66      0.63       442
           2       0.72      0.75      0.73       442

    accuracy                           0.70      1326
   macro avg       0.71      0.70      0.70      1326
weighted avg       0.71      0.70      0.70      1326
```



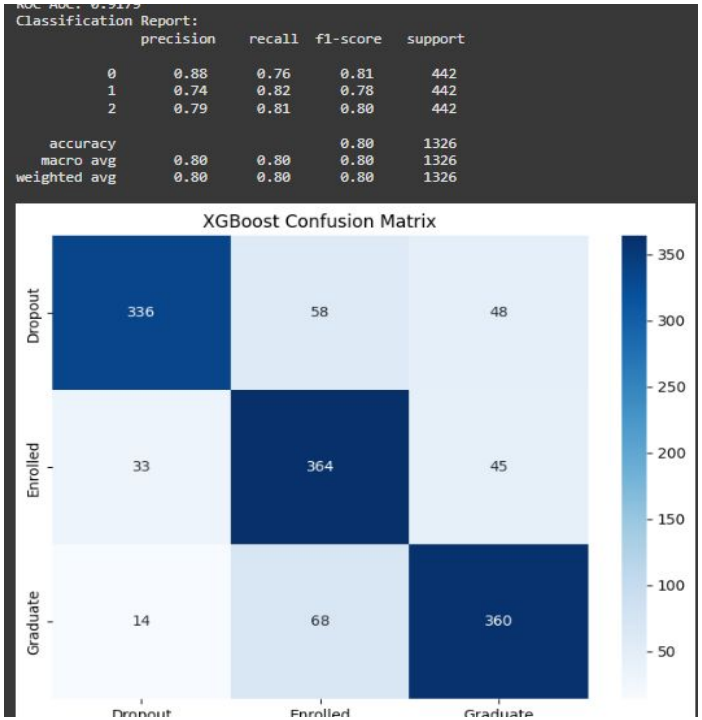
MODEL 3 – XGBoost

- Dimensionality Reduction – PCA to retain 90% variance in the dataset.
- Hyperparameter Tuning using Randomized Search :
  - `'n_estimators': randint(100, 300),  # Reduced range for faster training`
  - `'max_depth': randint(3, 10),`
  - `'learning_rate': loguniform(1e-3, 0.3),`
  - `'subsample': [0.8, 0.9, 1.0],`
  - `'colsample_bytree': [0.8, 0.9, 1.0],`
  - `'gamma': [0, 0.1, 0.2],`
  - `'reg_alpha': loguniform(1e-3, 10),`
  - `'reg_lambda': loguniform(1e-3, 10)`

```
Best Parameters: {'colsample_bytree': 1.0, 'gamma': 0.2,
'learning_rate': np.float64(0.0563600475052774), 'max_depth': 5,
'n_estimators': 262, 'reg_alpha': np.float64(1.2164139351417065),
'reg_lambda': np.float64(0.0019777828512462727), 'subsample': 1.0}
```

- Performed 15 iterations with 3-fold CV
- Model Evaluation after optimization :

```
Train Accuracy: 0.983

Test Accuracy: 0.805

ROC AUC: 0.930
```



RESULTS AFTER OPTIMIZATION

Model	Before Optimization			After Optimization		
	Train Accuracy	Test Accuracy	ROC AUC	Train Accuracy	Test Accuracy	ROC AUC
Random Forest (Bagging-parallel trees)	0.857	0.757	0.907	1.0	0.837	0.940
XGBoost (Sequential trees)	0.983	0.785	0.917	0.983	0.805	0.930
Logistic Regression	0.669	0.660	0.830	0.713	0.701	0.857

COMPARISON OF RESULTS AND INFERENCE

- **Random Forest** achieved the **highest test accuracy (83.7%)** and **ROC AUC (94.0%)**, making it the best-performing model.
- **XGBoost** delivered strong results with **80.5% accuracy** and **93.0% ROC AUC**, balancing accuracy and generalization.
- **Logistic Regression** showed minor improvement but remained the least effective, with **70.1% accuracy**.
- **Ensemble methods (Random Forest, XGBoost)** proved highly effective in predicting student dropout and academic success.
- **Dimensionality reduction and hyperparameter tuning** significantly boosted model performance and decision-making accuracy.

IMPACT OF PROJECT ON HUMAN, SOCIAL AND SUSTAINABLE DEVELOPMENT

- **Early dropout prediction** enables timely student support and intervention.
- **Reducing dropout rates** fosters social mobility and economic growth.
- **Ensuring fairness and privacy** promotes ethical AI use in education.
- **Supporting SDG 4 (Quality Education)** contributes to sustainable development.

LIMITATIONS

- **Data Dependency**: Model performance relies heavily on the quality and completeness of the dataset; missing or biased data can impact predictions.

- There is Class imbalance in our dataset.

- **Generalization Issues**: The model may perform well on the current dataset but might not generalize effectively across different institutions or regions.

- **Limited Real-Time Adaptability**: While predictive, the model does not continuously update in real-time unless integrated with live student performance

Conclusion and Future Work

Conclusion:

- Implemented and evaluated ML models for predicting student dropout and academic success.
- Ensemble models (Random Forest, XGBoost) outperformed Logistic Regression.
- Optimized Random Forest achieved **83.71% accuracy** (ROC AUC: **94.63%**).
- Dimensionality reduction and hyperparameter tuning enhanced model efficiency.

Future Work:

- **Continuous Model Improvement**: Regular updates with new student data.
- **Real-Time Monitoring**: Analyzing live student performance for proactive interventions.
- **Improved Explainability**: Developing interpretable models for better insights.
- **Integration with LMS**: Automating early intervention strategies for student success.

Learning Outcome

- Applied machine learning models to predict student dropout and academic success using real-world data.
- Explored the impact of dimensionality reduction and hyperparameter tuning on improving model performance.
- Evaluated multiple classification models using metrics like accuracy and ROC AUC.
- Gained experience in cloud deployment for hosting and accessing the predictive model remotely.
- Developed skills in interpreting results for data-driven decision-making in education.

REFERENCES

1. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=predict+student+dropout+and+academic+success+&btnG=#d=gs_qabs&t=1740282437590&u=%23p%3DKjCFZJzbpEsJ

2. https://ieeexplore.ieee.org/document/8523888

3. https://www.sciencedirect.com/science/article/pii/S0160791X24000228

4. https://datascience.codata.org/articles/10.5334/dsj-2019-014

5. https://colab.research.google.com/drive/1Vj9M9sk4Py3G7s_YJVhgAQNP9JshLu2x?usp=sharing