Application of Machine Learning Algorithms to Predict Academic Success.

MISSISSIPPI STATE UNIVERSITY™
JAMES WORTH BAGLEY
COLLEGE OF ENGINEERING

Lorena Benavides & John A Yepes – Dec 2024

Dataset Information

- **35 variables** – 34 predicted variables & one is the response/Target.
  - Kaggle from Portugal 
    - Each feature contains 4,424 data with no missing values.
      - **Type:**
      - Demographic data
      - Socioeconomic data.
      - Macroeconomic data.
      - Academic data at enrollment.
      - Academic data at the end of the first and second semesters.
      - The target

Dataset Info

Class of Variable		Variable	Type		Class of Variable		Variable	Type
Demographic data	1	Marital status	Numeric/discrete		Academic data at enrollment	18	Application mode	Numeric/discrete
	2	Nacionality	Numeric/discrete			19	Application order	Numeric/ordinal
	3	Displaced	Numeric/binary			20	Course	Numeric/discrete
	4	Gender	Numeric/binary			21	Daytime/evening attendance	Numeric/binary
	5	Age at enrollment	Numeric/discrete			22	Previous qualification	Numeric/discrete
	6	International	Numeric/binary					
					Academic data at the end of 1st semester	23	Curricular units 1st sem (credited)	Numeric/discrete
Socioeconomic data	7	Mother's qualification	Numeric/discrete			24	Curricular units 1st sem (enrolled)	Numeric/discrete
	8	Father's qualification	Numeric/discrete			25	Curricular units 1st sem (evaluations)	Numeric/discrete
	9	Mother's occupation	Numeric/discrete			26	Curricular units 1st sem (approved)	Numeric/discrete
	10	Father's occupation	Numeric/discrete			27	Curricular units 1st sem (grade)	Numeric/continuous
	11	Educational special needs	Numeric/binary			28	Curricular units 1st sem (without evaluations)	Numeric/discrete
	12	Debtor	Numeric/binary					
	13	Tuition fees up to date	Numeric/binary		Academic data at the end of 2nd semester	29	Curricular units 2nd sem (credited)	Numeric/discrete
	14	Scholarship holder	Numeric/binary			30	Curricular units 2nd sem (enrolled)	Numeric/discrete
						31	Curricular units 2nd sem (evaluations)	Numeric/discrete
Macroeconomic data	15	Unemployment rate	Numeric/continuous			32	Curricular units 2nd sem (approved)	Numeric/discrete
	16	Inflation rate	Numeric/continuous			33	Curricular units 2nd sem (grade)	Numeric/continuous
	17	GDP	Numeric/continuous			34	Curricular units 2nd sem (without evaluations)	Numeric/discrete
35 - Target/Response			Type: Categorical				Dropout / Graduate / Enrolled	

Problem – Classification case

Problem: Understanding student success.

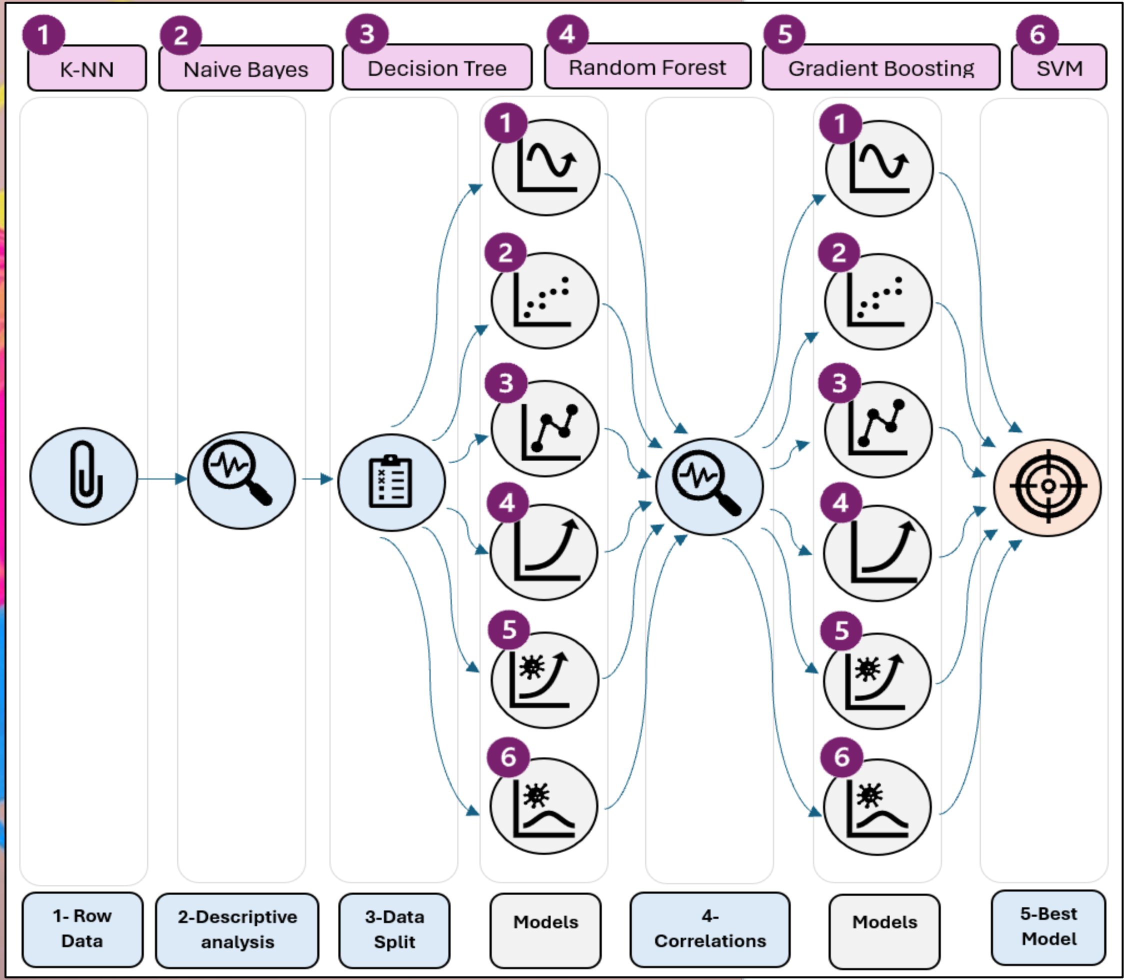
Goal: Detect students who are at risk of dropping out of their education.

- Tackling student dropout rates.
- Retaining students.

- **Target:**
  - Dropout, Graduate, and Enrolled

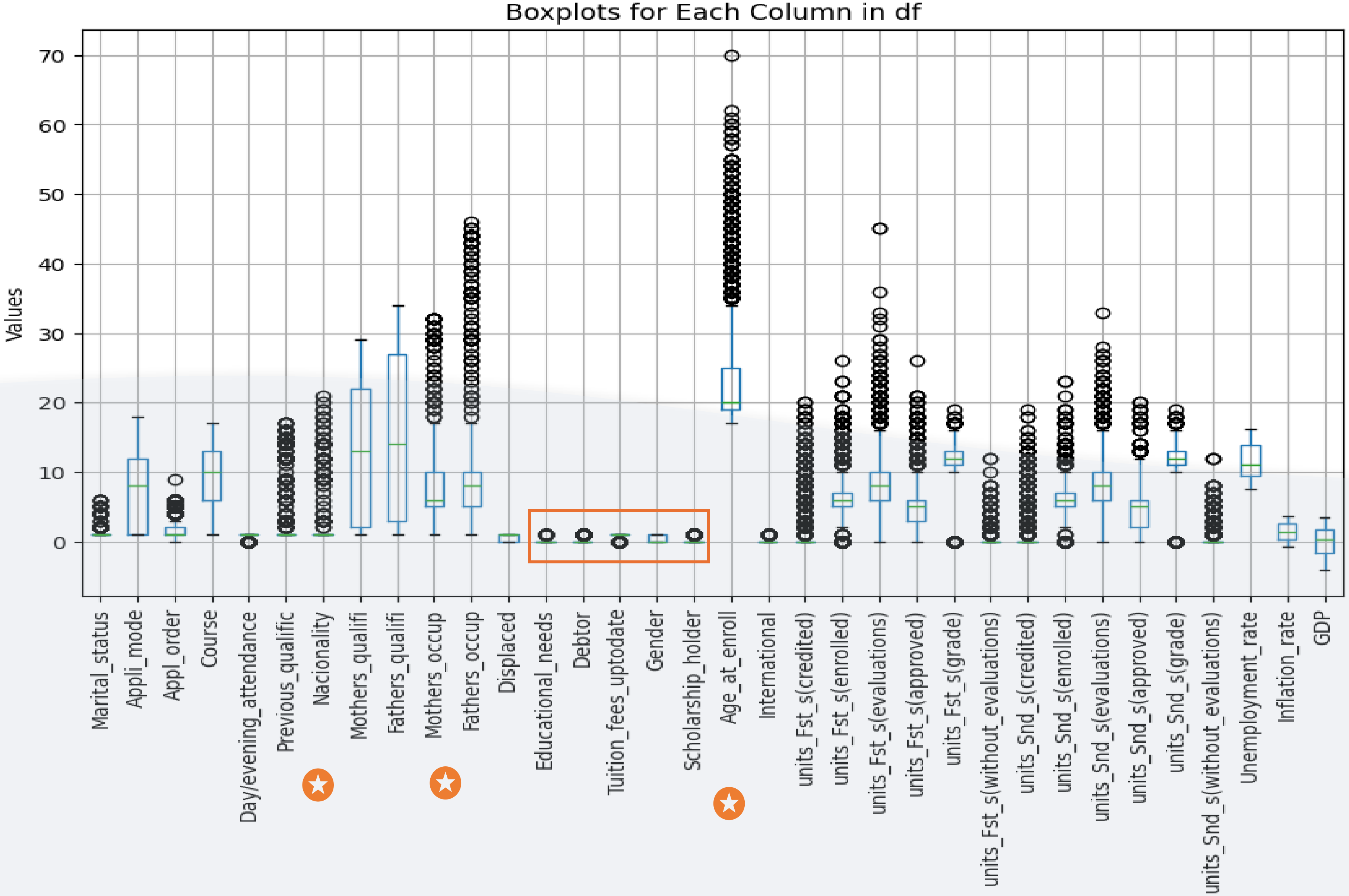
Methods & Deliverables

- **Visualizations (descriptive analysis)**
- **ML algorithms**
  - KNN
  - Naïve Bayes (Gaussian & Bernoulli)
  - Decision Tree
  - Random Forest
  - Gradient Boosting
  - Support Vector Machine
- **Results:**
  - Accuracy (Training & test) - Graph
  - Confusion Matrix
- **Best Model**
  - Metrix

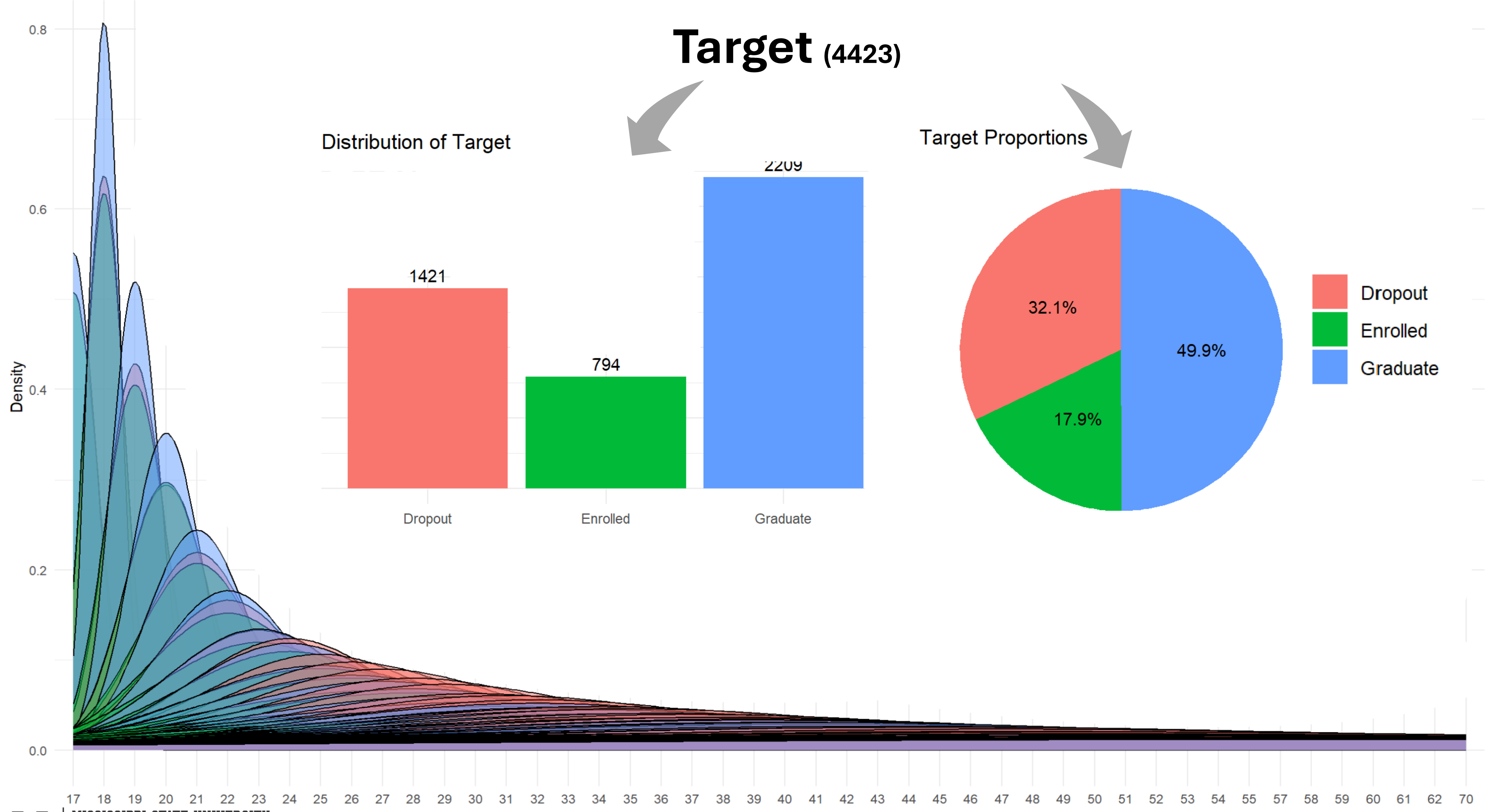


Visualizations

Boxplots by Target

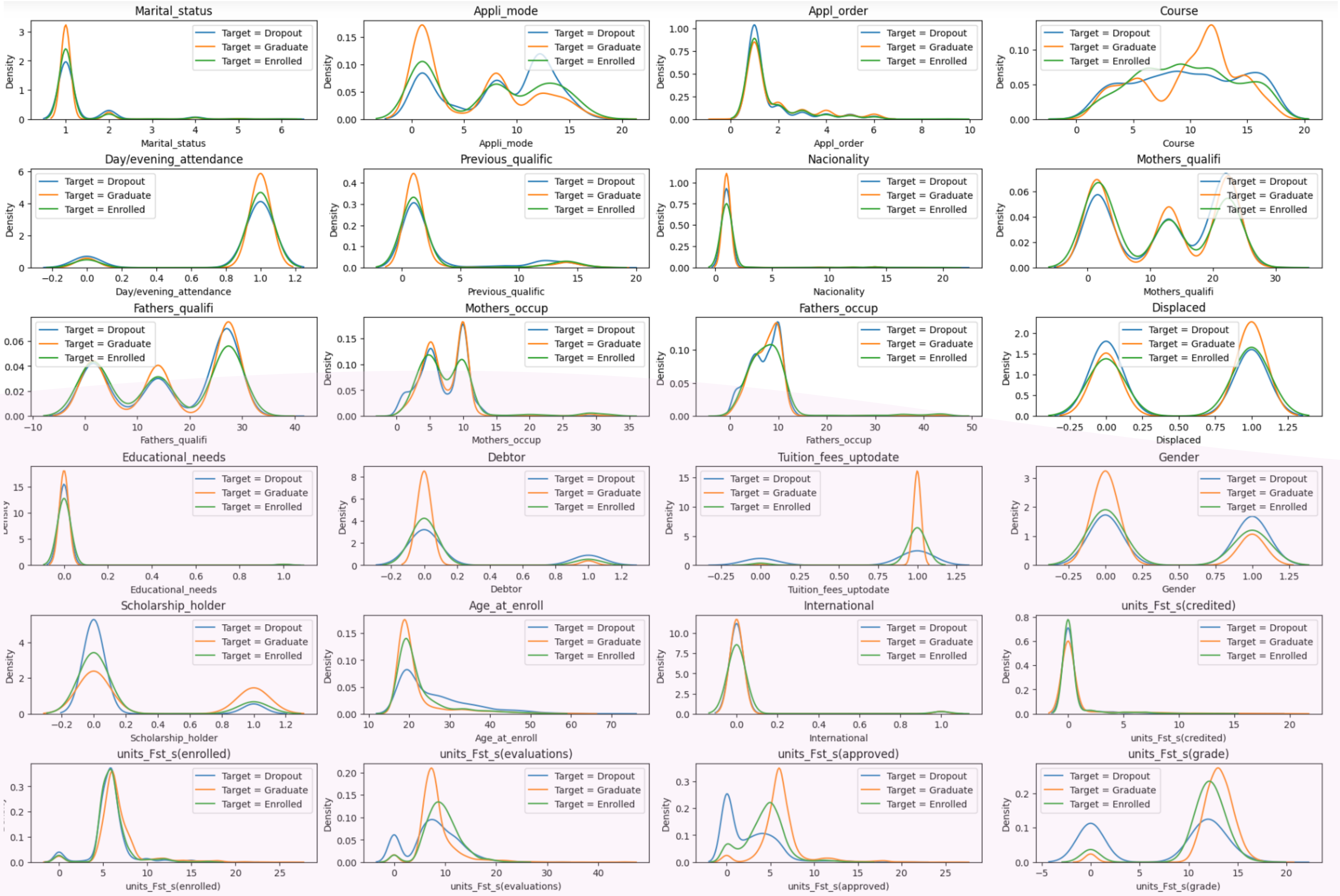


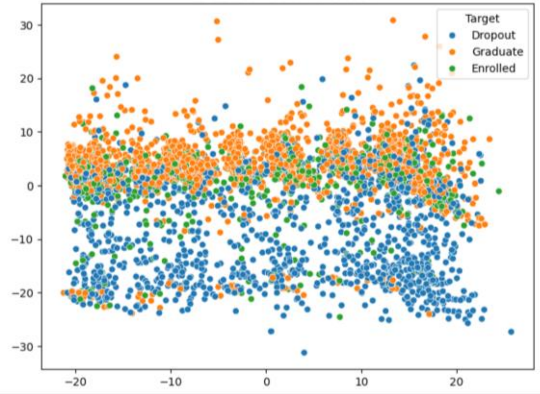
Target (4423)



Density
0.8
0.6
0.4
0.2
0.0
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 57 58 59 60 61 62 70
Distribution of Target
1421
794
2209
Dropout
Enrolled
Graduate
Target Proportions
32.1%
17.9%
49.9%
Dropout
Enrolled
Graduate

Density Plots by Target





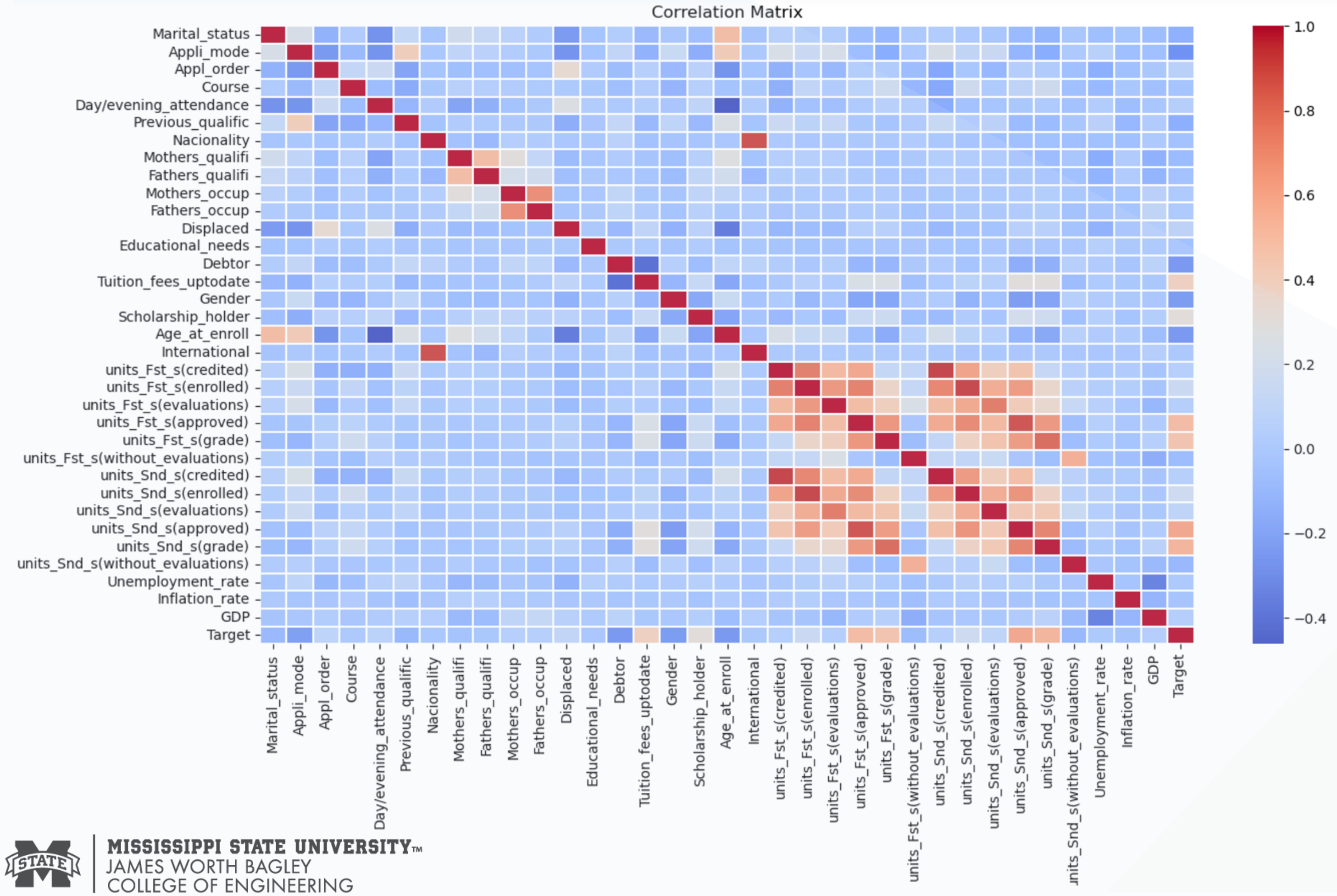
The variable shows each target class overlap, so the classes (Dropout, Enrolled, and Graduate) are not linearly separable.

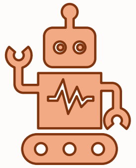
ML Process

1. Dataset
2. Filter by correlation extraction (>0.8)
3. Split the data between Training and Testing
4. Transforming dataset based on data type
5. New Models

Correlation > 0.8

	Feature1	Feature2	Correlation
0	Nacionality	International	0.911724





Full data (before extraction)	
Training set	3318,33
Testing set	1106,33

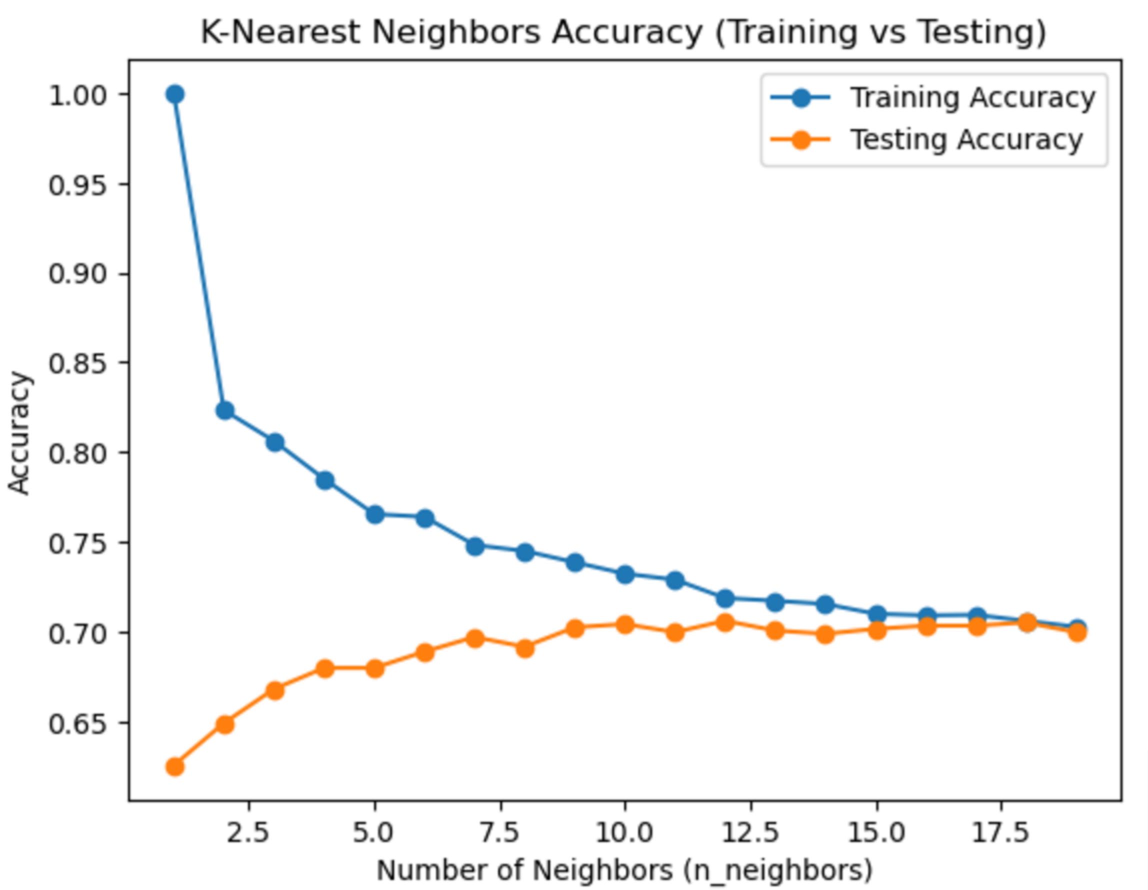
ColumnTransformer

ordinal	scaling	onehot
OrdinalEncoder	StandardScaler	OneHotEncoder

Data (After Transformation)	
Training set	3318,436
Testing set	1106,436

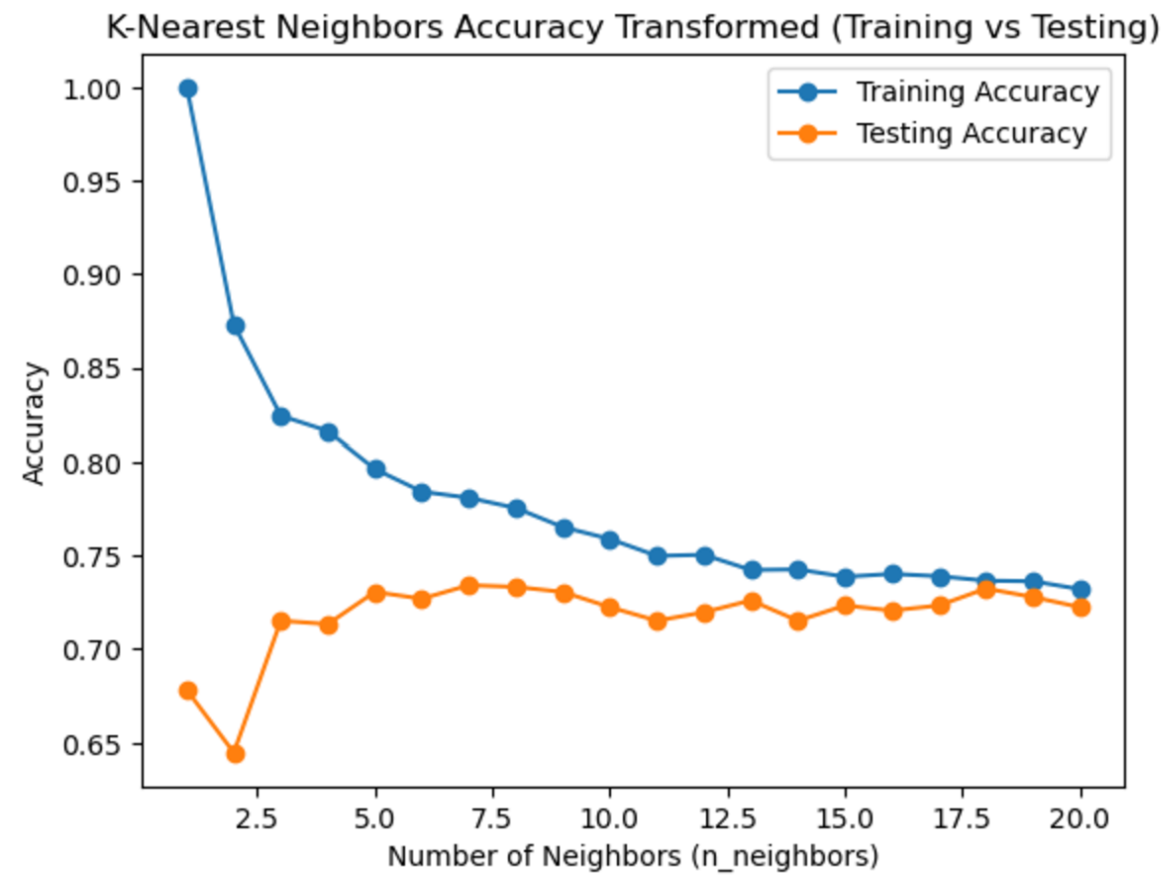
KNN

Raw Data



Best test accuracy occurs at n_neighbors=18
Training set is 0.705
Testing set is 0.705

Transformed Data



Best test accuracy occurs at n_neighbors=12
Training set is 0.750
Testing set is 0.728

Best cross-validation score: 0.699

Naive Bayes (Gaussian and Bernoulli)

Raw Data

```
# Naive Bayes Classifier
from sklearn.naive_bayes import GaussianNB
NB = GaussianNB()
brn = BernoulliNB()

NB.fit(X_train, y_train)

print("GBN Train set score: {:.2f}".format(NB.score(X_train, y_train)))
print("GBN Test set score: {:.2f}".format(NB.score(X_test, y_test)))

brn.fit(X_train, y_train)
print("BRN Train set score: {:.2f}".format(brn.score(X_train, y_train)))
print("BRN Test set score: {:.2f}".format(brn.score(X_test, y_test)))
```

```
GBN Train set score: 0.68
GBN Test set score: 0.67
BRN Train set score: 0.68
BRN Test set score: 0.69
```

Transformed Data

```
# Naive Bayes Classifier

NB = GaussianNB()
brn = BernoulliNB()

NB.fit(X_train_scaled, y_train2).predict(X_test_scaled)

print("GBN Train set score: {:.2f}".format(NB.score(X_train_scaled, y_train2)))
print("GBN Test set score: {:.2f}".format(NB.score(X_test_scaled, y_test2)))

brn.fit(X_train_scaled, y_train2).predict(X_test_scaled)
print("BRN Train set score: {:.2f}".format(brn.score(X_train_scaled, y_train2)))
print("BRN Test set score: {:.2f}".format(brn.score(X_test_scaled, y_test2)))
```

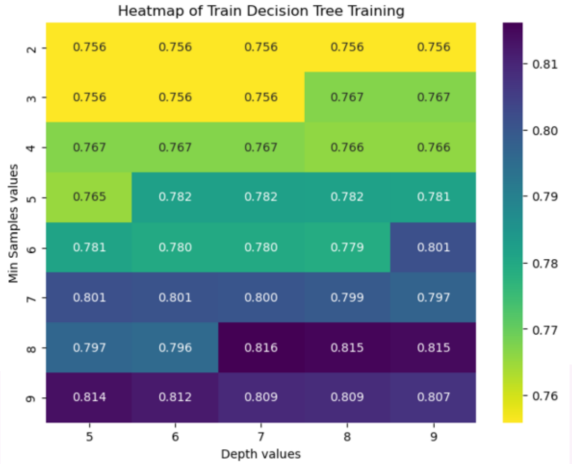
```
GBN Train set score: 0.27
GBN Test set score: 0.23
BRN Train set score: 0.72
BRN Test set score: 0.71
```

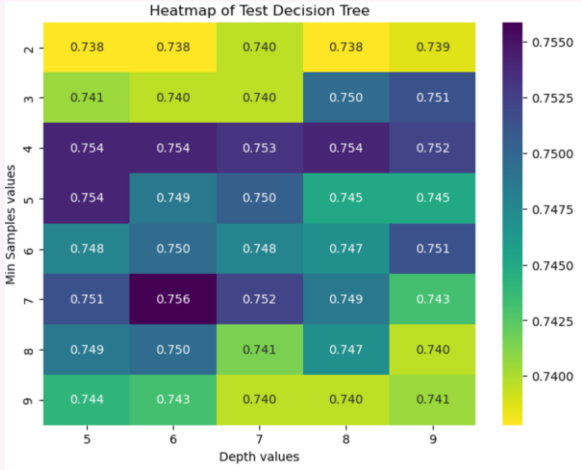
- Best model is Bernoulli after transforming and deleting variables.

- The small gap between train and test scores indicates no overfitting or underfitting

Decision Tree

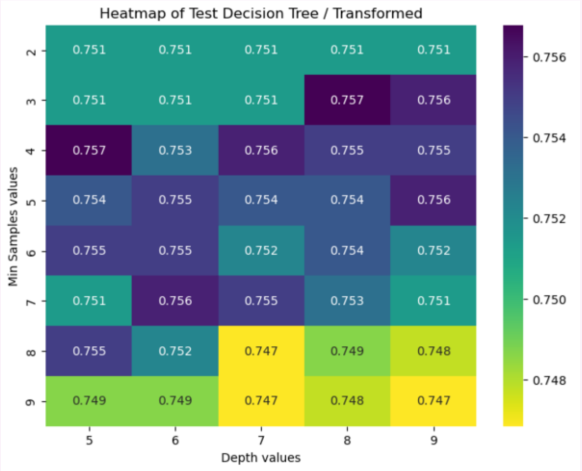
Raw Data





Best test accuracy occurs at max_depth=5, min_samples_split=7
Training set is 0.762
Testing set is 0.741

Transformed Data



Best test accuracy occurs at max_depth=8, min_samples_split=2
Training set 0.80
Testing set 0.7559

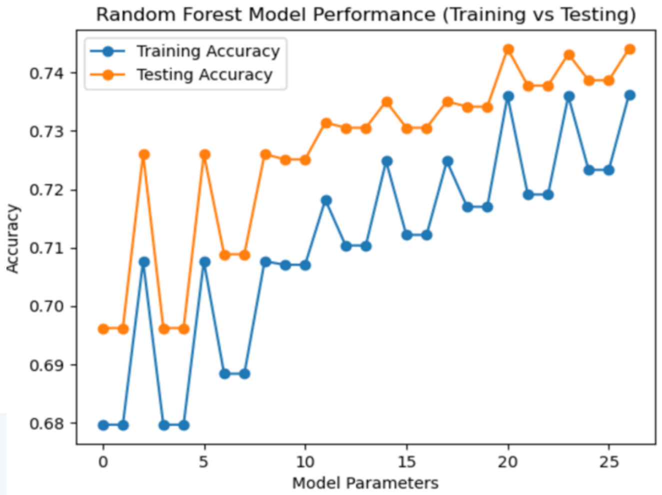
Transformed Data

```
Test set accuracy: 0.735
Test set precision: 0.715
Test set recall: 0.735
Test set F1 score: 0.717
Confusion matrix:
 [[253  33  67]
 [ 52  56  85]
 [ 34  22 504]]
ROC AUC score: 0.822
Classification report:
              precision    recall  f1-score   support

           0       0.82      0.68      0.74       353
           1       0.50      0.33      0.40       193
           2       0.77      0.94      0.85       560

    accuracy                           0.75      1106
   macro avg       0.70      0.65      0.66      1106
weighted avg       0.74      0.75      0.74      1106
```

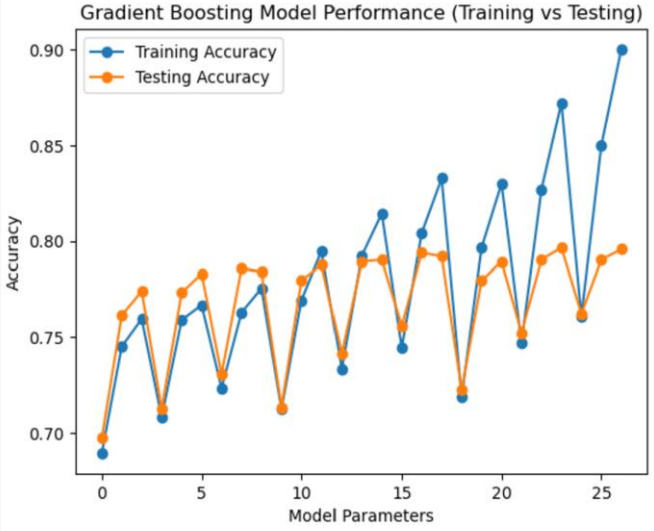
0 = Dropout
1= Enrolled
2=Graduate

Radom Forest Model

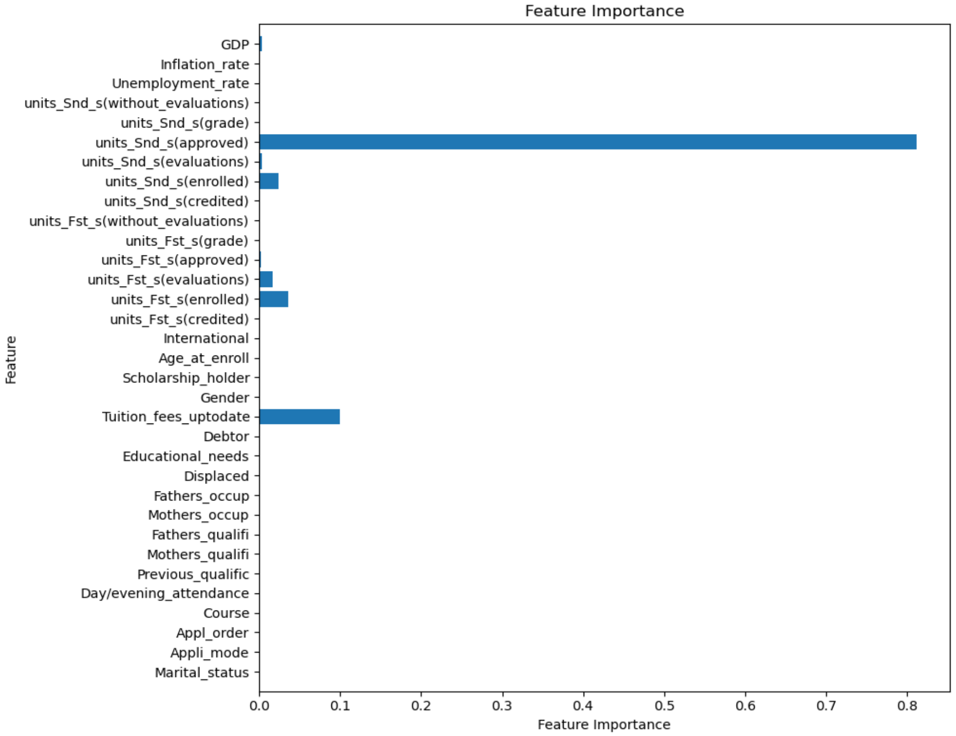
Raw Data



Testing set always is higher than training test.
No model with full data.



Transformed Data



Best parameters found: {'max_depth': 3, 'max_features': None, 'n_estimators': 50}
Best cross-validation score: 0.728
Train set is 0.74
Test set is 0.74

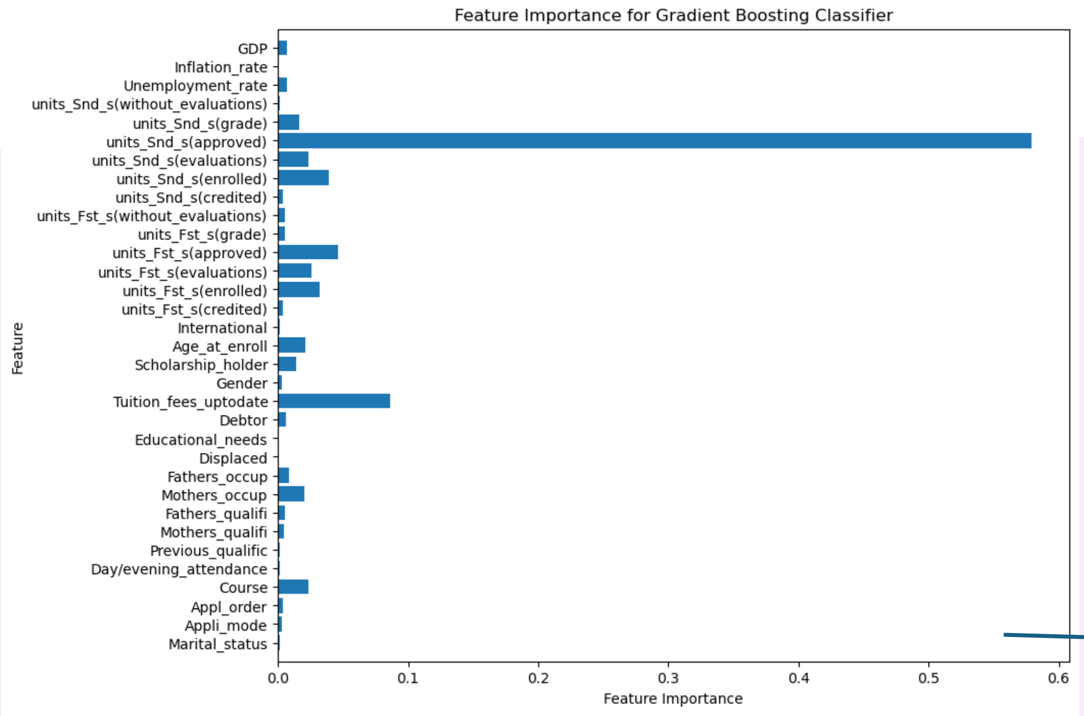
Gradient Boosting

Raw Data

Best parameters found: learning_reta=0.1, max_depth=2,n_estimatros = 100.

Train set is 0.792
Test set is 0.789

Transformed Data



0 = Dropout
1= Enrolled
2=Graduate

```
Train set score: 0.89
Test set score: 0.81
Test set accuracy: 0.807
Test set precision: 0.800
Test set recall: 0.807
Test set F1 score: 0.801
Confusion matrix:
 [[276  38  39]
 [ 37  97  59]
 [ 20  20 520]]
ROC AUC score: 0.897
Classification report:
              precision    recall  f1-score   support

           0       0.83      0.78      0.80       353
           1       0.63      0.50      0.56       193
           2       0.84      0.93      0.88       560

    accuracy                           0.81      1106
   macro avg       0.77      0.74      0.75      1106
weighted avg       0.80      0.81      0.80      1106
```

Best parameters found: learning_reta=0.2, max_depth=1,n_estimatros = 150.
Best cross-validation score: 0.770
Train set is 0.833
Test set is 0.794

SVM

Raw Data

```
# SVM not tuning
from sklearn.svm import SVC
svc_clf = SVC(kernel='rbf', C=100, gamma=0.1)
svc_clf.fit(X_train, y_train)
print("Train set score: {:.2f}".format(svc_clf.score(X_train, y_train)))
print("Test set score: {:.2f}".format(svc_clf.score(X_test, y_test)))
```

```
Train set score: 1.00
Test set score: 0.51
```

```
# SVM
from sklearn.svm import SVC
svc_clf = SVC(kernel='rbf', C=100, gamma='scale')
svc_clf.fit(X_train, y_train)
print("Train set score: {:.2f}".format(svc_clf.score(X_train, y_train)))
print("Test set score: {:.2f}".format(svc_clf.score(X_test, y_test)))
```

```
Train set score: 0.87
Test set score: 0.77
```

kernel=rbf C=100 gamma=scale
Train Accuracy = 0.87
Test Accuracy = 0.778

Transformed Data

```
svc_clf = SVC(kernel='rbf', C=10, gamma=0.01)
svc_clf.fit(X_train_scaled, y_train2)
print("Train set score: {:.2f}".format(svc_clf.score(X_train_scaled, y_train2)))
print("Test set score: {:.2f}".format(svc_clf.score(X_test_scaled, y_test2)))
```

```
Train set score: 0.89
Test set score: 0.81
```

Best parameters found: kernel=rbf, C=10 gamma=0.01, C=10
Train Accuracy = 0.89
Test Accuracy = 0.81

0 = Dropout
1= Enrolled
2=Graduate

```
Test set accuracy: 0.789
Test set precision: 0.779
Test set recall: 0.789
Test set F1 score: 0.781
Confusion matrix:
 [[274  38  41]
 [ 37  83  73]
 [ 17  27 516]]
ROC AUC score: 0.898
Classification report:
              precision    recall  f1-score   support

           0       0.84      0.78      0.80       353
           1       0.56      0.43      0.49       193
           2       0.82      0.92      0.87       560

    accuracy                           0.79      1106
   macro avg       0.74      0.71      0.72      1106
weighted avg       0.78      0.79      0.78      1106
```

Conclusions

- The models encounter an underfitting problem. Therefore, to obtain optimum performance 33 features were selected.
- Even thought for some models it is not required to scale the data, we found some improvement in general for all models.
- Gradient Boosting and SVM models algorithms perform with more accurate results in predicting dropouts of students with machine learning algorithms, while Naive Bayes has the lowest classification accuracy.
- The proposed model predicted the risk of dropout of students with 79% accuracy in average. Therefore, dropout risk can be predicted with this model in the future.

References

- [1] “P_R_Dashboard_2022Cohort,” Tableau Software. Accessed: Oct. 25, 2024. [Online]. Available: https://public.tableau.com/views/P_R_Dashboard_2022Cohort/PR_Dashboard?:embed=y&:showVizHome=no&:host_url=https%3A%2F%2Fpublic.tableau.com%2F&:embed_code_version=3&:tabs=no&:toolbar=yes&:animate_transition=yes&:display_static_image=no&:display_spinner=no&:display_overlay=yes&:display_count=yes&:language=en-US&:loadOrderID=0
- [2] A. W. Astin, *Assessment for excellence: The philosophy and practice of assessment and evaluation in higher education*. Rowman & Littlefield Publishers, 2012.
- [3] C. Gibson, S. Rankin, and T. York, “Defining and measuring academic success,” *Pract. Assess. Res. Eval.*, vol. 20, no. 5, p. 20, 2015.
- [4] R. Los and A. Schweinle, “The interaction between student motivation and the instructional environment on academic outcome: a hierarchical linear model,” *Soc. Psychol. Educ.*, vol. 22, no. 2, pp. 471–500, Apr. 2019, doi: 10.1007/s11218-019-09487-5.
- [5] A. I. Adekitan and E. Noma-Osaghae, “Data mining approach to predicting the performance of first year student in a university using the admission requirements,” *Educ. Inf. Technol.*, vol. 24, pp. 1527–1543, 2019.
- [6] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, “Systematic review of research on artificial intelligence applications in higher education–where are the educators?,” *Int. J. Educ. Technol. High. Educ.*, vol. 16, no. 1, pp. 1–27, 2019.

References

- [7] S. Hussain and M. Q. Khan, “Student-Performulator: Predicting Students’ Academic Performance at Secondary and Intermediate Level Using Machine Learning,” *Ann. Data Sci.*, vol. 10, no. 3, pp. 637–655, Jun. 2023, doi: 10.1007/s40745-021-00341-0.
- [8] M. Yağcı, “Educational data mining: prediction of students’ academic performance using machine learning algorithms,” *Smart Learn. Environ.*, vol. 9, no. 1, p. 11, Mar. 2022, doi: 10.1186/s40561-022-00192-z.
- [9] L. R. Pelima, Y. Sukmana, and Y. Rosmansyah, “Predicting University Student Graduation Using Academic Performance and Machine Learning: A Systematic Literature Review,” *IEEE Access*, vol. 12, pp. 23451–23465, 2024, doi: 10.1109/ACCESS.2024.3361479.
- [10] A. Singh, M. N., and R. Lakshmiganthan, “Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 12, 2017, doi: 10.14569/IJACSA.2017.081201.
- [11] N. Salmi and Z. Rustam, “Naïve Bayes classifier models for predicting the colon cancer,” in *IOP conference series: materials science and engineering*, IOP Publishing, 2019, p. 052068.
- [12] Y. Lu, T. Ye, and J. Zheng, “Decision Tree Algorithm in Machine Learning,” in *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, Aug. 2022, pp. 1014–1017. doi: 10.1109/AEECA55500.2022.9918857.

References

- [13] P. Probst and A.-L. Boulesteix, “To Tune or Not to Tune the Number of Trees in Random Forest”.
- [14] Z. Tian, J. Xiao, H. Feng, and Y. Wei, “Credit Risk Assessment based on Gradient Boosting Decision Tree,” *Procedia Comput. Sci.*, vol. 174, pp. 150–160, Jan. 2020, doi: 10.1016/j.procs.2020.06.070.
- [15] D. Isa, L. H. Lee, V. P. Kallimani, and R. RajKumar, “Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine,” *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1264–1272, Sep. 2008, doi: 10.1109/TKDE.2008.76.
- [16] “Find Open Datasets and Machine Learning Projects | Kaggle.” Accessed: Oct. 25, 2024. [Online]. Available: https://www.kaggle.com/datasets


THANK YOU