EDUANALYTICS

ENHANCING EDUCATION WITH
DATA MINING

1. Proposed model description

The Decision Tree model is a versatile and interpretable algorithm chosen for its ability to capture non-linear relationships within the student performance dataset. Decision Trees are particularly well-suited for educational data analysis, as they allow us to understand the decision-making process behind student outcomes. The model will leverage features such as 'math score,' 'reading score,' 'writing score and other relevant factors to predict student performance categories.

The Support Vector Classifier (SVC) is a powerful classification algorithm selected for its effectiveness in handling complex relationships within the data. SVC works well for both linear and non-linear patterns, making it suitable for capturing nuanced interactions between various features impacting student performance. The model will be trained to distinguish between different classes of student performance based on a range of input features.

The Naive Bayes model is chosen for its simplicity and efficiency, particularly when dealing with categorical features and relatively small datasets. Despite its 'naive' assumption of feature independence, Naive Bayes often performs well in practice and can provide insights into the conditional probabilities of different factors influencing student outcomes. The model will be applied to predict student performance categories based on the likelihood of various input features.

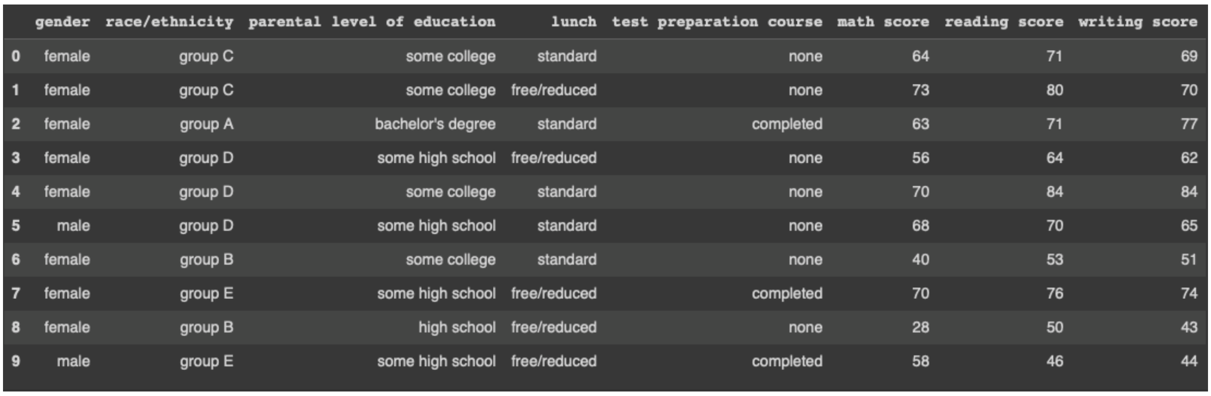
All proposed models will undergo a rigorous training process on the preprocessed dataset, which includes handling outliers, addressing class imbalance, and encoding categorical variables. The models will be evaluated using appropriate performance metrics such as accuracy, precision, recall, and F1-score.

2. Empirical Studies

2.1 Dataset description

The dataset, obtained from Royce Kimmons and consisting of 1000 instances , includes scores from three exams and various personal, social, and economic factors. The focus is on understanding how these factors interact and influence exam scores. This comprehensive dataset allows for a closer look at the relationships between academic performance, personal attributes, and socio-economic conditions.

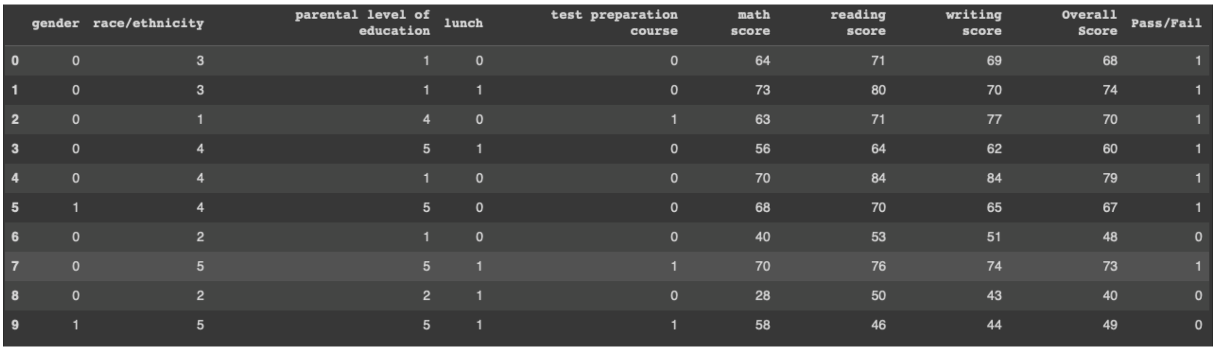
The following Figure presents the initial 10 rows of the dataset:



	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group C	some college	standard	none	64	71	69
1	female	group C	some college	free/reduced	none	73	80	70
2	female	group A	bachelor's degree	standard	completed	63	71	77
3	female	group D	some high school	free/reduced	none	56	64	62
4	female	group D	some college	standard	none	70	84	84
5	male	group D	some high school	standard	none	68	70	65
6	female	group B	some college	standard	none	40	53	51
7	female	group E	some high school	free/reduced	completed	70	76	74
8	female	group B	high school	free/reduced	none	28	50	43
9	male	group E	some high school	free/reduced	completed	58	46	44

Figure 1 Dataset

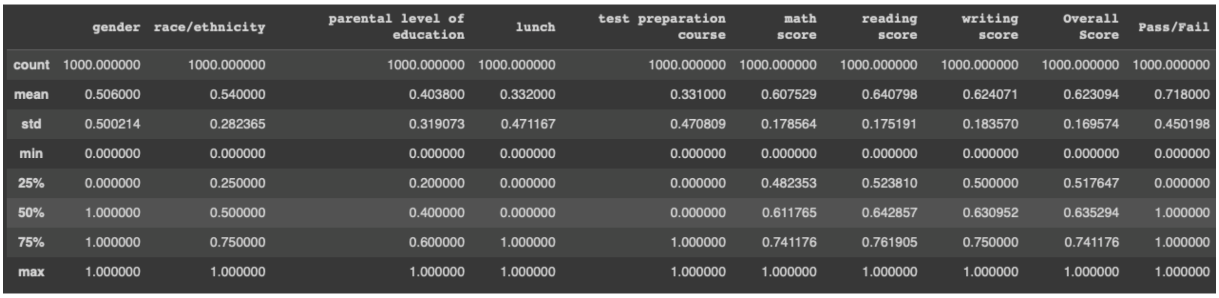
The following Figure presents the first 10 rows of the dataset after adding 'overall score' (calculated as the mean of math score, reading score, and writing score), 'pass/fail' (set to 1 if the overall score is greater than or equal to 60 and 0 otherwise), and replacing categorical values with numerical values:



	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Overall Score	Pass/Fail
0	0	3	1	0	0	64	71	69	68	1
1	0	3	1	1	0	73	80	70	74	1
2	0	1	4	0	1	63	71	77	70	1
3	0	4	5	1	0	56	64	62	60	1
4	0	4	1	0	0	70	84	84	79	1
5	1	4	5	0	0	68	70	65	67	1
6	0	2	1	0	0	40	53	51	48	0
7	0	5	5	1	1	70	76	74	73	1
8	0	2	2	1	0	28	50	43	40	0
9	1	5	5	1	1	58	46	44	49	0

Figure 2 Replacing categorical values with numerical

The following Figure presents the description of the dataset:



	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Overall Score	Pass/Fail
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.506000	0.540000	0.403800	0.332000	0.331000	0.607529	0.640798	0.624071	0.623094	0.718000
std	0.500214	0.282365	0.319073	0.471167	0.470809	0.178564	0.175191	0.183570	0.169574	0.450198
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.250000	0.200000	0.000000	0.000000	0.482353	0.523810	0.500000	0.517647	0.000000
50%	1.000000	0.500000	0.400000	0.000000	0.000000	0.611765	0.642857	0.630952	0.635294	1.000000
75%	1.000000	0.750000	0.600000	1.000000	1.000000	0.741176	0.761905	0.750000	0.741176	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figure 3 Dataset description

In order to assess the presence of outliers in the dataset, we employed a boxplot visualization. The boxplot provides a concise summary of the distribution of the data, highlighting the interquartile range (IQR) and identifying potential outliers beyond the whiskers. As shown in the figure 4, which includes boxplots with and without outliers, we proceeded to remove the outliers using Z-score normalization:

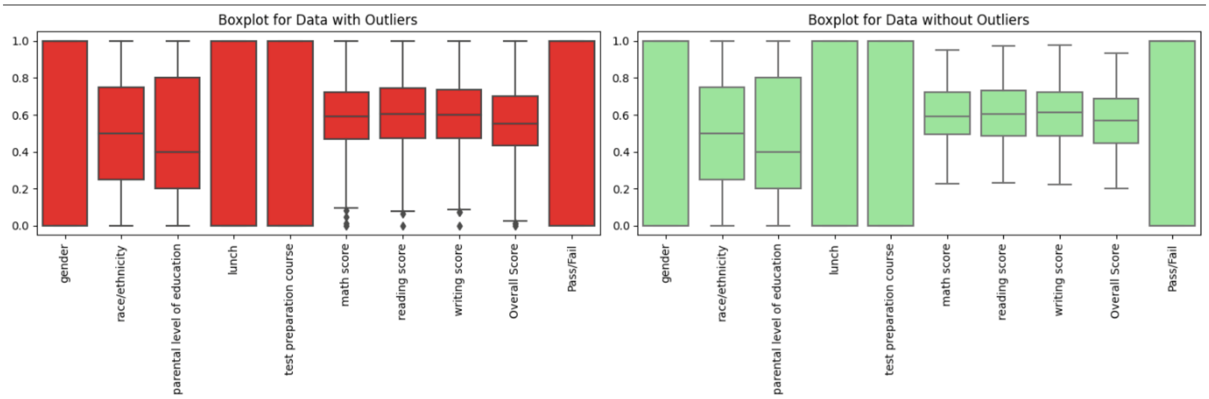
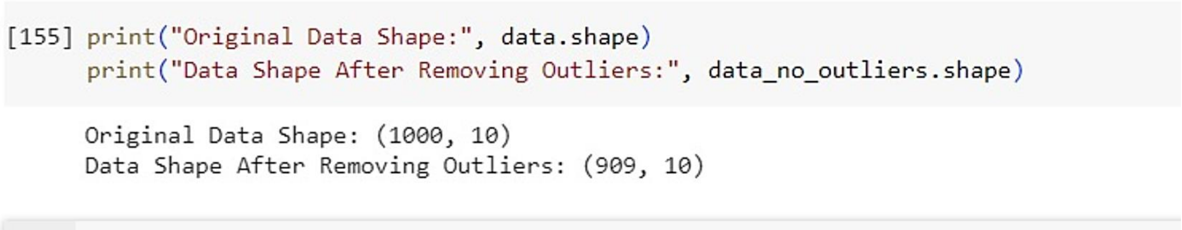


Figure 4 Remove the outlier



```
[155] print("Original Data Shape:", data.shape)
      print("Data Shape After Removing Outliers:", data_no_outliers.shape)

      Original Data Shape: (1000, 10)
      Data Shape After Removing Outliers: (909, 10)
```

Figure 5 Dataset size after remove the outliers

In order to address the issue of class imbalance in the dataset, we utilized a RandomOverSampler technique. This method involves oversampling the minority class to achieve a more balanced distribution of classes. As shown in the figure 6, which includes visualizations before and after oversampling, we applied RandomOverSampler to mitigate the class imbalance and enhance the representation of the minority class in the dataset:

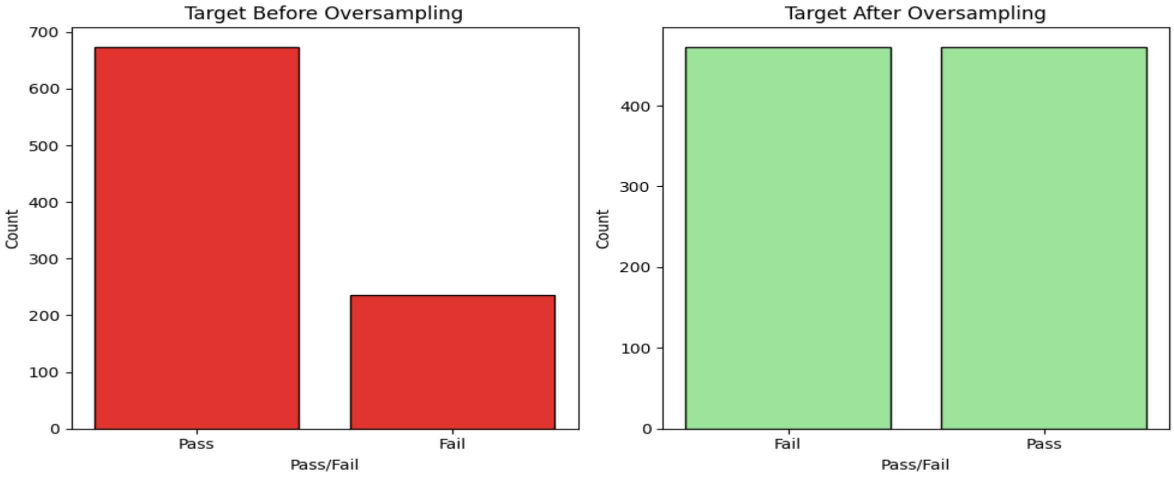
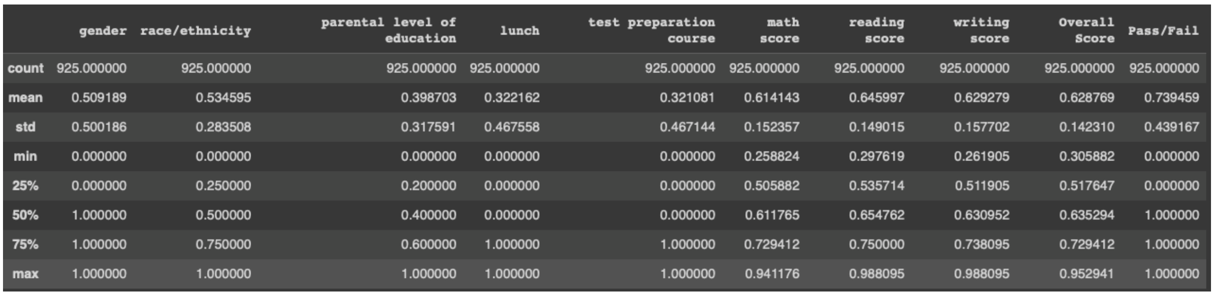


Figure 6 Over Sampling

Given the absence of any missing values, in figure 7 is the description of the dataset after addressing outliers and handling class imbalance:



	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Overall Score	Pass/Fail
count	925.000000	925.000000	925.000000	925.000000	925.000000	925.000000	925.000000	925.000000	925.000000	925.000000
mean	0.509189	0.534595	0.398703	0.322162	0.321081	0.614143	0.645997	0.629279	0.628769	0.739459
std	0.500186	0.283508	0.317591	0.467558	0.467144	0.152357	0.149015	0.157702	0.142310	0.439167
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.258824	0.297619	0.261905	0.305882	0.000000
25%	0.000000	0.250000	0.200000	0.000000	0.000000	0.505882	0.535714	0.511905	0.517647	0.000000
50%	1.000000	0.500000	0.400000	0.000000	0.000000	0.611765	0.654762	0.630952	0.635294	1.000000
75%	1.000000	0.750000	0.600000	1.000000	1.000000	0.729412	0.750000	0.738095	0.729412	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	0.941176	0.988095	0.988095	0.952941	1.000000

Figure 7 Handle imbalance

2.2 Experimental setup

Following the preprocessing stage, we will now investigate the experimental study's specifics. The integrated development environment (IDE) for the study was Jupyter Notebook, which was built with Python and the Scikit-learn module. The primary objective of this study is to predict whether a student's overall grade qualifies for a pass or not. For analysis, a dataset of 909 rows of data was obtained.

To begin the investigation, we separated the dataset into a training set and a testing set using Scikit-learn's "train_test_split" method. The training dataset accounted for 70% of the data, with the remaining 30% constituting the testing dataset. Following that, we built prediction models using various classification techniques, as described in Section 3. On the dataset, the chosen methods are Decision Tree, Support Vector Machines (SVM), and Naive Bayes were used independently.

Each model was trained and evaluated using relevant performance indicators, allowing for a thorough comparison of the three techniques. This comparison research provides significant insights into each algorithm's strengths and drawbacks in terms of prediction accuracy for forecasting student achievement using machine learning approaches.

2.3 Performance measure

The following performance measures have been used:

Accuracy: The overall correctness of the model's predictions.

$$\frac{(TP + TN)}{(TP + FP + FN + TN)}$$

Precision: The accuracy of the positive predictions made by the model.

$$\frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})}$$

Recall: The ability of the model to capture all positive instances.

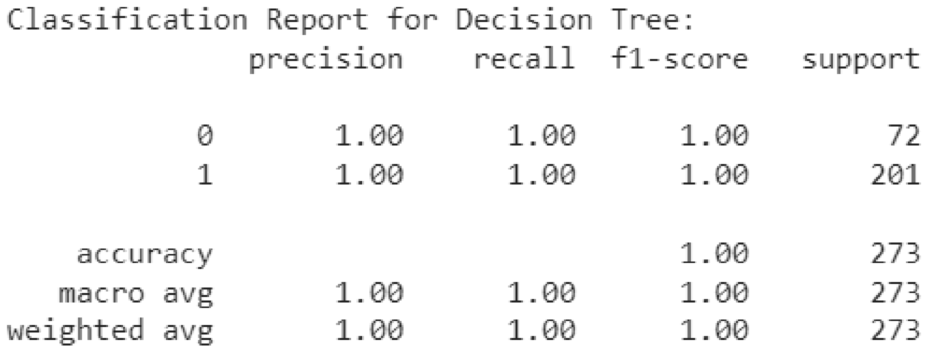
$$\frac{TP}{(TP + FN)}$$

F1 Score: A balanced measure combining precision and recall.

$$\frac{2(precision\ \ x\, recall)}{(precision\ +\ recall)}$$

3. Result and discussion

In the evaluation of the Decision Tree (DT) model, we achieve perfect predictions with an accuracy of 100%. For class 0, precision, recall, and the F1-score are all 100%, indicating flawless performance. Similarly, class 1 demonstrates impeccable precision, recall, and F1-score, contributing to the overall success of the model, Classification Report shown below:



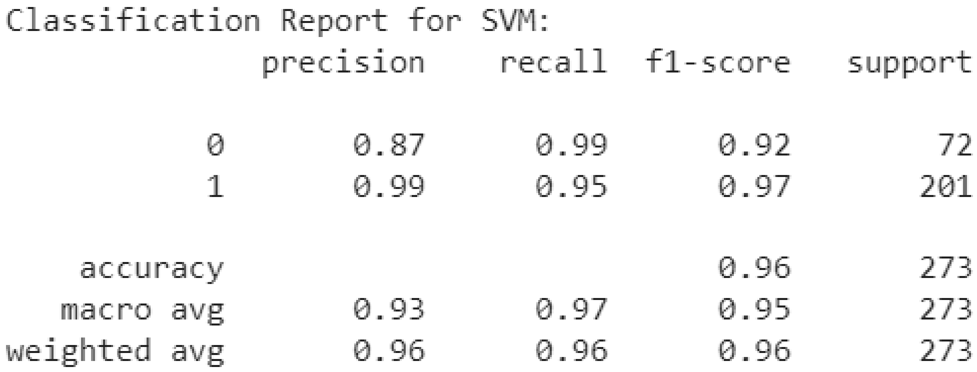
```
Classification Report for Decision Tree:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        72
           1       1.00      1.00      1.00       201

    accuracy                           1.00       273
   macro avg       1.00      1.00      1.00       273
weighted avg       1.00      1.00      1.00       273
```

Figure 8 DT Result

Moving on to the Support Vector Machines (SVM) model, we observe an accuracy of 96%. The precision for class 0 is 87%, indicating 87% accuracy in predicting instances of class 0. The recall for class 0 is 99%, showing the model's effectiveness in identifying 99% of actual instances of class 0. The F1-score for class 0, representing a balanced measure of the model's performance, is 92%. For class 1, the precision is 99%, the recall is 95%, and the F1-score is 97%. The macro-average and weighted average F1-scores stand at 95%, emphasizing the robust predictive capabilities of the SVM model, Classification Report shown below:



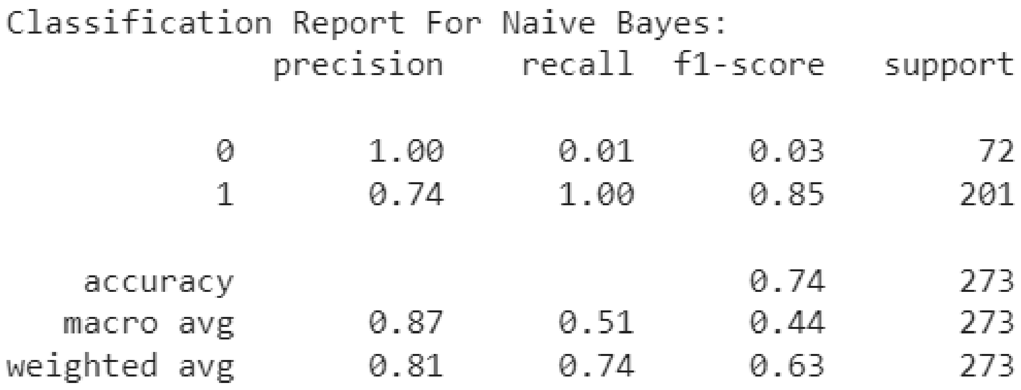
```
Classification Report for SVM:
              precision    recall  f1-score   support

           0       0.87      0.99      0.92        72
           1       0.99      0.95      0.97       201

    accuracy                           0.96       273
   macro avg       0.93      0.97      0.95       273
weighted avg       0.96      0.96      0.96       273
```

Figure 9 SVM Result

In the case of Naive Bayes, an accuracy of 74% is achieved. The precision for class 0 is 100%, indicating that all predicted instances of class 0 are correct. However, the recall for class 0 is just 1%, resulting in a nuanced performance for class 0. For class 1, the precision is 74%, the recall is 100%, and the F1-score is 85%. The macro-average F1-score, reflecting the average performance across both classes, is 44%, highlighting the model's limitations in dealing with class imbalances, Classification Report shown below:



```
Classification Report For Naive Bayes:
              precision    recall  f1-score   support

           0       1.00      0.01      0.03        72
           1       0.74      1.00      0.85       201

    accuracy                           0.74       273
   macro avg       0.87      0.51      0.44       273
weighted avg       0.81      0.74      0.63       273
```

Figure 10 NB Result

Decision Tree model stands out as the optimal choice, achieving flawless accuracy, precision, recall, and F1-scores in both classes. While the SVM performs well at 96% accuracy, the Decision Tree's perfect forecasts make it the preferred option for predicting student performance in this context. The Naive Bayes model, though accurate for class 0, faces imbalances and achieves a lower macro-average F1-score. Overall, the Decision Tree excels in forecasting student success.

A Confusion Matrix is a tabular representation of a classification algorithm's performance in machine learning. It breaks into true positive (TP), true negative (TN), false positive (FP), and false negative (FN) forecasts in great depth. These components allow for the computation of several performance measures including as accuracy, precision, recall, and F1-score. The matrix provides insight into how well a model performs, particularly in terms of correctly and wrongly predicted examples across different classes. Below shows the Confusion Matrix for each algorithm that was used:

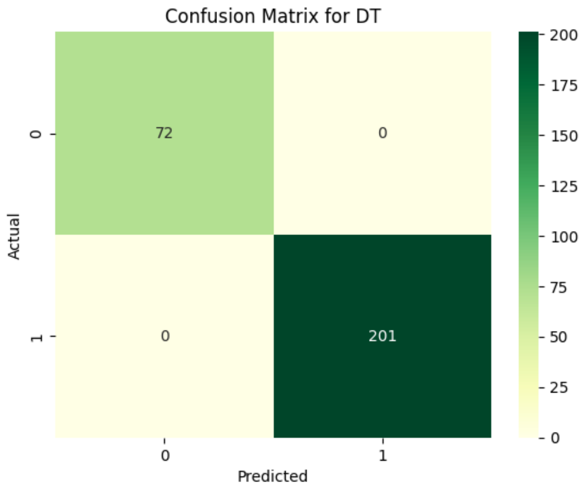


Figure 11 Confusion Matrix Of DT

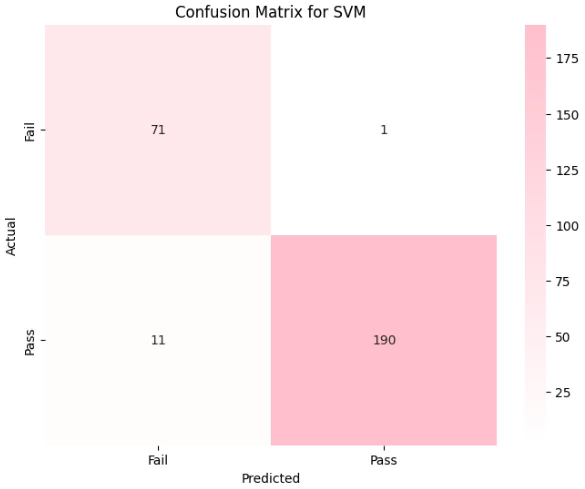


Figure 12 Confusion Matrix Of SVM

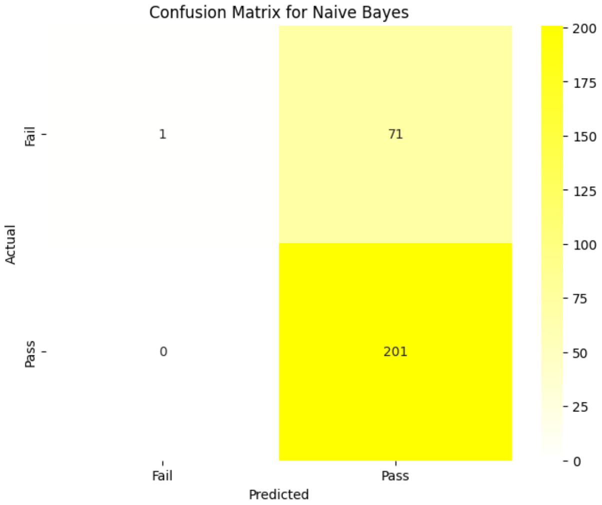


Figure 13 Confusion Matrix Of NB

4. Conclusion

In the end, we used three different machine learning models, including Decision Tree (DT), Support Vector Machines (SVM), and Naive Bayes, to predict student achievement based on a precisely produced dataset. The findings indicate the Decision Tree model's supremacy, as it not only attained faultless accuracy but also displayed remarkable precision, recall, and F1-scores for both pass and fail classes. This exceptional result establishes the Decision Tree as the most trustworthy option for forecasting student outcomes in this specific situation. Although the SVM model achieved 96% accuracy, the Decision Tree's perfect predictions made it the better choice for this dataset. Naive Bayes, on the other hand, struggled with imbalances, as seen by a lower macro-average F1-score. Finally, the Decision Tree model outperforms other educational prediction models in terms of forecasting student achievement.