Final Project Report

Course: CS4375.501
Group Members:

- Ruolan Liu (RXL200022)
- Natthiya Sae Ngow (NXS220110)

1. Introduction

We explore the datasets sources from Kaggle, selecting four datasets for in-depth analysis: Lifestyle Factors and Their Impact on Students, AI-Generated Ghibli Style Image Trends(2025), Average Daily Screen Time for Children, and Cost of International Education. The primary objective is to apply various machine learning techniques to each dataset, experiment with four distinct models, evaluate their performance, and conduct a comparative analysis of the results across all datasets.

2. Dataset Overview - Ruolan Liu

2.1 Dataset 1: student_lifestyle_dataset..csv

- Dataset Introduction:
  This dataset provides students' lifestyle patterns and their correlation with academic performance, which is represented by GPA. It contains 2000 records of students' daily habits including study, extracurriculars, sleep, etc. Each student's stress level is according to the study and the sleep time of each student. This dataset provides a deep understanding of how lifestyle affects academic outcomes. (Description adapted from the dataset source)

- Data Pre-processing:
  We examined the dataset for general information, missing values and category distributions. The dataset contains 2000 entries and 9 columns including both numeric and categorical variables.
  - Missing Values & Duplicated Values: There are no missing values or duplicated values in the columns, according to the output, we made a table here:

Column	Missing Values
Student_ID	0
Study_Hours_Per_Day	0
Extracurricular_Hours_Per_Day	0
Sleep_Hours_Per_Day	0
Social_Hours_Per_Day	0
Physical_Activity_Hours_Per_Day	0
Stress_Level	0
Gender	0
Grades	0

- Category Distributions:

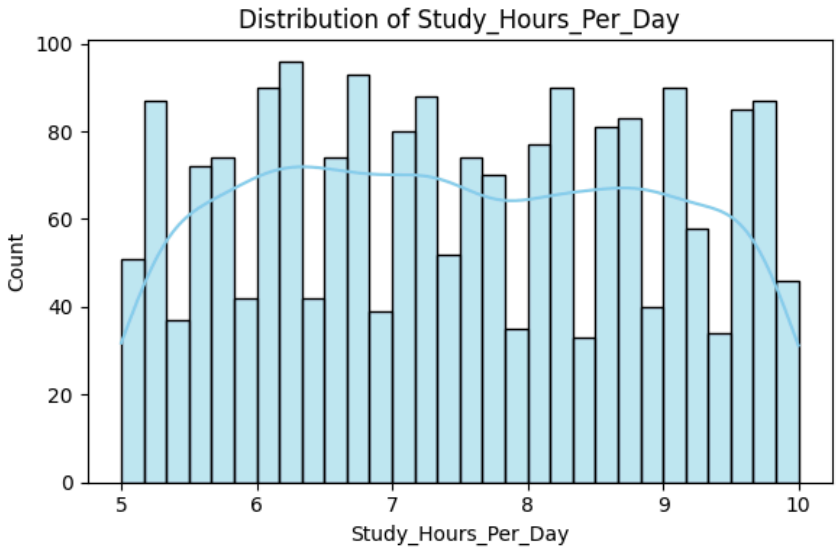
Stress Level	Counts
High	1029
Modeate	674
Low	297

Gender	Counts
Male	1016
Female	984

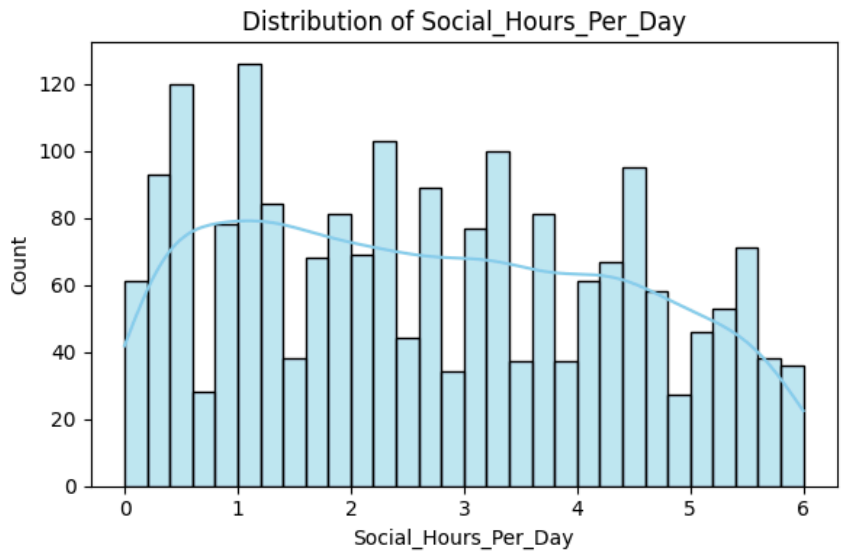
These shows that the dataset is complete and ready for further analysis.

Next, we plotted the histograms with KDE (Kernel Density Estimation) curves to understand the distribution of numeric features. The results are following:

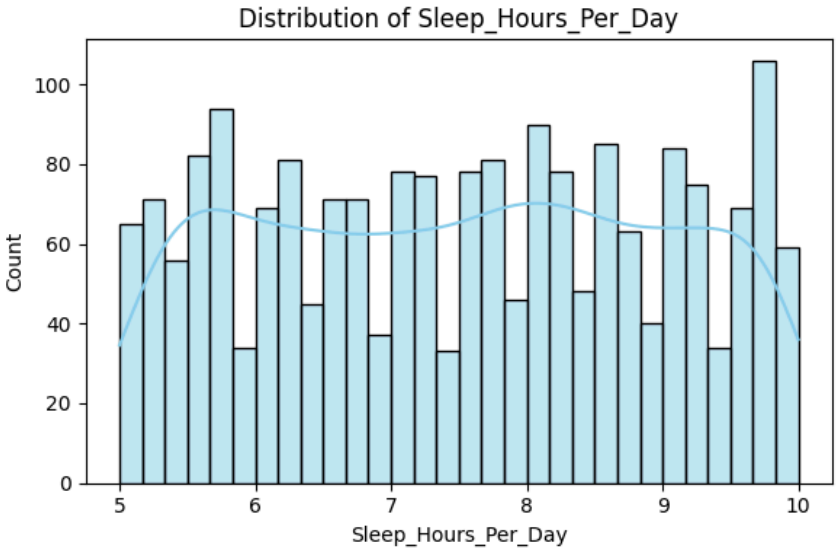
- Study_Hours_Per_Day: This distribution appears approximately uniform. Most of the values spread evenly between 5 and 10 hours.



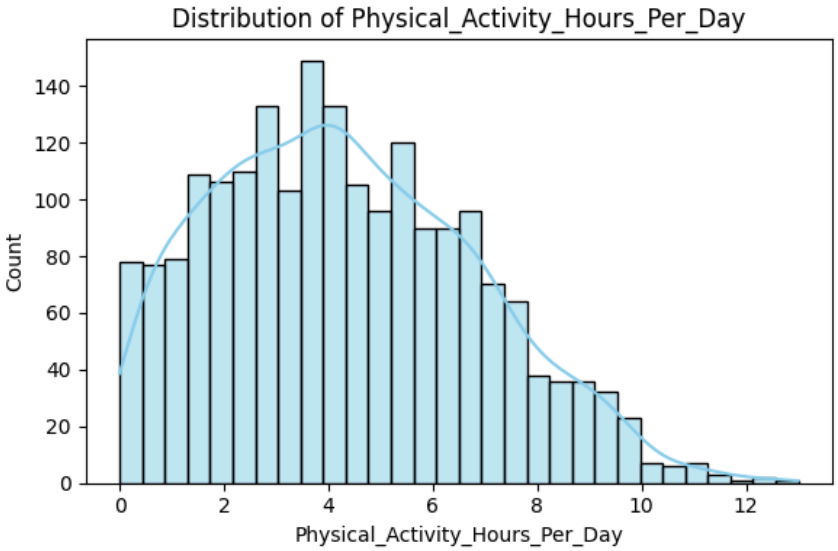
- Social_Hours_Per_Day: This distribution is right-skewed, more students are spending less hours on social activities.



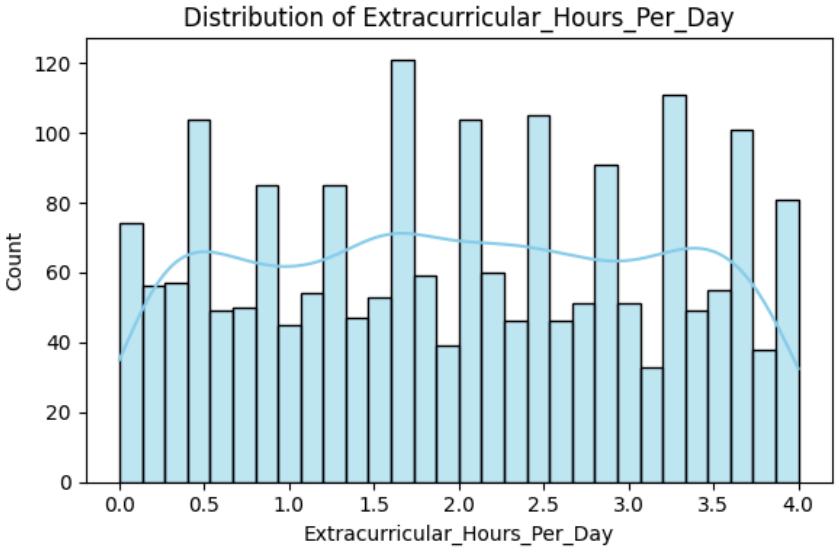
- Sleep_Hours_Per_Day: The values are approximately evenly distributed which range from 5 to 10 hours.



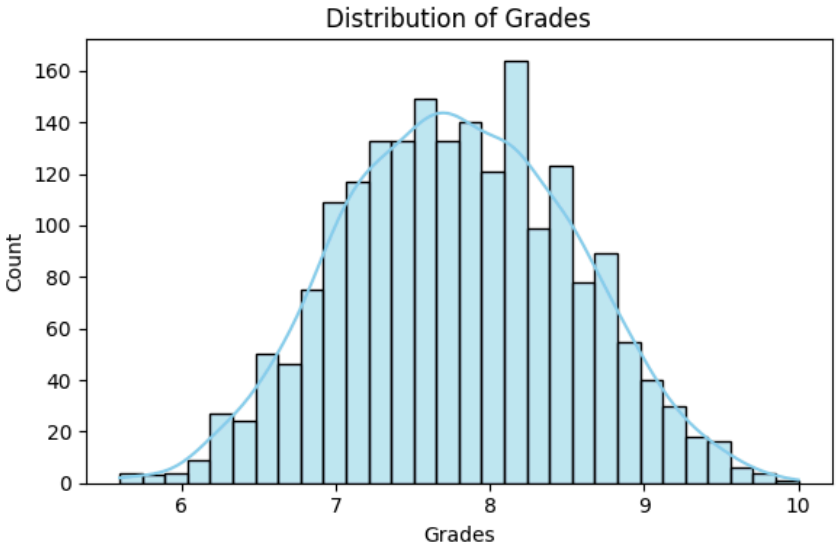
- Physical_Activity_Hours_Per_Day: This shows a moderately right-skewed distribution that concentrates around 2 to 6 hours.



- Extracurricular_Hours_Per_Day: This feature is close to a uniform distribution between 0 to 4 hours.

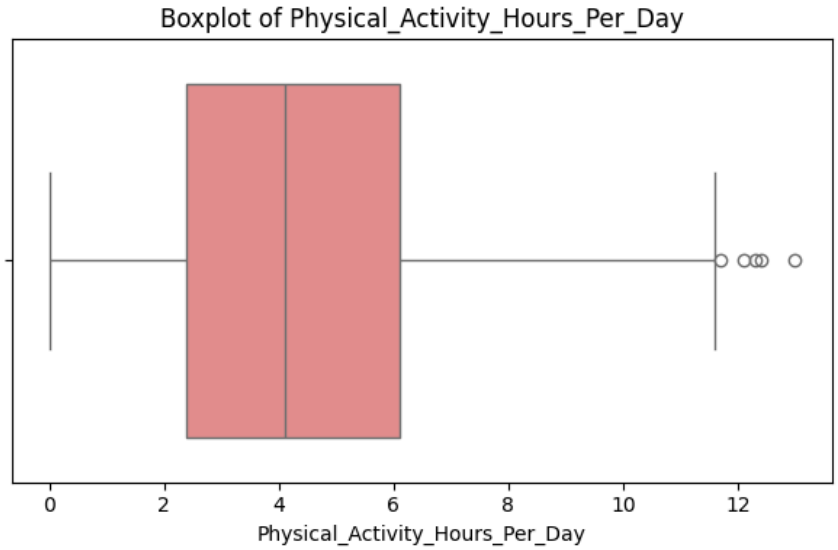


- Grades: The grades are approximately normally distributed. The major values are around 7 to 8.

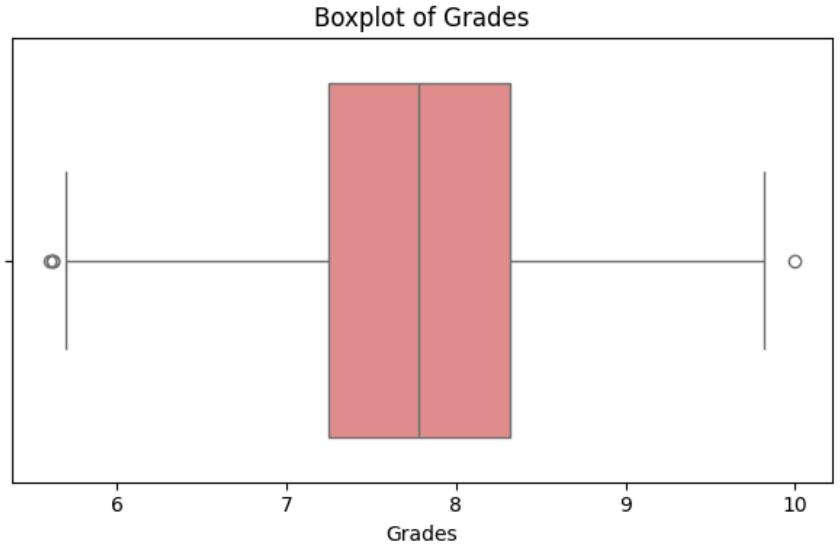


Then, we also used the boxplots to identify potential outliers in the numeric variables. Most of the features show compact interquartile ranges with no extreme outliers, except:

- Physical_Activity_Hours_Per_Day: A few values which are above 11-12 hours may be considered mild outliers.

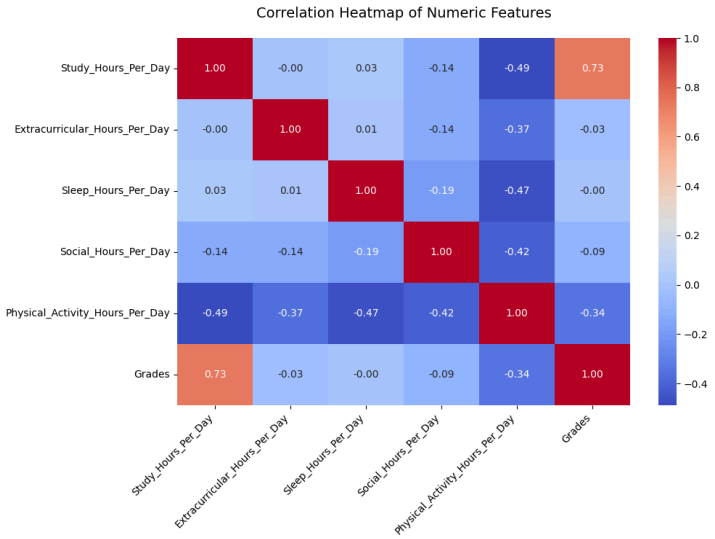


- Grades: A small number of points are lower than 6 and near 10.0 appear as mild outliers but still in the reasonable bounds.



These distributions confirmed that the numeric features are generally well-scaled with few variables that require outlier handling.

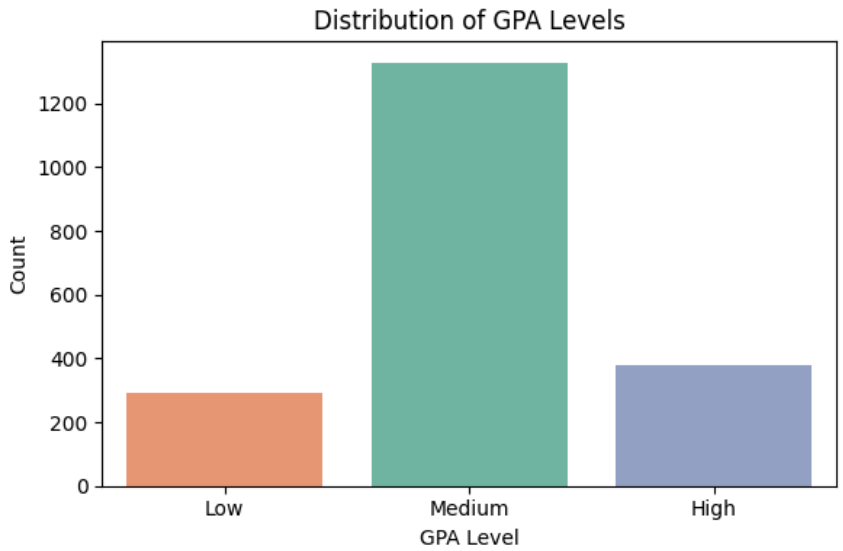
This correlation heatmap shows the pairwise Pearson correlation coefficients between all numeric variables. On the one hand, the strongest positive correlation with Grades is shown in Study_Hours_Per_Day (r = 0.73) shows that longer study hours are associated with higher academic performance. On the other hand, Physical_Activity_Hours_Per_Day has a moderate negative correlation with Grades (r = -0.34) that may indicate that students who spend more time on physical activities will get slightly lower grades. Other variables show relatively weak correlations.



We created a new target variable called GPA_Level to convert the continuous Grades variable into a categorical label which is suitable for classification. The Grades were categorized into three levels:

- Grades < 7.0 → “Low”
- 7.0 ≤ Grades < 8.5 → “Medium”
- Grades ≥ 8.5 → “High”

Here is the distribution of GPA Levels:



Among the 2000 records, the majority of students are in the “Medium” GPA range. The “Low” and “High” categories are relatively balanced in size.
Then, we convert three categorical variables, Gender, Stress_Level and GPA_Level to the numerical format by using Label Encoding that allows the models to process categorical inputs.

```
# Label Encoding for categorical variables
label_encoders = {}
for col in ['Gender', 'Stress_Level', 'GPA_Level']:
    le = LabelEncoder()
    df[col + '_Encoded'] = le.fit_transform(df[col])
    label_encoders[col] = le
```

We applied Z-score normalization by using StandardScaler from sklearn to ensure all numeric features are on the same scale. The scaled continuous features are following:

- Study_Hours_Per_Day
- Extracurricular_Hours_Per_Day
- Sleep_Hours_Per_Day
- Social_Hours_Per_Day
- Physical_Activity_Hours_Per_Day

After scaling, the mean of each feature is 0, and the standard deviation is 1 which helps to improve model convergence and interpretability.

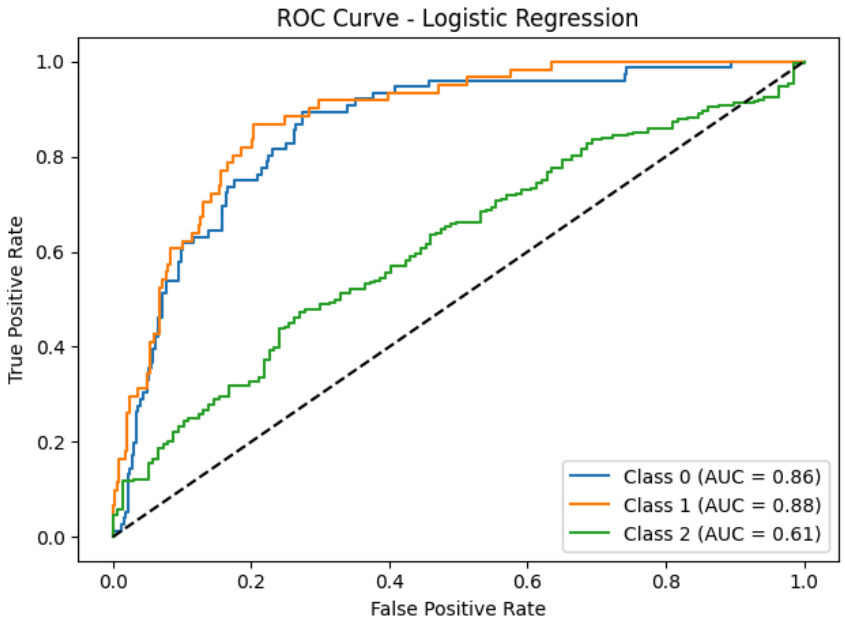
- **Experiments and Learning the Models:**
  For this dataset, we evaluated the dataset by using four classification models. They are Logistic Regression, Decision Tree, Random Forest and K-Nearest Neighbors (KNN). All models used the same set of features and the same target variable GPA_Level_Encoded is for comparison.
  - Data Splitting: The dataset was split into training and testing sets with an 80/20 ratio by using train_test_slit with a fixed random seed for reproducibility.
  - Feature Set Used: Features which used training included both numerical and encoded categorical variables.
    - Study_Hours_Per_Day
    - Extracurricular_Hous_Per_Day

- Sleep_Hours_Per_Day
- Social_Hours_Per_Day
- Physical_Activity_Hours_Per_Day
- Gender_Encoded
- Stress_Level_Encoded

Here are the 4 models' training results according to the output:

- Logistic Regression Evaluation:
  - Accuracy Score: 0.70
  - Classification Report:

Class	Precision	Recall	F1-score	Support
High	0.6	0.42	0.5	76
Low	0.59	0.31	0.41	61
Medium	0.73	0.87	0.79	263
Accuracy			**0.7**	**400**
Macro avg	0.64	0.53	0.57	400
Weighted avg	0.68	0.7	0.68	400

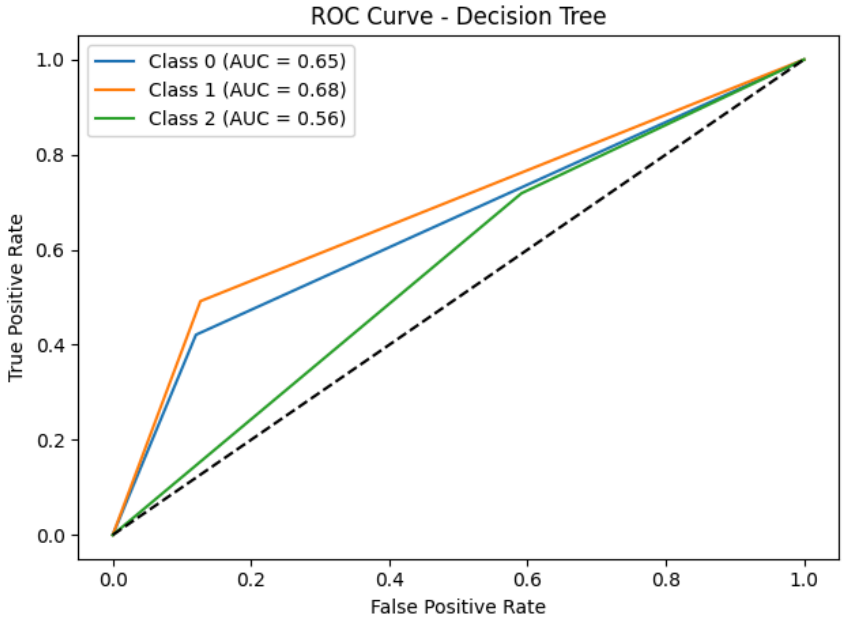


The classification report shows the Logistic Regression performs best on the Medium GPA class with a high recall 0.87 and balanced precision. However, it struggles with the Low GPA class with a 0.31 recall. The ROC vurve aligns with this: AUC values for Class 0 (0.86) and Class 1 (0.88) are higher than Class 2 (0.61). Logistic Regression achieves

a decent accuracy of 70% but maybe favor the majority class.

- Decision Tree Evaluation:
  - Accuracy score: 0.585
  - Classification report:

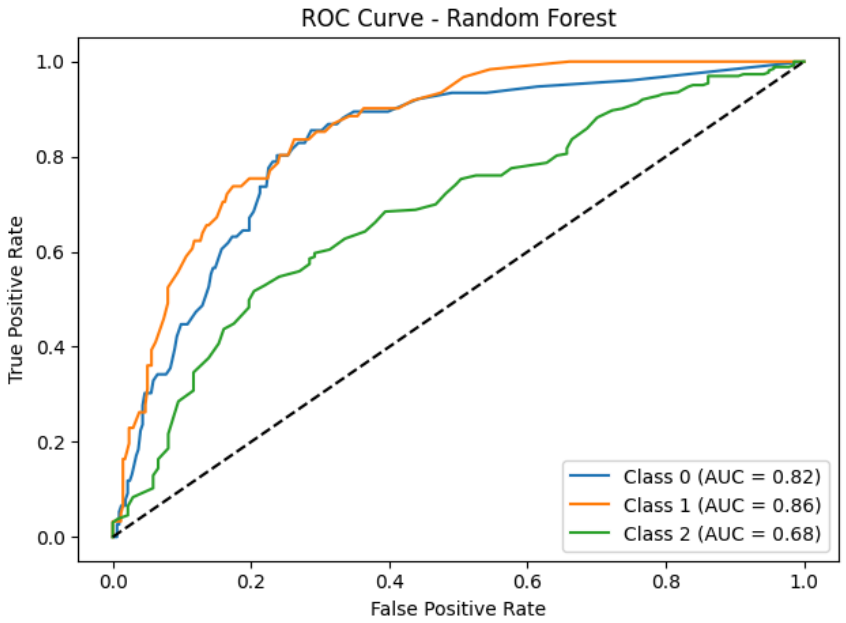
Class	Precision	Recall	F1-score	Support
High	0.42	0.41	0.42	76
Low	0.33	0.34	0.34	61
Medium	0.69	0.69	0.69	263
Accuracy			**0.58**	**400**
Macro avg	0.48	0.48	0.48	400
Weighted avg	0.59	0.58	0.59	400



This classification report shows the Decision Tree performs best on the Medium GPA class with a recall and precision both 0.69. However, it also struggles with the Low GPA class with precision 0.33 and recall 0.34. The ROC curve reflects similar behavior: AUC values for Class 0 (0.65) and Class 1 (0.68) are lower than those in Logistic Regression chart. The AUC value for Class 2 (0.56) is also lower than that for Class 0 and Class 1.

- Random Forest Evaluation:
  - Accuracy score: 0.69
  - Classification report:

Class	Precision	Recall	F1-score	Support
High	0.55	0.38	0.45	76
Low	0.61	0.33	0.43	61
Medium	0.72	0.86	0.79	263
Accuracy			**0.69**	**400**
Macro avg	0.63	0.52	0.55	400
Weighted avg	0.67	0.69	0.67	400

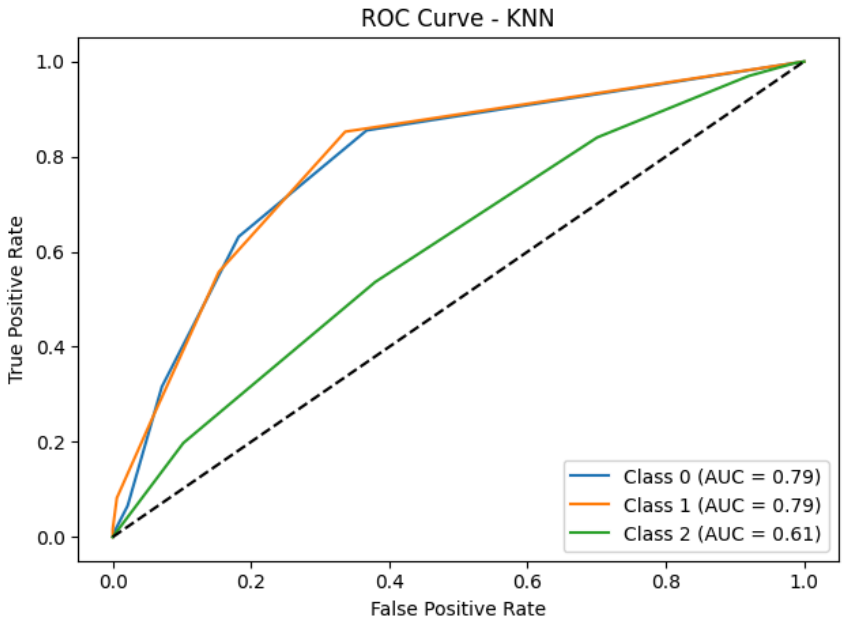


This classification report shows that the Random Forest Model performs best on the Medium GPA class with a recall 0.86 and a precision of 0.72. The Low GPA class has a precision 0.61 but a lower recall of 0.33 which indicates many false negatives. The High GPA class has a precision 0.55 and a lower recall 0.38. These show the model favors the majority class which is Medium GPA.

The AUC value for Class 0 (0.82) and Class 1 (0.86) show strong separability, while Class 2 AUC value 0.68 has weaker performance that shows the model is struggling to distinguish High GPA students.

- KNN Evaluation:
  - Accuracy score: 0.6525
  - Classification report:

Class	Precision	Recall	F1-score	Support
High	0.5	0.32	0.39	76
Low	0.46	0.26	0.33	61
Medium	0.7	0.84	0.79	263
Accuracy			**0.65**	**400**
Macro avg	0.55	0.47	0.49	400
Weighted avg	0.62	0.65	0.63	400

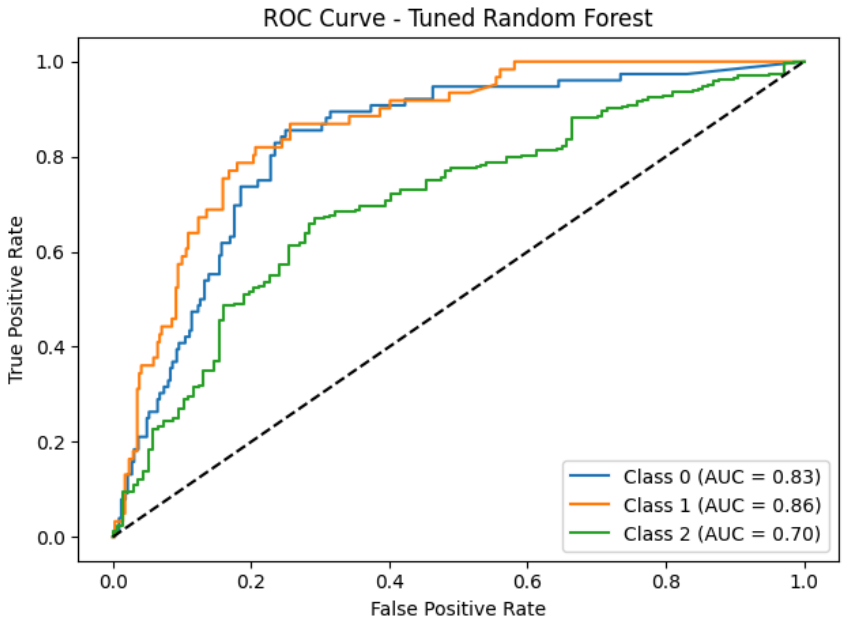


This classification report shows that KNN performs best on the Medium GPA class with a recall 0.84. It struggles with the Low GPA class with a recall 0.26 that shows many Low GPA students are misclassified. The High GPA class has a low recall of 0.32, but it has a precision 0.5.

The AUC values for Class 0 and Class 1 are same, which is 0.79, AUC value for Class 2 is 0.61. KNN has an accuracy of 65% which is weaker than Logistic Regression and Random Forest in this dataset.

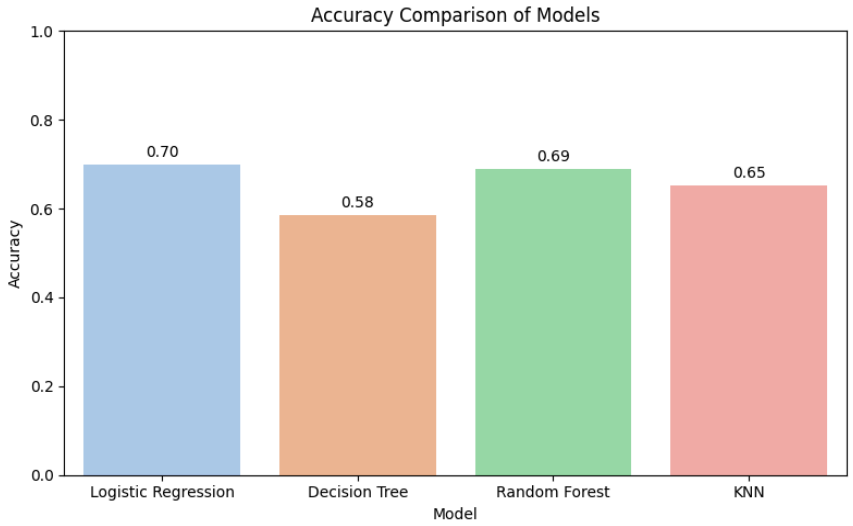
- Tuned Random Forest Evaluation:
  - Accuracy score: 0.6825
  - Classification report:

Class	Precision	Recall	F1-score	Support
High	0.55	0.37	0.44	76
Low	0.57	0.28	0.37	61
Medium	0.71	0.87	0.78	263
Accuracy			**0.68**	**400**
Macro avg	0.61	0.5	0.53	400
Weighted avg	0.66	0.68	0.66	400

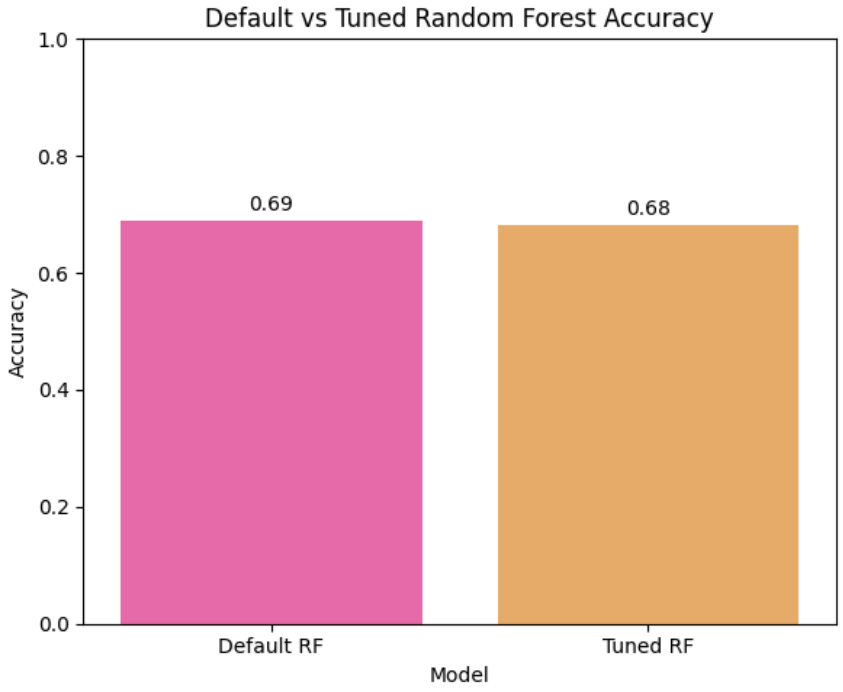


This classification report shows that the tuned Random Forest performs best on the Medium GPA class with a 0.87 recall which is better than the default one. However the accuracy is 0.68 which is lower than the default one (0.69) and the one in Logistic Regression (0.7).

- **Comparing the Models:**
  Here is the comparison chart of the accuracies in these 4 models.



According to the Accuracy Comparison chart, we can tell Logistic Regression has the highest accuracy 0.7, while the Decision Tree has the lowest accuracy 0.58.
Here is the comparison chart of the Accuracy for Tuned Random Forest and the Default Random Forest:



According to the chart, the Default Randon Forest Accuracy (0.69) is a bit higher than the Tuned version (0.68). So tuning Random Forest by using GridSearchCV did not improve its performance.

- **Conclusion:**
  According to the results above, Logistic Regression has the highest accuracy 70% and strong performance on identifying students with Medium GPA, though it struggled with low GPA classes. Random Forest performed well but its accuracy is a little bit lower than the accuracy of Logistic Regression. In conclusion, the results indicate that the Logistic Regression is the most suitable model for this dataset.

2.2 Dataset 2: ai_ghibli_trend_dataset_v2.csv

- **Dataset Introduction:**
  This dataset collects the trends in AI-generated Studio Ghibli-style images, including user engagement, generation metrics, and platform-specific interactions. It is designed to help people for instance researchers, data scientists and AI enthusiasts to analyze the impact of AI-generated pictures on social media and the performance of AI models in generating high-quality images. (Description adapted from the dataset source)

- **Data Pre-processing:**
  We started our analysis by testing the structure and quality of the dataset. This dataset has 500 records and 16 columns that cover metadata such as interaction metrics (likes, shares, comments), generation parameters (GPU usage, generation time), platform information and stylistic evaluation scores.
    - Missing Values & Duplicated Values: There are no missing values or duplicated values in the columns, according to the output, we made a table:

Column	Missing Values
image_id	0
user_id	0
prompt	0
likes	0
shares	0
comments	0
platform	0
generation_time	0
gpu_usage	0
file_size_kb	0
resolution	0
style_accuracy_score	0
is_hand_edited	0
ethical_concerns_flag	0
creation_date	0
top_comment	0

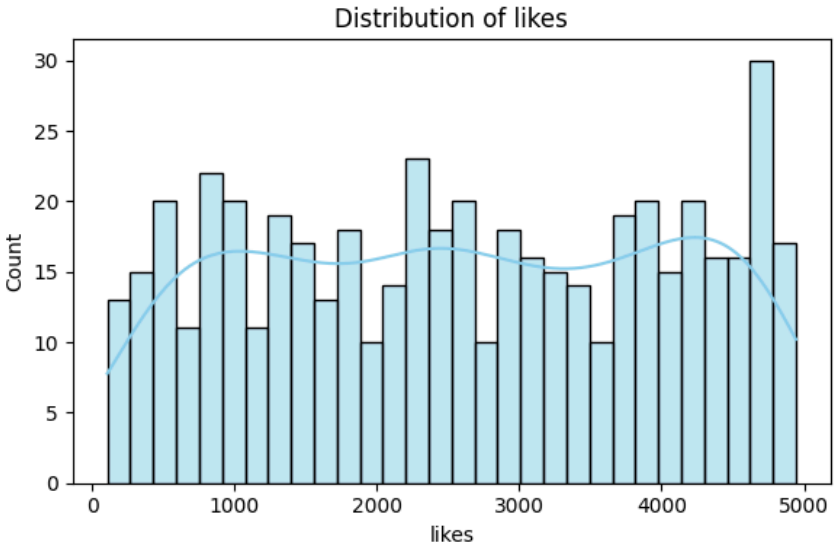
- Category Distributions:

Style_Level	Counts
Medium	204
High	148
Low	148

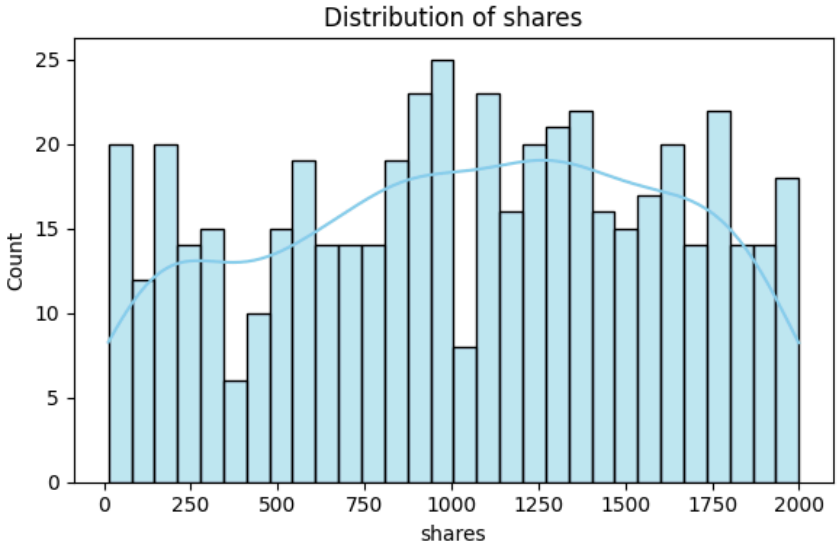
These show that the dataset is complete and there are no missing values.

Next, we plotted the histograms with KDE curves to understand the distributions of numeric features. The results are following:

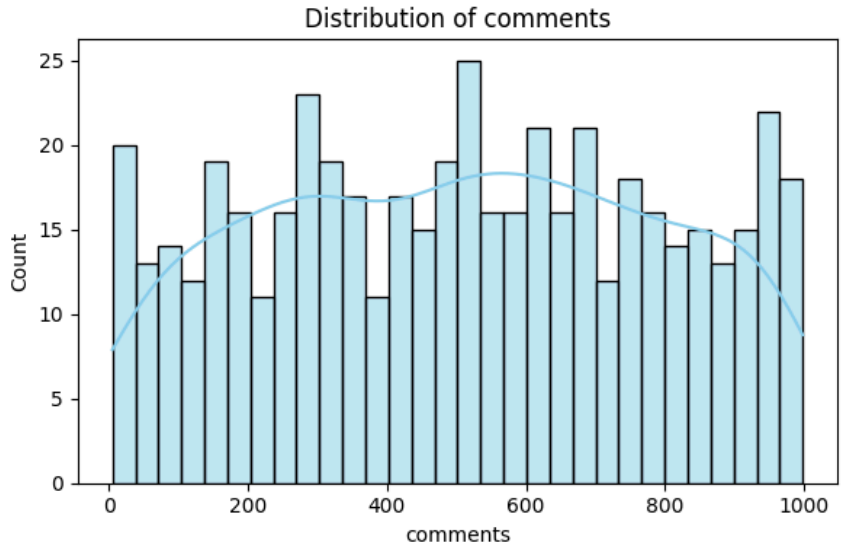
- likes: This distribution shows an approximately uniform distribution across most of its range and exists one noticeable spike at the upper boundary near 5000.



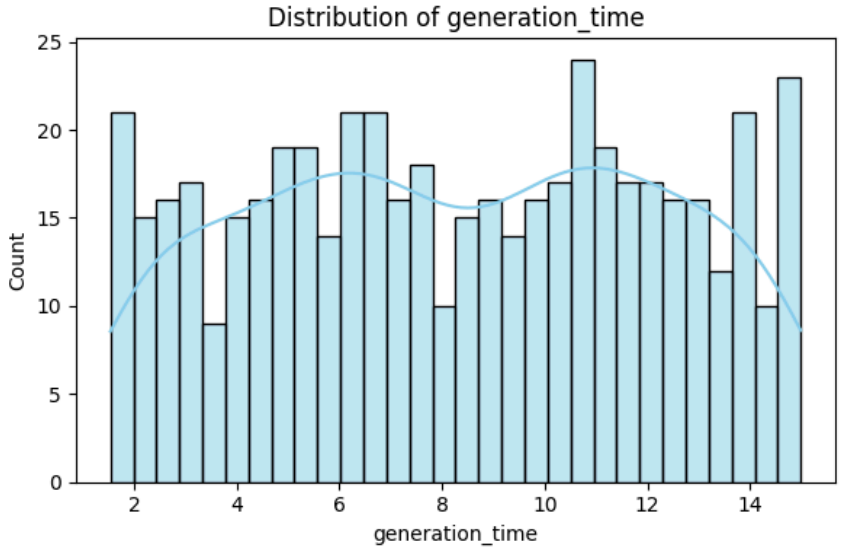
- shares: This one shows an approximately uniform districbution. with a slight increase between 700 and 1300 times.



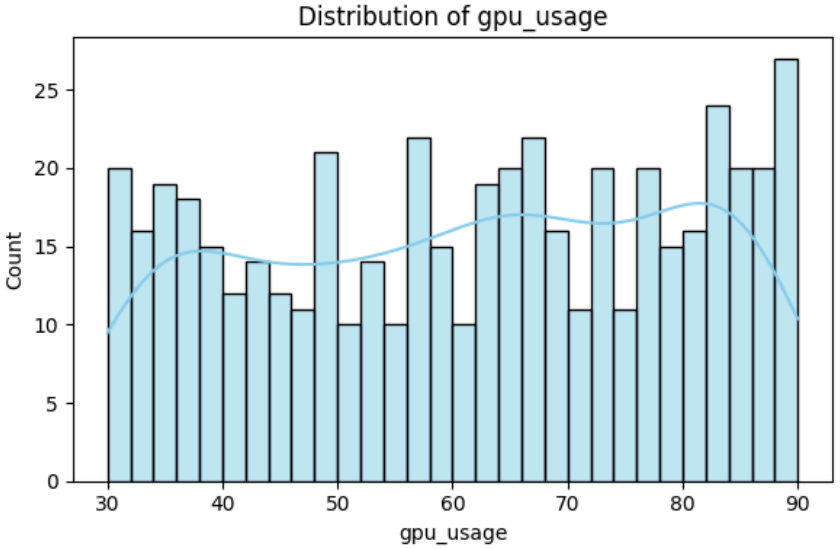
- comments: This one appears approximately uniform with a slight increase around the 400-700 range.



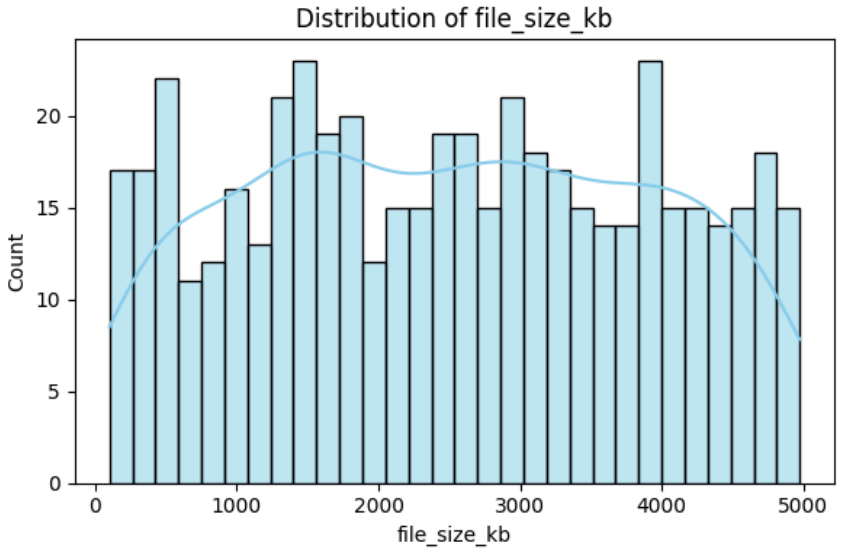
- generation_time: This distribution is also approximately symmetric. It has small picks around 11 seconds. There is no strong skew or tail, which suggests that it is a well-balanced generation environment.



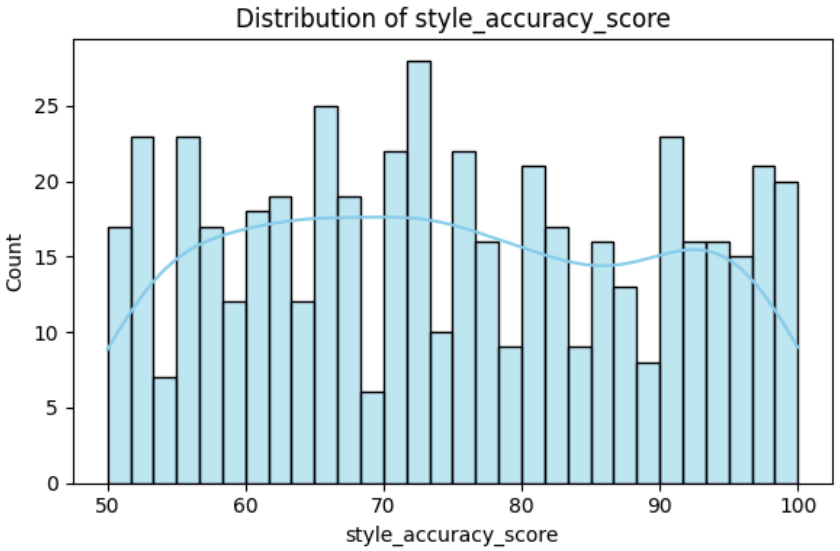
- gpu_usage: This distribution is approximately uniform with mild fluctuations across the values. There is a slight increase near the upper bound of 90 but the overall usage values are daily evenly spread.



- file_size_kb: This one shows an approximately uniform distribution. This shows that the image file sizes vary widely depending on the generation content.

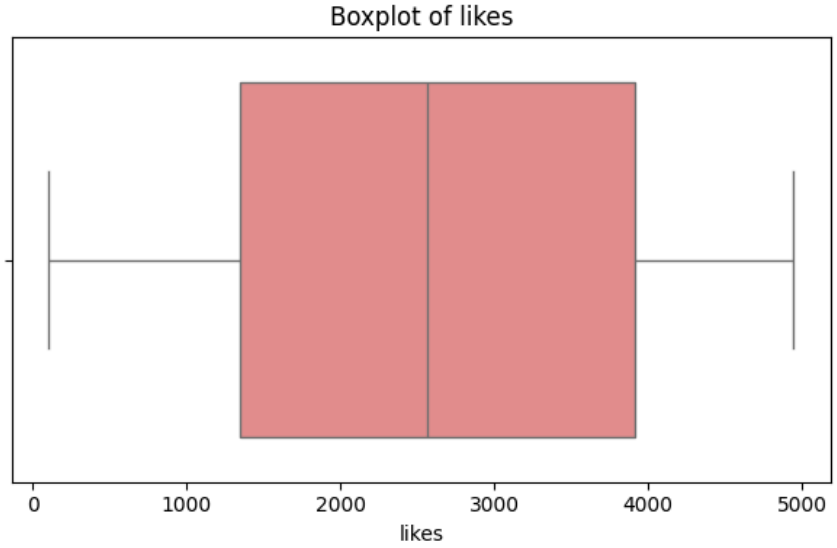


- style_accuracy_score: This one shows an approximately uniform distribution with slight multi-modal tendencies. The values are generally well spread across the 50 to 100 range. This balanced distribution makes it a good candidate for categorical segmentation into low, medium and high stylistic quality levels.

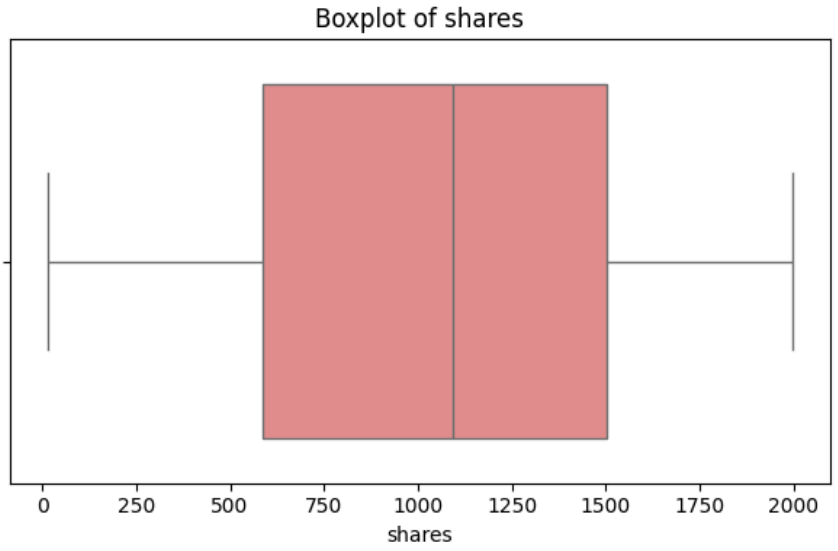


Then, we also examined each numeric feature by using boxplots to detect potential outliers. According to the result, none of the numeric features showed any extreme outliers in the boxplots. This confirms that the dataset is clean and consistent and no special outlier treatment is required before model training.
Here are the boxplots:

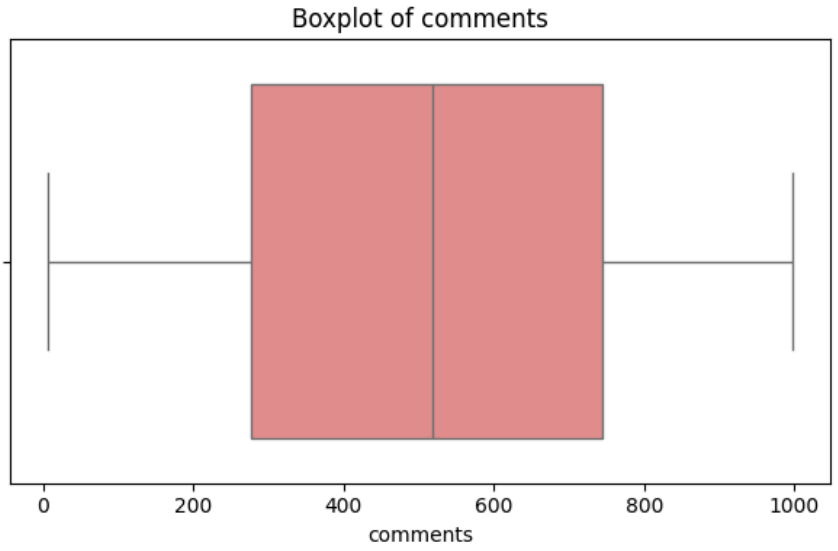
- likes:



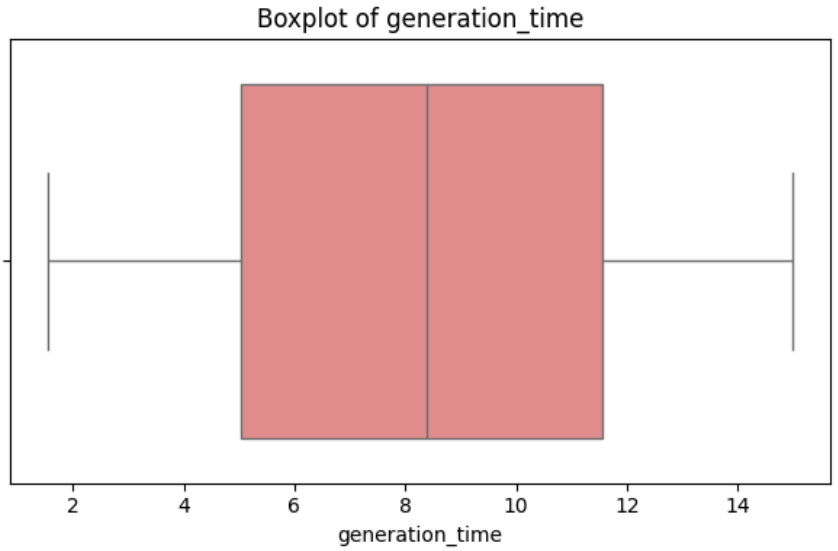
- shares:



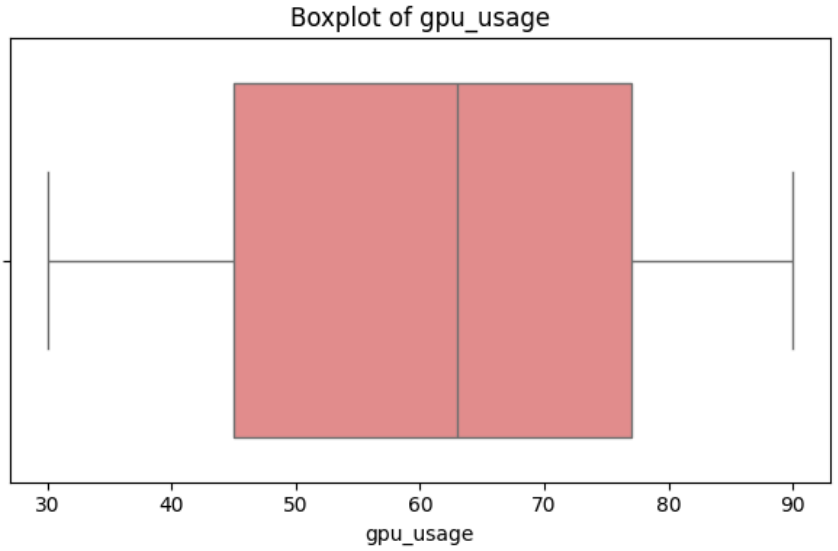
- comments:



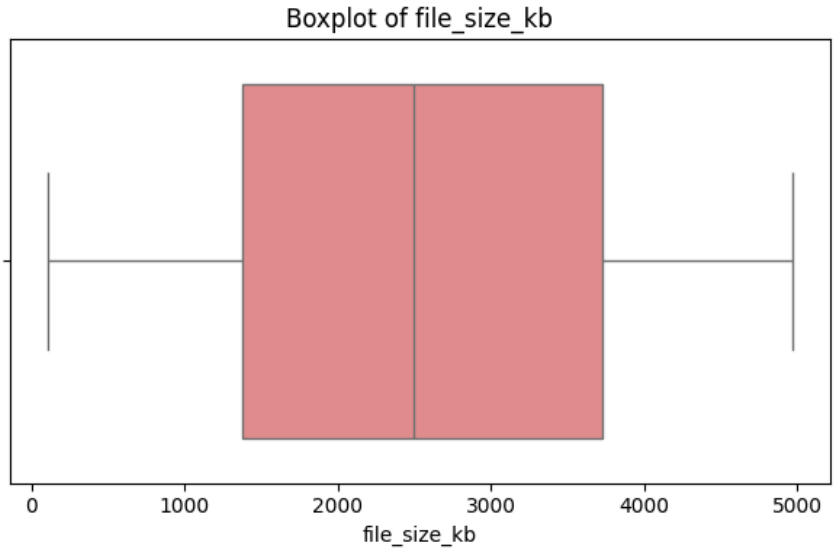
- generation_time:



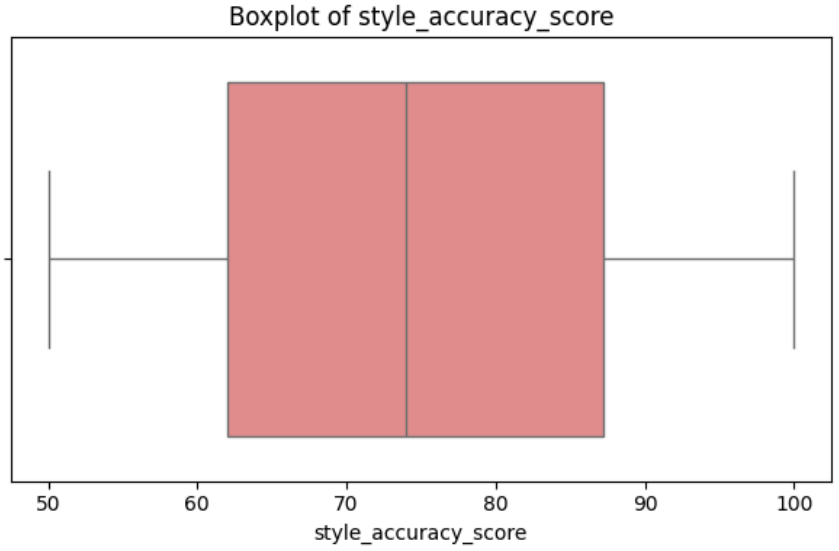
- gpu_usage:



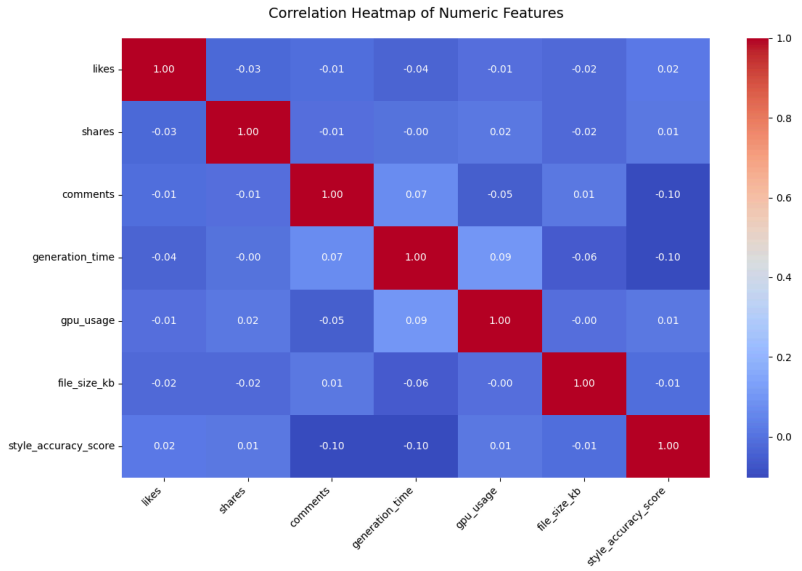
- file_size_kb:



- style_accuracy_score:



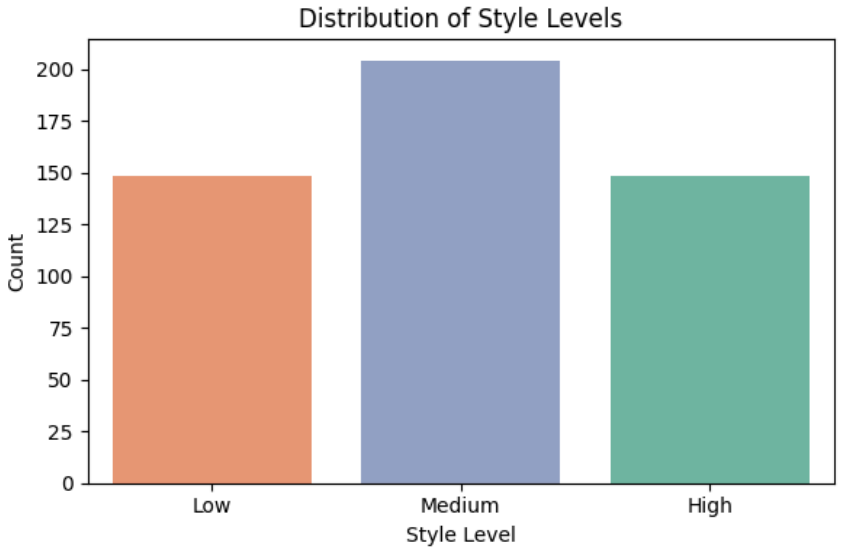
This correlation heatmap shows the pairwise Pearson correlation between all numeric variables. The correlation values are generally weak across the board that there are not strong linear relationships between most features. On the one hand, the strongest positive correlation is between generation_time and gpu_usage (r=0.09) which shows that longer generation time is mildly associated with higher GPU usage. On the other hand, style_accuracy_score shows very weak correlations with all other features. The variables appear to be fairly independent with most correlations falling in the -0.1 to +0.1 range.



We created a new target variable called Style_Level to convert the continuous style_accuracy_score into a categorical label suitable for multi-class classification. The scores were categorized into three levels:

- style_accuracy_score < 65 → “Low”
- 65 ≤ style_accuracy_score < 85 → “Medium”
- style_accuracy_score ≥ 85 → “High”

Here is the distribution of Style Levels:

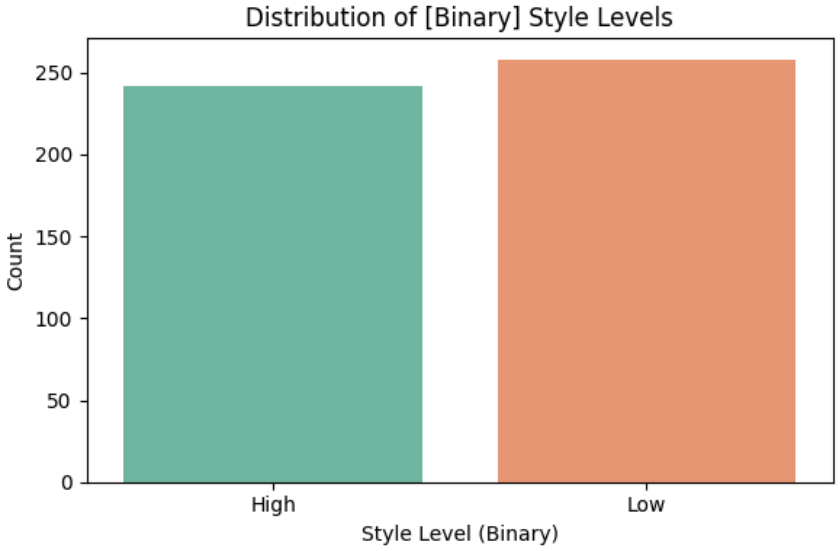


Among the 500 images, the majority are in the “Medium” style level range, the “Low” and “High” categories are nearly balanced which shows a well-distributed range of stylistic quality.

However, after evaluating the classification performance by using five models (Logistic Regression, Decien Tree, Random Forest, KNN and

SVM), we found that the accuracy of these five models was low. So we decided to redefine the problem as a binary classification task to simplify the learning process.
We also created a second target label called Style_Binary:

- style_accuracy_score ≥ 75 → “High”
- style_accuracy_score < 75 → “Low”



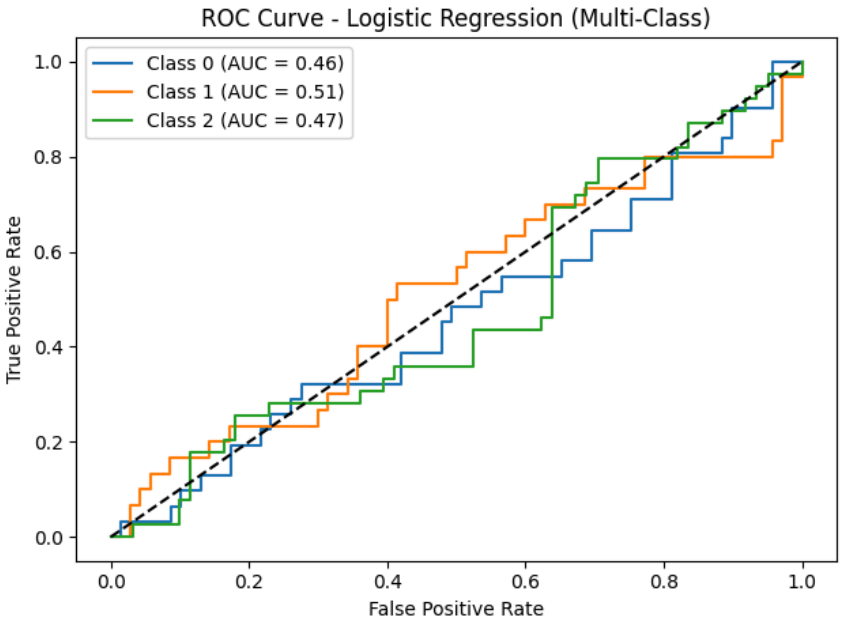
This one shows a clearer distinction. The new binary label exhibits an almost even distribution between the two classes. Then, we re-evaluated the same five models using this binary setup with an additional tuned Random Forest model.
In preparation for modeling, we applied Label Encoding to all categorical columns. We also parsed the resolution column into numeric width and height fields. We applied Z-score normalization by using StandardScaler to scale all numeric features. The scaled variables include:

- likes
- shares
- comments
- generation_time
- gpu_usage
- file_size_kb
- width
- hight
- platform_Encoded
- is_hand_edited_Encoded
- ethical_concerns_flag_Encoded

After scaling, each feature has a mean of 0 and a standard deviation of 1 that ensures consistent magnitude across inputs and improves model convergence.

- **Experiments and Learning the Models**
  For this dataset, we evaluated the dataset by using five classification models. They are Logistic Regression, Decision Tree, Random Forest, KNN and SVM.
  Here are the 5 models' training results for multi-class classification:
    - Logistic Regression Evaluation:
      - Accuracy Score: 0.4
      - Classification Report:

Class	Precision	Recall	F1-score	Support
High	0.14	0.03	0.05	31
Low	0.5	0.1	0.17	30
Medium	0.41	0.92	0.57	39
Accuracy			0.4	100
Macro avg	0.35	0.35	0.26	100
Weighted avg	0.36	0.4	0.29	100

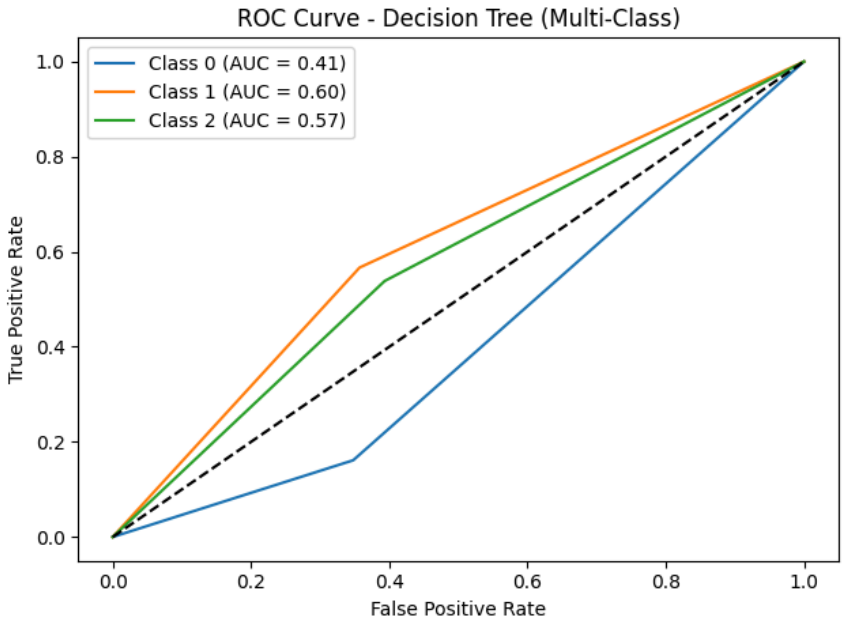


The classification report shows that the model tends to overpredict the Medium style class with a high recall 0.92 but with lower precision 0.41. The low style class has the

highest precision 0.5 that shows more reliable predictions when the model labels a sample as “Low” even though its recall is only 0.1. The Class 1 got the highest AUC 0.51, but still barely above chance. The AUC values of Class 0 and Class 2 are lower than that in Class 1. The models struggles to distinguish between these three style levels and performs only slightly better than random guessing.

- Decision Tree Evaluation:
  - Accuracy score: 0.35
  - Classification report:

Class	Precision	Recall	F1-score	Support
High	0.3	0.23	0.26	31
Low	0.41	0.43	0.42	30
Medium	0.33	0.38	0.36	39
Accuracy			0.35	100
Macro avg	0.35	0.35	0.35	100
Weighted avg	0.35	0.35	0.35	100

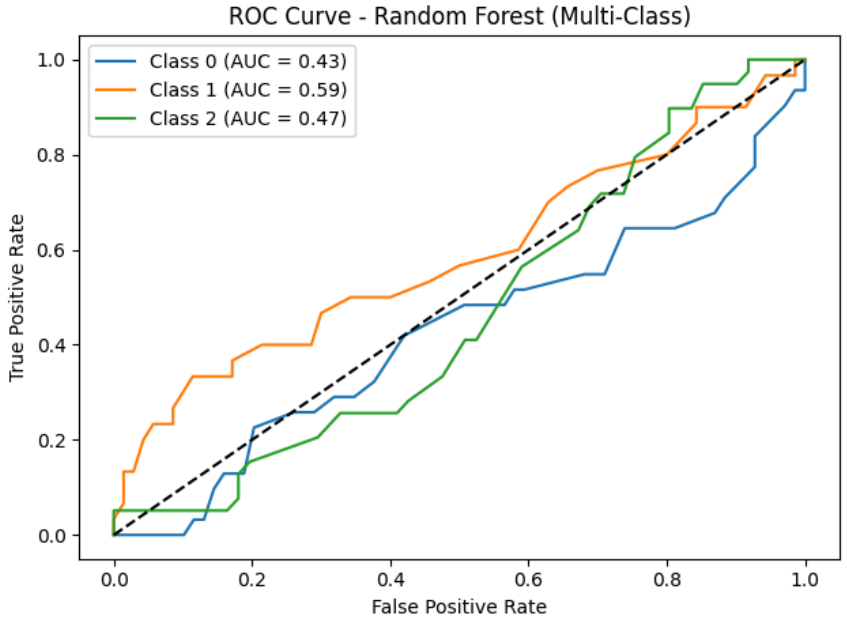


This classification report shows that the Decision Tree model performs best on the Low Style Level class with a precision 0.41 and recall 0.43. The Medium and High classes have relatively low performance with recall 0.38 and 0.23. The AUC values are 0.6 for Class 1, 0.57 for Class 2 and a much weaker 0.41 for Class 1. So the model accuracy is only 35%,

which indicates limited effectiveness for multi-class prediction in this dataset.

- Random Forest Evaluation:
  - Accuracy score: 0.3
  - Classification report:

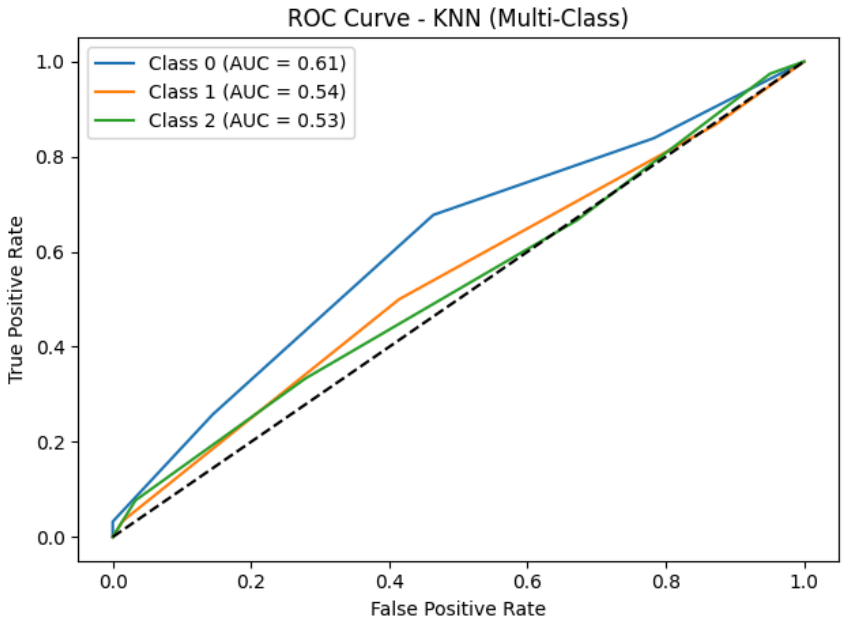
Class	Precision	Recall	F1-score	Support
High	0.05	0.03	0.04	31
Low	0.4	0.27	0.32	30
Medium	0.34	0.54	0.42	39
Accuracy			0.3	100
Macro avg	0.27	0.28	0.26	100
Weighted avg	0.27	0.3	0.27	100



The Random Forest classifier achieved an accuracy score of 0.3. The classification performance is weak across all classes.

- KNN Evaluation:
  - Accuracy: 0.39
  - Classification report:

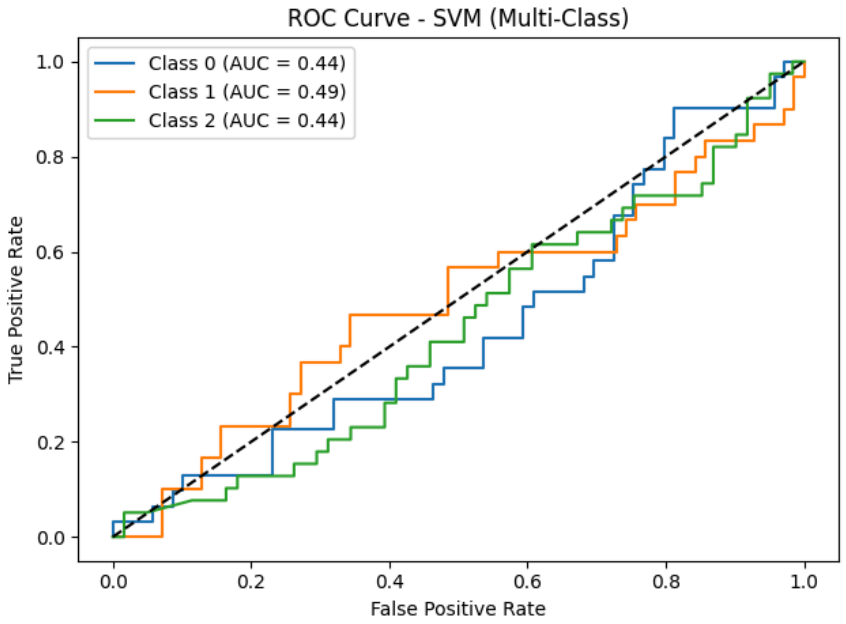
Class	Precision	Recall	F1-score	Support
High	0.39	0.55	0.45	31
Low	0.35	0.3	0.32	30
Medium	0.43	0.33	0.38	39
Accuracy			0.39	100
Macro avg	0.39	0.39	0.38	100
Weighted avg	0.39	0.39	0.38	100



This classification report shows the accuracy 0.39. The classification performance remains weak across all classes.

- SVM Evaluation:
  - Accuracy score: 0.43
  - Classification report:

Class	Precision	Recall	F1-score	Support
High	0.38	0.1	0.15	31
Low	0.75	0.1	0.18	30
Medium	0.42	0.95	0.58	39
Accuracy			0.43	100
Macro avg	0.52	0.38	0.3	100
Weighted avg	0.51	0.43	0.33	100

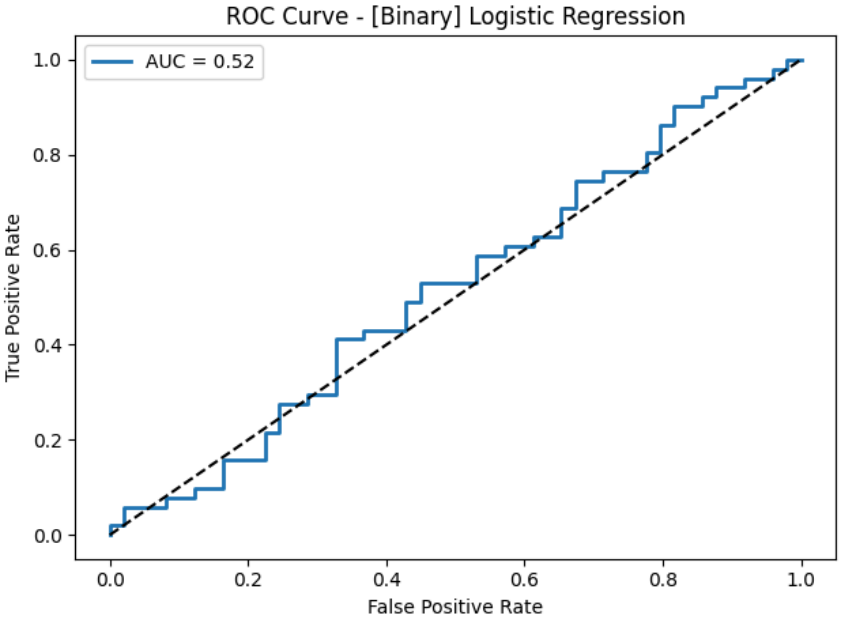


This one got the highest accuracy among multi-class models. SVM performs best on the Medium class recall 0.95 but fails to capture Low and High classes effectively. AUC values remain low and near chance level for all three classes. While SVM achieves the highest accuracy among multi-class models, the imbalance across classes suggests it primarily fits the domain class.

Now here are 5 models' training results for binary classification:

- Logistic Regression Evaluation:
  - Accuracy score: 0.51
  - Classification report:

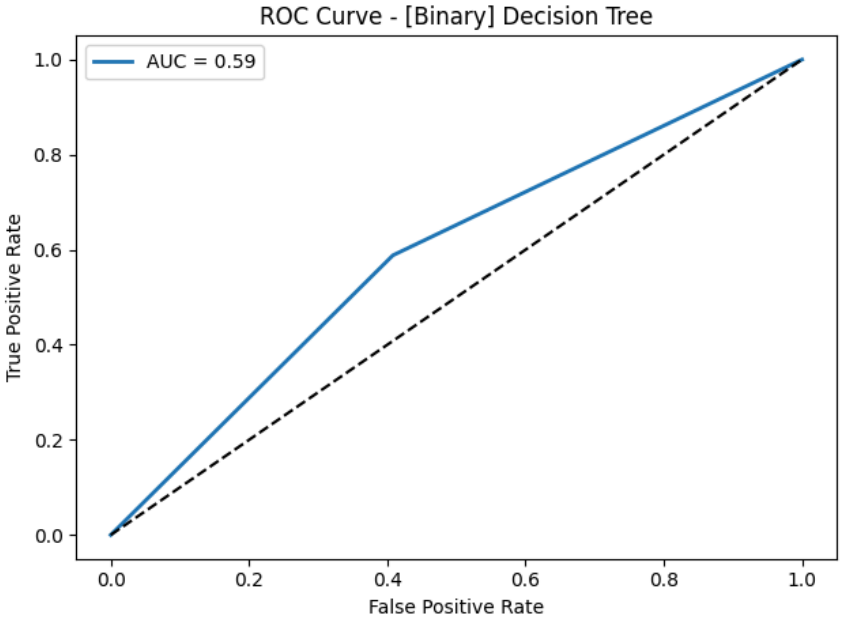
Class	Precision	Recall	F1-score	Support
High	0.5	0.41	0.45	49
Low	0.75	0.61	0.56	51
Accuracy			0.51	100
Macro avg	0.51	0.51	0.5	100
Weighted avg	0.51	0.51	0.51	100



This logic regression model for binary classification showed an accuracy 0.51 which is slightly better than most of the multi-class results. However the AUC score 0.52 shows that the overall separability between the two classes is still weak.

- Decision Tree Evaluation:
  - Accuracy score: 0.59
  - Classification report:

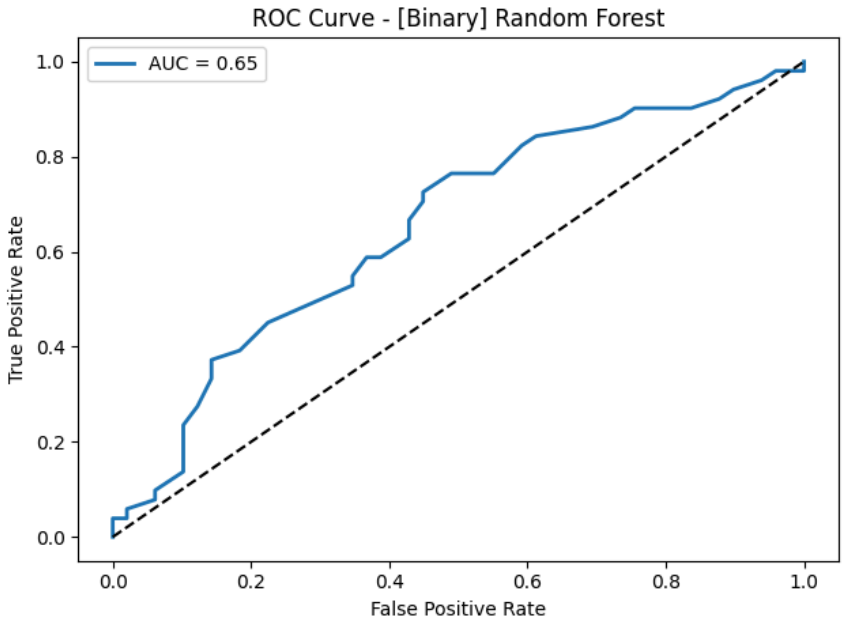
Class	Precision	Recall	F1-score	Support
High	0.58	0.59	0.59	49
Low	0.6	0.59	0.59	51
Accuracy			0.59	100
Macro avg	0.59	0.59	0.59	100
Weighted avg	0.59	0.59	0.59	100



The Decision Tree model in the binary classification achieved an accuracy of 59% which shows a balanced precision and recall across both classes. The ROC curve also shows improved separability compared to the multi-class version.

- Random Forest Evaluation:
  - Accuracy score: 0.62
  - Classification report:

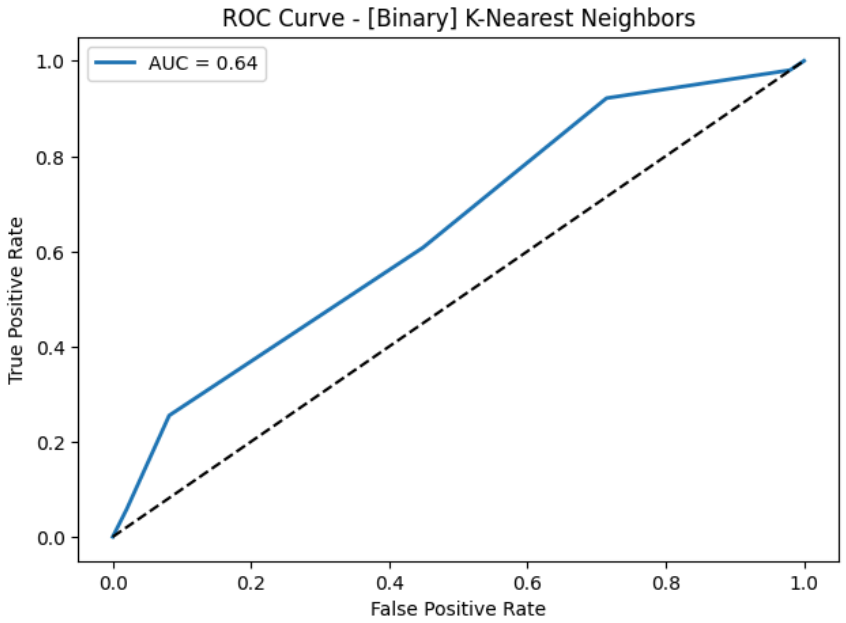
Class	Precision	Recall	F1-score	Support
High	0.62	0.57	0.6	49
Low	0.62	0.67	0.64	51
Accuracy			0.62	100
Macro avg	0.62	0.62	0.62	100
Weighted avg	0.62	0.62	0.62	100



The binary Random Forest model achieved an accuracy of 0.62 with a balanced precision and recall across both classes. It outperformed the previous models slightly with an AUC 0.65 which shows a moderate ability to distinguish between high and low style quality images.

- KNN Evaluation:
  - Accuracy score: 0.58
  - Classification report:

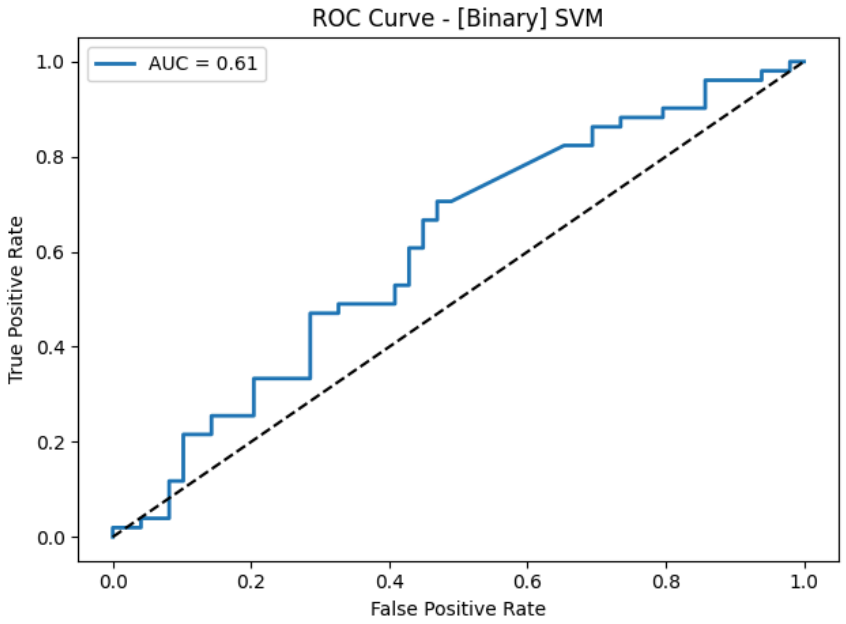
Class	Precision	Recall	F1-score	Support
High	0.57	0.55	0.56	49
Low	0.58	0.61	0.6	51
Accuracy			0.58	100
Macro avg	0.58	0.58	0.58	100
Weighted avg	0.58	0.58	0.58	100



The KNN model achieved an accuracy of 0.58. The AUC score of 0.64 shows moderate classification performance and a decent improvement compared to the multi-class results.

- SVM Evaluation
  - Accuracy score: 0.59
  - Classification report:

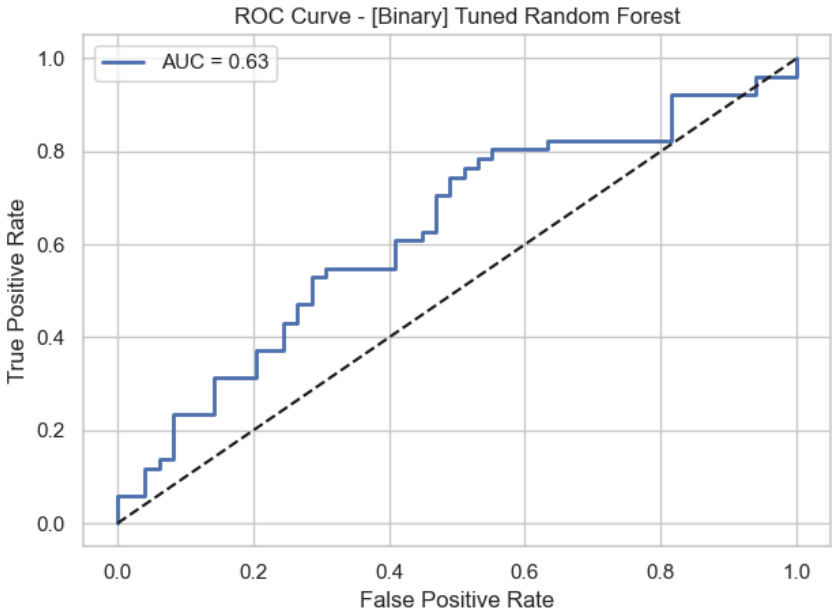
Class	Precision	Recall	F1-score	Support
High	0.59	0.55	0.57	49
Low	0.59	0.63	0.61	51
Accuracy			0.59	100
Macro avg	0.59	0.59	0.59	100
Weighted avg	0.59	0.59	0.59	100



The binary SVM model achieved an accuracy of 0.59 with balanced precision and recall across both classes. The AUC score 0.61 shows moderate separability between the high and low style levels.

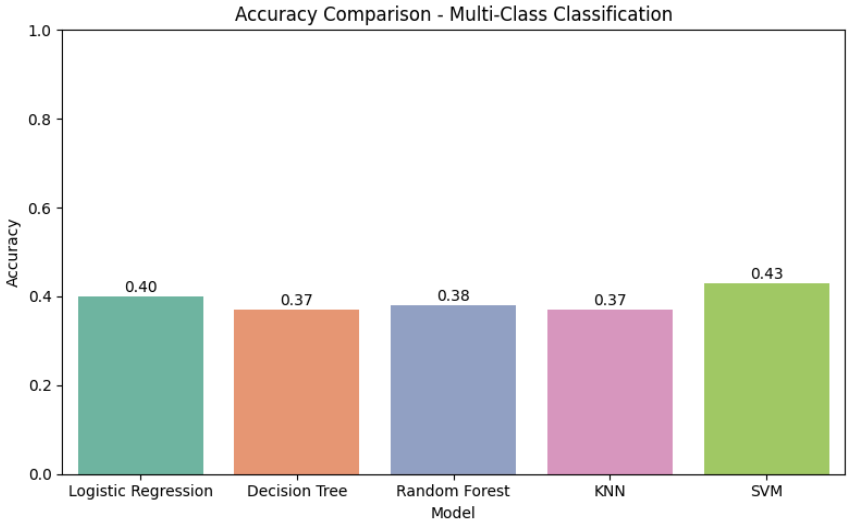
- Tuned Random Forest Evaluation:
  - Accuracy score: 0.59
  - Classification report:

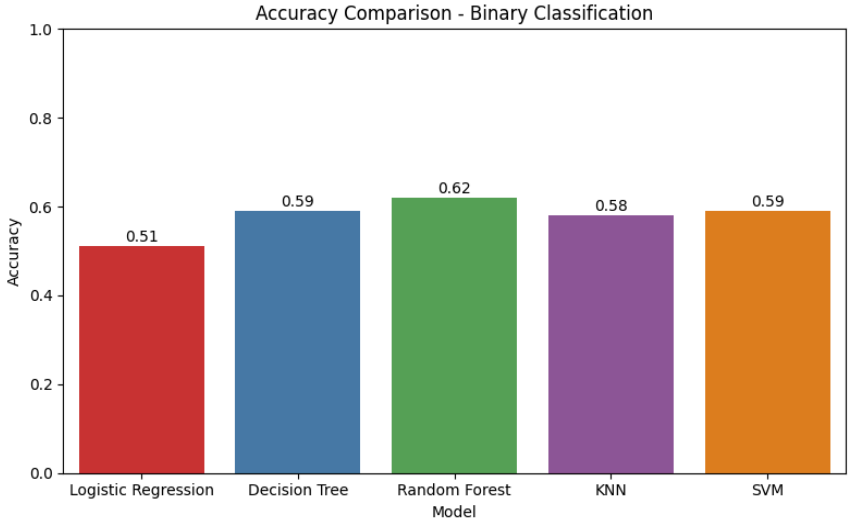
Class	Precision	Recall	F1-score	Support
High	0.59	0.53	0.56	49
Low	0.59	0.65	0.62	51
Accuracy			0.59	100
Macro avg	0.59	0.59	0.59	100
Weighted avg	0.59	0.59	0.59	100

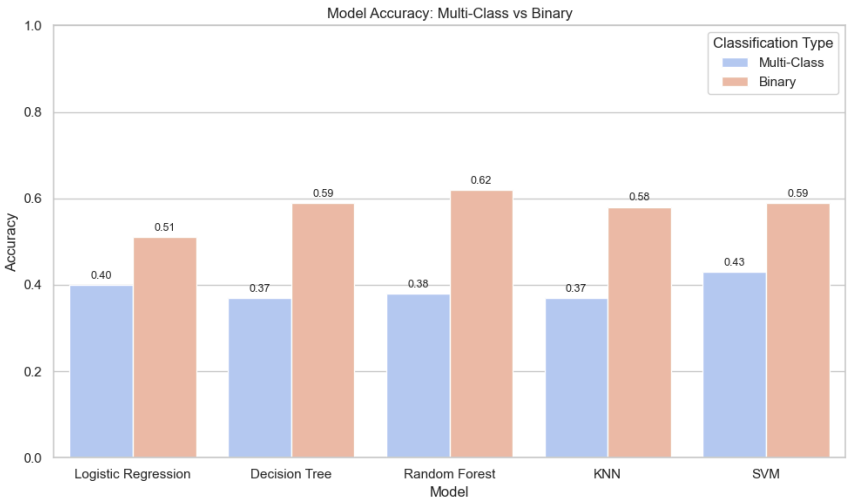


The tuned Random Forest showed similar performance to the default model. It got lower accuracy than the default one.

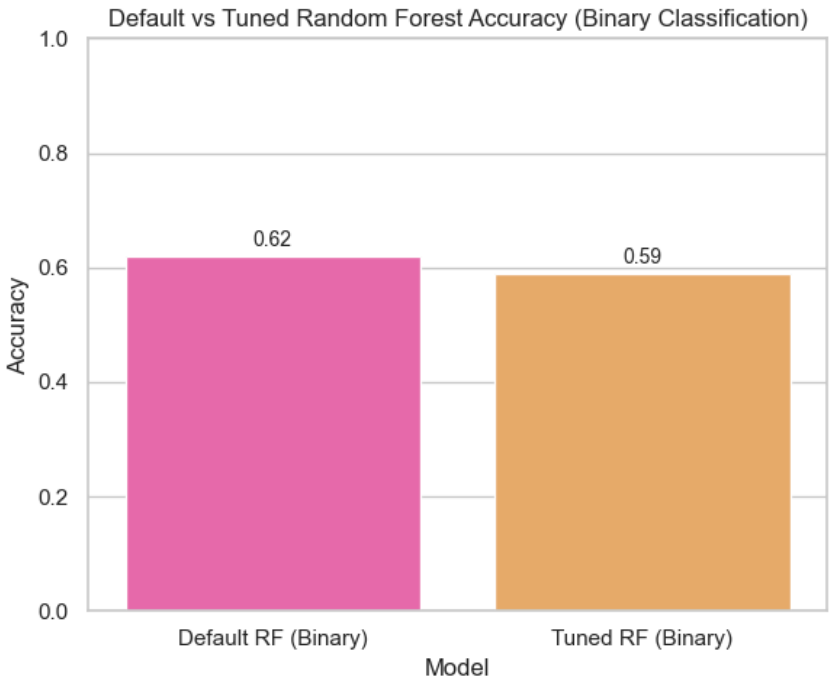
- **Comparing the Models:**
  Here are the Multi-Class Model chart, the Binary Classification chart, and the comparison between these two classifications.







According to these charts, the comparison charts show that the highest multi-class accuracy was only 0.43 (SVM), while the best binary accuracy was 0.62 (Random Forest). This shows that simplifying the target variable into two categories improves model performance and separability across the board.



For Random Forest, we also compared the default and tuned versions. The tuned version did not improve the performance and got an accuracy lower than the default one.

- **Conclusion:**
  According to the findings, we found that the binary classification got higher accuracy than the multi-class setup. Among all models, Random Forest (default binary) provided the best accuracy. This shows that reducing class complexity can help models distinguish Ghibli-style accuracy more effectively.

3. Dataset Overview - Natthiya Sae Ngow

3.1 Dataset 3: Average Daily Screen Time for Children

- **Dataset Introduction :**
  This dataset contains 198 observations and 6 variables. It captures real-world patterns in children's screen time usage across ages 5 to 15, including educational, recreational, and total screen time. Variables also account for gender (Male, Female, Other/Prefer not to say) and day type (Weekday, Weekend). The data reflects trends such as increasing screen time with age, higher usage on weekends, dominance of recreational

screen time, and slight gender difference. Sample size varies by age, from 500 responses at age 5 down to 300 at age 15, adding to the dataset's realism.

- **Data Pre-processing**:
  We began by cleaning the data, checking for missing values, outliers, and duplicates. The dataset was found to be complete, with no missing values or duplicate entries. We then conducted exploratory data analysis (EDA) to understand the structure and characteristics of the results are summarized as follows;
  - **Data Preview:**
    Displays the first few rows to preview the data structure, column names, and sample values, as shown in the following tables;

Dataset Head:

	Age	Gender	Screen Time Type	Day Type	Average Screen Time (hours)	Sample Size
0	5	Male	Educational	Weekday	0.44	500
1	5	Male	Recreational	Weekday	1.11	500
2	5	Male	Total	Weekday	1.55	500
3	5	Male	Educational	Weekend	0.5	500
4	5	Male	Recreational	Weekend	1.44	500

Data shape :(198, 6)

Dataset size is 198 entries, 6 variables including Age, Gender, Screen Time Type, Day Type , Average Screen Time (hours) and Sample Size.

Dataset Type:
RangeIndex: 198 entries, 0 to 197
Data columns (total 6 columns)

	Column	Non-Null Count	Dtype
0	Age	198 non-null	int64
1	Gender	198 non-null	object
2	Screen Time Type	198 non-null	object
3	Day Type	198 non-null	object
4	Average Screen Time(hours)	198 non-null	float64
5	Sample Size	198 non-null	int64

Gender	Count
Male	66
Female	66
Other/Prefer not to say	66

Dataset Describe:

The table below describes the distribution of categorical variables in the dataset (Gender, Screen Time Type, and Day Type):

	Gender	Screen Time Type	Day Type
Count	198	198	198
Unique	3	3	2
Most Frequent Category	Male	Educational	Weekday
Frequency	66	66	99

This table shows each of these features has no missing values (198 entries). The dataset is perfectly balanced across all categories for both Gender and Screen Time Type, each containing 66 samples for 3 categories. Additionally, Day Type is evenly split between Weekday and Weekend, with 99 samples each. This uniform distribution ensures that no category is overrepresented, reducing the risk of bias during model training.

Missing Value:

Age	0
Gender	0
Screen Time Type	0
Day Type	0
Average Screen Time(hours)	0
Sample Size	0

Descriptive Statistics and Distribution:

This table provides a summary of numerical variables.
(Age, Average Screen Time (hours, and Sample Size):

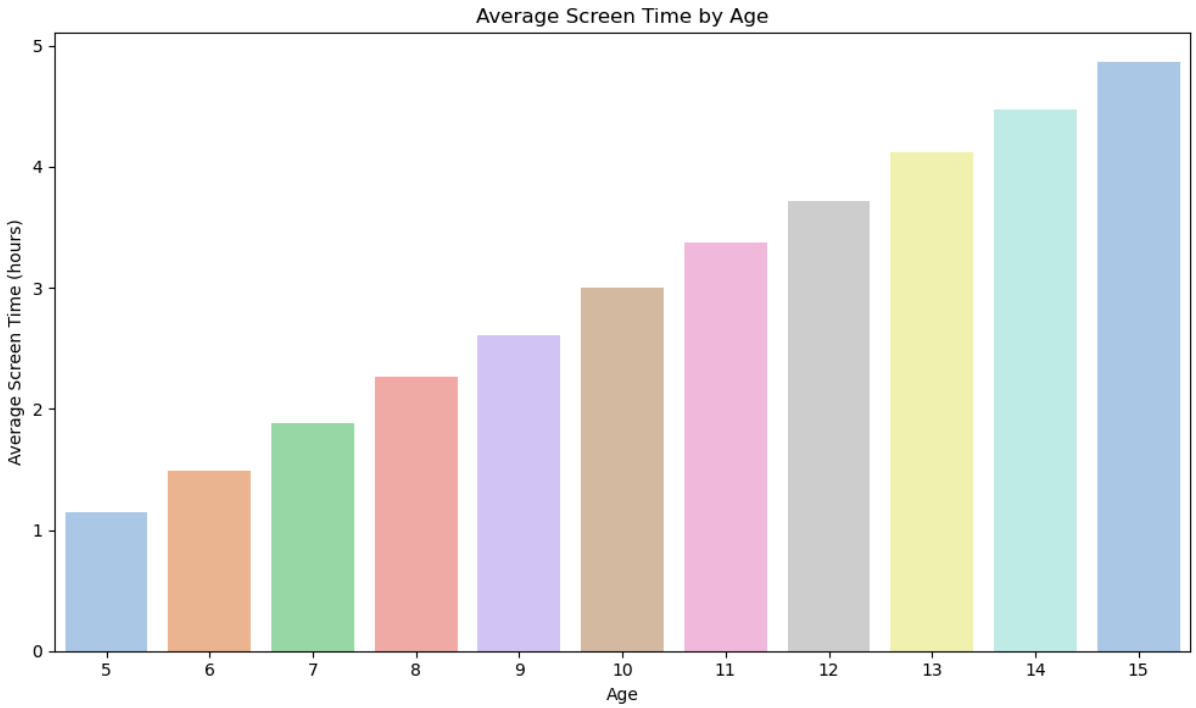
Variable	Mean	Std Dev	Min	25%	50% (Median)	75%	Max
Age	10	3.17	5	7	10	13	15
Average Screen Time (hrs)	2.99	1.92	0.44	1.4	2.49	4.4	8.19
Sample Size	400	63.41	300	340	400	460	500

This table shows that Age is distributed from 5 (Min) to 15 (Max) with a mean and median of 10 that confirms a symmetrical age range. Average Screen Time shows a wide range from 0.44 (Min) to 8.19 (Max) hours with median 2.49 slightly below the mean 2.99, indicating a slight right skewed distribution. In

addition, Sample Size decreases with age that more young participants, which aligns with realistic sampling in child studies.

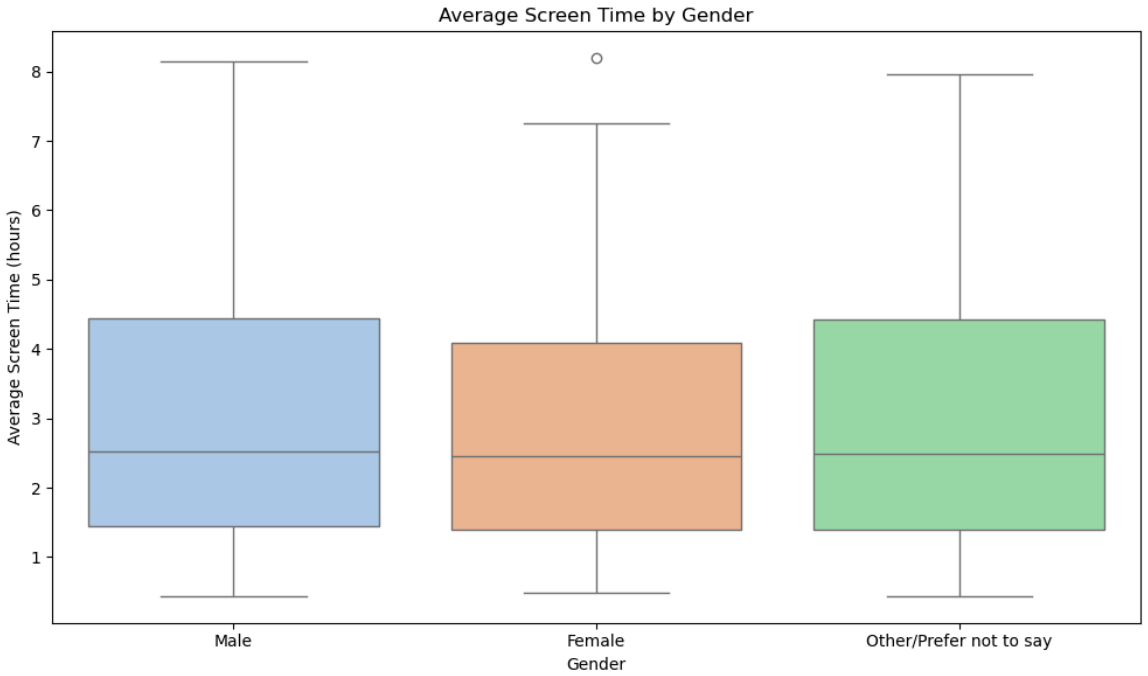
Data Visualizations:

1. Average Screen Time by Age (Bar Plot)



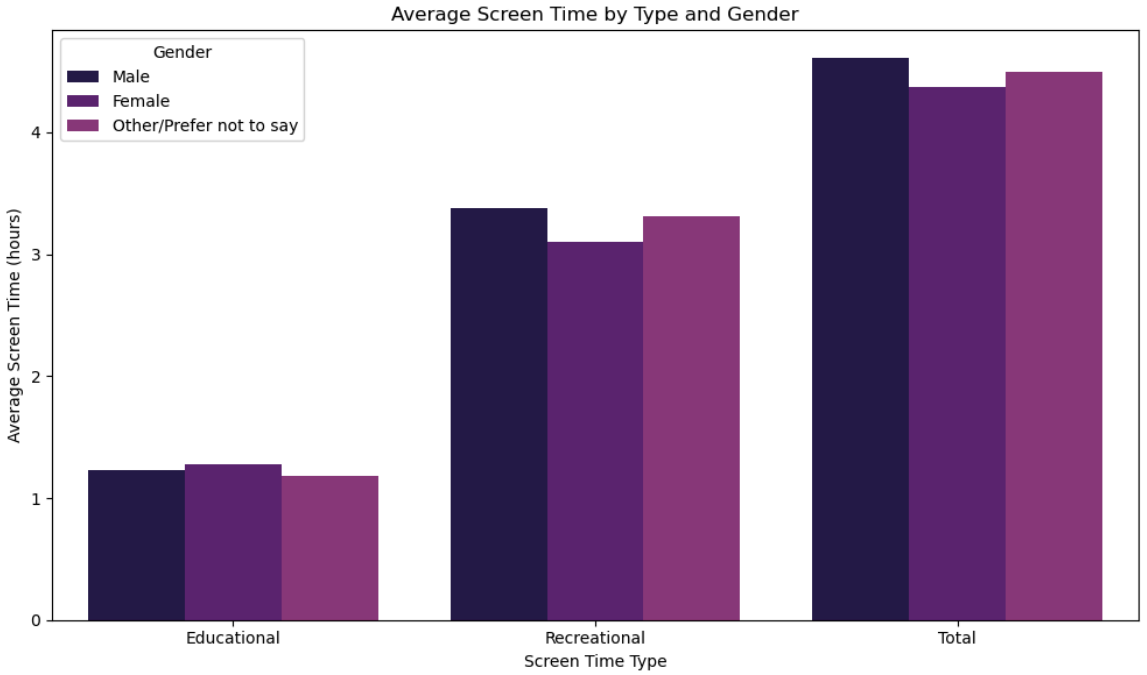
The bar plot shows that children between the ages of 5 and 15 spend approximately 1 to nearly 5 hours on screen time per day. The trend indicates a steady increase in average screen time as children grow older, with a more noticeable rise between ages 10 to 15. By age 15, screen time reaches nearly 5 hours, reflecting a significant increase in usage during adolescence.

2. Average Screen Time by Gender (Boxplot)



This box plot displays the distribution of average screen time across three gender categories: Male, Female, and Other/Prefer not to say. All groups show a similar median screen time of approximately 2.5 hours per day, with the Male and Other/Prefer not to say groups exhibiting slightly wider ranges than Females. Additionally, all categories include high-end outliers, indicating that a few individuals recorded screen time as high as 7 to 8 hours per day.

3. Average Screen Time by Type and Gender (Barplot)



This bar plot compares the average screen time across three types: Educational, Recreational and Total, and for each gender category Male, Female, and Other/Prefer not to say. The chart shows similar screen time patterns across all groups, with Males having the highest total screen time, reaching nearly 4.6 hours per day, representing the most prominent usage trend.

- **Data Preparation and Encoding :**

1. **Class Balance Check**

We examined the distribution of the target variable : High Screen Time, using a class balance check. This variable was derived by comparing each row's average screen time to a 3-hour threshold, resulting in two binary classes:

0 = Children with screen time ≤ 3 hours → "Low Screen Time"

1 = Children with screen time > 3 hours → "High Screen Time"

The output shows that **115 entries** fall into the low screen time group (class 0), while **83 entries** are classified as high screen time (class 1). This distribution indicates a slight class imbalance but remains suitable for binary classification tasks.)

Class Balance:

High Screen Time	
0	115
1	83

2. **Label Encoding of Categorical Variable and Prepare training data**

We convert categorical features : Gender, Screen Time Type and Day Type into numeric value by using LabelEncoder().

```
# Encode categorical variables
le = LabelEncoder()
s_time["Gender"] = le.fit_transform(s_time["Gender"])
s_time["Screen Time Type"] = le.fit_transform(s_time["Screen Time Type"])
s_time["Day Type"] = le.fit_transform(s_time["Day Type"])
print(s_time.head())
```

The print(s_time.head()) line shows the first five rows of the transformed dataset with encoded values.

Output : s_time.head()

	Age	Gender	Screen Time Type	Day Type	Average Screen Time (hours)	Sample Size	High Screen Time
0	5	1	0	0	0.44	500	0
1	5	1	1	0	1.11	500	0
2	5	1	2	0	1.55	500	0
3	5	1	0	1	0.5	500	0
4	5	1	1	1	1.4	500	0

To prepare the training data set x as defined as the input features for model training , while y was set as a target label (High Screen Time). The drop column Average Screen Time (hours) is excluded to prevent target leakage, since it's directly related to the label being predicted.

- **Experiments and Learning the Models**

1. **Train Test Split and Feature Scaling with StandardScaler**
   The dataset is split into 80% for training the model (X_train, y_train) and 20% for testing its performance (X_test, y_test), with random_state=42 set to ensure reproducibility by keeping the split consistent across runs. Feature scaling is applied using StandardScaler to standardize the features, transforming them to have zero mean and unit variance, which helps improve model performance and stability.

2. **Model Definition, Tuning, and Evaluation**
   We defined four machine learning models : **Logistic Regression**, **Gradient Boosting**, **AdaBoost**, and **Decision Tree**, and each paired with a grid of hyperparameter values for tuning. To keep the process efficient, the grid size was kept small to medium.

```
# === Model Definition & Tuning === #
# Define models and parameters
models = {
    "Logistic Regression": (LogisticRegression(), {'C': [0.01, 0.1, 1, 10]}),
    "Gradient Boosting": (GradientBoostingClassifier(), {
        'n_estimators': [50, 100], 'learning_rate': [0.05, 0.1], 'max_depth': [2, 3]
    }),
    "AdaBoost": (AdaBoostClassifier(), {
        'n_estimators': [50, 100], 'learning_rate': [0.5, 1.0]
    }),
    "Decision Tree": (DecisionTreeClassifier(), {
        'max_depth': [2, 4, 6, 8], 'criterion': ['gini', 'entropy']
    })
}
```

Hyperparameter tuning was performed using **GridSearchCV** with **5-fold cross-validation** and evaluated using **Accuracy**, **F1 Score**, **Precision**, **Recall**, and **ROC AUC**, enabling optimization of each model's key parameters. The final performance metrics are summarized in the table below.

Final Model Results;

Model	Best Params	Accuracy	F1 Score	Precision	Recall	ROC AUC
Logistic Regression	{'C': 1}	0.95	0.933333	0.933333	0.933333	0.986667
Gradient Boosting	{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}	0.925	0.896552	0.928571	0.866667	0.98
AdaBoost	{'learning_rate': 1.0, 'n_estimators': 100}	0.925	0.896552	0.928571	0.866667	0.98
Decision Tree	{'criterion': 'entropy', 'max_depth': 6}	0.925	0.896552	0.928571	0.866667	0.913333

Feature importance was performed by fitting 4 trained machine learning models **Logistic Regression**, **Gradient Boosting**, **AdaBoost**, and **Decision Tree** with the best-known hyperparameters. Each model is trained using the option found through GridSearchCV.

```
# === Feature Importances === #
# Train models with best-known parameters
log_reg = LogisticRegression(C=1).fit(X_train_scaled, y_train)
gb = GradientBoostingClassifier(learning_rate=0.1, max_depth=3, n_estimators=100).fit(X_train_scaled, y_train)
ada = AdaBoostClassifier(learning_rate=1.0, n_estimators=100).fit(X_train_scaled, y_train)
tree = DecisionTreeClassifier(criterion='entropy', max_depth=8).fit(X_train_scaled, y_train)
```

To compare which features were most influential in each model, a feature importance table was created using a pandas DataFrame (importance_df). This table includes the feature names, coefficients from Logistic Regression, and feature importance scores from Gradient Boosting, AdaBoost, and Decision Tree models. Feature importance was extracted using:

- log_reg.coef_[0] for Logistic Regression
- .feature_importances_ for tree-based models (Gradient Boosting, AdaBoost, Decision Tree)

The final performance feature importance is summarized in the table below.

Feature Importance Results :

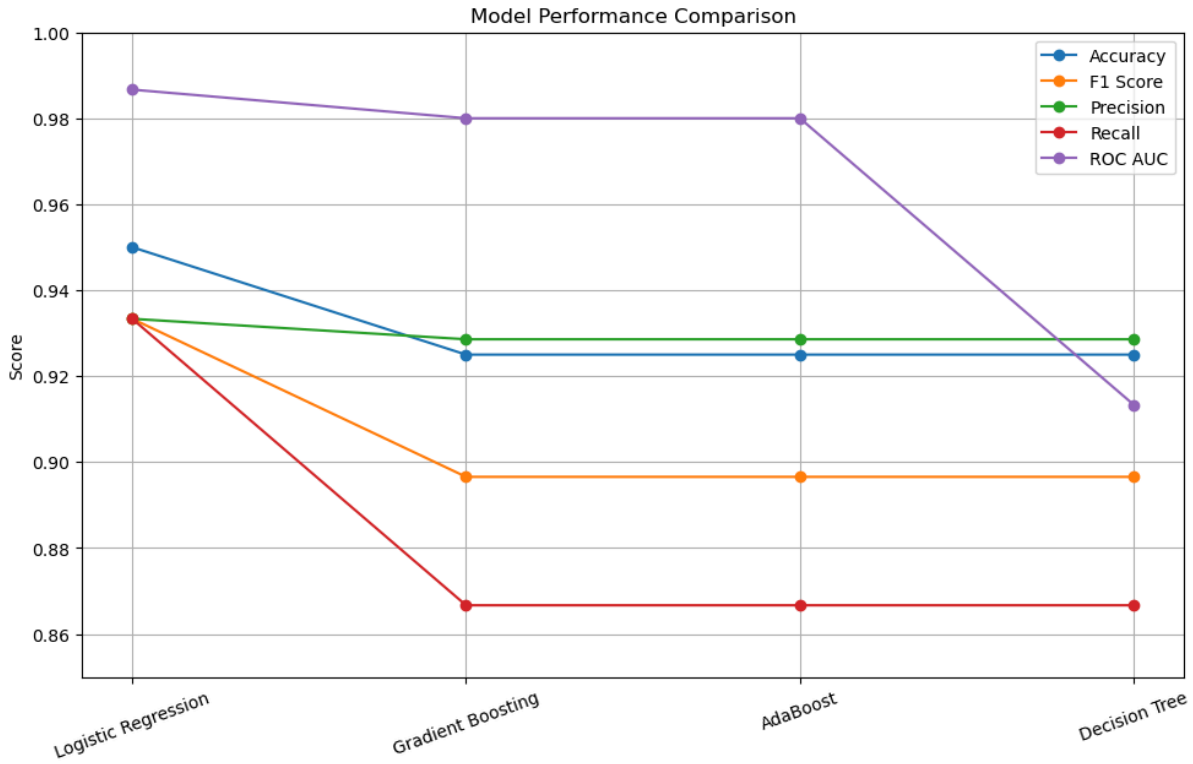
Column	Feature	Logistic Coef	Gradient Boosting	AdaBoost	Decision Tree
1	Age	1.495776	0.268391	0.199966	0.015985
2	Gender	0.136149	0.002416	0.01239	0.017684
3	Screen Time Type	3.169709	0.43041	0.481278	0.419831
4	Day Type	0.445351	0.025279	0.065994	0.074941
5	Sample Soze	-1.495776	0.273504	0.240372	0.471559

- **Comparing the Models:**

Based on the **Final Model Results**, **Logistic Regression** is the best performing model overall. It demonstrated strong performance across all evaluation metrics, with an Accuracy of 95%, F1 Score of 93.33%, Precision of

93.33%, Recall of 93.33%, and a ROC AUC of 98.67%. Its high AUC indicates excellent classification capability across probability thresholds.
Both Gradient Boosting and AdaBoost achieved identical scores, reflecting solid overall performance; however, their lower Recall (86.67%) compared to Logistic Regression suggests a higher rate of false negatives.
While the Decision Tree matched the ensemble models in most metrics, its lower ROC AUC of 91.33% indicates reduced reliability in ranking class probabilities relative to the other models.

The line graph displays various performance metrics of Accuracy, F1 Score, Precision, Recall, and ROC AUC to compare the relative changes and consistency in performance across different models.



Based on the **Feature Importance Results ;**

1. **The Logistic Regression model** identified the most influential features as **Screen Time Type** (+3.17), **Sample Size** (–1.50), and **Age** (+1.50). This indicates that **Screen Time Type** has the strongest positive association with high screen time, while **Sample Size** has a strong negative coefficient , suggesting that older age groups with smaller sample sizes tend to exhibit higher screen

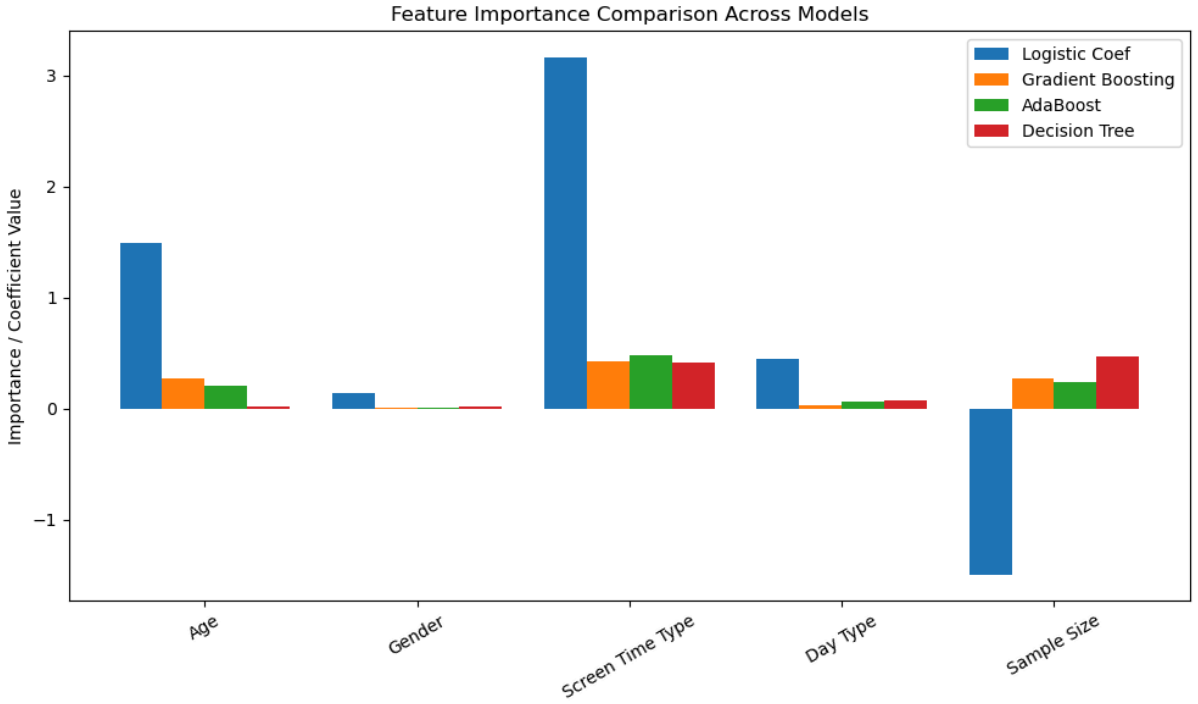
usage. In contrast, **Gender** and **Day Type** show relatively minimal influence.

2. **The Gradient Boosting model** identified **Screen Time Type** (0.43), **Sample Size** (0.27), and **Age** (0.27) as the top contributing features. This indicates that **Screen Time Type** is again the dominant predictor, while **Age** and **Sample Size** **Contribute** moderately. **Gender** and **Day Type** have minimal impact on the model's predictions.

3. **The AdaBoost model** identified **Screen Time Type** (0.48), **Sample Size** (0.24), and **Age** (0.20) as the most important features. Similar to the Gradient Boosting model, **Screen Time Type** is the dominant predictor, while **Age** and **Sample Size** play secondary but relevant roles. **Gender** remains negligible in influence.

4. **The Decision Tree model** identified **Sample Size** (0.47) and **Screen Time Type** (0.42) as the top contributing features. The model relies heavily on these two predictors, while **Age** and **Day Type** play minor roles. **Gender** has almost no influence on the model's decisions.

The bar plot compares **f**eature importance across four machine learning models: Logistic Regression, Gradient Boosting, AdaBoost, and Decision Tree.



- **Conclusion:**

  The dataset reveals screen time behavior that aligns with expected patterns related to Age usage. **The Logistic Regression Model** achieved the highest overall performance with the best balance of accuracy and interpretability. In terms of feature importance, **Screen Time Type** is the most influential predictor across all models , followed by Sample size and Age as important secondary features. In contrast, Gender and Day Type showed very low predictive power, indicating that how and why screen time is used matters more than who is using it.

3.2 Dataset 4 : Cost of International Education

- **Dataset Introduction :**

This dataset contains 907 observations and 12 variables, compiled detailed financial information for students pursuing higher education abroad. It spans multiple countries, cities and universities around the world. capturing a comprehensive spectrum of tuition fees, living expenses and key ancillary costs. Standardized fields include tuition in USD, living cost indices, rent, visa fees, insurance, and up-to-date exchange rates. Currency values were derived based on local currency units per U.S dollar at the time of data collection.

- **Data Pre-processing**:

We started cleaning the dataset, checking for missing values, outliers, and duplicates. The data was found to be complete, with no missing or duplicate entries. Exploratory Data Analysis (EDA) was then performed to better understand the structure and characteristics of the dataset. The key findings are summarized as follows:

Dataset Shape: (907, 12)

Feature 12 variables :

Feature	Description
Country	Country where the university is located (e.g., USA, UK, Canada)
City	City of the university
University	Name of the institution
Program	Program or field of study (e.g., Data Science, Engineering)
Level	Degree level (Undergarde ,Master and PhD)
Duration_Years	Duration of the program in years (1 year to 5 years)
Tuition_USD	Tuition fee in USD for the entire program ($0 to $62,000)
Living_Cost_Index	Relative cost of living index (27.8 to 122 ,higher = more expensive)
Rent_USD	Monthly rent in USD ($150 to $2,500)
Visa_Fee_USD	Visa processing fee for that country ($40 to $490)
Insurance_USD	Health or travel insurance fee for the student ($200 to $1,500)
Exchange_Rate	Exchange rate to USD (1 = USD-based country)

Dataset size is 907 entries, 12 variables including Country, City, University, Program, Level, Duration_Years, Tuition_USD, Living_Cost_Index, Rent_USD, Visa_Fee_USD, Insurance_USD, Exchange_Rate.

Dataset Head :

Country	City	University	Program	Level	Duration_Years
USA	Cambridge	Harvard University	Computer Science	Master	2
UK	London	Imperial College London	Data Science	Master	1
Canada	Toronto	University of Toronto	Business Analytics	Master	2
Australia	Melbourne	University of Melbourne	Engineering	Master	2
Germany	Munich	Technical University of Munich	Mechanical Engineering	Master	2

Country	Tuition_USD	Living_Cost_Index	Rent_USD	Visa_Fee_USD	Insurance_USD	Exchange_Rate
USA	$ 55,400	83.5	$ 2,200	$ 160	$ 1,500	1
UK	$ 41,200	75.8	$ 1,800	$ 485	$ 800	0.79
Canada	$ 38,500	72.5	$ 1,600	$ 235	$ 900	1.35
Australia	$ 42,000	71.2	$ 1,400	$ 450	$ 650	1.52
Germany	$ 500	70.5	$ 1,100	$ 75	$ 550	0.92

This dataset head table displays a sample from the International Education Cost dataset, highlighting financial and program-related details from five global universities. The table shows that the USA (Harvard) has the highest tuition fees at $55,400 for a master's degree, along with a high monthly rent of $2,200, indicating a substantial total cost of attendance. In contrast, Germany (Munich) offers exceptionally low tuition at $500 and a moderate rent of $1,100, suggesting relatively affordable living expenses. The Living Cost Index is highest for the USA and UK, reinforcing their reputation as expensive education environments. Additionally, programs such as Engineering and Computer Science appear frequently, reflecting the strong representation of technical fields in the dataset.

Dataset Info:
RangeIndex: 907 entries, 0 to 906
Data columns (total 12 columns):

Column	Non-Null Count	Dtype
Country	907 non-null	object
City	907 non-null	object
University	907 non-null	object
Program	907 non-null	object
Level	907 non-null	object
Duration_Years	907 non-null	float64
Tuition_USD	907 non-null	int64
Living_Cost_Index	907 non-null	float64
Rent_USD	907 non-null	int64
Visa_Fee_USD	907 non-null	int64
Insurance_USD	907 non-null	int64
Exchange_Rate	907 non-null	float64

Missing Values :

Column	Missing Values
Country	0
City	0
University	0
Program	0
Level	0
Duration_Years	0
Tuition_USD	0
Living_Cost_Index	0
Rent_USD	0
Visa_Fee_USD	0
Insurance_USD	0
Exchange_Rate	0
Total_Cost_USD	0
HighCost	0

Descriptive Statistics and Distribution:

This table provides a summary of Categorical variables and Numerical variables.

5 Categorical Variables:

(Country, City, University, Program and Level):

Feature	Count	Unique	Most Frequent Value	Frequentcy
Country	907	71	UK	93
City	907	556	Singapore	18
University	907	622	University of Washington	6
Program	907	92	Computer Science	312
Level	907	3	Master	451

This table shows a wide global representation across 71 Countries , 556 Cities, 622 Universities, 92 Program and 3 Levels. The most popular program is Computer Science and the majority of entries are the Master's level programs.

7 Numericals Variables :

(Duration_Year, Tuition_USD, Living_Cost_Index, Rent_USD, Visa_Fee_USD, Insurance_USD and Exchange_Rate)

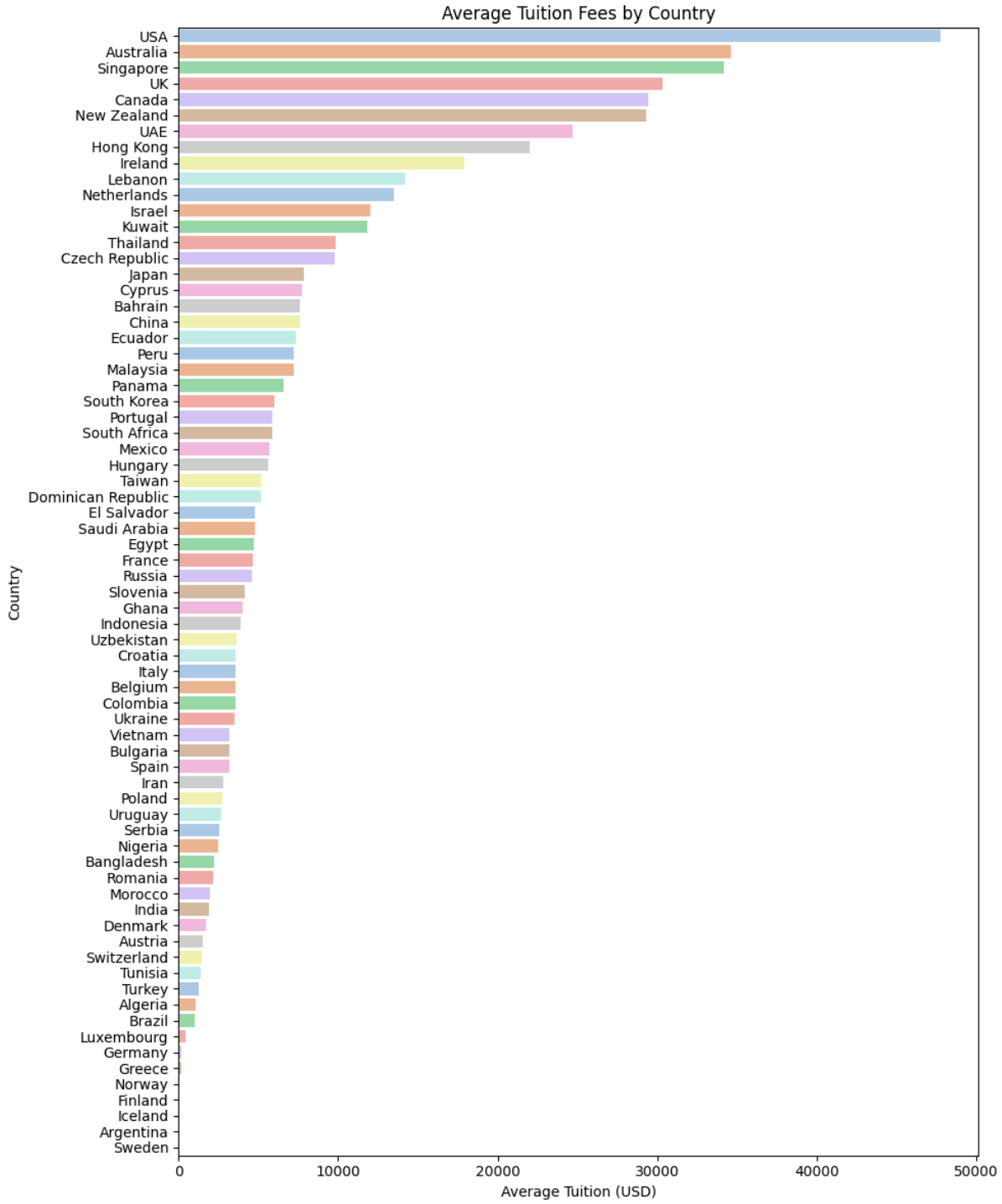
Data distribution:

Column	Count	Mean	Std Dev	Min	25%	Median	75%	Max
Duration_Years	907	2.84	0.95	1.00	2.00	3.00	4.00	5.00
Tuition_USD	907	16,705.00	16,582.00	-	2,850.00	7,500.00	31,100.00	62,000.00
Living_Cost_Index	907	64.44	14.06	27.80	56.30	67.50	72.20	122.40
Rent_USD	907	969.21	517.15	150.00	545.00	900.00	1,300.00	2,500.00
Visa_Fee_USD	907	211.40	143.44	40.00	100.00	160.00	240.00	490.00
Insurance_USD	907	700.08	320.37	200.00	450.00	650.00	800.00	1,500.00
Exchange_Rate	907	623.00	3,801.75	0.15	0.92	1.35	7.15	42,150.00

These statistics summarize the distribution of each numeric feature across all 907 observations. The average program duration is approximately 3 years. Tuition fees range from $0 (likely in public universities with full subsidies) to $62,000, with a mean of around $16,700, indicating a heavily skewed distribution. Living costs also vary significantly which monthly rent ranges from $150 to $2,500, reflecting the diversity of global housing markets. The Living Cost Index peaks at 122.4, likely corresponding to high-cost cities such as New York, London, or Tokyo. Visa and insurance fees also show wide variation: visa costs range from $40 to nearly $500, depending on country specific student visa policies, while insurance fees range from $200 to $1,500, reflecting differing coverage plans and national regulations. The exchange rate variable exhibits a large standard deviation and a maximum value of 42,150, suggesting the presence of countries with hyperinflated currencies. The median exchange rate is approximately 1.35, aligning with rates common in international education markets like Canada and Australia.

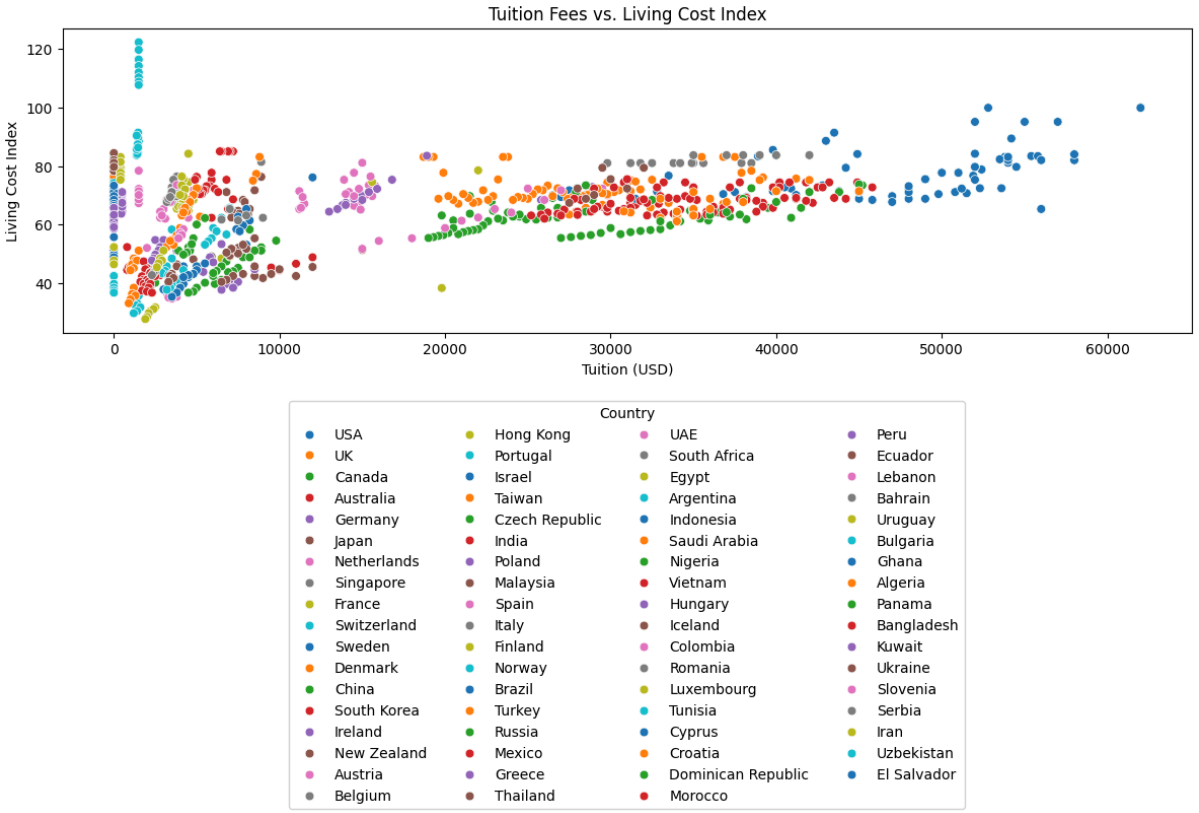
Data Visualizations:

1. Average Tuition Fees by Country (Bar Plot)



This bar plot shows that the USA has the highest average tuition fees, nearing $50,000. Other high-tuition countries include Australia, Singapore, the UK, and Canada. In contrast, Germany, Sweden, Finland, and Argentina appear at the bottom, reflecting low-cost or tuition-free education models.

2. Tuition Fees VS Living Cost Index (Scatter Plot)



The scatter plot shows a positive trend, indicating that higher tuition fees are generally associated with higher living costs. The USA, UK, Singapore, and Australia have many data points in the high-cost region. Clusters below $10,000 in tuition suggest countries offering more affordable education options. Additionally, some data points at tuition = $0 with high living costs imply the presence of fully subsidized programs in otherwise expensive countries, such as Germany or Sweden.

- **Data Preparation and Encoding :**

We created two engineered variables: Total_Cost_USD and HighCost. The Total_Cost_USD feature was calculated to represent the estimated total cost of attending a program, including both tuition and living expenses. The HighCost feature is a binary classification label indicating whether a program is considered high-cost or not.

```
# Create the total cost and binary target variable
cost_int["Total_Cost_USD"] = cost_int["Tuition_USD"] + cost_int["Duration_Years"] * (
    cost_int["Rent_USD"] * 12 + cost_int["Insurance_USD"] + cost_int["Visa_Fee_USD"]
)
cost_int["HighCost"] = (cost_int["Total_Cost_USD"] > cost_int["Total_Cost_USD"].median()).astype(int)
```

Class Balance Check

We examined the distribution of the target variable : High Cost , using a class balance check to label in two binary classes:

0 = a program with total cost $\leq$ the median → "Low total cost"

1 = a program with total cost $>$ 3 hours → "High total cost"

The output shows that **454 entries** fall into the low total cost group (class 0), while **453 entries** are classified as high total cost (class 1). This distribution indicates a class perfect balance for binary classification tasks.

Class Balance:

Class	Count
0	454
1	453

Standard Scale of Numerical Variables and One-hot Encoding of Categorical Variable and Prepare training data

We separated numerical variables : Duration_Year, Tuition_USD, Living_Cost_Index, Rent_USD, Visa_Fee_USD, Insurance_USD and Exchange_Rate by scaled using method StandardScaler(), applies Z-score normalization (mean = 0, std =1)
StandardScaler().

```
# Standard scale numerical
scaler = StandardScaler()
X_num_scaled = pd.DataFrame(scaler.fit_transform(X_num), columns=numerical_cols)
```

We convert categorical variables : Country, City, University, Program and Level) into binary dummy variables using one-hot encoding.

```
# One-hot encode categorical
encoder = OneHotEncoder(handle_unknown='ignore', sparse_output=False)
X_cat_encoded = pd.DataFrame(encoder.fit_transform(X[categorical_cols]))
X_cat_encoded.columns = encoder.get_feature_names_out(categorical_cols)
```

- **Experiments and Learning the Models**

3. **Train Test Split and Feature Scaling with StandardScaler**

   The dataset is split into 80% for training the model (X_train, y_train) and 20% for testing its performance (X_test, y_test), with random_state=42 set to ensure reproducibility by keeping the split consistent across runs.

Feature scaling is applied using StandardScaler to standardize the features, transforming them to have zero mean and unit variance, which helps improve model performance and stability.

4. **Model Definition, Tuning, and Evaluation**
   We defined four machine learning models : **Logistic Regression**, **Random Forest, Support Vector Machine (SVM)** and **Gradient Boosting**, and each paired with a grid of hyperparameter values for tuning. To keep the process efficient, the grid size was kept small to medium.

```
# Step 4: Train 4 models with Grid Search
models = {
    "LogisticRegression": (LogisticRegression(max_iter=1000), {
        "C": [0.01, 0.1, 1, 10]
    }),
    "RandomForest": (RandomForestClassifier(), {
        "n_estimators": [100, 200],
        "max_depth": [5, 10]
    }),
    "SVM": (SVC(), {
        "C": [0.1, 1],
        "kernel": ["linear", "rbf"]
    }),
    "GradientBoosting": (GradientBoostingClassifier(), {
        "n_estimators": [50, 100],
        "learning_rate": [0.05, 0.1], 'max_depth': [2, 3]
    })
}
```

Hyperparameter tuning was performed using **GridSearchCV** with **5-fold cross-validation** and evaluated using **Accuracy**, **F1 Score**, **Precision**, **Recall**, and **ROC AUC**, enabling optimization of each model's key parameters. The final performance metrics are summarized in the table below.

Final Model Results;

Model	Best Params	Accuracy	F1 Score	Precision	Recall	ROC AUC
Logistic Regression	{'C': 10}	0.983516	0.983516	0.983575	0.983516	0.998913
Gradient Boosting	{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}	0.983516	0.983516	0.983575	0.983516	0.99686
SVM	{'C': 1, 'kernel': 'linear'}	0.978022	0.978019	0.978253	0.978022	0.998551
Random Forest	{'max_depth': 10, 'n_estimators': 200}	0.956044	0.956039	0.956264	0.956044	0.994204

Feature importance was performed by fitting 4 trained machine learning models **Logistic Regression**, **Random Forest** , **SVC** and **Gradient**

Boosting with the best-known hyperparameters. Each model is trained using the option found through GridSearchCV.

```
# === Feature Importances === #
# Fit models with best-known parameters
log_reg = LogisticRegression(C=10, max_iter=1000).fit(X_train, y_train)
rf = RandomForestClassifier(max_depth=10, n_estimators=100).fit(X_train, y_train)
svm = SVC(C=1, kernel="linear", probability=True).fit(X_train, y_train)
gb = GradientBoostingClassifier(learning_rate=0.05, max_depth=3, n_estimators=100).fit(X_train, y_train)
```

To compare which features were most influential in each model, a feature importance table was created using a pandas DataFrame (importance_df). This table includes the feature names, coefficients from Logistic Regression and SVM and feature importance scores from Random Forest and Gradient Boosting models. Feature importance was extracted using:

- log_reg.coef_[0] for Logistic Regression
- svm.coef_[0] for SVM
- feature_importances_ for tree-based models (Random Forest and Gradient Boosting)

The final performance feature importance is summarized in the table below.

Feature Importance Results :
Top 10 Features by Importance for Each Model :

1. **Logistic Regression Model**

Feature	Coefficient
Tuition_USD	6.8648
Rent_USD	5.8316
Duration_Years	2.8699
Country_Canada	2.3838
University_University of Amsterdam	-1.922
Level_Master	-1.9093
Country_UAE	1.7303
Program_Artificial Intelligence	-1.4332
Level_PhD	1.3857
University_University of Lille	1.2557

This table shows that **Logistic Regression** identified the most influential features as Tuition_USD (6.8648), Rent_USD(5.8316), and Duration_Years (2.8699). Notable positive contributors include

Country_Canada, Country_UAE, and Level_PhD, while strong negative weights were assigned to University_University of Amsterdam, Level_Master, and Program_Artificial Intelligence. In this model, tuition and rent are the primary drivers, with university and program indicators significantly influencing the linear decision boundary.

2. Random Forest Model

Feature	Coefficient
Tuition_USD	0.1504
Insurance_USD	0.1165
Living_Cost_Index	0.1071
Visa_Fee_USD	0.0837
Exchange_Rate	0.0675
Rent_USD	0.0611
Country_USA	0.0368
Level_Master	0.025
Country_UK	0.0227
Program_Computer Science	0.0217

This table shows that the **Random Forest model** identified the most influential features as Tuition_USD (0.1504), Insurance_USD (0.1165), and Living_Cost_Index (0.1071). Additional important features include Visa_Fee_USD, Exchange_Rate, and Country_USA. This model captures a broader range of cost-related influences, highlighting the significance of various financial factors beyond just tuition and rent.

3. SVM

Feature	Coefficient
Tuition_USD	2.1589
Rent_USD	1.8505
University_University of Amsterdam	-0.9739
Duration_Years	0.8382
Level_Master	-0.8028
Country_UAE	0.7547
Country_Canada	0.7505
Program_Artificial Intelligence	-0.5751
University_University of Neuchatel	-0.5541
City_Neuchatel	-0.5541

This table shows that the **SVM model** identified the most influential features as Tuition_USD (2.1589), Rent_USD(1.8505), and Duration_Years (0.8382). Country-specific factors such as Country_Canada and Country_UAE,

along with program and university indicators, also had a notable impact. The model demonstrates sensitivity to both numerical and binary categorical features, which is consistent with the behavior of margin-based classification in SVMs.

4. **Gradient Boosting**

Feature	Coefficient
Tuition_USD	0.7165
Rent_USD	0.1466
Duration_Years	0.0479
Level_Master	0.031
Insurance_USD	0.0099
Exchange_Rate	0.0088
Level_PhD	0.0074
Country_Lebanon	0.0046
University_University of Amsterdam	0.0046
University_Arabian Gulf University	0.0029

This table shows that the **Gradient Boosting model** identified Tuition_USD (0.7165) and Rent_USD (0.1466) as the most influential features. Other relevant but weaker contributors include Duration_Years, Level_Master, and Exchange_Rate. The model also captures minor influence from low-frequency university and country indicators. Overall, it primarily focuses on tuition and rent while effectively accounting for smaller effects from rare but meaningful features.

- Comparing the Models:

Based on the **Final Model Results**, **Logistic Regression** emerged as the best-performing model overall. It demonstrated consistently strong performance across all evaluation metrics, achieving the **highest ROC AUC of 99.89%**, along with well-balanced **precision, recall, and F1 score of 98.35%**. This model is particularly effective when the data has strong linear relationships.

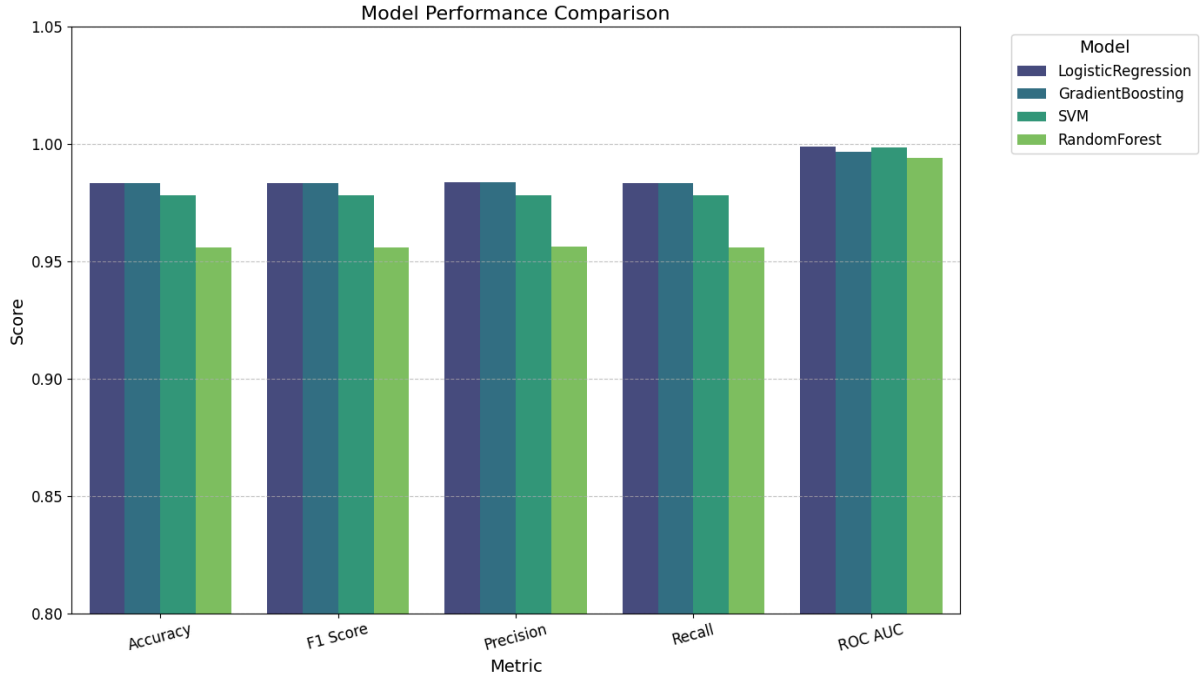
Gradient Boosting closely followed, matching Logistic Regression in all metrics except for a slightly lower ROC AUC of **99.68%**, indicating robust predictive power with added ability to capture non-linear patterns.

Support Vector Machine (SVM) also showed strong performance with an **F1 score of 97.80%** and a high **ROC AUC of 99.86%**, suggesting effective

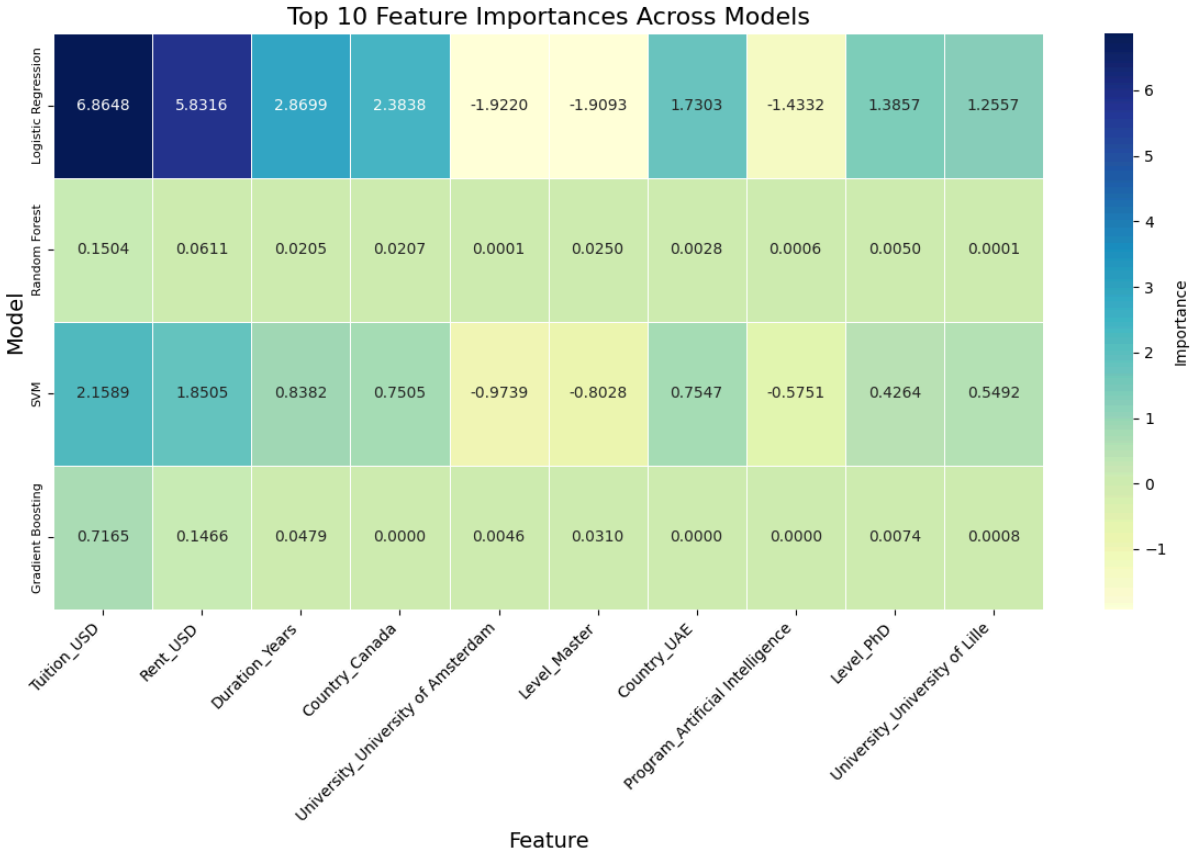
class separation. While slightly behind the top two models, SVM remains a solid performer, though it may be more computationally intensive depending on the kernel and data size.

Random Forest delivered the lowest performance among the four models, though still excellent. It achieved an **accuracy of 95.60%** and a respectable **ROC AUC of 99.42%**, making it a reliable choice, especially for quick baseline comparisons.

The bar plot compares four models across five evaluation metrics: Accuracy, F1 Score, Precision, Recall, and ROC AUC. **Logistic Regression** and **Gradient Boosting** emerge as the most consistent and top-performing models across all metrics. **SVM** also performs competitively, particularly in terms of ROC AUC and overall class separation. While **Random Forest** trails slightly in precision and recall, it still demonstrates strong classification performance, especially with a solid **ROC AUC of 99.4%**.



The heatmap compares feature importance across four models: **Logistic Regression**, **Random Forest**, **Support Vector Machine (SVM)**, and **Gradient Boosting**. The color intensity and numerical values indicate the strength of each feature's influence on the model's predictions.



The heatmap shows that Tuition_USD (6.86) and Rent_USD (5.83) are the most important features across all models, particularly in the **Logistic Regression model**. Both Logistic Regression and SVM assign positive or negative weights to categorical indicators, such as University_University of Amsterdam (−1.922) and Level_Master (−1.90). In contrast, **Random Forest** and **Gradient Boosting** assign minimal or zero weight to binary university and country indicators. Gradient Boosting, in particular, focuses on a few key features with relatively low but concentrated importance values.

- **Conclusion**:
  The dataset reveals that the total cost of international education is primarily driven by **tuition fees** and **rent**, with the **living cost index** and **program duration** serving as secondary contributors. Among the models evaluated, **Logistic Regression** and **Gradient Boosting** emerged as the

top performers, each achieving an **accuracy and F1 score of 98.35%**, with **ROC AUCs of 99.89% and 99.68%**, respectively highlighting their strong classification capabilities. In terms of feature importance, **Tuition_USD** and **Rent_USD** consistently ranked as the most influential predictors across all models.

4. Collaboration Summary

This project was completed by a two-member team. We selected four datasets from Kaggle: **Lifestyle Factors and Their Impact on Students**, **AI Generated Ghibli Style Image Trends (2025)**, **Average Daily Screen Time for Children**, and **Cost of International Education** to explore and analyze using various machine learning techniques. We divided the work by assigning two datasets to each member, who independently handled data cleaning, exploratory data analysis (EDA), model implementation, and evaluation. For the PowerPoint presentation, we collaborated by contributing slides and insights based on our respective datasets. Communication was maintained regularly through in-person meetings, Discord, and Google Docs, which we used to compile the final report. This structured division of tasks enabled us to work efficiently while ensuring consistency in both analysis and presentation.

5. Conclusion

Across all four datasets, we found that **Logistic Regression** consistently performed well in three of them, offering strong accuracy and interpretability, while **Random Forest** was the best performer in one dataset.

For **Student Lifestyle Factors**, Logistic Regression excelled at identifying medium-GPA students.
In the **Ghibli Image Trends** dataset, Random Forest outperformed other models in the binary classification setup, demonstrating that reducing class complexity improves accuracy.

In the **Screen Time** dataset, behavior patterns were strongly linked to **Screen Time Type** and **Age**, with Logistic Regression again delivering the best results.

For the **International Education Cost** dataset, both Logistic Regression and Gradient Boosting achieved top accuracy and high ROC AUC scores, with **Tuition** and **Rent** emerging as key predictors.

6. Datasets Sources

Lifestyle Factors and Their Impact on Students:
https://www.kaggle.com/datasets/charlottebennett1234/lifestyle-factors-and-their-impact-on-students

AI Generated Ghibli Style Image Trends (2025):
https://www.kaggle.com/datasets/uom190346a/ai-generated-ghibli-style-image-trends-2025

Average Daily Screen Time for Children :
https://www.kaggle.com/datasets/ak0212/average-daily-screen-time-for-children

Cost of International Education :
https://www.kaggle.com/datasets/adilshamim8/cost-of-international-education/data