CSL 2010

STUDENT PERFORMANCE PREDICTION

MAJOR PROJECT REPORT

Rahul Ahuja(B23CH1037)
Manan Ajmera(B23CH1026)
Vishwendra Pratap Singh(B23CH1048)
Chirag Bhutra(B23MT1012)

Introduction-

Understanding what influences academic success is key to improving learning, creating personalized education, and making better interventions. Traditional ways of evaluating performance, like focusing only on test scores and grades, often miss important aspects of a student's abilities and challenges.

This project uses machine learning to build a model that can predict student performance based on different factors, such as Study time (Weekly), Absences, attendance, Extracurricular activities, and behaviour. By analysing this information, the model can offer useful insights to help teachers as well as students to focus more on weaker part and adjust teaching methods to improve results

Flow chart-

Importing Data-

Importing and reading the csv file using pandas library.
Dataset used: [Student Performance Prediction]

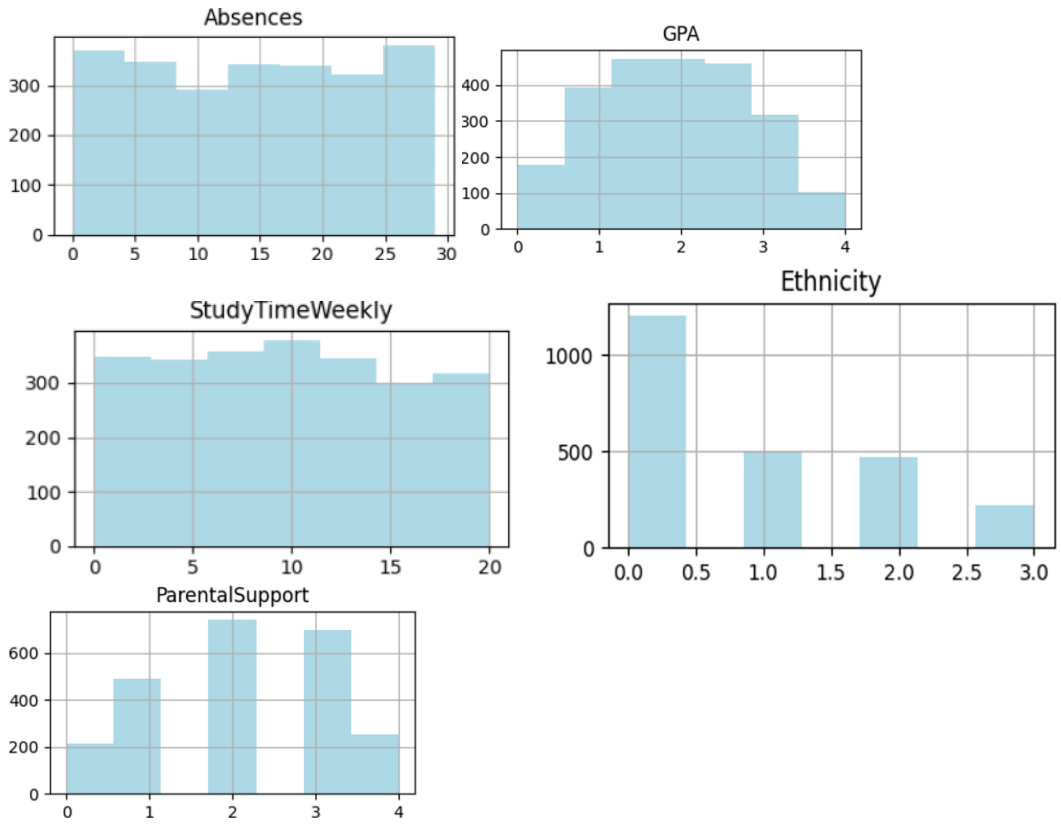
Preprocessing-

- ❖ Checking for Categorical and Numerical columns
  We get-
- No categorical columns
- Numerical columns: (['StudentID', 'Age', 'Gender', 'Ethnicity', 'ParentalEducation', 'StudyTimeWeekly', 'Absences', 'Tutoring', 'ParentalSupport', 'Extracurricular', 'Sports', 'Music', 'Volunteering', 'GPA', 'GradeClass'])
- No Null Values
- RangeIndex: 2392 entries, 0 to 2391
- Data columns (total 15 columns):
- Target Variable: Grade Class
- GradeClass: Classification of students' grades based on GPA:
  - 0: 'A' (GPA >= 3.5)
  - 1: 'B' (3.0 <= GPA < 3.5)
  - 2: 'C' (2.5 <= GPA < 3.0)
  - 3: 'D' (2.0 <= GPA < 2.5)
  - 4: 'F' (GPA < 2.0)
-

Explanatory Data Analysis()-

❖ **Histogram for Numerical columns-**

- The age group is centred around 15-18 years.
- Study time over the week is distributed around 10 hours with no extreme outliers.
- Absences are skewed towards 0–10, but extends up to 30, indicating some students have frequent absences.
- Students have less participation for Extracurricular like such as Music and Sports or Volunteering. (majority is 0 i.e 'no')
- Ethnicity is coded as- 0: Caucasian, 1: African American, 2: Asian, 3: Other
- Parental Education varies such as, 0: None, 1: High School, 2: Some college, 3: Bachelor's, 4: Higher. With most parents having College level education.
- Parental Support is coded as- 0: None, 1: Low, 2: Moderate, 3: High, 4: Very High. With most. Parents supporting in range moderate to high.



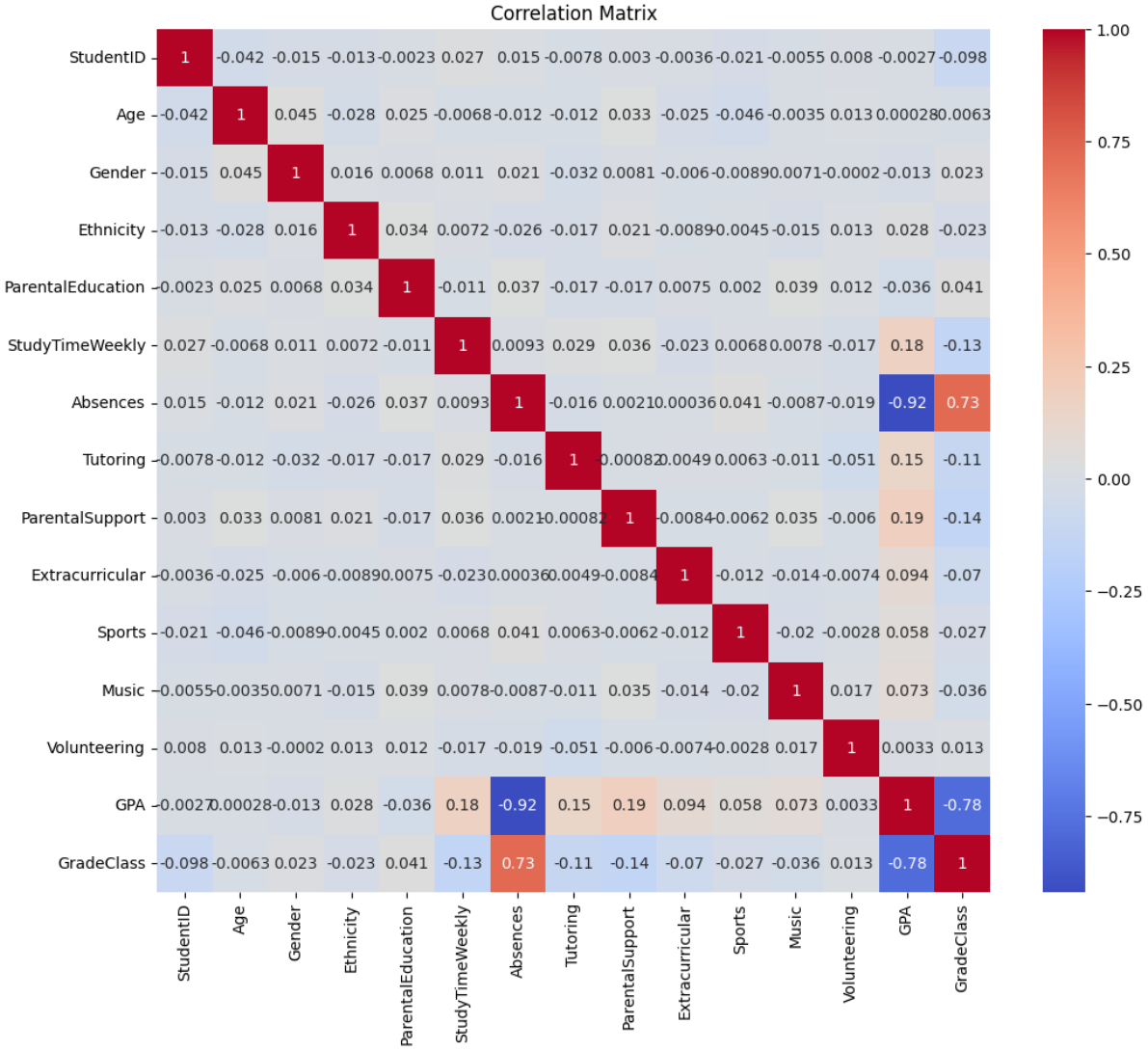
❖ Correlation heatmap

1. Strong Positive Correlation with GradeClass:
   - GPA (0.78) suggests that GPA has strong influence on a student's grades.
   - Absences (0.73) shows that students who more often remain absent are at a higher risk of obtaining lower grades. This relationship simply implies that maintaining good attendance is crucial for academic performance.

2. Strong Correlation between features:
   - GPA and Absences (-0.92) students who miss classes are associated with lower GPA.
   - Parental Support and GPA (0.19): Students receiving parental support tend to perform better.

3. Low correlation with GradeClass:
   - Gender, Age show weak (near-zero) correlation.
   - Both Sports and Volunteering have minimal correlation.

- Tutoring and GPA: Weak negative correlation (-0.11), implying that tutoring might be more common among students with lower GPAs.



Note: here grade class classification is given as GradeClass: Classification of students' grades based on GPA:

- 0: 'A' (GPA >= 3.5)
- 1: 'B' (3.0 <= GPA < 3.5)
- 2: 'C' (2.5 <= GPA < 3.0)
- 3: 'D' (2.0 <= GPA < 2.5)
- 4: 'F' (GPA < 2.0)

Hence the we can see inverese (negative) realtion between GradeClass and GPA, but in actual it is positve 0.78 Similarly Absences and GradeCLass have correlation of -0.73

Results-

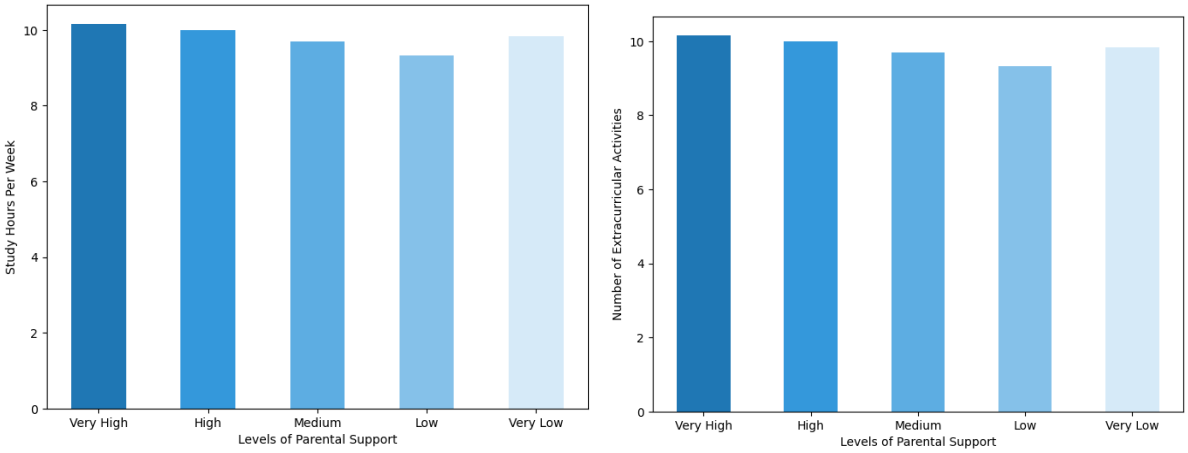
The features **StudentID**, **Music**, **Sports**, and **Volunteering** were dropped from the analysis due to their minimal or no correlation with our target variable, as shown in the correlation matrix. This removal simplifies the dataset for more focused insights.

❖ Boxplots-

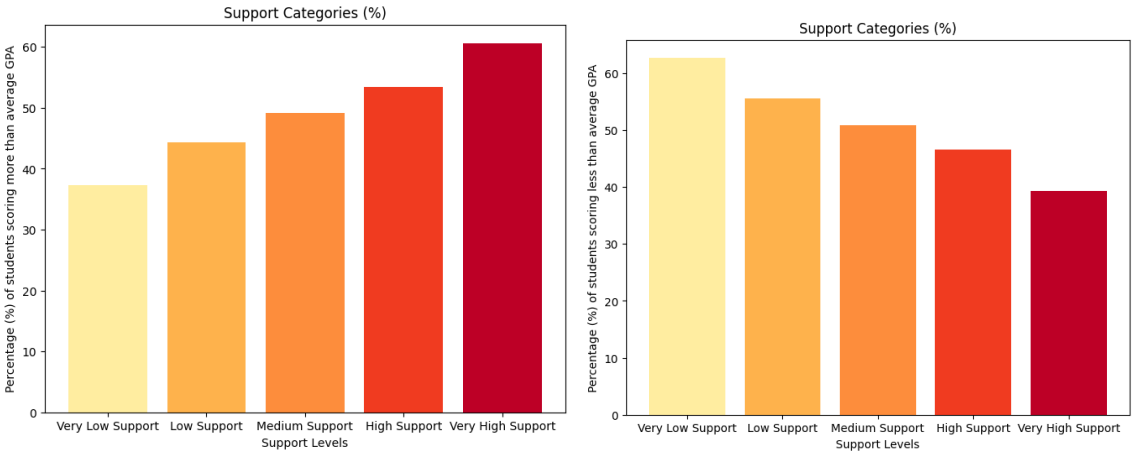
- They are used to visualize data distribution and mainly to detect outliers.
- The boxplots confirm what we see in the correlation matrix, showing that students are not very involved in activities like sports, music, and volunteering.
- Also, it is noted that there are no outliers in any of the features.

****

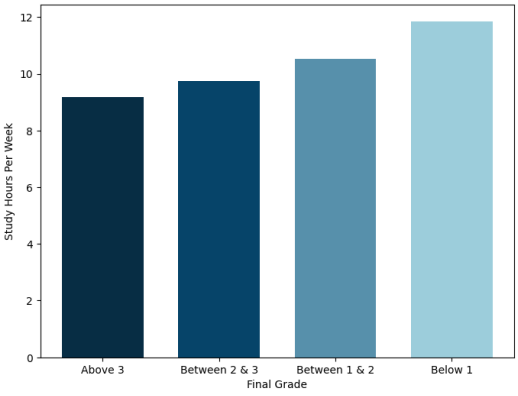
❖ Avanced EDA:

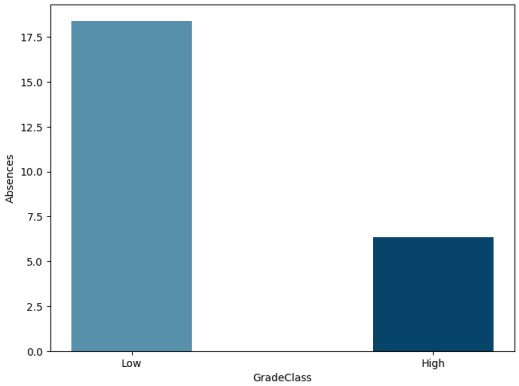


- -We can notice that study hours per week of students does not depend on level of parental support.
- Also participation in extracurricular activities of students does not depend on level of parental support.



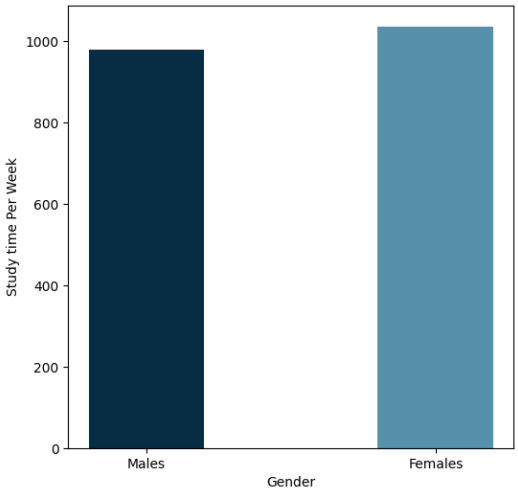
- It is noticed that a higher percentage of students with final GPA above average have very higher parental support.
- Similarly higher percentage of students with final GPA below average have very lower parental support

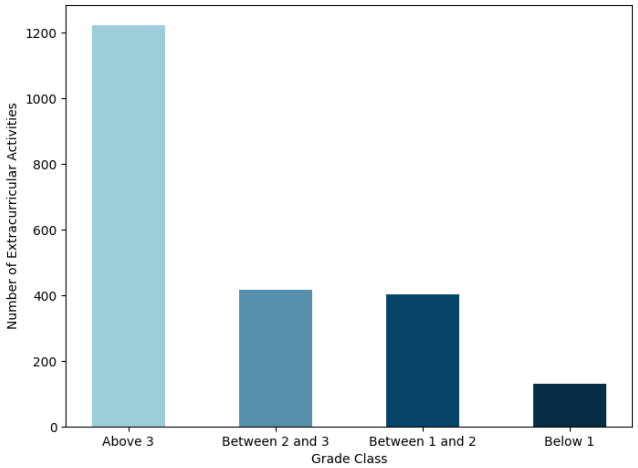




- From above plot, it is observed that students with final grade class above 3(Low grade) have smallest study hours per week.

- From above right side plot,we can notice that students with high grade class have less number of absence compared to those having low grade class.





- notice that number of students who is scoring more than average gpa is larger for females
- from the second plot it is observed that Students scoring less grade(above 3) involved in large number of extracurricular activities,while those scoring higher grade(below 1) involved in less number of extracurricular activities.

****

Standardizing Numerical Features-

We have used StandardScaler to standardize the numerical columns['Age', 'ParentalEducation', 'StudyTimeWeekly', 'Absences', 'ParentalSupport', 'GPA'
], which transform their mean to 0 and standard deviation to 1, making them easier to compare on a similar scale.
This helps improve the performance of machine learning models by reducing the impact of differences in measurement units.

Train Test Split-

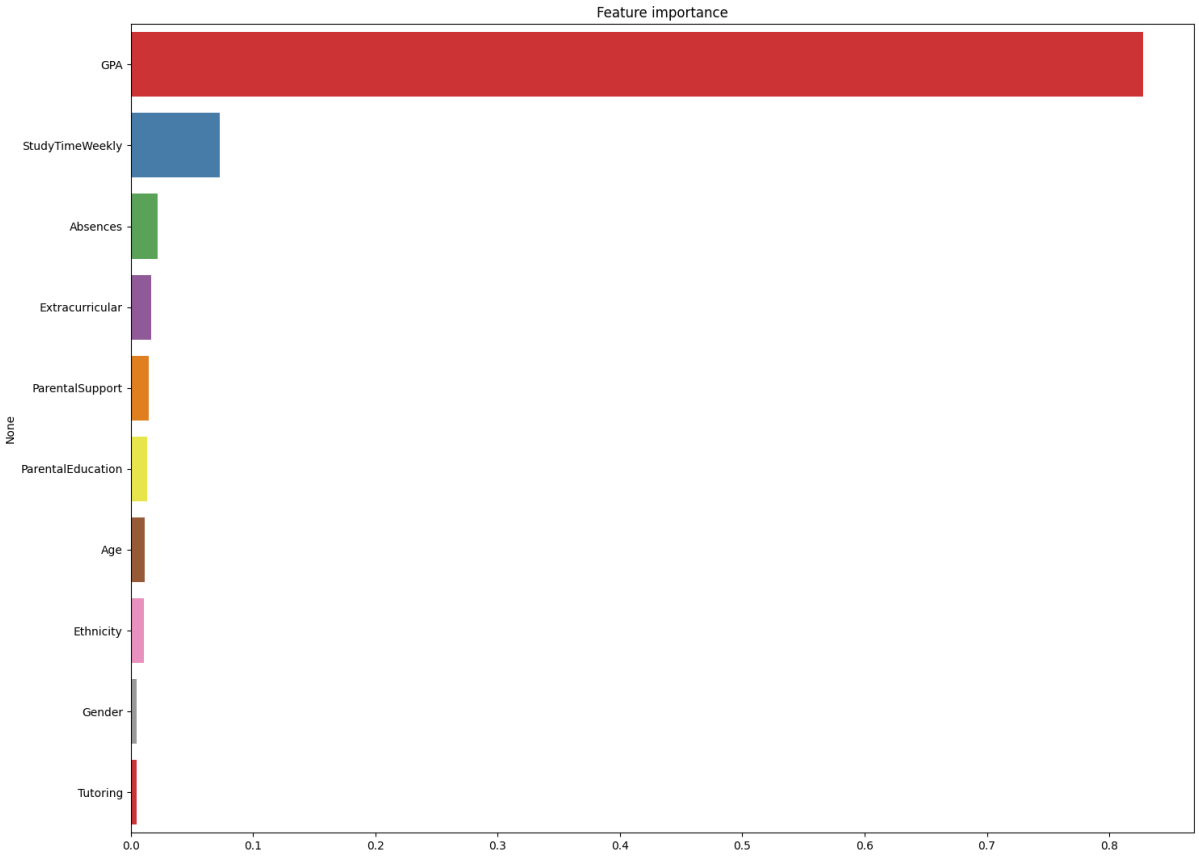
- Splitting the data into 2 parts-

80% for training and 20% for testing.

- Our target variable is **GradeClass**, and columns which have minimal correlation with GradeClass like **StudentID**, **Volunteering**, **Sports**, and **Music** are dropped as they are not contributing much.

❖ Feature Importance-

- We have used “RandomForestClassifier” to estimate feature importance.
- The following bar plot displays these feature importances, arranged from the most to the least significant.



❖ Model training-

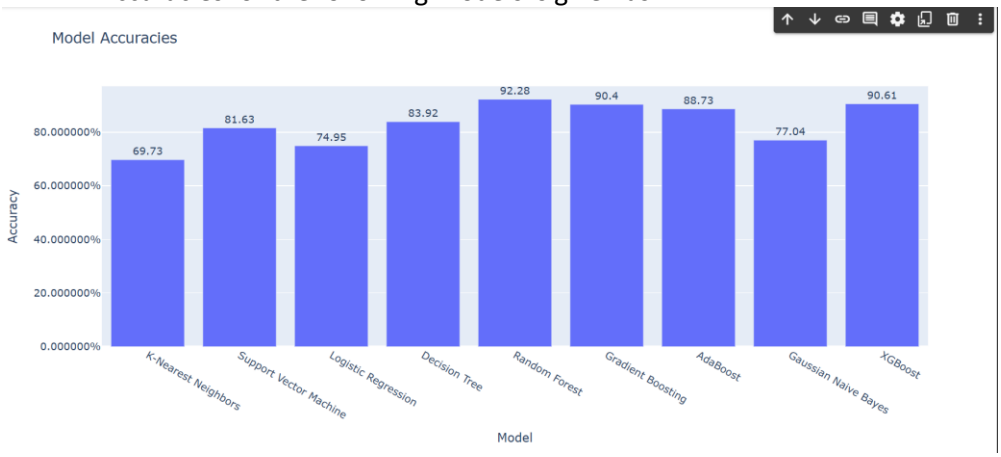
- Training various models to see which model is best for our dataset.
- We use following models in our dataset-

S no.	Models
1	K-Nearest Neighbors
2	Support Vector Machine
3	Logistic Regression
4	Decision Tree
5	Random Forest

6	Gradient Boosting
7	AdaBoost
8	Gaussian Naive Bayes
9	XGBoost

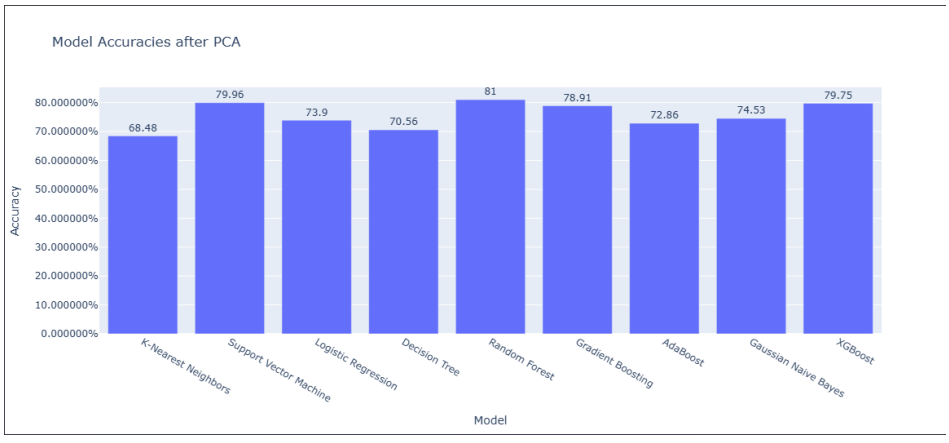
Each model was trained using the training set and tested on the test set to measure its accuracy. The bar chart shows the accuracy of each model as a percentage, making it easy to compare their performance.

- Accuracies for the following Models is given as-



Random Forest has highest accuracy (92.28%) among all and **K-Nearest Neighbours** has least(69.73%).

❖ **Applying PCA**



- We applied PCA with n_components=0.95 to retain 95% of the variance, reducing the feature space while keeping significant information.

The decrease in accuracy after applying PCA is due to several factors:

- **Loss of Information**: PCA reduces dimensionality by capturing the main variance, but some of the detailed structure that helps models classify accurately may be lost. Lower-variance features, though less prominent, can still hold critical information for classification tasks.
- **Feature Transformation**: PCA transforms features into principal components, which might not align well with the natural class boundaries. This can make it harder for models to capture class distinctions.
- **Non-linearity**: Some models (decision trees, boosting methods) perform better with the original feature set, as PCA components might not fully capture non-linear relationships crucial for these models.

❖ Model Testing-

To assess the effectiveness of various classification models in predicting student grade classifications, a range of performance metrics were calculated for each model. This approach provides a comprehensive view of how well each model performs across multiple dimensions, including accuracy, precision, recall, F1 score, and ROC-AUC.

	Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
0	K-Nearest Neighbors	0.697286	0.691175	0.697286	0.689827	0.831309
1	Support Vector Machine	0.816284	0.827685	0.816284	0.802512	0.917966
2	Logistic Regression	0.749478	0.714032	0.749478	0.728252	0.892948
3	Decision Tree	0.832985	0.837038	0.832985	0.833235	0.862169
4	Random Forest	0.920668	0.919915	0.920668	0.918836	0.921391
5	Gradient Boosting	0.906054	0.907010	0.906054	0.904905	0.912971
6	AdaBoost	0.887265	0.887558	0.887265	0.875741	0.851674
7	Gaussian Naive Bayes	0.770355	0.746119	0.770355	0.756784	0.899337
8	XGBoost	0.906054	0.905210	0.906054	0.904284	0.913114

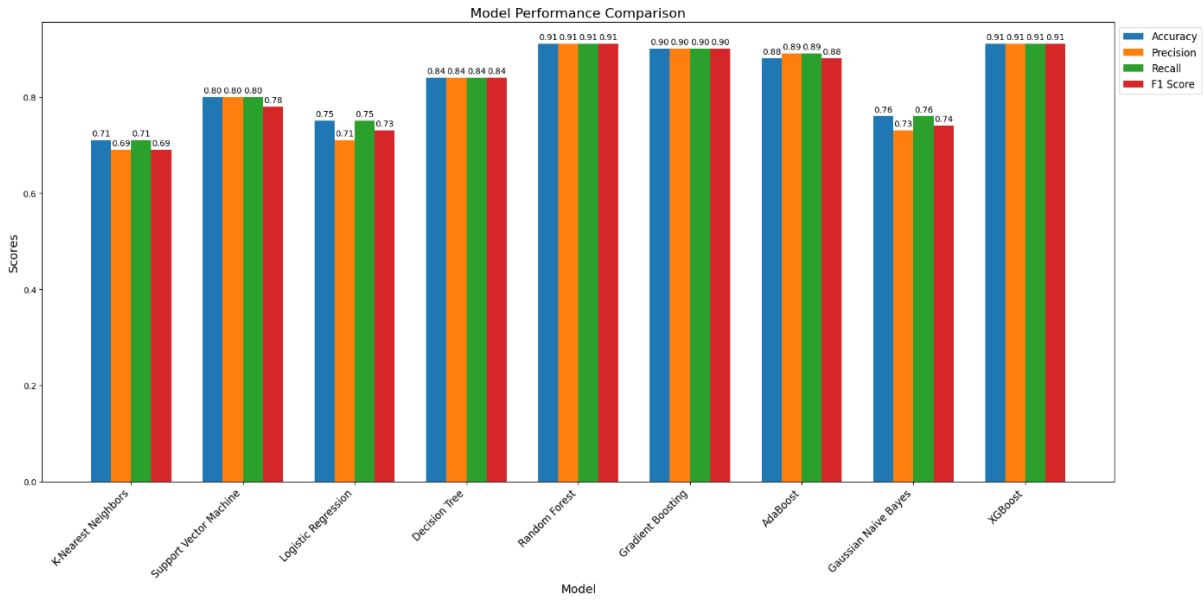
1. Random Forest, Gradient Boost and XGBoost achieves highest performance among the models (all the scores varying from 90%-92%).
This demonstrates excellent balance between precision, recall, and overall classification capability, which indicates that the models can accurately predict both positive and negative classes.

2. SVM showed a good performance with an accuracy of 0.82 and a high ROC-AUC score of 0.92, indicating it is effective at separating classes. However, its F1 score (0.80) suggests that it may slightly underperform in balancing precision and recall compared to the top models.

3. Decision tree and AdaBoost while being simpler then the ensemble models differ in scores about 5% (approx 85% accuracy and other scores). This suggests that the model be less consistent at identifying boundaries then the above models.

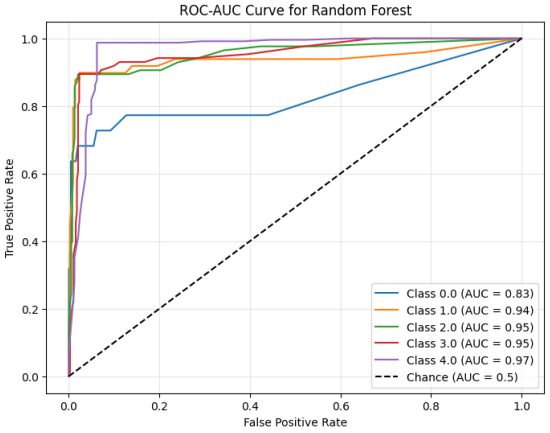
4. Logistic Regression and Gaussian Naive Bayes both have lower accuracies (0.75 and 0.77, respectively), indicating that they may not capture the complexity in the data as effectively as other models. However, they still offer reasonable ROC-AUC scores (0.89 and 0.90), which suggests they are capable of separating classes, though not as accurately as ensemble methods.

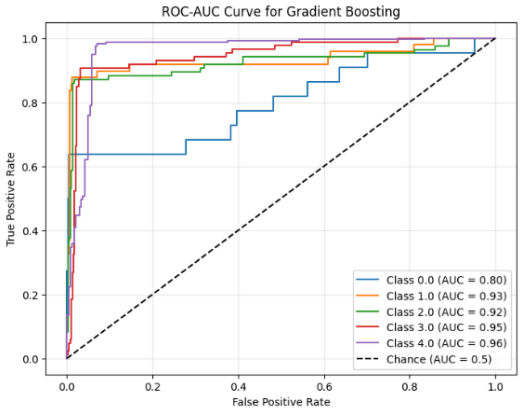
5. KNN has the lowest accuracy (0.70), precision, recall, and F1 score, as well as the lowest ROC-AUC score (0.83). This result implies that KNN might not be suitable for this dataset, potentially due to its sensitivity to the structure and scaling of data, or its reliance on local data points for classification.

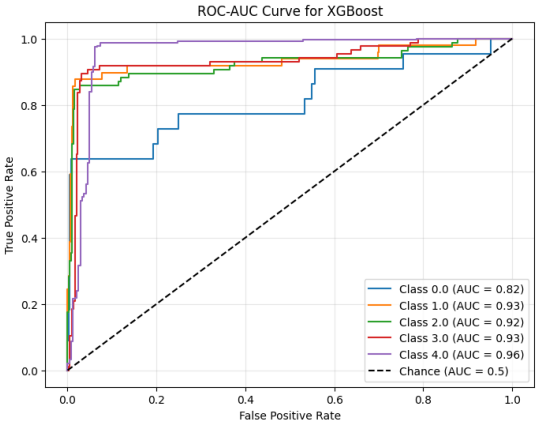


ROC-AUC plots for some models

- All three models achieve high AUC values for most classes, indicating strong performance in this classification task.
- Random Forest slightly outperforms the other models in overall AUC, especially for Class 4, which has the highest AUC across all models (0.97).
- Class 0 is consistently challenging for all models, with AUC values around 0.80 to 0.83, suggesting that distinguishing this class may require additional tuning or feature engineering.
- These plots demonstrate that while all models perform well, there are some class-specific variations in performance. Random Forest appears to have a slight edge in overall discriminative power.







❖ **Hyperparameter Tuning for:**

- Random Forest
- XGBoost

Hyperparameter Tuning-
To improve the efficiency of the model we have use Hyperparameter tuning . We employed Grid Search Cross-Validation (GridSearchCV) to find the best hyperparameters for each model based on accuracy.

Random Forest Classifier:
The Random Forest model is tuned using a parameter grid that includes the number of estimators (n_estimators), maximum depth of trees (max_depth), and minimum samples required to split an internal node (min_samples_split).

```
Best Random Forest Parameters: {'max_depth': 20, 'min_samples_split': 10, 'n_estimators': 100}
Best Random Forest Cross-Validated Accuracy: 0.9279
Random Forest Test Accuracy: 0.9207
```

XGBoost Classifier:
For XGBoost, the tuning parameters include n_estimators, max_depth, learning_rate, and gamma.

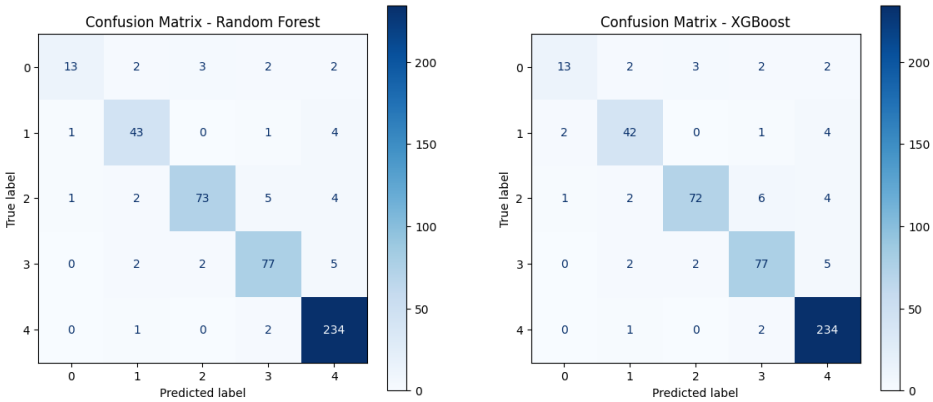
```
Best XGBoost Parameters: {'gamma': 0, 'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 200}
Best XGBoost Cross-Validated Accuracy: 0.9273
XGBoost Test Accuracy: 0.9144
```

- The hyperparameter-tuned Random Forest and XGBoost models achieved similar cross-validated accuracies (approximately 0.93) on the training data, demonstrating strong performance.

❖ **Plotted Confusion matrix for Random Forest and XGBoost:**

- For Class 0, both models correctly classified 13 instances, but misclassified some as other classes.
- For Class 1, the Random Forest correctly classified 43 instances, while XGBoost classified 42 correctly. XGBoost had slightly more misclassifications in this class.
- For Class 2, both models showed similar performance with 73 and 72 correct classifications for Random Forest and XGBoost, respectively.

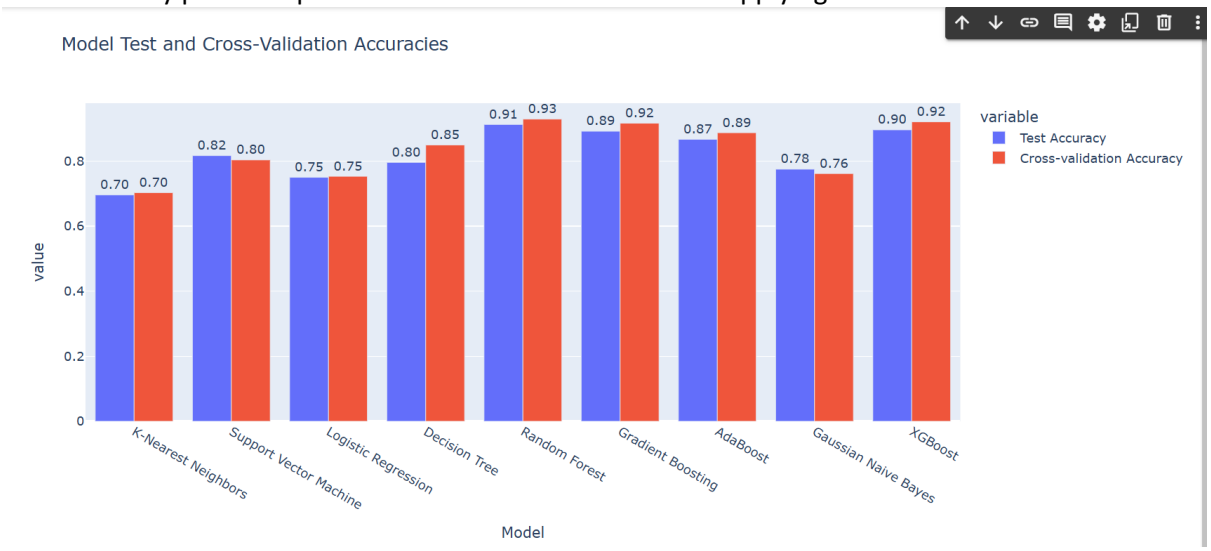
- For Class 3, both models classified 77 instances correctly, but had a few misclassifications in both cases.
    - For Class 4, both models performed very well, with 234 correct classifications each.



- Both models occasionally misclassified instances, but the confusion matrices show that these misclassifications are relatively sparse, indicating that both models have learned the classes well.

❖ **Train-Validation-Test split on all models:**

- Split the data into 80-10-10, then again train the models for train-validation-test.
- Finally print and plot the accuracies before and after applying train-validation-test.



- Overall the accuracies increase from 2-5% for each of the models

❖ **Conclusion:**

- The dataset is quite balanced with all class having considerate amount of instances, also no null values to be dealt with.
- There are a few columns which have very minimal correlation which target variable hence are dropped.
- The data is scaled by standard scaler.
- Feature importance shows that GPA is most important in the prediction of GradeClass.

- We train on various models with highest accuracy for random forest and lowest for knn.
- Further To assess the effectiveness of various classification models in predicting student grade classifications, a range of performance metrics were calculated for each model, including accuracy, precision, recall, F1 score, and ROC-AUC.
- Hyperparameter tuning for XGBoost and Random Forest, best parameters found which increases the accuracy.
- Cross-Validation performed on all the models which also increases the accuracies.

Contributions:
Manan Ajmera (B23CH1026)
Model training, hyperparameter tuning, Train-Validation-Test split, Project Report

Rahul Ahuja(B23CH1037)
Pre-Processing, Data Visualization, Advanced EDA, Model training, Project Report

Chirag Bhutra(B23MT1012)
Pre-Processing, Advanced EDA, Data Visualization, Model training, website deployement, PPT

Vishwendra Pratap Singh(B23CH1038)
Data Visualization, Model training, performance metrics , website deployement, PPT