PROJECT REPORT

Topic: Predicting students' academic performance

Course: CSE422

Section: 8

Prepared by:

- IBRAHEEM IBN ANWAR
- SIFAT E NAYNA CHHOYA

Date: 2 January 2025

Table of contents:

INTRODUCTION:

In this project, we aim to explore the different factors affecting the academic performance of a student using Machine Learning techniques. For this, we have selected a comprehensive dataset to analyze patterns and correlations between the various attributes and the outcomes present in the dataset.

Understanding these patterns can be helpful for educators, parents and even policy makers. This can help them make informed decisions. Resources for students can be allocated in a more efficient way, focus can be made to change key factors/attributes for a better performance. Our motivation behind this project is to lower the gap between potential and performance, and bring out the maximum from every student.

In this report we have presented details regarding the dataset, how we pre-processed it, the models trained and finally the results obtained. Through this analysis, we hope to contribute valuable insights into the determinants of student performance, paving the way for more effective educational strategies.

INFORMATION REGARDING THE DATASET:

Source: Kaggle
Link: https://www.kaggle.com/datasets/lainguyn123/student-performance-factors/data
Reference: Kaggle (2024). "Student Performance Factors Dataset". Available at: Student Performance Factors Dataset

This is a regression problem since the target variable 'Exam_score' is a continuous number representing the score achieved by students in an exam.

Total number of data points: 6607

Quantitative Features:

- Hours_Studied (Continuous)
- Attendance (Continuous)
- Sleep_Hours (Continuous)
- Previous_Scores (Continuous)
- Tutoring_Sessions (Discrete)
- Distance_from_Home (Continuous)
- Physical_Activity (Continuous)

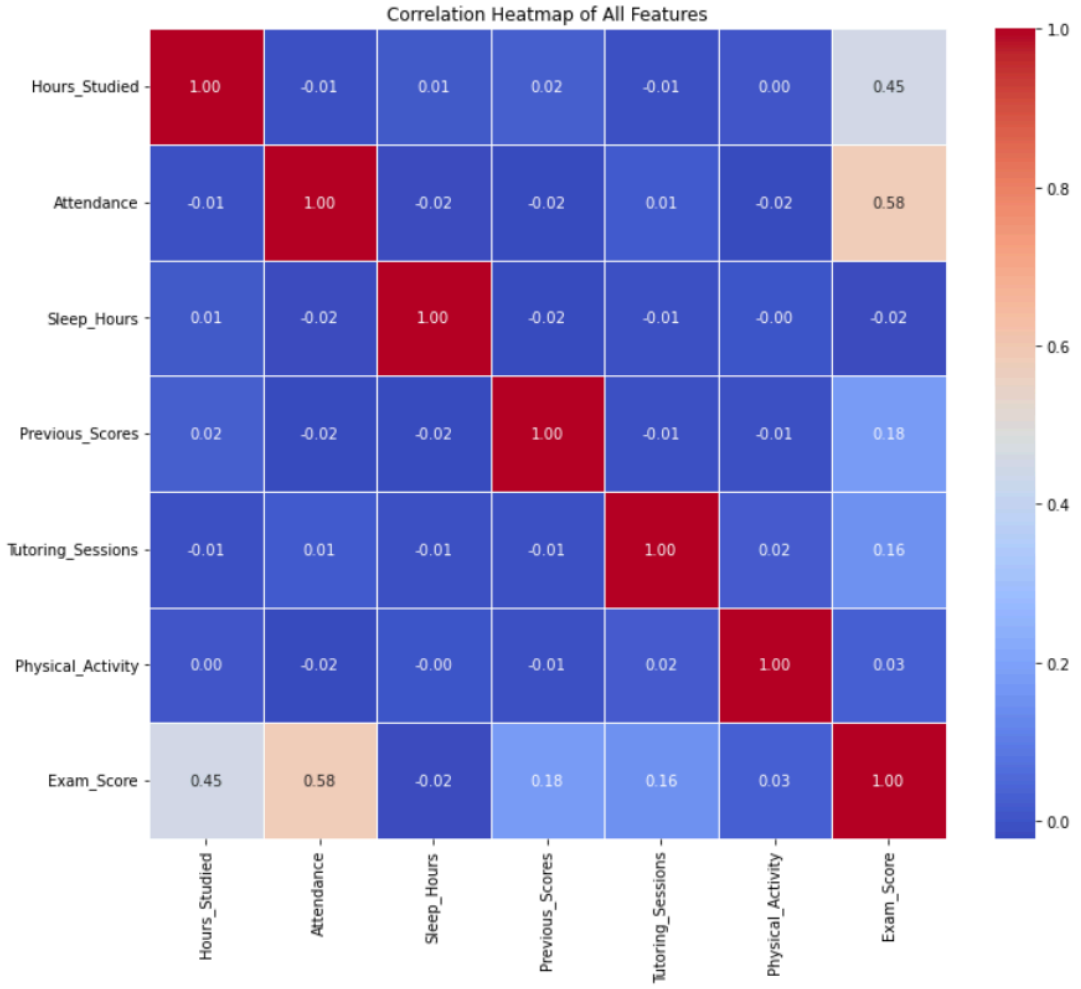
Categorical Features:

- Parental_Involvement (Nominal)
- Access_to_Resources (Nominal)
- Extracurricular_Activities (Nominal)
- Internet_Access (Nominal)
- Family_Income (Nominal)
- Teacher_Quality (Nominal)
- School_Type (Nominal)
- Peer_Influence (Nominal)
- Learning_Disabilities (Nominal)
- Parental_Education_Level (Ordinal)
- Motivation_Level (Ordinal)
- Gender (Nominal)

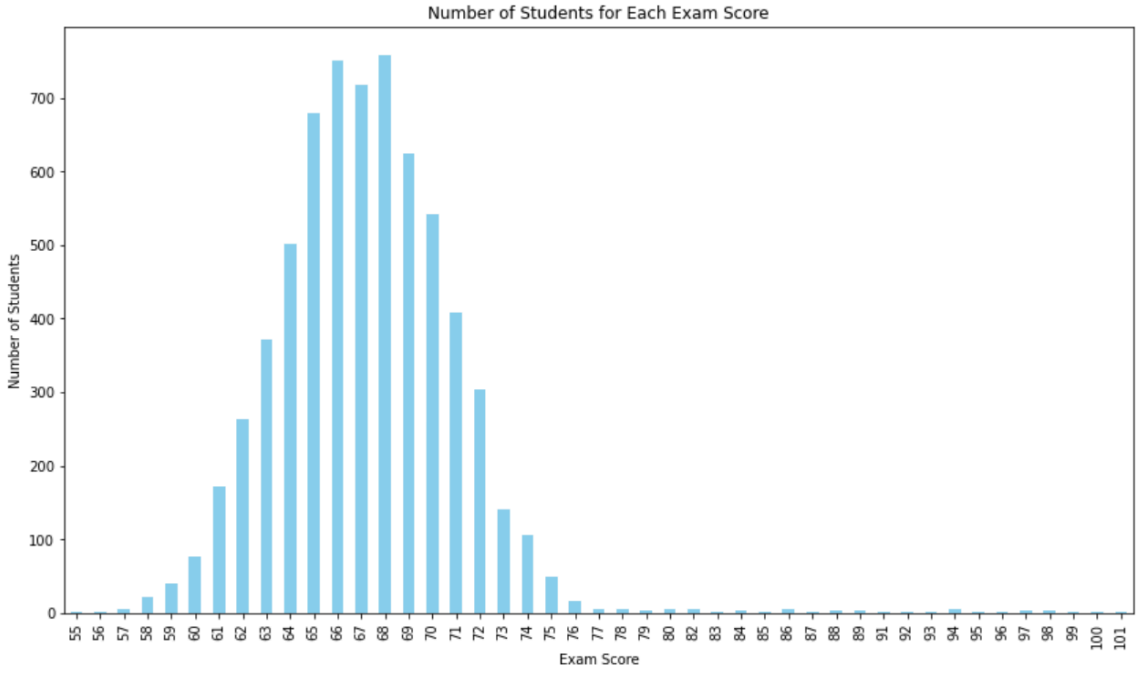
Target Variable:

- Exam_Score (Continuous)

Correlation of the features (Heatmap using seaborn library):



Distribution of exam scores:



PRE-PROCESSING OF THE DATASET:

Faults in the dataset:

Null values were found as given below:

Feature	Null_count
Teacher_Quality	78
Parental_Education_Level	90
Distance_From_Home	67

The Null values account for a very small portion of the dataset (around 1%). To handle these values, we have chosen Imputation instead of row deletion so that all of the original data is available and no data is discarded.
For the three features having Null Values, they are all Categorical features. Because of that we have chosen to impute the Null values with the Mode from their respective group.

Encoding of data:

The categorical features in our dataset are as given below:

1. Parental_Involvement
2. Access_to_Resources
3. Extracurricular_Activities
4. Motivation_Level
5. Internet_Access
6. Family_Income
7. Teacher_Quality
8. School_Type
9. Peer_Influence
10. Learning_Disabilities
11. Parental_Education_Level
12. Distance_from_Home
13. Gender

We used ordinal encoding for 7 of the Categorical features, which have an order. These features are:

1. Parental_Involvement (Low, Medium, High)
2. Access_to_Resources (Low, Medium, High)
3. Motivation_Level (Low, Medium, High)
4. Family_Income (Low, Medium, High)
5. Teacher_Quality (Low, Medium, High)
6. Parental_Education_Level (High School, College, Postgraduate)
7. Distance_from_Home (Near, Moderate, Far)

We used Label(Binary) encoding for 5 binary features, which can have only two possible values. These are:

1. Extracurricular_Activities (Yes, No)
2. Internet_Access (Yes, No)
3. School_Type (Public, Private)
4. Learning_Disabilities (Yes, No)
5. Gender (Male, Female)

We used One-Hot encoding in only one feature where there is no order for the values, and there are more than two categories. This feature is Peer_Influence (Positive, Negative, Neutral).

FEATURE SCALING:

For the feature scaling process, we used standardization using the StandardScaler from scikit-learn library. In the standardization process, the data have been transformed to have a mean of 0 and a standard deviation of 1. Using standardization we have ensured that all features were scaled to the same standard scale so that we can get maximum performance from our models.

DATASET SPLITTING:

The dataset was divided into two portions at random for training and testing purposes . 70% of the data was used for training the models and 30% of the data was used to test the model's performance.

MODEL TRAINING AND TESTING:

We experimented with three different machine learning models:

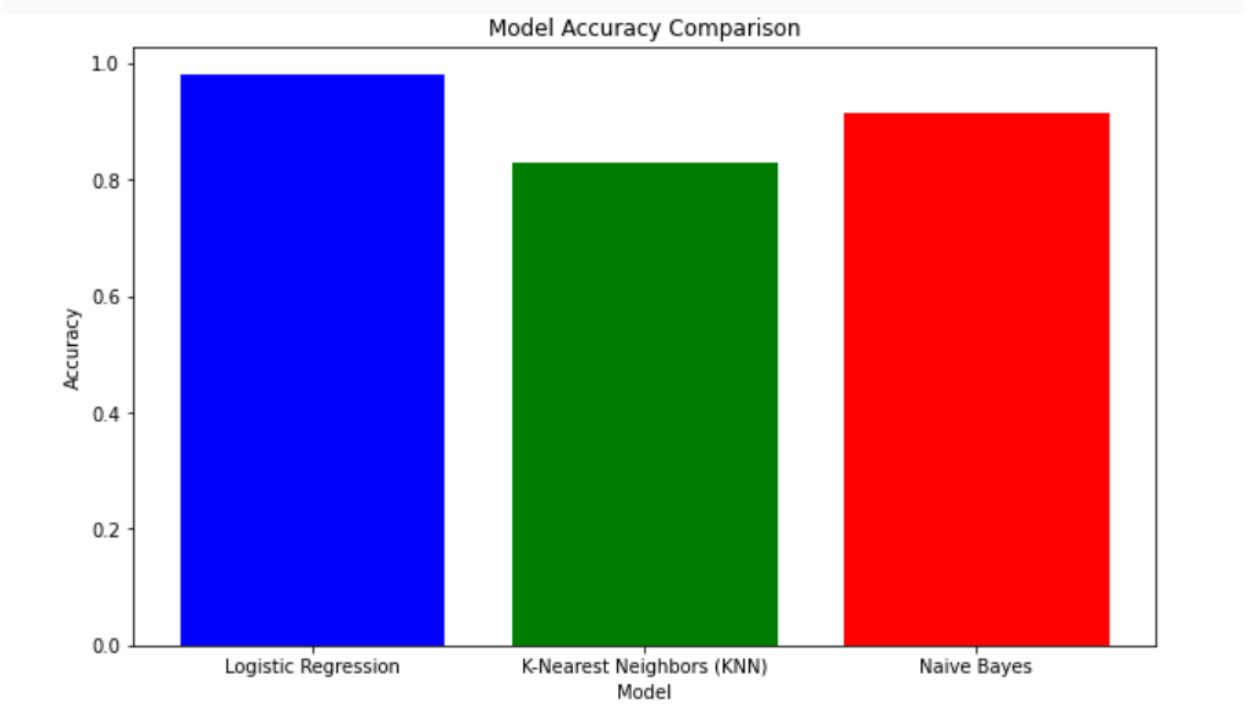
- Logistic Regression
- K-Nearest Neighbors (KNN)
- Naive Bayes

Each model was trained using the training set created by splitting the pre-processed dataset.

PERFORMANCE ANALYSIS OF THE MODELS:

To evaluate the performance of the models, we used several metrics including accuracy, confusion matrix, precision, recall, and classification report.
The accuracy results were found as given below:

Model used	Accuracy
Logistic Regression	0.9803328290468987
K-Nearest Neighbors (KNN)	0.8295511850731215
Naive Bayes	0.9152798789712556



Precision and recall values for the models:

Model used	Precision	Recall
Logistic Regression	0.9763355286401478	0.9803328290468987
K-Nearest Neighbors (KNN)	0.8192547123398706	0.8295511850731215
Naive Bayes	0.9109518287502637	0.9152798789712556

Comparison of Confusion matrices:

```
Logistic Regression Confusion Matrix:
 [[   0    3    5]
 [   0  468   22]
 [   0    9 1476]]

KNN Confusion Matrix:
 [[   0    2    6]
 [   0  225  265]
 [   0   65 1420]]

Naive Bayes Confusion Matrix:
 [[   0    3    5]
 [   0  366  124]
 [   0   36 1449]]
```

	Logistic Regression	K-Nearest Neighbors (KNN)	Naive Bayes
True Positives (A)	0	0	0
False Positives (A)	0	0	0
False Negatives (A)	8	8	8
True Positives (B)	468	225	366
False Positives (B)	9	65	36
False Negatives (B)	22	265	124
True Positives (C)	1476	1420	1449
False Positives (C)	27	271	129
False Negatives (C)	0	0	0

CONCLUSION:

From the performance Analysis we gained valuable insights into each of the models used for the dataset we have used. From among the three models, Logistic Regression performed better than the the other two models in terms of accuracy, precision and recall. It achieved the highest overall accuarcy of 0.980. From The confusion matrices we have seen that it predicted instances of Class B and Class C with minimal false positives and false negatives compared to the other two models. Naives Bayes showed moderate performance with an accuracy of 0.915 while KNN had the lowest accuracy of 0.830.
To conclude, Logistic Regression is the most suitable model for this particular dataset due its high accuracy and good performance across all the classes. Naive Bayes may be a second option giving good results, but KNN is not a suitable model for this dataset.