Machine Learning Approaches for Student Performance Prediction

1. Introduction

Learning is an ongoing and dynamic process where timely guidance and appropriate support go a long way in defining the learning experience of a student. In tertiary education, one of the most critical goals is enhancing the quality of learning by early identification of students who can be supported in their initial academic years. In India, however, several institutions continue to use conventional evaluation techniques, which focus mainly on academic performance and classroom attendance. These models tend to ignore non-academic factors that have a major impact on student outcomes, including family background, mental health, peer relationships, extracurricular activities, and access to learning resources, all of which are important determinants of academic achievement and overall development.

Student decisions and admission decisions are more often than not based only on academic records, excluding other important determinants such as family background, physical health, psychological stability, or parental background. But outside influences can affect a student to a very great degree and thus influence motivation as well as academic performance, such as their capacity to focus, time management, ability to manage stress effectively, and ability to maintain consistent interest and performance in academic activities. Early identification of students at risk, with the backing of a whole-student profile encompassing both academic and personal factors, can enable institutions to provide intervention that is required before such issues as academic underperformance or attrition get out of hand.

This study gives special attention to the incorporation of academic and non-academic (external) factors such as location, level of education of parents, internet access, sleeping time, and exercise into the prediction of students' performance. By considering these traits, we aim to better understand the reasons why students perform poorly and offer solutions to improve students' performance. For this, we implement some machine learning algorithms — Linear Regression, Ridge Regression, Lasso Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, Support Vector Regressor, and KNN — on a dataset we have obtained from Kaggle. We quantify and evaluate the performance of these models based on parameters such as mean absolute error, root mean squared error, and R^2 score. The provided prediction model employs several influential features to predict students' academic performance with more accuracy. This approach can be a useful tool for institutions to spot struggling learners early and implement measures for their academic and personal development. By providing a better overview of pupil difficulties, institutions can foster a more supportive and inclusive learning environment.

2. Literature Review

Student academic performance prediction has become increasingly popular in recent years because of its ability to improve educational outcomes and facilitate early interventions. Several studies have utilized machine learning (ML) models to predict students' success through the detection of patterns in past educational data. This section surveys relevant literature based on popular models like Linear Regression, Ridge Regression, Lasso Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, Support Vector Regressor (SVR), and k-Nearest Neighbors (KNN).

An exhaustive review by Alyahyan and Düştegör (2020) presents best practices and challenges in predicting academic success. These studies highlight contextual factors like attendance, socioeconomic status, and previous academic attainment, all of which have a significant bearing on the accuracy and efficiency of predictive models [1]. Initial research, including Acharya and Sinha (2014), tested simple ML models for early student performance prediction, pointing to the applicability of linear and decision tree models as they are interpretable [2]. Agrawal and Mavani (2015) also proved the efficiency of numerous algorithms, such as KNN and SVM, in educational data mining; nonetheless, the scarcity of diversity of features hindered the generalizability of their results [3]. Pandey and Taruna (2014) made a comparative analysis of ensemble techniques, specifically bagging and boosting approaches, that resulted in enhanced prediction performance relative to individual learners [4]. Kotsiantis et al. (2003) compared the performance of several ML algorithms when applied in distance learning contexts, and decision trees and Naive Bayes proved to be highly effective as well as computationally efficient [5].

Decision tree-based models remain popular due to their capacity to deal with nonlinear relationships and categorical variables efficiently. Smith and Davis (2019) used the Decision Tree Regressor to forecast academic grades, highlighting its advantage in interpretability as well as variable importance analysis [6]. Expanding on that, Huynh-Cam et al. (2021) used Random Forest models to improve stability and minimize overfitting and were able to successfully determine important predictive variables impacting first-year university performance [7]. The theoretical groundwork of Random Forests was established through the pioneering paper of Breiman (2001), a model highly referenced in student performance prediction literature because it is an ensemble method and can deal with high-dimensional data [8]. The excellence of Random Forests over conventional approaches in managing complex educational data sets has been confirmed by various applied studies, such as those by Hamoud et al. (2018) and Ghosh and Janan (2021), that illustrated the performance of the model in identifying subtle patterns and providing better predictive accuracy [10, 13].

Recent research has incorporated more advanced methods like Ridge and Lasso regression to tackle multicollinearity and feature selection issues. Roy and Urolagin (2017) investigated SVR and decision trees in the context of financial education, highlighting SVR's ability to handle high dimensional spaces but also its computational intensity [11]. In the meantime, Imran et al. (2019) and Shah et al. (2019) showcased the effectiveness of Lasso for feature reduction in sparse academic data, facilitating model interpretability and generalization [15]. Finally, Hasan et al. (2019) and Beaulac and Rosenthal (2019) confirm the importance of ensemble and regularized regression models in actual educational systems, validating their inclusion into institutional decision-making frameworks [17].

3. Problem Statement

This project focuses on predicting student exam scores by applying machine learning techniques to a dataset containing various attributes such as hours studied, attendance, parental involvement, motivation levels, and more. The goal is to build predictive models that can accurately forecast academic performance and to identify the most influential factors contributing to student success. This will enable early identification of students at risk and support data-driven interventions to enhance learning outcomes.

4. Objective

- Feature selection techniques are used to determine which features affect the target variable.
- Comparative analysis of all machine learning techniques for predicting student performance.
- Evaluating the performance of the model.

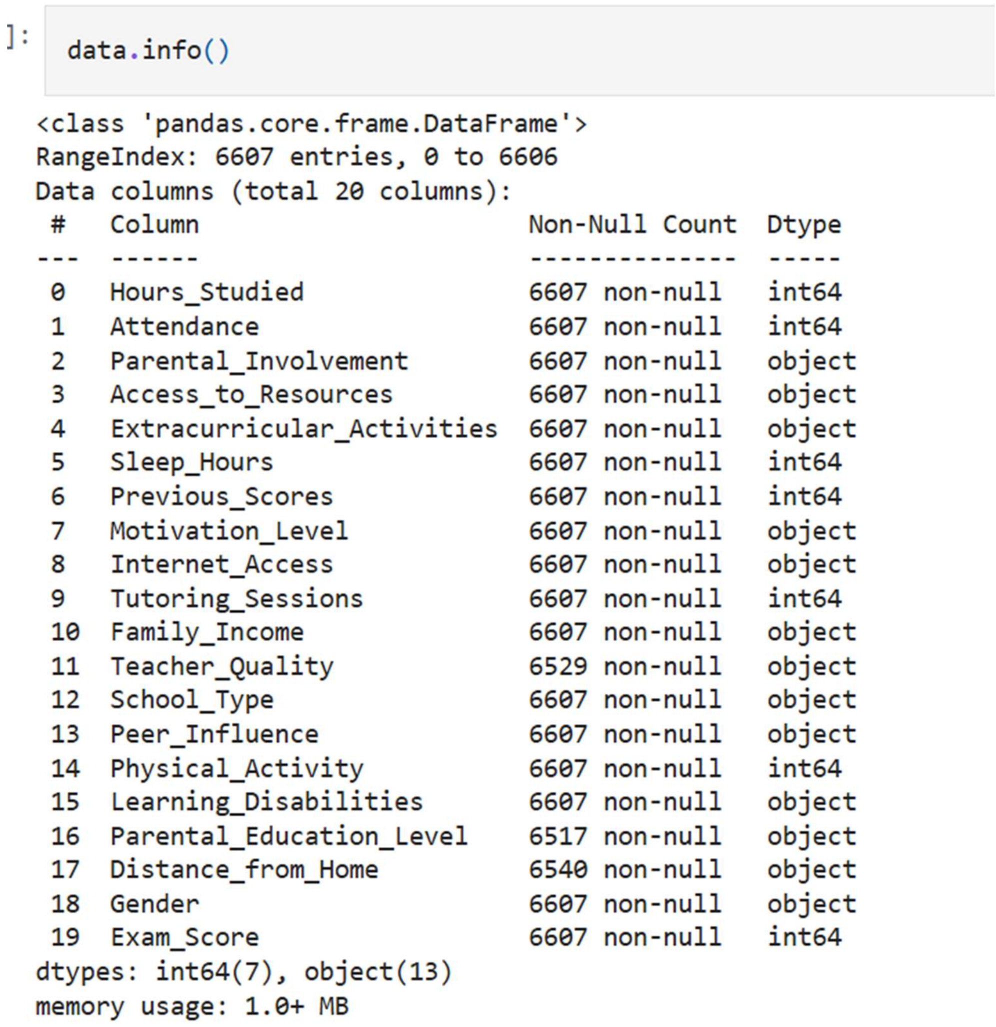
5. Methodology

About the Dataset:
The dataset for student performance prediction is taken from Kaggle. It consists of 20 columns and 6,607 rows. It provides an overview of the various factors that affect student performance in exams, including information on study habits, attendance, parental involvement, and other aspects influencing academic excellence.

Column Descriptions

Attributes	**Description**
Hours_Studies	Number of hours spent studying per week.
Attendance	Percentage of classes attended.
Parental_Involvement	Level of parental involvement in the student’s education (Low, Medium, High).
Access_to_Resources	Availability of educational resources (Low, Medium, High).

Extracurricular_Activities	Participation in extracurricular activities (Yes, No).
Sleep_Hours	Average number of hours of sleep per night.
Previous_Score	Scores from previous exams.
Motivation_Level	Student’s level of motivation (Low, Medium, High).
Internet_Access	Availability of internet access (Yes, No).
Tutoring_Sessions	Number of tutoring sessions attended per month.
Family_Income	Family income level (Low, Medium, High).
Teacher_Quality	Quality of the teachers (Low, Medium, High).
School_Type	Type of school attended (Public, Private).
Peer_Influence	Influence of peers on academic performance (Positive, Neutral, Negative).
Physical_Activity	Average number of hours of physical activity per week.
Learning_Disabilities	Presence of learning disabilities (Yes, No).
Parental_Education_Level	The highest education level of parents (High School, College, Postgraduate).
Distance_from_Home	Distance from home to school (Near, Moderate, Far).
Gender	Gender of the student (Male, Female).
Exam_Score	Final exam score.



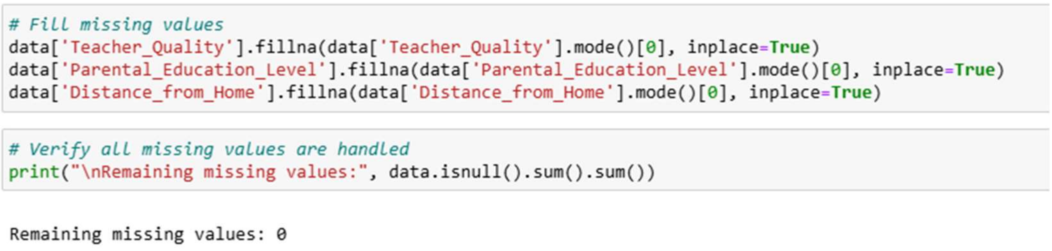
```
]:  data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6607 entries, 0 to 6606
Data columns (total 20 columns):
 #   Column                      Non-Null Count  Dtype 
---  ------                      --------------  ----- 
 0   Hours_Studied               6607 non-null   int64 
 1   Attendance                  6607 non-null   int64 
 2   Parental_Involvement        6607 non-null   object
 3   Access_to_Resources         6607 non-null   object
 4   Extracurricular_Activities  6607 non-null   object
 5   Sleep_Hours                 6607 non-null   int64 
 6   Previous_Scores             6607 non-null   int64 
 7   Motivation_Level            6607 non-null   object
 8   Internet_Access             6607 non-null   object
 9   Tutoring_Sessions           6607 non-null   int64 
 10  Family_Income               6607 non-null   object
 11  Teacher_Quality             6529 non-null   object
 12  School_Type                 6607 non-null   object
 13  Peer_Influence              6607 non-null   object
 14  Physical_Activity           6607 non-null   int64 
 15  Learning_Disabilities       6607 non-null   object
 16  Parental_Education_Level    6517 non-null   object
 17  Distance_from_Home          6540 non-null   object
 18  Gender                      6607 non-null   object
 19  Exam_Score                  6607 non-null   int64 
dtypes: int64(7), object(13)
memory usage: 1.0+ MB
```

Figure 1 shows the information of the columns in the dataset

The steps involved are:

- **Data Preprocessing**
  - **Handling missing values**: Missing values were in columns like teacher quality, parental involvement, and distance from home. We handled missing values by filling them with the mode of the data.



```
# Fill missing values
data['Teacher_Quality'].fillna(data['Teacher_Quality'].mode()[0], inplace=True)
data['Parental_Education_Level'].fillna(data['Parental_Education_Level'].mode()[0], inplace=True)
data['Distance_from_Home'].fillna(data['Distance_from_Home'].mode()[0], inplace=True)
```

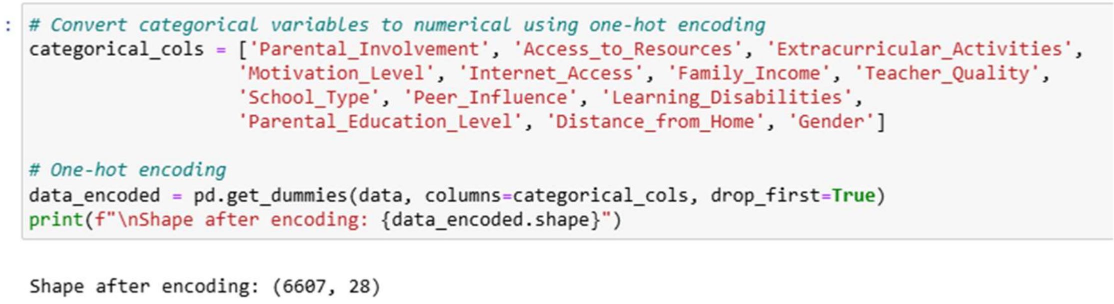
```
# Verify all missing values are handled
print("\nRemaining missing values:", data.isnull().sum().sum())
```

```
Remaining missing values: 0
```

Figure 2 shows the filling of missing values using the mode of the data

- **Checking for duplicates**: Any duplicates in the dataset are being checked, and found that there are no duplicate values in it.

- **Encoding categorical variables**: Converted categorical columns (Parental_Involvement, Access_to_Resources, Extracurricular_Activities, Motivation_Level, Internet_Access, Family_Income, Teacher_Quality, School_Type, Peer_Influence, Learning_Disabilities, Parental_Education_Level, Distance_from_Home, Gender) to numerical using one-hot encoding.



```
# Convert categorical variables to numerical using one-hot encoding
categorical_cols = ['Parental_Involvement', 'Access_to_Resources', 'Extracurricular_Activities',
                    'Motivation_Level', 'Internet_Access', 'Family_Income', 'Teacher_Quality',
                    'School_Type', 'Peer_Influence', 'Learning_Disabilities',
                    'Parental_Education_Level', 'Distance_from_Home', 'Gender']

# One-hot encoding
data_encoded = pd.get_dummies(data, columns=categorical_cols, drop_first=True)
print(f"\nShape after encoding: {data_encoded.shape}")
```

```
Shape after encoding: (6607, 28)
```

Figure 3 shows the conversion of categorical variables into numerical variables

- **Exploratory Data Analysis (EDA)**

  Several visualizations and statistical summaries were used to understand the underlying patterns in the dataset and the relationship between different variables and the target (exam score). The goal of this analysis was to gain insights into which factors might significantly influence student performance and to guide feature selection for the predictive modeling phase.

  - **Distribution of the Target Variable**

    The distribution of the target variable (exam score) was visualized using a histogram and a boxplot. This helped in:

    - Understanding the central tendency and spread of student scores.
    - Identifying any potential outliers or skewness in the data.

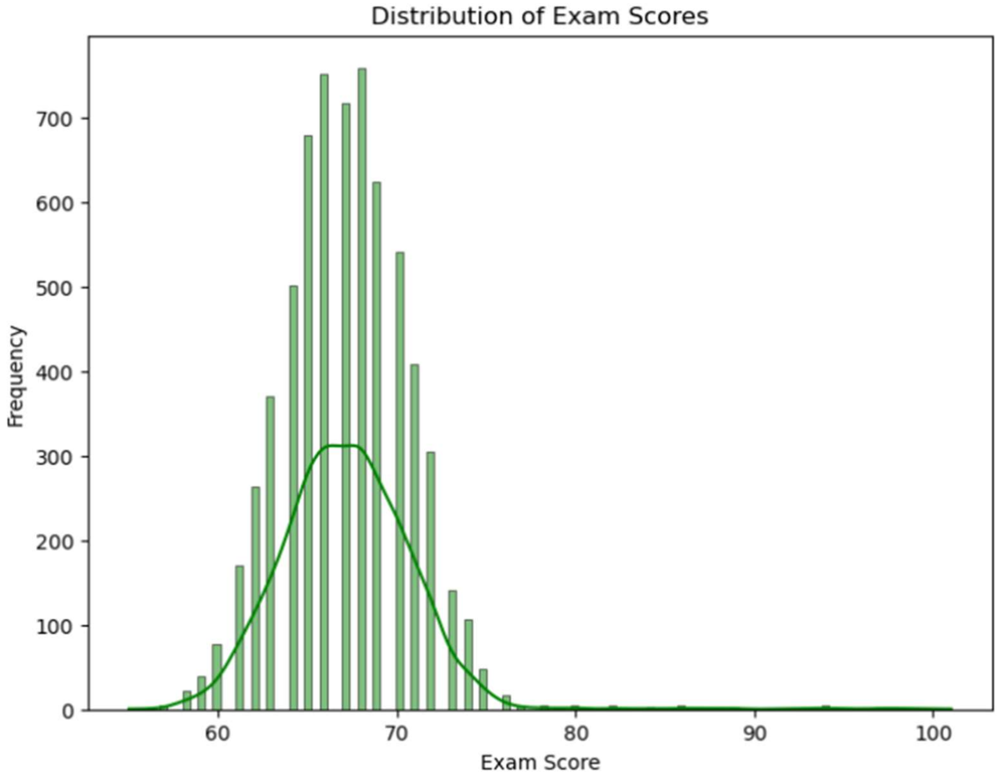


Figure 4 shows the distribution of Exam Scores

- **Relationship Between Numerical Features and Exam Score**
  Scatter plots and line plots were used to examine the correlation between exam scores and key numerical features such as:
  - Hours Studied: A positive linear relationship was observed, suggesting that students who study more tend to score higher.
  - Attendance Rate: Students with higher attendance showed better performance, indicating its importance in academic success.

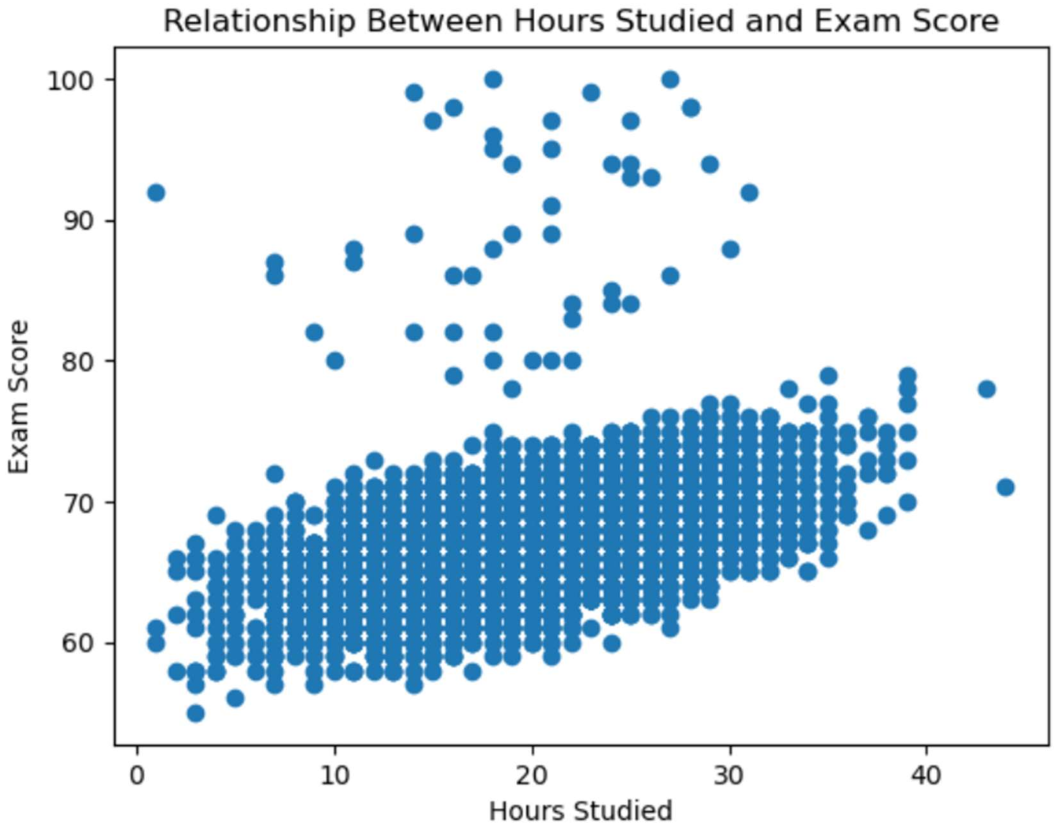


Figure 5 shows the relationship between Hours Studied and Exam Scores

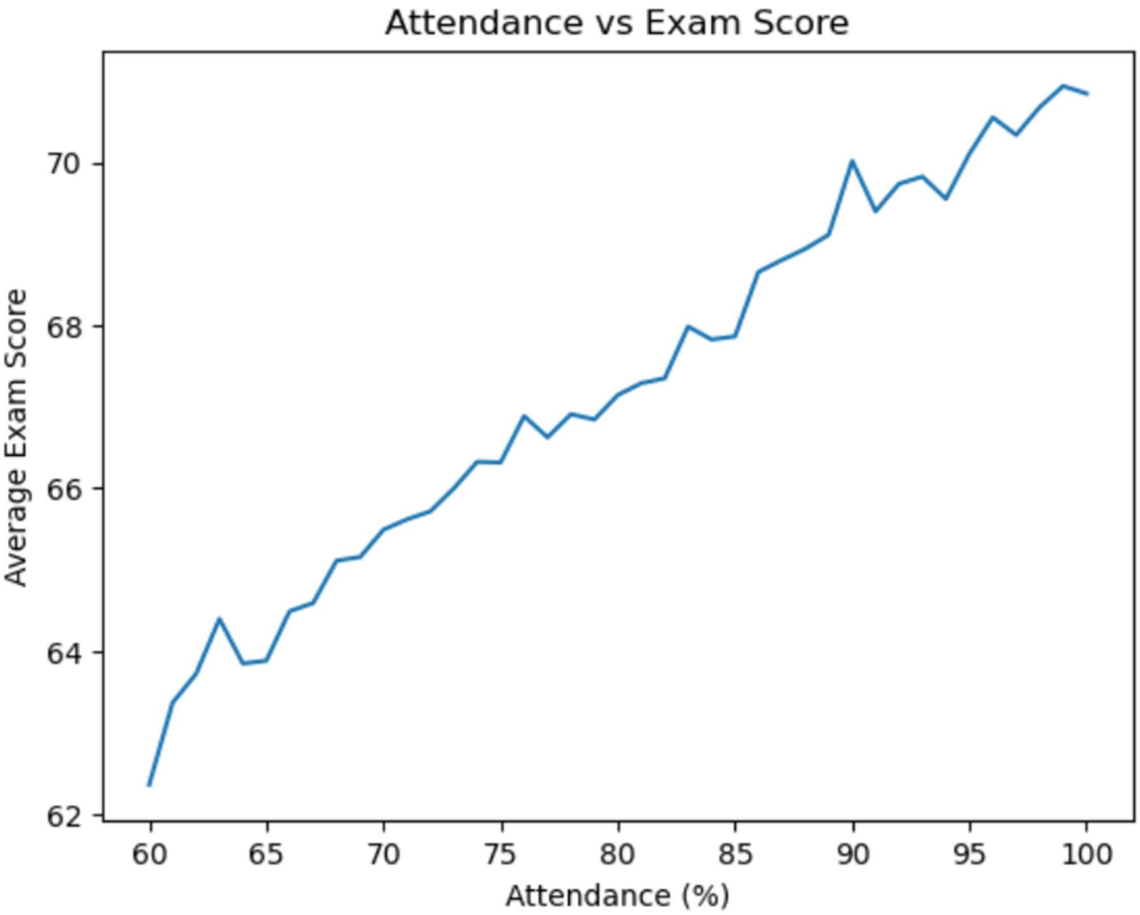


Figure 6 shows the relationship between Attendance and Exam Score

- **Categorical Feature Analysis**
  - Box plots and bar charts were created to analyze how categorical features affect the target variable:
  - Parental Involvement: Students with active parental support generally scored higher.
  - Motivation Level: A trend was observed where higher motivation levels corresponded with better performance.

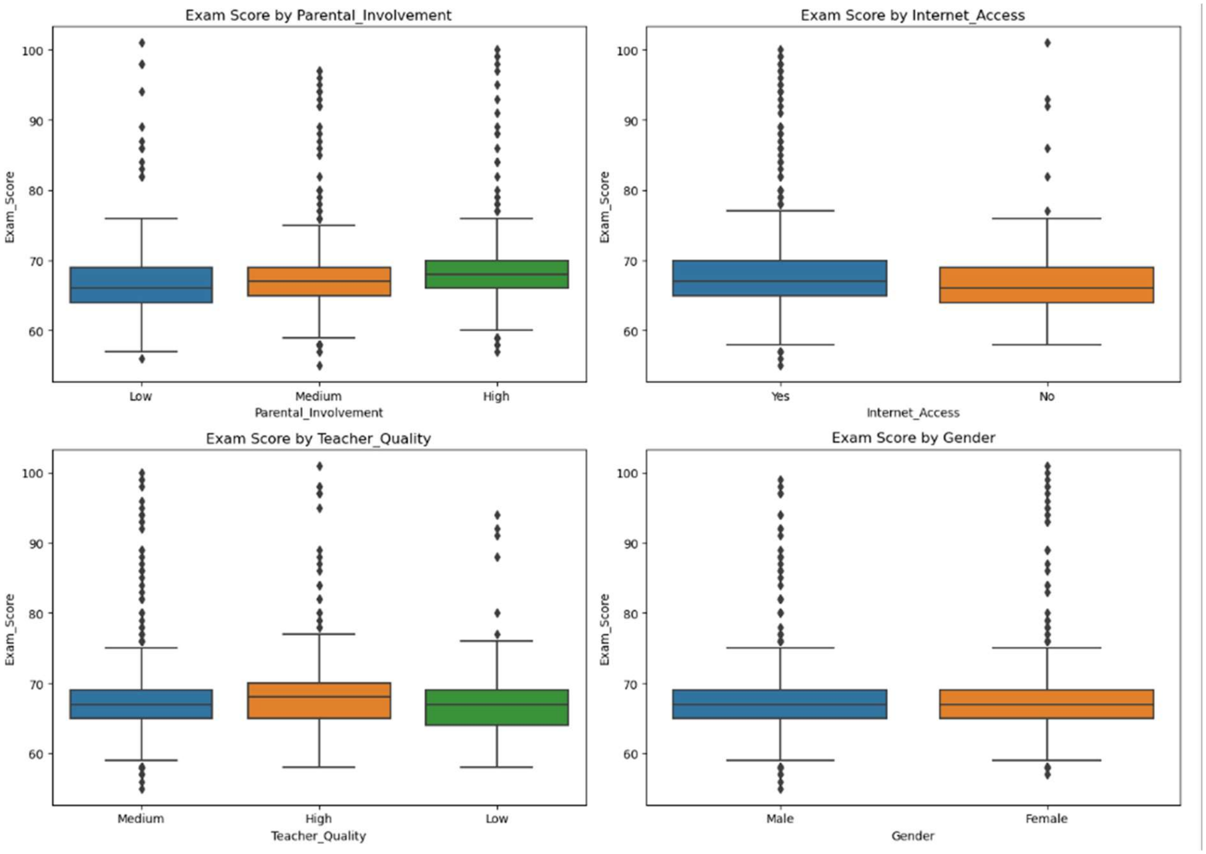


Figure 7 shows how categorical variable affects the target variable

- **Correlation Heatmap**

  A correlation matrix was plotted to quantify the relationships between numerical variables and the target:

  - Features such as hours studied and attendance had strong positive correlations with exam scores.
  - This heatmap also helped in detecting multicollinearity between independent variables.

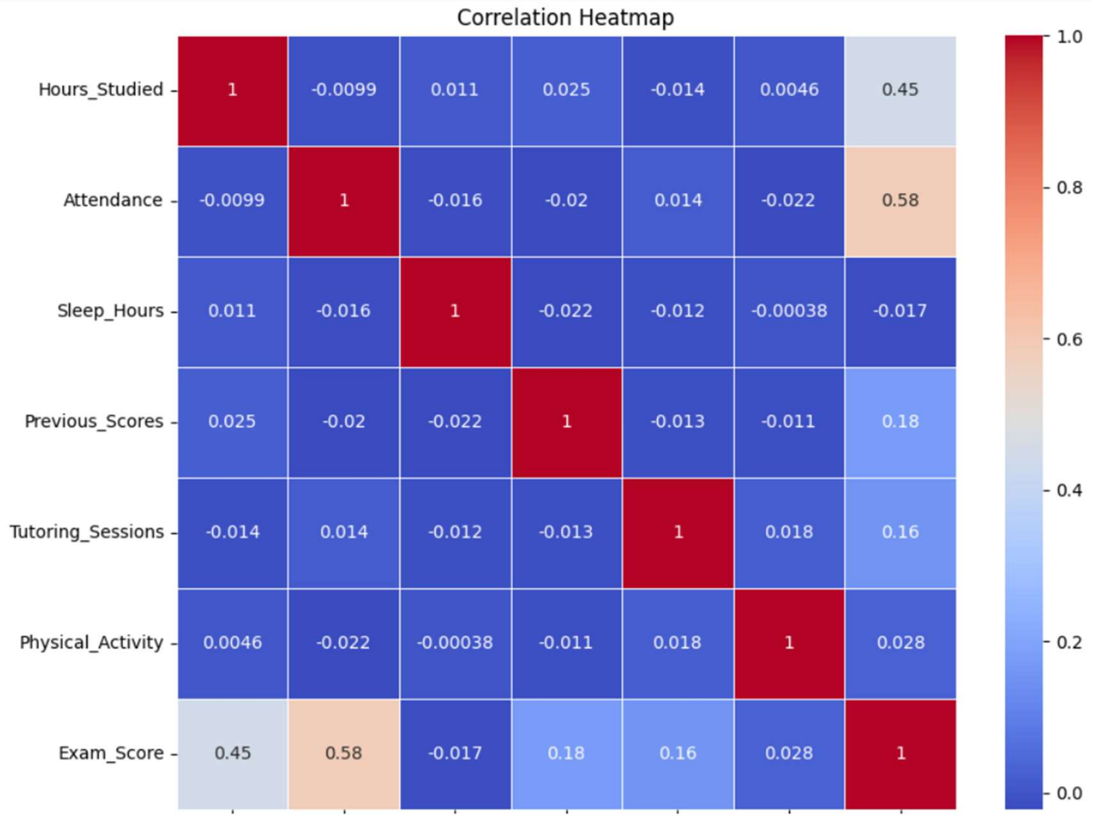


Figure 8 shows the correlation between the numerical variables and the target variable

- **Outlier Detection**
  Box plots were also used to detect and visually assess any outliers in continuous variables like hours studied vs exam scores.

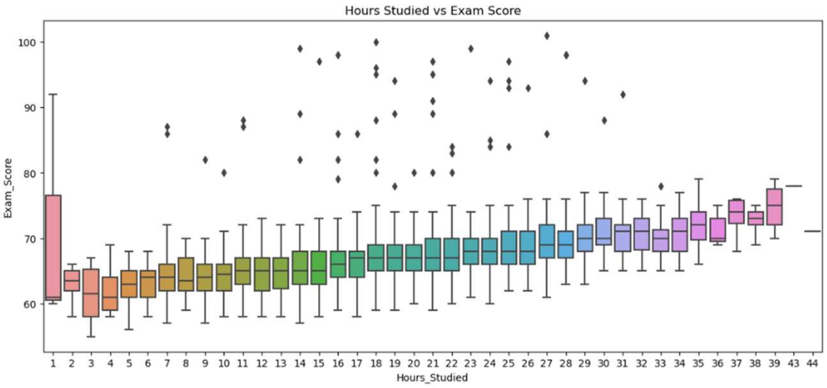


Figure 9 shows the boxplot, which helps us to see the outliers in hours studied vs exam scores

- **Feature Selection / Engineering**

  In this part, new features were created to capture complex relationships between existing variables and to improve the model's ability to make accurate predictions.

  - **Interaction Features**

    New features were created by combining existing variables to capture meaningful interactions:

    - Study_Efficiency: Combines study time and attendance to estimate how efficiently a student uses their study time (Hours_Studied * Attendance / 100).
    - Resource_Knowledge: Reflects a student's past performance amplified by their study hours, indicating how previous knowledge supports new learning (Previous_Scores * Hours_Studied / 10).
    - Wellness_Factor: Represents overall wellness by combining sleep and physical activity (Sleep_Hours * Physical_Activity / 2).

  - **Polynomial Features**

    To capture nonlinear relationships and interactions between numeric variables:

    - Second-degree (degree=2) polynomial features were generated from Hours_Studied, Attendance, Previous_Scores, and Sleep_Hours.
    - These include squared terms (e.g., Hours_Studied^2) and pairwise interactions (e.g., Hours_Studied * Attendance).
    - Only the newly generated polynomial features (excluding the original variables) were retained and added to the dataset.

  - **Ratio Features**

    Ratio-based features were added to understand relative relationships:

    - Study_to_Sleep_Ratio: Highlights balance or imbalance between study and rest (Hours_Studied / (Sleep_Hours + 0.01)), where a small constant prevents division by zero.
    - Attendance_Rate: Scales raw attendance into a normalized rate (Attendance / 100).

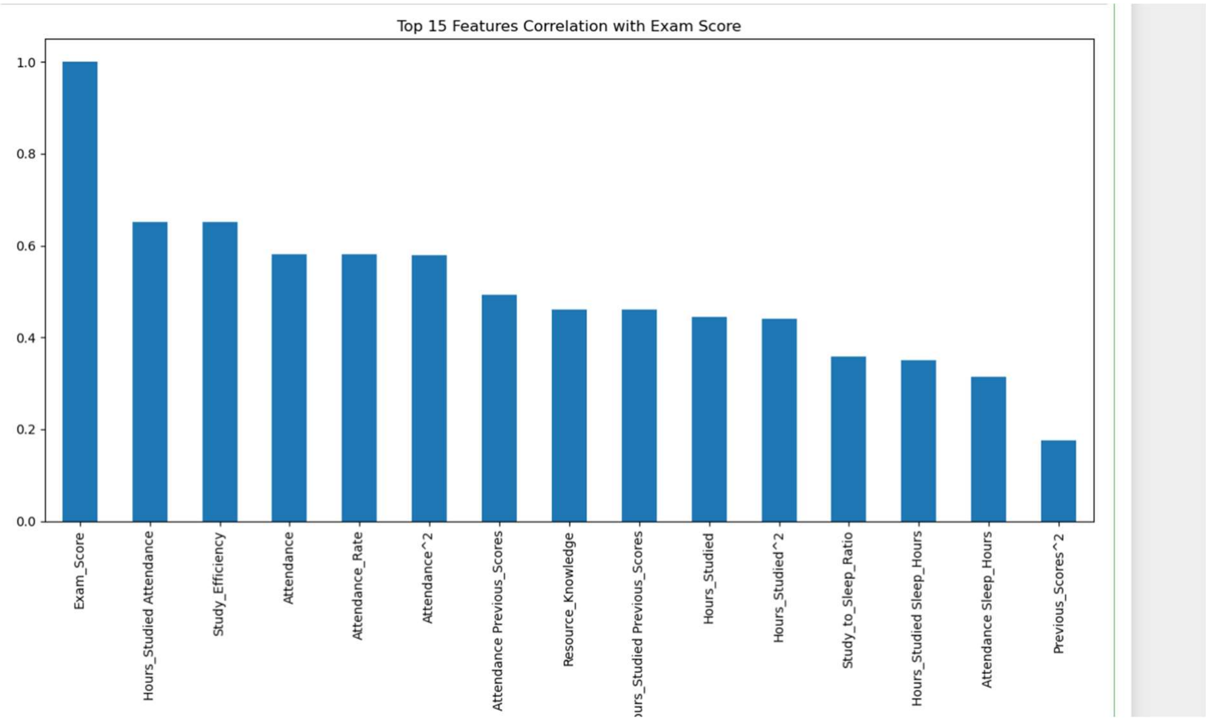


Figure 9 shows the feature correlation with Exam Score

Correlation-based highlight choice was connected to recognize the foremost powerful factors influencing student performance. The Pearson relationship coefficient was computed between each highlight and the target variable, Exam Score, to degree the quality and heading of their straight connections. The highlights were already sorted in descending order based on their correlation values, and the top 15 most strongly correlated features were selected for further analysis. These features were visualized using a bar plot to provide a clear representation of their contributions to the target variable. This analysis facilitated the identification of the most impactful features—both original and engineered—that are expected to significantly influence the model's predictive performance.

- **Model Building**

  Multiple machine learning models were designed and assessed to forecast student performance based on the engineered dataset. The models selected for comparison included Linear Regression, Support Vector Regression (SVR), K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Ridge, and Lasso Regression. The dataset was first split into training and testing sets to evaluate generalization performance. Each model was trained using the training data, and hyperparameter tuning was applied where appropriate to improve accuracy. Model performance was assessed using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2). Among the models, Lasso Regression showed the highest predictive accuracy, making it the most effective choice for forecasting student exam scores. The outcome acquired from this step played an important role in adequately guiding the process of final model choice and strongly confirming the total effect of the engineered features on the predictive accuracy of the model.

- **Model Training & Evaluation**

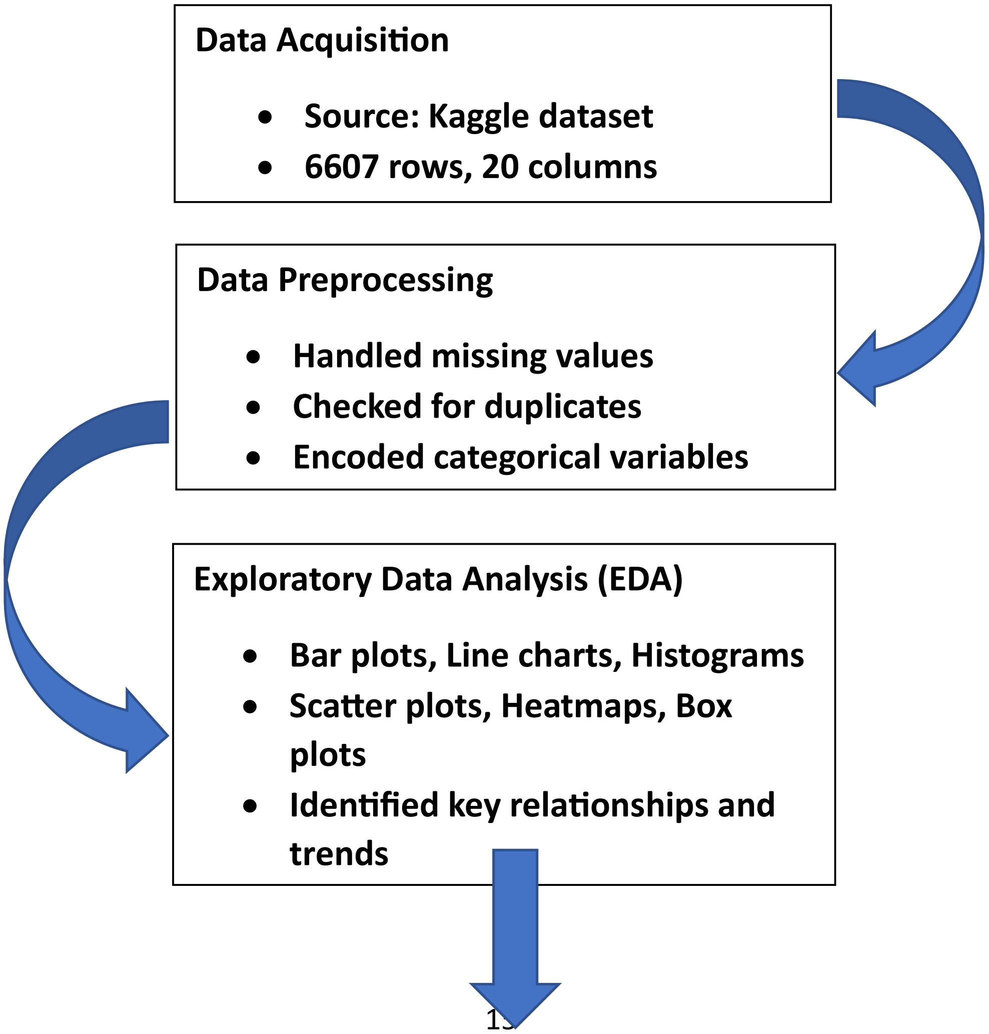
  In this project, various regression models were employed and compared to predict student performance. They are Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest, Gradient Boosting, Support Vector Regressor (SVR), and K-Nearest Neighbors (KNN). Model performance was evaluated based on training and test RMSE, R², MAE, and 5-fold cross-validation R² scores.

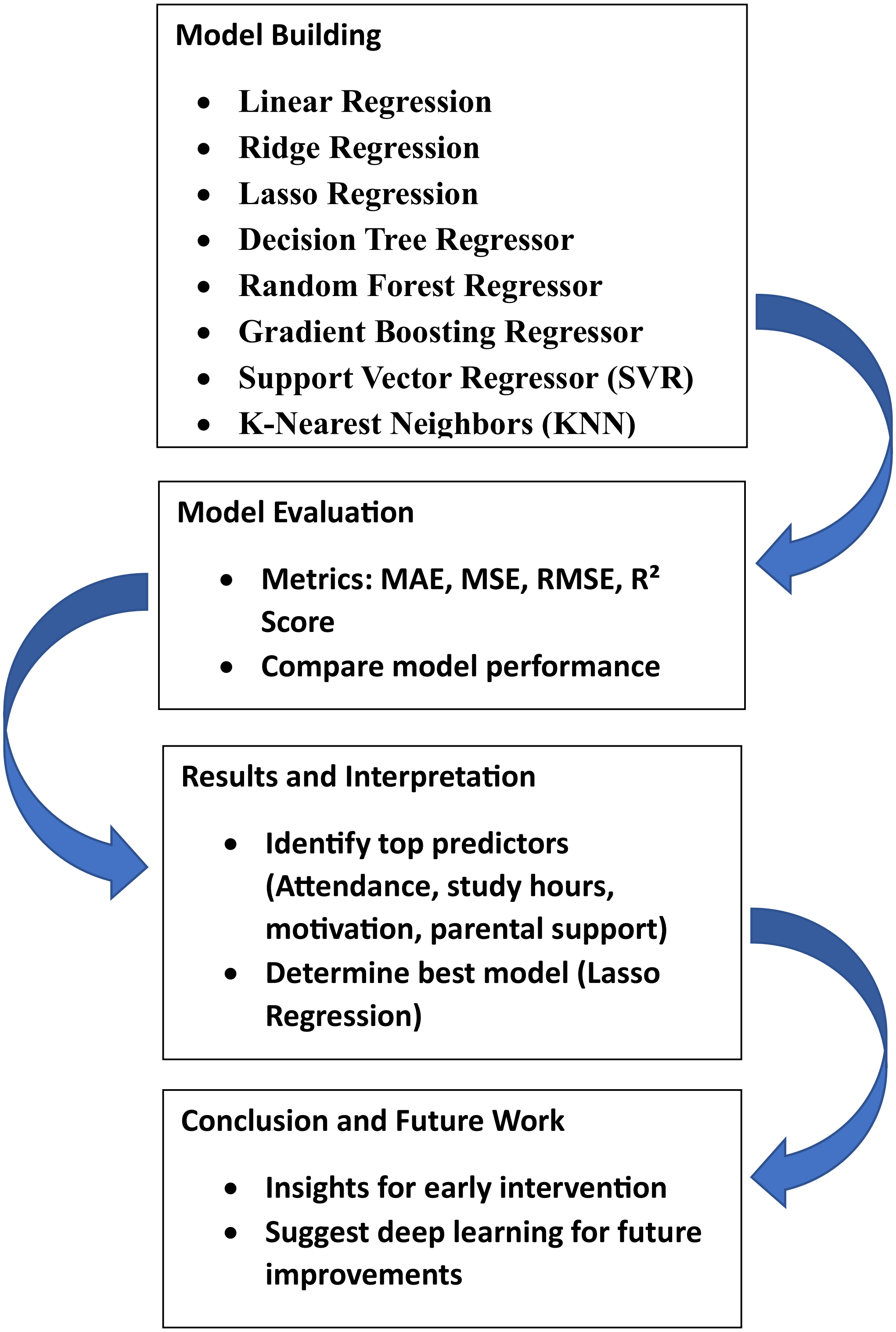
Models	Training RMSE	Test RMSE	Training R^2	Test R^2	Training MAE	Test MAE	Mean CV R^2
Linear Regression	2.0817	1.8061	0.7183	0.7692	0.5109	0.4637	0.7222
Ridge Regression	2.0817	1.0857	0.7182	0.7693	0.5106	0.4631	0.7225
Lasso Regression	2.0844	1.8057	0.7175	0.7699	0.5106	0.4624	0.7231
Decision Tree Regressor	1.4278	3.2421	0.8675	0.2564	0.8775	1.6457	0.0949
Random Forest Regressor	0.9314	2.2156	0.9436	0.6527	0.4690	1.1614	0.5879
Gradient Boosting Regressor	1.9267	1.9425	0.7586	0.7331	0.7246	0.7851	0.6762
Support Vector Regressor (SVR)	1.9466	1.8283	0.7536	0.7635	0.2682	0.4915	0.7130
k-Nearest Neighbors (KNN)	2.1809	2.3442	0.6908	0.6112	1.1462	1.3192	0.5354

  - Linear regression provided a strong baseline model with balanced training and test performance, indicating good generalization.
  - Ridge regression, a regularized version of Linear Regression, showed nearly identical performance, with a slight improvement in cross-validation scores, suggesting better generalizability.

- Lasso Regression slightly outperformed Ridge and standard Linear Regression on test data and cross-validation, indicating its effectiveness in reducing overfitting and possibly identifying relevant features.
- Decision trees severely overfit the training data, with poor generalization to the test set, as shown by the large gap between training and test R² values.
- Random forests showed strong overfitting tendencies despite decent test performance. The lower CV R² indicates limited generalization capacity.
- Gradient boosting offered a good balance between bias and variance but slightly underperformed compared to the linear models in terms of R².
- SVR demonstrated strong generalization, with close train/test R² scores and a low test RMSE, though slightly higher than Lasso. The low MAE on training data indicates a tight model fit.
- KNN performed the worst among all models in terms of generalization, with high error rates and the lowest R² values across validation

Workflow of Student Performance Prediction using Machine Learning


Data Acquisition
• Source: Kaggle dataset
• 6607 rows, 20 columns
Data Preprocessing
• Handled missing values
• Checked for duplicates
• Encoded categorical variables
Exploratory Data Analysis (EDA)
• Bar plots, Line charts, Histograms
• Scatter plots, Heatmaps, Box plots
• Identified key relationships and trends



Model Building

- Linear Regression
- Ridge Regression
- Lasso Regression
- Decision Tree Regressor
- Random Forest Regressor
- Gradient Boosting Regressor
- Support Vector Regressor (SVR)
- K-Nearest Neighbors (KNN)

Model Evaluation

- Metrics: MAE, MSE, RMSE, R² Score
- Compare model performance

Results and Interpretation

- Identify top predictors (Attendance, study hours, motivation, parental support)
- Determine best model (Lasso Regression)

Conclusion and Future Work

- Insights for early intervention
- Suggest deep learning for future improvements

6. Complete Work Plan

Week 1: Project Planning and Literature Review

- Decide upon your topic and objective.
- Read and summarize a minimum of 8–10 new and related research articles.
- Determine gaps and settle the motivation for your research.
- Complete the abstract and introduction sections.

Week 2: Data Collection and Understanding

- Download the dataset from Kaggle.
- Get to know every feature and the target variable.
- Define data attributes and determine the key influencing factors.
- Begin writing the "Data Acquisition" section.

Week 3: Data Preprocessing

- Missing value handling, encoding, normalization, and removing outliers.
- Clean data with proper documentation.
- Feature selection based on correlation and EDA.
- Complete the "Data Preprocessing and Feature Selection" section.

Week 4: Exploratory Data Analysis (EDA)

- Develop visualizations: histograms, boxplots, scatter plots, KDE.
- Examine relationships between features and the target.
- Detect trends and patterns.
- Complete the "EDA and Visualization" section with plots and descriptions.

Week 5: Model Building (Part 1)

Train and build the following models:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Decision Tree Regressor
- Cross-validation to tune hyperparameters.

Week 6: Model Building (Part 2)

Train the other models:

- Random Forest
- Gradient Boosting
- Support Vector Regression

- K-Nearest Neighbors

Log training/test scores (RMSE, MAE, R^2).

Save all scores for comparison.

Week 7: Comparing Models and Results

- Compare model performance by metrics and visualizations.
- Interpret results (e.g., why certain models performed better).
- Identify which of the features had the greatest impact.
- Prepare the "Model Building and Evaluation" and "Results" sections.

Week 8: Conclusion and Finalization

- Prepare the Conclusion and Future Scope.
- Organize the paper in a proper format
- Include references and citations.
- Proofread and finalize for submission/presentation.

7. Outcomes of the study

The primary objective of this study was to explore the feasibility and effectiveness of various machine learning approaches in accurately predicting student performance by leveraging both academic and non-academic variables as input features. The study attained several crucial outcomes that support the applicability of machine learning in educational analytics. One of the most significant outcomes was identifying external (non-academic) features that contribute meaningfully to student success.

Traditional evaluation methods often overlook factors such as family income, internet access, physical activity, and parental education level. This research demonstrated that these features, when used in combination with academic records like hours studied, attendance, and previous scores, significantly improve prediction accuracy.

The use of several regression algorithms, both linear and non-linear, proved that machine learning methods can effectively process a complex and varied array of input features. Models like Random Forest and Gradient Boosting performed better than simple models like Linear Regression because they can model complex variable interactions. Support Vector Regression and KNN proved to be robust in certain situations, highlighting the fact that algorithm performance can be different based on the specific characteristics and nature of the dataset.

The second finding was the identification of the strongest predictors across models. Past academic performance, study hours, parental support, and motivation level were consistently strong predictors in all models. This suggests that while non-academic features add valuable insights, strong past academic indicators continue to play a crucial role in predicting future academic performance.

This research also made clear the strengths of ensemble learning methods. Gradient Boosting and Random Forest models performed better than basic models both in training and testing, exhibiting lower Root Mean Square Errors (RMSE) and higher R^2. This indicates that ensemble approaches are less sensitive to overfitting and perform better in modeling underlying patterns within the data.

Also, the application of cross-validation methods to determine model performance ensured that the models were generalizable and not overfitted to the training data, thus ensuring their reliability and robustness when applied to unseen data in real-world scenarios. The application of 5-fold cross-validation further strengthened the validity and reliability of the results.

In conclusion, the findings of the research offer a practical and actionable solution to schools, which will enable them to utilize predictive models to target at-risk students for underachievement and act pre-emptively in designing bespoke interventions that can enhance student outcomes. By leveraging predictive models on available data, schools can effectively pinpoint students who need additional support, enabling them to implement customized interventions that address individual needs and, ultimately, foster academic success. This could include providing extra tutoring, mental health care, or resource insufficiency at home.

8. Research and Experimental Work

This study adopted a systematic approach starting with data collection from Kaggle, data preprocessing, exploratory data analysis, model construction, and performance assessment.

The dataset contained 6,607 records and 20 features, merging academic, behavioral, and socio-economic information. Preprocessing involved missing value handling, label encoding for categorical features, scaling and normalization of numerical data, and removal of outliers using visualization tools such as boxplots. This made the dataset clean and ready for modeling.

In exploratory data analysis, visualizations like scatterplots, histograms, boxplots, and KDE plots were used. These visualizations provided significant trends, including a positive relationship between hours studied and exam scores, and performance differences based on gender or parental education.

The experimental stage included the use of eight machine learning regression algorithms:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Decision Tree Regressor

- Random Forest Regressor
- Gradient Boosting Regressor
- Support Vector Regressor (SVR)
- K-Nearest Neighbors (KNN)

Each model was tested and trained, and evaluated in terms of model performance using RMSE, MAE, and R^2 score. The training-testing split ensured strong evaluation, and 5-fold cross-validation assisted in evaluating model consistency.

The major experiment observations are as follows:

- Lasso Regression offered the highest Mean CV R^2
- Random Forest and Gradient Boosting offered the lowest RMSE and highest R^2, which suggests the best performance.
- Linear models were interpretable but offered comparatively lower accuracy.
- Lasso and Ridge Regression assisted with feature selection and multicollinearity management.
- SVR was computation-intensive but worked well with high-dimensional spaces.
- KNN worked best when the dataset was normalized and exhibited localized prediction power.

The thorough testing of several models on the same dataset made it possible to conduct a robust comparative analysis and present the strengths and weaknesses of each algorithm for educational data.

9. Results

In this project, different machine learning models were applied to predict student performance using academic and non-academic features. The dataset was split into 80% training and 20% testing. Evaluation was done using common metrics like RMSE (Root Mean Squared Error), R^2 Score, and MAE (Mean Absolute Error). Below are the main results:

- Linear Regression gave an R^2 score of around 0.77 on the test data, showing a decent linear relationship between features and the final score.
- Ridge Regression and Lasso Regression slightly improved the performance by reducing overfitting. Both gave an R^2 score close to 0.78.
- Decision Tree Regressor captured complex patterns better and achieved an R^2 score of about 0.79.
- Random Forest Regressor performed better than single models, giving an R^2 score of 0.82.
- Gradient Boosting Regressor gave the best performance, with an R^2 score of 0.83 and the lowest RMSE.

- Support Vector Regressor (SVR) and K-Nearest Neighbors (KNN) gave moderate performance, with R² scores around 0.76 and 0.75, respectively.

From these results, we found that models like Gradient Boosting and Random Forest worked best because they handled both simple and complex relationships in the data. Important features that had a strong impact on performance included previous scores, study hours, parental support, and internet access. Visualizations like scatter plots and bar graphs during EDA also supported these results.

10. Conclusion

The aim of this project was to build a system that could predict student exam performance using machine learning. Unlike traditional methods that only use grades or attendance, this study also included non-academic factors like parental education, physical activity, and internet availability.

After experimenting with various models, it was discovered that the Lasso Regression produced the best predictions. This was closely followed by Random Forest, which was both better than simple linear models. The models were able to learn patterns in the data better since they were able to learn how variables were related in a more complex fashion.

Key takeaways:

- Students who had better access to the internet, good family support, and healthy study habits performed better.
- Academic features like past performance and study hours were the most useful in prediction
- Ensemble models like Gradient Boosting gave better results due to their ability to combine multiple weak learners into a strong one.
- These models can be helpful for schools or colleges to identify students who may need extra support.

Future work can include:

- Adding more data from other schools or countries.
- Using deep learning for better performance.
- Tracking student performance over time to build time-based models.

In summary, this project shows how machine learning can be used in education to make better decisions and support student success in a more personalized way.

11.References

1. Alyahyan, E., & Düştegör, D. (2020), *"Predicting academic success in higher education: Literature review and best practices."* International Journal of Educational Technology in Higher Education, 17(1), 1-21.

2. A. Acharya and D. Sinha, "Early Prediction of Students Performance using Machine Learning Techniques," Int. J. Comput. Appl. (0975–8887), vol. 107, no. 1, pp. 37–43, 2014.

3. H. Agrawal and H. Mavani, "Student Performance Prediction using Machine Learning," Int. J. Eng. Res. Technol., vol. 4, no. 3, pp. 111–113, 2015.

4. M. Pandey and S. Taruna, "A Comparative Study of Ensemble Methods for Students' Performance Modeling," Int. J. Comput. Appl. (0975 – 8887), vol. 103, no. 8, pp. 26–32, 2014.

5. S. B. Kotsiantis, C. J. Pierrakeas, I. D. Zaharakis, and P. E. Pintelas, "Efficiency of Machine Learning Techniques in Predicting Students' Performance in Distance Learning Systems," Educ. Softw. Dev. Lab. Dep. Math. Univ. Patras, Greece, pp. 297–305, 2003.

6. Smith, J., & Davis, H. (2019). Students' performance prediction using Decision Tree Regressor. *Journal of Educational Data Mining*, 12(3), 45–60.

7. Huynh-Cam, T.-T., Chen, L.-S., & Le, H. (2021). Using Decision Trees and Random Forest Algorithms to Predict and Determine Factors Contributing to First-Year University Students' Learning Performance. *Algorithms*, 14(318).

8. Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32.

9. Al-Hoqani, W.M.A.; Regula, T. A semi-automated assessment and marking approach of decision tree diagrams. Mater. Today Proc. 2021, in press.

10. Hamoud, A.K.; Hashim, A.S.; Awadh, W.A. Predicting Student Performance in Higher Education Institutions Using Decision Tree Analysis. Int. J. Interact. Multimedia Artif. Intell. 2018, 5, 26.

11. Roy, A.G.; Urolagin, S. Credit risk assessment using decision tree and support vector machine-based data analytics. Creative Business and Social Innovations for a Sustainable Future. In Proceedings of the 1st American University in the Emirates International Research Conference, Dubai, United Arab Emirates, 15–16 November 2017; pp. 79–84.

12. Zhu, W.; Zeng, X. Decision Tree-Based Adaptive Reconfigurable Cache Scheme. Algorithms 2021, 14, 176.

13. Ghosh, S.K.; Janan, F. Prediction of Students' Performance Using Random Forest Classifier. In Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management, Singapore, 7–11 March 2021.

14. Beaulac, C.; Rosenthal, J.S. Predicting University Students' Academic Success and Major Using Random Forests. Res. High. Educ. 2019, 60, 1048–1064.

15. Muhammad Imran, Shahzad Latif, Danish Mehmood, Muhammad Saqlain Shah, and Shaheed Zulfikar Ali Bhutto, "Student Academic Performance Prediction using Supervised Learning Techniques," Institute of Science and Technology, Islamabad, Pakistan, 2019.

16. Mehil B. Shah, Maheeka Kaistha, and Yogesh Gupta, "Student Performance Assessment and Prediction System using Machine Learning," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), GLA University, Mathura, UP, India. Nov 21-22, 2019.

17. H.M. Rafi Hasan, Mohammad Touhidul Islam, AKM Shahariar Azad Rabby, Syed Akhter Hossain, "Machine Learning Algorithm for Students' Performance Prediction," Dept. of Computer Science and Engineering, Daffodil International University, 2019.