Introduction and Problem Definition

The dataset to be examined in this report is the "Students Performance in Exams" dataset, which can be found at:

https://www.kaggle.com/datasets/spscientist/students-performance-in-exams?resource=download

The primary objective of this study is to predict a student's **math score** based on the best-performing predictive variables in this dataset. To achieve this goal, we will apply several machine learning techniques to select the most important features and evaluate the performance of different prediction algorithms to identify the best approach for predicting math scores.

This report will detail the methods used, present the findings, and disclose the best-performing model's metrics. To begin, we will conduct an in-depth exploratory data analysis to gain a comprehensive understanding of the dataset and its features. This will enable us to identify any patterns, trends, or relationships that might aid in the prediction.

Next, let's dive into our exploratory data analysis.

Exploratory Data Analysis

DATA OVERVIEW

This dataset consists of the following columns, or features. Here is a brief explanation of each:

Feature	Description	Data Type
Gender	Student's gender	Boolean/String
Race/Ethnicity	Student's ethnic group from groups A-E	String
Parental level of education	Describes the parent of the student's level of education	String
Lunch	Describes the lunch program of the student, e.g., standard/reduced/free	String
Test preparation course	If the student completed a test prep course, e.g., none/completed	String
Math score	Exam math score	Integer
Reading score	Exam reading score	Integer
Writing score	Exam writing score	Integer

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
..	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

Excluding the target column, **math score**, Our dataset comprises 8 columns in total. There are 5 non-numeric columns (**gender, race/ethnicity, parental level of education, lunch**, and **test preparation course**) and 3 numeric columns (**math score, reading score,** and **writing score**).

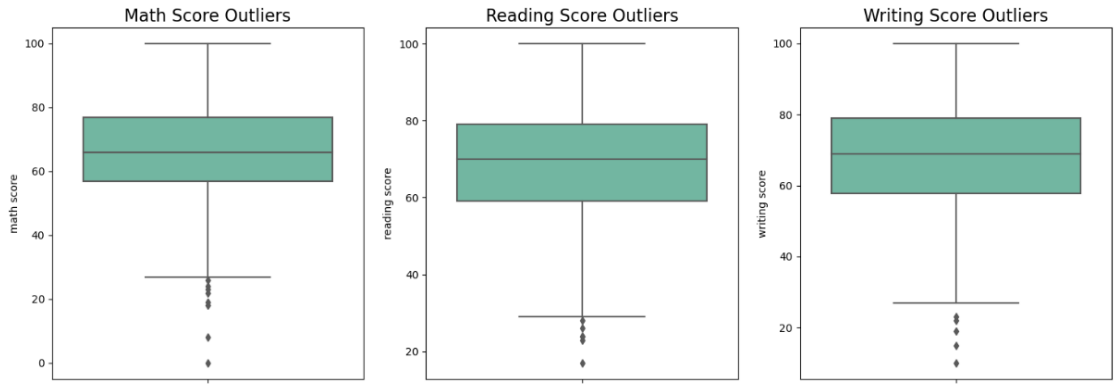
It is essential to note that the non-numeric columns will need to be converted to numeric values or encoded appropriately to be utilized in the machine learning algorithms. With this information in mind, let's delve deeper into the dataset and examine how the values interact with the target variable, **math score**.

FEATURES OF FOCUS

First, let's examine the heatmap to understand how various features in the dataset correlate with **math score**, our target variable. The heatmap provides a visually efficient and reader-friendly overview of the relevant data attributes and their relationships with the target.

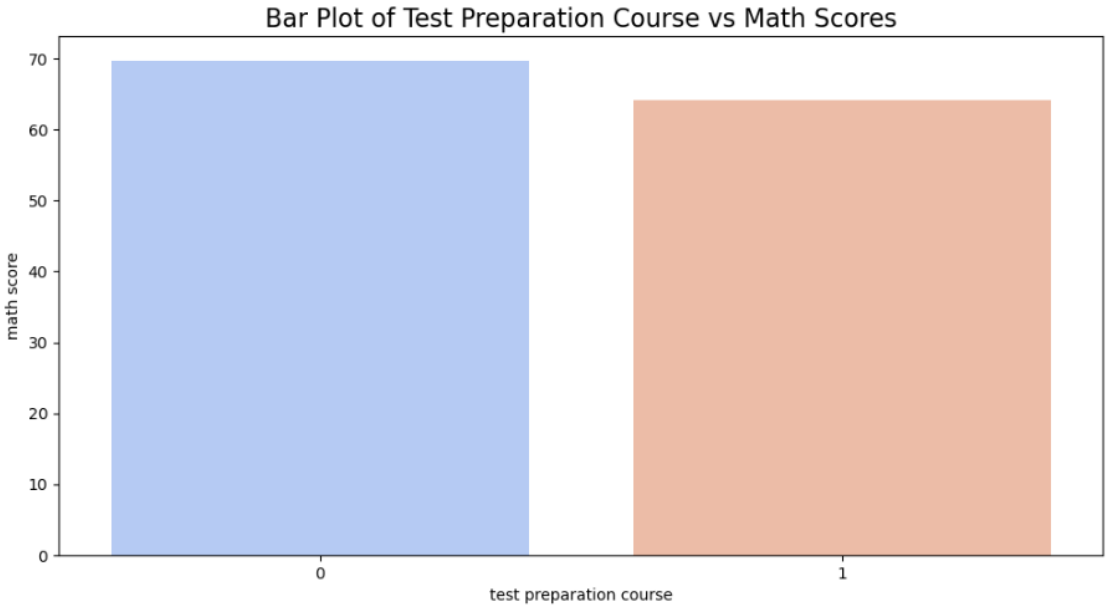
From the heatmap, we can observe that the features **writing score** and **reading score** have strong positive correlations with **math score**. On the other hand, **test preparation course** has a stronger negative correlation. These features will likely have a significant impact on predicting **math score**, and we'll pay close attention to them throughout our analysis.

Next, let's examine box plots for **math score**, **reading score**, and **writing score** to identify any potential outliers.



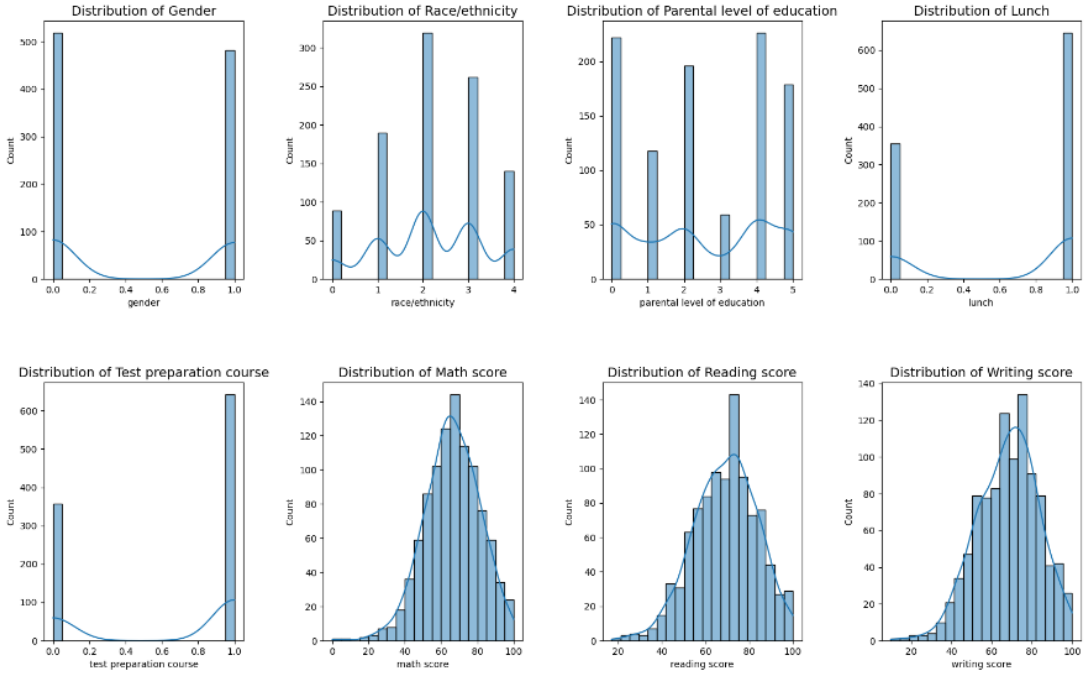
The box plots reveal that **writing score** and **reading score** have similar distributions with relatively few outliers. This similarity could explain why these features are strongly correlated with **math score**, as their distributions might reflect a common underlying relationship with the target variable.

Let's explore the bar plot of **test preparation course** against **math scores**.



This bar plot reveals an interesting insight: students who completed a test preparation course generally have higher math scores compared to those who did not. This trend suggests that the test preparation course might be a useful feature to consider when predicting **math score**.

Lastly, let's look at the data distribution graphs for all features in the dataset, with a focus on the important features identified earlier.



These distribution graphs provide an overview of the spread and patterns of the data. For instance, we can see that **math score**, **reading score**, and **writing score** have roughly normal distributions, while the non-numeric features exhibit varying degrees of skewness. This information will be valuable when preprocessing the data and selecting the appropriate machine learning algorithms.

In conclusion, our analysis has identified **writing score**, **reading score**, and **test preparation course** as the features of focus. We'll continue to investigate their relationships with **math score** in subsequent sections to build a robust predictive model.

Data Treatment

In preparing the dataset for modeling, several steps were taken to ensure that the data was suitable for processing. First, there were no missing values to deal with, as confirmed by the output of the **df.describe().T** function, which showed that the dataset was complete without any missing data.

	count	mean	std	min	25%	50%	75%	max
gender	1000.0	0.482	0.499926	0.0	0.00	0.0	1.0	1.0
race/ethnicity	1000.0	2.174	1.157179	0.0	1.00	2.0	3.0	4.0
parental level of education	1000.0	2.486	1.829522	0.0	1.00	2.0	4.0	5.0
lunch	1000.0	0.645	0.478753	0.0	0.00	1.0	1.0	1.0
test preparation course	1000.0	0.642	0.479652	0.0	0.00	1.0	1.0	1.0
math score	1000.0	66.089	15.163080	0.0	57.00	66.0	77.0	100.0
reading score	1000.0	69.169	14.600192	17.0	59.00	70.0	79.0	100.0
writing score	1000.0	68.054	15.195657	10.0	57.75	69.0	79.0	100.0

Second, as shown in the box plots of writing, reading, and math scores in the EDA section, the dataset contains outliers. However, these outliers are important parts of the dataset, as they represent genuine data points that provide valuable insights into student performance. Removing them would lead to loss of critical information. Therefore, no action is taken to eliminate or modify these outliers.

Third, the dataset includes several non-numeric columns, which need to be converted to numeric values to allow for data processing. To accomplish this, the **LabelEncoder** function is used, which iterates through each column in the dataset, identifies non-numeric columns, and converts them to numeric values using the **fit_transform** method. This ensures that all columns in the dataset are in a numeric format, which is essential for modeling purposes.

In summary, the data treatment process involved handling missing values, retaining important outliers, and converting non-numeric columns to numeric values, ensuring a prepared and suitable dataset for modeling and analysis.

Model Development and Tuning

FEATURE SELECTION

In this section, we discuss the development, tuning, and comparison of models using different feature selection methods. Three feature selection algorithms were employed to identify the most relevant features for predicting math scores in the dataset. These algorithms include **Recursive Feature Elimination (RFE),**

Forward Feature Selection (FFS), and **Chi-Square Test**. Below is a table of the selected features by each feature selection algorithm:

Feature Selection Algorithm	Selected Features
RFE	gender, lunch, test preparation course
FFS	lunch, reading score, writing score
Chi-Square	parental level of education, reading score, writing score

Based on the features selected by these algorithms, the statistically significant features from the OLS model summary, and insights from the EDA section, the final chosen features were **writing score, reading score**, and **test preparation course**.

With these selected features, multiple stand-alone models were developed, tuned, and compared.

MODEL TUNING

The Neural Network

The second model, a neural network model, was created using **1 input layer**, **2 hidden layers**, and **1 output layer,** with **10 neurons** each in the first 3 layers, and a single neuron in the output layer. The model is compiled using mean squared error as the loss function and **Stochastic Gradient Descent (SGD)** as the **optimizer**.

Grid searching is a method for hyperparameter tuning, where a search space of possible hyperparameter values is defined, and an exhaustive search is performed to find the best hyperparameter combination that minimizes the model's loss function. Hyperparameter tuning is essential to optimize a model's performance, as different hyperparameter values may result in significant differences in model performance.

Grid searching was performed on the neural network model to find the optimal hyperparameter values for the number of layers, neurons, learning rate, activation function, and kernel initializer. **KerasRegressor** was used to wrap the Keras model as a scikit-learn compatible estimator, and **GridSearchCV** from the scikit-learn library was used to perform the grid search on the Keras model. Grid searching showed that optimal parameters were **0.00001** for the learning rate**, RELU** for the activation function, and **he_uniform** as the kernel initializer.

The Stacked Model

In the stacked model, several base models are used, including **Ridge, Lasso, DecisionTreeRegressor, AdaBoostRegressor, RandomForestRegressor, GradientBoostingRegressor**, and **SVR**. The hyperparameters of these base models were also tuned using grid searching using parameter grids for each base model. **Linear Regression** was used for the stacked model and tuned as well.

Model Evaluation

Base Models for the Stacked Model:

Model	RMSE Avg.	RMSE Std. Dev.	MAE Avg.	MAE Std. Dev.	R-squared Avg.	R-squared Std. Dev.
Ridge	8.579	0.255	7.03	0.286	0.669	0.037
Lasso	8.578	0.257	7.039	0.287	0.669	0.037
Decision Tree Regressor	11.759	0.326	9.394	0.292	0.378	0.069
AdaBoost Regressor	8.873	0.279	7.209	0.275	0.646	0.041
Random Forest Regressor	8.91	0.188	7.243	0.256	0.644	0.027
Gradient Boosting Regressor	9.176	0.231	7.393	0.264	0.622	0.035
SVR	9.288	0.347	7.43	0.385	0.614	0.02

Model Comparison Table:

Model	RMSE Avg.	RMSE Std. Dev.	MAE Avg.	MAE Std. Dev.	R-squared Avg.	R-squared Std. Dev.
Model 1 (OLS)	8.592	0.162	7.046	0.118	0.667	0.017
Model 2 (Neural Network)	13.375	0.499	11.064	0.413	0.199	0.082
Model 3 (Stacked Model)	8.431	0.308	6.877	0.291	0.68	0.039

Out of the base models that make up the stacked model, the **Lasso Regression** model performs the best with the best average RMSE, though there are other models who have less fluctuation or perform better on the other metrics.

Based on the RMSE values, **Model 3 (Stacked Model)** performs the best with an RMSE of 8.426, which is slightly better than Model 1 (OLS) with an RMSE of 8.568. Model 2 (Neural Network) has the highest RMSE of 13.375, indicating it performs the worst out of the three models.

Conclusion

We explored three different models to predict students' math scores based on their writing scores, reading scores, and test preparation courses. The models analyzed were Ordinary Least Squares (OLS), a Neural Network model, and a Stacked Model consisting of various base models, including Ridge, Lasso, Decision Tree Regressor, AdaBoost Regressor, Random Forest Regressor, Gradient Boosting Regressor, and Support Vector Regressor.

A comprehensive comparison of the three models was conducted using the average Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared metrics. The results showed that the **Stacked Model** had the lowest average RMSE (8.431), followed by Model 1 (OLS) with an average RMSE of 8.592, and Model 2 (NN) with an average RMSE of 13.375. In addition, the Stacked model had the **highest R-squared value (0.68)**, indicating that it provided the best fit to the actual data.

Overall, the Stacked Model demonstrated the best performance among the three models, with the lowest RMSE and the highest R-squared value. This indicates that the Stacked model is the most suitable choice for predicting students' math scores based on the given features. It is important to note that the choice of base models and their hyperparameters may impact the performance of the Stacked Ensemble model. Future work could explore alternative base models or optimization techniques to further improve the predictive performance of the ensemble model.