STUDENT PERFORMANCE INDICATOR

CSEN 240 - Machine Learning Project
Spring 2024

AUTHORS:
Naveena Avula (07700009231)
Sri Sai Saketh Chillapalli (07700000089)

UNDER THE GUIDANCE OF

Dr. ALEX SUMARSONO

AUDIENCE

This project speaks to a wide audience interested in utilizing machine learning for improved student performance analysis. Here is the breakdown of the main groups:

Educator and Educational Leaders: This project will give valuable insights to teachers and school leaders who are seeking to understand factors that influence a child's performance. The findings can therefore be used to inform curriculum development and strategies for personalized learning as well as early intervention programs.

Data Scientists and Researchers: In this project, the application of different ML algorithms, including Linear Regression, Lasso, Ridge, KNN, Decision Tree, Random Forest, XGBoost, CatBoost, and AdaBoost, has been used to predict student performance. The results of the prediction can be valuable for researchers who are trying to get a novel approach in educational data analysis.

Educational Technologists and Software Developers: The findings of the project may be used as guidelines when developing educational technology tools and learning management systems that use ML for personalized learning and student performance analysis.

Table of Contents

Introduction

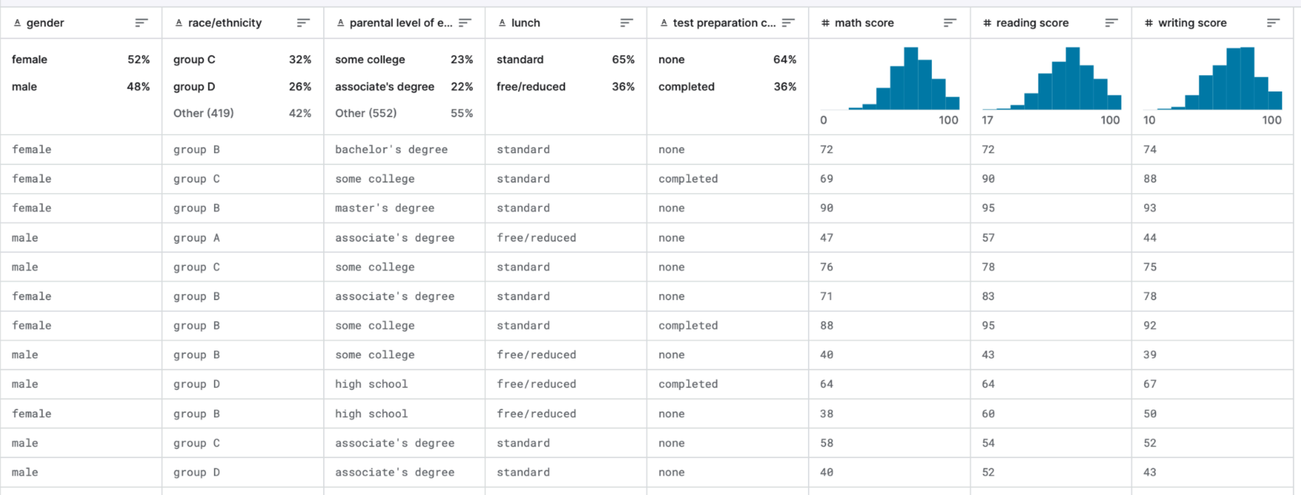
Welcome to the research project report on Student performance Indicator. It is very important for any means of improving educational outcomes. The present study explores the potential of a wide range of machine learning algorithms: **Linear Regression, Lasso, Ridge, K-Nearest Neighbors Regressor, Decision Tree, Random Forest Regressor, XGBoost Regressor, CatBoosting Regressor, and AdaBoost Regressor** in determining student performance indicators. Through comparison of how these models perform in the prediction of student performance, we identify the best way to identify the superior approach for institutions in educational practice that are to use machine learning for data-driven decision-making and personalized learning strategies.

Problem statement

We desire to understand how the student's performance (test scores) is affected by variables such as Gender, Ethnicity, Parental level of education, Lunch student had before test and Test preparation course. We plan to achieve this by data collection, performing data checks, **exploratory data analysis,data pre-processing, model training and then choosing the best model**.

Dataset Intended to be used for this project

https://www.kaggle.com/datasets/spscientist/students-performance-in-exams?datasetId=74977
There are 1000 rows and 8 columns in the data.



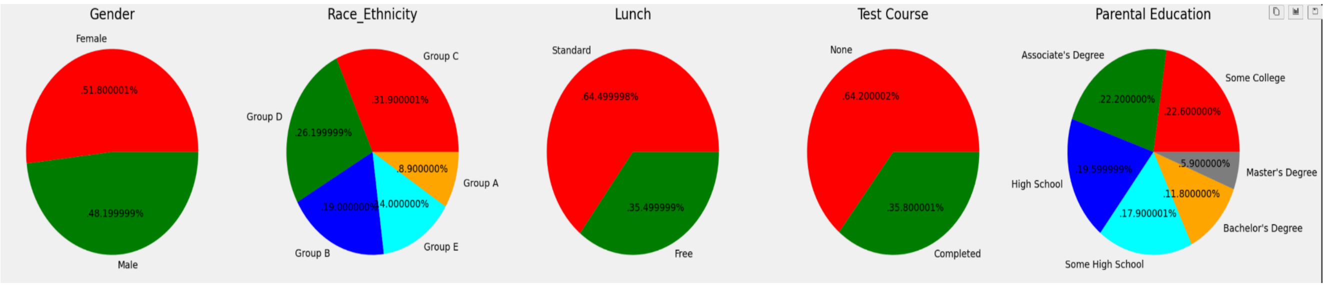
gender	race/ethnicity	parental level of e...	lunch	test preparation c...	# math score	# reading score	# writing score
female 52% male 48%	group C 32% group D 26% Other (419) 42%	some college 23% associate's degree 22% Other (552) 55%	standard 65% free/reduced 36%	none 64% completed 36%	0 100	17 100	10 100
female	group B	bachelor's degree	standard	none	72	72	74
female	group C	some college	standard	completed	69	90	88
female	group B	master's degree	standard	none	90	95	93
male	group A	associate's degree	free/reduced	none	47	57	44
male	group C	some college	standard	none	76	78	75
female	group B	associate's degree	standard	none	71	83	78
female	group B	some college	standard	completed	88	95	92
male	group B	some college	free/reduced	none	40	43	39
male	group D	high school	free/reduced	completed	64	64	67
female	group B	high school	free/reduced	none	38	60	50
male	group C	associate's degree	standard	none	58	54	52
male	group D	associate's degree	standard	none	40	52	43

Sample Data Set

Dataset Information

Below is the information of the columns, content of the columns and different categories present in each column

Column name	Content of the column	Categories in each column
Gender	Gender of students	Male, Female
Race/ethnicity	Ethnicity of students	Group A, B, C, D, E
Parent level of education	Parent’s final education	Bachelor’s degree, some college,master's degree,associate's degree,high school
Lunch	Having lunch before test	Standard, free/reduced
Test preparation course	Complete or not complete before test	None, completed
Math score	Score in math test	Numbers in the Range 0-100
Reading score	Score in reading test	Numbers in the Range 0-100
Writing score	Score in writing test	Numbers in the Range 0-100



Distribution of Categories for each Feature

Required Packages

Pandas: Built on top of NumPy, Pandas is a superhero for data analysis. It provides DataFrames, which is a primary data structure that combines labeled tables with both rows and columns, and intuitive tools for data cleaning, manipulation, and analysis.

NumPy: The fundamental package for scientific computing with Python. NumPy provides support for efficient arrays that can be used for performing operations on a large collection of elements. It is considered most useful for handling large datasets because of its speed and memory optimization in relation to other Python features.

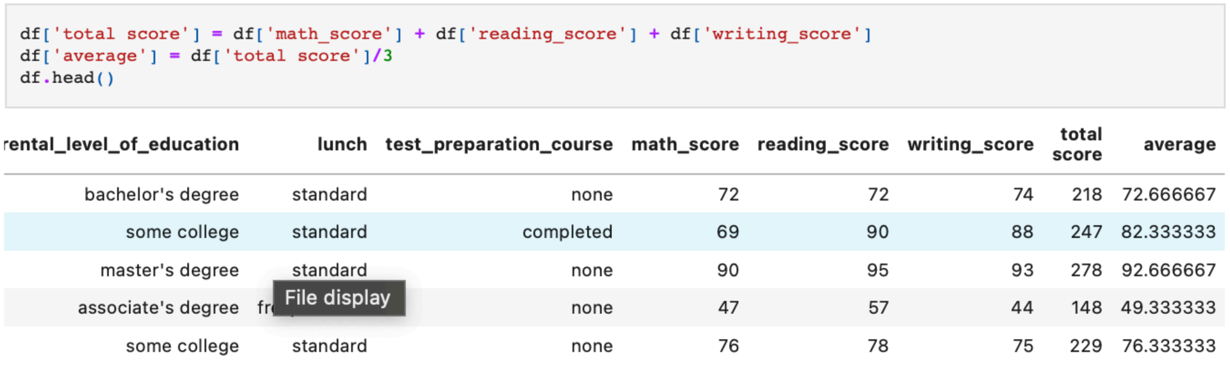
Matplotlib: A Python 2D plotting library that produces publication-quality figures in a variety of formats and interactive environments across platforms. It's a flexible library able to plot many types of graphs, starting from simple line plots to beautiful ones like heat maps. It lets you explore and present the data in an informative way.

Seaborn: The Seaborn library is built on top of Matplotlib. It aims to make visualization a central part of exploring and understanding data. It provides a high-level interface for creating beautiful and informative visualizations commonly used in data science. Seaborn also plays very well with Pandas DataFrames, making data visualization in Python very smooth.

Jupyter Notebook/Colab: puts code, text, and visualizations in one place, is given a powerful cloud makeover with Google Colab. This free platform allows you to run Jupyter Notebooks right from your browser, without the need for setup headaches. The free hardware acceleration using GPUs for machine learning in Colab can easily be accessed and is immediately compatible with Google Drive for easy storage and collaboration in your data science projects.

Data Checks Performed

Validate Missing values: Check which columns have empty cells and their effects.
Check Duplicates: Identifying and eliminating duplicate data points to prevent redundancy.
Check data type: Make sure that each column contains the right data type—for example, numbers or text.
Check values: Find out the amount of unique values within each column to get an idea of the variety of data.
Check stats: Summarize the central tendencies (mean, median) and spread (standard deviation) of numeric data.
Check categories: Examine the various categories that exist in each of your categorical columns.
No duplicates present: All data points are unique, meaning no rows are identical.
No missing values: Every data point has a value recorded, there are no blank entries.
Total score and Average columns addition: For using in exploratory data analysis.



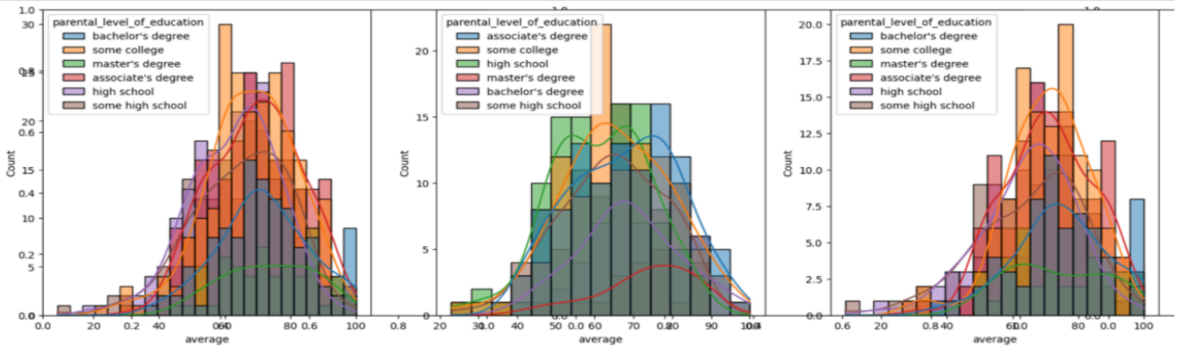
```
df['total score'] = df['math_score'] + df['reading_score'] + df['writing_score']
df['average'] = df['total score']/3
df.head()
```

rental_level_of_education	lunch	test_preparation_course	math_score	reading_score	writing_score	total score	average
bachelor's degree	standard	none	72	72	74	218	72.666667
some college	standard	completed	69	90	88	247	82.333333
master's degree	standard	none	90	95	93	278	92.666667
associate's degree	fr	none	47	57	44	148	49.333333
some college	standard	none	76	78	75	229	76.333333

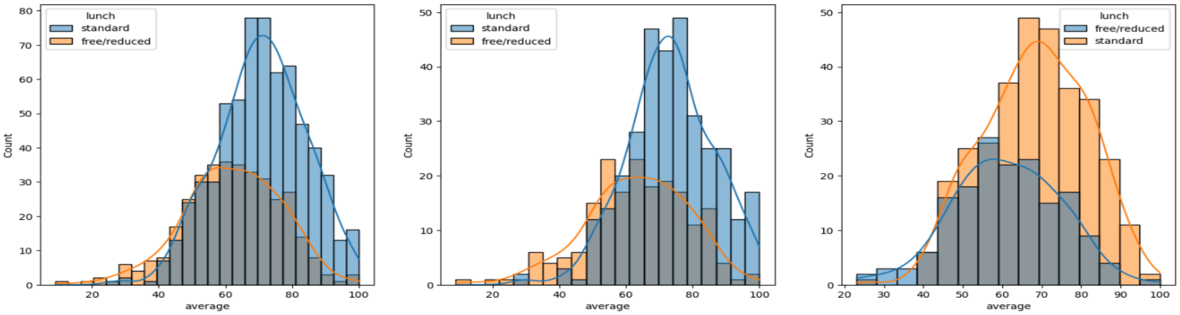
Exploratory Data Analysis (EDA)

It is the detective work of data science, in which you investigate your dataset to understand its composition. This involves inspecting for missing values and duplicates, as well as the assurance that data types are appropriate. After that, you delve into the single variables (univariate analysis) with summary statistics and an exploration of value distribution. The investigation of relationships between pairs of variables (bivariate analysis) will follow, conducted with the use of visualizations, such as scatter plots and cross tabulations. Throughout the EDA, you'll develop histograms, box plots, and heatmaps to uncover patterns and identify outliers. By the end, you'll have a profound understanding of your data, together with its strong and weak points, be armed with a set of initial hypotheses, and be well-prepared for further analysis—effectively transforming raw data into a springboard for meaningful insights.

Below are the Exploratory Data Analysis insight images analyzing the effect of different features on the test scores.

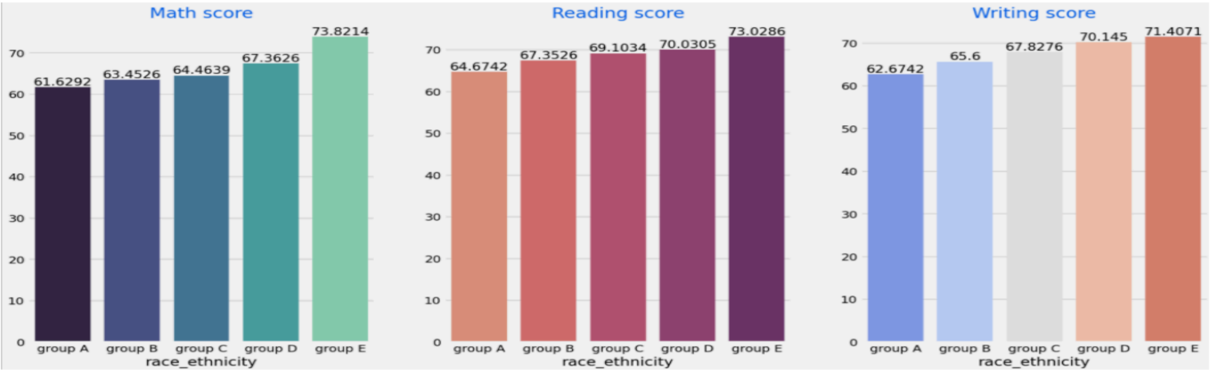


- 1st plot - In general parent's education don't help student perform well in exam. -
- 2nd plot - shows that parents whose education is of associate's degree or master's degree their male child tend to perform well in exam -
- 3rd plot - we can see there is no effect of parent's education on female students.

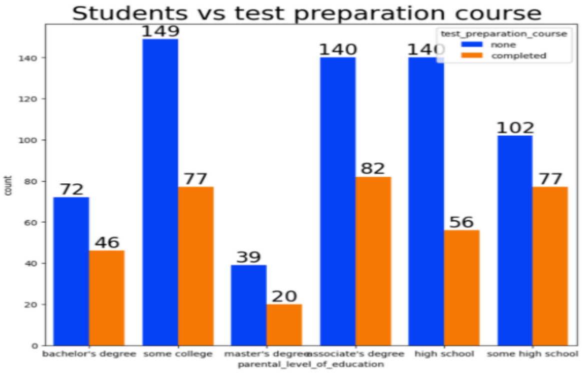


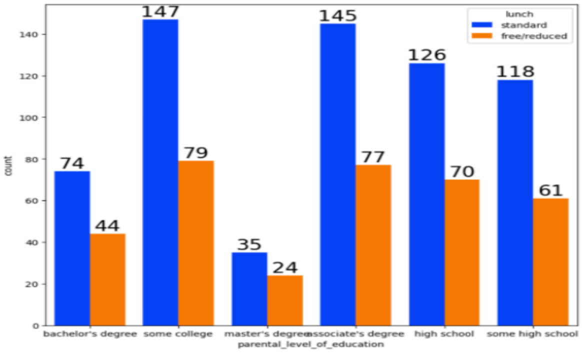
Insights

- Standard lunch helps perform well in exams.
- Standard lunch helps perform well in exams be it a male or a female.

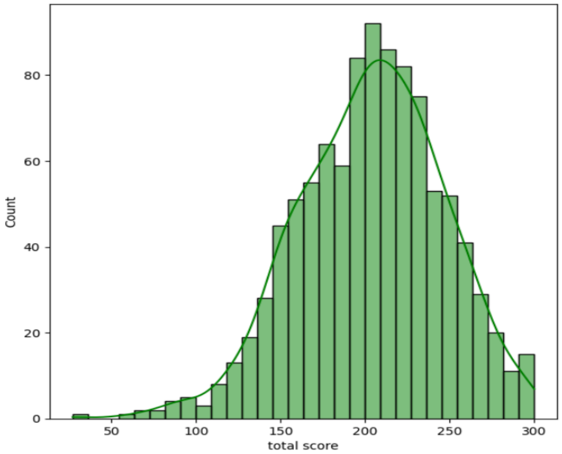


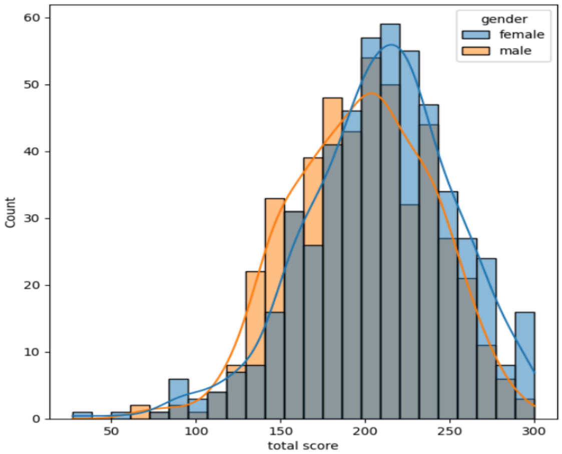
- Group E students have scored the highest marks.
- Group A students have scored the lowest marks.
- Students from a lower Socioeconomic status have a lower avg in all course subjects





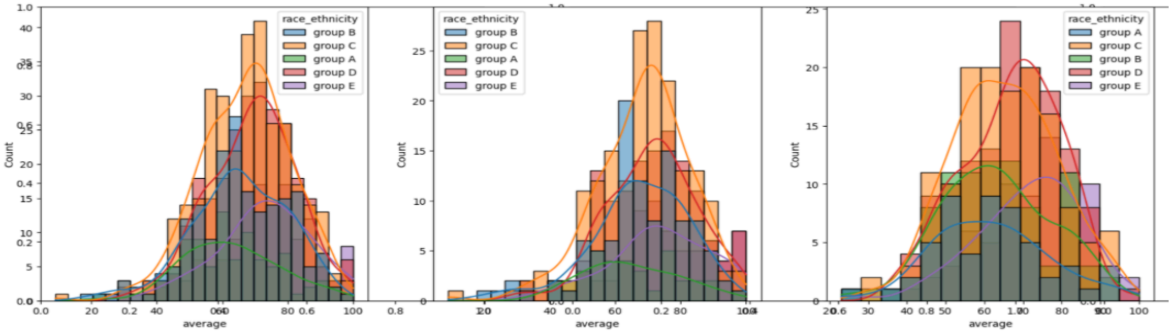
- Students who have completed the Test Preparation Course have scores higher in all three categories than those who haven't taken the course





Insights

- Female students tend to perform well then male students.



- Students of group A and group B tends to perform poorly in exam.
- Students of group A and group B tends to perform poorly in exam irrespective of whether they are male or female

Models Used

Linear Regression: It is considered to be the base algorithm for many regression techniques. Linear regression is an approach that tries to fit a straight line that explains the relationship of a set of independent variables (features) with one dependent variable (target). It is considered one of the most used and well-understood models. The coefficients of the equation show the strength and direction of the influence each feature has on the target variable. However, linear regression assumes a linear relationship between features and the target, something which doesn't always hold true for real-world data. Besides, it can be prone to overfitting if there is a large number of features or irrelevant features in the dataset.

Lasso and Ridge Regression: The main problem in linear regression is overfitting. Overfitting occurs when a model becomes too focused on capturing the noise in the training data and loses its ability to generalize well on unseen data. In this regard, Lasso regression offers a solution because it introduces a penalty term that results in some coefficients shrinking toward zero. The coefficients associated with the irrelevant features become equal to zero, effectively removing the features from the model. This induces sparsity and enhances model generalization. On the other hand, ridge regression applies a penalty that forces all the coefficients to shrink towards zero, but not to zero themselves. This approach retains all the features but reduces their influence, which leads to a smoother and more generalizable model.

K-Nearest Neighbors Regressor: This algorithm works on the principle of similarity for prediction. For the new data point, it looks at k nearest neighbors from the training data that are most similar, based on a distance metric. Subsequently, the predicted value for the new point is the average of target variable values taken from its k nearest neighbors. KNN is a non-parametric method because it doesn't make any assumptions about the underlying data distribution. It's simple to understand and use, and it can learn non-linear relations to some extent. On the downside, it could be a very slow algorithm when dealing with large sets of data, and the choice of the hyperparameter k might greatly influence the performance. Another con would be the presence of irrelevant features that might skew the distance calculations and, therefore, the accuracy of predictions.

Decision Tree Regressor: This method creates a tree-like structure to predict the target variable. It starts with the entire data set, beginning at the root node of the tree, and recursively divides the data into further smaller branches based on the decision rules applied to the features. The procedure defines the most important splits that divide the data points with different values of the target variable. A prediction for a new point is obtained by traversing the tree according to the rules that best agree with the features of the new point, until a leaf node with a predicted value is reached. Since the decision rule is intuitive and can be easily visualized, it is interpretable. Moreover, it is a good performer in capturing non-linear relationships. It can be sensitive to small changes, which result in predicting instability. In addition, if the tree grows very deep, it might overfit to the data.

Random Forest Regressor: This method harnesses the power of ensemble learning to overcome limitations of a single decision tree. It builds a number of decision trees, each one on a randomly selected subset of features and on a random sample of the training data with replacement. This method of training, known as bootstrapping, is said to weaken the tendency of trees to overfit data. In making predictions with this ensemble model, the new data point is used on all the decision trees to get the final average of all the predicted results. While random forests offer higher accuracy and robustness compared to single decision trees, they come at the cost of reduced interpretability as the final prediction is a combination of multiple, potentially complex, decision trees.

XGBoost Regressor and CatBoost Regressor: These are advanced ensemble methods based on gradient boosting. These models iteratively build decision trees, focusing on areas where previous trees made errors. They are highly flexible and powerful for complex datasets, but interpretability is limited.

AdaBoostRegressor: Another ensemble method for combining predictions from weak learners is AdaBoost Regressor, which often includes simple decision trees. It focuses on improving predictions for the data points where the prior models have performed poorly. High in power, though low in interpretability.

Precautions while modeling

Overfitting

- **High Variance:** Overfitting leads to high variance because the model becomes overly sensitive to the specific training data it's exposed to. This sensitivity stems from the model capturing irrelevant details or noise in the training data.
- **Low Bias:** Overfitting models often have low bias because they closely fit the training data. However, this fit comes at the cost of generalizability. The model essentially memorizes the training data without learning the underlying relationships between features and the target variable.

Underfitting

- **High Bias:** Underfitting results in high bias because the model is too simple and cannot capture the complexities of the data. This bias stems from limitations in the model's capacity to learn the relationships between features and the target variable, often due to an underrepresentation of the underlying function.

- **Low Variance:** Underfitting models tend to have low variance because they are not overly sensitive to the specifics of the training data. However, this comes at the cost of poor performance on both the training and testing data. The model fails to learn the true patterns in the data, leading to consistently large errors.

RMSE, MAE and R^2

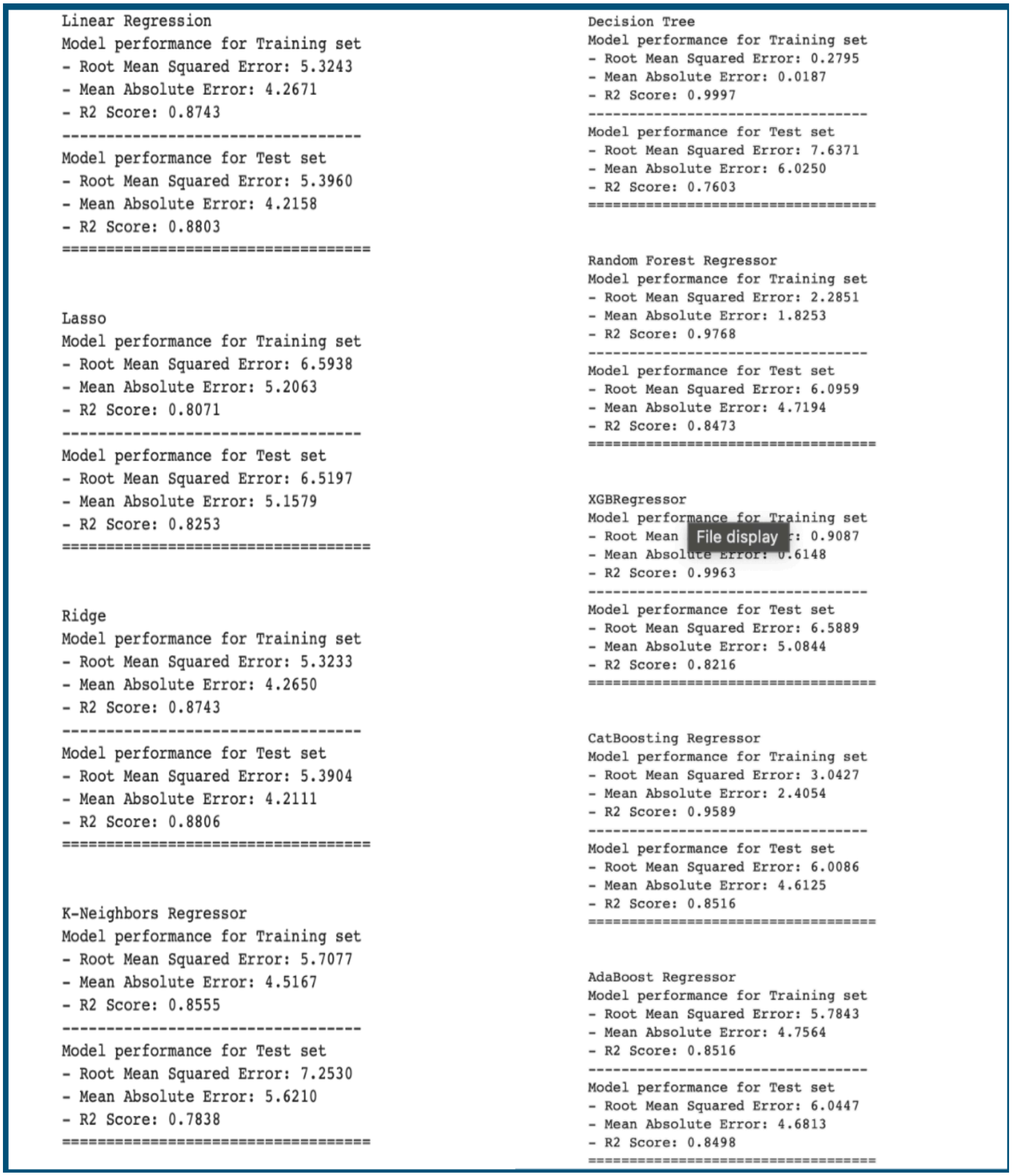
Root Mean Squared Error (RMSE): Think of RMSE as the error between your predicted values and the true values being distances from a bullseye. RMSE squares those distances, takes the average of those squared distances, and then takes the square root to get a unit back on the scale of your target variable. It gives heavier weight to larger errors, which can be valuable in many applications since larger errors may be particularly undesirable. Lower RMSE values mean a better fit.

Mean Absolute Error (MAE): This measures the average magnitude of the errors, but it makes no specification about the direction an error points. It simply computes the average of the absolute difference between predicted and actual values. MAE is much easier to understand than RMSE, since it is the average error in the same units as your target. Lower MAE values mean a better fit.

R-squared (R^2): The metric describes the proportion of variance in the dependent variable that is predictable from the independent variable(s). Think about it as the extent to which your model: drawing a line between all the data points and drawing a different line between all the data points, which represents the model's predictions. For example, your R^2, which ranges from 0 to 1, with a better model fit being closer to 1. However, R^2 doesn't see the real errors' magnitude, hence misleading you if the model just predicted an average value close to the mean. It is best used with other metrics so that you get a fuller picture.

Results

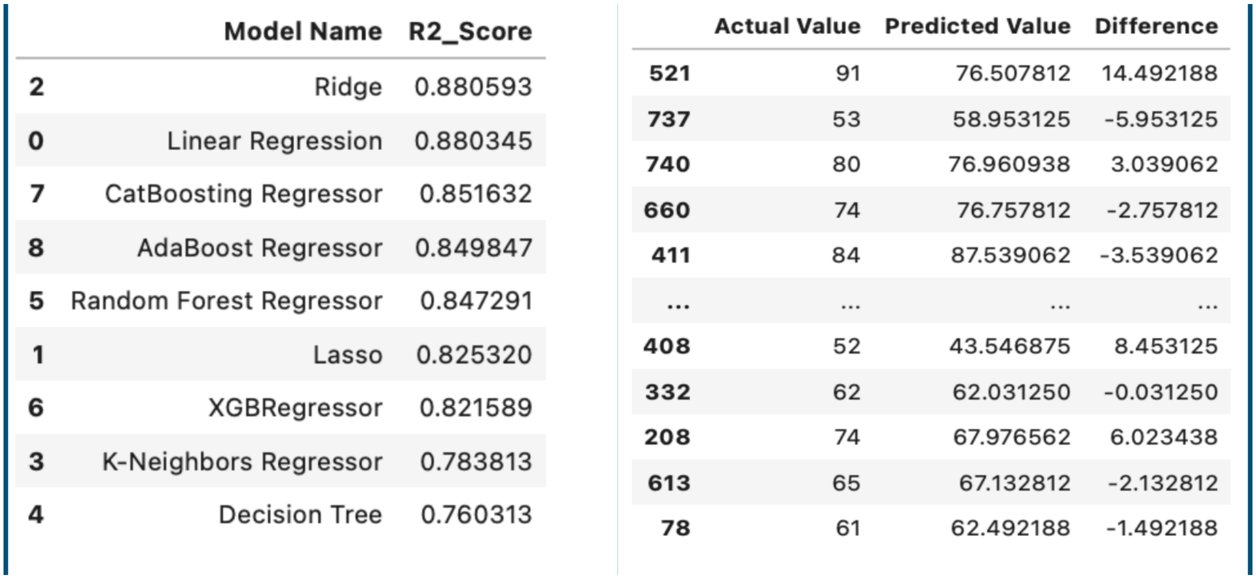
The following image shows the **RMSE, MAE and R^2** for each model



```
Linear Regression
Model performance for Training set
- Root Mean Squared Error: 5.3243
- Mean Absolute Error: 4.2671
- R2 Score: 0.8743
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 5.3960
- Mean Absolute Error: 4.2158
- R2 Score: 0.8803
===================================


Lasso
Model performance for Training set
- Root Mean Squared Error: 6.5938
- Mean Absolute Error: 5.2063
- R2 Score: 0.8071
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 6.5197
- Mean Absolute Error: 5.1579
- R2 Score: 0.8253
===================================


Ridge
Model performance for Training set
- Root Mean Squared Error: 5.3233
- Mean Absolute Error: 4.2650
- R2 Score: 0.8743
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 5.3904
- Mean Absolute Error: 4.2111
- R2 Score: 0.8806
===================================


K-Neighbors Regressor
Model performance for Training set
- Root Mean Squared Error: 5.7077
- Mean Absolute Error: 4.5167
- R2 Score: 0.8555
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 7.2530
- Mean Absolute Error: 5.6210
- R2 Score: 0.7838
===================================
```

```
Decision Tree
Model performance for Training set
- Root Mean Squared Error: 0.2795
- Mean Absolute Error: 0.0187
- R2 Score: 0.9997
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 7.6371
- Mean Absolute Error: 6.0250
- R2 Score: 0.7603
===================================


Random Forest Regressor
Model performance for Training set
- Root Mean Squared Error: 2.2851
- Mean Absolute Error: 1.8253
- R2 Score: 0.9768
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 6.0959
- Mean Absolute Error: 4.7194
- R2 Score: 0.8473
===================================


XGBRegressor
Model performance for Training set
- Root Mean [illegible]r: 0.9087
- Mean Absolute Error: 0.6148
- R2 Score: 0.9963
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 6.5889
- Mean Absolute Error: 5.0844
- R2 Score: 0.8216
===================================


CatBoosting Regressor
Model performance for Training set
- Root Mean Squared Error: 3.0427
- Mean Absolute Error: 2.4054
- R2 Score: 0.9589
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 6.0086
- Mean Absolute Error: 4.6125
- R2 Score: 0.8516
===================================


AdaBoost Regressor
Model performance for Training set
- Root Mean Squared Error: 5.7843
- Mean Absolute Error: 4.7564
- R2 Score: 0.8516
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 6.0447
- Mean Absolute Error: 4.6813
- R2 Score: 0.8498
===================================
```

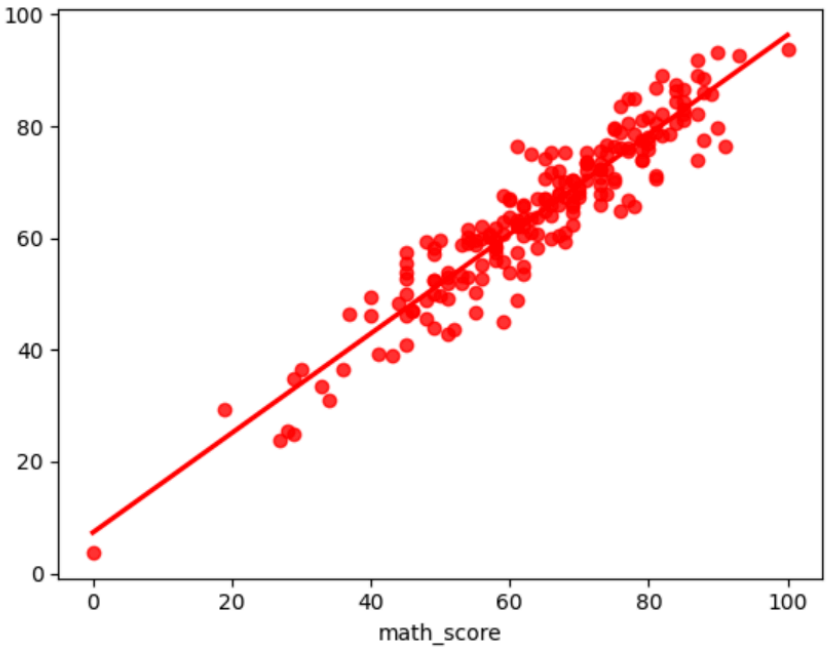
R^2 Scores of Each Model



	Model Name	R2_Score
2	Ridge	0.880593
0	Linear Regression	0.880345
7	CatBoosting Regressor	0.851632
8	AdaBoost Regressor	0.849847
5	Random Forest Regressor	0.847291
1	Lasso	0.825320
6	XGBRegressor	0.821589
3	K-Neighbors Regressor	0.783813
4	Decision Tree	0.760313

	Actual Value	Predicted Value	Difference
521	91	76.507812	14.492188
737	53	58.953125	-5.953125
740	80	76.960938	3.039062
660	74	76.757812	-2.757812
411	84	87.539062	-3.539062
...	...	...	...
408	52	43.546875	8.453125
332	62	62.031250	-0.031250
208	74	67.976562	6.023438
613	65	67.132812	-2.132812
78	61	62.492188	-1.492188

Ridge Regression Model (Best Accuracy Model)



Final Insights and Conclusions

- Student's performance is related to lunch, race, parental level education
- Females lead in pass percentage and also are top-scorers
- Student's Performance is not much impacted by test preparation course
- Finishing the preparation course is beneficial.
- Ridge has the highest R-squared score (0.880593) among all the models. R-squared is a metric that indicates how well a regression model fits the data. A higher R-squared value signifies a better fit.
- While other models like Lasso (0.825320) and Random Forest Regressor (0.847291) also have decent R-squared scores,Ridge outperforms them in this aspect.

References

ML Models
https://www.coursera.org/articles/machine-learning-models

Overfitting and Underfitting
https://medium.com/@aakriti.sharma18/andrew-ngs-machine-learning-simplified-part-8-overfitting-and-regularization-6ac38c943077

RMSE,MAE and R^2
https://www.studytonight.com/post/what-is-mean-squared-error-mean-absolute-error-root-mean-squared-error-and-r-squared#:~:text=MSE%20and%20MAE%20report%20the,variable%20described%20by%20the%20model.

EDA
https://www.ibm.com/topics/exploratory-data-analysis

Tools
https://dev.to/stevenmcgown/python-for-mlai-13-matplotlib-seaborn-jupyter-notebooks-2a76

Others
Lecture Notes and Andrew Ng Notes