جامعة الطائف
TAIF UNIVERSITY

Project Machine Larning

Student performance

Name of student : Al-Batool Mohammed Abu Al-Nada
ID : 44151371
D.R : Nada AL-Twarqi
Class : 4232

- What is the name of your data ?

The name of my dataset is " Student Performance ".

- The source of the data (which database) ?

The dataset was obtained from the Kaggle platform it is titled " Student Performance – Multiple Linear Regression " .

- Link to the original data ?

Here is the original dataset link on Kaggle :
https://www.kaggle.com/datasets/nikhil7280/student-performance-multiple-linear-regression

- Explain the data in words :

The dataset contains information about students academic and behavioral patterns that may affect their overall performance , each row represents a single student , and the columns include features such as :

i. Hours Studied
ii. Previous Scores
iii. Extracurricular Activities
iv. Sleep Hours
v. Sample Question Papers Practiced
vi. Performance Index

- Is it a regression or classification problem ?

This is **a regression** problem ,the target variable , Performance Index , is a continuous numerical value ranging from 0 to 100 therefore the objective is to predict a continuous outcome rather than assigning classes or categories , which is the case in classification problems.

- How many attributes ?

The dataset contains **6 attributes (features)** excluding the target variable .
If we count only **independent features** , then there are **5 attributes** , and **1 dependent variable** (Performance Index) .

- How many samples ?

The dataset contains a total of 100 samples (rows) , each sample represents data for one student , including their study habits , academic history , and other behavioral factors .

- What are the properties of the data? (statistics)

The dataset contains **10,000 samples** , and the statistical summary of the features is as follows :

i. **Hours Studied** :
   - Mean : 4.99 hours
   - Range : 1 to 9

ii. **Previous Scores** :
   - Mean : 69.45
   - Range : 40 to 99

iii. **Sleep Hours** :
   - Mean : 6.53 hours
   - Range : 4 to 9

iv. **Sample Question Papers Practiced** :
   - Mean : 4.58 papers
   - Range : 0 to 9

v. **Performance Index (Target)** :
   - Mean : 55.22
   - Range : 10 to 100
   - Standard Deviation : 19.21

These statistics indicate that the data is well-distributed across students with varying habits and academic backgrounds.

Additional **Data Summary (Python Output)**

The following table (1) as shown represents the statistical summary of the dataset generated using the python command : df.describe ( )

Table 1: Statistical Summary of the Student Performance Dataset

Feature	Count	Mean	Std Dev	Min	25%	50%	75%
Hours Studied	10000	4.99	2.59	1.00	3.00	5.00	7.00
Previous Scores	10000	69.45	17.34	40.0	54.0	69.0	85.0
Sleep Hours	10000	6.53	1.69	4.00	5.00	7.00	8.00
Sample Question Papers Practiced	10000	4.58	2.86	0.00	2.00	5.00	7.0076
Performance Index (Target)	10000	55.22	19.21	10.0	40.0	55.0	71.0

- Are there any missing data? How did you fill in the missing values ?

After loading and exploring the dataset , verified that there are no missing values all feature columns returned a count of 0 missing values, so no imputation or data-filling techniques were needed .

- Visualize the data :

To gain a better understanding of the data and model behavior , several visualizations were created :

i. **Model Comparison Plots** :
   - A **bar plot** was used to visualize the **Mean Squared Error (MSE)** for each model .
   - Another **bar plot** displayed the **R^2 Score** values to compare the accuracy of different algorithms .
   - Each bar was labeled with its exact value , making it easier to interpret the differences between models .

ii. **Actual vs Predicted Performance** :
   - For each model , a **line graph** was plotted showing the actual vs predicted values for the first 10 samples .
   - This visual comparison helped identify how closely each model's predictions matched the true performance index .
   - Models like Linear Regression and ANN showed strong alignment , while Naive Bayes had significant deviations.

iii. **Data Distribution (before modeling)** :
   - These visualizations provided clear insights into model performance and highlighted which algorithms were most effective for the dataset.

In addition to model performance visualizations , conducted an exploratory data analysis (EDA) to better understand the features and their relationships with the target :

- A **correlation heatmap** was used to reveal how features relate to each other and to the target " Performance Index " look at figure (1) .
- **Histograms** showed the distribution of numeric values as shown figure (2) , including study hours , sleep , and previous scores
- A **boxplot** was created to analyze the effect of sleep hours and extracurricular activities as shown figure (3) on student performance .

These visualizations helped us understand data skewness , variance , and feature importance before model training.

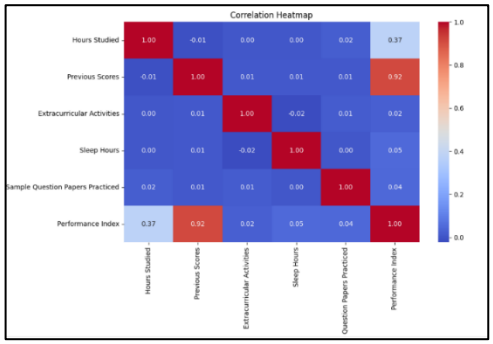


Figure (1) : Feature Correlation Matrix

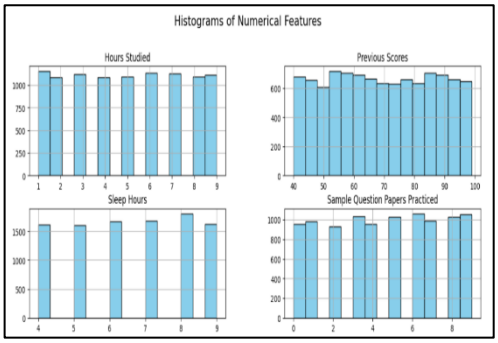


Figure (2) : Distribution of Input Features

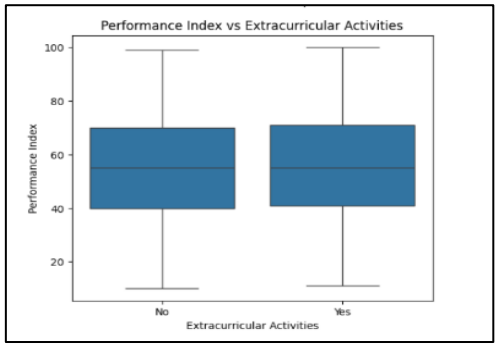


Figure (3) : Performance by Extracurricular Activities

- Did you normalize or standardize any of your data ? Why ?

No , we did not apply any normalization or standardization to the dataset.
This decision was based on most of the features were already in relatively similar numerical ranges , tree-based models like Decision Tree and Random Forest are not affected by feature scaling ,the performance of models like Linear Regression, SVM, and KNN was still high without scaling.

- What type of preprocessing did you apply to your data ? List everything and explain why ?

Applied the following preprocessing steps to prepare the data for model training :

i. Label Encoding :

The column Extracurricular Activities contained categorical values (" Yes " and " No ") .
converted them to numerical values using :
df [' Extracurricular Activities '] = df ['Extracurricular Activities'].map({ ' Yes ' : 1 , ' No ' : 0})
This was necessary because machine learning models require numerical input.

ii. Train-Test Split :

split the dataset into training and testing sets using an 80 / 20 ratio to evaluate the models on unseen data.
X_train , X_test , y_train , y_test = train_test_split (X , y , test_size = 0.2 , random_state = 42)

iii. Missing Values Check

We checked for null values using df.isnull( ).sum( ) , so No missing values were found no imputation was needed.

iv. Exporting Preprocessed Data :

saved the cleaned and split data into CSV files :
X.csv , X_test.csv , Y.csv , Y_test.csv , this was done to organize the dataset into a structured format for training and evaluation.

- How did you divide the train and test data ? What are the proportions ?

Divided the dataset into two sets using the **train_test_split** function from sklearn.model_selection :
80% of the data was used for training the models , 20% of the data was used for testing and evaluating the model performance , this split was done to ensure that models are trained on a sufficient amount of data while still being evaluated on unseen data to check for generalization .

- Apply all the machine learning models you have learned in this course to your data and report the results. What is the best / worst performing model ? Why ?

Applied all the machine learning models covered in the course to our regression problem using the Student Performance dataset. These models are :

i. Linear Regression
ii. Decision Tree Regressor
iii. Random Forest Regressor

iv. K-Nearest Neighbors Regressor (KNN)
v. Support Vector Machine (SVM)
vi. ANN (Artificial Neural Network)
vii. Naive Bayes

Each model was trained and evaluated based on Mean Squared Error (MSE) and R^2 Score , visualized the performance results using comparison charts , and generated actual vs. predicted.

The best Performing Model Linear Regression and ANN (Artificial Neural Network) both achieved the lowest MSE values (≈ 4.08 and 4.15) and highest R^2 scores (≈ 0.989) in otherwise **the worst Performing Model** Naive Bayes , this model had the highest MSE (≈ 37.30) and the lowest R^2 score (≈ 0.899) the reason is that Naive Bayes is not designed for regression problems ; it assumes feature independence and is better suited for classification tasks , not continuous output like performance scores.

- The accuracy of all models using tables and figures ?

We evaluated the performance of all applied models using two key metrics , Mean Squared Error (MSE) - Lower is better and R^2 Score - Closer to 1 is better as shown in the table (2)

table (2) : Model Evaluation Results Using MSE and R^2 Score

Model	MSE	R^2 Score
Linear Regression	4.08	0.989
Decision Tree	8.84	0.976
Random Forest	5.17	0.986
SVM	5.39	0.985
KNN	5.98	0.984
ANN	4.15	0.989
Naive Bayes	37.30	0.899

MSE Comparison Chart clearly shows that Linear Regression and ANN had the lowest errors as shown in figure (4) .

R^2 Score Comparison Chart as shown in figure (5) confirms both models also had the highest R^2 values, indicating excellent predictive ability .

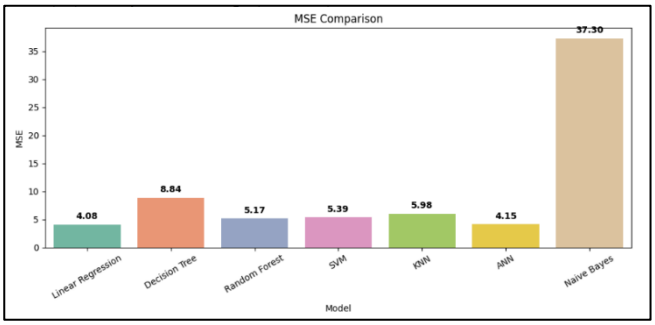


Figure (4) : MSE Comparison Across Regression Models

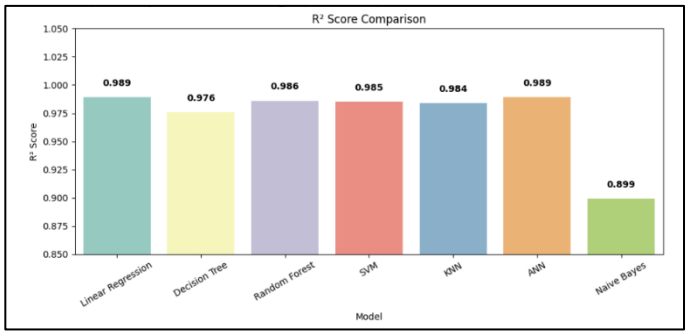


Figure (5) : R^2 Score Comparison Across Regression Models

Additionally , each model has an " Actual vs Predicted " plot showing how close its predictions were to real values.

نتائج نموذج: Linear Regression

	Actual	Predicted
0	51.0	54.711854
1	20.0	22.615513
2	46.0	47.903145
3	28.0	31.289767
4	41.0	43.004570
5	59.0	59.071252
6	48.0	45.903475
7	87.0	86.459118
8	37.0	37.700140
9	73.0	72.055925

نتائج نموذج: Decision Tree

	Actual	Predicted
0	51.0	58.0
1	20.0	24.0
2	46.0	45.0
3	28.0	27.0
4	41.0	45.0
5	59.0	59.0
6	48.0	49.0
7	87.0	85.0
8	37.0	35.0
9	73.0	73.0

نتائج نموذج: Random Forest

	Actual	Predicted
0	51.0	56.190000
1	20.0	21.950000
2	46.0	46.300000
3	28.0	29.181048
4	41.0	42.808333
5	59.0	59.432500
6	48.0	47.008000
7	87.0	86.132333
8	37.0	36.710000
9	73.0	72.180000

نتائج نموذج: KNN

	Actual	Predicted
0	51.0	56.6
1	20.0	24.4
2	46.0	48.4
3	28.0	30.0
4	41.0	43.4
5	59.0	58.4
6	48.0	44.2
7	87.0	88.4
8	37.0	37.6
9	73.0	72.6

نتائج نموذج: SVM

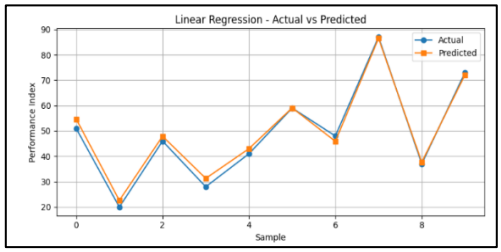
	Actual	Predicted
0	51.0	54.822844
1	20.0	24.263054
2	46.0	46.578362
3	28.0	31.183212
4	41.0	42.419642
5	59.0	58.454746
6	48.0	45.235555
7	87.0	84.897595
8	37.0	37.140106
9	73.0	71.745022

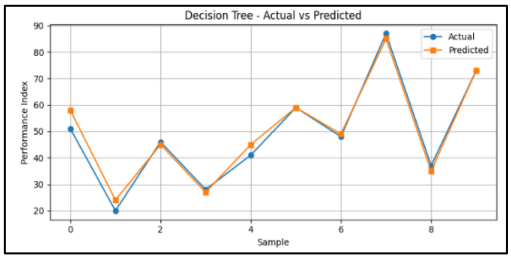
نتائج نموذج: ANN

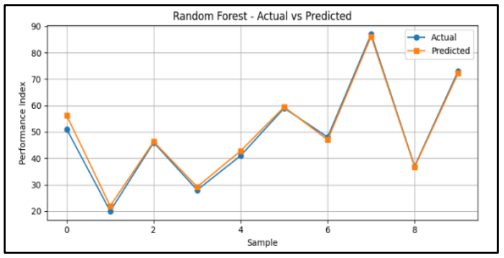
	Actual	Predicted
0	51.0	54.517687
1	20.0	22.465174
2	46.0	47.643245
3	28.0	31.014309
4	41.0	42.827872
5	59.0	58.807274
6	48.0	45.763084
7	87.0	86.092325
8	37.0	37.371856
9	73.0	71.728129

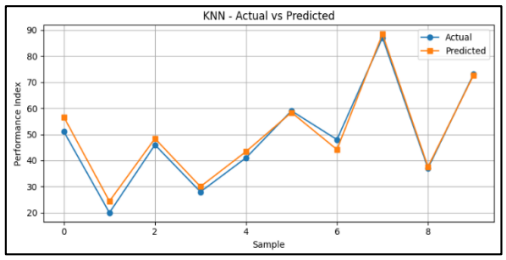
نتائج نموذج: Naive Bayes

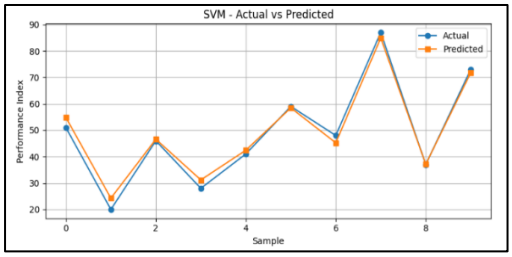
	Actual	Predicted
0	51.0	57.0
1	20.0	24.0
2	46.0	41.0
3	28.0	29.0
4	41.0	41.0
5	59.0	49.0
6	48.0	43.0
7	87.0	84.0
8	37.0	34.0
9	73.0	62.0

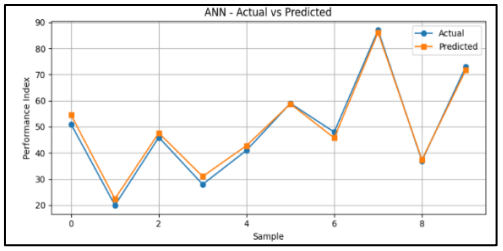












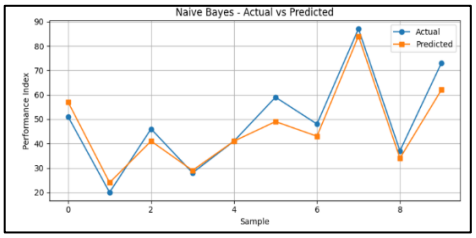


Figure (6) : Actual vs Predicted Performance Index Across All Regression Models

- If your ability to present the result is advanced (using plot libraries such as seaborn android techniques) you will get 5 marks bonus

Enhanced the visualization of our results by using Seaborn , a powerful Python library for statistical graphics. Our bar plots were designed with distinct colors for each model , and the exact values of MSE and R^2 Score were clearly annotated above each bar for clarity. This makes it easy to distinguish the performance of each algorithm at a glance.

Additionally, we created line graphs showing the actual vs predicted values for the first 10 samples of each model. These comparisons offered deep insights into how closely each model's predictions matched the real performance index.

This advanced visual representation makes the findings more interpretable and effective, thus meeting the criteria for the 5-mark bonus.

- Explain in 20 lines , font size 20 , Font : Times New Roman , What is the reason you picked up this data? What is the importance of your data in reality, and what is the importance of your best-performing model ? Is there any insight you could share from the data and the model ?

Chose this dataset because it reflects a realistic and relatable academic scenario. Understanding student performance is essential in today's educational landscape .
The data includes meaningful features like study hours , previous scores , sleep patterns , and more .
These attributes are common among students and can significantly affect academic outcomes.
This dataset offers a solid foundation to apply regression techniques and extract useful predictions.
It also helps in developing early intervention strategies for struggling students .
By predicting student performance , educators can provide support before final results are released .
Among the applied models , Linear Regression and ANN performed the best with the lowest MS .
Both models had an R^2 Score of 0.989 , showing their high prediction accuracy .
This makes them highly suitable for continuous outcome prediction like performance index .
The worst-performing model was Naive Bayes with an MSE of 37.3 and R^2 of 0.899 . This confirms that Naive Bayes might not suit continuous regression tasks effectively .
Our model plots clearly show how actual scores align with predicted values.
The visualizations validate our model selection and allow easy comparison .
Also discovered that sleep hours and practice played key roles in high scores .
This insight can guide educational policies and personal student strategies . With these predictions , schools can design smarter study plans for students .
Furthermore , this project helped reinforce the strengths and weaknesses of each ML model .
It bridged theoretical learning with a real-world educational application .
Overall , this dataset provided a meaningful and insightful machine learning experience.

- Link to your code and data , remember the code is in the main folder , the data is in the folder Data, and save the original data + another folder the train test devision (features and targets) ?

https://github.com/batool1998-hub/Student-Performance-project

- In the Data folder , create a folder called Result and add test set and the predicted value from all models (to check the accuracy) without the features.

Done √