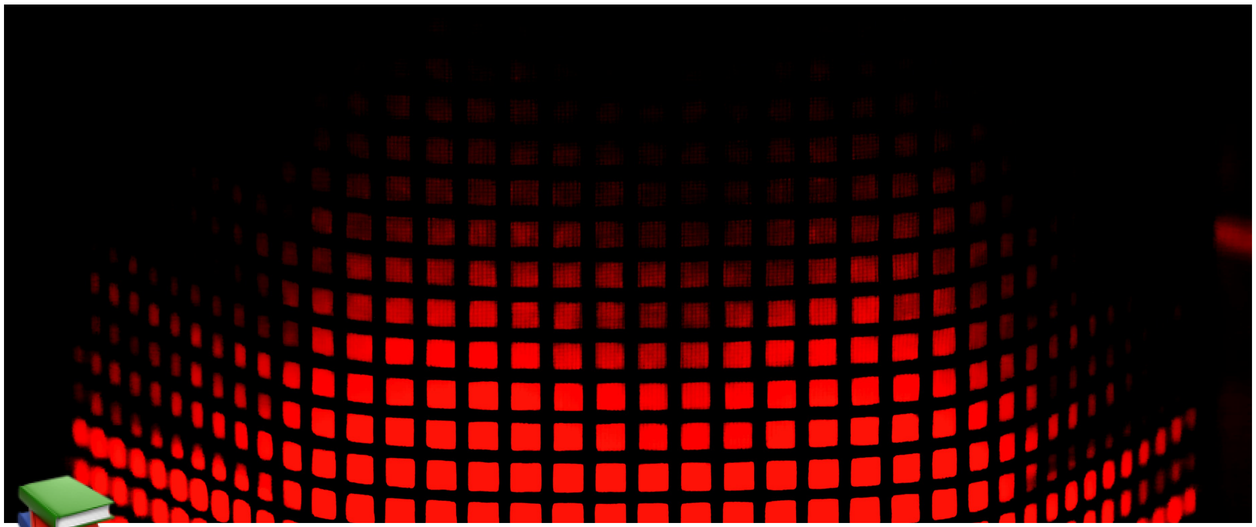


DS 675 Project - Milestone 4

Analysis and Modeling of the Student Performance Factors Dataset

Members → Kriss Sitapara, Paridhi Bhardwaj, Jyotsna Vatti

Find our .ipynb file, video, report in this git link → https://github.com/kkrriissss/ds675project

Introduction→

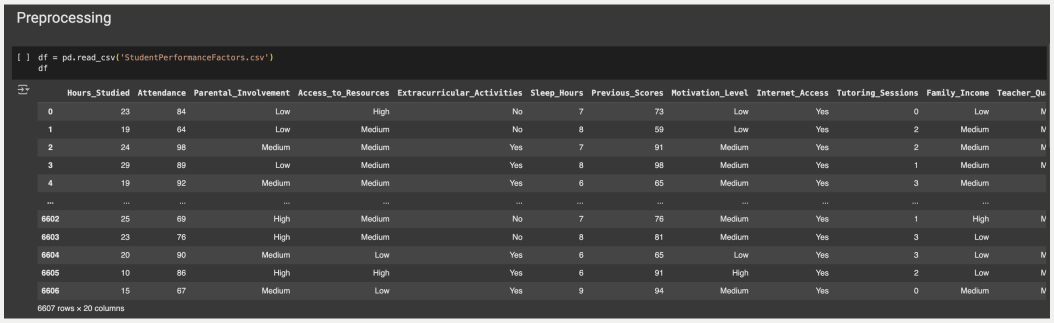
The team selected upon the Dataset **Student Performance Factors** (https://www.kaggle.com/datasets/lainguyn123/student-performance-factors) that focuses on improving the accuracy and insights derived from machine learning models.

About the Dataset→ The purpose of this dataset is to provide insights into students performance and its contributing factors. It gives an overview of various factors affecting student performance in exams. This information includes data on their study habits, attendance, parental involvement which is helpful to determine their success.

Building upon the analyses from Milestone 2, we have introduced advanced preprocessing techniques, exploratory data analysis (EDA), improved the original work done by Ahmed and then created our own model.

Preprocessing Steps→

- We started the code by importing the necessary python libraries such as pandas, numpy, seaborn, matplotlib, various sklearn models, SVMs etc and then *read our csv file.*



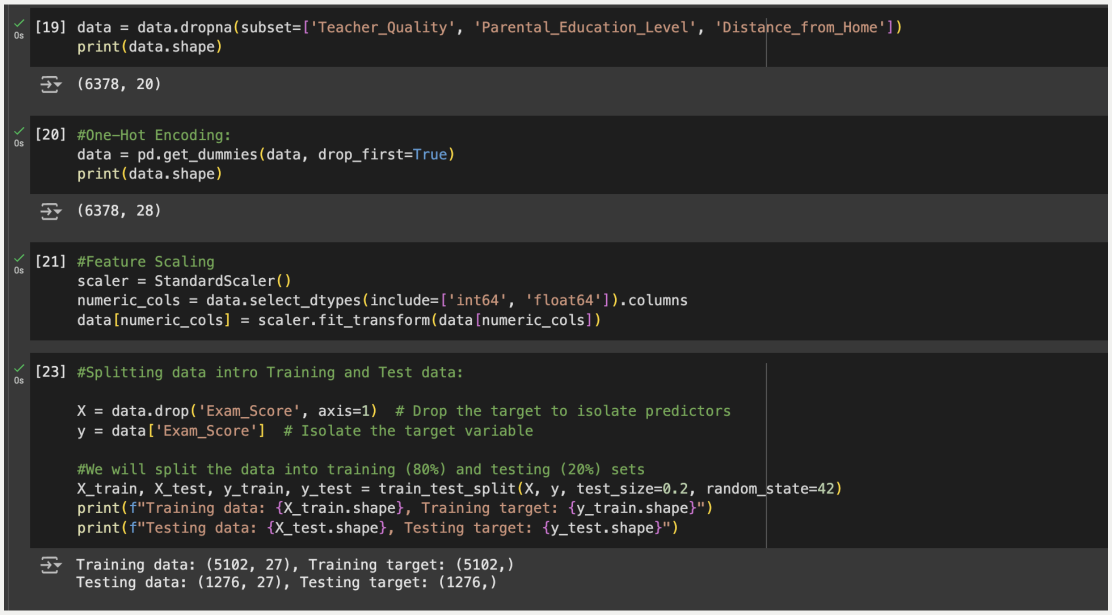
Preprocessing

```
[ ] df = pd.read_csv('StudentPerformanceFactors.csv')
    df
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level	Internet_Access	Tutoring_Sessions	Family_Income	Teacher_Qu
0	23	84	Low	High	No	7	73	Low	Yes	0	Low	M
1	19	64	Low	Medium	No	8	59	Low	Yes	2	Medium	M
2	24	98	Medium	Medium	Yes	7	91	Medium	Yes	2	Medium	M
3	29	89	Low	Medium	Yes	8	98	Medium	Yes	1	Medium	M
4	19	92	Medium	Medium	Yes	6	65	Medium	Yes	3	Medium	
...	...	...	...	...	...	...	...	...	...	...	...	
6602	25	69	High	Medium	No	7	76	Medium	Yes	1	High	M
6603	23	76	High	Medium	No	8	81	Medium	Yes	3	Low	
6604	20	90	Medium	Low	Yes	6	65	Low	Yes	3	Low	M
6605	10	86	High	High	Yes	6	91	High	Yes	2	Low	M
6606	15	67	Medium	Low	Yes	9	94	Medium	Yes	0	Medium	M

6607 rows × 20 columns

- We then searched if there is any *duplicate values and any missing values.* On running, we observed that we do not have any duplicate values although we did had some missing values. Our original data had **6607 X 20 columns** but the missing values brought down the rows to **6378 X 20 columns.**
- We then proceeded on doing the *one-hot encoding, feature scaling, and also ensured to split our dataset into training and test data*



```
[19] data = data.dropna(subset=['Teacher_Quality', 'Parental_Education_Level', 'Distance_from_Home'])
     print(data.shape)

     (6378, 20)

[20] #One-Hot Encoding:
     data = pd.get_dummies(data, drop_first=True)
     print(data.shape)

     (6378, 28)

[21] #Feature Scaling
     scaler = StandardScaler()
     numeric_cols = data.select_dtypes(include=['int64', 'float64']).columns
     data[numeric_cols] = scaler.fit_transform(data[numeric_cols])

[23] #Splitting data intro Training and Test data:

     X = data.drop('Exam_Score', axis=1)  # Drop the target to isolate predictors
     y = data['Exam_Score']  # Isolate the target variable

     #We will split the data into training (80%) and testing (20%) sets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
     print(f"Training data: {X_train.shape}, Training target: {y_train.shape}")
     print(f"Testing data: {X_test.shape}, Testing target: {y_test.shape}")

     Training data: (5102, 27), Training target: (5102,)
     Testing data: (1276, 27), Testing target: (1276,)
```

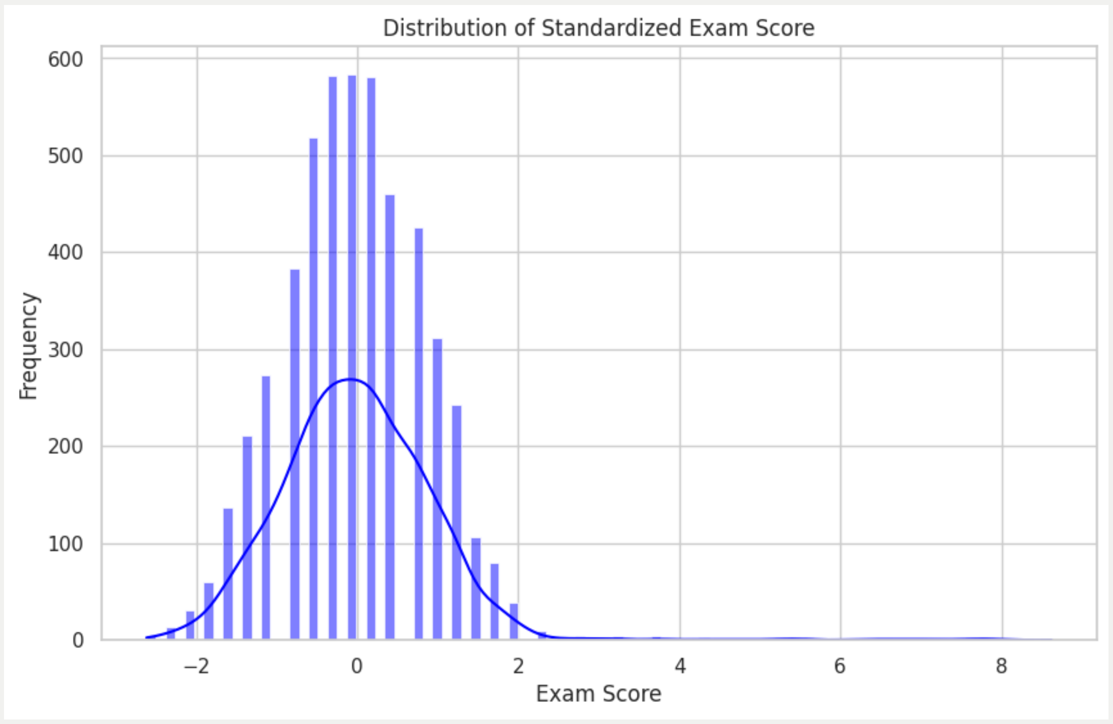
Exploratory Data Analysis (EDA) →

EDA can be defined as the process that involves in investigating and summarizing the main characteristics of a dataset. The goal of EDA is to understand the data, uncover patterns, identify anomalies if any present, test hypothesis and check assumptions through visual and statistical methods.

- We created *histograms to check the distributions for every factor.* The factors includes →
  - Hours_Studied, Attendance, Sleep_Hours, Previous_Scores, Tutoring_Sessions, Physical_Activity, Parental_Involvement_Low, Parental_Involvement_Medium, Access_to_Resources_Low, Access_to_Resources_Medium, Extracurricular_Activities_Yes, Motivation_Level_Low, Motivation_Level_Medium, Internet_Access_Yes, Family_Income_Low, Family_Income_Medium, Teacher_Quality_Low, Teacher_Quality_Medium, School_Type_Public, Peer_Influence_Neutral, Peer_Influence_Positive, Learning_Disabilities_Yes, Parental_Education_Level_HighSchool,

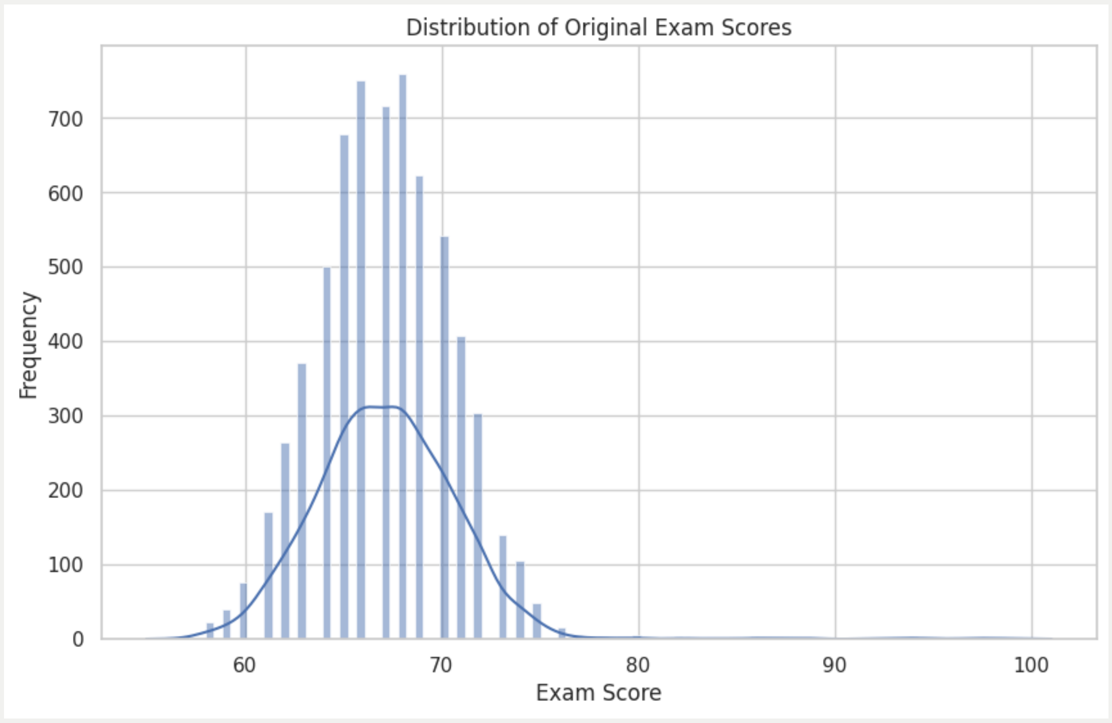
Parental_Education_Level_Postgraduate, Distance_from_Home_Moderate, Distance_from_Home_Near, Gender_Male

- We then proceeded on doing the *Target variable distribution* which is used to understand the distribution in EDA as it provides us with insights into the characteristics of the variable. We selected ***Standardized Exam Score*** as our variable. The following output was given and we learned that the **center** is around 0 which confirms that the data is **standardized**. The **spread** values falls within -2 to 2 which also corresponds to the normal distribution's empirical rule. We did see some **outliers** on the right hand side and the data is nearly **symmetric**.

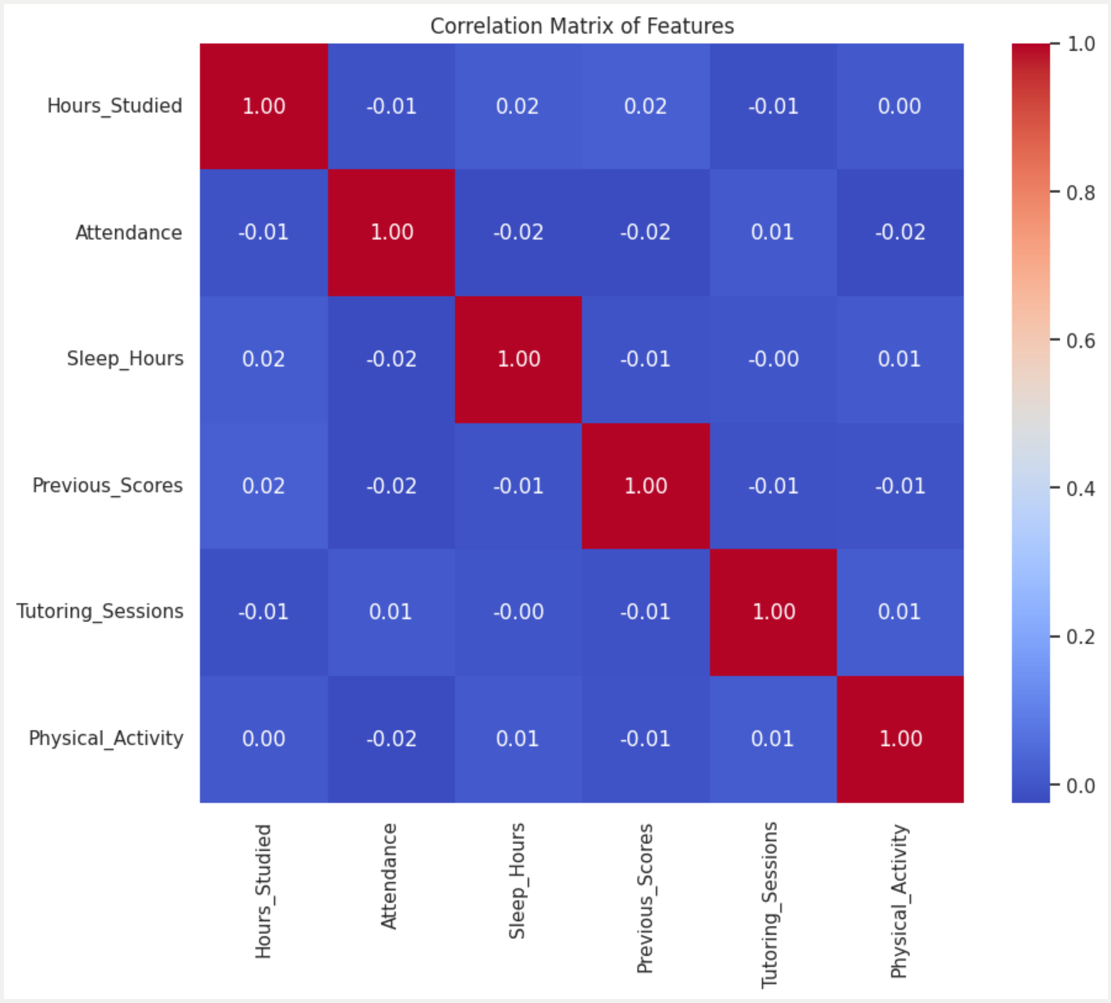


- We then graphed the *Original Exam Scores* the distribution of the original exam scores is used to understand their spread, central tendency and if there is any potential outliers. We see that the majority of scores fall between 60 and 80 and peaks around 70.

📌

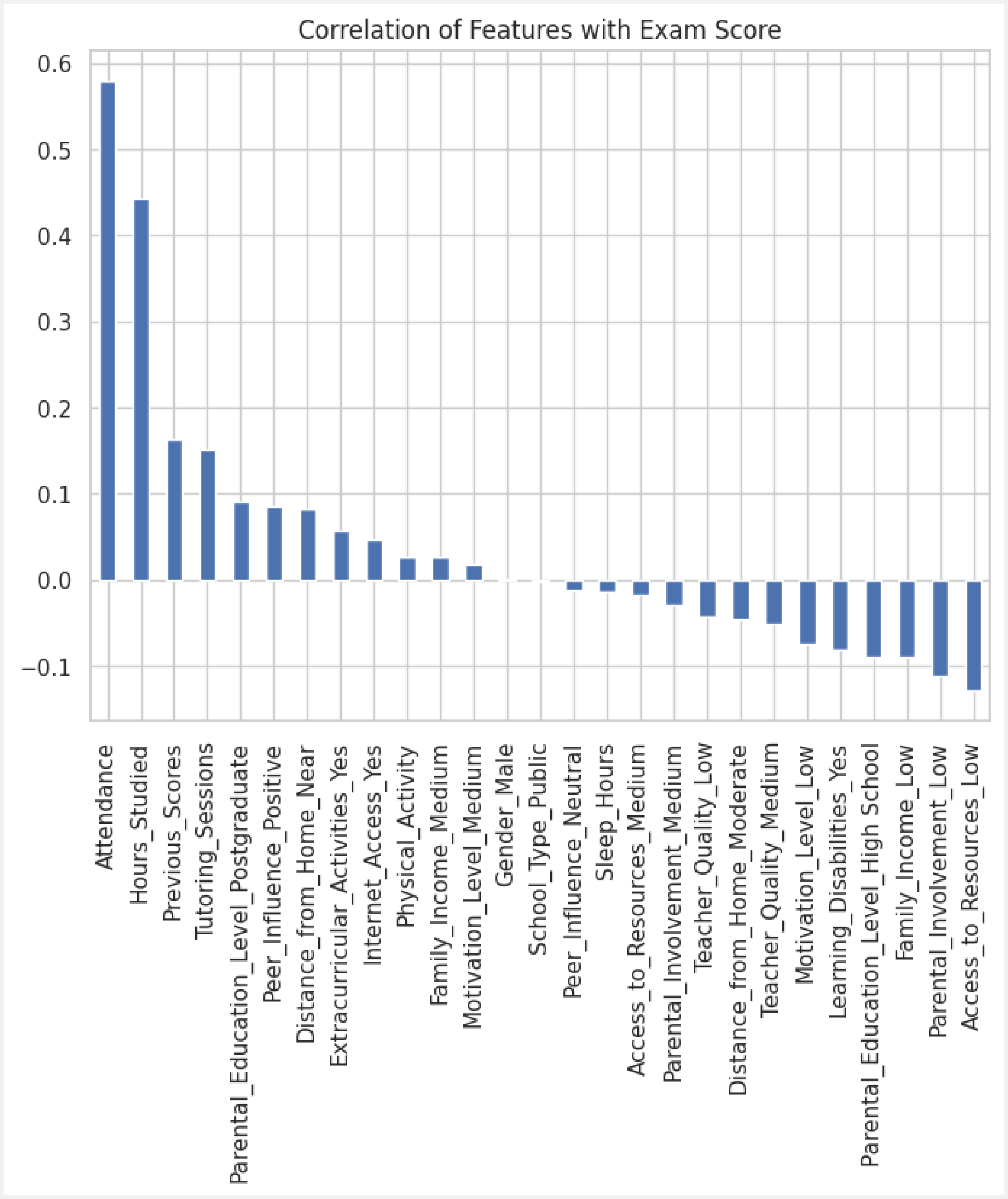


- We also created the *heatmap in order to visualize the correlation matrix*

📌 

	Hours_Studied	Attendance	Sleep_Hours	Previous_Scores	Tutoring_Sessions	Physical_Activity
Hours_Studied	1.00	-0.01	0.02	0.02	-0.01	0.00
Attendance	-0.01	1.00	-0.02	-0.02	0.01	-0.02
Sleep_Hours	0.02	-0.02	1.00	-0.01	-0.00	0.01
Previous_Scores	0.02	-0.02	-0.01	1.00	-0.01	-0.01
Tutoring_Sessions	-0.01	0.01	-0.00	-0.01	1.00	0.01
Physical_Activity	0.00	-0.02	0.01	-0.01	0.01	1.00

- We then worked upon creating the *correlation of features with the target variable Exam_Score*. The bar chart visualizes how strongly each features in the dataset correlates with the target variable. This features are important to show us the strongest relationship with the target variable and can be used in our models. The correlations can be another factors that contribute positively and negatively to student's exam performance.
    - **Positive Correlations** → Attendance, Hours_Studied, Previous_Scores
    - **Negative Correlations** → Parental_Involvement_Low and Access_to_Resources_Low



Modifying the previous code →

We decided to modify the code worked by Ahmed Ezzat Ibrahem
https://www.kaggle.com/code/ahmedezzatibrahem/student-performance-factors

He build the 3 models →

1. Linear regression model which was used on the scaled training data
2. Ridge regression model on the scaled training data (with regularization alpha 1.0)
3. Support Vector Regression SVR model

We focused on the SVR Model that Ahmed made and went on to fine tune that.

We were able to **reduce his MSE** from 0.291 to 0.285. We emphasized on *parameter tuning* which is a crucial step when comes to optimizing the model because it fine tunes the settings that the learning algorithm uses to maximize its performance. Ahmed used SVR model and its parameters such as Kernel, C(Regularization), Epsilon, and degree.

What was Ahmed's model?

Ahmed employed the SVR model using the default parameters

- Kernel → "rbf" as it is the default in many implementations
- C (Regularization) → it is also set at 1.0 by default
- Epsilon → commonly 0.1 by default
- Degree → it is not applicable for the 'rbf' kernel but defaults to 3 if a polynomial kernel is used.

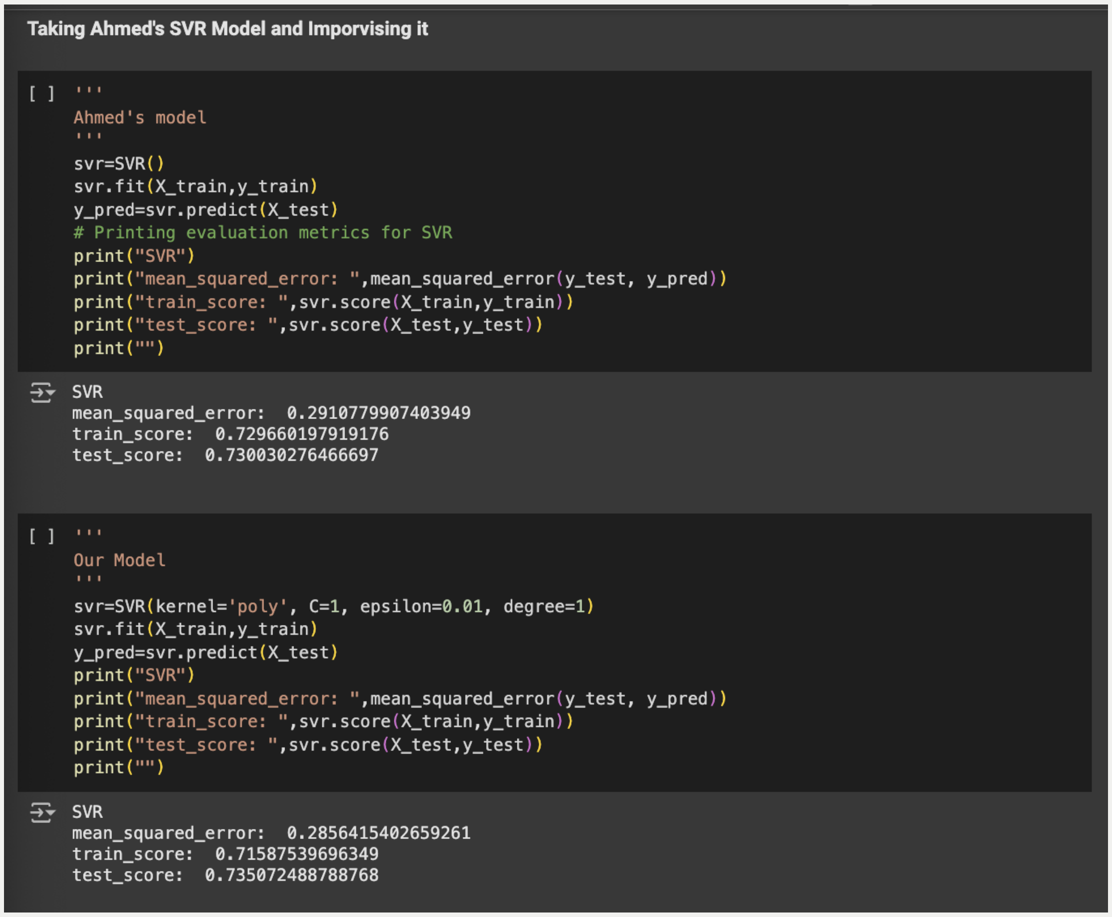
What we did different?

Our model did better because we fine-tuned our SVR model due to the kernel setting. By switching to a polynomial kernel of degree 1 we were able to benefit from the linear relationships in the data that are more effectively than possibly assuming the default rbf kernel. The polynomial kernel with 1 degree also ensured that there is **no overfitting** which can be introduced due to unnecessary complexity. This made our model **simple** which was better suited to the structure and trends in our dataset leading to higher accuracy.

- For our enhanced model we set the **kernel to poly** which allows the model to consider feature interactions up the specified degree this provides flexibility in capturing linear and non-linear patterns

- we set the **degree to 1** which simulated a linear relationship under the poly kernel settings
- We maintained his **C (regularization) of 1.0** which balanced the margin maximization with error minimization
- The **epsilon was set to 0.1** which helped in the model to not react aggressively to minor errors.

📌


Taking Ahmed's SVR Model and Imporvising it

```
'''
Ahmed's model
'''
svr=SVR()
svr.fit(X_train,y_train)
y_pred=svr.predict(X_test)
# Printing evaluation metrics for SVR
print("SVR")
print("mean_squared_error: ",mean_squared_error(y_test, y_pred))
print("train_score: ",svr.score(X_train,y_train))
print("test_score: ",svr.score(X_test,y_test))
print("")
```

```
SVR
mean_squared_error:  0.2910779907403949
train_score:  0.729660197919176
test_score:  0.730030276466697
```

```
'''
Our Model
'''
svr=SVR(kernel='poly', C=1, epsilon=0.01, degree=1)
svr.fit(X_train,y_train)
y_pred=svr.predict(X_test)
print("SVR")
print("mean_squared_error: ",mean_squared_error(y_test, y_pred))
print("train_score: ",svr.score(X_train,y_train))
print("test_score: ",svr.score(X_test,y_test))
print("")
```

```
SVR
mean_squared_error:  0.2856415402659261
train_score:  0.71587539696349
test_score:  0.735072488788768
```

Our model 1 - K means→

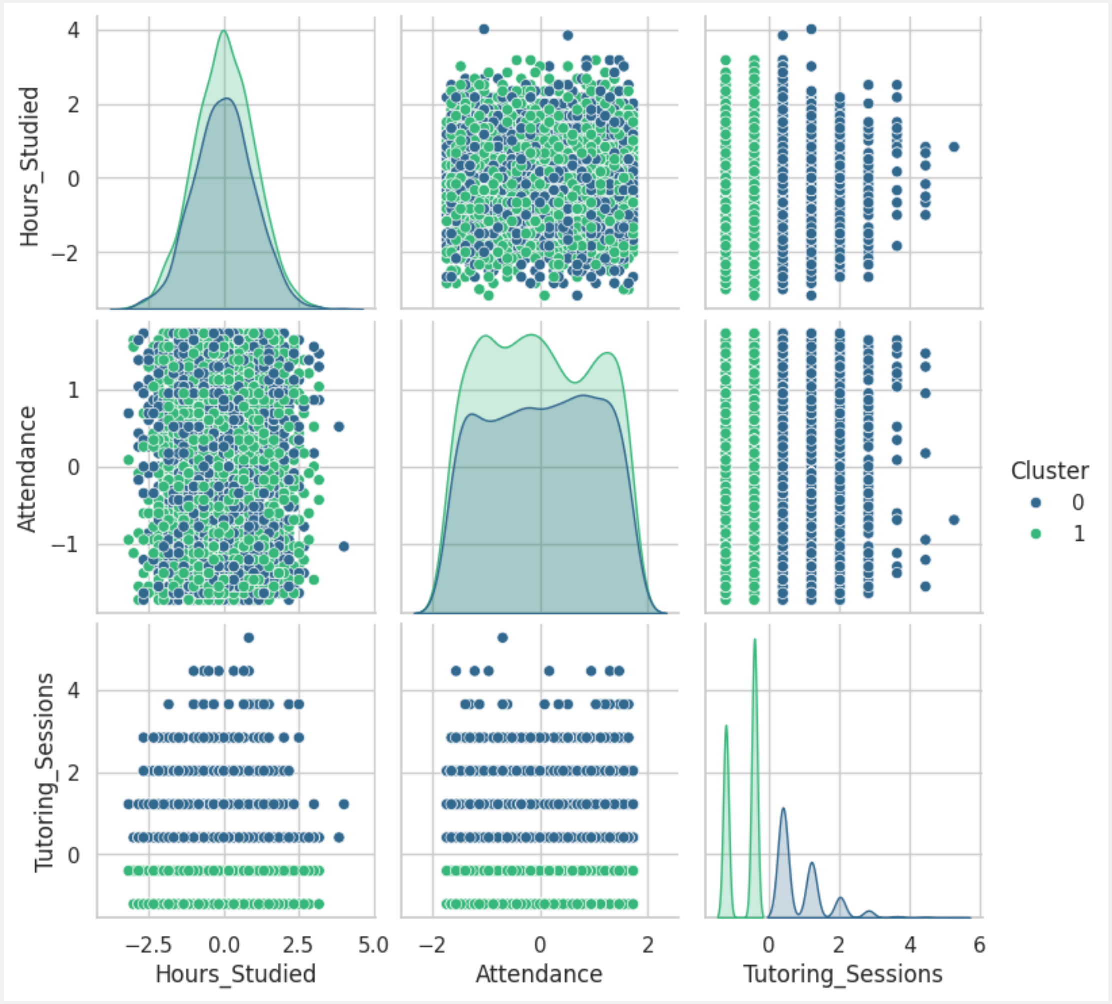
We made our first model with **k-means clustering** with two clusters (0 and 1) based on Hours_Studied, Attendance, Tutoring_Sessions.

Cluster 0 has centroids near zero for Hours_Studied and slightly positive for Attendance but a notable positive for Tutoring_Sessions. This suggests that students have average study hours, slightly above-average attendance and significant in tutoring sessions.

Cluster 1 shows higher Hours_Studied and Attendance but very low Tutoring_Sessions. This group seems to consists of students who study more and have better attendance but rely less on tutoring, therefore are self-sufficient.

Cluster 1 shows higher hours studied compared to cluster 0. Both clusters cover a wide range for **attendance** although cluster 1 tends to have slightly higher attendance. Cluster 0 students are more likely to attend **tutoring sessions.**

This can be used to learn relevant insights for academic support so knowing which students rely more on tutoring.



Our model 2- Linear with Lasso→

Linear Regression Performance:

- The Linear Regression model performed reasonably well with an R² score of approximately 0.731. This indicates that about 73% of the variance in the our target variable (Exam_scores) can be explained by the model, which is a decent performance for this dataset cause of its nature.

Lasso Regression Performance:

- Initially, the Lasso Regression model provided a poor R² score due to an inappropriate alpha value, which led to extreme regularization. After applying cross-validation with LassoCV, we found a more suitable alpha, and the performance significantly improved.

- The Lasso model selected with cross-validation had an R^2 close to 0.731, matching the performance of the Linear Regression model, suggesting that it is capturing much of the same predictive information but with potentially fewer features due to the regularization effect.

Combined Model Performance:

- The combined weighted model, which used a weighted average of predictions from both the Linear and Lasso regression models, improved the R^2 score to about 0.731. This enhancement, albeit slight, indicates that integrating these models brought some incremental benefits, potentially balancing bias and variance more effectively.

Impact of Outliers: The presence of outliers with larger residuals indicates that the model struggles to predict more extreme or atypical cases accurately. These outliers have a considerable impact on the R^2 score, pulling it down from potentially higher values. Addressing these outliers could lead to improvements in the model's overall accuracy and predictive performance. Now this was something that we noticed not only in our data models, but other people's models as well that worked on this dataset

Conclusion →

This milestone was very insightful for us because we got to practice our skills on developing our own machine learning model. This project made us extensively think about how to analyze data and analyze patterns. We used our theoretical knowledge from classes such as the concepts of Lasso, K-means which were used to develop our models. The team took the Machine Learning class for the first time and learned a lot especially from this project and while coding we were also able to review the topics which will be tested on the final exam.