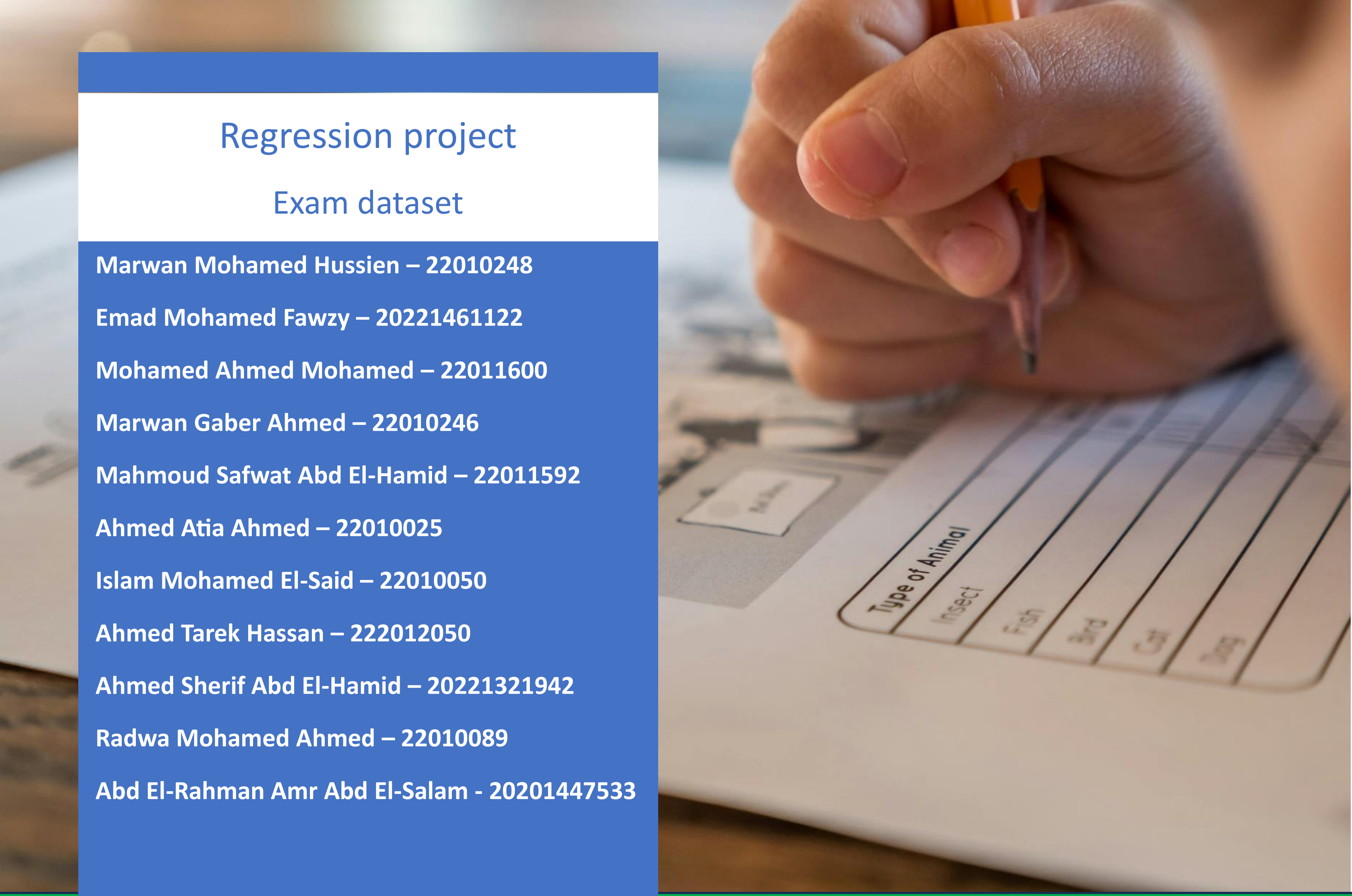


Regression project

Exam dataset

Marwan Mohamed Hussien – 22010248

Emad Mohamed Fawzy – 20221461122

Mohamed Ahmed Mohamed – 22011600

Marwan Gaber Ahmed – 22010246

Mahmoud Safwat Abd El-Hamid – 22011592

Ahmed Atia Ahmed – 22010025

Islam Mohamed El-Said – 22010050

Ahmed Tarek Hassan – 222012050

Ahmed Sherif Abd El-Hamid – 20221321942

Radwa Mohamed Ahmed – 22010089

Abd El-Rahman Amr Abd El-Salam - 20201447533

Introduction

In this project we aim the determine the factors which affects the students' performance during exams.

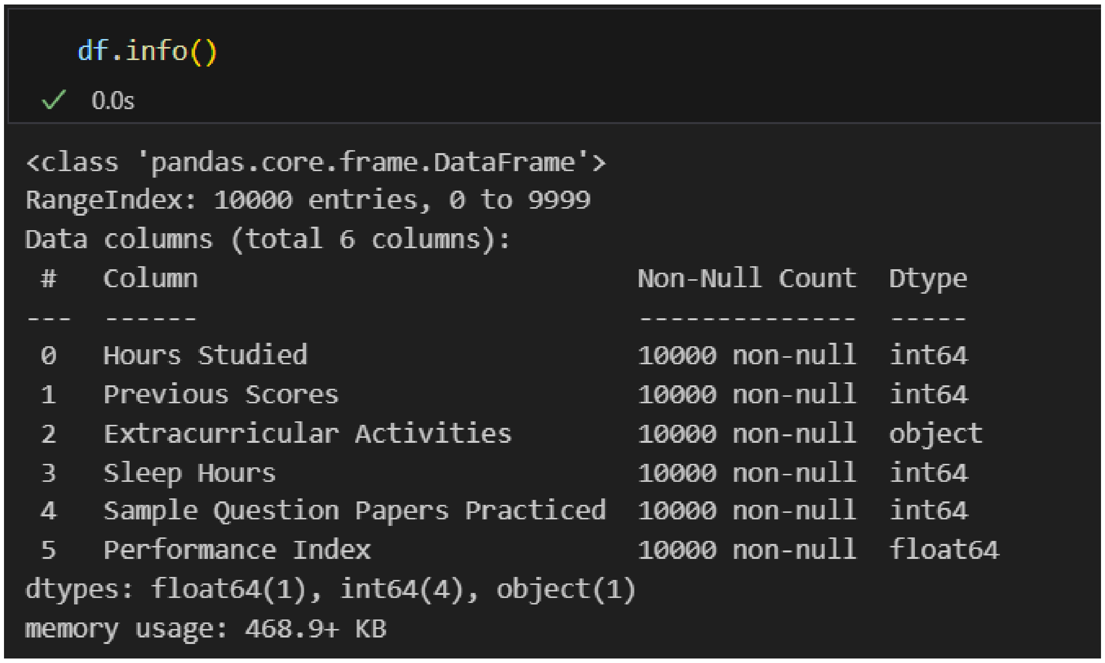
To achieve our goal, we are going to use the dataset at:

https://www.kaggle.com/datasets/nikhil7280/student-performance-multiple-linear-regression

We are using python as our programming language of choice along with pandas for data storing , matplotlib and seaborn for data visualization , and SK-Learn for the regression process itself.

Data Cleaning

We start by importing the data into a pandas data frame and then we see the make of the dataset.

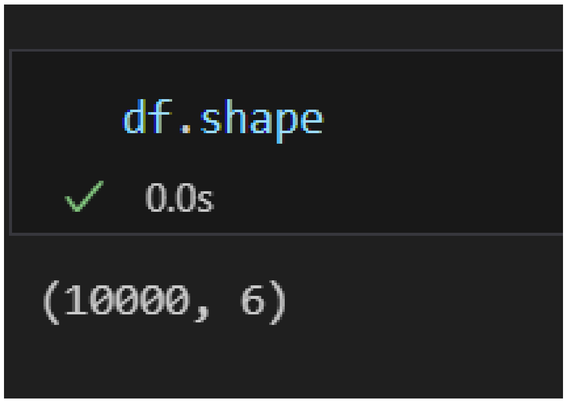


```
df.info()
✓ 0.0s
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Hours Studied                     10000 non-null  int64
 1   Previous Scores                   10000 non-null  int64
 2   Extracurricular Activities        10000 non-null  object
 3   Sleep Hours                       10000 non-null  int64
 4   Sample Question Papers Practiced  10000 non-null  int64
 5   Performance Index                 10000 non-null  float64
dtypes: float64(1), int64(4), object(1)
memory usage: 468.9+ KB
```

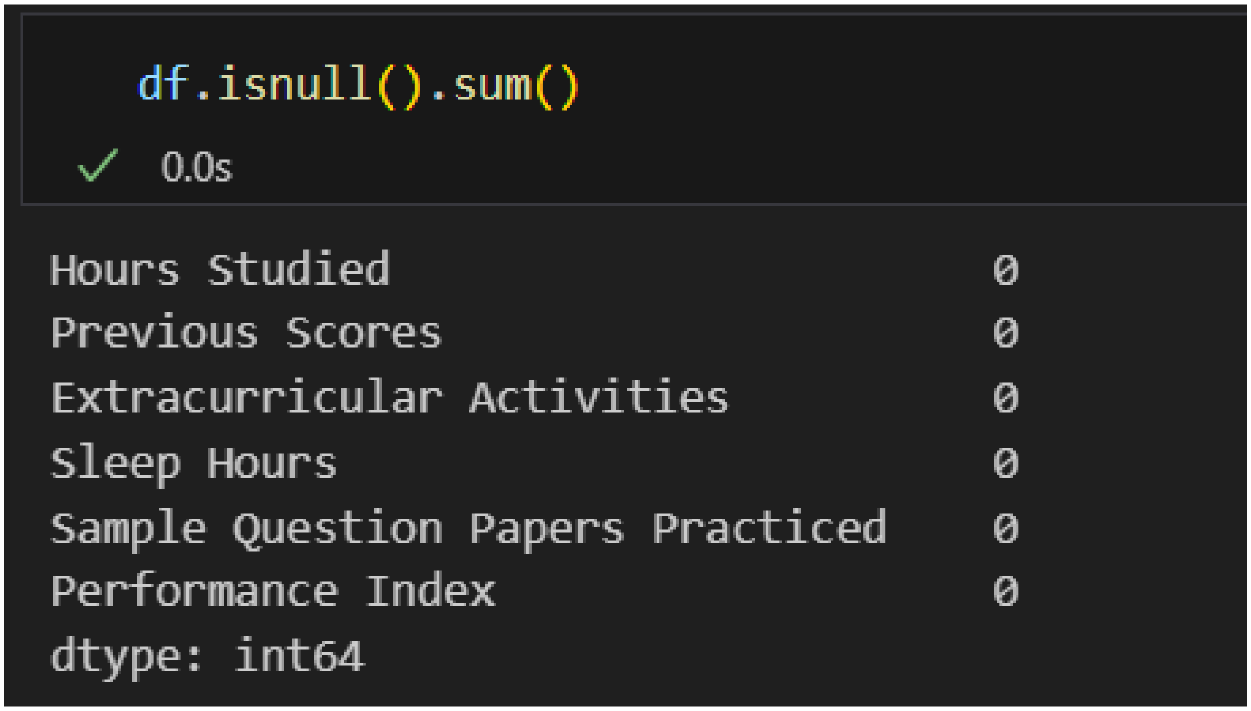
We see that we have 6 columns : Hours studied, Previous Scores, Extracurricular Activities, Sleep Hours, Sample Question Papers Practiced, and Performance Index.

We print the shape of the dataset and see that we have 10,000 rows and six columns.



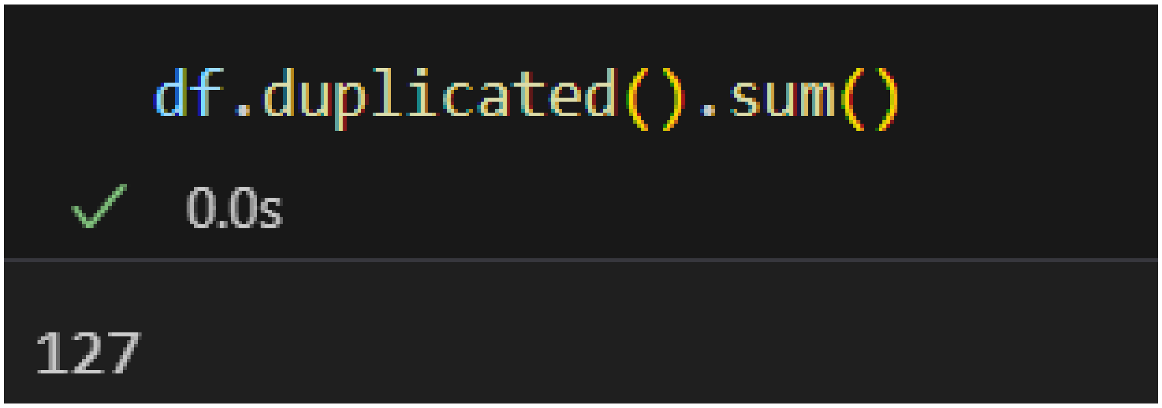
```
df.shape
✓ 0.0s

(10000, 6)
```

We then check for null values of which there were none.



```
df.isnull().sum()
✓ 0.0s

Hours Studied                       0
Previous Scores                     0
Extracurricular Activities          0
Sleep Hours                         0
Sample Question Papers Practiced    0
Performance Index                   0
dtype: int64
```

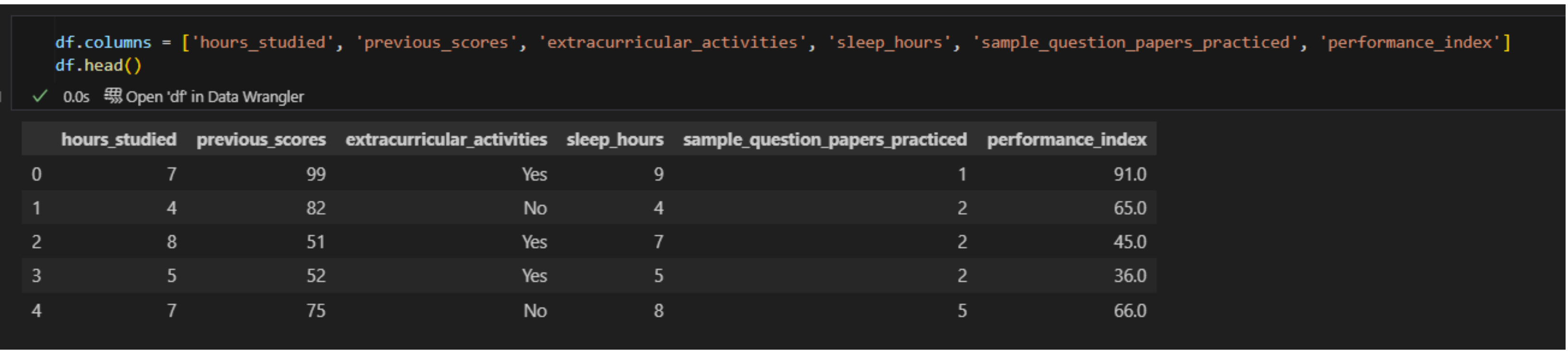
Next, we checked for duplicate values which there were 127.



```
df.duplicated().sum()
✓ 0.0s

127
```

We removed the null values and then the data were reduced to 9873 entries.

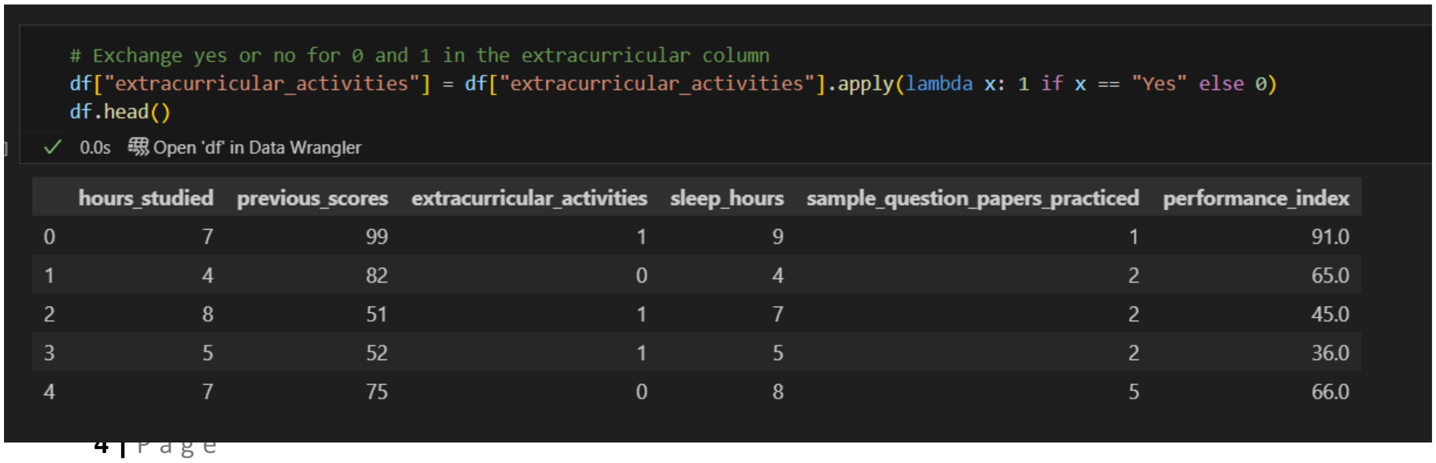
After that, we rename the columns to remove the spaces.



```
df.columns = ['hours_studied', 'previous_scores', 'extracurricular_activities', 'sleep_hours', 'sample_question_papers_practiced', 'performance_index']
df.head()
✓ 0.0s  Open 'df' in Data Wrangler
```

	hours_studied	previous_scores	extracurricular_activities	sleep_hours	sample_question_papers_practiced	performance_index
0	7	99	Yes	9	1	91.0
1	4	82	No	4	2	65.0
2	8	51	Yes	7	2	45.0
3	5	52	Yes	5	2	36.0
4	7	75	No	8	5	66.0

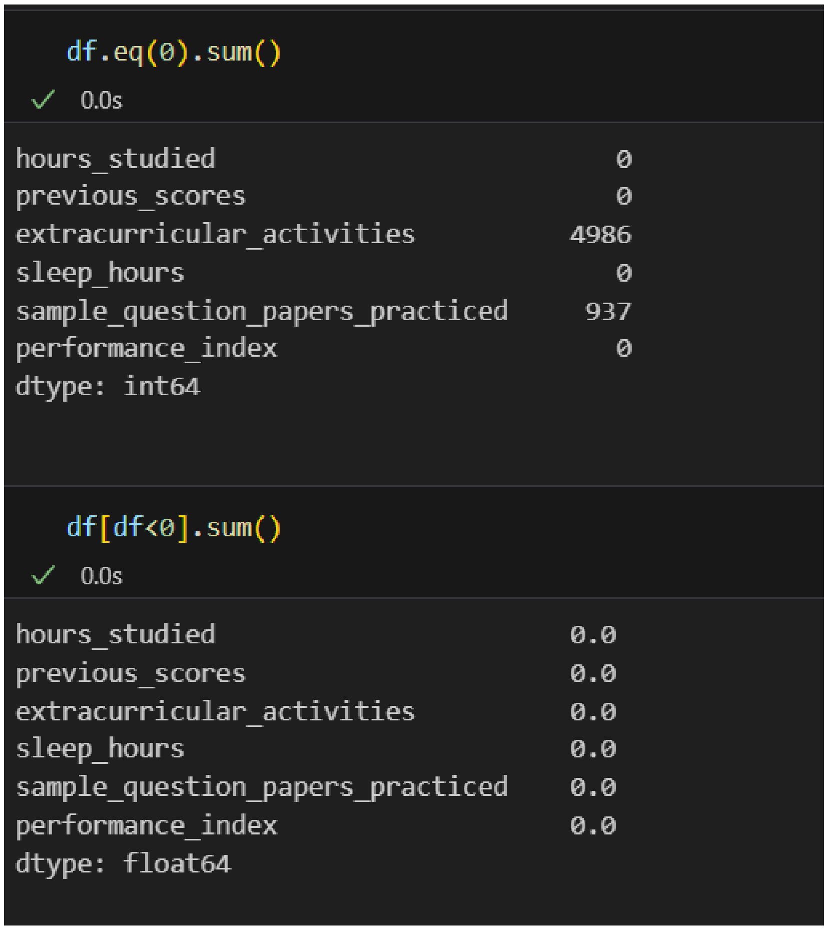
We also edit the extracurricular_activities column to numeric values instead of yes or no.



```
# Exchange yes or no for 0 and 1 in the extracurricular column
df["extracurricular_activities"] = df["extracurricular_activities"].apply(lambda x: 1 if x == "Yes" else 0)
df.head()
✓ 0.0s  Open 'df' in Data Wrangler
```

	hours_studied	previous_scores	extracurricular_activities	sleep_hours	sample_question_papers_practiced	performance_index
0	7	99	1	9	1	91.0
1	4	82	0	4	2	65.0
2	8	51	1	7	2	45.0
3	5	52	1	5	2	36.0
4	7	75	0	8	5	66.0

Then we check for zero or below zero values.

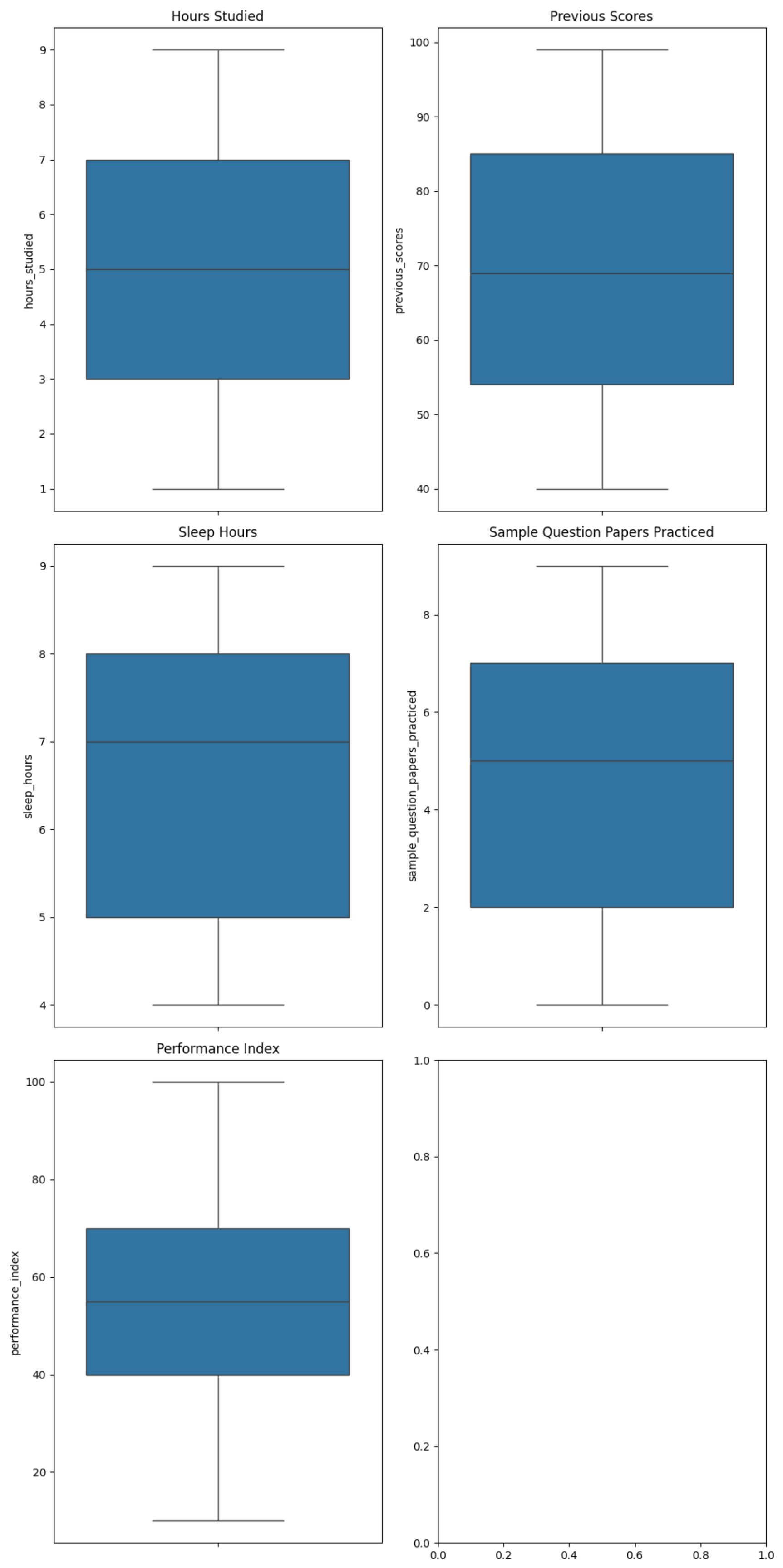


```
df.eq(0).sum()
✓ 0.0s

hours_studied                          0
previous_scores                        0
extracurricular_activities          4986
sleep_hours                            0
sample_question_papers_practiced     937
performance_index                      0
dtype: int64
```

```
df[df<0].sum()
✓ 0.0s

hours_studied                       0.0
previous_scores                     0.0
extracurricular_activities          0.0
sleep_hours                         0.0
sample_question_papers_practiced    0.0
performance_index                   0.0
dtype: float64
```

The columns with zero values are normal to have them.

Lastly, we check for outliers with box plots.

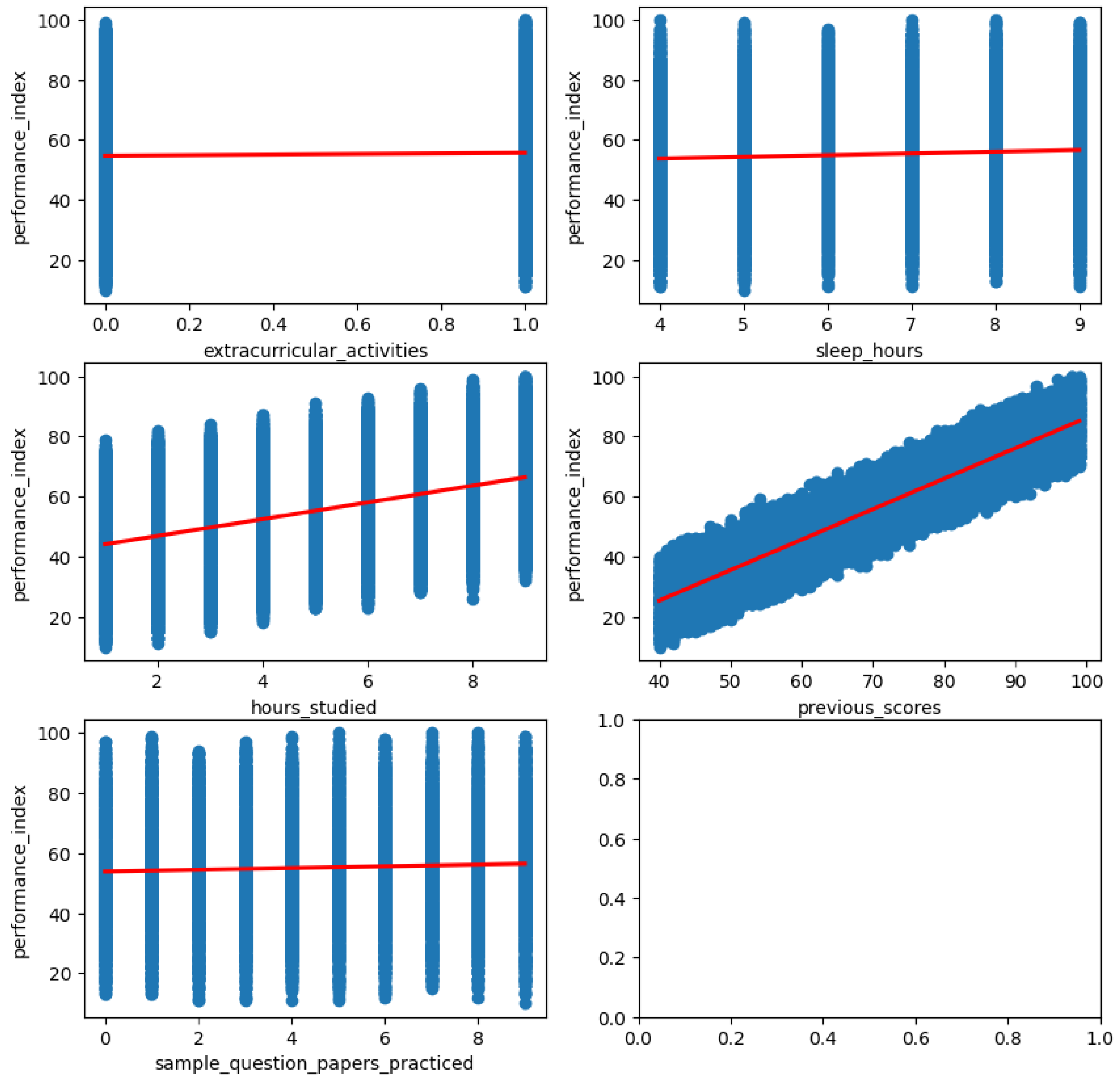


The box plots show no outlier values, so we are good to move ahead with the regression.

Feature selection

In this section we are selecting the features that affect the outcome of the performance of the students.

The first thing is to draw a scatter of the variables and visualize their relation to the target.

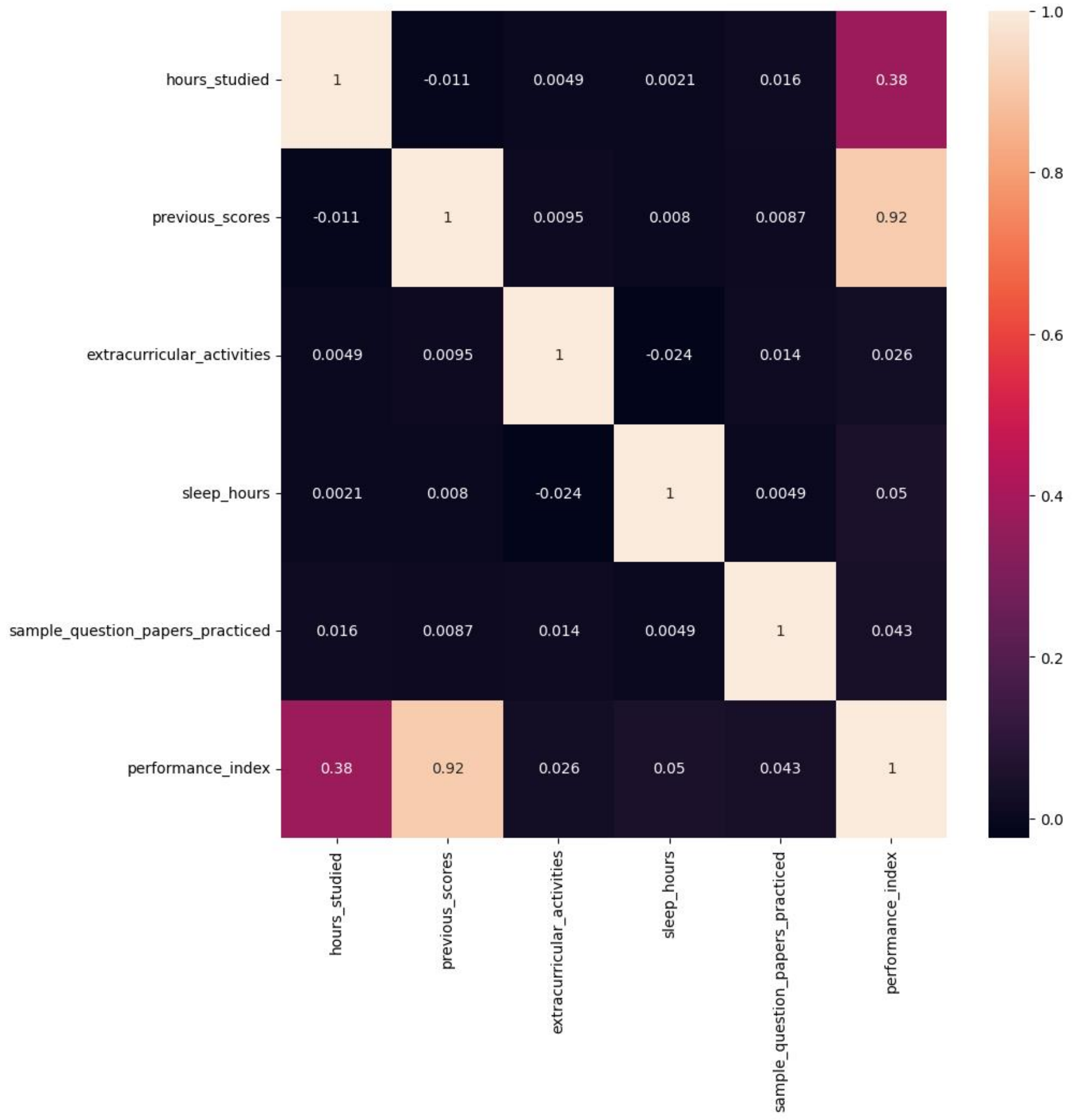


The plot shows a big correlation with the previous scores specially making it the dominant variable with the others contributing to smaller amounts.

We have two methods to choose our features.

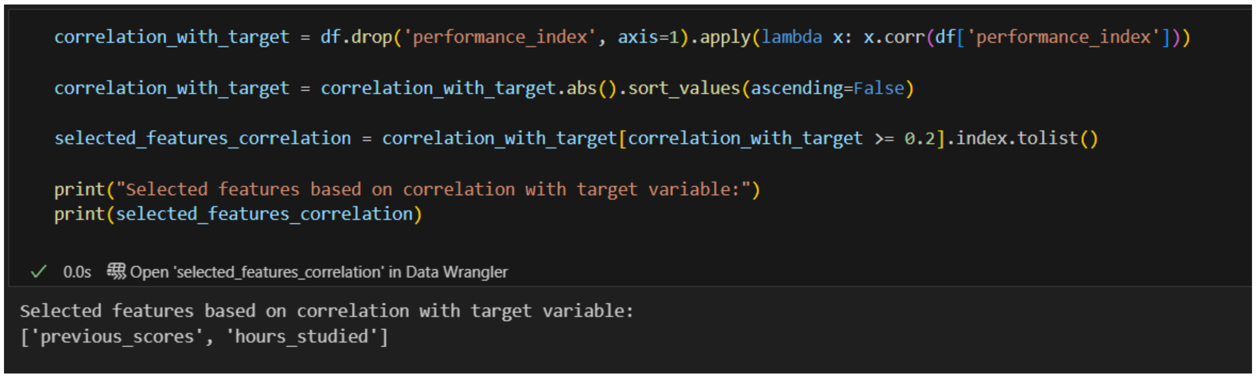
1. Correlation

To visualize the correlation between our variables we will draw a correlation matrix.



We can summarize from the matrix that the hours studied and the previous scores have the most impact on the outcome and that most of the independent variables are not correlated with one another.

We will choose to remove any variables with correlation with the target of less than 0.2.



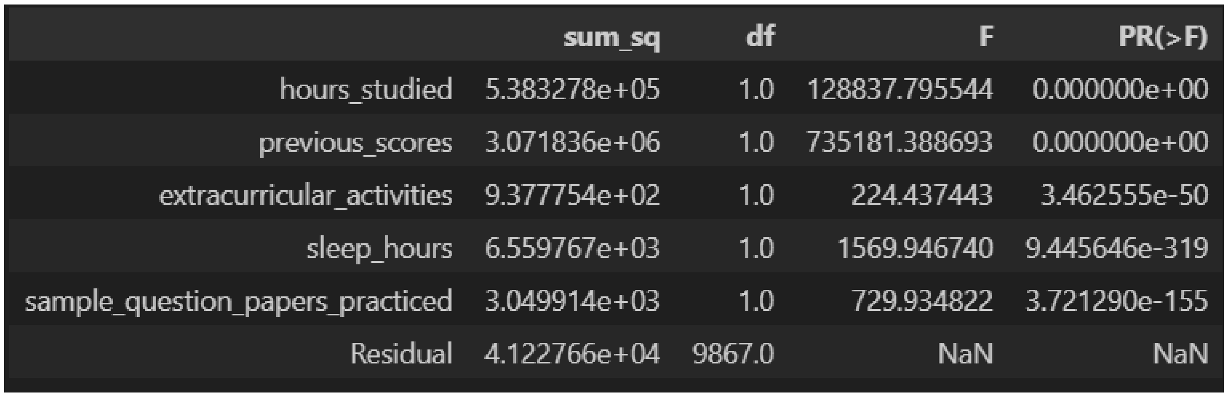
```
correlation_with_target = df.drop('performance_index', axis=1).apply(lambda x: x.corr(df['performance_index']))

correlation_with_target = correlation_with_target.abs().sort_values(ascending=False)

selected_features_correlation = correlation_with_target[correlation_with_target >= 0.2].index.tolist()

print("Selected features based on correlation with target variable:")
print(selected_features_correlation)
```

```
Selected features based on correlation with target variable:
['previous_scores', 'hours_studied']
```

This leaves us with previous scores and hours studied.

2. Anova Table

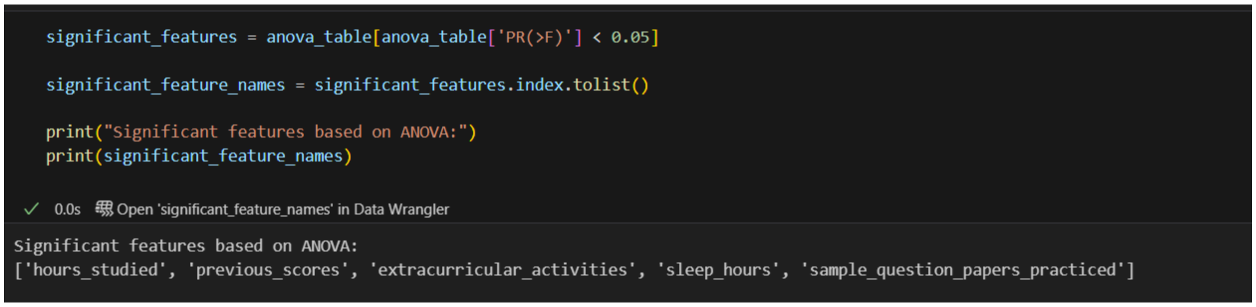
We start by constructing the table to find the p-values of each variable.



	sum_sq	df	F	PR(>F)
hours_studied	5.383278e+05	1.0	128837.795544	0.000000e+00
previous_scores	3.071836e+06	1.0	735181.388693	0.000000e+00
extracurricular_activities	9.377754e+02	1.0	224.437443	3.462555e-50
sleep_hours	6.559767e+03	1.0	1569.946740	9.445646e-319
sample_question_papers_practiced	3.049914e+03	1.0	729.934822	3.721290e-155
Residual	4.122766e+04	9867.0	NaN	NaN

As we can see all the variables have very low p-values which indicates that all of them are needed for the regression.

Taking a 95% confidence level we get that all the variables are needed.



```
significant_features = anova_table[anova_table['PR(>F)'] < 0.05]

significant_feature_names = significant_features.index.tolist()

print("Significant features based on ANOVA:")
print(significant_feature_names)
```

✓ 0.0s Open 'significant_feature_names' in Data Wrangler

```
Significant features based on ANOVA:
['hours_studied', 'previous_scores', 'extracurricular_activities', 'sleep_hours', 'sample_question_papers_practiced']
```

Now that we have two different outcomes we will do the model with both and then decide which one is better.

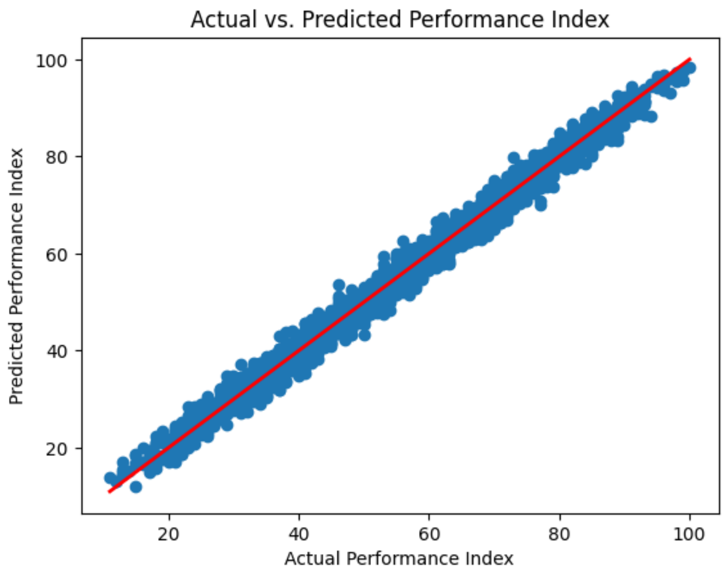
Models

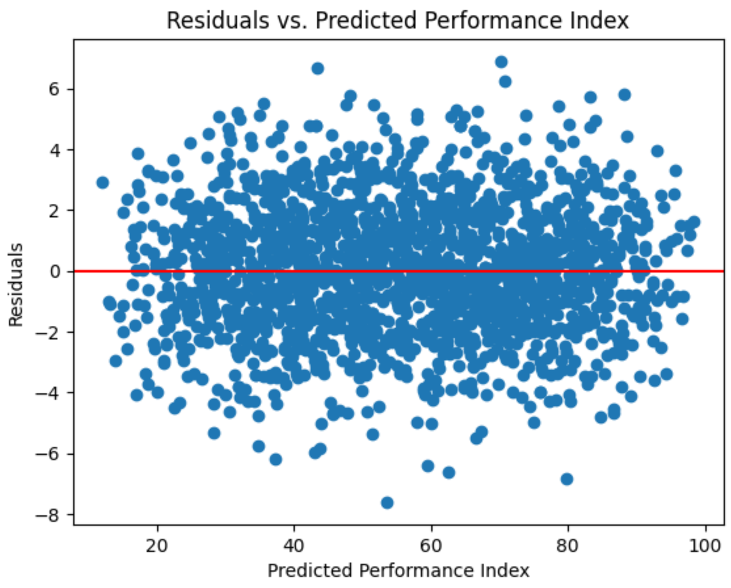
The first model we tried was without removing any features and the results we got was as follows:

The Mean Squared Error is 4.305900938538476.

The R2 score is 0.9884301209927054.

The adjusted R2 score is 0.9884007409038094.





Which indicates a very well fitted model.

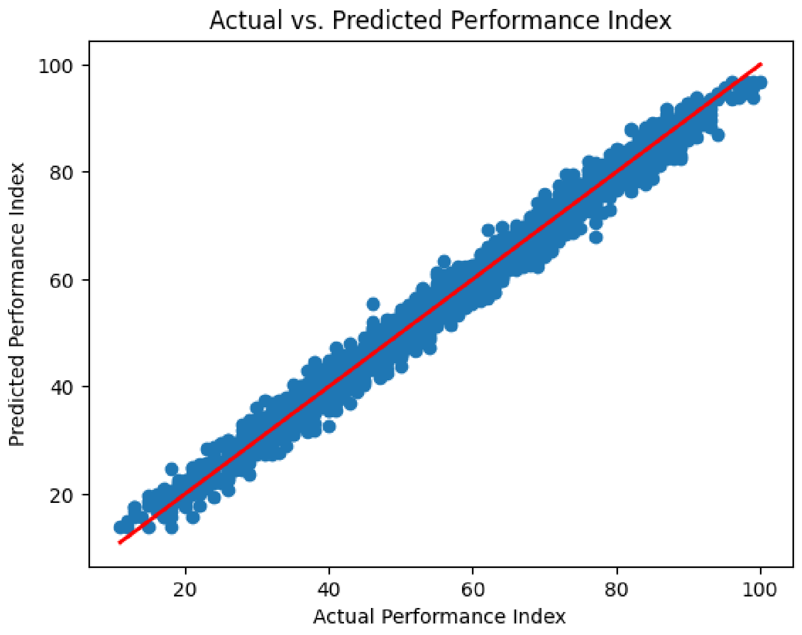
The second model we tried we only kept the hours studied and the previous scores.

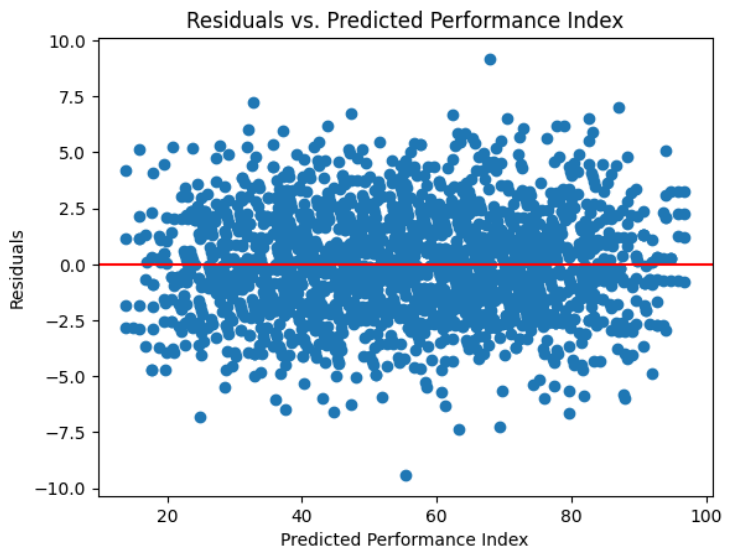
The results were :

The Mean Squared Error is 5.573764139536915.

The R2 score is 0.9850233951895029.

The adjusted R2 score is 0.9850082059351312.





Which does not vary a lot from the previous model.

Conclusion

After trying both models we reached the following points:

1. The two most significant variables were hours studied and previous scores as the model without them got an R2 adjusted with a difference of just 0.003.
2. The relation with the variables was very linear as the model was very well fitted.
3. The correlation between the previous scores and higher performance is because if a student is already excellent, he is more likely to stay so and vice versa.
4. While some variables like hours slept did not affect the performance in the regression, it does not mean that they do not have an outright impact on the longer term performance or other health reasons.

Conclusion