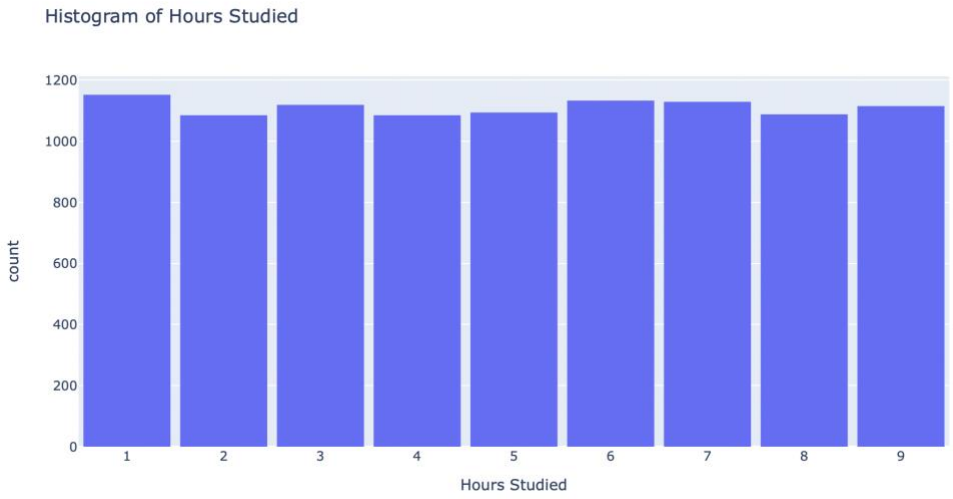
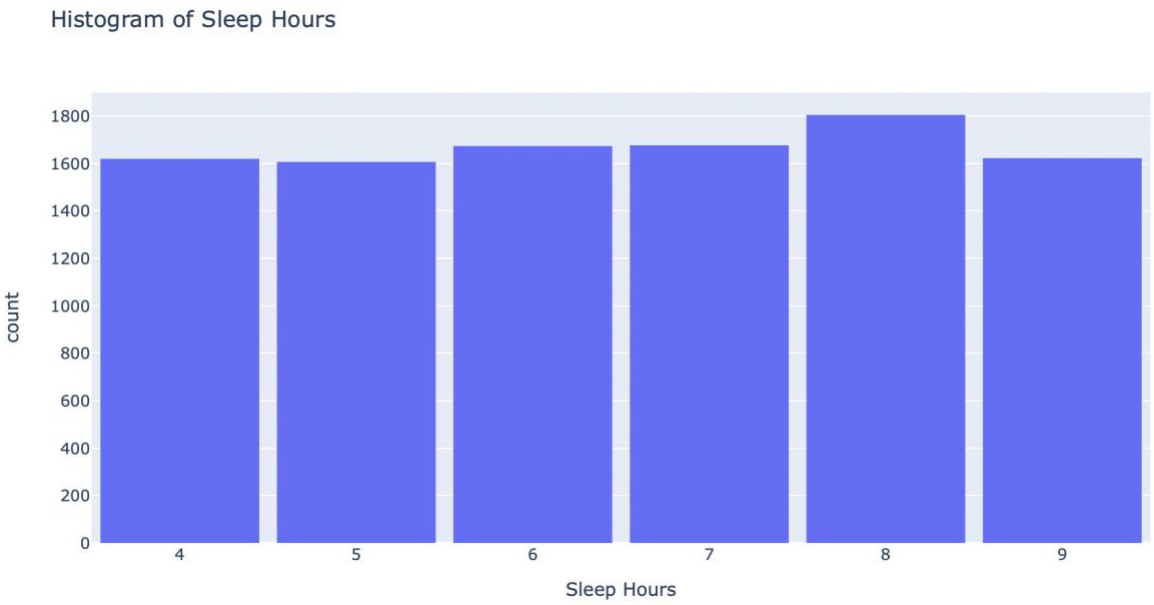
Student Performance Index Analysis

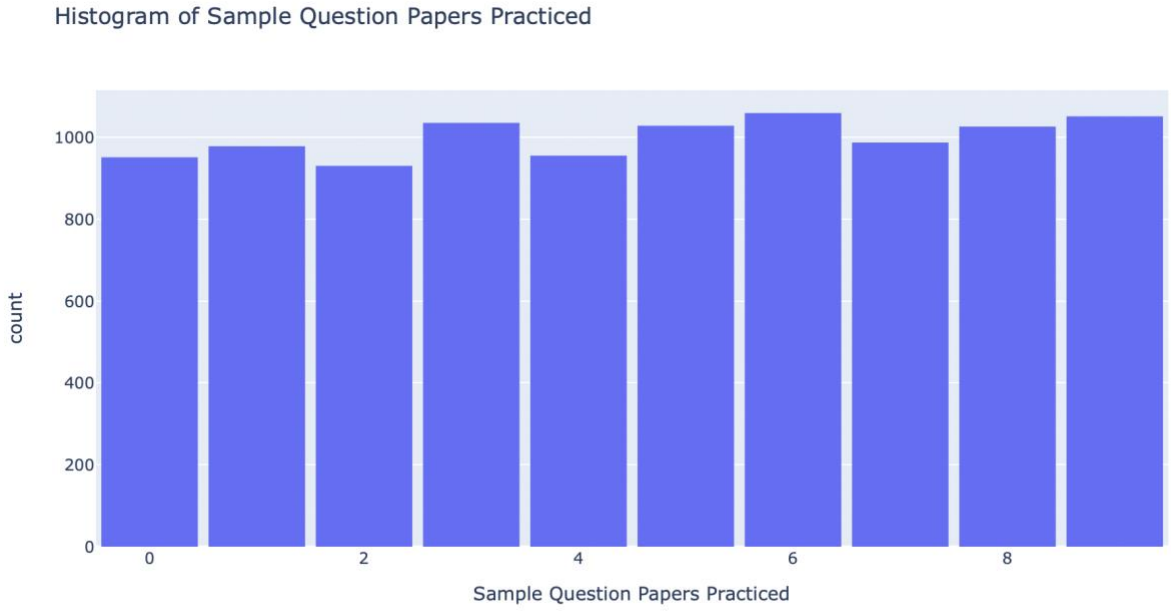
In this project, I used different Regression techniques in order to come up with the best model to predict Student Performance Index from a dataset on Kaggle.

EDA

Starting out, I did exploratory data analysis on the data set by looking at summary statistics for the variables Hours Studied, Previous Scores, Sleep Hours, and Sample Question Papers Practiced. In addition to the summary statistics, I made histograms to view the spread and distribution of the data for these variables as well. The plots can be seen below.







Following the analysis on those predictor variables, I also had to clean up the Extracurricular Activities predictor variable by changing it into a dummy variable for Yes/No. After creating the dummy variable and adding it into the original data frame, I dropped the 'No' column and renamed the 'Yes' column to 'EC Activities' where 0 signified no extracurricular activities and 1 signified extracurricular activities.

The final few steps of my EDA included a heatmap to analyze the correlation between the different variables, which can be seen below. As you can see in the diagram, the relationship between Hours Studied and Performance Index is relatively strong and positive, and the

relationship between Previous Scores and Performance Index is very strong and positive, something we will examine further later.

Lastly, I just wanted to see what a scatterplot would look like of Previous Scores and Performance Index, as the correlation between these two variables is very high so I wanted to see the strong, positive, linear relationship illustrated in graphical form.

Linear Regression

The first model I ran for this data set was a Linear Regression model where Performance Index was my y variable, and Hours Studied, Previous Scores, Sleep Hours, Sample Question Papers Practiced, and EC Activities were my x variables. My R Squared on the training data was .9886 and my R Squared on the testing data was .9891, but I wanted my model to be improved by scaling my independent variables as the variable values in this dataset were different in scale (Hours Studied, Previous Scores, etc.). I used standard scaler to do that, and my new model evaluation values were the following:

R Squared Value for Linear Regression Training Set	0.989
R Squared Value for Linear Regression Testing Set	0.989
MSE Value for Linear Regression Model	4.158

As we can see, the R squared values for both the training and testing sets are extremely high, as it is great how close the values are to 1. Additionally, the MSE value is not bad either, as the closer the value is to 0, the better.

KNN Regression

The next model I ran for this data set was KNN Regression, this time I focused on the scaled data only with the same y variable and x variables. I ran a for loop to determine the best k value to use based on the testing dataset results, and as we can see from the code the values settle around .977 at k = 10. I then ran a GridSearchCV to find the best parameter value for k, from k = 2 to k = 14. It came out with k = 14, but due to the values settling at around .977 at k = 11, I decided to use that to make the code less computationally taxing. After running the model with the scaled variables and k = 11 as my choice, the model evaluation values were the following:

R Squared Value for KNN Regression k=11 Training Model	0.981
R Squared Value for KNN Regression k=11 Testing Model	0.977
MSE Value for KNN Regression Model	8.707

As you can see, the R squared value on the testing dataset is lower than the one for Linear Regression, and the MSE value was also higher than what was had for Linear Regression. Overall, the model values are good, just not quite as good as the Linear Regression model.

Decision Tree Regression

I then decided to run a Decision Tree Regression Model on this dataset using the same scaled variables and x/y variables. The results of the model can be found below:

R Squared Value for Decision Tree Regression Training Model	0.999
R Squared Value for Decision Tree Regression Testing Model	0.975
MSE Value for Decision Tree Regression Model	9.39

For this model, the R Squared on the training set is very high, however of the 3 models ran so far, the R Squared on the testing set is the lowest. Additionally, the MSE value is the highest of the 3 models ran so far, indicating this is the worst model of the 3, even though this is still a good model with a relatively low MSE and an R Squared of .975 on the testing set.

Random Forest Regression

The last regression model I ran was a Random Forest Regression Model. I used hyperparameter tuning again, to determine the best value for the number of estimators. As we can see in the code, that value came out to be 120. After running the model, the results can be seen below:

R Squared Value for Random Forest Regression Training Model	0.998
R Squared Value for Random Forest Regression Testing Model	0.985
MSE Value for Random Forest Regression Model	5.564

This model has great results, as the R Squared on both the training and testing values are very close to 1. In addition to that, the MSE value is closer to 0 than on some of the prior models, at just over 5.

Conclusion

After building out all those models mentioned above, the two best models are the Linear Regression and Random Forest Regression models. They have the highest R Squared values on the testing data, and also have the lowest MSE values as well. Either of the two are good models to use to predict Student Performance Index based on Hours Studied, Previous Scores, Sleep Hours, Sample Question Papers Practiced, and Doing Extracurricular Activities as the percentage of variation of Student Performance Index explained by the variation in all of the features is either .985 or .989—roughly .99.

Appendix

I decided to do some further investigation into model construction using the two best models (Linear Regression and Random Forest Regression) using only the most important features. This could be useful for larger datasets or datasets where we can't use all of our features. In order to do this, I ran both models the same as I did before, just with Hours Studied and Previous Scores as the independent variables, as the importance scores for those two variables were the highest. The new model results can be found below:

R Squared Value for New Linear Regression Training Set	0.986
R Squared Value for New Linear Regression Testing Set	0.986
MSE Value for New Linear Regression Model	5.232

R Squared Value for New Random Forest Regression Training Model	0.987
R Squared Value for New Random Forest Regression Testing Model	0.985
MSE Value for New Random Forest Regression Model	5.69

As we can see, the new model results for Linear Regression are very slightly worse on the training set, testing set, and MSE value, but they are still very respectable and prove the model is still very effective. For the Random Forest Regression, the results are also very slightly worse across the board, other than the R Squared on the testing set which is the same as the older model with more x variables. Overall though, if needed the new models with only the important features would be more than satisfactory to use as they can provide great results while requiring less computational power.