Machine Learning Observation:

**Math Scores**

* **Labels:** “math score”
* **Features:** dropping “math score”
* **Techniques:**
* RandomUnderSampler – Reduces the number of samples in the majority class.
* Balance class distribution.
* DecisionTreeRegressor – Split the data into subsets and minimize the variance (label/target)
* **Split data:** Split data into training and testing to estimate the performance of a machine learning model on new data.
* **R-Squared:** Statistical measure to evaluate the performance of a ML model (regression).
* **Summary:** The aim of the ML model was to demonstrate its ability to accurately predict the math scores of students based on the given features. The model achieved an R-squared score of 0.82, indicating that it can explain 82% of the variance in the target variable. However, there was an unexplained variance of 0.08, which suggests that there may be other factors that affect the math scores of students that were not included in the model. Overall, the model can be used to predict the math scores of students based on the given features.
* **Evaluation:** To evaluate the performance of the ML model, we tested it with a new set of features. The model predicted a math score of 81 for the given features, but the actual score was 82. This suggests that the model was not able to accurately predict the math score for this particular data. By testing the model with new data, we can determine how well it performs on unseen instances. In this case, the model has a relatively high R-squared score, but it was not able to accurately predict the math score for this particular data. This highlights the importance of evaluating the model with new data and not relying solely on the R-squared score to assess its performance.

**Reading Scores**

* **Labels:** “reading score”
* **Features:** dropping “reading score”
* **Techniques:**
* RandomOversampler – Increases the number of samples in the minority class. (Duplicates)
* Gave a higher R-squared.
* DecisionTreeRegressor
* **Split data:**
* **R-Squared:**
* **Summary:** The aim of the ML model was to demonstrate its ability to accurately predict the reading scores of students based on the given features. The model achieved an R-squared score of 0.98, indicating that it can explain 98% of the variance in the target variable. However, there was an unexplained variance of 0.02, which suggests that there may be other factors that affect the reading scores of students that were not included in the model. Overall, the model can be used to predict the reading scores of students based on the given features.
* **Evaluation:** To evaluate the performance of the ML model, we tested it with a new set of features. The model predicted a reading score of 90 for the given features, but the actual score was 91. This suggests that the model was not able to accurately predict the reading score for this particular data. By testing the model with new data, we can determine how well it performs on unseen instances. In this case, the model has a relatively high R-squared score, but it was not able to accurately predict the reading score for this particular data. This highlights the importance of evaluating the model with new data and not relying solely on the R-squared score to assess its performance.

**Writing Scores**

* **Labels:** “writing score”
* **Features:** dropping “writing score”
* **Techniques:**
* RandomUnderSampler:
* DecisionTreeRegressor:
* **Split data:**
* **R-Squared:**
* **Summary:** The aim of the ML model was to demonstrate its ability to accurately predict the writing scores of students based on the given features. The model achieved an R-squared score of 0.98, indicating that it can explain 94% of the variance in the target variable. However, there was an unexplained variance of 0.06, which suggests that there may be other factors that affect the writing scores of students that were not included in the model. Overall, the model can be used to predict the writing scores of students based on the given features.
* **Evaluation:** To evaluate the performance of the ML model, we tested it with a new set of features. The model predicted a writing score of 96 for the given features, but the actual score was 88. In this case, the model has a relatively high R-squared score, but it was not able to predict the writing score for this particular data. The model has overfit the training data, which means it has learned the noise in the data instead of the underlying patterns.

**Things to Consider**

* Use different models, algorithms, techniques.
* Random Forest – Combination of decision trees to create a more accurate model.
* Gradient Boosting – Combination of weak models to create a more accurate model.
* Support Vector Regression – Finding the best decision boundary that separates the classes in all possible features.
* Use only the most important features that will have the most impact on the label/target.
* Use/collect more data.

**Further Evaluation**

* We developed a Flask and HTML to handle fresh data and make predictions on test score performance using the machine learning models built.
* Dataset size: 10
* **Evaluation:** Based on our results, the math score prediction model had a 70% accuracy rate, the reading score prediction model had a 90% accuracy rate, and the writing score prediction model had a 100% accuracy rate.