**Data Analysis Report on PRCP-1026-** **Teaching Assistance**

**Project Members: Sameera Tanveer,**

**Arunmozhi Muthamizhan**

**Project Team ID: PTID-CDS-OCT-24-2112**

**Date: 15-11-2024**

**Table of Contents:**

1. Introduction

2. Data Preprocessing & Feature Engineering

3. Modelling

4. Conclusion

**Introduction:**

1. The purpose of this report is to provide a complete explanation of the data analysis of the given project.
2. **Overview of the dataset:**
3. The provided dataset concerns the teaching assistant evaluation, which is a classification problem and contains the assessment data.
4. The domain of the dataset is “Education”. The data consists of teaching performance over three regular and two summer semesters.
5. The dataset contains 151 records and 6 columns (attributes/characteristics).
6. Whether of not the TA is a native English speaker (binary),1=English speaker, 2=non-English speaker.
7. Course instructor (categorical, 25 categories)
8. Course (categorical, 26 categories)
9. Summer or regular semester (binary) 1=Summer, 2=Regular
10. Class size (numerical)
11. Class attribute (categorical) 1=Low, 2=Medium, 3=High
12. Missing Attribute Values: None

**Objective:**

The main objective of the project is to build a model that can accurately predict the performance of the students.

**2. Data Preprocessing & Feature Engineering:**

**1. Adding Column Names to the Data Frame:** The dataset does not have Column Names. The column names were provided in the text file along with the metadata of the attributes. The first step was to set the column names of the data frame.

**2. Handling Missing Values:** No null values are present in the data set.

**3. Check for the duplicates:** There is a total of 41 duplicated values in this dataset.

**4. Univariate Analysis:** Performance, Summer regular and speaker’s native are counted using the countplot. Course, Class size Corse instructor are counted using the histplot.

**5. Skewness:** Skewness of the entire data has been calculated.

**6. Dealing with the Outliers:** We have used the Visualization technique of Box Plots to detect the possible candidates of Outliers. There are no outliers present in the data.

**7. Correlation:** Using the corr() function,the correlation of the data has been interpreted. Using a heatmap, the multicollinearity has been achieved.

**3. Modelling:**

**1. Train\_test\_split:** To randomly split the data into training and testing sets, the train test split has been imported from sklearn model selection. By doing this we prevented the overfitting and underfitting. Based on the given test size, the model’s performance has been evaluated.

**2. Scaling:** To speed up the training and improve the model performance scaling is done using the testing and training set.

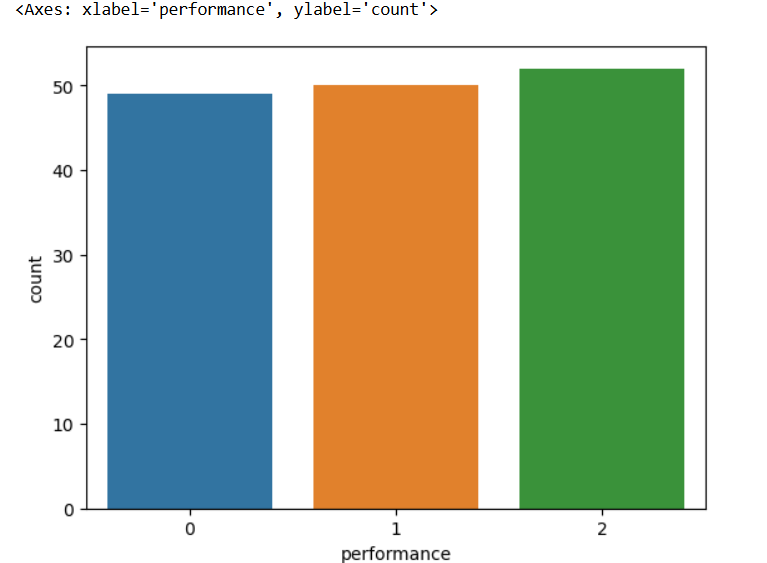
**3. Model Comparison:** To find the suitable model for the task using various classifiers.

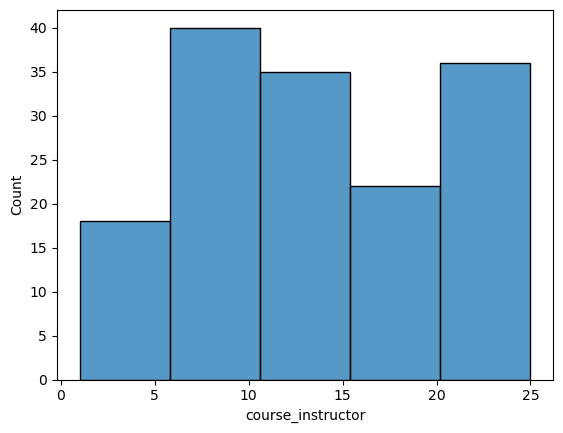
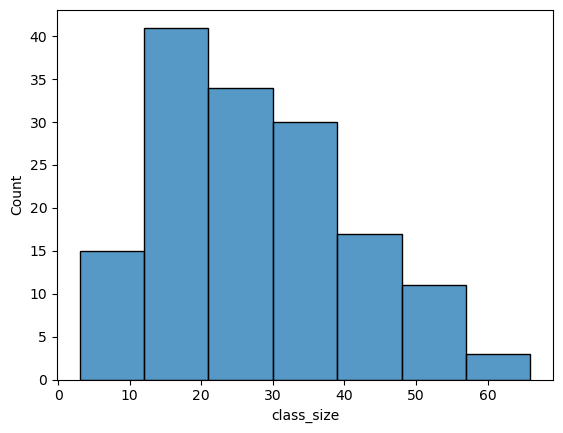
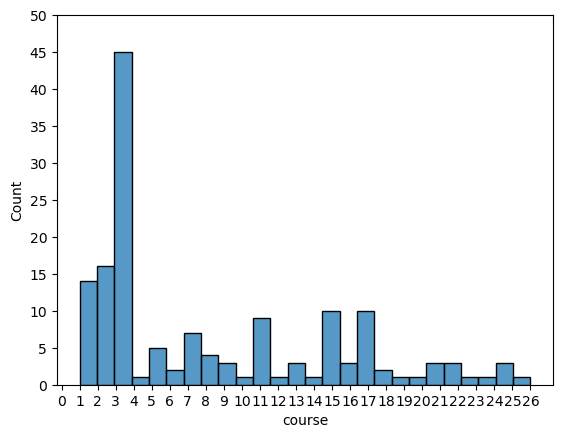
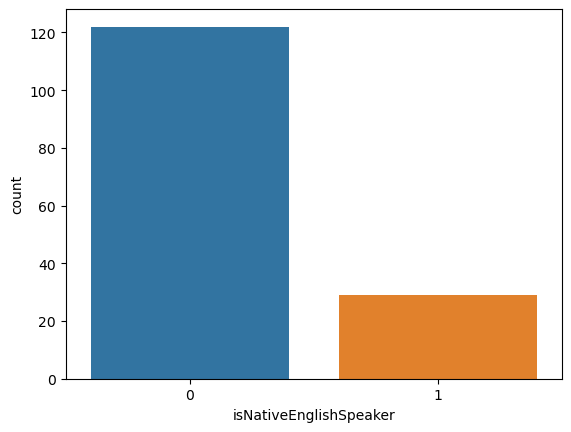
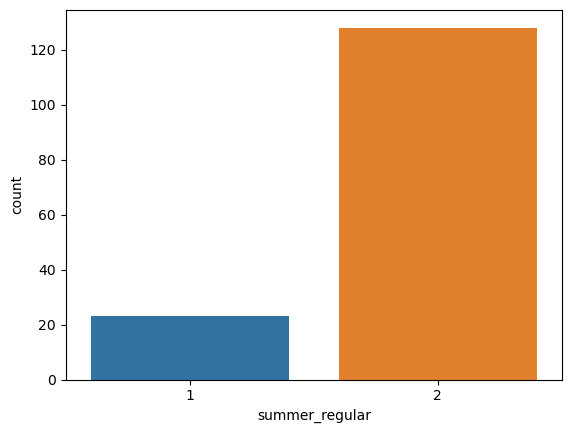
1. **Decision Tree Classifier:** using the decision tree classifier, we achieved an accuracy score of 64.5%.
2. **Random Forest Classifier:** In random forest we achieved an accuracy of 70.9%
3. **Gradient Boosting Classifier:** Using gradient boosting, we got 74.19 %.
4. **XGB Classifier:** Extreme Gradient Boosting, an optimized implementation provided by the xgboost is 70.9%.

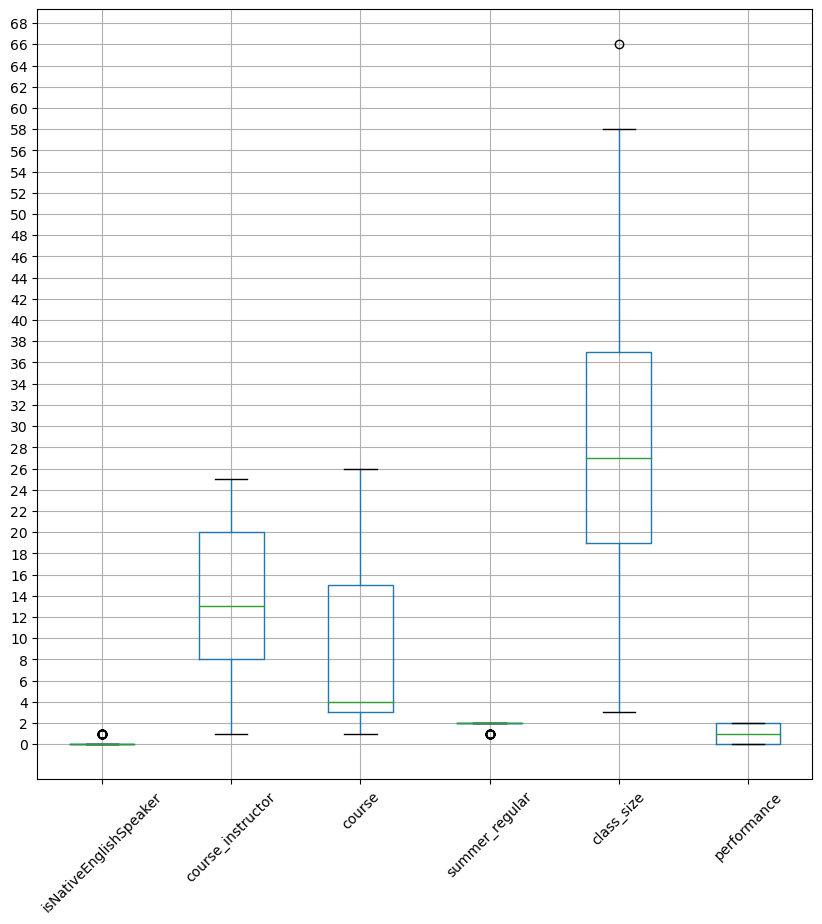
Even after the feature transformation, we achieved the same results using the classifiers as mentioned above.

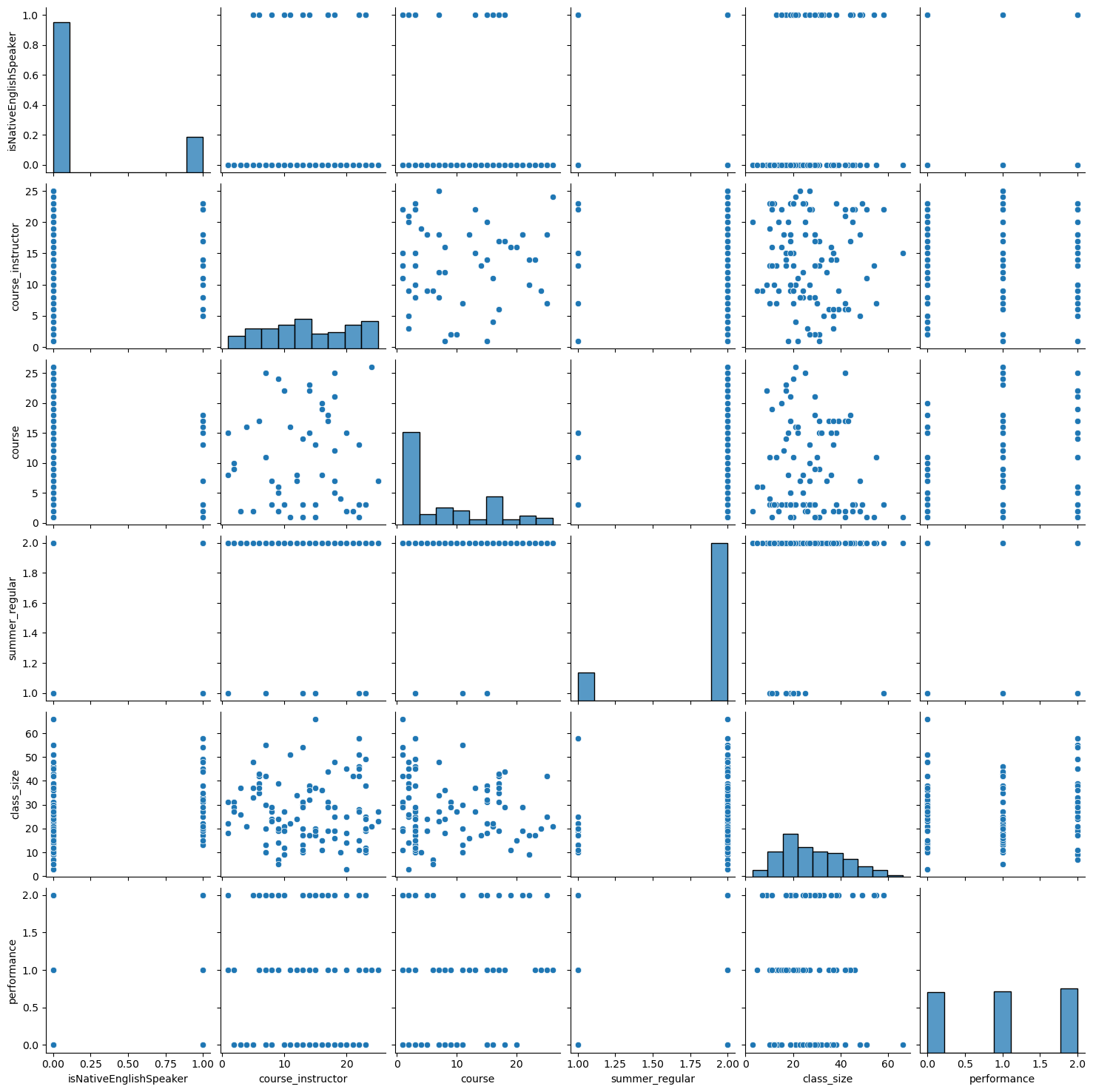
**4. Hyperparameter Tuning:** After hyperparameter tuning for the gradient boosting, the accuracy score is just 29.03% which is much less than the original model’s performance.

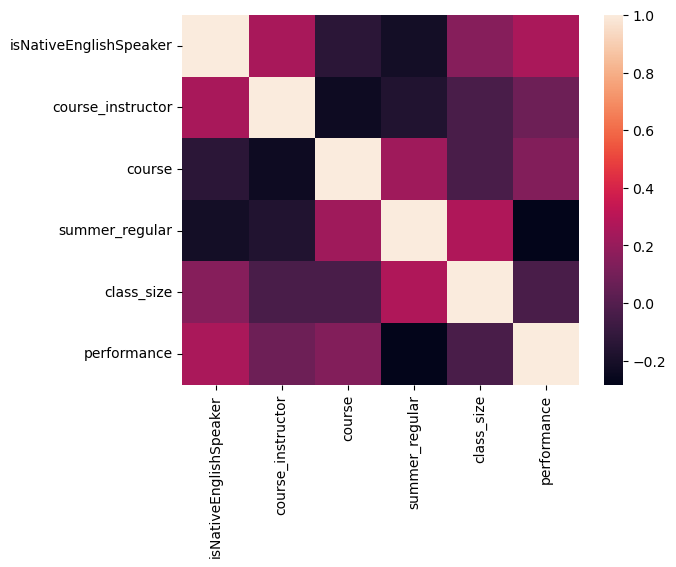
**Visualizations and Insights:**

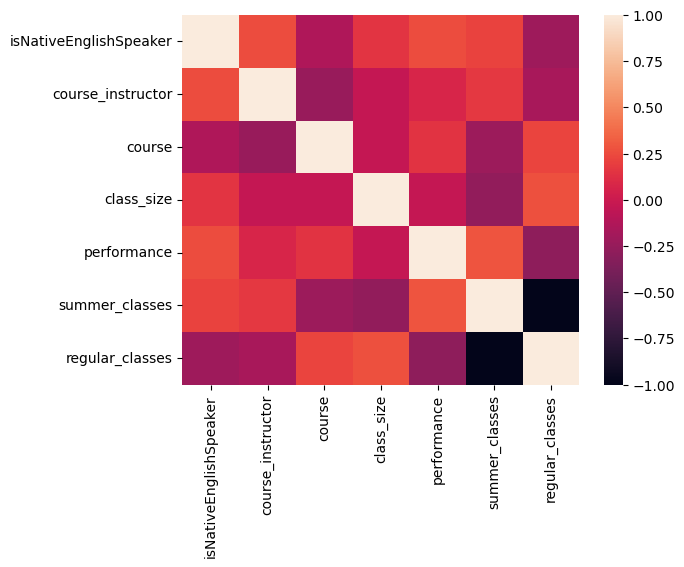












**Challenges faced with the data set:**

1. Column names were in the other file and an explanation about the data is attached in txt file.

2. The best accuracy achieved is 74.19 using the gradient boosting classifier.

3. The accuracy is dropping after the hyperparameter tuning.

**Conclusion**:

The given data set is properly explored and the best-fitting model is found after implementing the different classifiers. The best fitting model for this data set is the ‘**Gradient Boosting Classifier**’