1. **Introduction:**

The Teaching Assistant (TA) Performance Evaluation dataset contains data from 151 TA assignments at the University of Wisconsin-Madison, including attributes like native language, instructor, course, semester type, and class size. The target variable categorizes TA performance into three levels: low, medium, and high, based on these factors. The goal is to predict the performance level of a teaching assistant based on various attributes such as whether the TA is a native English speaker, the instructor they worked with, the course they assisted in, the semester type, and the class size. The target variable is the class attribute, which classifies the TA's performance into three categories: low, medium, or high.

**2. Dataset Overview:**

1. The dataset includes 6 explanatory variables describing different aspects of teaching assistance.
2. Class Attribute is the target variable, which represents the performance of the teachers.

**Attribute Information:**

1. Native Teacher: Whether or not the TA is a native English speaker (binary); 1=English

speaker, 2=non-English speaker

2. Instructor: Course instructor (categorical, 25 categories)

3. Course (categorical, 26 categories)

4. Semester: Whether instructor took classes in Summer (1) or Regular (2)

5. Class size (numerical): How many participants joined the session.

6. Class attribute (categorical) 1=Low, 2=Medium, 3=High

**3. Data** **Summary**:

Number of records: 151

Number of features: 6

**Data Types:**

1. Numerical Features - 6

**Null Values:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null Values** | **Type** |
| Native Teacher | 0 | Numerical |
| Instructor | 0 | Numerical |
| Course | 0 | Numerical |
| Semester | 0 | Numerical |
| Class Size | 0 | Numerical |
| Class Attribute | 0 | Numerical |

**4. Data Cleaning and Preprocessing:**

* 1. Group by Native Teacher and evaluate Class attribute.
  2. Label encode the target variable (Class attribute).
  3. One-hot encode categorical columns like Instructor and Course.

**5. Exploratory Data Analysis:**

**Correlation Analysis**

* Identified strong correlations between Class Attribute and features like Semester, Course, Class Size, Native Teacher and Instructor.

**Data Visualization**

* Distribution of Class Sizes (Histplot, Boxplot)
* Feature relationships (Scatter plots, Heatmaps)
* Multivariate analysis using pair plot.

**6. Modeling & Prediction**

**Model Selection**

* Implemented various models:
  + Logistic Regression
  + Random Forest
  + Gradient Boosting Classifier
  + Support Vector Machine

**Model Evaluation**

* Performance measured using:
  + Accuracy
  + Precision
  + Recall
  + F1-Score
* Best performing model: **Random Forest Classifier** shows the highest accuracy point.

**7. Model Comparison Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| Logistic Regression | 0.68 | 0.64 | 0.82 | 0.72 |
| **Random Forest** | **0.74** | **0.71** | **0.91** | **0.80** |
| Gradient Boosting | 0.74 | 0.75 | 0.82 | 0.78 |
| Support Vector Machine | 0.52 | 0.46 | 0.55 | 0.50 |

* **Best Model Recommendation**: Random Forest

**8. Flask Deployment Link**

<http://127.0.0.1:5000/predict>

**9. Challenges Faced and Solutions**

**Data Quality Issues**

* Categorical Data Transformation: Applied label encoding for the target variable and one-hot encoding for categorical features like Instructor and Course.
* Class Imbalance: Addressed using SMOTE for oversampling and class weight adjustment during model training.
* Multicollinearity: Managed by conducting correlation analysis and removing or combining highly correlated features.

**Model Performance**

* Model Selection & Tuning: Implemented multiple models and used cross-validation and grid search for hyperparameter optimization.
* Overfitting: Mitigated using pruning techniques and tuning parameters like max depth and n\_estimators in tree-based models.
* SVM Underperformance: Optimized kernels and regularization parameters to improve SVM results.

**10. Conclusion**

* The Teaching Assistant Performance Evaluation project successfully predicted TA performance based on various factors such as native language, instructor, course, semester type, and class size.
* Through comprehensive data cleaning, preprocessing, and exploratory data analysis, we identified key correlations and relationships within the dataset.
* Various machine learning models, including Logistic Regression, Random Forest, Gradient Boosting, and Support Vector Machine, were tested, with Random Forest demonstrating the highest accuracy and overall performance.
* Challenges such as handling categorical data, managing class imbalance, and addressing overfitting were effectively overcome using techniques like encoding, SMOTE, and pruning.
* The Random Forest model was recommended as the best-performing model for predicting TA performance.