MODEL OPTIMIZATION AND HYPERPARAMETER TUNING REPORT.

INTRODUCTION.

Model optimization and hyperparameter tuning are crucial for enhancing machine learning performance. This process includes identifying key hyperparameters and making manual adjustments to assess their impact. Automated techniques such as grid search, random search, and Bayesian optimization streamline this exploration. Advanced ensemble methods, like stacking, blending and voting classifiers, leverage multiple models' strengths. Additionally, understanding feature importance through SHAP values and partial dependence plots aids in refining models. Ultimately, the goal is to select the final model based on performance, interpretability, and computational efficiency.

Detailed results of different tuning methods

- **Grid Search**: The performance metrics for the XGBoost model (Grid Search) indicate strong predictive capabilities. With an accuracy of approximately 88.26%, the model correctly identifies a high proportion of instances. The precision of 87.84% suggests that most positive predictions are accurate, while the recall of 84.67% indicates that the model captures a significant portion of actual positive cases. The F1-Score of 85.97% reflects a good balance between precision and recall, highlighting the model's effectiveness in handling the classification task.

- **Random Search**: The performance metrics for the XGBoost model (Random Search) demonstrate improved predictive performance compared to the Grid Search variant. With an accuracy of approximately 89.62%, the model effectively classifies a high percentage of instances. The precision of 89.53% indicates that nearly all positive predictions are correct, while the recall of 86.22% shows that the model identifies a substantial portion of actual positive cases. The F1-Score of 87.59% reflects a strong balance between precision and recall, underscoring the model's robustness in classification tasks.

- **Bayesian Optimization (Hyperopt):** The output shows the results of five hyperparameter optimization trials, each with 50 iterations. The best loss values indicate the model's performance during each trial, with lower values representing better performance. The trials varied significantly in duration, with the fastest taking about 11 seconds per trial and the slowest taking over 22 minutes. The best loss achieved was approximately -0.8791 in the last trial, indicating an improvement in model performance. These results highlight the effectiveness of hyperparameter tuning in optimizing the model's predictive capabilities.

- **Stacking Classifier**: this help achieved a validation accuracy of 85.07%, indicating that it performs well on the validation dataset. The test accuracy is slightly higher at 86.68%, suggesting that the model generalizes effectively to unseen data. These results demonstrate the model's robustness and its ability to maintain performance across different datasets, highlighting its effectiveness in the classification task.

- **Blending Classifier**: This Classifier was able to achieve a validation accuracy of 88.01%, indicating strong performance on the validation dataset. The test accuracy is even higher at 88.71%, suggesting excellent generalization to unseen data. These results reflect the model's effectiveness and reliability in making accurate predictions, showcasing its advantages over other classification approaches.

- **Voting Classifier**: The Voting Classifier achieved a validation accuracy of 87.33%, indicating solid performance on the validation dataset. Its test accuracy of 88.26% suggests that the model generalizes well to unseen data. These results demonstrate the model's effectiveness in combining predictions from multiple classifiers, leading to reliable and accurate outcomes in classification tasks.

Analysis of hyperparameter impact on model performance

The analysis of hyperparameter tuning reveals significant impacts on model performance across various algorithms. In Logistic Regression, adjusting the regularization strength (C) balances overfitting and underfitting, enhancing interpretability. For Decision Trees, parameters like max_depth, min_samples_split, and min_samples_leaf improve accuracy and reduce variance. In Random Forests, increasing n_estimators and optimizing tree complexity through max_depth enhances stability and predictive power. SVM performance benefits from tuning C, kernel, and gamma, while XGBoost shows improved accuracy with adjustments to n_estimators, learning_rate, and max_depth. Overall, careful tuning of hyperparameters is essential for maximizing model effectiveness and generalization.

Feature importance plots

The analysis of feature importance across different models highlights key influences on decision-making. In the Random Forest model, features like "Curricular units 2nd sem (approved)" and financial factors such as "Tuition fees up to date" are most significant, with importance values ranging from 0.00 to 0.14. The Decision Tree model similarly prioritizes "Curricular units 2nd sem (approved)" and "Tuition fees up to date," with a wider range of values from 0.00 to 0.40. XGBoost emphasizes "Curricular units 2nd sem (approved)" as critical, along with "Tuition fees up to date," showing values from 0.00 to about 0.20. Overall, academic performance indicators and financial factors dominate feature importance across all models.

SHAP summary and dependency plots

The SHAP (SHapley Additive exPlanations) plot visually represents the impact of various features on model predictions. Features are displayed on the vertical axis with their SHAP values on the horizontal axis, indicating their contributions. Positive SHAP values enhance the likelihood of positive predictions, while negative values reduce it. The density of points shows the distribution of feature values, with features like Feature 30 and Feature 24 demonstrating varied impacts. A color gradient indicates feature values, aiding in understanding their correlation with SHAP values. Overall, the plot highlights key features, particularly Feature 30 and Feature 16, which are crucial for model interpretability.

Partial dependence plots for key features

The Partial Dependence Plot (PDP) analysis across Random Forest, Decision Tree, and XGBoost models highlights the influence of three features: x0, x1, and x2. In all models, x0 shows a stable relationship with minimal impact on predictions. x1 demonstrates a more dynamic relationship, indicating moderate influence. In contrast, x2 consistently exhibits a strong positive correlation, significantly affecting predictions. Overall, x2 is identified as the most influential feature, followed by x1, while x0 has limited impact across all models.

CONCLUSION

The evaluation of classifiers reveals that Random Forest achieved the highest accuracy (0.8894) and precision (0.8908), although its recall (0.7465) indicates some missed positive cases. XGBoost follows closely with an accuracy of 0.8849 and precision of 0.8699, demonstrating balanced performance. The Decision Tree model showed the lowest accuracy (0.8375) and precision (0.7500), suggesting a need for refinement. Ultimately, Random Forest is selected as the final model due to its strong predictive capabilities, interpretability through feature importance, and computational efficiency. XGBoost serves as an alternative model, excelling in handling complex patterns but requiring more tuning. Overall, Random Forest is deemed the most robust choice for deployment.