Technical Documentation: Student Dropout Prediction Pipeline

Overview

This report details the end-to-end machine learning pipeline developed to predict student dropout rates. The pipeline includes data preprocessing, feature engineering, model development, evaluation, and deployment using a Random Forest model. The model was integrated into a user-friendly web application via **Streamlit** to provide actionable predictions. This report also covers the ethical considerations and limitations associated with the model. The main goal of this project is to build a predictive model to identify students at risk of dropping out, based on academic, demographic, and financial features. The predictions can help educational institutions intervene early, providing support to at-risk students and improving retention rates.

Data Preprocessing.

Data preprocessing is crucial for cleaning and preparing the raw data before applying machine learning algorithms. The following steps were taken to achieve that:

- Handling Missing Values: Imputation of Missing values were done for numerical features using mean values and for categorical features using mode values. Data Type Corrections was done on Date fields and categorical variables to properly format and ensure that they could be correctly interpreted by the model.
- Data Transformation: Normalization of Features such as grades and the number of approved curricular units was done using **MinMaxScaler** to ensure all variables had a uniform scale. Categorical Encoding of Features like "Tuition fees up to date" and "Age at enrollment" were label-encoded into numerical values.
- Statistical Analysis: Descriptive Statistics which shows a summary of each feature, including mean, median, and standard deviation, was computed. For the Correlation Analysis, Pearson and Spearman correlations were used to identify relationships between features. Lastly, for the Hypothesis Testing, Chi-square and t-tests were conducted to test significant associations between features and the target variable (dropout status).

Data Exploration and Visualization

- Univariate Analysis**:** Histograms and Box Plots were created for each numerical feature to understand the distribution of the data and identify potential outliers while Bar Charts were Generated for categorical features such as "Tuition fees up to date" to visualize class distribution.
- Bivariate and Multivariate Analysis: Scatter Plots was used to examine relationships between pairs of numerical variables, such as grades and approved units. Correlation Heatmaps created to visualize the correlation between numerical features to detect multicollinearity. Chi-Square Tests was conducted to assess relationships between categorical features and the target variable.
- Advanced Visualization: Interactive Visualizations were created with Plotly to allow users to explore data interactively and developed using Streamlit to present key insights visually, including feature distributions and correlations with dropout risk.

Feature Engineering and Selection

- Feature Creation: New features such as study time per credit and time since last evaluation were created to enhance the predictive power of the model.

- Feature Transformation: Log Transformation was applied to skewed numerical features like grades to reduce skewness.
- Binning: Continuous features like age were binned into categorical age groups to better capture non-linear relationships.
- Feature Selection: For Filter Methods, Correlation analysis was used to remove highly correlated features that provided redundant information. Embedded Methods, The importance of each feature was ranked using the built-in feature importance measure of Random Forest. Also for Wrapper Methods, Recursive Feature Elimination (RFE) was applied to identify the most impactful feature subsets. Lastly, for Dimensionality Reduction, Principal Component Analysis was used to reduce dimensionality and visualize high-dimensional data.

Model Development and Training

- Data Splitting: The dataset was split into training (80%), validation (10%) and testing (10%) sets to ensure that the model was robust and generalizable.
- Baseline Model: A Logistic Regression model was used as a baseline. Its performance was measured using accuracy, precision, recall, and F1-score. This baseline helped to benchmark the performance of more complex models.
- Traditional Machine Learning Models: After comparing multiple models (e.g., Decision Trees, SVMs), Random Forest was selected as the final model due to its superior performance in handling non-linear relationships and its ability to capture interactions between features. Models were evaluated using cross-validation to mitigate overfitting and ensure that the model generalizes well to unseen data.
- Deep Learning Models (Optional): An additional neural network architecture was experimented with but ultimately discarded in favor of the Random Forest model due to its complexity and lack of significant performance improvement.

Model Optimization and Hyperparameter Tuning

- Manual Hyperparameter Tuning: Key hyperparameters for the Random Forest model, such as the number of trees (`n_estimators`), maximum tree depth, and minimum samples for splitting, were manually tuned by observing performance on the validation set.
- Automated Hyperparameter Tuning: Grid Search was exhaustively used to seach for the best hyperparameters by testing all possible combinations. Random Search was also used for a broader, more efficient hyperparameter search. Lastly, Bayesian Optimization was applied to further refine the search process, balancing exploration and exploitation.

Model Evaluation and Comparison

- Evaluation Metrics**:** Accuracy gave the overall percentage of correct predictions, Precision and Recall indicates how many of the predicted dropouts were actual dropouts and measured how many of the actual dropouts were correctly identified. F1-Score helped to balanced combining precision and recall. ROC-AUC (Receiver Operating Characteristic Area Under the Curve) was used to assess the model's ability to distinguish between classes.
- Cross-validation was employed to evaluate models on different subsets of data, providing a more reliable estimate of model performance.

Feature Importance and Model Interpretability

- Feature Importance**:** For the Tree-based Models, Feature importance plots were generated to show the relative contribution of each feature. While SHAP (SHapley Additive exPlanations) was used to interpret the predictions of the Random Forest model. SHAP values provided insight into how much each feature contributed to individual predictions, increasing transparency in decision-making.

Model Deployment

- Model Serialization: The best-performing Random Forest model was serialized using the `pickle` library to allow for deployment in the Streamlit web application.
- Deployment Using Streamlit: The final model was deployed using Streamlit, enabling educators and administrators to input student data and receive predictions in real-time. The interface is user-friendly, allowing even non-technical users to interact with the model seamlessly.

Ethical Considerations

- Bias Mitigation: The data was carefully analyzed to ensure that the model did not propagate biases, particularly in terms of socio-economic and demographic factors.
- Privacy: Sensitive student information was handled securely, and privacy policies were implemented to protect the data.
- Decision-Making: The model's predictions are intended as a decision-support tool, not as a standalone decision-making system. Human intervention is required for final decisions.

Conclusion

The pipeline for predicting student dropout rates was successfully implemented using a Random Forest model, with the entire process culminating in a deployable Streamlit application. The model performs well in predicting dropout risk, but it should be used responsibly, considering the ethical and practical limitations discussed. Further work could involve incorporating additional socio-economic data or using a more diverse dataset to enhance generalization.