Predicting Student Performance Using Synthetic Big Data and XGBoost

Overview

This project predicts student academic performance (GPA categories) using a synthetically generated dataset scaled to 50 million records. By leveraging XGBoost and big data techniques, we address challenges of privacy, data scarcity, and scalability in educational data mining.

What Has Been Done

- Generated a synthetic student dataset using Gaussian Copula Synthesizer to scale from 10M to 50M records.
- Normalized and stored the dataset in a PostgreSQL database (via Docker) for efficient querying.
- Conducted Exploratory Data Analysis (EDA) to validate feature distributions and check data integrity.
- Applied dimensionality reduction:
  - PCA for numerical features (retaining 90% variance).
  - Chi-Squared test for categorical feature selection (p-value < 0.1).
- Implemented XGBoost classifier with hyperparameter tuning using Optuna (30 trials).
- Evaluated model performance with accuracy, precision, recall, F1-score, and confusion matrices.
- Visualized feature importance, classification metrics, and class distributions.

Roadmap

- Expand visualizations to interpret model decision-making and feature interactions.
- Incorporate advanced evaluation metrics such as Cohen’s Kappa and Matthews Correlation Coefficient (MCC).
- Test additional baseline models (Logistic Regression, Decision Tree) for comparative analysis.
- Optimize training for full dataset using incremental learning and GPU acceleration where feasible.

- Automate anomaly detection pipelines for future synthetic datasets.

How It Will Be Done

1. **Data Preparation**: Synthetic dataset scaled and stored in PostgreSQL using Docker.
2. **Update Database**: Use initdb scripts to populate the database with synthetic data.
3. **Feature Engineering**: PCA and Chi-Squared tests to reduce dimensions and retain significant predictors.
4. **Model Training**: XGBoost tuned with Optuna for optimal hyperparameters.
5. **Evaluation**: Held-out test set and visualizations for performance assessment.
6. **Big Data Handling**: Query data in chunks using Ibis for memory-efficient processing in Colab.

Plan to Cover Topics

The project workflow addresses:

- **Synthetic Data Generation**: Overcomes privacy and scarcity issues for educational datasets.
- **Scalable ML Modeling**: Applies XGBoost with large datasets using database-backed workflows.
- **Result Interpretation**: Visualizations and advanced metrics for comprehensive performance analysis.
- **Future Extensions**: Improve robustness, automation, and expand to alternative ML approaches.

Deliverables

1. **Google Colab Notebook** EDA
2. **Phase 2 Report PDF**: Detailed methodology, results, and analysis.
3. **PostgreSQL Docker Setup**: Scripts and configuration for database deployment.
4. **Synthetic Data Generation Scripts**: Python scripts for generating and processing synthetic data.
5. **README.TXT**: This document.

Installation

Prerequisites

Install Docker and UV. Docker is used to spin up postgres, while UV is used to manage python dependencies.

Usage

Clone this repo

```
git clone https://github.com/IssacL891/CSCI-620-Group-Project.git
cd CSCI-620-Group-Project
```

Generate Synthetic Data

Copy the data from student-performance into the student-performance folder.

Run

```
cd "Generate Data"
uv sync
uv venv
```

To generate the virtual env.

Follow the jupyter notebook.

Setup Postgres server

Run

```
cd postgres
docker-compose up -d
```

To setup the postgres server. More details in the readme in that folder.