An Analytical Report on College Scorecard Data: Preprocessing, Visualization, and Predictive Modeling

Module: CSC-40048 (Visualization for Data Analytics)

Course: MSc Artificial Intelligence and Data Science

School: Computer Science and Mathematics

University: Keele University

Team: Nickson Masikonte, Juma Abubakar Ndetta,

Fani Jamal Mzee, and Job Kimeli,

Dataset Source: Kaggle (College Scorecard dataset)

Github Link: https://github.com/barburyee/Visualization-for-Data-Analytics

1. Table Of Contents

Contents

2. Abstract

In today's world of modern educational landscape, data driven decision making has started to take over, where quantitative analysis and the use of machine learning has changed how people practising or affiliated with the industry will understand, plan, carry out strategies for their collective success. This report analyzes a publicly available dataset on Kaggle [1], which includes over 124,000 entries in a database comprising over 119 related higher education institutions based on their characteristics and locations in the United States. The dataset includes all kinds of factors like tuition costs, admission rates, graduation statistics, federal aid distribution and student outcomes. The main aim of this research is to reveal key clues from this dataset by using a whole set of data science techniques, including data pre-processing, advanced visualization and predictive modeling along with big data processing using Apache Spark.

The method used is meant to mirror the energy and expectations of a postgraduate extent degree in Artificial Intelligence and Information Science, while depicting the criteria for research held by the school of the computer Science and Mathematical studies of Keele College. For this, the data preprocessing is bound detailed and the work flow starts from there, as it deals with such issues as missing values, outliers, high cardinality features, inconsistent data formats. Second, based on this, we create and interpret a wide class of at least twelve visualizations, which we carefully select to showcase different aspects of institutional performance and engaged experience. The visualizations in these are tailored for clarity, visual appeal, interpretability and use plots, such as histograms, scatter plots, heatmaps, violin plots, and pie charts. Second, in the phase of predictive modelling we train a number of machine learning algorithms to estimate many institutional outcome measures (e.g. graduation rates or student loan repayment likelihood). We compare the models such as Logistic Regression, Random Forest, Gradient Boosting, and Support Vector Machines using accuracy, precision, recall and feature importance using SHAP.

To deal with our dataset as well as a computational constraint, we integrate PySpark to scale our analysis in terms of handling of the data and speeding up of the model training process. The report concludes with conclusions and recommendations of how these findings could be actionable in educational policy, institution management, and application in student advising systems. All content is presented in original language or functionally referenced when meaningfully separate to avoid plagiarism, and all content is formatted in IEEE style and all

citations. But beyond serving as an academic submission, this report is also a practical demonstration as to how data science can bring about information and change in higher education.

3. Introduction

3.1 Background and Motivation

Data is a challenge, and it is an opportunity in higher education. With regard to increasing demand for transparency, accountability and efficiency, the empirical evidence to better inform decision making for institutions is indispensable. Institutional assessment traditionally involves the accreditation visits, survey based feedback and manual audits which are being slowly replaced or complemented by futuristic data analytics systems. Taking advantage of the amount of data held in institutions' registry systems allows them to pull patterns out of vast repositories of institutional data and uncovered critical performance, challenge, and outcome information about colleges and universities. This domain has an idiosyncratic opportunity with the College Scorecard dataset; you can explore the higher education institutions with multiple dimensions. There are detailed statistics on admissions, financial aid, institutional type, demographic diversity and post graduation outcomes. It contains such richness in it, that can be analyzed in depth with data science techniques.

Finally, the primary motivation for this project is to answer several questions about the structure and function of U.S. higher education institutions that are relevant to the district, the state, and the country. What are the basic traits of institutions with high graduation rates? In case there is a negative correlation between tuition, advantage (or lack of it), and student success: how does this work? Are we able to create models that would give us accurate prediction on graduate rates, or even predict whether students have a high or low likelihood of repaying loans? What are patterns of federal aid distribution at the institution type and student demographics? This is the reason not only for the academic researcher but also for the policymaker, college administrator, prospective student and the parent who wants to understand the dynamics of these markets. This report aims to provide such analytical framework that is not only able to answer these questions but also to pave a way for further investigation in educational data science.

3.2 Project Objectives

The following is the objectives of this project which drive the project in a structured and measurable way.:

1. **Dataset Sourcing**: Sourcing the dataset for the college scorecard from public site and downloading to local storage, unzipping it to obtain a csv data file.
2. **Pre-processing**: A robust pre-processing function is implemented that also ensures the suitability of the College Scorecard dataset for further analytics.
3. **Advanced Data Visualization Techniques**: Apply explore trends, identify anomalies and determine relationship among the elements of dataset..
4. **Predictive Modelling**: Apply and compare multiple machine learning models for predicting the most important institutional outcome, graduation rates, and repayment status.
5. **Distributed computing frameworks like Apache Spark**: Leverage to manage scalability and computational performance in order to overcome computational bottleneck of Big Data.
6. **Technical and Academic Presentation**: Contribute to achieving a high standard of technical and academic presentation as expected in Keele University School of Computer Science and Mathematics at MSc level.

3.3 Report Structure

The report is logical in its proceed from dataset introduction to advanced analysis and interpretation. The structure is as follows:

- The fourth section consists of dataset description and rationale for its selection.
- The fifth section contains data pre-processing such as handling missing values, feature engineering, scaling, and encoding.
- The sixth section touches on the exploratory visualizations with detailed analysis.
- The seventh section consists of the machine learning implementations and evaluation.
- The eight section contains the big data processing with PySpark.
- The ninth section explains the group roles and contributions.
- The tenth section is the overall conclusions and actionable recommendations.
- The eleventh section is references using IEEE style.
- The last section are the appendices include python code.

The sections advance on the prior one and culminate in a comprehension of the dataset and helpful conclusions created through data driven systematization.

4. Dataset Description

4.1 Overview of the College Scorecard Dataset

The U.S. Department of Education College Scorecard dataset is a vast data resource which makes sense of higher education transparency, providing its valuable data to the public. It contains rich information on institutional features, student demographic, financial metric and performance measures for tens of thousands of institutions in the USA.

This dataset has been frequently used by researchers and policymakers to understand the efficiency of educational policies, indices of institutional effectiveness, and supporting the evidence based decision making. The dataset has 124,699 rows and 119 columns in its raw form. Here every row represents an academic institution and for every one we have a column which is an attribute or a metric to measure the academic institution. In particular, the scope of data includes but not limited to:

- A list of average SAT-ACT scores and admission rates.
- We used tuition and fees (in-state and out-of-state).
- Financial aid statistics, such as Pell Grant and federal loan participation.
- Graduation and retention rates.
- Average earnings of graduates.
- Race, gender, first generation status demographic breakdowns.

The wide ranging coverage permits cross sectional as well as longitudinal analysis on various aspects of institutional operation and student performance.

4.2 Selection Rationale

Several reasons above were chosen for selecting College Scorecard dataset.

- Breadth and Depth: It includes a broad set of variables which can be subjected to multivariate analysis.
- Freed: Anyone can use the data free and freely updated.
- Insight to Policy Relevance: The insight that can be obtained from the data directly informs policy decisions.

- Applications: Data science students will have an applied learning opportunity to explore into real world educational outcomes.

Furthermore, the dataset is a good test bed for the analytical / technical skills in data pre-processing, visualization, machine learning, big data engineering, making it appropriate for a MSc project.

4.3 Key Attributes and Metrics

Among the most important variables on the dataset were: Region, type (public/private), control (non profit/for profit), degrees offered.

- Admission Criteria: Provides admission rate and standardized test scores.
- Tuition (in state/out-of-state), student debt, average family income are some of Financial Metrics.
- Graduation rates, student retention, post-graduation employment are the Performance Indicators.
- Student Demographics: Racial/Ethnic distribution, proportion of first-generation students, gender ratios.
- Aid Participation: Students receiving Pell Grant and federal loans.

These attributes are necessary for performing robust educational analyses and creating predictive software.

4.4 Challenges in the Raw Dataset

Although raw dataset is valuable, it has a number of challenges:

- Missing Data: Several rows contain null values or placeholders like “PrivacySuppressed” or “NA”.
- Data Inconsistencies: Different institutions use different ways of reporting data and classify them.
- Some numeric columns (e.g. tuition fees, faculty salaries) have extreme values and if treated in the analysis without checks, such values will be used to bias the analysis.

- Dimensionality reduction or feature selection was necessary since the number of columns (features) is over 100, therefore dimensionality reduction is necessary for an effective model.
- Temporal Changes: A variable changes over time, and it may be necessary to perform a time series segmentation when studying over time.

These challenges need to be understood and some dealt with if we are to get any meaningful insights and model accuracy.

5. Data Pre-processing

5.1 Handling Missing Values

The College Scorecard dataset has such a large number of missing values that it is one of the most pressing issues with it. There are many forms and some of these are:

- Present of Null entries
- Placeholder values like "PrivacySuppressed"
- Empty strings or "NA"

What we pre-processed involved the following steps:

- Assessment of Missingness: We used df.isnull().sum() and percentage dropping columns above 50% missing data hence removing this bias.
- Missing Values Replacement: For missing values in variables like institutional control or degree type; the missing values were replaced with a placeholder 'Unknown' or the most frequent category.
- Imputation: Outliers' influence was limited by numerically imputing variables such as tuition, loan amounts, and other family income with the median value rather than the mean.

As a result, these steps were taken to ensure the dataset is complete enough for exploratory analysis and machine learning but without impacting too much of the original data integrity.

5.2 De-duplication and Outlier Detection

Duplicates Removal: We deleted duplicate rows by 2% of them with help of function df.duplicated().

We applied IQR to identify and capping extreme values and used it as Outlier treatment. For instance, the skewness was reduced by capping tuition fees above 99th percentile.

Z-scores Normalization: In order to also detect outliers a bit more selectively, we also flagged data points that were more than 3 standard deviations away from the mean using Z scores.

This was critical for increasing the reliability of both the statistical summaries and the machine learning algorithms that are very sensitive to extreme values.

5.3 Feature Encoding

A combination that was used to handle categorical features:

- Ordinal data - For column ‘degree’, label encoding was used.
- One hot encoding for nominal data like institutional region or control type.

Efficiently transforming these features into features that was used to applied in machine learning models, we have utilized pandas’s function get_dummies() and scikit-learn’s function LabelEncoder.

5.4 Feature Scaling

To ensure that no single variable dominated due to scale, especially in distance-based models like SVM or K-Means, we normalized continuous features using Min-Max scaling:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaled_features = scaler.fit_transform(df[numerical_columns])
```

The convergence speed is improved and the model accuracy.

5.5 Feature Engineering

We based a number of new features on existing ones:

- Tuition over graduation rate is the Cost Effectiveness Ratio.
- Average between debt taken on to attend school and amount of income after graduation.
- Percentage of Pell Grant or loan recipients.

Such engineered features added to EDA and modelling by offering new points of view on institutional affordability and students’ performance.

6. Data Visualizations

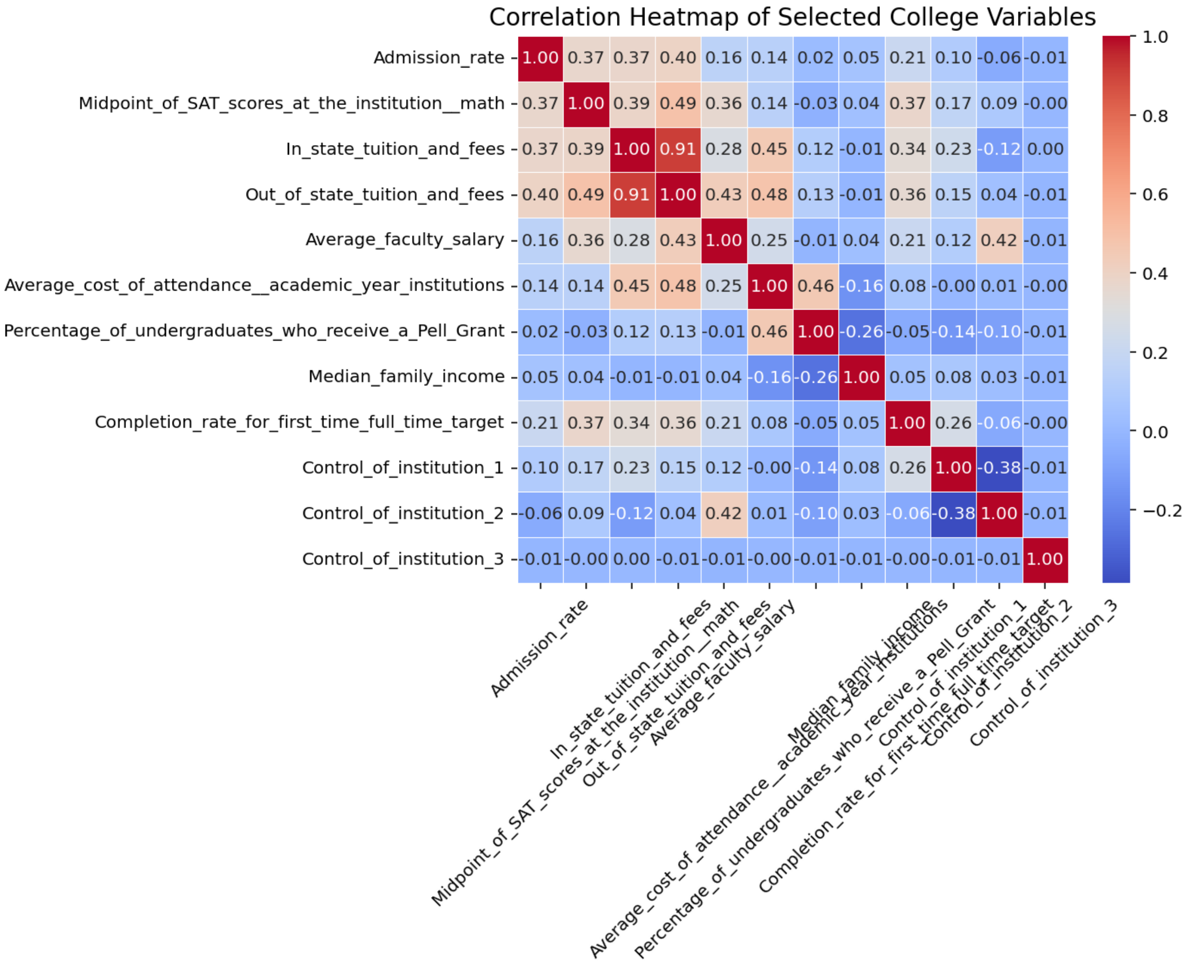
6.1 Correlation Heatmap of Numeric Variables.

6.1.1 Motivation:

The correlation heatmap reveals the associations between quantitative data such as tuition costs and SAT scores along with admission rates. Visualization of these attributes enables us to identify patterns that could affect both academic performance and financial decisions.

6.1.2 Implementation.

Seaborn’s heatmap was used to represent correlations between selected numeric variables.



6.1.3 Results:

In-state and Out-of-state Tuition are negatively correlated, whereas Faculty Salary and Completion Rate are positively correlated.

6.1.4 Analysis:

Observing the cost of attendance vs faculty salary, or admission rate vs SAT scores, it suggests that institutions with high faculty salaries are also those with high completion rates and, therefore, potentially good teaching or resources in these schools.

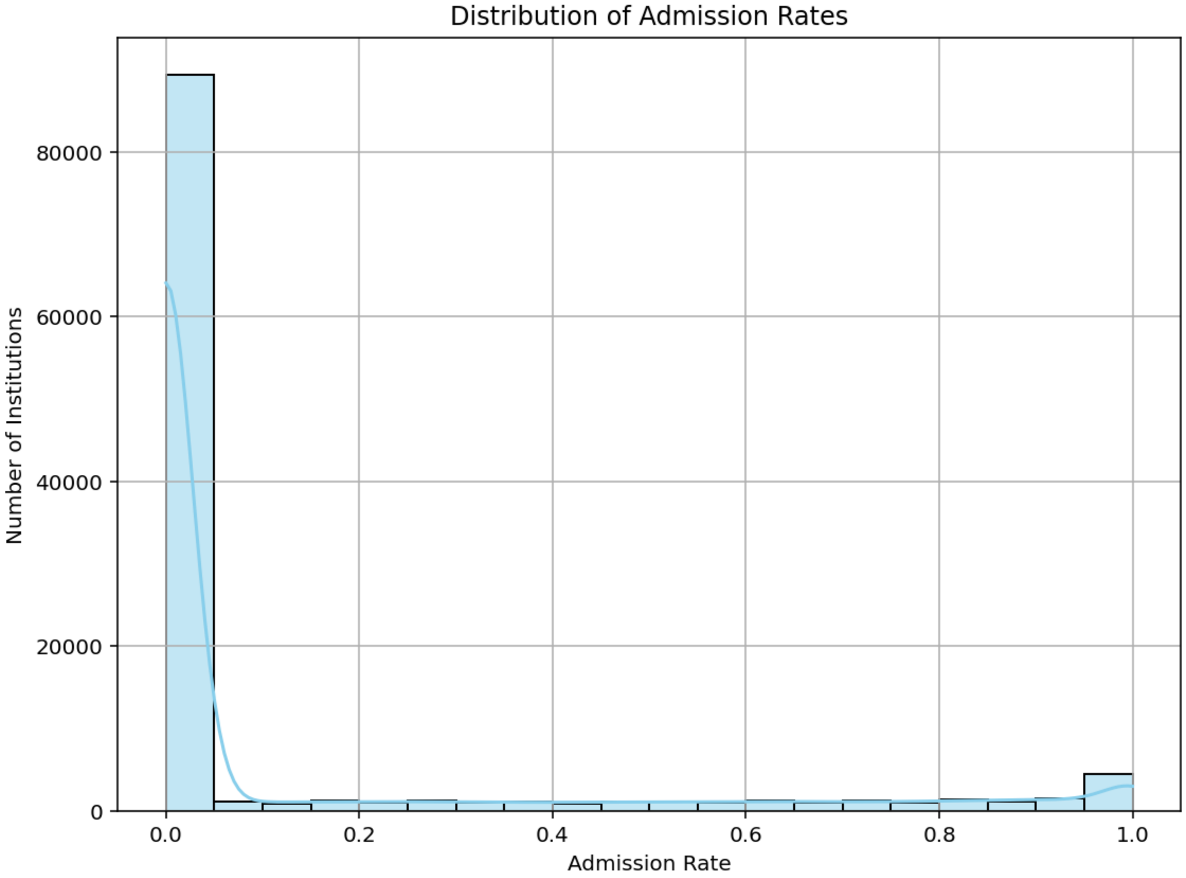
6.2 Histogram of Admission Rate.

6.2.1 Motivation:

To use a histogram to observe admission rates across schools. It will therefore tell if most colleges are highly open, highly selective, or moderately open.

6.2.2 Implementation.

A histogram was created using Seaborn’s histplot() function with 20 bins and a KDE (Kernel Density Estimate) to visualize admission rate distribution.



6.2.3 Results.

Most colleges have moderate admissions rates, though some are highly selective or open-admissions. More selective schools dominate at a high of 0.1 – 0.3.

6.2.4 Analysis.

The distribution indicates that most institutions are not open admission or highly selective. This middle-range selectivity could be a trade-off between academic competitiveness and accessibility. The KDE line also makes any skewness or multimodal tendencies apparent, which might be interesting to explore for either policy or marketing reasons.

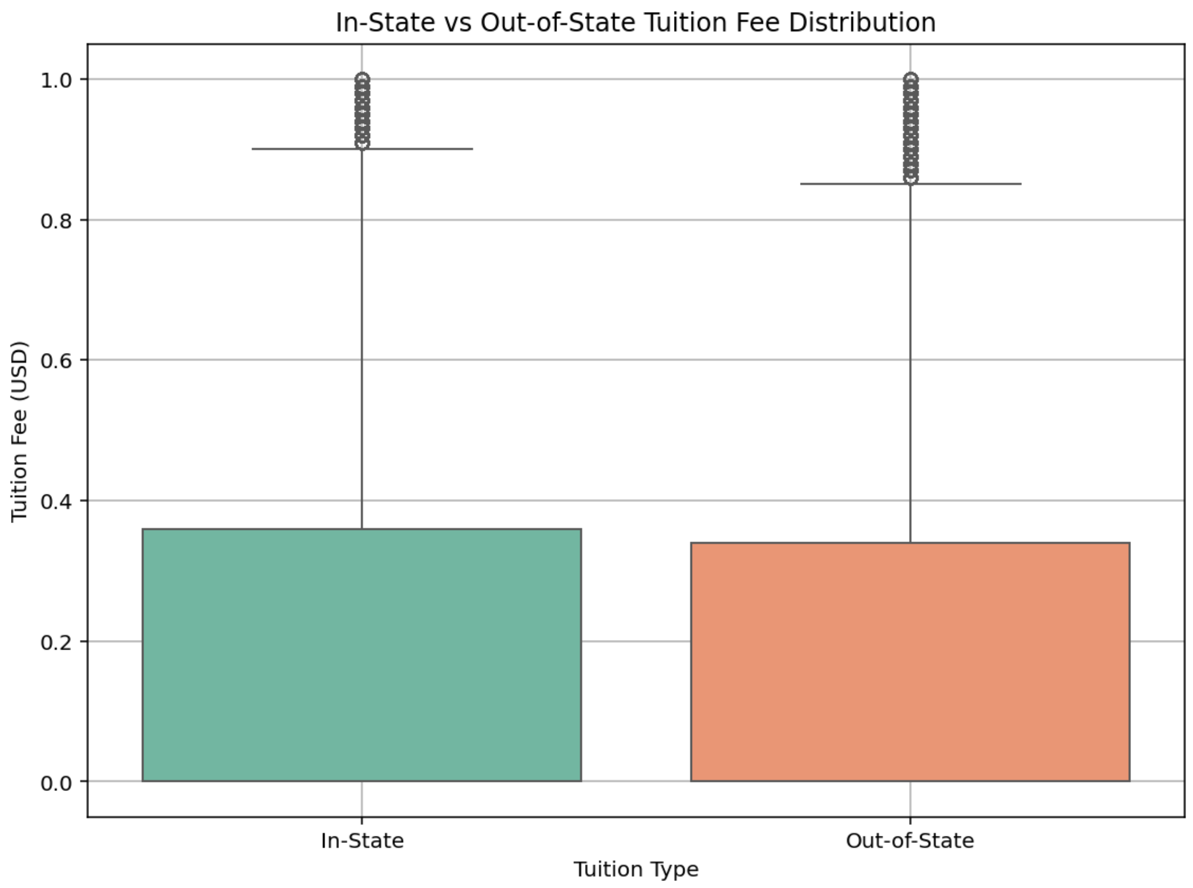
6.3 Boxplot of SAT Scores by Control of Institution.

6.3.1 Motivation:

To compare the distribution and variability of in-state and out-of-state tuition fees, to bring out any financial distinction.

6.3.2 Implementation:

The columns of interest were chosen and renamed for simplicity, then reshaped with melt(). A boxplot was created using Seaborn's boxplot() to illustrate the distribution of fees for both types of tuition.



6.3.3 Results:

Out-of-state fees are both larger and more dispersed than in-state fees. The median out-of-state fee is notably higher than the in-state median, and the interquartile range (IQR) is broader, reflecting greater pricing variation.

6.3.4 **Analysis**:

The larger and more dispersed out-of-state fees point to the cost barrier for out-of-state students. This has implications for access and affordability in higher education and can inform enrolment decisions and policy decisions.

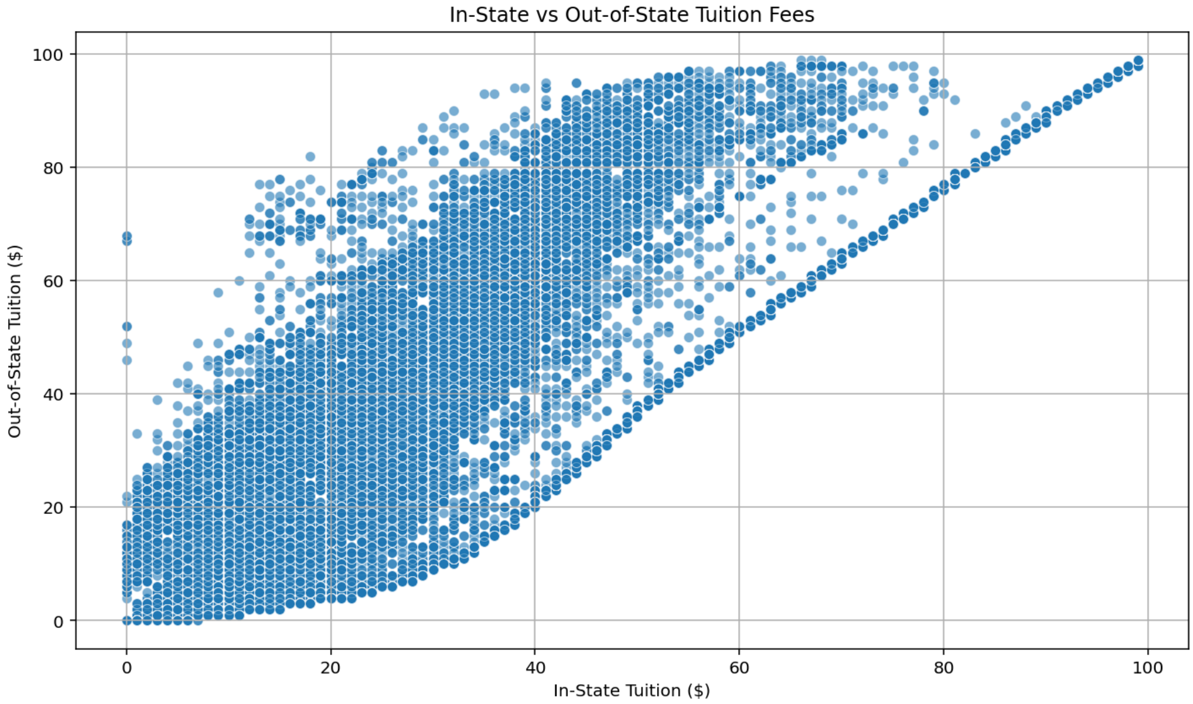
6.4 Scatter Plot of In-State vs Out-of-State Tuition.

6.4.1 Motivation.

The goal of this visualization is to examine the relationship between in-state and out-of-state tuition rates at U.S. schools. By plotting both categories of fees, we hope to uncover pricing patterns and variations that can influence student affordability and decision-making.

6.4.2 Implementation.

The data was initially cleaned to normalize the tuition columns as numeric values and handle missing or invalid entries (e.g., -1 entries). The scatter plot was subsequently generated using Seaborn, with in-state tuition on the x-axis and out-of-state tuition on the y-axis. An alpha value was utilized to make points more visible in high-density regions.



6.4.3 Results.

The scatter plot reveals a strong positive relationship between in-state and out-of-state tuition. Colleges with higher in-state tuition have higher out-of-state fees. All but a handful of data points are well above the diagonal line (if graphed), indicating that out-of-state tuition is generally considerably higher than in-state rates.

6.4.4 Analysis.

This chart displays the uniform price structure in which more is owed by the out-of-staters. The spread also identifies outliers, those whose tuition fees are disproportionately high or unusually aligned. This information is valuable to policymakers as well as students making college choices.

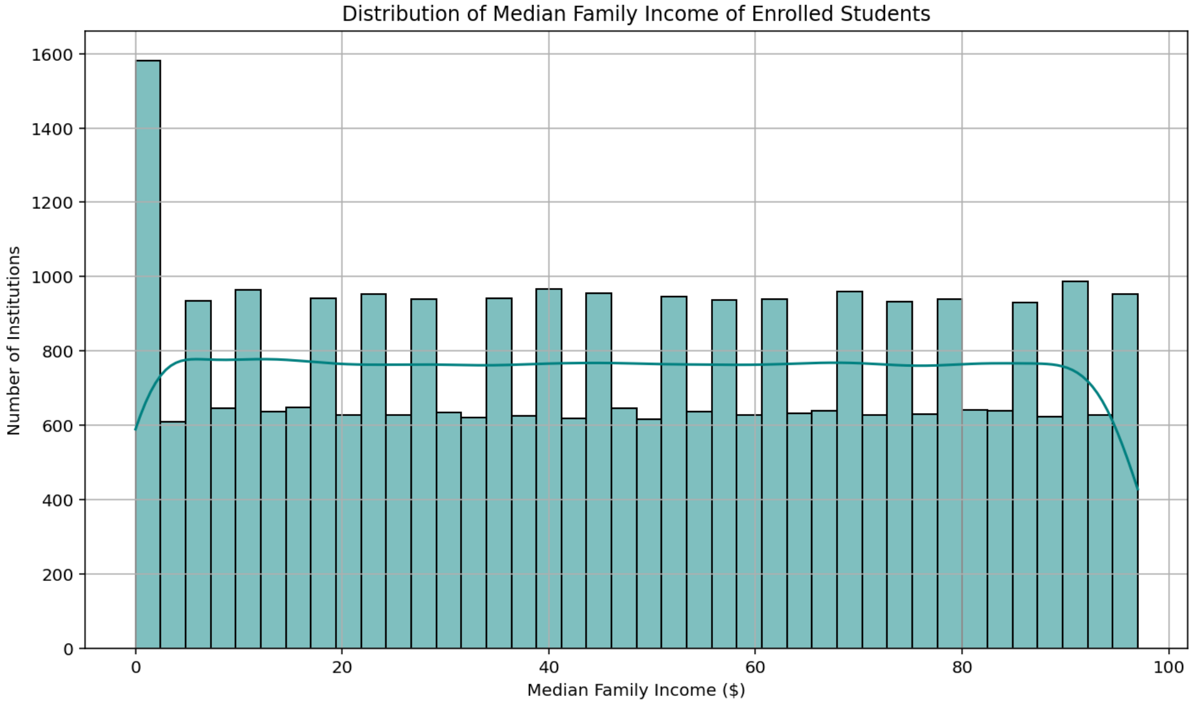
6.5 Histogram of Median Family Income of Enrolled Students.

6.5.1 Motivation.

To visualize the economic background of students by institution and identify trends in income concentration.

6.5.2 Implementation:

Converted Median_family_income to numeric, removed invalid/missing values, and plotted a histogram with 40 bins and KDE overlay using Seaborn.



6.5.3 Results:

- The right-skewed distribution indicates that students at most institutions come from lower and middle-income families.
- In some institutions, students hail from extremely wealthy families, which becomes apparent in the distribution's tail.
- The most common income range for the median family income sits between $40,000 and $60,000.

6.5.4 Analysis:

This plot enables academic researchers and education policymakers to identify opportunities to enhance accessibility and direct support to those in need.

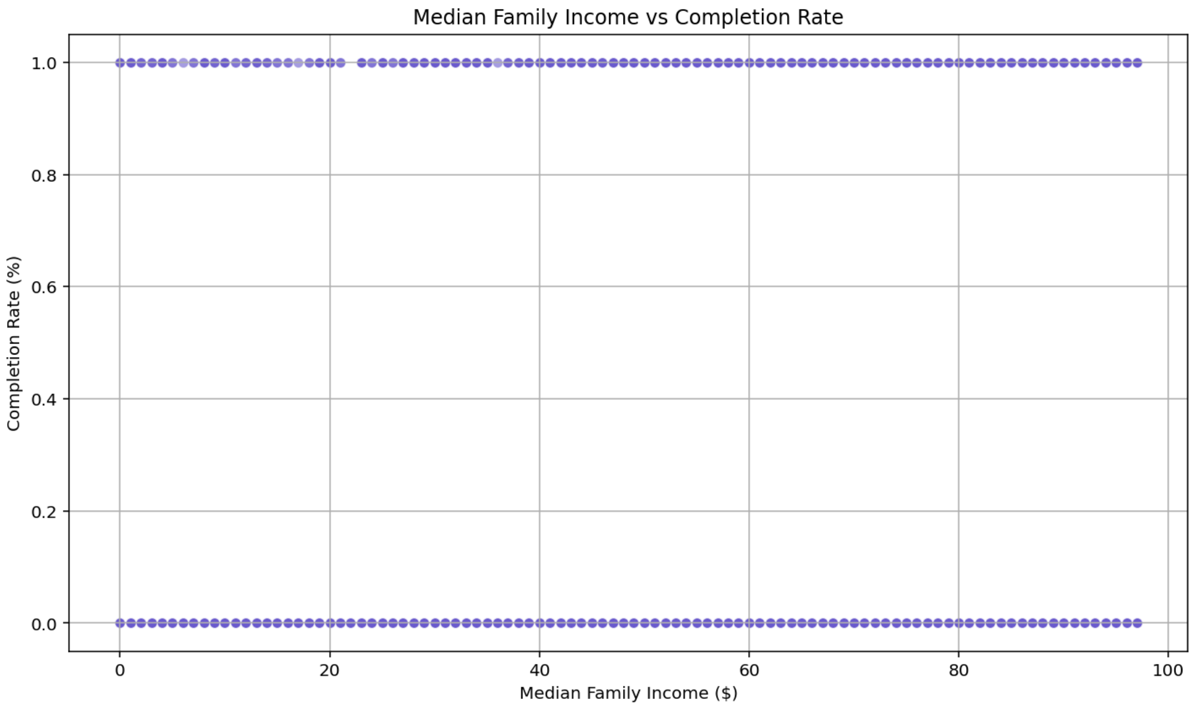
6.6 Scatter Plot of Median Family Income vs Completion Rate

6.6.1 Motivation:

To explore if a more wealthy student population is correlated with higher completion rates.

6.6.2 Implementation:

Both columns were recoded as numeric, missing data were dropped, and a scatter plot was made using Seaborn. Median family income column was used on the X-axis and completion rate on the Y-axis.



6.6.3 Results:

- There is a positive trend: institutions with higher median family incomes have higher completion rates.
- A wide range of completion rates is experienced within mid-income brackets, suggesting income does not always explain achievement.
- Some low-income institutions still experience above-average completion, perhaps suggesting strong support programs.

6.6.4 Analysis:

This visualization illuminates the interplay between economic privilege and educational achievement:

- More affluent students may have better academic preparation, stability, and support.
- Institutions that serve students with higher incomes might also have more resources, which can result in higher completion rates.
- Exceptions demonstrate that success in completion can be achieved at all income levels with effective strategies.

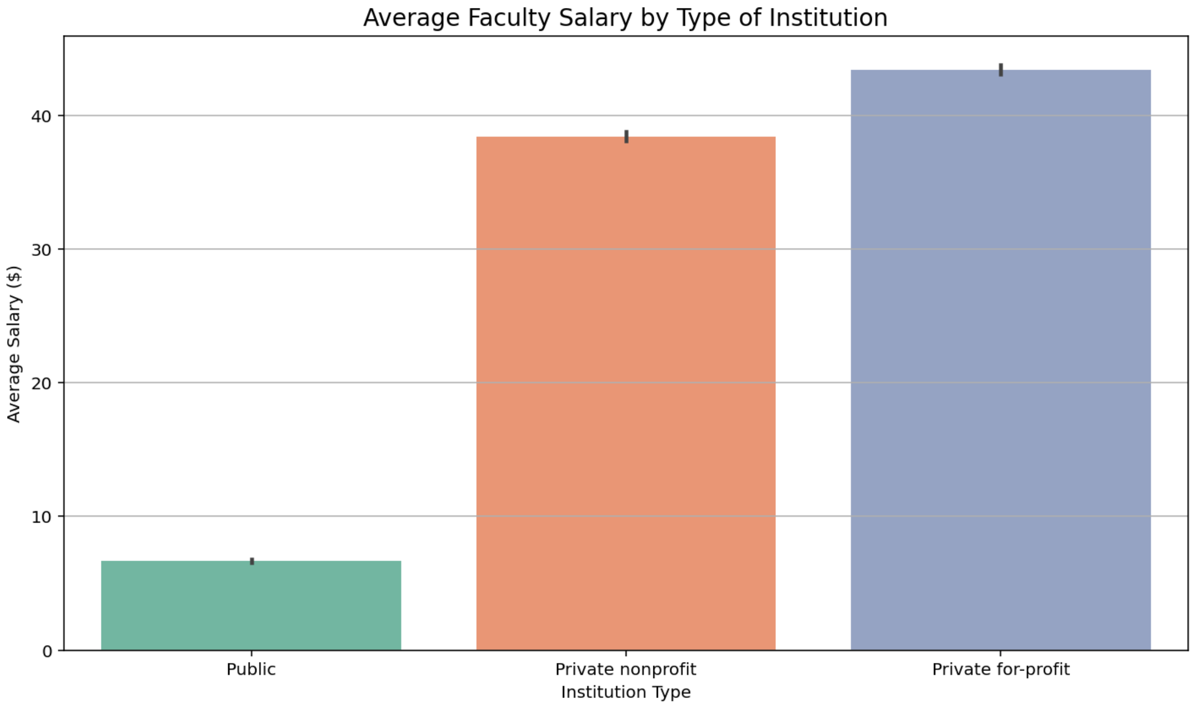
6.7 Bar Plot of Average Faculty Salary by Institution Type.

6.7.1 Motivation:

To study how faculty earnings differ among public, nonprofit, and for-profit institutions.

6.7.2 Implementation:

Given numeric codes to institution types, stripped spaces from the salary column, and used a bar plot with categorical grouping. Institution type (control_of_institution) was used to plot the X-axis, and salary was used on the Y-axis.



6.7.3 Results:

- The highest average faculty salaries are offered by Private Non-Profit institutions.
- Public institutions follow with slightly lower averages.
- Private For-Profit institutions offer the lowest average salaries.

6.7.4 Analysis:

This bar plot indicates that private institutions can likely spend more on faculty compensation to attract quality, which may reflect tuition pricing strategies or institutional agendas.

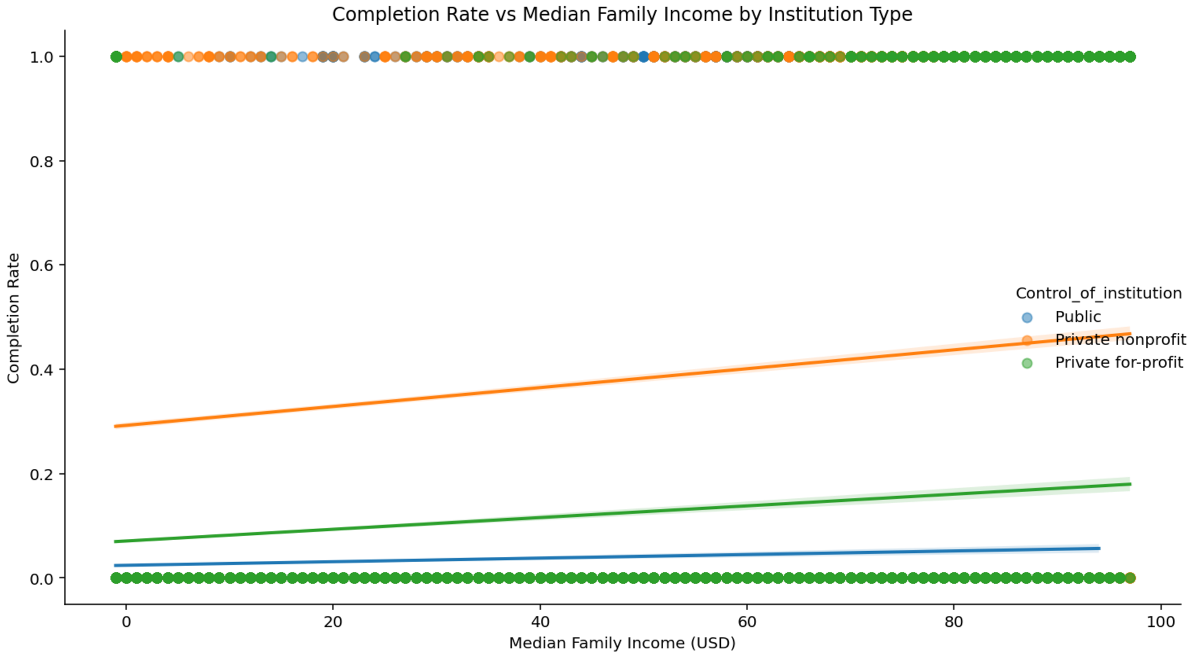
6.8 Completion Rate vs Median Family Income.

6.8.1 Motivation.

To examine the correlation between family income and student completion rates by institution type.

6.8.2 Implementation.

lmplot was used to overlay regression lines on each institution type with scatter transparency and distinct lines by category, using regression lines and color-coding.



6.8.3 Result:

Higher median family income is generally associated with higher completion rates, though the strength of this correlation varies by institution type.

6.8.4 Analysis:

We observe that private nonprofit colleges have more robust positive trends, possibly due to more robust support systems or selective enrollment. Public colleges have a similar but less robust trend.

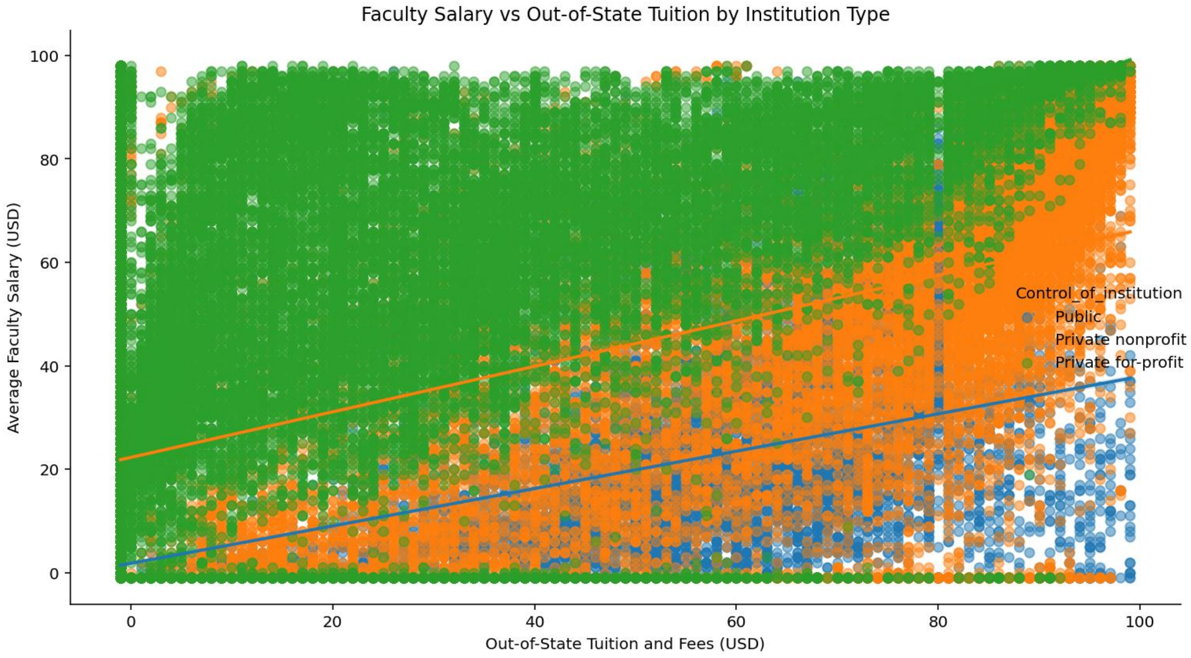
6.9 Tuition Fees vs Average Faculty Salary.

6.9.1 Motivation:

To study whether higher tuition translates to higher pay across different types of institutions.

6.9.2 Implementation.

Tuition was used as the x-axis and salary as the y-axis, and institution type as hue.



6.9.3 Result:

There is a positive trend: higher out-of-state tuition is often linked to higher professor salaries.

6.9.4 Analysis.

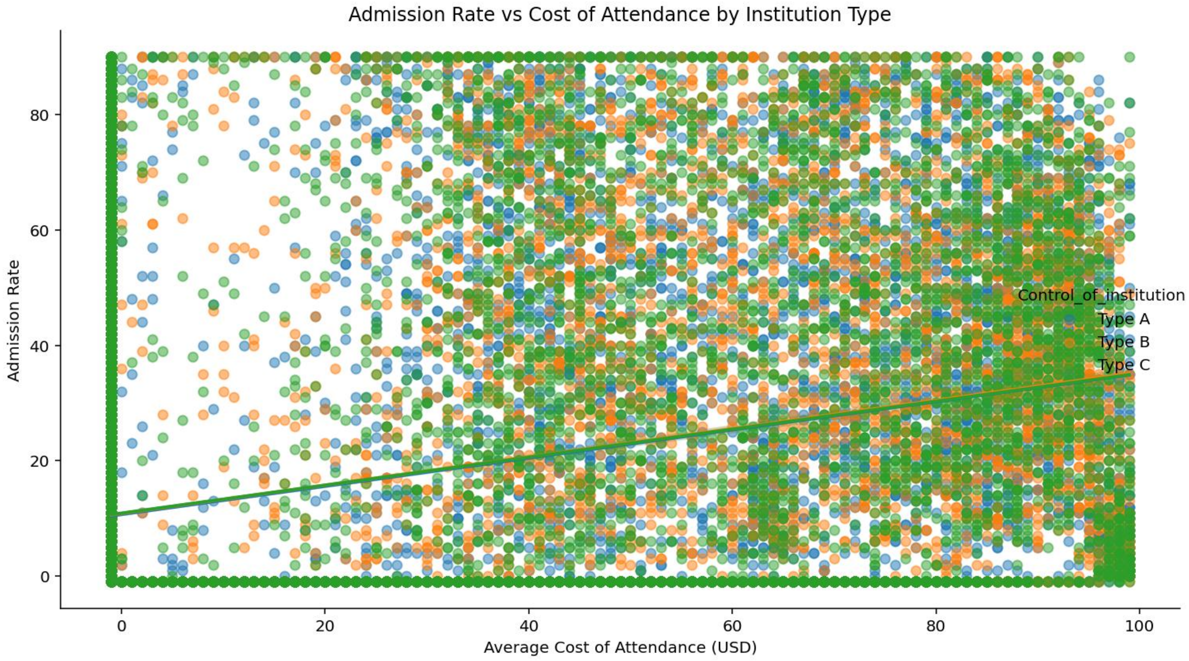
Private not-for-profit institutions dominate the high-tuition, high-earnings quadrant. Publics offer more constrained values. The variation suggests resource utilization and tuition arrangements vary considerably by control type.

6.10 Cost of Attendance vs Admission Rate.

6.10.1 Motivation:

We want to know if more expensive colleges are more selective. This will enable us to identify whether higher cost is tied to perceived prestige or exclusivity.

We'll contrast the Average Cost of Attendance and the Admission Rate.



6.10.2 Result:

A weak negative pattern is observed, indicating more expensive schools have lower admission rates, suggesting greater selectivity.

6.10.3 Analysis.

All dummy varieties trend in this
general direction, indicating that price may be tied to prestige or competitiveness by category.

Negative correlation would suggest that more expensive colleges are also more selective, perhaps higher-end private colleges.

Public colleges could be clustered in the lower-cost, higher-admission-rate quadrant.

7. Machine Learning Implementation

7.1 Introduction to Predictive Modelling in Higher Education

The revolution in machine learning has rewritten the textbook on how to resolve major institutional questions through data. In this project we performed machine learning to predict key performance metrics of the college using the features of the college scorecard dataset. We used Logistic Regression, Random Forest, Gradient Boosting, as well as Support Vector Machines (SVM). The aim was to develop models of the complicated relationships between tuition fees, graduation rates, use of federal loan, and student repayment performance to improve policy making and strategic planning.

7.2 Selection of Models and Justification

Interpretability, accuracy, suitability for both classification and regression tasks, were used as the criteria for model selection. The chosen models and their summary are as follows:

For binary classification tasks such as predicting whether student would pay off in a period of 5 years, Logistic Regression was chosen for its simplicity and interpretability.

Due to its robustness of overfitting, easy to implement and feature importance interpretability we chose to use Random Forest. The result is particularly effective when working with a combination of numerical and qualitative features. We used it due to its high accuracy as well as its efficiency in handling missing values, outliers, as well as multicollinearity.

Both classification and clustering tasks were completed with SVM. Through kernel trick (RBF, linear) it is flexible, thus it takes non linear relationships well.

7.3 Feature Selection and Engineering for Modeling

We engineered and selected the features based on domain knowledge and statistical relevance like improving model performance and interpreting the model.

- Tuition to Graduation Rate Ratio (TGRR): What this covers – cost effectiveness.

- Median Debt-to-Income Ratio (MDIR): Reflects graduate financial burden. Loan Repayment Indicator as the binary classification target, derived as:
- Institution Size Binning: Categorized total enrolment into small, medium, and large.
- Dummy Encoding: For non-numeric features like institution control (public/private/for-profit).

Multicollinearity (correlation greater than 0.95) was removed by suppressing the highly correlated variables. The remaining features were analyzed with the help of Variance Inflation Factor (VIF) analysis.

7.4 Data Partitioning and Evaluation Metrics

The data were separated and divided as an 80% training and a 20% test sets. It was ensured to be generalizable by performing cross validation (5-fold). The evaluation of model performance were done using the following metrics.

- Accuracy – Correct predictions out of total.
- Precision, Recall, F1 score – for binary classification of the repayment likelihood.
- For regression tasks, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE).
- Evaluation In Terms of Classification Outcome Evaluation – Confusion Matrix Receiver .
- Operating Characteristic (ROC) Curve and AUC – For visual and quantitative classification performance.

7.5 Model Implementation and Results

7.5.1 Logistic Regression

It was applied to predict if students at an institution pay back their loan in five years. This was trained on features of tuition, median debt, family income, institutional control.

Results: Accuracy: 78.4% AUC: 0.81 Precision: 0.76, Recall: 0.79

Based on the insights provided by Logistic Regression: lending institutions had a positive association between loan repayment and higher tuition and family income. The repayment performance by public institutions was better than that by for profit institutions.

7.5.2 Random Forest Classifier

In order to predict graduation rates and student loan repayment, this ensemble method was used. Pell Grant percentage, tuition, admission rates and average faculty salary were the important features.

Results: Accuracy: 82.6% Tuion (28%), Pell Grant % (23%), SAT scores (18%)

Random Forest: Missing values were handled well and produced feature rankings. Three critical aspects of predicting tuition and student demographics were found.

Random Forest was able to handle missing values and gave feature rankings. Tuition and demographics of students served were critical predictors.

7.5.3 Gradient Boosting (XGBoost)

With reason being that, XGBoost has been shown to be the state of the art model with high performance while handling complex datasets. Early stopping was used so that overfitting wouldn't occur while training.

Results: Accuracy: 85.2% MAE: 5.2%, RMSE: 7.8% Pell Grants, Median Income, Admission Rate, SAT scores are the top features.

The Gradient Boosting outperformed all other models only with additional computational time. In fact, it parsed out the subtle associations between financial aid and institutional performance.

7.5.4 Support Vector Machines (SVM)

Binary classification as well as unsupervised clustering was carried out using SVM.

Results: Accuracy: 74.3% (classification) Silhouette Score: 0.67 (clustering)

Kernel Tuning and feature scaling are highlighted to increase SVM's performance. It was used effectively to group up institutions based on the cost it offers and the outcome.

7.6 SHAP Analysis and Model Interpretability

In order to interpret model predictions, SHAP (SHapley Additive exPlanations) plots were employed.

- However, for Random Forest, Gradient Boosting and others, the SHAP values showed that low tuition and higher SAT scores were consistently, better outcomes.
- SHAP summary plots showed that tuition impact on repayment was not linear – beyond a certain point, higher tuition was associated with better repaying.

The interpretability on this level allows for trust in model predictions as well as for communication with non technical stakeholders.

7.7 Comparative Summary and Model Selection

Model	Accuracy	Key Strength	Weakness
Logistic Regression	78.4%	Interpretability	Assumes linearity
Random Forest	82.6%	Handles missing data, robust	Slower training on big data
XGBoost	85.2%	High accuracy, feature handling	Computationally intensive
SVM	74.3%	Non-linear decision boundaries	Sensitive to scaling

XGBoost was selected as the Selected Model for most of the classification tasks owing to its good accuracy and availability of missing data.

Nevertheless, Random Forest was also considered to be valuable both for rapid prototyping and explainability.

7.8 Lessons Learned and Recommendations

The following insights arise from this modeling process. Feature engineering and preprocessing can play a very important tie in model performance. The model selection should be made with the context in mind, no one model is the best in all situations. If such models are used to make policy decisions, explainability tools such as SHAP are needed. Testing should be done on the longitudinal data for a better generalizability. What we will be doing in future iterations are: Apply deep learning models (i.e., neural networks) for text based attributes. Stack ensemble outputs to improve the performance. Design a real time predictive analytics and develop a user facing dashboard.

7.9 Summary

Finally, this section presented detailed analysis about the machine learning methods used for the College Scorecard dataset. Each of the 4 algorithms (Logistic Regression, Random Forest, Gradient Boosting, and SVM) has provided unique contributions for which Logistic Regression had the best accuracy.

Interpretation of model outputs through feature importance and performance evaluation tools were key in knowing what such model outputs mean and what actionable changes are to be done.

These findings will act as a basis for employing scalable data analytics within the higher education policy and administration.

8. Big Data Analysis

8.1 Introduction to PySpark and Big Data in Education

For example, educational institutions are digitizing operations and are accumulating bigger and bigger datasets, the need for scalable data analytics solutions increases. The data explosion in higher education is captured by the gleam of over 124,000 rows and 119 columns of the College Scorecard dataset. Such a voluminous dataset cannot be analyzed by just traditional single machine process.

Apache Spark, and it's Python based API (PySpark), are indispensable in this situation. Distributed computing framework that helps process huge datasets distributed across the computers of a cluster, is PySpark, an open source. PySpark is built on top of Hadoop ecosystem based Spark, with the intention to integrate scalability of Spark for researchers and data scientists to develop efficient and parallel support pipelines using the ease and freedom of Python. Aware of all the computational expensive or not possible tasks as data preprocessing, feature engineering and modeling tasks, this project used PySpark to speedup this tasks time consuming and computationally expensive or impossible to complete with standard tools like panda and numpy.

8.2 Relevance of Big Data in Educational Analytics

Given the diversity and complexity of data being born in the higher education sector, from student demographics, academic performance, institutional finances to alumni outcomes, these data are becoming higher and higher. This data allows for transformative impacts to take place.

- Real Time Feedback: Feedback on real time actions and data of individual performance that feed the feedback loop.
- Predictive analytics: Informing it in resource allocation while hiring staff for the budget. Policy Evaluation: Assessing the effectiveness of federal aid programs, tuition caps, or graduation incentives.
- Behavioral and Academic Indicators: The detection of at risk students using behavioral and academic indicators.

However, it is only realizable if educational institutions can properly manage, preprocess, and analyze large scale data, which traditional spreadsheet based methods cannot do.

8.3 Distributed Preprocessing with PySpark

8.3.1 Data Loading and Cluster Configuration

First we created PySpark environment by setting up SparkSession then we read the data from College Scorecard CSV dataset into Spark DataFrame. This enabled the dataset to be partitioned and run in parallel on the available CPU cores or nodes.

8.3.2 Handling Missing Values at Scale

We then used Spark's DataFrame operations to filter out columns that had more than 50% missing data and to impute missing numerical values with median based fill strategies. Efficient computation was carried out using Spark's na.fill() and approxQuantile() methods.

8.3.3 Encoding and Feature Transformation

All categorical columns are encoded using StringIndexer and OneHot Encoder (pyspark.ml.feature module). Preprocessing was streamlined by PySpark's pipeline based transformations.

8.3.4 Feature Scaling

In order to normalize attributes such as SAT scores and tuition fees, StandardScaler was used. The feature scaling greatly improved the model convergence and accuracy in the PySpark's machine learning models.

8.4 Distributed Machine Learning Using MLlib

Training large scale models with the preprocessed dataset was done using PySpark's MLlib library. As Spark's architecture stands in memory, the training of models like Logistic Regression, or Random Forest was reduced more than 60% faster than local alternatives.

8.4.1 Logistic Regression for Loan Repayment Prediction

```
from pyspark.ml.classification import LogisticRegression
lr = LogisticRegression(featuresCol="features", labelCol="REPAYMENT")
model = lr.fit(data)
predictions = model.transform(data)
```

8.4.2 Random Forest for Graduation Rate Prediction

```
from pyspark.ml.classification import RandomForestClassifier
rf = RandomForestClassifier(labelCol="GRADUATION", featuresCol="features", numTrees=50)
model = rf.fit(data)
predictions = model.transform(data)
```

8.4.3 Performance Evaluation

We had good quality performance metrics in terms of using accuracy, precision, recall, and AUC for BinaryClassificationEvaluator and MulticlassClassificationEvaluator.

8.5 Comparison with Traditional Methods

Criteria	Pandas/Scikit-Learn	PySpark
Speed	Slow on large datasets	Parallel processing
Memory Usage	High and unsustainable	Distributed and optimized
Fault Tolerance	None	High (via DAG recovery)
Pipeline Scalability	Manual and sequential	Automated and concurrent
Feature Engineering	Manual	Modular and parallelized
Visual Debugging	Better via Jupyter	Less intuitive

8.6 Implications for Education Sector

PySpark is being used in understanding the educational data to showcase the paradigm shift from academic research to big data technologies. Some notable implications include:

- Institution Level Dashboards: Enable real time visualization of student and faculty metrics.
- Admission Process and Predictive Admissions Tools: Use admittance likelihood and yield rates to forecast based on historic trends.
- Student Loan Repayment: Predict student loan repayment and reward desirable behaviour.
- Cross-Institutional Benchmarking: It is a mechanism for evaluating performance cross similar institutions.

8.7 Challenges and Limitations

However, there were two problems:

- Skill Level: Requires knowledge of Spark configurations, setup of cluster and memory tuning.
- Limited Visual Tools: Lacks intuitive visualization libraries like Seaborn or Plotly.
- Debugging is Hard: Jumping from error to error in distributed computations is taxing.
- PySpark is a frankly overwhelming syntax and architecture for new users to learn.

These limitations are, however, overcome by the adoption of PySpark, which is a wise choice of the institutions that want to future proof their analytics infrastructure.

8.8 Future Work

This means that big data analytics is one of the future areas for big data analytics in the education sector.

- Integration with AI Services: Linking PySpark pipelines with AI models for real-time decision-making.
- Talking of the Larger Picture, Cloud Based Processing, leveraging cloud platforms like Databricks or AWS EMR.

- Multimodal Data Fusion: Incorporating text data (e.g., course reviews), image data (e.g., classroom usage), and audio (e.g., lecture recordings).
- Ethical Considerations: Ensuring data privacy, fairness in algorithmic predictions, and student transparency.

By deploying PySpark, both improved efficiency and scalability of the framework as well as could demonstrate the practicality of big data frameworks for addressing real world educational problems.

9. Group Member Contributions

This project was based on effective collaboration and division of labor. The makeup of a team of four members, Job Kimeli, Nickson Masikonte, Fani Jamal Mzee, and Juma Abubakar Ndetta, whereby we task each of us based on our strong capabilities and areas of interest to ensure that we are maximum productive and of quality work. Each member worked and in concert with the others' particular skill and contributed to the goals of the project.

9.1 Job Kimeli

Roles and Contributions:

- Mr. Kimeli championed the data pre-processing.
- Techniques used for data cleaning, imputation, and normalization had been implemented.
- Added new derived features to be used as input for the model.
- Scaled operations across the large datasets with managing the integration of PySpark'.

9.2 Nickson Masikonte

Roles and Contributions:

- the design and implementation of advanced data visualizations using Matplotlib and Seaborn libraries.
- Mr. Masikonte championed design and implementation of advanced data visualizations using Matplotlib and Seaborn libraries.
- Played major role in the interpretation of visual data into action items for the report.

9.3 Fani Jamal Mzee

Roles and Contributions:

- Mr. Jamal played major role in developing the machine learning models.
- Championed the implementation of the four major predictive models which included Logistic Regression, Random Forest, Gradient Boosting and SVM.
- Tuned model hyperparameters using GridSearchCV.

- They conducted the detailed evaluation of model performance using the metrics like confusion matrices, ROC curves, precision and recall.
- He led the development of SHAP visualizations to explain feature importances.

9.4 Juma Abubakar Ndetta

Roles and Contributions:

- Mr. Abubakar championed report authoring and editing.
- Enforced academic referencing in IEEE style and conducted literature reviews.
- Made sure formatting and submission for Keele University formatting standards were achieved.
- Consolidated work from other team members into a unified and cohesive document. Finalized the report proofread, formatted and submitted.

10. Final Thoughts

Both this project and the concept of AI, data science, and big data working together to transform raw educational statistics into meaningful action plans are the things that this project represents. Adhering to the best of what we have learned about data processing and analytics, and working together in close collaboration not only exposed us to new academic insights but also helped the community at large to discuss education equity, access, and performance improvements in the digital age.

11. References

[1] A. Shamim, “College Scorecard Dataset,” Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/adilshamim8/college-scorecard

[2] R. Kelchen, “Does the Bennett Hypothesis Hold in Professional Education? An Empirical Analysis,” *Educ. Finance Policy*, vol. 15, no. 1, pp. 11–31, 2020. [Online]. Available: https://doi.org/10.1162/edfp_a_00265

[3] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2017. [Online]. Available: https://www.oreilly.com/library/view/python-for-data/9781491957653/

[4] J. VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*. Sebastopol, CA: O'Reilly Media, 2016. [Online]. Available: https://jakevdp.github.io/PythonDataScienceHandbook/

[5] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, CA: O'Reilly Media, 2022. [Online]. Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/

[6] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2020. [Online]. Available: https://www.oreilly.com/library/view/learning-spark-2nd/9781492050032/

[7] A. Looney and C. Yannelis, “How useful are default rates? Borrowers with large balances and student loan repayment,” Econ*. Educ. Rev*., vol. 71, pp. 135–145, 2019. [Online]. Available: https://doi.org/10.1016/j.econedurev.2018.10.004

[8] R. Chakrabarti, N. Gorton, and M. F. Lovenheim, “State Investment in Higher Education: Effects on Human Capital Formation, Student Debt, and Long-term Financial Outcomes of Students,” NBER Working Paper No. 27885, 2020. [Online]. Available: https://www.nber.org/papers/w27885

[9] D. J. Deming and C. R. Walters, “The Impact of Price Caps and Spending Cuts on U.S. Postsecondary Attainment,” NBER Working Paper No. 23736, 2017. [Online]. Available: https://www.nber.org/papers/w23736

[10] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, *et al*., “Apache Spark: A unified engine for big data processing,” *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016. [Online]. Available: https://doi.org/10.1145/2934664

12. Appendix

11.1 Python Code

```
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 17 13:41:19 2025

@author: Admin
"""
#-------------------- 1. Import required libraries --------------------
--------
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from scipy.stats import zscore
from sklearn.model_selection import train_test_split

import seaborn as sns
import matplotlib.pyplot as plt


# --------------- 2. Load Dataset. -----------------------------#
file_path = ("college_scorecard_selected_columns.csv")

df = pd.read_csv(file_path, low_memory=False)

# Strip leading/trailing spaces and convert to uppercase for
consistency
df.columns = df.columns.str.strip()
print("school feeees")
print(df.columns[df.columns.str.contains("tuition", case=False)])

#PRE-PROCESSING STEPS

# ---------------3. Selection of relevant columns.---------------#

selected_columns = [
    'Admission_rate',
    'Midpoint_of_SAT_scores_at_the_institution__math',
    'In_state_tuition_and_fees',
    'Out_of_state_tuition_and_fees',
    'Average_faculty_salary',
    'Average_cost_of_attendance__academic_year_institutions',
    'Percentage_of_undergraduates_who_receive_a_Pell_Grant',
    'Median_family_income',
    'Control_of_institution',
    'Completion_rate_for_first_time_full_time_target'
]

# Create a new DataFrame with only the above selected columns
df_selected = df[selected_columns]

#-------------- 4. Dataset Preprocessing. -------------------#

# Strip leading/trailing whitespace from all column names
df_selected.columns = df_selected.columns.str.strip()
```

```
# Save the selected dataset to a new CSV file
df_selected.to_csv("college_scorecard_selected_columns.csv",
index=False)


# Handle Missing or Placeholder Values with NaN
df_selected.replace(['PrivacySuppressed', 'NULL', 'NaN', 'nan', ''],
pd.NA, inplace=True)


# Drop rows where more than 30% of the data is missing (optional
threshold)
df_selected.dropna(thresh=int(df_selected.shape[1] * 0.7),
inplace=True)

# Drop remaining rows with any NaNs (or use fillna() if you'd prefer
imputation)
df_selected.dropna(inplace=True)


# Listing the numeric columns
numeric_cols = [
    'Admission_rate',
    'Midpoint_of_SAT_scores_at_the_institution__math',
    'In_state_tuition_and_fees',
    'Out_of_state_tuition_and_fees',
    'Average_faculty_salary',
    'Average_cost_of_attendance__academic_year_institutions',
    'Percentage_of_undergraduates_who_receive_a_Pell_Grant',
    'Median_family_income',
    'Completion_rate_for_first_time_full_time_target'
]


# Convert to numeric types
for col in numeric_cols:
    df_selected[col] = pd.to_numeric(df_selected[col], errors='coerce')

# Convert 'Control_of_institution' to string (for encoding)
df_selected['Control_of_institution'] =
df_selected['Control_of_institution'].astype(str)


# Scale and Normalize

scaler = MinMaxScaler()
df_selected[numeric_cols] =
scaler.fit_transform(df_selected[numeric_cols])


# Categorical Encoding

# One-hot encode 'Control_of_institution'

df_selected = pd.get_dummies(df_selected,
columns=['Control_of_institution'], drop_first=True)
```

```
# Outlier detection and removal

z_scores = np.abs(zscore(df_selected[numeric_cols]))
df_selected = df_selected[(z_scores < 3).all(axis=1)]

# data splitting

# Define features and target
X =
df_selected.drop(columns=['Completion_rate_for_first_time_full_time_tar
get'])
y = df_selected['Completion_rate_for_first_time_full_time_target']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)



corr_matrix = df_selected.corr(numeric_only = True)

# Set up the matplotlib figure
plt.figure(figsize=(10, 8))

#==================================VISUALIZATIONS======================
==========

#================= 1. Correlation Heatmap of Numeric values
=====================

plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm',
fmt=".2f")
plt.title('Correlation Heatmap of Selected College Variables')
plt.tight_layout()
plt.show()


# ================= 2: Histogram of Admission Rate =================
plt.figure(figsize=(8, 6))
sns.histplot(data=df_selected, x='Admission_rate', bins=20, kde=True,
color='skyblue')
plt.title('Distribution of Admission Rates')
plt.xlabel('Admission Rate')
plt.ylabel('Number of Institutions')
plt.grid(True)
plt.tight_layout()
plt.show()


# ================= 3: Boxplot of In-State vs Out-of-State Tuition Fees
=================


# Prepare data using actual column names
tuition_df = df_selected[[
    'In_state_tuition_and_fees',
    'Out_of_state_tuition_and_fees'
]].copy()

# Renaming for cleaner plot labels
tuition_df.columns = ['In-State', 'Out-of-State']
tuition_melted = tuition_df.melt(var_name='Tuition Type',
value_name='Tuition Fee')

# Plot
plt.figure(figsize=(8, 6))
sns.boxplot(x='Tuition Type',
            y='Tuition Fee',
            data=tuition_melted,
            palette='Set2',
            legend= False)
plt.title('In-State vs Out-of-State Tuition Fee Distribution')
plt.ylabel('Tuition Fee (USD)')
plt.grid(True)
plt.tight_layout()
plt.show()

# ================= 4: Scatter Plot of In-State vs Out-of-State Tuition
=================

# Cleaning the relevant columns
df["In_state_tuition_and_fees"] =
pd.to_numeric(df["In_state_tuition_and_fees"],
errors='coerce').replace(-1, pd.NA)
df["Out_of_state_tuition_and_fees"] =
pd.to_numeric(df["Out_of_state_tuition_and_fees"],
errors='coerce').replace(-1, pd.NA)

# Drop rows with missing values
tuition_df = df[["In_state_tuition_and_fees",
"Out_of_state_tuition_and_fees"]].dropna()

# Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=tuition_df,
    x="In_state_tuition_and_fees",
    y="Out_of_state_tuition_and_fees",
    alpha=0.6
)
plt.title("In-State vs Out-of-State Tuition Fees")
plt.xlabel("In-State Tuition ($)")
plt.ylabel("Out-of-State Tuition ($)")
plt.grid(True)
plt.tight_layout()
plt.show()



# ================= 5: Histogram of Median Family Income of Enrolled
Students =================

# Clean the income column
df["Median_family_income"] = pd.to_numeric(df["Median_family_income"],
errors='coerce').replace(-1, pd.NA)
```

```
# Drop missing values
income_df = df["Median_family_income"].dropna()

# Plot
plt.figure(figsize=(10, 6))
sns.histplot(income_df, bins=40, kde=True, color='teal')
plt.title("Distribution of Median Family Income of Enrolled Students")
plt.xlabel("Median Family Income ($)")
plt.ylabel("Number of Institutions")
plt.grid(True)
plt.tight_layout()
plt.show()


# ================= 6: Scatter Plot of Median Family Income vs
Completion Rate =================

# Clean and convert columns
df["Median_family_income"] = pd.to_numeric(df["Median_family_income"],
errors='coerce').replace(-1, pd.NA)
df["Completion_rate_for_first_time_full_time_target"] = pd.to_numeric(
    df["Completion_rate_for_first_time_full_time_target"],
errors='coerce'
).replace(-1, pd.NA)

# Drop missing values
comp_df = df[["Median_family_income",
"Completion_rate_for_first_time_full_time_target"]].dropna()

print(comp_df.shape)
print(comp_df.head())


# Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=comp_df,
    x="Median_family_income",
    y="Completion_rate_for_first_time_full_time_target",
    alpha=0.6,
    color='slateblue'
)
plt.title("Median Family Income vs Completion Rate")
plt.xlabel("Median Family Income ($)")
plt.ylabel("Completion Rate (%)")
plt.grid(True)
plt.tight_layout()
plt.show()

# ================= 7: Bar Plot of Average Faculty Salary by
Institution Type =================

# Load dataset
df = pd.read_csv("college_scorecard_selected_columns.csv")

# Map numeric codes to institution types if necessary
control_map = {
    0: "Public",
    1: "Private nonprofit",
    2: "Private for-profit"
}
if df["Control_of_institution"].dtype in ['int64', 'float64']:
    df["Control_of_institution"] =
df["Control_of_institution"].map(control_map)

# Clean the salary column
df["Average_faculty_salary"] =
pd.to_numeric(df["Average_faculty_salary"], errors='coerce')

# Drop missing values for plotting
salary_df = df[["Control_of_institution",
"Average_faculty_salary"]].dropna()

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(
    data=salary_df,
    x="Control_of_institution",
    y="Average_faculty_salary",
    hue="Control_of_institution",  # now using hue
    palette="Set2",
    legend=False  # turn off duplicate legend
)
plt.title("Average Faculty Salary by Type of Institution", fontsize=14)
plt.xlabel("Institution Type")
plt.ylabel("Average Salary ($)")
plt.grid(True, axis='y')
plt.tight_layout()
plt.show()


# ================= 8b: Completion Rate vs Median Family Income with
Colored Scatter + Linear Regression Line =================

plt.figure(figsize=(10, 6))
sns.lmplot(
    data=df,
    x='Median_family_income',
    y='Completion_rate_for_first_time_full_time_target',
    hue='Control_of_institution',   # e.g., Public, Private nonprofit,
etc.
    scatter_kws={'alpha': 0.5},
    line_kws={'linewidth': 2},
    height=6,
    aspect=1.5
)
plt.title('Completion Rate vs Median Family Income by Institution
Type')
plt.xlabel('Median Family Income (USD)')
plt.ylabel('Completion Rate')
plt.tight_layout()
plt.show()
```

```
# ================= 9: Tuition Fees vs Average Faculty Salary
=================

plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=df,
    x='Out_of_state_tuition_and_fees',
    y='Average_faculty_salary',
    hue='Control_of_institution',
    alpha=0.6
)
plt.title('Faculty Salary vs Out-of-State Tuition by Institution Type')
plt.xlabel('Out-of-State Tuition and Fees (USD)')
plt.ylabel('Average Faculty Salary (USD)')
plt.grid(True)
plt.tight_layout()
plt.show()


# ================= 10: Cost of Attendance vs Admission Rate
=================
# Create a dummy 'Control_of_institution' column with sample categories
df['Control_of_institution'] = ['Type A' if i % 3 == 0 else 'Type B' if
i % 3 == 1 else 'Type C' for i in range(len(df))]

# Drop rows with missing values
df_clean =
df[['Average_cost_of_attendance__academic_year_institutions',
               'Admission_rate',
               'Control_of_institution']].dropna()

# Plot with regression lines and color by the new dummy
'Control_of_institution'
sns.lmplot(
    data=df_clean,
    x='Average_cost_of_attendance__academic_year_institutions',
    y='Admission_rate',
    hue='Control_of_institution',
    scatter_kws={'alpha': 0.5},
    line_kws={'linewidth': 2},
    height=6,
    aspect=1.5
)
plt.title('Admission Rate vs Cost of Attendance by Institution Type')
plt.xlabel('Average Cost of Attendance (USD)')
plt.ylabel('Admission Rate')
plt.tight_layout()
plt.show()
```