IT5006 Project Presentation

Building a University Recommendation Engine

Chong Si Qing
Lee Ming Xuan
Premi Jeevarathinam
Venessa Tan

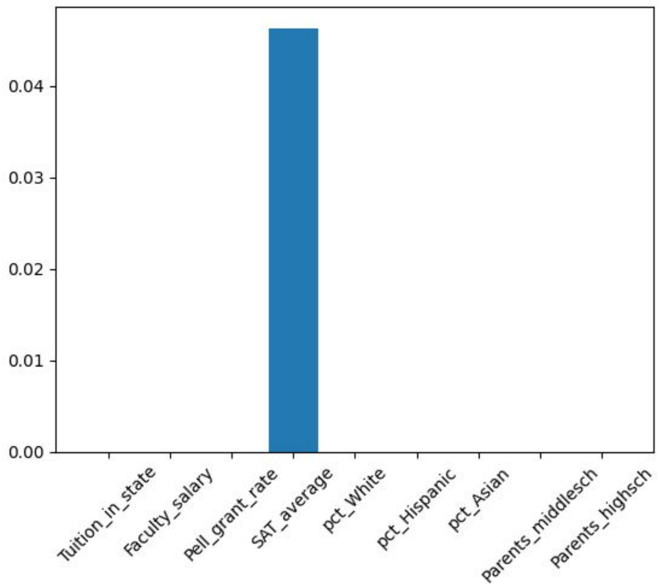
Contents

Kaggle Submission

Kaggle Submission

Initial Approach

- Basic model with OLS regression
- Analyzed feature correlation with LASSO regression
- Simple linear regression to check linearity and heteroscedasticity
- Fine-tuned model with polynomial features
- Achieved the value of R-squared: 0.768



Kaggle Submission

Improvements to model

- XGBoost model was explored
- Combines boosting, regularisation, and bagging with decision trees as base
- Effectively correlated weak features to dependent variable, reduced variance, and prevented overfitting
- Improved prediction accuracy and R-squared value
- Enabled feature importance analysis to identify impactful predictors

Model	Mean R-Score	Std R-Score
1st submission (OLS with poly features)	0.768	0.021
XGBoost	0.828	0.023
XGBoost after tuning	0.849	0.020

Conclusion

- R-score improved significantly with XGBoost model
- Heteroscedasticity observed at highest and lowest ends of completion rate
- Feature of the data, further tuning may result in overfitting

College Dataset Analysis

Introduction and Motivations

Problem Statement

- Choosing a university is a crucial decision for those considering higher education.
- Numerous factors come into play when making this decision
  - Potential earning power against the cost of studying
  - Projected Completion
  - Projected Debt Repayment
- We aimed to develop a recommendation engine using machine learning to help the students make informed decisions on their higher education options

Objectives and Outcomes

Objective: Develop a recommendation engine that generates the top universities for a student based on their unique profile.

Input

- Student’s SAT scores
- Background (Race, family income)
- Preferences (Desired field, region and locale of study

Output

- Recommendations of the top matching universities
- Key Predictive Value: Projected earnings

Dataset

Scope of Data

- Most recent data
  - No time series as earnings increase over time
- Undergraduate programs only
  - Focus on undergraduate programs within the past 10 years, to ensure relevance and accuracy.
- Institution-level data only
  - Field of study data has too many ‘Privacy Suppressed’ values

Data Preprocessing

- **Data Cleaning -**

  Dropped all NA values

- **Feature Engineering -**

  Aggregation of Locale data and State data

- **Standardisation -**

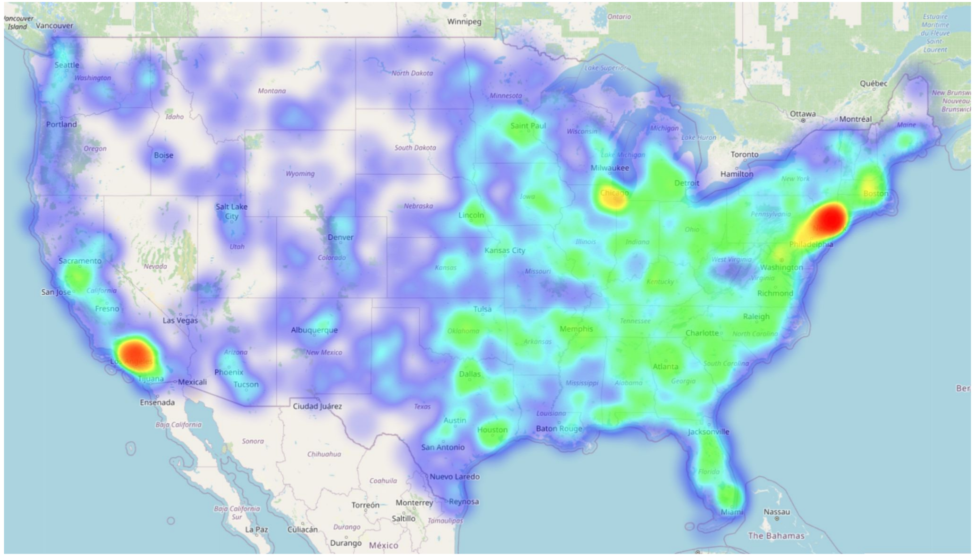
  Standardisation using Standard Scaler

Exploratory Analysis and Approach

Exploratory Analysis

Target Variable: Earnings

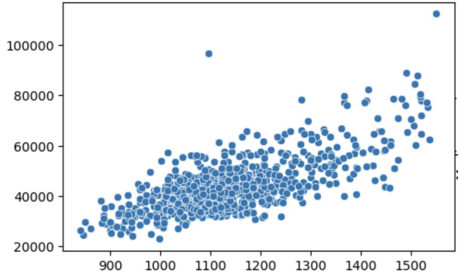
- High variability
- Right-skewed distribution
- Earnings are concentrated around the East and in NY, LA



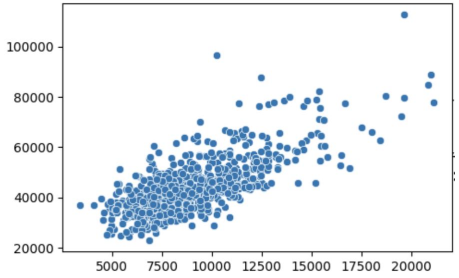
SD = 10,654 | **Skewness** = 1.6 | **Kurtosis** = 4.4

Feature Analysis

High Correlation

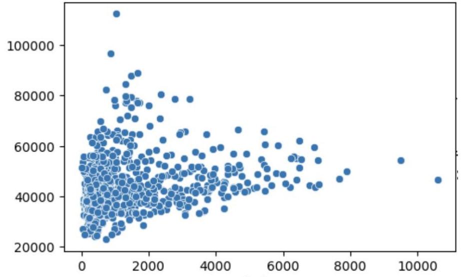


Average SAT Score

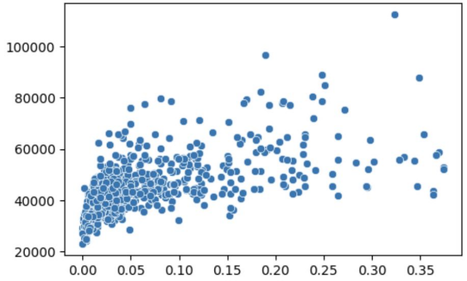


Faculty Salary

Some Correlation

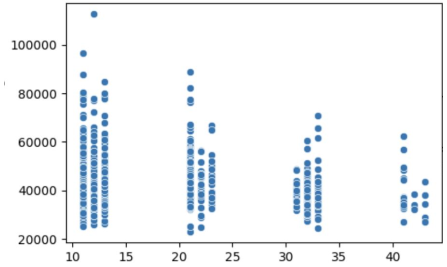


Completion Rate

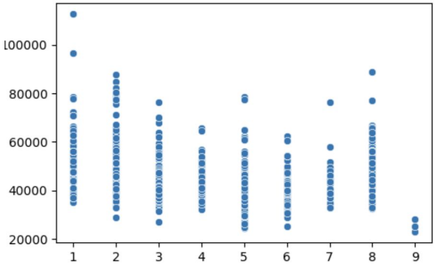


Percentage of Asians

Poor Correlation



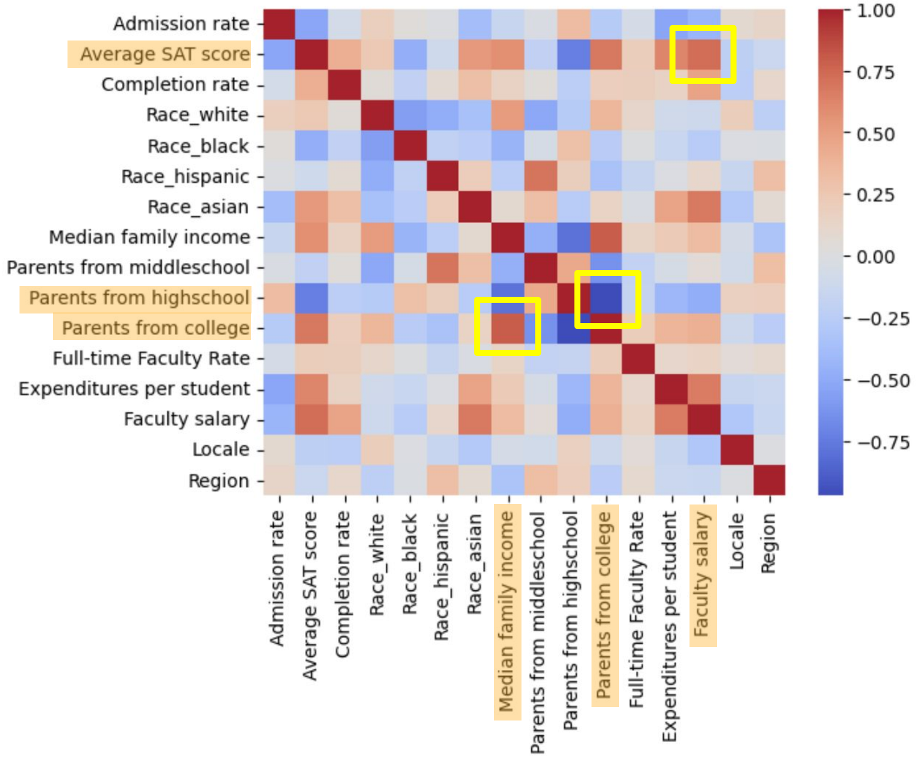
Locale



Region

Exploratory Analysis

Feature Correlation Analysis



- Highly correlated features:
  - Parents from college and Median family income
  - Parents from college and Parents from high school
  - SAT scores and Faculty salary

Exploratory Analysis

Feature Selection

Student-specific data

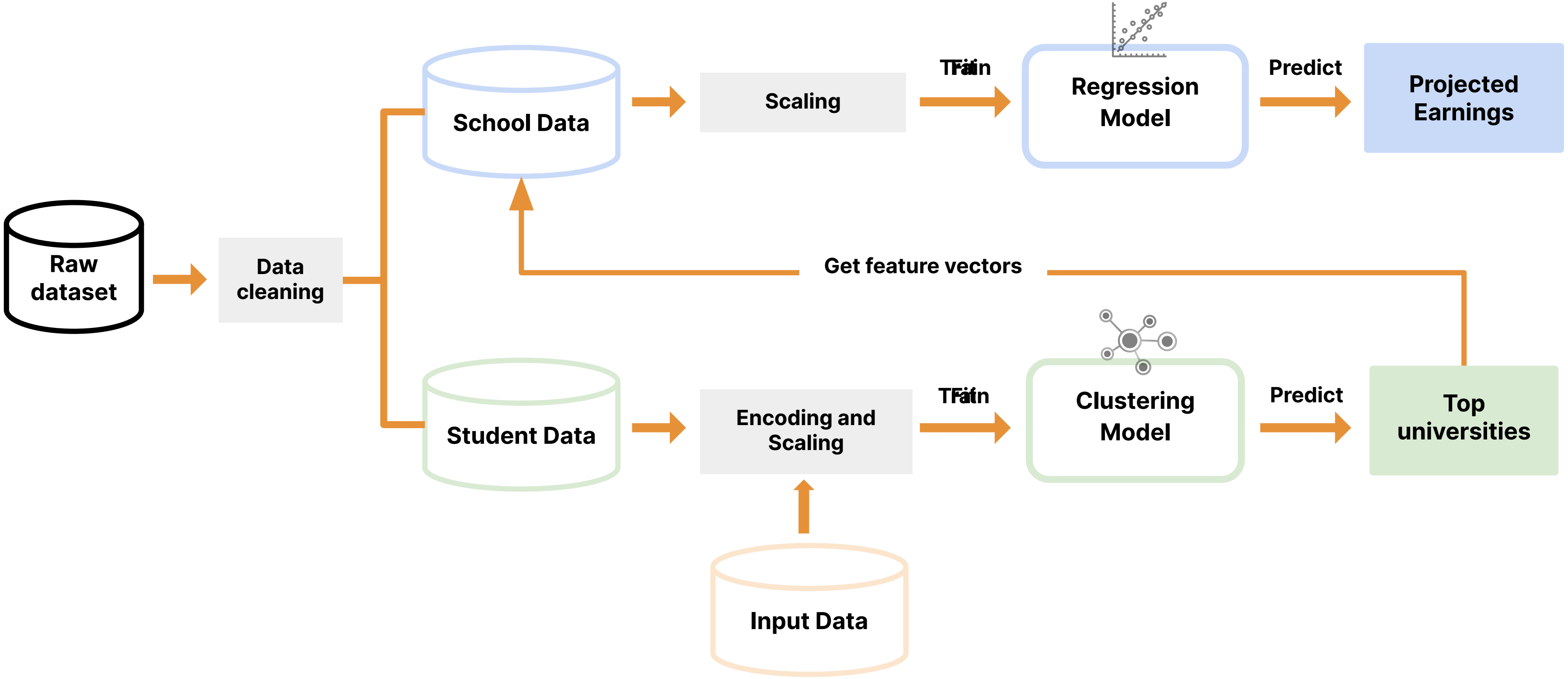
Race
SAT score
Family Income
Parents' education level
Desired Locale
Desired Region
Desired Field of Study

University-specific data

Admission rate
Faculty salary
Completion rate
Race demographics
Family income
Parents' Education
Expenditures

Outcomes of EDA

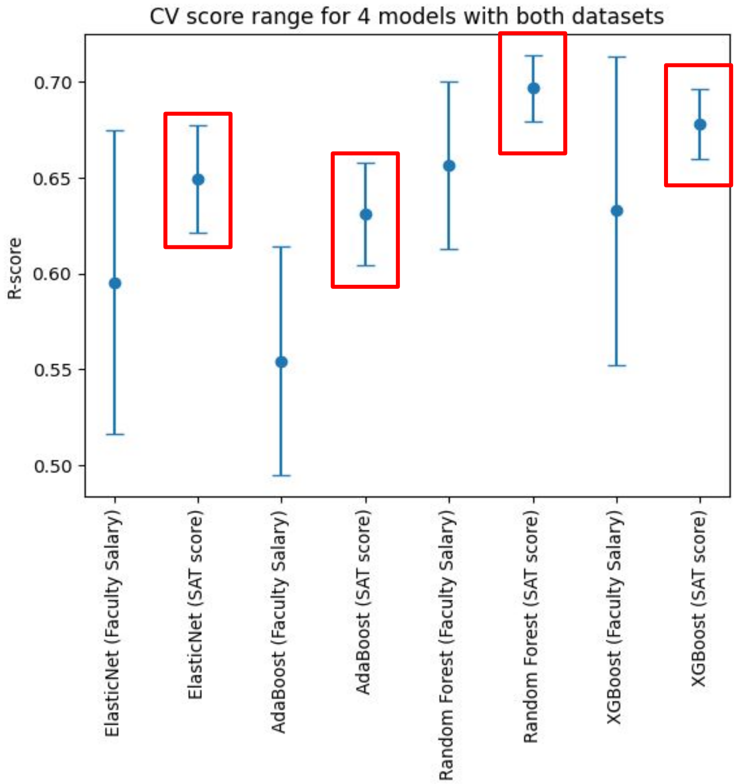
Our Approach



Building our Model and Recommendation Engine

Building our Model

Regression Models - Selection



1. Data with using SAT score had a better R-score and also less overfitting issues across models as compared to using Faculty Salary.

2. Random Forest and XGBoost showed the best initial R-score

Building our Model

Regression Models - Hyperparameter tuning

Tuning of hyperparameters is sometimes a tradeoff between highest overall score vs overfitting

Rank of Score	Min Child Weight	Mean Score	Std of Score
1	2	0.721674	0.032104
2	3	0.705202	0.026371
3	1	0.703679	0.037338
4	4	0.701058	0.027847
5	5	0.696828	0.03232

R-Score improved but spread increased

Evaluating K-Nearest Neighbours

- The distance score from the model does not tell us how good the results are. However, the recommended schools' SAT scores, median family income and demographics generally fall quite close to the student's profile.

- Due to the filtering, there are sometimes limited colleges with profile that matches the student. We can observe this when the range of the median family income of the recommended college becomes large. (next slide)

Building our Model

Results from random student profiles

Between 0 - 10%

No. of Schools left after filter	Avg difference of SAT Score between student and schools recommended	Standard deviation of SAT Score for schools recommended	Avg difference of family income between student and schools recommended	Standard deviation of family income for schools recommended	Mean earnings of recommended schools	Predicted Earnings	% diff between predicted earnings and mean earnings
25	15.14	154.95	10.96	8587.3	48098.13	46997.06	2.29
43	5.89	84.15	23.55	12693.5	45688.53	48178.42	5.45
43	7.64	128.6	15.49	13886.8	48231.53	46784.22	3.00
128	8.44	110.56	9.74	8691.49	51595.07	51907.80	0.61
58	11.54	63.94	14.81	10414.1	37909.47	39409.39	3.96
43	10.50	128.6	14.83	13886.8	48231.53	46784.22	3.00
81	10.68	121	76.76		56974.27	59659.59	4.71
27	10.13	111.18	37.46	16812.6	43286.73	45760.93	5.72
43	10.23	128.6	15.87	13886.8	48231.53	46784.22	3.00
43	9.01	128.6	45.43	13886.8	48231.53	46784.22	3.00

Range of family income of recommend schools is large

Challenges with model / dataset

1. **Challenge #1:** Limited college dataset. For example: initial size of 800 colleges can drop to below 50 if the student choose a particular profile of schools (e.g. there were only 26 colleges in the rural area) → Can lead to poorer result from K-Nearest Neighbour.

2. **Challenge #2:** Non-granular dataset. Student might be looking out for other things besides features we had put into the K-Nearest Neighbour model

3. **Challenge #3:** K-Nearest Neighbour is a simple model (e.g. hard to include features that are categorical in nature)

Evaluation and Summary

Qualitative Analysis

We have evaluated the recommendation engine based on two aspects:

1. **Cost and Benefits**: A two-step model would be computationally more expensive than Content-Based Filtering and Matrix Factorisation. However, given the significance of college decisions and dataset limitations, we have prioritised personalisation and model explainability.
2. **User Feedback**:
    a. **Useful as a discovery tool** to check out the top colleges based on predicted earnings.
    b. **Need to enable weighting of factors and multiple selection** for greater flexibility and to better fine-tune results

Conclusion

- We developed a recommendation engine that takes in a student's unique characteristics, and provides them with a personalised list of top universities that are most likely to provide the best return on their education investment
- While we wanted to predict more factors, we focused on earnings as a first step.
- Based on user feedback, we recognise that college selection is a complex decision, and the methodology can be applied to other predictive outcomes (e.g. completion likelihood, debt repayment). Further enhancements also include weighting of factors, and multiple option selection.

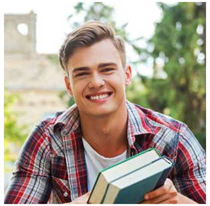
Model Selection and Tuning

1. 3 main approach + 1 a combo of all 3
    - Linear Regression with Regularisation (ElasticNet)
    - Gradient Boosting (using AdaBoost)
    - Bagging with Decision Tree (i.e. Random Forest)
    - Decision Tree with Boosting, Bagging and Regularisation (i.e. XGBoost)
    -
2. Grid search to tune the hyperparameter of the top 2 model

MX: I think this slide is not needed based on the guideline provided by the lecturer

Our Model in Action

Student Profiles



SAT Score: 1550
Race: White
Family Income: 120000
Desired Region: Southeast
Desired Locale: City
Field of Study: Computer Science

Recommended Schools:
Predicted Earnings:

SAT Score: 1600
Race: Asian
Family Income: 80000
Desired Region: Southeast
Desired Locale: City
Field of Study: Computer Science

Recommended Schools:
Predicted Earnings:

SAT Score: 1550
Race: Black
Family Income: 60000
Desired Region: Southeast
Desired Locale: City
Field of Study: Business

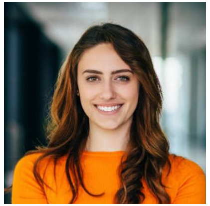
Recommended Schools:
Predicted Earnings:

Our Model in Action

Student Profiles

SAT Score: 1600
Race: Black
Family Income: 70000
Desired Region: Southeast
Desired Locale: City
Field of Study: Engineering

Recommended Schools:
Predicted Earnings:



SAT Score: 1500
Race: White
Family Income: 60000
Desired Region: Southwest
Desired Locale: City
Field of Study: Psychology

Recommended Schools:
Predicted Earnings:

SAT Score: 1600
Race: Asian
Family Income: 50000
Desired Region: Southeast
Desired Locale: City
Field of Study: Computer Science

Recommended Schools:
Predicted Earnings:

Our Model in Action

Student Profiles

SAT Score: 1400
Race: White
Family Income: 50000
Desired Region: Midwest
Desired Locale: Suburb
Field of Study: Fitness

Recommended Schools:
Predicted Earnings:

SAT Score: 1500
Race: Asian
Family Income: 80000
Desired Region: Northeast
Desired Locale: City
Field of Study: History

Recommended Schools:
Predicted Earnings:

SAT Score: 1500
Race: Hispanic
Family Income: 70000
Desired Region: Southeast
Desired Locale: City
Field of Study: Business

Recommended Schools:
Predicted Earnings: