GTU Admissions & Scholarship Decision-Support Tool

Created by: Touseef Ahmed

□ GTU Admissions & Scholarship Decision-Support Tool

This project addresses a real-world regression case study for **Global Tech University (GTU)**, a research-focused institution seeking to improve the speed and accuracy of its admissions and merit scholarship decisions through machine learning.

□ Problem Statement

GTU receives over 16,000 international applications—double pre-pandemic levels—but faculty reviewers are limited to 10 hours/week. The Provost has tasked the Data Science Center with building a **decision-support tool** that:

- Predicts the probability of admission for applicants.
- Forecasts how many merit scholarships can be safely promised without exceeding the budget.

□ Dataset

Source: Slate CRM export of GTU's admissions data from 2019 to 2023.

Features used:

| Feature | Description |
|

--|

--|
GRE	GRE score
TOEFL	TOEFL score
University Rating	Reputation of applicant's university
SOP	Statement of Purpose quality (1–5)
LOR	Letter of Recommendation strength
CGPA	Undergraduate GPA (out of 10)

| Research | Binary indicator of research work |
| Chance of Admit | Target variable (0–1 scale) |

□ Machine Learning Models Used

1. Multiple Linear Regression
- Used to understand how academic metrics (GRE, TOEFL, CGPA) influence admission odds.
- Interpretability helps Deans understand impact of policy levers.

2. Decision Tree Regressor
- Mimics human-like decision logic using if-else rules.
- Suitable for “what-if” scenario analysis.

3. Random Forest Regressor
- Robust ensemble model that reduces variance.
- Produces more reliable top-200 ranking under noisy data.

□ Model Tuning & Enhancements

✅Random Forest Regressor Tuning

Used `RandomizedSearchCV` to tune the following hyperparameters:
- `n_estimators`: Number of trees in the forest
- `max_depth`: Maximum depth of each tree
- `min_samples_split`: Minimum number of samples to split an internal node
- `min_samples_leaf`: Minimum samples required at each leaf node
- `max_features`: Number of features to consider when looking for the best split

> □ This significantly improved model accuracy and reduced overfitting compared to default parameters.

✅Decision Tree Regressor Tuning

Used `GridSearchCV` to tune:
- `max_depth`
- `min_samples_split`
- `min_samples_leaf`

This helped optimize decision trees to:
- Prevent overfitting (deep trees)

- Avoid underfitting (shallow trees)
- Improve MSE on test data

✅Feature Selection Using Random Forest Importances

- Extracted feature importances from the tuned Random Forest.
- Selected the **top 5 most important features**:
  - (e.g., CGPA, GRE, LOR, University Rating, Research)
- Retrained Random Forest and Decision Tree models using only these five features.
- Compared performance to full-feature models.

> Result: Minimal performance drop with fewer features → simpler, faster models.

✅Handling Multicollinearity in Linear Regression

Multicollinearity can make regression coefficients unreliable.

- Applied **Variance Inflation Factor (VIF)** analysis.
- Identified and dropped features with **VIF > 10**.
- Retrained the Linear Regression model.

> This resulted in better coefficient stability and more interpretable results for the admissions committee.

Evaluation Metrics

Model	Features Used	MSE (Test Set)
Linear Regression	All	_(Insert value)_
Linear Regression	VIF Filtered	_(Insert value)_
Decision Tree	Tuned, All	_(Insert value)_

Decision Tree	Top 5 Features	_(Insert value)_
Random Forest	Tuned, All	_(Insert value)_
Random Forest	Top 5 Features	_(Insert value)_

▯ Sample Inference

Test Applicant:
- GRE: 322
- TOEFL: 111
- CGPA: 8.9
- University Rating: 3
- Strong Research: ✅

▯ **Predicted Admit Probability: 74%**

✅Decision Tree model supports this via a clearly explainable decision path.

▯ Repository Structure

```
├── data/
│   └── Admission_Predict.csv
├── notebooks/
│   └── Final_DS11_Regression_case_study.ipynb
├── README.md
```

▯ Tech Stack

- Python
- Pandas, NumPy
- scikit-learn
- Matplotlib, Seaborn
- Jupyter Notebook

▯ Key Takeaways

- Decision support tools can help universities scale admissions decisions efficiently.
- Hyperparameter tuning is critical for tree-based models.

- VIF and feature selection techniques improve model interpretability and speed.
- Final model serves as a transparent and data-backed assistant to human reviewers.

✅Next Steps

- Deploy the model behind a web interface for admissions teams.
- Integrate live data feeds from Slate CRM or application portals.
- Extend model to predict scholarship thresholds based on budget simulations.

Contact us for improvement at: Touseefahmed00710@gmail.com