

Artificial Intelligence and Machine Learning Project Documentation.

Introduction and Problem Statement.

The global landscape of higher education is increasingly influenced by student migration, driven by factors such as scholarship opportunities, visa policies, and academic performance. This project aims to predict the placement status (employed or unemployed) of international students' post-graduation using machine learning techniques. The problem stems from the need to understand how educational and socio-economic variables impact employability, providing insights for students, universities, and policymakers. With the rise in international student mobility, accurately forecasting placement outcomes can help tailor support systems and address disparities in job market access. Our study focuses on leveraging a comprehensive dataset to build predictive models, addressing a critical gap in actionable data-driven insights for global education trends.

Dataset Description

Source and Structure

<https://www.kaggle.com/datasets/atharvasoundankar/global-student-migration-and-higher-education-trends>

The *Global Student Migration & Higher Education Trends* dataset is published on Kaggle, created by Atharva Soundankar.

It contains data on international student migration flows and higher-education metrics across countries and years. Typical features include

- **Origin or Home_country:** the country from which students originate
- **Destination or Host_country:** the country where students go for higher education
- **Number_of_students** (or similar): count of students migrating from origin to destination in that year
- Possibly additional features/columns related to higher-education trends (e.g., tertiary enrolment, outbound/inbound ratios, etc.)

Key Characteristic

- **Number of records:** - 5,000 observations (rows)
- **Number of features (columns):** - 20
- **Feature types** - Categorical (e.g., origin country, host country), numerical (student counts, enrolment numbers).
- **Temporal span** - Data covers multiple years around 2019-2023.
- **Geographic span:** Global (various countries of origin and destination, enabling cross-country comparisons and migration flow analysis).
- **Data quality:** Does not contain missing values, Does not contain duplicate rows.
- **Distribution & balance:** The target variable is balanced, with approximately 50% of students marked as “Placed” and 50% as “Not Placed”.

Preprocessing & Exploratory Data Analysis

six-step preprocessing pipeline transforms raw data into a clean, machine-readable format:

1. **Scaling:** Standardizes numerical features to ensure equal contribution across variables.
2. **Label Encoding:** Converts categorical target labels into numeric form for model compatibility.
3. **One-Hot Encoding:** Transforms categorical features into binary columns to represent each category distinctly.
4. **Outlier Removal:** Detects and removes extreme values to improve model stability and accuracy.
5. **Feature Engineering:** Creates new informative features to enhance predictive performance.

Before creating a model using this data set, data set should be preprocessed.

After checking The *Global Student Migration & Higher Education Trends* dataset, there are not any duplicates in records, but three columns have null values.

When using **supervised learning** for tabular data set, there are three options for using regret null values.

1. Drop columns
2. Add mean or mode values
3. Replace null value as 'null' or 0

Three features have null values 2491 null values of *placement country* ,2491 null values of *placement company* ,982 null values of *language proficiency test*.

According to this data set, first option is not matched to largest null values features because of the balancing of data set. When it would be explained, this data set's target column is placement status. There are two values and they are 'placed' and 'not placed'. There is 50% for each. When dropping columns, it would be unbalanced. And also, after dropping null values, it doesn't have enough data for train and test in model. So, it uses 982 null values data.

Second option is better for numerical data. These features are categorical therefore second option is rejected.

The last option is the best one for using avoid categorical features null values. So, placement country and placement company null values are replaced as 'not placed'.

Standardized Country

Origin country, Destination Country, Placement Country have different formatted country names. When the model training it cannot identify the equality of the country name which formats in different terms.

Solution- import '**pycountry**' library and standardize to one format like that:

```
"usa": "United States",
"u.s.a": "United States",
"us": "United States",
"uk": "United Kingdom",
"Uae": "United Arab Emirates",
"england": "United Kingdom"
```

Encode

In Supervised Learning there are three types of Encoding types.

- Label Encoding
- One Hot Encoding
- Target Encoding

For this data set, label encoding uses at least three category data features. Among the features of data set, '*placement status*' and '*scholarship received*' is used to encode. Other categorical data features are used One Hot Encoding.

Updated data set

record -4018
features-208

Scaling

There are two techniques of Scaling in Supervised Learning.

1. Min -Max Scaler

Study duration which is created by feature engineering and starting salary USD values are in various range. So, using min max scaler get these features values in same range.

2. Standard Scaler

Using standard scaler, normalize starting salary USD as low, medium and high.

Scaling like:

```
bins = [0, 0.333, 0.667, 1.0]
labels = ['Low', 'Medium', 'High']
```

Feature Engineering

In Feature engineering part, the developers create a new feature using two raw data features:

$$\text{Study Duration} = \text{year of enrollment} - \text{graduation year}$$

Under the feature creation part, new features are created and drop year of enrollment and graduation year to preprocess.

The part of feature engineering and feature selection handle main role. Under the selection part developers use variance threshold method and correlation method to get analyzed what are the unnecessary features (noises) included in data set and drop to create a best model without overfitting.

Model Design and Implementation

We use three machine learning models that were implemented and evaluated to predict the target outcome using the preprocessed dataset. The models include **Logistic Regression**, **Support Vector Classifier (SVC)** and **Random Forest**. Each model was tuned using **GridSearchCV** to identify the optimal hyperparameters.

Model Selection

1. Logistic regression

- Data set focuses on binary classification problem (placed /not placed)
- Coefficients clearly show feature importance

2. Support Vector Classifier

- Data set has many features
- Works well with small to medium tabular datasets

3. Random Forest

- features interact with each other
- Automatically captures feature interactions

Parameters Used for Tuning

1. Logistic regression

- 'C' : [0.01, 0.1, 1, 10],
- 'solver' : ['lbfgs', 'liblinear']

2. Support Vector Classifier

- 'C' : [0.1, 1, 10]
- 'Kernel' : ['Linear', 'rbf', 'poly']
- 'gamma' : ['scale', 'auto']

3. Random Forest

- 'n_estimators' : [100, 300, 500]
- 'Max_depth' : [None, 10, 20]
- 'Min_sample_split' : [2, 5, 10]

Evaluation

Parameters of classification Report

- Precision
- Recall
- F1-score
- Support
- Accuracy

Estimated Accuracy

- Logistic Regression -0.995
- Random Forest -0.995
- SVC- 0.998

Best model

Random Forest

Same accuracy range is in every models . so we cannot get best model under the Accuracy.

Random forest is best because it offers comparable performance with better robustness, scalability, minimal preprocessing, and built-in feature importance, making it more suitable for tabular data.

Root case for Random Forest model – Accuracy and parameters of the model training are showing **1.0**.

So, developers find the reason which is affected to overlap. Finally, data leakage affected the problem. Random Forest is best for reducing confusion in **overlapping regions** and better generalization and fewer misclassifications.

Deployment

Pros & Cons

Pros

Balanced Data set

Cons

Data leakage

Not enough data records

Technics

Library -numpy , pandas, Scikit-learn , Matplotlib , Seaborn, pycountry

Language -Python

Thanks

