Full Stack Intelligent Solution for the problems

Group Name: Aditi

Group Members:

First name	Last Name	Student number
Aditi	Pithiva	C0933875
Daivik	Pelathur	C0938273
Dhrumil	Tarde	C0926130
Maitriben	Balar	C0938274
Raj	Tamakuwala	C0934427

Submission date: ***[13/12/2024]***

Template Prepared by: William Pourmajidi

Updated by Vahid Hadavi Nov 2023

Contents

Abstract

In this report, we discuss the analysis and visualization of the student dataset. We also present a full-stack intelligent solution for predicting First Year Persistence using Neural Networks. We began my performing Exploratory Data Analysis, which gave us very definitive insights. We observed that most of the features were categorical yet labeled encoded. At first glance, it seemed our dataset had no missing values; after replacing '?' with np.nan, we observed the count of missing values in each feature. After handling the missing values and outliers and fixing the datatypes, we created several visualizations using Plotly Express, Matplotlib, and Seaborn. We finally created a dashboard using Plotly Express and Dash; we then built a GraphQL API powered by Neural Networks to build a prediction model. Furthermore, we build a FAST Frontend to allow users to interact with our model and analyze the visualizations we built.

Introduction

This project analyzes the student data to predict student retention. First, year persistence is the likelihood that a student will continue their education. We began by fixing the dataset by removing the first 24 rows, renaming the column headers, handling missing data, and removing outliers.

Several visualizations were created to understand key insights; we found a higher count of Domestic Students whose first language is English than International Students. We also observed the age groups with the most students; we analyzed the gender distribution among domestic and international students, the English proficiency level of domestic and international students, GPA Comparison, and many more using Plotly Express, matplotlib, and Seaborn.

A Full Stack Intelligent solution was created to predict the First Year Persistence using Neural Networks; we created a FAST Frontend integrated with GraphQL to visualize the Dashboard created using Plotly Express and Dash. This solution allows users to interact with the Frontend, gain insights, and make predictions using our model.

Methods

Preparing Dataset for EDA

In this project stage, we begin by importing the libraries and removing the first 24 rows of our data because all records above the 24th row are the metadata and are not required for our analysis.

To create interactive dashboards, we need to run pip install dash on Jupyter

```
#!pip install dash
```

```
import pandas as pd
import numpy as np
import plotly.express as px
from dash import Dash, html, dcc, Input, Output
from plotly.subplots import make_subplots
import plotly.graph_objects as go
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

Importing Libraries

```
filePath = "Student data.csv"
headers = [
    'First Term Gpa', 'Second Term Gpa', 'First Language', 'Funding', 'School',
    'Fast Track', 'Coop', 'Residency', 'Gender', 'Prev Education',
    'Age Group', 'High School Average Mark', 'Math Score', 'English Grade', 'FirstYearPersistence'
]
```

```
df = pd.read_csv("Student data.csv", skiprows=24, header=None)  # we are skipping the first 24 rows
df.columns = headers
df.shape
```

```
(1437, 15)
```

Fixing Dataset

Our dataset contains 1437 records and 15 columns.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1437 entries, 0 to 1436
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   First Term Gpa            1437 non-null   object
 1   Second Term Gpa           1437 non-null   object
 2   First Language            1437 non-null   object
 3   Funding                   1437 non-null   int64
 4   School                    1437 non-null   int64
 5   Fast Track                1437 non-null   int64
 6   Coop                      1437 non-null   int64
 7   Residency                 1437 non-null   int64
 8   Gender                    1437 non-null   int64
 9   Prev Education            1437 non-null   object
 10  Age Group                 1437 non-null   object
 11  High School Average Mark  1437 non-null   object
 12  Math Score                1437 non-null   object
 13  English Grade             1437 non-null   object
 14  FirstYearPersistence      1437 non-null   int64
dtypes: int64(7), object(8)
memory usage: 168.5+ KB
```

Df.info()

By taking a closer look at our features, we observe that most of the features are categorical but have been encoded with numerical values.

Template Prepared by: William Pourmajidi

Updated by Vahid Hadavi Nov 2023

We also notice that there are no missing values, but in reality, there are missing values; the dataset contains several '?' signs, which indicate that the record is missing.

```
df.replace('?', np.nan, inplace=True)
df.isnull().sum()
```

```
First Term Gpa              17
Second Term Gpa            160
First Language             111
Funding                      0
School                       0
Fast Track                   0
Coop                         0
Residency                    0
Gender                       0
Prev Education               4
Age Group                    4
High School Average Mark   743
Math Score                 462
English Grade               45
FirstYearPersistence         0
dtype: int64
```

Replacing the "?" value with nan

Exploratory Data Analysis

In this project stage, we analyze each feature to find insights on the data; we discovered many datatypes are shown as numerical but categorical in nature. There are 1437 records and 15 features; the featured school has only one category – Engineering so we decided to drop it. High school averages are higher than 1oo, which is not possible. These are outliers. Moreover, the High School Average has over 51 percent missing data.

Data Preprocessing

In this project stage, we correct the datatypes, handle the missing values, and fix outliers.

Fixing Datatype and Handling Missing Data

We created functions in Python to handle the missing data, remove outliers, and fix datatypes.

Features Name	Datatype before converting	Datatype after converting	Missing Values before handling	Missing Values after handling	Imputation Technique
First Term Gpa	Object	Float	17	-----	Mean
Second Term Gpa	Object	Float	160	-----	Mean

First Language	Object	Float	111	-----	Mode
Funding	Integer	Integer	-----	-----	-----
School	Integer	Integer	-----	-----	-----
Fast Track	Integer	Integer	-----	-----	-----
Coop	Integer	Integer	-----	-----	-----
Residency	Integer	Integer	-----	-----	-----
Gender	Integer	Integer	-----	-----	-----
Prev Education	Object	Object	04	-----	Mode
Age Group	Object	Object	04	-----	Dropped Rows
High School Average Mark	Object	-----	743	-----	We Dropped
Math Score	Object	Float	462	-----	Mean
English Grade	Object	Object	45	-----	Mode
First Year Persistence	Integer	Integer	-----	-----	-----

We have converted the columns to numeric format: -

```
def convert_to_numeric(df, columns):
    """
    Now let us convert specified columns to numeric, coercing errors to NaN.
    """
    for col in columns:
        df[col] = pd.to_numeric(df[col], errors='coerce')
    return df
```

Function to convert to numeric

We have written a function for the imputation technique; since here our columns are numerical, it is ideal to impute it with the mean technique to get an average of the data.

```python
def impute_with_mean(df, columns):
    """
    Now let us impute specified columns with their mean.
    """
    for col in columns:
        df[col].fillna(df[col].mean(), inplace=True)
    return df
```

Function to impute with mean

In this case, since our columns are in categorical format, we have utilized the mode technique to impute missing values for categorical values.

```python
def impute_with_mode(df, columns):
    """
    Now let us impute specified columns with their mode.
    """
    for col in columns:
        df[col].fillna(df[col].mode()[0], inplace=True)
    return df
```

Impute with mode function

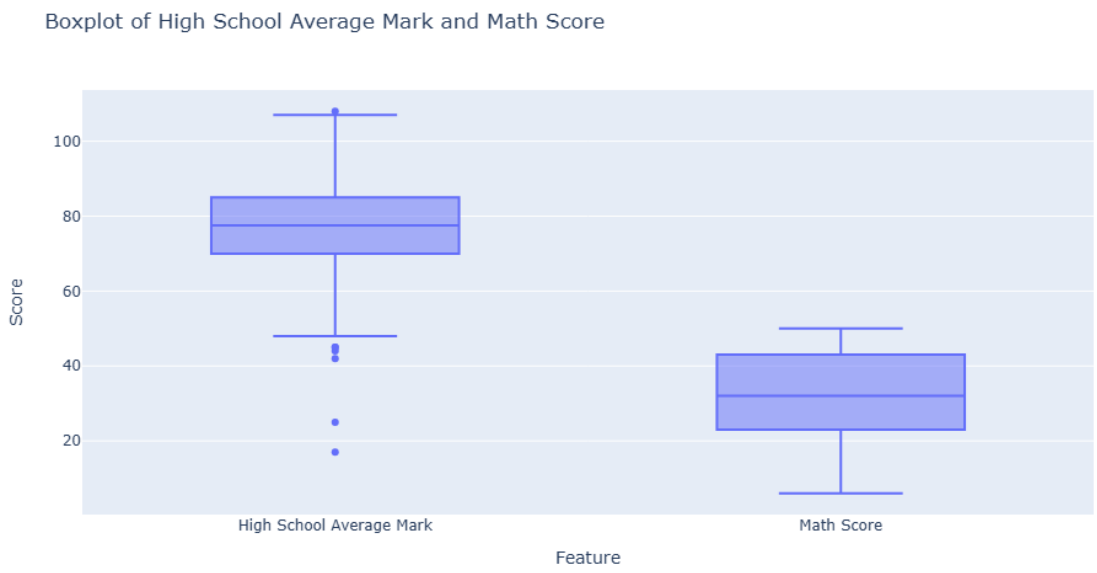
```
In [100]: def preprocess_data(df):
              """
              Now this function is the full preprocessing pipeline for the dataframe.
              """
              # Lets convert specific columns to numeric
              numeric_columns = ['First Term Gpa', 'Second Term Gpa', 'First Language', 'Math Score']
              df = convert_to_numeric(df, numeric_columns)

              # Let us impute the GPA columns with mean
              gpa_columns = ['First Term Gpa', 'Second Term Gpa', 'Math Score']
              df = impute_with_mean(df, gpa_columns)

              # Let us impute categorical features with mode
              categorical_columns = ['First Language', 'Prev Education', 'English Grade']
              df = impute_with_mode(df, categorical_columns)

              df = df.dropna(subset=['Age Group'])

              columns_to_drop = ['High School Average Mark', 'School'] # school only has engineering, and high school average has more th
              df = drop_columns(df, columns_to_drop)

              return df

In [101]: df = preprocess_data(df)
```

Preprocess data function

A preprocessing pipeline was built to convert data into numerical format and then impute the features with mean mode. We dropped 2 features – High School Average mark due to the large count of missing values (over 50% of the data was missing), and School as it had only a single label across the dataset.

Outlier Detection

In this stage, we analyze the outliers in High School Average Marks and Math Scores. We observe few records with values above 100; this does not make sense as the marks range from 0-100 in this feature. So, we remove all records with a High School Average Mark greater than 100.



Outlier Detection Using Boxplot

Over here we drop High School Average Mark and School

```
# More than 50% of the data is missing in High School Average Mark -> Lets drop this feature
df.drop(columns=['High School Average Mark'], inplace=True)
```

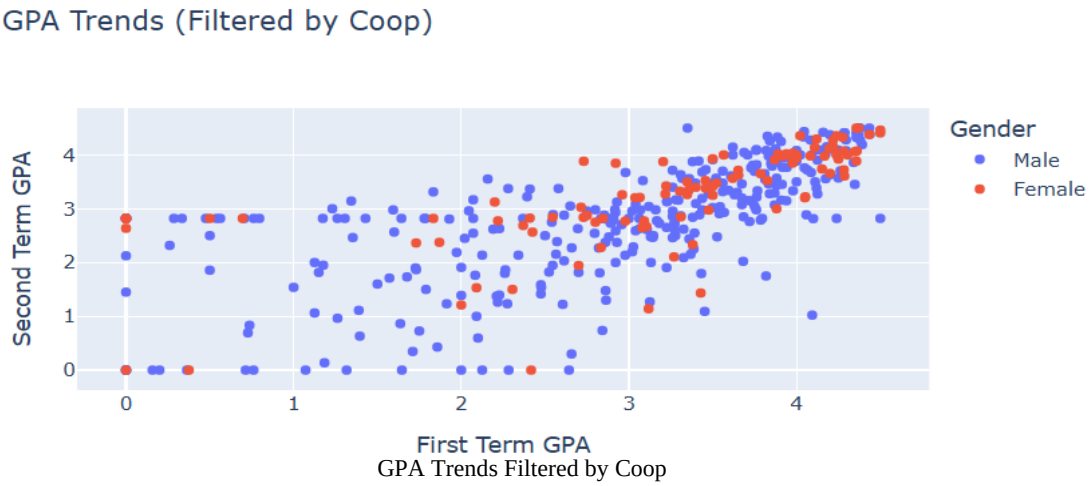
```
df.drop(columns=['School'], inplace=True)
```

Dropping the Unwanted Data

```
df.isnull().sum()
```

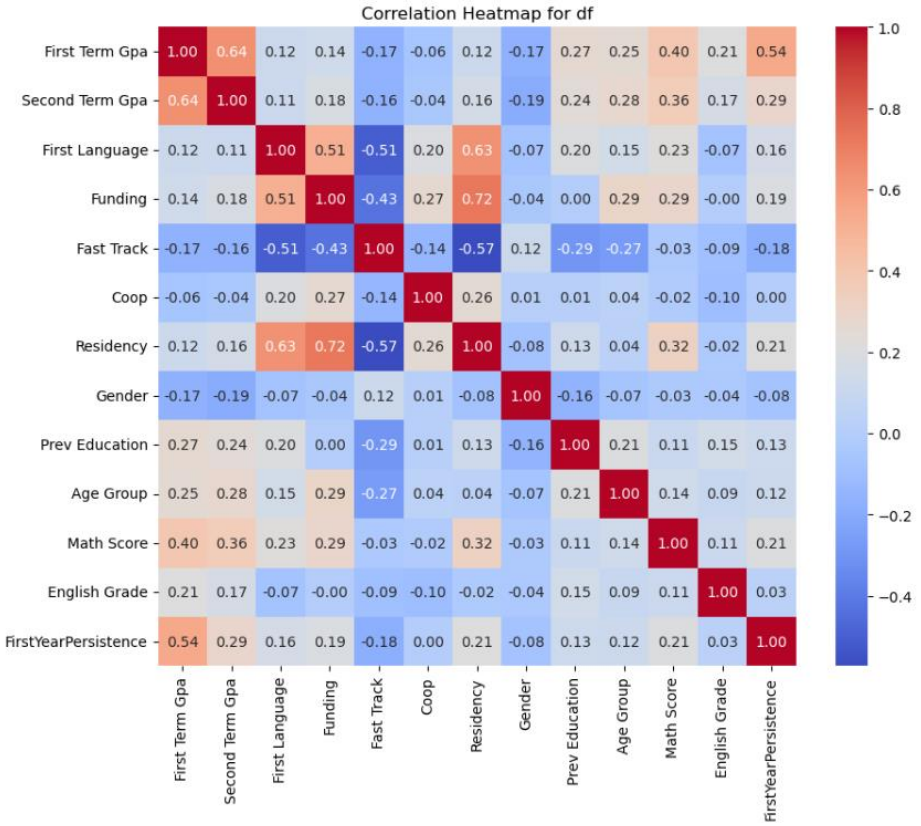
```
First Term Gpa          0
Second Term Gpa         0
First Language          0
Funding                 0
Fast Track              0
Coop                    0
Residency               0
Gender                  0
Prev Education          0
Age Group               0
Math Score              0
English Grade           0
FirstYearPersistence    0
dtype: int64
```

Data Visualization



GPA Trends Filtered by Coop

The feature Gender has a major class imbalance; there are 1111 records for Males and only 325 records for Females. From the above scatterplot we observe that Female students tend to perform better, with higher GPA Scores.



Heatmap

From the following heatmap, we can see the feature with strong and weak correlation

Strong Positive Correlation:

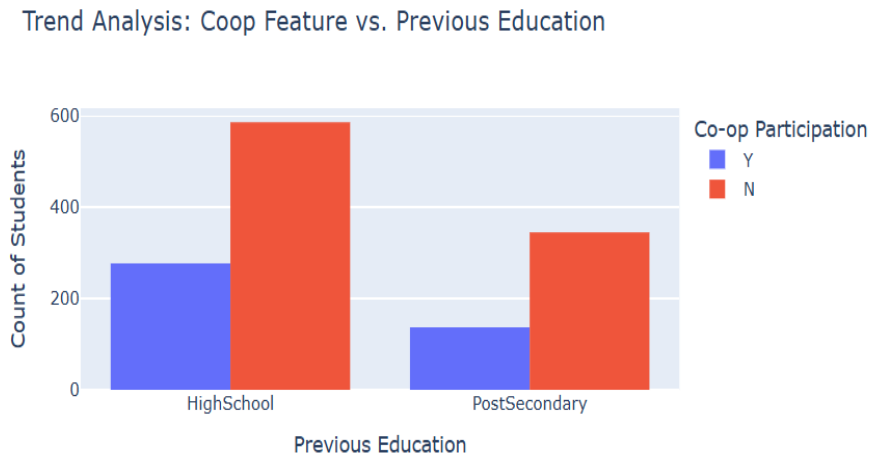
- First Term GPA and First Year Persistence (0.54)
- First Language and Residency (0.63)
- Funding and Residency (0.72)

Strong Negative Correlation:

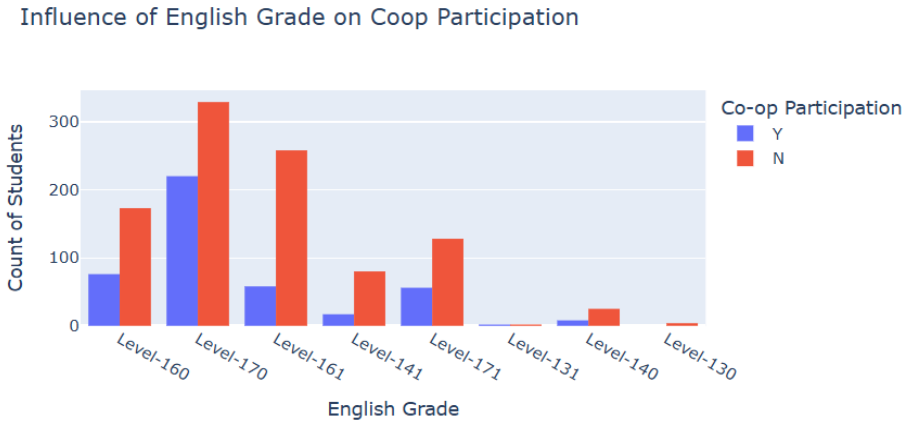
- First Language and Fast Track (-0.51)
- Residency and Fast Track (-0.57)

Neutral/ Weak Correlation:

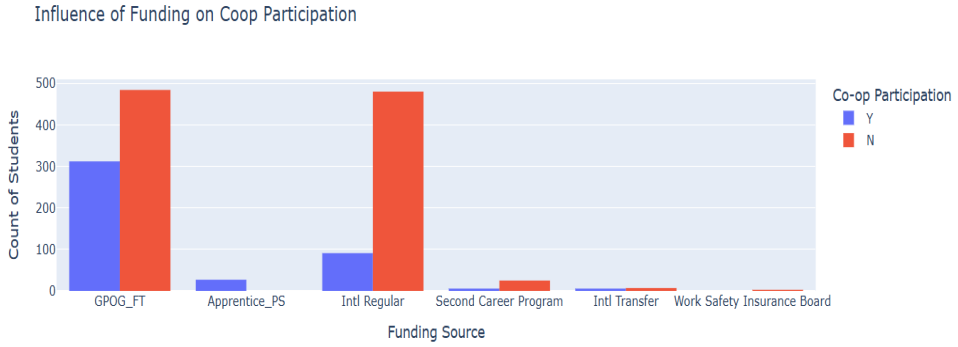
- Gender and First-Year Persistence (-0.08)
- Age Group and English Grade (0.09)
- Math Score and Coop (0.01)



Prev Education has 863 records corresponding to high school and 482 records corresponding to post-secondary education. From this visualization, we can observe that there are more students who dont participate in co-op overall, and Students whose previous education was Post Secondary have a lower participation in Co-op, this can be due to alternate career paths, internships, etc



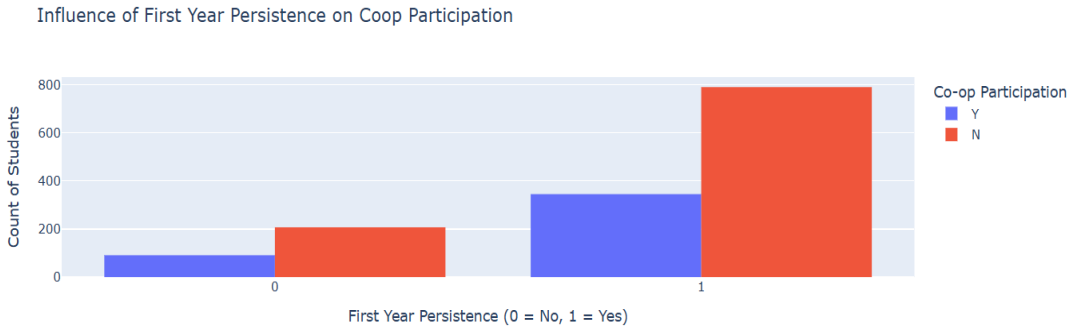
As we can see more students are participating in coop for students with English Grade Level 160 and Level 161, while Level 170 has the highest non-coop count. This suggests that coop participation is more common for those with Higher English Grades



Influence of Funding on COOP Participation

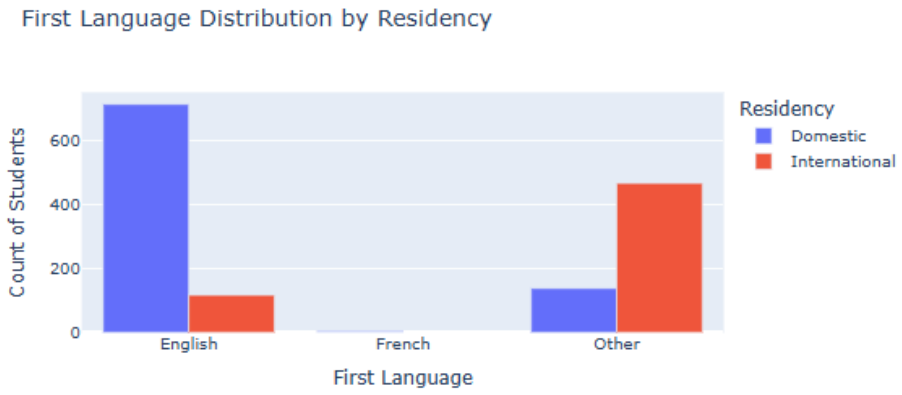
We can notice that GPOG_FT and Intl Regular funding categories have the highest student counts, with non-Co-op participants outnumbering Co-op participants.

Other funding categories, such as Apprentice_PS and Second Career Program, show minimal participation in co-op and non-co-op programs.



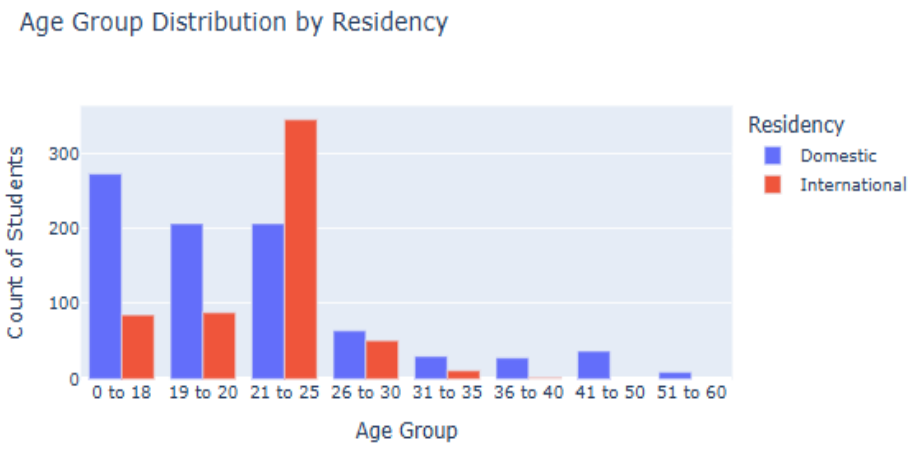
Influence of First-Year Persistence on Coop Participation

Here, we see a bar chart that looks at the effect of first-year persistence (0 = No, 1 = Yes) on co-op participation (Y/N). Students who persisted after semester 1 (1) outnumber those who did not (0) by a large margin in "persistent" students; non-participants (red) vastly outnumber participants (blue). For the non-persistent students, attendance is minimal, suggesting a strong association between the first year of persistence and co-op participation. It also emphasizes how crucial stable academic pathways are for co-op involvement.



First Language Distribution by Residency

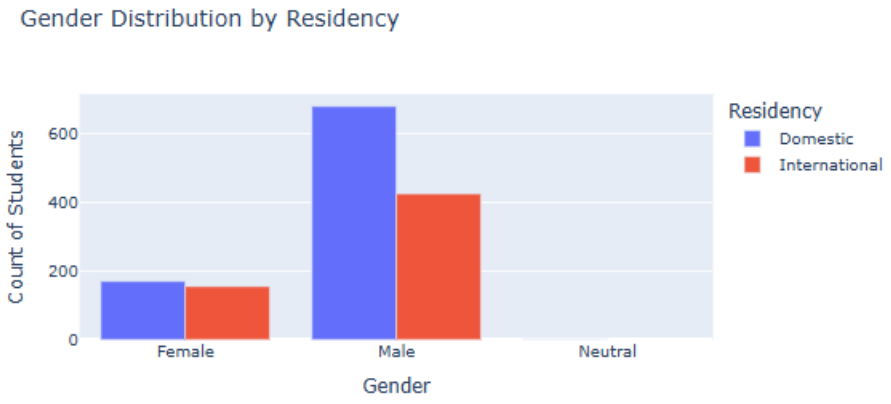
From this graph, we can understand that English is the language widely spoken among domestic residents compared to international residents. On the other hand, international residents are more comfortable with their own native language (Other) than English. However, no student speaks the French language.



Age Group Distribution by Residency

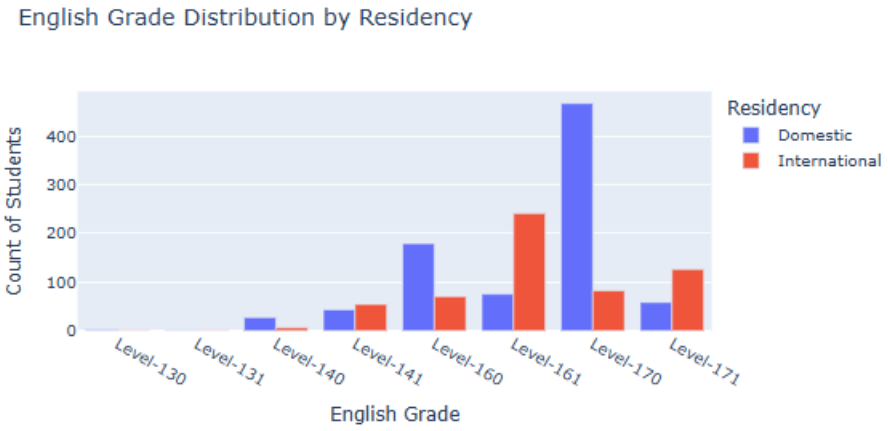
We can observe that Domestic Students fall in the age group from 0-25 with few students in higher age groups. Whereas International Students mostly fall in the age group of 20-25 with few studetns in between 0-20. This tells us that most students who come to study from other countries are from the age group 21-25.

Also, there are no International students who pursue their education after the age of 40, whereas there are 37 Domestic students who fall in the age group 41-50 and only 9 students who are older than 51



Gender Distribution by Residency

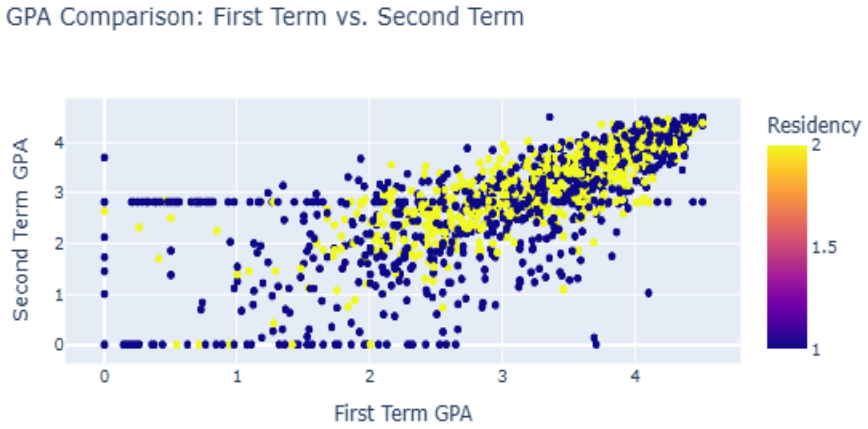
From the above visualization, male students dominate the domestic and international residency with high counts. Female students are less compared to males but have a balanced view in terms of residency, whether it is domestic or international. On the other hand, the neutral gender represents very little in both the residency (domestic and international).



English Grade Distribution By Residency

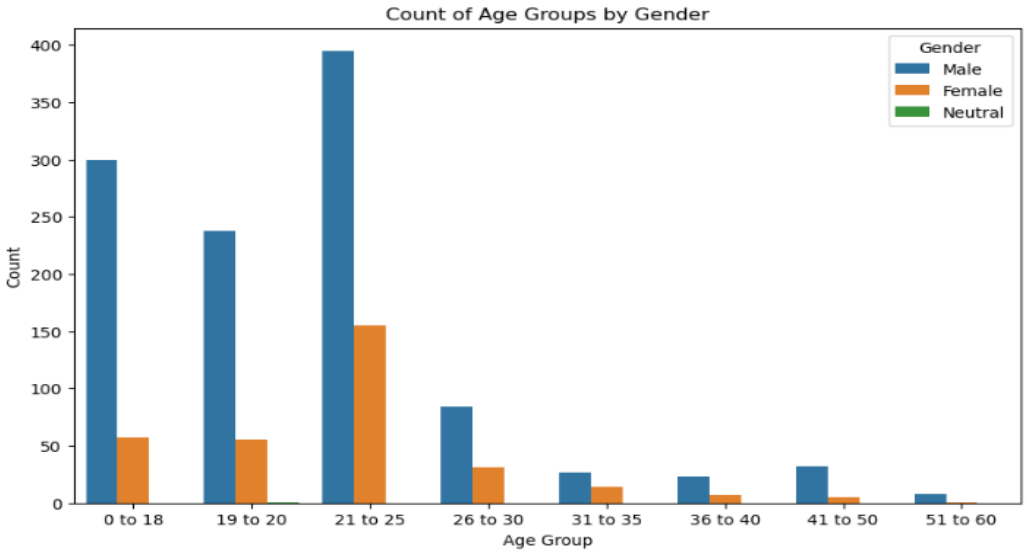
Majority of Domestic Students have a English Grade of Level 170, with few students in Englsih Grade Level 160, 161 and 171. Whereas majority of International Students have a English Grade of Level 161, 171, and 170, with few students with English Grade of Level 160,141, and very few with lower English Grade.

This tells us that while majority of Domestic Students have a higher English Grade compared to International Students



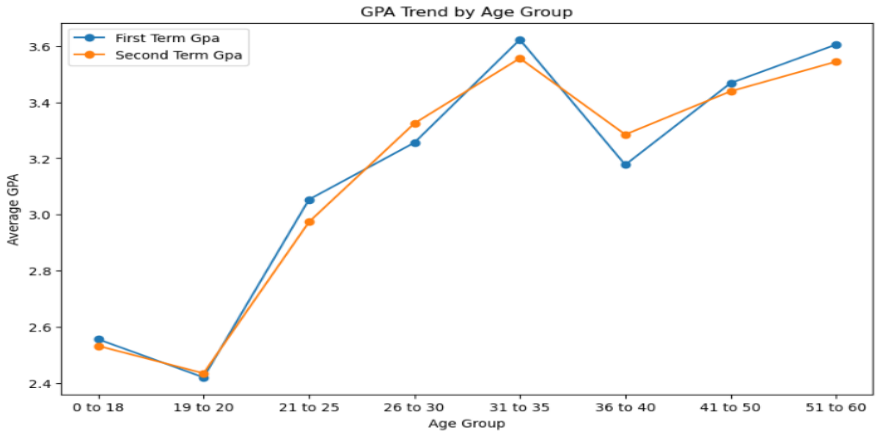
Scatterplot – First Term GPA vs Second Term GPA

Here, the data highlights students'GPAs for the first and second term, and higher residency students have better academic performance, clustering towards the upper right side of the chart. While some students maintain their performance and other fluctuations.



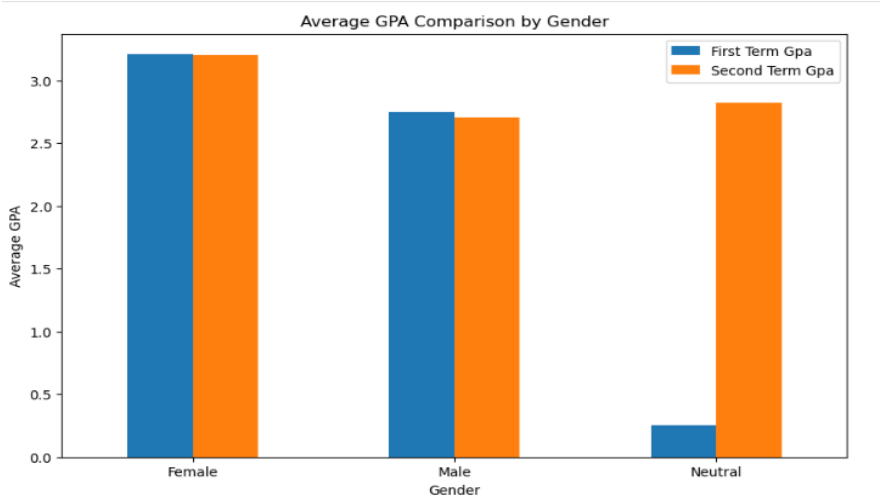
Age Groups by Gender

So, we can see that the age group which falls between 21 and 25 has a high count, stating a larger population in that range. Overall, the male population appears to be higher than females and is neutral in every age category. Well, another trend we can see is that as the age group increases, a decline can be noticed in the overall population count.



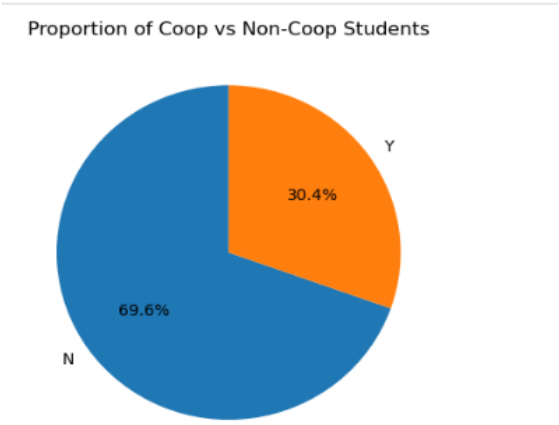
GPA Trends by Age Group

This chart shows the GPA trend based on age group. Overall, the first-term and second-term GPAs show no notable changes across the age groups. Both the first term Gpa and Second Term Gpa start with low in the younger age groups (0 to 18 and 19 to 20) and increase drastically to almost 3 gpa by the age (21 to 25) and show growth by (31 to 35), which was the peak and then dropped to 3.2 by (36 to 40) before showing a positive growth.



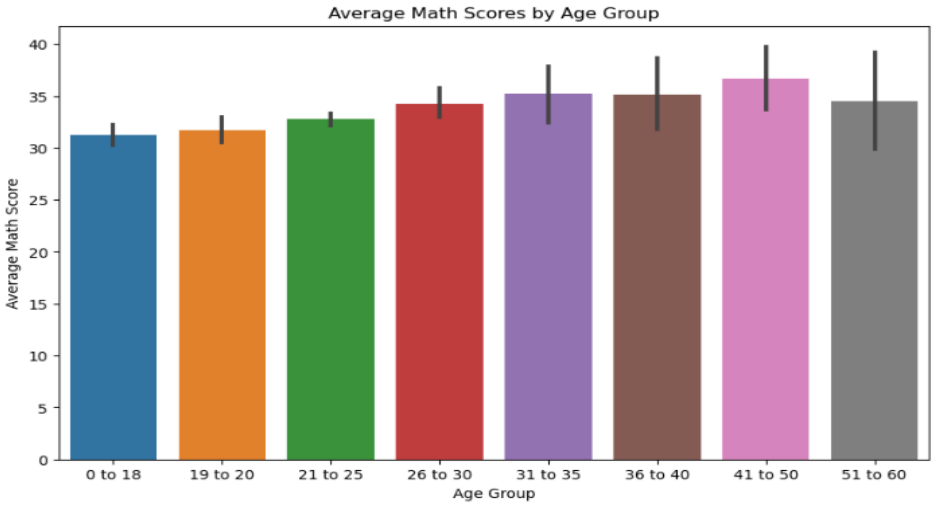
Average GPA Comparison by Gender

This chart highlights the gender-based differences in their academics. The female students have higher Gpa in the terms “First Term Gpa” and “Second Term Gpa” than males and neutrals. Also, we can see a significant growth in GPA for neutral categories in the second term compared to the first term.



Proportion of Coop vs Non Coop Students

This chart depicts the proportion for Coop and Non-Coop Students, where the greatest number of students fall into the non-coop category, making it about 69.6%, making up over 2/3rd of the total student population. at the same time, the rest of them are Coop students, representing 30.4%.



Average Math Scores by Age Group

The maximum number for the math score is 50, and after visualizing the Average Math Score with Age Group to get an insight into which age group has the highest number of Average Math Score and from the graph we can see that as the Age increases, chances of getting high Math score also increases except for the Age Group "51 to 60" which shows a drop in the math score which can be due to age factor.

Artificial Neural Networks to Predict First-Year Persistence

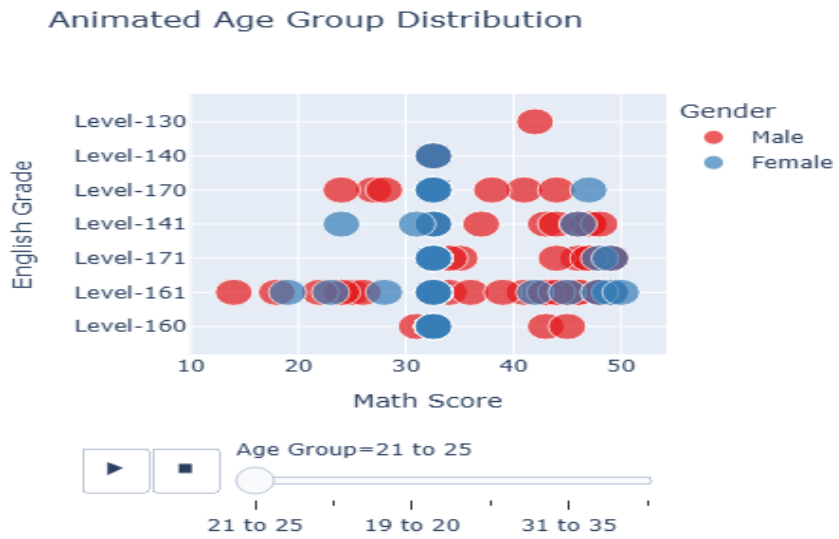
Implementing the Sequential Model with 2 hidden layers, using ReLu activation we successfully predicted the First Year Persistence. Scoring an accuracy of 86% on our Train Data, and 88% accuracy on our Test Data. This indicates our model is performing well on both seen and unseen data.

```
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.8573 - loss: 0.3532 - val_accuracy: 0.8565 - val_loss: 0.3533
Epoch 19/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8577 - loss: 0.3453 - val_accuracy: 0.8826 - val_loss: 0.3399
Epoch 20/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.8545 - loss: 0.3542 - val_accuracy: 0.8783 - val_loss: 0.3430
Epoch 21/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.8435 - loss: 0.3710 - val_accuracy: 0.8739 - val_loss: 0.3397
Epoch 22/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.8531 - loss: 0.3643 - val_accuracy: 0.8826 - val_loss: 0.3387
Epoch 23/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.8525 - loss: 0.3650 - val_accuracy: 0.8826 - val_loss: 0.3385
Epoch 24/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - accuracy: 0.8458 - loss: 0.3650 - val_accuracy: 0.8783 - val_loss: 0.3376
Epoch 25/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - accuracy: 0.8353 - loss: 0.3711 - val_accuracy: 0.8783 - val_loss: 0.3425
Epoch 26/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.8488 - loss: 0.3557 - val_accuracy: 0.8652 - val_loss: 0.3480
Epoch 27/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.8701 - loss: 0.3325 - val_accuracy: 0.8783 - val_loss: 0.3373
Epoch 28/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.8684 - loss: 0.3389 - val_accuracy: 0.8913 - val_loss: 0.3384
Epoch 29/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.8552 - loss: 0.3613 - val_accuracy: 0.8739 - val_loss: 0.3445
Epoch 30/30
29/29 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.8624 - loss: 0.3575 - val_accuracy: 0.8696 - val_loss: 0.3455
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9079 - loss: 0.3049
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8731 - loss: 0.3406
Train Accuracy: 0.86
Test Accuracy: 0.88
```

Train and Test Accuracy Screenshot

Plotly Dashboard

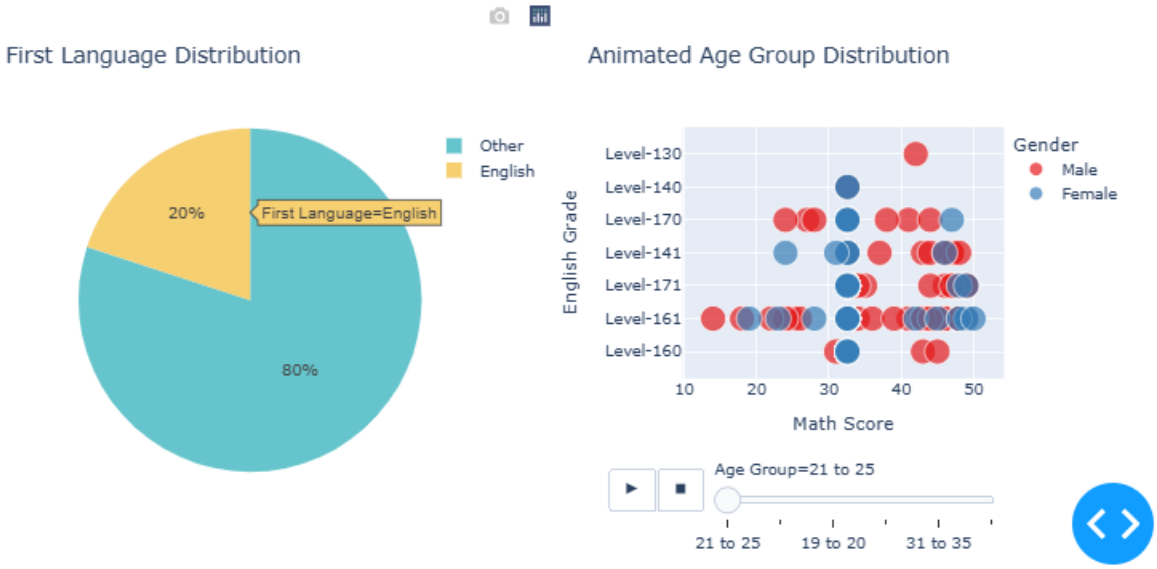
We created 4 plotly visualizations with the slicer for Residency that lets us visualize the data for Domestic and International students. We also added an Animation Slider of the Age to explore the age groups dynamically.



Dashboard Visualizations – Scatterplot – Age Distribution

In this chart, we can observe the distribution of the age group of students with their English Grade; the red color dots represent Males, and the blue color dots represent Females. The animation slider allows the exploration of age groups dynamically.

Dashboard Visualizations for Domestic



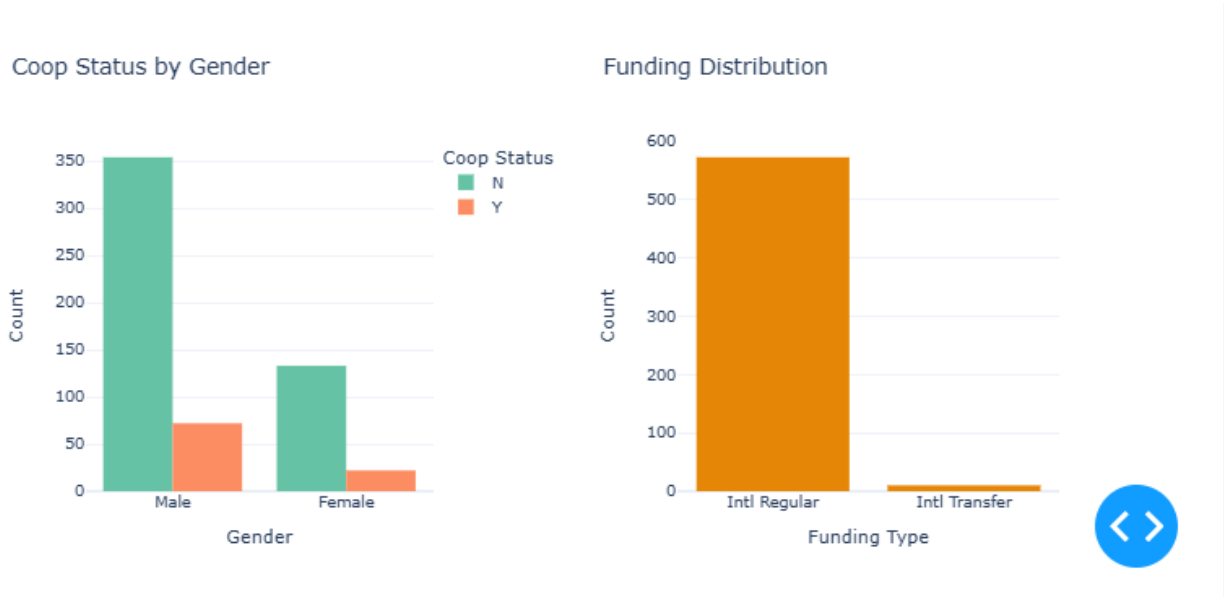
Dashboard Visualizations for International

We can see that most first-language people are English for Domestic and International Students. International Students don't speak French at all while there is a very small count of students who speak French who are domestic.

Dashboard Visualizations for Domestic

We can observe the coop status by Gender for Domestic and International Students, There is no neutral Gender for International Students.

The Funding Distribution barchar shows us the Funding these students have. Domestic Students have more funding and a variety of Funding, whereas International Students only have Intl Regular, Intl Transfer.

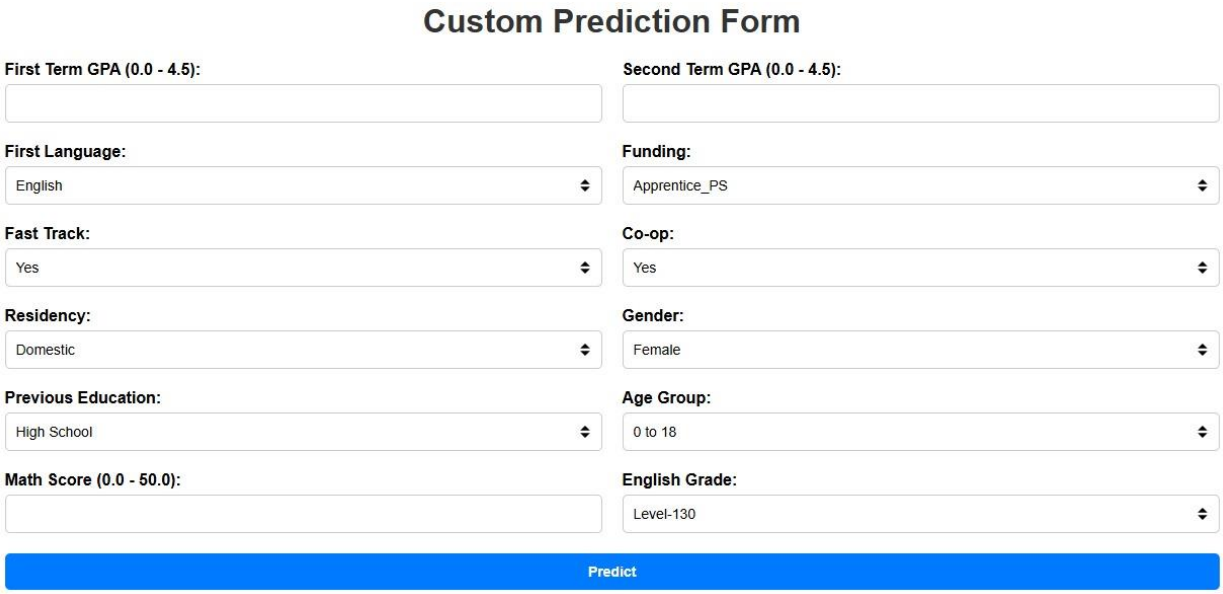


Dashboard for International Students

We save the entire code into a .py file so that we can use it to visualize it locally on a website by integrating it with GraphQL..

FAST Frontend

A front end was built using FAST; we integrated it with GraphQL by Apollo Client, which dynamically renders data and visualizations. We also integrated our Neural Networks model and created a custom predictions form so that users can input the features to check whether a Student will continue his education based on the model.



Custom Prediction form on website

Result

We have successfully built a Full Stack Intelligent Solution to predict the First Year Persistence in the Student Dataset. Using FAST to build the front end, we have integrated it with GraphQL. We have also performed several visualizations using Plotly Express, Matplotlib, and Seaborn. Using Plotly Express and Dash, we have created a dashboard that conveys useful insights on the Residency of Students; the dashboard has 4 main visualizations – A pie chart that explains the distribution of First Languages, A Scatterplot on the Age Distributions, The coop status by Gender, and Funding Distribution.

Conclusions and Future Work

In this project, we analyzed the student data and made useful insights; we dove deep into various features and used Plotly Express and Dash to create a dashboard. It was a very informative project. We also use artificial neural networks to generate a prediction for first-year persistence. We also built a frontend using FAST and integrated the visualizations using GraphQL.

In the future, we can implement more complex models with better parameters to increase the Accuracy on our Train and Test Data. We can also gather a larger dataset to help train our model.

References

Plotly. (n.d.). *Python: Animations*. Retrieved December 13, 2024, from https://plotly.com/python/animations/