Academic Performance and Success Predictor

Background

The data was collected from the Faculty of Engineering and Faculty of Educational Sciences students in 2019. The information collected from each student ranges from basic demographics to classroom and study habits. Along with the information collected was the student's cumulative GPA from the past semester out of a 4.0, their expected cumulative GPA during graduation, and their grades ranging from pass to fail (0: Fail, 1: DD, 2: DC, 3: CC, 4: CB, 5: BB, 6: BA, 7: AA).

Problem Statement

In Higher Education, having a high GPA is how to place on the honor roll, giving the perk of graduating with honors, sum cum laude, or magna cum laude. However, for many students their GPA is important for maintaining their scholarships and perhaps securing an internship or co-op. By being able to predict a student's expected GPA and what factors help contribute to GPA, we could notify students and create a plan to improve their GPA before the end of the semester.

Exploratory Data Analysis and Data Wrangling

The dataset contains a total of thirty-three features. Each of the features can be grouped into the following categories: basic demographic, classroom habits, study habits, and grades. Each of the categorical variables are mapped out to numerical values (i.e Mother's occupation: 1: retired, 2: housewife, 3: government officer, 4: private sector employee, 5: self-employment, 6: other).

Correlations between the target variable 'EXP_GPA' and the features were relatively low. Understandably, the only feature which has the most correlation at 0.66 is 'CUML_GPA' which is the student's cumulative GPA from the past semester.

The following graphs display the correlation between the features and the target, EXP_GPA:

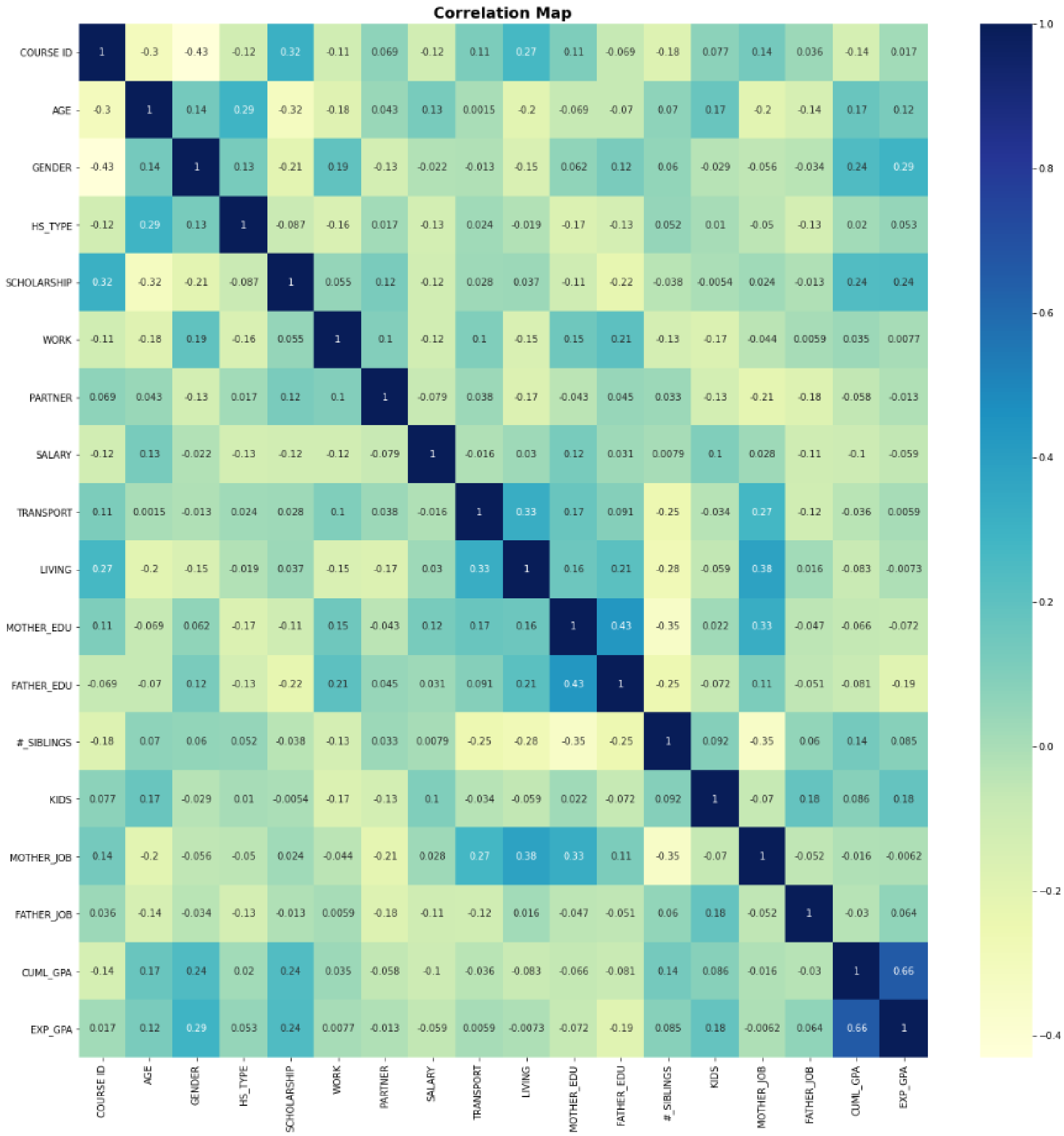


Figure 1: Correlation between basic demographic features and 'EXP_GPA'

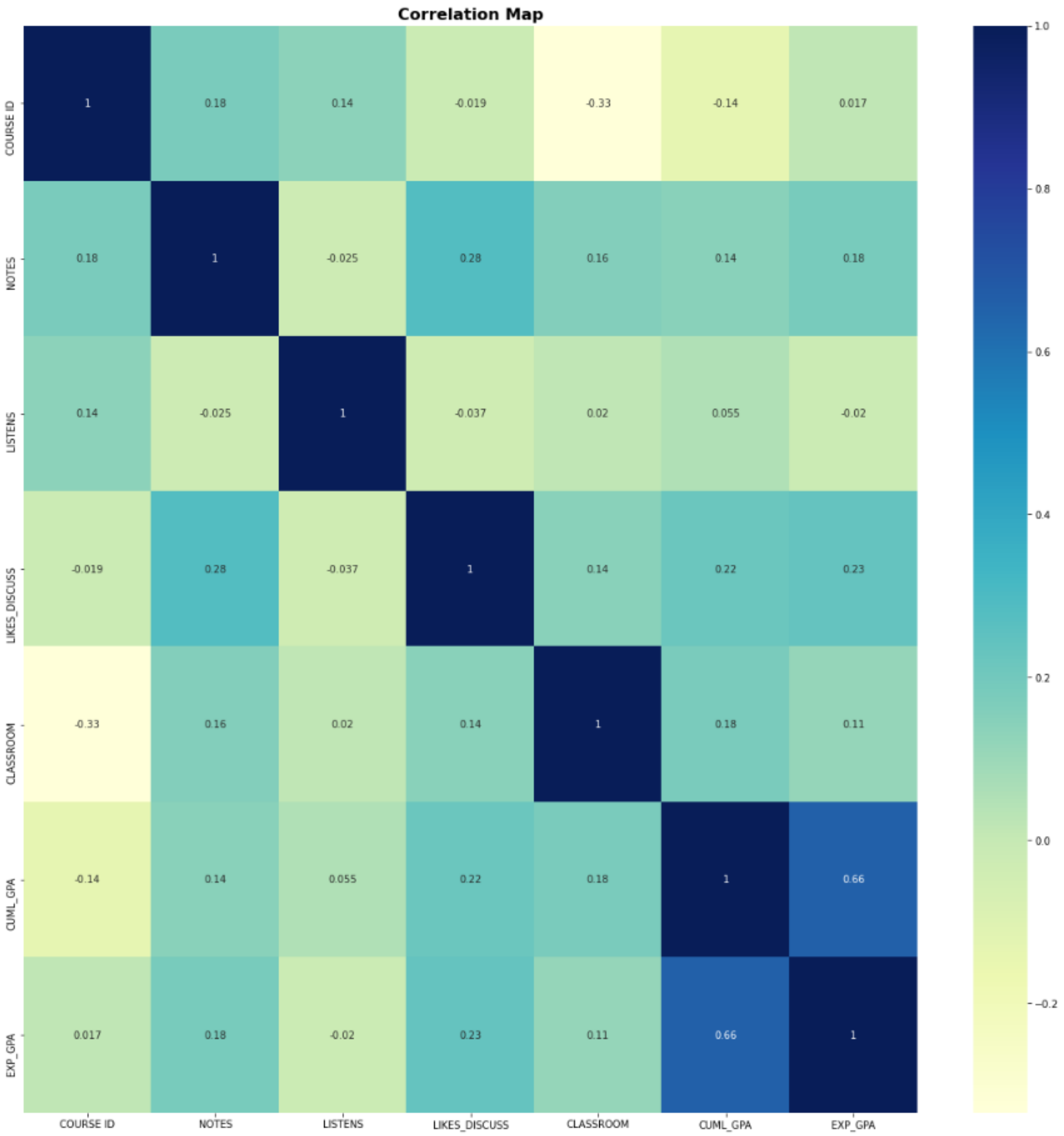


Figure 2:Correlation between classroom habit features and 'EXP_GPA'

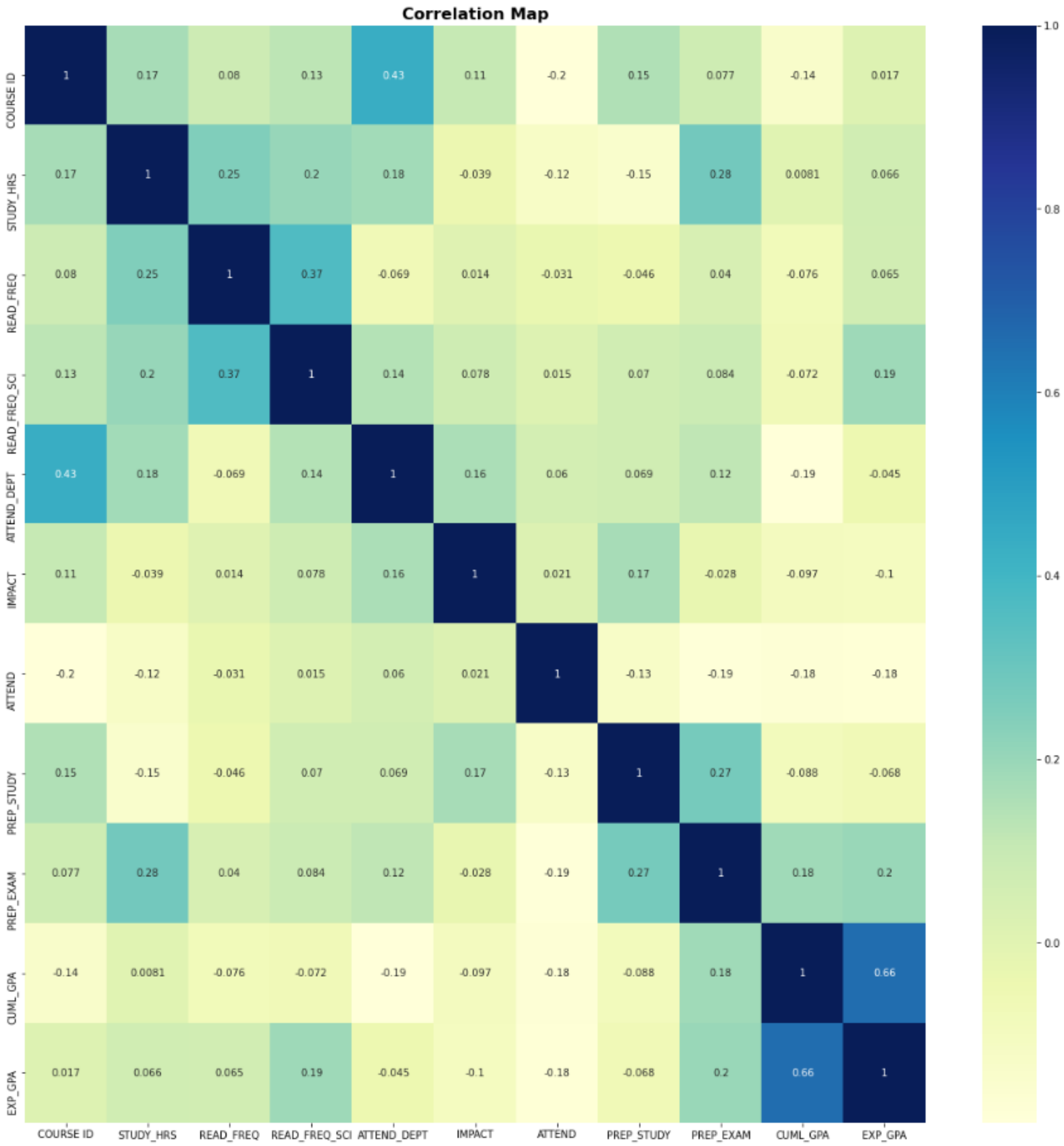


Figure 3: Correlation between study habit features and 'EXP_GPA'

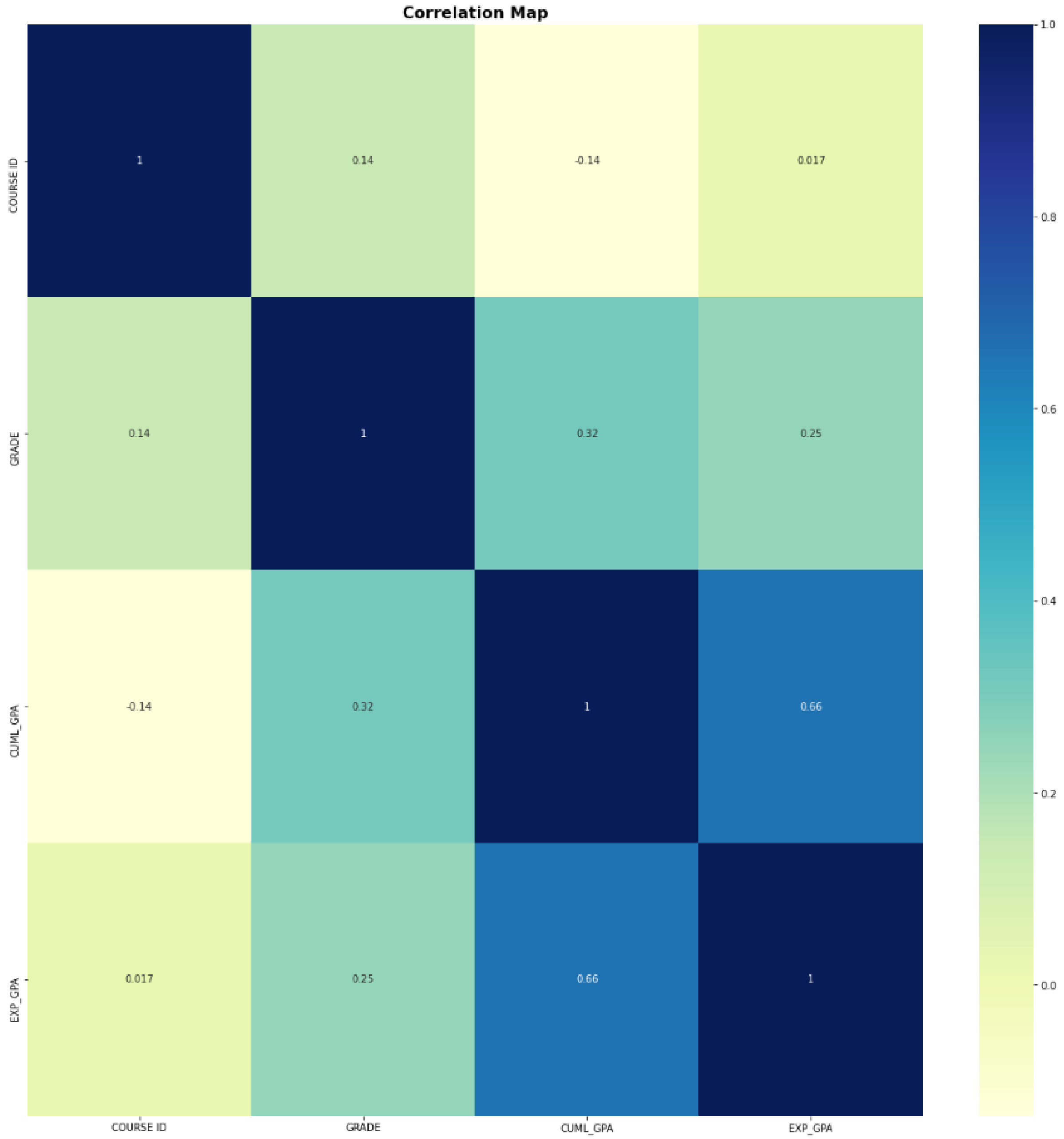


Figure 4: Correlation between grade features and 'EXP_GPA'

After exploring all the features, I wanted to take a closer look at features which have at least a correlation of 0.2. These features included:

- SCHOLARSHIP
- LIKES_DISCUSS
- GRADE

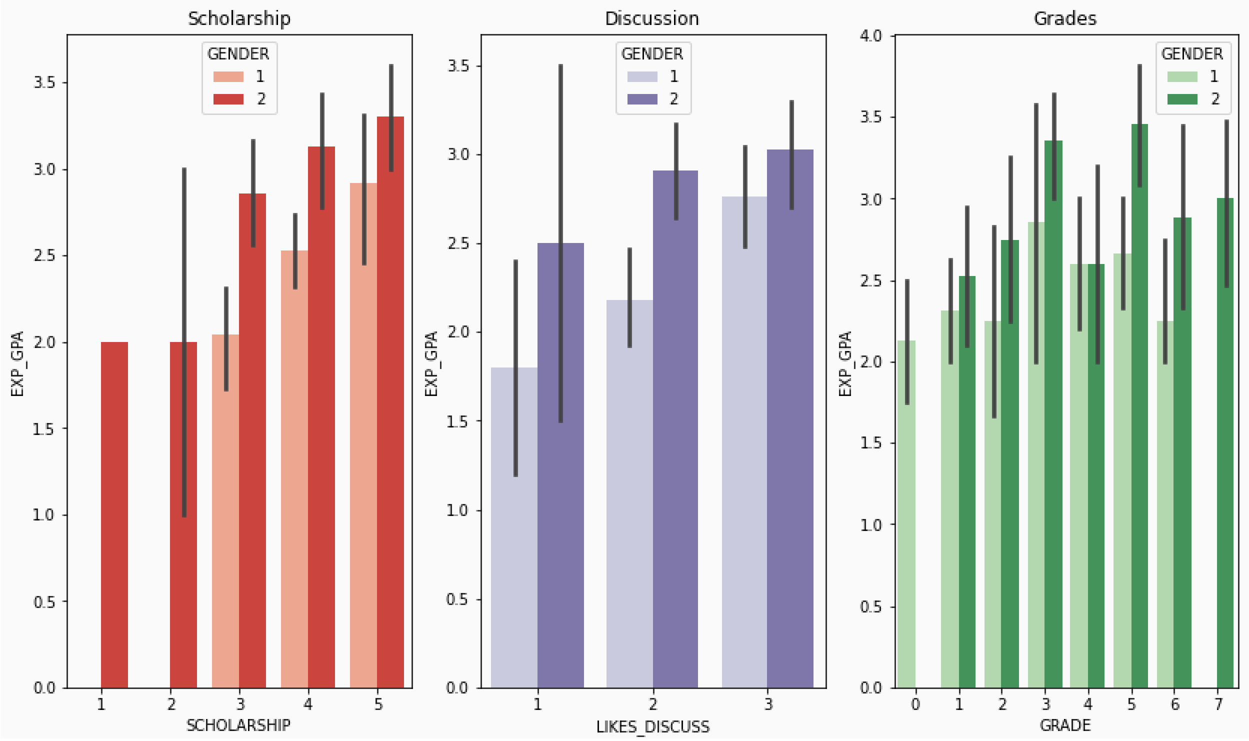


Figure 5: comparing gender, scholarship, discussion, and grades to EXP_GPA

From these graphs which compare gender, exp GPA to the following scholarship, likes discussions and grades, some patterns I noticed include:

- Students who receive a scholarship that covers at least 75% of tuition tend to have a higher expected GPA
- Generally, students who participate or like discussing in class have a higher GPA than those who never discuss (with a couple outliers)
- In general, males tend to have better grades

Let's take a closer look at the courses and the grades of the students in those courses and how that compares to the expected GPA…

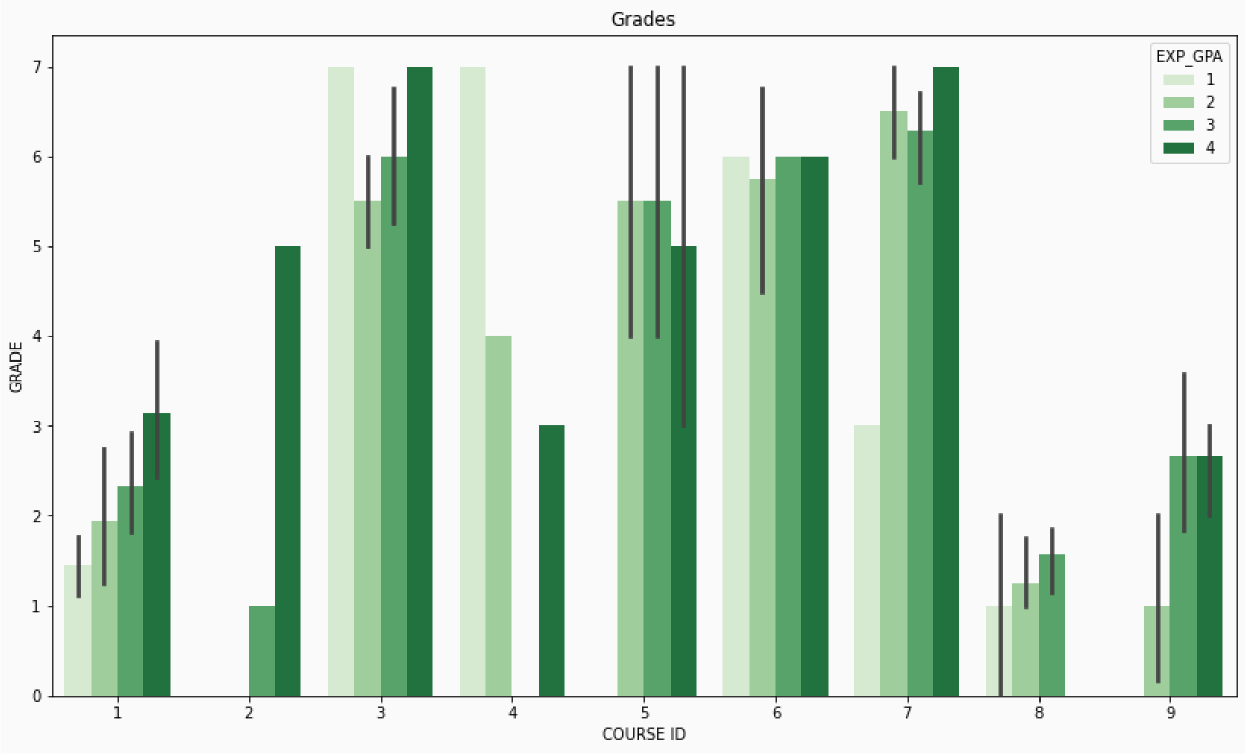


Figure 6: In-depth look at courses and grades to EXP_GPA

From these graphs:

In general, if you perform well in a course your expected GPA should increase. However, there are some unusual patterns:

- In course 2, students who performed poorly (1:DD) still have an expected GPA of 3
- In course 3, students who perform well (7:AA) have an expected GPA of 1
- In course 4, students who perform well (7:AA) have an expected GPA of 1 and students who performed average (3: CC) have an expected GPA of 4
- In course 5 & 6, students performed the same yet expected GPA ranges between them
- In course 8, despite the students performing below average (2:DC) they have expected GPAs of 1 to 3
- In course 9, students perform average (4:CB) and still have expected GPA of 4

Exploring the target variable: ‘EXP_GPA’

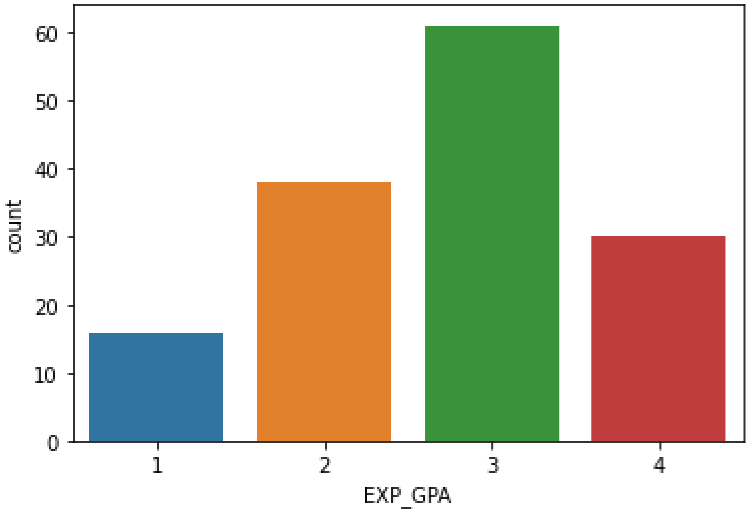


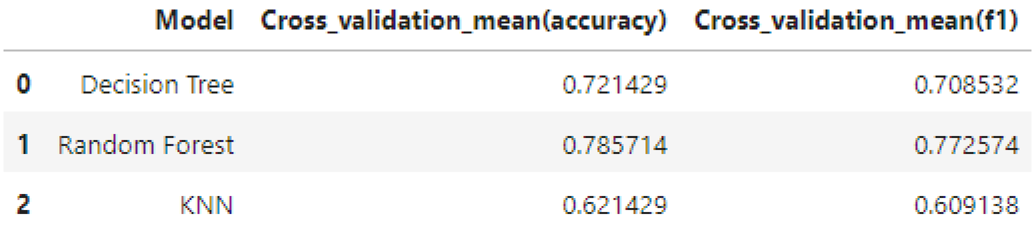
Figure 7: Overview of target variable distribution

Viewing the target feature, we can see how the target variable is imbalanced, so this is something that I needed to handle before creating the models. Since the dataset is small (145 rows and 33 columns) I used oversampling techniques such as:

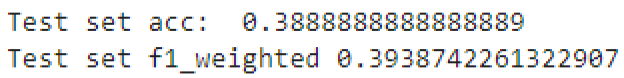
- Simple Random Oversampling
- SMOTE
- ADASYN

Model Selection

Oversampling



	Model	Cross_validation_mean(accuracy)	Cross_validation_mean(f1)
0	Decision Tree	0.721429	0.708532
1	Random Forest	0.785714	0.772574
2	KNN	0.621429	0.609138

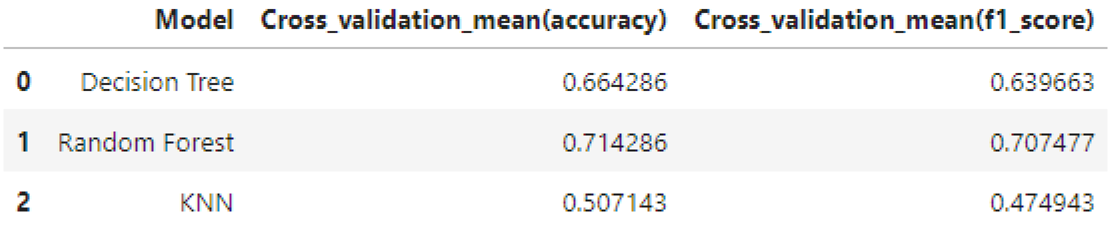


```
Test set acc:  0.3888888888888889
Test set f1_weighted 0.3938742261322907
```

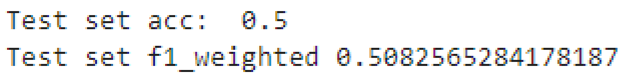
Since this is a multiclass classification, I decided to use 2 metrics to compare the models: accuracy and f1_weighted score

On the training data it looks decent with scores as high as .7, but the test is poor, meaning that this oversampling method did not generalize well across the data leading into an issue of overfitting

SMOTE



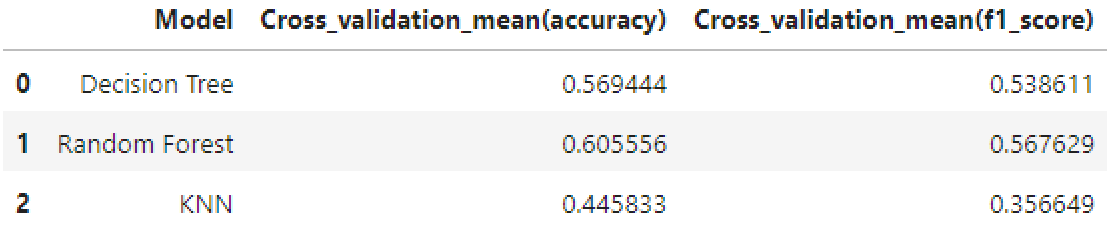
	Model	Cross_validation_mean(accuracy)	Cross_validation_mean(f1_score)
0	Decision Tree	0.664286	0.639663
1	Random Forest	0.714286	0.707477
2	KNN	0.507143	0.474943



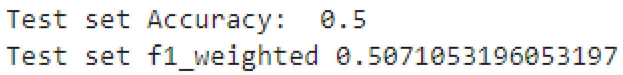
```
Test set acc:  0.5
Test set f1_weighted 0.5082565284178187
```

On the training data it looks decent with scores as high as .7, but the test is poor, meaning that this oversampling method did not generalize well across the data leading into an issue of overfitting

ADASYN



	Model	Cross_validation_mean(accuracy)	Cross_validation_mean(f1_score)
0	Decision Tree	0.569444	0.538611
1	Random Forest	0.605556	0.567629
2	KNN	0.445833	0.356649



```
Test set Accuracy:  0.5
Test set f1_weighted 0.5071053196053197
```

While the training data is scoring around .5, .6, isn't the greatest score, the test set appears to score the same as well. So I can see that using the ADASYN method did not result in overfitting leading this to be the best model so far.

Model Improvements:

For each of the models, I tuned their parameters to improve the model accuracy.

For the decision tree, I found which maximum depth would be the most accurate. For the Random Forest, I tuned the n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf. For the KNn model, I found which number of neighbors would produce the most accurate result.

Model	Accuracy Before	Accuracy After
Decision Tree	0.569	0.638

Random Forest	0.605	0.693
KNN	0.443	0.521

Key-Takeaways and Future Work

From the results of each model and the oversampling methods, it seems as though the best model is a Random Forest with the oversampling method of ADASYN. Other oversampling methods resulted in overfitting of the data whereas the ADASYN method was better suited in generalizing over the model. The following features that were related to the model includes:

- WORK
- LIKES_DISCUSS
- READ_FREQ
- IMPACT
- CUML_GPA

With an accuracy score of ~69%, we can perform further research to improve the models.

Ways to improve:

- Experiment with other oversampling methods
- Test out different classification models
- Gather more data