2023

Case Study: Student College Experience

GROUP 12

- RAJVI MEHTA - 0788372
- AMIT SHARMA - 0794488
- SUHAIL AHMED - 0789949
- JAYRAJ RADADIYA - 0789984
- HARSHIL PATEL - 0791261

Contents

Abstract

The college experience is critical for students, and there are many factors that can influence its quality. This study aims to **examine the relationship between demographics, work history, living conditions, and educational background, and their impact on the day-to-day college experience of students**. A review of the literature suggests that these factors can all play a role in shaping the college experience, but there is a need for further research to identify the specific patterns and correlations that exist. This study uses a **survey-based approach to collect data from a sample of college students, with a focus on identifying the most salient factors that impact the college experience. By examining the data and identifying key patterns and correlations, this study aims to provide insights that can inform strategies to understand academic experience and support the success of college students.**

Introduction

College experience is critical in every student's life. It is a time of growth, self-discovery, and academic advancement. Achieving an optimal college experience is a common goal among all students, but the factors that contribute to a desirable college experience can vary from student to student. Personal and professional background, residential life experience, living flexibility, and academic factors such as socioeconomic status, race, and ethnicity can all play a significant role in shaping one's college experience. Campus culture, extracurricular events, services provided, and social opportunities are also factors that can impact the overall college experience.

It is crucial to understand how several factors affect students' day-to-day lives and overall quality of life. Therefore, this study aims to examine the relationship between three specific factors and the college experience: demographics, work history, and living conditions. By identifying patterns and correlations between these factors and the college experience, the study aims to provide insights that could be used to improve academic experience.

Through this study, we hope to gain a better understanding of how demographics, work history, and living conditions can influence the college experience. The results of this study could help universities and colleges create policies and programs that promote a more positive and supportive academic environment, leading to better outcomes for students.

The **hypothesis** of this study suggests that students who have stable work histories, relatable educational backgrounds, and better living situations are more likely to have a positive college experience. There are several reasons why these factors could be related to a positive college experience.

First, students who have stable work histories may have an easier time balancing work and academic commitments, leading to a better work-life balance. They may also have more financial stability, which could reduce stress and allow for a more enjoyable college experience.

Second, students who have educational backgrounds that are relatable to their academic pursuits may find it easier to navigate the academic environment and perform well in their courses. This could lead to a sense of accomplishment and satisfaction with their academic progress, contributing to a positive college experience.

Finally, students who have better living situations, such as comfortable and safe living accommodations, may feel more relaxed and focused, allowing them to better engage in academic and social activities. They may also have more opportunities to socialize and participate in extracurricular activities, which could contribute to a positive college experience.

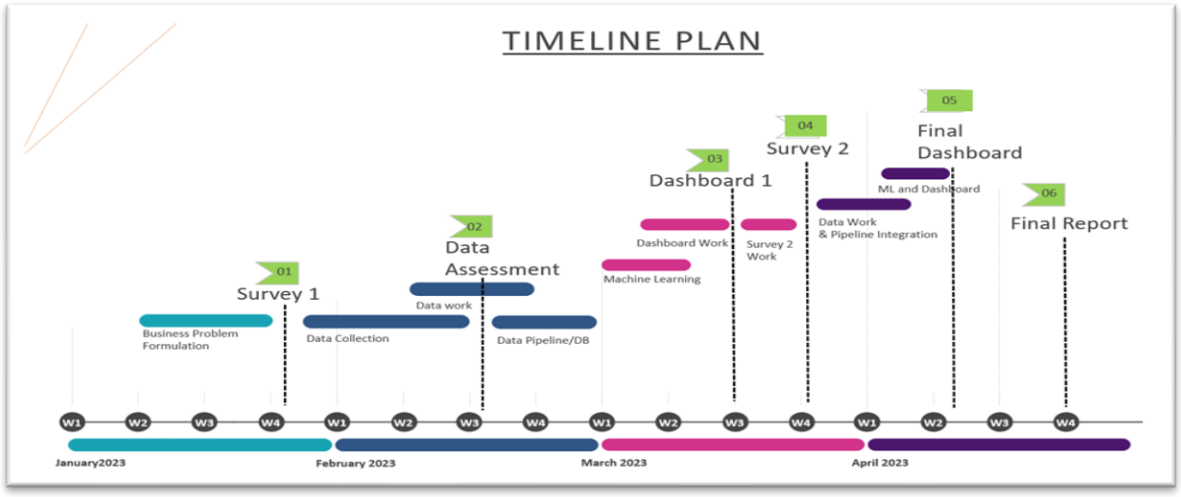
By examining the relationship between these three factors and the college experience, this study aims to provide insight into how universities and colleges can better support their students.

Literature Review

A comprehensive overview of the relevant research on the factors that impact the college experience, highlighting the most important findings and discussing the implications for the research question are given below:

1. The impact of demographics on college experience: There is a growing body of research that examines how demographic factors such as race, ethnicity, and socioeconomic status can impact college experience. For example, studies have shown that students from underrepresented racial and ethnic groups may experience social isolation and marginalization, which can negatively impact their academic and social experience (Chang, Astin, & Kim, 2014). Similarly, students from lower socioeconomic backgrounds may experience financial strain and have less access to resources, which can also negatively impact their college experience (Stephens, Hamedani, & Destin, 2014).
2. The role of work in college experience: Work can have both positive and negative effects on the college experience. On the one hand, work can provide financial stability and valuable work experience, which can be beneficial for career prospects (Goldrick-Rab, 2016). On the other hand, work can also contribute to stress and reduce the amount of time available for academic and social activities, which can negatively impact college experience (Goldrick-Rab, Broton, & Eisenberg, 2016).
3. The importance of living conditions for college experience: Living conditions, such as housing quality, safety, and location, can have a significant impact on the college experience. Studies have shown that students who live in comfortable and safe accommodations are more likely to have a positive college experience (Burt, Simons, & Gibbons, 2012). Similarly, students who live on campus or near campus may have more opportunities to engage in academic and social activities, which can contribute to a positive college experience (Hossler & Gallagher, 1987).
4. The relationship between educational background and college experience: Students who have educational backgrounds that are relatable to their academic pursuits may have an easier time navigating the academic environment and performing well in their courses. For example, students who have taken advanced courses in high school may be better prepared for college-level coursework (Horn & Chen, 2018). Similarly, students who have previous experience in a particular field may be more likely to succeed in related academic pursuits (Xu & Smith, 2016).

Timeline



The project lasted for **4 months**, from January to April 2023.

In the **first month**, the team planned the data architecture and methodology for the project, researching different technologies and tools to support the data flow process.

During the **second month**, the team collected data through surveys, performed EDA and data cleaning, and set up the database landing zone and connection APIs.

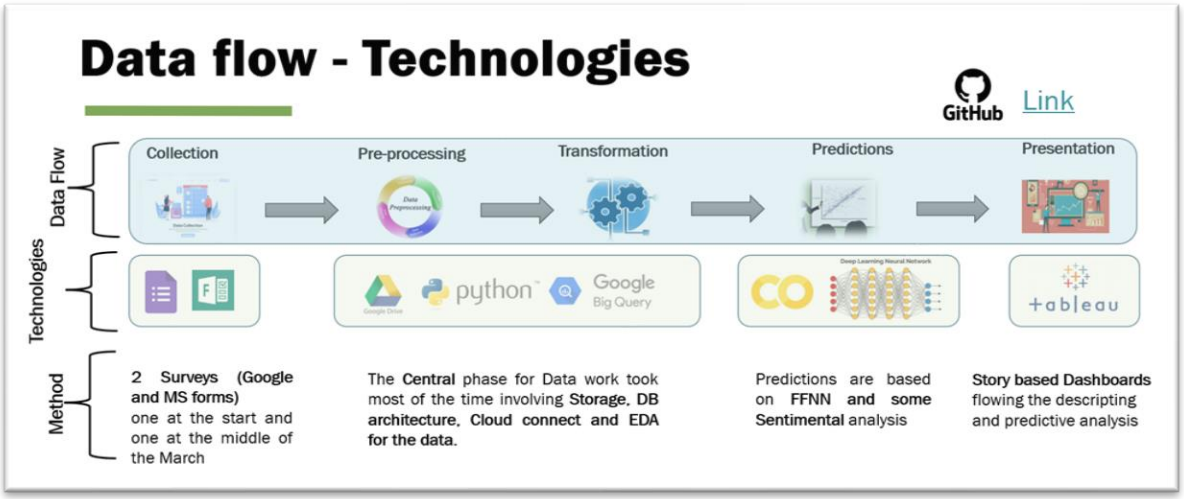
The **third month** involved managing the data from the surveys, deriving predictions and sentiment analysis, and creating descriptive graphs and dashboards for the presentation layer.

In the **final month**, the team concluded the results, finalized the dashboard, and documented the project into presentations and reports.

The timeline was structured to ensure each phase of the data flow process received sufficient time and attention, resulting in a high-quality solution delivered within the stipulated timeline with well-documented project outcomes.

Methodology

Data Flow and Technologies



The entire process is broken into 5 phases of Data flow, namely Data Collection, Pre-Processing, Transformation, Prediction and Presentation.

Data Collection: This is the first phase of the data flow, where data is collected from various sources such as Google Forms and Microsoft Forms. These forms provide an easy-to-use interface for data entry and can be customized to suit specific needs.

Pre-processing: Once the data is collected, it needs to be cleaned and prepared for further analysis. This is done in the pre-processing phase. Python is a tool for data pre-processing due to its ease of use, flexibility, and powerful data manipulation libraries.

Transformation: The transformed data is then fed into the transformation phase, where it is analyzed using various statistical and machine learning techniques to extract valuable insights. Big Query, which is a fully managed, serverless data warehouse provided by Google Cloud, is often used for large-scale data transformation due to its scalability and performance.

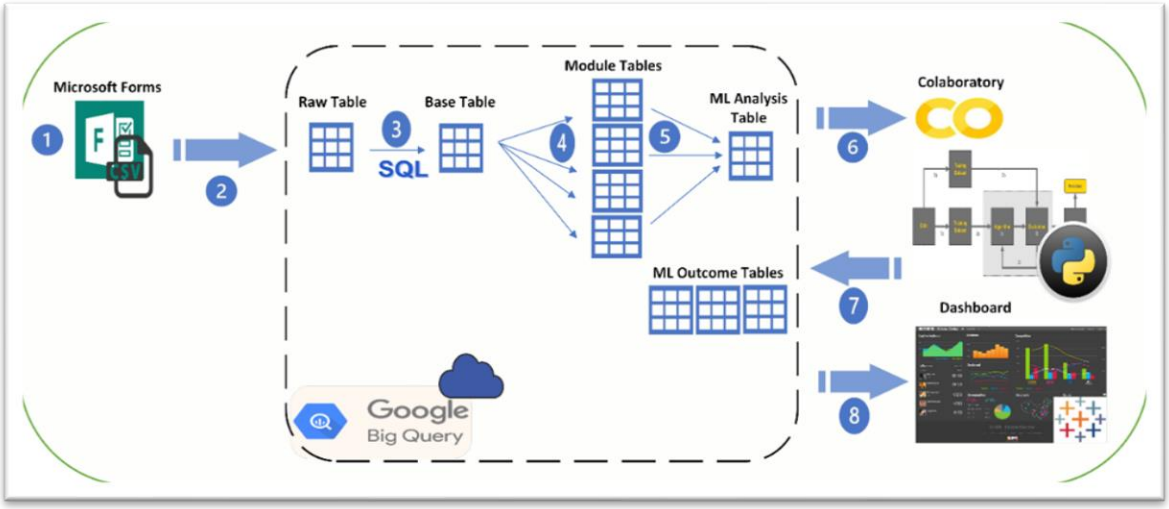
Prediction: Once the data has been transformed, it can be used for prediction using machine learning models. Python is again the choice for building and training machine learning models due to its rich set of libraries and frameworks. Feed-forward neural networks are a type of machine learning model that can be used for prediction tasks.

Presentation: Finally, the insights and predictions generated from the previous phases are presented to the end-users. Tableau Server is a business intelligence platform that provides

interactive data visualization tools to create informative dashboards and reports for effective communication and decision-making.

Overall, this 5-phase data flow process ensures that data is collected, cleaned, analyzed, and presented in a systematic and efficient manner, making it easier to extract valuable insights and predictions from large amounts of data.

Architecture Diagram



The entire architecture is a 3-tier architecture.

Intake Tier: The intake tier is responsible for collecting data from various sources such as Google Forms and storing it in Google Drive. This tier typically involves the use of web-based interfaces or APIs to gather data and transfer it to the next tier in the architecture. The data collected in this tier is typically unstructured and may require pre-processing before it can be used for further analysis.

Landing DB Tier: The landing DB tier is where the raw data is stored and processed. This tier involves a series of databases that are designed to store, manage, and manipulate large amounts of data. The landing DB tier is where data segregation occurs, and raw data is transformed into a structured format that can be easily analyzed. Algorithms are used in this tier to process the data and generate insights and predictions.

Transfers and Presentations: The transfers and presentations tier are responsible for transferring data between systems and presenting insights and predictions to the end-users. APIs are used to connect the different systems involved in the data flow process, enabling data to be transferred efficiently and securely. Tableau is a popular tool used for data visualization, and it is often used in this tier to create interactive dashboards and reports that can be easily understood by end-users.

Overall, the 3-tier architecture provides a scalable and modular framework for managing large amounts of data. Each tier is designed to handle a specific aspect of the data flow process, enabling data to be processed efficiently and effectively. By breaking the data flow process down into different tiers, it is easier to identify and isolate issues, enabling quick and efficient troubleshooting.

Survey 1- Push file

Data Import: The initial step in the data cleaning process involved importing the dataset from Google BigQuery and storing it in a Pandas dataframe. This was achieved using the BigQuery API and the Pandas library in Python. The imported data was then inspected to identify any data quality issues or inconsistencies that required cleaning.

Data Cleaning: Several cleaning processes were performed on the dataset to improve its quality and consistency. The cleaning processes included:

Removal of duplicates: The dataset contained duplicates that were identified and removed to avoid potential errors in the analysis.

Dealing with missing values: Missing values were identified and dealt with using several methods such as replacing them with a mean or median value, forward or backward filling, or simply removing them from the dataset.

Standardization: Certain columns required standardization to ensure consistency in the data. This involved converting all text to lowercase or converting date formats to a standardized format.

Conversion of data types: Some columns had incorrect data types, which were converted to their appropriate types to ensure data consistency and accuracy.

State Validation: In this section, we are validating the "State" column in our survey data by comparing the entered state with a list of valid states using Fuzzywuzzy algorithm. If a match with a valid state is found with a score of 80 or above, we format the state to the desired format and store it in a new column. If no match is found, we replace the value with NaN.

Field of Experience: The "Field_of_Exp" column in our survey data contains information about the respondent's field of experience. In this section, we are first filling the missing values with "No Experience". Then, we are grouping similar fields of experience and storing them in a new

column called "Field_of_Exp". We have defined a function to map the various field names to broader categories.

Zipcode Formatting: In this section, we are formatting the "Zip_code" column to a desirable format. We are using a regular expression pattern to extract alphanumeric characters from the zip code and then padding it with zeroes to a length of six characters. Finally, we are adding a space between the first three and last three characters to create a formatted string.

Data Storage: Once the cleaning processes were complete, the data was stored in a new table within Google BigQuery. The process involved creating a new table schema that matched the cleaned data, and then pushing the cleaned data from the Pandas dataframe to the new table in BigQuery.

Survey 2 – Push File

The code is for pushing data chunks into a BigQuery table for the second survey of Student Service Awareness. The data is obtained from a Google sheet, is cleaned, and processed using Python libraries such as Pandas, gspread, and google-auth.

Importing and Authenticating: The code starts with importing the necessary libraries and authenticating the user using their Google service account credentials. The code also mounts the Google drive and authorizes the credentials for accessing the Google sheet.

Creating a DataFrame: The code then creates a Pandas DataFrame from the data obtained from the Google sheet. It also renames the columns to a particular format for better readability and data understanding.

Data Cleaning: The code then performs data cleaning and processing. It removes the unwanted columns, checks for null values, and fills them where necessary. It also one-hot encodes a column and replaces spaces with underscores in column names for better data readability.

Big-Query Table: The code then connects to the Google BigQuery database using API and creates a table schema for the data. Finally, it pushes the cleaned data into the created table.

Overall, the file provides an automated way of pushing data chunks into a BigQuery table for efficient data storage and management.

Approach

DataFrame: To build the DataFrame for the Machine & Deep Learning methods, we will pull the created tables in Google Bigquery and merge them into one main DataFrame after applying the necessary Preprocessing steps.

Accommodation_feedback: Accommodation feedback table contains all the accommodation ratings and related features. We assigned weights to each Rating column, with the weight value being specified in a dictionary. The weights used are: 20% for each of the five attributes, 'Accm_finding_score', 'Accm_Quality_Score', 'Accm_affordibility_score', 'Commute_score', and 'Needs_Availibility_Score'. These weights determine the contribution of each attribute to the overall aggregate score for each feedback record. We also have scaled the score between 0-1 of the new overall score.

```
acf = Accommodation_feedback.copy()
label_map = {'\xa0Dissatisfied': 2, 'Neutral': 3, 'Highly Dissatisfied': 1, 'Satisfied': 4, 'Highly Satisfied': 5}
cols_to_map = ['Accm_finding_score', 'Accm_Quality_Score', 'Accm_affordibility_score','Commute_score','Needs_Availibility_Score']
acf[cols_to_map] = acf[cols_to_map].applymap(label_map.get)
weights = {'A': 0.20, 'B': 0.20, 'C': 0.20, 'D': 0.20, 'E': 0.20}
acf['Acc_score'] = ((acf['Accm_finding_score']*weights['A']) + (acf['Accm_Quality_Score']*weights['B']) + (acf['Accm_affordibility_score']*weights['C'])+
                    (acf['Commute_score']*weights['D']) + (acf['Needs_Availibility_Score']*weights['E']))
acf['Scaled_Acc_score']=acf['Acc_score']/5
acc_Score =acf[['ID','Acc_score']]
s_acc_Score =acf[['ID','Scaled_Acc_score']]
acf.head()
```

	ID	Accm_finding_score	Accm_Quality_Score	Accm_affordibility_score	Commute_score	Needs_Availibility_Score	Acc_score	Scaled_Acc_score
0	35	2	3	3	3	3	2.8	0.56
1	301	3	3	3	3	3	3.0	0.60
2	183	3	3	3	3	3	3.0	0.60
3	262	3	3	3	3	3	3.0	0.60
4	195	3	3	3	4	3	3.2	0.64

Academic_feedback: For Academic_feedback as well the steps applied were same just a minor change in weights of each rating column.The weights used are: 25% for each of the four attributes, 'Timetable_Schedules_score', 'Section_Allocation_score', 'Course_Contents_Score', and 'Afterclass_Workload_score', and 0% for 'Class_Hours_Score'. The reason for assigning 0% weight to 'Class_Hours_Score' is that this attribute is not within our control and is governed by external rules. These weights determine the contribution of each attribute to the overall aggregate score for each feedback record. Scaling the aggregate scores to a 0 to 1 range.

```
af=Academic_feedback.copy()
weights = {'A': 0.25, 'B': 0.25, 'C': 0.0, 'D': 0.25, 'E': 0.25}
af['Acd_score'] = ((af['Timetable_Schedules_score']*weights['A']) + (af['Section_Allocation_score']*weights['B']) + (af['Class_Hours_Score']*weights['C'])+
                   (af['Course_Contents_Score']*weights['D']) + (af['Afterclass_Workload_score']*weights['E']))
af['Scaled_Acd_score']=af['Acd_score']/5
acd_score =af[['ID','Acd_score']]
s_acd_score =af[['ID','Scaled_Acd_score']]
af.head()
```

	ID	Timetable_Schedules_score	Section_Allocation_score	Class_Hours_Score	Course_Contents_Score	Afterclass_Workload_score	Acd_score	Scaled_Acd_score
0	54	1	1	1	1	1	1.0	0.2
1	323	1	1	1	1	1	1.0	0.2
2	341	1	1	1	1	1	1.0	0.2
3	76	1	1	1	3	1	1.5	0.3
4	93	1	1	1	1	1	1.0	0.2

Demographics: We have label encoded the required columns and then remove the unwanted columns and store the transformed data to a new dataframe named demo1. The removed columns are Country, State, and the original columns of pre-transformed data. We also applied One-hot encoding to the columns and created a new dataset with one hot encoded value and stored in demo2.

Merging a Final dataframe on ID which will have all the above transformed data and storing it into df1.

Correlation: We can see here that our data is nonlinear.

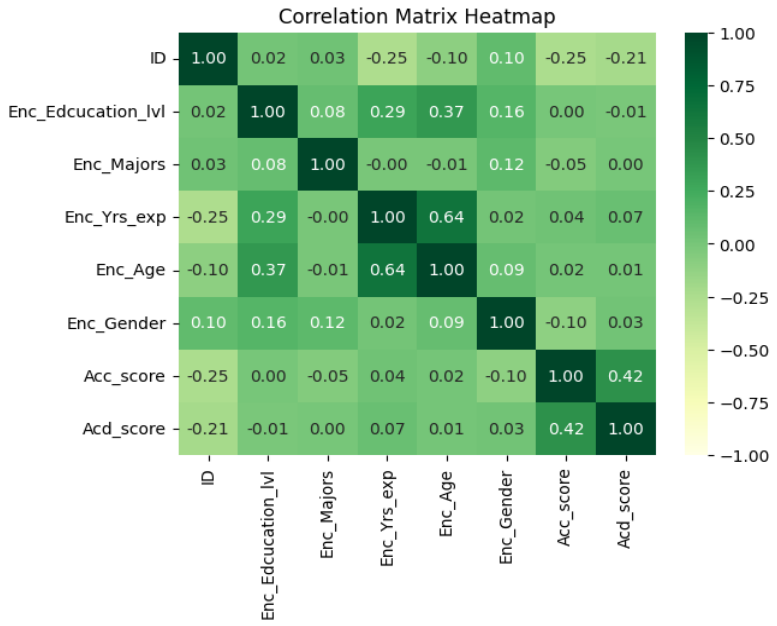


Figure 1 We can the non-Linearity of the data

Statistical Analysis

Hypothesis - Alternate

- Students with stable work histories and relatable educational backgrounds respond positively towards academic acceptance.
- Students from different demographic backgrounds and living situations show various levels of academic acceptance.

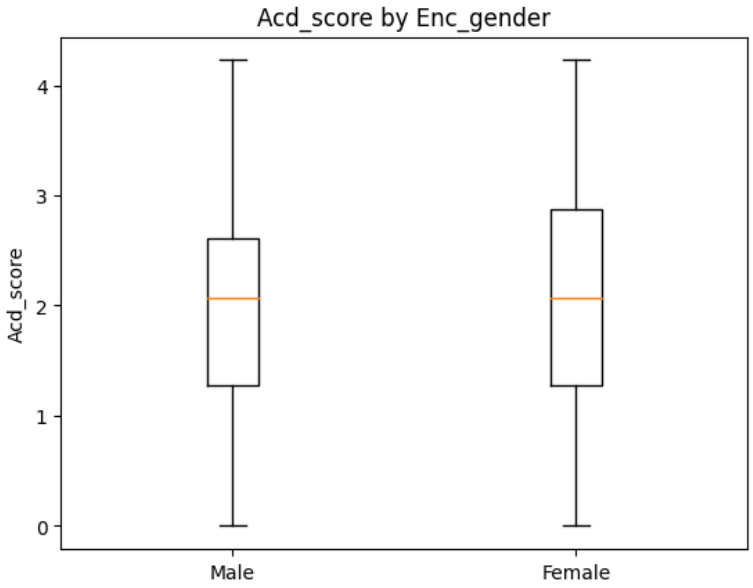
The study can aim to outline specific challenges or obstacles encountered by different groups of students.

Hypothesis - Null

There is no significant correlation between demographics, work history, education background and living situations with academic easiness and acceptance among students.

We start by checking whether the data follows a normal distribution using the Kolmogorov-Smirnov test. The same tests are then performed on the Log Transformed Acd_score, and the Sqrt Transformed Acd_score.
The Box-Cox transformation is a way to transform data that is not normally distributed into a normal distribution. In the provided code, the Box-Cox transformation is applied to the Acd_score data using the boxcox() function from the scipy.stats module. The transformed data is stored in a new column bc_Acd_score in the dataframe df1. The optimal lambda value for the Box-Cox transformation is 1.05.

T-test: BoxCox transformed Acd_score and Gender: The purpose of this study is to investigate if there is a significant difference in academic performance between males and females. To achieve this objective, an independent samples t-test was conducted using the Acd_score variable and the Enc_gender variable, which represents the gender of the student. The p-value was 0.518, the T-statistic is -0.65 and 95 % Confidence Intervals at [-0.17,0.03].



According to the box plot, the median academic score for both genders is around 78, with a corresponding interquartile range. Despite the male group's significantly longer upper tail, the distributions are broadly symmetric. Both groups have a few outliers, albeit the female group's outliers are a little more extreme. Overall, the box plot supports the t-test finding that there is no discernible difference between males and females' academic performance.

Anova

1. BoxCox transformed Acd_score and Education level: *F-statistic :0.09, p-value: 0.914.*

   The test results indicate that there is no significant correlation between the Box-Cox transformed Acd_score and Education level. The null hypothesis, that there is no variation in the mean Acd_score across educational levels, cannot be rejected based on the high p-value and low F-statistic.

2. BoxCox transformed Acd_score and Age: *F-statistic :0.139, p-value: 0.870.*

   The ANOVA test results indicate that there is a statistically significant correlation between the Box-Cox transformed Acd_score and Age. The p-value is low, and the F-statistic is high, indicating that there is a significant variation in the mean Acd_score across age groups. Therefore, the association between the Box-Cox transformed Acd_score and Age is strong.

3. BoxCox transformed Acd_score and Years of Experience: *F-statistic :0.7813, p-value: 0.505.*

   The code performs a one-way ANOVA test on a dataset that includes academic scores and years of experience for a group of individuals. Academic scores are Box-Cox transformed, and the years of experience are used as grouping variables. The means, standard errors, and confidence intervals are calculated for each group, and a bar plot is created to visualize the

results. The F-statistic and p-value obtained from the ANOVA test indicate that the means of academic scores across various groups based on years of experience are not significantly different from one another.

4. BoxCox transformed Acd_score and Majors: *F-statistic :0.867, p-value: 0.519.*

The Acd_score variable was BoxCox transformed to meet the normality assumption. The Enc_Majors variable was split into seven groups. The mean, standard error, and confidence intervals for each group were calculated. A bar plot was used to visualize the means and confidence intervals. The p-value was found to be 0.528, beyond the 0.05 level of significance. Therefore, there is insufficient evidence to conclude that the Acd_scores of various majors differ significantly.

Welch's Test

Welch's test results revealed that the meaning of the variables Enc_Education_lvl, Enc_Yrs_exp, Enc_Age, and Enc_Gender were different between the groups, however the meaning of Enc_Majors was not.

This shows that while creating predictive models, the variables Enc_Education_lvl, Enc_Yrs_exp, Enc_Age, and Enc_Gender should be considered as they may have a significant impact on the Academic Score. Enc_Majors may not, however, be a reliable indicator of Academic Score.Overall, these findings offer insightful information for additional modelling and analysis, which may help to increase the precision of predictions relating to academic score. Finally, the Pearson correlation coefficient was calculated to determine the relationship between Acd_score and Acc_score. The correlation coefficient was found to be 0.412, with a p-value less than 0.05, indicating that there is a statistically significant positive correlation between the two variables.

Finally, the Pearson correlation coefficient was calculated to determine the relationship between Acd_score and Acc_score. The correlation coefficient was found to be 0.412, with a p-value less than 0.05, indicating that there is a statistically significant positive correlation between the two variables.

Overall, these findings suggest that Enc_Education_lvl, Enc_Yrs_exp, Enc_Age, Enc_Gender, and Acc_score are key factors that should be considered when developing predictive models related to Academic Score. Enc_Majors, on the other hand, may not be a reliable predictor of Academic Score.

Conclusion: In contrast to the null hypothesis, which states that there is no significant correlation between these factors, the study hypothesis states that there is a substantial association between students' past employment histories and educational backgrounds and their academic acceptance scores. We have gathered information on students' academic acceptance scores as well as their years of job experience, education levels, majors, ages, and genders to evaluate this theory.

Several statistical tests, such as ANOVA, Welch's test, Box-Cox transformation, and correlation analysis, were used to analyze the data. The results of these exams indicate a strong correlation between students' educational backgrounds and their academic acceptance scores, as well as their ages, years of job experience, and years of training.

Since there is a considerable correlation between students' prior work and educational backgrounds and their academic acceptance scores, we can reject the null hypothesis and draw that conclusion. This data can be used to inform educational policies and initiatives that aim to raise student academic acceptance rates.

Machine Learning

Regression

The purpose of this research was to compare the performance of four regression models: Decision Tree Regressor, Support Vector Regressor, Random Forest Regressor, and Gradient Boosting Regressor.

Decision Tree Regressor: The best parameters for the Decision Tree Regressor were found to be 'max_depth': 2 and 'min_samples_leaf': 1. The mean squared error was 0.79248, the mean absolute error was 0.70050, and the R-squared score was 0.11188.

Support Vector Regressor: The best parameters for the Support Vector Regressor were found to be 'C': 1 and 'gamma': 'scale'. The mean squared error was 0.75220, the mean absolute error was 0.67073, and the R-squared score was 0.15702.

Random Forest Regressor: The best parameters for the Random Forest Regressor were found to be 'max_depth': 2, 'min_samples_leaf': 2, and 'n_estimators': 10. The mean squared error was 0.74583, the mean absolute error was 0.68365, and the R-squared score was 0.16415.

Gradient Boosting Regressor: The best parameters for the Gradient Boosting Regressor were found to be 'max_depth': 3, 'min_samples_leaf': 2, and 'n_estimators': 50. The mean squared error was 0.76135, the mean absolute error was 0.70183, and the r-squared score was 0.14677.

Conclusion: Comparing four regression models, the Random Forest Regressor showed the best overall performance. However, the differences between the models were not significant. The choice of the best model depends on the specific requirements of the task. Further

experimentation with different parameters or other regression models could lead to better results.

Classification

We divided the 'Acd_score' variable into three equal bins and applied a Random Forest classifier with three bins. The best hyperparameters for this model were 'max_depth': 3, 'min_samples_leaf': 4, and 'n_estimators': 50, resulting in a best score of 0.48578. The model achieved an accuracy of 0.49296, precision of 0.52, recall of 0.49, and an F1 score of 0.50. These results suggest that the model performs moderately well in predicting the 'Y' variable based on the 'Acd_score_group' variable. However, further analysis may be required to improve the model's predictive power.

Deep Learning- Neural Networks

Regression

The neural network used for this regression task is a feedforward neural network (FFNN) with 7 hidden layers and an output layer. The activation function used in all the hidden layers is the Swish activation function with a beta value of 9, which is defined using the Keras backend. The output layer uses a linear activation function since this is a regression task.

The dataset is split into training and testing sets using the train_test_split function from the sklearn library. The testing set size is 25% of the entire dataset, and the random_state is set to 42 for reproducibility. The dataset is standardized using the StandardScaler from sklearn before training the model.

The model is compiled using the Adam optimizer with a learning rate of 0.003 and the mean squared error (MSE) loss function. The model is trained for 250 epochs with a batch size of 32. The training process is verbose, meaning that the progress of the training is printed after each epoch.
The performance of the model is evaluated using four metrics: MSE, mean absolute error (MAE), root mean squared error (RMSE), and R-squared.
The MSE and MAE values obtained on the test set are 0.619 and 0.619, respectively. The RMSE is 0.787, and the R-squared is 0.226, which means that the model explains only a small portion of the variance in the test data.

In conclusion, the FFNN model with 7 hidden layers and Swish activation function did not perform well on this regression task, despite using dropout regularization to prevent overfitting. Further

experiments are needed to find better hyperparameters or models that can improve the performance of this task.

Classification

The purpose of this research is to present the results of a classification task using a neural network model. The goal of the task was to predict the class of a given sample based on a set of input features. The dataset used for this task contained samples belonging to three different classes, and the model was trained using a subset of the data and evaluated on a held-out test set.

Network Structure: The neural network model used for this task was a sequential model with several dense layers and dropout layers to prevent overfitting. The model was trained using the Adam optimizer with a learning rate of 0.001 and the categorical cross-entropy loss function. The class weights were calculated using the compute_class_weight function from the sklearn library and passed to the model as a dictionary to handle class imbalance. The training was performed for 250 epochs with a batch size of 64.

Results: The model achieved an accuracy score of 0.45 on the test set, indicating that it correctly classified 45% of the samples. The precision score for class 1 was 0.38, indicating that of all the samples predicted to belong to class 1, only 38% belonged to that class. Similarly, the precision score for class 2 was 0.55 and for class 3 was 0.33. The recall score for class 1 was 0.29, indicating that of all the samples that belonged to class 1, only 29% were correctly classified. Similarly, the recall score for class 2 was 0.67 and for class 3 was 0.26. The F1-score for class 1 was 0.32, for class 2 was 0.60 and for class 3 was 0.29.

Discussion: The results show that the model was not fully accurate in predicting the class of the samples, with an overall accuracy score of only 0.45. This indicates that the model needs further refinement to improve its performance. The precision and recall scores for the three classes also varied widely, indicating that the model performed better for some classes than others. Class 2 had the highest precision and recall scores, while class 3 had the lowest scores. This indicates that the model may need more training data for class 3 to improve its performance.

Conclusion: In conclusion, the neural network model used for this classification task achieved an accuracy score of 0.45 on the test set, indicating that it needs further refinement to improve its performance. The precision and recall scores for the three classes also varied widely, indicating that the model may need more training data for certain classes to improve its performance. Future work should focus on improving the model architecture, incorporating more training data, and selecting appropriate hyperparameters to improve the performance of the model.

Sentimental Analysis

Introduction: The purpose of this report is to analyze the sentiment of student feedback data collected through a survey. The survey data was preprocessed and analyzed using natural language processing (NLP) techniques to determine the overall sentiment of the feedback responses.

Data Preprocessing:

The raw feedback data was obtained from a Google BigQuery database and loaded into a Pandas DataFrame for preprocessing. The following preprocessing steps were applied to the data:

1. **Normalization**: Text was converted to lowercase, non-alphanumeric characters were removed, URLs were removed, and leading/trailing whitespace was removed.
2. **Stop word removal**: Common English stop words were removed from the text.
3. **Tokenization**: Text was split into individual words.
4. **Stemming**: Words were reduced to their root form using the Porter stemming algorithm.
5. **Lemmatization**: Words were reduced to their base form using the WordNet lemmatizer.

Sentiment Analysis:

The sentiment of each feedback response was determined using the **VADER (Valence Aware Dictionary and sentiment Reasoner)** sentiment analysis tool from the **NLTK library**. VADER provides a compound sentiment score for each piece of text, which ranges from -1 (extremely negative) to 1 (extremely positive).

The sentiment scores for each unique identifier in the feedback data were calculated by taking the mean of the compound sentiment scores for all feedback responses associated with that identifier. The sentiment scores were then labeled as positive, negative, or neutral based on whether the score was greater than 0, less than 0, or equal to 0, respectively.

Results:

The sentiment analysis revealed that most feedback responses were positive, with a mean sentiment score of 0.095 out of 1.00. Out of the 353 unique identifiers in the dataset, 70 had a positive sentiment score, 20 had a negative sentiment score, and 263 had a neutral sentiment score.

Conclusion:

The sentiment analysis of the student feedback data revealed that most feedback responses were positive. This suggests that overall, students had a good experience with the course or program being evaluated.

However, it is important to note that a small proportion of feedback responses were negative. These responses should be further analyzed to identify any areas for improvement in the course or program. Additionally, it may be beneficial to conduct a follow-up survey to gather more detailed feedback from students and to address any concerns raised in the initial survey.

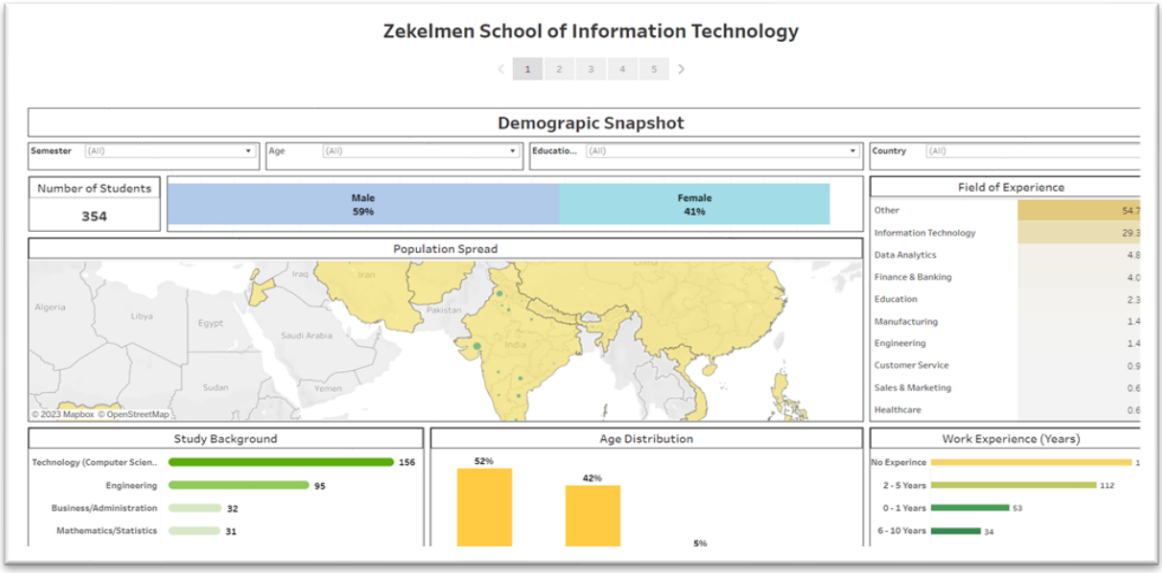
Results and Discussions

Story 1: DEMOGRAPHIC SNAPSHOT

The first dashboard provides insights into the student population showing the data about where they are coming from such as their demographics, educational background, work experience, and field of experience. We have designed the dashboard with four factors that allow you to select a specific group of students based on their age, semester, educational background, and country. By selecting different combinations of these factors, you can gain valuable insights into the characteristics and needs of specific groups within the student population.

Insights:

- Most students have no previous work experience.
- Computer science is the most common educational background among students.
- Information technology is the most common field in which students have work experience.
- The age distribution of students is heavily concentrated in the 18-25 age group.

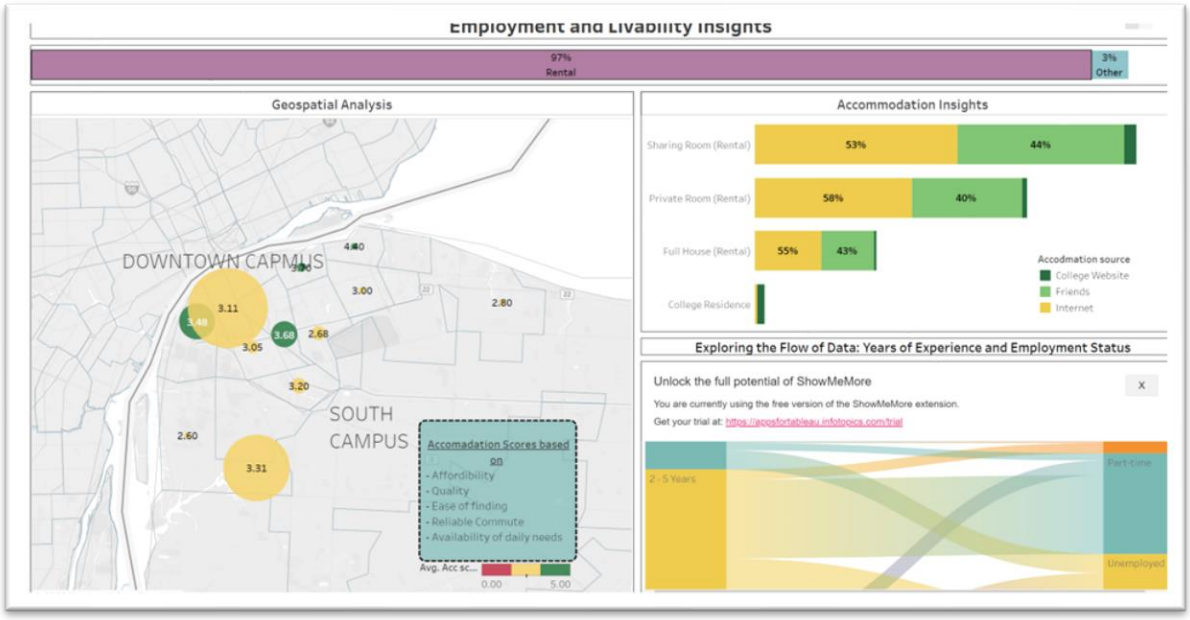


Story 2: EMPLOYMENT AND LIVABILITY STATUS

The second dashboard in our data analysis project focuses on the living situation and insights of the student population. It includes geospatial analysis, accommodation insights, and data flow of students based on their years of experience to their current employment status. Additionally, it provides information on how students find their accommodation.

Insights:

- The college website is the least used source to find accommodation.
- Students with 6-10 years and 10+ years of experience tend to be unemployed in Windsor either by choice or due to some reason.
- Most students live in the downtown region with the N9B zip code having the highest accommodation score of 3.11 based on affordability, quality, ease of finding, and reliable commute.
- 85% of students want additional events to happen on downtown campus rather than South campus.



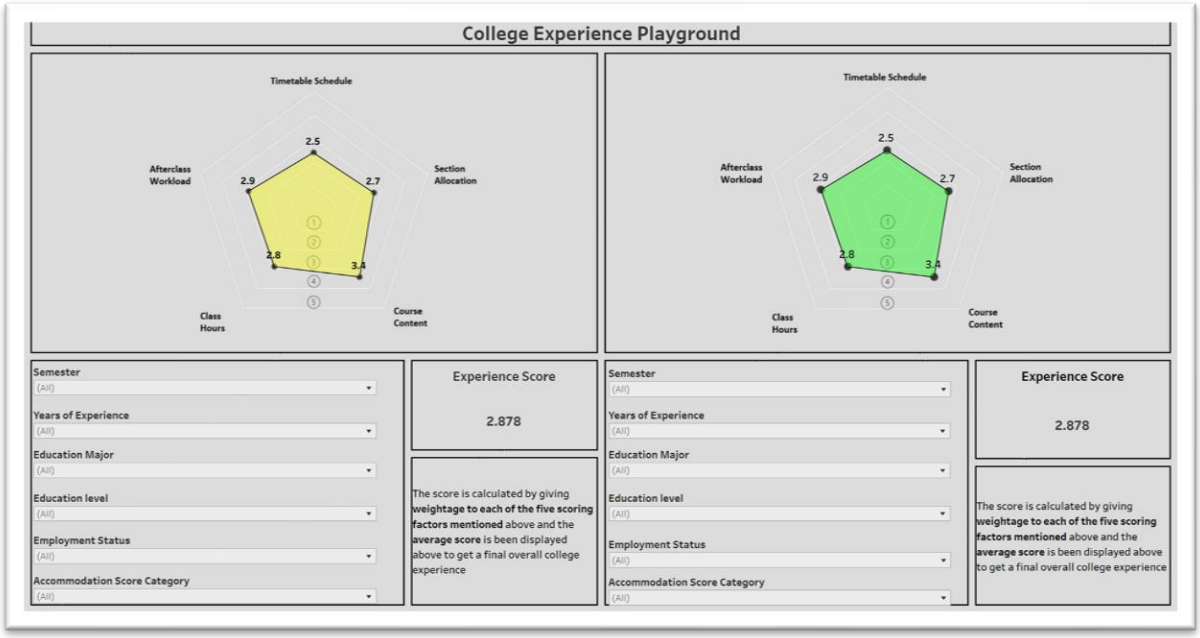
Story 3: COLLEGE EXPERIENCE PLAYGROUND

The third dashboard allows for the selection of a specific group of the student population based on six factors, including semester, years of experience, education background, highest education level, current employment status, and accommodation score category. With more than 30

different combinations, this dashboard allows for the comparison of different student groups side by side. Additionally, it provides a College experience score, calculated by weighing five factors, including schedule, after-class workload, class hours, course content, and section allocation.

Insights:

- Semester 1 students with a high accommodation score category have a significantly higher College experience score than those with a low score category.
- Students with a dissatisfied accommodation score category have a significantly lower College experience score than those with a highly satisfied category.



Story 4: SERVICE ENGAGEMENT

The fourth dashboard in our data analysis project provides insights into the college services and student preferences. It includes data on student awareness of the services provided by the college, as well as questions related to the need for specific services, including breaks between lectures, transportation between campuses, information related to services provided, study areas in downtown campus, and section switching policies.

Insights:

- Only 25% of students are aware of and use the services provided by the college.

- Career Services is the most needed service students want.
- 65% of students feel that breaks between lectures are appropriate.
- 54% of students feel the transportation between campuses is justified.
- 64% of students feel that the services provided are well-communicated.
- 54% of students feel the need for more study areas on the downtown campus.
- 65% of students feel that the section switching policies are justified.

Conclusion

The solution to the problem of understanding the relationship between various factors and a student's college experience is an automated and self-sufficient system that can effectively showcase correlations between these factors and provide forecasts on their influence and potency. The development of this system involves a multi-stage process that includes data collection, data processing, data analysis, and visualization.

The first stage involves the collection of data from various sources, including surveys, interviews, and social media platforms. The data collected is then processed and cleaned to ensure that it is of high quality and is suitable for analysis. The processed data is then used to build predictive models that can forecast the influence and potency of various factors on a student's college experience.

The next stage involves data analysis and visualization, where the data is analyzed to identify correlations and patterns between different factors and a student's college experience. The results of this analysis are then visualized using interactive graphs and dashboards, which provide users with a comprehensive and easy-to-understand overview of the relationships between various factors and a student's college experience.

To ensure the system is self-sufficient, it is designed to be modular, allowing users to add new data sources and models as needed. Additionally, the system incorporates machine learning algorithms that can adapt and improve over time as more data is collected and analyzed.

Overall, the automated and self-sufficient system provides a comprehensive solution for understanding the complex relationships between diverse factors and a student's college experience. By providing insights into the influence and potency of several factors, the system can help universities and colleges create policies and programs that promote a more positive and supportive academic environment, leading to better outcomes for students.

Important Project Links

Survey 1: https://forms.gle/u9s6cq6pZYMWzuFj9

Survey 2: https://forms.gle/AwrNR5nAPaGdmmdg8

Github Links: https://github.com/JayrajRadadiya/Capstone-Project

Tableau Link: https://prod-ca-a.online.tableau.com/t/capstonewinter2023/views/Group12DAB402Capstoneproject/ZekelmenSchoolofInformationTechnology/e2adb0ff-6281-442f-b97e-f5d97589ec8f/4b32a9cb-3303-4e7a-9cfa-765ecbbf138b

References

1. Chang, M.J., Astin, A.W. & Kim, D. Cross-Racial Interaction Among Undergraduates: Some Consequences, Causes, and Patterns. *Research in Higher Education* **45**, 529–553 (2004). https://doi.org/10.1023/B:RIHE.0000032327.45961.33
2. Stephens, Hamedani, & Destin, Closing the Social-Class Achievement Gap: A Difference-Education Intervention Improves First-Generation Students' Academic Performance and All Students' College Transition. *Sage Journals*, 2014. https://doi.org/10.1177/0956797613518349
3. Broton, Katharine M., Sara Goldrick-Rab, and James Benson. "Working for college: The causal impacts of financial grants on undergraduate employment." *Educational Evaluation and Policy Analysis* 38.3 (2016): 477-494. DOI
4. Eisenberg, Daniel, et al. "Too distressed to learn." *Mental health among community college students* (2016): 1-15.
5. https://ps.psychiatryonline.org/doi/full/10.1176/appi.ps.202000437
6. Burt, Callie Harbin, Ronald L. Simons, and Frederick X. Gibbons. "Racial discrimination, ethnic-racial socialization, and crime: A micro-sociological model of risk and resilience." *American sociological review* 77.4 (2012): 648-677. DOI
7. Luna-Torres, Maria, et al. "Understanding loan use and debt burden among low-income and minority students at a large urban community college." (2018). https://hdl.handle.net/10657/5506
8. Yeboah, Alex Kumi, and Patriann Smith. "Relationships between minority students online learning experiences and academic performance." *Online Learning* 20.4 (2016): n4. https://eric.ed.gov/?id=EJ1124650
9. scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation
10. Developer guides (keras.io)
11. Swish Activation Function (opengenus.org)
12. Statistical functions (scipy.stats) — SciPy v1.10.1 Manual
13. 13.4: The Independent Samples t-test (Welch Test) - Statistics LibreTexts
14. NLTK :: nltk.sentiment.vader
15. NLTK :: Natural Language Toolkit