FINAL REPORT

A Comprehensive Analysis of Logistic Regression:
Methods and Applications

Vo Nguyen Minh Duy - 17105

Nguyen Van Khanh - 17096

Tran Quang Minh - 17061

Nguyen Khoa - 14820

25.11.2023

1. Introduction

- **Purpose**: discuss how Logistic Regression can be effectively applied to high-dimensional data, addressing its challenges and demonstrating our research methodology
- **Background:** Logistic Regression was developed by statistician David Cox in 1958. However, the basic concept was formulated earlier in the 19th century in the context of modeling population growth. It is based on the concept of probability and utilizes the logistic function to model the probability of a certain class or event. This function is an S-shaped curve that can take any real-valued number and map it between 0 and 1, but not exactly at those limits.
- **Logistic Regression plays a significant role in high-dimensional data analysis:** particularly in fields where binary outcomes are of interest. High-dimensional data refers to datasets with a large number of variables (features), often much larger than the number of observations. This scenario is common in modern applications like genomics, text classification, and image recognition
- **Example**:
  Genomics and Bioinformatics: In genomics, logistic regression is used to identify the relationship between genetic variants (like SNPs - Single Nucleotide Polymorphisms) and binary traits (like the presence or absence of a disease). The high dimensionality comes from the large number of genetic markers.
  Financial Fraud Detection: Logistic regression can be employed to detect fraudulent activities in finance, such as identifying fraudulent credit card transactions. The high-dimensional data includes numerous transaction attributes.
  Image Recognition: While more complex models like neural networks are often preferred for image recognition, logistic regression can be used for simple binary classification tasks in images (e.g., distinguishing between two types of objects). The high dimensionality arises ─from the large number of pixels in each image.

2. Methodology

- Logistic Regression is used when the dependent variable is binary. Unlike linear regression, which predicts a continuous outcome, logistic regression predicts the probability of an outcome occurring (e.g., yes/no, success/failure).
- **Mathematical formulation:**

$$P(Y = 1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\ldots\beta_k x_k)}}$$

  Where:
  P(Y=1) is the probability of the dependent variable equaling a case$\beta_1, \beta_2, \ldots \beta_k$ are the coefficients

$x_0, \ldots x_k$ are the independent variables

- **Assumptions and limitations:**
  Binary Outcome: The dependent variable should be binary.
  Linearity: Assumes a linear relationship between the independent variables and the logit of the dependent variable.
  No Multicollinearity: Independent variables shouldn't be too highly correlated.
  Large Sample Size: Performs better with a larger sample size.
  Independence of Observations: Assumes that the observations are independent of each other.

- **Data Preprocessing:**
  Handling Missing Values: Impute or remove missing values.
  Feature Scaling: While not always necessary, feature scaling can be beneficial, especially for convergence in optimization algorithms.
  Encoding Categorical Variables: Convert categorical variables into a format that can be provided to the model (using R).
- **Model Building**:
  Variable selection:
  Expert Judgment: Based on domain knowledge.
  Statistical Tests: Like Chi-square tests for categorical variables.
  Regularization Techniques: Like LASSO for automated feature selection.
  Model fitting:
  Splitting Data: Divide the data into training and testing sets.
  Training the Model: Fit the logistic regression model on the training set.
- **Model evaluation:**
  Confusion Matrix: To evaluate the number of correct and incorrect predictions.
  Accuracy: The proportion of correctly predicted observations.
  Precision and Recall: Especially important in imbalanced datasets.
  ROC Curve and AUC: To assess the model's ability to distinguish between the classes.
  Cross-Validation: For assessing the model's performance on unseen data.

3. Strengths and Weaknesses of Logistic Regression

- **Advantage of logistic regression:**
  Simplicity and Interpretability: Logistic Regression models are straightforward and easy to interpret, making them a good starting point for binary classification problems.
  Probabilistic Approach: It provides probabilities for outcomes, which can be a more informative way of looking at results compared to simply predicting classes directly.
  Good Performance with Small Datasets: It can perform well with a smaller number of observations, unlike some complex models that require large amounts of data.

Robust to Noise: Logistic Regression can be less sensitive to small noise in the data compared to more complex models.
Efficient and Scalable: It is computationally less intensive, making it efficient to train, even with a large number of features.
Handles Non-linear Effects: Through transformations and interactions, logistic regression can model non-linear effects.

- **Disadvantages of logistic regression:**

Assumption of Linearity: It assumes a linear relationship between the independent variables and the logit of the dependent variable, which is not always the case in real-world scenarios.
Difficulty with Complex Relationships: Logistic Regression may not perform well with complex relationships between variables, where more sophisticated models might excel.
Not Suitable for a Large Number of Categorical Features: When dealing with data with a large number of categorical features, especially after one-hot encoding, logistic regression might become less efficient.
Sensitive to Imbalanced Data: In cases of highly imbalanced datasets, logistic regression might not perform well without proper handling of the imbalance.
Multicollinearity: Performance can be impacted if there is high multicollinearity among the independent variables.

4. Practical Implementation and Results

4.1 Data Collection Process

Data collection is the process of collecting and evaluating information or data from multiple sources to find answers to research problems, answer questions, evaluate outcomes, and forecast trends and probabilities. It is an essential phase in all types of research, analysis, and decision-making, including that done in the social sciences, business, and healthcare.

- First step, data is needed for the Calculation so a survey is conducted with some main properties:

+ The data collected must be clear and easy to work on.
+ A large number of participants is also needed as the more data the better.
+ A set of variables is required and the questions should be based on them.
+ An essential part like a research question can’t be overlooked.

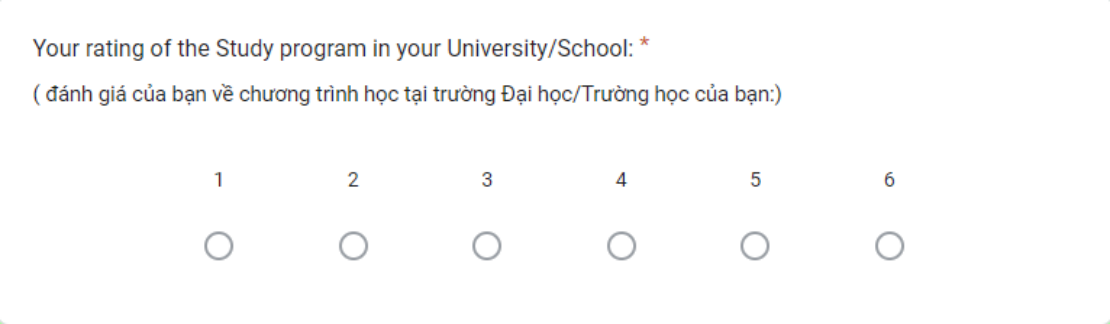
=> After some Discussion, we see that a rating survey type would be the best for this scenario as it satisfies all the requirements listed above

- Second step, the details of our survey:

+ The topic of the survey is the satisfaction level of students at the University
+ Variables which the questions in the survey will base on:

- Sex?
- Age?
- Year?
- Study program rating?
- Facilities rating?
- Club activities rating?
- Study environment rating?
- Opportunity rating?(scholarship/internship)
- Faculty rating?
- Overall rating?

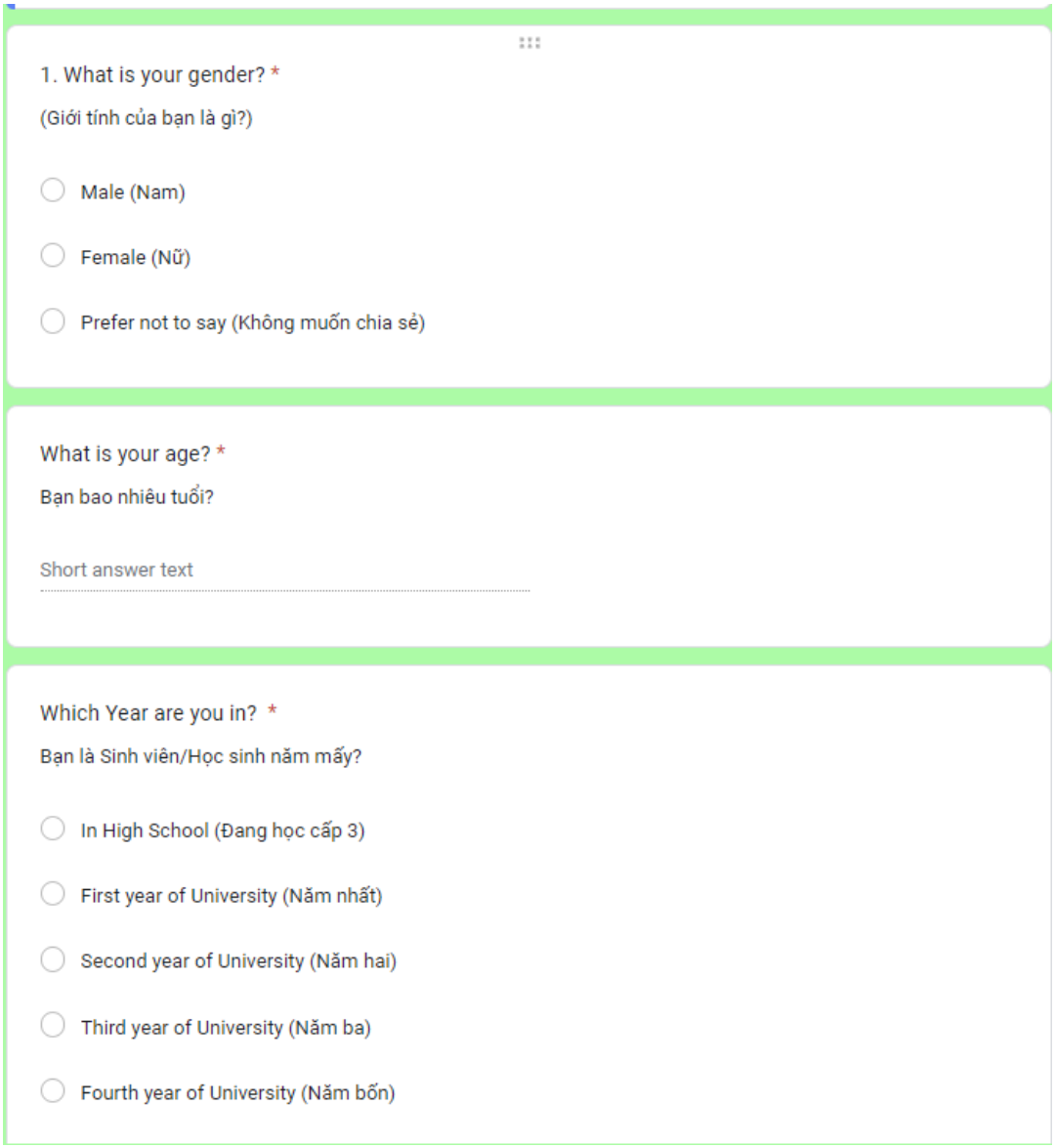
+ And Research questions for this project:
  => What is the overall satisfaction level of students about many aspects of their university or school that they are studying?
  => Which Aspect satisfies most of the students?
  => Which Aspect is not satisfied by most of the students?
  => Can we apply this analysis method to other rating survey data?
+ The Rating system is based on number levels " *The higher the number, the greater the Satisfaction"*. For example:

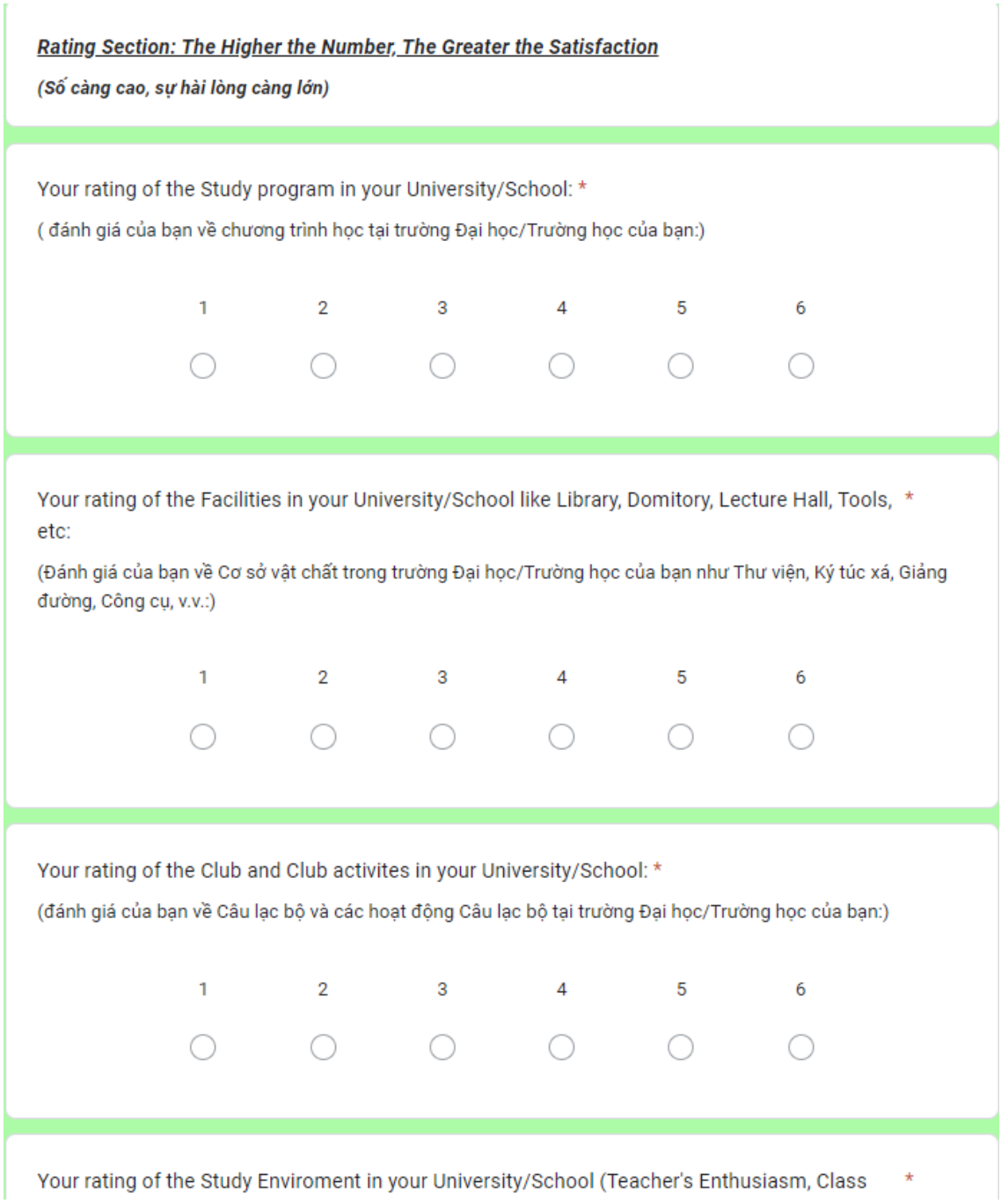


This is a simple question taken directly from our final survey. The Rating system is self-explained here, the bigger the number the more students are satisfied with 1 being the smallest and 6 being the greatest.

- The third step is to deliver this survey to our group of choice which is mostly *university students.* In this case, *Google Forms* is proven to be the best tool as it gives an output already in .csv file format with a clear view of all variables and their results.

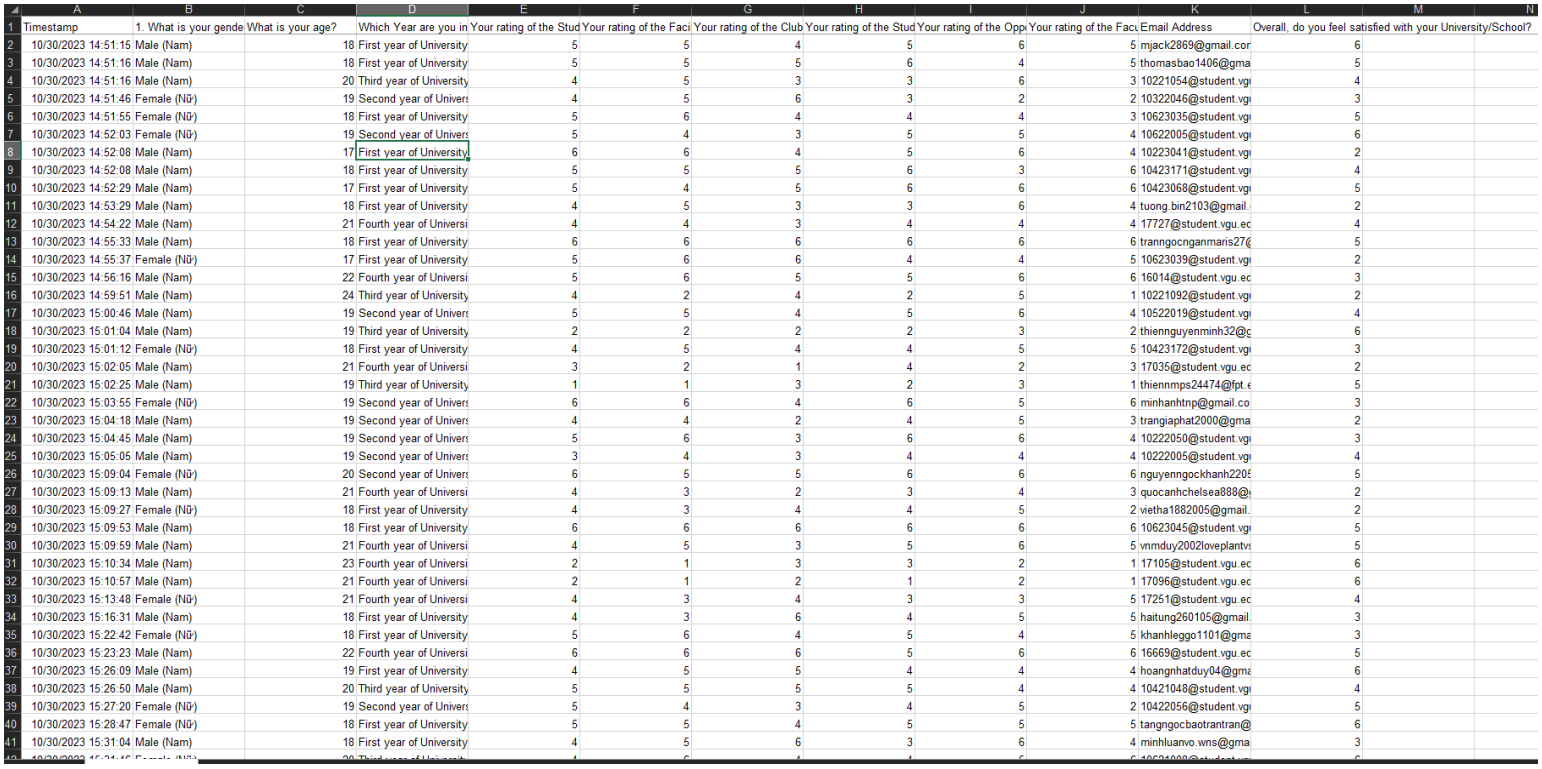
Here are some looks at our final survey:





The actual link for the final survey is provided in section 6.Reference

- Fourth step, after gathering enough responses which in our case is *128 participants.* The data assembled will need to be reviewed before taking part in the calculation stage, this will be performed in the data preparation stage. Here is a glance at the results before any cleanup is done:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Timestamp	1. What is your gende	What is your age?	Which Year are you in	Your rating of the Stud	Your rating of the Faci	Your rating of the Club	Your rating of the Stud	Your rating of the Opp	Your rating of the Facu	Email Address	Overall, do you feel satisfied with your University/School?		
2	10/30/2023 14:51:15	Male (Nam)	18	First year of University	5	5	4	5	6	5	mjack2869@gmail.cor	6		
3	10/30/2023 14:51:16	Male (Nam)	18	First year of University	5	5	5	6	4	5	thomasbao1406@gma	5		
4	10/30/2023 14:51:16	Male (Nam)	20	Third year of University	4	5	3	3	6	3	10221054@student.vg	4		
5	10/30/2023 14:51:46	Male (Nam)	19	Second year of Univers	4	5	6	3	2	2	10322046@student.vg	3		
6	10/30/2023 14:51:55	Female (Nữ)	18	First year of University	5	6	4	4	4	3	10623035@student.vg	5		
7	10/30/2023 14:52:03	Female (Nữ)	19	Second year of Univers	5	4	3	5	5	4	10622005@student.vg	6		
8	10/30/2023 14:52:08	Male (Nam)	17	First year of University	6	6	4	5	6	4	10223041@student.vg	2		
9	10/30/2023 14:52:08	Male (Nam)	18	First year of University	5	5	5	6	3	6	10423171@student.vg	4		
10	10/30/2023 14:52:29	Male (Nam)	17	First year of University	5	4	5	6	6	6	10423068@student.vg	5		
11	10/30/2023 14:53:29	Male (Nam)	18	First year of University	4	5	3	3	6	4	tuong.bin2103@gmail.	2		
12	10/30/2023 14:54:22	Male (Nam)	21	Fourth year of Universi	4	4	3	4	4	4	17727@student.vgu.ec	4		
13	10/30/2023 14:55:33	Male (Nam)	18	First year of University	6	6	6	6	6	6	tranngocnganmaris27@	5		
14	10/30/2023 14:55:37	Female (Nữ)	17	First year of University	5	6	6	4	4	5	10623039@student.vg	2		
15	10/30/2023 14:56:16	Male (Nam)	22	Fourth year of Universi	5	6	5	5	6	6	16014@student.vgu.ec	3		
16	10/30/2023 14:59:51	Male (Nam)	24	Third year of University	4	2	4	2	5	1	10221092@student.vg	2		
17	10/30/2023 15:00:46	Male (Nam)	19	Second year of Univers	5	5	4	5	6	4	10522019@student.vg	4		
18	10/30/2023 15:01:04	Male (Nam)	19	Third year of University	2	2	2	2	3	2	thiennguyenminh32@g	6		
19	10/30/2023 15:01:12	Female (Nữ)	18	First year of University	4	5	4	4	5	5	10423172@student.vg	3		
20	10/30/2023 15:02:05	Male (Nam)	21	Fourth year of Universi	3	2	1	4	2	3	17035@student.vgu.ec	2		
21	10/30/2023 15:02:25	Male (Nam)	19	Third year of University	1	1	3	2	3	1	thiennmps24474@fpt.e	5		
22	10/30/2023 15:03:55	Female (Nữ)	19	Second year of Univers	6	6	4	6	5	6	minhanhtnp@gmail.co	3		
23	10/30/2023 15:04:18	Male (Nam)	19	Second year of Univers	4	4	2	4	5	3	trangiaphat2000@gma	2		
24	10/30/2023 15:04:45	Male (Nam)	19	Second year of Univers	5	6	3	6	6	4	10222050@student.vg	3		
25	10/30/2023 15:05:05	Male (Nam)	19	Second year of Univers	3	4	3	4	4	4	10222005@student.vg	4		
26	10/30/2023 15:09:04	Female (Nữ)	20	Second year of Univers	6	5	5	6	6	6	nguyenngockhanh2205	5		
27	10/30/2023 15:09:13	Male (Nam)	21	Fourth year of Universi	4	3	2	3	4	3	quocanhchelsea888@g	2		
28	10/30/2023 15:09:27	Female (Nữ)	18	First year of University	4	3	4	4	5	2	vietha1882005@gmail.	2		
29	10/30/2023 15:09:53	Male (Nam)	18	First year of University	6	6	6	6	6	6	10623045@student.vg	5		
30	10/30/2023 15:09:59	Male (Nam)	21	Fourth year of Universi	4	5	3	5	6	5	vnmduy2002loveplantvs	5		
31	10/30/2023 15:10:34	Male (Nam)	23	Fourth year of Universi	2	1	3	3	2	1	17105@student.vgu.ec	6		
32	10/30/2023 15:10:57	Male (Nam)	21	Fourth year of Universi	2	1	2	1	2	1	17096@student.vgu.ec	6		
33	10/30/2023 15:13:48	Female (Nữ)	21	Fourth year of Universi	4	3	4	3	3	5	17251@student.vgu.ec	4		
34	10/30/2023 15:16:31	Male (Nam)	18	First year of University	4	3	6	4	5	5	haitung260105@gmail.	3		
35	10/30/2023 15:22:42	Female (Nữ)	18	First year of University	5	6	4	5	4	5	khanhleggo1101@gma	3		
36	10/30/2023 15:23:23	Male (Nam)	22	Fourth year of Universi	6	6	6	5	6	6	16669@student.vgu.ec	5		
37	10/30/2023 15:26:09	Male (Nam)	19	First year of University	4	5	5	4	4	4	hoangnhatduy04@gma	6		
38	10/30/2023 15:26:50	Male (Nam)	20	Third year of University	5	5	5	5	4	4	10421048@student.vg	4		
39	10/30/2023 15:27:20	Female (Nữ)	19	Second year of Univers	5	4	3	4	5	2	10422056@student.vg	5		
40	10/30/2023 15:28:47	Female (Nữ)	18	First year of University	5	5	4	5	5	5	tangngocbaotrantran@	6		
41	10/30/2023 15:31:04	Male (Nam)	18	First year of University	4	5	6	3	6	4	minhluanvo.wns@gma	3		

The actual data file will be provided in section [7. Reference]

4.2 Data Analysis Using Logistic Regression

4.2.1 Introduction

Data analysis using logistic regression in R is a powerful method for modeling binary or categorical outcomes. In this section, we explore the practical application of logistic regression in R for analyzing and interpreting a dataset related to satisfaction or dissatisfaction. We employ the glm function, part of the base R package, to fit logistic regression models. The analysis focuses on model development, interpretation, and performance evaluation.

4.2.2 Data Preparation

Data preparation is the process of preparing raw data so that it is suitable for further processing and analysis. Key steps include collecting, cleaning, and labeling raw data into a form suitable for machine learning (ML) algorithms and then exploring and visualizing the data.

Therefore, before using R for computing and developing any logistic regression model we need to perform some data cleanup first.

- Initially, cluster and unnecessary elements such as Vietnamese translation, timestamp or email address need to be cleared. The questions also should be shortened down to

variable names for a clearer understanding of what data they represent. This is the result after going through the above steps:

	A	B	C	D	E	F	G	H	I	J
1	Sex	Age	Year	Program	Facilities	Club	Environm	Opportun	Faculties	Overall
2	Male	18	First year	5	5	4	5	6	5	5
3	Male	18	First year	5	5	5	6	4	5	5
4	Male	20	Third year	4	5	3	3	6	3	4
5	Female	19	Second ye	4	5	6	3	2	2	3
6	Female	18	First year	5	6	4	4	4	3	4
7	Female	19	Second ye	5	4	3	5	5	4	4
8	Male	17	First year	6	6	4	5	6	4	5
9	Male	18	First year	5	5	5	6	3	6	4
10	Male	17	First year	5	4	5	6	6	6	5
11	Male	18	First year	4	5	3	3	6	4	3
12	Male	21	Fourth ye	4	4	3	4	4	4	4
13	Male	18	First year	6	6	6	6	6	6	6
14	Female	17	First year	5	6	6	4	4	5	4
15	Male	22	Fourth ye	5	6	5	5	6	6	5
16	Male	24	Third year	4	2	4	2	5	1	2
17	Male	19	Second ye	5	5	4	5	6	4	5
18	Male	19	Third year	2	2	2	2	3	2	2
19	Female	18	First year	4	5	4	4	5	5	4
20	Male	21	Fourth ye	3	2	1	4	2	3	3
21	Male	19	Third year	1	1	3	2	3	1	2
22	Female	19	Second ye	6	6	4	6	5	6	5
23	Male	19	Second ye	4	4	2	4	5	3	4
24	Male	19	Second ye	5	6	3	6	6	4	4
25	Male	19	Second ye	3	4	3	4	4	4	4
26	Female	20	Second ye	6	5	5	6	6	6	6
27	Male	21	Fourth ye	4	3	2	3	4	3	3
28	Female	18	First year	4	3	4	4	5	2	4
29	Male	18	First year	6	6	6	6	6	6	6
30	Male	21	Fourth ye	4	5	3	5	6	5	5
31	Male	23	Fourth ye	2	1	3	3	2	1	2
32	Male	21	Fourth ye	2	1	2	1	2	1	1
33	Female	21	Fourth ye	4	3	4	3	3	5	3
34	Male	18	First year	4	3	6	4	5	5	4
35	Female	18	First year	5	6	4	5	4	5	5
36	Male	22	Fourth ye	6	6	6	5	6	6	6
37	Male	19	First year	4	5	5	4	4	4	4
38	Male	20	Third year	5	5	5	5	4	4	5
39	Female	19	Second ye	5	4	3	4	5	2	4
40	Female	18	First year	5	5	4	5	5	5	5
41	Male	18	First year	4	5	6	3	6	4	4
42	Female	20	Third year	4	6	4	4	5	6	5
43	Male	18	First year	5	6	1	4	4	5	3

- After the result looks comprehensible then next is when R language is applied for grouping and factoring the data in order for the logistic regression function to work properly.
    - Grouping data: in our survey the satisfaction levels are separated into 6 levels so the responses collected are also divided into 6 levels. This is not good since too complex data will make the output varied and difficult to analyze. So we decided to group them into only half which means 3 levels with the rule:
        - Levels 1 and 2 will group into level 0
        - Levels 3 and 4 will group into level 1
        - Levels 5 and 6 will group into level 2

With the exception of the Overall variable which will only have 2 binary levels 0 represents 1, 2, 3 for satisfied and 1 represents 4, 5, 6 for unsatisfied. This variable will be used as a response variable in the calculation for a binary outcome.

```
##### Read the data
data = read.csv("E:/Data analysis/data/Rating_data_2.csv")
data

##### Group data into smaller class types
data[data == "Male"] <- 1
data[data == "Female"] <- 0
data[data == "Prefer not to say"] <- -1

data["Program"][data["Program"] == 1 | data["Program"] == 2] = 0
data["Program"][data["Program"] == 3 | data["Program"] == 4] = 1
data["Program"][data["Program"] == 5 | data["Program"] == 6] = 2

data["Facilities"][data["Facilities"] == 1 | data["Facilities"] == 2] = 0
data["Facilities"][data["Facilities"] == 3 | data["Facilities"] == 4] = 1
data["Facilities"][data["Facilities"] == 5 | data["Facilities"] == 6] = 2

data["Club"][data["Club"] == 1 | data["Club"] == 2] = 0
data["Club"][data["Club"] == 3 | data["Club"] == 4] = 1
data["Club"][data["Club"] == 5 | data["Club"] == 6] = 2

data["Environment"][data["Environment"] == 1 | data["Environment"] == 2] = 0
data["Environment"][data["Environment"] == 3 | data["Environment"] == 4] = 1
data["Environment"][data["Environment"] == 5 | data["Environment"] == 6] = 2

data["Opportunities"][data["Opportunities"] == 1 | data["Opportunities"] == 2] = 0
data["Opportunities"][data["Opportunities"] == 3 | data["Opportunities"] == 4] = 1
data["Opportunities"][data["Opportunities"] == 5 | data["Opportunities"] == 6] = 2

data["Faculties"][data["Faculties"] == 1 | data["Faculties"] == 2] = 0
data["Faculties"][data["Faculties"] == 3 | data["Faculties"] == 4] = 1
data["Faculties"][data["Faculties"] == 5 | data["Faculties"] == 6] = 2

##### Change Overall data to binary
data["Overall"][data["Overall"] == 1 | data["Overall"] == 2 | data["Overall"] == 3] = 0
```

```
data["Overall"][data["Overall"] == 4 | data["Overall"] == 5 | data["Overall"] == 6] = 1
##### Check the data type
str(data)
```

- Factoring data: to ensure logistic regression functions in R will work flawlessly, the data need to be factorized into factor type. We can check which type the data is at the present with command str(data)

```
'data.frame':   127 obs. of  10 variables:
 $ Sex          : chr  "1" "1" "1" "0" ...
 $ Age          : int  18 18 20 19 18 19 17 18 17 18 ...
 $ Year         : chr  "First year" "First year" "Third year" "Second year" ...
 $ Program      : num  2 2 1 1 2 2 2 2 2 1 ...
 $ Facilities   : num  2 2 2 2 2 1 2 2 1 2 ...
 $ Club         : num  1 2 1 2 1 1 1 2 2 1 ...
 $ Environment  : num  2 2 1 1 1 2 2 2 2 1 ...
 $ Opportunities: num  2 1 2 0 1 2 2 1 2 2 ...
 $ Faculties    : num  2 2 1 0 1 1 1 2 2 1 ...
 $ Overall      : num  1 1 1 0 1 1 1 1 1 0 ...
```

In order to change the data type to factor, some commands can help implement that:

```
#####  Factorize
###  DO NOT FACTORIZE THE VARIABLE IF ONLY DRAWING CORRELATION MATRIX
data$Sex <- as.factor(data$Sex)
data$Overall <- as.factor(data$Overall)
data$Faculties <- as.factor(data$Faculties)
data$Opportunities <- as.factor(data$Opportunities)
data$Environment <- as.factor(data$Environment)
data$Club <- as.factor(data$Club)
data$Facilities <- as.factor(data$Facilities)
data$Program <- as.factor(data$Program)
```

And if str(data) is used again, the data type has changed to factor and is now suitable for calculating anything related to logistic regression

```
'data.frame':   127 obs. of  10 variables:
 $ Sex          : Factor w/ 3 levels "-1","0","1": 3 3 3 2 2 2 3 3 3 3 ...
 $ Age          : int  18 18 20 19 18 19 17 18 17 18 ...
 $ Year         : chr  "First year" "First year" "Third year" "Second year" ...
 $ Program      : Factor w/ 3 levels "0","1","2": 3 3 2 2 3 3 3 3 3 2 ...
 $ Facilities   : Factor w/ 3 levels "0","1","2": 3 3 3 3 3 2 3 3 2 3 ...
 $ Club         : Factor w/ 3 levels "0","1","2": 2 3 2 3 2 2 2 3 3 2 ...
 $ Environment  : Factor w/ 3 levels "0","1","2": 3 3 2 2 2 3 3 3 3 2 ...
 $ Opportunities: Factor w/ 3 levels "0","1","2": 3 2 3 1 2 3 3 2 3 3 ...
 $ Faculties    : Factor w/ 3 levels "0","1","2": 3 3 2 1 2 2 2 3 3 2 ...
 $ Overall      : Factor w/ 2 levels "0","1": 2 2 2 1 2 2 2 2 2 1 ...
```

4.2.3 Model Development

Model development in logistic regression is a crucial step that involves constructing a predictive model for binary or categorical outcomes. In this section, we discuss the process of developing a logistic regression model using R, focusing on key steps, considerations, and insights gained during the development phase.

First of all, variable selection is a critical aspect of logistic regression model development. In our analysis, we carefully considered the inclusion of predictor variables based on their relevance to the research questions and the overall satisfaction outcome. An exploratory data analysis was conducted to identify potential variables that might influence overall satisfaction.

Before developing the model, it's essential to load and preprocess the dataset. This includes handling missing values, encoding categorical variables, and splitting the data into training and testing sets. Let's revisit the relevant code:

```
# Read the data
data = read.csv("Rating_data_2_demo.csv")
# Explore the structure of the dataset
str(data)
summary(data)
```

To ensure the reliability and generalizability of our logistic regression model, we employed a data-splitting strategy. The dataset was randomly divided into two subsets: a training set (80% of the data) and a test set (20% of the data). The training set was utilized for developing the logistic regression model, while the test set served as an independent dataset to evaluate the model's performance.

```
# Split the data into training and testing sets
set.seed(101)
sample = sample(c(TRUE, FALSE), nrow(data), replace = TRUE, prob =
c(0.8, 0.2))
train = data[sample, ]
test = data[!sample, ]
```

The logistic regression model was fitted using the *glm* function in R. The model formula included the response variable '*Overall*' and predictor variables '*Program*', '*Facilities*', '*Club*', '*Environment*', '*Opportunities*'and '*Faculties*'. The choice of the binomial family was appropriate for predicting binary outcomes.

```
# Logistic regression model development
logistic_full <- glm(Overall ~ Program + Facilities + Club +
Environment + Opportunities + Faculties, data = train, family =
"binomial")
```

4.2.4 Model Interpretation

The Logistic Regression model was developed to understand the relationship between predictor variables (*Program*, *Facilities*, *Club*, *Environment*, *Opportunities*, *Faculties*) and the binary outcome variable, *Overall* satisfaction. The model aimed to identify significant predictors and assess their impact on predicting the likelihood of overall satisfaction.

```
Call:
glm(formula = Overall ~ Program + Club + Opportunities + Faculties,
    family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1321   0.1220   0.2060   0.3634   1.3861

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    -7.764      2.534  -3.064  0.00218 **
Program1        1.029      1.467   0.701  0.48316
Program2        2.028      1.671   1.214  0.22482
Club1           3.582      1.466   2.443  0.01457 *
Club2           4.638      1.668   2.780  0.00543 **
Opportunities1  0.850      1.254   0.678  0.49786
Opportunities2  2.675      1.307   2.047  0.04066 *
Faculties1      3.163      1.470   2.151  0.03146 *
Faculties2      3.321      1.484   2.237  0.02527 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 95.959  on 96  degrees of freedom
Residual deviance: 48.750  on 88  degrees of freedom
AIC: 66.75

Number of Fisher Scoring iterations: 6
```

The coefficients and significance of the model represent the log odds of the outcome variable. Based on the result obtained above, there are a few things that can be acknowledged:

- Intercept: The intercept term represents the log odds of the reference category (baseline) when all predictor variables are zero. In this case, it is the log odds of overall satisfaction when all factors (Program, Club, Opportunities, Faculties) are at their reference levels.
- $Program1$, $Program2$, $Club1$, $Club2$, $Opportunities1$, $Opportunities2$, $Faculties1$, $Faculties2$: The coefficients for Club1, Club2, Opportunities2, Faculties1, and Faculties2 are statistically significant with p-values less than 0.05. This suggests that these specific levels within each predictor significantly impact the log odds of overall satisfaction.

Deviance residuals measure the difference between observed and predicted values. In our model, deviance residuals close to zero indicate a good fit, suggesting that the model effectively captures the patterns in the data. Moreover, the model's robustness is reflected in low deviance residuals, suggesting an excellent fit. The significance of certain predictor variables implies that, for this analysis, those factors significantly contribute to or detract from overall satisfaction

4.3 Model validating

4.3.1 Method

After approximating a fitting model to above 97 observations as the training dataset, we want to test how effective the model is by testing with observations it may have never seen or learned. In order to accomplish this, we devise the following strategy to measure the precision of the model:

1. Prepare a loop that perform step 2 to step 5 **100 times** as well as an empty confusion matrix to store data of 100 matrices: The idea is to measure the model against various types of observations, known or unknown, and collect various confusion matrices before finalizing into a big confusion matrix in order to interpret the overall performance of the model. In order to accomplish this, an empty matrix is created beforehand and a for-loop is used in R to execute all 4 steps below, orderly and repeatedly.
2. Choose randomly 30 observations from the 128 observations:

   ```
   test_Loop = data[sample(nrow(data), 30), ]
   ```

   By randomizing the observations, we extract mixes between known and unknown data. Regardless, there are some observations the model may not have seen before, by which we can observe how the model can actually predict its response variable.
3. Predict the target values for 30 chosen observations using the model we've just computed above:

   ```
   prediction = predict(logistic_full_no_program,test_Loop,type =
   "response")
   ```

4. Using the predict() function in R, the model will return a vector of predicted response variables corresponding to their observations. Some response variables are predicted to be very high (near 1 or near 0), indicating the model may have learned this data pattern before. However, there exist few values surrounding 0.5, showing that the model might have been confused about the given observations.
5. Design one threshold value in order to classify the prediction into class 1 or class 0:

   ```
   classification <- ifelse(prediction > 0.4,1,0)
   ```

   It is necessary to decide whether each prediction is actually meant by the model to be either response variable 1 (Good) or value 0 (Bad) since all 30 classified predictions will be compared against their original response variables. A bigger or lower threshold majorly affects the number of false classifications the model makes, while slightly to indifferently interfering with the counts of true classifications.
6. Draw a confusion matrix counting pair of predicted and actual target values of each observation:

   ```
   confusionMatrix_Loop <- table(Predicted = classification, Actual
   = testOrderedOverall_Loop)
   ```

   A Confusion Matrix is a common method used to summarize the performance of a fitting line on a set of data. The dimension for our confusion matrix is 2X2 since we classify the prediction into 2 different classes. The matrix counts the number of following attributes when comparing the predicted response variables against the actual response variables of the dataset:
   + True Positive value (predicted class is 1 - actual class is 1)
   + False Positive value (predicted class is 1 - actual class is 0)
   + True Negative value (predicted class is 0 - actual class is 0)
   + False Negative value (predicted class is 0 - actual class is 1)
7. Visualize the final confusion matrix that already stores 100 confusion matrices for 100 tests:

   ```
   confusionMatrixTotal = confusionMatrixTotal +
   confusionMatrix_Loop
   print(confusionMatrixTotal)
   ```

   After every test is conducted, we draw our big confusion matrix that has successfully captured 100 confusion matrices for 100 tests done. This matrix will be primarily used to evaluate the overall performance of the model as a whole using the following metrics in the next steps.
8. Calculate the Precision, Recall, and F1-score of the big confusion matrix to statistically understand the model's performance: The Confusion Matrix basis introduces some metrics to provide the accuracy of the model based on its 4 matrix attributes. The metrics will be

interpreted based on a percentage value in order to understand how well the model works against given observations. We will calculate 3 metrics as follows:

+ Accuracy: Measure the overall performance of the model based on its correction predictions:

$$\frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

+ Precision: Measure the accuracy of the model's positive prediction:

$$\frac{True\ Positive}{True\ Positive + False\ Positive}$$

+ Recall: Measure the accuracy of the model's prediction of a class:

$$\frac{True\ Positive}{True\ Positive + False\ Negative}$$

4.3.2 Result

	Actual	
Predicted	1	0
1	2425	202
0	46	327

- After repetitively testing the model with a randomized dataset and forming confusion matrices 100 times, the final confusion matrix is visualized above. The values are highly skewed, depending on the threshold chosen to determine the final response variable from the model's prediction. The above matrix is the result of the threshold 0.4, which will classify any prediction above 0.4 to response variable 1 and any prediction below 0.4 to response variable 0.
- The matrix is shown to have a very high True Positive value, followed by a high True Negative value. Additionally, the False Positive value is less than the True Negative value by more than 100 counts and the False Negative value being less than 50 counts is the lowest value in the matrix.
- Certain threshold values provide different ratios of False Negative / False Positive values. In the above example, 0.4 was chosen since it provides us with the lowest False Negative values possible while maintaining relatively low False Positive values. Other thresholds can provide slightly lower values of False Positive values but increase False Negative values, resulting in a bigger combination of False values.

```
> Accuracy
[1] 0.9176392
> Precision
[1] 0.9231062
> Recall
[1] 0.9817814
```

- Recall metric has a very strong value, indicating that the model has effectively predicted most of the observations with its corresponding classes.

- The precision metric is also very high, showing that the model can predict the positive class very accurately with a very rare chance that the model attempts a false classification.
- A large accuracy value implies that the model can correctly predict and classify observations to their actual response variables while maintaining low chances of making a false prediction.

4.4 Visual Presentation

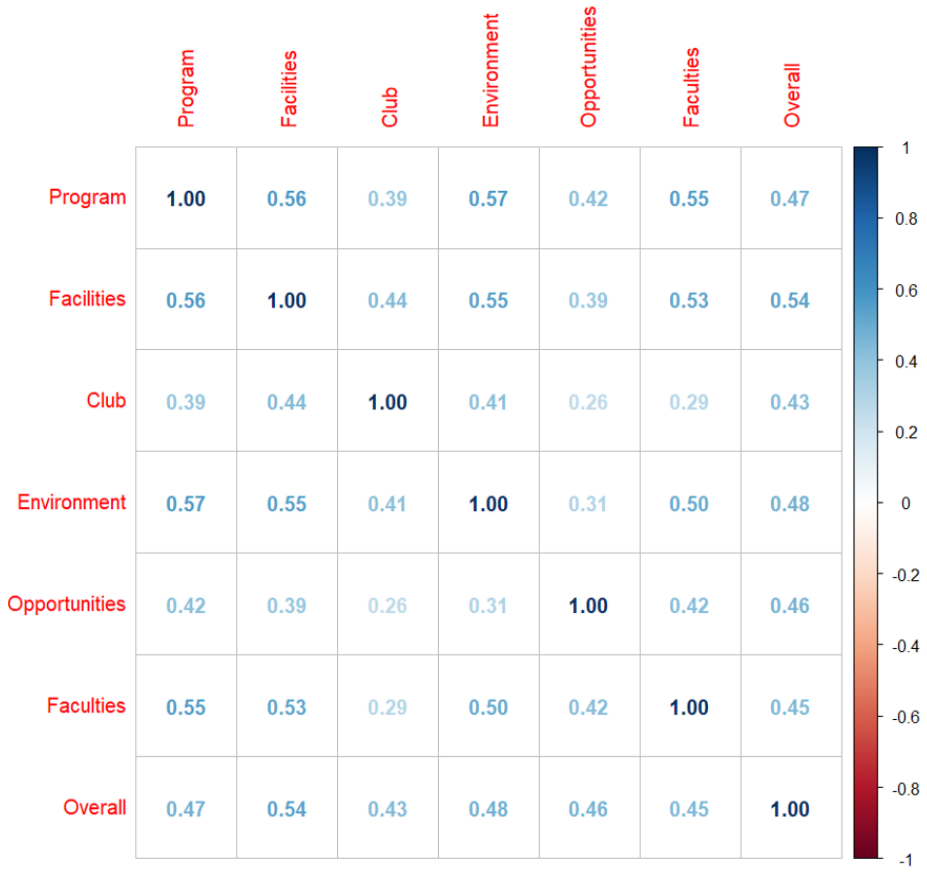
- Correlation between variables: Correlation describes the strength of an association between variables. This is the correlation coefficient.

$$\text{Correlation} = \rho = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

- In the case of this project, R can provide a function to visualize the correlation between each variable.

```
#### Create subset for chosen variables
data_cor=subset(data, select = -c(Sex,Year, Age))
#### Calculate correlation
cor(data_cor)
#### Create correlation table
corrplot(cor(data_cor), method="number", title = "Correlation")
```

The resulting graph:



There are some key takeaways that the graph shows:

+ There is no negative correlation *(-ρ)* or zero correlation *(ρ=0)* so all variables have a perfect positive relationship *(+ρ)*
+ However, the correlation coefficient between all variables is not great with the strongest being $\rho = 0.57$ and the weakest is $\rho = 0.26$

=> Despite all that, our calculation revolves around the Overall variable so only the correlation between the Overall rating and everything else should be considered. The last row in the graph represents what we need and the number is not too low so the data is acceptable.

- Predicted binary logistic regression model: based on section 4.3 Model validating, with the calculated data, library *ggplot2* in R can be used to create a graph representing the result.

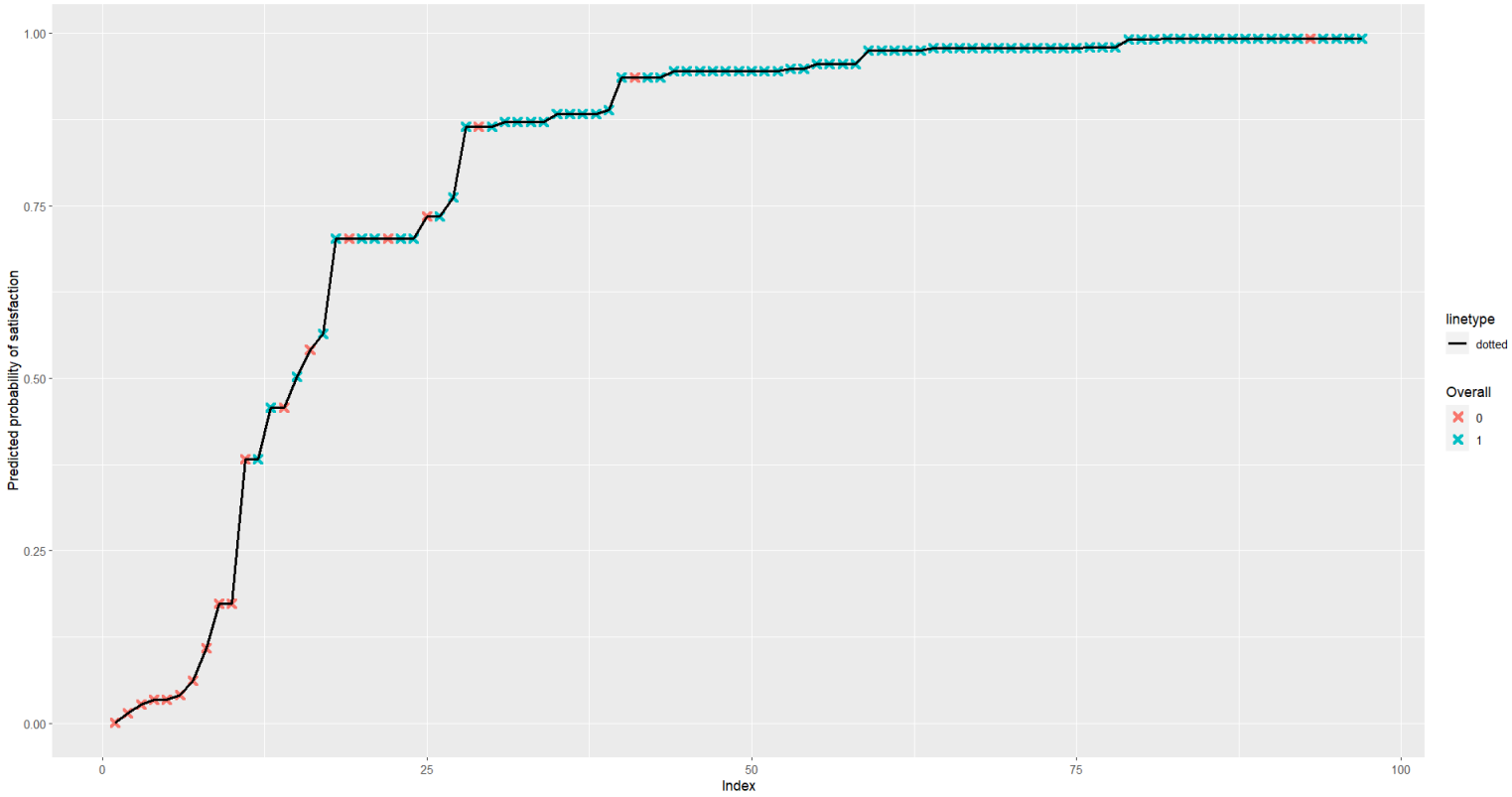
```
#### Enable the package
library(ggplot2)
#### Create data frame
predicted.data_1 <- data.frame(

probability.of.Overall=logistic_full_no_big_std_error$fitted.values,
```

```
  Overall=train$Overall
)
predicted.data_2 <- predicted.data_1[
  order(predicted.data_1$probability.of.Overall, decreasing=FALSE),]

predicted.data_2$rank <- 1:nrow(predicted.data_2)
####create predicted graph
ggplot(data=predicted.data_2, aes(x=rank, y=probability.of.Overall))
+
  geom_point(aes(color=Overall), alpha=1, shape=4, stroke=2) +
  xlab("Index") +
  ylab("Predicted probability of satisfaction") +
  geom_line(aes(linetype="dotted"), size=1)
```

The resulting graph:



More details image: [Predicted Graph]

5. Discussion

- Result Interpretation
  - Based on the standard errors and significance value of each variable in the model, variables such as Club ratings, Opportunity ratings, and Faculties ratings will majorly impact the shifting of outcome variables toward either target variable 1 or 0.
  - Both null and residual deviance being small implies that the model is also expected to provide precise outcome variables to be 1 or 0, regardless of any observations. The interpretation is further supported by our testing method using Confusion Matrix, which gives a prediction accuracy of more than 90%.
  - Additionally, metrics such as Precision and Recall value both provide similarly high percentages, justifying that the model effectively makes accurate predictions and sporadically confuses its prediction. The prediction confusion may happen only when facing very skewed observations, whose occurrences are very unlikely.
- Answers to Research Questions
  - What is the overall satisfaction level of students about many aspects of their university or school that they are studying? The variable Overall ratings are gathered as data in our survey since we resolve our calculation around that. Moreover, most students feel satisfied with their University with the binary data lean towards 1 which is satisfied
  - Which Aspect satisfies most of the students? Because the variables Club ratings, Opportunity ratings, and Faculty ratings are significant enough to affect an institution's target variable Overall rating, it is assumed that students will evaluate an institution's quality mainly based on its Club, Opportunity, and Faculty ratings.
  - Which Aspect is not satisfied by most of the students? Facilities and Environment variables are not significant enough and instead very skewed in making a prediction of the target variable's Overall rating. Therefore, it is assumed that students are unlikely to refer to Facilities and Environment aspects in order to assess a university.
  - Can we apply this analysis method to other rating survey data? This method can be used in most kinds of rating survey data, as long as the variables in observations are particularly categorical. Additionally, the Logistic regression method can be utilized for any rating survey to determine the overall ratings of the survey based on correlations between given ratings in the survey.

6. Conclusion

In summary, using Logistic Regression analysis in a rating-based survey has shown to be a useful and efficient way to extract insightful information. Through the modeling of the correlation between

the independent variables (survey factors) and the dependent variable (ratings), Logistic Regression enables us to decipher the elements influencing the ratings and forecast future results.
Through the model analysis, we managed to determine a shared relationship between all the factors in the survey with numerous different observations. We also identified the key variables that influence the most of the final overall rating of the survey and how much they impact.
By employing the Confusion Matrix testing method, we further strengthened our interpretation of the models by consistently inspecting the model with various observations and calculating the accuracy of the predictions of the model. A very high accuracy discloses that the model is very effective in determining the dependent variable based on its deep learning of the independent variables' correlations.
Overall, using Logistic Regression analysis to analyze rating-based surveys has shown to be a reliable and perceptive method that yields important information about the variables affecting ratings. Based on a detailed analysis of the survey results, this approach can greatly aid in decision-making processes, enabling businesses to optimize their strategies and raise consumer satisfaction.

7. References

- This project's survey: Satisfaction level of students
- Survey's collected data: Collected survey data
- logistic growth functions (for precalculus) - youtube.com
- Logistic Regression: A Mathematical Explanation with a Real-World Example - linkedin.com
- 5 Real-world Examples of Logistic Regression Application - ActiveWizards.com
- What is Logistic Regression? A Beginner's Guide [2023] - careerfoundry.com
- Advantages and Disadvantages of Logistic Regression - GeeksforGeeks
- 02119.pdf - tinbergen.nl
- Short History of the Logistic Regression Model Jeffrey R. Wilson & Kent A. Lorenz
- Harrell, F. E. (Year). "Regression Modeling Strategies" Springer.
- Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). "Applied Logistic Regression." John Wiley & Sons
- Agresti, A. (2012). "Categorical Data Analysis." John Wiley & Sons.

8. Appendix

```
##### Read the data
data = read.csv("Rating_data_2_demo.csv")
View(data)
```

```
##### Group data into smaller class types
data[data == "Male"] <- 1
data[data == "Female"] <- 0
data[data == "Prefer not to say"] <- -1

data["Program"][data["Program"] == 1 | data["Program"] == 2] = 0
data["Program"][data["Program"] == 3 | data["Program"] == 4] = 1
data["Program"][data["Program"] == 5 | data["Program"] == 6] = 2

data["Facilities"][data["Facilities"] == 1 | data["Facilities"] == 2] = 0
data["Facilities"][data["Facilities"] == 3 | data["Facilities"] == 4] = 1
data["Facilities"][data["Facilities"] == 5 | data["Facilities"] == 6] = 2

data["Club"][data["Club"] == 1 | data["Club"] == 2] = 0
data["Club"][data["Club"] == 3 | data["Club"] == 4] = 1
data["Club"][data["Club"] == 5 | data["Club"] == 6] = 2

data["Environment"][data["Environment"] == 1 | data["Environment"] == 2] = 0
data["Environment"][data["Environment"] == 3 | data["Environment"] == 4] = 1
data["Environment"][data["Environment"] == 5 | data["Environment"] == 6] = 2

data["Opportunities"][data["Opportunities"] == 1 | data["Opportunities"] == 2] = 0
data["Opportunities"][data["Opportunities"] == 3 | data["Opportunities"] == 4] = 1
data["Opportunities"][data["Opportunities"] == 5 | data["Opportunities"] == 6] = 2

data["Faculties"][data["Faculties"] == 1 | data["Faculties"] == 2] = 0
data["Faculties"][data["Faculties"] == 3 | data["Faculties"] == 4] = 1
data["Faculties"][data["Faculties"] == 5 | data["Faculties"] == 6] = 2

##### Change Overall data to binary
data["Overall"][data["Overall"] == 1 | data["Overall"] == 2 | data["Overall"] == 3] = 0
data["Overall"][data["Overall"] == 4 | data["Overall"] == 5 | data["Overall"] == 6] = 1

##### Factorize
### Do not factorize if drawing Correlation Matrix
data$Program = as.factor(data$Program)
data$Facilities = as.factor(data$Facilities)
data$Club = as.factor(data$Club)
data$Environment = as.factor(data$Environment)
```

```
data$Opportunities = as.factor(data$Opportunities)
data$Faculties = as.factor(data$Faculties)
data$Overall = as.factor(data$Overall)

##### Summary the data types in data
str(data)

##### Split into training samples and test samples
set.seed(101)
sample = sample(c(TRUE,FALSE),nrow(data),replace = TRUE, prob = c(0.8,0.2))
train = data[sample, ]
test = data[!sample, ]

##### Set the option to disable scientific notion
options(scipen=999)

##### Initial fitting model to evaluate the variables
logistic_full = glm(Overall ~
                      Program +
                      Facilities +
                      Club +
                      Environment +
                      Opportunities +
                      Faculties
                    , data = train, family = "binomial")
print(summary(logistic_full),show.residuals=TRUE)
logistic_full
confint(logistic_full)

##### Facilities and Environment have biggest std errors -> Removed from
the model and re-train
logistic_full_no_big_std_error = glm(Overall ~
                                       Program +
                                       Club +
                                       Opportunities +
                                       Faculties
                                     , data = train, family = "binomial")
print(summary(logistic_full_no_big_std_error),show.residuals=TRUE)
logistic_full_no_big_std_error
confint(logistic_full_no_big_std_error)

##### Testing the model with test sample
```

```
logitModelPred_response = predict(logistic_full_no_big_std_error, test,
type = "response")
logitModelPred_response

#### Create subset for chosen variables
data_cor=subset(data, select = -c(Sex,Year, Age))
#### Calculate correlation
cor(data_cor)
#### Create correlation table
corrplot(cor(data_cor), method="number", title = "Correlation")

#### Enable the package
library(ggplot2)
#### Create data frame
predicted.data_1 <- data.frame(
  probability.of.Overall=logistic_full_no_big_std_error$fitted.values,
  Overall=train$Overall
)
predicted.data_2 <- predicted.data_1[
  order(predicted.data_1$probability.of.Overall, decreasing=FALSE),]

predicted.data_2$rank <- 1:nrow(predicted.data_2)
#### Create predicted graph
ggplot(data=predicted.data_2, aes(x=rank, y=probability.of.Overall)) +
  geom_point(aes(color=Overall), alpha=1, shape=4, stroke=2) +
  xlab("Index") +
  ylab("Predicted probability of satisfaction") +
  geom_line(aes(linetype="dotted"), size=1)

########## VALIDATING THE MODEL 100 TIMES
##### Form an empty confusion matrix
PredictedEmpty <- factor(levels = c(1, 0))
ActualEmpty <- factor(levels = c(1, 0))
confusionMatrixTotal <- table(Predicted = PredictedEmpty, Actual =
ActualEmpty)

##### Repeat the testing 100 times
for (x in 1:100) {
  ### Sampling 30 samples as testing data set from the responses
  test_Loop = data[sample(nrow(data), 30), ]

  ### Predict the Overall variable for 30 samples using final model
```

```
  prediction = predict(logistic_full_no_program,test_Loop,type =
"response")

  ### Label class 1 or 0 based on the position of the above prediction
table
  ## Predicted over 50% -> Label 1 and vice versa
  classification <- ifelse(prediction > 0.4,1,0)

  ### Rearrange the sample orders from table of predicted value into
ascending order
  classification <- ordered(classification, levels = c(1, 0))

  ### Rearrange the sample orders from testing sample into ascending order
  testOrderedOverall_Loop <- ordered(test_Loop$Overall, level = c(1, 0))

  ### Form and print a confusion matrix
  confusionMatrix_Loop <- table(Predicted = classification, Actual =
testOrderedOverall_Loop)

  confusionMatrixTotal = confusionMatrixTotal + confusionMatrix_Loop
}

##### Print the confusion matrix of the model after testing 100 times
print(confusionMatrixTotal)

##### Calculate Accuracy
Accuracy = (confusionMatrixTotal[1,1] + confusionMatrixTotal[2,2]) /
(confusionMatrixTotal[1,1] + confusionMatrixTotal[1,2] +
confusionMatrixTotal[2,1] + confusionMatrixTotal[2,2])
Accuracy
##### Calculate Precision
Precision = confusionMatrixTotal[1,1] / (confusionMatrixTotal[1,1] +
confusionMatrixTotal[1,2])
Precision
##### Calculate Recall
Recall = confusionMatrixTotal[1,1] / (confusionMatrixTotal[1,1] +
confusionMatrixTotal[2,1])
Recall
```