YaleNUSCollege

Success Predictions in QR using Machine Learning modelling

Talha Ahmad Farooqui

Capstone Final Report for BSc (Honours) in

Mathematical, Computational and Statistical Sciences

Supervised by: Dr. Tim Wertz

AY 2020/2021


YaleNUSCollege

Yale-NUS College Capstone Project

DECLARATION & CONSENT

1. I declare that the product of this Project, the Thesis, is the end result of my own work and that due acknowledgement has been given in the bibliography and references to ALL sources be they printed, electronic, or personal, in accordance with the academic regulations of Yale-NUS College.

2. I acknowledge that the Thesis is subject to the policies relating to Yale-NUS College Intellectual Property (Yale-NUS HR 039).

ACCESS LEVEL

3. I agree, in consultation with my supervisor(s), that the Thesis be given the access level specified below: [check one only]

o Unrestricted access
Make the Thesis immediately available for worldwide access.

o Access restricted to Yale-NUS College for a limited period
Make the Thesis immediately available for Yale-NUS College access only from _____ (mm/yyyy) to _____ (mm/yyyy), up to a maximum of 2 years for the following reason(s): (please specify; attach a separate sheet if necessary):
_____.

After this period, the Thesis will be made available for worldwide access.

o Other restrictions: (please specify if any part of your thesis should be restricted)
_____
_____

TALHA AHMAD FAROOQUI, SAGA
Name & Residential College of Student

_____ 01/04/21
Signature of Student Date

Tim Watts _____ 01/04/21
Name & Signature of Supervisor Date

Acknowledgements

This capstone project marks the end of my undergraduate journey at Yale-NUS College. The last 4 years have been a blur. I am extremely grateful to the college and the MCS faculty for providing me with a rewarding education. As my final gift to the college, I hope the results of this capstone are conclusive enough to be implemented for the classification of the future cohorts.

I am deeply indebted to my supervisor, Professor Tim Wertz, for his valuable feedback. Due to constant motivation and his postive attitude, the project felt like a breeze. This project was only possible under his tutelage. I look up to him as a mentor, as well as a remarkable human being. From QR to capstone, we have come full circle. Thank you so much for everything.

To Sanat, Manraaj, Adarsh, ZiGi, and Aiman, thank you for making college a memorable experience.

I would like to thank ZiGi Cheong and Zainab Abaid for painstakingly proofreading this thesis and constructive criticism.

Studying abroad is a tough journey; all international students live a parallel life. I am extremely thankful to my family, my parents and my sister. I am everything because of you. This thesis is dedicated to you.

YALE-NUS COLLEGE

Abstract

B.Sc (Hons)

Success Predictions in QR using Machine Learning modelling

by Talha FAROOQUI

The course *YCC1122 Quantitative Reasoning* (QR) is a prerequisite for all Yale-NUS College students. Students are carefully hand-placed into teams by the QR facilitator based on a variety of demographic factors as well as a prediction of how well they will perform in the course. To date, this assessment has been based on educated guesswork. In this capstone, we train various machine learning models based on the last few years of QR data to predict successful candidates in QR. We use various Supervised Learning techniques to classify students into two discrete categories: High-performing and Low-performing. We successfully classified students at an accuracy of 80.6% (Logistic Regression), 81.5% (Support Vector Machines), 80.6% (Random Forests), and 82.5% (Artificial Neural Networks).

Keywords: Data Science, Machine Learning, Classification, Supervised Learning, Data Imputation, Neural Networks, Generalized Linear Models, SVM

Contents

List of Figures

“All models are wrong, but some are useful.”
-George E. P. Box

Chapter 1

Introducing the Problem

1.1 Introduction

All first-year students in Yale-NUS College have to take the course Quantitative Reasoning as part of the Common Curriculum. Students are carefully hand-placed into teams by the QR facilitator based on a variety of demographic factors as well as a prediction of how well they will perform in the course. To date, this assessment has been based on educated guesswork. The teams are split to ensure a similar level of performance. While this sound right in theory, things do not often work out well. The goal of this capstone project is to develop Machine Learning models based on the last few years of QR data to improve the allocation of students to teams to ensure a fair split of teams. Using data from QR and the admissions office, we trained various Supervised Learning models to classify students into two discrete categories: Low-performing and High-performing. Our results consist of models in which we predict a binary variable from a mix of categorical (e.g., intended major) and quantitative (e.g., standardised testing scores).

Chapter 1 describes the problem and explores the data sets. Chapter 2 is

largely centered on missing data imputation and conversion of data. Chapter 3 focuses on data pre-processing and feature engineering for classification. Chapter 4 describes the performance of various supervised learning models and compares them. Chapter 5 concludes this project with the results and the limitations.

1.2 Motivation of the Problem

All incoming students into Yale-NUS College have to take the course Quantitative reasoning as part of their first, ungraded semester. This course helps them hone their math and statistical skills, as well as give them an introduction to statistical programming in R. Presumably, students with former background in Mathematics and Statistics settle in relatively easier into the class because of the overlap of topics they learn in high school and the first few weeks of QR. People from other streams have a harder time settling in. QR is partially a team-based course, where the grade depends on individual, as well as collective performance. Students from each class are split into three to four different teams, with each team consisting of 6 students each. The facilitators want the teams to be split fairly. The instructors thus use a metric of predicting how well the students will perform beforehand, and split the students accordingly. Data from the Admissions office has demographic information used as predictors. A variety of factors are considered: the students' gender, their academic interests and backgrounds, and their demographics and geographical status. It would not be fair to put all the STEM (Science, Technology, Engineering and Math) students in one group and all the non-STEM ones into another, as this would cause the STEM team to be at an advantage because of their mathematical background. Similarly, groups can not have all same sex members because

of bias. Instead of algorithms, the facilitators use educated guesswork and allocate the teams as they see fit. This could lead to potential biases and could be detrimental to the performance of students if they are allocated sub-optimal partners.

We aim to solve this problem by eliminating human bias and training Machine Learning models based on the last few years of QR data to classify students and improve the allocation of students to teams. We used various supervised learning techniques including Generalised Linear Models (Linear and Logistic Regression), Support Vector Machines (SVMs), Decision Trees, Random Forests Classifiers, and Neural Networks. We will fit these models on two data sets to classify and sort students. We will wrangle and clean data by imputing missing values, as well as the transformation of continuous and categorical variables into quantitative (dummy) variables for prediction. We will compare and contrast the results of each model, and decide which model is the best classifier.

1.3 Data and Exploratory Data Analysis

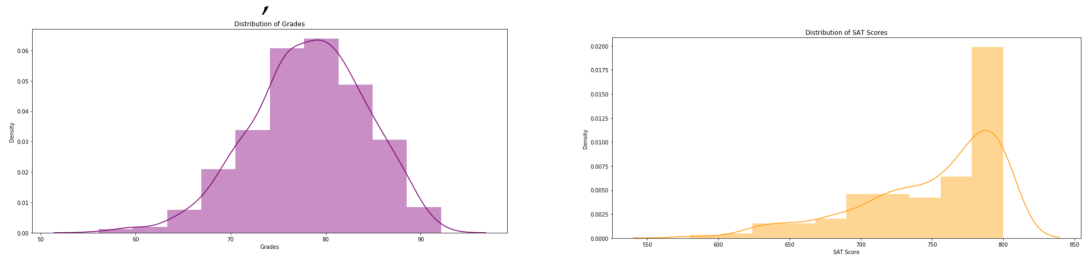
This project was based on two different data sets: admissions data and QR data. The first data set was supplied by the Yale-NUS Admissions Office and contained data on the 1170 students who matriculated from 2016 to 2020. The columns included the matriculation date, gender, Academic Interests 1 and 2 (students' intended majors when applying) , and the standardised math test scores for the Scholastic Assessment Test (SAT) and American College Testing (ACT). The data set also comprised of high school grades from three different educational disciplines.

The second data set (1012 rows $\times$56 columnns) had a complete record of the student performances in QR– Final Grades, Midterm and Exam results, IRAT and TRAT scores, peer and instructor evaluations, as well as their attendance and writing reports score. This data set also contained the students' actual declared majors, and their regions. The data was anonymised and the countries where students came from were consolidated into seven regions to further protect student identities. Students hailed from seven unique regions: Singapore, East Asia, Southeast Asia, South Asia, North America, Europe, and 'Other' Regions.

A substantial portion of the admissions data was missing, particularly high school grades and standardised test scores. Yale-NUS College did not require applicants to take the SAT or ACT in the past (they were optional). Students thus do not feel the need to take these tests. Similarly, most international students receive their high school examination results after their admission. Yale-NUS College has a non-conditional admission policy in which students are not required to submit their scores. Therefore, students are not compelled to report scores and grades once admitted because it does not impact their admission. 'Missingness Maps' were created using the R package `Amelia` which show the spread of missing values for both datatsets. The Admissions data had 77% missing data (Figure A.9), while the QR data had 7% missing data (Figure A.10).

Exploratory Data Analysis (EDA) was done on two data sets showing the distribution of data. The QR final grades have a mean of 78.9 (out of 100) and standard deviation of 6.15. 50% of students scored at least 79.2 and 75% scored at least 83.3. The SAT score distribution was highly skewed with mean 750 (out of 800) and standard deviation 48.6. 50% of students scored at least

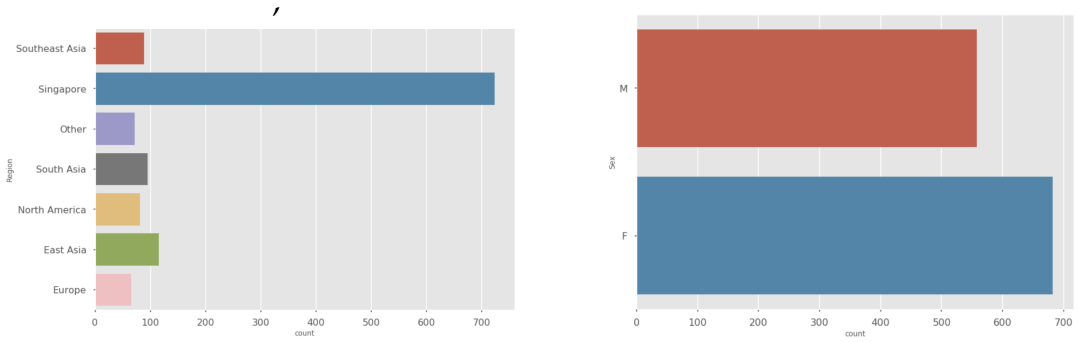
760 and 75% scored at least 790. Figure 1.1 shows the density plots for the distribution of SAT scores and QR final grades.



(A) QR Final Grade (B) SAT Scores

FIGURE 1.1: Density plot of QR grades and SAT scores.

The student distribution shows there were more female students than male, with a ratio of approximately 700 : 550 students. Overall, there were roughly 700 Singaporean students and 550 internationals (Figure 1.2). The batch size of the students grew steadily from 2014 to 2019.



(A) Region (B) Gender

FIGURE 1.2: Bar plot of student regions and gender.

Yale-NUS College offers 14 different majors, as well as a Double Degree Law programme with NUS. The distribution of the declared majors by QR students reveals that the majority declared (Philosophy, Politics, and Economics) as their major, followed by MCS (Mathematical, Computational and Statistical Sciences) and Economics (Figure A.6). Looking at the spread of students' first Intended Majors, PPE is the most popular response with 160 candidates,

followed by Global Affairs and Economics (130). MCS and Life Sciences have 80 responses each. PPE is the most favourable choice for the second intended major (165), while MCS is one of the least popular choices. Figures 1.3 and 1.4 show the spread of intended majors.

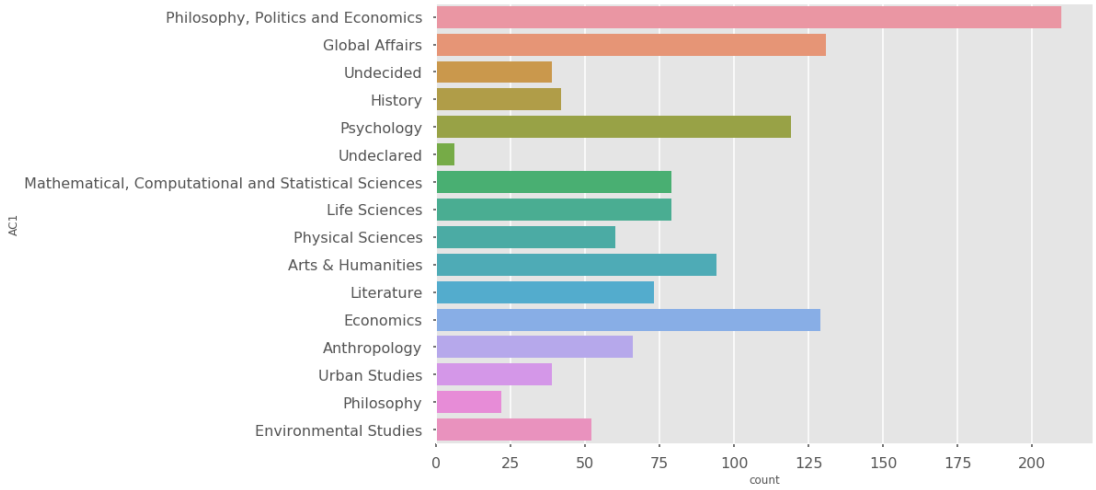


FIGURE 1.3: Bar Plot of students' intended major 1

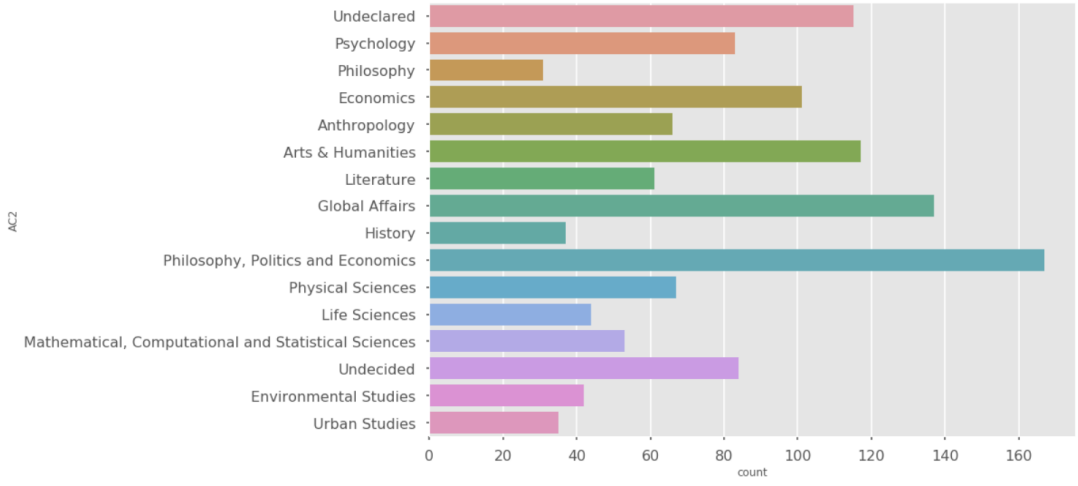


FIGURE 1.4: Bar Plot of students' intended major 2

Chapter 2

Score Concordance and Missing Data Imputation

2.1 Standardised Testing Scores and Grades Concordance

In this capstone, we were investigating variables that could help us predict the QR grade. Looking at students' performance in standardised tests was the next logical step in predicting their in-class performance. 85% international colleges require students to take at least one Standardised Test for undergraduate admissions (Schneider and Dorans, 1999). Yale-NUS College has a holistic admissions process in which grades are just one of the many contributing factors towards a students' admission. Standardised Testing is a requirement for admissions for international students, where they must submit either the Scholastic Assessment Test (SAT) or the American College Testing (ACT). 60% data was missing since Singaporean students are not required to provide standardised tests for their university applications.

The SAT and ACT are exams required for undergraduate university admissions. The SAT is split into two sections: English and Mathematics. Each

section is graded on a scale of 200-800 (in increments of 10-points). The ACT is split into four subsection, and each subsection is graded on a scale of 1-36 (in increments of 1 point). The ACT and SAT are similar in their multiple-choice format as well as exam duration. The Yale-NUS Admissions Office only reported us the math scores from the standardised tests.

We noticed a discrepancy between the SAT and ACT Math scores: the amount of SAT scores was three times the amount of ACT scores. Therefore, a strong need for score *concordance* was felt. Concordance "refers to establishing a relationship between scores on assessments that measure similar constructs" (from the website *ACT/SAT Concordance*). According to a study, concorded scores on non-parallel tests (such as SAT and ACT) are not "equal", but "comparable". (Marco, Abdel-fattah, and Baron, 1992) A potential use of score conversion would be to gauge the probability of students performing well in QR.

Even though conversions of data from one form to another results in a loss of information, the missing data posed a graver issue. Using the official ACT/SAT Concordance table, ACT scores were converted into SAT math scores. If students had taken both the SAT and the ACT, the higher of the two scores was chosen. Ultimately, concordance helped us reduce the missing data problem within the standardised scores.

2.2 High School Grade Conversion

High School Grades were another variable of interest to predict students' performance in QR. The admissions data contained grades from the three types of educational programmes: International Baccalaureate (IB), Cambridge GCE A-Levels, and Advanced Placement (AP). We converted various interdisciplinary education boards scores into a common scale.

The students reported grades for different subjects. Singaporean students have to take a minimum of 6 subjects for Cambridge GCE A-levels, while international students only have to take three. The admissions data set only showed the grades for Singaporean students for five subjects: Two lower level H1 subjects (Mathematics and Physics) and three higher level H2 subjects (Mathematics, Physics, and Computer Science). Students giving the International Baccalaureate (IB) exams have to take six subjects, which are divided into a mix of High Level (HL) and Standard Level (SL) subjects. The admissions data set showed results for seven IB subjects: Math Studies (SL), Mathematics (HL and SL), Statistics (HL), Further Mathematics (SL), and Computer Science (HL and SL). Advance Placement (AP) students have to take 5-8 AP classes. Grades were only reported for four subjects: Calculus, Statistics, Physics 1, Computer Science.

Since the majority of students (Singaporeans) in Yale-NUS College took Cambridge GCE A-level exams, the new shared scale was determined to follow the grading pattern of GCE A-levels (A, B, C, and so on). Using various resources, the spread, distribution and the difficulty level of the IB grades and AP grades were considered in the allocation of grades following the A-level conventions. Similarly, the distribution of the A-level grades was researched and grades were selected after averaging. Lastly, these new grades were compiled manually in a column named **HSG** (High School Grades). The HSG column takes the form of ordinal grades ranging from A to D. This will be helpful for classification ahead.

2.3 Missing Data Imputation using MICE

Various machine learning models (both supervised and unsupervised learning algorithms), are used for analytics . Algorithms train on massive data sets and use some variables to make predictions or classifications. The data set in question must be properly cleaned and wrangled for the models to make predictions. This step of cleaning and refining the data is known as data preprocessing, and is a crucial beginning step to train the models. The most common problem encountered by analysts at this step is the management of missing data and the identification of outliers. Sometimes data is either missing from the data sets, or it ends up being in the data set as a result of being incorrectly entered. This leads to a significant decrease in the data quality, which results in subsequent wrongful predictions by the models (Khan and Hoque, 2020). Furthermore, missing values in data sets may significantly increase computational and operational costs, lead to a decrease in efficiency and time management, bias the outcome, and frustrate analysts (Graham, Cumsille, and Shevock, 2012). The presence of missing values also hinders the use of various algorithms. For instance, some supervised machine learning techniques (e.g., Neural Networks, Logistic Regression, Support Vector Machines, etc.) cease to work with missing values (Slavakis, Giannakis, and Mateos, 2014).

Even after conversion and concordance of both the standardised test scores and grades, 547 SAT values and 742 HSG values were missing due to admission rules and a non-conditional admission policy (securing a seat once admitted does not depend on high school grades). For large data sets, it is usually an easy solution to just drop the rows with the missing values and conveniently ignore those observations. In a smaller data set like ours, the removal of missing values is detrimental: it not only causes a significant loss in the data

information but also decreases the data efficiency and statistical power (Kwak and Kim, 2017).

Considering the scale of the missing data, the two widely used solutions: list-wise deletion and mean/median substitution– would not have worked. Instead, we used Multiple Imputation. In this approach, we use the distribution of the observed variables to estimate potential values for the missing data, instead of substituting missing values with a predetermined value as per the general convention. The advantages are immense: this method helps account for the uncertainty around the true value, obtains unbiased estimates, and allows more flexibility in choosing the number of imputations. We used the `MICE` package, which stands for *Multivariate Imputation by Chained Equations in R*. This algorithm was implemented by Stef van Buuren in 2010, and it can impute a variation of ordered/unordered categorical, binary and continuous data (Buuren and Groothuis-Oudshoorn, 2010).

2.4 Working of MICE package

The package `MICE` works in the following three steps:

- Selects values that keep the relationship within the data set intact in place of missing values
- Forms independently drawn imputed data sets (default 5)
- Calculates new standard errors using variation across data sets caused by the imputed data sets

The mice function imputes data in one line of code:

```
mice(data,m=5,maxit=50,meth='pmm',seed=500).
```

Several parameters are involved in the function, which are described in the official R package documentation (Buuren et al., 2015). The parameters used in the function above are as follows:

- **data**: A data frame containing the incomplete data.
- **m**: The number of imputed data sets created by MICE. The default is $m = 5$, however, it can range from 3 to 10.
- **maxit**: A scalar giving the maximum number of iterations.
- **meth**: Specifies the imputation method to be used for each column in data. The default method is 'pmm' (Predictive Mean Matching).
- **seed**: An integer that is used as an argument by the set.seed() for offsetting the random number generator.

MICE has more than twenty methods of imputations for different types of missing data e.g. logistic regression for ordered categorical variables and polytomous regression for unordered data etc. After careful research, we found that the best method to impute our continuous data is the 'PMM' method, which stands for Predictive Mean matching. Buuren writes that PMM "is the only method that yields plausible imputations and preserves the original data distributions" (Vink et al., 2014). PMM is also "independent from the missingness mechanism and preserves the correlation in the data when outliers are not considered". Unlike other imputation methods, PMM is also free from distributional assumptions, and thus offers flexibility in its imputations. The next section describes the results with the 'pmm' imputation method in detail.

2.5 Data Imputation and EDA

Before imputing, we can see the patterns of the missing data using the `md.pattern` function from MICE. The missing data pattern plot is shown in Figure A.12. The plot indicates 342 samples lack only the HSG values, 147 samples miss the SAT values, and 400 samples miss both. The number of missing values in the SAT column is 547, and the number of missing values in the HSG column is 742. Another visual representation of the missing data is obtained using the `aggr` function. The aggr function stands for Aggregations for missing/imputed values, and it "plots the amount of missing/imputed values in each variable and the amount of missing/imputed values in certain combinations of variables" (Templ et al., 2020). From the plot, we can infer that almost 40% of the samples are missing HSG and SAT information, and 34% are missing the HSG values. The function also provides plots of the histogram of the missing data and sorts the variables by the number of missing data points. In our case, HSG had a higher missing value count (0.7449) compared to SAT (0.5492). Figure 2.1 shows the aggregation plots of the missing and imputed data.

Another visualisation tool is the `marginplot` function. It stands for scatterplot with additional information in the margins, and it shows information about missing and imputed values as well. The newly imputed values are present in the scatterplot. This plot restricts one to plot just two variables. For our case, two is feasible as we are just imputing two variables: SAT and HSG. Figure 2.2 shows the margin plots of the missing and imputed data.The blue box plot (on the bottom) shows the distribution of HSG with SAT missing, while the red box plot shows the distribution of the remaining data points. The red box plot (on the left) shows the distribution of SAT with HSG missing while the blue box plot shows the distribution of the remaining data points.

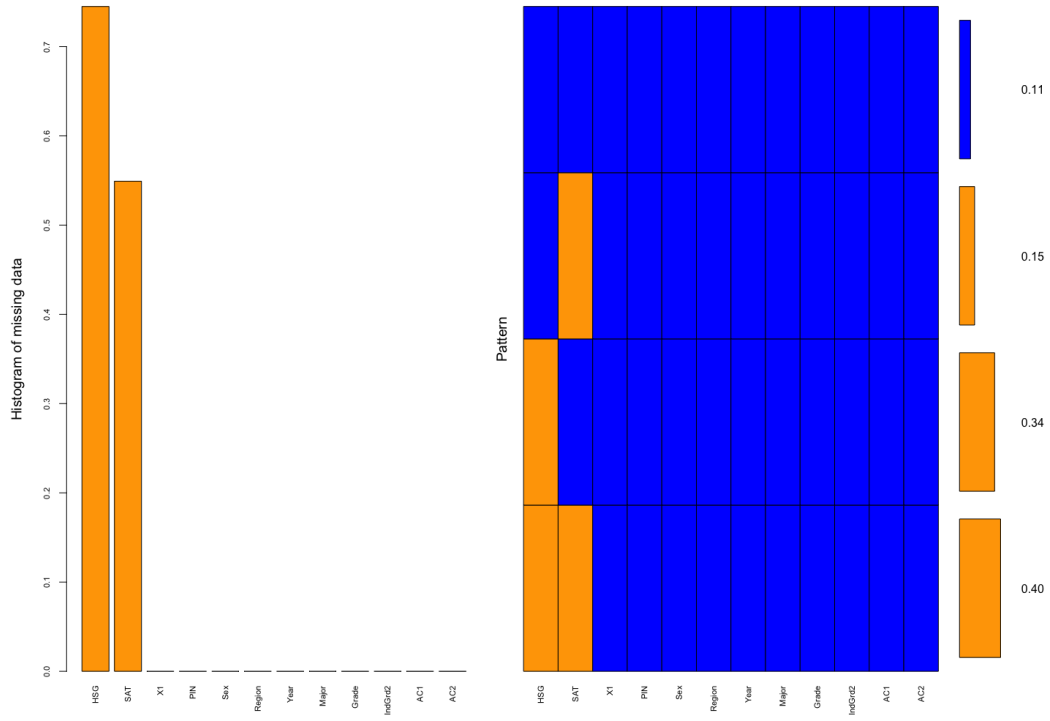


FIGURE 2.1: Aggregations for missing/imputed values. HSG had a higher missing value count than SAT.

Both the box plots look similar.

After successfully imputing the data, we can see the distribution of the imputed data in comparison with the original data. We use the high-level R function `xyplot`, which creates Common Bivariate Trellis Plots, designed mainly for two continuous variates. We will use this function to compare the distributions of original and imputed data using by using a scatter plot and plotting HSG and SAT values against the Grade variables. The red points on the figure 2.3 and A.13 are the imputed data points, while blue points are the original (observed) data. These red (imputed) points are similarly spread as the original data points, which indicates that our imputed data is feasible.

Lastly, we consider the use of density plots, using the R function `densityplot`.

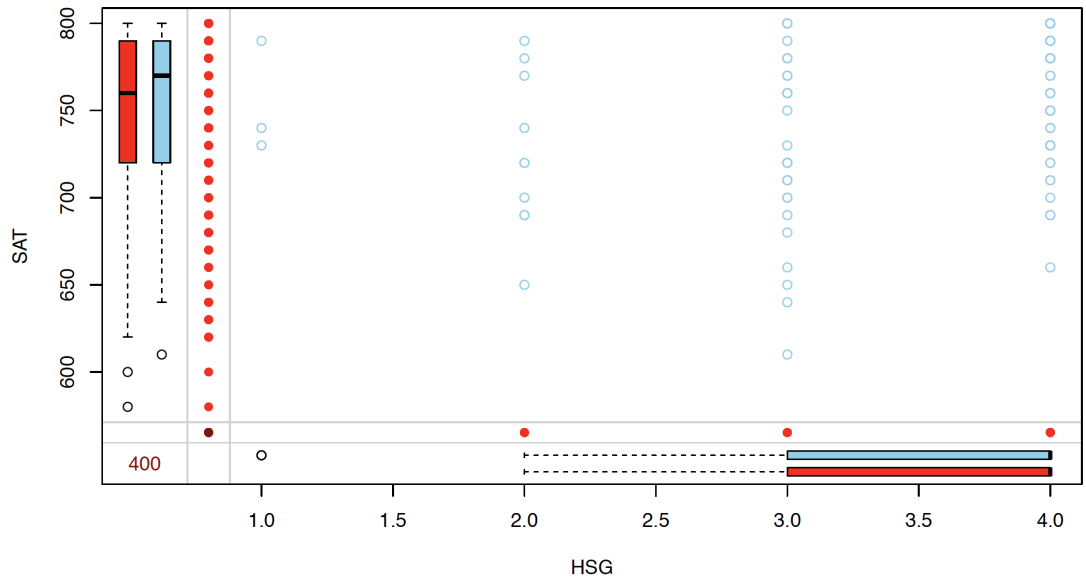


FIGURE 2.2: Margin Plot of SAT and HSG

which "constructs and graphs non-parametric density estimates" (R documentation). The density plots of the missing and the imputed data are shown in Figure 2.4. Here, the blue curves represent the density of the original data, while the pink points indicate the newly imputed data. The distributions are similar.

Overall, the imputed data using Predictive Mean Matching from MICE looks plausible and is similarly spread like our original data. Using concordance and MICE imputation helped us solve the missing data problem. We still need to further wrangle and manipulate data in order to use it to train models. The next chapter deals with data pre-processing and feature engineering.

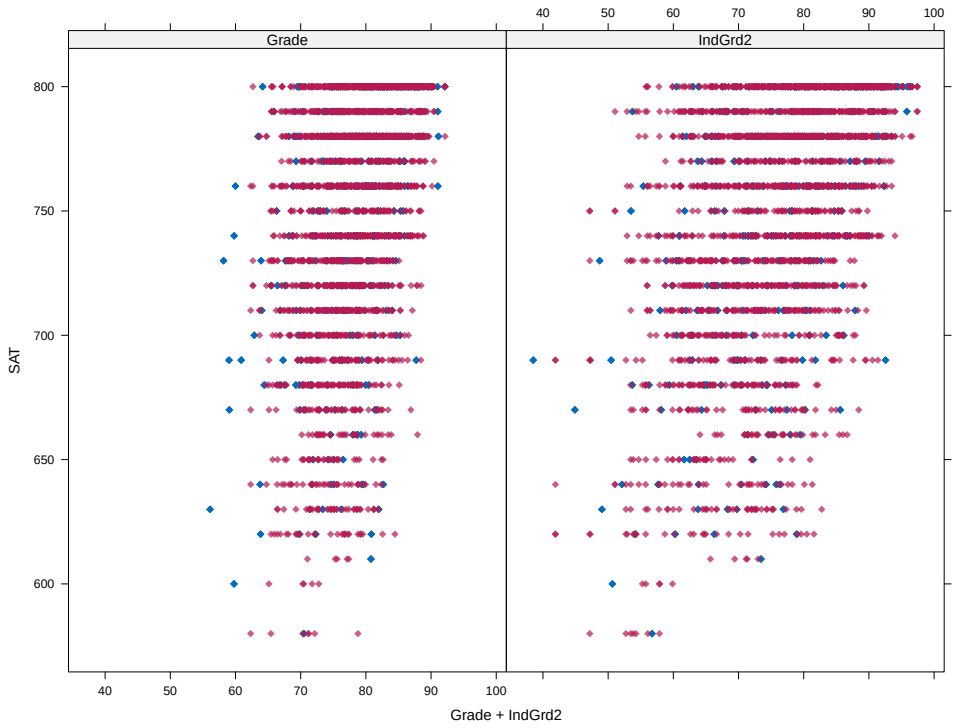


FIGURE 2.3: High level Trellis function of SAT. The blue points are original data, the pink points are imputed data.

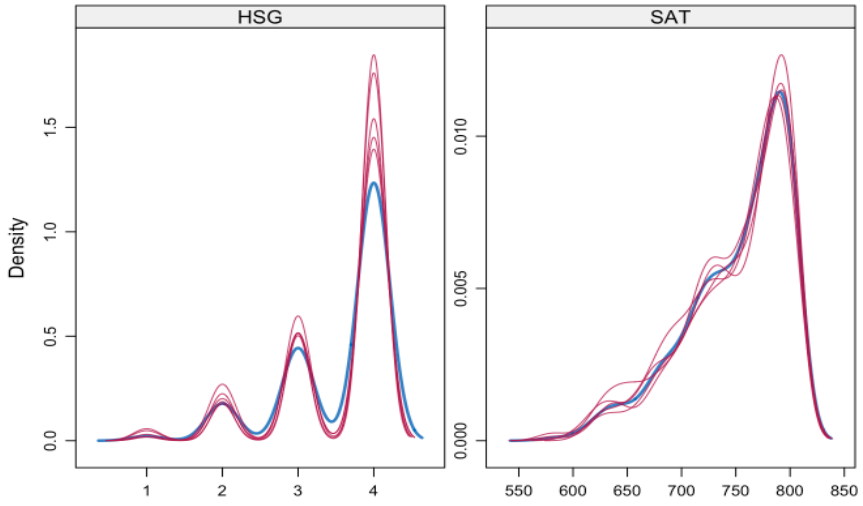


FIGURE 2.4: Density Plot of missing/imputed values

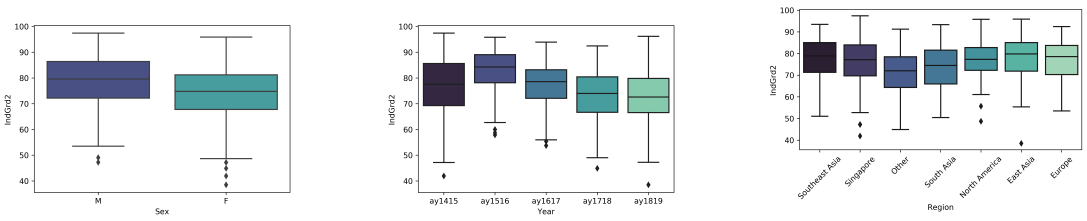
Chapter 3

Data Preprocessing, Feature Engineering, Diagnostics

3.1 Extracting Individual Grades

The final grade in QR is comprised of different components– both team based and individual. Usually, 60% of the final grade component is team based, while 40% is individual. The grade is calculated into IRATs, TRATs, midterm and final exam, peer and instructor evaluation, reports and presentations. Out of these, only the two exams and IRATs are performed individually. For classifying students, we decided that the optimal predictor variable was a student's individual grade in QR. For this, we separated the three individual components from the QR Final Grade. IRATs have a 10%, weightage, while the Midterm Exam and Final Exam each have 15% weightage in the final grade. We wrote a function in R that reweighs the aforementioned variables out of 100, and then takes an average to get a rescaled grade. We saved these in our dataframe in a column titled **IndGrd2**. This is essential for modelling as we will be focusing on using only these individual grades as the response (dependent) variable for classification. There was a strong correlation of 89% between individual

grades and final grades, proving the final grade in QR depends on individual performance. The following boxplots show the distribution of Individual Grades with respect to gender, region, and student batch. Male students have a higher individual grade than female students. Students scored higher marks in some years than others, e.g ay1516. Singapore and Southeast Asian students had the highest scores, while students from 'other' regions and North America scored lower.



(A) Gender (B) Academic Year (C) Region

FIGURE 3.1: Boxplots showing spread of Individual Grades

3.2 The final Dataset

Using the imputed data from MICE, a final data set 1240×7 (1240 rows and 7 columns) was created. The top five rows are shown in Figure 3.2 .The description of each column is as follows:

1. **Sex** Gender of the student (M= Male, F= Female)
2. **Region** The geographical region of the student
3. **HSG** Stands for High School Grades, which are the student's (converted) grades in High School, on a range of 4-1, with 4 being the best and 1 being the worst (4 represents an A, 3 represents a B, 2 represents a C, and 1 represents a D)

4. **AC1** Students' Academic Interest 1 (first intended major)

5. **AC2** Students' Academic Interest 2 (second intended major, if any)

6. **SAT** The students' SAT scores (after concordance)

7. **IndGrd** Students' Individual Grade on a binary (1/0) scale. The column IndGrd2 was converted into binary labels after seeing the distribution of the student's individual grades. The mean was 77.3 and the standard deviation was 9.66. To classify students into high and low performing categories, a threshold of 1 standard deviation from the mean was set. Therefore, students with marks above 86.9 were represented by 1 and all other students were assigned 0. This will be our response (dependent) variable, and we will be classifying students on the basis of this.

	Sex	Region	HSG	AC1	AC2	SAT	IndGrd
0	M	Southeast Asia	3	Philosophy, Politics and Economics	Undeclared	770	1.0
1	M	Singapore	4	Global Affairs	Psychology	680	0.0
2	F	Singapore	4	Undecided	Philosophy	730	0.0
3	F	Singapore	3	Philosophy, Politics and Economics	Psychology	790	0.0
4	F	Singapore	3	Philosophy, Politics and Economics	Economics	800	1.0

FIGURE 3.2: Top five rows of the final dataset (1240 x7)

3.3 Feature Engineering

3.3.1 One-Hot Encoding

Using the `dtypes` function fromt the Python library `pandas` reveals the datatypes of the variables in our dataset. We have a mix of categorical, numerical, and

continuous data in the dataset. Sex, Region, AC1 and AC2 are categorical variables so they are encoded as strings. Our numerical variables are IndGrd, SAT, and HSG. Some columns are categorical variables with two levels (e.g Sex), so we transform them into binary variables using the map function. Here, 1 represents a Yes, and a 0 represents No. For a lot of other variables, they take other values which are not yes and no. Therefore, using the industry standards, One-Hot-Encoding is used. We select some categorical, but not ordinal features which have more than 2 values, and transform them into dummy variables using the `get_dummies` function from pandas. Each label is mapped to a binary vector. After feature engineering, the 1240×7 data frame was converted to a 1240×44 data frame (Figure A.16). We will be using this data frame with dummy variables for further testing and classification. This dataset is cleaned and wrangled so that most Machine Learning models (e.g SVMs and NNs) work perfectly on it.

	HSG	SAT	IndGrd	Sex_F	Sex_M	Region_East Asia	Region_Europe	Region_North America	Region_Other	Region_Singapore	...
0	3	770	1.0	0	1	0	0	0	0	0	...
1	4	680	0.0	0	1	0	0	0	0	1	...
2	4	730	0.0	1	0	0	0	0	0	1	...
3	3	790	0.0	1	0	0	0	0	0	1	...
4	3	800	1.0	1	0	0	0	0	0	1	...

FIGURE 3.3: Top five rows of the One-Hot Encoded data set (truncated)

3.3.2 Training and Testing split

Using the **train_test_split** function from the Python library `sklearn`, we split the data randomly into two subsets:

1. **Training Dataset** Used to train the machine learning model by fitting on this subset

2. **Testing Dataset** Used to test the model by evaluating the final model fit

Splitting data into training and testing helps reduce computational cost and eliminates bias in the predictive performance of the models. Since we have a sufficiently large dataset, we set a configuration parameter to 0.2. 80% of the data will be used for training and 20% will be used for testing. The IndGrd column will be used for prediction. The `train_test_split` function splits a data frame into input (X) and output (y) columns, and then further splits X and y columns to training and testing subsets. Our input column X is the dataset containing all the columns except the 'IndGrd' variables. Our output column y is the IndGrd column (the one with $1/0$ values for the student QR performance). We will use the training data to train various supervised learning models, and use the testing set to evaluate the model performances. Some diagnostic tools are used as metrics to test the model performance, which are defined in the next section.

3.4 Diagnostics of a Classifier

The following definitions are taken from the textbook Data Science and Big Data Analytics, page 185,224-225 (Dietrich et al., 2015).

- **True Positive:** Positive instances correctly identified as positive
- **True Negative:** Negative instances correctly identified as negative
- **False Positive:** Negative instances incorrectly identified as Positive
- **False Negative:** Positive instances incorrectly identified as Negative

- **True Positive Rate (TPR)/Recall:** Percentage of positive instances correctly identified by the classifier. TPR= Number of true positives / Number of positives = TP/ FP + TN

- **False Positive Rate (FPR):** Percent of negatives the classifier marked as positive. FPR= Number of false positives / Number of negatives = FP/FP + TN

- **Precision:** Precision is the percentage of instances marked positive that really are positive. Precision = TP/ TP + FP

- **Accuracy:** The rate at which a model has classified the records correctly. Defined as the sum of TP and TN divided by the total number of instances. Accuracy = (TP + TN/TP + TN + FP + FN) * 100

- **Confusion Matrix:** A 2×2 matrix which reports the number of FP,FN,TP, and TN. It helps us calculate other metrics used to measure model performance (e.g accuracy and precision).

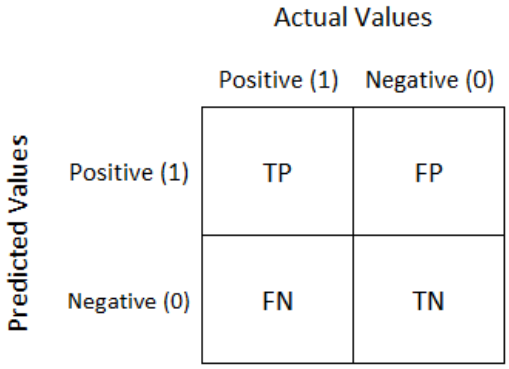


FIGURE 3.4: An illustration of Confusion Matrix (sourced from Towards Data Science)

- **F1 Score:** The harmonic mean of the precision and recall (on a scale of 0-1). F1 score= 2 * (precision * recall) / (precision + recall)

Chapter 4

Machine Learning Modelling

This chapter focuses on fitting several supervised learning techniques on our datas et and looking at their performance. In a supervised learning model, the data set comprises of labeled data. We use these labels to train algorithms to classify data into two discrete labels (1: High-Performing students, 0: Low-performing student). For our models, the Individual Grades on a scale 0-1 (IndGrd) will be our binary response (dependent) variable (except linear regression, where we use the former individual grades). The remaining six variables (Sex, Gender, HSG, SAT, AC1 and AC2) will be our explanatory (independent) variables.

4.1 Generalized Linear Models (GLMs)

A Generalized Linear Model (GLM) with k variables takes the form:

$$g(\mu_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \tag{4.1}$$

Here, the linear predictor (or response variable) $y_i; i = 1, 2, 3, \ldots, k$ is modelled by a link function g that links $\mathbb{E}[Y]$ with a linear function of explanatory variables $x_i; p = 1, 2, 3, \ldots, k$ (Turner, 2008).

A GLM has three elements:

1. A probability distribution belonging to the exponential family
2. A link function which shows how the mean (expected value) depends on the linear predictor $g(\mathbb{E}[Y]) = g(\mu_i) = y_i$.
3. A variance function that describes how the variance depends on the mean $\text{var}(Y_i) = \phi V(\mu)$, where ϕ (dispersion parameter) is a constant (Turner, 2008).

4.1.1 Linear Regression

Using R, Linear Regression was done on our dataset. We used the identity link function $g(\mu_i) = \mu_i$, and the variance function $V(\mu_i) = 1$. Using the R package `fastDummies`, categorical variables were converted into dummy variables. This is useful because we can use a single linear regression equation to represent multiple groups. After creating the dummy variables, the R package `stats` was used to fit the model using the `glm` function in R. The function takes the form:

```
glm(formula, family= ''; (link=), data = )
```

The glm formula has the following arguments:

- **Formula:** Formula of the response variable (y) as a function of the explanatory variables (x).
- **Family:** A description of the error distribution. There are multiple probability distribution functions that can be used depending on the type of regression, e.g binomial, gaussian, poisson, gamma etc.

- **Link:** The link function to be used in the model. The link function depends on the family type, e.g logit for binomial, and identity for gaussian.
- **Data:** Dataframe, list, or environment containing the model variables.

The results for the linear regression using the originally extracted Individual Grades (on a scale of 0-100) as the dependent variable and Academic Interest 1 and 2 as the independent variable are shown in the appendix in figures A.17 and A.18. The statistically significant predictors' p-values are circled in red. For the first fit, it shows that having Arts and Humanities, Global Affairs, History, and Literature as the first intended major will have a negative impact on the final QR grade, decreasing the score by 4 units. Alternatively, Physical sciences as intended major leads to an expected grade increase of 3.5 units. Having MCS as an Academic Interest 1 is statistically insignificant. However, for AC2, MCS is statistically significant, showing an increase in the final grade by 5 units. Literature is also statistically significant predictor, leading to a decrease of roughly 4 units. This model shows that Academic Interests (students' intended majors) thus play a significant part in a students QR grade. For instance, if a student selects Physics as their first academic interest and MCS as their second, they are likely to be one of the top-performers in QR. Alternatively, students selecting Arts and Humanities as their first academic interest and Literature as their second are likely to under perform. While the geographic region of the students did not indicate significant differences, it is notable that students from the region "other" tend to not perform as well as the remaining six regions (Figure A.19).

Running a linear regression model with binary Individual grade (0-1 scale) as dependent variable and all other variables as independent variables shows

that only SAT, HSG, and SexM (male students) are statistically significant. Table 4.1 shows the results of these statistically significant variables.

Predictor	Est.	S.E.	t val.	p
(Intercept)	-1.48	0.19	-7.99	0.00
SexM	0.09	0.02	4.18	0.00
HSG	0.08	0.01	5.23	0.00
SAT	0.00	0.00	7.78	0.00

TABLE 4.1: Regression Results (statistically significant)

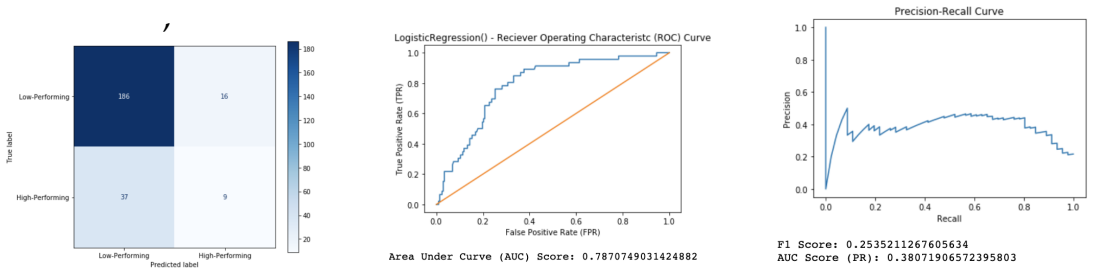
4.1.2 Logistic Regression Classifier

This section will focus on using Logistic Regression to build a model where we have a binary response variable. In our case, the response variable Y is the students' Individual Grade–a categorical response variable with two levels: 1 and 0. Instead of modelling the outcome response Y, logistic regression instead models how the probability that Y belongs to a particular category. The high performing students take the value 1 with probability p and low-performing students take the value 0 with probability $1 - p$. The probability p is modelled in relation to the predictor variables. The logistic regression uses the *logit* link function, the negative derivative of the binary entropy function. The logistic regression model thus models the log odds as a linear transformation of the predictor variables (Hastie, Tibshirani, and Friedman, 2009). It takes the form:

$$logit(Y) = \log\left(\frac{\mathbb{E}[Y]}{1 - (\mathbb{E}[Y])}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \tag{4.2}$$

We fit the logistic regression model on using the function `LogisticRegression` from the Python library `scikit-learn` (sklearn). Using the `.predict` function,

we get the predicted classes as well as probabilities of the students. Our Logistic Regression model is 80.6% accurate. The confusion matrix shows 167 True Negatives, 9 True Positives, and just 16 False Positive and 37 False Negatives each. Out of 248 students, the logistic regression model correctly predicted 167 low-performing students and 9 high-performing students. The model performance can be further evalutaed by the Receiver Operating Characteristic (ROC) curve, an indicator of the performance of a binary classifier. ROC is a plot of TPR (True Positive Rate) against the FPR (False Positive Rate). Having a high TPR and a low FPR is ideal. The higher the area under the curve (AUC), the better the model is at classification. Our AUC score is 0.787. The precision recall graph is also presented below. The F1 score (a harmonic mean of precision and recall) is 0.25, while the area under curve (AUC) score is 0.38. Figure 4.1 shows the aforementioned diagnostics for Logistic Regression.

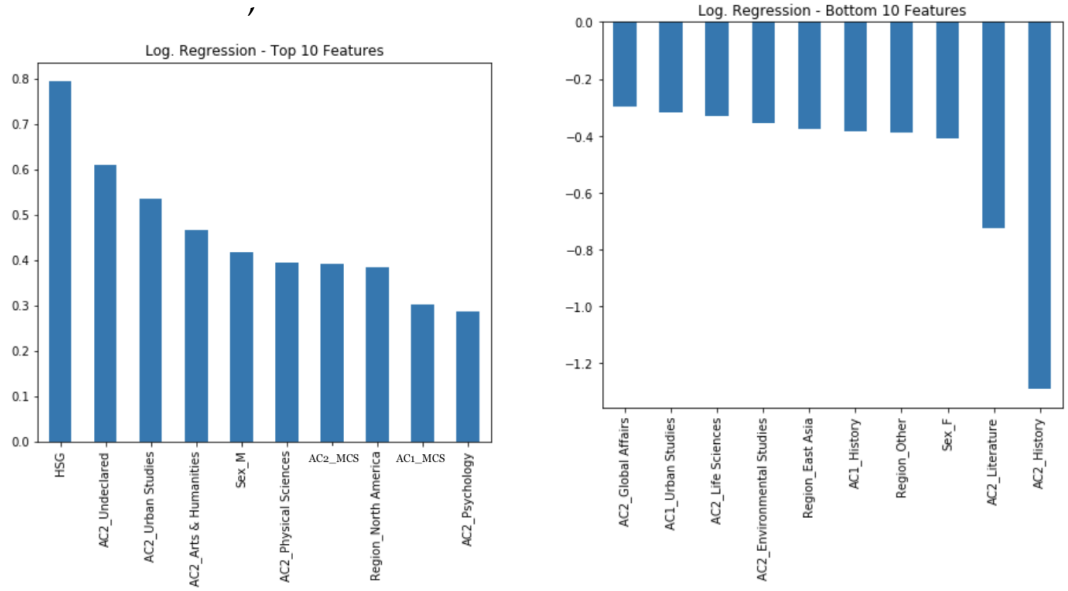


(A) Confusion Matrix (B) ROC Curve (C) Precision-Recall Curve

FIGURE 4.1: Logistic Regression Classifier diagnostics

Next we use a function that prints the top 10 and bottom 10 features that predict student's performance in QR. This shows us the features with respect to their coefficients (weights). The graphs are shown in Figure 4.2. From the plots, we can infer HSG is the biggest predictor, followed the intended (second) majors Urban Studies and Arts and Humanities. The male gender and MCS

intended majors also are helpful for predicting student performance. The features that have the least or no impact on the model are the the regions Other and East Asia, and the female gender.



(A) Top 10 (B) Bottom 10

FIGURE 4.2: Logistic Regression Model Features

4.1.3 Pros and Cons

Both linear and logistic regression rely on the linearity assumption where the response variable is a linear additive function of the explanatory variables. When the linearity assumption does not hold, then data must be appropriately transformed (Dietrich et al., 2015). Linear Regression can not be used if the outcome variable is categorical. Logistic Regression is a better choice for that.

4.2 Decision Trees and Random Forest Classifier

Decision Trees are another method used for regression and classification. Here, the predictor space is divided or stratified into a number of regions which can all be summarised in a tree. Tree-based methods are easy to use and visualise, and are and useful for interpretation. In terms of prediction accuracy however, they may lag behind some other supervised learning techniques. To combat thise, we can use the techniques of bagging (merging the same type of predictions) and random forests in which "multiple trees are combined to yield yield a single consensus prediction" (Hastie, Tibshirani, and Friedman, 2009). While this is harder to interpret, it offers increased prediction accuracy. Given a set of categorical or continuous input variables $X = x_1, x_2, \ldots, x_n$, the goal of a decision tree is to predict an output variable Y. At every testing point (node), a specific branch represents a decision and goes down the tree. Ultimately, a final point (leaf node) is reached, and a final prediction can be made (Dietrich et al., 2015).

Our decision tree formed 607 nodes and reached a maximum depth of 27. On training data, the Decision Tree formed 491 nodes with maximum depth 32. The decision tree is presented in Figure A.20 (Appendix). From the testing dataset of 248 students, the decision tree classifier correctly predicted 11 high performing (TP), and 189 low performing students (TN). There are 13 FP and 35 FN which bring the accuracy of the classifier down to 73.3%.

We will next use a random forest classifier for better predictions. Random forests are defined as "a combination of trees such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest" (Breiman, 2001). An illustration of a random forest is shown in Figure 4.3

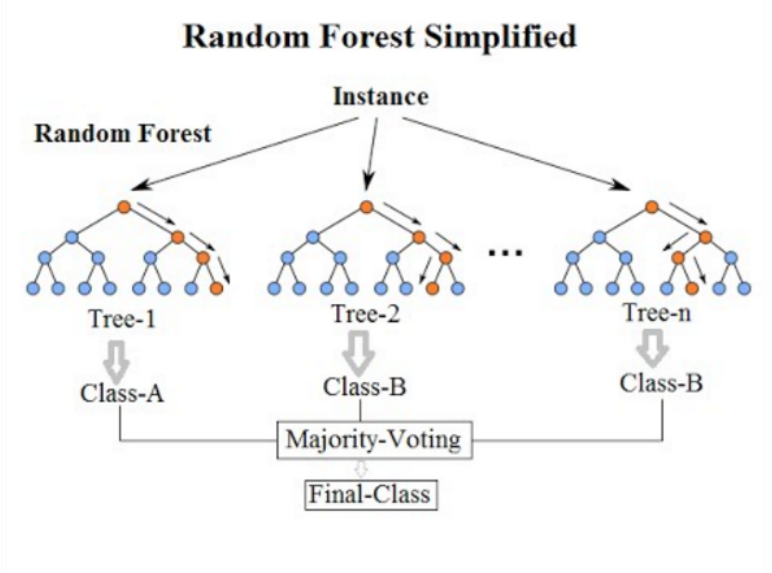
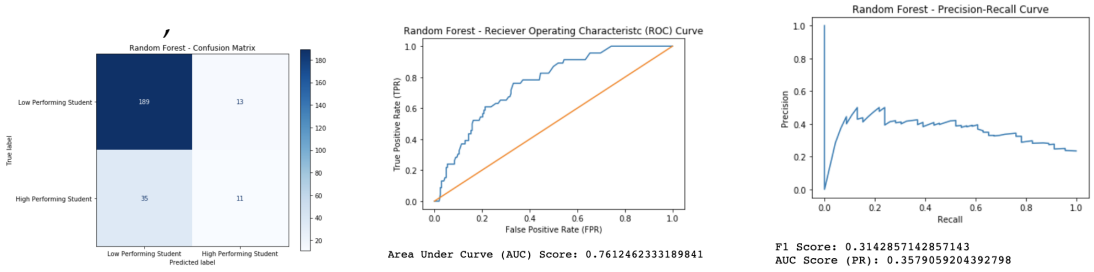


FIGURE 4.3: An illustration of a Random Forest classifier (sourced from wikipedia). Multiple decision trees are combined together and averaged to give the final output.

Random forests combine trees such that they reduce the correlations between trees, hence improving the quality of the predictions. This is done by randomly subsetting the features for each tree. A random sample of $n = \sqrt{p}$ predictors is chosen as split candidates for a total number of p trees (Hastie, Tibshirani, and Friedman, 2009). The confusion matrix for Random Forest classifier shows an accuracy of 80.6%. It had 11 TP, 189 TN, 13 FP, and 35 FN; an improvement from the decision tree classifier. The confusion matrix, ROC curve (AUC 0.76) and the precision-recall curve for the random forest classifier are shown in Figure 4.4.

Random Forests also help us visualise the feature importance in a model. Using the sklearn function `.feature_importances`, we plot the impurity-based importance of each variable. This importance is calculated based on the mean decrease in impurity (MDI). Figure 4.5 below shows the feature importance scores. The most important feature is SAT with a feature importance score of



(A) Confusion Matrix (B) ROC Curve (C) Precision-Recall Curve

FIGURE 4.4: Random Forest Classifier performance diagnostics

0.25. Other top features are HSG, Singapore (region), and Economics and PPE as the first intended major. The least important features are History (AC1 and AC2), and AC2 Life Sciences.

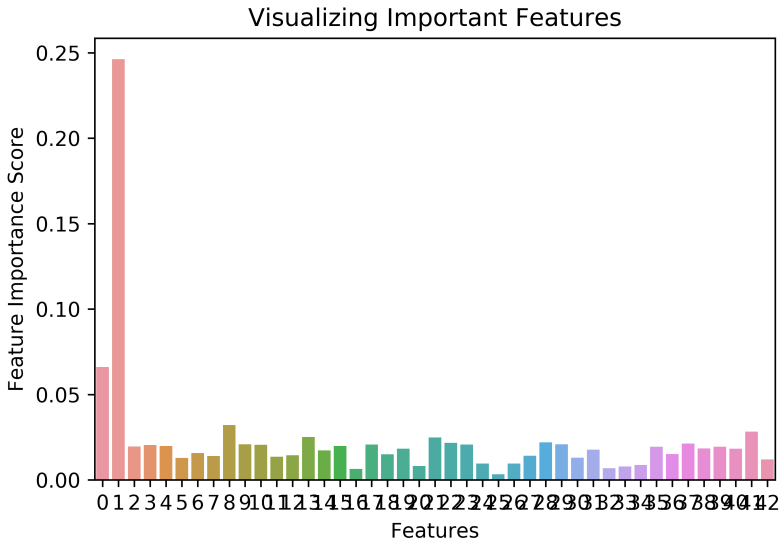


FIGURE 4.5: Feature Importance Graph (RF)

4.2.1 Pros and Cons

Random Forests are useful for accurate prediction, regression, and classification. They can handle a mix of continuous and categorical data, and are easy

to interpret. Random forests also do not overfit because of Law of Large Numbers; they converge to a lower Mean Squared Error. (Hastie, Tibshirani, and Friedman, 2009)

4.3 Support Vector Machines (SVM)

Next, we consider the use of a Support Vector Machine. "SVMs are a set of related methods for supervised learning, applicable to both classification and regression problems. A SVM classifiers creates a maximum-margin hyperplane that lies in a transformed input space and splits the example classes, while maximizing the distance to the nearest cleanly split examples." (Shmilovici, 2009).

In an *n-dimensional* space, a hyperplane is a flat subspace of the dimension $n-1$.For example, in a 2-D space, a unidimensional line is a hyperplane, and in a 3-D space, a 2-D plane will be a hyperplane. An n-dimensional hyperplane takes the form of the equation:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n = 0 \tag{4.3}$$

The hyperplane divides an *n-dimensional* space into two distinct classes, with the points in any class being either > 0 or < 0 (Hastie, Tibshirani, and Friedman, 2009). An example of 2-D and 3-D hyperplane is given in Figure 4.6.

The smallest perpendicular distance from each training observation is referred to as the margin. SVMs are based on maximal margin hyperplane (MMH)- a separating hyperplane that maximizes the margin. Data points close to the hyperplane influence it, and are called *support vectors* as they are vectors *n-dimensional* space. The MMH depends not on the data points, but rather the

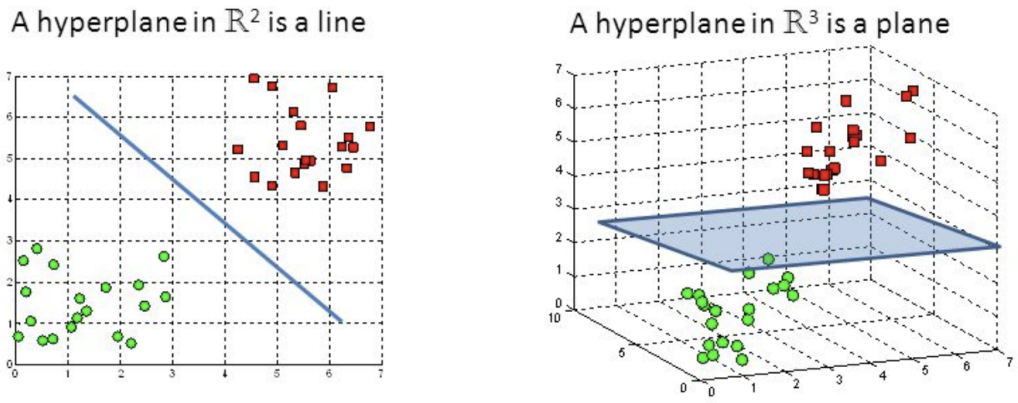


FIGURE 4.6: Hyperplanes in 2-D and 3-D (sourced from Towards Data Science)

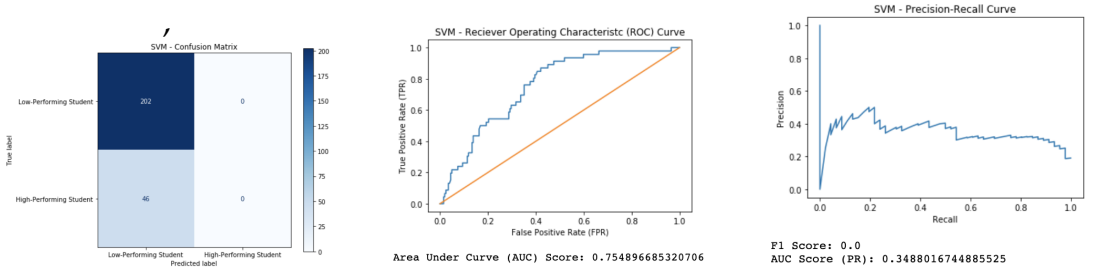
the support vectors (Hastie, Tibshirani, and Friedman, 2009). This hyperplane is also sensitive to data; a change in even one data point leads to a shift in the orientation and position of the hyperplane, which affects classification. It also might lead to overfitting of training data. Thus, a support vector classifier (SVC) is considered which uses a hyperplane that does not perfectly classify two classes, but allows a few points to be wrongly classified to account for an increased robustness to individual observations and improved classification of majority of observations. A SVM is thus "an extension of the SVC that results from enlarging the feature space using kernels" (Hastie, Tibshirani, and Friedman, 2009).

SVMs are ideal for classifying data with non-linear decision boundaries, by using various *kernels*. A kernel is function that transforms data into a suitable space. A kernel is defined as a generalization of the inner product of two points. Different kernels transform data into different forms. There are a few options e.g linear, polynomial, radial basis function (rbf), and sigmoid. These kernel functions help for space transformation and separating non-linear points. Using `sklearn`, a SVM is fit on our data in just two lines of

code. We experiment with different kernels. A linear kernel is a generalization of a the inner product, and takes the form of $K(x, y) = x^T y$. The polynomial kernel of degree n is defined as

$$K(x, y) = \left(\sum_{i=1}^{p} x^T y + 1 \right)^n$$

The polynomial kernel gives us the best performance, with an accuracy of 81.5%. The Confusion matrix shows 202 TN, and 46 FN, with the AUC score is 0.75. The Confusion Matric, ROC Curve, and Precision-Recall curve are present in Figure 4.7.



(A) Confusion Matrix (B) ROC Curve (C) Precision-Recall Curve

FIGURE 4.7: SVM performance diagnostics

4.4 Artificial Neural Networks (ANN)

Lastly, we consider an Artificial Neural Network, an effective tool for classification. We will use a three layer sequential feed-forward neural network with sigmoid function in input and output layer. We will use `Keras`, a high-level deep learning API and the `TensorFlow` backend for the neural network. The neural network function takes the form of the equation

$$y_k(x, w) = \sigma(\left(\sum_{j=0}^{M} {w_{kj}}^{(2)} h(\left(\sum_{i=0}^{D} {w_{ji}}^{(1)} x_i \right) \right) .$$

Here, M linear combinations of the input variables $x_1, \ldots, x_D$ are in the first layer, and w_{kj} are the weights and w_{ji} are the second-layer bias. Therefore, a neural network is "a nonlinear function from a set of input variables x_i to a set of output variables y_k controlled by an adjustable vector of weights w" (Bishop, 2006). In the equation, σ represents a sigmoid activation function. The sigmoid activation function is

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

and is ideal for binary classification since its output lies between 0 and 1 (Bishop, 2006). Figure 4.8 illustrates a neural network with 2 layers described in our equation.

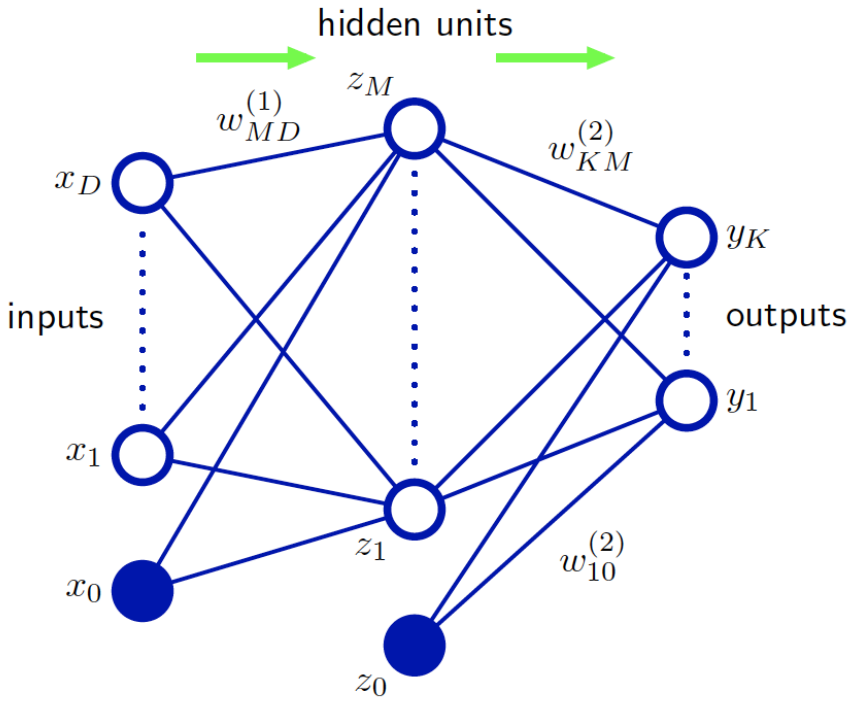
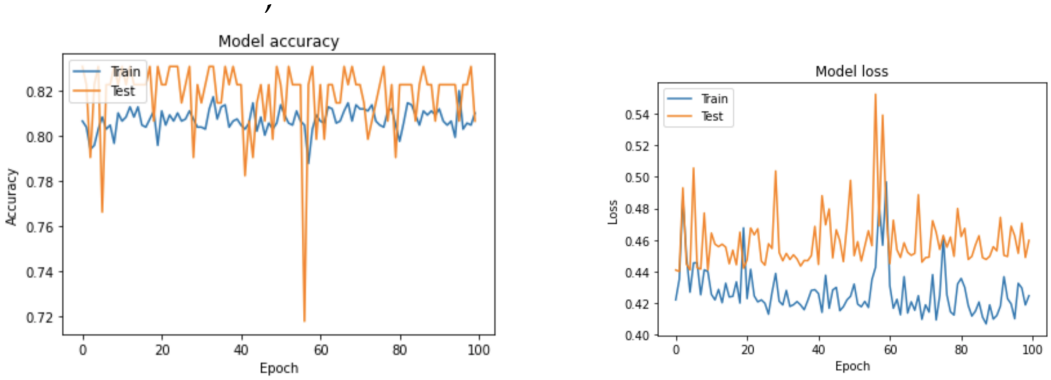


FIGURE 4.8: Illustration of a 2 layered Neural Network (Bishop, 2006)

We first define our model using the Sequential model class in Keras (which groups a linear stack of layers into the model). Our model will have three layers. Since we have 49 input (dummy) variables, our first layer has 49 nodes. Our first hidden layer has 40 nodes, and our second hidden layer has 24 nodes. Our final output layer has two nodes for binary classification and use the sigmoid activation function.The neural net has two hyperparameters that need tuning to optimise the model. These include the number of layers, and the number of nodes. Generally, "a neural net never needs more than two hidden layers to solve most problems" (Lapedes and Farber, 1988). There is no right answer for what are the best hyper-parameters. Finding the best hyper parameters is often a trial and error approach.

The model is compiled using the TensorFlow backend. Lastly, we start training our model on our test data. Training a neural net requires multiple iterations of passing the dataset through the model, and each iteration is called an epoch. We can split a large dataset into smaller batches, so that it is computationally inexpensive to run the model. For our dataset, we use a 100 epochs, and a batch size of 64. After training, the model will converge at a certain error level. We evaluate the model to see its performance. The final model testing accuracy is 82.5%. Figure 4.9 shows the accuracy and the loss of training and testing data in 100 epochs. The orange lines indicate testing accuracy and loss, and the blue line indicate training accuracy and loss. Our goal is to increase the accuracy and decrease the loss iteratively. From the figures, it seems testing accuracy and testing loss do not change much.

The confusion matrix of the ANN gives 203 TN, 45 FN, and has an accuracy of 82.5%.



(A) Accuracy (B) Loss

FIGURE 4.9: Training and Testing Accuracy and Loss graphs

Chapter 5

Discussions

5.1 Results

We successfully classified students at an accuracy of 80.6% (Logistic Regression), 81.5% (Support Vector Machine), 80.6% (Random Forests), and 82% (Artificial Neural Networks). The Support Vector Machine and Artificial Neural Network only classified low-performing students. They falsely classified high-performing students. Thus, they were not very helpful. Random Forests and Logistic Regression gave us very similar results. Both these models classified low and high performing students at an accuracy of 80.6%. Logistic Regression, however, had a higher AUC score (0.78) compared to Random Forest (0.76). Therefore, Logistic Regression is the best classifier in our capstone project. Moreover, our Logistic Regression classifier indicates that the high school grades (HSG), male gender and MCS intended majors are helpful for predicting student performance. This model can be used to predict successful candidates in QR by the MCS facilitators next semester.

5.2 Limitations

There are a few limitations and possible caveats to our Machine Learning project.

- Yale-NUS College has a non conditional admission policy. Students are not required to submit their final high school grades for admission. Furthermore, Singaporean students are also not required to submit standardised testing scores. This results in a huge amount of missing data. In our Admissions data set, 77% of the data was missing (Figure A.9). While we used to imputation and concordance to solve this missing data problem, it still let to a decrease in accuracy in classification. Logistic Regression and Random Forest classifiers were used to test classification in the original data. The 1240x7 dataset was reduced to a 101 x 7 dataset after removal of all missing values, a 92% decrease. The accuracy of the classifiers on this data was 97% (Logistic Regression) and 100% (Random Forest). On the data set with imputation and concordance, both Random Forest and Logistic Regression gave the same accuracy of 80.6%. Thus, missing data led to a decrease in accuracy by approximately 20%.

- The concordance between SAT and ACT has some limitations. The scores are not equivalent, but comparable. The scales are different for both exams which means there exist many-to-one possible score conversions (e.g a 34 in ACT can take more than one value if converted to a SAT score). Lastly, students with concorded scores have a higher SAT score than the average scores (Schneider and Dorans, 1999). Therefore, concordance is helpful to some extent, but is not a perfect way to transform data.

- The first semester at Yale-NUS College is ungraded. Students either pass or fail the course. This potentially may affect the students' motivation to perform well in the course (unlike the other graded courses). Therefore, even a student predicted to be a high-performing student might not perform well in QR due to not having a grade contributing to their GPA.

- Admission in Yale-NUS College is competitive; the usual acceptance rate ranges between 3-6%. Due to a high admission threshold, Yale-NUS College students are talented individuals and get exceptional results in high school examinations. There is low variance in the high school grades as shown in the Figure A.11. Most students have As in their examination results. Low variance in the grade data means students are clustered together and are indistinguishable. Thus, low variance in the HSG predictor is a limitation for classification of students into discrete categories.

Bibliography

ACT/SAT Concordance. URL: https://www.act.org/content/act/en/products-and-services/the-act/scores/act-sat-concordance.html.

Bishop, Christopher M (2006). *Pattern recognition and machine learning*. springer.

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Buuren, S van and Karin Groothuis-Oudshoorn (2010). "mice: Multivariate imputation by chained equations in R". In: *Journal of statistical software*, pp. 1–68.

Buuren, Stef van et al. (2015). "Package 'mice'". In: *Computer software*.

Dietrich, David et al. (2015). *Data science and big data analytics: discovering, analyzing, visualizing and presenting data*. John Wiley & Sons,

Graham, John W, Patricio E Cumsille, and Allison E Shevock (2012). "Methods for handling missing data". In: *Handbook of Psychology, Second Edition* 2.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). "An introduction to statistical learning". In:

Khan, Shahidul Islam and Abu Sayed Md Latiful Hoque (2020). "SICE: an improved missing data imputation technique". In: *Journal of big data* 7.1, pp. 1–21.

Kwak, Sang Kyu and Jong Hae Kim (2017). "Statistical data preparation: management of missing values and outliers". In: *Korean journal of anesthesiology* 70.4, p. 407.

Lapedes, Alan and Robert Farber (1988). "How neural nets work". In: *Evolution, learning and cognition*. World Scientific, pp. 331–346.

Marco, Gary L, AA Abdel-fattah, and Patricia A Baron (1992). "METHODS USED TO ESTABLISH SCORE COMPARABILITY ON THE ENHANCED ACT ASSESSMENT AND THE SAT®". In: *ETS Research Report Series* 1992.1, pp. i–19.

Schneider, Dianne and Neil Dorans (1999). "Concordance between SAT® I and ACT™ Scores for Individual Students. Research Notes. RN-07." In: *College Entrance Examination Board*.

Shmilovici, Armin (2009). "Support vector machines". In: *Data mining and knowledge discovery handbook*. Springer, pp. 231–247.

Slavakis, Konstantinos, Georgios B Giannakis, and Gonzalo Mateos (2014). "Modeling and optimization for big data analytics:(statistical) learning tools for our era of data deluge". In: *IEEE Signal Processing Magazine* 31.5, pp. 18–31.

Templ, Matthias et al. (2020). *Package 'VIM'*.

Turner, Heather (2008). "Introduction to generalized linear models". In: *Rapport technique, Vienna University of Economics and Business*.

Vink, Gerko et al. (2014). "Predictive mean matching imputation of semicontinuous variables". In: *Statistica Neerlandica* 68.1, pp. 61–90.

Appendix A

Supplementary and Additional Figures

A.1 Code

The complete code can be found in this GitHub repository. The repository includes all the R code, R Markdown files, Jupyter Notebooks, Plots and Figures, and a copy of the thesis.

A.2 Supplementary and Additional Figures

This section has the following supplementary and additional figures:

1. Exploratory Data Analysis plots
2. Missing data and imputation plots
3. Data sets
4. Machine Learning figures

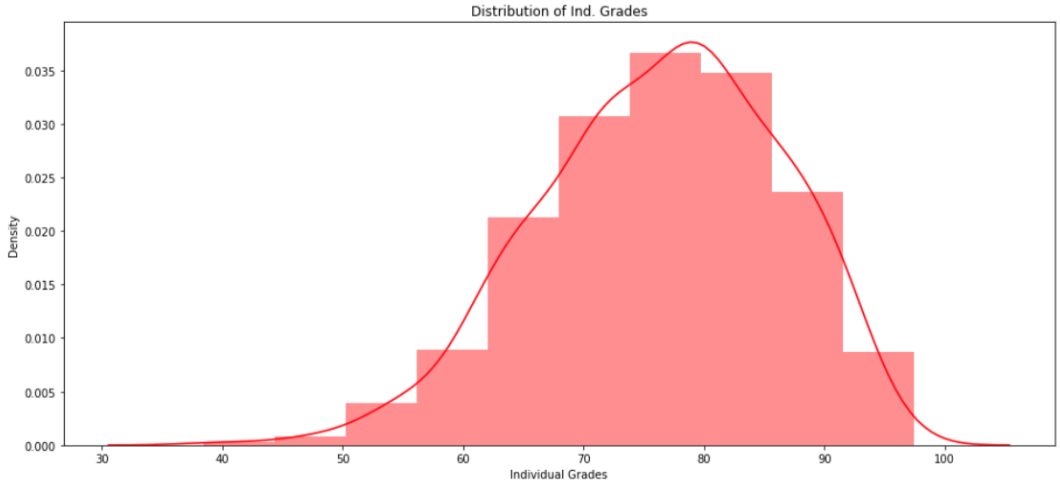


FIGURE A.1: Density Plot of Individual Grades in QR

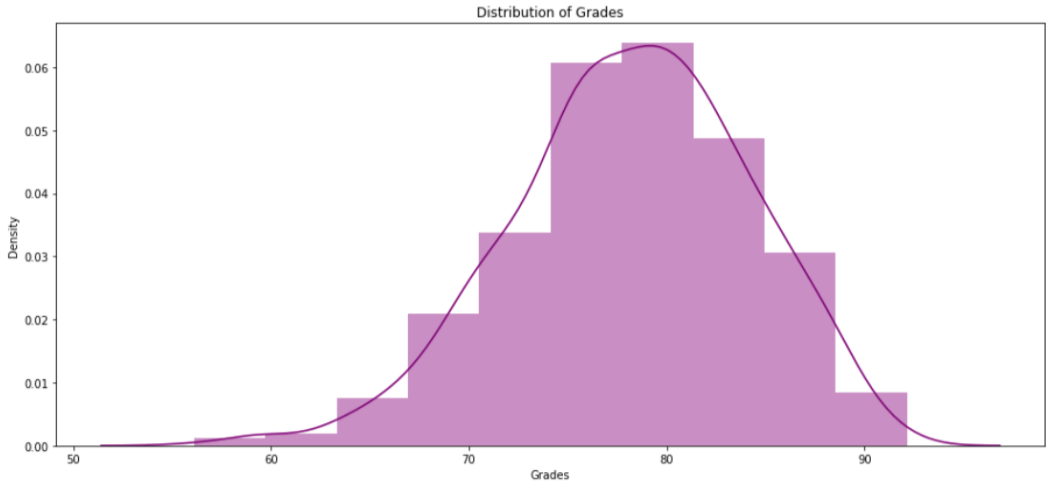


FIGURE A.2: Density Plot of Final Grades in QR

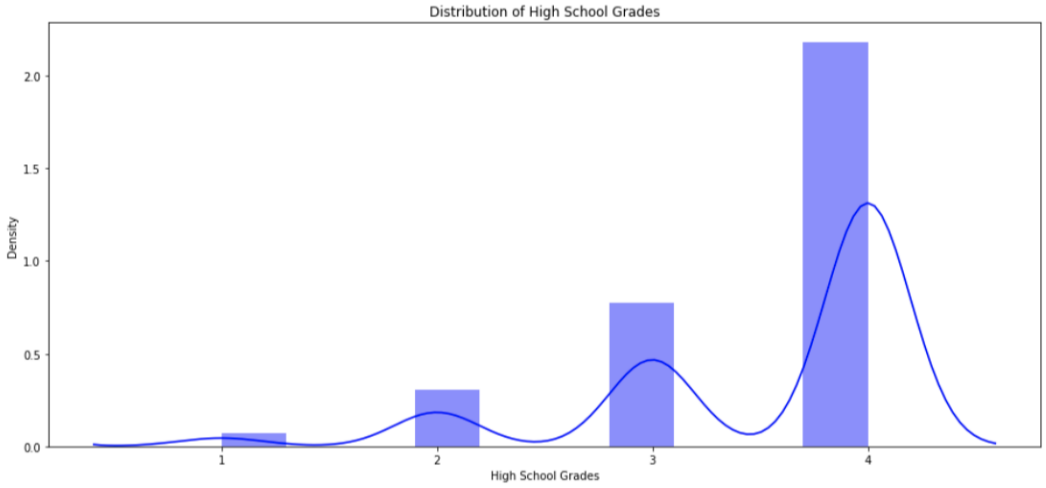


FIGURE A.3: Density Plot of High School Grades

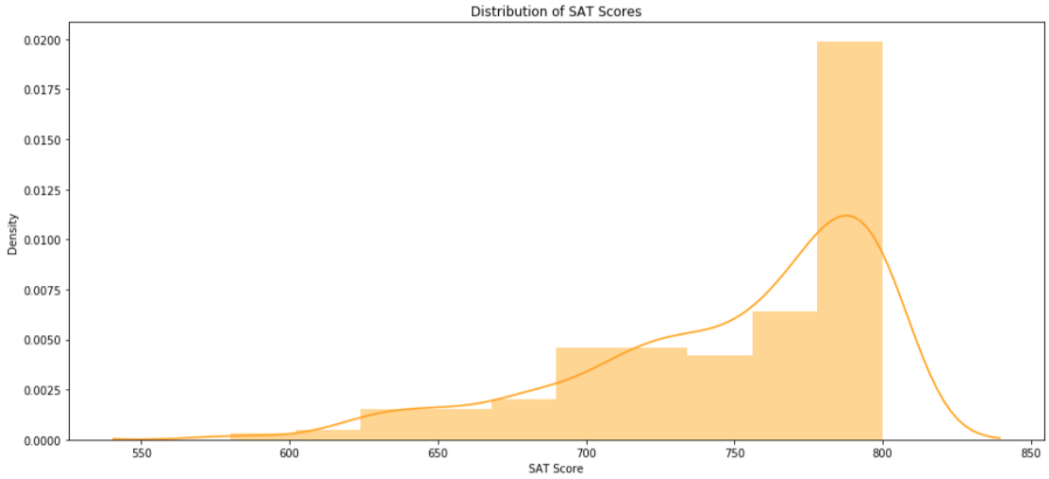


FIGURE A.4: Density Plot of SAT scores

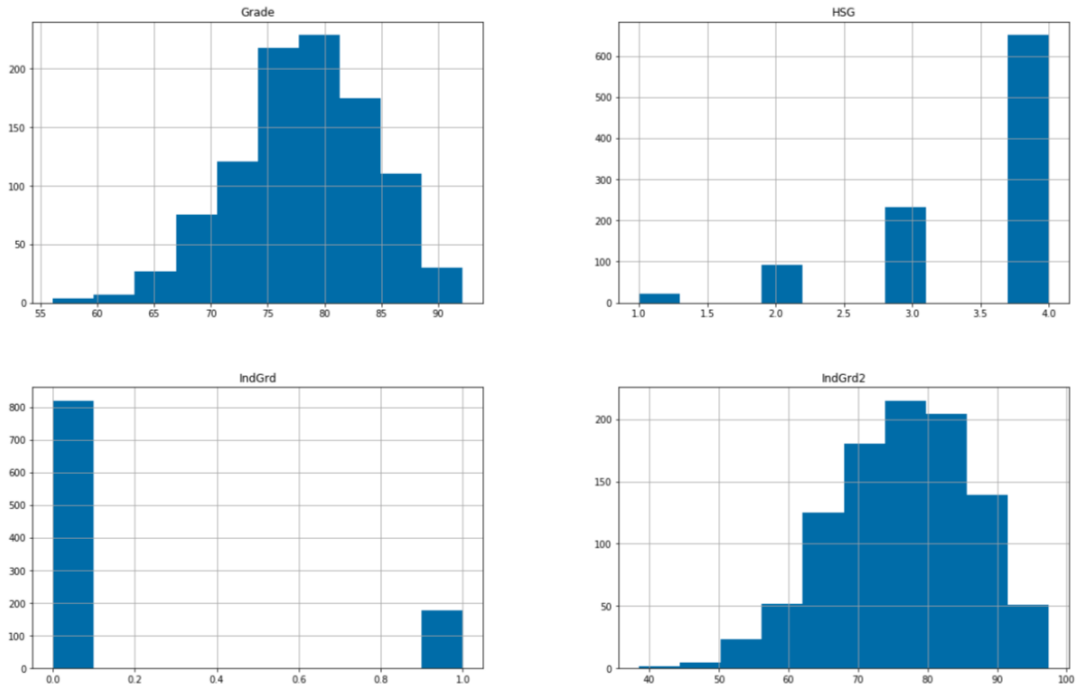


FIGURE A.5: Histogram of Numerical Data

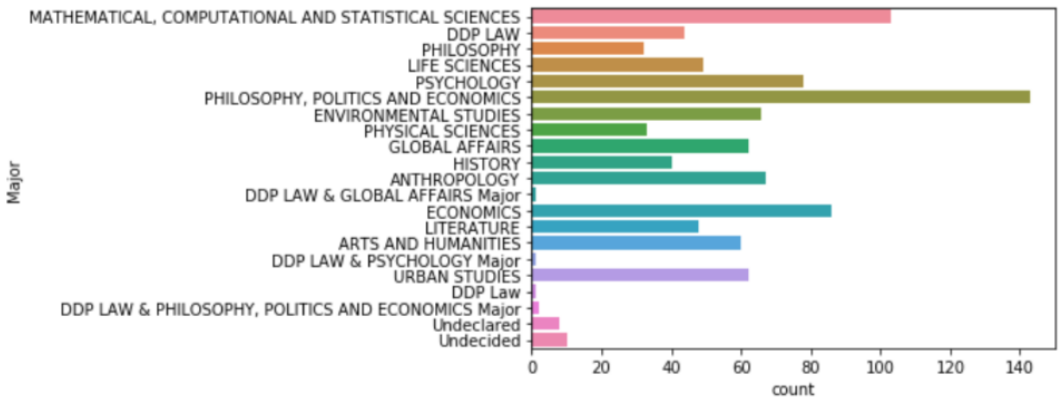


FIGURE A.6: Bar Plot of students declared major

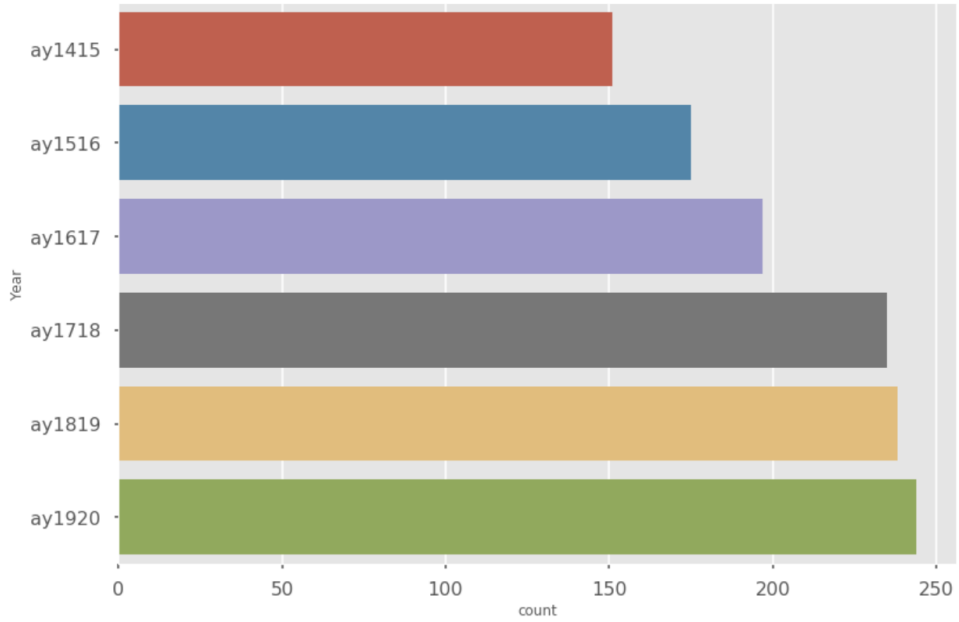


FIGURE A.7: Bar Plot of student batches. Batch size grew steadily from 2014 to 2019.

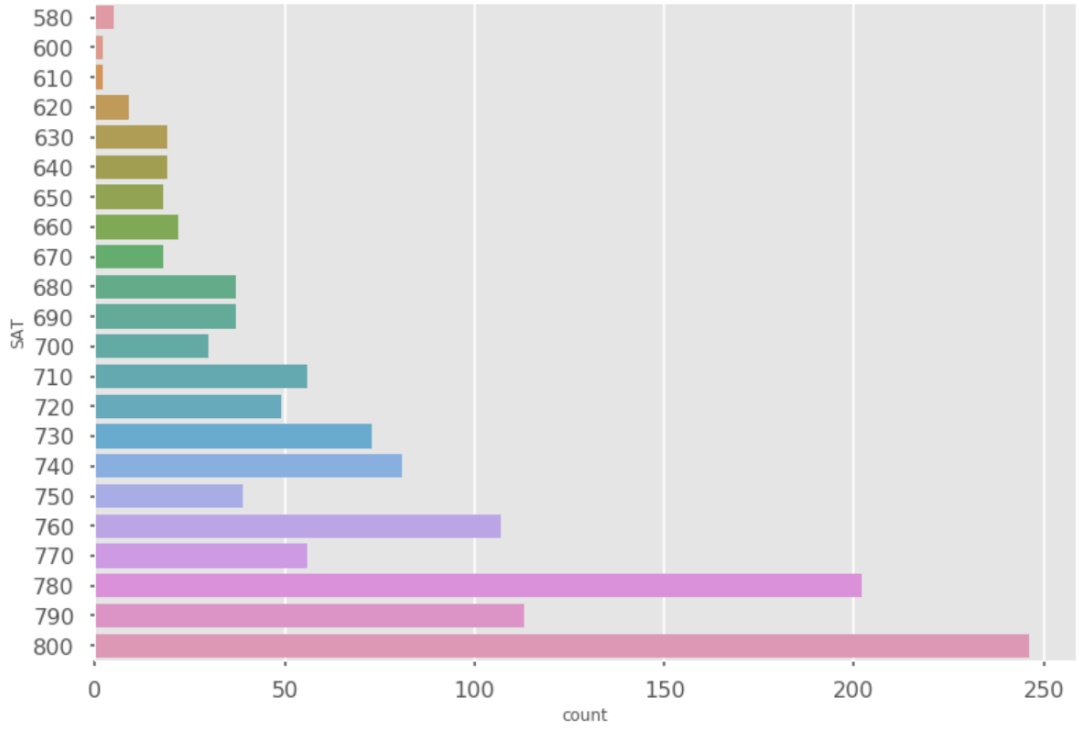


FIGURE A.8: Bar Plot of SAT scores

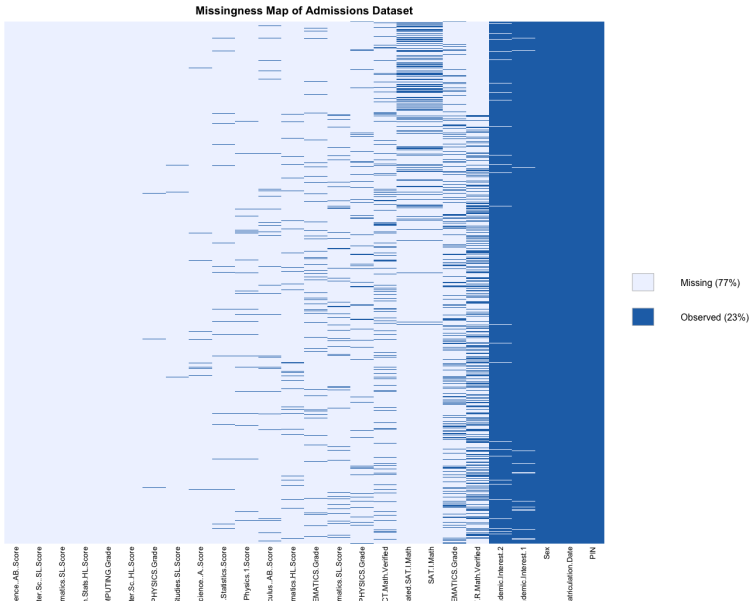


FIGURE A.9: Missingness Map of Admissions Data. 77% of data is missing.

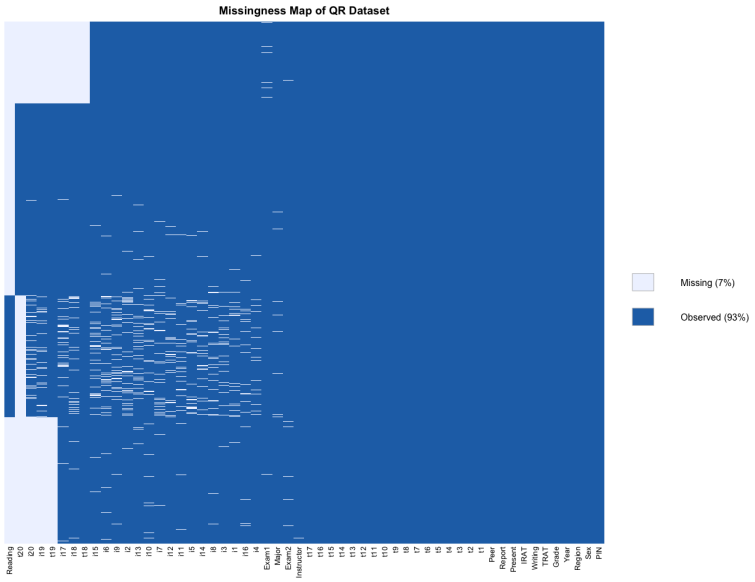


FIGURE A.10: Missingness Map of QR Data. Only 7% of data is missing.

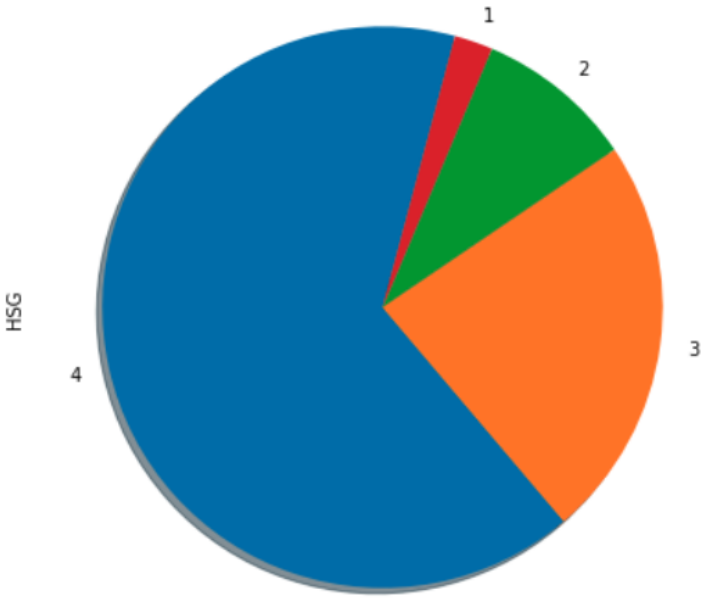


FIGURE A.11: Pie Chart for High School Grades

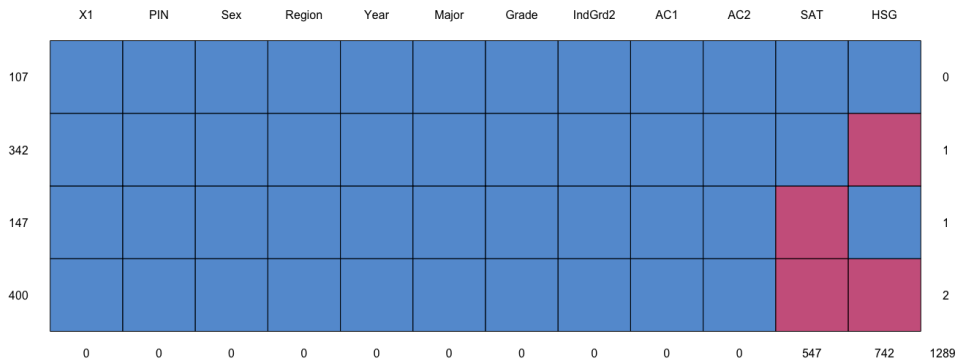


FIGURE A.12: Missing Data Pattern for HSG and SAT. There are 547 missing SAT values and 742 HSG values.

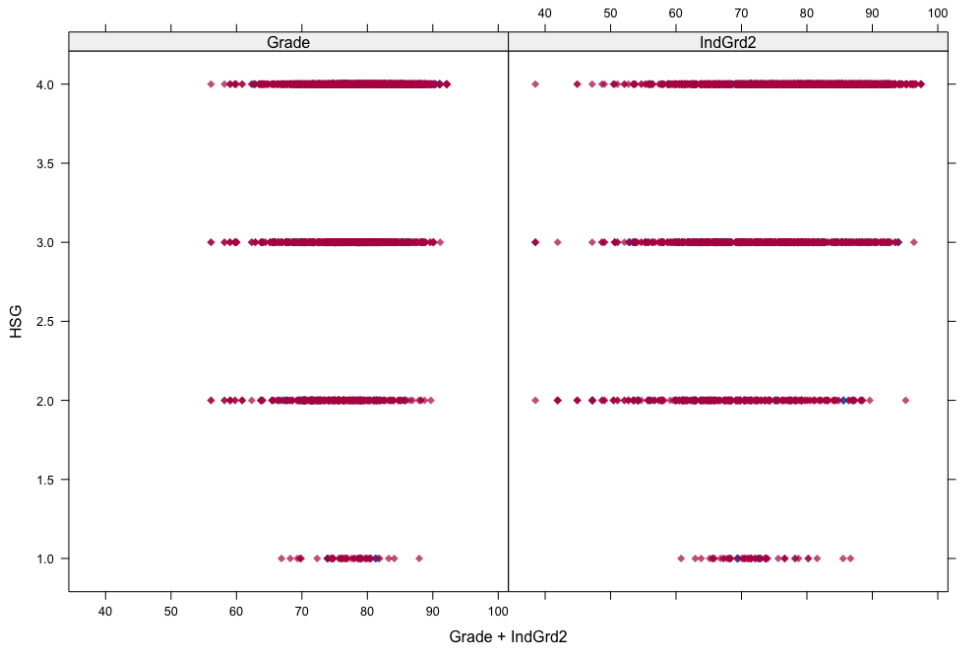


FIGURE A.13: High level Trellis function of HSG. The blue points are original data, the pink points are imputed data.

	Sex	Region	HSG	AC1	AC2	SAT	IndGrd
0	M	Southeast Asia	3	Philosophy, Politics and Economics	Undeclared	770	1.0
1	M	Singapore	4	Global Affairs	Psychology	680	0.0
2	F	Singapore	4	Undecided	Philosophy	730	0.0
3	F	Singapore	3	Philosophy, Politics and Economics	Psychology	790	0.0
4	F	Singapore	3	Philosophy, Politics and Economics	Economics	800	1.0
...	...	...	...	...	...	...	...
1235	M	Singapore	3	Urban Studies	Environmental Studies	770	0.0
1236	M	Singapore	4	Urban Studies	Arts & Humanities	750	0.0
1237	M	Singapore	4	Urban Studies	Undecided	780	0.0
1238	M	South Asia	3	Urban Studies	History	800	0.0
1239	F	Singapore	3	Urban Studies	Economics	780	1.0

1240 rows × 7 columns

FIGURE A.14: Final Dataset Cleaned and Imputed (1240 rows x 7 columns)

	HSG	SAT	IndGrd	Sex_F	Sex_M	Region_East Asia	Region_Europe	Region_North America	Region_Other	Region_Singapore	...	AC2_Life Sciences	AC2_Literature	AC2_Mati Computa
0	3	770	1.0	0	1	0	0	0	0	0	...	0	0	
1	4	680	0.0	0	1	0	0	0	0	1	...	0	0	
2	4	730	0.0	1	0	0	0	0	0	1	...	0	0	
3	3	790	0.0	1	0	0	0	0	0	1	...	0	0	
4	3	800	1.0	1	0	0	0	0	0	1	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1235	3	770	0.0	0	1	0	0	0	0	1	...	0	0	
1236	4	750	0.0	0	1	0	0	0	0	1	...	0	0	
1237	4	780	0.0	0	1	0	0	0	0	1	...	0	0	
1238	3	800	0.0	0	1	0	0	0	0	0	...	0	0	
1239	3	780	1.0	1	0	0	0	0	0	1	...	0	0	

1240 rows × 44 columns

FIGURE A.15: Final Dataset with One-Hot Encoding (truncated) (1240 rows x 44 columns)

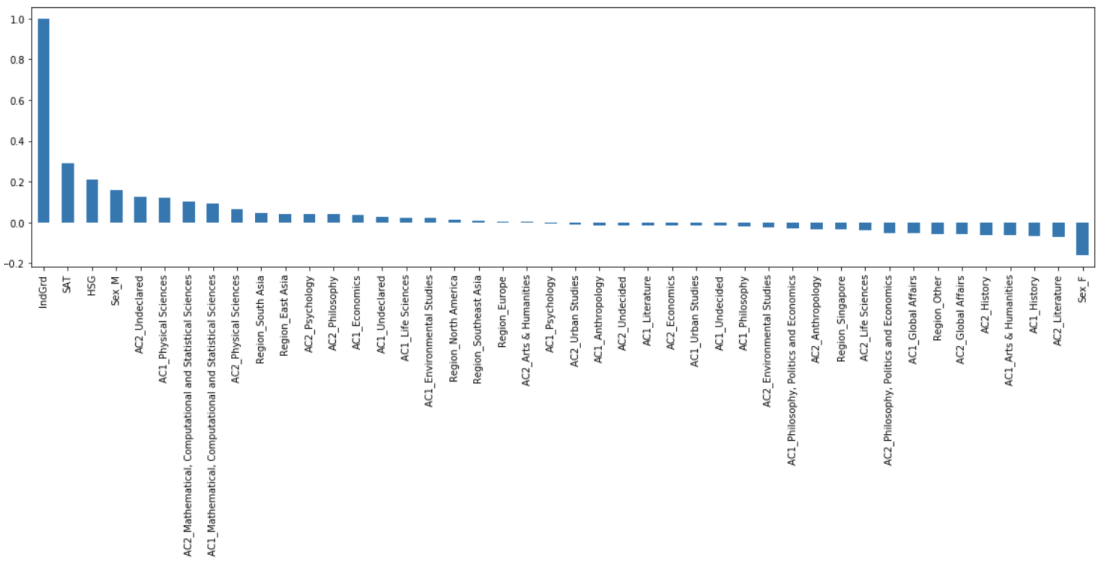


FIGURE A.16: Correlation plot of Individual Grades (IndGrd) with respect to all other dummy variables

	Est.	S.E.	t val.	p
(Intercept)	78.63	1.51	51.91	0.00
Academic.Interest.1_Anthropology	-3.09	1.91	-1.62	0.11
Academic.Interest.1_Arts & Humanities	-4.57	1.80	-2.54	0.01
Academic.Interest.1_Economics	1.21	1.73	0.70	0.48
Academic.Interest.1_Environmental Studies	1.54	2.00	0.77	0.44
Academic.Interest.1_Global Affairs	-3.99	1.72	-2.32	0.02
Academic.Interest.1_History	-4.08	2.10	-1.94	0.05
Academic.Interest.1_Life Sciences	-0.37	1.84	-0.20	0.84
Academic.Interest.1_Literature	-4.25	1.87	-2.27	0.02
Academic.Interest.1_Mathematical, Computational and Statistical Sciences	1.61	1.85	0.87	0.38
Academic.Interest.1_Philosophy	-1.32	2.52	-0.52	0.60
Academic.Interest.1_Philosophy, Politics and Economics	-2.42	1.65	-1.47	0.14
Academic.Interest.1_Physical Sciences	3.59	1.94	1.85	0.06
Academic.Interest.1_Psychology	-0.66	1.75	-0.38	0.71
Academic.Interest.1_Undecided	-1.47	2.13	-0.69	0.49
Academic.Interest.1_Urban Studies	NA	NA	NA	NA
Academic.Interest.1_NA	NA	NA	NA	NA

Standard errors: MLE

FIGURE A.17: Regression Results using Individual Grades (0-100 scale) as dependent variable, and Academic Interest 1 as independent variable. The statistically significant values are circled.

(Intercept)	77.46	1.61	47.99	0.00
Academic.Interest.2_Anthropology	-0.79	2.00	-0.40	0.69
Academic.Interest.2_Arts & Humanities	-2.57	1.84	-1.40	0.16
Academic.Interest.2_Economics	0.16	1.87	0.08	0.93
Academic.Interest.2_Environmental Studies	-0.29	2.19	-0.13	0.90
Academic.Interest.2_Global Affairs	-2.62	1.81	-1.45	0.15
Academic.Interest.2_History	0.60	2.25	0.27	0.79
Academic.Interest.2_Life Sciences	-0.30	2.16	-0.14	0.89
Academic.Interest.2_Literature	-3.85	2.02	-1.91	0.06
Academic.Interest.2_Mathematical, Computational and Statistical Sciences	5.00	2.08	2.40	0.02
Academic.Interest.2_Philosophy	0.28	2.35	0.12	0.91
Academic.Interest.2_Philosophy, Politics and Economics	-1.32	1.77	-0.75	0.46
Academic.Interest.2_Physical Sciences	0.80	1.98	0.40	0.69
Academic.Interest.2_Psychology	1.30	1.92	0.68	0.50
Academic.Interest.2_Undecided	-0.82	1.92	-0.43	0.67
Academic.Interest.2_Urban Studies	NA	NA	NA	NA
Academic.Interest.2_NA	NA	NA	NA	NA

Standard errors: MLE

FIGURE A.18: Regression Results using Individual Grades (0-100 scale) as dependent variable, and Academic Interest 2 as independent variable. The statistically significant values are circled.

	Est.	S.E.	t val.	p
(Intercept)	77.73	1.25	62.35	0.00
Region_East Asia	0.33	1.62	0.20	0.84
Region_Europe	-1.32	1.85	-0.71	0.48
Region_North America	-0.73	1.71	-0.43	0.67
Region_Other	-5.97	1.82	-3.29	0.00
Region_Singapore	-1.31	1.31	-1.00	0.32
Region_South Asia	-3.62	1.74	-2.08	0.04
Region_Southeast Asia	NA	NA	NA	NA
Region_NA	NA	NA	NA	NA

FIGURE A.19: Regression Results using Individual Grades (0-100 scale) as dependent variable, and region as independent variable. The statistically significant values are circled.

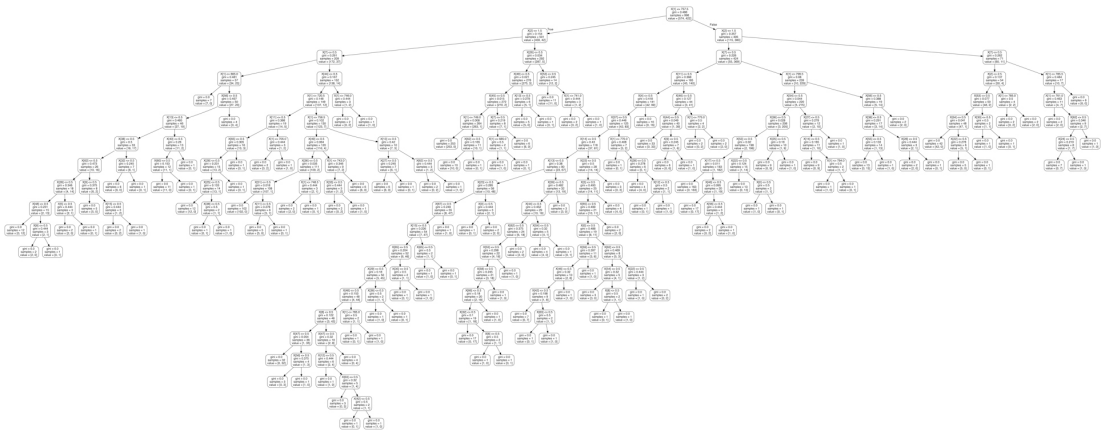


FIGURE A.20: Decision Tree Classifier with 607 nodes (depth 27)