OPTIMAL SALARY FORCASTING

By

Rinda Sai Kiran

Aayushi Bhattacharya

Monica Kulkarni

Vikram Arikath

Yashvardhan Rathi

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to INSOFE and Dr. Sridhar Pappu for his support and encouragement. We, as an academic group, would like to thank him for allowing us to work on data recorded by INSOFE'S consulting team. We would also like to thank Dr. Sourav Chatterjee for giving us this opportunity to take up this project.

INDEX

1	Introduction	2
2	Data Cleaning and Feature Engineering	8
3	Imputation	15
4	Data Exploration	27
5	Model Training and Evaluation	39
6	Conclusion	53
7	Appendix	54

LIST OF TABLES

1	Attributed and its description	3
2	Imputation methods: Performance for the Variable ‘Age’	22
3	Imputation methods: Performance for the Variable ‘Score.In.Domain’	24
4	Performance measures of the 3 algorithms without using PCA on Validation Data	42
5	Error matrix for the models trained using Principal Components	49

Introduction

"Many receive advice, only the wise profit from it."—Harper Lee

Data is an asset to any organization, regardless of the domain in which it specializes. A key ingredient for a company's success is how they make use of data. The need to implement a sustainable decision model for the growth of the company, based on feedback received from data gathered over time, cannot be overemphasized.

Organizations generally come up with a revenue forecast for a financial year. In order to reach the figures that were estimated in the forecast, these organizations plan a budget for each one of their sectors. For any organization, salaries paid to the employees constitute the biggest part of their expenditures. Behind every successful company is a dedicated and motivated workforce that delivers as and when required. Hence, it is imperative for a company to keep its employees satisfied, both with the work that is assigned and the pay that the employee deserves. While there is no dearth of work usually, it is the remunerations offered to the employees which are limited within a budget, and thus a company is often found wanting in this aspect[1].

A study conducted by Glassdoor Economic Research indicated that a *"10% higher base pay is associated with a 1.5-percentage-point increase in the likelihood that workers will stay at their current company the next time they move to a new role"* [2]which is a statistically significant link. Essentially, all of this points to the fact that the lack of deserving salaries leads to a high employee attrition rate - a HR nightmare.
To address this issue, effective budgeting for payrolls must be performed by taking into account the ***optimal figure*** that each individual deserves. The big question that looms now is this - ***What is an optimal pay and how to estimate this number***? This issue becomes all the more difficult to address considering that the same job title can be filled in by different individuals with relatively different experience levels and skill sets. One approach to tackle this problem would be to **train a salary prediction engine** based on past data.

This report deals with the analysis of one such dataset where information regarding an individual's demographics, academic background, soft skills, and technical

[1] "Effective Budgeting,"GPayroll,last modified Jul 12, 2016,https://www.gpayroll.com/blog/payroll/effective-budgeting-for-payroll.html.

[2] Andrew Chamberlain,"GoodPay,"last modified Mar 06,2017,https://hbr.org/2017/03/why-do-employees-stay-a-clear-career-path-and-good-pay-for-starters.

capacities has been recorded along with the salaries they receive, which they consider to be fair for the amount of work they put in and the skill-set they possess. Although this data comes from one country - India - alone, a consensus was reached that it is acceptable to conduct a generalized analysis in a region that boasts of a humongous work-force and a steadily increasing industrial presence, along with the fact that its population accounts for approximately 18 percent of the world's[3] - a very significant fraction.

This data consists of 34 variables including the annual salaries in Indian currency which is listed as "**Pay_in_INR".** Other details such as the individual's **Gender**, the **College from which they graduated**, their **native state** have been recorded. Of particular interest is a set of scores that appear in this list. These scores reflect the performance of an individual in a very popular test which quantify the technical and soft skills that are required in a professional environment. These scores are widely utilized by recruiters in India to filter the required candidates for a particular position. This analysis was performed with the understanding that these scores cannot be excluded from the forecasting engine. The model described has been trained and validated using RStudio. A full list of the variables can be found in the table below:

ATTRIBUTE NAME	DATA DESCRIPTION
Candidate ID	Unique ID number denoting each candidate.
Pay_In_INR	Salary of each candidate
Gender	Gender of the candidate
Date of Birth	The date of birth of the candidate in the yyyy-mm-dd hh : mm format
Score in Tenth	Marks obtained out of 100 in the 10^{th} standard

[3] Jason Burke,"India Census,"last modified May 31, 2011,https://www.huffingtonpost.com/2011/03/31/india-census-indian-population_n_843247.html.

School Board in Tenth	School board studied under in the 10th standard
Year of Twelfth Completion	Year in which the candidate completed 12th standard (High School)
Score in Twelfth	Marks obtained out of 100 in the 12th standard (High School)
Board in Twelfth	School board studied under in the 12th standard (High School)

Table I : Attributes and their Descriptions

ATTRIBUTE NAME	DATA DESCRIPTION
College Code	Codified identifiers for different colleges
College Tier	Level assigned to the college based on student performance and reviews
Graduation	Degree pursued by the candidate in college
Discipline	Major studied in college by the candidate
GPA Score in Graduation	Marks obtained out of 100 in college
City Code	Codified identifiers for different cities
City Tier	Level assigned to the city based on population
State	State in which the college in located
Year of Graduation Completion	Year in which the candidate completed graduation
Score in English language	Score quantifying the candidate's skill in English
Score in Logical skill	Score quantifying the candidate's skill in Logical reasoning

Table I : Attributes and their Descriptions (Contd.)

ATTRIBUTE NAME	DATA DESCRIPTION
Score in Quantitative ability	Score quantifying the candidate's skill in Quantitative ability
Score in Domain	Score quantifying the candidate's skill in the Domain that encompasses his job.
Score in Computer Programming	Score quantifying the candidate's skill in Computer Programming
Score in Electronics and Semicon	Score quantifying the candidate's skill in Electronics and Semiconductors concepts
Score in Computer Science	Score quantifying the candidate's skill in Computer Sciences
Score in Mechanical Engg	Score quantifying the candidate's skill in Mechanical Engineering concepts
Score in Electrical Engg	Score quantifying the candidate's skill in Electrical Engineering concepts
Score in Telecom Engg	Score quantifying the candidate's skill in Telecommunication Engineering concepts
Score in Civil Engg	Score quantifying skill in Civil Engineering concepts

Table I : Attributes and their Descriptions (Contd.)

ATTRIBUTE NAME	DATA DESCRIPTION
Score in conscientiousness	Score quantifying the candidate's conscientiousness
Score in agreeableness	Score quantifying the candidate's agreeableness
Score in extraversion	Score quantifying the candidate's extraversion
Score in neuroticism	Score quantifying the candidate's neuroticism
Score in openness to experience	Score quantifying the candidate's openness to experience

Table I : Attributes and their Descriptions (Contd.)

The dataset consists of 26415 observations for 34 features or variables. Using the **str()** function, it was observed that out of the 34 variables, 7 variables have been recorded as categorical variables and 27 variables have been recorded as either numeric or integer variables. However, there might be inconsistencies in the manner in which the variables are recorded and what they convey. Verifying whether the variables have been recorded in an acceptable manner and rectifying any errors, if present, constitute a part of **Data Cleaning and Feature Engineering.**

Data Cleaning and Feature Engineering

"80 percent of Data Science is cleaning and preparing data.
The other 20 percent is complaining about it."

Data, in its raw form, is difficult to comprehend. Before diving into the model training phase, it is necessary to gain at the very least, a top level understanding of the variables involved and the domain of analysis, sometimes referred to as 'Domain Knowledge'. This information allows an analyst to make informed decisions regarding whether or not to perform any modifications on the dataset, and whether such modifications retain the essence and information captured by the original data.

In the dataset chosen for this report, it was observed that the variable **"Date.Of.Birth"** has been recorded as a categorical variable with **2639 unique values**. Along with the date, this variable also recorded the time of birth. Since the time of birth has no relevance in estimating the salary, it was removed. The function, **str_sub()** was used to split the dates and times, after converting the variable into a character type. It was observed that the separated 'Time' had no unique values with all entries being "00:00", providing enough reason to eliminate it.

Since it is unnatural to have a categorical variable with 2500 levels, The remaining date column, 'D.O.B' was converted to date type and the year alone was extracted. This year was then compared with the current year extracted using the **Sys.Date()** function to obtain the **"Age"** variable. The intermediate columns were eliminated.

Date.Of.Birth
1990-06-13 00:00
1992-10-08 00:00
1991-09-17 00:00

```
data$Date.Of.Birth = as.character(data$Date.Of.Birth)
data$D.O.B = str_sub(data$Date.Of.Birth, 1,-6)
data$Birth.Time = str_sub(data$Date.Of.Birth, start = -5)

data$D.O.B = as_date(data$D.O.B)

data$Y.O.B = year(data$D.O.B)
x = format(Sys.Date(), "%Y")

data$Age = as.numeric(x) - data$Y.O.B
```

Age
28
26
27

Fig i : Transformation from Date.Of.Birth to Age

It was noted that a few observations had times specified but no dates, in the Date.Of.Birth column. Upon converting to Age, these observations assumed an **'NA'** level. Usually, missing values are codified as **NA** in R indicating that the variable **Age** had missing values.

In the variables **"Board.in.Twelfth"** and **"School.Board.in.Tenth"**, both of which are recorded as categorical variables, it was observed that there is a level specified as "**0**" occuring in a number of observations. This essentially denotes that the entry for these variables is missing. These "0" levels were transformed to NA. Similarly, the variable **"Score.in.Domain"**, a numeric variable which was standardized before being recorded, had a few negative values all of which were **-1**. This variable comes from a popular test in India which does not assign any negative scores, with zero being the minimum. These negative values were converted to missing values. All the missing values in the dataset are spread **across 6344 rows** and **total up to 11072 instances.**

```
levels(data$Board.in.Twelth)[levels(data$Board.in.Twelth)=="0"] = NA
levels(data$School.Board.in.Tenth)[levels(data$School.Board.in.Tenth)=="0"] = NA
data$Score.in.Domain[data$Score.in.Domain == -1] = NA
```

Fig ii : Converting absent entries to NA

Few variables were recorded under the incorrect data types. "**CollegeTier"** and "**CityTier"** are actually categorical variables which assume numeric values. However, these numerals specify a particular factor or level. For CollegeTier, the values were observed to be 1 and 2 with '1' indicating a higher level and '2' indicating a relatively lower level. Similarly for CityTier, these values were 0 and 1 with '0' indicating a tier lower than '1'. Both these **variables were listed as numeric variables** and were **converted** using the **as.categorical()** function.

From the variables **"Year.of.Graduation.Completion"** and **"Year.Of.Twelfth.Completion"**, both of which were integer data types, a new variable, **"Gap_in_Education"** was derived by considering the difference between the year of graduation completion and year of twelfth completion. Along with this numeric variable, a categorical variable **"Gap_Scenario"** was introduced which cross references the "Gap_in_Education" with the existing variable **"Graduation"**. The "Graduation" variable contains the information regarding each individual's degree. A categorical variable, it assumes 3 values - 'B.Tech/B.E.' which generally

take 4 years after high school (twelfth) to be earned; 'M.Tech/M.E.' which generally take 6 years after high school; 'MCA' which usually takes between 5 and 6 years after high school. The variable "Gap_in_Education" and the "Graduation" variables were compared and based on the ideal scenarios mentioned above, **"Gap_Scenario"** was assigned two levels - 'ideal' and 'non-ideal'.

In **Fig iii(a)**, the new variables along with their parent variables have been depicted. With a gap in education of 5 years for a B.tech/B.E. degree, the fourth observation was assigned a 'non-ideal' level under "Gap_Scenario".

Year.Of.Twelth.Completion	Year.of.Graduation.Completion	Gap_in_education	Graduation	Gap_Scenario
2007	2011	4	B.Tech/B.E.	ideal
2009	2013	4	B.Tech/B.E.	ideal
2010	2014	4	B.Tech/B.E.	ideal
2008	2013	5	B.Tech/B.E.	non-ideal

Fig iii(a) : Derived variables and their values based on the Parent variables

```
data$Gap_in_education = data$Year.of.Graduation.Completion - data$Year.Of.Twelth.Completion
x = c(5,6)
data$Gap_Scenario = ifelse(data$Graduation == "B.Tech/B.E." & data$Gap_in_education == 4, "ideal",
                           ifelse(data$Graduation == "M.Tech./M.E." & data$Gap_in_education == 6,"ideal",
                                  ifelse(data$Graduation == "MCA" & data$Gap_in_education %in% x,"ideal","non-ideal")))

data$Gap_Scenario = as.factor(data$Gap_Scenario)
```

Fig iii(b) : Code that derives the new variables Gap_in_Education and Gap_Scenario

For the numeric variables **"Score.in.CivilEngg"**, **"Score.in.TelecomEngg"**, **"Score.in.ElectricalEngg"**, **"Score.in.MechanicalEngg"**, **"Score.in.Comp.Science"**, **"Score.in.ElectronicsAndSemicon"**, **"Score.in.ComputerProgramming"**, the values reflect scores assigned to each individual's technical capacities in various fields. The test from which these scores were collected makes these fields optional for appearing candidates, meaning that not all these fields have to be tested. As such, there are numerous candidates surveyed for the dataset who were not tested in a majority of these fields, depending on the job position they were applying for. That is, *the fields in which candidates were tested can differ from one observation to the other*.However, while

recording this data, the fields in which a candidate was not tested were filled in with a value of **'-100'**. Since these do not indicate missing values, they cannot be replaced with NA's. These are indicators that this field was not a part of the test or they were not opted for.

To provide this information to the forecasting engine, **7 new variables were introduced**. **Each new variable corresponds to one of the optional fields**. These 7 categorical variables are binary variables that assume the value '0' wherever a -100 occurs in the score column, and a value of '1' elsewhere.

```
data$CP.opt = as.factor(ifelse(data$Score.in.ComputerProgramming == -100, '0','1'))
data$ESem.opt = as.factor(ifelse(data$Score.in.ElectronicsAndSemicon == -100, '0','1'))
data$CS.opt = as.factor(ifelse(data$Score.in.ComputerScience == -100, '0','1'))
data$Mech.opt = as.factor(ifelse(data$Score.in.MechanicalEngg == -100, '0','1'))
```

Fig iv(a) : Code that introduces new variables conveying information about the field status

Score.in.ComputerScience	CS.opt	Score.in.ElectronicsAndSemicon	ESem.opt	Score.in.ComputerProgramming	CP.opt
-100	0	366	1	-100	0
376	1	-100	0	335	1
-100	0	196	1	385	1
-100	0	-100	0	365	1
125	1	-100	0	525	1

Fig iv(b) : Depiction of a sample of Score variables and their corresponding 'opted' variables

After introducing the 7 'opted' variables, the entries of -100 in the Score variables were set to 0 in order to avoid needless and inaccurate variation within the variable.

It was noted that the variables **"School.Board.in.Tenth"** and **"Board.in.Twelfth"**, both categorical, had an unnaturally high number of levels. "Board.in.Twelfth" had 329 unique levels while "School.Board.in.Tenth" had 262 unique levels. Upon inspection, it was observed that most of the levels represented the same Educational Board. However, they were addressed differently leading to the unusually high number of levels. Some entries listed the name of the institution instead of the board to which they are affiliated. This further contributed to the high number of levels. Each state in India has its own educational board. Apart from these, there exist a few common Educational Boards with which many institutions

across the country have an affiliation. **These unique levels were therefore grouped together based on the state to which they belong**. Attempts were made to match entries with names of institutes instead of boards to their respective states. **The entries that could not be grouped under any state's board or national board were separately placed into a level named 'Other'**. The number of **levels**, after such grouping, reduced **from 329 to 24 and 262 to 23** for "Board.in.twelfth" and "School.Board.in.Tenth" respectively.

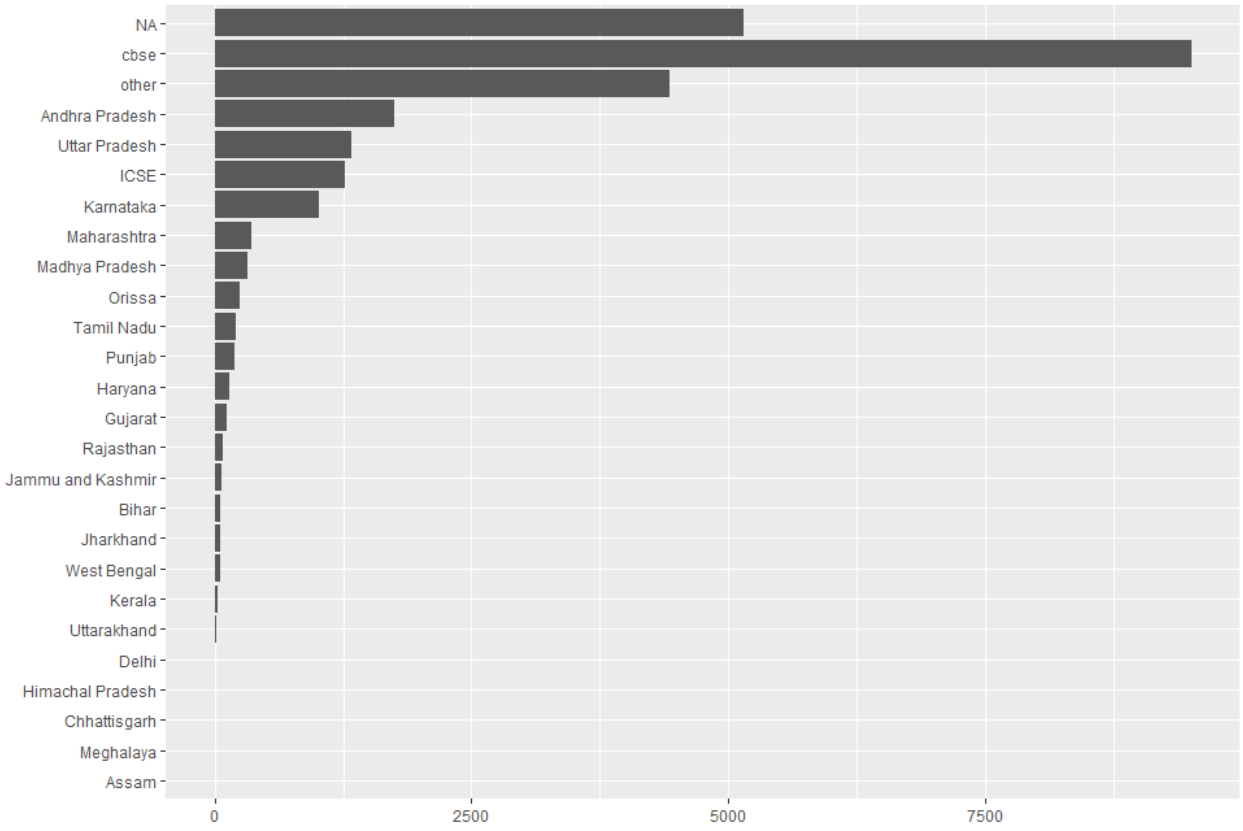


Fig v(a) : Bar Plot showing various levels in the variable "Board.in.Twelfth" and their frequencies after combining levels. The bar plot for the variable "School.Board.in.Tenth exhibits a similar trend.

A similar issue was observed in the variable **"Discipline"** which listed the various majors that were chosen by individuals. These fields of study could be grouped into a single category in a broad sense. Levels such as 'Computer Engineering', and 'Computer Networking' were grouped under the broad category of 'Computer Sciences' and so on. **This led to 46 different levels present in "Discipline" being reduced to 14 unique levels**.

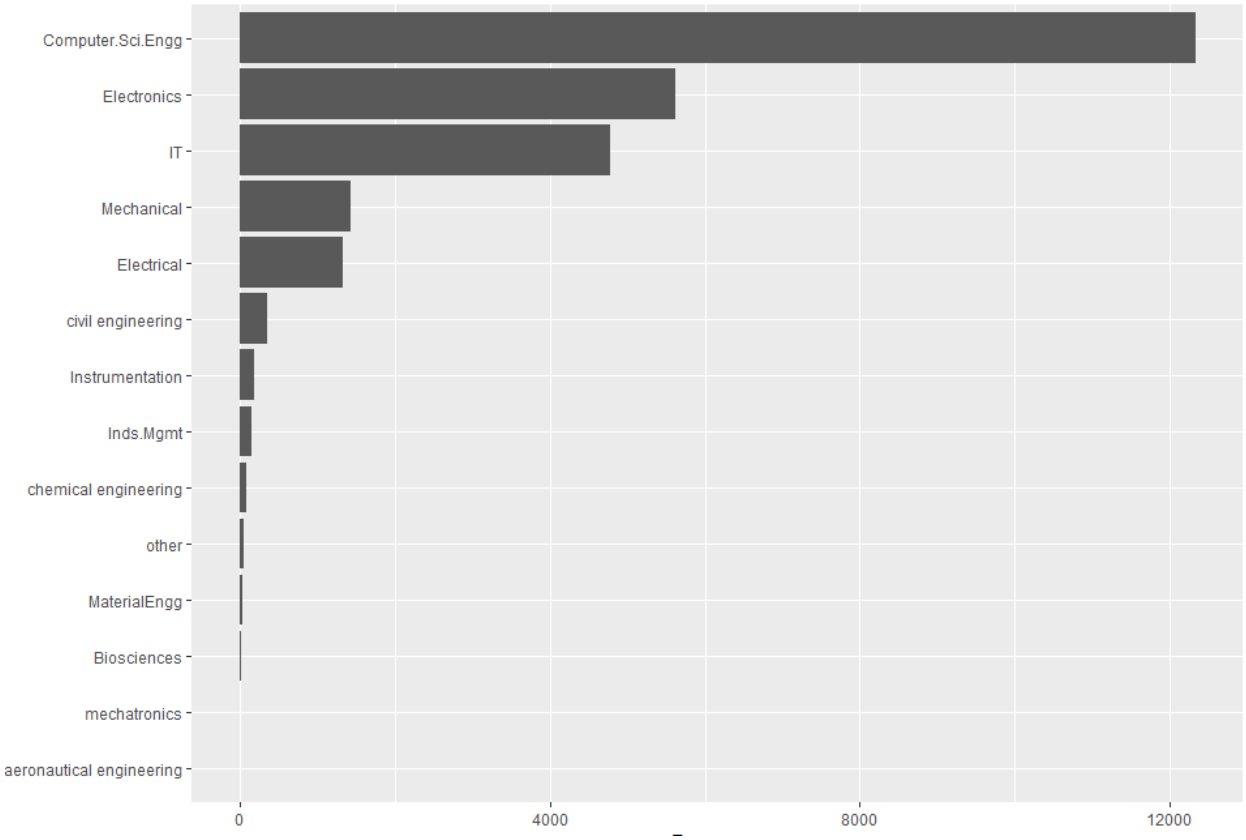


Fig v(b) : Bar Plot depicting unique levels in the variable "Discipline" and their frequencies after combining levels

Although the variable **"State"** has many levels, the premise on which it needs to be grouped remains unclear. It was suggested to group states based on region. However, such an approach would be meaningless. The issue being addressed is how best to predict optimal salaries, which might be affected by a state's economy. Grouping based on region is not proper, given that states in the same region can have vastly different economies. Case in point Tamil Nadu and Kerala[4]. Economy itself is affected by various factors. As such, arriving at a premise to group states was unsuccessful.

After examining the variables, a few instances were identified with missing values. These NA's were found in **"Age"**, **"Board.in.Twelfth"**, **"School.Board.in.Tenth"**, and **"Score.in.Domain"**.

4 "Comparing Indian," StatisticsTimes,accessed Nov 26, 2018,https://statisticstimes.com/economy/comparing-indian-states-and-countries-by-gdp.php.

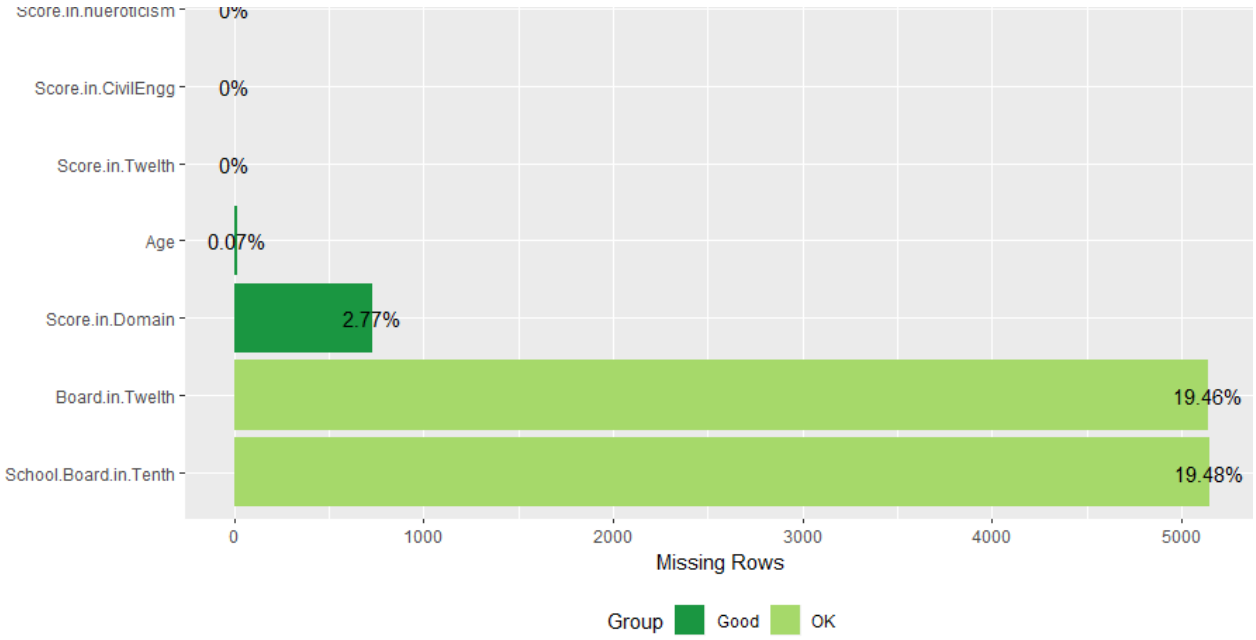


Fig vi : Percentage of rows with missing values under each column. Such a graph for all the variables in the dataset can be plotted using the plot_missing() function.

The various approaches used to estimate values for these missing instances constitute **Imputation.**

Imputation

There can exist 3 different types of missing values - Missing Completely at Random (MCAR), Missing at Random (MAR), Not Missing at Random (NMAR). MCAR type is based on the assumption that the missing data is "completely unrelated to the other information in the dataset". That is, data is missing purely because of some random phenomenon. MAR assumes that the missing data is somehow related to the rest of the information available and can be predicted by utilizing this information. NMAR is based on the idea that information was willfully withheld and no standard method used to handle missing data would have a proper effect.[5]

The data in this report was collected based on an unbiased survey. Hence, it can be deduced that the missing values in this dataset fall under the Missing at Random (MAR) category. As such, they were estimated using predictive methods that make use of the rest of the information available in the dataset. Specifically, 4 different approaches were tested and the best one was chosen. The four different approaches include **Central Imputation, KNN Imputation, Tree Imputation,** and **Imputation using the MissForest Package**. Apart from these 4 approaches, a **direct substitution based on the results of an independence test** was utilized.

The categorical variables "School.Board.in.Tenth" and "Board.in.Twelfth" had a high number of levels despite combining and grouping levels previously. As such, it was difficult to train a model that could come up with one of the existing 24 odd levels in place of the missing entries. Since the variable "State" seemed to have a huge bearing upon both these variables, It was proposed that the missing values in these variables can be imputed using the values present in the "State" variable.

In order to verify the association between the variables "State" and "School.Board.in.Tenth", a **Chi square test of independence** was performed. This method was chosen over the other tests of independence owing to the large size of data. A Chi square test is a hypothesis testing method used to test the statistical significance of the observed relationship with respect to the expected relationship. The null hypothesis is the assumption that there is no association between the two variables being considered for the study. Based on the probability

[5] Tim Bock, "What are,"Displayr,accessed Nov 26, 2018, https://www.displayr.com/different-types-of-missing-data/.

of occurrence of the value of test statistic obtained[6], The null hypothesis is either rejected or cannot be rejected.

Using the **chisq.test()** function, it was noticed that with a **p-value of 0.0004998**, sufficient evidence was present to **reject the null hypothesis of independence**. A correlation plot, depicted in **Fig vii**, between the levels of the 2 variables threw light on some interesting observations:

- Individuals belonging to a particular state seem to enrol themselves in educational institutes that are affiliated to the state's educational board
- Individuals from West Bengal seem to prefer enrolling with Jharkhand's educational board instead of West Bengal's.
- Tamil Nadu and Andhra Pradesh seem to have a higher number of institutes whose affiliation cannot be determined, suggesting the possibility that they might be unrecognized.

6 "Chi Square Test,"Statistics Solutions,accessed Nov 26, 2018,https://www.statisticssolutions.com/chi-square-test/.

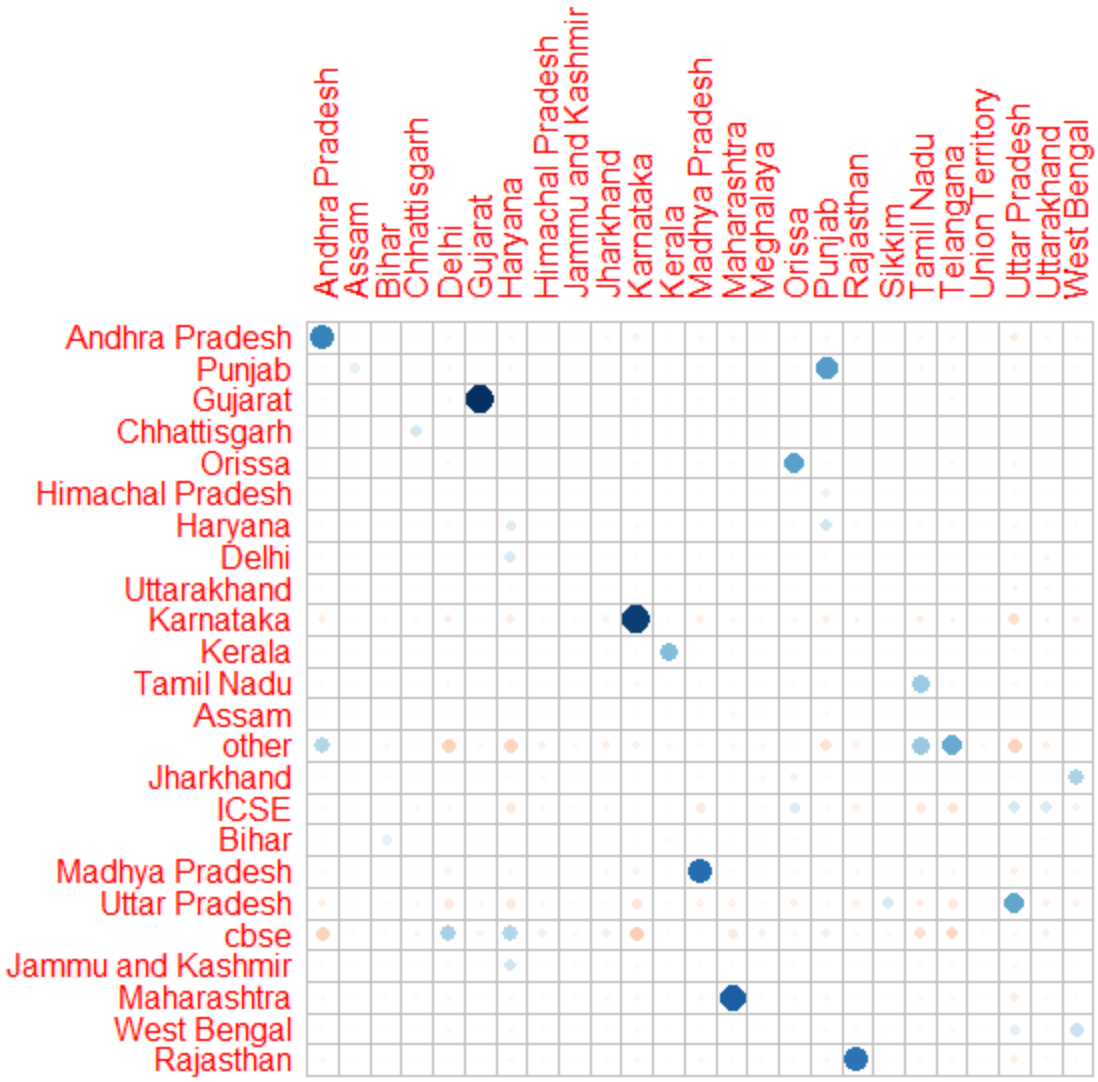


Fig vii(a) : Correlation plot between the levels of State (HORIZONTAL AXIS) and the levels of School.Board.in.Tenth (VERTICAL AXIS)

It must be noted that all the observations drawn from the correlation plot were merely suggestive and are not claimed to be facts. Causation cannot be inferred from Correlation. These correlations merely provide a basis for further investigation.

Based on the results of the chi square test as well as the common knowledge that most, if not all, individuals generally enroll themselves in institutes affiliated to their native state's board, the missing values in "School.Board.in.Tenth" were filled in using the corresponding "State" variable's values if the value appeared in the levels of "School.Board.in.Tenth".

```
xc = chisq.test(imputers$School.Board.in.Tenth, imputers$State, simulate.p.value = T)
xc
corrplot(xc$residuals, is.corr = F)
#data:  imputers$School.Board.in.Tenth and imputers$State
#Chi-squared = 47953, df = NA, p-value = 0.0004998
```

Fig vii(b) : Code used to perform the Chi square test in R and the corresponding output.

```
for(i in 1:nrow(imputers_miss)){
  if((imputers_miss$State[i] %in% levels(data$School.Board.in.Tenth)) &
     (is.na(imputers_miss[i,"School.Board.in.Tenth"]) == T)){
  imputers_miss$School.Board.in.Tenth[i] = imputers_miss$State[i]
  }
  if((imputers_miss$State[i] %!in% levels(data$School.Board.in.Tenth)) &
     (is.na(imputers_miss[i,"School.Board.in.Tenth"]) == T)){
  imputers_miss$School.Board.in.Tenth[i] = "other"
  }
}
```

Fig vii(c) : Code used to impute School.Board.in.Tenth with entries in State variable.

Similar chi square tests were performed for the variables “School.Board.in.Tenth” and “Board.in.Twelfth” as well as “Board.in.Twelfth” and “State”. It was observed that both these tests established the interdependency of these 3 variables. Hence, the values of “School.Board.in.Tenth” were used to fill in the missing values in “Board.in.Twelfth”. A correlation plot, shown in **Fig viii**, between the levels of these 2 variables was referred to before such an imputation.

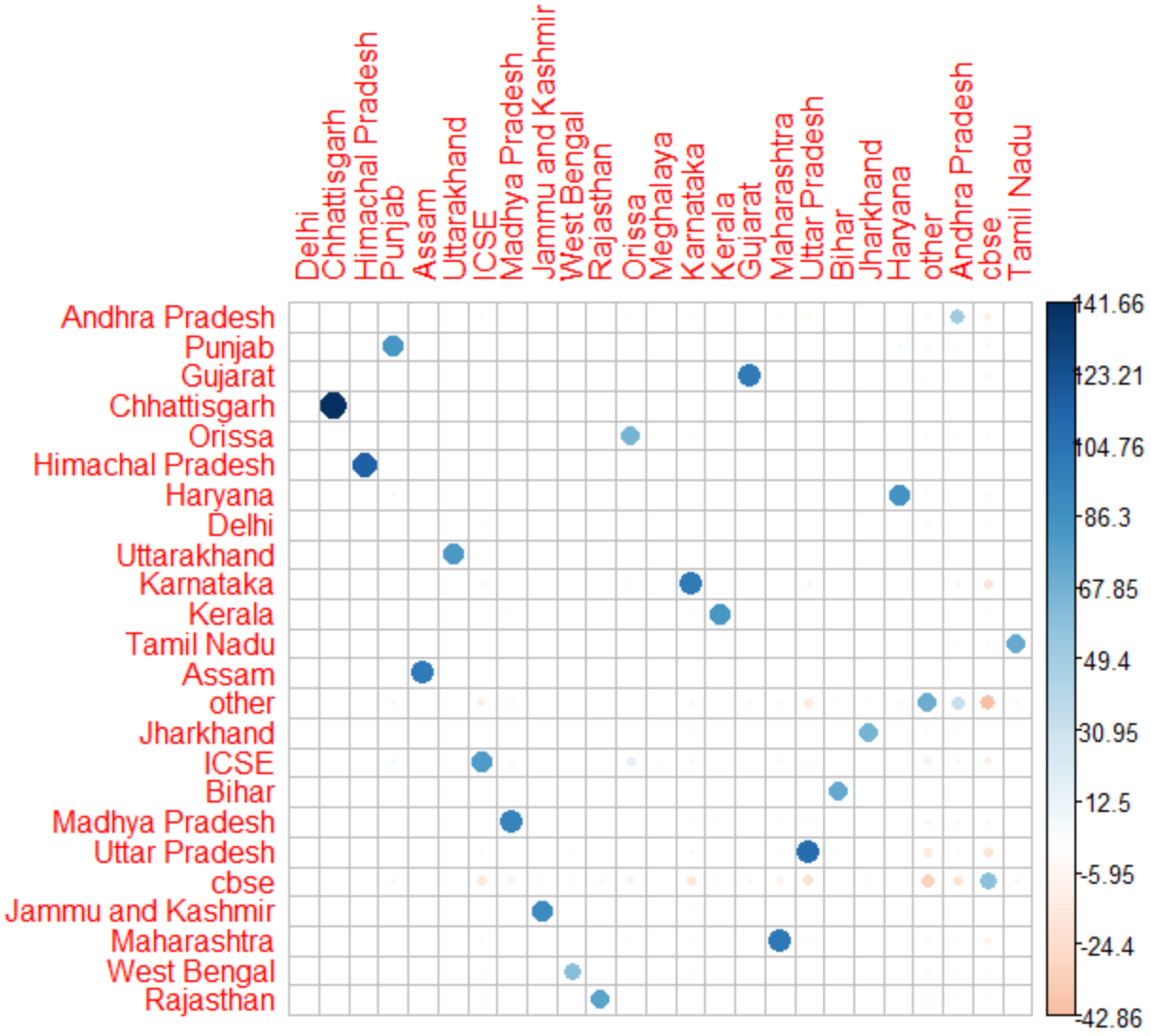


Fig viii : Correlation plot between levels of Board.in.Twelfth (HORIZONTAL AXIS) and School.Board.in.Tenth(VERTICAL AXIS)

It was observed that there is a strong correlation between the corresponding levels of these 2 variables. Meaning that, if an individual's educational institute was affiliated to a particular board in the tenth standard, it seems more likely that he/she would continue in institutes affiliated to the same board in their twelfth.

It was initially proposed that using standard imputation procedures which generally use a central value to fill in the missing values can be used for these 2 variables. However, it was decided that such an approach would be inaccurate. Given that these are categorical variables, the missing values would be filled using the **mode**

of the column. Given that most of the observations assumed either ICSE or CBSE, the missing instances would be imputed with these entries, increasing the already high number of instances of these entries and creating a bias. The values of the State variable were used to avoid this.

Having imputed the 2 categorical variables, quite a few imputation techniques were considered to fill in the continuous variables. **KNN Imputation** and **Central Imputation** were utilized and their results were compared. KNN imputation is a non-parametric method which fills in the missing values in an observation based on observations that are **closest** to it. Whether or not two observations are "close" to each other, or similar to each other, is decided by using various distance measures such as **euclidean distance** or **manhattan distance** which are commonly used for continuous variables.[7] The K in KNN refers to the number of such Nearest Neighbours to be considered while imputing. Either a **weighted average**, **median**, **mode** or **mean** of the values of these nearest neighbours is determined which is used to impute the missing values. It is essential to standardize the numeric variables before using KNN. The **knnImputation()** function in R, by default, standardizes the numeric variables supplied in the dataset. **Central Imputation** is a primitive imputation technique which uses a central value to plug in the missing instances. The central value can be mean, median or mode for continuous variables and mode for categorical variables.
The **MissForest** package, in simple terms, builds a random forest for each variable. **Random Forest** is an ensemble technique used to predict or classify data. At this point, it is sufficient to know that Random Forest is a machine learning algorithm. This will be further discussed and elaborated upon in the **Model Training and Evaluation** part.

The MissForest package trains a Random Forest for every variable using every other variable in the dataset as predictors. However, owing to the sizeable number of columns in the dataset coupled with the fact that some of the categorical variables have more than 10 levels, it was deemed computationally expensive and time consuming.

It was then proposed to train individual Random Forests for the 2 continuous variables with missing values. Similarly, individual Decision Trees, which is another

[7] Deepanshu Bhalla,"K Nearest," last modified Dec 18, 2017, https://www.r-bloggers.com/k-nearest-neighbor-step-by-step-tutorial/.

machine learning algorithm, were trained for these 2 variables, an idea directly based on the **Tree Imputation method[8].**

All the observations that had no missing values for Age and Score.in.Domain were subsetted using the **complete.cases()** function and NA's were introduced in the 2 variables for this subset using the **prodNA()** function. Individual imputation methods were tested on these artificially introduced NA's and the results were compared to pick the best method. Additionally, **while training the random forests and decision trees, the target variable and the other variable with missing values were not used.**

The performance of each imputation method was captured by plotting the actual and predicted variables. The error metric "**Mean Absolute Percentage Error"** was also considered. MAPE is obtained by calculating the average of the absolute value of the difference between actual and predicted values divided by the actual value, and expressing this number in percentage. The **lower this metric, better is the model**.
The plots for these models depict their performance. If the predicted values fall closer to the actual values, the **points congregate around the 45° line cutting the plot.**

Imputation Method	MAE	MSE	RMSE	MAPE
KNN	0.258970359	0.375975039	0.613168035	**0.009108568**
Central Imp.	1.32917317	2.85881435	1.69080287	**0.04624213**
Decision Tree	0.59126365	0.76287051	0.87342459	**0.02082801**
Random Forest	0.170826833	0.208268331	0.456364252	**0.006040812**

[8] Peerapon Vateekul, Kanoksri Sarinnapakorn, "Tree-Based Approach,"IEEE Xplore, last modified Dec 28, 2009,https://ieeexplore.ieee.org/document/5360523.

Table ii : Imputation methods' performance for the variable 'Age'

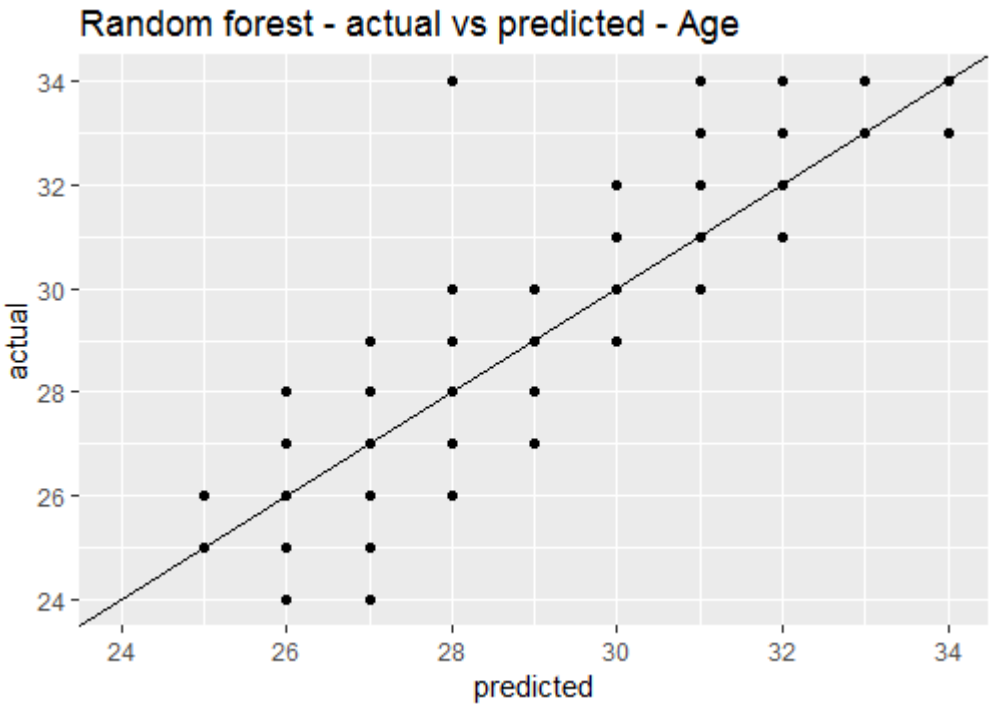


Fig ix(a & b) : Plot between Actual & Predicted Age. The points congregate around the 45° line for the Random Forest (above; ix(a)), indicating a good fit while the points are spread out for the Decision tree(below ; ix(b)), indicating a bad fit.

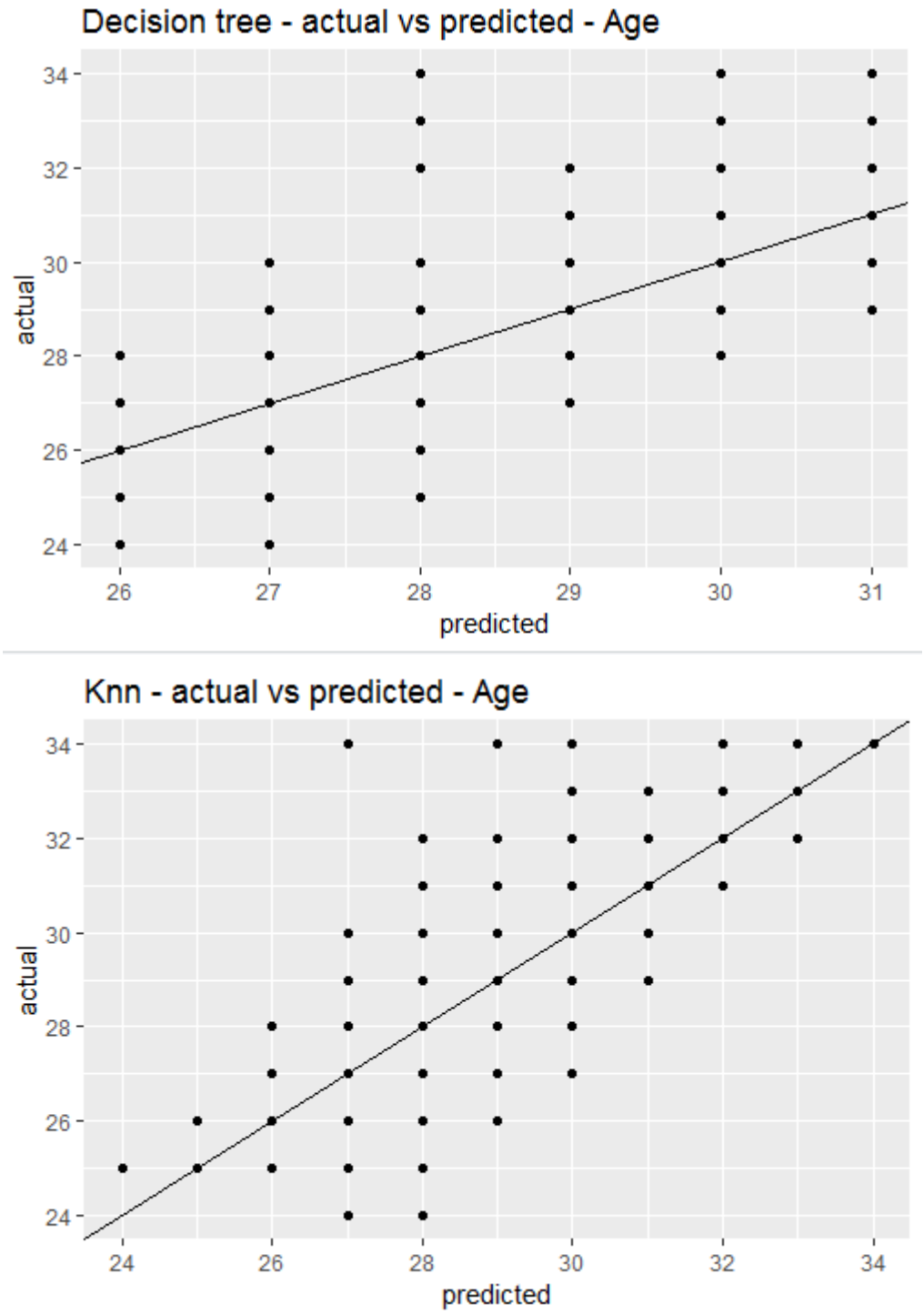
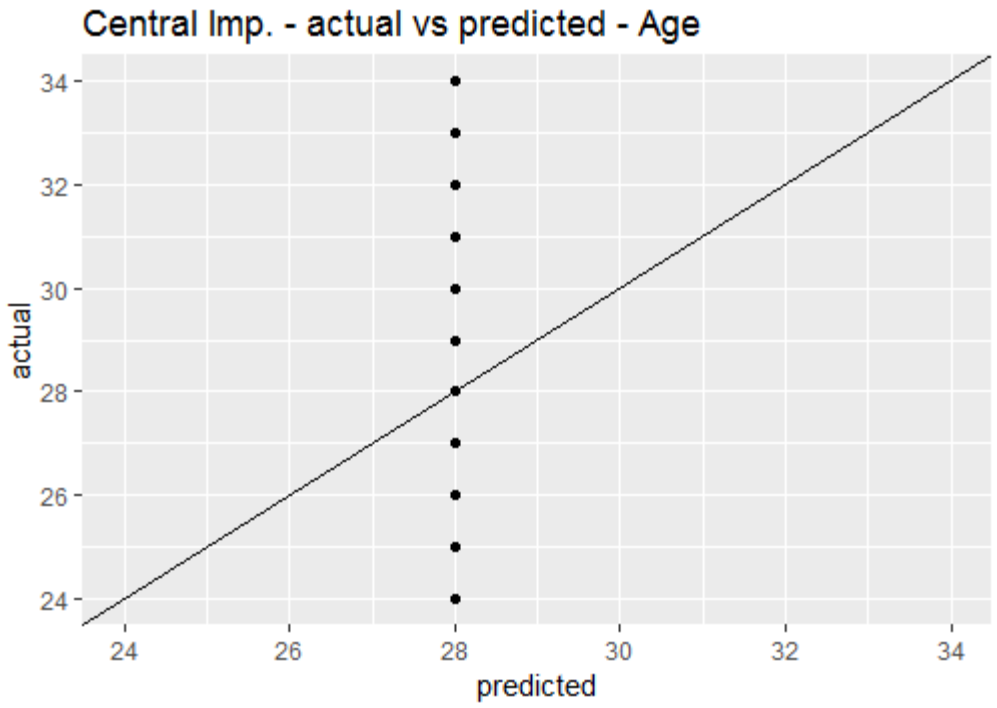


Fig ix(c & d) : Plot between Actual & Predicted Age. The points congregate around the 45° line for the KNN (above; ix(c)), but not as much as the Random Forest, indicating a good fit, while all the predictions have the same value for central imputation(below ; ix(d)). This indicates a bad fit as the actual values differ quite a bit from the central value.



Imputation Method	MAE	MSE	RMSE	MAPE
KNN	0.054632584	0.008399962	0.091651306	**0.099354060**
Central Imp.	0.17427507	0.04557488	0.21348273	**0.34012735**
Decision Tree	0.09481679	0.01733919	0.13167835	**0.17521269**
Random Forest	0.034948234	0.004105507	0.064074227	**0.067670623**

Table iii : Imputation methods' performance for the variable 'Score.in.Domain'

For the visualizations for "Score.in.Domain", both the actual and predicted values were multiplied by 100 to rescale the original values recorded/predicted in a 0-1 scale.

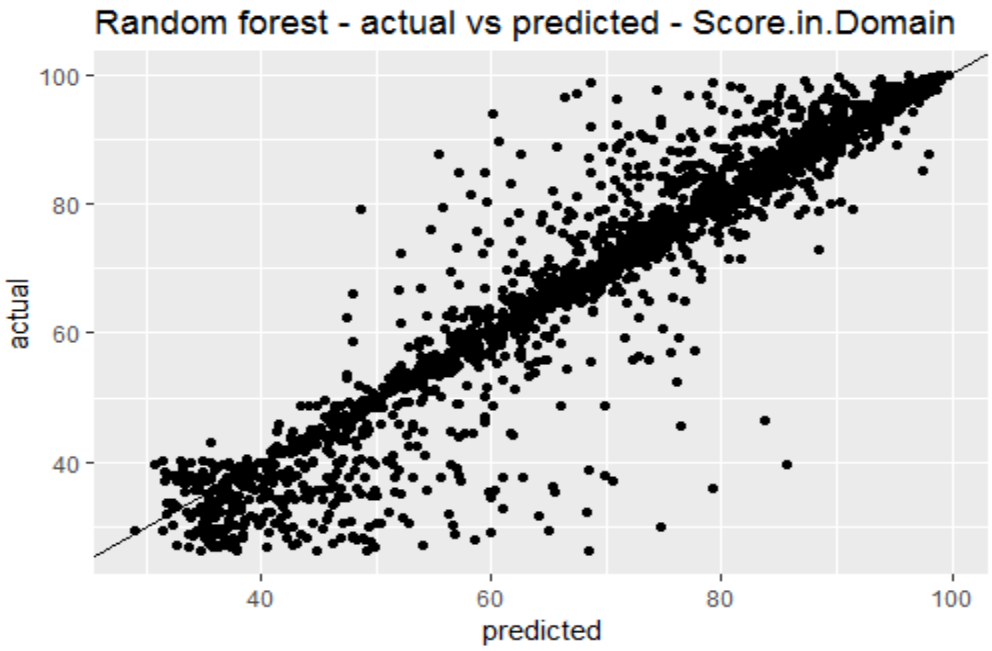


Fig x(a) : Plot between Actual & Predicted Score.in.Domain using a Random Forest, most of the points accumulate near the 45° line indicating a good fit.

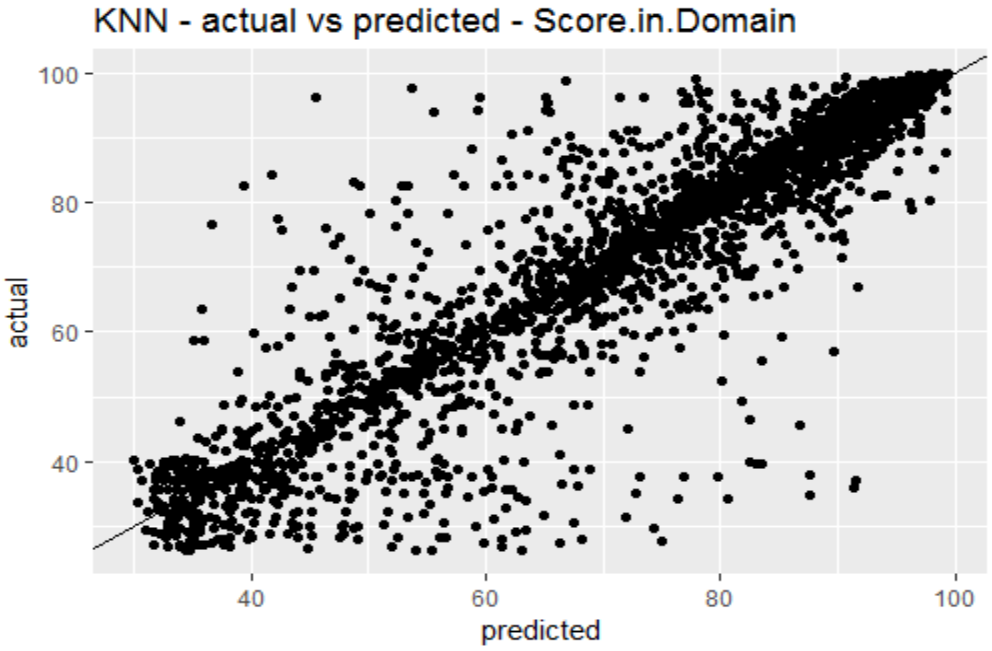


Fig x(b) : Plot between Actual & Predicted Score.in.Domain using KNN Imputation. The points congregate around the 45° line, however the distribution is not as compact as it was for the Random Forest.

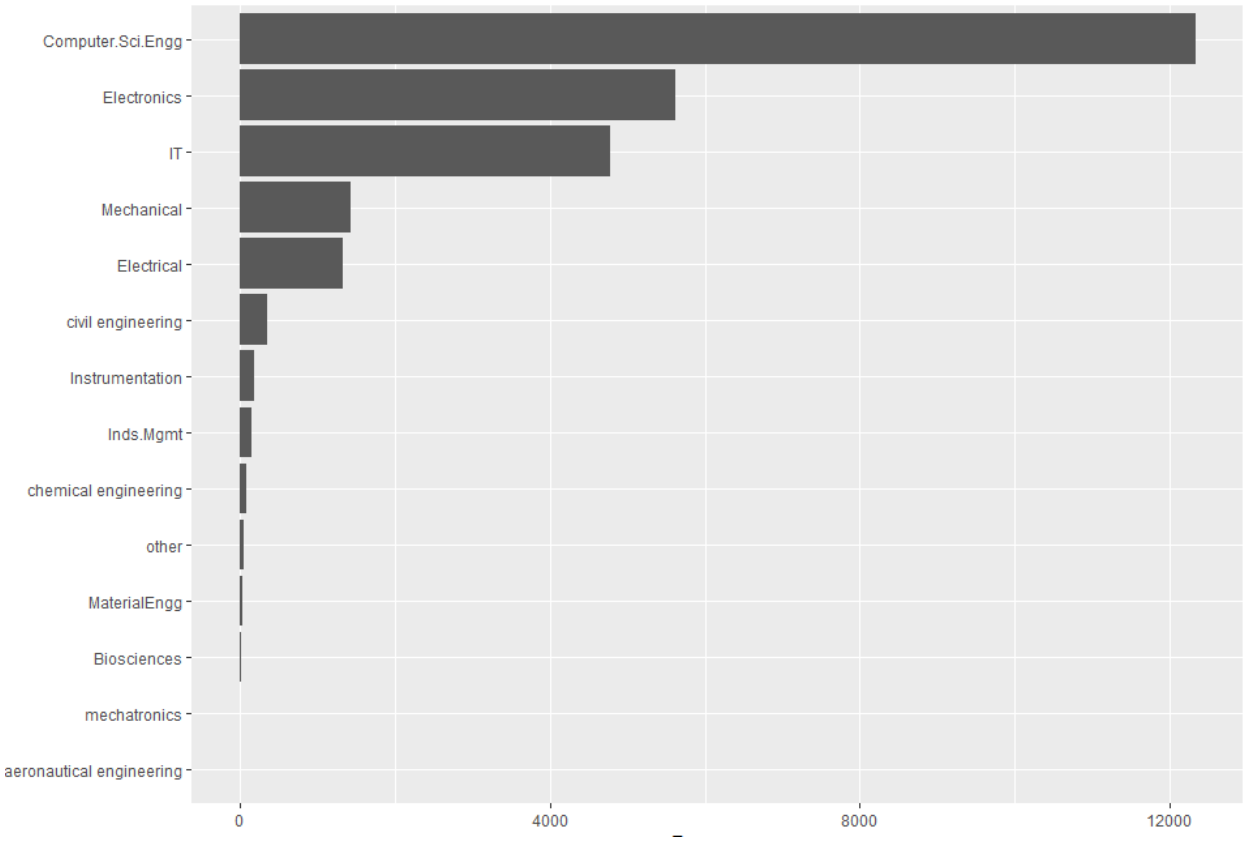
```
ggplot(mapping = aes(age_rf, age_actu))+geom_point()+geom_abline(intercept = 0, slope = 1)+
  xlim(24,34) +
  ggtitle("Random forest - actual vs predicted - Age") + xlab("predicted") + ylab("actual")

ggplot(mapping = aes(score_rf*100, score_actu*100))+geom_point()+
  geom_abline(intercept = 0, slope = 1)+ ylab("actual") +
  ggtitle("Random forest - actual vs predicted - Score.in.Domain") + xlab("predicted")
```

Fig x(c) : The Code used to generate visualizations for the random forest imputations for Age and Score.in.Domain

From the tabulated error metrics and the plotted visualizations, it was concluded that **Random Forest imputation had the best performance and hence was used to impute Age and Score.in.Domain variables**. It is to be noted that while training such an imputation algorithm, the actual Target variable must be excluded to avoid false correlations being introduced between the Target and imputed variables.

Data Exploration

While cleaning data, it was observed that the variables Discipline, School.Board.in.Tenth and Board.in.Twelfth had a high number of levels **(refer Fig-v(a) & Fig-v(b)** . After reducing this number, it was noticed that certain levels had an extremely low frequency of occurrence in the dataset. Training a model with isolated cases adds to the complexity of the model. Analysts have to make a decision whether it is worth the complexity to include these levels with isolated cases. If the *changes* in the final metrics of the model are deemed to be insignificant, then these levels can be grouped into the existing levels based on similarity or they can all be put under a new level.



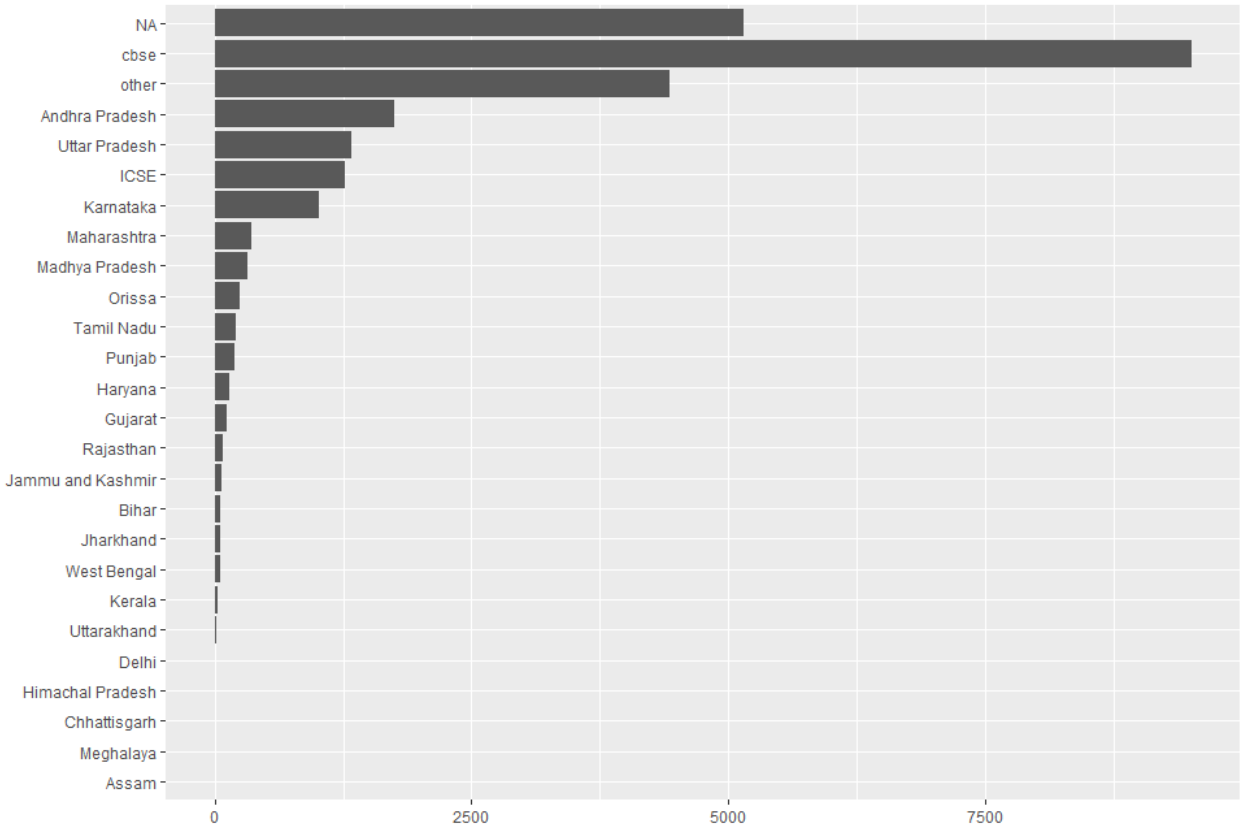


Fig v(a & b) revisited : Levels in Discipline(previous page) and School.Board.in.Twelfth(above) and their frequencies of occurrence. Note the levels with very low frequencies.

In the 3 variables mentioned, there exists a level named **"Other"**. A consensus was reached that the isolated cases only added to the complexity of the data. Hence, these isolated cases were identified and grouped into this "Other" level under the respective variable.

After such grouping, the number of levels in Discipline reduced from 14 to 9 and the levels in School.Board.in.Tenth and Board.in.Twelfth were reduced from 24 to 18 and 26 to 19 respectively.

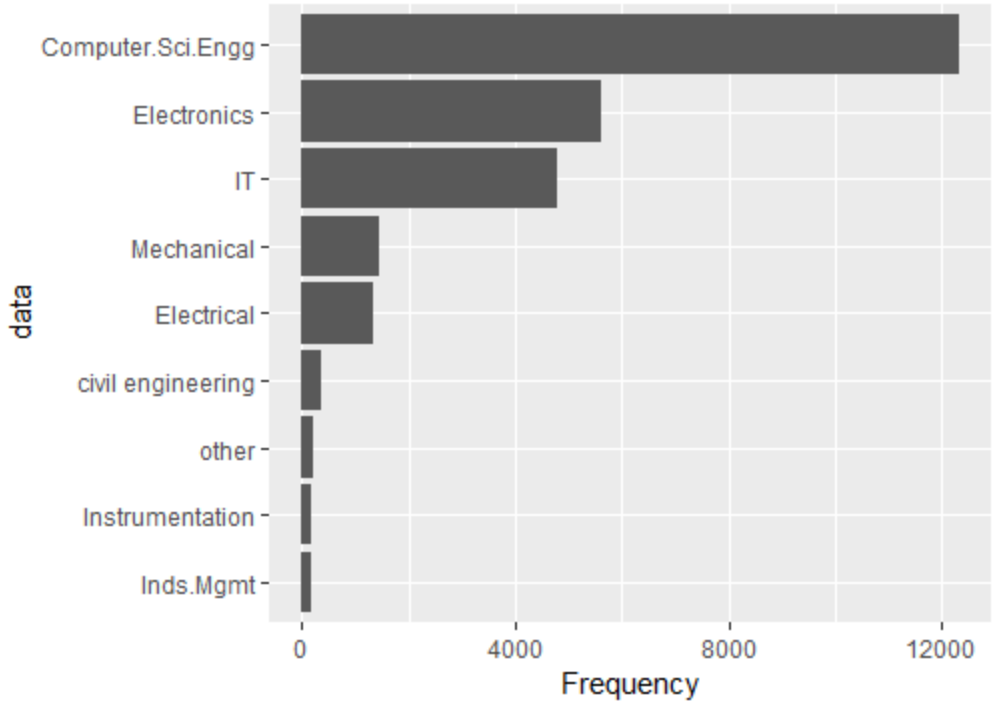


Fig xi(a) : Levels in the variable Discipline reduced from 14 to 9

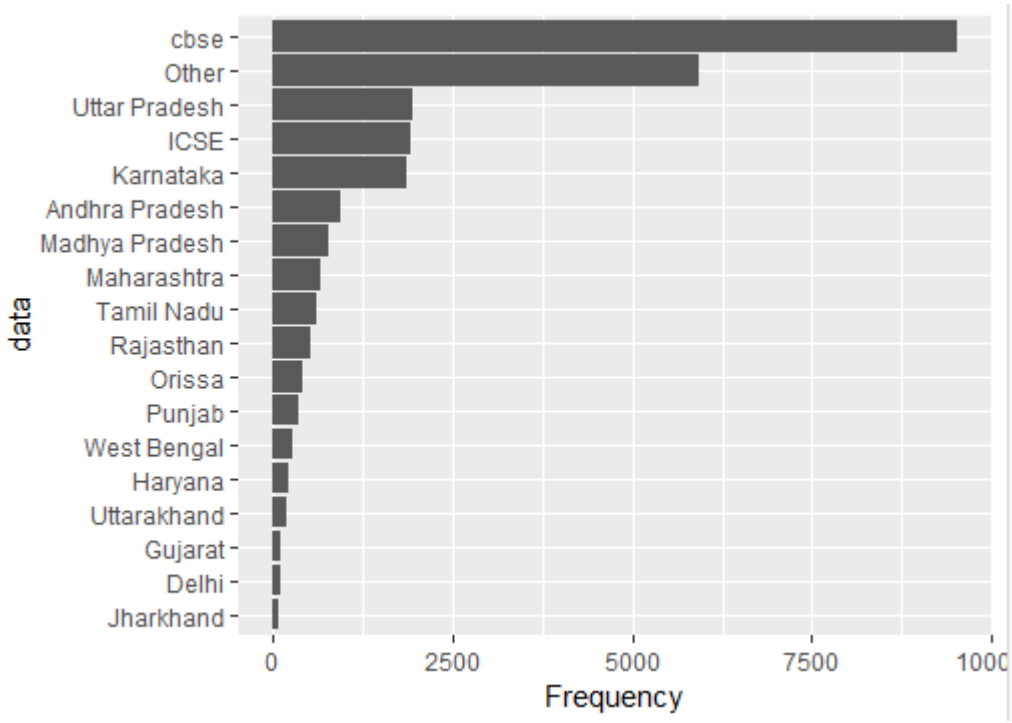


Fig xi(b) : Levels in School.Board.in.Tenth reduced from 24 to 18

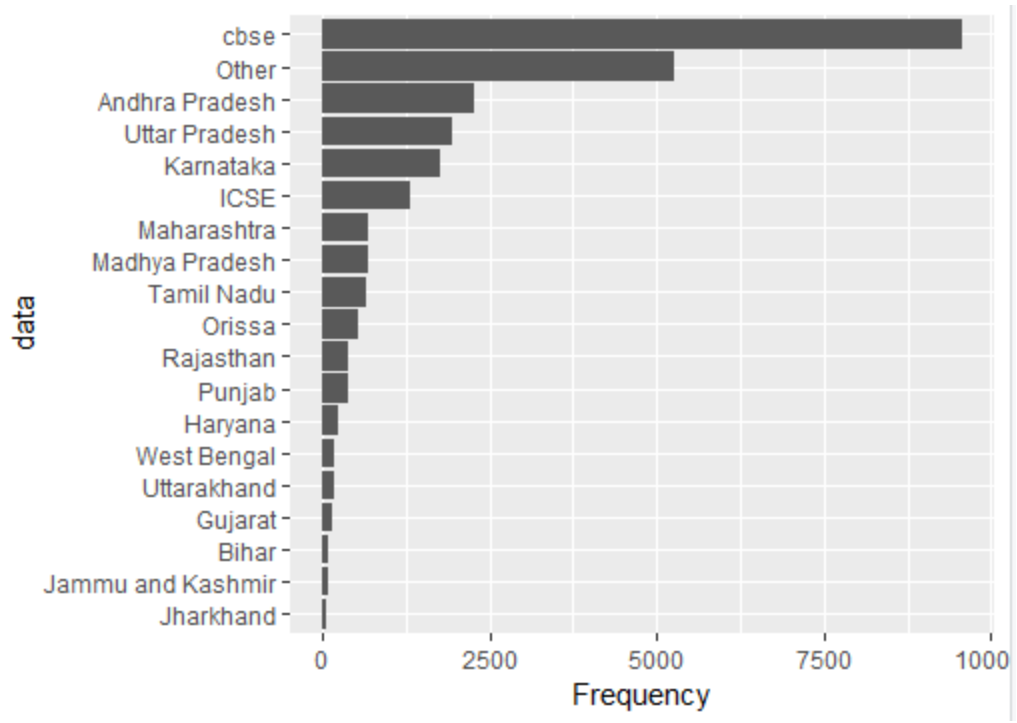


Fig xi(c) : Levels in Board.in.Twelfth reduced from 26 to 19.

Following these minor modifications, the data was **visually explored** using plots to observe the behaviour of each variable and how they affected the target variable. Few interesting trends and behaviors were uncovered during the course of such explorations, some of which have been mentioned here.

```
ggplot(visdata) + geom_point(aes(x = Age, y = Pay_in_INR/10**6, color = Gender)) +
  ggtitle("Scatterplot between Age and Salary \n with distinction on the basis of Gender") +
  xlab("Age (Years)") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))
```

Fig xii(a) : Code used to generate the plot between Age and Salary. A similar version of this piece of code was used to generate the rest of the plots.

Fig xii(b) : Plot between Age and Salary, distinction on the basis of Gender.

The individuals in the age bracket of **28-31 years seem to earn the highest salaries**. An interesting observation seen here, in the age group of 28 years, women seem to outnumber men when it comes to earning higher pay.

```
visdata$std_GPA = scale(visdata$GPA.Score.in.Graduation)
visdata$std_GPA = rescale(visdata$std_GPA, to = c(0,10))

ggplot(visdata) + geom_point(aes(x = std_GPA, y = Pay_in_INR/10**6, color = Gender)) +
  ggtitle("Scatterplot between GPA(grad) and Salary \n with distinction on the basis of Gender"
          ) +
  xlab("GPA (Std)") + ylab("Salary") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))
```

Fig xiv(a) : Code to generate the plot between college GPA and Salary.

Before plotting the scatterplot between GPA (graduation) and Salary, the variable GPA was standardized as different institutes use different scales for grading. Later, it was rescaled to values between 0 and 10 to make the plot interpretation easier.

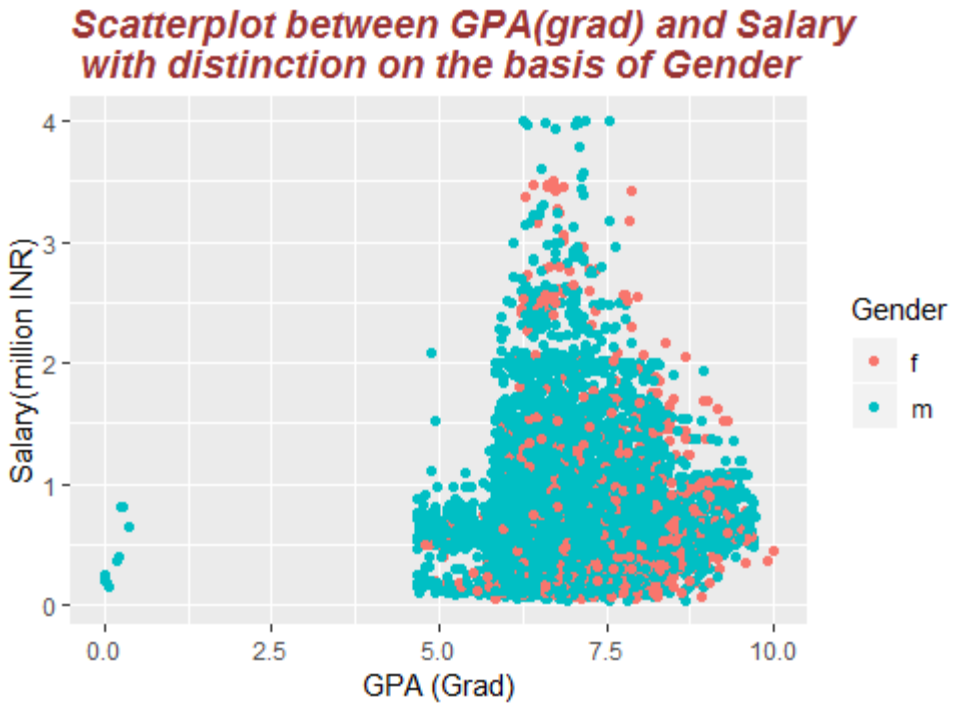


Fig xiv(b) : Plot between scaled college GPA and Salary.

A **scaled GPA range of 6.2 to 8.0 out of a maximum of 10** seems to be the interval into which most individuals fall. It is also this bracket where **maximum pay** is observed. An interesting observation is that the **highest GPA scores did not match with the highest salaries**, but an average to decent GPA range boasts of the highest pay.

```
visdata$std_XII = scale(visdata$Score.in.Twelth)
visdata$std_XII = rescale(visdata$std_XII, to = c(0,10))

ggplot(visdata) + geom_point(aes(x = std_XII, y = Pay_in_INR/10**6, color = Gender)) +
  ggtitle("Scatterplot between GPA(12th) and Salary \n with distinction on the basis of Gender")
+xlab("GPA (Std)") + ylab("Salary") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))
```

Fig xv(a) : Code to generate plot between 12th GPA and Salary.

Scaling and re-scaling was performed for the GPA in high school too for reasons similar to those mentioned for Graduation GPA.

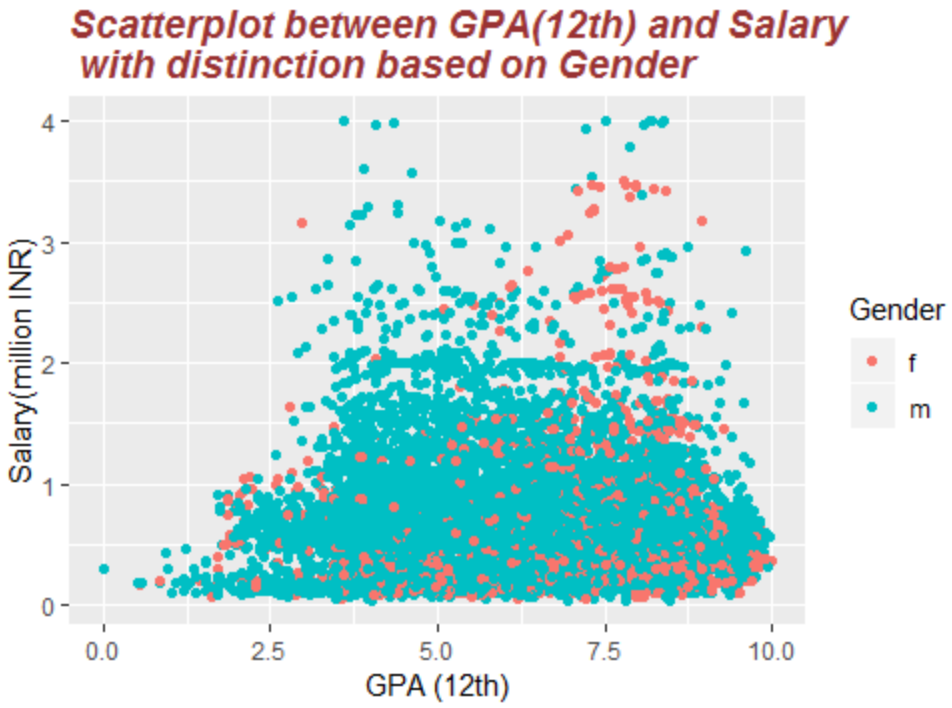


Fig xv(b) : Plot between scaled High School GPA and salary.

GPA in 12th does not seem to have as much impact on the salaries as graduation GPA did. However, **2 distinct peaks can be observed in the ranges 3.0 - 5.0 and 7.0 - 8.0, where people earn the highest**. An interesting insight that was observed was that **females seemed to perform better than males in High school.**

It was observed from **Fig xvi** that **English** speaking skills **did not have much impact** on individuals' salary, with less fluent individuals earning on par with the fluent ones.
However, there seems to be a range for individuals earning between 1 and 2 million with most of them scoring between 3.0 - 8.0.

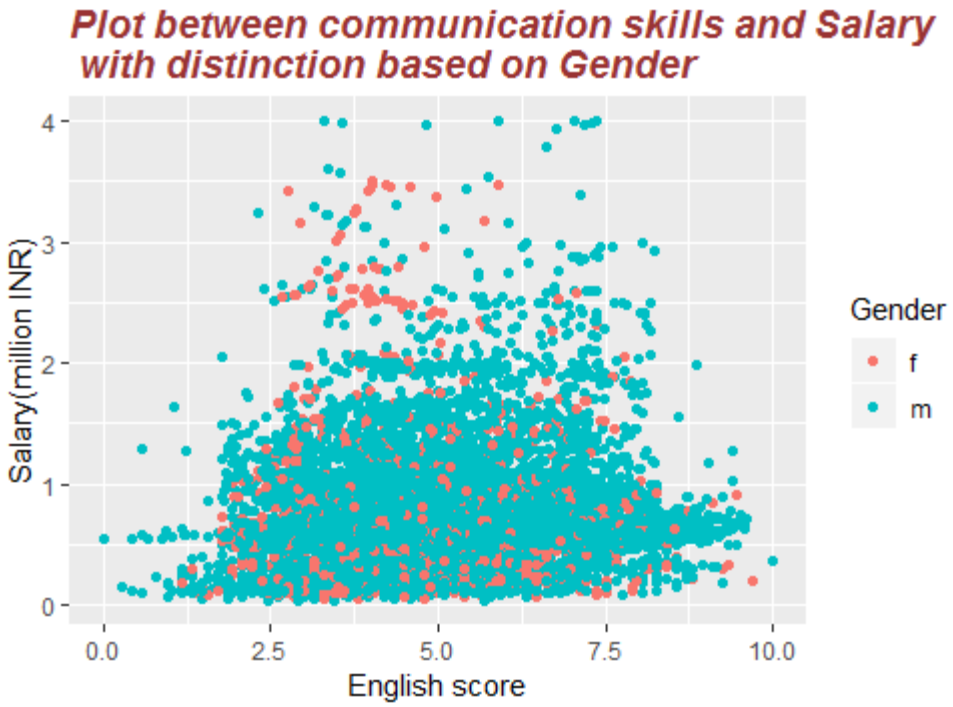


Fig xvi : Plot between English Score and Salary.

```
ggplot(visdata) + geom_point(aes(x = Graduation, y = Pay_in_INR/10**6, color = Gender,
                                 shape = Gap_Scenario)) +
  ggtitle("Plot between Graduation and Salary \n with distinction based on Gender and Gap Scenario")+
  xlab("Graduation degree") + ylab("Salary") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))
```

Fig xvii(a) : Code to generate plot between Graduation Degree and Salary. Distinctions were created on the basis of Gender and Gap_Scenario.

The plot between Graduation Degree and Salary was plotted with distinctions made for Gender as well as Gap_Scenario. However, it was suggested that these distinctions should be broken down into different plots to make interpretation easier.

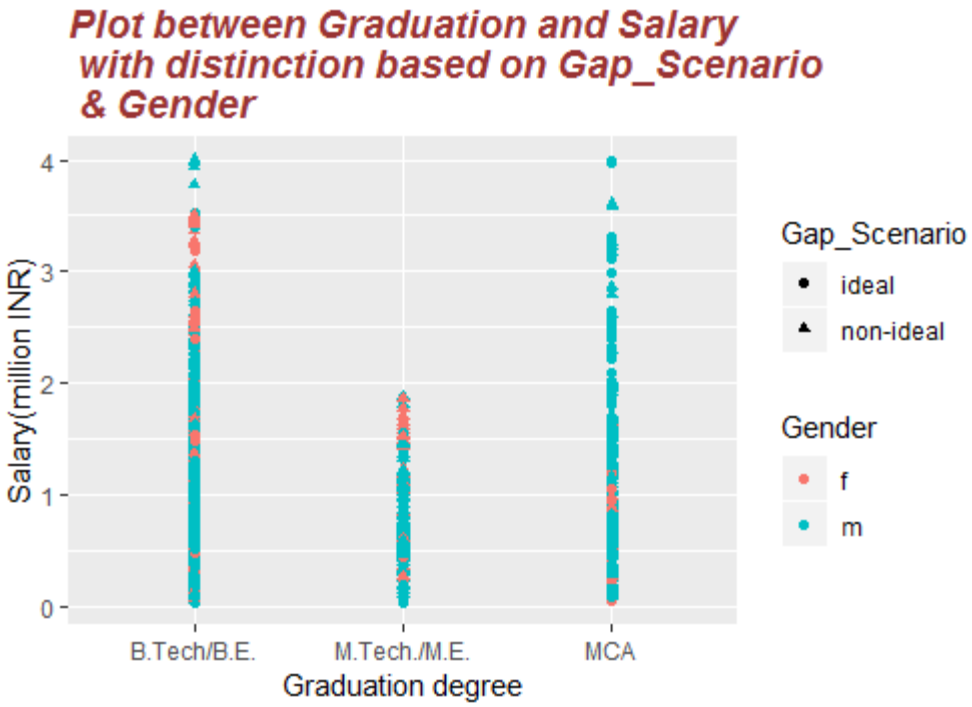


Fig xvii(b) : Plot between Graduation and Salary

Although the Gap_Scenario distinctions were difficult to interpret from this chart (**Fig-xvii(b)**), It was noticed that individuals who graduated with an **M.Tech/M.E. degree earn significantly less than their B.Tech/B.E. and MCA counterparts**. An interesting insight drawn from the plot was that a majority of the females do not opt for higher degrees with B.Tech /B.E. field consisting of most of the females.

Upon removing the Gender filter (**Fig xvii(c)**), it was noticed that **individuals with a non-ideal career gap and a B.Tech/B.E. degree were among the highest paid.** A **non-ideal Educational gap occurs frequently among the highest paid cases** for each of the Graduation levels, contradicting the popular notion that a gap in education can be detrimental to the chances of securing a good salary.

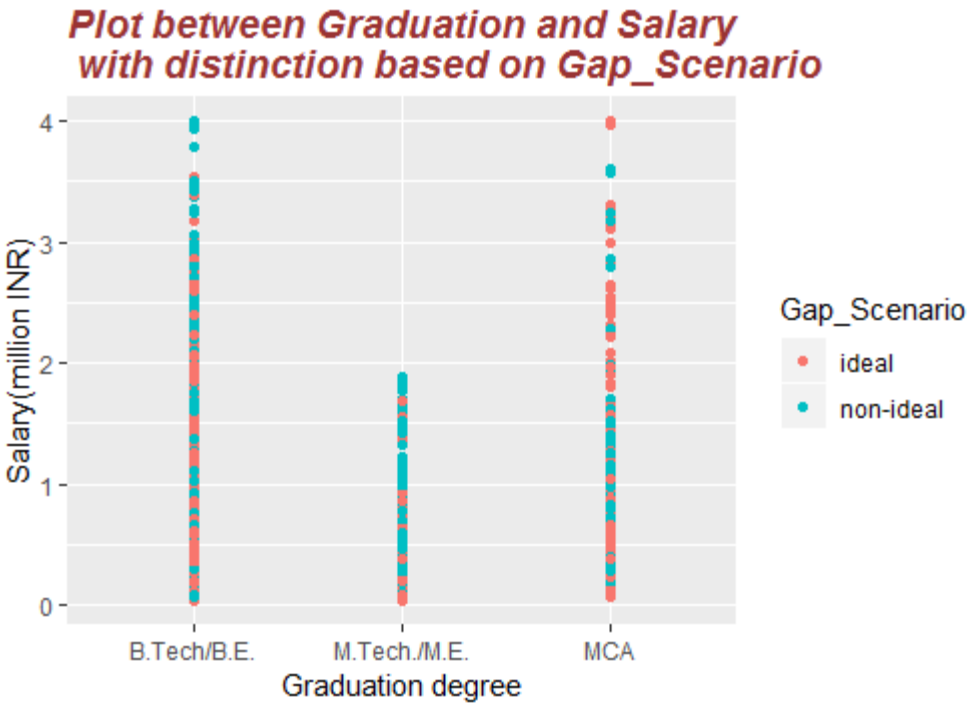


Fig xvii(c) : Plot between Graduation degree and Salary with distinction based on Gap_Scenario.

```
tier2 = visdata[which(visdata$CollegeTier == '2'),]
tier1 = visdata[which(visdata$CollegeTier == '1'),]
#Tier 1 is a better rating, Tier 2 is lower

ggplot(tier1) + geom_point(aes(x = Gender, y = Pay_in_INR/10**6)) +
  ggtitle("Plot between Gender and Salary \n for tier 1 college grads") +
  xlab("Gender") + ylab("Salary") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))
```

Fig xviii(a) : Code that generates a plot between Gender and Salary for individuals graduating from Tier-1 colleges.

The data set was split on the basis of “CollegeTier”, a variable with 2 levels, and each subset was plotted separately since it was observed that multiple filters were making it difficult to interpret the plots.

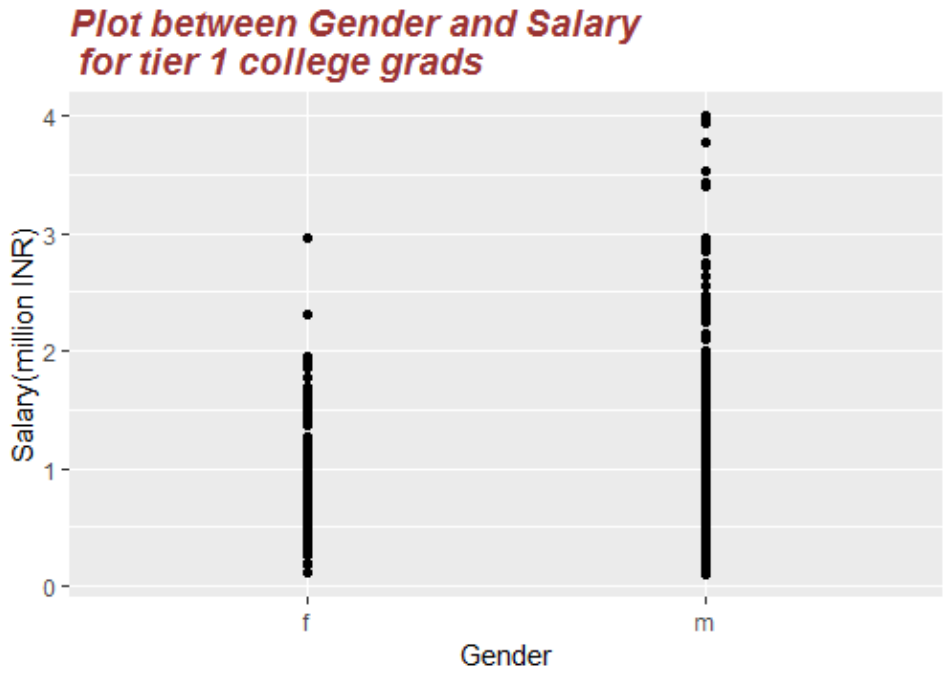


Fig xviii(b) : Plot between Gender and Salary for Tier-1 college graduates

It was observed that males graduating from Tier-1 colleges earned significantly more than their female counterparts.

Fig xviii(c) : Plot between Gender and Salary for Tier-2 college graduates

Comparing **Fig viii(b)** and **Fig viii(c)** it was noticed that **females from Tier-2 colleges earn significantly more than female graduates from Tier-1 colleges.** This becomes interesting when the fact that **Tier-1 colleges are ranked higher than Tier-2 colleges** is considered.

Class-wise average salary was obtained, based on the classes of the variable **"Discipline"**.

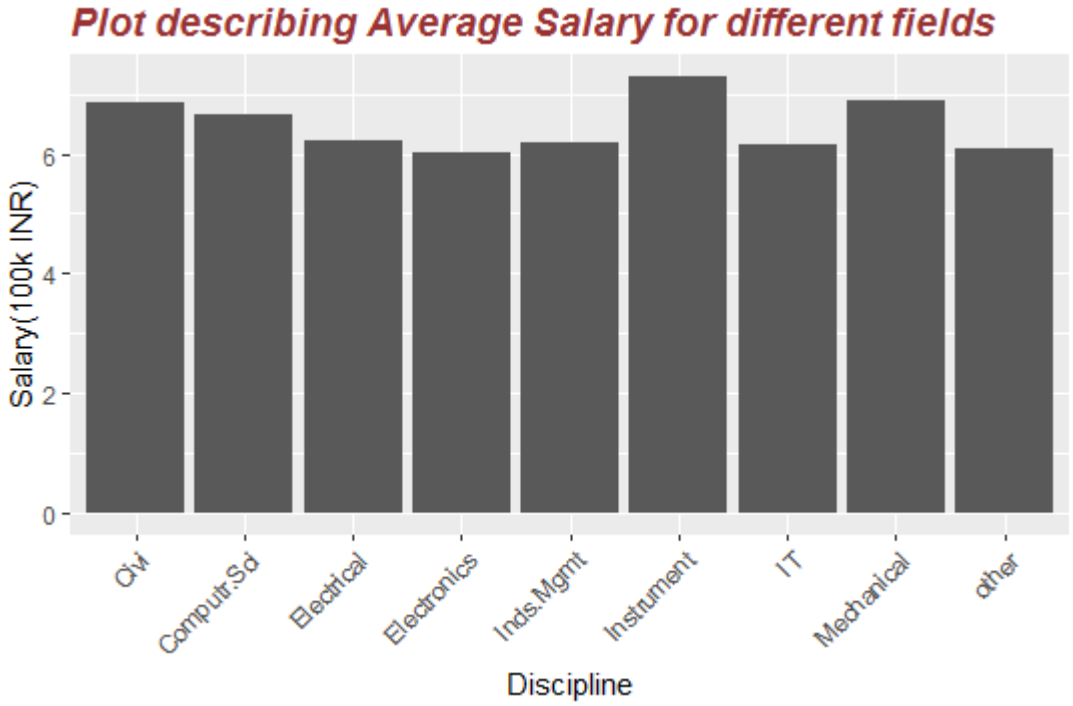


Fig xix : Plot depicting the average pay for various Disciplines.

It was observed that despite Computer Sciences and IT being the most frequently occuring discipline, the **highest salaries** were given to individuals coming from fields less opted for, such as **Instrumentation, Mechanical, and Civil Engineering**.

Model Training and Evaluation

"In God we trust, all others bring data." - W. Edwards Deming

Having sufficiently explored the data, three prediction algorithms were zeroed in on - **Random Forest**, **Decision Tree** and **Multiple Linear Regression.** In layman's terms, a prediction algorithm can be likened to a tool which is used by analysts to predict a value for the required field, based on a given set of features. The algorithm is trained to identify the variations in this set of features, and associate each one these variations with the variable that is to be predicted or the **target variable**. When a new 'observation' arrives, this set of features from the new observation is fed to the algorithm, which then predicts a value for the target variable.

In a **linear regression** algorithm, this set of features, or **predictors**, are assigned weights or **coefficients**. The purpose of assigning weights is to establish a linear relationship between the predictors[9] and the target variable, meaning that a change in one of the predictors should bring a direct change in the target variable. How much of a change is induced in the target variable due to a change in a single predictor's value determines the **significance** of that predictor.

A **decision tree** relies on **splitting** the dataset on which the algorithm is being trained, into smaller subsets. For a prediction case, each predictor is separately used to predict the target variable and the **values of the predictor which gives the least error are used to split** the dataset[10]. This process is repeated recursively, with such checks performed for each split and ends when there is no change in the error metric.

A **random forest** is an extension of the logic behind a decision tree. Instead of training a single tree, multiple such trees are trained with different subsets of predictors and the outputs of all these trees are averaged to arrive at the final output of this collection of trees or **forest**.

---

9 "Multiple Linear Regression,"Yale University,accessed Nov 26, 2018, http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm.

10 Bala Deshpande,"2 main differences between," last modified Jul 13,2011,http://www.simafore.com/blog/bid/62482/2-main-differences-between-classification-and-regression-trees.

It is common practice to split the available data into **train, validation** and **test** datasets. The algorithms are trained on the train data, validated on validation to improve the performance and a final check is performed on the test set. This was done using the **createDataPartition()** function.

Before training these algorithms, the numeric variables in the dataset were **standardized**. This was done to bring all the continuous values onto the same scale and avoid cases where variables are given more/less significance or importance owing to their large/small values. This was performed on the subset of all continuous variables using the **scale()** function. Standardization of the train, validation, and test datasets was done separately.

In order to reduce the complexities involved in training models with many features, it was proposed to utilize dimensionality reduction techniques to obtain a smaller feature set. **Principal Component Analysis** was one such technique discussed and used. This method introduces **principal components**, derived from the existing set of features. Each principal component consists of a portion of all the given features, that is, it constructs a new set of features based on a combination of old ones.[11] Based on the percentage of variance explained, an adequate number of principal components can be chosen to train a model.

It must be noted that before performing PCA, all the categorical variables must be dummified. This was done using the **dummyVars()** function. All the algorithms were trained with and without PCA to draw comparisons. Principal components were derived using the **prcomp()** function. Based on the Screeplot and the summary, required number of Principal Components were chosen.

[11] Meigarom Lopes,"Dimensionality Reduction,"Towards Data Science, last modified Jan 10, 2017,https://towardsdatascience.com/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32.

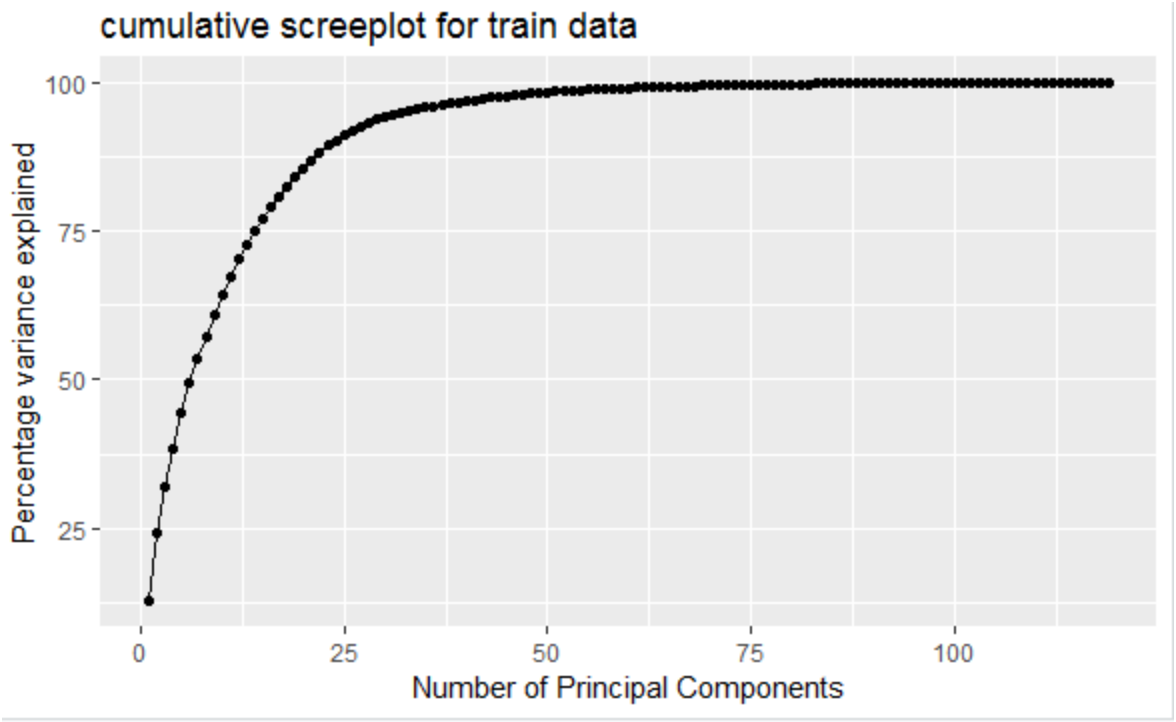


Fig xx(a): Plot between the number of principal components and cumulative percent variance explained

It was inferred from the plot that more than 90 percent of the variance was captured by 50 principal components. This was corroborated by the cumulative variance values observed in the summary of the **prcomp()** object.

	PC49	PC50	PC51	PC52	PC53	PC54	PC55	PC56
Standard deviation	0.18367	0.17366	0.17012	0.15759	0.14876	0.14352	0.14003	0.13453
Proportion of Variance	0.00119	0.00107	0.00102	0.00088	0.00078	0.00073	0.00069	0.00064
Cumulative Proportion	0.98333	0.98440	0.98542	0.98630	0.98708	0.98781	0.98850	0.98914

Fig xx(b) : Summary of the prcomp() object generated.

After generating Principal Components, the 3 algorithms were **trained twice - using PC's and without using PC's** and the results were compared. The error metric chosen was MAPE.

Without PCA	MAE	MSE	RMSE	**MAPE**
Random Forest	5.988434e+04	1.263334e+10	1.123981e+05	**1.245342e-01**
Decision Tree	1.662965e+05	7.982575e+10	2.825345e+05	**3.483401e-01**
Linear regression	1.810622e+05	8.996289e+10	2.999381e+05	**3.549146e-01**

Table iv : Performance measures of the 3 algorithms trained without using PCA on validation data.

It was observed that Random Forest outperformed the rest of the models, with a MAPE value of 0.124. The random forest was trained with **100 trees.** 12 variables were considered at each node to make a split. Although 100 trees were used, the error rate stabilized after 50 trees, as can be seen upon plotting the model :

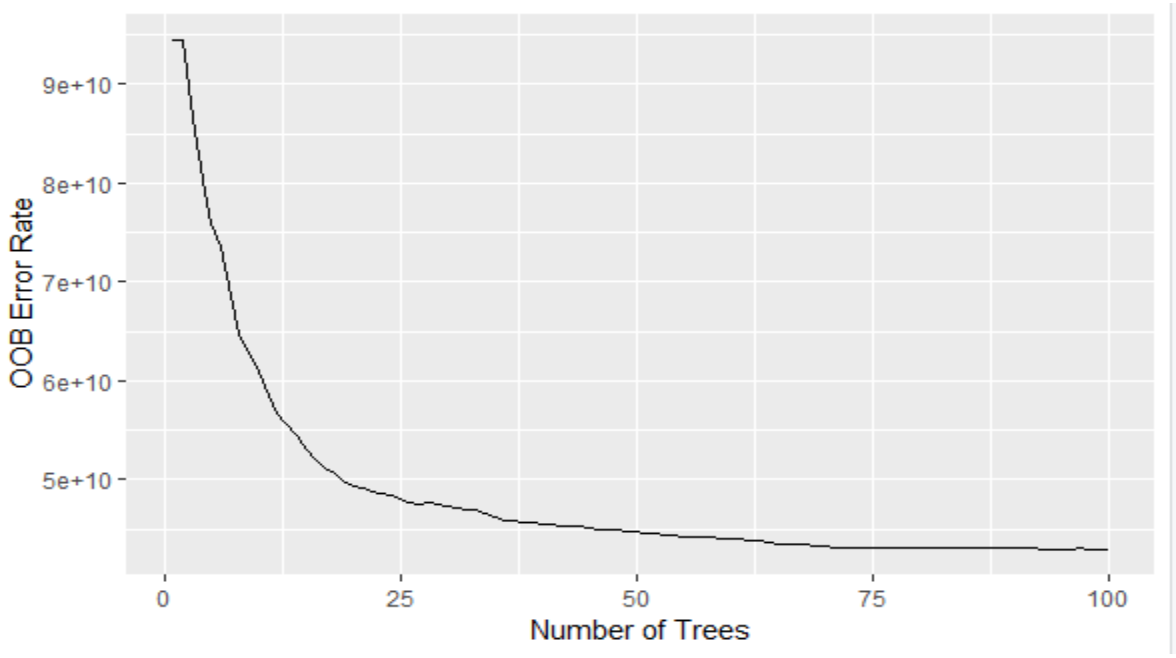


Fig xxi(a) : Plot between number of trees and error rate for the trained random forest without using PCA

From the variable importance plot **(Fig xxi(b))**, it was noticed that the State variable was very significant when it comes to predicting salary accurately. Quantitative skills also proved to be an important factor affecting salary regardless of the individual's Discipline. The **random forest algorithm can also be used for feature selection** with variables being chosen on the basis of their standings on the VarImpPlot. The plot depicts the **decrease in Mean Square Error** when a variable is permuted. A large decrease indicates that the variable is **important**. Variables are depicted in decreasing order of importance.

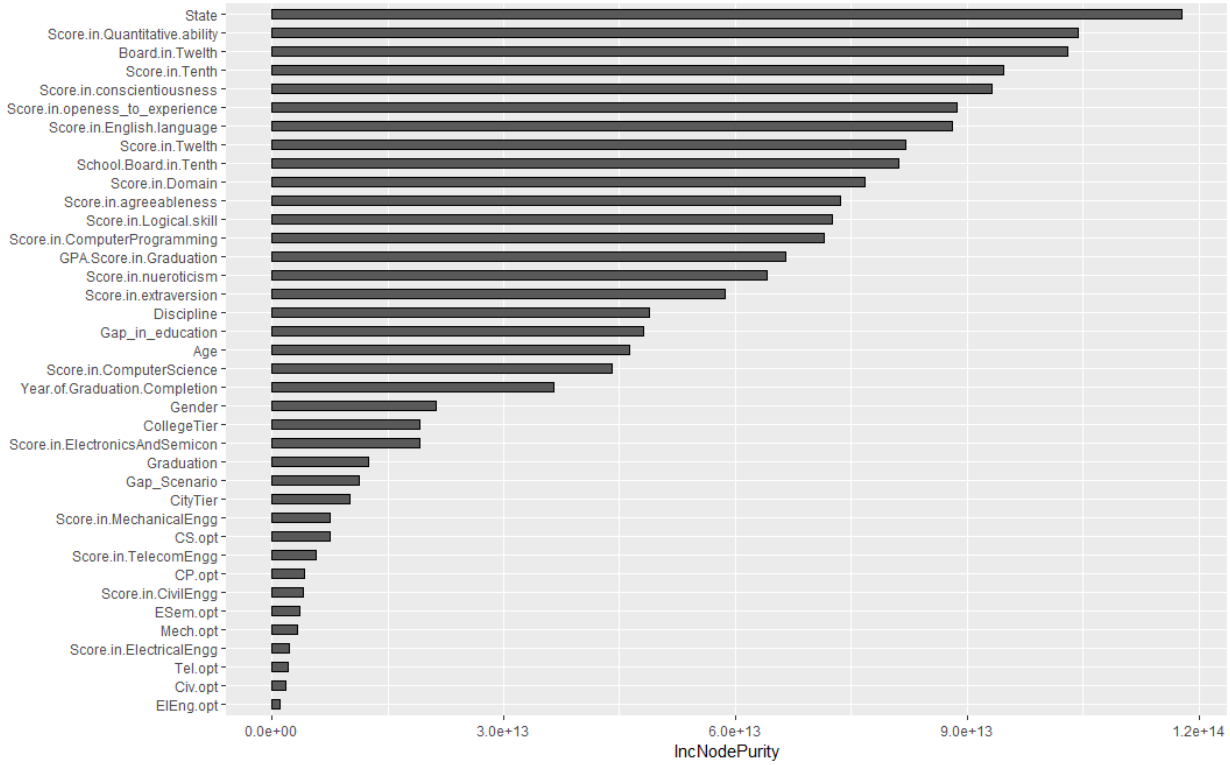


Fig xxi(b) : Variable Importance Plot for the random forest without using PCA.

Similarly, for the **decision tree**, a **plot between actual and predicted values** was observed (**Fig xxiii(a)**). The **values failed to congregate around the 45 degree** line, confirming that the Random Forest was a better performer. The rules were nonetheless extracted using the **rpart.rules()** function.

```
model_dt = rpart(Pay_in_INR~.,train, method = "anova")
dt_raw = predict(model_dt, valid[,-39])
regr.eval(valid[,39],dt_raw)
#         mae          mse         rmse         mape
#1.662965e+05 7.982575e+10 2.825345e+05 3.483401e-01
rpart.rules(model_dt, clip.facs = T)
plot(model_dt)
prp(model_dt)
fancyRpartPlot(model_dt)
```

Fig xxii : Code that trains and evaluates the Decision Tree. The error metrics obtained are displayed alongside the code together with the commands to extract rules and plot the tree.

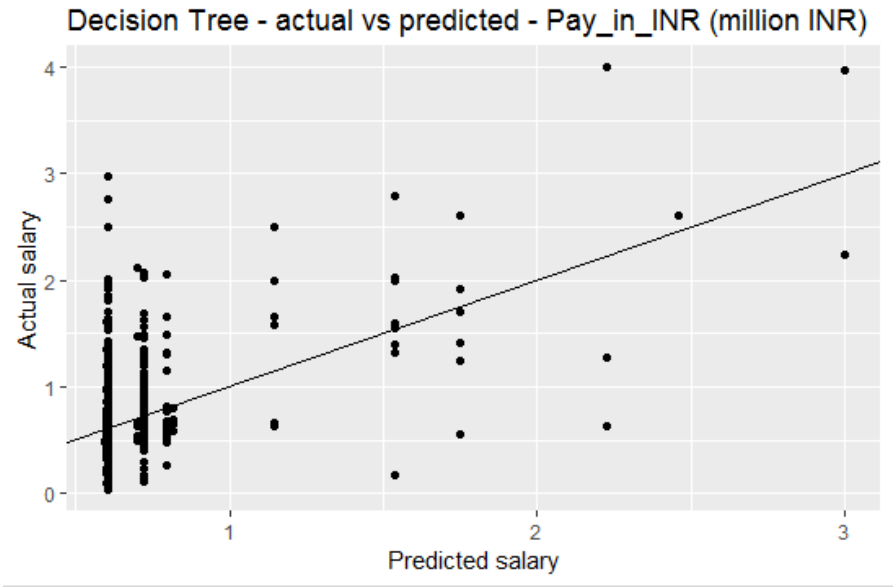


Fig xxiii(a) : Actual vs Predicted salaries using a Decision tree, observed that the model is a poor fit, predictions do not follow any trends, seem almost random.

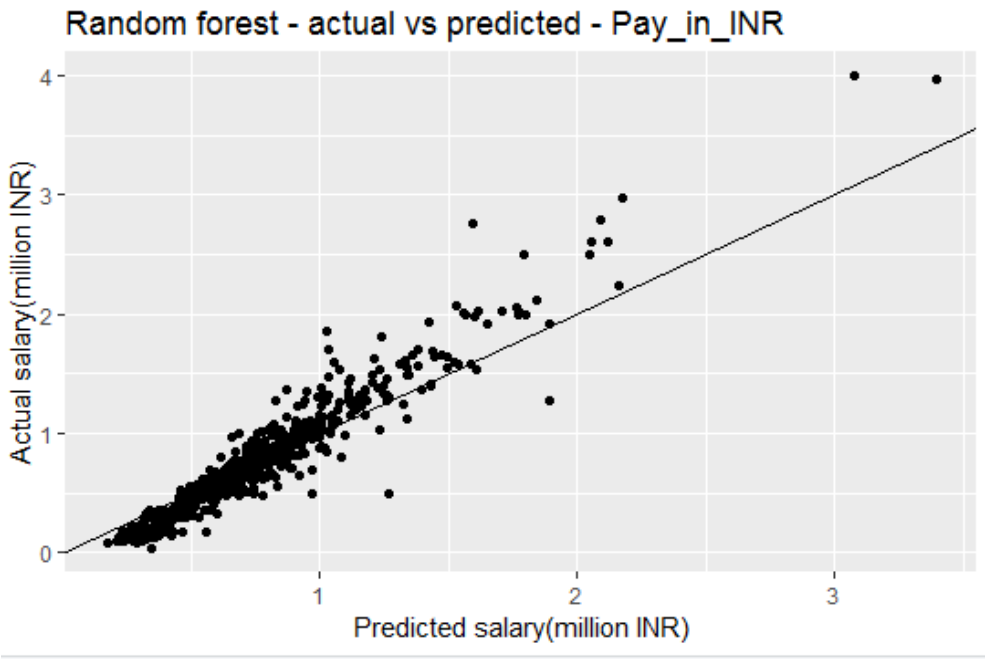


Fig xxiii(b) : Actual vs Predicted salaries using the Random Forest. A much better fit.

A **Lift chart** is depicts the number of correct predictions in the 'n' groups of observations fed to the algorithm. These groups are made after sorting based on the probability of prediction.
Lift charts were compared for these algorithms which further confirmed Random Forest's superiority over the Decision Tree. **A good model has high initial lift.**

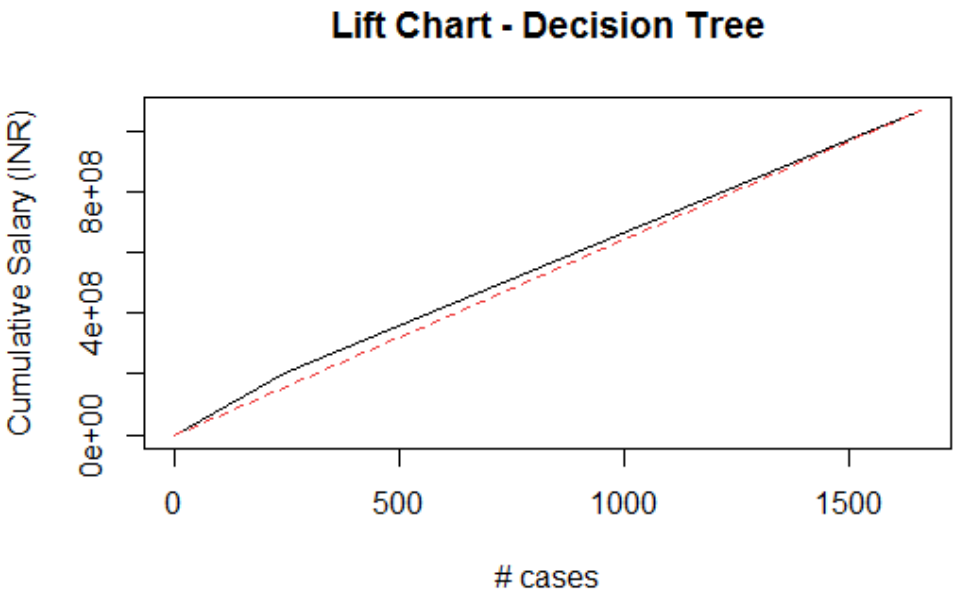
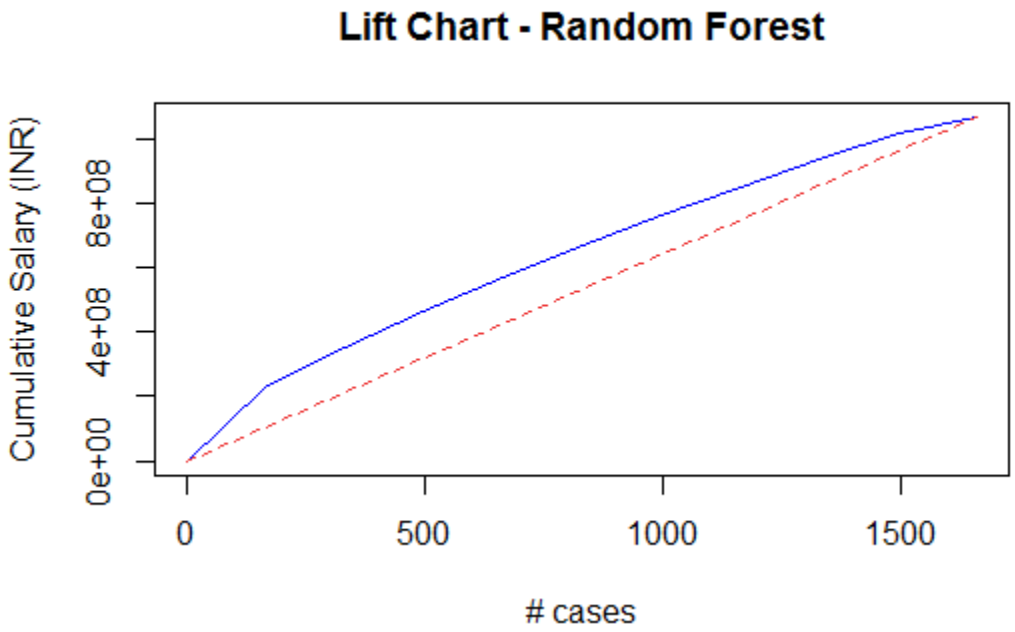


Fig xxiv(a & b) : Lift chart for the Decision Tree (above) and Lift chart for the Random Forest (below). It was observed that the <u>Decision tree performs about as good as a baseline model or random guessing</u>, while the random forest exhibits a good initial lift.



Similar comparisons were made between **Linear regression** model and the random forest. The **Decile Wise lift charts** for both the models had a high initial lift but the gradual 'stairway' descent was observed only for the Random Forest.

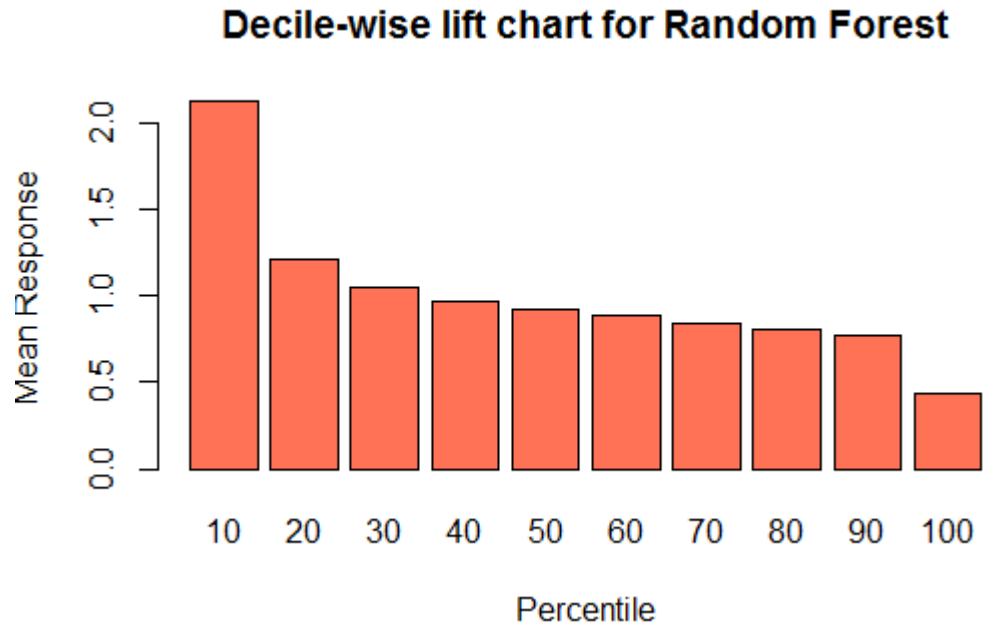
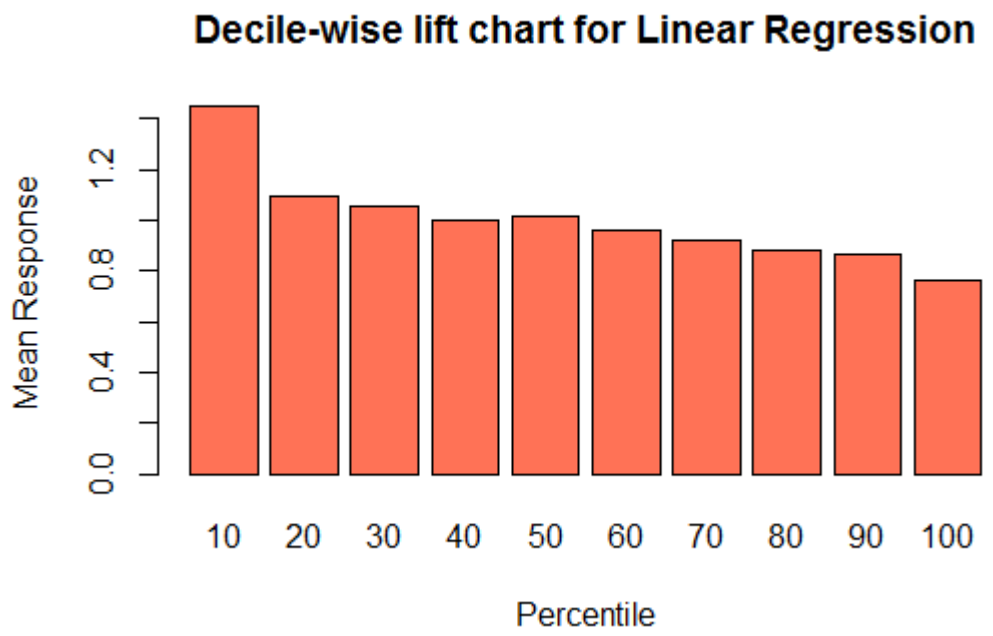


Fig xxv(a & b): Decile wise lift charts for Random Forest and Linear Regression.



It was observed from the decile wise lift charts that multiple linear regression has an uneven descent of the decile peaks. **A decile wise lift chart is obtained when predicted values are sorted based on their probabilities of prediction and the proportion of correct predictions is plotted for each decile.** In such a scenario, the peaks must gradually recede as is seen in the random forest's decile wise lift chart.

Nevertheless, significant variables obtained from multiple linear regression were compared with those that were considered important by the random forest.

```
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                       401442      52856   7.595 3.24e-14 ***
Score.in.Tenth                      3025       3946   0.767 0.443346
Score.in.Twelth                    -3065       3959  -0.774 0.438933
GPA.Score.in.Graduation            15652       3107   5.038 4.75e-07 ***
Year.of.Graduation.Completion     -32175       5663  -5.682 1.35e-08 ***
Score.in.English.language          20223       3128   6.466 1.04e-10 ***
```

Fig xxvi(a) : a part of the summary of linear regression, with significant predictors highlighted

It was noted that **most of the important variables indicated by Random Forest do not match with the ones highlighted as significant by the linear**

regression model. In fact, **some of the variables deemed to be less important** by the random forest **are highlighted as significant ones** by the regression model such **Year.of.Graduation.Completion.**

Similarly, important variables were extracted from the decision tree using **summary()** function on the rpart object. Though some of them seem to match with those that occur in random forest's important variable plot, the relative level of importance attached to these variables in the decision tree model was observed to be different.

```
Variable importance
Score.in.openess_to_experience                9
          Board.in.Twelth                     8
Score.in.English.language                     8
     School.Board.in.Tenth                    7
Score.in.ComputerProgramming                  7
          Score.in.Tenth                      6
    Score.in.Logical.skill                    6
          Score.in.Domain                     6
                   State                      6
         Gap_in_education                     4
          Score.in.Twelth                     4
                  Gender                       4
               Discipline                     4
                     Age                      3
Score.in.nueroticism                          3
   GPA.Score.in.Graduation                    3
Score.in.Quantitative.ability                 2
              CollegeTier                     2
     Score.in.extraversion                    2
                   CS.opt                     2
Score.in.ComputerScience
```

Fig xxvi(b) : Important variables as shown by the Decision Tree model

It was interesting to note that some of the **common variables being assigned vastly different levels of importance** by two different models **resulted in significant difference in the models' performance.**
It was hence concluded that **random forest performed the best among the models trained without using PCA.**

It was observed that the **models trained using the first 50 principal components,** which accounted for more than 95 percent of the variance in the data**, recorded higher error rates than models trained without using principal components.**

Using PCA	MAE	MSE	RMSE	**MAPE**
Random Forest	6.229023e+04	1.291048e+10	1.136243e+05	**1.337623e-01**
Decision Tree	1.701781e+05	8.854554e+10	2.975660e+05	**3.550161e-01**
Linear regression	1.830637e+05	9.280413e+10	3.046377e+05	**3.584974e-01**

Table v : Error metrics for models trained using Principal Components.

The purpose of dimension reduction is to reduce the complexity of the model being trained by incorporating lower number of features. This is usually done at the expense of a minimal drop in goodness of the model. However, in PCA this drop in complexity of training is filled in by the complexity of making sense out of the model. Since **each principal component is composed of all the original features, it is difficult to explain how the target variable is influenced by the original set of predictors**.

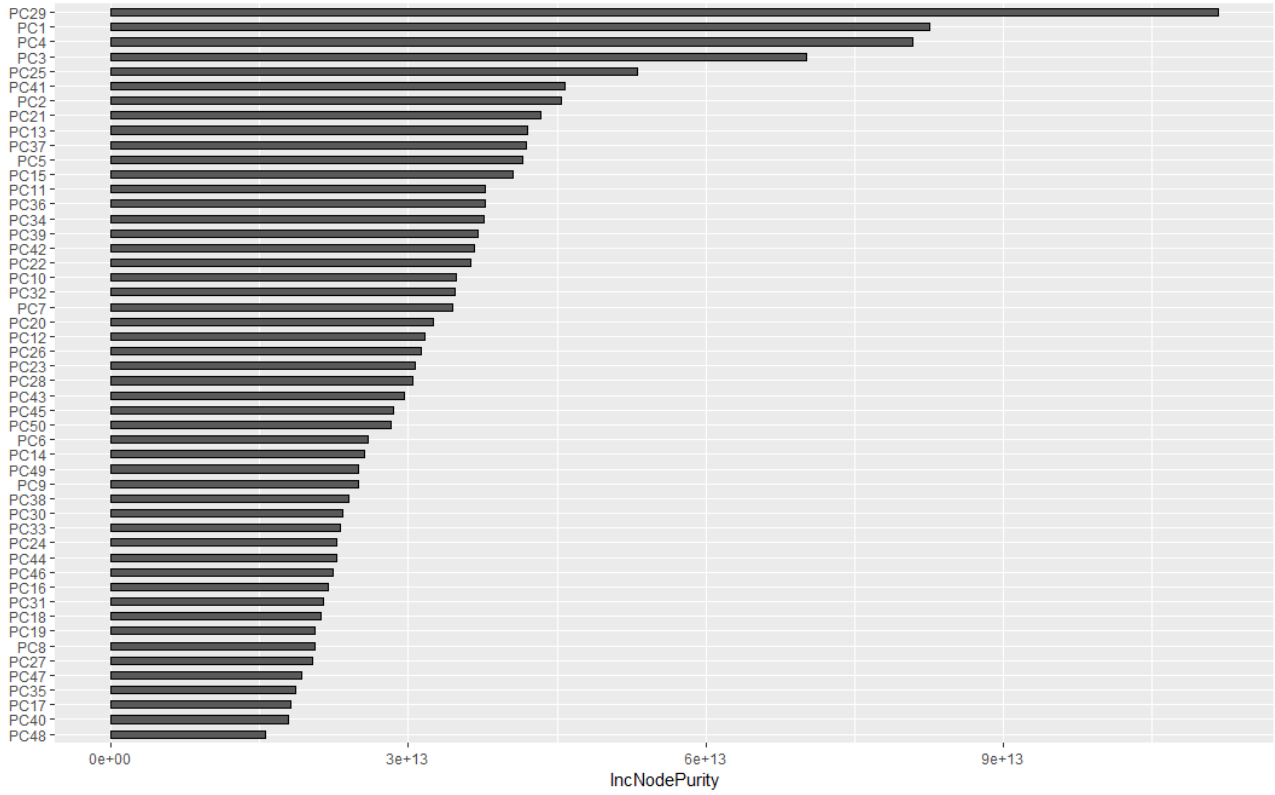


Fig xxvii : Variable Importance plot for the random forest trained on Principal Components

It was observed that **principal components 29**, **1,and 4** are assigned a high importance value when **a random forest is trained using the principal components.**

The contributions of the original set of predictors to these 3 principal components was plotted using the **fviz_contrib()** function.

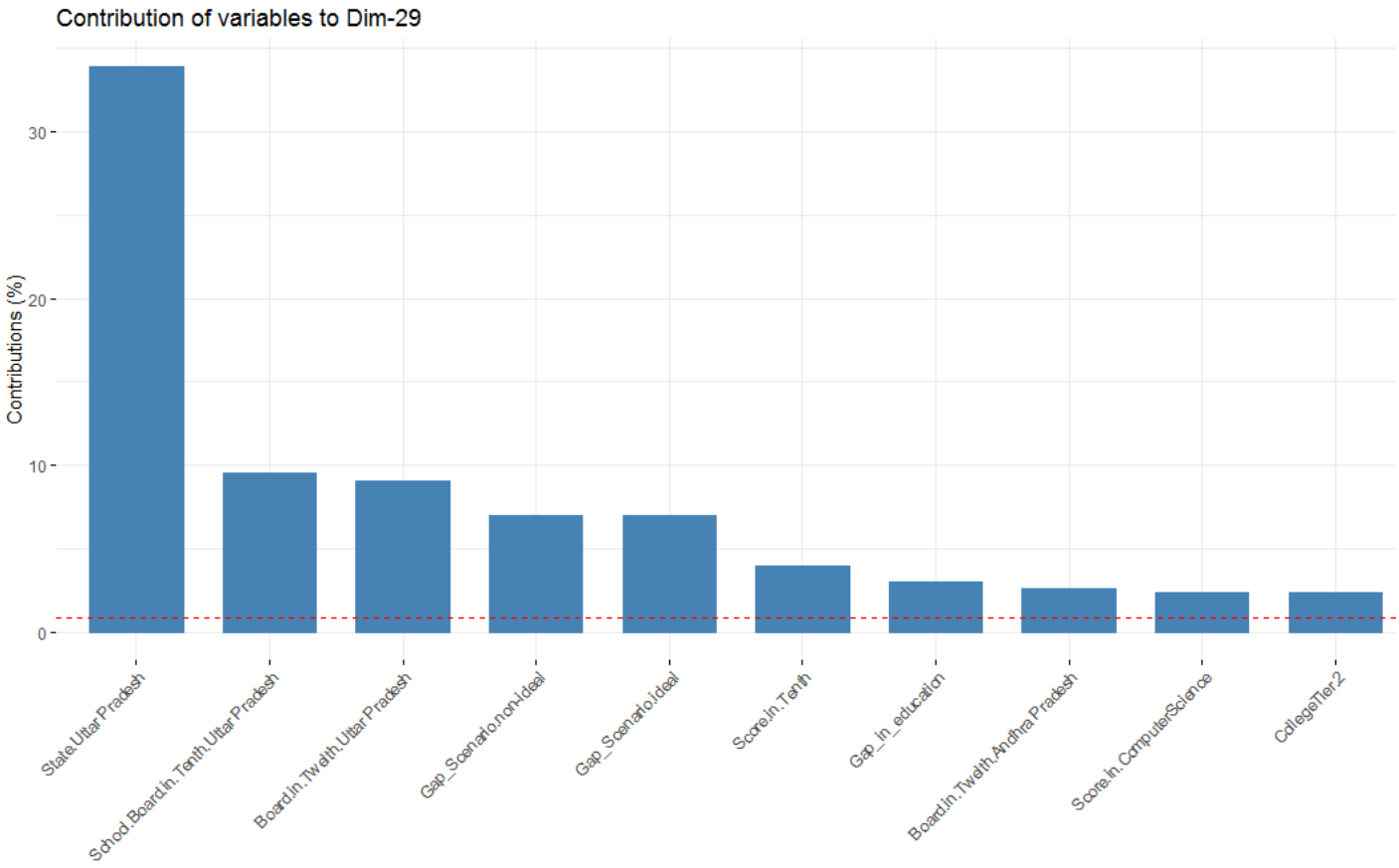


Fig xxviii(a) :Top 10 Contributions of the original predictors to PC-29

PC 29 was shown to be most important in the random forest model trained on principal components. The contributions to PC 29 seem to corroborate the information provided by the **random forest trained on original variables** which **mentioned "State" as the most important** variable. In this contribution plot, **it was observed that 30% of the most important PC is constituted of the level "Uttar Pradesh" from the state variable.** The red line passing through the graph indicates the expected average contribution. It must be noted **that only the top 10 contributors have been depicted in this plot**.

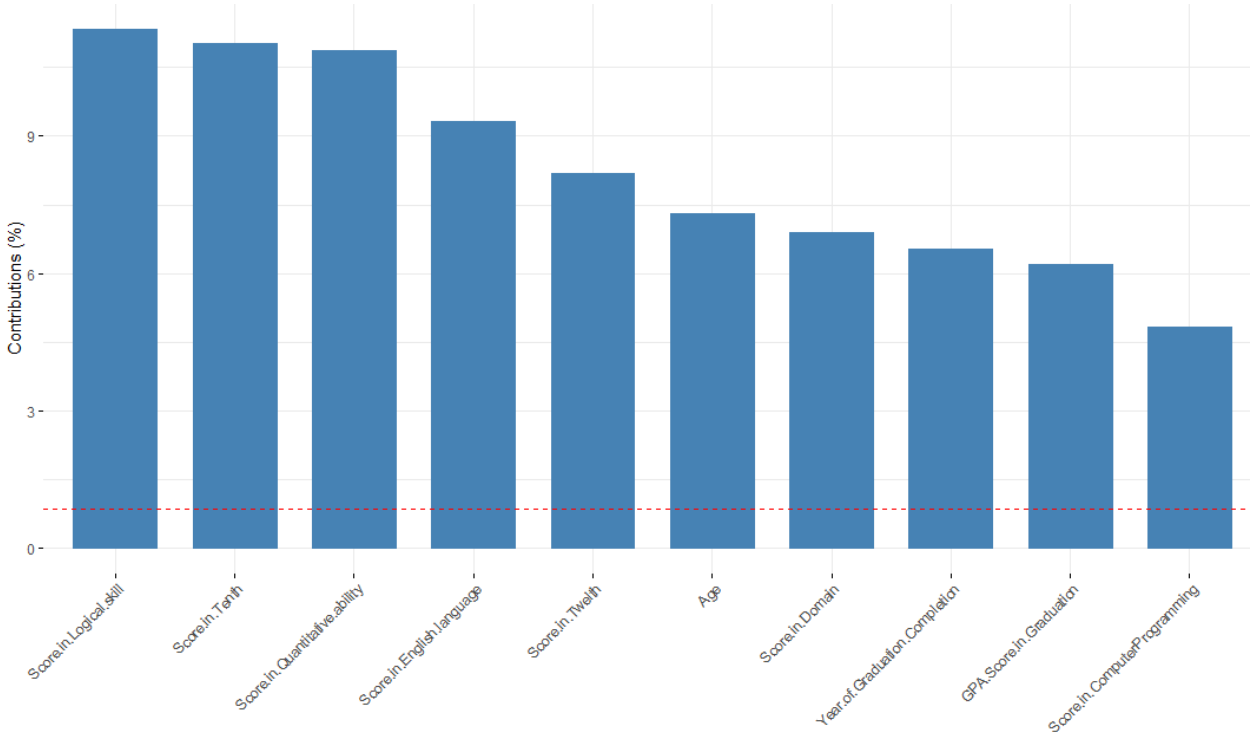


Fig xxviii(b) : Top 10 contributions of the original predictors to PC-1

The second most important PC in the random forest trained using principal components, PC-1 too was observed to denote extreme similarities with the important variables extracted from the first random forest. The variables **"Score.in.Tenth"**, **"Score.in.Quantitative.ability"**, **"Score.in.English"** and **"Score.in.Twelfth"** which contribute the most to PC-1, all featured in the top most important variables of Random forest - 1 trained without principal components.

Based on the evidence presented, it was concluded that the random forest algorithm should be used to predict the **test** values. Upon predicting using the random forest, a **MAPE value of 0.25444 was obtained**

```
> test_rf = predict(model_rf, test[,-39])
> regr.eval(test[,39],test_rf)
         mae          mse         rmse         mape
1.111599e+05 4.622382e+10 2.149972e+05 2.544471e-01
>
```

Fig xxix : Random Forest was used to make predictions on test data, the above metrics were obtaine

Conclusion

A prediction model with decent, if not good, error metrics was trained. To summarize, the data was **cleaned**, **feature engineering** was performed as and when necessary, missing values were **imputed**, data was **explored** to uncover interesting insights, and finally **models were trained and analyzed**, and the best one was chosen to **predict** on unseen data. The limitations, if any, arose due to computational capacities.

Appendix

```
data = read.csv("Indata.csv", sep = ",", header = T, na.strings = " ")
str(data)

#install.packages("rockchalk")
#install.packages("mice")
#install.packages("missForest")
#install.packages("missRanger")
#install.packages("DataExplorer")
#install.packages("ggRandomForests")
#install.packages("rpart.plot")
#install.packages("rattle")

library(scales)
library(stringr)
library(lubridate)
library(rockchalk)
library(plyr)
library(caret)
library(rpart)
library(DMwR)
library(corrplot)
library(DataExplorer)
library(dplyr)
library(missForest)
library(gains)
library(mice)
library(missRanger)
library(ggRandomForests)
library(rpart.plot)
library(rattle)
library(devtools)
#install_github("vqv/ggbiplot")
library(ggbiplot)
library(factoextra)
```

```
sum(is.na(data))
str(data)
summary(data)

data$Date.Of.Birth = as.character(data$Date.Of.Birth)
data$D.O.B = str_sub(data$Date.Of.Birth, 1,-6)
data$Birth.Time = str_sub(data$Date.Of.Birth, start = -5)

data$D.O.B = as_date(data$D.O.B)

data$Y.O.B = year(data$D.O.B)
x = format(Sys.Date(), "%Y")

data$Age = as.numeric(x) - data$Y.O.B
rm(x)

levels(data$Board.in.Twelth)[levels(data$Board.in.Twelth)=="0"] = NA
levels(data$School.Board.in.Tenth)[levels(data$School.Board.in.Tenth)=="0"] =
NA
data$Score.in.Domain[data$Score.in.Domain == -1] = NA

sum(is.na(data)) #11072
nrow(data[!complete.cases(data),]) #6344

data$CityTier = as.factor(data$CityTier)
data$CollegeTier = as.factor(data$CollegeTier)

data$Gap_in_education = data$Year.of.Graduation.Completion -
data$Year.Of.Twelth.Completion
x = c(5,6)
data$Gap_Scenario = ifelse(data$Graduation == "B.Tech/B.E." &
data$Gap_in_education == 4, "ideal",
                    ifelse(data$Graduation == "M.Tech./M.E." &
data$Gap_in_education == 6,"ideal",
                           ifelse(data$Graduation == "MCA" & data$Gap_in_education
%in% x,"ideal","non-ideal")))
```

```
data$Gap_Scenario = as.factor(data$Gap_Scenario)

data$Y.O.B = NULL
data$D.O.B = NULL
data$Birth.Time = NULL
data$Date.Of.Birth = NULL
data$Candidate.ID = NULL
data$CityCode = NULL
data$CollegeCode = NULL
data$Year.Of.Twelth.Completion = NULL

data$CP.opt = as.factor(ifelse(data$Score.in.ComputerProgramming == -100,
'0','1'))
data$ESem.opt = as.factor(ifelse(data$Score.in.ElectronicsAndSemicon == -100,
'0','1'))
data$CS.opt = as.factor(ifelse(data$Score.in.ComputerScience == -100, '0','1'))
data$Mech.opt = as.factor(ifelse(data$Score.in.MechanicalEngg == -100, '0','1'))
data$ElEng.opt = as.factor(ifelse(data$Score.in.ElectricalEngg == -100, '0','1'))
data$Tel.opt = as.factor(ifelse(data$Score.in.TelecomEngg == -100, '0','1'))
data$Civ.opt = as.factor(ifelse(data$Score.in.CivilEngg == -100, '0','1'))

data$Score.in.ComputerProgramming[data$Score.in.ComputerProgramming == -
100] = 0
data$Score.in.ElectronicsAndSemicon[data$Score.in.ElectronicsAndSemicon == -
100] = 0
data$Score.in.ComputerScience[data$Score.in.ComputerScience == -100] = 0
data$Score.in.MechanicalEngg[data$Score.in.MechanicalEngg == -100] = 0
data$Score.in.ElectricalEngg[data$Score.in.ElectricalEngg == -100] = 0
data$Score.in.TelecomEngg[data$Score.in.TelecomEngg == -100] = 0
data$Score.in.CivilEngg[data$Score.in.CivilEngg == -100] = 0

data = data[,c(1:17,24,39,23,38,22,37,21,36,20,35,19,34,18,33,25:32)]


data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                            levs = c("board of secondary education - andhra pradesh",
```

```
                                "board ofsecondary education ap","board of ssc
education andhra pradesh",
                                "apsche","apssc","ap state board for secondary
education",
                                "board of intermediate education","school
secondary education andhra pradesh",
                                "state board of secondary education ap","andhra
pradesh board ssc",
                                "andhra pradesh state board","state board of
secondary education andhra pradesh",
                                "board of secondary education ap","ssc board of
andrapradesh",
                                "ssc-andhra pradesh","ap state board",
                                "board of secondary education andhra pradesh"),
                        newLabel = "Andhra Pradesh")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("pseb","punjab school education board mohali"),
                        newLabel = "Punjab")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("gujarat state board","gujarat
board","gseb","gsheb"),
                        newLabel = "Gujarat")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("cgbse raipur","cgbse"),
                        newLabel = "Chhattisgarh")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("board of secendary education orissa","board of
secondary education (bse) orissa",
                                "board of secondary education odisha","board of
secondary education(bse) orissa",
                                "bsc orissa","board of secondary education
orissa","bse odisha",
                                "hse  orissa","bse orissa","bse"),
```

```
                        newLabel = "Orissa")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("himachal pradesh board","himachal pradesh
board of school education",
                                 "state borad hp"),
                        newLabel = "Himachal Pradesh")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("board of school education harayana","board of
school education haryana",
                                 "hbse","haryana board of school education","hbsc",
                                 "haryana board of school education (hbse)"),
                        newLabel = "Haryana")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("delhi board","delhi public school"),
                        newLabel = "Delhi")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("ua","uttranchal board","uttaranchal shiksha
avam pariksha parishad",
                                 "uttrakhand board","uttaranchal state
board","uttarakhand board",
                                 "board of school education uttarakhand"),
                        newLabel = "Uttarakhand")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("state(karnataka board)","karnataka state
education examination board",
                                 "karnataka secondary school of
examination","karnataka board of higher education",
                                 "karnataka secondary education","karnataka board
of secondary education",
                                 "kea","karnataka secondary board","ksseb(karnataka
state board)",
```

```
                                "karantaka secondary education and examination
borad","ksseb","karnataka sslc board bangalore",
                                "karnataka education board (keeb)","state board of
karnataka","ksbe","karnataka state examination board",
                                "karnataka","karnataka secondory education
board","sslc board","karnataka education board",
                                "kseeb(karnataka secondary education examination
board)","karnataka state secondary education board",
                                "kseeb","karnataka secondary education
examination board","sslc karnataka",
                                "karnataka state board","karnataka secondary
eduction","karnataka secondary education board",
                                "sslc","karnataka board"),
                        newLabel = "Karnataka")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("kerala state board","education board of
kerala","kerala state technical education",
                                "kerala university","kerala","kseb"),
                        newLabel = "Kerala")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("tn state board","tamilnadu matriculation board",
                                "sri kannika parameswari highier secondary school
udumalpet",
                                "stjoseph of cluny matrhrsecschool neyveli
cuddalore district",
                                "tamil nadu state","kalaimagal matriculation higher
secondary school",
                                "tamilnadu state board","stjosephs girls higher sec
school dindigul"),
                        newLabel = "Tamil Nadu")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("seba(assam)","seba"),
                        newLabel = "Assam")
```

```
data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                            levs = c("state bord","board of secondaray
education","sarada high scchool",
                                     "secondary school certificate","ghseb","bharathi
matriculation school",
                                     "bsemp","national public school","secondary school
cerfificate",
                                     "maticulation","board of secondary school
education","anglo indian",
                                     "gyan bharati school","don bosco maatriculation
school","mhsbse",
                                     "stmary's convent inter college","board secondary
education",
                                     "holy cross matriculation hr sec school","mirza
ahmed ali baig",
                                     "hse board","state board","hsc","matric board",
                                     "bse(board of secondary education)","little jacky
matric higher secondary school",
                                     "certificate of middle years program of
ib","matric","board of ssc",
                                     "secondary school education","board of  secondary
education","cluny",
                                     "jawahar navodaya vidyalaya","ms
board","metric","matriculation board",
                                     "secondary state certificate","state board of
secondary education( ssc)",
                                     "board of secondary education","secondary school
of education",
                                     "stmary higher secondary","ssc
regular","hsce","matriculation","hse",
                                     "kiran english medium high
school","state","stateboard","ssc","state board","state board "),
                            newLabel = "other")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                            levs = c("jbse jharkhand","jharkhand secondary board",
                                     "jharkhand secondary examination board (ranchi)",
```

```
                                "jharkhand accademic council",
                                "jharkhand secondary examination board ranchi",
                                "state board (jac ranchi)","jharkhand secondary
education board",
                                "jharkhand academic council","jharkhand acedemic
council","jseb"),
                        newLabel = "Jharkhand")


data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("icse board new delhi","icse board","council for
indian school certificate examination",
                                "cicse","icse"),
                        newLabel = "ICSE")


data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("bihar school examination board
patna","bihar","biharboard",
                                "bihar board","bihar school examination board",
                                "bihar secondary education board patna","bihar
examination board patna",
                                "bsepatna","bseb patna","bseb"),
                        newLabel = "Bihar")
data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("mp-bse","mp state board","madhya pradesh
board","state boardmp board ","mp","mp board bhopal",
                                "mpbse","mpboard","mp board"),
                        newLabel = "Madhya Pradesh")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("up bourd","u p","up borad","bright way college
(up board)","upbhsie","up baord",
                                "up board allahabad","uttar pradesh
board","up","up(allahabad)","u p board",
                                "up bord","upboard","board of high school and
intermediate education uttarpradesh",
```

```
                                "uttar pradesh","up-board","up board"),
                        newLabel = "Uttar Pradesh")
data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("dav public school sec 14","cbse
board","cbsc","aisse",
                                "central board of secondary education new
delhi","cbse[gulf zone]",
                                "central board of secondary education","dav public
school hehal",
                                "cbse","cbse "),
                        newLabel = "cbse")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("j&k state board of school education","j & k
bord","jkbose"),
                        newLabel = "Jammu and Kashmir")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("maharashtra state board of secondary and
higher secondary education",
                                "ssc maharashtra board","maharastra board","pune
board","msbshse pune",
                                "kolhapur divisional board maharashtra","nagpur
board","mumbai board",
                                "maharashtra sate board","maharashtra board
pune",
                                "maharashtra state boar of secondary and higher
secondary education",
                                "maharashtra nasik board","nagpur","latur board",
                                "maharashtra state(latur board)","latur",
                                "maharashtra state board of secondary & higher
secondary education",
                                "aurangabad board","kolhapur",
                                "maharashtra state board of secondary and higher
secondary education pune",
                                "maharashtra state board pune","nagpur divisional
board","nagpur board nagpur",
```

```
                                "maharashtra state board mumbai divisional
board","maharashtra satate board",
                                "pune","nasik","nashik board","sss
pune","maharashtra",
                                "maharashtra state board for ssc","maharashtra
state board",
                                "maharashtra board"),
                        newLabel = "Maharashtra")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("west bengal  board of secondary education",
                                "west bengal board of secondary eucation",
                                "west bengal board of secondary examination
(wbbse)",
                                "west bengal board of secondary education",
                                "state board - west bengal board of secondary
education : wbbse",
                                "wbbsce","wbbse"),
                        newLabel = "West Bengal")

data$School.Board.in.Tenth = combineLevels(data$School.Board.in.Tenth,
                        levs = c("rbse (state board)","secondary education board
of rajasthan",
                                "rbse ajmer","rajasthan board ajmer","secondary
board of rajasthan",
                                "board of secondary education rajasthan(rbse)",
                                "rajasthan board of secondary education","board of
secondary education rajasthan",
                                "rajasthan board","rbse"),
                        newLabel = "Rajasthan")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                      levs = c("bte delhi","board of technical education","board of
technicaleducation delhi"),
                      newLabel = "Delhi")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
```

```
                    levs = c("cgbse raipur","cgbse"),
                    newLabel = "Chhattisgarh")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("himachal pradesh board","himachal pradesh board
of school education"),
                    newLabel = "Himachal Pradesh")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("psbte","punjab state board of technical education &
industrial training",
                             "pseb","punjab state board of technical education &
industrial training chandigarh"),
                    newLabel = "Punjab")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("ahsec"),
                    newLabel = "Assam")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("ua","uttranchal board","uttaranchal shiksha avam
pariksha parishad",
                             "uttrakhand board","uttaranchal state
board","uttarakhand board",
                             "board of school education uttarakhand"),
                    newLabel = "Uttarakhand")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("isc board new delhi","aissce","isc
board","isce","council for indian school certificate examination",
                             "cicse","icse","isc"),
                    newLabel = "ICSE")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("mp-bse","bsemp","madhya pradesh
board","madhya pradesh open school",
```

```
                        "mp board bhopal","mp","mpbse","mpboard","mp
board"),
                    newLabel = "Madhya Pradesh")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("j&k state board of school education","j & k
board","jkbose"),
                    newLabel = "Jammu and Kashmir")


data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("west bengal board of higher secondary
education","west bengal council of higher secondary eucation",
                        "west bengal council of higher secondary examination
(wbchse)",
                        "west bengal council of higher secondary
education","wbscte",
                        "state board - west bengal council of higher secondary
education : wbchse",
                        "west bengal state council of technical
education","wbchse","wbbhse"),
                    newLabel = "West Bengal")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("rbse (state board)","secnior secondary education
board of rajasthan",
                        "rajasthan board ajmer","secondary board of
rajasthan","board of secondary education rajasthan(rbse)",
                        "rajasthan board of secondary education","board of
secondary education rajasthan",
                        "rajasthan board","rbse"),
                    newLabel = "Rajasthan")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("state board of technical education and
training","chsc","scte vt orissa",
```

```
                        "certificate for higher secondary education
(chse)orissa","scte & vt (diploma)",
                        "chse(concil of higher secondary education)","board of
higher secondary orissa",
                        "chse odisha","scte&vt","scte and vt orissa","chse
orissa","chse"),
                    newLabel = "Orissa")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("mbose"),
                    newLabel = "Meghalaya")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("department of technical education
bangalore","karnataka sslc",
                        "dpue","sjrcw","pue","preuniversity board(karnataka)",
                        "dept of pre-university education","pre university board
of karnataka",
                        "karnataka pre-university","karnataka board secondary
education",
                        "karnataka board of university","department of pre-
university education(government of karnataka)",
                        "pre university board","karnataka secondary education
board",
                        "pre university board katnataka","dte","karnatak pu
board","pu board karnataka",
                        "karnataka pre unversity board","karnataka pre
university board",
                        "department of pre-university education","state board
of karnataka",
                        "department of pre university education","karnataka
state examination board",
                        "karanataka secondary board","karnataka
state","karnataka pu board",
                        "karnataka secondary education","jyoti
nivas","karnataka pu",
```

```
                        "department of pre-university eduction","karnataka
education board",
                        "karnataka state pre- university board","karnataka pre-
university board",
                        "directorate of technical education
banglore","karnataka state board",
                        "govt of karnataka department of pre-university
education",
                        "department of pre-university education bangalore","s j
polytechnic",
                        "pu  board karnataka","pre-university board","pu board
karnataka",
                        "p u board karnataka","pu board","pre-
university","pub","kea","pu",
                        "karnataka board"),
                    newLabel = "Karnataka")


data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("board of higher secondary examination
kerala","kerala state board",
                        "kerala state hse board","kerala university","kerala"),
                    newLabel = "Kerala")
data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("gseb/technical education
board","gshseb","gseb","gujarat board","ghseb",
                        "gsheb"),
                    newLabel = "Gujarat")
data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("amravati divisional board","diploma in engg (e &tc)
tilak maharashtra vidayapeeth",
                        "diploma ( maharashtra state board of technical
education)",
                        "hsc maharashtra board","pune board","msbshse
pune","kolhapur divisional board maharashtra",
                        "nagpur board","government polytechnic mumbai
mmbai board",
```

```
                    "stmiras college for girls","maharashtra state boar of
secondary and higher secondary education",
                    "maharashtra nasik board","nagpur","latur
board","diploma(msbte)",
                    "maharashtra board pune","maharashtra state(latur
board)",
                    "maharashtra state board of secondary & higher
secondary education",
                    "msbte (diploma in computer technology)","aurangabad
board","msbte pune",
                    "nagpur divisional board","kolhapur","nagpur board
nagpur",
                    "maharashtra state board mumbai divisional
board","maharashtra satate board",
                    "msbte","ms board","nasik","nashik board","hsc
pune","maharashtra",
                    "latur","maharashtra state board for hsc","maharashtra
state board",
                    "maharashtra board"),
                newLabel = "Maharashtra")
data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                levs = c("bteup","uo board"," upboard","up bourd","u
p","lucknow public college",
                    "upboard","bright way college (up
board)","upbhsie","up baord",
                    "up board allahabad","uttar pradesh board","bte
up","up(allahabad)",
                    "u p board","up bord","board of high school and
intermediate education uttarpradesh",
                    "uttar pradesh","up-board","up board","up"),
                newLabel = "Uttar Pradesh")
data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                levs = c("bseb patna","bihar school examination board
patna","bihar","bihar board",
                    "bihar intermediate education council","bciec
patna","intermediate council patna",
```

```
                        "biec","bihar intermediate education council
patna","biec-patna","bice",
                        "biec patna","bseb"),
                    newLabel = "Bihar")
data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("jiec","jstb jharkhand","jharkhand acamedic council
(ranchi)",
                        "jharkhand accademic council","state board (jac
ranchi)","jharkhand board",
                        "jharkhand academic council","sbte
jharkhand","jharkhand acedemic council"),
                    newLabel = "Jharkhand")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("state board of technical education harayana","ssm
srsecschool","hse",
                        "board of school education harayana","haryana state
board of technical education chandigarh",
                        "technical board punchkula","state board of technical
eduction panchkula",
                        "hbsc","hbse"),
                    newLabel = "Haryana")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                    levs = c("jaswant modern school","mpc","narayana junior
college",
                        "aligarh muslim university","state broad","pre
university","state bord",
                        "department of technical
education","intermideate","puboard","intermediate state board",
                        "all india board","apsb","higher secondary","state
syllabus","board of secondary school of education",
                        "ibe","diploma in computers","state board","sda matric
higher secondary school",
                        "ssc","international baccalaureate (ib) diploma","board
of secondary education",
```

```
                              "higher secondary education","hisher seconadry
examination(state board)",
                              "science college","sri sankara
vidyalaya","intermidiate","nios",
                              "matric board","matriculation","state","staae
board","hsc",
                              "higher secondary state certificate","intermediate
board","intermediate",
                              "stateboard","puc","intermedite","state board "),
                        newLabel = "other")

data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                        levs = c("board of intrmediate education ap","intermediate
board of education",
                              "board of intermidiate examination","intermediate
board of education andhra pradesh",
                              "board fo intermediate education ap","andhra pradesh
board of secondary education",
                              "board of intermeadiate education","borad of
intermediate",
                              "board of intmediate education ap","board of
intermediate ap",
                              "board of intermediate(bie)","baord of intermediate
education",
                              "intermediate board examination","board of
intermidiate","andhra pradesh",
                              " board of intermediate","sbtet","ap board for
intermediate education",
                              "board of intermediate education:ap hyderabad",
"intermediate board of andhra pardesh",
                              "board of intermidiate education ap","andhpradesh
board of intermediate education",
                              "state board of technical education","board of
intermediate education hyderabad",
                              "andhra pradesh state
board","boardofintermediate","andhra board",
```

```
                              "state  board of intermediate education andhra
pradesh","bieap","bie",
                              "board of intermediate education","ipe","board of
intermediate education ap",
                              "sri chaitanya junior kalasala","apbsc","board of
intermediate","ap board",
                              "apbie","ap intermediate board","board of intermediate
education andhra pradesh"),
                       newLabel = "Andhra Pradesh")
data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                       levs = c("cbse board","dav public school sec
14","cbsc","jawahar higher secondary school",
                              "cbse new delhi","dav public school","central board of
secondary education new delhi",
                              "central board of secondary education","dav public
school hehal","cbese",
                              "cbse"),
                       newLabel = "cbse")


data$Board.in.Twelth = combineLevels(data$Board.in.Twelth,
                       levs = c("state board - tamilnadu","stateboard/tamil
nadu","jaycee matriculation school",
                              "tn state board","tamilnadu higher secondary education
board","electonincs and communication(dote)",
                              "hslc (tamil nadu state board)","sri kannika parameswari
highier secondary school udumalpet",
                              "stjoseph of cluny matrhrsecschool neyveli cuddalore
district",
                              "st joseph hr sec school","tamil nadu polytechnic","tamil
nadu state",
                              "ks rangasamy institute of technology","holy cross
matriculation hr sec school",
                              "tamilnadu state board","tamilnadu stateboard","dote
(diploma - computer engg)",
                              "tamil nadu state board","srv girls higher sec school
rasipuram"),
```

```
                    newLabel = "Tamil Nadu")

data$Discipline = combineLevels(data$Discipline,
                    levs = c("electronics", "electronics & instrumentation eng",
"electronics & telecommunications", "electronics and communication
engineering","electronics and computer engineering","electronics and
instrumentation engineering", "electronics engineering", "embedded systems
technology", "telecommunication engineering"),
                    newLabel = "Electronics")


data$Discipline = combineLevels(data$Discipline,
                    levs = c("computer and communication engineering" ,
"computer application", "computer engineering", "computer networking",
"computer science", "computer science & engineering", "computer science and
technology"),
                    newLabel = "Computer.Sci.Engg")

data$Discipline = combineLevels(data$Discipline,
                    levs = c("information & communication technology",
"information science", "information science engineering", "information
technology"),
                    newLabel = "IT")

data$Discipline = combineLevels(data$Discipline,
                    levs = c("applied electronics and instrumentation",
"instrumentation and control engineering", "instrumentation engineering",
"control and instrumentation engineering"),
                    newLabel = "Instrumentation")

data$Discipline = combineLevels(data$Discipline,
                    levs = c("automobile/automotive engineering", "internal
combustion engine", "mechanical & production engineering", "mechanical and
automation", "mechanical engineering"),
                    newLabel = "Mechanical")

data$Discipline = combineLevels(data$Discipline,
```

```
                    levs = c("electrical and power engineering", "electrical
engineering", "electronics and electrical engineering", "power systems and
automation"),
                    newLabel = "Electrical")

data$Discipline = combineLevels(data$Discipline,
                    levs = c("industrial engineering", "industrial & production
engineering", "industrial & management engineering"),
                    newLabel = "Inds.Mgmt")


data$Discipline = combineLevels(data$Discipline,
                    levs = c("ceramic engineering", "metallurgical engineering",
"polymer technology"),
                    newLabel = "MaterialEngg")

data$Discipline = combineLevels(data$Discipline,
                    levs = c("biomedical engineering", "biotechnology"),
                    newLabel = "Biosciences")

#Imputations
names = colnames(data)[colSums(is.na(data)) > 0]
names
#"School.Board.in.Tenth" "Board.in.Twelth"  "Age" "Score.in.Domain"
imputers = data[complete.cases(data),]
imputers_miss = data[!complete.cases(data),]

xc = chisq.test(imputers$School.Board.in.Tenth, imputers$State, simulate.p.value
= T)
xc
corrplot(xc$residuals, is.corr = F)
#data:  imputers$School.Board.in.Tenth and imputers$State
#Chi-squared = 47953, df = NA, p-value = 0.0004998

xd = chisq.test(imputers$School.Board.in.Tenth, imputers$Board.in.Twelth,
simulate.p.value = T)
xd
```

```
corrplot(xd$residuals, is.corr = F)
#data:  imputers$School.Board.in.Tenth and imputers$Board.in.Twelth
#Chi-squared = 185820, df = NA, p-value = 0.0004998
xe = chisq.test(imputers$Board.in.Twelth, imputers$State, simulate.p.value = T)
xe
corrplot(xe$residuals, is.corr =F)
#data:  imputers$Board.in.Twelth and imputers$State
#X-squared = 54817, df = NA, p-value = 0.0004998

'%!in%' = function(x,y)!('%in%'(x,y))

for(i in 1:nrow(imputers_miss)){
  if((imputers_miss$State[i] %in% levels(data$School.Board.in.Tenth)) &
     (is.na(imputers_miss[i,"School.Board.in.Tenth"]) == T)){
    imputers_miss$School.Board.in.Tenth[i] = imputers_miss$State[i]
  }
  if((imputers_miss$State[i] %!in% levels(data$School.Board.in.Tenth)) &
     (is.na(imputers_miss[i,"School.Board.in.Tenth"]) == T)){
    imputers_miss$School.Board.in.Tenth[i] = "other"
  }
}
rm(i)
sum(is.na(imputers_miss$School.Board.in.Tenth))

imputers_miss$Board.in.Twelth[is.na(imputers_miss$Board.in.Twelth)] =
imputers_miss$School.Board.in.Tenth[is.na(imputers_miss$Board.in.Twelth)]
sum(is.na(imputers_miss$Board.in.Twelth))

newdata = rbind(imputers, imputers_miss)

names = colnames(newdata)[colSums(is.na(newdata)) > 0]
names
#"Score.in.Domain" "Age"

set.seed(001)
rand_ind = sample(nrow(newdata))
newdata = newdata[rand_ind,]
```

```
nrow(newdata[complete.cases(newdata),])
imputers = newdata[complete.cases(newdata),]
imputers_miss = newdata[!(complete.cases(newdata)),]

exp = imputers
exp_num = exp[,c(37,17)]
sum(is.na(exp_num))
exp_miss = prodNA(exp_num, noNA = 0.1) #10% missing values introduced in
exp_miss

exp_age = cbind(exp[,-c(1,37,17)],"Age" = exp_miss$Age)
#created a df with 'Age' containing NA's and all other columns excluding 'Pay' and
#'Score.in.Domain' named exp_age
exp_score = cbind(exp[,-c(1,37,17)],"Score.in.Domain" =
exp_miss$Score.in.Domain)

age_train = exp_age[complete.cases(exp_age),]
age_valid = exp_age[!complete.cases(exp_age),]

score_train = exp_score[complete.cases(exp_score),]
score_valid = exp_score[!complete.cases(exp_score),]

rpart_age = rpart(Age ~ ., data= age_train, method="anova")
age_rpart = predict(rpart_age, age_valid)
age_rpart = round(age_rpart)
age_actu = exp_num[is.na(exp_miss$Age),"Age"]
regr.eval(age_actu, age_rpart)
#      mae       mse      rmse      mape
#0.59126365 0.76287051 0.87342459 0.02082801

ggplot(mapping = aes(age_rpart,
age_actu))+geom_point()+geom_abline(intercept = 0, slope = 1)+
  ggtitle("Decision tree - actual vs predicted - Age") + xlab("predicted") +
ylab("actual")

knn_age = knnImputation(exp_age)
```

```
age_knn = knn_age[is.na(exp_age$Age), "Age"]
age_knn = round(age_knn)
regr.eval(age_actu, age_knn)
#       mae       mse       rmse       mape
#0.258970359 0.375975039 0.613168035 0.009108568

ggplot(mapping = aes(age_knn, age_actu))+geom_point()+geom_abline(intercept
= 0, slope = 1)+
  ggtitle("Knn - actual vs predicted - Age") + xlab("predicted") + ylab("actual")

ci_age = centralImputation(exp_age)
age_ci = ci_age[is.na(exp_age$Age), "Age"]
age_ci = round(age_ci)
regr.eval(age_actu, age_ci)
#      mae       mse      rmse      mape
#1.32917317 2.85881435 1.69080287 0.04624213

ggplot(mapping = aes(age_ci, age_actu))+geom_point()+geom_abline(intercept =
0, slope = 1)+
  xlim(24,34) +
  ggtitle("Central Imp. - actual vs predicted - Age") + xlab("predicted") +
ylab("actual")

rpart_score = rpart(Score.in.Domain ~ ., data= score_train, method="anova")
score_rpart = predict(rpart_score, score_valid)
score_actu = exp_num[is.na(exp_miss$Score.in.Domain),"Score.in.Domain"]
regr.eval(score_actu, score_rpart)
#      mae       mse      rmse      mape
#0.09481679 0.01733919 0.13167835 0.17521269

ggplot(mapping = aes(score_rpart*100, score_actu*100))+geom_point()+
  geom_abline(intercept = 0, slope = 1)+
  ggtitle("Dec.Tree - actual vs predicted - Score.in.Domain") +
  xlab("predicted") + ylab("actual")

knn_score = knnImputation(exp_score)
score_knn = knn_score[is.na(exp_score$Score.in.Domain), "Score.in.Domain"]
```

```
regr.eval(score_actu, score_knn)
#       mae       mse       rmse       mape
#0.054632584 0.008399962 0.091651306 0.099354060

ggplot(mapping = aes(score_knn*100, score_actu*100))+geom_point()+
  geom_abline(intercept = 0, slope = 1)+
  ggtitle("KNN - actual vs predicted - Score.in.Domain") +
  xlab("predicted") + ylab("actual")

ci_score = centralImputation(exp_score)
score_ci = ci_score[is.na(exp_score$Score.in.Domain),"Score.in.Domain"]
regr.eval(score_actu, score_ci)
#      mae       mse      rmse      mape
#0.17427507 0.04557488 0.21348273 0.34012735

ggplot(mapping = aes(score_ci*100, score_actu*100))+geom_point()+
  geom_abline(intercept = 0, slope = 1)+
  xlim(35,100)+
  ggtitle("Central Imp - actual vs predicted - Score.in.Domain") +
  xlab("predicted") + ylab("actual")

sum(is.na(age_valid))
rf_age = randomForest(age_train[,-37], age_train[,37], ntree = 120)
age_rf = predict(rf_age, age_valid[,-37])
age_rf = round(age_rf)
regr.eval(age_actu, age_rf)
#       mae       mse       rmse       mape
#0.170826833 0.208268331 0.456364252 0.006040812

ggplot(mapping = aes(age_rf, age_actu))+geom_point()+geom_abline(intercept =
0, slope = 1)+
  xlim(24,34) +
  ggtitle("Random forest - actual vs predicted - Age") + xlab("predicted") +
ylab("actual")
```

```
rf_score = randomForest(score_train[,-37], score_train[,37], ntree = 120)
score_rf = predict(rf_score, score_valid[,-37])
regr.eval(score_actu, score_rf)
#       mae       mse       rmse       mape
#0.034948234 0.004105507 0.064074227 0.067670623

ggplot(mapping = aes(score_rf*100, score_actu*100))+geom_point()+
  geom_abline(intercept = 0, slope = 1)+ ylab("actual") +
  ggtitle("Random forest - actual vs predicted - Score.in.Domain") +
xlab("predicted")

age_newdata = predict(rf_age, newdata[is.na(newdata$Age),-c(1,17,37)])
age_newdata = round(age_newdata)
newdata$Age[is.na(newdata$Age)] = age_newdata

score_newdata = predict(rf_score, newdata[is.na(newdata$Score.in.Domain), -
c(1,17,37)])
newdata$Score.in.Domain[is.na(newdata$Score.in.Domain)] = score_newdata

rm(age_newdata, score_newdata,exp, exp_age, exp_miss, exp_num,
exp_score,score_actu, score_rf,
   age_actu, age_rf,ci_score, score_ci,knn_score, score_knn,rpart_score,
score_rpart,
   ci_age, age_ci,knn_age, age_knn,rpart_age, age_rpart,rand_ind, imputers,
imputers_miss,
   xc,xd,xe)

sum(is.na(newdata))
str(newdata)
summary(newdata)
colnames(newdata)

newdata$Discipline = combineLevels(newdata$Discipline,
                          levs = c("other","MaterialEngg","Biosciences",
                                "mechatronics","aeronautical engineering",
                                "chemical engineering"),
```

```
                        newLabel = "other")


newdata$School.Board.in.Tenth =
combineLevels(newdata$School.Board.in.Tenth,
                                levs = c("Himachal Pradesh","Chhattisgarh",
                                      "Kerala","Assam","Bihar",
                                      "Jammu and Kashmir","other"),
                                newLabel = "Other")

newdata$Board.in.Twelth = combineLevels(newdata$Board.in.Twelth,
                                levs = c("Himachal Pradesh","Chhattisgarh",
                                      "Kerala","Assam","Meghalaya",
                                      "Delhi","other"),
                                newLabel = "Other")
visdata = newdata

create_report(visdata)
ggplot(visdata) + geom_point(aes(x = Age, y = Pay_in_INR/10**6, color = Gender))
+
 ggtitle("Scatterplot between Age and Salary \n with distinction on the basis of
Gender") +
 xlab("Age (Years)") + ylab("Salary(million INR)") +
 theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))

visdata$std_GPA = scale(visdata$GPA.Score.in.Graduation)
visdata$std_GPA = rescale(visdata$std_GPA, to = c(0,10))

ggplot(visdata) + geom_point(aes(x = std_GPA, y = Pay_in_INR/10**6, color =
Gender)) +
 ggtitle("Scatterplot between GPA(grad) and Salary \n with distinction on the
basis of Gender"
 ) +
 xlab("GPA (Grad)") + ylab("Salary(million INR)") +
 theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))
```

```
visdata$std_XII = scale(visdata$Score.in.Twelth)
visdata$std_XII = rescale(visdata$std_XII, to = c(0,10))

ggplot(visdata) + geom_point(aes(x = std_XII, y = Pay_in_INR/10**6, color =
Gender)) +
  ggtitle("Scatterplot between GPA(12th) and Salary \n with distinction based on
Gender")+
  xlab("GPA (12th)") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))

visdata$std_english = scale(visdata$Score.in.English.language)
visdata$std_english = rescale(visdata$std_english, to = c(0,10))

ggplot(visdata) + geom_point(aes(x = std_english, y = Pay_in_INR/10**6, color =
Gender)) +
  ggtitle("Plot between communication skills and Salary \n with distinction based
on Gender")+
  xlab("English score") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))


ggplot(visdata) + geom_point(aes(x = Graduation, y = Pay_in_INR/10**6, color =
Gender, shape =  Gap_Scenario)) +
  ggtitle("Plot between Graduation and Salary \n with distinction based on
Gap_Scenario \n & Gender")+
  xlab("Graduation degree") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))


ggplot(visdata) + geom_point(aes(x = Graduation, y = Pay_in_INR/10**6, color =
Gap_Scenario)) +
  ggtitle("Plot between Graduation and Salary \n with distinction based on
Gap_Scenario")+
  xlab("Graduation degree") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic")
tier2 = visdata[which(visdata$CollegeTier == '2'),]
tier1 = visdata[which(visdata$CollegeTier == '1'),]
```

```
#Tier 1 is a better rating, Tier 2 is lower

ggplot(tier1) + geom_point(aes(x = Gender, y = Pay_in_INR/10**6)) +
  ggtitle("Plot between Gender and Salary \n for tier 1 college grads") +
  xlab("Gender") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))
#However, Tier 2 colleges seem to get a better salary

ggplot(tier2) + geom_point(aes(x = Gender, y = Pay_in_INR/10**6)) +
  ggtitle("Plot between Gender and Salary \n for tier 2 college grads") +
  xlab("Gender") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))
#Tier 1 females seem to earn substantially less than their tier 1 peers

table(visdata$Discipline)

ggplot(visdata) + geom_point(aes(x = Discipline, y = Pay_in_INR/10**6, color =
Gender)) +
  ggtitle("Plot between Grad Discipline and Salary \n with distinction on the basis
of Gender") +
  xlab("Discipline") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"))

civil_engineering = mean(visdata$Pay_in_INR[which(visdata$Discipline == "civil
engineering")])
Electronics = mean(visdata$Pay_in_INR[which(visdata$Discipline ==
"Electronics")])
Computer.Sci.Engg = mean(visdata$Pay_in_INR[which(visdata$Discipline ==
"Computer.Sci.Engg")])
IT = mean(visdata$Pay_in_INR[which(visdata$Discipline == "IT")])
Instrumentation = mean(visdata$Pay_in_INR[which(visdata$Discipline ==
"Instrumentation")])
Mechanical = mean(visdata$Pay_in_INR[which(visdata$Discipline ==
"Mechanical")])
Electrical = mean(visdata$Pay_in_INR[which(visdata$Discipline == "Electrical")])
Inds.Mgmt = mean(visdata$Pay_in_INR[which(visdata$Discipline ==
"Inds.Mgmt")])
```

```
other = mean(visdata$Pay_in_INR[which(visdata$Discipline == "other")])

dis =
c(civil_engineering,Electronics,Computer.Sci.Engg,IT,Instrumentation,Mechanical,
      Electrical,Inds.Mgmt,other)
Disciplines = c("Civi","Electronics","Computr.Sci","IT","Instrument",
            "Mechanical","Electrical","Inds.Mgmt","other")

ggplot() + geom_col(aes(x = Disciplines, y = dis/10**5)) +
  ggtitle("Plot describing Average Salary for different fields") +
  xlab("Discipline") + ylab("Salary(100k INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"),
      axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(visdata) + geom_point(aes(x = School.Board.in.Tenth, y =
Pay_in_INR/10**6, color = Gender)) +
  ggtitle("Plot between Tenth Board and Salary \n with distinction on the basis of
Gender") +
  xlab("Discipline") + ylab("Salary(million INR)") +
  theme(plot.title = element_text(color="#993333", size=14, face="bold.italic"),
      axis.text.x = element_text(angle = 90, hjust = 1))

rm(dis, Disciplines, civil_engineering, Electronics, Electrical, Computer.Sci.Engg, IT,
   Instrumentation, Mechanical, other, Inds.Mgmt,visdata, tier1, tier2)
finale = newdata
set.seed(914)
frodo = createDataPartition(finale$Pay_in_INR, p = 0.7, list = F)
train = finale[frodo,]
test = finale[-frodo,]
set.seed(915)
aragorn = createDataPartition(train$Pay_in_INR, p = 0.9, list = F)
train = train[aragorn,]
valid = train[-aragorn,]

numeric_train = Filter(is.numeric, train)
categ_train = Filter(is.factor, train)
```

```
numeric_valid = Filter(is.numeric, valid)
categ_valid = Filter(is.factor, valid)

numeric_test = Filter(is.numeric, test)
categ_test = Filter(is.factor, test)

numeric_train = scale(numeric_train[,-1])
numeric_valid = scale(numeric_valid[,-1])
numeric_test = scale(numeric_test[,-1])

train = as.data.frame(cbind(numeric_train, categ_train, "Pay_in_INR" =
train$Pay_in_INR))
valid = as.data.frame(cbind(numeric_valid, categ_valid, "Pay_in_INR" =
valid$Pay_in_INR))
test = as.data.frame(cbind(numeric_test, categ_test, "Pay_in_INR" =
test$Pay_in_INR))

train = train[,c(1:8,9,31,10,32,11,33,12,34,13,35,14,36,15,37,16:30,38,39)]
valid = valid[,c(1:8,9,31,10,32,11,33,12,34,13,35,14,36,15,37,16:30,38,39)]
test = test[,c(1:8,9,31,10,32,11,33,12,34,13,35,14,36,15,37,16:30,38,39)]

#tuneRF(train[,-39], train[,39], ntreeTry = 120)
model_rf = randomForest(train[,-39], train[,39], ntree = 100)
rf_raw = predict(model_rf,valid[,-39])
regr.eval(valid[,39],rf_raw)
#       mae        mse       rmse       mape
#5.988434e+04 1.263334e+10 1.123981e+05 1.245342e-01
plot(gg_vimp(model_rf))
oob = gg_error(model_rf)
plot(oob)
varImpPlot(model_rf)

model_dt = rpart(Pay_in_INR~.,train, method = "anova")
dt_raw = predict(model_dt, valid[,-39])
regr.eval(valid[,39],dt_raw)
#       mae        mse       rmse       mape
#1.662965e+05 7.982575e+10 2.825345e+05 3.483401e-01
```

```
rpart.rules(model_dt, clip.facs = T)
plot(model_dt)
prp(model_dt)
fancyRpartPlot(model_dt)
summary(model_dt)

model_lm = lm(Pay_in_INR~., data = train)
lm_raw = predict(model_lm, valid[,-39])
regr.eval(valid[,39],lm_raw)
#       mae         mse         rmse        mape
#1.810622e+05 8.996289e+10 2.999381e+05 3.549146e-01
summary(model_lm)

train_dum = dummyVars("~.", data = train)
dum_train = as.data.frame(predict(train_dum,train))

valid_dum = dummyVars("~.", data = valid)
dum_valid = as.data.frame(predict(valid_dum, valid))

test_dum = dummyVars("~.", data = test)
dum_test = as.data.frame(predict(test_dum, test))

PC = prcomp(dum_train[,-120])
std_dev = PC$sdev
pr_var = std_dev^2
summary(PC)
prop_varex = pr_var/sum(pr_var)

#SCREE PLOT
plot(prop_varex, xlab = "Principal Component",
    ylab = "Proportion of Variance Explained",
    type = "b")
#cumulative scree plot
plot(cumsum(prop_varex), xlab = "Principal Component",
    ylab = "Cumulative Proportion of Variance Explained",
    type = "b")
```

```
ggplot(mapping = aes(x = 1:119, y =
cumsum(prop_varex)*100))+geom_line()+geom_point()+
  ggtitle("cumulative screeplot for train data") + xlab("Number of Principal
Components") +
  ylab("Percentage variance explained")

train_pca = data.frame("Pay_in_INR" = train$Pay_in_INR, PC$x)
train_pca = train_pca[,1:51]
test_pca =  data.frame(predict(PC,dum_test[,-120]))
valid_pca = data.frame(predict(PC,dum_valid[,-120]))
model_rf2 = randomForest(train_pca[,-1], train_pca[,1], ntree = 100)
rf_pca = predict(model_rf2,valid_pca[,1:50])
regr.eval(valid$Pay_in_INR, rf_pca)
#       mae         mse        rmse        mape
#6.229023e+04 1.291048e+10 1.136243e+05 1.337623e-01
plot(gg_vimp(model_rf2))
varImpPlot(model_rf2)

model_dt2 = rpart(Pay_in_INR~., data = train_pca, method = "anova")
dt_pca = predict(model_dt2, valid_pca[,1:50])
regr.eval(valid$Pay_in_INR, dt_pca)
#       mae         mse        rmse        mape
#1.701781e+05 8.854554e+10 2.975660e+05 3.550161e-01

model_lm2 = lm(Pay_in_INR~.,data = train_pca)
lm_pca = predict(model_lm2, valid_pca[,1:50])
regr.eval(valid$Pay_in_INR, lm_pca)
#       mae         mse        rmse        mape
#1.830637e+05 9.280413e+10 3.046377e+05 3.584974e-01

ggplot(mapping = aes(rf_raw/10**6,
valid$Pay_in_INR/10**6))+geom_point()+geom_abline(intercept = 0, slope = 1)+
  ggtitle("Random forest - actual vs predicted - Pay_in_INR") + xlab("Predicted
salary(million INR)") + ylab("Actual salary(million INR)")
ggplot(mapping = aes(dt_raw/10**6,
valid$Pay_in_INR/10**6))+geom_point()+geom_abline(intercept = 0, slope = 1)+
```

```
  ggtitle("Decision Tree - actual vs predicted - Pay_in_INR (million INR)") +
xlab("Predicted salary") + ylab("Actual salary")
ggplot(mapping = aes(lm_raw/10**6,
valid$Pay_in_INR/10**6))+geom_point()+geom_abline(intercept = 0, slope = 1)+
  ggtitle("Linear regression - actual vs predicted - Pay_in_INR (million INR)") +
xlab("Predicted salary") + ylab("Actual salary")
ggplot(mapping = aes(rf_pca/10**6,
valid$Pay_in_INR/10**6))+geom_point()+geom_abline(intercept = 0, slope = 1)+
  ggtitle("Random forest PCA actual vs predicted - Pay_in_INR (million INR)") +
xlab("Predicted salary") + ylab("Actual salary")
ggplot(mapping = aes(dt_pca/10**6,
valid$Pay_in_INR/10**6))+geom_point()+geom_abline(intercept = 0, slope = 1)+
  ggtitle("Decision Tree PCA actual vs predicted - Pay_in_INR (million INR)") +
xlab("Predicted salary") + ylab("Actual salary")
ggplot(mapping = aes(lm_pca/10**6,
valid$Pay_in_INR/10**6))+geom_point()+geom_abline(intercept = 0, slope = 1)+
  ggtitle("Linear regression PCA actual vs predicted - Pay_in_INR (million INR)") +
xlab("Predicted salary") + ylab("Actual salary")
gain_rf = gains(valid[,39],rf_raw)
plot(c(0,gain_rf$cume.pct.of.total*sum(valid[,39]))~c(0,gain_rf$cume.obs),
    xlab="# cases", ylab="Cumulative Salary (INR)", main="Lift Chart - Random
Forest",
    col = "blue1", type="l")

lines(c(0,sum(valid[,39]))~c(0,nrow(valid)), col="brown2", lty=2)
barplot(gain_rf$mean.resp/mean(valid[,39]), names.arg = gain_rf$depth,
      xlab = "Percentile", ylab = "Mean Response", main = "Decile-wise lift chart for
Random Forest",
      col = "coral1")
gain_dt = gains(valid[,39], dt_raw, groups = 10)
plot(c(0,gain_dt$cume.pct.of.total*sum(valid[,39]))~c(0,gain_dt$cume.obs),
    xlab = "# cases", ylab="Cumulative Salary (INR)", main="Lift Chart - Decision
Tree",
    col = "black", type="l")
lines(c(0,sum(valid[,39]))~c(0,nrow(valid)), col="brown2", lty=2)
barplot(gain_dt$mean.resp/mean(valid[,39]), names.arg = gain_dt$depth,
```

```
      xlab = "Percentile", ylab = "Mean Response", main = "Decile-wise lift chart for
Decision Tree",
      col = "coral1")
gain_lm = gains(valid[,39], lm_raw, groups = 10)
plot(c(0,gain_lm$cume.pct.of.total*sum(valid[,39]))~c(0,gain_lm$cume.obs),
   xlab = "# cases", ylab="Cumulative Salary (INR)", main="Lift Chart",
   col = "black", type="l")
barplot(gain_lm$mean.resp/mean(valid[,39]), names.arg = gain_lm$depth,
      xlab = "Percentile", ylab = "Mean Response", main = "Decile-wise lift chart for
Linear Regression",
      col = "coral1")
#Principal Component Contributions
fviz_pca_var(PC, col.var = "black")
fviz_contrib(PC, choice = "var", axes = 29,top = 10)
fviz_contrib(PC, choice = "var", axes = 1,top = 10)
#Test predictions
test_rf = predict(model_rf, test[,-39])
regr.eval(test[,39],test_rf)
```

Individual Report

Yashvardhan Rathi

This being my first project in R-programming, I was really excited about working on the same. We had a tough time deciding our dataset. We had shortlisted on 2 different datasets to work on. Our primary dataset was the FlightDelays dataset but we dropped it since it was already used in class exercises. We ended up finalizing our 2nd dataset, the dataset which involved predicting the possible earnings package for the candidate based on the profile i.e., with the help of various factors such as demographic/personal details, academic background, different soft skills details. Once we finalized our dataset we realized that the actual tough time is going to start now.

We started by cleaning the data and transforming some variables, I researched on various imputation techniques in R like imputing using Hmisc package, Amelia Package, tree imputation, random forest. After going through results from Monica we decided to go forward with random forest results since it gave the lowest error rates and was computationally inexpensive. I tested the data with PCA and implemented decision tree and random forest algorithms to improve the error metric. Creating innovative visualizations was also done since we had to put lot of them in the report. I also learnt how to write an analysis report and what to include so that it is best understood.

In the project I learnt a lot about coding in R and how to start working with a dataset. It was my first time getting my hands dirty with analysis and I enjoyed every bit of it. I learnt about different data repositories, cleaning and exploration of data, transforming variables, running and evaluating different models to find the results which give the best predictions and displaying valuable visualizations of the predictions. We faced problems while coding and working on problems which were new. We made extensive use of gitHub and StackOverflow to solve a lot of coding related problems. Sometimes if we used to have problems applying or interpreting an algorithm we made use of our textbook, YouTube videos and guidance from the professor. Different tasks for the project were divided among the 5 of us. All the members in the group were well organized. Everyone usually had a list of questions/doubts ready to be asked to the other members to check-in with progress made by everyone. Apart from the group meetings there was lot of communication on the personal level with each other member. Virtual and immediate communication was more important since most of the doubts had to be solved immediately. Whenever the lot met as a group there was a brainstorming session which helped us get valuable insights and do our remaining work efficiently. There was no negativity in the group and everyone spoke what they wanted to. We were pretty clear in all the decisions made for the project and always made sure that they were mutually unified decisions.

Rinda was the one who helped everyone a lot since he was the only one who had worked on R before. I really appreciate his help. He was patient and composed at most almost all times. He worked mainly on data gathering, data exploration and model analysis. **Aayushi** was very accommodating and optimistic at all times during the project. She was jovial and was equipped with food every time. Model training and

analysis was mainly performed by her. **Vikram** stayed near my apartment so we used to meet up often and do work together. He did a lot of data cleaning and feature engineering. Mine and Vikram's work had a lot of connections so the fact that he lived nearby really helped a lot. **Monica** and me also had similar jobs assigned so we had a lot of interaction and work together which we managed to finish pretty efficiently. She worked on some imputations and algorithm techniques too. I am happy with all the group members since I connected with them on a professional level as well as a personal level. Would love to work with this pack again!

After completing this project I feel very confident about doing more analysis projects and taking up projects on my own. It was like doing a crash course of the whole BUAN 6356 class. I have gained important domain and technical knowledge. Looking forward I intend to start working on different types of projects and improve my skillset.