SCHOOL OF ENGINEERING AND TECHNOLOGY

COURSEWORK FOR THE BSC (HONS) INFORMATION SYSTEMS (BUSINESS ANALYTICS); YEAR 3

ACADEMIC SESSION AUGUST 2021; SEMESTER 7, 8, 9

IST2334: WEB AND NETWORK ANALYTICS

DEADLINE: 3rd December 2021 4:00PM

GROUP: 6

INSTRUCTIONS TO CANDIDATES

- This assignment will contribute 25% to your final grade.
- This is a group assignment. Each group consists of 4-5 members.

IMPORTANT

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

- Coursework submitted after the deadline but within 1 week will be accepted for a maximum mark of 40%.
- Work handed in following the extension of 1 week after the original deadline will be regarded as a non-submission and marked zero.

Students' declaration:

	(Name)	**(ID)**	**(Signature)**
We 1.	Alicia Chong Tsui Ying	20074290	*Alicia*
2.	Wong Yi Qing	19028570	*Yi qing*
3.	Kusselin Yuthasax A/P Prak Chuap	18094409	*Kusselin*
4.	Yap Yee Mun	19001189	*Yee Mun*
5.	Soh Wee Chee	20013520	*Wee Chee*

received the assignment and read the comments.

Academic Honesty Acknowledgement

"We (names stated above) verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment."

1) Alicia Chong Tsui Ying	Alicia (3/12/2021) (Student's signature / Date)
2) Wong Yi Qing	Yi qing (3/12/2021) (Student's signature / Date)
3) Kusselin Yuthasax A/P Prak Chuap	Kusselin (3/12/2021) (Student's signature / Date)
4) Yap Yee Mun	Yee Mun (3/12/2021) (Student's signature / Date)
5) Soh Wee Chee	Wee Chee (3/12/2021) (Student's signature / Date)

Marking rubrics

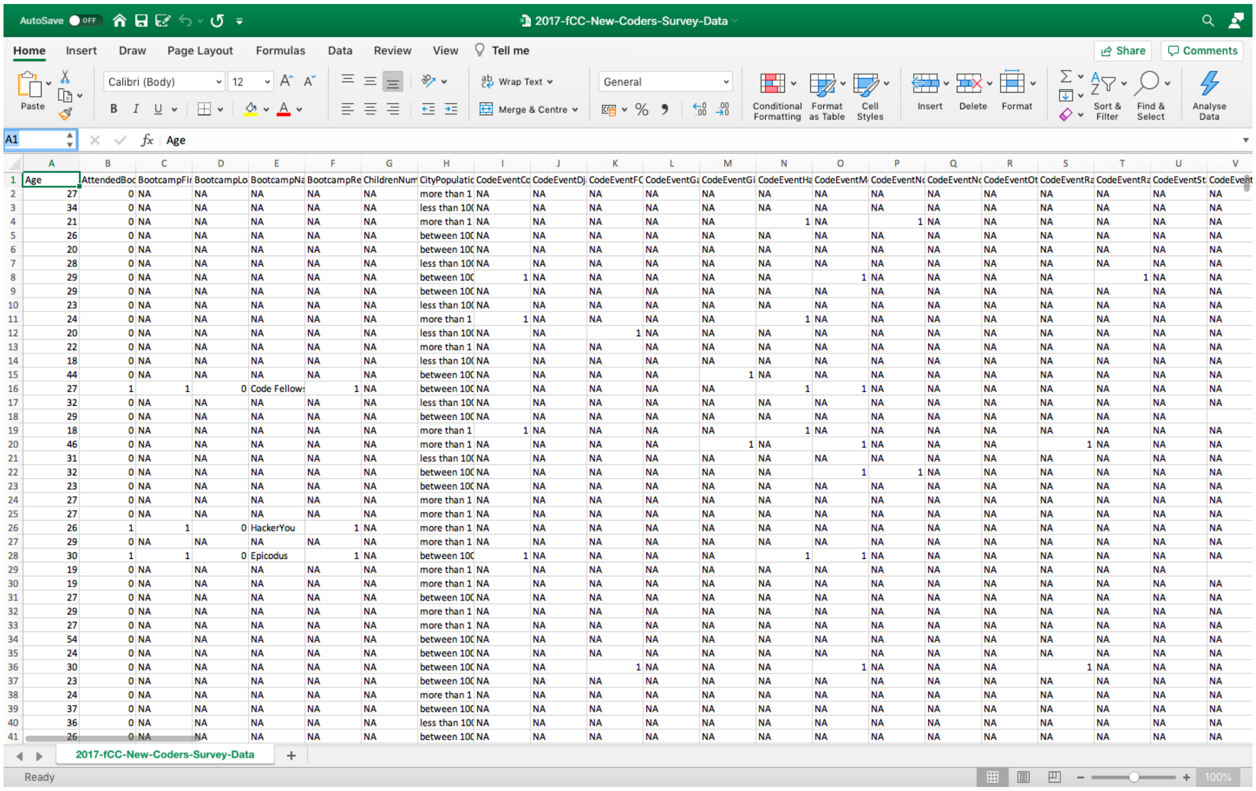
	Excellent	Good	Adequate	Unsatisfactory
	9 - 10	6 - 8.99	4 - 5.99	0 – 3.99
Introduction and motivation of the work [10%]				
	9 - 10	6 - 8.99	4 - 5.99	0 – 3.99
Elaboration of the data sets [10%]				
	13.5 - 15	9 – 13.49	6 – 8.99	0 – 5.99
Presentation of the analysis – techniques used, rationale, results and explanation [15% x 3]				
	9 - 10	6 - 8.99	4 - 5.99	0 – 3.99
Coding in R [10%]				
	4.5 - 5	3 – 4.49	2 – 2.99	0 – 1.99
Lesson learned [5%]				
	9 - 10	6 - 8.99	4 - 5.99	0 – 3.99
Individual Reflections [10%]				
	9 - 10	6 - 8.99	4 - 5.99	0 – 3.99
Formatting, grammar, and style of writing [10%]				

Table of Contents

1. Introduction and motivation of the work

Coding has emerged as the number 1 in-demand skill in recent years. Technology disruption is almost everywhere that significantly alters the way consumers, industries, and businesses operate. After the pandemic of Covid-19, there's a surge of E-commerce, online news sites, ride-sharing apps, GPS systems, and more which illustrates the importance of technologies and the skill of coding. Coding is one of the most important skills of current for future generations to demystify the digital world. Customer expectations are also evolving as they prefer a speedy buying experience along with personalized product recommendations. Therefore, services provided by programmers are a key success factor of a business. Apart from software and application programmers, business analytics, data analytics also acquire coding skills. The reason for this is because data is becoming more valuable for business decision-making insights and of course resulting in data breaches on the other hand. Thus, the number of cyber security jobs is also increasing. Data is so essential along with technology disruption in the current digital world, we couldn't deny the importance of coding skills. Therefore, we are interested in looking into insights of new coder surveys to understand more about what the population and generation think. The dataset used was from an open dataset of freeCodeCamp's surveys. This dataset surveys thousands of people who learn coding for less than 5 years. There are more than 31,000 new coders who participated and responded to this survey. According to observations, 67% of them live outside the United States and 21% are women. They have been coding for an average of 21 months and 17% of them have already landed their first developer job. The median age among the respondents is around 30 years old.

2. Elaboration of the data sets



A1 | fx Age

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Age	AttendedBoo	BootcampFir	BootcampLo	BootcampNa	BootcampRe	ChildrenNum	CityPopulatio	CodeEventCo	CodeEventDj	CodeEventFC	CodeEventGa	CodeEventGi	CodeEventHa	CodeEventM	CodeEventNo	CodeEventNo	CodeEventOt	CodeEventRa	CodeEventRa	CodeEventSt	CodeEvent
2	27	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	34	0	NA	NA	NA	NA	NA	less than 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	21	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	1	NA	1	NA	NA	NA	NA	NA	NA
5	26	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
6	20	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
7	28	0	NA	NA	NA	NA	NA	less than 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
8	29	0	NA	NA	NA	NA	NA	between 100	1	NA	NA	NA	NA	NA	1	NA	NA	NA	NA	1	NA	NA
9	29	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
10	23	0	NA	NA	NA	NA	NA	less than 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	24	0	NA	NA	NA	NA	NA	more than 1	1	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA
12	20	0	NA	NA	NA	NA	NA	less than 100	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
13	22	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
14	18	0	NA	NA	NA	NA	NA	less than 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
15	44	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
16	27	1	1	0	Code Fellows	1	NA	between 100	NA	NA	NA	NA	NA	1	1	NA	NA	NA	NA	NA	NA	NA
17	32	0	NA	NA	NA	NA	NA	less than 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
18	29	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
19	18	0	NA	NA	NA	NA	NA	more than 1	1	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA
20	46	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	1	NA	1	NA	NA	NA	1	NA	NA	NA
21	31	0	NA	NA	NA	NA	NA	less than 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
22	32	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	1	1	NA	NA	NA	NA	NA	NA
23	23	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
24	27	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
25	27	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
26	26	1	1	0	HackerYou	1	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
27	29	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
28	30	1	1	0	Epicodus	1	NA	between 100	1	NA	NA	NA	NA	1	1	NA	NA	NA	NA	NA	NA	NA
29	19	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
30	19	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
31	27	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
32	29	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
33	27	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
34	54	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
35	24	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
36	30	0	NA	NA	NA	NA	NA	between 100	NA	NA	1	NA	NA	NA	1	NA	NA	NA	1	NA	NA	NA
37	23	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
38	24	0	NA	NA	NA	NA	NA	more than 1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
39	37	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
40	36	0	NA	NA	NA	NA	NA	less than 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
41	26	0	NA	NA	NA	NA	NA	between 100	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

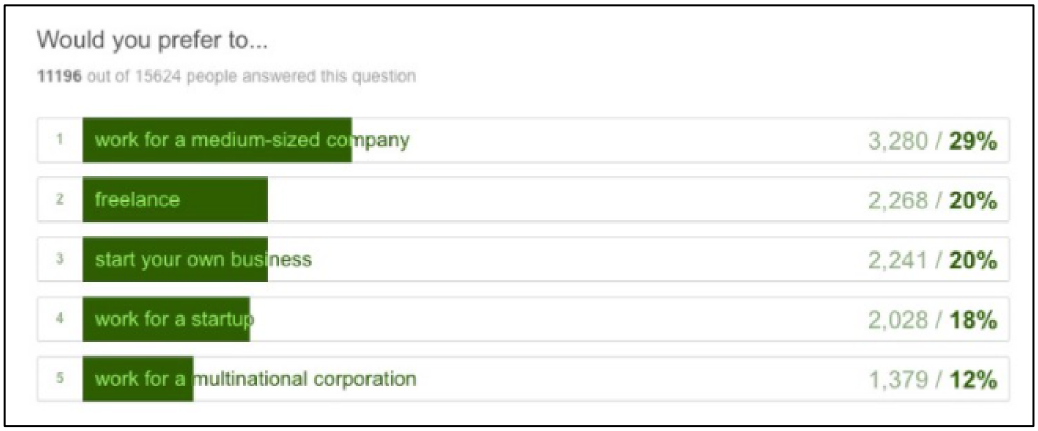
Figure 1: Snippet of Data set in csv format

This dataset is an open dataset from FreeCodeCamp's New Coders Survey in 2017 that consists of a total of 48 survey questions. As a basic overview, the given data set contains 138 columns and 18175 entries of information regarding respondents' demographics and their coding preferences. Many N/A are identified as the attribute are categorical data with multiple options. The selected responses are marked as "1". Moreover, some of the responses such as whether they have attended bootcamp, are underemployed, or if they are a software developer are recorded in the form of binary "0" or "1" (see figure 1).

According to an article by Quincy Larson (2016), many interesting insights were gathered by analyzing the FreeCodeCamp's New Coders Survey in 2016. Some of the statistics and figures (see figure 2 &3) from this article that caught our interest are listed below:

- Of the 15,655 people who responded to the survey, 21% are women
- Of the 15,655 people who responded to the survey, their median age is 27 years old

- 40% want to either freelance or start their own business.
- 42% consider themselves under-employed (working a job that is below their education level)

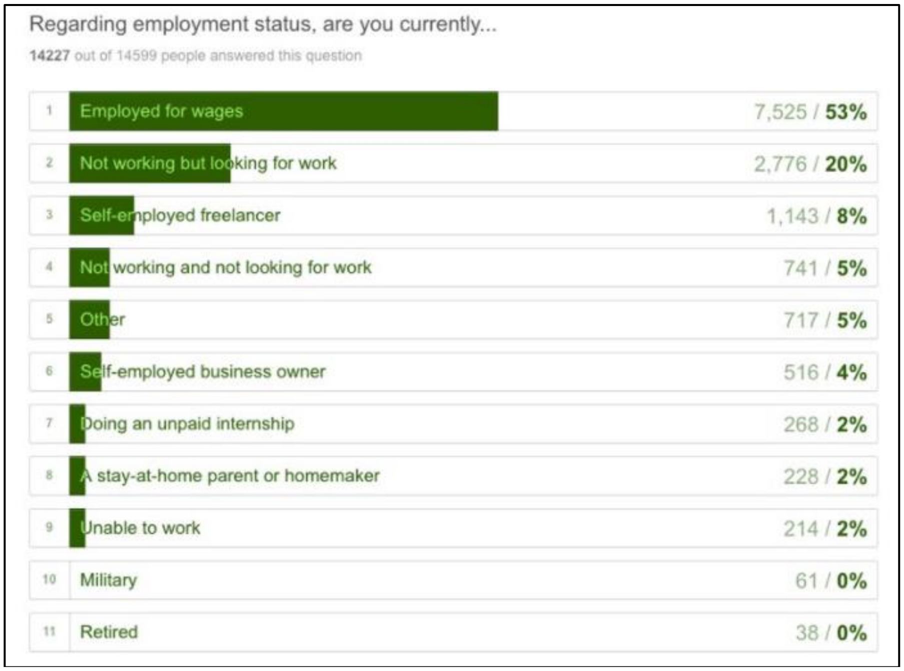


Would you prefer to...

11196 out of 15624 people answered this question

1	work for a medium-sized company	3,280 / 29%
2	freelance	2,268 / 20%
3	start your own business	2,241 / 20%
4	work for a startup	2,028 / 18%
5	work for a multinational corporation	1,379 / 12%

Figure 2:Frequency chart of Respondent's job preferences (source from FreeCodeCamp)



Regarding employment status, are you currently...

14227 out of 14599 people answered this question

1	Employed for wages	7,525 / 53%
2	Not working but looking for work	2,776 / 20%
3	Self-employed freelancer	1,143 / 8%
4	Not working and not looking for work	741 / 5%
5	Other	717 / 5%
6	Self-employed business owner	516 / 4%
7	Doing an unpaid internship	268 / 2%
8	A stay-at-home parent or homemaker	228 / 2%
9	Unable to work	214 / 2%
10	Military	61 / 0%
11	Retired	38 / 0%

Figure 3: Frequency chart of Respondent's current employment status (source from FreeCodeCamp)

This information induced us to do our own analysis regarding the following aspects, which are the demographics of the respondents, what are the variable's affecting the respondent's income, how is the respondent's employment status and ideal job preferences. For this assignment, we are interested to do analysis on the FreeCodeCamp's 2017 New Coders Survey to see if there is any distinction between the findings from the survey in 2016 and 2017.

3. Presentation of the three analyses

The three areas of our analysis are Exploratory analysis, Predictive Analysis with Logistic Regression Analysis, and Predictive Analysis with Decision Tree Analysis. With Exploratory analysis, we would like to perform initial investigations on the data so that we can expect to discover patterns, spot anomalies, and find interesting insights with graphical representations before making further analysis. After getting an overview of the dataset, we are interested in finding correlations between variables and seeing if we could make predictions based on the data that we have, therefore, regression analysis and decision tree analysis were carried out.

3.1 Exploratory Analysis

For this Exploratory Analysis, we are interested in getting a better understanding of our dataset. The first thing that we would like to explore is the demographics of the respondents of this survey; therefore, we looked at the respondent's gender and age. We also performed analysis on the respondents' income and job status as well as job preferences.

To understand the gender distribution of the respondents, a pie chart was created with their percentages being labeled. As shown in figure 4, approximately 79.88% of the respondents are male and 18.97% of the respondents are female. Genderqueer, trans and agender have a percentage of 0.467%, 0.45%, and 0.24% respectively. This suggests that the programming field is still mainly dominated by men.

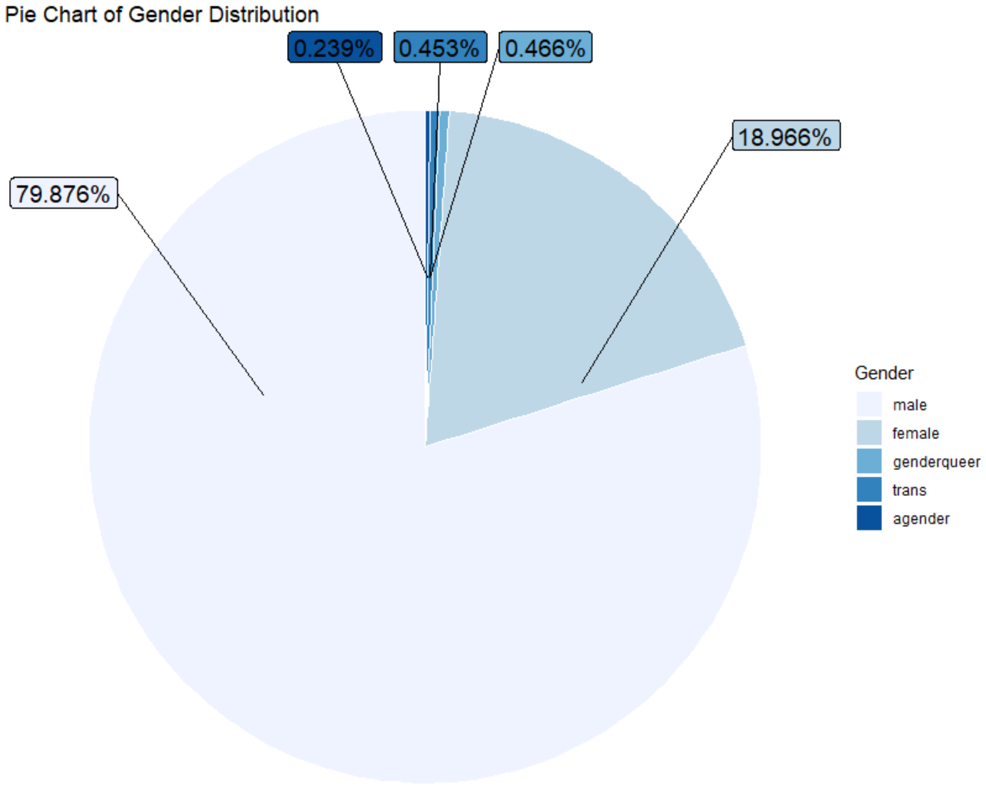


Figure 4: Pie Chart of Gender Distribution

A bar chart was also plotted for the age group by gender. Based on figure 5, we know that the number of genderqueers, trans and agender respondents are relatively smaller; therefore, for better visual representation of the bar chart, these 3 minor gender groups are grouped together as “others” in a new variable named Gendergrp. From figure 5, it is observed that each gender group has its peak around age 25, which means most new coders from each gender group are around age 25.

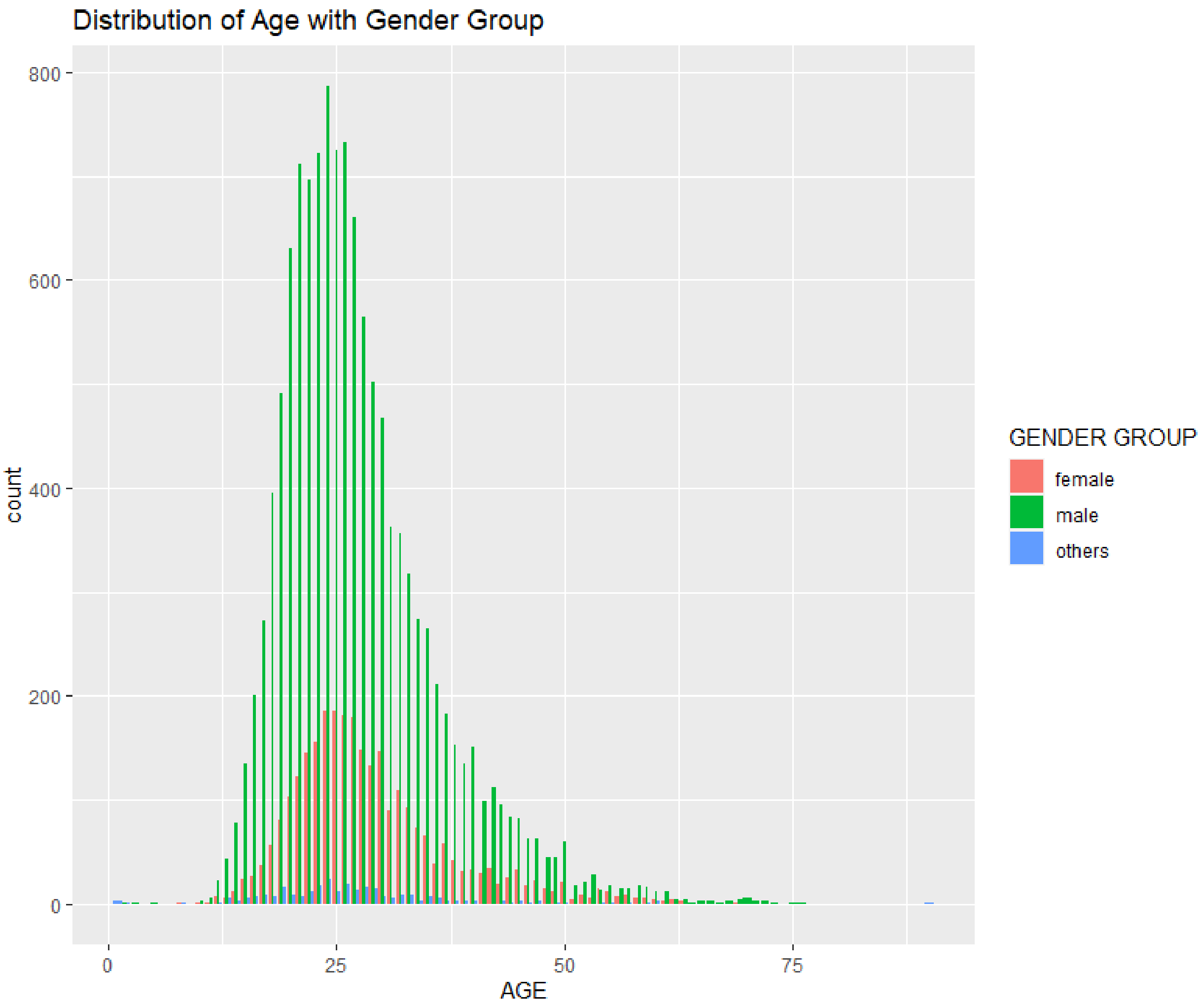


Figure 5: Distribution Plot of Age with Gender Group

We are also interested to identify if there is a pay difference between male, female and other gender groups. From the jitter plot in figure 6, we can observe that there are relatively more green dots (male) in the higher income range as compared to women and other gender groups. It can be assumed that male coders have a higher income as compared to women and other gender groups. This suggested that the pay gap between males and other genders exists. Not only are women grossly under-represented among developers, but they are grossly underpaid.

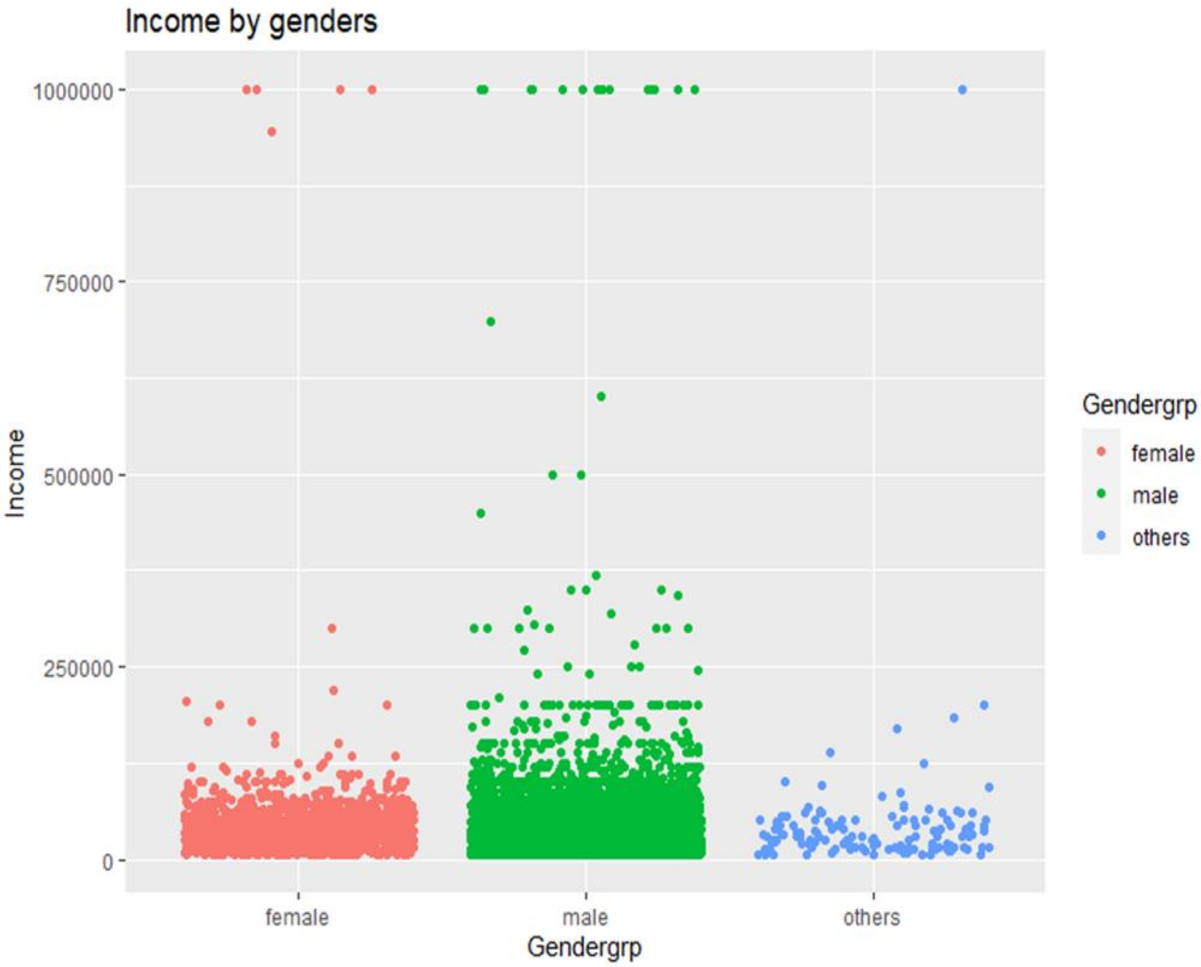


Figure 6: Jitter Plot of Income by Gender Group

After that, we looked at the income and age variables. We are expecting to see a pay increase with age since people who are older are usually more experienced, therefore they should be able to have higher income. To validate our assumption, a jitter plot of income by age group is created. Before plotting, the age variable was filtered to only include respondents that are above the age of 10. For better graphical presentation, age is divided into six groups, which are labeled as “11-20”, “21-30”, “31-40”, “41-50”, “51-60”, and “60 and above". As shown in figure 7, we can observe that age group “21-30”, “31-40” and “41-50” seemed to have a relatively higher income as compared to other age groups, however, to validate our assumption, a box plot of income by age group was created (see figure 8) The black dot in the box plot indicates the median income for each age group. To our surprise, there is a slight increase in the median income with the increase of age. This verifies our assumption: the older an individual is, the more likely he or she will get a higher pay.

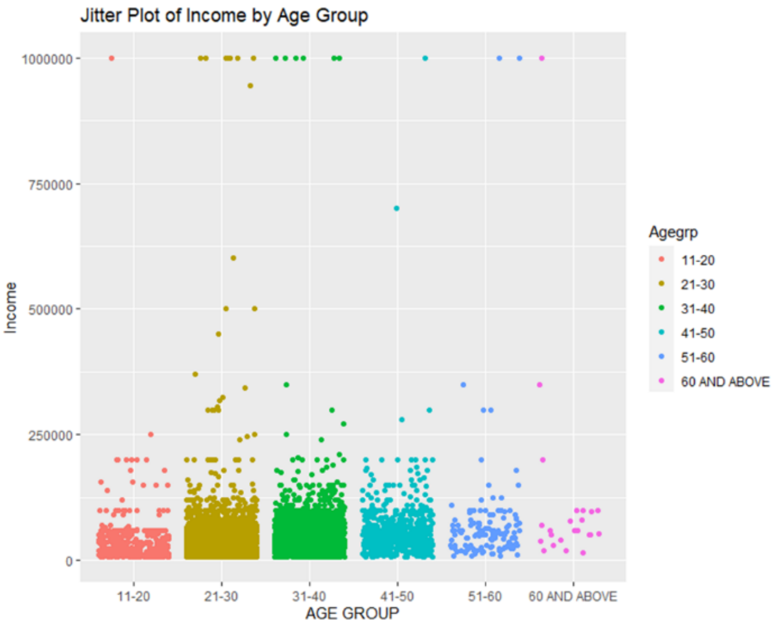


Figure 7 7: Jitter Plot of Income by Age Group

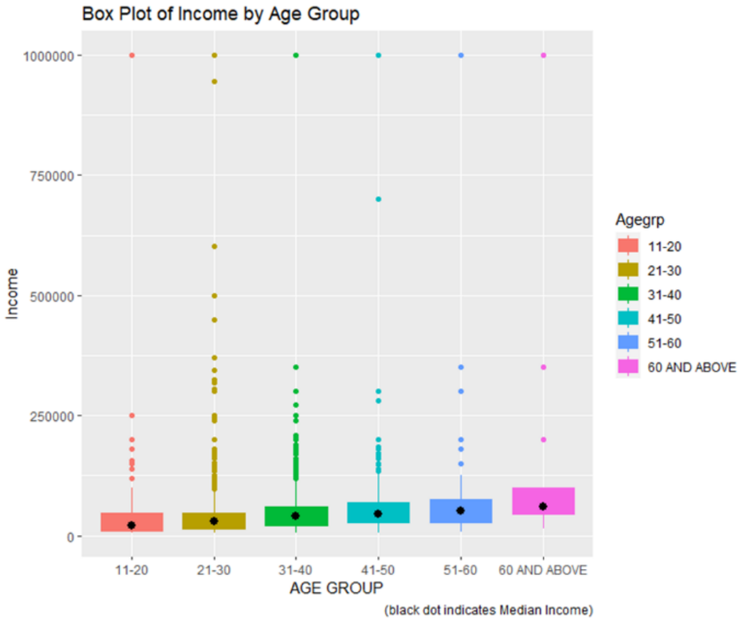


Figure 8: Box Plot of Income by Age Group (black dot indicates Median Income)

After exploring the demographics and income attributes of the respondents, we were interested to find out the level of association between the respondents' current job and their ideal job. Variables EmploymentStatus and JobPref are used to represent respondents' current job and ideal job respectively. As seen in figure 9, a heat map is used to show the relationship between these variables EmploymentStatus and JobPref. By observing the heat map, it is revealed that respondents who are currently employed for wages would like to work for a medium-sized company, this trend has a frequency count of 1227, which is the most significant relationship in this plot. Next, it is also observed that respondents who are currently employed for wages would also like to start their own business (frequency count: 954), be a freelancer (frequency count: 887), or work for a startup (frequency count: 714). On the other hand, respondents who are currently not working but looking for work also have a tendency to work for a medium-sized company, be a freelancer, start their own business or work for a start-up.

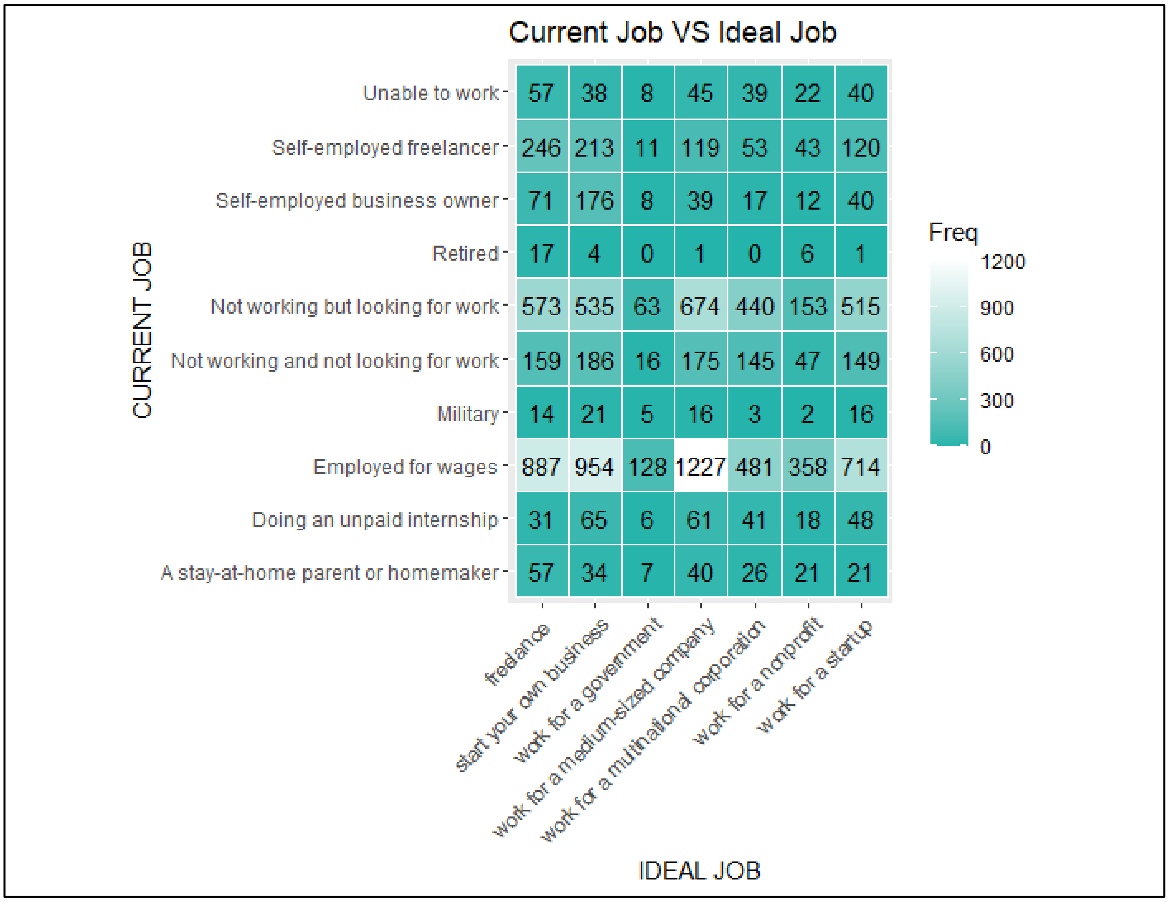


Figure 8: Heat Map of EmploymentStatus and JobPref

3.2 Predictive Analysis (Logistic Regression Analysis)

After doing some exploratory analysis on the dataset, we were interested in predicting whether or not a new coder is underemployed and whether or not there's a correlation between being under-employed and the other attributes of the respondent. We selected a total of 7 input variables of our interest. Here shows the description for the input and output variables of our regression model (refer to the appendix for detailed description):

Input Variables:

- **Age**: A numeric type answer regarding the age of the respondent
- **AttendedBootcamp**: A boolean type answer regarding whether or not the respondent has attended a full-time coding boot camp.
- **SchoolDegree**: A categorical type answer regarding the highest degree of level of school the respondent has completed.
- **Income**: A numeric type answer regarding the amount of money made by the respondent last year in US dollars.
- **Gender**: A categorical type answer regarding the respondent's gender type.
- **HoursLearning**: A numeric type answer regarding the number of hours spent learning each week.
- **MonthsProgramming**: A numeric type answer regarding the number of months the respondent has been programming.

Output Variables:

- **IsUnderEmployed**: A boolean type answer regarding whether or not the respondent considers themselves to be underemployed (to work a job that is below their education level)

Multiple logistic regression was used to predict the probability of whether or not a new coder is underemployed based on Age, AttendedBootcamp, SchoolDegree, Income, Gender, HoursLearning and MonthsProgramming. The dataset was imported and a subset that only contains the variables of our interest is created. Then, data of the subset was explored, filtered, mutated, cleaned, and converted to other data types before fitting them into a model. After that, the cor() function was used to create a correlation matrix and corrplot() was used to generate the correlation plot to see if there exists any strong correlation among the input variables. As seen in figure 10, no strong correlation is observed between the variables, the multicollinearity is low, so no variables need to be dropped for further analysis.

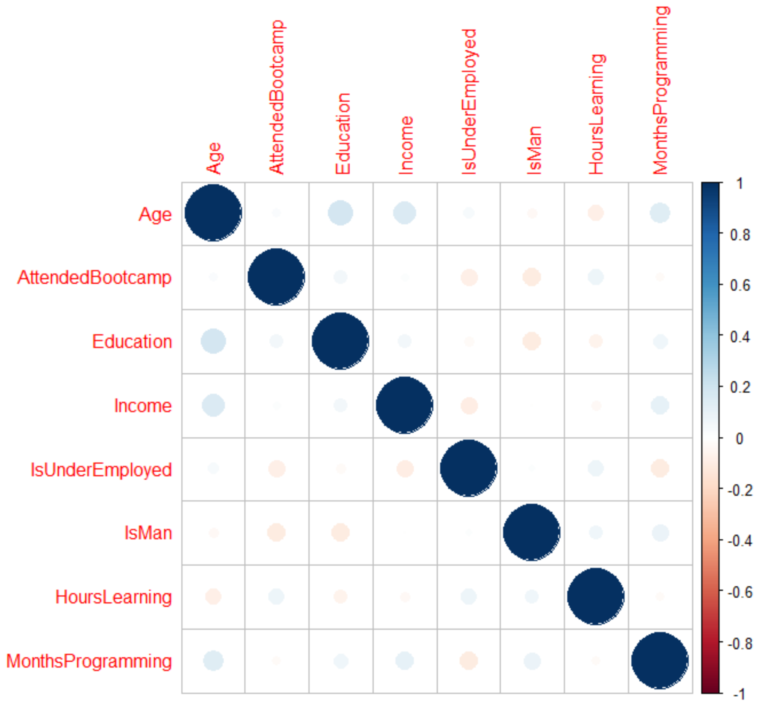
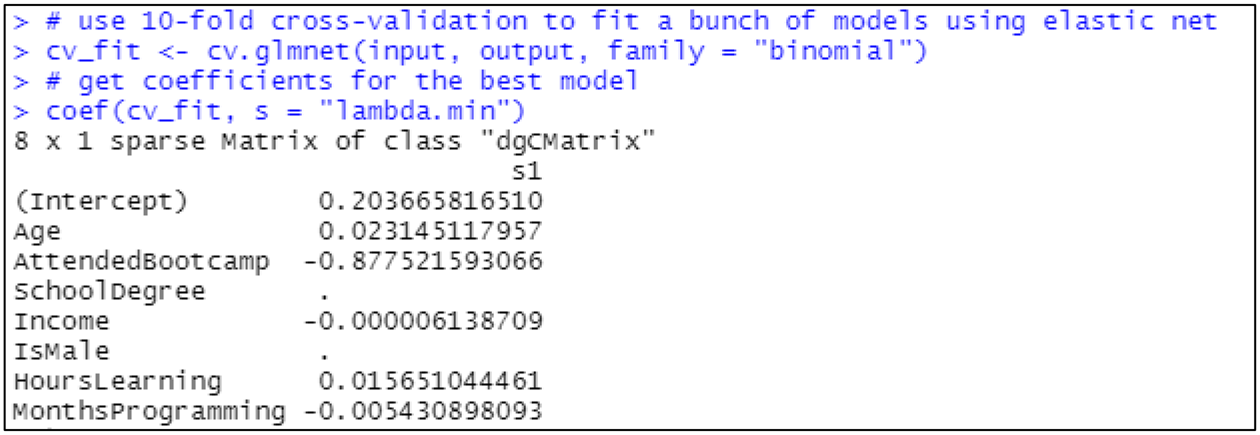


Figure 9: Correlation Plot

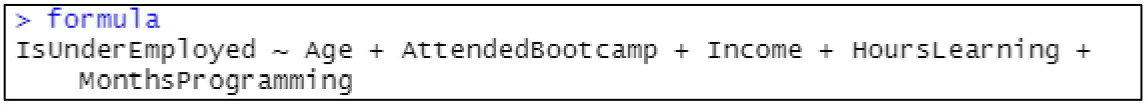
Before fitting our model, we split the dataset into a training set and a testing set, with a ratio of 70:30. Then, we used 10-fold cross-validation to fit our models with Elastic Net to select our variables so that our model can be regularized, and the probability of overfitting can be avoided by making sure that none of the coefficients end up being ridiculously large.



```
> # use 10-fold cross-validation to fit a bunch of models using elastic net
> cv_fit <- cv.glmnet(input, output, family = "binomial")
> # get coefficients for the best model
> coef(cv_fit, s = "lambda.min")
8 x 1 sparse Matrix of class "dgCMatrix"
                              s1
(Intercept)        0.203665816510
Age                0.023145117957
AttendedBootcamp  -0.877521593066
SchoolDegree       .
Income            -0.000006138709
IsMale             .
HoursLearning      0.015651044461
MonthsProgramming -0.005430898093
```

Figure 10: Output of 10-fold cross-validation to fit models

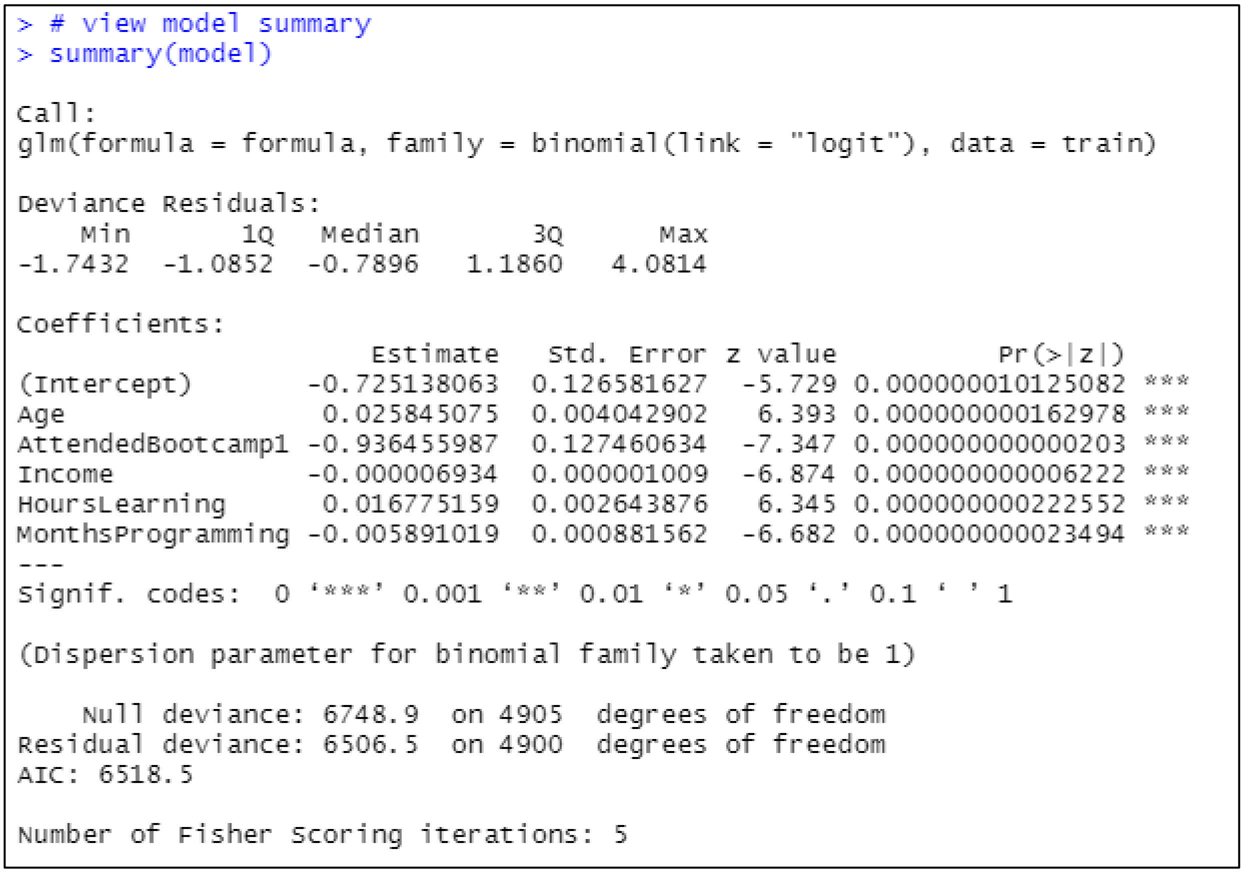
As seen in figure 11, all features except School Degree and IsMale were somehow useful in predicting whether or not a respondent is underemployed. Therefore, to create our best model, we only selected the variables with non-zero coefficients out of our Elastic Net model to fit a new regression model. Figure12, shows the formula of our new model.



```
> formula
IsUnderEmployed ~ Age + AttendedBootcamp + Income + HoursLearning +
    MonthsProgramming
```

Figure 11: Formula for regression model

To fit our logistic regression model with a general linear model, the formula in figure 12 is passed into the **glm()** function. We then viewed the summary of our model with the **summary()** method (see figure 13).

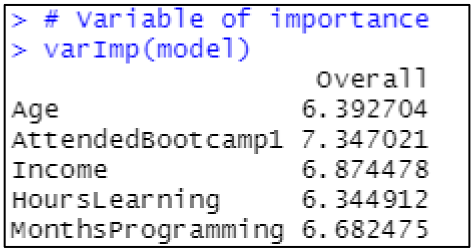


```
> # view model summary
> summary(model)

Call:
glm(formula = formula, family = binomial(link = "logit"), data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.7432  -1.0852  -0.7896   1.1860   4.0814  

Coefficients:
                      Estimate   Std. Error z value           Pr(>|z|)    
(Intercept)       -0.725138063  0.126581627  -5.729 0.000000010125082 ***
Age                0.025845075  0.004042902   6.393 0.00000000162978 ***
AttendedBootcamp1 -0.936455987  0.127460634  -7.347 0.00000000000203 ***
Income            -0.000006934  0.000001009  -6.874 0.000000000006222 ***
HoursLearning      0.016775159  0.002643876   6.345 0.00000000222552 ***
MonthsProgramming -0.005891019  0.000881562  -6.682 0.00000000023494 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6748.9  on 4905  degrees of freedom
Residual deviance: 6506.5  on 4900  degrees of freedom
AIC: 6518.5

Number of Fisher Scoring iterations: 5
```

Figure 12: Output of Model Summary

It is observed that the estimated effect of AttendedBootCamp on underemployed is -0.94, which has the highest estimated value for this model, however, it also has a relatively high standard error as compared to the other variables. Besides, it seems like all variables are fairly important in predicting the model as they all have very low p-values.

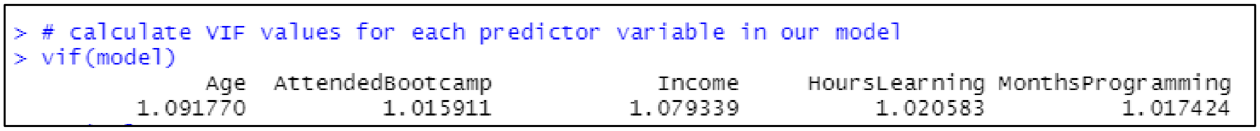
To access our model fit, we computed the importance of each predictor variable in the model by using the **varImp()** function from the *caret* package. Figure 14 shows that all of our variables have fairly equal importance. These results match up nicely with the p-values from the model (see figure 14).



```
> # Variable of importance
> varImp(model)
                     Overall
Age                 6.392704
AttendedBootcamp1 7.347021
Income              6.874478
HoursLearning       6.344912
MonthsProgramming 6.682475
```

Figure 13: Output of the variable of importance of the model

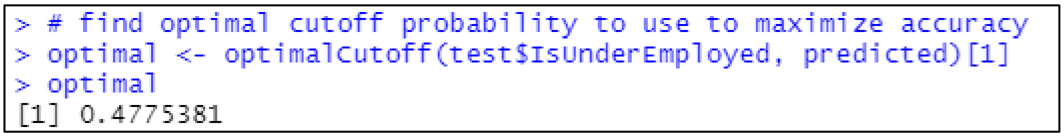
We also calculated the VIF values of each variable in the model to see if multicollinearity is a problem. As a rule of thumb, VIF values above 5 indicate severe multicollinearity. Since none of the predictor variables in our models have a VIF over 5, we can assume that multicollinearity is not an issue in our model (see figure15).



```
> # calculate VIF values for each predictor variable in our model
> vif(model)
              Age  AttendedBootcamp            Income     HoursLearning MonthsProgramming
         1.091770          1.015911          1.079339          1.020583          1.017424
```

Figure 14: Output of the VIF values for each predictor variable in the model

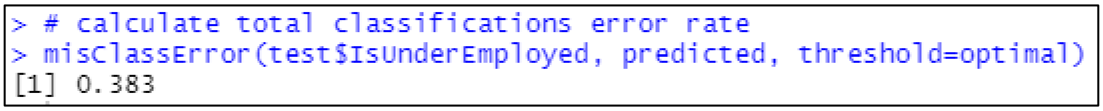
After fitting our logistic regression model, we used our model on the test dataset to make predictions on whether one is underemployed for every respondent in that dataset. Before analyzing how well our model performs on the test dataset, we used the **optimalCutoff**() function from the *InformationValue* package to find the optimal probability to use to maximize the accuracy of our model (see figure 16).



```
> # find optimal cutoff probability to use to maximize accuracy
> optimal <- optimalCutoff(test$IsUnderEmployed, predicted)[1]
> optimal
[1] 0.4775381
```

Figure 15: Output of the optimal cutoff probability

As shown in figure 16, the optimal probability cutoff to use is 0.4775381. Thus, any respondent with a probability of being underemployed of 0.4775381 or higher will be predicted to be underemployed, while any respondent with a probability less than this number will be predicted to not underemployed. Using this threshold, we calculate the misclassification rate of our model.



```
> # calculate total classifications error rate
> misClassError(test$IsUnderEmployed, predicted, threshold=optimal)
[1] 0.383
```

Figure 16: Output of Misclassification Error of the model

The total misclassification error rate is 38.3% for this model. In general, the lower the rate the better the model is able to predict outcomes, so our model turns out to be not very good at predicting whether a respondent is underemployed. Lastly, we plotted the ROC (Receiver Operating Characteristic) Curve which displays the percentage of true positives predicted by the model as the prediction probability cutoff is lowered from 1 to 0. The higher the AUC (area under the curve), the more accurately our model is able to predict outcomes. As shown in figure 18, we can see that the AUC is 0.639, which is relatively low. This indicates that our model is not very good at predicting whether or not a respondent is underemployed. We believe that this might be due to the fact that there are too many other variables affecting the probability of whether someone is underemployed, our selected input variables are not significant in predicting the outcome.

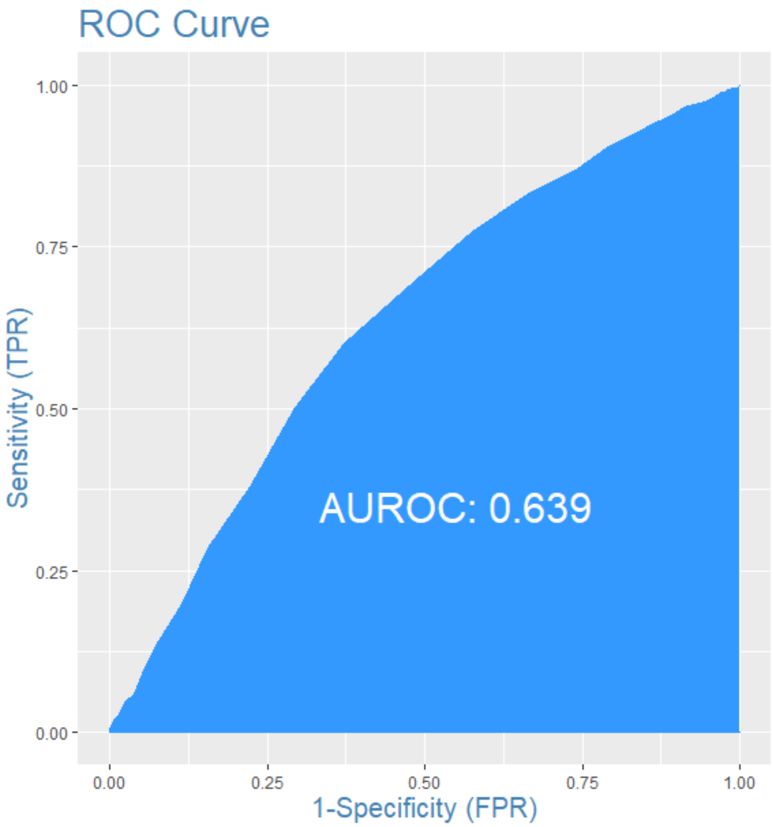


Figure 17: ROC Curve of the model

3.3 Predictive Analysis (Decision Tree Analysis)

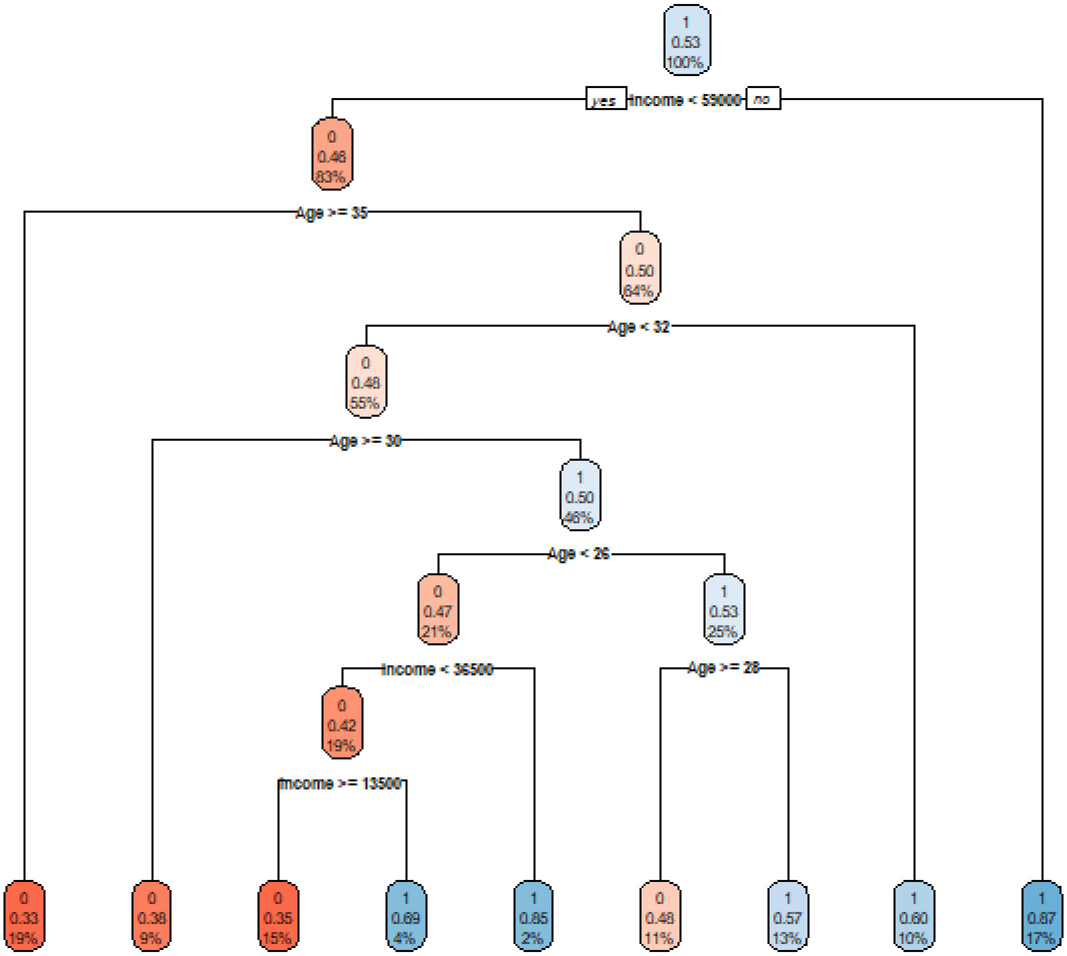


Figure 18: Decision Tree Map generated from the decision tree model

After predicting whether or not a respondent is underemployed, we were interested in predicting whether or not a respondent is a software developer and whether or not there's a correlation between being a software developer and the other attributes of the respondent. We selected a total of three variables of our interest to be used in creating a decision tree (refer to the appendix for a detailed description of each variable).

Age: a numeric type answer regarding the age of respondent

Income: a numeric type answer regarding the total money of the respondent earned last year (in US dollars)

IsSoftwareDev: a boolean type answer regarding whether a respondent is already working as a software developer

The decision tree is being used to break down the complex branches. We used the decision tree to see whether the respondent is working as a software developer and whether their age affects his or her income. In the figure 19 above, we can observe the nodes being split and leaves, representing the respondent already working as a software developer or not a software developer.

In the leaves, we can observe three numbers.

- The first numbers, 0 and 1, indicate whether the respondent already works as a software developer.

  Value = 1 (green): the respondent already works as a software developer

  Value = 0 (orange): the respondent is not working as a software developer

- The second number in the leaves shows the level of purity of nodes.

  Nodes in orange: the purity is represented in inverse
  Nodes in blue: the impurity is represented

  Darker blue/orange: more purity
  Light blue/orange: more impurity
  The darker the shade, the greater the level of purity

  In this case, purity states that it consists of information that respondents are not working as software developers.

- The percentage in the leaves shows the overall data that ends up with this classification

For examples,

- If the respondent's income is higher than $59000 and age is less than 35,
  - value of nodes = 0 indicates that the respondent is likely not a software developer
  - purity of nodes = 0.33, which means that 33% of data collected has some impurity
  - 19% shows the percentage of overall data that ends up with this classification

- If the respondent's income is higher than $59000 and age is between 35 and 33,
  - value of nodes = 1 indicates that the respondent is likely a software developer
  - purity of nodes = 0.6, which means that 60% of data collected has some impurity
  - 10% shows the percentage of overall data that ends up with this classification

4. Coding in R (R script)

This r script has three sections, which are the exploratory analysis, regression analysis, and decision tree analysis. The dataset is read from the Free Code Camp's Github repository, but if the link is not available at the moment, the dataset can also be imported from the CSV file that was attached in the submission zip file.

```
####################################################################################
#################### Web and Network Analytics Assignment #########################
####################################################################################


# Load dataset from github
coders_response <- read.table("https://raw.githubusercontent.com/freeCodeCamp/2017-new-coder-
survey/ed8a2c5118209fa26cc823fd33fedcf6fe5661ec/clean-data/2017-fCC-New-Coders-Survey-Data.csv",
                              header=TRUE, sep=",")


# importing the dataset (only if dataset couldn't be loaded from github)
# coders_response <- read.csv('2017-fCC-New-Coders-Survey-Data.csv',header = TRUE,
#                             na.strings = NA, stringsAsFactors = FALSE)


# install necessary packages
# install.packages("dplyr")
# install.packages("ggplot2")
# install.packages("tidyr")
# install.packages("gmodels")
# install.packages("scales")
# install.packages("ggrepel")
# install.packages("forcats")
# install.packages("tidyverse")
# install.packages("glmnet")
# install.packages("InformationValue")
# install.packages("caret")
# install.packages("car")
# install.packages("corrplot")
# install.packages("readr")
# install.packages("rpart")
# install.packages("rpart.plot")


# load necessary packages
library(dplyr)
library(ggplot2) # Data visualization
library(tidyr)
library(gmodels)
library(scales) # to calculate percentages
library(ggrepel) # to create labels for pie chart
library(forcats)
library(tidyverse)
library(glmnet) # fit glms with elastic net
library(InformationValue) # analyze how well our model performs on the test dataset
library(caret) # to compute importance of each predictor variable
library(car) # to calculate VIF values of each variable in the model
library(corrplot) # to generate correlation plot
library(readr)
library(rpart)
library(rpart.plot)
```

4.1 Exploratory Analysis

```
######################################################################################
############################ Exploratory Analysis ##################################
######################################################################################


# get an overview of the structure of the data
glimpse(coders_response)


# Get summary of data set
summary(coders_response)


# View the first 5 rows
head(coders_response)


# Get number of rows and columns
ncol(coders_response)
nrow(coders_response)


# Get column names
names(coders_response)


# Get class information
str(coders_response)


# Check distinct Age Values
unique(coders_response$Age)


max(coders_response$Age, na.rm = T)


########################### Data Preparation ########################################


coders_response$Agegrp[coders_response$Age > 0  & coders_response$Age <= 10] <- '1-10'
coders_response$Agegrp[coders_response$Age > 10 & coders_response$Age <= 20] <- '11-20'
coders_response$Agegrp[coders_response$Age > 20 & coders_response$Age <= 30] <- '21-30'
coders_response$Agegrp[coders_response$Age > 30 & coders_response$Age <= 40] <- '31-40'
coders_response$Agegrp[coders_response$Age > 40 & coders_response$Age <= 50] <- '41-50'
coders_response$Agegrp[coders_response$Age > 50 & coders_response$Age <= 60] <- '51-60'
coders_response$Agegrp[coders_response$Age > 60] <- '60 AND ABOVE'


# check distinct Agegrp Values
unique(coders_response$Agegrp)


# check distinct Gender Values
unique(coders_response$Gender)


# check frequency of each gender group
table(coders_response$Gender)


# Merge agender, trans and genderqueer into one category to main data frame
coders_response$Gendergrp[coders_response$Gender == 'agender'] <- 'others'
coders_response$Gendergrp[coders_response$Gender == 'trans'] <- 'others'
coders_response$Gendergrp[coders_response$Gender == 'genderqueer'] <- 'others'
coders_response$Gendergrp[is.na(coders_response$Gender)] <- 'others'
coders_response$Gendergrp[coders_response$Gender == 'female'] <- 'female'
coders_response$Gendergrp[coders_response$Gender == 'male'] <- 'male'


# check distinct Gender Values
unique(coders_response$Gendergrp)
```

```
# check frequency of each gender group
table(coders_response$Gendergrp)


# count frequency for each age group by gender group
age_gender_freq = coders_response %>%
  # select(Age, Gender)%>%
  group_by(Agegrp, Gendergrp)%>%
  summarize(count=n())


###########################  Pie chart of Gender ###############################


table(Gender = coders_response$Gender) %>%
  as.data.frame() %>%
  arrange(desc(Freq)) %>%
  mutate(prop = percent(Freq / sum(Freq))) %>%
  ggplot(aes( x= '', y=Freq,  fill = fct_inorder(Gender))) +
  geom_bar(stat="identity", width=1, color="white") +
  coord_polar("y", start=0) +
  theme_void() +
  scale_fill_brewer("Blues") +
  geom_label_repel(aes(label = prop), size=5,show.legend = F, nudge_x = 2) +
  guides(fill = guide_legend(title = "Gender")) +
  ggtitle('Pie Chart of Gender Distribution')


################ Bar Chart of Age Distribution by Gender Group ##################


coders_response %>%
  select(Age, Gendergrp)%>%
  filter(Gendergrp %in% c('female', 'male', 'others')) %>%
  filter(Age > 0) %>%
  ggplot(aes(x=Age)) +
  geom_bar(aes(fill=Gendergrp), position = position_dodge()) +
  ggtitle('Distribution of Age with Gender Group') +
  labs(x ="AGE", Y="COUNT") +
  scale_fill_discrete(name="GENDER GROUP")


############### Geom jitter plot of income by gender groups ###################


coders_response %>%
  select(Gendergrp, Income) %>%
  ggplot(aes(Gendergrp,Income, color = Gendergrp, fill=Gendergrp)) +
  geom_jitter() +
  ggtitle("Income by genders")



################# Jitter plot of Income by AGe group ###########################


coders_response %>%
  select(Agegrp, Income)%>%
  filter(!is.na(Income),!is.na(Agegrp))%>%
  ggplot(aes(Agegrp,Income, color = Agegrp, fill=Agegrp)) +
  geom_jitter() +
  labs(title = "Jitter Plot of Income by Age Group",
       x ="AGE GROUP", Y="INCOME")


#################### Box Plot of Income by Age Group ###########################


coders_response %>%
  select(Agegrp, Income)%>%
  filter(!is.na(Income),!is.na(Agegrp))%>%
  ggplot(aes(Agegrp,Income, color = Agegrp, fill=Agegrp)) +
  geom_boxplot() +
  stat_summary(fun=median, geom="point", shape=20, size=4,
              color="black", fill="black") +
  labs(title = "Box Plot of Income by Age Group",
       caption = "(black dot indicates Median Income)",
       x ="AGE GROUP", Y="INCOME")


################## Heat map of Current job vs ideal job ########################


table(current_job = coders_response$EmploymentStatus,
      ideal_job = coders_response$JobPref) %>%
  as.data.frame() %>%
  ggplot(aes(ideal_job,current_job))+
  geom_tile(aes(fill = Freq), colour = "white")+
  coord_fixed(ratio = 1)+
  theme(axis.text.x=element_text(angle=45,vjust = 1, hjust = 1)) +
  labs(title = "Current Job VS Ideal Job",
       x = "IDEAL JOB", y = "CURRENT JOB") +
  geom_text(aes(label=Freq)) +
  scale_fill_gradient(low = "#24b4ab", high = "white")
```

4.2 Regression Analysis

```
################################################################################
####################### Logistic Regression Analysis ###########################
################################################################################


# Find the maximum learning hours per week
max(coders_response$HoursLearning, na.rm = T)


# create a subset of the data with only our variables of interest
dataSubset <- coders_response %>%
  filter(Age > 10) %>% # filter data that doesn't make sense
  filter(HoursLearning < 105) %>%
  mutate(IsMale = as.integer(Gender == "male")) %>%
  mutate(SchoolDegree = as.numeric(factor(SchoolDegree,
                                          levels=c("no high school (secondary school)",
                                                   "some high school",
                                                   "high school diploma or equivalent (GED)",
                                                   "trade, technical, or vocational training",
                                                   "some college credit, no degree",
                                                   "associate's degree",
                                                   "bachelor's degree",
                                                   "master's degree (non-professional)",
                                                   "professional degree (MBA, MD, JD, etc.)",
                                                   "Ph.D.")))) %>%
  select(Age, AttendedBootcamp, SchoolDegree,
         Income,IsUnderEmployed, IsMale,
         HoursLearning, MonthsProgramming ) %>%
  na.omit() # remove non-numeric values


# Understand the structure of the subset
glimpse(dataSubset)
summary(dataSubset)


# Compute correlation matrix
correlations <- cor(dataSubset[,1:8])


# Compute correlation plot
par(mfrow = c(1,1))
corrplot(correlations, method="circle")


# covert all categorical variables to factors
```

```
dataSubset$AttendedBootcamp <- as.factor(dataSubset$AttendedBootcamp)
dataSubset$SchoolDegree <- as.factor(dataSubset$SchoolDegree)
dataSubset$IsMale <- as.factor(dataSubset$IsMale)


# Take another look at the subset
glimpse(dataSubset)
summary(dataSubset)


# Understand the data set by checking the freqeuncy count, mean and proportion
table(dataSubset$AttendedBootcamp)
table(dataSubset$IsUnderEmployed)
mean(dataSubset$IsUnderEmployed)


# Check on the proportion
prop.table(table(dataSubset$IsUnderEmployed))


# find the total rows in dataset
nrow(dataSubset)


#################### Create Training and Test Samples #######################


#  initialize a pseudorandom number generator to make this example reproducible
set.seed(1)


# Use 70% of data set as training set and remaining 30% as testing set
sample <- sample(c(TRUE, FALSE), nrow(dataSubset), replace=TRUE, prob=c(0.7,0.3))
train <- dataSubset[sample, ]
test <- dataSubset[!sample, ]


########################### Fitting a model #################################


# convert input variables to matrix
input <- train %>%
  select(-IsUnderEmployed) %>% # Exclude variable to be predicted
  data.matrix()  %>%
  na.omit()


# get a vector with our output variable
output <- train$IsUnderEmployed


# use 10-fold cross-validation to fit a bunch of models using elastic net
cv_fit <- cv.glmnet(input, output, family = "binomial")


# get coefficients for the best model
coef(cv_fit, s = "lambda.min")


# get a (non-sparse) matrix of the coefficients for the best model
coef_matrix <- coef(cv_fit, s = "lambda.min") %>%
  as.matrix()
coef_matrix


# get the variables with a coefficient that's not 0
variables <- row.names(coef_matrix)[coef_matrix != 0] %>%
             setdiff("(Intercept)") #remove the intercept
# check on the variables that only has our selected features
variables


# turn list of formulas into a variable
variables_selected <- paste(variables, collapse="+")
variables_selected
```

```
formula <- paste("IsUnderEmployed ~ ",variables_selected,sep = "") %>%
  as.formula()
formula


# fit logistic regression model with general linear model (glm)
model <- glm(formula, data = train,
            family = binomial(link="logit"))


# view model summary
summary(model)


# Check the confidence level for this model
confint(model)


# added-variable (partial regression) plots for model
avPlots(model)



######################### Assessing Model Fit ##############################


# Variable of importance
varImp(model)


# calculate VIF values for each predictor variable in our model
vif(model)


# # to analyze the table of deviance
# anova(model, test="Chisq")


###################### Use the model to make predictions #####################


# calculate probability of default for each individual in test data set
predicted <- predict(model, test, type="response")


############################ Model Diagnostics ############################


# find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$IsUnderEmployed, predicted)[1]
optimal


# calculate total classifications error rate
misClassError(test$IsUnderEmployed, predicted, threshold=optimal)


# plot the ROC curve
plotROC(test$IsUnderEmployed, predicted)
```

4.3 Decision Tree Analysis

```
################################################################################
########################## Decision Tree Analysis ##############################
################################################################################


#remove all NA observations
na.coders_response <- coders_response[complete.cases(coders_response[ , 5:6]),]


#build models
tree <- rpart(IsSoftwareDev~ Age + Income, method = "class", na.coders_response)


#plot & visualize decision tree, remove scientific notation
rpart.plot(tree, box.palette="RdBu",digits = -2)
```

5. Lessons Learned and Conclusion

In conclusion, from the analysis of the given dataset, we were able to detect trends and patterns as well as get interesting insights. For instance, we were able to observe that the majority of the respondents are male, are around the age of 25, and have relatively higher incomes. Furthermore, our analysis also reveals that there is a linear relationship between age and income: the higher the age, the higher the income, this suggests that employers do value aged and experienced employees. We also found some interesting trends between respondents' current job and their ideal job. The most significant trend is that respondents who are currently employed for wages would like to work for a medium-sized company. Moreover, we also did predictive analysis regarding the probability that a respondent is underemployed as well as the probability that a respondent is a software developer. These analyses can provide useful information in identifying challenges of the society such as gender inequality in the programming field. Though much was discovered during our analysis, we also faced problems as well as made mistakes in our analysis. For instance, we failed to create a predictive model to predict the probability of a respondent being underemployed due to several possible reasons. One reason could be that our input variables for the model are not significant to predict the outcome, this leads to the creation of a model that does not have high predictive ability. This has taught us the importance of feature selection. Hence, we will be more careful in terms of choosing the right features for our model in the future.

6. Individual Reflection

Alicia Chong Tsui Ying

Personally, being able to have first-hand experience of analysing such a huge data set has allowed me to gain a lot of R programming and data analysis knowledge throughout the course of the assignment. Furthermore, this was also my first time performing predictive analysis with programming tools; therefore, it took me a considerably large amount of time in trying to learn all that I can regarding this topic from the Internet. The process was challenging, but the outcome was also rewarding. Although there still consists of many limitations and imperfections in our assignment, I think we have tried our best to perfect this assignment. This assignment has certainly played a role in opening my eyes to the potential of data analysis and its future implications to the society.

Wong Yi Qing

From this assignment, I realised that analysis is a very powerful tool in our everyday lives. This is because it can accurately analyse large-scale data and aid us in making decisions that are beyond human capabilities. Furthermore, being able to obtain and grasp this knowledge is beneficial for my future, as a data analytics student. During the process of our assignment, we were faced with certain challenging parts such as getting the results that we did not expect, and we had to find out what was wrong and modify the codes. Lastly, this assignment gave me experience of what it feels like to code as a team, on how to communicate with my teammates efficiently. I am thankful to my teammates, as our teamwork is the catalyst to the completion of this assignment.

Kusselin Yuthasax A/P Prak Chuap

In this assignment, at first, I find it very challenging at first but as a data analytics student I believe that I have gained something from this subject in terms of analytics skills. At first, we have to change our chosen data set many times as the earlier dataset could not give us much information that we can analyze or some of them was even analyzed so that there was nothing we could do. In the end we used New Coders Survey Data, but problems occurred as we tried to decide on which area of analysis to use for this dataset, which is considered to be too large for us. One part that I found difficult for me is the logistic regression part as the process itself is complex and we have to explore, filter, mutate, clean and even convert the data types before fitting them into a model. We had to come up with different formulas for this part and the key learning point is I learned how to use varImp() functions from the caret package to compute the importance of each predictor variable in the models. Last but not least, without my teammates I believe that I will not be able to come this far as I still think this subject is very challenging, with their efforts and guidance we manage to finish this project just on time.

Soh Wee Chee

Throughout this assignment, I gained a greater insight into the importance of analysis. It creates valuable information for real-world applications by organizing, interpreting, and structuring the data. Thus, I understand the need for analysis in research and surveys that makes data analysis more straightforward and accurate. Our group faces some coding difficulties as the outcome is not as expected. We had to change and modify our codes a few times to get a better result. I realized I would have to study more and be more consistent in order to solve the coding problem in the future. Finally, I want to thank my group mates for their support in completing this project.

Yap Yee Mun

Analysis brings hidden insights and patterns of the dataset to make better data-driven decisions. Throughout the exploratory and analysis process, I realised that there are many factors that shaped the future of developers. From the exploratory analysis, it is found that companies value age and experience as programmer/developer. This piece of information shows the importance of years of experience in coding. Thus, I’ve learned to be consistent in learning more new technologies and programming languages to build up my coding skills and participate in more projects to gain different experience and challenges. Lastly, I would like to thank my great groupmates for exploring this dataset and bringing out different insights together.

References

Larson, Q. (2016, May 3). *We asked 15,000 people who they are, and how they’re learning to code*. FreeCodeCamp.Org. Retrieved December 3, 2021, from https://www.freecodecamp.org/news/we-asked-15-000-people-who-they-are-and-how-theyre-learning-to-code-4104e29b2781/#.pdpnnbaw7

Appendix

Metadata of Variables used for analysis

Variables	Question	Type	Options / Note
Age	How old are you?	Numeric	User Input
AttendedBootcamp	Have you attended a full-time coding bootcamp?	Boolean	Options: Yes (1), No (0)
Gender	What’s your gender?	Categorical	• female • male • agender • trans • genderqueer
SchoolDegree	What's the highest degree of level of school you have completed?	Categorical	• no high school (secondary school) • some high school • high school diploma or equivalent (GED) • some college credit, no degree • trade, technical, or vocational training • associate's degree • bachelor's degree • master's degree (non-professional) • professional degree (MBA, MD, JD, etc.) • Ph.D.
EmploymentStatus	Regarding employment status, are you currently...	Categorical	• Employed for wages • Self-employed freelancer • Self-employed business owner • Doing an unpaid internship • Not working but looking for work • Not working and not looking for work • A stay-at-home parent or homemaker • Military • Retired • Unable to work • Other Note: The "Other" option, if selected, creates a new column for user input.
Income	About how much money did you make last year in (US dollars)?	Numeric	Options: User input
IsUnderEmployed	Do you consider	Boolean	Options: Yes (1), No (0)

	yourself under-employed?		Note: The explanation for this question was, "Under-employed means working a job that is below your education level"
IsSoftwareDev	Are you already working as a software developer?	Boolean	Options: Yes (1), No (0)
JobPref	Would you prefer to...	Categorical	Options: • work for a startup • start your own business • work for a multinational corporation • freelance • work for a medium-sized company
JobRoleInterest	Which one of these roles are you most interested in?	Categorical	Options: • Front-End Web Developer • Mobile Developer • Data Scientist/Data Engineer • User Experience Designer • Back-End Web Developer • Full-Stack Web Developer • Quality Assurance Engineer • Product Manager • DevOps/SysAdmin • Other Note: The "Other" option gave the individual the freedom to type in whatever they wish. This field is the column directly after this question's column. Thus, this "Other" column is mostly empty.
HoursLearning	About how many hours do you spend learning each week?	Numeric	Options: User input
MonthsProgramming	About how many months have you been programming for?	Numeric	Options: User input

*Note: Refer to the link below for the full dataset metadata for the 2016 New Coder Survey. (some descriptions may vary for the 2017 New Coder Survey dataset) https://github.com/freeCodeCamp/2016-new-coder-survey/blob/master/clean-data/survey-data-dictionary.md