**Predictive Gender Model**

Donghyeon KIM(814762)

1. Abstract

The data analysis process is important to make a suitable model and new insight. The data analysis has four types and the data analytics process has six steps. Before analysing data and making model, the given data set is necessary to clean by filtering unnecessary data and removing incorrect, outliers, missing and duplicate values as well as normalisation. Then, exploratory data analysis is one of the critical parts of the process by identifying descriptive statistics such as the measure of tendency, frequency, variation and position, and suitable plots and graphs. According to each phase performed, the model is developed to predict by using the numerous voice data in order to differentiate by gender. The candidate models are created through diverse methods such as linear regression and artificial neural networks. The candidate models are continually developed and compared each other, then the most satisfied model is selected. After that, the selected model is evaluated. When this performance evaluation meets the initial purpose and follows significant results, the next phase is moved as communicating results and to operationalising such as briefing, final reports and technical documentation including code. If the performance evaluation is dissatisfied, the past phase is iterated. This report performs in accordance with the data analytics process then obtains the satisfied model by using supervised and unsupervised learning in the R application.

1. Table of Contents and Figures

Figure 1: descriptive statistics table 1…..…………………………………………………………………………4

Figure 2: descriptive statistics for male table 2……………………………………………………………………5

Figure 3: descriptive statistics for female table 3………………………………………………………………….6

Figure 4: descriptive statistics boxplot 1.………………………………………………………………………….7

Figure 5: PCA table 1…..………………………………………………………………………………………….8

Figure 6: PCA plot 1……………………………………………………………………………………………….8

Figure 7: PCA ggbiplot …….……………………………………………………………………………………...9

Figure 8: PCA biplot …………….………………………………………………………………………………...9

Figure 9: PCA plot 2…………….………………………………………………………………………………..10

Figure 10: PCA plot 3…………………………………………………………………………………………….10

Figure 11: correlation plot 1……………………………………………………………………………………...11

Figure 12: correlation plot 2……………………………………………………………………………………...12

Figure 13: the descriptive statistics about meanfun quartile 1…..……………………………………………….12

Figure 14: the descriptive statistics about IQR quartile 2…………………..……………………………………12

Figure 15: ANNs plot 1…………………………………………………………………………………………..13

Figure 16: ANNs plot 2…………………………………………………………………………………………..13

Figure 17: ANNs plot 3………..…………………………………………………………………………………14

Figure 18: ANNs plot 4…………..……………………………………………………………………………....14

Figure 19: confusion matrix table 1……………..……………………………………………………………….15

Figure 20: confusion matrix table 2……………………………………………………………………………...16

Figure 21: confusion matrix table 3……………………………………………………………………………...16

Figure 22: confusion matrix table 4……………………………………………………………………………...16

Figure 23: confusion matrix table 5……………………………………………………………………………...17

Figure 24: confusion matrix table 6……………………………………………………………………………...17

Figure 25: confusion matrix table 7……………………………………………………………………………...18

Figure 26: confusion matrix table 8……………………………………………………………………………...18

Figure 27: confusion matrix table 9……………………………………………………………………………...18

Figure 28: confusion matrix table 10…………………………………………………………………………….18

Figure 29: distance matrix table 1………………………………………………………………………………..18

Figure 30: distance matrix table 2………………………………………………………………………………..19

Figure 31: distance matrix table 3………………………………………………………………………………..19

Figure 32: candidate model table 1……………………………………………………………………………....20

Figure 33: bar plot 1…….………………………………………………………………………………………..20

1. Introduction

Many methods are existed to differentiate between gender throughout the real world. One of the methods is the voice that can classify gender. The classification of gender through voice is easy for humans, but difficult for computers so after having past data analyse, as making model and learning by machine learning, the computer can differentiate between gender by the feature of voice. However, the human can be also hard to classify by gender when a lot of the variables and quantifying data are given to analyse and to make model. It is related to the problem which is to the diverse a lot of variables in the given data set to make the best model classifying by gender. Hence, the solution is that a suitable model is selected by using effective methods, tools and techniques according to the data analysis process. Hence that the final satisfied model has more accurate prediction and classification of gender despite of numerous data.

The types of data analysis consist of descriptive analytics, decision analytics, predictive analytics and finally prescriptive analytics. In addition, the components of the data analytics process are discovery, data preparation, model planning, model building, communicate results and final phase is operationalization. As the process of data analysis is departmentalized, the first step is pre-processing data for data analytics. When this step proceeds, the optimised can be created to analyse by data cleansing. The removed six features of dirty data such as incomplete, duplicate, incorrect, inaccurate, business rule violations and inconsistent data, are implemented to make model through techniques.

The compositions of procedure of technical implementation are data clearing methods, principal component analysis (PCA), correlation, linear regression modelling, classifying between testing, validation and testing, artificial neural networks (ANN), k-nearest neighbour (k-NN) and hierarchical clustering. The making model is used by supervised and unsupervised learning. For instances of using learning in this report, supervised learnings such as a linear regression model, ANNs and k-NN are performed. Otherwise, unsupervised like hierarchical clustering is executed. After the results are revealed by using that, candidate models are evaluated and compared by using statistical metrics, tables and plots. The final model is identified as close as the prediction and the accurate.

1. Technical Implementation

The first phase is to understand the problem and the purpose to create a model so it is necessary to recognise the desired results. The framing this problem is the significant model to classify the gender according to the procedure of data analytics. Before pre-processing the data set, the types of variable and data, as well as the definitions of the independent and dependent variable are identified such as data, numeric, character.

The second step is to prepare data in order to optimise the environment of analysis by cleansing data. The data is gender voice to determine based on quantifying voice. This data is secondary data so it is already collected and recorded. Furthermore, the data set do not have missing values and inconsistency data. However, the data is necessary to clean and identify before analysis and creation of the model. It helps to improve the accurate model and results. Additionally, cleansing data removes and decreases anomaly and skew. The data set is cleaned by using Excel and R. If Excel is available to clean data such as spell check, removing duplicate rows, change the case of text, removing spaces, fixing numbers, merge and split as well as transforming and rearranging columns.

However, by using R, it is easy to clean data than using Excel so the data set is imported in R. Besides, the results will be improved by removing outliers. Therefore, firstly as seen from the R script, a column is added to remove outliers easily. This additive column represents by NA or one so NA is an outlier but one is not. After this process, by using na.omit function, the outliers and other missing values are removed. Additionally, by using a unique function, the duplicate data are erased. Before over the next step, the consistency of data types is assessed so one of the columns as the label is necessary to be changed string to numeric in order to analyse to be useful and make model. The results of change are male’s rows represent by one and female represent by two. Therefore, the 3168 observations in 21 variables change the 2230 observations in 21 variables by cleaning the data set.

After removing outliers and cleaning data set, the values in the data set can be able to standardise or normalize because they help to easily understand quantity data. The standardisation identifies how far apart the data value is from the mean by rescaling data to have a mean of 0 and a standard deviation of 1. Otherwise, the normalization is the process of matching data ranges or making distributions similar when comparing and dealing with large amounts of data so the data set in this report need to normalise. In order to normalise the data set, the function is created and the formula is necessary so the formula is that the difference value between the value and the minimum value in the variable is divided by the difference value between the maximum and minimum value in the variable.

The next phase is model planning. This phase is related to data analysis and dimensionality reduction. Firstly, for the below Figure 1, the table is indicated the factors of descriptive statistics. The factors are the number of valid observations (nbr.val), the number of null (nbr.null), the number of not available (nbr.na), the minimum value (min), the maximum value (max), the range, the sum, the median, the standard error of the mean (SE.mean), the 95% confidence interval of the mean (CI.mean.0.95), the variance(var), the standard deviation (std.dev) and the coefficient of variation (coef.var). These factors are important to understand each variable and to compare between variables. This table is able to identify the overall tendency. The values are rounded to the three-decimal place. The median values and mean values can be similar because outliers are already removed and normalisation is applied. After normalization, the maximum range is 1. The range value of kurt has the maximum value of all of the variables. The mindom’s coefficient of variation has the highest value. It means that the mindom is the greatest level of dispersion around the mean.

However, this model is to differentiate by gender so the descriptive statistics table is created by gender to identify how to be the difference in each variable. As seen from Figure 2 and 3, Figure 2 is for male and 3 is for female. After cleaning the data set, the number of observations for the male is 1138 and for the female is 1092. For the measure of tendency such as mean, median, the female’s value can be higher than the male. Moreover, for the measure of variation such as range, variance and standard deviation, the female tends to be higher.

(Figure 1: descriptive statistics table 1)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | meanfreq | Sd | median | Q25 | Q75 | IQR | Skew | kurt | sp.ent |
| nbr.val | 2230 | 2230 | 2230 | 2230 | 2230 | 2230 | 2230 | 2230 | 2230 |
| nbr.null | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nbr.na | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| min | 0.004 | 0.001 | 0.004 | 0.001 | 0.006 | 0.001 | 0.011 | 0.090 | 0.030 |
| max | 0.01 | 0.004 | 0.01 | 0.01 | 0.011 | 0.008 | 0.169 | 1.000 | 0.038 |
| Range | 0.005 | 0.003 | 0.006 | 0.009 | 0.004 | 0.007 | 0.158 | 0.910 | 0.008 |
| Sum | 16.212 | 4.848 | 16.725 | 12.726 | 19.963 | 7.237 | 182.612 | 752.584 | 78.147 |
| median | 0.007 | 0.002 | 0.008 | 0.006 | 0.009 | 0.004 | 0.079 | 0.291 | 0.035 |
| Mean | 0.007 | 0.002 | 0.008 | 0.006 | 0.009 | 0.003 | 0.082 | 0.337 | 0.035 |
| SE.mean | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.004 | 0.000 |
| CI.mean.0.95 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.007 | 0.000 |
| Var | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.031 | 0.000 |
| std.dev | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.002 | 0.027 | 0.176 | 0.002 |
| coef.var | 0.135 | 0.275 | 0.155 | 0.290 | 0.088 | 0.469 | 0.334 | 0.522 | 0.047 |
|  | | | | | | | | | |
|  | Sfm | Mode | Centroid | meanfun | minfun | maxfun | Meandom | Mindom | max  dom |
| nbr.val | 2230 | 2230 | 2230 | 2230 | 2230 | 2230 | 2230 | 2230 | 2230 |
| nbr.null | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nbr.na | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Min | 0.003 | 0.000 | 0.004 | 0.003 | 0.000 | 0.009 | 0.002 | 0.000 | 0.005 |
| Max | 0.032 | 0.011 | 0.010 | 0.009 | 0.004 | 0.011 | 0.089 | 0.006 | 0.560 |
| Range | 0.029 | 0.011 | 0.005 | 0.006 | 0.003 | 0.002 | 0.087 | 0.006 | 0.554 |
| Sum | 34.478 | 15.889 | 16.212 | 12.615 | 3.340 | 23.372 | 78.871 | 3.197 | 483.57 |
| median | 0.014 | 0.008 | 0.007 | 0.005 | 0.002 | 0.011 | 0.034 | 0.001 | 0.215 |
| Mean | 0.015 | 0.007 | 0.007 | 0.006 | 0.001 | 0.010 | 0.035 | 0.001 | 0.217 |
| SE.mean | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.002 |
| CI.mean.0.95 | 0,000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.005 |
| Var | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.014 |
| std.dev | 0.007 | 0.003 | 0.001 | 0,001 | 0.001 | 0.001 | 0.018 | 0.002 | 0.118 |
| coef.var | 0.438 | 0.354 | 0.135 | 0.202 | 0.393 | 0.054 | 0.523 | 1.124 | 0.542 |
|  | | | | | | | | | |
|  | Dfrange | | Modindx | | label |  | | | |
| nbr.val | 2230 | | 2230 | | 2230 |
| nbr.null | 0 | | 0 | | 0 |
| nbr.na | 0 | | 0 | | 0 |
| Min | 0.003 | | 0.00 | | 1.000 |
| Max | 0.559 | | 0.036 | | 2.000 |
| Range | 0.556 | | 0.036 | | 1.000 |
| Sum | 480.369 | | 13.884 | | 3322 |
| median | 0.214 | | 0.005 | | 1.000 |
| Mean | 0.215 | | 0.006 | | 1.490 |
| SE.mean | 0.002 | | 0.000 | | 0.011 |
| CI.mean.0.95 | 0.005 | | 0.000 | | 0.021 |
| Var | 0.014 | | 0.000 | | 0.250 |
| std.dev | 0.118 | | 0.004 | | 0.500 |
| coef.var | 0.546 | | 0.605 | | 0.336 |

(Provided by R Studio)

(Figure 2: descriptive statistics for male table 2)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | meanfreq | Sd | median | Q25 | Q75 | IQR | skew | kurt | sp.ent |
| nbr.val | 1138 | 1138 | 1138 | 1138 | 1138 | 1138 | 1138 | 1138 | 1138 |
| nbr.null | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nbr.na | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| min | 0.004 | 0.002 | 0.004 | 0.001 | 0.007 | 0.001 | 0.021 | 0.090 | 0.033 |
| max | 0.009 | 0.004 | 0.010 | 0.009 | 0.011 | 0.007 | 0.159 | 0.986 | 0.038 |
| range | 0.004 | 0.002 | 0.005 | 0.008 | 0.004 | 0.006 | 0.137 | 0.896 | 0.005 |
| sum | 7.880 | 2.805 | 8.147 | 5.467 | 10.279 | 4.812 | 82.623 | 336.202 | 40.80 |
| median | 0.007 | 0.002 | 0.007 | 0.005 | 0.009 | 0.004 | 0.068 | 0.245 | 0.036 |
| mean | 0.007 | 0.002 | 0.007 | 0.005 | 0.009 | 0.004 | 0.073 | 0.295 | 0.036 |
| SE.mean | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.005 | 0.000 |
| CI.mean.0.95 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.010 | 0.000 |
| var | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.027 | 0.000 |
| std.dev | 0.001 | 0.000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.025 | 0.163 | 0.001 |
| coef.var | 0.115 | 0.125 | 0.169 | 0.190 | 0.083 | 0.155 | 0.345 | 0.553 | 0.028 |
|  | | | | | | | | | |
|  | sfm | Mode | centroid | meanfun | minfun | maxfun | Meandom | mindom | max  dom |
| nbr.val | 1138 | 1138 | 1138 | 1138 | 1138 | 1138 | 1138 | 1138 | 1138 |
| nbr.null | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nbr.na | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| min | 0.003 | 0.000 | 0.004 | 0.003 | 0.000 | 0.009 | 0.002 | 0.000 | 0.005 |
| max | 0.032 | 0.011 | 0.009 | 0.007 | 0.003 | 0.011 | 0.088 | 0.006 | 0.551 |
| range | 0.029 | 0.011 | 0.004 | 0.004 | 0.003 | 0.002 | 0.086 | 0.006 | 0.546 |
| sum | 20.108 | 7.742 | 7.880 | 5.368 | 1.676 | 11.896 | 37.387 | 1.495 | 222.46 |
| median | 0.017 | 0.007 | 0.007 | 0.005 | 0.002 | 0.011 | 0.031 | 0.001 | 0.197 |
| mean | 0.018 | 0.007 | 0.007 | 0.005 | 0.001 | 0.010 | 0.033 | 0.001 | 0.195 |
| SE.mean | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.003 |
| CI.mean.0.95 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.006 |
| var | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.010 |
| std.dev | 0.006 | 0.003 | 0.001 | 0.001 | 0.001 | 0.001 | 0.016 | 0.001 | 0.099 |
| coef.var | 0.328 | 0.410 | 0.115 | 0.118 | 0.384 | 0.056 | 0.479 | 1.060 | 0.507 |
|  | | | | | | | | | |
|  | Dfrange | | Modindx | | label |  | | | |
| nbr.val | 1138 | | 1138 | | 1138 |
| nbr.null | 0 | | 0 | | 0 |
| nbr.na | 0 | | 0 | | 0 |
| min | 0.005 | | 0.001 | | 1 |
| max | 0.550 | | 0.036 | | 1 |
| range | 0.546 | | 0.036 | | 0 |
| sum | 220.964 | | 7.387 | | 1138 |
| median | 0.196 | | 0.005 | | 1 |
| mean | 0.194 | | 0.006 | | 1 |
| SE.mean | 0.003 | | 0.000 | | 0 |
| CI.mean.0.95 | 0.006 | | 0.000 | | 0 |
| var | 0.010 | | 0.000 | | 0 |
| std.dev | 0.099 | | 0.004 | | 0 |
| coef.var | 0.511 | | 0.657 | | 0 |

(Provided by R Studio)

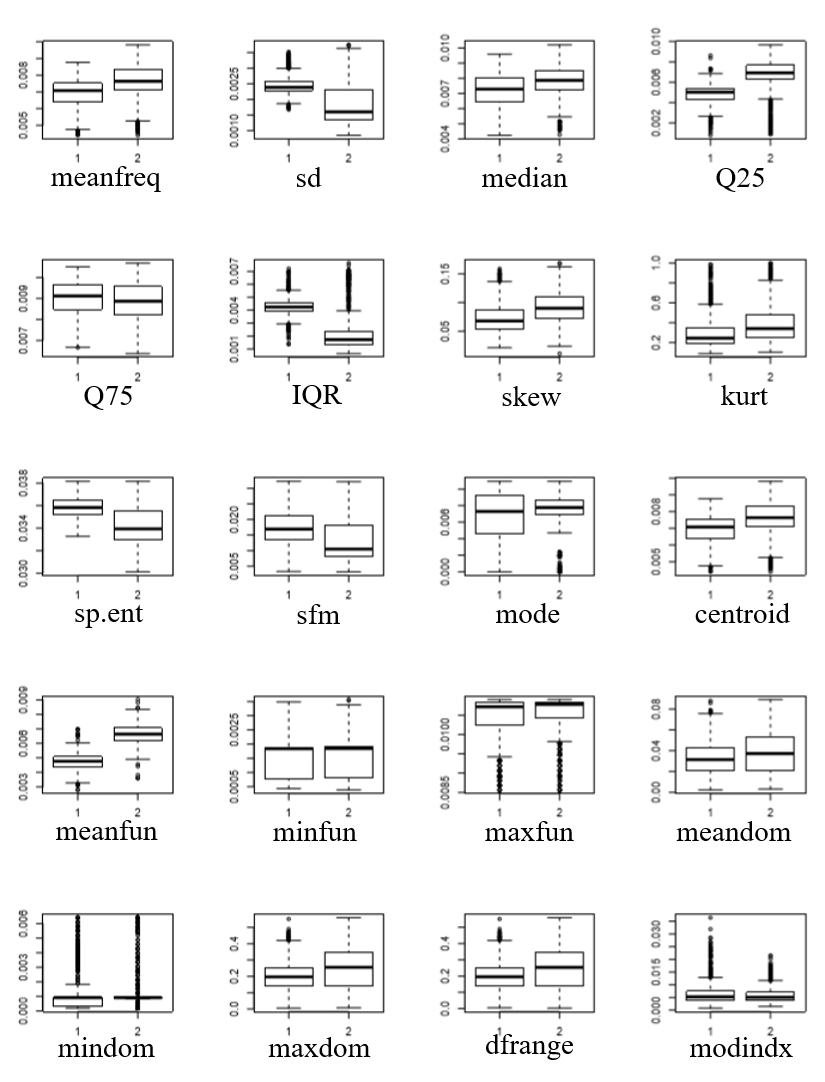
(Figure 3: descriptive statistics for female table 3)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | meanfreq | Sd | median | Q25 | Q75 | IQR | skew | kurt | sp.ent |
| nbr.val | 1092 | 1092 | 1092 | 1092 | 1092 | 1092 | 1092 | 1092 | 1092 |
| nbr.null | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nbr.na | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| min | 0.004 | 0.001 | 0.004 | 0.001 | 0.006 | 0.001 | 0.011 | 0.103 | 0.030 |
| max | 0.010 | 0.004 | 0.010 | 0.010 | 0.011 | 0.008 | 0.169 | 1.000 | 0.038 |
| range | 0.005 | 0.003 | 0.006 | 0.009 | 0.004 | 0.007 | 0.158 | 0.897 | 0.008 |
| sum | 8.333 | 2.043 | 8.579 | 7.259 | 9.684 | 2.424 | 99.989 | 416.382 | 37.344 |
| median | 0.008 | 0.002 | 0.008 | 0.007 | 0.009 | 0.002 | 0.090 | 0.341 | 0.034 |
| mean | 0.008 | 0.002 | 0.008 | 0.007 | 0.009 | 0.002 | 0.092 | 0.381 | 0.034 |
| SE.mean | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.005 | 0.000 |
| CI.mean.0.95 | 0.000 | 0,000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.002 | 0.011 | 0.000 |
| var | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.032 | 0.000 |
| std.dev | 0.001 | 0.001 | 0.001 | 0.002 | 0.001 | 0.001 | 0.026 | 0.178 | 0.002 |
| coef.var | 0.135 | 0.359 | 0.126 | 0.260 | 0.092 | 0.672 | 0.287 | 0.468 | 0.051 |
|  | | | | | | | | | |
|  | sfm | Mode | centroid | meanfun | minfun | maxfun | meandom | mindom | max  dom |
| nbr.val | 1092 | 1092 | 1092 | 1092 | 1092 | 1092 | 1092 | 1092 | 1092 |
| nbr.null | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nbr.na | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| min | 0.003 | 0.000 | 0.004 | 0.004 | 0.000 | 0.009 | 0.003 | 0.000 | 0.007 |
| max | 0.032 | 0.011 | 0.010 | 0.009 | 0.004 | 0.011 | 0.089 | 0.006 | 0.560 |
| range | 0.029 | 0.011 | 0.005 | 0.005 | 0.003 | 0.002 | 0.086 | 0.006 | 0.553 |
| sum | 14.369 | 8.147 | 8.333 | 7.247 | 1.663 | 11.476 | 41.484 | 1.702 | 261.11 |
| median | 0.011 | 0.008 | 0.008 | 0.007 | 0.002 | 0.011 | 0.037 | 0.001 | 0.256 |
| mean | 0.013 | 0.007 | 0.008 | 0.007 | 0.002 | 0.011 | 0.038 | 0.002 | 0.239 |
| SE.mean | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.004 |
| CI.mean.0.95 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.008 |
| var | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.017 |
| std.dev | 0.007 | 0.002 | 0.001 | 0.001 | 0.001 | 0.001 | 0.021 | 0.002 | 0.130 |
| coef.var | 0.528 | 0.291 | 0.135 | 0.102 | 0.402 | 0.052 | 0.544 | 1.157 | 0.545 |
|  | | | | | | | | | |
|  | Dfrange | | Modindx | | label |  | | | |
| nbr.val | 1092 | | 1092 | | 1092 |
| nbr.null | 0 | | 0 | | 0 |
| nbr.na | 0 | | 0 | | 0 |
| min | 0.003 | | 0.001 | | 2 |
| max | 0.559 | | 0.022 | | 2 |
| range | 0.556 | | 0.020 | | 0 |
| sum | 259.405 | | 6.497 | | 2184 |
| median | 0.253 | | 0.005 | | 2 |
| mean | 0.238 | | 0.006 | | 2 |
| SE.mean | 0.004 | | 0.000 | | 0 |
| CI.mean.0.95 | 0.008 | | 0.000 | | 0 |
| var | 0.017 | | 0.000 | | 0 |
| std.dev | 0.131 | | 0.003 | | 0 |
| coef.var | 0.550 | | 0.529 | | 0 |

(Provided by R Studio)

However, the above tables can be uncomfortable when the number of factors is increased. To improve to comprehend the data and to quickly catch the point, the data visualisation is necessary. For the below Figure 4, the boxplots illustrate descriptive statistics each 20 variables by gender. The male is 1 and the female is 2. Specifically, some plots such as the sd, the Q25, the IQR, the sp.ent and the meanfun indicate the distributions can be explicitly separated by genders by using interquartile range and mean. Smoothly, the variables such as the menfreq, the median, the skew, the kurt and the centroid could be also possible to show the distribution is different by gender by using interquartile range and mean as a central tendency. Moreover, some central tendency of the male values is lower than the female values such as the meanfreq, the median, the Q25, the skew, the kurt, the centroid, the meanfun, the meandom, the maxdom and the frange. By contrast, the female value is lower than the male such as the sd, the IQR and the sp.ent. As the compared between tables and plots, the visualisation helps to quickly understand and to catch the more and hidden insight.

(Figure 4: descriptive statistics boxplot 1)



(Provided by R Studio)

By using Principal Components Analysis (PCA), the dimension is reduced and the number of variables is decreased to analyse so it helps to improve accuracy and to apply in the candidate models to the data for classifying, clustering or finding relationships. The PCA is important when the data set has numerous variables so this data set needs to be applied.

As seen from the below Figure 5, this table is the results of attributes of PCA using R. These numbers each principal component represent by standard deviation, the proportion of variance and cumulative proportion. That proportion is important to reflect how much make important variable up between 0 to 1. When the proportion is as most closely as 1 and the highest value of variance, it is the first principal component. The next order of components is the descend dependent on the value of variance. This order is easy to analyse to reduce dimension. For the below Figure 6, the cumulative proportion is depicted at a glance easily so it is useful to choose components to analyse. For this data set, PC1 to PC4 is chosen.

(Figure 5: PCA table 1)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PC1 | PC2 | PC3 | PC4 | PC5 | | PC6 | PC7 | PC8 | | PC9 |
| Standard deviation | 2.991 | 1.8759 | 1.33969 | 1.15181 | 1.07600 | | 0.93751 | 0.87441 | 0.8299 | | 0.74806 |
| Proportion of Variance | 0.426 | 0.1676 | 0.08547 | 0.06317 | 0.055513 | | 0.04185 | 0.03638 | 0.0328 | | 0.02665 |
| Cumulative Proportion | 0.426 | 0.5936 | 0.67907 | 0.74224 | 0.79737 | | 0.83923 | 0.87561 | 0.9084 | | 0.93506 |
|  | | | | | | | | | | | |
|  | PC10 | PC11 | PC12 | PC13 | PC14 | | PC15 | PC16 | PC17 | | PC18 |
| Standard deviation | 0.69381 | 0.52108 | 0.42142 | 0.41271 | 0.33165 | | 0.27519 | 0.20961 | 0.15682 | | 0.09358 |
| Proportion of Variance | 0.02292 | 0.01293 | 0.00846 | 0.00811 | 0.00524 | | 0.00361 | 0.00209 | 0.00117 | | 0.00042 |
| Cumulative Proportion | 0.95798 | 0.97091 | 0.97936 | 0.98748 | 0.99271 | | 0.99632 | 0.99841 | 0.99958 | | 1.00000 |
|  | | | | | | | | | | | |
|  | PC19 | | PC20 | | | PC21 | | | |  | |
| Standard deviation | 8.424e-9 | | 2.843e-11 | | | 2.008e-15 | | | |
| Proportion of Variance | 0.000e+00 | | 0.000e+00 | | | 0.000e+00 | | | |
| Cumulative Proportion | 1.000e+00 | | 1.000e+00 | | | 1.000e+00 | | | |

(Provided by R Studio)

(Figure 6: PCA plot 1)

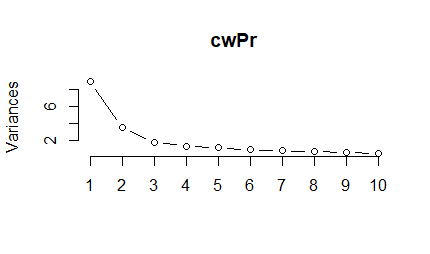
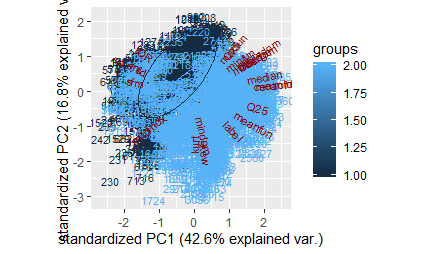
(Provided by R Studio)

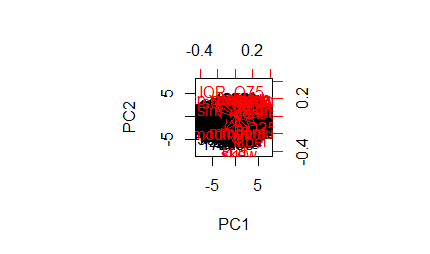
Figure 7, 8, 9 and 10 are constructed by using R. The functions of construction are ggbiplot, biplot and two ggplot respectively. As seen from the made ggbiplot, the x-axis and y-axis indicate PC1 and PC2 respectively. The variables as red colour’s letter express the linear regression so their gradients indicate the slope of the straight line. The made by biplot means same as previously analyzing the made ggbiplot but biplot has expressing directly straight line unlike expression of ggbiplot. When the slope is positive, the expressed value as red-letter on between PC1 and PC2 are positively related but when the slope is negative, the relation is negative.

(Figure 7: PCA ggbiplot)



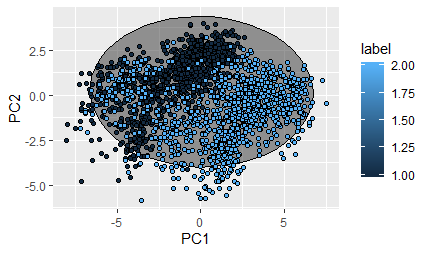
(Provided by R Studio)

(Figure 8: PCA biplot)



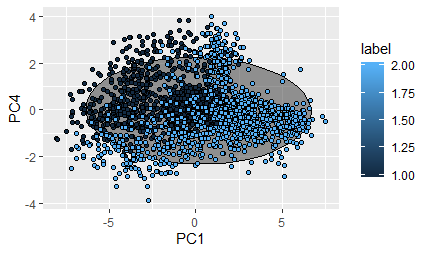
(Provided by R Studio)

(Figure 9: PCA plot 2)



(Provided by R Studio)

(Figure 10: PCA plot 3)

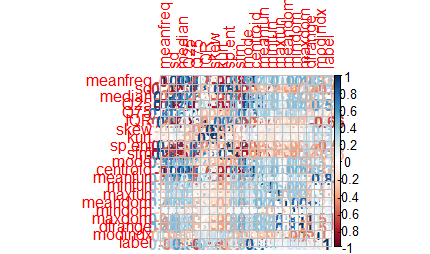


(Provided by R Studio)

Furthermore, for the above ggbiplot from Figure 7, the values are grouped by gender as expressing the circle in the plot so it means the values in the circle and skewed circle are the most related to gender. Likewise, for the above Figures 9 and 10 as the made ggplot, R function, the dots in both the plots between PC1 and PC2 as well as between PC1 and PC4 are fairly able to differentiate by gender at glance. Therefore, PC1, PC2 and PC4 are important attributes to analyse in the data set by using PCA before making the model to use voice data for separating by gender.

As seen from the below Figure 11, this plot depicts by the number how much close as the relationship between variables. When this number is a strong relationship in two variables between vertical and horizontal value each other, the number is as close 1 or -1 as possible so if the vertical and horizontal value is same, this correlation’s value is 1. The variable, 'label', as one of both axes in this plot is selected because the desired model as discriminate by gender(label) is necessary.

(Figure 11: correlation plot 1)



(Provided by R Studio)

The highest correlation coefficient by variables is the meanfun variable (0.86) as the average of fundamental frequency measured across the acoustic signal. It means that the meanfun and the label are the most related to one another. The next higher value of correlation coefficient is the IQR (-0.66) as the interquartile range (in kHz) which is the difference between lower(Q25) and upper quartile(Q75). This value is negative so when the IQR value is increased, the label value is decreased and vice versa. These two values are useful to make accurate linear regression model later.

See below Figure 12, by using other values as results to obtain from the previous phase by analyzing PCA, the correlation coefficient is determined. The highest value is PC2 (-0.51). It means that PC2 is the most one of the principal components related to differentiating by gender. The next higher value is PC1 (0.5). Nevertheless, these coefficients are not strong evidence that they are related in gender variable because they are not significantly close to 1 or -1.

(Figure 12: correlation plot 2)



(Provided by R Studio)

Before the linear model is created, the meanfun and the IQR by gender are identified the descriptive analysis of data such as mean and standard deviation. It is helpful to identify how the quantity values as descriptive data are differentiative by gender. As seen from the below Figure 13 and Figure 14, for the measure of the position such as quartiles, the upper and lower quartile values about the meanfun for the male is lower than female but the IQR’s upper and lower quartile value for the female is lower than male.

(Figure 13: the descriptive statistics about meanfun quartile 1)

|  |  |  |
| --- | --- | --- |
| Quartile | Male | Female |
| 0 | 0.004403210 | 0.0043998809 |
| 25(lower) | 0.00642605 | 0.007113942 |
| 50 | 0.007077998 | 0.007635901 |
| 75(upper) | 0.007528061 | 0.008348438 |
| 100 | 0.009812562 | 0.007213559 |

(Figure 14: the descriptive statistics about IQR quartile 2)

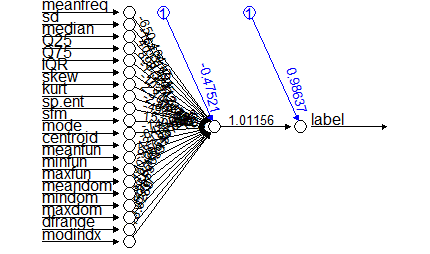
|  |  |  |
| --- | --- | --- |
| Quartile | Male | Female |
| 0 | 0.001360165 | 0.0006481204 |
| 25(lower) | 0.003915518 | 0.0013164284 |
| 50 | 0.004239664 | 0.0017282543 |
| 75(upper) | 0.004567052 | 0.0023719913 |
| 100 | 0.007213559 | 0.0076401272 |

The first model, linear model, is made by using R. The resulting of the previous phase on analyzing PCA and correlation is referred and contributes to this model by choosing component and variables as the most related with gender variable. The four linear models by selected variables are created. The two linear models are made by the variables as the highest and next higher correlation and the two left models are made by the principal components as the highest and next higher correlation obtained an analysis of PCA. The created linear model has calculated the quantity values so when the value is close to 1 or rounded to 1, it can be determined male and when the value is close to 2 or rounded to 2, it can be decided female. Then, the test of four linear models can be evaluated by using the confusion matrix. This model is created manually so after the factors of the linear model such as slope and intercept and the number of testing data is identified, the results of slope and intercept in each model put in the linear equation by using the function methods.

Before, the ANNs model is tested, performance resilience through cross validation is necessary to examine. The second model is made by using an artificial neural network in R studio. An Artificial Neural Networks (ANN) is a computational model as perceptron that comprises a combination of simple processing units that mutually exchange information by transmitting the signal to each other over a large number of weighted connections as well as adding to bias then using activation function. The perceptron has four main steps to the procedure that is to receive inputs, to weight inputs, to sum inputs and to achieve outputs. This model is the difference in the linear model because the ANN model is considered all of the variables by selected input. When the hidden value is more than 2, the backpropagation method is necessary to improve more accurate. The results of these models have automatically calculated the prediction so it can be compared and evaluated between predictive and actual values by using confusion matrix.

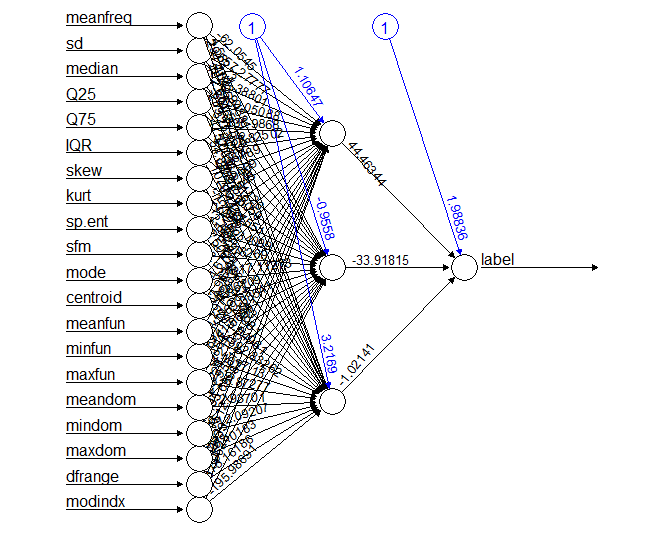
The four types of the second model will be tested. The below Figure 15, 16, 17 and 18 are computed neural network. The plots indicate the inputs, neurons as second located circles in its hidden layer, weights as a black line and the bias as the blue line. The first one is by putting in 20 variables so the numbers of inputs and are 20 as well as the number of neurons in its the 1 hidden layer and bias are 1. Secondly, the model is tested by changing the number of the neuron and its hidden layers from 1 to 3 and the numbers of inputs is the same as the previous model and the number of bias is 3. Next, when the usage of selected four principal components is instead of the variables and the numbers of inputs as is 4 and the numbers of neurons in its 1 hidden layer and bias are 1, the model will be tested. Finally, the number of neurons and its hidden layer changes from 1 to 3, as well as the numbers of input are the same and the number of bias is 3 then the model is tested.

(Figure 15: ANNs plot 1)

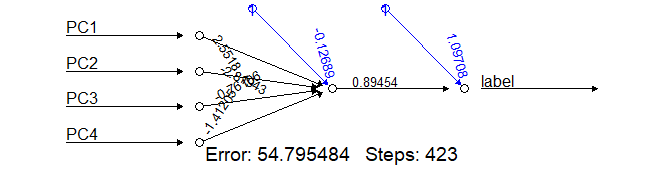


(Provided by R Studio)

(Figure 16: ANNs plot 2)

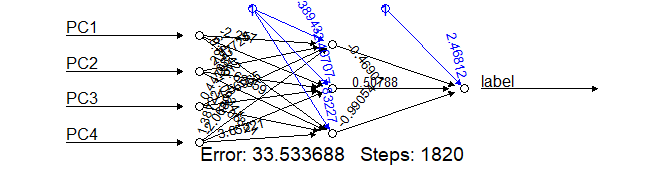
(Provided by R Studio)

(Figure 17: ANNs plot 3)



(Provided by R Studio)

(Figure 18: ANNs plot 4)



(Provided by R Studio)

The k-nearest neighbour algorithm makes the third model by using R. This is supervised learning like ANNs and the example of instance-based learning. By using distance measure as the Euclidean, the distance is calculated between the stored data and new instance. This report has four types of the model by changing k and inputs such as original variables and selected four principal components. Firstly, when k is equal to 1, the inputs are all of 20 variables. Moreover, the second type of model will be tested when k changes 1 to 2 and inputs are same. Thirdly, inputs replace variables to selected four principal components and k is 1 then the model will be tested. Finally, when k changes 1 to 2 and inputs are selected four principal components, the model will be performed. The results of these models have automatically calculated the prediction thus comparison and evaluation are available by using confusion matrix.

The final algorithm is the hierarchical clustering for grouping objects based on the similarity in data set so as to make a suitable model. This is the unsupervised learning and has the two types of algorithm which are agglomerative and divisive clustering. The agglomerative is that the one single cluster is determined by merging between the most similar clusters but the divisive clustering is that all observation is in their own cluster by dividing between the most heterogeneous cluster. The three types of models will use in this report. The first type is when the model uses only 20 variables. Second, the inputs change 20 variables to selected four principal components. The final model is the usage of 20 variables and the selected four principal components. The results of the three models have automatically calculated the prediction. Hence, it can be compared and evaluated by using the distance matrix.

1. Performance Evaluation

Before the diverse models are created and evaluated, the data set needs to tune by dividing by three departments. Specifically, for the first in the process, the labelled data set is separated training set and test set and then the training set is isolated as a training set and validation set in the learning process. After that, the training set learns models to select a model by combining the validation set. From that time, the learned model is checked by the test set. By typically splitting labelled data set, 60% of the data for training, 20% of the data for validation, and 20% of the data for testing comprise the overall data set. In the R application, in order to split the data into training and testing sample, the samplesize is created as the 70 percentages of data set. By using the set.seed function in R, it can generate the random number to set constraints for the test. Therefore, the training data has 1561 observers as the 70 percentages of data set, and the testing data has 669 observers as the left of data set. The candidate models use the testing data to test and evaluate.

For the below tables, these are confusion matrix. The male is 1 and the female is 2. Furthermore, the vertical is prediction and the horizontal is actual. It is useful to evaluate the model by identifying accuracy, precision, recall and false positive rate. Before analysis of this confusion matrix, the terms are identified. There are four types of results in this table when classification predictions are performed by using ANNs. The first type is true positives (TP) which occur when predicted and actual observations belong to the same class. The next type is true negatives (TN) which occur when the predicted values do not belong to a class likewise it actual values does not exist that class. The third type is false positives (FP) which are when actual values do not belong to a class but predicted values belong to that class. The final type is false negative (FN) which are when actual values do not belong to a class but predicted values belong to that class. The first metric in the confusion matrix is accuracy which is the sum of the TP and TN is divided by the sum of all results. However, a measure of the model performance cannot be suitable for accuracy so precision is necessary. Its calculation is the TP is divided by the sum of the TP and the FP. Another matric is the recall as true positive rate which is the TP is divided by the sum of TP and the FN. The final metric is false positive rate which is the FP is divided by the sum of the FP and the TN.

The linear models’ evaluations have the four types for the below four confusion matrixes. The first linear model is made by using only the meanfun variables as the highest correlation value with gender because the linear model is composed one independent and one dependent variable. The linear model equation is prediction = 371.1902 \* explanatory variable(meanfun) -0.6017 as rounded to first decimal place. For below Figure 19, the 3 is also created because the values round to the first decimal. However, it is errors of prediction so its value will be able to classify type 2 error when male is true but when female is true, its errors can be classified type 1 error. when the male is positive and female is negative, the true positive is 336, the true negative is 294, the false positive is 33 and the false negative is 6. The accuracy is 0.9417 as rounded the fourth decimal place so it means accuracy is same as representing 94%. The precision is 0.9106, 91%. The recall is 0.9825 and the false positive rate is 0.1092. Otherwise, when female is positive, the precision is 0.98. The recall is 0.8991 and the false positive rate is 0.0175.

(Figure 19: confusion matrix table 1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Confusion matrix (hidden 1) | | Prediction | | |
| 1 | 2 | 3 |
| Actual | 1 | 336 | 4 | 0 |
| 2 | 33 | 294 | 2 |

(Provided by R Studio)

Secondly, by using the IQR variables as the next higher correlation, the linear model is created. The second linear model equation is prediction = -212.673 \*explanatory variable (IQR)+2.184 as rounded to first decimal place. For below Figure 20, when the male is true and female is false, the true positive is 331, the true negative is 275, the false positive is 54 and the false negative is 9. The accuracy is 0.9058, 91%. The precision is 0.8597, 86%. The recall is 0.9735, 97% and the false positive rate is 0.1641, 16%. On the other hand, when female is true, the precision is 0.9683, 96%. The recall is 0.8358, 84% and the false positive rate is 0.0264, 3%.

(Figure 20: confusion matrix table 2)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 1) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 331 | 9 |
| 2 | 54 | 275 |

(Provided by R Studio)

The next two models are by using the selected principal component variables as the highest and next higher correlation. The third linear model equation in terms of PC1 is prediction = 0.08369 \*explanatory variable (PC1) + 1.49521 as rounded to first decimal place. For below Figure 21, when the male is positive and female is negative, the true positive is 248, the true negative is 241, the false positive is 88 and the false negative is 92. The accuracy is 0.7309, 73%. The precision is 0.7381, 74%. The recall is 0.7294, 73% and the false positive rate is 0.2674, 27%. Differently, when female is positive, the precision is 0.7237, 73%. The recall is 0.7325, 73% and the false positive rate is 0.2705, 27%.

(Figure 21: confusion matrix table 3)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 1) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 248 | 92 |
| 2 | 88 | 241 |

(Provided by R Studio)

The final linear model equation as next higher correlation with respect to principal component 2 is prediction = -0.1394 \*explanatory variable (PC2) + 1.4923 as rounded to first decimal place. As seen from the Figure 22, when the male is true and female is false, the true positive is 274, the true negative is 220, the false positive is 109 and the false negative is 66. The accuracy is 0.7384, 74%. The precision is 0.7154, 72%. The recall is 0.8058, 81% and the false positive rate is 0.3313, 33%. Differently, when female is positive, the precision is 0.7692, 77%. The recall is 0.6686, 67% and the false positive rate is 0.1941, 19%.

(Figure 22: confusion matrix table 4)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 1) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 274 | 66 |
| 2 | 109 | 220 |

(Provided by R Studio)

As an analysis of the four types of the linear model, when the accuracies are compared, the highest value is 91% as the second linear model by using IQR variable. The third and fourth linear model’s accuracies are decreased than the first and second model. This means that the variables are decided to improve more accuracy than the selected principal components when the linear regression model is created. In addition, overall, the precision of the predicted to female is higher the predicted to male.

When ANNs models are evaluated, k-fold cross validation is performed by using ‘for’ function as looping. The variations in the accuracy of the model is considered because of setting the duration of the training. The training data with lengths of 10 to 20 is examined. When training data are large, the accuracy of the model is developed. In addition, the Root Mean Square Error (RMSE) which is the standard deviation of the residuals is calculated the samples for each of length of 50 are selected. In this report, the k-fold cross validation is performed when one of four types of ANNs by using 1 hidden and 20 variables is tested.

The model is significantly developed by using artificial neural networks (ANNs). The four types of ANNs are tested according to earlier explanation. Firstly, when applying classification prediction, the male is positive and female is negative. As seen from the Figure 23, when the four types of outcomes that could occur in this confusion matrix by inputting hidden as 1, true positive is 335, the true negative is 319, the false positive is 10 and the false negative is 5. The accuracy is 0.9776 as rounded the fourth decimal place so it means accuracy is 98%. The precision is 0.9710, 97% when the male is positive. The recall is 0.9853, 99% and the false positive rate is 0.0304, 3%. Otherwise, when female is positive, the precision is 0.9845, 98%. The recall is 0.9696, 97% and the false positive rate is 0.0147, 1%.

(Figure 23: confusion matrix table 5)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 1) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 335 | 5 |
| 2 | 10 | 319 |

(Provided by R Studio)

See the below Figure 24, when backpropagation is performed and to increase hidden than the previous model, the results are a little bit changed. When the male is 1 and the female is 2, the true positive is 332, the true negative is 319, the false positive is 10 and the false negative is 8. Hence, the accuracy is 0.9731 and the precision is 0.9708. The recall is 0.9764 and the false positive rate is 0.0304. When the female is positive, the precision is 0.9755. The recall is 0.9696 and the false positive rate is 0.0235.

(Figure 24: confusion matrix table 6)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 3) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 332 | 8 |
| 2 | 10 | 319 |

(Provided by R Studio)

When the third model and fourth model are tested, the results of two models’ confusion matrixes are the same as the first ANNs model. The test is continually iterated to identify that the results still are the same. The results do not change. Sometimes, the second ANNs model as changing hidden 1 to 3 is the same results in a confusion matrix of the first ANNs model. It means that ANNs optimise automatically regardless of changing hidden as well as the usage of variables or principal components.

The next tested models are k-nearest neighbour (k-NN) as supervised learning like ANNs. When the first model is when the k is equal to 1 and the data is the 20 variables. For the below Figure 25, when the male is true and female is false, the four outputs are out such as true positive, the true negative, the false positive and the false negative. The outputs are 259, 227, 102 and 81 respectively. Then, the accuracy is 0.7265, 73%. The precision, the recall and the false positive rate are 0.7174, 0.7618 and 0.3100 respectively. Otherwise, when female is positive, the precision is 0.7370, 74%. The recall is 0.6899, 69% and the false positive rate is 0.2382, 24%.

(Figure 25: confusion matrix table 7)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 3) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 259 | 81 |
| 2 | 102 | 227 |

(Provided by R Studio)

As seen from the below Figure 26, when increasing k value 1 to 2 and the same 20 variables put in this model, the results are a little bit changed. When the second k-NN model is tested, the accuracy is 0.7234, 72%. When male is true, the precision, the recall and the false positive rate are 0.7146, 0.7588 and 0.3130 respectively. Moreover, when the female is true, the precision, the recall and the false positive rate are 0.7337, 0.6869 and 0.2411 respectively.

(Figure 26: confusion matrix table 8)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 3) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 258 | 82 |
| 2 | 103 | 226 |

(Provided by R Studio)

For the below Figure 27, the third k-NN model is tested when k value is 1 and the four principal components put in this model. The third k-NN model is developed more than the two previous k-NN model because the accuracy is growing such as 0.9446, 94%. When male is positive, the precision, the recall and the false positive rate are 93%, 96% and 10% respectively. On the other hand, when the female is positive, the precision, the recall and the false positive rate are 96%, 93% and 4% respectively.

(Figure 27: confusion matrix table 9)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 3) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 326 | 14 |
| 2 | 23 | 306 |

(Provided by R Studio)

When the final k-NN model is tested, the below Figure 28 is identified. This model’s conditions are that k value is 2 and the four principal components put in this model. The fourth k-NN model is also developed more than the two previous k-NN model and the results are similar as third model. The overall accuracy is 94% When male is true, the precision, the recall and the false positive rate are 92%, 96% and 9% respectively. On the other hand, when the female is positive, the precision, the recall and the false positive rate are 95%, 91% and 4% respectively.

(Figure 28: confusion matrix table 10)

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion matrix (hidden 3) | | Prediction | |
| 1 | 2 |
| Actual | 1 | 325 | 15 |
| 2 | 28 | 301 |

(Provided by R Studio)

When the four types of above k-NN model, the model can be compared, the higher overall accuracies are identified by using principal components without original variables and regardless of the number of k. In addition, like linear and ANNs, the precision of the predicted to female is higher the predicted to male.

For the below tables, those are distance matrix. Unlike confusion matrix, the number is not labelled and the matrix is automatically separated by merging between the objects as minimum distance. Therefore, the matrix shows the clusters for similar features.

Firstly, when the hierarchical clustering is tested, the below Figure 29 is identified. The 20 variables without principal components put in the model. This model is divided into three clusters even if the goal is to make a model differentiate by gender as two clusters. In addition, the 1 is comprised 84% of all. The male values in testing data is 340, so approximate accuracy is 59%. It means this model cannot be accurate to differentiate by gender because the male and female values are evenly distributed as much as possible in the testing data set.

(Figure 29: distance matrix table 1)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 1 | 566 | 0 | 0 |
| 2 | 0 | 75 | 0 |
| 3 | 0 | 0 | 28 |

(Provided by R Studio)

Secondly, the four selected principal components are replaced from 20 variables, when the second hierarchical clustering is tested. This model is also divided by three clusters like the first model but the number of comprised values in each group as labelled 1 and 2 is evenly more distributed than the first model. The 1 is comprised 31% of all and the 2 is the 65% percentages of all. In addition, based on male value, the approximate accuracy is 60%. the male values in testing data is 340. It means this model is close to the goal for differentiating by gender. Moreover, the model can be evaluated that the usage of principal components affects more accurate the results than the usage of original variables for the occasion that the model is created by using hierarchical clustering.

(Figure 30: distance matrix table 2)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 1 | 207 | 0 | 0 |
| 2 | 0 | 432 | 0 |
| 3 | 0 | 0 | 30 |

(Provided by R Studio)

Finally, the four selected principal components and 20 original variables put in the final hierarchical clustering model. When the final hierarchical clustering is tested, the model is developed and more accurate to differentiate by two clusters. This model is also divided by three clusters like the first and second model but the number of 3 values are decreased. In addition, 1 and 2 are evenly more distributed than the first and second model. The 1 is comprised 58% in testing data and the 2 is the 41 percentages of all. The approximate accuracy is 87.4% in terms of male values. Therefore, the model can be evaluated that the model is more accurate when the combination of selected principal components and original variables put in the hierarchical clustering.

(Figure 31: distance matrix table 3)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 1 | 389 | 0 | 0 |
| 2 | 0 | 276 | 0 |
| 3 | 0 | 0 | 4 |

(Provided by R Studio)

The models are necessary to compare so that one of the candidates is selected. Firstly, for the below Figure 32 and 33, the ANNs model has the highest performance and accuracy by inputting every 20 variables without selected principal components. The four types of ANNs model are more accurate than 97% and the false positive rate is definitely low. The two types of the linear regression model by using selected variables as highest and next higher correlation are the next higher model as more than 91%. However, the two types of k-NN models are also higher by using selected principal components. This means that selected one of the variables as the highest correlation prefer to use linear regression modelling and selected one of the principal components as the highest variance prefer to use k-NN modelling. As a result, even if ANNs modelling is the highest accuracy and needs to test, the other types of model can need to test so as to develop and compare the final model.

When the models compared by other factors such as the mean square of error, type 1/2 error, F-score and P-value. Firstly, the mean square of error is checked, the mean square of error indicates the difference between prediction and actual values. Mathematically, mean square of error is equal to the pooled standard deviation so the mean in the testing data is about 1.48. This means that when the model is satisfied, the mean square of error is as close as mean (1.48). Therefore, the mean square of error is the inverse proportion of accuracy thus the ANNs model is also the lowest values of the mean square of error.

For the below Figure 32, the models can be compared by type 1/2 error. The type 1 error is false positive in the confusion matrix so the null hypothesis is correct then the test is rejected. Moreover, the type 2 error is false negative in the confusion matrix so the null hypothesis is not correct then the test is accepted. When male change positive to negative, the type 1 error and type 2 error values are exchanged. Overall, when the male is positive, the type 1 error of male values is higher than female likewise the type 2 error of female values is lower than male. Back to the comparing model, these errors are also in inverse proportion of accuracy thus ANNs is the lowest values of type 1 and type 2 error.

The next checked factor is F-score or F-measure. It is the harmonic mean of precision and recall so it is possible to calculate by using confusion matrix. The F-score formula is F = 2/(1/recall +1/precision) or F = 2\*(precision\*recall)/(precision+recall). The higher F-score is as close as 1 thus ANNs models’ F-scores are higher than other models. Therefore, when the accuracy of the model is high, the F-score is also high. Furthermore, the p-value is identified, the p-value is also important to reject the null hypothesis. For Figure 32, the results of p-values are micro-value thus they represent the same values all of the models.

In conclusion, the satisfied model is selected based on comparing the candidate model. For testing the model, the ANNs model by using 20 variables is determined to differentiate by gender and close to the goal or the approach. The other models such as the first and second linear model and the third and fourth k-NN models can be decided and continually considered because of demonstration such as accuracy. The data analytics process is repeated so that the satisfied models are searched and the final model is developed and able to close to approach the goal.

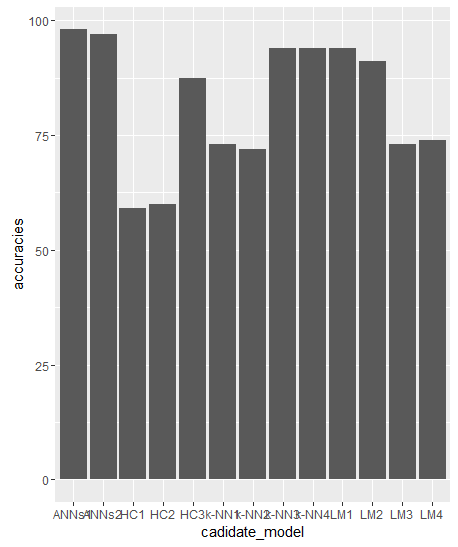
(Figure 32: candidate model table 1)

(m/f: male / female)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Mean square of error | Type 1  error  m/f | Type 2  error  m/f | F-score  m/f | p-value |
| Linear model 1 | 94% | 0.2118991 | 5%/1% | 1%/5% | 0.945/0.0.938 | 2.2e-16 |
| Linear model 2 | 91% | 0.2875236 | 8%/1% | 1%/8% | 0.913/0.0.897 | 2.2e-16 |
| Linear model 3 | 73% | 0.4440764 | 13%/14% | 14%/13% | 0.734/0.728 | 2.2e-16 |
| Linear model 4 | 74% | 0,4393427 | 16%/10% | 10%/16% | 0.758/0.715 | 2.2e-16 |
| ANNs model 1 | 98% | 0.1481165 | 1%/1% | 1%/1% | 0.978/0.977 | 2.2e-16 |
| ANNs model 2 | 97% | 0.1613021 | 1%/1% | 1%/1% | 0.974/0.973 | 2.2e-16 |
| k-NN model 1 | 73% | 0.4440764 | 15%/12% | 12%/15% | 0.739/0.713 | 2.2e-16 |
| k-NN mode 2 | 72% | 0.4537485 | 15%/12% | 12%/15% | 0.736/0.710 | 2.2e-16 |
| k-NN model 3 | 94% | 0.2117411 | 3%/2% | 2%/3% | 0.945/0.945 | 2.2e-16 |
| k-NN model 4 | 94% | 0.2113732 | 4%/2% | 2%/4% | 0.940/0.930 | 2.2e-16 |
| Hierarchical clustering 1 | 59% | 0.5948236 | 34%/34% | 34%/34% | 0.508/0.225 | 2.2e-16 |
| Hierarchical clustering 2 | 60% | 0.5799312 | 20%/20% | 20%/20% | 0.629/0.765 | 2.2e-16 |
| Hierarchical clustering 3 | 87.4% | 0.3237849 | 7%/9% | 9%/7% | 0.870/0.845 | 2.2e-16 |

(Provided by R Studio)

(Figure 33: bar plot 1)



(Provided by R Studio)

1. Reference

Cady, F. (2017). *The Data Science Handbook.* Wiley

R (Accessed: 18 Nov 2019)

R documentation[Online] Available at:  [https://www.rdocumentation.org//](%20https://www.rdocumentation.org//) (Accessed: 18 Nov 2019)

Schutt, R and O’Neil, C (2013). *Doing Data Science*. O’Reilly

Stack Overflow[Online] Available at: <https://stackoverflow.com/> (Accessed: 18 Nov 2019)

1. Appendix A – R script

##Coursework R Script

## Importing Data (csv)

voice\_data <- read.csv("M:/Data Science and Analytics/gender\_voice\_dataset.csv")

## Variable transformation (gender: char -> numeric)

voice\_data$label = ifelse(voice\_data$label == "male", 1, 2)

##data cleansing by removing outliers

outliers <- boxplot(voice\_data$meanfreq, plot = FALSE)$out

voice\_data[voice\_data$meanfreq %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$sd, plot = FALSE)$out

voice\_data[voice\_data$sd %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$median, plot = FALSE)$out

voice\_data[voice\_data$median %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$Q25, plot = FALSE)$out

voice\_data[voice\_data$Q25 %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$Q75, plot = FALSE)$out

voice\_data[voice\_data$Q75 %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$IQR, plot = FALSE)$out

voice\_data[voice\_data$IQR %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$skew, plot = FALSE)$out

voice\_data[voice\_data$skew %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$kurt, plot = FALSE)$out

voice\_data[voice\_data$kurt %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$sp.ent, plot = FALSE)$out

voice\_data[voice\_data$sp.ent %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$sfm, plot = FALSE)$out

voice\_data[voice\_data$sfm %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$mode, plot = FALSE)$out

voice\_data[voice\_data$mode %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$centroid, plot = FALSE)$out

voice\_data[voice\_data$centroid %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$meanfun, plot = FALSE)$out

voice\_data[voice\_data$meanfun %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$minfun, plot = FALSE)$out

voice\_data[voice\_data$minfun %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$maxfun, plot = FALSE)$out

voice\_data[voice\_data$maxfun %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$meandom, plot = FALSE)$out

voice\_data[voice\_data$meandom %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$mindom, plot = FALSE)$out

voice\_data[voice\_data$mindom %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$maxdom, plot = FALSE)$out

voice\_data[voice\_data$maxdom %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$dfrange, plot = FALSE)$out

voice\_data[voice\_data$dfrange %in% outliers, "isOutlier"] = "1"

outliers <- boxplot(voice\_data$dfrange, plot = FALSE)$out

voice\_data[voice\_data$modindx %in% outliers, "isOutlier"] = "1"

voice\_data$isOutlier = ifelse(is.na(voice\_data$isOutlier), "1", NA)

##data cleansing by removing missing and dupicate values

voice\_data <- na.omit(voice\_data)

voice\_data <- unique(voice\_data) #removing duplicate value

##Use the function stat.desc() to compute basic descriptive statistics

install.packages("pastecs", INSTALL\_opts = c('--no-lock'))

library(pastecs)

resDesc4 <- stat.desc(voice\_data[, 1:21])

round(resDesc4, 3)

##Standardisation: rescales data to have a mean (μ) of 0 and standard deviation (σ) of 1 (unit variance) it helps to be identical between variables

##voice\_data <- data.frame(scale(voice\_data[,1:20]), voice\_data$label)

##colnames(voice\_data)[21] = c("label")

##Normalisation: rescales the values into a range of [0,1]. This might be useful in some cases where all parameters need to have the same positive scale. However, the outliers from the data set are lost.

normalisation <- function(x) {

return ((x - min(x)) / (max(x) - min(x)))

}

voice\_data <- data.frame(normalisation(voice\_data[,1:20]), voice\_data$label)

colnames(voice\_data)[21] = c("label")

##Use the function stat.desc() to compute basic descriptive statistics

install.packages("pastecs", INSTALL\_opts = c('--no-lock'))

library(pastecs)

resDesc <- stat.desc(voice\_data[, 1:21])

round(resDesc, 3)

#differenciation by gender

voice\_male <- voice\_data[voice\_data$label == 1,]

voice\_female <- voice\_data[voice\_data$label == 2,]

##descriptive statistics for male

resDesc2 <- stat.desc(voice\_male[, 1:21])

round(resDesc2, 3)

##descriptive statistics for female

resDesc3 <- stat.desc(voice\_female[, 1:21])

round(resDesc3, 3)

##data visualisation (boxplot) with variables

par(mfrow = c(5,4))

boxplot(meanfreq~label, data=voice\_data)

boxplot(sd~label, data=voice\_data)

boxplot(median~label, data=voice\_data)

boxplot(Q25~label, data=voice\_data)

boxplot(Q75~label, data=voice\_data)

boxplot(IQR~label, data=voice\_data)

boxplot(skew~label, data=voice\_data)

boxplot(kurt~label, data=voice\_data)

boxplot(sp.ent~label, data=voice\_data)

boxplot(sfm~label, data=voice\_data)

boxplot(mode~label, data=voice\_data)

boxplot(centroid~label, data=voice\_data)

boxplot(meanfun~label, data=voice\_data)

boxplot(minfun~label, data=voice\_data)

boxplot(maxfun~label, data=voice\_data)

boxplot(meandom~label, data=voice\_data)

boxplot(mindom~label, data=voice\_data)

boxplot(maxdom~label, data=voice\_data)

boxplot(dfrange~label, data=voice\_data)

boxplot(modindx~label, data=voice\_data)

## PCA

cwPr <- prcomp(voice\_data[,1:21], center = TRUE, scale = TRUE)

summ\_cwPr <- summary(cwPr)

plot(cwPr, type = "l")

biplot(cwPr, scale = 0)

## Extract PC Scores

str(cwPr)

cwPr$rotation #identify eigenvalue

voice\_data[,22:25] <- cwPr$x[,1:4]

colnames(voice\_data)[22] = c("PC1")

colnames(voice\_data)[23] = c("PC2")

colnames(voice\_data)[24] = c("PC3")

colnames(voice\_data)[25] = c("PC4")

##data visualisation (boxplot) with variables

par(mfrow=c(2, 2))

boxplot(label~PC1 + PC2 + PC3 + PC4, data=voice\_data)

## Plot with GGBIPLOT

install.packages("devtools")

install.packages("usethis")

install.packages("Rcpp")

library(usethis)

library(Rcpp)

library(devtools)

install\_github("vqv/ggbiplot") #choose 3(NONE) and with library(devtools)

install.packages("ggplot2")

install.packages("plyr")

install.packages("scales")

install.packages("grid")

library(ggplot2)

library(plyr)

library(scales)

library(grid)

library(ggbiplot)

ggbiplot(cwPr, labels=rownames(voice\_data))

ggbiplot(cwPr, ellipse=TRUE, labels=rownames(voice\_data), groups=voice\_data$label)

## Plot with GGPLOT

ggplot(voice\_data, aes(PC1, PC2, col = label, fill = label)) + stat\_ellipse(geom = "polygon", col = "black", alpha = 0.5) + geom\_point(shape = 21, col = "black")

ggplot(voice\_data, aes(PC1, PC4, col = label, fill = label)) + stat\_ellipse(geom = "polygon", col = "black", alpha = 0.5) + geom\_point(shape = 21, col = "black")

##PC1/PC2, PC1/PC4

## Correlations between Vars and PCs

cor(voice\_data[, 1:21], voice\_data[,22:25])

##Visualizing the correlation matrix with variables by gender

install.packages("corrplot")

library(corrplot)

cor\_data = cor(voice\_data[1:21])

corrplot(cor\_data)

corrplot(cor\_data, method="number")

##Visualizing the correlation matrix with PC by gender

cor\_data2 = cor(voice\_data[21], voice\_data[22:25])

corrplot(cor\_data2)

corrplot(cor\_data2, method="number")

##descriptive analysis by gender by using group\_by after stong correlation

install.packages("dplyr")

library(dplyr)

voice\_data %>%

group\_by(label) %>%

summarise(mean\_gender = mean(meanfun), sd\_gender = sd(meanfreq))

quantile(voice\_male$meanfreq)

quantile(voice\_female$meanfreq)

voice\_data %>%

group\_by(label) %>%

summarise(mean\_gender = mean(IQR), sd\_gender = sd(IQR),qt\_gender = quantile(meanfreq))

quantile(voice\_male$IQR)

quantile(voice\_female$IQR)

##descriptive analysis by gender by using group\_by after stong correlation with PC

voice\_data %>%

group\_by(label) %>%

summarise(mean\_gender = mean(PC1), sd\_gender = sd(PC1))

voice\_data %>%

group\_by(label) %>%

summarise(mean\_gender = mean(PC2), sd\_gender = sd(PC2))

##Split the data into training and testing samples

samplesize = 0.70\*nrow(voice\_data)

set.seed(80)

index = sample(seq\_len(nrow(voice\_data)),size=samplesize)

datatrain = voice\_data[index,]

datatest = voice\_data[-index,]

## Linear Regression and Linear Model

linearMod1 <- lm(label~meanfun, data = datatest)

summary(linearMod1)

print(linearMod1)

for(i in 1:669) {

datatest[i,26] = round(371.1902\*datatest$meanfun[i]-0.6017,digits=0)

}

colnames(datatest)[26] = c("lmres1")

resultslm1 <- data.frame(actual = datatest$label, prediction = datatest$lmres1)

attach(resultslm1)

table(actual, prediction)

linearMod2 <- lm(label~IQR, data = datatest)

summary(linearMod2)

print(linearMod2)

for(i in 1:669) {

datatest[i,27] <- round(-212.673\*datatest$IQR[i] +2.132,digits=0)

}

colnames(datatest)[27] = c("lmres2")

resultslm2 <- data.frame(actual = datatest$label, prediction = datatest$lmres2)

attach(resultslm2)

table(actual, prediction)

linearMod3 <- lm(label~PC1, data = datatest)

summary(linearMod3)

print(linearMod3)

for(i in 1:669) {

datatest[i,28] <- round(0.08369\*datatest$PC1[i] +1.49521,digits=0)

}

colnames(datatest)[28] = c("lmres3")

resultslm3 <- data.frame(actual = datatest$label, prediction = datatest$lmres3)

attach(resultslm3)

table(actual, prediction)

linearMod4 <- lm(label~PC2, data = datatest)

summary(linearMod4)

print(linearMod4)

for(i in 1:669) {

datatest[i,29] <- round(-0.1394\*datatest$PC2[i] +1.4923,digits=0)

}

colnames(datatest)[29] = c("lmres4")

resultslm4 <- data.frame(actual = datatest$label, prediction = datatest$lmres4)

attach(resultslm4)

table(actual, prediction)

##ANN

install.packages("neuralnet")

library(neuralnet)

## Cross validation of neural network model

install.packages("boot")

install.packages("plyr")

library(boot)

library(plyr)

set.seed(50)

k = 50

RMSE.NN = NULL

List = list( )

for(j in 10:20){

for (i in 1:k) {

index = sample(1:nrow(voice\_data),j )

datatrainKF = voice\_data[index,]

datatestKF = voice\_data[-index,]

NN = neuralnet(label ~ meanfreq + sd + median + Q25 + Q75 + IQR + skew + kurt + sp.ent + sfm + mode + centroid + meanfun + minfun + maxfun + meandom + mindom + maxdom + dfrange + modindx, datatrainKF, hidden=1, linear.output =T)

predict\_testNN = compute(NN, datatestKF[,c(1:21)])

predict\_testNN = (predict\_testNN$net.result\*(max(voice\_data$label)-min(voice\_data$label)))+min(voice\_data$label)

RMSE.NN [i]<- (sum((datatest$label - predict\_testNN)^2)/nrow(datatestKF))^0.5

}

List[[j]] = RMSE.NN

}

Matrix.RMSE = do.call(cbind, List)

Matrix.RMSE

## hidden 1

NN = neuralnet(label ~ meanfreq + sd + median + Q25 + Q75 + IQR + skew + kurt + sp.ent + sfm + mode + centroid + meanfun + minfun + maxfun + meandom + mindom + maxdom + dfrange + modindx, datatrain, hidden=1, linear.output =T)

predict\_testNN = compute(NN, datatest[,c(1:21)])

results <- data.frame(actual = datatest$label, prediction = predict\_testNN$net.result)

plot(NN)

##predicting and making confusion matrix

roundedresults <- sapply(results,round,digits=0)

roundedresultsdf=data.frame(roundedresults)

attach(roundedresultsdf)

table(actual,prediction)

## hidden more than 2(3)

NN\_rprop\_multi <- neuralnet(label ~ meanfreq + sd + median + Q25 + Q75 + IQR + skew + kurt + sp.ent + sfm + mode + centroid + meanfun + minfun + maxfun + meandom + mindom + maxdom + dfrange + modindx, datatrain, algorithm = "rprop+", hidden=3, stepmax=1e6) #algorithm <- backproporation(if hidden is more than 2)

predict\_testNN2 = compute(NN\_rprop\_multi, datatest[,c(1:21)])

results2 <- data.frame(actual = datatest$label, prediction = predict\_testNN2$net.result)

plot(NN\_rprop\_multi)

##predicting and making confusion matrix (more than 2)

roundedresults2 <- sapply(results2,round,digits=0)

roundedresultsdf2=data.frame(roundedresults2)

attach(roundedresultsdf2)

table(actual,prediction)

## neural with PC (Hidden1)

NN\_PC = neuralnet(label ~ PC1+PC2+PC3+PC4, datatrain, hidden=1, linear.output =T)

predict\_testNN\_PC = compute(NN\_PC, datatest[,c(21:25)])

results\_PC <- data.frame(actual = datatest$label, prediction = predict\_testNN\_PC$net.result)

plot(NN\_PC)

##predicting and making confusion matrix

roundedresults3 <- sapply(results,round,digits=0)

roundedresultsdf3=data.frame(roundedresults3)

attach(roundedresultsdf3)

table(actual,prediction)

## neural with PC (Hidden3)

NN2\_PC = neuralnet(label ~ PC1+PC2+PC3+PC4, datatrain, algorithm = "rprop+", hidden=3, stepmax = 1e6)

predict\_testNN2\_PC = compute(NN2\_PC, datatest[,c(21:25)])

results2\_PC <- data.frame(actual = datatest$label, prediction = predict\_testNN2\_PC$net.result)

plot(NN2\_PC)

##predicting and making confusion matrix

roundedresults4 <- sapply(results,round,digits=0)

roundedresultsdf4=data.frame(roundedresults4)

attach(roundedresultsdf4)

table(actual,prediction)

##KNN with variable when k=1

install.packages("class")

library(class)

knn\_CW <- knn(datatrain[,1:20],datatest[,1:20],datatrain[,21],k=1)

table(datatest[,'label'],knn\_CW)

##KNN with variable when k=2

knn\_CW2 <- knn(datatrain[,1:20],datatest[,1:20],datatrain[,21],k=2)

table(datatest[,'label'],knn\_CW2)

##KNN with PC when k=1

knn\_CW3 <- knn(datatrain[,22:25],datatest[,22:25],datatrain[,21],k=1)

table(datatest[,'label'],knn\_CW3)

##KNN with PC when k=2

knn\_CW4 <- knn(datatrain[,22:25],datatest[,22:25],datatrain[,21],k=2)

table(datatest[,'label'],knn\_CW4)

##hierarchical clustering with variables (distance matrix)

datatest3 <- datatest[,1:20]

dist\_mat <- dist(datatest3, method = 'euclidean')

hclust\_avg <- hclust(dist\_mat, method = 'average')

plot(hclust\_avg)

cut\_avg <- cutree(hclust\_avg, k=3)

plot(hclust\_avg)

rect.hclust(hclust\_avg, k=3, border = 2:6)

suppressPackageStartupMessages(library(dplyr))

voice\_cl <- mutate(datatest3, cluster = cut\_avg)

count(voice\_cl, cluster)

table(voice\_cl$cluster, voice\_cl$cluster)

##hierarchical clustering with PC1 to PC4 (distance matrix)

datatest4 <- datatest[,22:25]

dist\_mat2 <- dist(datatest4, method = 'euclidean')

hclust\_avg2 <- hclust(dist\_mat2, method = 'average')

cut\_avg2 <- cutree(hclust\_avg2, k=3)

voice\_cl2 <- mutate(datatest4, cluster = cut\_avg2)

count(voice\_cl2, cluster)

table(voice\_cl2$cluster, voice\_cl2$cluster)

##hierarchical clustering with variables and PC (distance matrix)

datatest5 <- datatest[,-21]

dist\_mat3 <- dist(datatest5, method = 'euclidean')

hclust\_avg3 <- hclust(dist\_mat3, method = 'average')

cut\_avg3 <- cutree(hclust\_avg3, k=3)

voice\_cl3 <- mutate(datatest5, cluster = cut\_avg3)

count(voice\_cl3, cluster)

table(voice\_cl3$cluster, voice\_cl3$cluster)

## Analysis of results about accuracy of cadidate model

accuracy <- accuracy <- data.frame('cadidate\_model'=c('LM1','LM2','LM3','LM4','ANNs1','ANNs2','k-NN1','k-NN2','k-NN3','k-NN4','HC1','HC2','HC3'),'accuracies'=c(94,91,73,74,98,97,73,72,94,94,59,60,87.4))

ggplot(data = accuracy) +

geom\_bar(mapping = aes(x=cadidate\_model, y=accuracies), stat = "identity")

##Mean of square of the mean (=sdres)

install.packages("ANOVAreplication")

library(ANOVAreplication)

pooled.sd(resultslm1)

pooled.sd(resultslm2)

pooled.sd(resultslm3)

pooled.sd(resultslm4)

pooled.sd(roundedresultsdf)

pooled.sd(roundedresultsdf2)

pooled.sd(roundedresultsdf3)

pooled.sd(roundedresultsdf4)

pooled.sd(knn1)

pooled.sd(knn2)

pooled.sd(knn3)

pooled.sd(knn4)

pooled.sd(voice\_cl)

pooled.sd(voice\_cl2)

pooled.sd(voice\_cl3)

## F-score / F-measurement (F = 2/(1/recall +1/precision) or F = 2\*(precision\*recall)/(precision+recall))

precision <- c(0.9106, 0.8597, 0.7381, 0.7154, 0.9710, 0.9708, 0.7174, 0.7146, 0.93, 0.92, 0.6, 0.6, 0.86)

recall <- c(0.9825,0.9735, 0.7294, 0.8058, 0.9853, 0.9764, 0.7618, 0.7588, 0.96, 0.96, 0.44, 0.66, 0.88)

precision2 <- c(0.98, 0.9683, 0.7237, 0.7692, 0.9845, 0.9755, 0.7370, 0.7337, 0.96, 0.95, 0.22, 0.76, 0.85)

recall2 <- c(0.8991, 0.8358, 0.7325, 0.6686, 0.9696, 0.9696, 0.6899, 0.6869, 0.93, 0.91, 0.23, 0.77, 0.84)

F\_scoreM <- c(1:13)

F\_scoreF <- c(1:13)

for(i in 1:13) {

F\_scoreM[i] <- round((2/((1/recall[i])+(1/precision[i]))), digits=3)

F\_scoreF[i] <- round((2/((1/recall2[i])+(1/precision2[i]))), digits=3)

}

F\_scoreM

F\_scoreF

##analysis of results,

t.test(resultslm1)

t.test(resultslm2)

t.test(resultslm3)

t.test(resultslm4)

t.test(roundedresultsdf)

t.test(roundedresultsdf2)

t.test(roundedresultsdf3)

t.test(roundedresultsdf4)

knn1 <- table(datatest[,'label'],knn\_CW)

t.test(knn1)

knn2 <- table(datatest[,'label'],knn\_CW2)

t.test(knn2)

knn3 <- table(datatest[,'label'],knn\_CW3)

t.test(knn3)

knn4 <- table(datatest[,'label'],knn\_CW4)

t.test(knn4)

hc1 <- table(voice\_cl$cluster, voice\_cl$cluster)

t.test(hc1)

hc2 <- table(voice\_cl2$cluster, voice\_cl2$cluster)

t.test(hc2)

hc3 <- table(voice\_cl3$cluster, voice\_cl3$cluster)

t.test(hc3)