

Data 630 9040

Machine Learning 2215

Professor Bati Firdu

Melissa Hunfalvay

Date: 7-20-2021

Assignment 4

Introduction

Objective

The dataset used for this project was the diabetes data from Pima Indian females (Murphy & Aha, 1994; <https://www.kaggle.com/kumargh/pimaindiandibetescsv>).

The objective of the analysis was to explore the factors that impact a positive or negative diagnosis of diabetes. More specifically, using the Neural Networks (NN) method of supervised learning, the objective is to classify the diabetes diagnosis using various input variables to include health, demographics, and family history. Understanding what factors affect the Pima Indian female's high incidence rate of diabetes may provide some insight into strategies for mitigation.

Supervised learning refers to giving the dataset a predetermined, already known “right answer” or outcome (Ng, 2021). This dataset contained women, in which, for every woman there was a known outcome, either positive or negative for diabetes. Diabetes was the dependent variable, and the input variables of the women's health and family history are the independent variables in the dataset. Some examples of the independent variables include, number of pregnancies, Body Mass Index (BMI) and age.

Problem Domain

The Pima (pronounced pi:me) are a group of Native Americans who live in Arizona (https://en.wikipedia.org/wiki/Pima_people). In 1965 to 1989, rates of cardiovascular disease were recorded in the Pima, Sioux, Navajo, and several other Oklahoma tribes (Hoehner, Williams, Sievers et al., 2006). Results showed that ischemic heart disease became the leading cause of death in Pima Indians in those with diabetes (Sievers et al, 1996).

The extent to which diabetes was driving the increased deaths from heart disease in the Pima community was not well understood. As the years progressed, there continued to be an increase in the average duration of diabetes and the use of dialysis (Pavkov, Sievers, Knowler, Bennett & Nelson, 2004). This increase continued even when trends in from heart disease and diabetes and death were declining in other portions of the United States population, in particular the Caucasian populations (Cooper, Stamler, Dyer & Garside, 1978). In a study comparing Pima Indians to Caucasians, Pima Indians had a 19-fold greater incidence of diabetes (Knowler, Bennett, Hamman, Miller, 1978).

To try and understand the factors contributing to the continued rise of heart disease and diabetes in the Pima community, each member who was five years or older, participated in a research examination every 2 year. The research examinations for diabetes were conducted by the National Institute of Diabetes and Digestive and Kidney Disease.

The data collected in the research examinations were broad in scope to examine which factors may contribute to the increased incidence of diabetes. Therefore, the data collected included health factors, known to impact diabetes, and those less well understood. For instance, Body Mass Index is a well-known contributor to diabetes (Narayan, Boyle, Thompson, Gregg, & Williamson, 2007). Factors less well understood includes factors such as number of pregnancies and genetics; recorded in the Diabetes Pedigree Function (<https://machinelearningmastery.com/case-study-predicting-the-onset-of-diabetes-within-five-years-part-1-of-3/>).

Diabetes is defined according to the World Health Organization (WHO Study Group, 1985) as “*chronic elevation of the concentration of glucose in the blood...sometimes accompanied by severe thirst, profuse urination....deficient production or action of insulin, a*

hormone that controls glucose, fat...” Clinical diagnostic criteria for diabetes is via an oral glucose tolerance test only after fasting and at 2 hours after a 75 gram oral glucose load. Figure 1 shows the related values for diabetes diagnosis. This definition and diagnostic criteria was used to classify the Pima women as diabetic or not, along with a review of clinical records in the course of routine clinical care (Hoehner et al, 2006).

	Glucose concentration, mmol/litre (mg/dl)				
	Whole blood		Plasma		
	Venous	Capillary	Venous	Capillary	
<i>Diabetes mellitus</i>					
Fasting value	≥ 6.7 (≥ 120)	≥ 6.7 (≥ 120)	≥ 7.8 (≥ 140)	≥ 7.8 (≥ 140)	mmol/lit
2 hrs after glucose load ^b	≥ 10.0 (≥ 180)	≥ 11.1 (≥ 200)	≥ 11.1 (≥ 200)	≥ 12.2 (≥ 200)	
<i>Impaired glucose tolerance</i>					
Fasting value	< 6.7 (< 120)	< 6.7 (< 120)	< 7.8 (< 140)	< 7.8 (< 140)	Mg/dl
2 hrs after glucose load ^b	6.7–10.0 (120–180)	7.8–11.1 (140–200)	7.8–11.1 (140–200)	8.9–12.2 (160–220)	

^aSee footnote to Fig. 1.

^bFor epidemiological or population screening purposes the 2-hour value after 75 g oral glucose may be used alone or with the fasting value. The fasting value alone is considered less reliable since true fasting cannot be assured and spurious diagnosis of diabetes may more readily occur.

Table 1: World Health Organization Meeting, 1985; Diabetes diagnostic criteria

Therefore, the purpose of this analysis is to examine which health, demographic and family history health variables contribute with the highest degree of statistical weight to a positive or negative diabetes diagnosis for the women in the Pima community.

Method Rationale

The main methodology chosen was a Neural Network (NN), which is a statistical classification techniques and form of supervised learning.

The rationale for *supervised learning* methodology includes:

- a) A known outcome of diabetes (positive or negative)
- b) Variables within the dataset that may be used to evaluate against the known outcomes

The rationale for using NN includes:

- a) The research question is quantitative in nature requiring a tool for quantitative modeling
- b) The target/dependent variable (diabetes diagnosis) that forms a classification problem
- c) The specific NN used is a neuralnet because the dependent variable has only two classes, rather than nnet which is easier to use when the dependent variable has more than two classes.
- d) The research question is looking to identify the relationship among a set of variables or patterns in the data (Maimon & Rokach, 2010).
- e) The independent variables are numeric, rather than factor variables. Age, measured in years, is an example of numeric (interval) data within the dataset. NeuralNet data does not handle factor variables and they would need to be changed to dummy variables for each value.
- f) The dependent variable, diabetes is a factor variable with two levels that can be changed to a numeric variable with zero and one values.
- g) The nature of the problem is to inform the Pima community and health care professional of factors that may be influencing an abnormally high incident level of diabetes within the community. Therefore, a classification model with precision will help inform the community.

The Pima community; the National Institute of Diabetes and Digestive and Kidney Disease; and health care professionals serving the Pima community all want to ask questions of a dataset like this to include:

1. What are the primary contributing factors for a positive diabetes diagnosis in Pima women?

2. What are the primary factors contributing to a negative diabetes diagnosis in Pima women?
3. By understanding these factors, what mitigation strategies would be most beneficial to reduce the incidence of diabetes in Pima women and the Pima community as a whole? Mitigating strategies may take the form of:
 - a. Education for example, dietary changes or choices, symptom awareness, and/or exercise
 - b. Awareness of a possible increase in risk via genetic counseling
 - c. Regular doctor's visits
 - d. Medication

Analysis

Data

This data set was collected by the National Institute of Diabetes and Digestive and Kidney Disease on Pima females only. All women were over the age of 21 years and were examined at the Indian Health Service Hospital in Phoenix, Arizona. The goal of the data was to examine the health, demographic and family history factors that predict a positive or negative diagnosis of diabetes and by doing so, help to inform the community and provide mitigating strategies to reduce incidence.

The number of observations in this dataset was 768. Each observation was unique. There were nine variables in the data. The dependent variable chosen was class which had two levels: tested positive or tested negative.

Health related variables included the number of times the woman fell *pregnant* (preg).

Pregnancies were represented in the dataset as integers (whole number) variable.

A second health related variable included the *plasma glucose concentration* (plas) which is a measure of blood sugar/glucose levels in the blood measured in milligrams per deciliter (mg/dL). In the Pima dataset glucose was measured 2 hours after glucose administration of 75 grams of carbohydrates, using an oral glucose tolerance test. The diagnosis of diabetes was made with this test and in the course of the routine medical exam (Hoehner et al, 2006).

Other health related variable was *diastolic blood pressure* (pres) measured in a millimeter of mercury (mm Hg). Diastolic blood pressure is considered normal when below 80 mmHg. High blood pressure (stage 1) is characterized as diastolic blood pressure between 80-89 mm Hg. High blood pressure (stage 2) is characterized as diastolic blood pressure between 90-119 mm Hg. A hypertensive crisis is characterized as diastolic blood pressure above 120 mm Hg and requires immediate consultation with a medical professional (<https://www.healthline.com>). High blood pressure is correlated with both heart attack and diabetes both of which are known higher health incidence in the Pima population (<https://www.heart.org>).

A *triceps skin fold thickness* measured in millimeters (mm; skin) also contributed to the health variables in the dataset. A measurement of the subcutaneous skinfolds in the triceps is a widely used index of body fat (Ruiz, Colley & Hamilton, 1978). Normative values exist, and values for females are higher than for males (Moffatt, Sady, Owen, 1980). Furthermore, age significantly impacts results. Figure 2 is the normative values across various demographics when measuring skinfold thickness. Special attention should be paid to the women's group as they are most aligned with the group in the Pima community dataset.

Sex and Age (years)	N	Triceps		
		Mean	SD	Median
Male		(mm)		(mm)
18-24	60	13.9	7.07	12.0
25-34	132	14.8	7.61	12.2
35-44	114	15.5	6.57	14.8
45-54	90	15.6	5.24	15.0
55-64	66	15.4	5.63	14.0
65+	69	15.3	6.26	14.3
All ages	531	15.1	6.55	14.2
Female				
18-24	69	19.4	6.64	19.8
25-34	133	25.5	8.29	23.5
35-44	101	26.3	8.97	24.8
45-54	99	27.6	7.54	26.2
55-64	94	29.7	7.95	30.0
65+	60	27.2	7.18	26.5
All ages	556	26.1	8.41	24.8

Figure 2: Comparison of Sample Sizes, Means (in millimeters), Standard Deviation, and Median values for Triceps Measurement of Skinfolds in U.S. Men and Women. Taken from Moffatt, Sady, Owen (1980).

Insulin (insu) values also contributed to the health factors in the dataset. This was a 2-hour serum insulin test with measurement units as milli-international units per liter (mu U/ml).

Normal insulin levels after 2 hours of fasting are 16-166 mu U/ml

(<https://www.medicinenet.com>)

Body Mass Index (BMI; mass) is a health factor in this dataset. High levels of body fat are known to correlate with diabetes (Bays, Chapman & Grandy, 2007). For women, a BMI from 18.5 to 24.9 is considered normal. A BMI of 25 to 29.9 is considered overweight, 30-39.9 is considered obese, equal to or over 40 is considered morbidly obese (<https://medlineplus.gov/>).

Diabetes Pedigree Function (DPF; pedi) is a family history variable in the dataset. The DPF score the likelihood of diabetes based on family history and the genetic relationship to relatives of the patient who have diabetes (<https://github.com/susanli2016/Machine-Learning-with-Python/issues/26>). The measure gives an idea of hereditary risk associated with the onset

of diabetes. There are no normative values for this variable, nor a scale for interpretation of risk.

The only demographic variable in the dataset is age. Age is measured in years and is therefore an integer variable.

Exploratory Analysis

There are 9 variables in the data set, as shown by the str function (Figure 3). All variables were initially categorized by R as either numeric (num), integers (int) or factors. There are 768 rows (or observations) of data. Each row represents one unique woman.

```
.  
# Section 2: Data Exploration  
.  
# Check data types  
str(pima_diabetes) # note int, num, and factors, no Identifier  
data.frame': 768 obs. of 9 variables:  
 $ preg : int 6 1 8 1 0 5 3 10 2 8 ...  
 $ plas : int 148 85 183 89 137 116 78 115 197 125 ...  
 $ pres : int 72 66 64 66 40 74 50 0 70 96 ...  
 $ skin : int 35 29 0 23 35 0 32 0 45 0 ...  
 $ insu : int 0 0 0 94 168 0 88 0 543 0 ...  
 $ mass : num 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...  
 $ pedi : num 0.627 0.351 0.672 0.167 2.288 ...  
 $ age : int 50 31 32 21 33 30 26 29 53 54 ...  
 $ class: Factor w/ 2 levels "tested_negative",...: 2 1 2 1 2 1 2 1 2 2 ...
```

Figure 3: str function output without any pre-processing

An initial review of the data revealed no variables needed to be removed.

Missing data values were checked. None were found in this data set using this command (Figure 4).

```
# Check if the data has missing values  
colSums(is.na(pima_diabetes))  
preg plas pres skin insu mass pedi age class  
0 0 0 0 0 0 0 0 0
```

Figure 4: apply command to check for the missing values within each variable in the data set

To further explore the data, the summary command was run for all variables (Figure 5). The summary command displays descriptive statistics that assist in determining distribution of variables and implied skewness. This command is used as a first step in further decision making and understanding of the dataset. It was observed when using the string command that many of the variables had minimums of zero. This raised a red flag as to the legitimacy of those values.

```
> # Explore the descriptive nature of all variables
> summary(pima_diabetes) #check descriptive statistics
```

preg		plas		pres		skin		insu		mass		pedi		age	
Min.	: 0.000	Min.	: 0.0	Min.	: 0.00	Min.	: 0.00	Min.	: 0.0	Min.	: 0.00	Min.	:0.0780	Min.	:21.00
1st Qu.:	1.000	1st Qu.:	99.0	1st Qu.:	62.00	1st Qu.:	0.00	1st Qu.:	0.0	1st Qu.:	27.30	1st Qu.:	0.2437	1st Qu.:	24.00
Median :	3.000	Median :	117.0	Median :	72.00	Median :	23.00	Median :	30.5	Median :	32.00	Median :	0.3725	Median :	29.00
Mean :	3.845	Mean :	120.9	Mean :	69.11	Mean :	20.54	Mean :	79.8	Mean :	31.99	Mean :	0.4719	Mean :	33.24
3rd Qu.:	6.000	3rd Qu.:	140.2	3rd Qu.:	80.00	3rd Qu.:	32.00	3rd Qu.:	127.2	3rd Qu.:	36.60	3rd Qu.:	0.6262	3rd Qu.:	41.00
Max.	:17.000	Max.	:199.0	Max.	:122.00	Max.	:99.00	Max.	:846.0	Max.	:67.10	Max.	:2.4200	Max.	:81.00

```

class
tested_negative:500
tested_positive:268

```

Figure 5: Summary Command showing descriptive statistics for all variables in the data set.

The head command was used to view the raw values for the first 6 rows of each variable (Figure 6). When viewing the summary command and/or the head command unusual data inputs were noticed. For example, BMI (mass) below 16 is considered anorexia nervosa (<https://www.fibre2fashion.com/>) yet there are zero values in this variable. These unusual values were noted and further explored in more depth with each individual variable.

```
> head(pima_diabetes)# show first 6 rows
```

	preg	plas	pres	skin	insu	mass	pedi	age	class
1	6	148	72	35	0	33.6	0.627	50	tested_positive
2	1	85	66	29	0	26.6	0.351	31	tested_negative
3	8	183	64	0	0	23.3	0.672	32	tested_positive
4	1	89	66	23	94	28.1	0.167	21	tested_negative
5	0	137	40	35	168	43.1	2.288	33	tested_positive
6	5	116	74	0	0	25.6	0.201	30	tested_negative

Figure 6: Head command showing the first 6 rows of data

For each integer variable the following commands were examined to explore the specific variable. The head command which showed the first 100 rows of data, the summary command showing descriptive statistics including minimum, maximum (by default range), median, and mean as well as 1st and third quartiles. Also examined was the standard deviation, and mode.

Visualizations for each variable included histograms and boxplots. For the factor variable a barplot was chosen as the visualization. When needed to further understand a variable, percentages and counts per unique value within the variable was viewed.

Exploratory Analysis: Number of Pregnancies (preg). The number of pregnancies ranged from zero to seventeen. Zero pregnancies are a legitimate value. However, pregnancies of 10 or more seem very high, especially if the woman is young. This was noted for a “data sanity” check in the preprocessing stage of the analysis.

Exploratory Analysis: Plasma glucose concentration (plas). Figure 6 shows the count associated with each unique value in the plasma variable and the summary command. Glucose (plas) levels below 70 are hypoglycemic, below 54 mg/dL is cause for immediate action by a doctor (<https://medlineplus.gov>). Five zero values are seen in the head command for the glucose variable. One further count is seen at 44 mg/dL. This is noted for removal of all rows of data below 54 mg/dL in the preprocessing section.

```
> table(pima_diabetes$plas) # Count the number of values within each category
 0  44  56  57  61  62  65  67  68  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98
 5   1   1   2   1   1   1   1   3   4   1   3   4   2   2   2   4   3   6   6   3   6  10   7   3   7   9   6  11   9   9   7   7  13   8   9   3
99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
17 17   9  13   9   6  13  14  11  13  12   6  14  13   5  11  10   7  11   6  11  11   6  12   9  11  14   9   5  11  14   7   5   5   5   6   4
136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172
 8   8   5   8   5   5   6   7   5   9   7   4   1   3   6   4   2   6   5   3   2   8   2   1   3   6   3   3   4   3   3   4   1   2   3   1
173 174 175 176 177 178 179 180 181 182 183 184 186 187 188 189 190 191 193 194 195 196 197 198 199
 6   2   2   2   1   1   5   5   5   1   3   3   1   4   2   4   1   1   2   3   2   3   4   1   1

> summary(pima_diabetes$plas) # Descriptive Statistics
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    99.0   117.0   120.9   140.2   199.0
```

Figure 6: Count per unique value and summary statistics for plasma glucose concentration (plas)

Exploratory Analysis: Diastolic blood pressure (pres): Diastolic blood pressure is considered normal when below 80 mmHg. Hypotension is when the diastolic blood pressure is below 60 mmHg. A hypertensive crisis is characterized as diastolic blood pressure above 120 mm Hg and requires immediate consultation with a medical professional (<https://www.healthline.com/>). When viewing the summary data (Figure 7), boxplot (Figure 8) and histogram (Figure 9), there is one woman whose diastolic blood pressure value is above 120

mmHg. There are 434 women with below 60 mmHg diastolic blood pressure (Figure10)! Thirty-five of these women have a diastolic blood pressure of zero. Although it is not impossible to have lower than 60 mmHg (hypotensive), especially for diabetics, it is impossible to have a zero value. Therefore, in preprocessing of this variable the entire data row is removed if the values are zero and the one value of above 120 mmHg.

```
> summary(pima_diabetes$pres) # Descriptive Statistics
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.00   62.00   72.00   69.11   80.00  122.00
```

Figure 7: Summary statistics for diastolic blood pressure

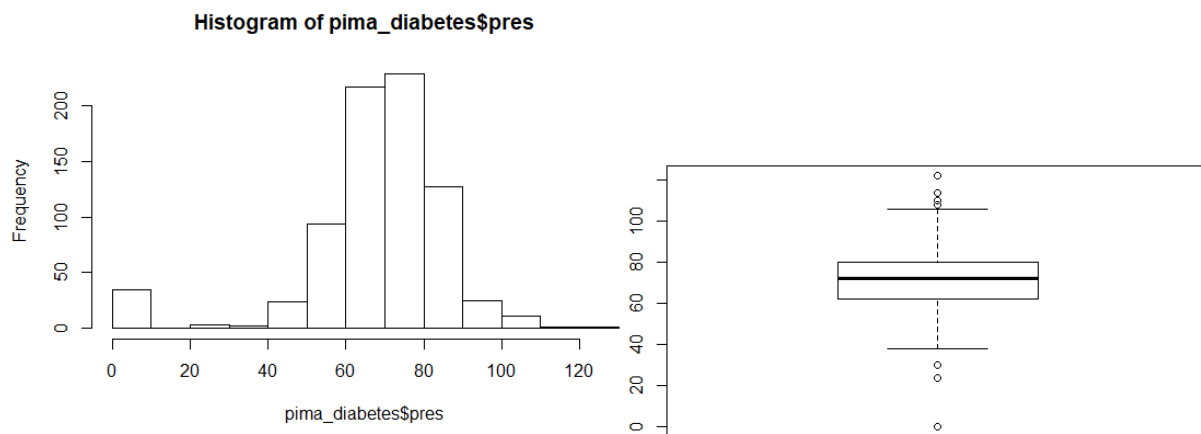


Figure 8: Histogram; Figure 9: Box plot for diastolic blood pressure.

```
> sum(pima_diabetes$insu<60) # how many records less than 60?
[1] 434
```

Figure 10: The number of observations below 60 mmHg indicating hypotension

Exploratory Analysis: Triceps Skin fold Thickness (skin): When exploring this variable (Figure 11), it is observed that there are almost 30% of zero values! Normative values range from 19.4-26.1% for women (Moffatt, Sady, Owen, 1980). This is a highly unlikely finding and like a zero Body Mass Index would indicate severe anorexia nervosa which is not a condition that correlates with diabetes. Hence, the assumption is that the researchers input a value of zero if they did not have any data.

```

> summary(pima_diabetes$skin) # Descriptive Statistics
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.00   0.00   23.00   20.54   32.00   99.00
> table(pima_diabetes$skin)/length(pima_diabetes$skin) # Percentage
 0          7          8          10          11          12          13          14          15          16          17          18
0.295572917 0.002604167 0.002604167 0.006510417 0.007812500 0.009114583 0.014322917 0.007812500 0.018229167 0.007812500 0.018229167 0.026041667
19          20          21          22          23          24          25          26          27          28          29          30
0.023437500 0.016927083 0.013020833 0.020833333 0.028645833 0.015625000 0.020833333 0.020833333 0.029947917 0.026041667 0.022135417 0.035156250
31          32          33          34          35          36          37          38          39          40          41          42
0.024739583 0.040364583 0.026041667 0.010416667 0.019531250 0.018229167 0.020833333 0.009114583 0.023437500 0.020833333 0.019531250 0.014322917
43          44          45          46          47          48          49          50          51          52          54          56
0.007812500 0.006510417 0.007812500 0.010416667 0.005208333 0.005208333 0.003906250 0.003906250 0.001302083 0.002604167 0.002604167 0.001302083
60          63          99
0.001302083 0.001302083 0.001302083
> sd(pima_diabetes$skin) # Standard deviation
[1] 15.95222
> names(sort(-table(Data_skin$skin)))[1] # Mode
[1] "0"

```

Figure 11: Descriptive Statistics, unique counts per value, standard deviation, and mode values for Triceps Skin fold Thickness (skin)

The problem with this level of missing data is that to remove the 30% of missing the observations would render a Neural Network analysis very difficult as NN requires a large amount of data (Hastie, Tibshirani, Friedman, 2007). Replacing the missing variables with a mean value for this large a percentage of missing data would significantly impact the “true” value of the variable.

Skin fold is a measure of subcutaneous fat measured at the triceps location. The dataset has another measure of fat in Body Mass Index, which is a measure of weight over height. Therefore, if the skin variable is removed there is still an indication of fat levels in the dataset. A decision was therefore made to remove the skin data due to the significant loss from within the variable and as there is a similar measure in BMI.

Exploratory Analysis: Two-hour Serum Insulin (mu U/ml): Normal levels of insulin after two-hours of fasting are 16-166 mu U/ml (<https://www.medicinenet.com/>). Type 1 diabetics do not make enough insulin, this insulin leads to hyperglycemia or diabetic ketoacidosis (<https://www.mayoclinic.org/>). Therefore, in Type 1 diabetes the insulin levels are low.

Figure 12 shows the minimum and maximum values and percentages of each value within the insulin variable. It can be observed that almost 49% of the data for insulin is zero!

```

> summary(pima_diabetes$insu) # Descriptive Statistics
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.0    0.0   30.5   79.8   127.2   846.0
> table(pima_diabetes$insu)/length(pima_diabetes$insu) # Percentage

```

0	14	15	16	18
0.486979167	0.001302083	0.001302083	0.001302083	0.002604167

Figure 12: Descriptive statistics and Percentages of values within the Insulin Variable

Type 2 diabetics have too much insulin in the blood as their body is less responsive to ridding the blood of insulin. Therefore, in type 2 diabetes the insulin levels are high.

It is unclear if the positive diagnosis of diabetes in the Pima data set is due to Type 1 or Type 2 diabetes. However, having low insulin levels of zero would cause immediate hospitalization due to diabetic ketoacidosis. Therefore, it is assumed that the zero values of insulin are, in fact, researcher's way of saying "no data available." This leaves two possible choices in pre-processing:

1. Keep the variable knowing there are incorrect values Make this decision because the Neural Network needs as much data as possible.
2. Remove the variable, because half the values are incorrect, even though the Neural Network may suffer.

The choice is to remove the variable. The priority should be accurate data, even at cost of the analytical methodology. The methodology should match the data, not the other way around.

Exploratory Analysis: Body Mass Index (BMI): BMI is a measure of body fat based on height and weight. For women, a BMI from 18.5 to 24.9 is considered normal. The mode for BMI in the data set is 32. There are 11 observations with a BMI of zero (see Figure 13). This is likely incorrect and would indicate severe anorexia nervosa as previously stated. Therefore, given the low number, these observations should be removed during the pre-processing phase.

The data is normally distributed (see Figure 14). Box plot indicates outliers (Figure 15) below the normal range, these are the zero values. The next value above zero is 18.2 (Figure 13). Therefore, only the zero values should be removed.

The boxplot also shows values above the 75th percentile. Although very high BMI's indicating obesity, they are legitimate, especially for Type 2 diabetics. A BMI of 25 to 29.9 is considered overweight, 30-39.9 is considered obese, equal to or over 40 is considered morbidly obese (<https://medlineplus.gov/ency/article/003101.htm>). Therefore, these observations are to be kept within the dataset.

```
> table(pima_diabetes$mass) # Count the number of values within each category
```

0	18.2	18.4	19.1	19.3	19.4	19.5	19.6	19.9	20	20.1	20.4	20.8	21	21.1	21.4
11	3	1	1	1	1	2	3	1	1	1	2	2	2	4	
23.3	23.4	23.5	23.6	23.7	23.8	23.9	24	24.1	24.2	24.3	24.4	24.5	24.6	24.7	24.8
2	1	3	3	2	2	2	4	1	6	4	3	1	4	5	
26.4	26.5	26.6	26.7	26.8	26.9	27	27.1	27.2	27.3	27.4	27.5	27.6	27.7	27.8	27.9
3	3	4	1	4	1	2	3	2	4	5	5	7	4	7	
29.6	29.7	29.8	29.9	30	30.1	30.2	30.3	30.4	30.5	30.7	30.8	30.9	31	31.1	31.4
4	8	3	5	7	9	1	1	7	7	1	9	5	2	1	1
33.2	33.3	33.5	33.6	33.7	33.8	33.9	34	34.1	34.2	34.3	34.4	34.5	34.6	34.7	34.8
7	10	1	8	5	5	2	6	4	8	6	4	5	5	4	
36.3	36.4	36.5	36.6	36.7	36.8	36.9	37	37.1	37.2	37.3	37.4	37.5	37.6	37.7	37.8
3	2	4	5	1	6	3	1	2	4	1	3	2	5	5	
39.3	39.4	39.5	39.6	39.7	39.8	39.9	40	40.1	40.2	40.5	40.6	40.7	40.8	40.9	41
1	7	3	1	1	2	3	2	1	1	3	4	1	1	2	
43.2	43.3	43.4	43.5	43.6	44	44.1	44.2	44.5	44.6	45	45.2	45.3	45.4	45.5	45.6
1	5	2	2	2	2	1	2	2	1	1	1	3	1	1	
50	52.3	52.9	53.2	55	57.3	59.4	67.1								
1	2	1	1	1	1	1	1								

```
> summary (pima_diabetes$mass) # Descriptive Statistics
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	27.30	32.00	31.99	36.60	67.10

```
> table(pima_diabetes$mass)/length(pima_diabetes$mass) # Percentage
```

0	18.2	18.4	19.1	19.3	19.4
0.014322917	0.003906250	0.001302083	0.001302083	0.001302083	0.001302083

Figure 13: Counts of unique values, summary command and percentages within the BMI variable.

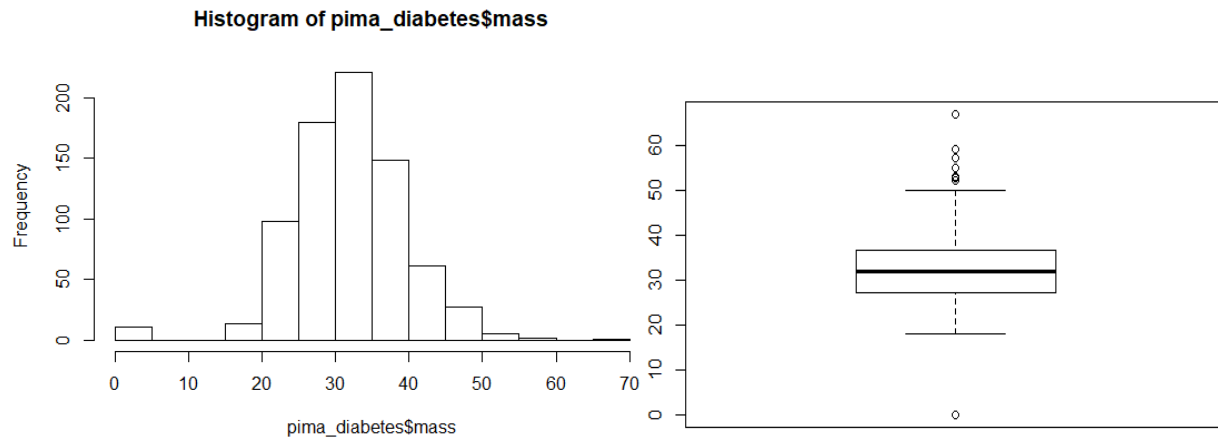


Figure 14: Histogram showing normal distribution; Figure 15: Boxplot showing outliers for the BMI variable.

Exploratory Analysis: Diabetes Pedigree Function (DPF): DPF variable shows a minimum score of 0.078 and a maximum score of 2.420. Standard deviation is 0.331 and mode is 0.254. There are no zero values in this variable. After considerable research trying to find the scale and understand the inputs, there is only very general descriptions of the scale stating that the scale collecting information on “diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient.” Therefore, without any further information, it would be pre-emptive to remove, or modify any observations. Therefore, there is no additional pre-processing needed for this variable.

Exploratory Analysis: Age: Women selected for this data set were at least 21 years of age of which there were 63 women. The oldest woman was 81 ($n = 1$). Most women were 33 (mean = 33.24, mode = 33). Data is right skewed (Figure 16). No pre-processing is needed for the age variable.

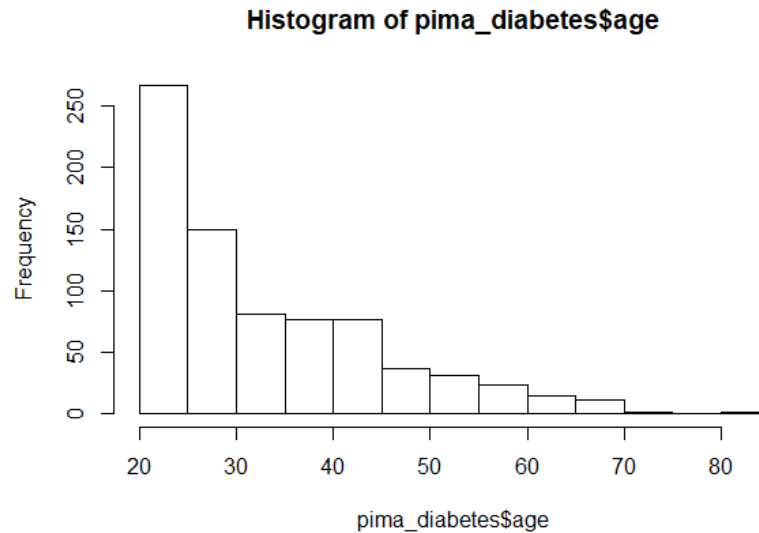


Figure 16: Histogram of Age for Pima Women

Exploratory Analysis: Class/Diagnosis: Class is the dependent variable in the dataset.

This is a factor variable with two levels: `tested_negative` and `tested_positive`. There are 500 non-diabetics in the data set (65%; `tested_negative`) and 268 diabetics (35%; `tested_positive`; Figure 17). As this is a factor variable and Neural Networks (NN) does not handle factor variables it will need to be transformed to a dummy variable in pre-processing.

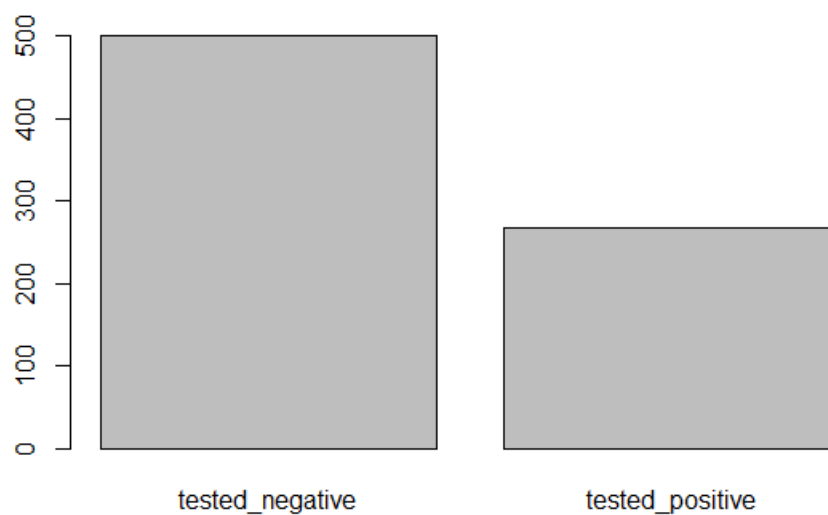


Figure 17: Bar chart of Diabetes Negative and Positive Test Results

Preprocessing

Armed with the objective and exploratory analysis, as well as the model type preprocessing was conducted. As the data was quite “messy” including missing values, outliers and transformations pre-processing was relatively extensive.

Preprocessing: Pregnancy (preg): pregnancies of 10 or more seem very high, especially if the woman is young. As a “data sanity” check women with more than 10 pregnancies were examined by age (Figure 18). The youngest women with more than 10 pregnancies were 35 (n = 2). It is possible for women 35 and over to have more than 10 pregnancies therefore the variable was not amended in any way during preprocessing.

```
> # Pregnancy variable
> # Check women with greater than 10 pregnancies against age.
> summary(pima_diabetes$preg) # Descriptive Statistics - check range
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000  1.000   3.000   3.845   6.000  17.000
> boxplot(pima_diabetes$preg) # Check outliers
> max(pima_diabetes$preg)
[1] 17
> pima_diabetes[which(pima_diabetes$preg>10),"age"] # legitimate, no one is 20's or early 30's
[1] 51 57 35 42 45 43 47 40 38 44 51 52 46 43 44 44 48 58 41 38 46 41 42 35 62 45 50 38 42 51 44 48 39 46
> |
```

Figure 18: Data Sanity Code Check for Number of Pregnancies above 10 Against Age

Preprocessing: Plasma (plas): Five zero values and one value of 44 mg/dL were found in data exploration. As previously explained, the rows of data below 54 mg/dL needed to be removed. Figure 19 shows removal of the 5 values of zero and the 1 value of 44 mg/dL.

```

> # Plasma/Glucose variable
> # Is it legitimate to have a zero plasma/glucose level? what is a reasonable lowest number? below 54 mg/dL Ref: https://medlineplus.gov/ency/patientinstructions/000085.htm
> summary(pima_diabetes$plas) # Descriptive Statistics - check range
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.0    99.0   117.0   120.9   140.2   199.0
> boxplot(pima_diabetes$plas) # Check outliers
> table(pima_diabetes$plas) # Unique counts before

 0  44  56  57  61  62  65  67  68  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98
 5   1   1   2   1   1   1   1   3   4   1   3   4   2   2   2   4   3   6   6   3   6  10   7   3   7   9   6  11   9   9   7   7  13   8   9   3
99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
17 17   9  13   9   6  13  14  11  13  12   6  14  13   5  11  10   7  11   6  11  11   6  12   9  11  14   9   5  11  14   7   5   5   6   4
136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172
 8   8   5   8   5   5   5   6   7   5   9   7   4   1   3   6   4   2   6   5   3   2   8   2   1   3   6   3   3   4   3   3   4   1   2   3   1
173 174 175 176 177 178 179 180 181 182 183 184 186 187 188 189 190 191 193 194 195 196 197 198 199
 6   2   2   2   1   1   5   5   5   1   3   3   1   4   2   4   1   1   2   3   2   3   4   1   1

> pima_diabetes = pima_diabetes[which(pima_diabetes$plas > 50),] # remove the patients with less than 54 mg/dL
> summary(pima_diabetes$plas) # Descriptive Statistics - check range
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
56.00   99.25  117.00  121.79  141.00  199.00
> boxplot(pima_diabetes$plas) #after
> table(pima_diabetes$plas) # Unique counts after

56  57  61  62  65  67  68  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
 1   2   1   1   1   1   3   4   1   3   4   2   2   2   4   3   6   6   3   6  10   7   3   7   9   6  11   9   9   7   7  13   8   9   3  17  17
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137
 9  13   9   6  13  14  11  13  12   6  14  13   5  11  10   7  11   6  11  11   6  12   9  11  14   9   5  11  14   7   5   5   6   4   8   8
138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174
 5   8   5   5   5   6   7   5   9   7   4   1   3   6   4   2   6   5   3   2   8   2   1   3   6   3   3   4   3   3   4   1   2   3   1   6   2
175 176 177 178 179 180 181 182 183 184 186 187 188 189 190 191 193 194 195 196 197 198 199
 2   2   1   1   5   5   5   1   3   3   1   4   2   4   1   1   2   3   2   3   4   1   1

```

Figure 19: Show the Removal of Illegitimate Plasma Values Below 54 mg/dL

Preprocessing: Diastolic Blood Pressure (pres): In the exploratory analysis it was determined that data rows where values are zero and the one value of above 120 mmHg needed to be removed. These were clipped (Figure 20)

```

> # Diastolic Blood Pressure: pres, type = int
> table(pima_diabetes$pres) # Count the number of values within each category

 0  24  30  38  40  44  46  48  50  52  54  55  56  58  60  61  62  64  65  66  68  70  72  74  75  76  78  80  82  84  85  86  88  90  92  94  95
35   1   2   1   1   4   2   4  13  11  11   2  12  21  37   1  33  43   7  30  43  57  44  51   8  39  45  39  30  23   6  21  25  22   8   6   1
96 98 100 102 104 106 108 110 114 122
 4   3   3   1   2   3   2   3   1   1

> summary(pima_diabetes$pres) # Descriptive Statistics
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.00   62.50   72.00   69.12   80.00  122.00
> sum(pima_diabetes$pres<1) # how many records less than 1?
[1] 35
> sum(pima_diabetes$pres>120) # how many records more than 120?
[1] 1
> pima_diabetes = pima_diabetes[which(pima_diabetes$pres > 0),] # remove the patients with zero
> pima_diabetes = pima_diabetes[which(pima_diabetes$pres < 121),] # remove the patients with zero
> summary(pima_diabetes$pres) # Descriptive Statistics
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
24.00   64.00   72.00   72.38   80.00  114.00
> sum(pima_diabetes$pres<1) # how many records less than 1?
[1] 0
> sum(pima_diabetes$pres>120) # how many records more than 120?
[1] 0

```

Figure 20: Show the Removal of Illegitimate Diastolic Blood Pressure Values <1 & >120 mmHg

Pre-Processing: Triceps Skin Fold Thickness: A decision was made, in the data exploration section, to remove the skin data due to the significant loss from within the variable and as there is a similar measure in BMI. This was done and can be seen in the string command after pre-processing (Figure 23).

Preprocessing: Two-hour Serum Insulin (μ U/ml): The choice is to remove the variable as explained in the data exploration section. The priority should be accurate data, even at cost of the analytical methodology. The methodology should match the data, not the other way around. This was done and can be seen in the string command after pre-processing (Figure 23).

Preprocessing: Body Mass Index (BMI): Removal of zero values was determined in the data exploration section. Figure 21 shows the code and verification of removal of all zero records within the BMI variable.

```
> # Mass (BMI) variable
> table(pima_diabetes$mass) # Count the number of values within each category
 0  18.2 18.4 19.1 19.3 19.4 19.5 19.6 19.9 20 20.1 20.4 20.8 21 21.1 21.2 21.7 21.8 21.9 22.1 22.2 22.3 22.4 22.5 22.6 22.7 22.9 23 23.1 23.2
4  3  1  1  1  1  2  2  1  1  1  2  2  2  3  1  1  5  3  2  2  1  1  3  2  1  2  2  2  4  3
23.3 23.4 23.5 23.6 23.7 23.8 23.9 24 24.1 24.2 24.3 24.4 24.5 24.6 24.7 24.8 24.9 25 25.1 25.2 25.3 25.4 25.5 25.6 25.8 25.9 26 26.1 26.2 26.3
2  1  2  3  1  2  2  4  1  6  4  3  1  4  4  3  1  5  4  3  5  2  4  2  6  2  7  4  3  4  1
26.4 26.5 26.6 26.7 26.8 26.9 27 27.1 27.2 27.3 27.4 27.5 27.6 27.7 27.8 27.9 28 28.1 28.2 28.3 28.4 28.5 28.6 28.7 28.8 28.9 29 29.2 29.3 29.5
3  3  4  1  4  1  2  3  2  4  5  4  7  3  7  2  4  1  2  2  5  3  2  7  2  5  5  1  5  5
29.6 29.7 29.8 29.9 30 30.1 30.2 30.3 30.4 30.5 30.7 30.8 30.9 31 31.1 31.2 31.3 31.6 31.9 32 32.1 32.2 32.3 32.4 32.5 32.6 32.7 32.8 32.9 33.1
4  8  2  5  4  9  1  1  7  7  1  9  5  2  1  12  1  12  2  12  1  1  2  9  6  1  3  9  8  3
33.2 33.3 33.5 33.6 33.7 33.8 33.9 34 34.1 34.2 34.3 34.4 34.5 34.6 34.7 34.8 34.9 35 35.1 35.2 35.3 35.4 35.5 35.6 35.7 35.8 35.9 36 36.1 36.2
7 10 1 8 5 4 2 6 4 8 6 4 5 5 4 2 6 4 3 2 3 4 7 2 4 5 5 2 3 1
36.3 36.4 36.5 36.6 36.7 36.8 36.9 37 37.1 37.2 37.3 37.4 37.5 37.6 37.7 37.8 37.9 38 38.1 38.2 38.3 38.4 38.5 38.6 38.7 38.8 38.9 39 39.1 39.2
2  4  5  1  6  3  1  2  4  1  3  2  5  5  3  2  2  3  4  1  2  5  1  3  1  1  3  4  2
39.3 39.4 39.5 39.6 39.7 39.8 39.9 40 40.1 40.2 40.5 40.6 40.7 40.8 40.9 41.2 41.3 41.5 41.8 42 42.1 42.2 42.3 42.4 42.6 42.7 42.8 42.9 43.1 43.3
1  7  3  1  1  2  2  2  1  1  3  4  1  1  2  1  3  2  1  1  2  1  2  1  3  2  1  4  1  4
43.4 43.5 43.6 44 44.1 44.2 44.5 44.6 45 45.2 45.3 45.4 45.5 45.6 45.7 45.8 46.1 46.2 46.3 46.5 46.7 46.8 47.9 48.3 48.8 49.3 49.6 49.7 50 52.3
2  2  2  2  1  1  2  1  1  1  3  1  1  2  1  1  2  2  1  1  1  2  2  1  1  1  1  1  1  1
52.9 53.2 55 57.3 59.4 67.1
1  1  1  1  1  1
> summary(pima_diabetes$mass) # Descriptive Statistics
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.00  27.43   32.35   32.31   36.60   67.10
> sum(pima_diabetes$mass<1) # how many records less than 1?
[1] 4
> pima_diabetes = pima_diabetes[which(pima_diabetes$mass >0),] # remove the patients with zero
> summary(pima_diabetes$mass) # Descriptive Statistics
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
18.20  27.50   32.40   32.49   36.60   67.10
> sum(pima_diabetes$mass<1) # how many records less than 1?
[1] 0
```

Figure 21: Removal of all zero records within the BMI variable

No pre-processing was required for the *Diabetes Pedigree Function* (pedi) or *age*.

Pre-Processing: Class/Diagnosis: Class is the dependent variable in the dataset. This is a factor variable with two levels: *tested_positive* and *tested_negative*. Neural Networks (NN) does not handle factor variables and therefore during pre-processing it was transformed to a dummy 'num' variable.

At the completion of pre-processing two final steps were taken. First, the variables were scaled so the weights can remain reasonably small and not explode or vanish and the model can

converge correctly (Figure 22) and second the string command was re-run to examine changes in the dataset (Figure 23).

	preg	plas	pres	mass	pedi	age	class
1	0.63336064	0.8476983491	-0.02826062	0.16107019	0.45706397	1.41303761	1
2	-0.85230012	-1.2084210508	-0.51796066	-0.85598219	-0.37276212	-0.20001473	0
3	1.22762495	1.9899869046	-0.68119400	-1.33544973	0.59236170	-0.11511724	1
4	-0.85230012	-1.0778737873	-0.51796066	-0.63804239	-0.92597951	-1.04898964	0
5	-1.14943227	0.4886933745	-2.63999415	1.54135555	5.45105357	-0.03021974	1
6	0.33622849	-0.1966797588	0.13497272	-1.00127538	-0.82375455	-0.28491222	0
7	-0.25803582	-1.4368787619	-1.82382742	-0.21669212	-0.68244359	-0.62450218	1
9	-0.55516797	2.4469023268	-0.19149397	-0.28933872	-0.95303905	1.66773009	1

Figure 22: Scaling of data after all other pre-processing steps are completed

In summary, after pre-processing was complete there remained 722 observations, 46 observations were removed. Seven variables remained; two variables were removed. The dependent variable, class, was transformed to a numeric variable for Neural Networks modeling.

```
> str(pima_diabetes)
'data.frame': 722 obs. of 7 variables:
 $ preg : num 0.633 -0.852 1.228 -0.852 -1.149 ...
 $ plas : num 0.848 -1.208 1.99 -1.078 0.489 ...
 $ pres : num -0.0283 -0.518 -0.6812 -0.518 -2.64 ...
 $ mass : num 0.161 -0.856 -1.335 -0.638 1.541 ...
 $ pedi : num 0.457 -0.373 0.592 -0.926 5.451 ...
 $ age : num 1.413 -0.2 -0.1151 -1.049 -0.0302 ...
 $ class: num 1 0 1 0 1 0 1 1 0 1 ...
```

Figure 23: String command showing changes in variables, observations, and data categories after pre-processing

Algorithm Intuition

Neural Networks (NN) are an important class of tools for quantitative modeling and data mining used to answer questions and provide insights such as pattern classification, series analysis, prediction, data association, filtering, conceptualization, and clustering (Zhang, 2008). NN, also called *Artificial NN*, grew out of research in artificial intelligence, in an attempt to

mimic the learning of biological neural networks of the human brain. Although, NN are a much simpler model of human brain processing, NN does share two characteristics that are thought to mimic biological networks. The first, is parallel processing of information and the second is learning and generalizing from experience (or in the case of NN from more and more data).

Basic Structure: The basic structure (Figure 24) of a NN involves input from the independent variables (x_1, x_2, x_3) multiplied by weights (w_1, w_2, w_3), run through an activation function and ultimately to an output (y).

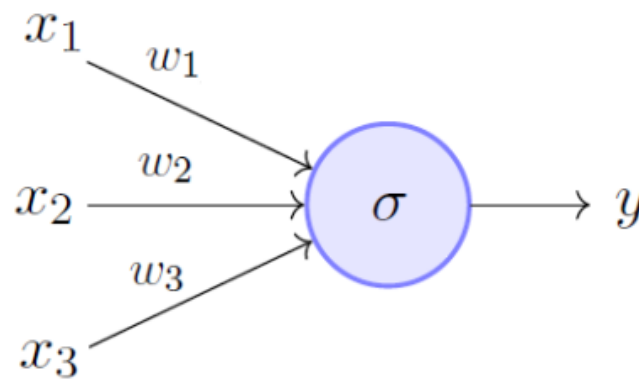


Figure 24: Basic structure of a Neural Network

Digging a little deeper into the process, the data (independent variables) will pass the sum of the input and weight through a non-linear function called an *activation function*. This will then create an output (Figure 25).

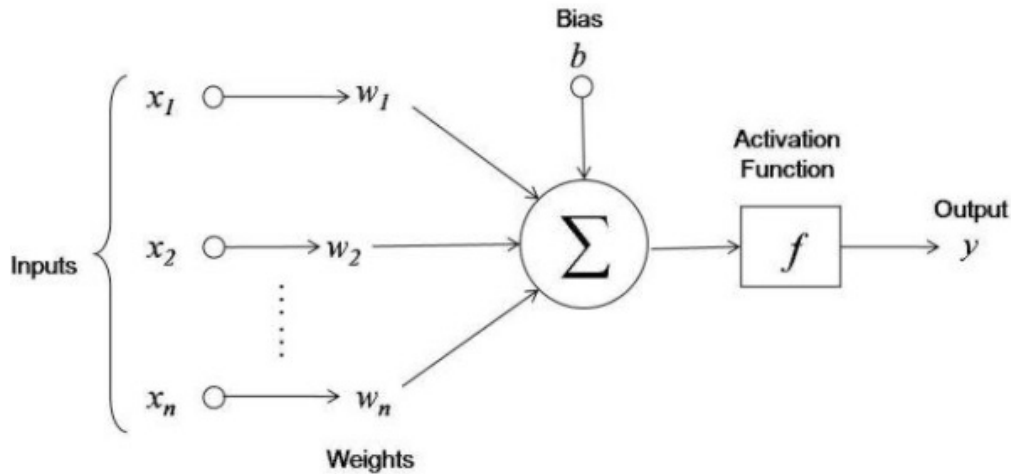


Figure 25: Basic Structure of a Neural Network with the Activation Function

There are various types of activation functions, here we will focus on the *sigmoid function* and how it differs from the *perceptron function*. The sigmoid neuron is preferable to the perceptron as it allows more than just dichotomous (0,1) variable inputs. By allowing inputs between these values, small changes can influence the model without it potentially giving vastly different outputs and in turn, this makes the model more useful as it is considered both more malleable and more stable.

The sigmoid function is a small, smooth, iterative, “S” shaped function, rather than the perceptron, which is a step function, or “jump”, caused by the dichotomous nature of variable input. The desired state is to create a small change in the model output when either the weights or biases are adjusted. One option for achieving this desired state is to use the sigmoid function. The difference in these functions is visualized in Figure 26 (Sigmoid function) and Figure 27 (Step function).

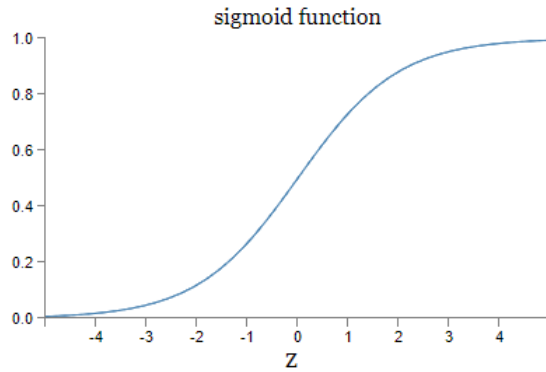


Figure 26: Sigmoid Function.

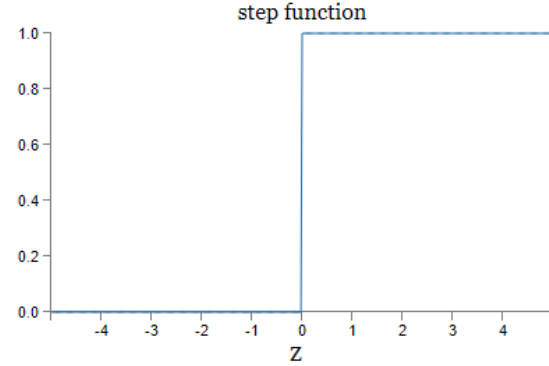


Figure 27: Step Function.

The sigmoid function (Figure 28) is mathematically explained by 1 divided by 1 plus e. E is the exponent variable.

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}}$$

Figure 28: Sigmoid function whereby e is the exponent variable. Z is the output.

Inputs from the independent variables ($x_1, x_2, x_3 \dots x_n$) are multiplied by the applied weights and then aggregated. Then biases are applied and transferred through the activation function, that is the sigmoid function (Figure 29).

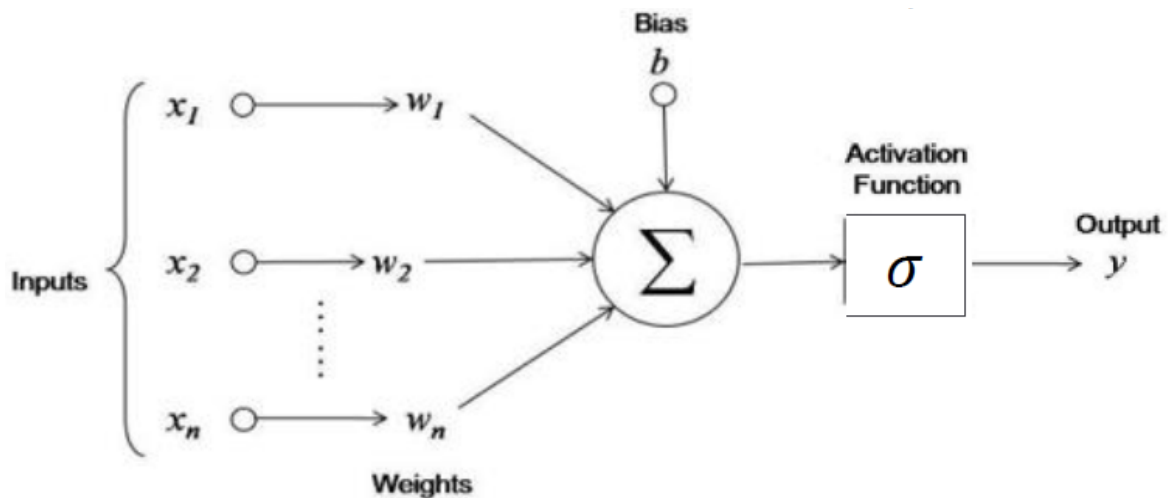
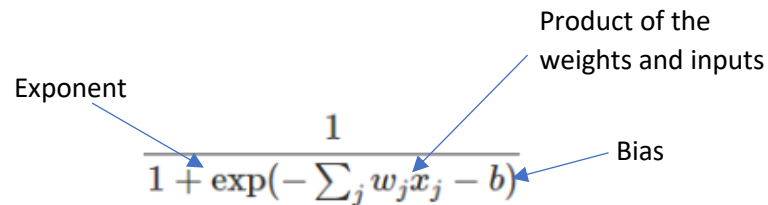


Figure 29: Visualization of the neural network using the sigmoid function

Figure 30 shows the formula for the entire model whereby, 1 divided by 1 exponent to the entire process minus the sum of the product of the weights and inputs to that neuron.



The diagram shows the sigmoid function formula: $\frac{1}{1 + \exp(-\sum_j w_j x_j - b)}$. Three blue arrows point to specific parts of the formula: one points to the '1' in the denominator's exponent, labeled 'Exponent'; another points to the summation term $-\sum_j w_j x_j$, labeled 'Product of the weights and inputs'; and a third points to the bias term $-b$, labeled 'Bias'.

Figure 30: Mathematically shows the process that is visualized using the sigmoid function in Figure 29.

The Sigmoid function allows application for predicting continuous values or for classification using a threshold. It will give similar results to the perception for model values when the values are in a moderate range. However, when the z- values are on the extremes (largely positive or largely negative) then the sigmoid function is advantageous as it does not create large global changes in the model.

Bias: Bias (b) is a measure of how easy it is to get the activation function to “fire” (Nelson, 2019). Bias is like the intercept in a linear equation as it is an additional parameter used to adjust the output along with the weights and inputs to the neuron. Bias is a constant that can help “fit” the model (<https://www.geeksforgeeks.org/effect-of-bias-in-neural-network/>). Bias helps to control the value at which the activation function will trigger. This can be seen in Figure 31 as the change in bias increases the value of triggering the activation function.

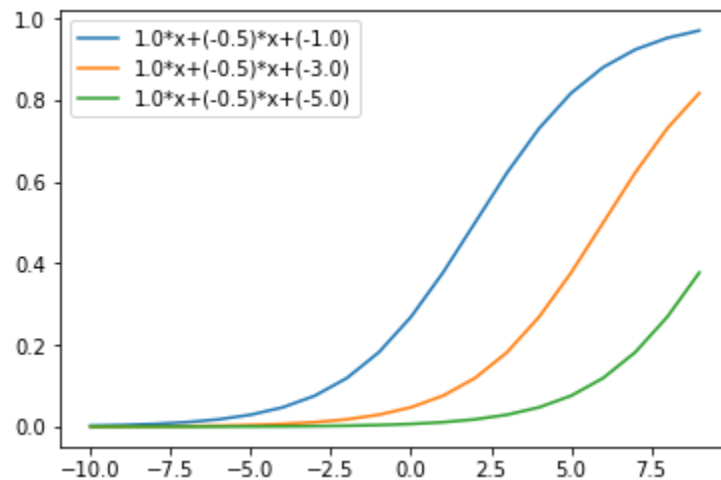


Figure 31: Shows bias ranging from -1.0 to -5.0 and the affect on triggering the activation function (<https://www.geeksforgeeks.org/effect-of-bias-in-neural-network/>)

Architecture: The process of assigning values to the variables and associated weights then running these through the sigmoid function, occurs for each input variable. This may occur multiple times with multiple layers. The first time this activation function occurs is called the input layer. The second time is called the “hidden” layer. The number of layers is iterated over to determine the best result for classifying the class variable (see Figure 32).

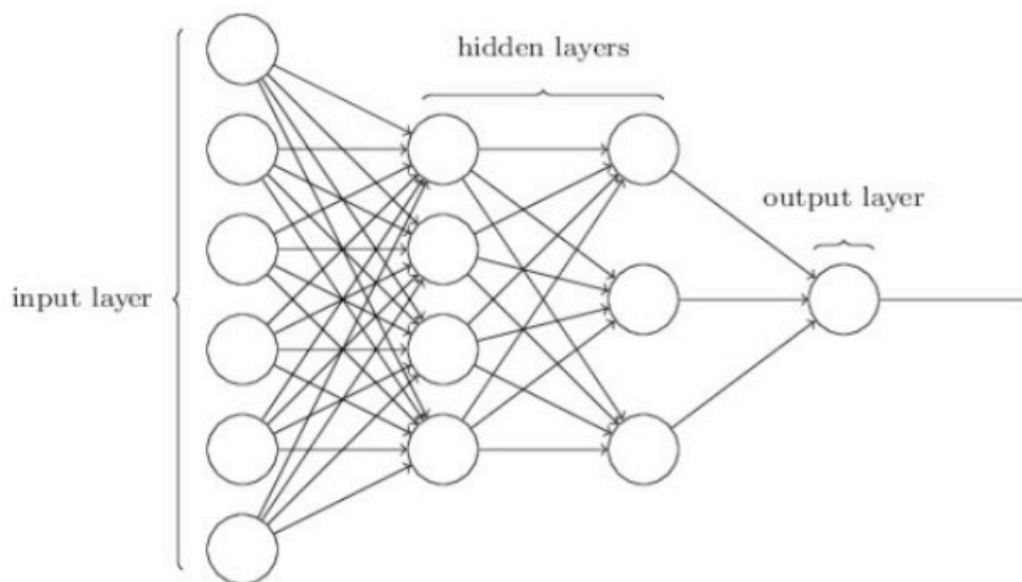


Figure 32: Architecture of the Layers of Neural Networks (Nelson, 2019)

Learning in Neural Networks: NN use inputs from the data to “learn” and modify the structure of the network to optimize the weight assigned for each input variable.

Backpropagation is the central mechanism by which NN learn

(<https://wiki.pathmind.com/backpropagation>). The purpose of backpropagation is to send information back to the NN and correct an error. The error is the guess the NN makes about the output. Therefore, backpropagation is synonymous with correction of error within the NN.

The NN backpropagates in a step-by-step fashion by first making a guess about the data using the input parameters, then measuring a loss function and finally the error is backpropagated to adjust the wrong-headed parameters. Backpropagation takes the error that is associated with the wrong answer and uses it to adjust the NN and reduce error.

The goal of the NN is to reduce error and to optimize correct outputs. The *gradient descent* is an optimization formula that is used to find error. Taking repeated steps in the opposite direction of the gradient (backwards in the NN) will map the gradient and find the minimum of the function to assist in reducing the error in the NN. In turn this assists in learning and improving the accuracy (reducing the error) of the model.

Model Fitting

The key steps used to fit the model were:

Step 1: To make sure the results were reproducible by using the `set.seed` command

Step 2: To split the data into 70% training data and 30% test data. Inspect the results via the `str` Command (Figure 33).

```
> str(train.data)          > str(test.data)
'data.frame':   482 obs. of  7 variables: 'data.frame':   240 obs. of  7 variables:
```

Figure 33: Results of splitting data into training data (left hand side) and test data (right hand side)

Step 3: Build the model using the dependent (target) variable all the independent variables (Figure 34). Class is diabetes test, if positive then 1, if negative then 0.

```
> #Build the model. If you receive a warning, rerun the command.  
> nn<-neuralnet(formula = class~preg+plas+pres+mass+pedi+age, data = train.data, hidden=2, err.fct="ce", linear.output = FALSE)
```

Figure 34: Shows the dependent variable as class and all independent variables

Step 4: Print and inspect the model from the first iteration of training set data

Step 5: Create a visualization on the training data to visualize the results

Step 6: Print and inspect the confusion matrix

Step 7: Print and inspect additional statistics to include sensitivity, specificity, positive predictive values, negative predictive values, prevalence, detection rate, detection prevalence and balanced accuracy.

Step 8: Print and inspect the training data classification accuracy

Step 9: Print and inspect the model from the test set data

Step 10: Create a visualization on the test data to visualize the results

Step 11: Print and inspect the confusion matrix on the test data

Step 12: Print and inspect additional statistics to include sensitivity, specificity, positive predictive values, negative predictive values, prevalence, detection rate, detection prevalence and balanced accuracy.

Step 13: Print and inspect the test data classification accuracy

Step 14: Compare the output statistics from the training data and test data to determine if the model can be generalized

Step 15: Iterate over the model to reduce error and increase accuracy by modifying the following parameters:

- a) Changing the number of hidden layers
- b) Changing the number of nodes in the hidden layer
- c) Using a subset of the independent variables
- d) Changing the threshold

Step 16: Experiment and iterate using decision making logic as it pertains to the objective or question being asked of the data.

Step 17: Do a final summary and inspection of the model in relation to the stated objectives

Results

Output

The default parameters of the model included all independent variables, and the target (dependent) variable was class (diabetes test, if positive then 1, if negative then 0). Results of the textual output for the summary model can be seen in Figure 35. The top area of Figure 35 shows the starting weights for the input to the hidden layer nodes and bias in the mode activation function in the model. The column heading is the hidden layer nodes of which there are two. The bottom area of Figure 35 shows weights and node activation function from hidden layer to output node.

```

> nn$weights
[[1]]
[[1]][[1]]
      [,1] [,2]
[1,]  0.1291475  0.6816551
[2,] -1.6498591 -2.3776608
[3,]  0.5862526 -0.9950384
[4,]  0.6045980  0.9271821
[5,]  0.4662801 -0.2997232
[6,]  1.0796786  0.8376053
[7,]  1.7797984  1.7113147

[[1]][[2]]
      [,1]
[1,] -0.7831101
[2,]  6.2324437
[3,] -5.5877430

```

Hidden layer nodes

Initial node activation function

Initial weights from input to hidden layer

Node activation function from hidden layer to

Weights from hidden layer to output layer

Figure 35: First iteration of the model output

The predicted probability for the class/diabetes label after the final iteration of the model is shown in Figure 36. The amount of error the initial NN generated was 2.12. The number of steps taken to generate the model is 2088. The stopping threshold reached was 0.009.

```

> nn$result.matrix      # number
error                   2.119149e+02
reached.threshold      9.722137e-03
steps                   2.088000e+03
Intercept.to.1layhid1  1.291475e-01
preg.to.1layhid1       -1.649859e+00
plas.to.1layhid1       5.862526e-01
pres.to.1layhid1       6.045980e-01
mass.to.1layhid1       4.662801e-01
pedi.to.1layhid1       1.079679e+00
age.to.1layhid1        1.779798e+00
Intercept.to.1layhid2  6.816551e-01
preg.to.1layhid2       -2.377661e+00
plas.to.1layhid2       -9.950384e-01
pres.to.1layhid2       9.271821e-01
mass.to.1layhid2       -2.997232e-01
pedi.to.1layhid2       8.376053e-01
age.to.1layhid2        1.711315e+00
Intercept.to.class     -7.831101e-01
1layhid1.to.class      6.232444e+00
1layhid2.to.class      -5.587743e+00

```

Amount of error in the model

Threshold (stopping criteria) reached

Number of training steps taken to get the result

Weights for the first hidden layer node

Weights for the second hidden layer node

Weights for the output node

A snapshot of the model can be found when viewing the first ten predicted outputs generated by the NN (Figure 37). The test is positive if the probability is 0.5 or above. The test is negative if the probability is below 0.5. Positive tests included the fourth, fifth and sixth tests. Negative tests included the first, second, third, seventh, eighth, ninth and tenth observations. When compared to actual values in the first 10 observations, the model correctly predicted 40% (in green) and incorrectly predicted 60% (in red).

```
> nn$net.result[[1]][1:10] # display the first 10 predicted probabilities
[1] 0.47918085 0.10786682 0.06753899 0.68128539 0.54939549 0.51295036 0.44485451 0.21043983 0.33522285 0.27117746
```




Figure 37: First 10 predicted outputs generated by the NN and compared with actual results.

Results of the training data, first version of the NN (Figure 38) and related statistics (Figure 39) shows an adequate first model. As previously stated there six input variables and two hidden layers in the model.

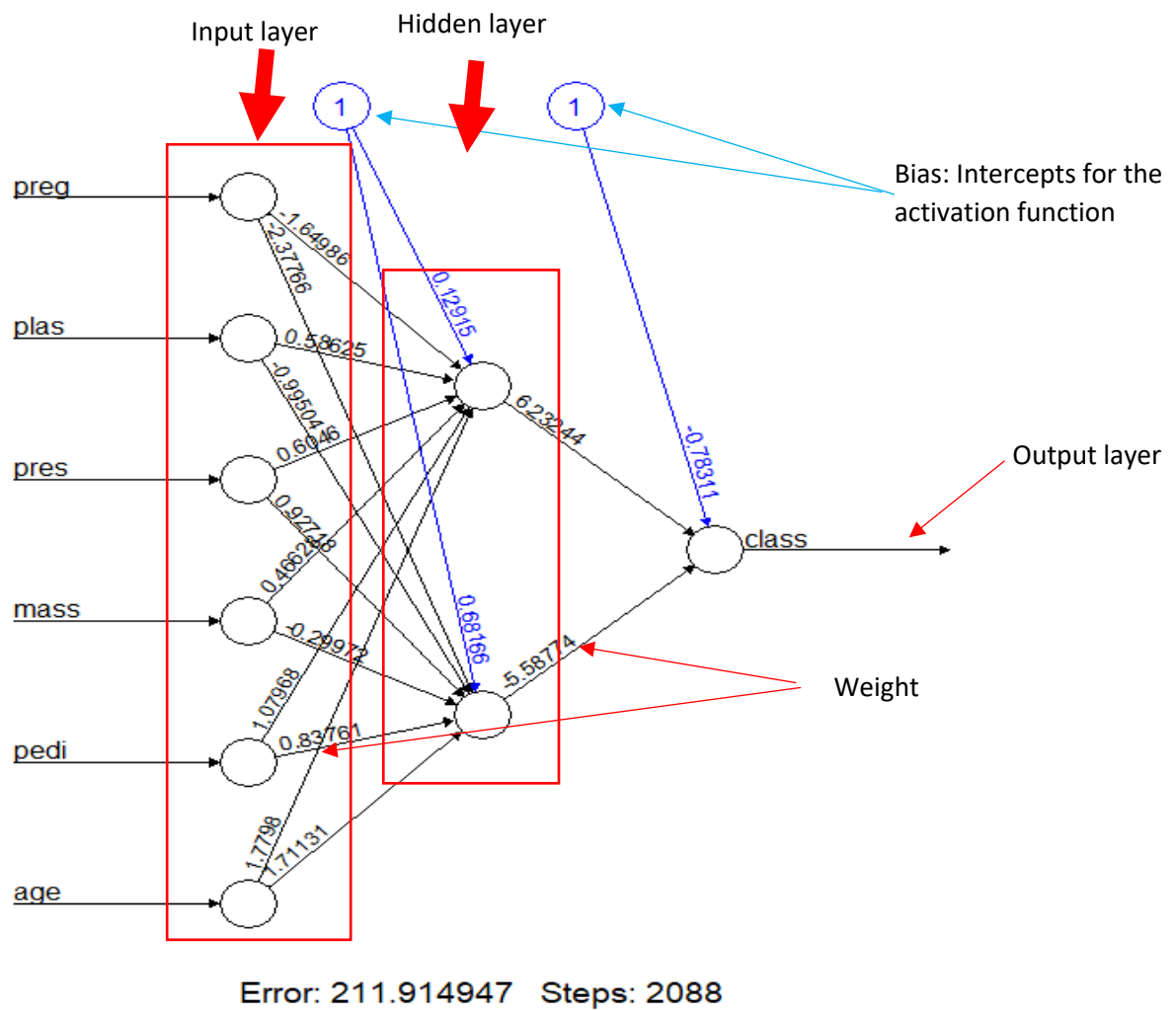


Figure 38: Visualization of the NN from the first iteration

Accuracy for the training data was 78% (Figure 39). Sensitivity, which indicates the correct classification of Pima women who tested positive for diabetes was high at 91%. Specificity is a measure of the Pima women who tested negative for diabetes was low-to-moderate at 54%.


```

> # confusion matrix for the training set
> trainPred <- compute(nn, train.data[, 0:6])$net.result
> trainPred<-apply(trainPred, c(1), round)
> table(trainPred, train.data$class, dnn =c("Predicted", "Actual"))
      Actual
Predicted 0  1
      0 286  77
      1  30  89
> mean(trainPred==train.data$class)
[1] 0.7780083
> confusionMatrix(table(trainPred, train.data$class), dnn=c("predicted", "actual"))
Confusion Matrix and Statistics

trainPred    0    1
      0 286   77
      1  30   89

      Accuracy : 0.778
      95% CI : (0.7382, 0.8143)
No Information Rate : 0.6556
P-value [Acc > NIR] : 3.169e-09

      Kappa : 0.473

McNemar's Test P-Value : 8.708e-06

      Sensitivity : 0.9051
      Specificity : 0.5361
Pos Pred Value : 0.7879
Neg Pred Value : 0.7479
Prevalence : 0.6556
Detection Rate : 0.5934
Detection Prevalence : 0.7531
Balanced Accuracy : 0.7206

'Positive' Class : 0

```

Figure 39: Confusion Matrix for the Training Data.

The stated objective of the analysis was to explore the factors that impact the Pima women's negative or positive test results for diabetes. Understanding what factors affect the Pima Indian female's high incidence rate of diabetes may provide some insight into strategies for mitigation.

If the level of accuracy is between 70-80% the model is categorized as a "good" model (Vallantin, 2020). A 91% sensitivity rate provides further evidence that the model is sound. Sensitivity, specificity, and accuracy on the test data was also examined at this time and was within range of the training data results.

In summary, these results are informative for the Pima community, health care

professional and the National Institute of Diabetes and Digestive and Kidney Disease in that they can evaluate the factors that contribute to a positive or negative diabetes test result with a good level of accuracy. More specifically, the Pima community, health care professionals and the National Institute of Diabetes and Digestive and Kidney Disease can view the woman's positive versus negative test results in terms of variables that the organization and community can work to influence. Most encouragingly would be the health-related variables, in particular BMI highly contributed to the model and can be a controllable factor. Therefore, the stated objective to explore the factors that impact Pima women's diabetes test results was achieved by this analysis.

Model Properties

After examining the initial model, two more specific goals were considered:

1. Can specificity be increased while maintaining high sensitivity and accuracy?
2. Can health variables, be used to create better guidance for the Pima community? This was considered so that the guidance back to the community could include actionable, controllable steps to reduce the incidence of diabetes.

To achieve these goals, input parameters were iterated over and included changing the number of hidden layers and node, using a subset of the independent variables, and changing the threshold levels.

In one of the iterations the uncontrollable variables were removed from the model, these included age and Diabetic Pedigree Function, which is a measure of genetics and family history. Unfortunately, that left only negative test results. When adding age back into the model this slightly improved the already very high sensitivity to 97%, slightly increased the overall accuracy to 79% but dropped the specificity further to a low level of 46%. This is an unacceptable specificity rate and therefore this iteration of the model was not adopted.

In further experimentation, when adding 5 nodes the accuracy was 82%, sensitivity was 95% and specificity was 56%. However, the model decreased by 10% when comparing the training data to test data, indicating a possible issue with overfitting and therefore generalization of the results. When reducing the nodes to 4 or 3 the specificity rate remained lower than 50%, therefore these models were set aside. Thresholds of 0.1 and 0.01 were examined but also proved inadequate. Addition hidden layers were added but those also proved inadequate.

The model with the best overall results when considering the objective had class as the dependent variable, and number of pregnancies, plasma, blood pressure, BMI (mass) and age as independent variables (Figure 40).

```
> nn$model.list          # list dependent and independent variables in the model
$response
[1] "class"

$variables
[1] "preg" "plas" "pres" "mass" "age"
```

Figure 40: Lists the independent and dependent variables in the final version of the model.

The number of hidden layers was ne with six nodes (Figure 41).

```
> #Run the commands to display the network properties
> nn$call          # the command we ran to generate the model
neuralnet(formula = class ~ preg + plas + pres + mass + age,
  data = train.data, hidden = 6, err.fct = "ce", linear.output = FALSE)
```

Figure 41: Command used to generate the final version of the model

The overall accuracy was high at 83% and sensitivity and specificity were also high at 83% and 81% respectively. Also noted was a decrease in the error from 211 in the original model to 178. When examined against the test data results remained high.

This model (Figure 42), and set of parameters, was evaluated against the overall objective: to explore the factors that impact a positive or negative diagnosis of diabetes in Pima women for the Pima community, health care professionals and the National Institute of Diabetes

and Digestive and Kidney Disease. More specifically, this information, should the model be accurate enough, could assist in providing mitigation strategies to controllable elements associated with positive testing such as Body Mass Index (mass) and other health related variables. Given the objective and the need to accurately classify those with a positive diagnosis (sensitivity) *and* Pima women who tested negative for diabetes (specificity) this model was adopted (See Appendix A for complete details on the adopted model).

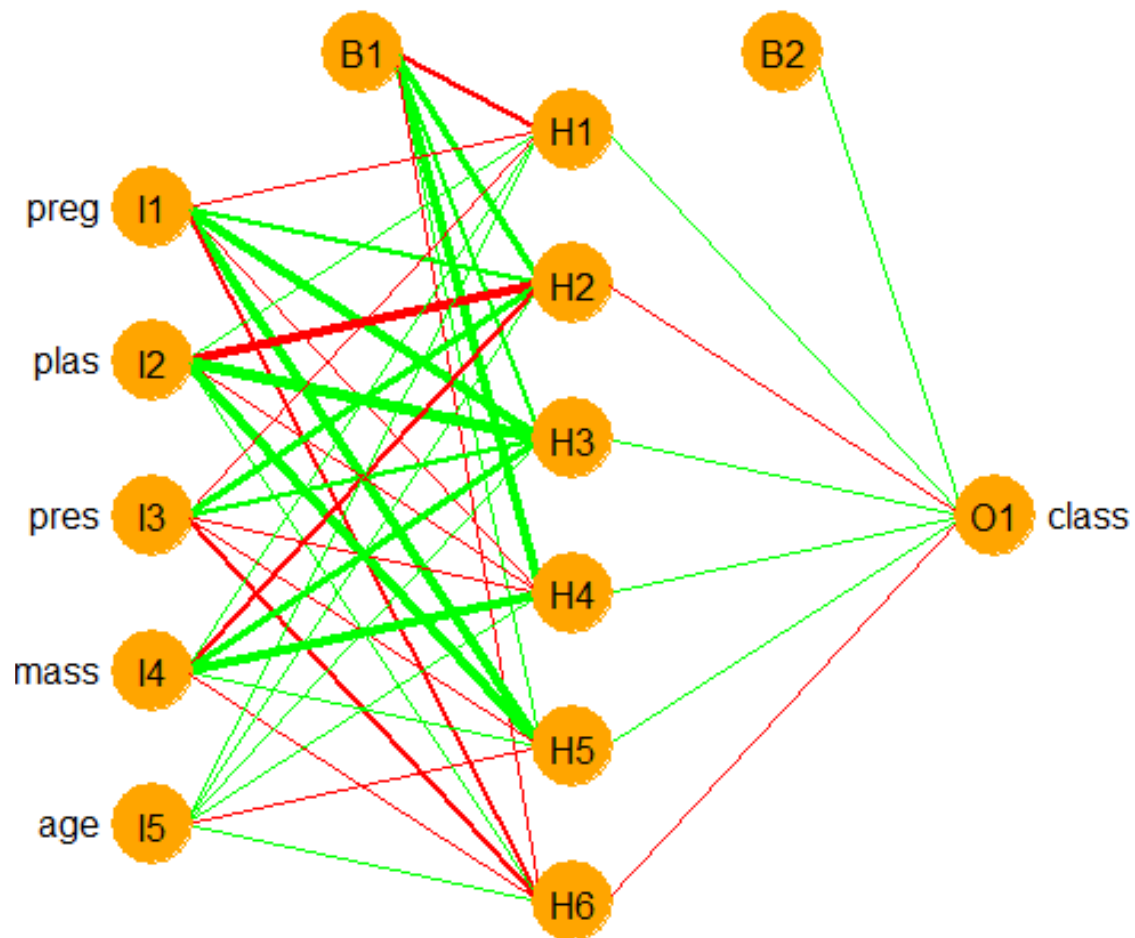


Figure 42: NN visualization of the final iteration of the model

The additional, specific goal of finding the contributions of controllable health variables to take measurable steps to reduce the incidence of diabetes in the female Pima community was also found (Figure 43). BMI (mass) and Plasma Glucose Concentration (plas)

were the most important (highest weighted variables in contributing to the outcome of a positive or negative diabetes test. This is an important finding that can provide actional education and health care initiatives for the Pima community.

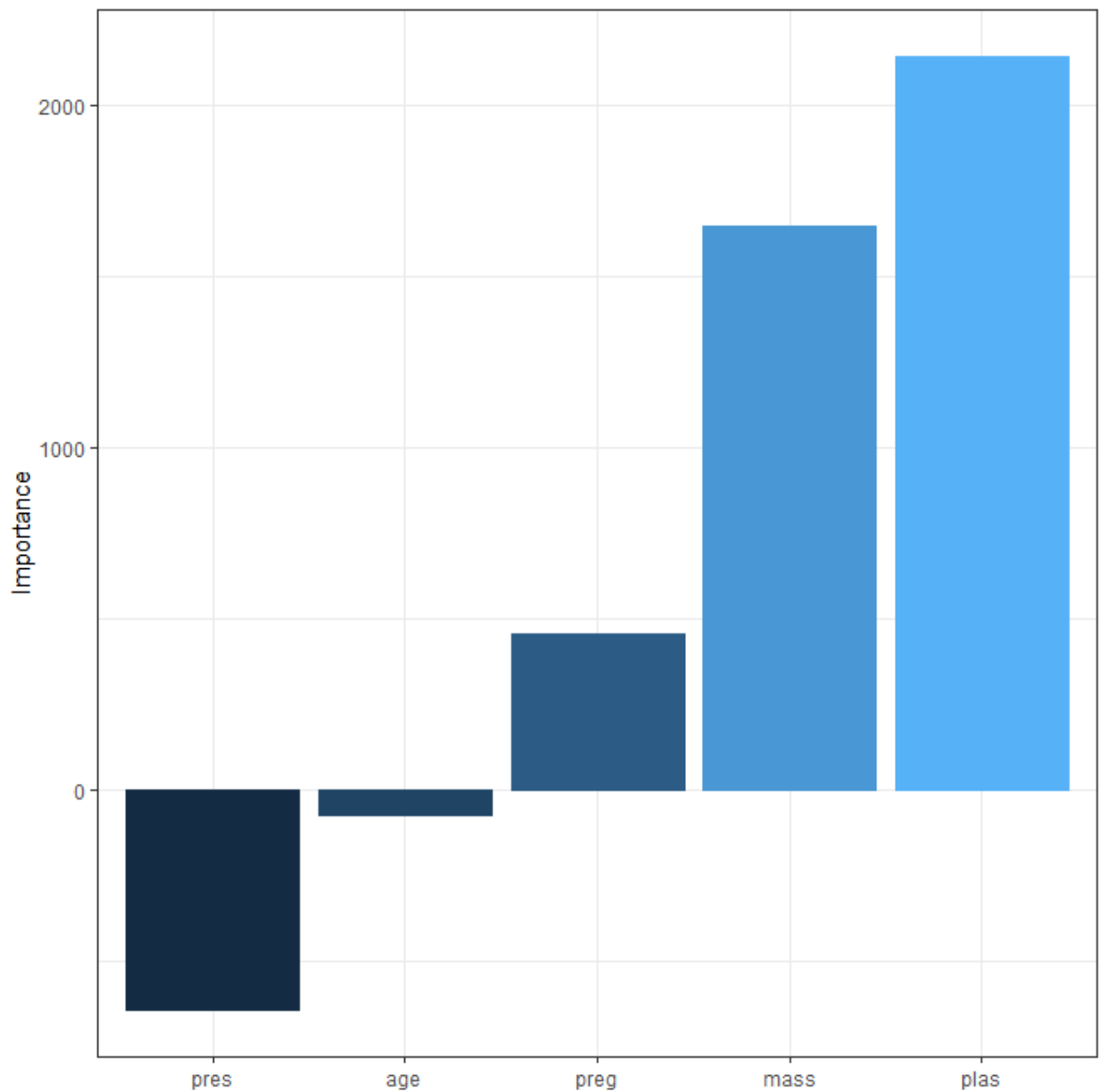


Figure 43: Relative Importance of each Independent Variable

In summary, the results on the training data indicate that the model was good. Next, the generalizability of the model is to be examined using the test data. The main reason for separating the training and test data sets is to mitigate overfitting.

Evaluation

Evaluate the model on the test data: The model accuracy on the test data was found to be 74% (Figure 44). A 4% discrepancy between the training and test data was deemed acceptable and not indicative of overfitting.

```
> # confusion matrix for the test set
> testPred <- compute(nn, test.data[, 0:5])$net.result
> testPred<-apply(testPred, c(1), round)
> table(testPred, test.data$class, dnn =c("Predicted", "Actual"))
      Actual
Predicted 0  1
      0 119 24
      1  38 59
> confusionMatrix(table(testPred, test.data$class), dnn=c("predicted", "actual"))
Confusion Matrix and Statistics

testPred   0   1
      0 119  24
      1  38  59

              Accuracy : 0.7417
              95% CI   : (0.6814, 0.7958)
    No Information Rate : 0.6542
    P-Value [Acc > NIR] : 0.002261

              Kappa   : 0.4509

  Mcnemar's Test P-value : 0.098738

    Sensitivity : 0.7580
    Specificity : 0.7108
   Pos Pred Value : 0.8322
   Neg Pred Value : 0.6082
    Prevalence   : 0.6542
    Detection Rate : 0.4958
    Detection Prevalence : 0.5958
    Balanced Accuracy : 0.7344

    'Positive' Class : 0
```

Figure 44: Model output from test data

Confusion Matrix (Figure 44): results from the confusion matrix command show how many records in the test data have each predicted positive or negative diabetes test results. The number of correctly classified instances in the test data set = $119 + 59 = 178$. The number of misclassified instances in the test data set = $38 + 24 = 62$. The total number of instances in the test dataset = $178 + 62 = 240$ (which corresponds to the str command output in Figure 33 of test data observations). The classification accuracy is the sum of numbers on diagonal/sum of all numbers = $178/240 = 74\%$ *classification accuracy*.

Sensitivity Rate /True Positives: Sensitivity rate indicates the correct classification of Pima women who tested positive for diabetes was 76%. The number of true positives in the dataset is 119 and is found at the top left cell of the confusion matrix (Figure 44).

Formula: true positive/total actual positive

Calculation: $119/(119 + 59) = 0.7580$ or **76%**

Specificity Rate /True Negative: Specificity rate is a measure of the Pima women who tested negative for diabetes was 71%. The number of true negatives in the dataset is 59 and is found at the bottom right of the confusion matrix (Figure 44).

Formula: true negative/total actual negative

Calculation: $59/(119 + 59) = 0.7108$ or **71%**

Positive Predictive Value: is the true positives divided by the predicted positives. It is found by using the top row of the confusion matrix (Figure 44).

Formula: True positive/predicted as positive

Calculation: $119/(119 + 24) = 0.8322$ or **83%**

Negative Predictive value: is the true negatives divided by the predicted negatives. It is found by using the bottom row of the confusion matrix (Figure 44).

Formula: True negative/predicted as negative

Calculation: $59/(38 + 59) = 0.6082$ or **61%**

False Positive: also known as a *Type I error* occur when the null hypothesis is incorrectly rejected. The creates a “false positive” that leads to a conclusion that the alternate hypothesis is true when it is not. The number of false positives in the dataset is 38 and is found in the bottom left of the confusion matrix (Figure 44). Therefore, **38** people in this dataset may be misclassified as having a significant difference when there is not one.

False Negative: also known as a *type II error* is the non-rejection of a false null hypothesis. Whereby a true difference is not found. The number of false negatives in the dataset is **24** and is found at the top right of the confusion matrix (Figure 44).

Prevalence: is the number of true positives over the total number of instances. In other words, how often does the “yes” condition occur.

Formula: True positive/total number of instances

Calculation: $119/(119+24+38+59) = 0.4958$ or **50%**

Precision: is the number of true positives over the number of predicted positives (Figure 44). In other words, when the model predicts yes, how often is it correct?

Formula: true positive/number of predicted positives

Calculation: $59/(59 + 24) = 0.7108$ or **71%**

Kappa statistic: Is the agreement between predicted and observed considering the accuracy by chance (Figure 44). Values range from -1 to +1 which indicate perfect agreement. Values

closer to zero indicate no agreement. This data set has a Kappa value of 0.4509, which is low-to-moderate.

Formula: (Bati, 2021):

$$Kappa = \frac{n_a - n_c}{n - n_c} = \frac{p_a - p_c}{1 - p_c}$$

n: number of cases

n_a: number of agreement

n_c: number of agreement due to chance

p_a: proporiotn of observation in agreement,

p_c: proportion of agreement due to chance

Conclusion

Summary

The objective of the analysis was to explore the factors that impact a positive *or* negative diagnosis of diabetes in Pima women for the Pima community, health care professionals and the National Institute of Diabetes and Digestive and Kidney Disease. Armed with this information mitigation strategies such as community education, regular health visits and dietary modification could be employed to reduce the high level of diabetes in Pima women. Table 1 provides the analytical outcome and a possible mitigation strategy.

Outcome	Strategy
High levels of plasma glucose concentration results in a high probability of a positive diabetes test result.	Diet with low refined carbohydrates, reduction in sugar intake, increase in fiber, and water intake (https://www.healthline.com/nutrition/15-ways-to-lower-blood-sugar)

High BMI results in a high probability of a positive diabetes test result.	Increase exercise, set goals, and change diet to include more fruits and vegetables (https://www.everydayhealth.com/diet-nutrition/bmi/how-you-reduce-your-bmi-science-backed-steps/)
High number of pregnancies increased the probability of a positive diabetes test result.	Consider family planning counseling and contraception
Diastolic blood pressure, too high or too low influences both diabetes test results and heart disease (Bays, Chapman, Grandy, 2007).	Reduce salt intake, limit saturated fats, alcohol, caffeine smoking, stress. Include more garlic in diet (https://www.healthline.com/health/high-blood-pressure-hypertension/how-to-lower-diastolic-blood-pressure#tips-to-lower-blood-pressure)

Table 1: Summary Outcome and Mitigation Strategies to Reduce Diabetes in the Pima Community

In conclusion, there are various health related strategies that can make positive changes to the Pima women's health which are scientifically known to reduce diabetes. It is important for all members of the community and those who serve the community to be aware, educated, and supportive of these changes in order to increase the well-being of the Pima women in the hopes of a happy, health and long life.

Limitations

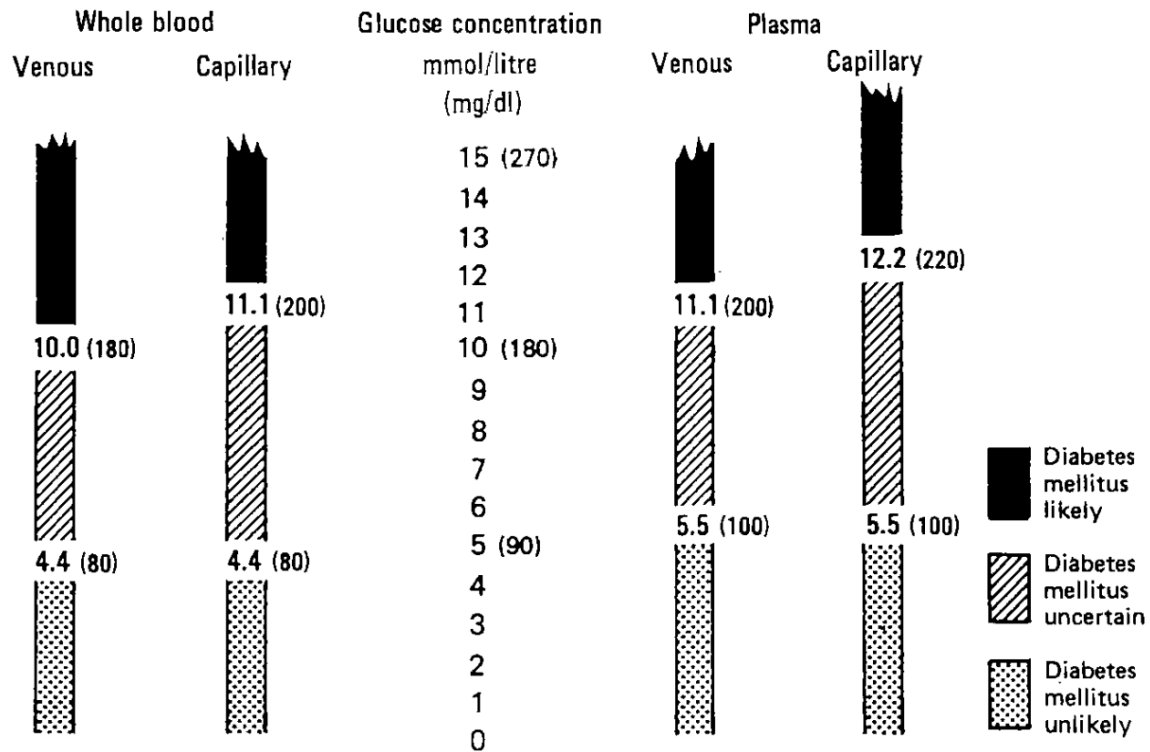
Several limitations were found while conducting the analysis. Data limitations were

significant. This is a small dataset for a NN analysis. The data was further reduced by some significant limitations.

First, the plasma glucose concentration variable did not have any units specified in the data explanation. After extensive additional research, two types of metrics can be used for this measurement (mg/DL and mmol). Millimoles per liter (mmol) was assumed, based on the numbers in the dataset. However, this was never able to be confirmed. The assumption was made based on the World Health Organization (WHO) definition and numbers in data set (http://apps.who.int/iris/bitstream/handle/10665/39592/WHO_TRS_727.pdf?sequence=1, Figure 45).

The issue is further exacerbated when examining the level of plasma against the positive or negative test for diabetes. High levels of plasma which would indicate a positive test result did not always correspond to a positive test result. This is an unresolved issue in the data set.

A second uncertainty in the data set came from the uncertain diagnosis seen in the data as described by the WHO (Figure 45) but this is not captured in data set. The dependent variable in the data was dichotomous, indicating a clear positive or negative test result. However, this may not be accurate.



WHO 851235

^a Blood glucose values in the second report of the WHO Expert Committee on Diabetes Mellitus (1) were rounded to the nearest mmol/litre. This decision was challenged on the grounds that it created comparatively large differences between the SI unit and the traditional unit (mg/dl) which could introduce potentially serious biases in diagnostic categories. For this reason, SI units have now been rounded to the nearest tenth of a mmol.

Figure 45: Summary data from the World Health Organization from the Pima Data Set

After significant research, the Diabetes Pedigree Function could not be found or explained beyond the most basic of explanations: “score the likelihood of diabetes based on family history and the genetic relationship to relatives of the patient who have diabetes” (<https://github.com/susanli2016/Machine-Learning-with-Python/issues/26>). No information regarding the scoring or scaling could be found. This makes the interpretation of this variable difficult.

Limitations for the insulin variable were also found. If you have diabetes, you might not make enough insulin (type 1 diabetes) or you might be less responsive to it (type 2 diabetes). As

a result, insulin levels may be very low (type 1 diabetes) or very high (type 2 diabetes). No information could be found as to the type of diabetes the women had. Without this information, it becomes more difficult to providing effective mitigation strategies as those strategies may differ depending on the type of diabetes.

A further data limitation was found during data exploration, likely missing data. When using the following command: `colSums(is.na(pima_diabetes))` to explore if missing data existed, none was found (Figure 46). However, when exploring each variable in the data exploration, it was clear that there was some incorrect data. For example, the insulin variable showed 375 records with zero insulin. Which would effectively indicate a person was in a coma (https://www.medicinenet.com/what_is_a_high_insulin_level/article.htm). this was clearly not the case and therefore it was assumed, but never confirmed, that the data was entered by the researchers as 0 if not data was available.

```
> # Check if the data has missing values
> colSums(is.na(pima_diabetes))
  preg  plas  pres  skin  insu  mass  pedi  age  class
    0     0     0     0     0     0     0     0     0
> |
```

The extent of this assumed missing data had follow on effects as it further reduced the already small sample size that is needed for running NN. Furthermore, the skin fold thickness variable was also found to have suspected zero input for missing data further reducing the data size.

Limitations also exist in the model. NN are still primarily used for prediction because the resulting rules are not intuitive enough for decision makers. Therefore, although the model can be classified as “good” as the level of accuracy is between 70-80% (Vallantin, 2020) it may still not be effective enough for some medical providers who are often used to seeing accuracy levels

in the mid-90% for drug related studies for example (<https://www.aafp.org>).

The limitations in the model are further exacerbated as the model can be very difficult to explain unlike decision trees for example. The “black box” nature of NN, greater computational burden and a proneness to overfitting (not found in this model) are disadvantages of NN. In a review paper by Tu (1996) comparing NN to logistic regression, e states: “*logistic regression remains the clear choice when the primary goal of model development is to look for possible causal relationships between the independent and dependent variables, and the modeler wishes to easily understand the effect of the predictor variables on the outcome.*” It would be advantageous to compare the NN model to a logistic regression. The regression would likely be more easily understood with more direct guidance on the influence of the independent variables on the output. However, what would be missed by the regression are *all* the interactions between the independent variables and how those influence the outcome.

Finally, when examining the outcome of the Pima women’s diabetes data, the goal is always to have the highest level of accuracy, sensitivity, and specificity. Although the model produced was good, with more data the improvement in these metrics may be found. This could be important for widespread adoption of the recommendations.

Improvement Areas

This data was collected about thirty years ago, should future research be conducted, adding more input variables and more observations would improve the possibility of a more meaningful outcome. This is especially the case in improving the model from good to excellent levels of accuracy.

A better understanding of the variables would also improve the understanding of the results. This is especially the case for the Diabetes Pedigree Function whereby understanding

both the scale and the questions that influence the score are needed. This is also the case for the plasma glucose concentration where a confirmation of the scale is needed.

An understanding of what happened to the uncertain diabetes diagnosis and how those were captured (or not) in this data set would also be an improvement.

Confirmation of the zeros in the dataset would be helpful so not data is assumed. Future recommendations would include proper training of research staff so that non-numeric codes for missing data were used. One example could be to use N/A.

Specific recommendations for additional independent variables include weight, height, dietary habits, exercise habits. Health visits and family habits (e.g. number of meals eaten at home per week) would also be interesting additions.

Further exploration and possible improvement could come from exploring other analytical approaches. One example may be a logistic regression method for classification. Another may be the use of other activation functions beyond the sigmoid function. For instance, a Gaussian function or hyperbolic tangent may be interesting to explore further.

Finally, another improvement areas, given the limitation of NN may be to explore easier to understand visualizations. There are many R packages for visualizations and most people are visual learners. The intuitive nature of the “biological network” may be more easily represented.

Appendix A

Output of the Final Model

```
> #make sure that the result is reproducible

> set.seed(12345)

> #split the data into a training and test set

> ind <- sample(2, nrow(pima_diabetes), replace = TRUE, prob = c(0.7, 0.3))

> train.data <- pima_diabetes[ind == 1, ]

> test.data <- pima_diabetes[ind == 2, ]

> str(train.data)

'data.frame':   482 obs. of  6 variables:

 $ preg : num  -1.149 0.336 -0.258 -0.555 1.822 ...
 $ plas : num   0.489 -0.197 -1.437 2.447 0.554 ...
 $ pres : num  -2.64 0.135 -1.824 -0.191 0.625 ...
 $ mass : num   1.541 -1.001 -0.217 -0.289 -0.783 ...
 $ age  : int   33 30 26 53 57 59 31 31 33 32 ...
 $ class: num    1 0 1 1 0 1 1 1 0 1 ...

> str(test.data)

'data.frame':   240 obs. of  6 variables:

 $ preg : num   0.6334 -0.8523 1.2276 -0.8523 0.0391 ...
```



```
$ plas : num 0.848 -1.208 1.99 -1.078 -0.393 ...
```

```
$ pres : num -0.0283 -0.518 -0.6812 -0.518 1.6041 ...
```

```
$ mass : num 0.161 -0.856 -1.335 -0.638 0.742 ...
```

```
$ age : int 50 31 32 21 30 34 51 41 41 43 ...
```

```
$ class: num 1 0 1 0 0 1 1 1 1 1 ...
```

```
> #Build the model. If you receive a warning, rerun the command.
```

```
> nn<-neuralnet(formula = class~preg+plas+pres+mass+age, data = train.data, hidden=6, err.fct="ce",  
linear.output = FALSE)
```

```
> #names command displays the available neural network properties
```

```
> names(nn)
```

```
[1] "call"          "response"      "covariate"     "model.list"    "err.fct"       "act.fct"
```

```
[7] "linear.output" "data"          "exclude"       "net.result"    "weights"
```

```
"generalized.weights"
```

```
[13] "startweights"  "result.matrix"
```

```
> #Run the commands to display the network properties
```

```
> nn$call          # the command we ran to generate the model
```

```
neuralnet(formula = class ~ preg + plas + pres + mass + age,
```

```
data = train.data, hidden = 6, err.fct = "ce", linear.output = FALSE)
```

```
> nn$response[1:10]    # actual values of the dependent variable for first 10 records
```

```
[1] 1 0 1 1 0 1 1 1 0 1
```

```
> nn$covariate [1:12,] # input variables that were used to build the model for first 12 records
```

	preg	plas	pres	mass	age
5	-1.1494323	0.4886934	-2.6399942	1.5413555	33
6	0.3362285	-0.1966798	0.1349727	-1.0012754	30
7	-0.2580358	-1.4368788	-1.8238274	-0.2166921	26
9	-0.5551680	2.4469023	-0.1914940	-0.2893387	53
13	1.8218893	0.5539670	0.6246728	-0.7833356	57
14	-0.8523001	2.1858078	-1.0076607	-0.3474560	59
17	-1.1494323	-0.1314061	0.9511395	1.9336472	31
18	0.9304928	-0.4904111	0.1349727	-0.4201026	31
19	-0.8523001	-0.6209584	-3.4561609	1.5704142	33
20	-0.8523001	-0.2293166	-0.1914940	0.3063634	32
21	-0.2580358	0.1296884	1.2776061	0.9892414	27
22	1.2276249	-0.7515056	0.9511395	0.4225979	50

```
> nn$model.list # list dependent and independent variables in the model
```

```
$response
```

```
[1] "class"
```

```
$variables
```

```
[1] "preg" "plas" "pres" "mass" "age"
```

```
> nn$net.result[[1]][1:10] # display the first 10 predicted probabilities
```

```
[1] 0.99693183 0.01920288 0.02395400 0.89078311 0.09195994 0.89078311 0.61192704 0.52938319
```

```
0.15105766 0.34039901
```

```
> nn$weights # network weights after the last method iteration
```

```
[[1]]
```

```
[[1]][[1]]
```

```
 [,1] [,2] [,3] [,4] [,5] [,6]
```

```
[1,] -116.405583 193.0068 106.540357 219.71002250 20.057813 -38.992276
```

```
[2,] -36.035283 113.4846 247.139044 -9.31351029 276.198127 -88.161204
```

```
[3,] 29.705218 -222.5608 282.816111 -54.13189201 253.353703 3.528106
```

```
[4,] -4.499902 157.4869 137.128333 -48.81178615 -67.339369 -75.305590
```

```
[5,] 3.896758 -103.6648 166.225168 261.77827205 34.103671 -21.263658
```

```
[6,] 2.818514 15.0540 9.103547 0.07217405 -2.947379 7.172052
```

```
[[1]][[2]]
```

```
 [,1]
```

[1,] 1.033929

[2,] 1.981097

[3,] -5.328165

[4,] 2.181713

[5,] 2.407644

[6,] 1.643330

[7,] -1.820781

> nn\$startweights # weights on the first method iteration

[[1]]

[[1]][[1]]

[,1] [,2] [,3] [,4] [,5] [,6]

[1,] -1.81283376 1.6683572 0.5817001 2.2597720 -0.1505571 0.4981722

[2,] 0.28860021 0.6726145 0.6660207 -0.4772723 0.9042436 0.1195602

[3,] -0.18962258 -0.2775180 -0.7789022 -0.1025805 2.2420361 -0.3672027

[4,] 0.01786021 -0.1460264 1.1633248 0.3686962 -1.1951229 0.2623311

[5,] 0.65043024 1.7014318 -1.9645442 -0.5354330 -0.4185226 0.2627041

[6,] 0.31025499 0.4713579 0.7691707 0.5066019 0.7982514 0.6407389

```
[[1]][[2]]
```

```
[,1]
```

```
[1,] 0.307089831
```

```
[2,] -0.033129401
```

```
[3,] -1.374750527
```

```
[4,] 0.627965113
```

```
[5,] 0.002143951
```

```
[6,] 0.284377723
```

```
[7,] -1.001779086
```

```
> nn$result.matrix      # number of trainings steps, the error, and the weights
```

```
[,1]
```

```
error      1.781355e+02
```

```
reached.threshold  9.623280e-03
```

```
steps      3.908100e+04
```

```
Intercept.to.1layhid1 -1.164056e+02
```

```
preg.to.1layhid1    -3.603528e+01
```

plas.to.1layhid1	2.970522e+01
pres.to.1layhid1	-4.499902e+00
mass.to.1layhid1	3.896758e+00
age.to.1layhid1	2.818514e+00
Intercept.to.1layhid2	1.930068e+02
preg.to.1layhid2	1.134846e+02
plas.to.1layhid2	-2.225608e+02
pres.to.1layhid2	1.574869e+02
mass.to.1layhid2	-1.036648e+02
age.to.1layhid2	1.505400e+01
Intercept.to.1layhid3	1.065404e+02
preg.to.1layhid3	2.471390e+02
plas.to.1layhid3	2.828161e+02
pres.to.1layhid3	1.371283e+02
mass.to.1layhid3	1.662252e+02
age.to.1layhid3	9.103547e+00
Intercept.to.1layhid4	2.197100e+02
preg.to.1layhid4	-9.313510e+00
plas.to.1layhid4	-5.413189e+01

pres.to.1layhid4	-4.881179e+01
mass.to.1layhid4	2.617783e+02
age.to.1layhid4	7.217405e-02
Intercept.to.1layhid5	2.005781e+01
preg.to.1layhid5	2.761981e+02
plas.to.1layhid5	2.533537e+02
pres.to.1layhid5	-6.733937e+01
mass.to.1layhid5	3.410367e+01
age.to.1layhid5	-2.947379e+00
Intercept.to.1layhid6	-3.899228e+01
preg.to.1layhid6	-8.816120e+01
plas.to.1layhid6	3.528106e+00
pres.to.1layhid6	-7.530559e+01
mass.to.1layhid6	-2.126366e+01
age.to.1layhid6	7.172052e+00
Intercept.to.class	1.033929e+00
1layhid1.to.class	1.981097e+00
1layhid2.to.class	-5.328165e+00
1layhid3.to.class	2.181713e+00

```

1layhid4.to.class    2.407644e+00

1layhid5.to.class    1.643330e+00

1layhid6.to.class    -1.820781e+00

> #Model evaluation; Round the predicted probabilities

> mypredict<-compute(nn, nn$covariate)$net.result

> mypredict<-apply(mypredict, c(1), round)

> mypredict [1:10]

    5  6  7  9 13 14 17 18 19 20

    1  0  0  1  0  1  1  1  0  0

> # confusion matrix for the training set

> trainPred <- compute(nn, train.data[, 0:5])$net.result

> trainPred<-apply(trainPred, c(1), round)

> table(trainPred, train.data$class, dnn =c("Predicted", "Actual"))

      Actual
Predicted 0  1
0 263  31
1  53 135

> mean(trainPred==train.data$class)

[1] 0.8257261

```



```
> confusionMatrix(table(trainPred, train.data$class), dnn=c("predicted", "actual"))
```

Confusion Matrix and Statistics

```
trainPred 0 1
```

```
0 263 31
```

```
1 53 135
```

Accuracy : 0.8257

95% CI : (0.7889, 0.8585)

No Information Rate : 0.6556

P-Value [Acc > NIR] : < 2e-16

Kappa : 0.6258

Mcnemar's Test P-Value : 0.02195

Sensitivity : 0.8323

Specificity : 0.8133

Pos Pred Value : 0.8946

Neg Pred Value : 0.7181

Prevalence : 0.6556

Detection Rate : 0.5456

Detection Prevalence : 0.6100

Balanced Accuracy : 0.8228

'Positive' Class : 0

```
> # confusion matrix for the test set
```

```
> testPred <- compute(nn, test.data[, 0:5])$net.result
```

```
> testPred<-apply(testPred, c(1), round)
```

```
> table(testPred, test.data$class, dnn =c("Predicted", "Actual"))
```

	Actual	
Predicted	0	1
0	119	24
1	38	59

```
> mean(testPred==test.data$class)
```

```
[1] 0.7416667
```

```
> confusionMatrix(table(testPred, test.data$class), dnn=c("predicted", "actual"))
```

Confusion Matrix and Statistics

```
testPred 0 1
```

```
0 119 24
```

```
1 38 59
```

Accuracy : 0.7417

95% CI : (0.6814, 0.7958)

No Information Rate : 0.6542

P-Value [Acc > NIR] : 0.002261

Kappa : 0.4509

Mcnemar's Test P-Value : 0.098738

Sensitivity : 0.7580

Specificity : 0.7108

Pos Pred Value : 0.8322

Neg Pred Value : 0.6082

Prevalence : 0.6542

Detection Rate : 0.4958

Detection Prevalence : 0.5958

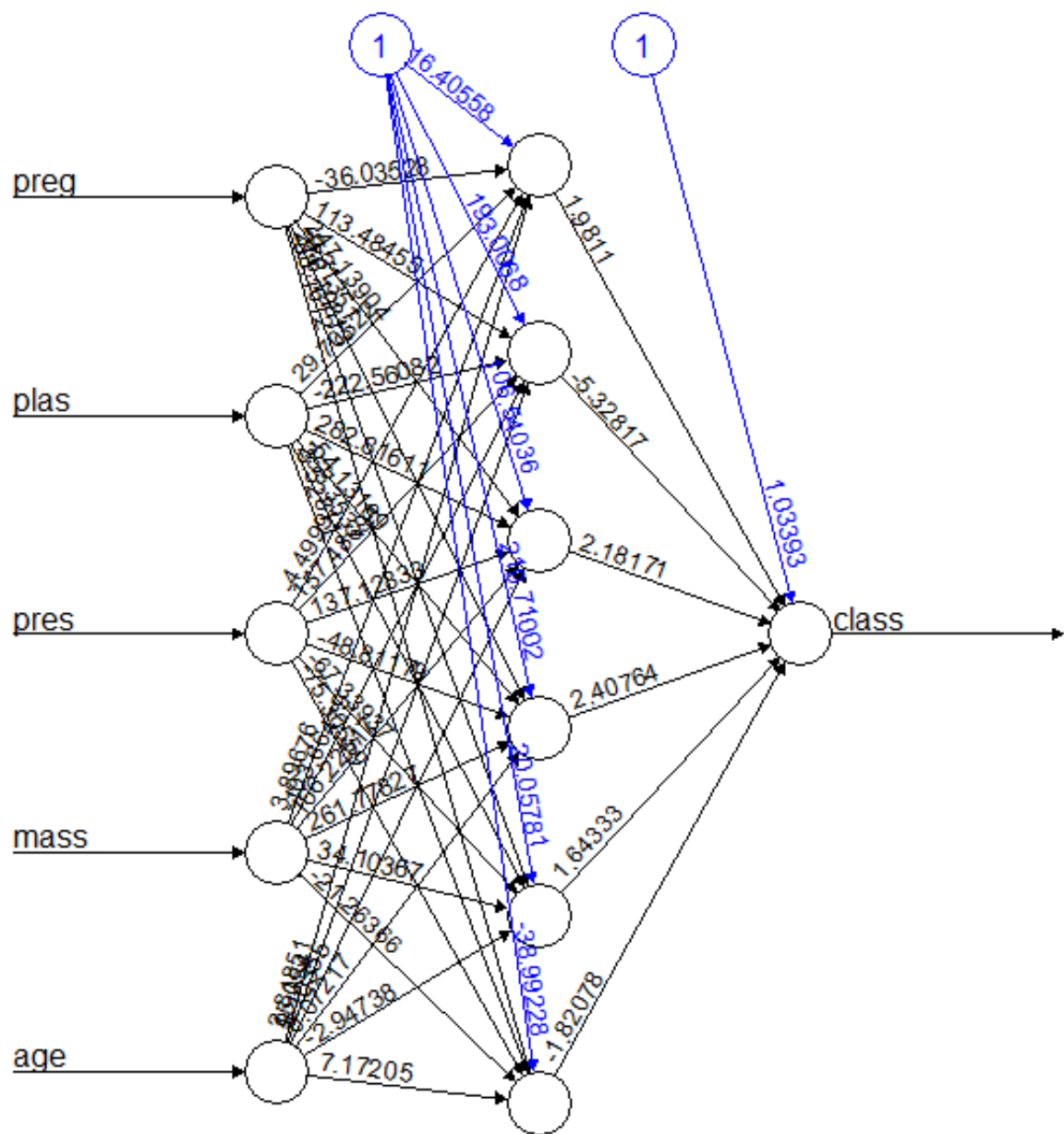
Balanced Accuracy : 0.7344

'Positive' Class : 0

> # Visualize the model

> plot(nn) # plot the network

>



Error: 178.135536 Steps: 39081

Appendix B

References

- AAFP.org (2021). *Clinical Practice Guideline Manual*. Retrieved from:
<https://www.aafp.org/family-physician/patient-care/clinical-recommendations/cpg-manual.html#i>
- Bati, F. (2021). Lecture: *Classification*. University of Maryland University College. Data 630: Module 4.
- Bati, F. (2021b). Lecture: *Model Explainability*. University of Maryland University College. Data 630: Module 4.
- Bays, H. E., Chapman, R. H., Grandy, S., & SHIELD Investigators' Group (2007). The relationship of body mass index to diabetes mellitus, hypertension, and dyslipidemia: comparison of data from two national surveys. *International journal of clinical practice*, 61(5), 737–747. <https://doi.org/10.1111/j.1742-1241.2007.01336.x>
- Brownlee, J. (2014). *Case study: Predicting the onset of diabetes within five years*. Weka Machine Learning. <https://machinelearningmastery.com/case-study-predicting-the-onset-of-diabetes-within-five-years-part-1-of-3/>
- Cooper, R., Stamler, J., Dyer, A., & Garside, D. (1978). The decline in mortality from coronary heart disease, U.S.A., 1968–1975. *Journal of Chronic Diseases*, 31, 709 – 720.
- Deng, H., Runger, G., Tuv, E. (2011). [*Bias of importance measures for multi-valued attributes and solutions*](#). Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN).

Everydayhealth.com (2019). *9 Steps Proven to Help you Lower your BMI*. Retrieved from:

<https://www.everydayhealth.com/diet-nutrition/bmi/how-you-reduce-your-bmi-science-backed-steps/>

Fibre2fashion.com (2021). *Size Zero Debates – Hot or Not?* Retrieved from:

<https://www.fibre2fashion.com/industry-article/3445/size-zero-debates-hot-or-not>

James, G., Whitten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Application in R*. Springer, New York, NY.

W. C. Knowler, P. H. Bennett, R. F. Hamman, and M. Miller (1978). Diabetes incidence and prevalence in Pima Indians: A 19-fold greater incidence than in Rochester, Minnesota. *American Journal of Epidemiology*, 108(6), 497–505.

W. C. Knowler, D. J. Pettitt, P. J. Savage, and P. H. Bennett (1981). Incidence in Pima Indians: Contributions to obesity and prenatal diabetes. *American Journal of Epidemiology*, 113(2), 144– 156.

Geekforgeeks.org (2018). *Effect of Bias in Neural Network*. Retrieved from:

<https://www.geeksforgeeks.org/effect-of-bias-in-neural-network/>

Github.com (2021). *What's the Definition of Diabetes Pedigree Function?* Retrieved from:

<https://medlineplus.gov/ency/article/003101.htm>

Hastie, T., Tibshiraani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. 2nd Ed. Springer, New York, NY.

Healthline.org (2021). *Blood Pressure Readings Explained*. Retrieved from:

<https://www.healthline.com/health/high-blood-pressure-hypertension/blood-pressure-reading-explained>

Healthline.org (2021). *15 Easy Ways to Lower Blood Sugar Levels Naturally*. Retrieved from:

<https://www.healthline.com/nutrition/15-ways-to-lower-blood-sugar>

Healthline.org (2021). *20 Ways to Lower Diastolic Blood Pressure*. Retrieved from:

https://www.healthline.com/health/high-blood-pressure-hypertension/how-to-lower-diastolic-blood-pressure#_noHeaderPrefixedContent

Heart.org (2021). *Health Threats from High Blood Pressure*. Retrieved from:

<https://www.heart.org/en/health-topics/high-blood-pressure/health-threats-from-high-blood-pressure#:~:text=In%20most%20cases%2C%20the%20damage,flow%20to%20the%20heart%20muscle>

Hoehner, C. M., Williams, D. E., Sievers, M. L., Knowler, W. C., Bennett, P. H., & Nelson, R.

G. (2006). Trends in heart disease death rates in diabetic and nondiabetic Pima

Indians. *Journal of Diabetes and Its Complications*, 20(1), 8–13. <https://doi->

[org.ezproxy.umgc.edu/10.1016/j.jdiacomp.2005.06.003](https://doi-org.ezproxy.umgc.edu/10.1016/j.jdiacomp.2005.06.003)

O. Maimon, L. Rokach (eds.), *Data Mining and Knowledge Discovery Handbook*, 2nd ed.,

DOI 10.1007/978-0-387-09823-4_21, Springer Science+Business Media, LLC 2010

Pavkov, M., Sievers, M. L., Knowler, W. C., Bennett, P. H., & Nelson, R. G. (2004). An

explanation for the increase in heart disease mortality rates in diabetic Pima Indians:

Effect of renal replacement therapy. *Diabetes Care*, 27, 1132– 1136.

Piryonesi S. Madeh; El-Diraby Tamer E. (2020). "Role of Data Analytics in Infrastructure Asset

Management: Overcoming Data Size and Quality Problems". *Journal of Transportation*

Engineering, Part B: Pavements. 146 (2) [doi:10.1061/JPEODX.0000175](https://doi.org/10.1061/JPEODX.0000175)

- Piryonesi, S. Madeh; El-Diraby, Tamer E. (2021). "Using Machine Learning to Examine Impact of Type of Performance Indicator on Flexible Pavement Deterioration Modeling". *Journal of Infrastructure Systems*. **27** (2) doi:[10.1061/\(ASCE\)IS.1943-555X.0000602](https://doi.org/10.1061/(ASCE)IS.1943-555X.0000602)
- Medlineplus.gov (2021). *Low Blood Sugar – Self-Care*. Retrieved from:
<https://medlineplus.gov/ency/patientinstructions/000085.htm>
- Medicinenet.com (2021). What is a High Insulin Level? Retrieved from:
https://www.medicinenet.com/what_is_a_high_insulin_level/article.htm
- Medplus.gov (2021). Overweight. Retrieved from:
<https://medlineplus.gov/ency/article/003101.htm>
- P. M. Murphy and D. W. Aha (1994). UCI repository of machine learning databases (machinereadable data depository). Department of Information and Computer Science, University of California, Irvine, CA.
- Moffatt, R. J., Sady, S. P., & Owen, G. M. (1980). Height, weight and skinfold thickness of Michigan adults. *American journal of public health*, 70(12), 1290–1292.
<https://doi.org/10.2105/ajph.70.12.1290>
- Narayan, K.M.V., Boyle, J.P., Thompson, T.J., Gregg, E.W. & Williamson, D.F. (2007). *Effect of BMI on Lifetime Risk of Diabetes in the U.S.* Diabetes Care Journal, 30 (6) 1562-1566; DOI: 10.2337/dc06-2544
- Nielsen, M. (2019) Neural Networks and Deep Learning:
<http://neuralnetworksanddeeplearning.com/>

Ng, A. (2021). *Introduction to Supervised Learning*. Retrieved from:

<https://www.coursera.org/lecture/machine-learning/supervised-learning-1VkCb>

Ruiz, L., Colley, J.R.T & Hamilton, P.J.S. (1971). Measurement of triceps skinfold thickness an investigation of sources of variation. *British Journal of Prev. Social Medicine*, 25, 165-167.

Sievers, M. L., Nelson, R. G., & Bennett, P. H. (1996). Sequential trends in overall and cause-specific mortality in diabetic and nondiabetic Pima Indians. *Diabetes Care*, 19, 107–111.

Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49(11), 1225–1231. [https://doi-org.ezproxy.umgc.edu/10.1016/s0895-4356\(96\)00002-9](https://doi-org.ezproxy.umgc.edu/10.1016/s0895-4356(96)00002-9)

Vallantin, L. (2020). *Why you Should not Trust only in Accuracy to Measure Machine Learning Performance*. Retrieved from: <https://medium.com/@limavallantin/why-you-should-not-trust-only-in-accuracy-to-measure-machine-learning-performance-a72cf00b4516>

Wikipedia.org *Pima People*. Retrieved from: https://en.wikipedia.org/wiki/Pima_people

Wiki.pathmind.com (2021). A Beginner's Guide to Backpropagation in Neural Networks. Retrieved from: <https://wiki.pathmind.com/backpropagation>

World Health Organization (1985). Who Technical Series Report, No. 646. WHO: Diabetes Mellitus. Geneva, SW.

Zhang, G. P. Neural Networks for Data Mining. *Soft Computing for Knowledge Discovery and Data Mining*. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-69935-6_2