Noa Pereira Prada Schnor

PIMA INDIANS DIABETES DATASET

The Pima Indian Diabetes Dataset, originally from the National Institute of Diabetes and Digestive and Kidney Diseases, contains information of 768 women from a population near Phoenix, Arizona, USA. The outcome tested was Diabetes, 258 tested positive and 500 tested negative. Therefore, there is one target (dependent) variable and the following attributes (TYNECKI, 2018):

- Pregnancies (number of times pregnant),
- Oral glucose tolerance test - OGTT (two hour plasma glucose concentration after 75g anhydrous glucose in mg/dl),
- Blood Pressure (Diastolic Blood Pressure in mmHg),
- Skin Thickness (Triceps skin fold thickness in mm),
- Insulin (2 h serum insulin in mu U/ml),
- BMI (Body Mass Index in kg/m^2),
- Age (years),
- Pedigree Diabetes Function ('function that represents how likely they are to get the disease by extrapolating from their ancestor's history')

PIMA INDIANS AND DIABETES

Pima are descendants of people that inhabited the Sonoran desert and Sierra Madre areas for centuries. Around 300 B.C. they moved to Gila River Valley at the time in Mexico, but region that was acquired by the United States in 1853. A Pima reservation was created in Arizona in 1959 and they adapted to their desert homeland by directing water to support a subsistence agriculture. Around 1900 the number of population of white settlers increased and a diversion of the water happened. That had an impact of Pima's food intake and way of life. Pima Indians used to farm sustained through physical labour to a little labour and scarce of food. As a consequence they food intake became high in fat and their lifestyle was mainly sedentary. That resulted in development of diabetes among the Arizona Pimas, and it drawed attention as they had the highest recorded prevalence and incidence of type 2 diabetes (T2DM) of any geographically-defined population (SCHULZ et al, 2015). The Pima population has been under study by the National Institute of Diabetes and Digestive and Kidney Diseases at intervals of 2 years since 1965. As epidemiological evidence indicates that T2DM results from interaction of genetic and environmental factors, the Pima Indians Diabetes Dataset includes information about attributes that could and should be related to the onset of diabetes and its future complications.

TRICEPS SKIN FOLD THICKNESS

Chandra-Selvi at al (2016) defined adipose tissue as a *'loose connective tissue composed mainly of adipocytes that has an unlimited growth potential at any stage of life, excess of adipose tissue predisposes to many diseases-development of insulin resistance'*. Their study found that the skinfold thickness gradually decreased in diabetic patients as the duration of disease increased. However, the Pima Indians Diabetes Dataset does not provide information on the duration of the Diabetes. The researchers discussed the results and concluded that in diabetic patients, there is an efflux of free fatty acids from the adipose tissue, resulting in a decrease of skin fold thickness as the duration of disease increases (CHANDRA-SELVI at al, 2016). Neverthless, there is still very limited prospective information evaluating the relationship of skinfold thickness with incident T2DM (RUIZ-ALEJOS et al, 2020). Triceps skinfold thickness in millimeters for females aged 20 and over and number of examined persons, mean, standard error of the mean, and selected percentiles, by race and ethnicity and age: United States, 2007–2010 (FRYAR et al, 2012).

Race and ethnicity and age	Number of examined persons	Mean	Standard error of the mean	Percentile								
				5th	10th	15th	25th	50th	75th	85th	90th	95th
All racial and ethnic groups[1]						Millimeters						
20 years and over	5,198	23.7	0.17	11.9	14.0	15.8	18.0	23.5	29.0	31.9	33.7	35.9
20–29 years	815	22.2	0.44	11.0	13.2	14.0	16.5	21.3	27.7	31.1	32.6	35.1
30–39 years	858	24.2	0.40	12.1	14.7	16.2	18.6	24.1	30.0	32.2	33.8	36.0
40–49 years	943	24.2	0.26	11.5	14.1	15.9	18.7	24.7	29.5	32.1	33.9	35.8
50–59 years	769	24.6	0.30	12.9	15.6	16.8	19.2	24.5	29.8	32.2	34.0	36.3
60–69 years	845	24.7	0.34	13.0	15.8	17.2	20.2	24.9	29.4	31.9	33.8	35.9
70–79 years	613	23.4	0.33	11.5	14.3	16.3	18.9	22.9	28.0	31.0	32.8	35.2
80 years and over	355	19.4	0.34	9.4	11.2	12.4	14.7	19.0	23.2	25.9	27.8	31.7

OBESITY AND DIABETES

Obesity, assessed by Body Mass Index (BMI), is intimately associated with diabetes and its impact on the development of T2DM has been largely described in large cohort prospective studies (VAN GAAL and SCHEEN, 2015; WILDING, 2014; RUIZ-ALEJOS et al, 2020). In fact, most of the individuals with T2DM are overweight or obese [5]. Despite the link between obesity and T2DM not all obese develops diabetes and not all diabetics are obese people. Diabetic lean people probably have a stronger genetic component for T2DM than overweight and obese individuals (WILDING, 2014). Accordingly to Ruiz-Alejos and col (2020) 'there is no specific recommendation about the use of an anthropometric measurement as a marker for T2DM risk prediction besides BMI and waist circumference'. Unfortunately the waist circumference is a measurement not included in the Pima Indians Diabetes Dataset. BMI provides a simple, yet accurate method for indicating nutritional status in adults (as it can be seem in the Table below), and it can be calculated by dividing the individual's weight (in kg) by the square of their height (in metres).

$$\text{BMI} = \frac{\text{weight (kg)}}{(\text{height (m)})^2}$$

BMI	Nutritional status
Below 18.5	Underweight
18.5–24.9	Normal weight
25.0–29.9	Pre-obesity
30.0–34.9	Obesity class I
35.0–39.9	Obesity class II
Above 40	Obesity class III

Table. Nutritional Status. Source: World Health Organization.

OBJECTIVE

The objective of this project is:

1. Analyse the dataset under the point of view of a Dietitian.
2. Apply machine learning techniques resulting in bridging the gap between datasets and human knowledge.

```
#Load packages

import pandas as pd
import numpy as np
import seaborn as sns
import sklearn.model_selection as mod
import sklearn.neighbors as nei
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import KFold
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import LeaveOneOut
from sklearn import metrics
from sklearn.metrics import accuracy_score, confusion_r
from sklearn.naive_bayes import MultinomialNB
#from sklearn.cross_validation import cross_val_score
from sklearn.feature_selection import RFECV
from sklearn.svm import SVC
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import operator
%matplotlib inline

url="/content/diabetes.csv"

#Load CSV file using Pandas

pima = pd.read_csv(url)
```

Check the DataFrame & Descriptive Statistics and Analysis of Pima Indians Dataset

```
# Check dimension of the DataFrame

pima.shape
```

```
(768, 9)
```

```
# Check the type of 'pima'
type(pima)
```

```
pandas.core.frame.DataFrame
```

```
# Get row indices

pima_row_idx = pima.index
pima_row_idx
```

```
RangeIndex(start=0, stop=768, step=1)
```

```
# Get the column names

pima_col_idx = pima.columns
pima_col_idx
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

```
# Get data type for each attribute

pima.dtypes
```

```
Pregnancies                 int64
Glucose                     int64
BloodPressure               int64
SkinThickness             float64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
dtype: object
```

```
# Check the first 5 rows

pima.head (10)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
0	6	148	72	NaN	0	33.6	
1	1	85	66	29.0	0	26.6	
2	8	183	64	0.0	0	23.3	
3	1	89	66	23.0	94	28.1	
4	0	137	40	35.0	168	43.1	
5	5	116	74	0.0	0	25.6	
6	3	78	50	32.0	88	31.0	
7	10	115	0	0.0	0	35.3	
8	2	197	70	45.0	543	30.5	
9	8	125	96	0.0	0	0.0	

Check missing values

pima.isnull().sum()

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               1
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Create Nutritional status column

Nutritional_status = pd.Series([])

```
<ipython-input-106-1e90f2aaece1>:3: FutureWarning: The default dtype for empty Series will be 'object' instead of 'float64' in a future
  Nutritional_status = pd.Series([])
```

```
# Nutritional status based on BMI

for i in range(len(pima)):
    if pima['BMI'][i] == 0.0:
        Nutritional_status[i]="NA"

    elif pima['BMI'][i] < 18.5:
        Nutritional_status[i]="Underweight"

    elif pima['BMI'][i] < 25:
        Nutritional_status[i]="Normal"

    elif pima['BMI'][i] >= 25 and pima['BMI'][i] < 30:
        Nutritional_status[i]="Overweight"

    elif pima['BMI'][i] >= 30:
        Nutritional_status[i]="Obese"

    else:
        Nutritional_status[i]= pima['BMI'][i]

# Insert new column - Nutritional Status
pima.insert(6, "Nutritional Status", Nutritional_statu

# Check df containing new column
pima.head (5)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
0	6	148	72	NaN	0	33.6	
1	1	85	66	29.0	0	26.6	
2	8	183	64	0.0	0	23.3	
3	1	89	66	23.0	94	28.1	
4	0	137	40	35.0	168	43.1	

```
pima['Nutritional Status'].value_counts()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self,
key, method, tolerance)
   3801             try:
-> 3802                 return self._engine.get_loc(casted_key)
   3803             except KeyError as err:

  ---------------------------------- 4 frames ----------------------------------
pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Nutritional Status'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self,
key, method, tolerance)
   3802                 return self._engine.get_loc(casted_key)
   3803             except KeyError as err:
-> 3804                 raise KeyError(key) from err
   3805             except TypeError:
   3806                 # If we have a listlike key, _check_indexing_error will raise

KeyError: 'Nutritional Status'
```

11 women don't have information about BMI. Only 106 of 758 women have normal weight. Most of the women present overweight or obesity.

Create OGTT_Interpretation (Interpretation of Glucose

OGTT_Interpretation = pd.Series([])

```
<ipython-input-111-4879ae9795be>:3: FutureWarning: The default dtype for empty Series will be 'object' instead of 'float64' in a future
  OGTT_Interpretation = pd.Series([])
```

```
# Interpretation of OGTT (Glucose) - using OGTT levels

for i in range(len(pima)):
    if pima['Glucose'][i] == 0.0:
        OGTT_Interpretation [i]="NA"

    elif pima['Glucose'][i] <= 140:
        OGTT_Interpretation [i]="Normal"

    elif pima['Glucose'][i] > 140 & pima['Glucose'][i]
        OGTT_Interpretation [i]="Impaired Glucose Tole

    elif pima['Glucose'][i] > 198:
        OGTT_Interpretation[i]="Diabetic Level"

    else:
        OGTT_Interpretation [i]= pima['Glucose'][i]
```

```
# Insert new column - Glucose Result
pima.insert(2, "Glucose Result", OGTT_Interpretation)
```

```
pima['Glucose Result'].value_counts()
```

```
Normal                      571
Impaired Glucose Tolerance  192
NA                            5
Name: Glucose Result, dtype: int64
```

Not a single individual from the sample showed OGTT result at "Diabetic levels". Would be interesting to calculate HOMA-IR to get information about insulin resistance of the sample. ***DISCUSS HERE ABOUT GLUCOSE INTOLERANCE AND INSULIN RESISTANCE****

```
Impaired_Glucose_Tolerance_Diabetic = ((pima ['Glucose
Impaired_Glucose_Tolerance_Diabetic
```

```
131
```

Not every women with impaired glucose tolerance have diabetes. That can show that the ones with impaired glucose tolerance might be in risk of developing diabetes or are diabetic, but not already diagnosed.

```
Normal_Glucose_Diabetic = ((pima ['Glucose'] != 0 ) & (
Normal_Glucose_Diabetic
```

```
134
```

Half of the diabetic women showed normal glucose level. Information regarding drug therapy is needed.

```
# Create Percentile of skin thickness column

Percentile_skin_thickness = pd.Series([])
```

Show hidden output

```
# Check how many women are 80 or older (the Percentile

pima['Age'].value_counts()
```

```
22    72
21    63
25    48
24    46
23    38
28    35
26    33
27    32
29    29
31    24
41    22
30    21
37    19
42    18
33    17
38    16
36    16
32    16
45    15
34    14
46    13
43    13
40    13
39    12
35    10
50     8
51     8
52     8
44     8
58     7
47     6
54     6
49     5
48     5
57     5
53     5
60     5
66     4
63     4
62     4
55     4
67     3
56     3
59     3
65     3
69     2
61     2
72     1
81     1
64     1
70     1
68     1
Name: Age, dtype: int64
```

```
#  Check skin fold thickness Percentile

for i in range(len(pima)):


    if pima["Age"][i] >= 20.0 and pima["Age"][i] <= 79

        if pima["SkinThickness"][i] == 0.0:
            Percentile_skin_thickness[i]=" 0 NA"

        elif pima["SkinThickness"][i] < 11.9:
            Percentile_skin_thickness[i]="1 <P5th"

        elif pima["SkinThickness"][i] == 11.9:
            Percentile_skin_thickness[i]="2 P5th"

        elif pima["SkinThickness"][i] > 11.9 and pima['
            Percentile_skin_thickness[i]="3 P5th - P10t

        elif pima["SkinThickness"][i] == 14.0:
            Percentile_skin_thickness[i]="4 P10th"

        elif pima["SkinThickness"][i] > 14.0 and  pima[
            Percentile_skin_thickness[i]="5 P10th - P1!

        elif pima["SkinThickness"][i] == 15.8:
            Percentile_skin_thickness[i]="6 P15th"

        elif pima["SkinThickness"][i] > 15.8 and pima['
            Percentile_skin_thickness[i]="7 P15th - P2!

        elif pima["SkinThickness"][i] == 18.0:
            Percentile_skin_thickness[i]="8 P25th"

        elif pima["SkinThickness"][i] > 18.0 and pima['
```

```
            Percentile_skin_thickness[i]="9 P25th - P5(

        elif pima["SkinThickness"][i] == 23.5:
            Percentile_skin_thickness[i]="10 P50th"

        elif pima["SkinThickness"][i] > 23.5 and pima['
            Percentile_skin_thickness[i]="11 P50th - P7

        elif pima["SkinThickness"][i] == 29.0:
            Percentile_skin_thickness[i]="12 P75th"

        elif pima["SkinThickness"][i] > 29.0 and pima['
            Percentile_skin_thickness[i]="13 P75th - P8

        elif pima["SkinThickness"][i] == 31.9:
            Percentile_skin_thickness[i]="14 P85th"

        elif pima["SkinThickness"][i] > 31.9 and pima['
            Percentile_skin_thickness[i]="15 P85th - P9

        elif pima["SkinThickness"][i] == 33.7:
            Percentile_skin_thickness[i]="16 P90th"

        elif pima["SkinThickness"][i] > 33.7 and pima['
            Percentile_skin_thickness[i]="17 P90th - P9

        elif pima["SkinThickness"][i] == 35.9:
            Percentile_skin_thickness[i]="18 P95th"

        elif pima["SkinThickness"][i] > 35.9:
            Percentile_skin_thickness[i]="19 >P95th"

    elif pima["Age"][i] >= 80.0:  #Only 1 woman is 81 y
        if  pima["SkinThickness"][i] > 31.7:
```

```
Percentile_skin_thickness[i]="20 >P95th"
```

Note: As I have 2 >P95 I chose to add numbers in front of each percentile to easily visualize which >P95 it belonged.

```
# Insert new column - Percentile of skin thickness

pima.insert(4, "Percentile skin thickness", Percentile_

# Check the first 5 rows

pima.head(5)
```

	Pregnancies	Glucose	Glucose Result	BloodPressure	Percentile skin thickness	SkinThickness	Insulin	BM
0	6	148	Impaired Glucose Tolerance	72	NaN	NaN	0	33
1	1	85	Normal	66	12 P75th	29.0	0	26
			Impaired					

It looks like the Percentile of the triceps skin fold thickness can predict the Nutritional Status.

```
# Check number of women x Percentile of skin thickness

pima['Percentile skin thickness'].value_counts()
```

```
 0 NA              227
19 >P95th          145
11 P50th - P75th    87
9 P25th - P50th     79
15 P85th - P90th    50
13 P75th - P85th    46
17 P90th - P95th    22
8 P25th             20
7 P15th - P25th     20
3 P5th - P10th      18
12 P75th            17
1 <P5th             15
5 P10th - P15th     14
4 P10th              6
20 >P95th            1
Name: Percentile skin thickness, dtype: int64
```

227 women don't have information regarding skin fold. Most of the sample have Percentile of skin thickness greater than 95th. 53 women had value below P15 indicating that they might be malnourished. Only one woman is 80 years old or older and she presents a Percentile of skin thickness greater than 95th.

```
diabetic_malnourished_st = ((pima ['SkinThickness'] < 
diabetic_malnourished_st
```

```
94
```

```
diabetic_malnourished_bmi = ((pima ['BMI'] < 18.5) & (p
diabetic_malnourished_bmi
```

```
2
```

```
diabetic_malnourished_bmi_st = ((pima ['BMI'] < 18.5) &
diabetic_malnourished_bmi_st
```

```
2
```

Interesting to note that 94 diabetic women were classified as underweight/malnourished based only on Skin thickness, but only 2 diabetic women presented underweight/malnutrition when considering BMI or BMI & Skin Thickness. Skinfold thicknesses are difficult measurements to make with precision and accuracy without rigorous training, for that reason they are rarely used today and not universally applicable. Besides, skinfold thickness can be problematic to identify malnutrition in many individuals as they vary with diseases such as diabetes, menopause, and hydrational status. That way, more data should be included in this dataset to help to identify malnutrition using skinfold thickness measurement.

```
# Minimum

pima.min()
```

```
<ipython-input-67-77cb051e5797>:1: FutureWarning: The default value of numeric_only in DataFrame.min is deprecated. In a future version,
  pima.min()
Pregnancies                                        0
Glucose                                            0
Glucose Result             Impaired Glucose Tolerance
BloodPressure                                      0
SkinThickness                                    0.0
Insulin                                            0
BMI                                              0.0
Nutritional Status                               NaN
DiabetesPedigreeFunction                       0.078
Age                                               21
Outcome                                            0
dtype: object
```

It can be seem that only adult women were included as the minimum age is 21 years old. It can be noticed that a number of women don't have information of some of the attributes, such as glucose, blood pressure, etc as the minimum value is zero.

```
# Maximum

pima.max()
```

```
<ipython-input-68-8a467d058a68>:1: FutureWarning: The default value of numeric_only in DataFrame.max is deprecated. In a future version,
  pima.max()
Pregnancies                     17
Glucose                        199
Glucose Result              Normal
BloodPressure                  122
SkinThickness                 99.0
Insulin                        846
BMI                           67.1
Nutritional Status             NaN
DiabetesPedigreeFunction      2.42
Age                             81
Outcome                          1
dtype: object
```

The nutritional status 'underweight' should not be classified as Max Value in medical point of view. However, it was considered as a Max value as it starts with the letter U.

```
#Check if the sample were classified as Underweight pre

pima_underweight = pima[pima['Nutritional Status'] =='U
pima_underweight
```

Glucose Percentile

```
pima['Outcome'].value_counts()
```

```
0    500
1    268
Name: Outcome, dtype: int64
```

500 not diabetic and 268 are diabetic

```
# Another way of counting the outcome (diabetes)
count_not_diabetic = len(pima[pima['Outcome']==0])
count_not_diabetic
```

```
500
```

```
# Check the average of features grouped by Outcome (Dia

pima.groupby('Outcome').mean()
```

```
<ipython-input-72-26a675b3ae2e>:3: FutureWarning: The default value of numeric_only in D
  pima.groupby('Outcome').mean()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Nu
Outcome							
0	3.298000	109.980000	68.184000	19.664000	68.792000	30.304200	
1	4.865672	141.257463	70.824627	22.116105	100.335821	35.142537	

It looks like the average of all features is higher in diabetic women.

```
pima.mean()
```

```
<ipython-input-73-fb60b930057b>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version
  pima.mean()
Pregnancies                   3.845052
Glucose                     120.894531
BloodPressure                69.105469
SkinThickness                20.517601
Insulin                      79.799479
BMI                          31.992578
Nutritional Status                 NaN
DiabetesPedigreeFunction      0.471876
Age                          33.240885
Outcome                       0.348958
dtype: float64
```

As some of the women have no information of a number of attributes, such as glucose, blood pressure and BMI the average of those items might not be correct. To get the correct value only the women with a value > zero should be included to calculate the average.

```
# Shows women that contains information about Glucose

pima_glucose = pima.loc[pima['Glucose'] != 0]

pima_glucose.shape
```

```
(763, 12)
```

Most women of the sample have data regarding glucose (only 5 of them don't have any glucose value) as it is an important information to check the health status of a diabetic person.

```
# Check average of glucose from women that don't have a

pima_glucose['Glucose'].mean()
```

```
121.6867627785059
```

The average of glucose is at the normal range (less than 140 mg/dl).

```
pima_glucose.groupby('Outcome').mean()
```

```
<ipython-input-78-15367d43ad27>:1: FutureWarning: The default value of numeric_only in D
  pima_glucose.groupby('Outcome').mean()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Nu
Outcome							
0	3.311871	110.643863	68.213280	19.631791	69.160966	30.317304	
1	4.860902	142.319549	70.800752	22.007547	101.090226	35.106015	

```
# Check minimum and maximum values of glucose from wome

pima_glucose['Glucose'].min()
```

```
44
```

```
pima_glucose['Glucose'].max()
```

```
199
```

As it shows above, some women have information about glucose but no information about insulin. Moreover, it shows that 5 women don't have information about glucose as the result shows 763 rows. Furthermore, the maximum value of glucose is 199, considered a Diabetic level. Therefore, even diagnosed with diabetes they don't present high glucose value. The dataset should have information about drug therapy.

```
# Shows women that have information about Blood pressu

pima_BloodPressure = pima.loc[pima['BloodPressure'] !=

pima_BloodPressure.shape
```

```
(733, 12)
```

```
# Check the average of blood pressure (only from women

pima_BloodPressure['BloodPressure'].mean()
```

```
72.40518417462484
```

```
# Minimum and maximum of Blood Pressure from women tha

pima_BloodPressure['BloodPressure'].min()
```

```
24
```

```
pima_BloodPressure['BloodPressure'].max()
```

```
122
```

35 women don't have information about Blood Pressure as the result shows 733 rows. The maximum value of Diastolic Blood Pressure shows that there are a possibility of some women to have hypertension (>90 mmHg) as it can be diagnosed when the blood pressure is 140/90 mmHg or higher (NICE, 2019). Diabetic individuals should be treated to Diastolic blood pressure goal of less than 90 mmHg. It is recommended that blood pressure should be measured at least annually in an adult with type 2 diabetes without previously diagnosed hypertension or renal disease (NICE, 2019). Systolic Blood Pressure measure should have been included in this dataset as studies show that blood pressure equal or greater than 115/75 mmHg is associated with increased rates of several diseases, including heart failure, kidney disease and mortality (BOER et al, 2017). Looking at the minimum value of Diastolic Blood Pressure (24 mmHg) it is totally out of the normal range even when the heart is 'relaxing' (between 40 to 160 mmHg). Individual with Diastolic blood pressure less or equal to 60 mmHg has low blood pressure (BLOOD PRESSURE UK).

```
pima_insulin = pima.loc[pima['Insulin'] != 0]

pima_insulin.shape
```

```
(394, 12)
```

```
# Check average value of insulin from women that don't

pima_insulin['Insulin'].mean()
```

```
155.5482233502538
```

```
# Check minimum and maximum of Insulin value from wome
```

```
pima_insulin['Insulin'].min()
```

```
14
```

```
pima_insulin['Insulin'].max()
```

```
846
```

Normal range of 2h insulin is from 16 to 166 mIU/L. The average value of 2h insulin of the samples show a normal range. However, some women of the sample seem to present high level of insulin level. It might be due obesity, early stage of T2DM or excessive insulin administration.

```
# Check women that don't have zero value of BMI

pima_BMI = pima.loc[pima['BMI'] != 0]

pima_BMI.shape
```

```
(757, 12)
```

```
# Check average of BMI from women that don't have zero

pima_BMI['BMI'].mean()
```

```
32.457463672391015
```

The average value of BMI indicates obesity (BMI >= 30 kg/m2)

```
# Check minimun and maximum value of BMI from women tha

pima_BMI['BMI'].min()
```

```
18.199999999999999
```

```
pima_BMI['BMI'].max()
```

```
67.099999999999994
```

The minimum value of BMI shows that there is no case of underweight, but the maximum value shows cases of women with morbid obesity (BMI >= 40kg/m2).

```
# Check only the women that have all the values of BMI,

pima_all = pima.loc[(pima['BMI'] != 0) & (pima['Insulin

pima_all.shape
```

```
(392, 12)
```

Only 392 women have information about all the attributes. That number represents less than half of the sample (around 49% of the women of the sample have all information of all attributes).

pima_all['Outcome'].value_counts()

```
0    262
1    130
Name: Outcome, dtype: int64
```

262 women don't have DM and 130 have DM.

pima_all.mean()

```
Pregnancies                   3.301020
Glucose                     122.627551
BloodPressure                70.663265
SkinThickness                29.145408
Insulin                     156.056122
BMI                          33.086224
DiabetesPedigreeFunction      0.523046
Age                          30.864796
Outcome                       0.331633
dtype: float64
```

pima_all.groupby('Outcome').mean()

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Di
Outcome							
0	2.721374	111.431298	68.969466	27.251908	130.854962	31.750763	
1	4.469231	145.192308	74.076923	32.961538	206.846154	35.777692	

Diabetic women tend to have higher number of pregnancies, higher level of glucose, higher Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function and Age. Both groups present BMI that indicates obesity. Women that have Diabetes have an average of insulin that is higher than the normal range (16 - 166 mIU/L). Both groups have average of glucose higher than the normal range (<= 100 mg/dL). It might indicate that some non diabetic women are in risk of have Diabetes in the future, specially the ones with higher levels of insulin (that might have insulin resistance).

pima_all.min()

```
Pregnancies                                           0
Glucose                                              56
Glucose Result               Impaired Glucose Tolerance
BloodPressure                                        24
Percentile skin thickness                      1 <P5th
SkinThickness                                         7
Insulin                                              14
BMI                                                18.2
Nutritional Status                               Normal
DiabetesPedigreeFunction                          0.085
Age                                                  21
Outcome                                               0
dtype: object
```

pima_all.max()

```
Pregnancies                              17
Glucose                                 198
Glucose Result                       Normal
BloodPressure                           110
Percentile skin thickness    9 P25th - P50th
SkinThickness                            63
Insulin                                 846
BMI                                    67.1
```

```
Nutritional Status              Underweight
DiabetesPedigreeFunction               2.42
Age                                      81
Outcome                                   1
dtype: object
```

```
diabetic_malnourished = ((pima_all ['SkinThickness'] <
diabetic_malnourished
```

```
4
```

4 from 130 Diabetic women are malnourished.

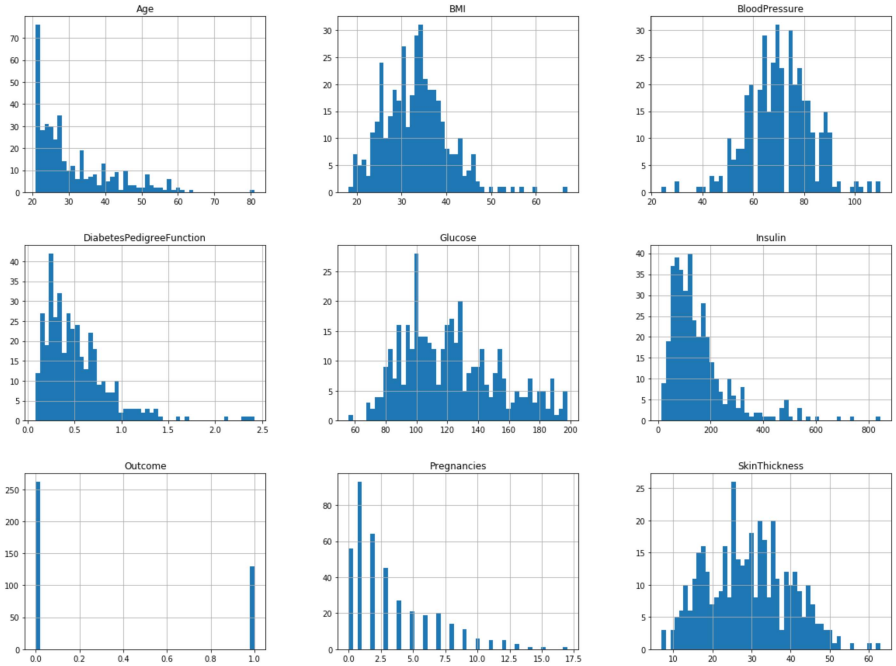
```
diabetic_overweight_obese = ((pima_all ['BMI'] <= 30) &
diabetic_overweight_obese
```

```
20
```

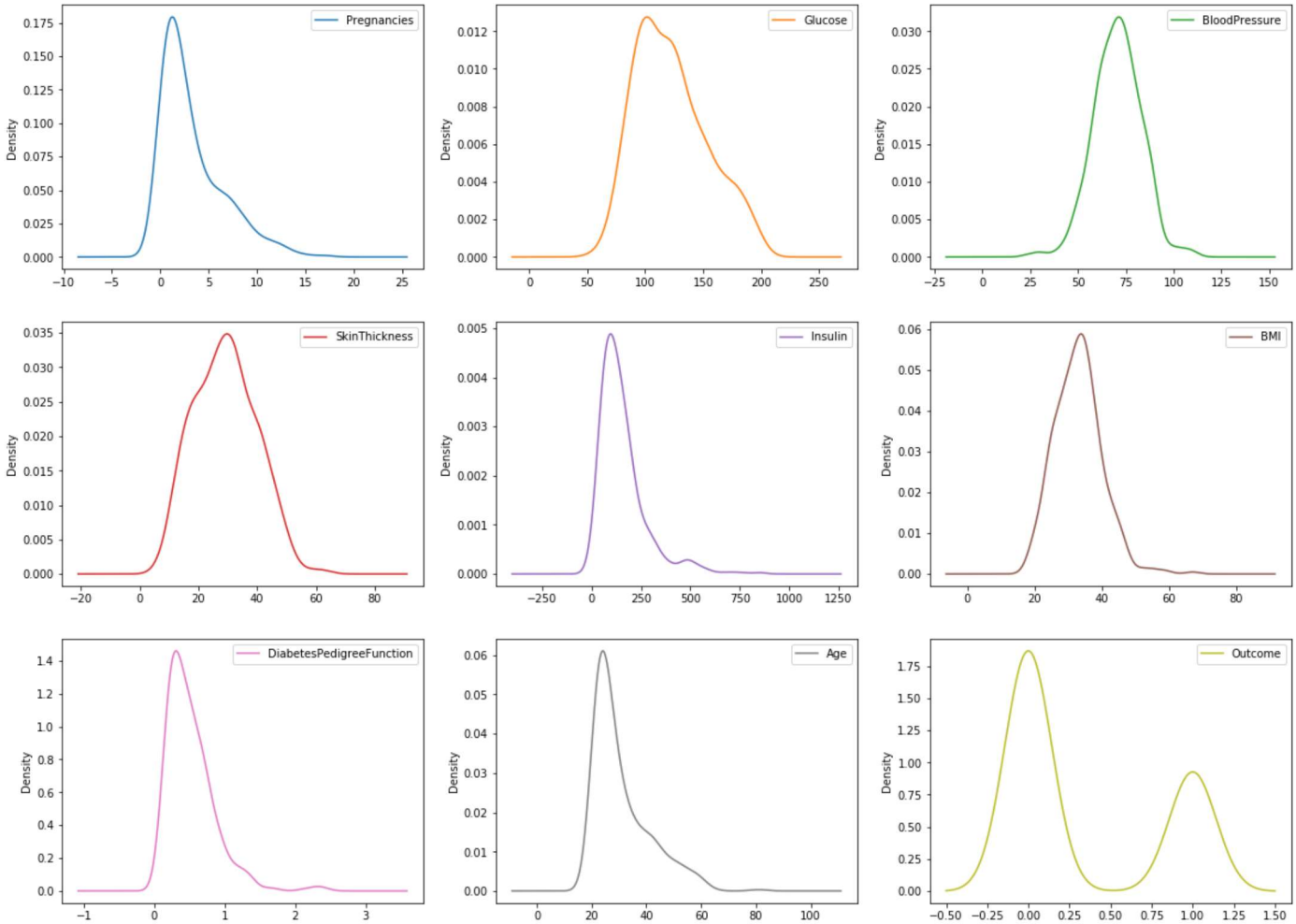
20 from 130 Diabetic women are overweight or obese.

Attributes Distribution

```
# Histogram
pima_all.hist(bins=50, figsize=(20, 15))
plt.show()
```

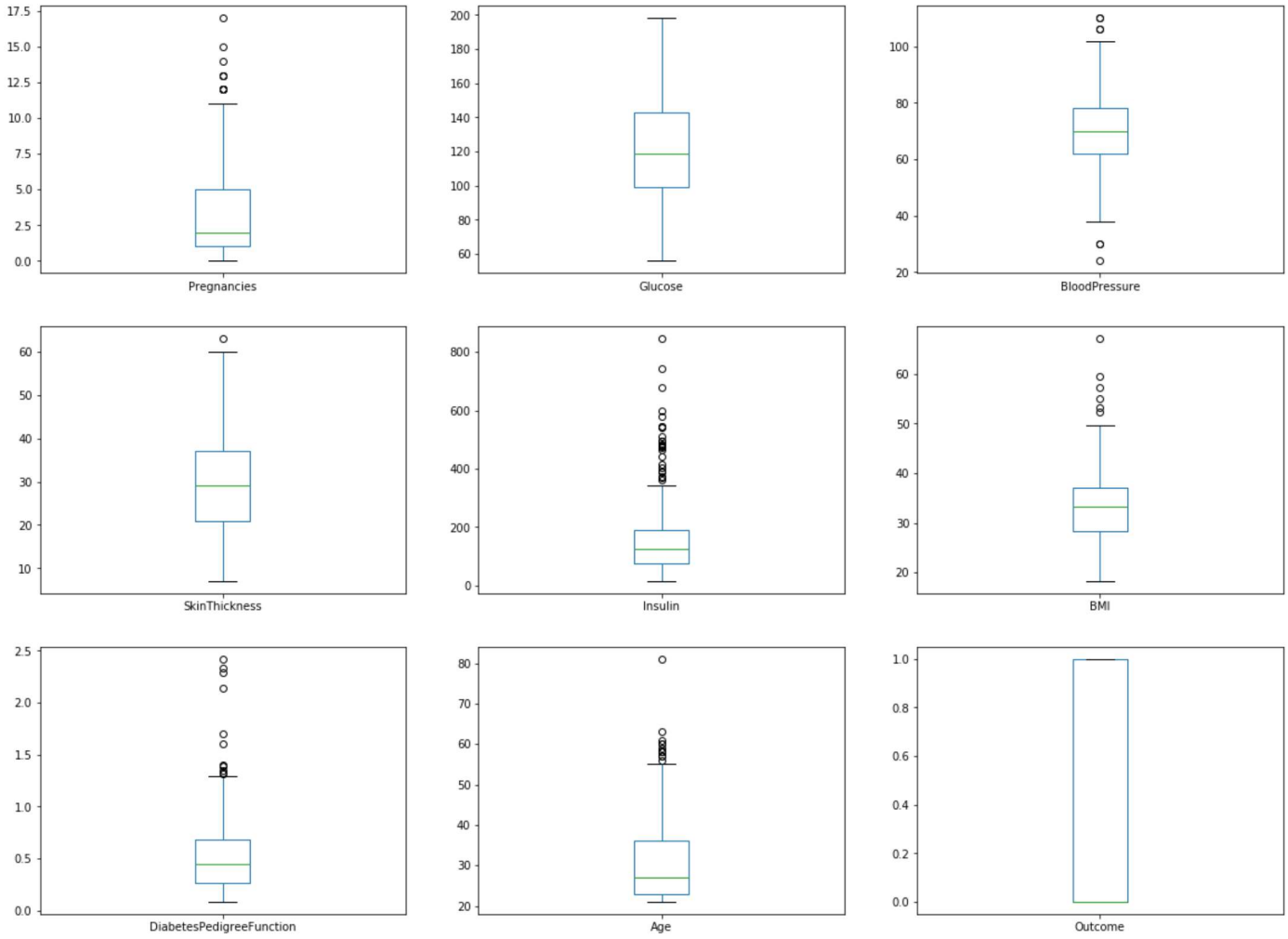



```
# Density plots for all attributes to visualize the dis
pima_all.plot(kind='density', subplots=True, layout=(3,
plt.show()
```




```
#Box and Whisker plot to visualize the distribution of
pima_all.plot(kind= 'box' , subplots=True, layout=(3,3
```

```
Pregnancies                   AxesSubplot(0.125,0.657941;0.227941x0.222059)
Glucose                     AxesSubplot(0.398529,0.657941;0.227941x0.222059)
BloodPressure               AxesSubplot(0.672059,0.657941;0.227941x0.222059)
SkinThickness                 AxesSubplot(0.125,0.391471;0.227941x0.222059)
Insulin                     AxesSubplot(0.398529,0.391471;0.227941x0.222059)
BMI                         AxesSubplot(0.672059,0.391471;0.227941x0.222059)
DiabetesPedigreeFunction          AxesSubplot(0.125,0.125;0.227941x0.222059)
Age                           AxesSubplot(0.398529,0.125;0.227941x0.222059)
Outcome                       AxesSubplot(0.672059,0.125;0.227941x0.222059)
dtype: object
```



#Skew of attributes distributions
skew=pima_all.skew(axis = 1)

Bell shape curve: Blood Pressure

Right-Skewed: Age, Insulin, Pregnancies, Diabetes Pedigree Function

Short IQR: insulin, Diabetes Pedigree Function, Blood Pressure and BMI

At least 75% of the women:

- are 25 years old or older
- have BMI nearly 30 kg/m2
- have insulin level 100 or more
- have 1 or more pregnancies

- have glucose level of 100 mg/dL or more
- have blood pressure of 60 mmHg or more

Correlation

```
# Correlation between the different characteristics. C
```

```
corr_matrix_pearson = pima_all.corr(method='pearson')
corr_matrix_pearson
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Out
Pregnancies	1.000000	0.198291	0.213355	0.093209	0.078984	-0.025347	0.007562	0.679608	0.25
Glucose	0.198291	1.000000	0.210027	0.198856	0.581223	0.209516	0.140180	0.343641	0.51
BloodPressure	0.213355	0.210027	1.000000	0.232571	0.098512	0.304403	-0.015971	0.300039	0.19
SkinThickness	0.093209	0.198856	0.232571	1.000000	0.182199	0.664355	0.160499	0.167761	0.25
Insulin	0.078984	0.581223	0.098512	0.182199	1.000000	0.226397	0.135906	0.217082	0.30
BMI	-0.025347	0.209516	0.304403	0.664355	0.226397	1.000000	0.158771	0.069814	0.27
DiabetesPedigreeFunction	0.007562	0.140180	-0.015971	0.160499	0.135906	0.158771	1.000000	0.085029	0.20
Age	0.679608	0.343641	0.300039	0.167761	0.217082	0.069814	0.085029	1.000000	0.35
Outcome	0.256566	0.515703	0.192673	0.255936	0.301429	0.270118	0.209330	0.350804	1.00

There are no strong correlation between the features. The 'strongest' ones are the following (as expected):

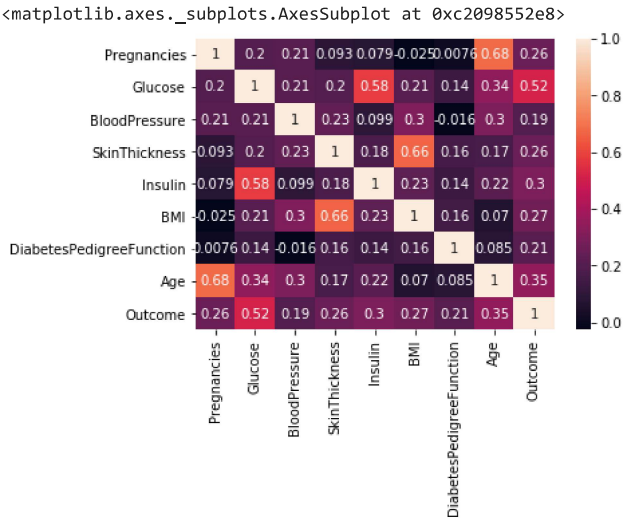
- Age x pregnancies (0.68) - Older women tend to have higher number of pregnancies
- Glucose x insulin (0.58)
- Glucose x outcome (0.52) - Women that have higher level of glucose tend to have higher level of insulin and have DM
- Skin fold thickness x BMI (0.66) - Women with higher skin fold thickness value have higher BMI (and probably are overweight/obese)

Negative correlation:

- BMI x Pregnancies (-0.025)
- Blood Pressure x Diabetes Pedigree Function (-0.016)

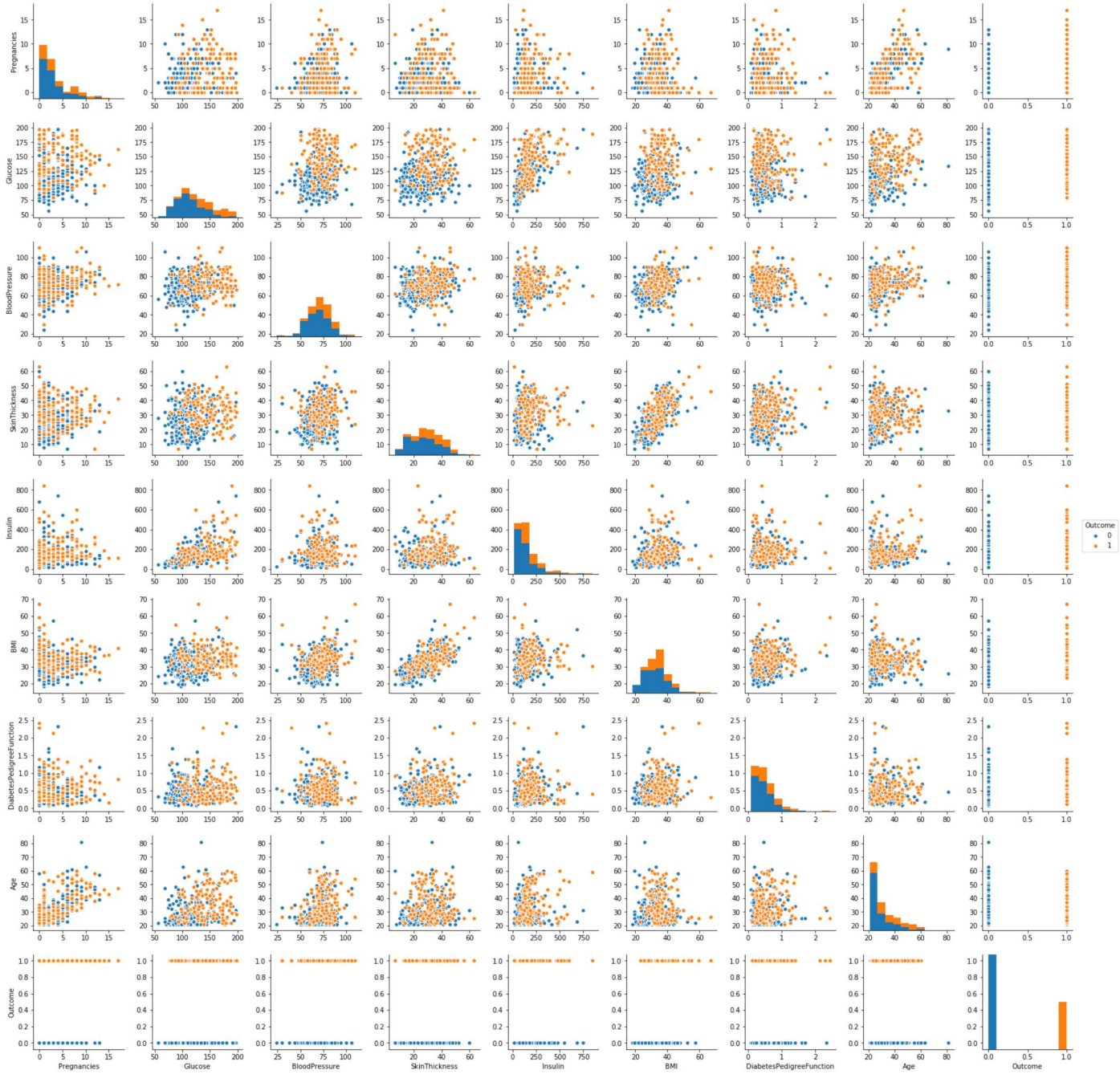
```
sns.heatmap(corr_matrix_pearson, annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0xc2098552e8>
```



```
# Pairplot

sns.pairplot(pima_all, hue='Outcome')
```

```
<seaborn.axisgrid.PairGrid at 0xc209276b70>
```



Diabetic women tend to show larger values of age, BMI, insulin, skin thickness, blood pressure, and pregnancies.

The feature that it is possible to see 2 distinct groups (diabetic and non diabetic) is glucose.

https://www.collaborat.com/pima-diabetes-data-discovery-predictive-model/

Logic Regression

```
#1st Iteration - 8 variables
var1=["Pregnancies", "Glucose","BloodPressure","SkinTh
X=pima_all[var1]
y=pima_all.Outcome

## Defining the model and assigning Y (Dependent) and 
logit_model=sm.Logit(y,X)

## Fitting the model and publishing the results
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.563677
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:                Outcome   No. Observations:                  392
Model:                          Logit   Df Residuals:                      384
Method:                           MLE   Df Model:                            7
Date:                Mon, 14 Dec 2020   Pseudo R-squ.:                  0.1128
Time:                        16:22:42   Log-Likelihood:                -220.96
converged:                       True   LL-Null:                       -249.05
                                        LLR p-value:                 8.717e-10
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
Pregnancies                  0.1299      0.049      2.655      0.008       0.034       0.226
Glucose                      0.0174      0.005      3.765      0.000       0.008       0.026
BloodPressure               -0.0484      0.009     -5.123      0.000      -0.067      -0.030
SkinThickness                0.0284      0.015      1.898      0.058      -0.001       0.058
Insulin                      0.0019      0.001      1.598      0.110      -0.000       0.004
BMI                         -0.0365      0.022     -1.669      0.095      -0.079       0.006
DiabetesPedigreeFunction     0.4636      0.344      1.347      0.178      -0.211       1.138
Age                          0.0005      0.016      0.031      0.976      -0.031       0.032
==============================================================================================
```

This model can explain 56% of the variation in dependent variable. SkinThickness, BMI, Diabetes Pedigree Function, Age, and Insulin are attributes to be eliminated in the next model. as variables that influence the outcome should have a p-value less than 0.05.

```
#2nd Iteration - 3 variables
var2=["Pregnancies", "Glucose","BloodPressure"]
X=pima_all[var2]
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.574607
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:                Outcome   No. Observations:                  392
Model:                          Logit   Df Residuals:                      389
Method:                           MLE   Df Model:                            2
Date:                Mon, 14 Dec 2020   Pseudo R-squ.:                 0.09558
Time:                        16:22:42   Log-Likelihood:                -225.25
converged:                       True   LL-Null:                       -249.05
                                        LLR p-value:                 4.597e-11
===================================================================================
                      coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------
Pregnancies         0.1405      0.037      3.826      0.000       0.069       0.212
Glucose             0.0210      0.004      5.709      0.000       0.014       0.028
BloodPressure      -0.0525      0.007     -7.449      0.000      -0.066      -0.039
===================================================================================
```

This model can explain 57% of the variation in dependent variable. All the variables show a p-value < 0.005.

```
logreg = LogisticRegression()
var2=["Pregnancies", "Glucose","BloodPressure"]
X=pima_all[var2]
y=pima_all.Outcome
logreg.fit(X,y)

## Defining the y_pred variable for the predicting valu
y_pred=logreg.predict(X)

## Calculating the precision of the model
print(classification_report(y,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.79      0.89      0.84       262
           1       0.71      0.53      0.61       130

avg / total       0.77      0.77      0.76       392
```

The model precision of model #2 (3 variables) is 77%.

```
## Confusion matrix gives the number of cases that the
confusion_matrix = confusion_matrix(y, y_pred)
print(confusion_matrix)
```

```
[[234  28]
 [ 61  69]]
```

The confusion matrix shows that 234 + 69 are correct predictions and 61 + 28 are incorrect predictions.

Classifier

```
# Predict the outcome - knn

inputs = pima_all[['Glucose', 'Insulin', 'SkinThicknes

outputs = pima_all['Outcome']

knn = nei.KNeighborsClassifier(n_neighbors=5)

knn.fit(inputs, outputs)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
           weights='uniform')
```

```
# Evaluate knn

(knn.predict(inputs) == outputs).sum()
```

```
322
```

From 392, 322 got correct outcome predicted.

```
#Split the train set - 1/3 is to test

inputs_train, inputs_test, outputs_train, outputs_test

knn = nei.KNeighborsClassifier(n_neighbors=5)
knn.fit(inputs_train, outputs_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
           weights='uniform')
```

```
(knn.predict(inputs_test) == outputs_test).sum()
```

```
99
```

```
#Count how many women are Positive (1) and Negative (0)

outputs_test.value_counts()
```

```
0    94
1    36
Name: Outcome, dtype: int64
```

Check classification accuracy with knn = 5

```
outputs_pred = knn.predict(inputs_test)
accuracy = metrics.accuracy_score(outputs_test, outputs
accuracy
```

```
0.7615384615384615
```

```
# 10-fold cross-validation with knn = 5

scores = cross_val_score (knn, inputs, outputs, cv =10,
scores
```

```
array([ 0.725     ,  0.8       ,  0.64102564,  0.61538462,  0.64102564,
        0.74358974,  0.66666667,  0.82051282,  0.76923077,  0.79487179])
```

```
scores.mean()
```

```
0.72173076923076918
```

Optimal value of k

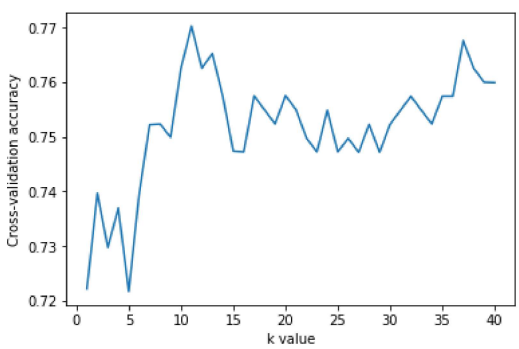
```
k_range = range(1,41)
k_scores = []

for k in k_range:
    knn = nei.KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score (knn, inputs, outputs, cv
    k_scores.append(scores.mean())

k_scores
```

```
[0.72224358974358971,
 0.73967948717948717,
 0.72974358974358966,
 0.73698717948717951,
 0.72173076923076918,
 0.73961538461538456,
 0.75224358974358974,
 0.75237179487179484,
 0.74993589743589739,
```

```
 0.76256410256410256,
 0.77025641025641023,
 0.76256410256410256,
 0.76525641025641022,
 0.75743589743589745,
 0.74737179487179484,
 0.74724358974358973,
 0.75749999999999995,
 0.7549358974358974,
 0.75237179487179495,
 0.75756410256410256,
 0.755,
 0.74980769230769229,
 0.74724358974358973,
 0.7548717948717949,
 0.74724358974358984,
 0.74974358974358979,
 0.74717948717948723,
 0.75230769230769234,
 0.74717948717948723,
 0.75230769230769234,
 0.7548717948717949,
 0.75743589743589745,
 0.7548717948717949,
 0.75237179487179484,
 0.75743589743589745,
 0.75743589743589745,
 0.76762820512820507,
 0.76256410256410256,
 0.76000000000000001,
 0.7599358974358974]
```

```
#Visualise best k number

plt.plot(k_range, k_scores)
plt.xlabel ('k value')
plt.ylabel('Cross-validation accuracy')
```

```
Text(0,0.5,'Cross-validation accuracy')
```




```
# Optimal value for k is 10

knn = nei.KNeighborsClassifier(n_neighbors=10)
knn.fit(inputs_train, outputs_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=10, p=2,
           weights='uniform')
```

```
(knn.predict(inputs_test) == outputs_test).sum()
```

```
103
```

```
outputs_pred = knn.predict(inputs_test)
accuracy = metrics.accuracy_score(outputs_test, output
accuracy #76.1%
```

```
0.79230769230769227
```

```
# 10-fold cross-validation with knn = 3

scores = cross_val_score (knn, inputs, outputs, cv =10
scores #accuracy scores arrays (from 61.5% to 84.6%)
```

```
array([ 0.8       ,  0.8       ,  0.69230769,  0.61538462,  0.69230769,
        0.84615385,  0.71794872,  0.87179487,  0.79487179,  0.79487179])
```

```
# Average accuracy score

scores.mean()
```

```
0.76256410256410256
```

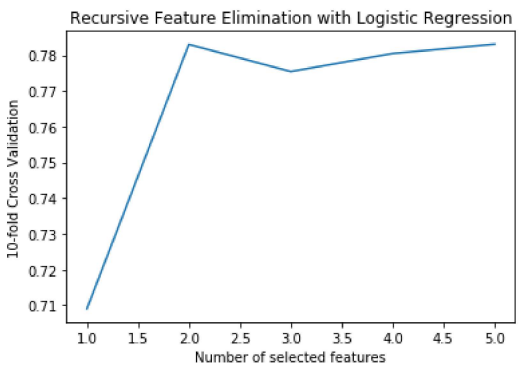
Cross Validation

Recursive feature elimination with cross-validation

```
kf = KFold(10, False, 1)

skf = StratifiedKFold(n_splits=10, random_state=42)

lg = LogisticRegression()

rfecv = RFECV (estimator=lg,step=1, cv=skf, scoring='a

rfecv.fit(inputs, outputs)
```

```
RFECV(cv=StratifiedKFold(n_splits=10, random_state=42, shuffle=False),
   estimator=LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False),
   n_jobs=1, scoring='accuracy', step=1, verbose=0)
```

```
plt.figure()
plt.title('Recursive Feature Elimination with Logistic
plt.xlabel('Number of selected features')
plt.ylabel('10-fold Cross Validation')
plt.plot(range(1, len(rfecv.grid_scores_) + 1), rfecv.g

plt.show()

#Optical number of features (=5) more suitable to pred:
```




```
feature_names = pima_all.columns[:10]
feature_names
```

```
Index(['Pregnancies', 'Glucose', 'Glucose Result', 'BloodPressure',
       'Percentile skin thickness', 'SkinThickness', 'Insulin', 'BMI',
       'Nutritional Status', 'DiabetesPedigreeFunction'],
      dtype='object')
```

```
X = pima_all[feature_names]

new_features = list(filter(lambda x: x[1],zip(feature_n
new_features
```

```
[('Pregnancies', True),
 ('Glucose', True),
 ('Glucose Result', True),
 ('BloodPressure', True),
 ('Percentile skin thickness', True)]
```

new_features are the features more suitable to predict the outcome (Diabetes)

Knn and Accuracy score - new features (inputs) and optimal k value

```
new_inputs = pima_all[['Pregnancies','Glucose', 'BloodP

knn = nei.KNeighborsClassifier(n_neighbors=10)
```