Diagnostically predict having diabetes. Based on independent body data.

Nanayakkara H.K.[#1], Amarasiri YHS[#3]

Department Of Computer science Engineering

University of Moratuwa

***Abstract*—Diabetes is a disease in which glucose level or sugar level of the blood becomes too high in the human body. Glucose comes from the foods that they eat. Insulin is the hormone that allows glucose to get into human cells in order to give them energy. Having a high glucose level in the human body can cause serious problems. It can affect human nerves, kidneys, eyes and may cause heart diseases, stroke and even need to remove a limb.**

***Keywords*— Include at least 5 keywords or phrases**

I. INTRODUCTION

Pregnant women can also be having the possibility of getting diabetes, called gestational diabetes. During the pregnancy period of women, the placenta makes hormones that can lead to a buildup of glucose in human blood. Usually, the pancreas can make enough insulin to handle that. If not, blood sugar levels will rise and can cause gestational diabetes [1].

The dataset was originally collected from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of this study is to diagnostically predict whether or not a female patient has diabetes based on glucose, blood pressure, skin thickness, insulin, age, BMI and no of pregnancies.

Analysing the dataset, we have obtained the data which we analyse here contain Metric Continuous and Metric Discrete .

How to analyze diabetes data and identify the various patterns to predict the occurrence of diabetes using machine learning techniques?

II. BACKGROUND

Variable Name	**Variable Description**	**Data Type**
Pregnancies	Number of times pregnant	Int64
Glucose	Plasma glucose concentration 2 hours in an oral glucose tolerance test	Int64
Blood Pressure	Diastolic blood pressure (mm Hg)	Int64
Skin Thickness	Triceps skinfold thickness (mm)	Int64
Insulin	2-Hour serum insulin (mu U/ml)	Int64
BMI	Body mass index (weight in kg/(height in m)^2)	float64
Diabetes Pedigree Function	Diabetes pedigree function	float64
Age	Women Age (years)	Int64
Outcome	Class variable (1: tested positive for diabetes, 0: tested negative for diabetes) 268 of 768 are 1, the others are 0	Int64

The dataset contains Seven hundred sixty-eight data rows and ten attributes including several medical predictor (independent) variables and one class (dependent) variable, Outcome. All patient's data in the dataset are females at least 21 years old of Pima Indian heritage [2].

Problem Statement we intend to focus here is how to analyze diabetes data and identify the various patterns to predict the occurrence of diabetes using machine learning techniques.

III. Methodology

We have used the python packages which required libraries and import the diabetes dataset to the Jupyter notebook.

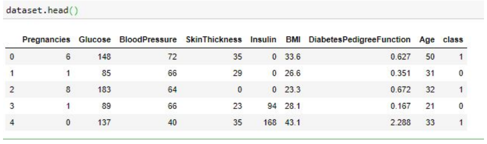
Pandas Library: Python’s pandas’ is an open source mostly used library for data analysis and it makes manipulation of data (importing, cleaning, transforming, analyzing, and visualizing) much easier [3].

NumPy: NumPy is a general-purpose array processing package which provides a high performance multidimensional array object, and tools for working with these arrays.

The dataset (diabetes.csv) file is updated by removing the column names and the path was given to the python code in order to import the dataset to the program. Then column names are given for the data as 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'Age', 'DiabetesPedigreeFunction', 'class'.

This step is important to familiarize with the dataset collected, in order to gain some understanding of the potential features and to see if data cleaning is needed.

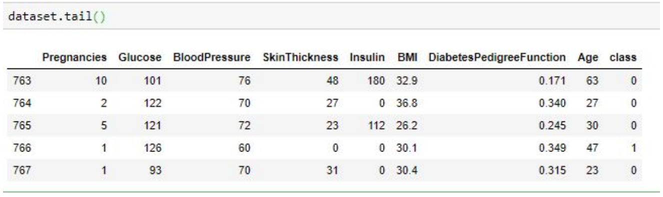
A. First Five Rows of The Dataset



```
dataset.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	class
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

B. Last Five Rows of The Dataset

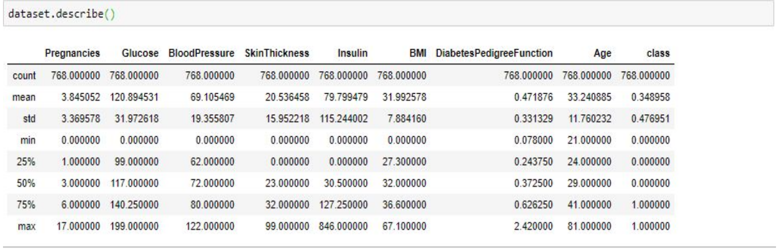


```
dataset.tail()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	class
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

As the Shape of The Dataset we can see, the dataset consists of 768 rows and 9 columns. ‘class’ is the column that is going to predict, which indicates whether the patient is having diabetic or not. 0 means the person is not having the diabetics and 1 means the person is having the diabetic. Based on the dataset statistics we can identify that out of the 768 persons, 500 are labeled as 0 (non-diabetic) and 268 as 1 (diabetic).

C. Descriptive analysis

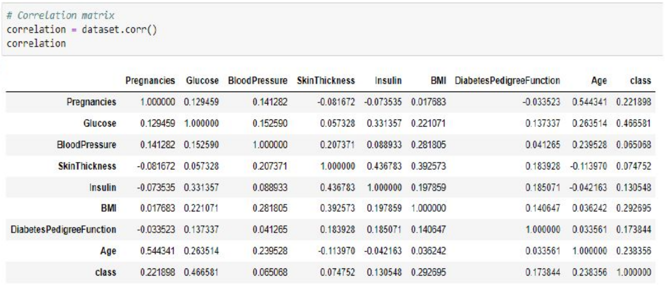


```
dataset.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	class
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

The above figure shows the descriptive statistics of the whole dataset which summarize the central tendency, shape and description of a dataset’s distribution, excluding all null values.

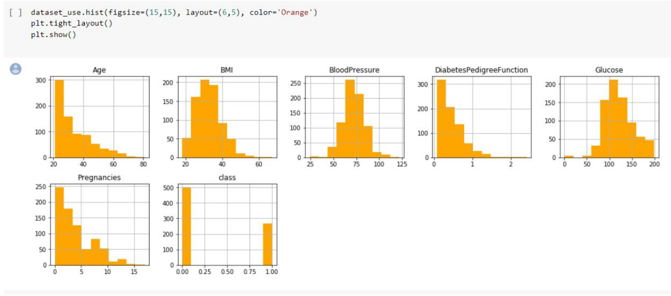
Correlation Matrix of The Dataset The correlation matrix is an important fact that helps to understand the correlation between the different parameters in the dataset. The values range from -1 to 1 and the closer a value is to 1 the better correlation there is between two characteristics [6].



```
# Correlation matrix
correlation = dataset.corr()
correlation
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	class
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
class	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

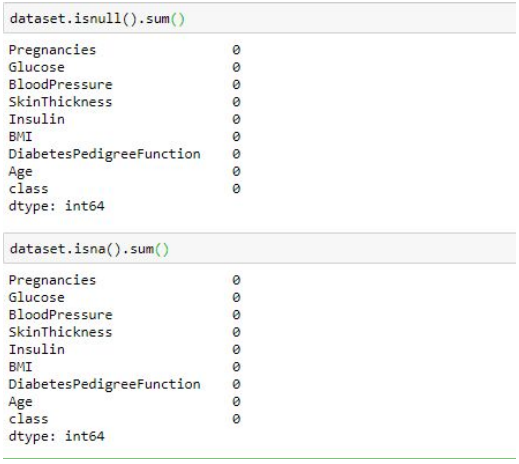
In order to consider Data Distribution of Each Feature In The Dataset, We can focus on below histograms, there are people with zero values for some variables, which cannot be possible. Usually, Blood Pressure, Insulin, BMI and Glucose level of a living person cannot have 0 values. Therefore, it indicates that this dataset should be pre-processed (Data Cleaning) before training the machine learning models.



```
[ ]  dataset_use.hist(figsize=(15,15), layout=(6,5), color='Orange')
     plt.tight_layout()
     plt.show()
```

D. Data Cleaning (Pre Processing)

Check whether there are any missing or null data points in the dataset.



```
dataset.isnull().sum()

Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
class                       0
dtype: int64

dataset.isna().sum()

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

According to above results we can observe that columns with “Zero” values: ‘BMI’, ‘insulin’, ‘blood pressure’, ‘Glucose level’ There are many ways to clean those data,

1. Remove or eliminate these data:

The easiest solution for this problem is, eliminating all those patients who are having zero values for the above columns. But it is not usually possible, because it will lose a lot of important data.

2. Calculate the median value for a specific column:

This might work for this diabetes dataset. Thus, the median value can be calculated for each column by considering the other data point in the specific column.We have used this method for our calculations.

As the next step, the dataset is divided into two parts as independent variables (features) and dependent variables (class variables) which will be predicted. The first eight columns of the dataset will be used as independent variables (features) and it is indicated using variable X_Data. The last column “class” is the dependent variable and it is indicated using the Y_Data variable. For this purpose, we have used python slice to select the columns in the NumPy array.

The loaded dataset is split into two sets as, 80% of data (training dataset) will be used to train the machine learning models and 20% of data (testing dataset) will be used to validate the accuracy of the models developed [8].

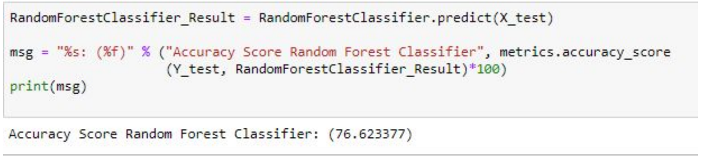
E. Predictive analysis

Initially, it is difficult to select a classification algorithm that suited the selected dataset. Therefore, evaluating multiple classification techniques will help to choose the algorithm that fits for this problem. Five algorithms will be evaluated in order to find the best model.

Random Forest Classifier.
K-Nearest Neighbors (KNN).
Logistic Regression (LR)
Linear Discriminant Analysis (LDA)
Classification and Regression Trees (CART).

Using the above-mentioned algorithms with their default parameters, the dataset will be trained, and the accuracy of each model will be considered in order to determine which model performs better with this diabetes dataset.
For this purpose, relevant libraries should be imported to the notebook.
Sklearn Library: Scikit-learn is a free machine learning library which can be used in Python programming language. It includes various clustering, classification and regression machine learning algorithms including gradient boosting, Logistic regression, k-means, support vector machines (SVM), random forests etc. Also, this library is designed to interoperate with the Python numerical and scientific libraries such as NumPy and SciPy [9].

1) *Random Forest Classifier*

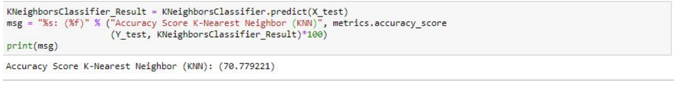


```
RandomForestClassifier_Result = RandomForestClassifier.predict(X_test)

msg = "%s: (%f)" % ("Accuracy Score Random Forest Classifier", metrics.accuracy_score
                    (Y_test, RandomForestClassifier_Result)*100)
print(msg)
```

```
Accuracy Score Random Forest Classifier: (76.623377)
```

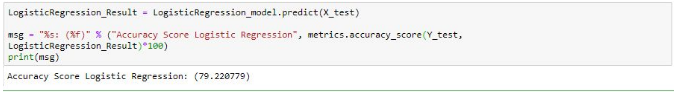
2) K-Nearest Neighbors (KNN).



```
KNeighborsClassifier_Result = KNeighborsClassifier.predict(X_test)
msg = "%s: (%f)" % ("Accuracy Score K-Nearest Neighbor (KNN)", metrics.accuracy_score
                    (Y_test, KNeighborsClassifier_Result)*100)
print(msg)
```

```
Accuracy Score K-Nearest Neighbor (KNN): (70.779221)
```

3) Logistic Regression (LR)

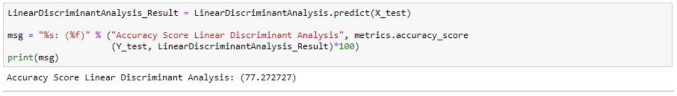


```
LogisticRegression_Result = LogisticRegression_model.predict(X_test)

msg = "%s: (%f)" % ("Accuracy Score Logistic Regression", metrics.accuracy_score(Y_test,
LogisticRegression_Result)*100)
print(msg)
```

```
Accuracy Score Logistic Regression: (79.220779)
```

4) Linear Discriminant Analysis (LDA)

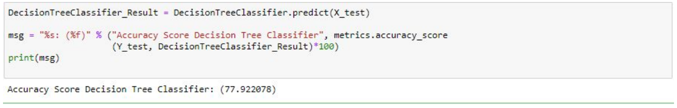


```
LinearDiscriminantAnalysis_Result = LinearDiscriminantAnalysis.predict(X_test)

msg = "%s: (%f)" % ("Accuracy Score Linear Discriminant Analysis", metrics.accuracy_score
                    (Y_test, LinearDiscriminantAnalysis_Result)*100)
print(msg)
```

```
Accuracy Score Linear Discriminant Analysis: (77.272727)
```

5) Classification and Regression Trees (CART).



```
DecisionTreeClassifier_Result = DecisionTreeClassifier.predict(X_test)

msg = "%s: (%f)" % ("Accuracy Score Decision Tree Classifier", metrics.accuracy_score
                    (Y_test, DecisionTreeClassifier_Result)*100)
print(msg)
```

```
Accuracy Score Decision Tree Classifier: (77.922078)
```

Accuracy Comparison For Each Model

Machine Learning Model	Accuracy (%)
Random Forest Classifier	77
K-Nearest Neighbors (KNN)	70
Logistic Regression (LR)	79
Linear Discriminant Analysis (LDA)	77
Classification and Regression Trees (CART)	75

IV. Conclusions

As future works, Dataset can be updated with more data records and features, in order to improve the accuracy of the learning model. Also, we can apply the ensemble method, which means we can simply combine the result of multiple machine learning models to produce better results. Considering diabetes, there are two levels (Type I and Type II) of diabetes available. As the next step, a model can be developed to predict the level of diabetes also.