

Diabetes Prediction Using Deep Learning with Python

The Ultimate Guide for building Diabetes Classifier and Predicting Possibility of having Diabetes in a Patient by using Neural Networks



**Blog By: Chaithanya Vamshi Sai**

**Student Id: 21152797**

# **Introduction**

Diabetes is a disease that occurs when the blood glucose is too high. Blood glucose is the main source of energy and comes from the food we eat.

Over time, having too much glucose in the blood can cause health problems and can lead to death. Preventing Diabetes is important and with advancements in AI, predicting the possibility of having Diabetes allows a person to take steps to manage diabetes and treat such situations at early stages making sure that more people can live healthy lives.

In this blog, I will demonstrate and show how we can harness the power of Deep Learning and apply it in healthcare. I will walk you through the entire process of how to classify and predict the possibility of having diabetes in a person using Neural Networks with Python.

# **Problem Statement**

The objective of this task is we are given a data set of patient medical records of Pima Indians and whether they had an onset of diabetes within five years.

Using these data, we will build an effective and optimised Binary Classifier and Predict the possibility of Diabetes in a person using GridSearchCV with Neural Networks and make predictions on test data.

# **Dataset Description**

The dataset comprises of 2 .csv files which are named “train.csv” and “test.csv”.

1. Train.csv: Training data to use with Model Training and Validation.

* Number of entries: 668
* Attributes: 9

1. Test.csv: Testing data to use with Predictions.

* Number of entries: 100
* Attributes: 8

|  |  |
| --- | --- |
| Attribute | Description |
| A1 | Number of times pregnant |
| A2 | Plasma Oral Glucose tolerance test |
| A3 | Diastolic blood pressure |
| A4 | Triceps skinfold thickness |
| A5 | 2-Hour serum insulin |
| A6 | Body mass index |
| A7 | Diabetes pedigree function |
| A8 | Age (years) |
| Class | 0 or 1 (1= tested positive for diabetes) |

# **Steps to Build a Neural Network Model using Keras & Optimisation with GridSearchCV**

1. Importing Libraries and Loading the dataset

2. Data Exploration on all Attributes

3. Data Visualisation on all Attributes

4. Feature Scaling on all Attributes

5. Compiling the Neural Network Model

6. Hyperparameter Tuning of Neural Network Model using GridSearchCV

7. Model Building and Optimisation of Neural Network Model with Best Hyper Parameters

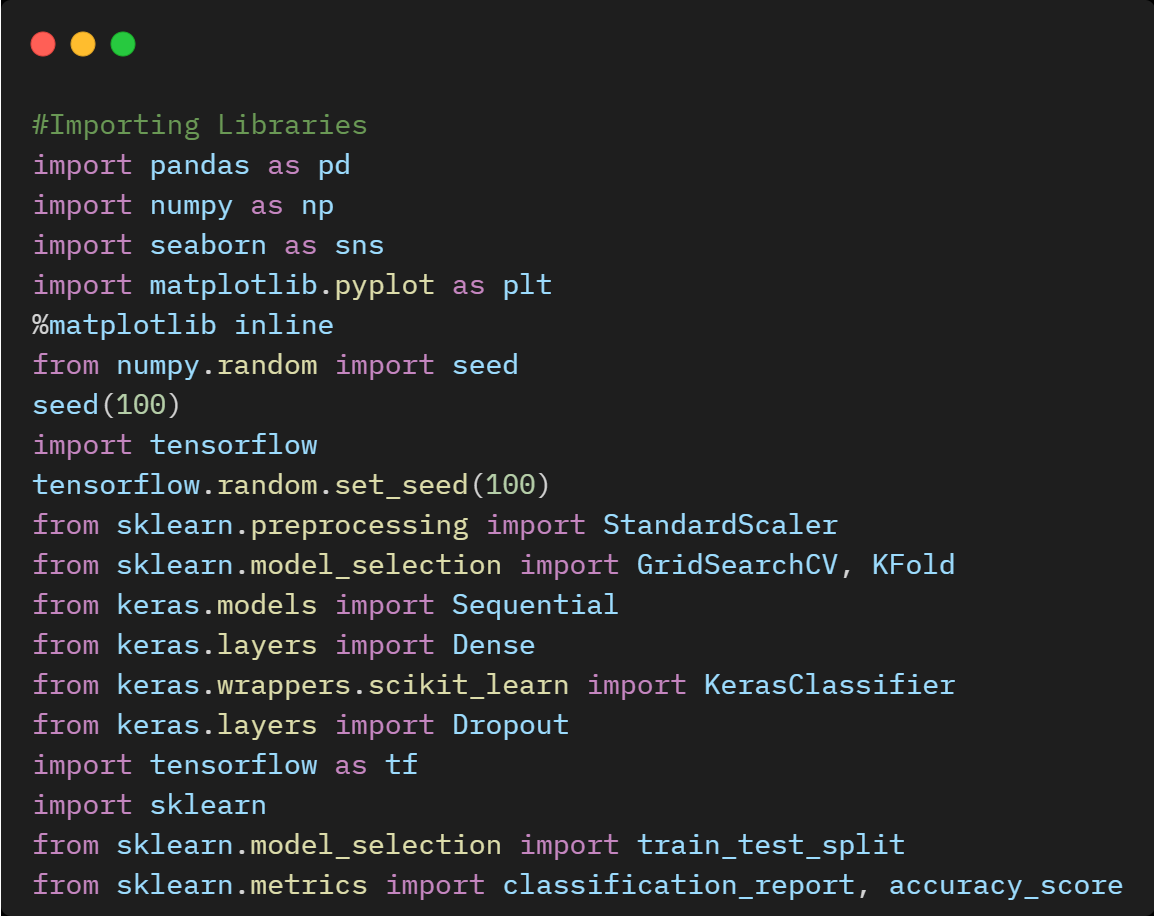
8. Evaluating Model Performance on Training and Validation data

9. Predictions on Test Data

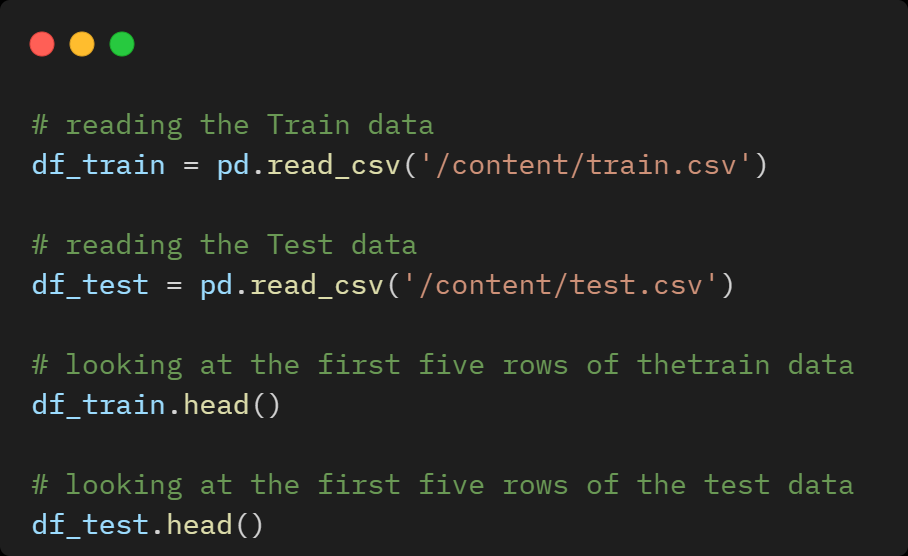
# **Importing Libraries**

Keras is a powerful and easy-to-use free open-source Python library for developing and evaluating deep learning models.

It wraps the efficient numerical computation libraries like TensorFlow and allows to define and train neural network models in just a few lines of code.



# **Reading the Train and Test Dataset**

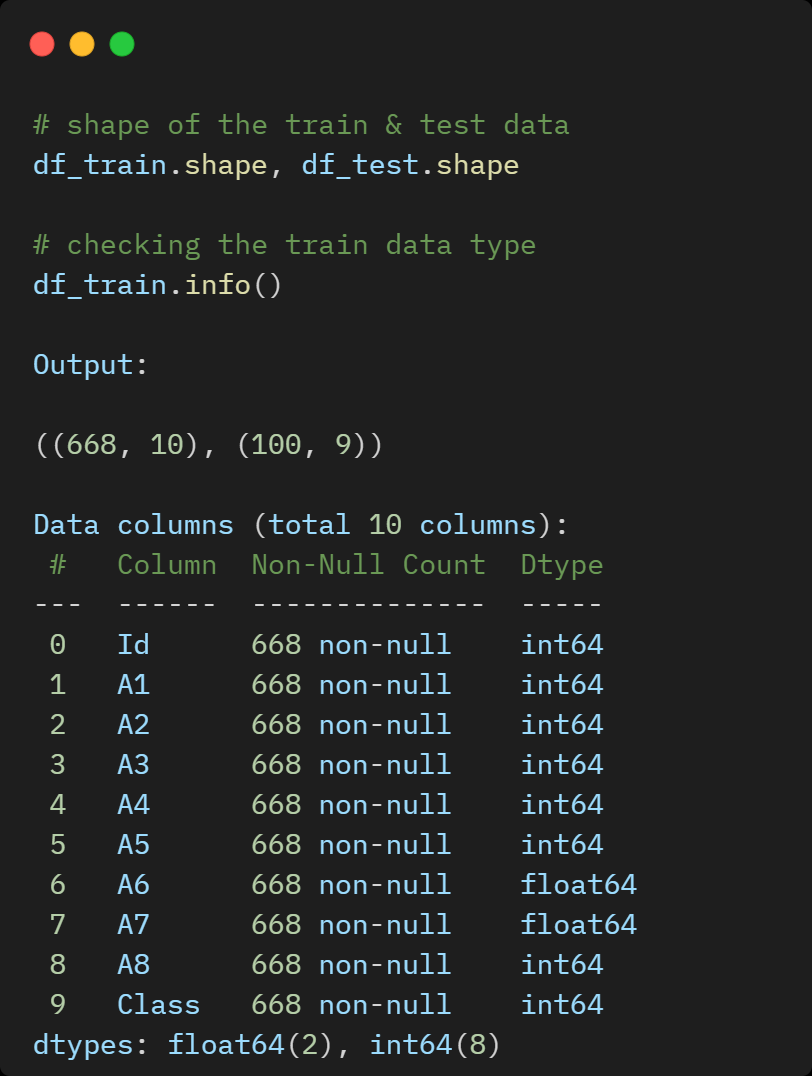


# **Data Exploration**

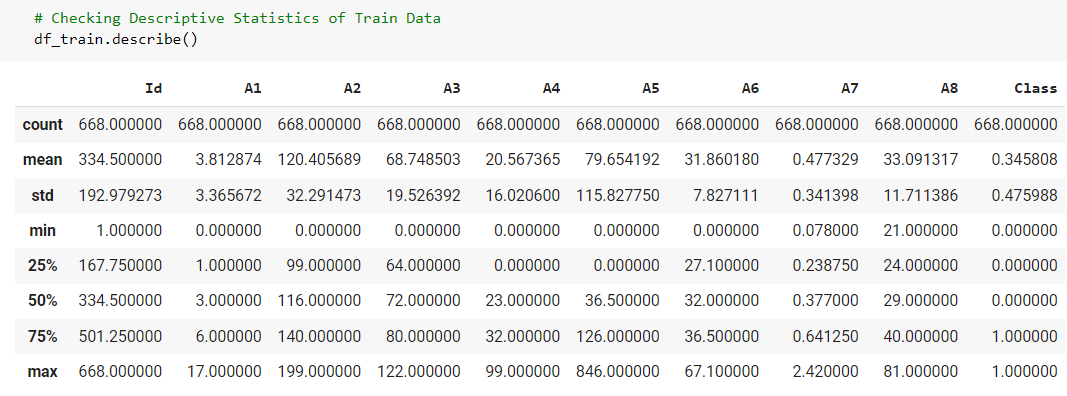
Data Exploration is an initial step to get insights from the data set to know which features have contributed more in predicting diabetes in a person.

It is always a good practice to understand the data first and try to gather as many insights from it.

## **Checking Shape and Data Types of Train & Test Data**



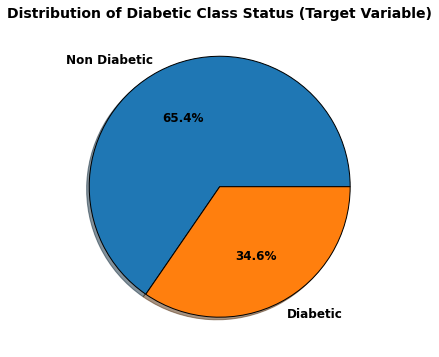
## **Checking Descriptive Statistics of Train Data**



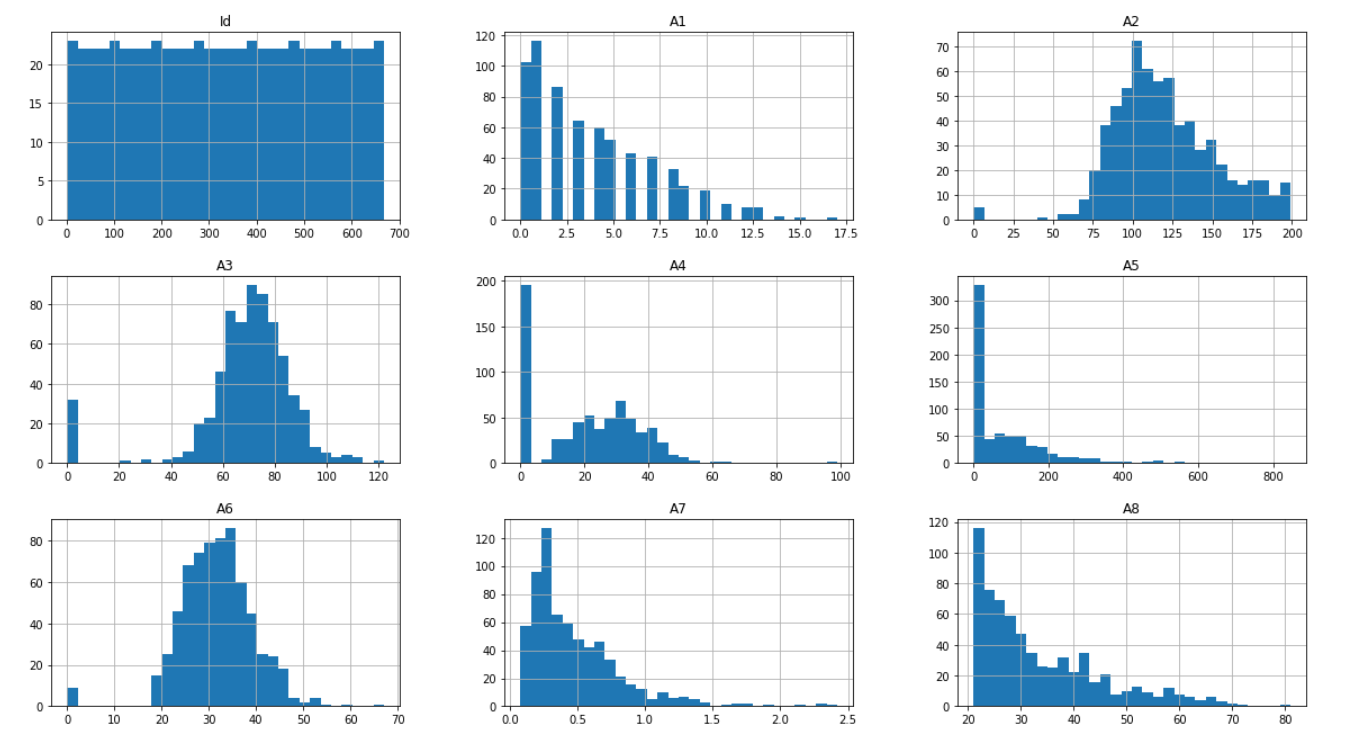
# **Data Visualisation**

## **Checking Distribution of "Class" (Target Attribute) in Train Data**

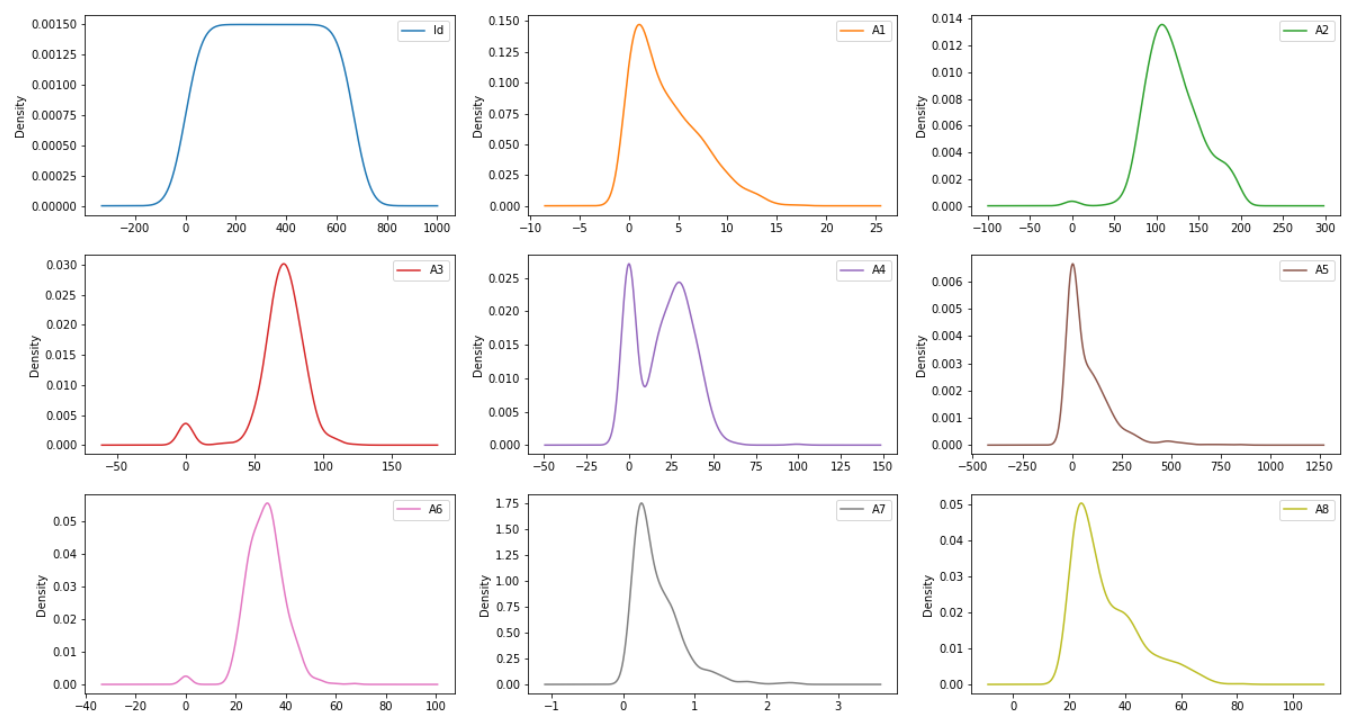
From the below pie chart, we can depict that 65.4% are non-diabetic and 34.6% are diabetic in the training data.



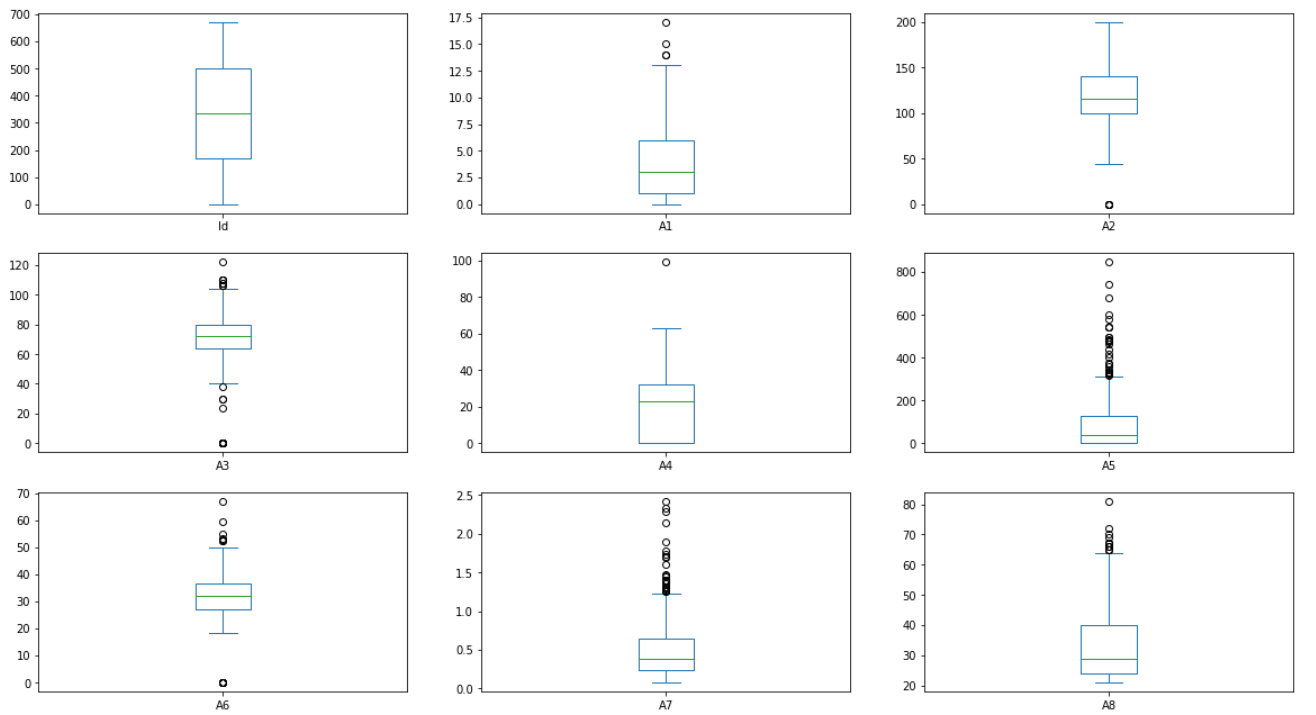
## **Plotting Histogram to Check the Distribution of all Attributes**



## **Plotting Density plots to check the Distribution of all Attributes**



## **Plotting Box plots to visualize the Distribution of all Attributes**



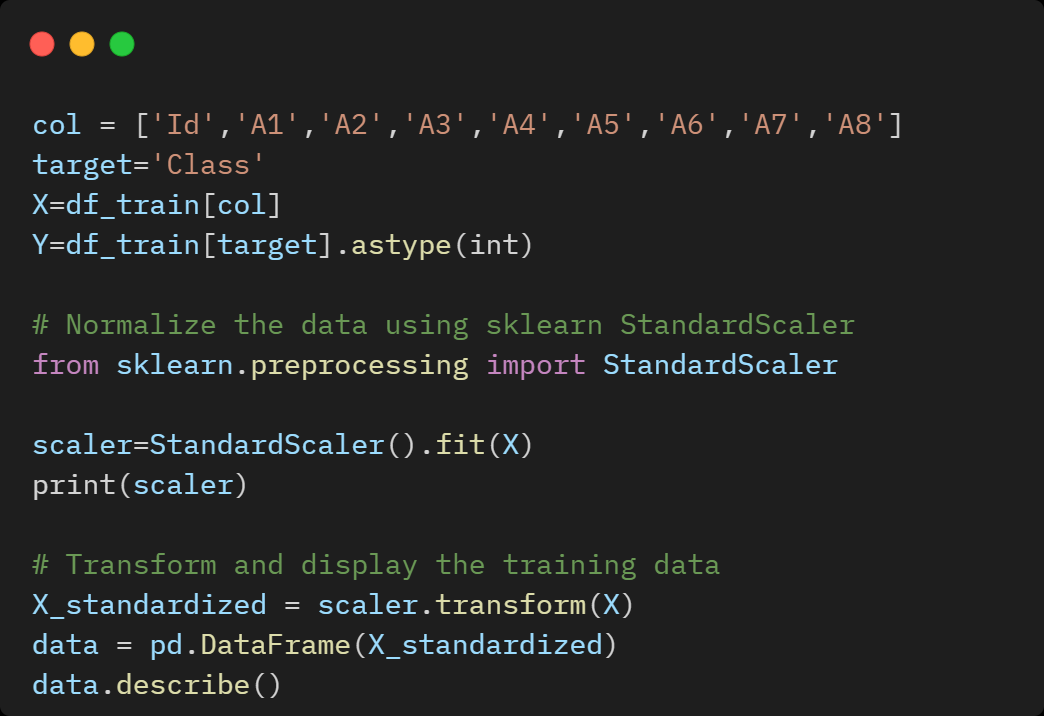
## **Summary of Data Visualisation:**

1. Bell shape curve: Blood Pressure
2. Right-Skewed: Age, Insulin, Pregnancies, Diabetes Pedigree Function
3. At least 75% of the people are

* 25 years old more
* BMI nearly 30 kg/m2
* Insulin level 100 or more
* 1 or more pregnancies
* The glucose level of 100 mg/dL or more
* Blood pressure of 60 mmHg or more

# **Feature Scaling on all Attributes**

Feature Scaling is an essential step in modeling the algorithms with the datasets. Different scales of the data features affect the modeling of a dataset and it leads to a biased outcome of predictions in terms of misclassification error and accuracy rates. Thus, it is necessary to Scale the data before model building.



# **Hyperparameter Tuning of the Neural Network Model using GridSearchCV**

Developing deep learning models is an iterative process and have lots of hyperparameters. It is very hard to tune the hyperparameter manually to get a model that can be trained efficiently in terms of time and compute resources.

So, there is a way where we can adjust the setting of the neural networks which is called hyperparameters and the process of finding a good set of hyperparameters is called hyperparameter tuning.

In this task, I have implemented hyperparameter tuning using GridSearchCV to build an optimised Neural Network model with better Accuracy and Performance.

1. Hyperparameter Tuning "Epochs" and "Batch size"
2. Hyperparameter Tuning "Learning Rate" and "Drop Out Rate"
3. Hyperparameter Tuning "Activation Function" and "Kernel Initializer"
4. Hyperparameter Tuning "Hidden Layer Neuron 1" & "Hidden Layer Neuron 2"

Python Implementation of Hyperparameter tuning of the Neural Network Model using GridSearchCV is available on [**GitHub**](https://github.com/ChaithanyaVamshi/Diabetes_Prediction_Deeplearning)

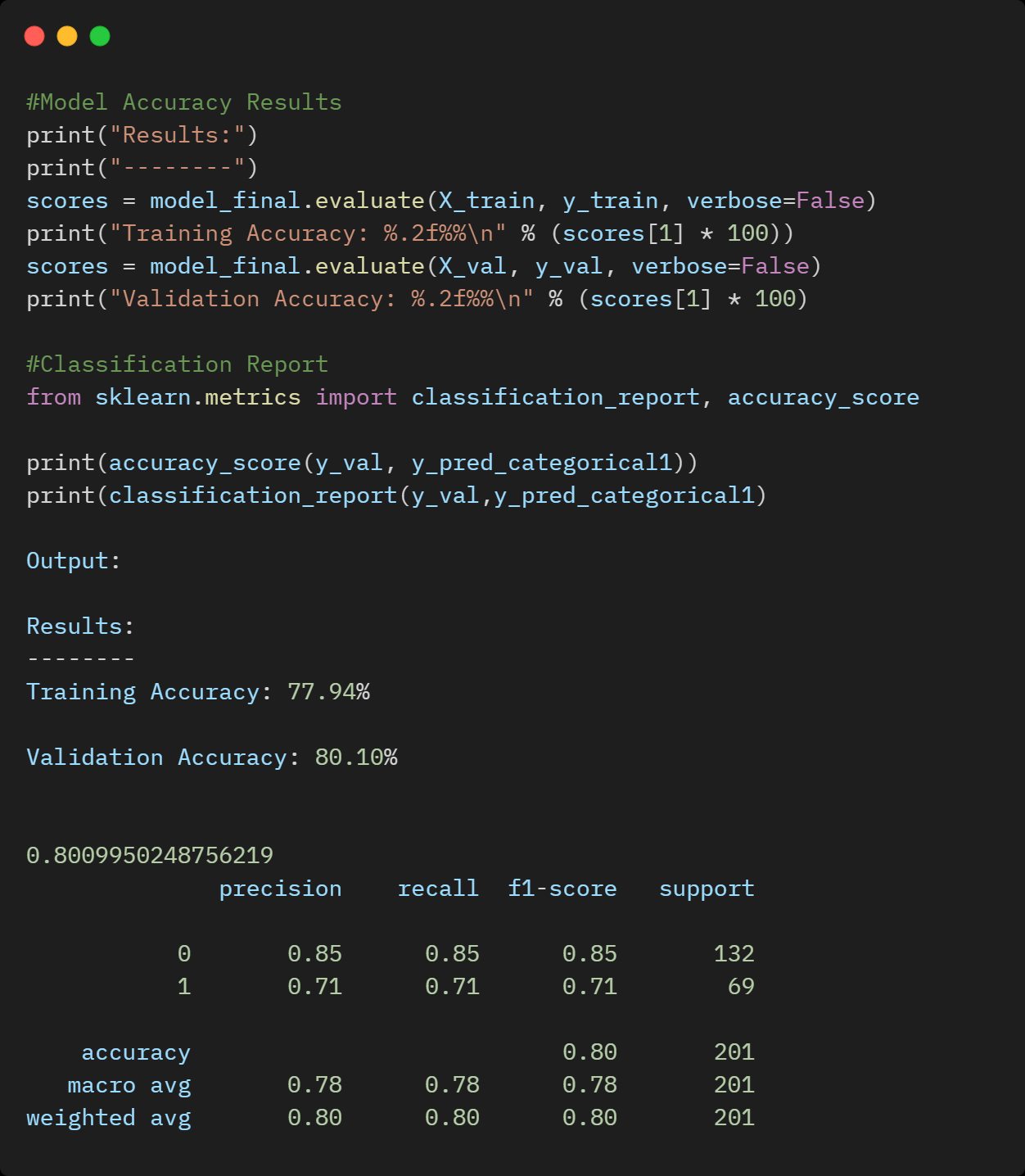
# **Model Building: Neural Network with the Best Hyperparameters Tuning using GridSearchCV**

After performing hyperparameter tuning using GridSearchCV on various parameters like Epochs, Batch size, learning rate, Activation function, Hidden layer Neurons etc we need to build a neural network model using the best hyperparameters obtained while training to get an optimised model for better Accuracy and Performance.

# **Evaluating Model Performance on Train & Validation Data**

## **Model Accuracy and Classification Report on Train & Validation Data**

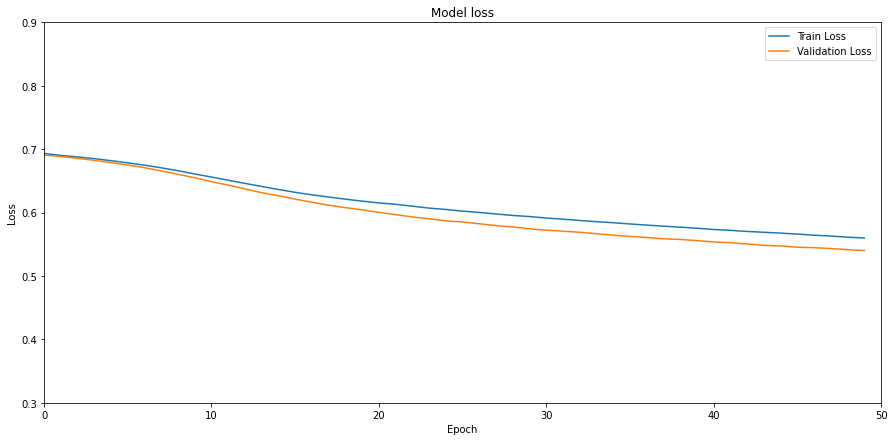
After hyperparameter tuning using GridSearchCV on the Neural Network model, we have obtained a **Training Accuracy of 78%** and a **Validation Accuracy of 80.1%** which implies a better and an optimised model.



## **Neural Network: Model Loss on Train and Validation Data**

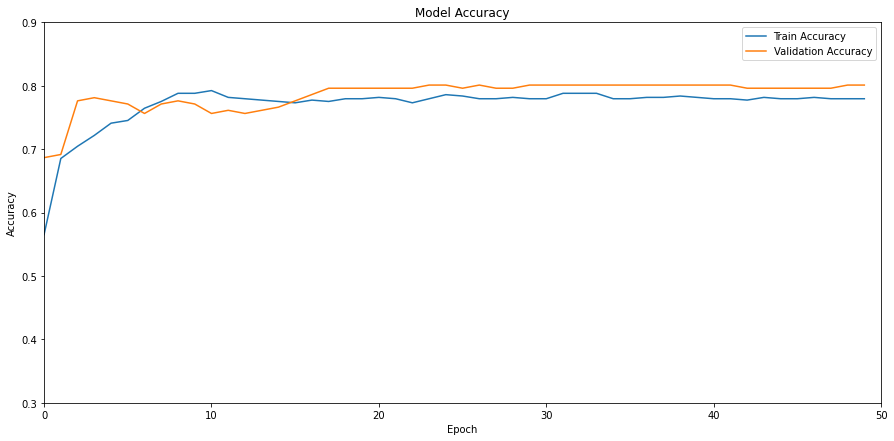
From the Model Loss chart, we can depict that as the number of epochs increases, the neural network model tends to lower the loss/cost on both training and validation data.

Hence, the Neural network model built is more effective and an optimised model.



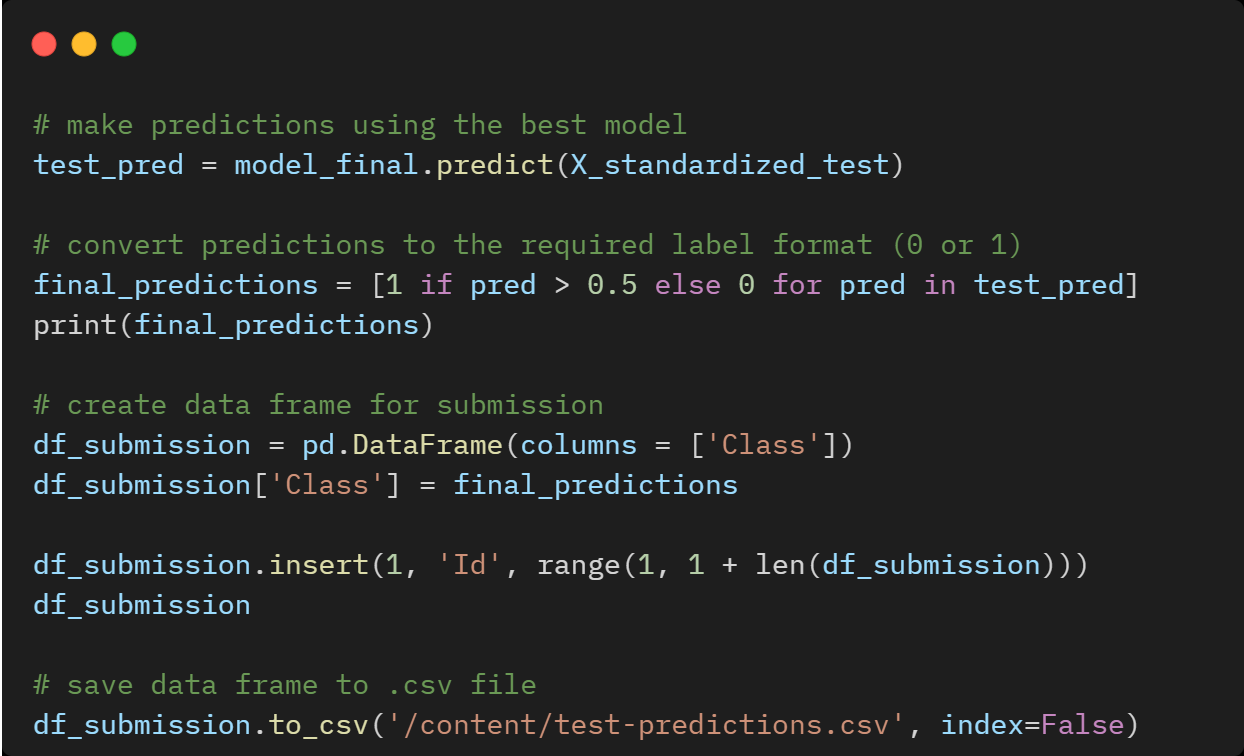
## **Neural Network: Model Accuracy on Train and Validation Data**

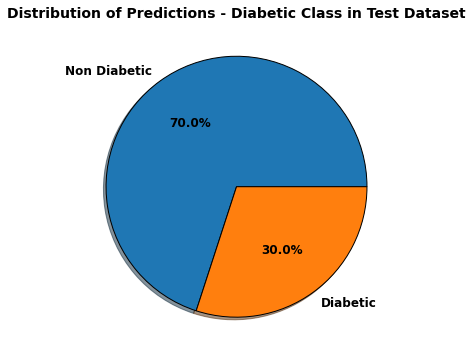
From the Model Accuracy chart, we can depict that as the number of epochs increases, the neural network model tends to increase the accuracy on both training and validation data. Hence, the Neural network model built is more effective and an optimised model.



# **Predictions on Test Data**

Using the Optimised Neural Network Model with the best hyperparameters and Accuracy, we will make predictions on test data and save predictions on .csv file name “test-predictions.csv”





# **Conclusion**

In this blog, we discussed how to approach the classification problem and predict the possibility of diabetes by implementing a Neural networks model using Keras and GridSearchCV.

We can explore this work further by trying to improve the accuracy by using advanced Deep Learning algorithms.

# **References**

<https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>

<https://medium.datadriveninvestor.com/hyperparameter-tuning-with-deep-learning-grid-search-8630aa45b2da>