

Machine Learning (CS 4131)

Lab Report

|  |  |
| --- | --- |
| **Name** | **Enrolment No** |
| Mudit Mohit | BT21GCS057 |
| Jinesh Chhajer | BT21GCS347 |
| Rishabh Pareek | BT21GCS340 |
| Manami Maity | BT21GCS351 |



# Lab 1:

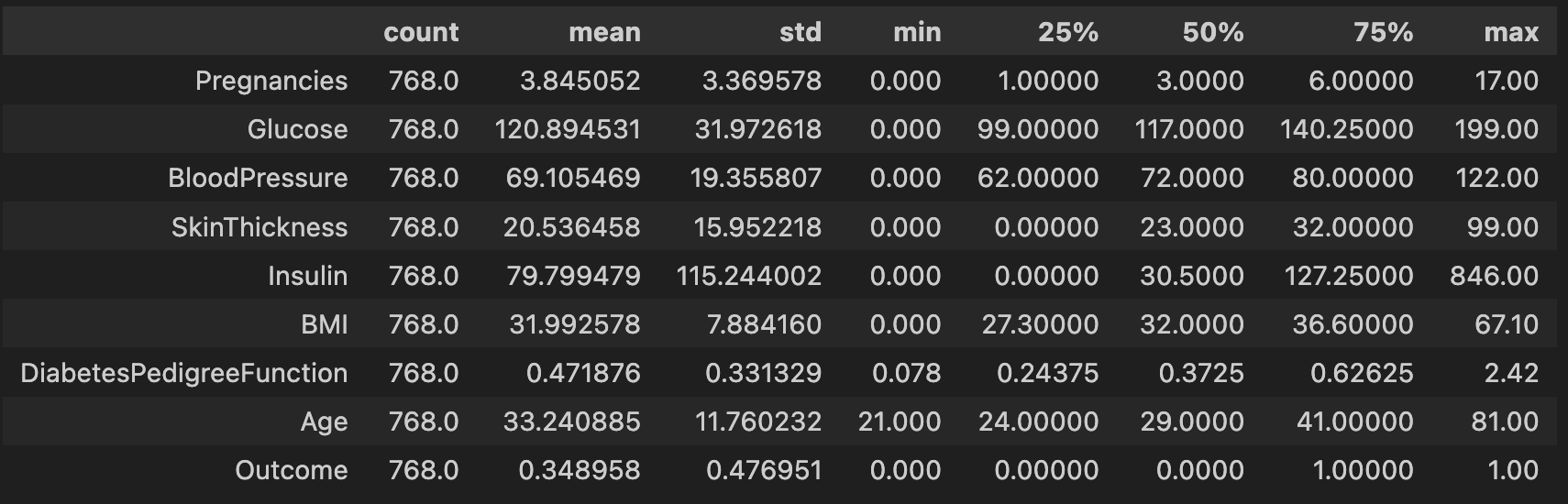
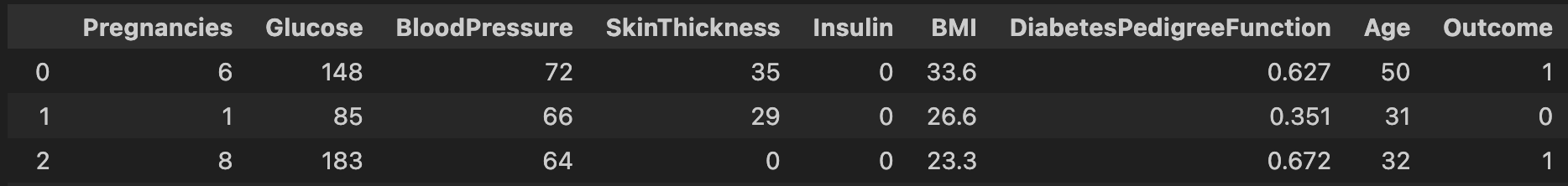
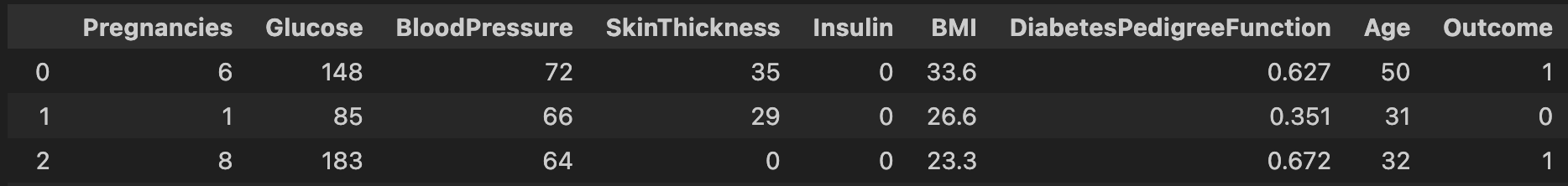
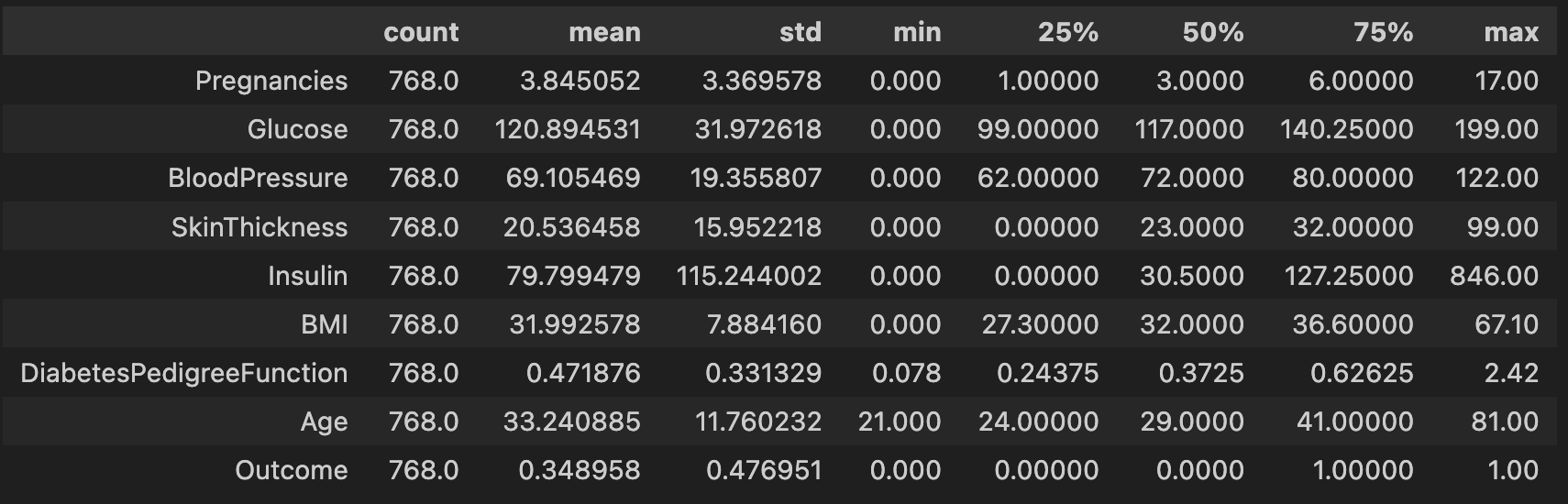
## Title: Linear Regression

## Objective:

The purpose of this laboratory was to deploy an Artificial Neural Network (ANN) on a given dataset within the domain of machine learning. It methodically investigated various configurations by tweaking parameters like epochs, units, and hidden layers to improve accuracy. Through analyzing the impact of these adjustments on accuracy, the aim was to refine the performance of the ANN and uncover insights into its behavior. This iterative process sought to enhance comprehension of how diverse configurations influence the ANN's learning and predictive abilities, thereby enriching the understanding of neural network methodologies in the field of machine learning.

## Description of dataset:

The Diabetes Dataset is sourced from the National Institute of Diabetes and Digestive and Kidney Diseases, targeting the prediction of diabetes within a particular demographic: females of Pima Indian descent aged 21 years or older. With nine distinct features, the dataset is customized for a binary classification objective, aiming to ascertain the probability of diabetes occurrence (Outcome variable: 0 indicating absence of diabetes, 1 indicating presence of diabetes).

The features include essential diagnostic measurements:

1. Pregnancies: The number of times a woman has been pregnant.

2. Glucose: Plasma glucose concentration 2 hours after an oral glucose tolerance test (measured in mg/dL).

3. Blood Pressure: Diastolic blood pressure in mmHg.

4. SkinThickness: Triceps skinfold thickness in mm.

5. Insulin: 2-Hour serum insulin level in micro-units per milliliter (mu U/ml).

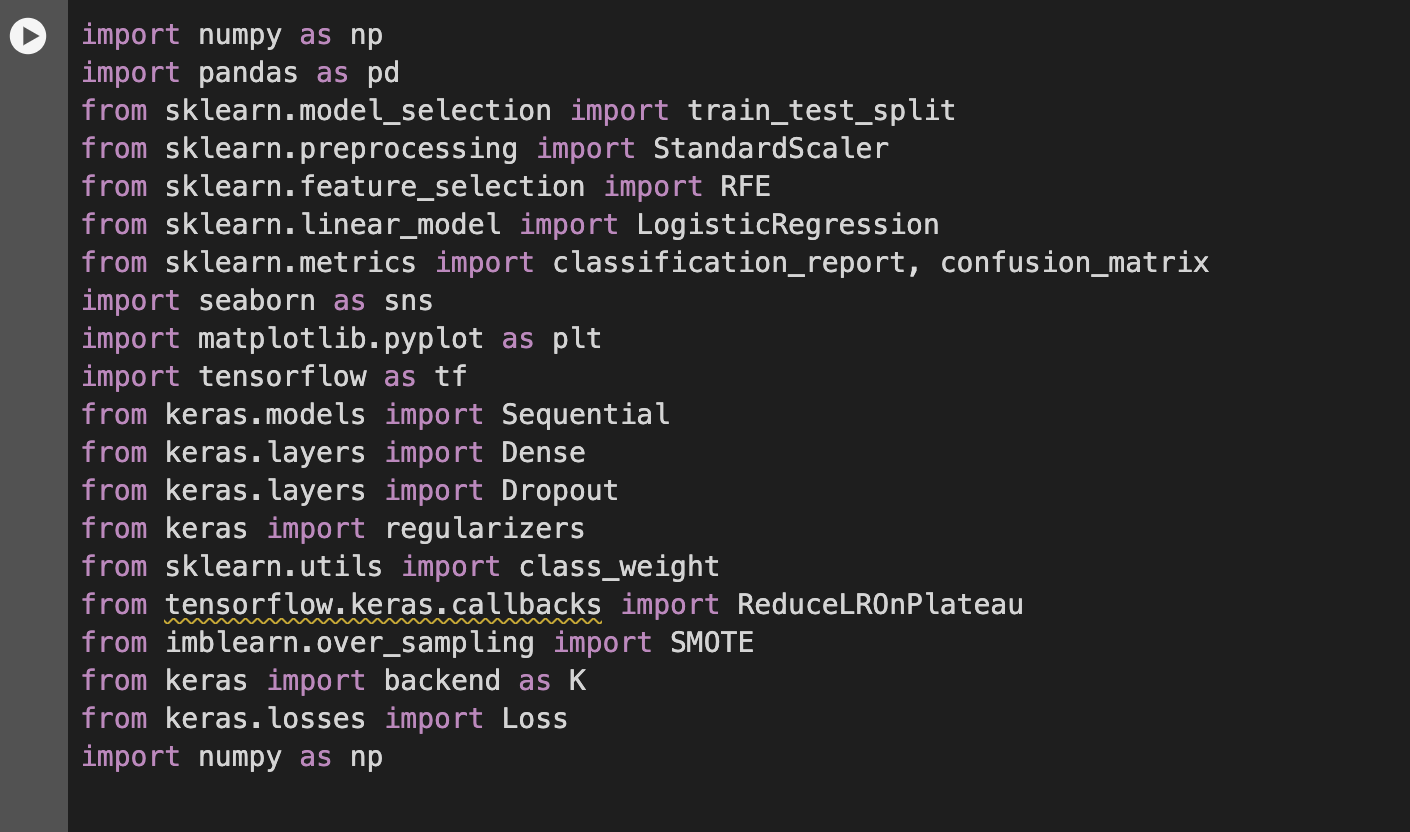
6. BMI (Body Mass Index): Calculated as weight in kilograms divided by the square of height in meters (kg/(m^2)).

7. Diabetes Pedigree Function: A function assessing the genetic influence of diabetes based on family history.

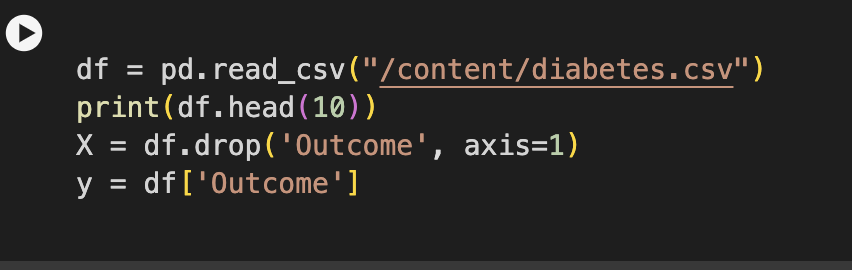
8. Age: The age of the individual in years.

9. Outcome: A binary variable indicating the presence (1) or absence (0) of diabetes.

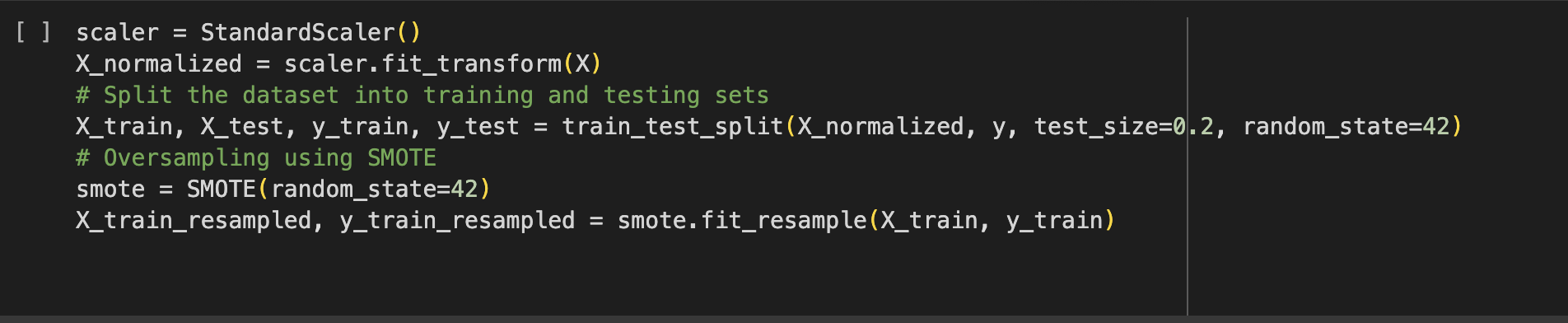
## Importing libraries:

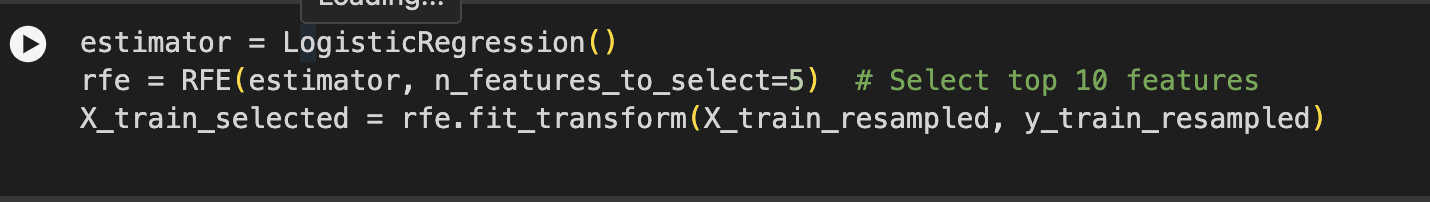


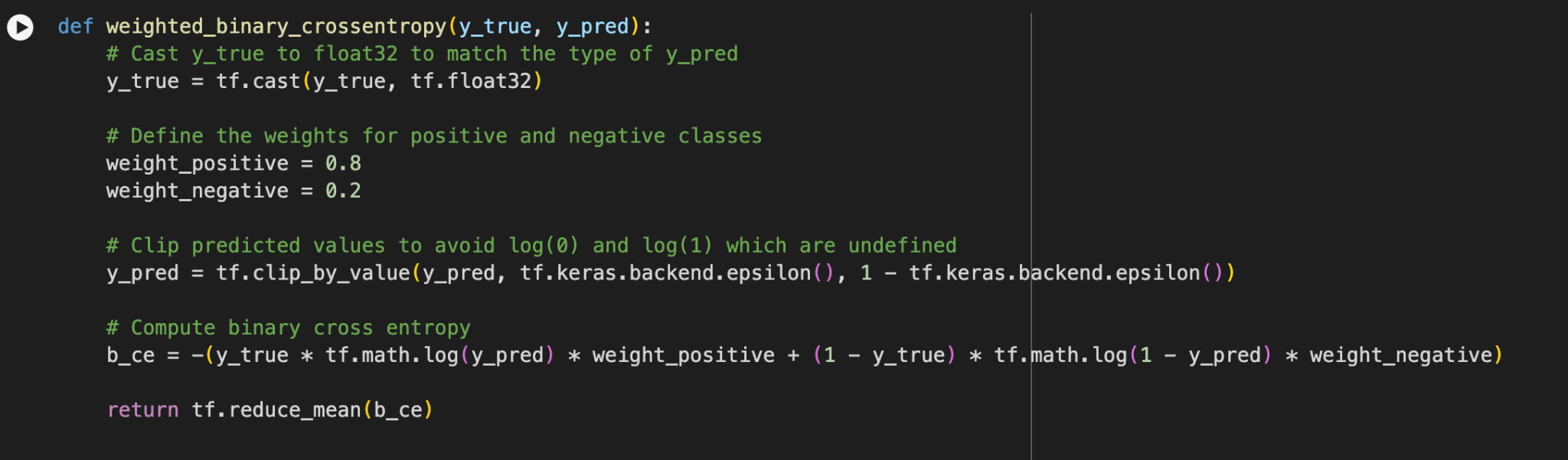
## Reading Data:



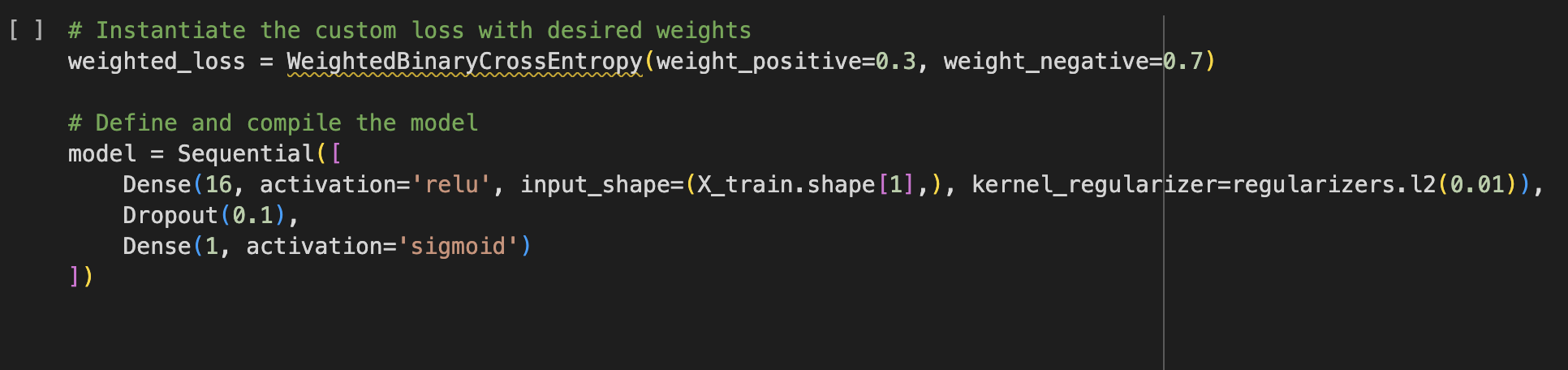
## Normalizing the splitting:



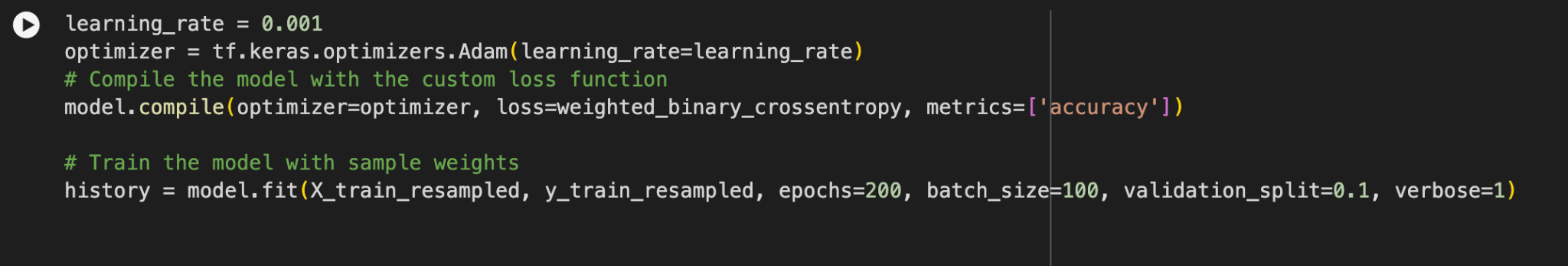
**Class weights:**

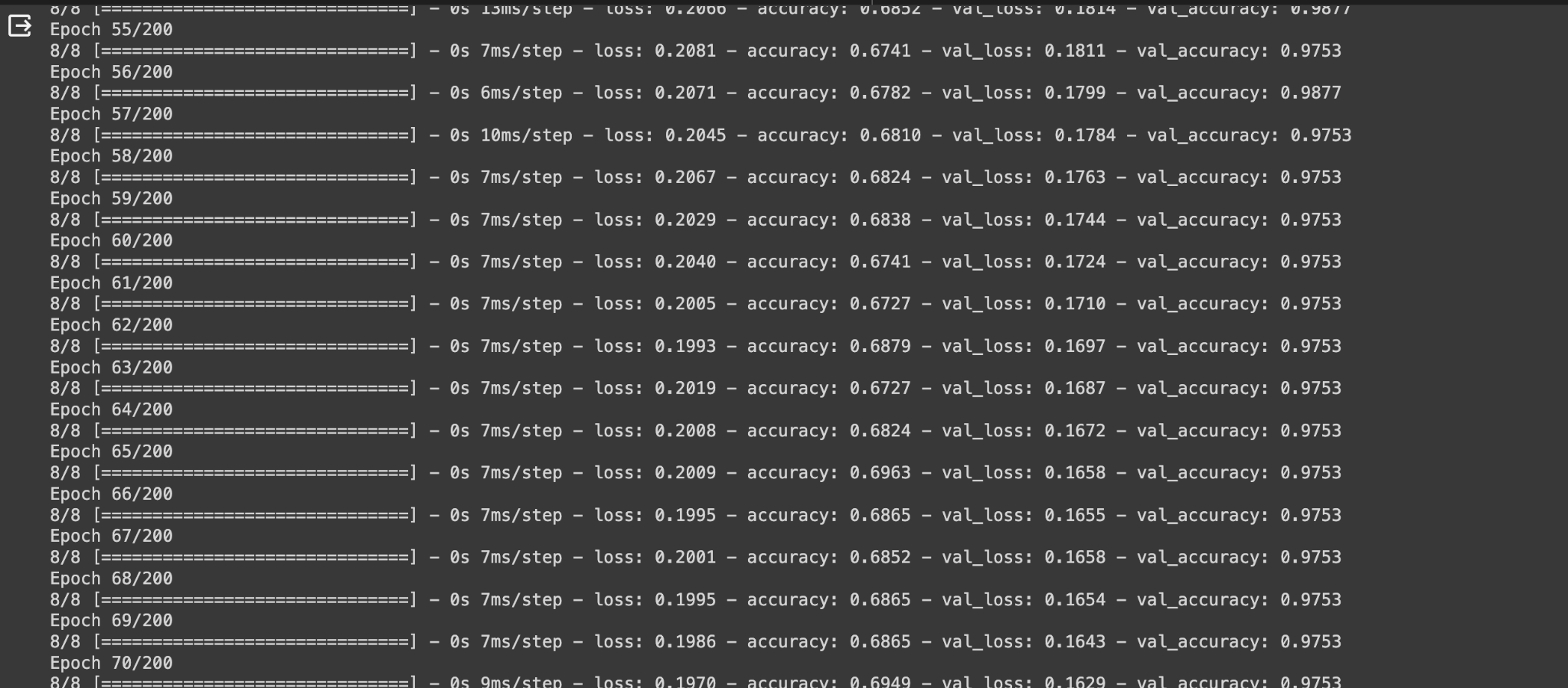
****

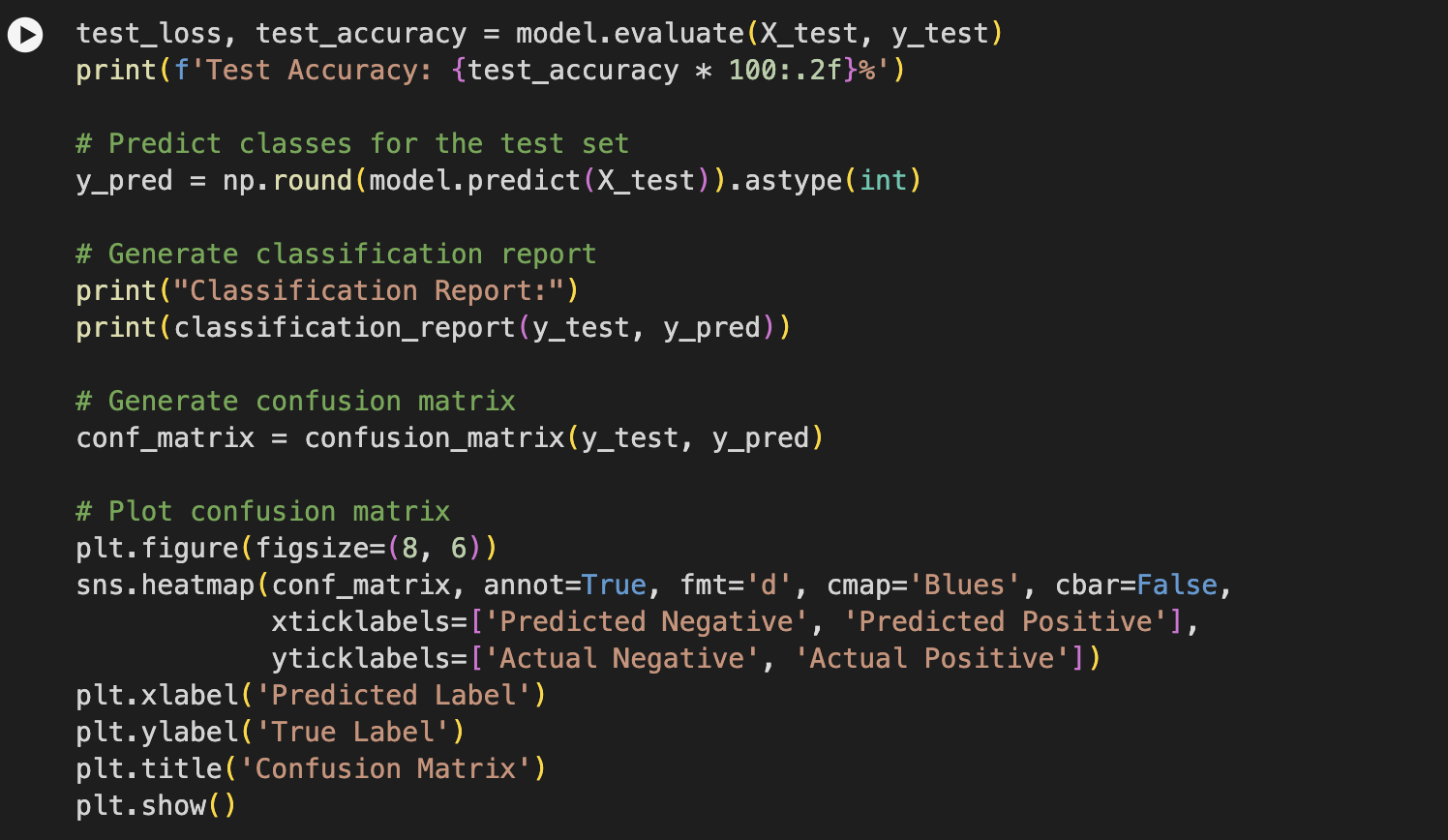
**Model:**

****

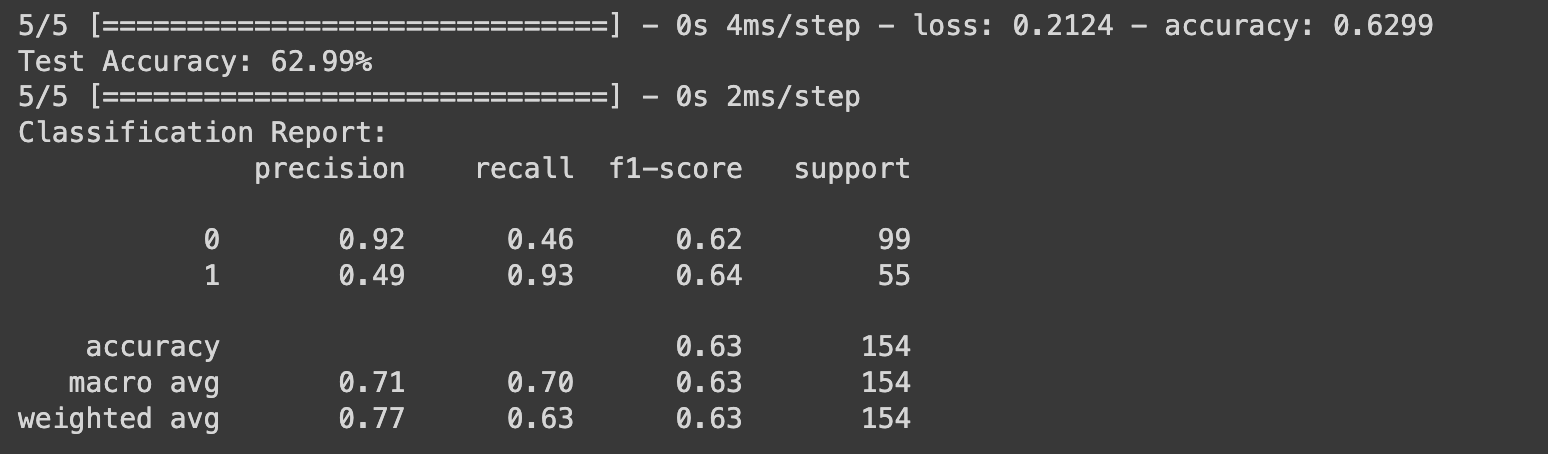
**Compiling:**



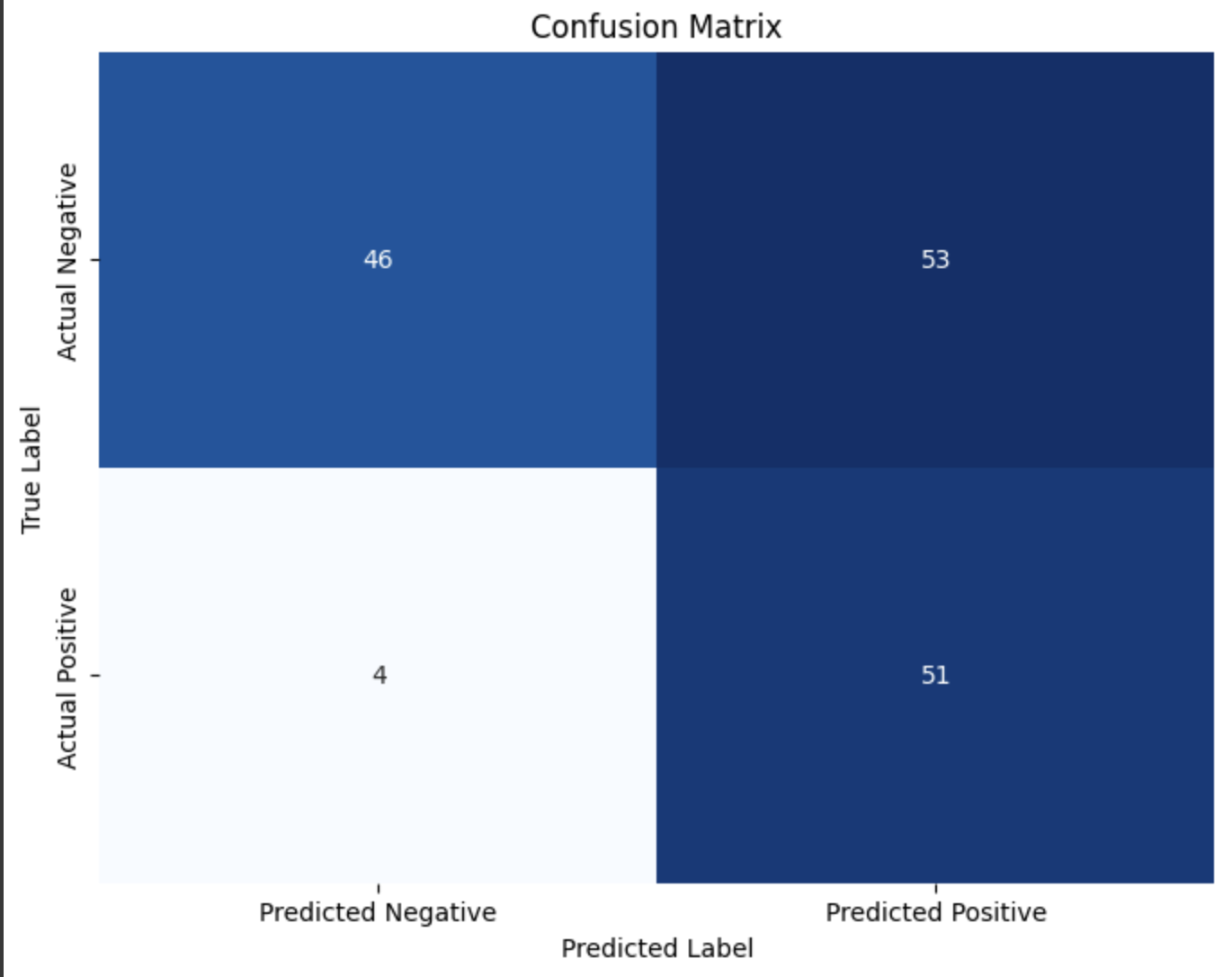
**Running model:**

**Evaluation :**

**Results:**



**Confusion Matrix:**



# Lab 2:

**Title:** Logistic Regression

## Objective:

The objective was to utilize logistic regression within the realm of machine learning, encompassing an understanding of its fundamental principles, dataset preparation, algorithm implementation, and performance evaluation. The overarching aim was to construct a predictive model for precise data classification, thereby gaining insights into the applicability and efficacy of logistic regression. This endeavor facilitated a deeper comprehension of various machine learning methodologies.

## Description of dataset:

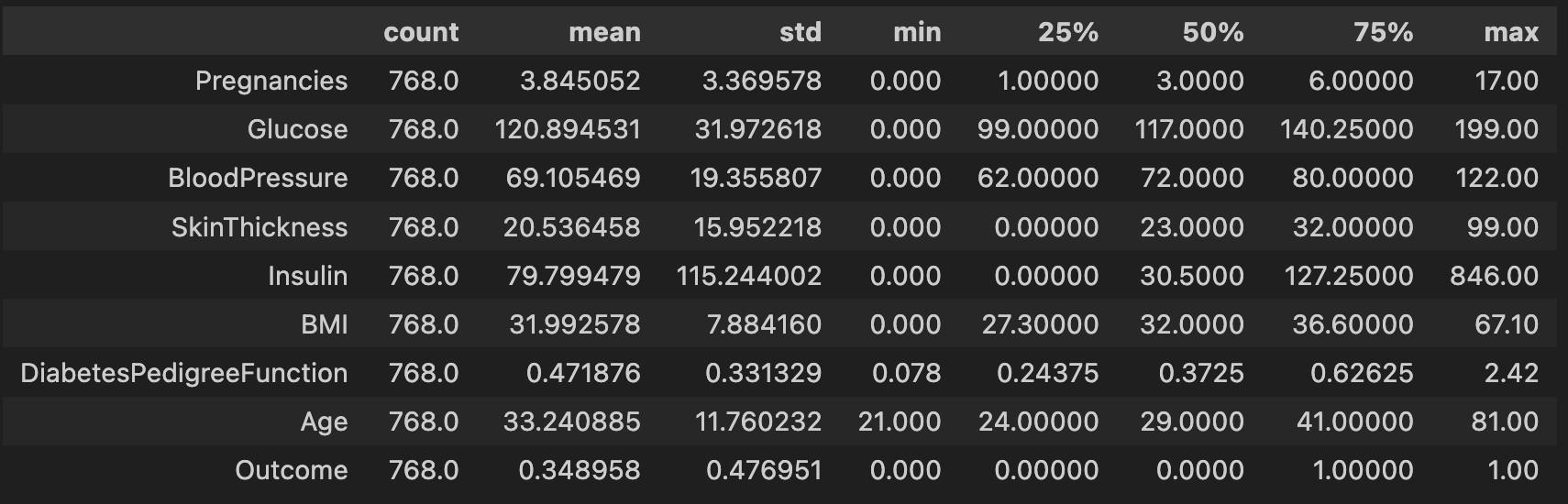
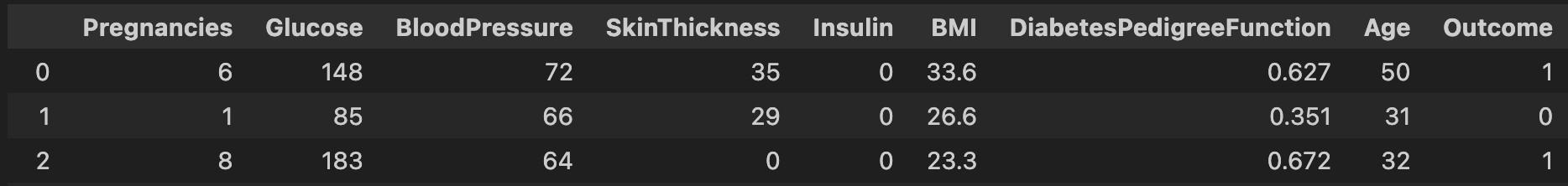
### For breast cancer dataset:

The dataset consists of attributes extracted from digitized images of breast mass Fine Needle Aspiration (FNA), employed for predicting the malignancy of tumors. It encompasses columns such as unique patient identifiers, diagnosis labels indicating malignancy or benignancy, and several mean measurements representing tumor characteristics including radius, texture, perimeter, area, smoothness, compactness, concavity, and concave points..



### For diabetes dataset:

The This dataset comprises attributes derived from digitized images of breast mass Fine Needle Aspiration (FNA), utilized to forecast tumor malignancy. It includes various columns such as unique patient identifiers, diagnosis labels denoting malignancy or benignancy, and numerous mean measurements depicting tumor characteristics like radius, texture, perimeter, area, smoothness, compactness, concavity, and concave points.

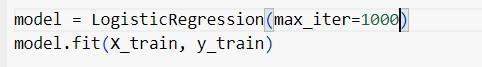


The features include essential diagnostic measurements:

* 1. Pregnancies: The number of times a woman has been pregnant.
  2. Glucose: Plasma glucose concentration 2 hours after an oral glucose tolerance test (measured in mg/dL).
  3. Blood Pressure: Diastolic blood pressure in mmHg.
  4. SkinThickness: Triceps skinfold thickness in mm.
  5. Insulin: 2-Hour serum insulin level in micro-units per milliliter (mu U/ml).
  6. BMI (Body Mass Index): Calculated as weight in kilograms divided by the square of height in meters (kg/(m^2)).
  7. Diabetes Pedigree Function: A function assessing the genetic influence of diabetes based on family history.
  8. Age: The age of the individual in years.
  9. Outcome: A binary variable indicating the presence (1) or absence (0) of diabetes.

## Comparison of model. fit

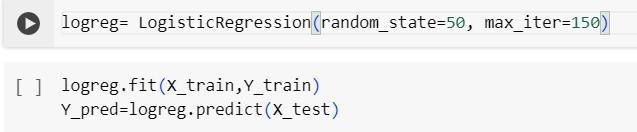
### For the breast cancer dataset



1. `model = LogisticRegression(max\_iter=1000)`: This code snippet initializes an instance of the LogisticRegression model and assigns it to the variable named 'model'. The 'max\_iter' parameter is set to 1000, indicating the maximum number of iterations for the solver to converge. It's a hyperparameter that can be adjusted based on the specific requirements of the problem.

2. `model.fit(X\_train, y\_train)`: This line of code trains (fits) the logistic regression model using the training data. 'X\_train' represents the feature matrix (input variables) of the training dataset, while 'y\_train' represents the target variable (output) for the training dataset. During this training phase, the model learns the relationship between the features and the target variable, refining its parameters to make accurate predictions.

### For Diabetes Dataset



The logistic regression has a random state of 50 and a maximum of 150 iterations for convergence. The model is trained on the provided x\_train and y\_train datasets.

## Method:

Logistic regression models the probability of binary outcomes, suitable for classification. Standardizing data involves transforming features to have zero mean and unit variance, promoting model stability and comparability. This is achieved by subtracting the mean and dividing by the standard deviation for each feature. The steps are to Load the data into pandas dataframe from csv. Pre-process the data if required, split into train and test using sklearn module, define a LogisticRegression

model using TensorFlow and Keras library, compile the model using optimisation technique, define learning rate, define batch size, fit the model and save the history, plot the history to monitor training process, evaluate the model using test set.

CODE:

FOR CANCER DATASET:

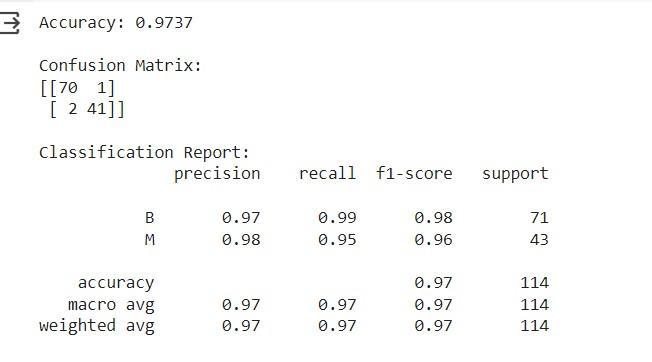


FOR DIABETES DATASET:

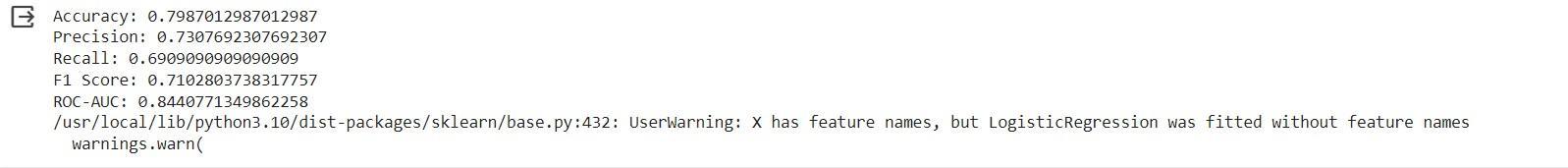


## Evaluation:

1. For breast cancer dataset:



1. For the diabetes dataset:



# Lab 3:

**Title:** Implementation of ANN

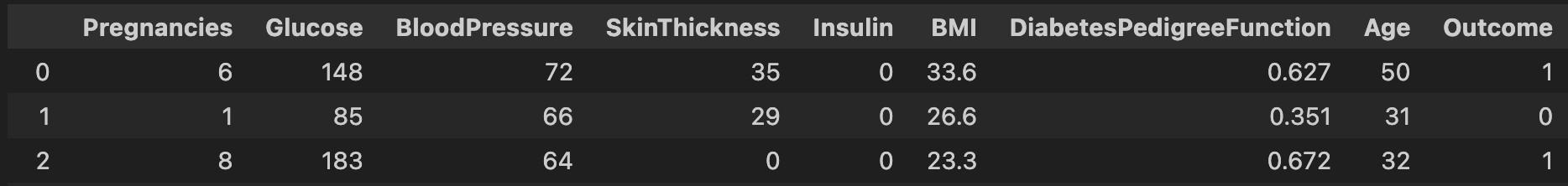
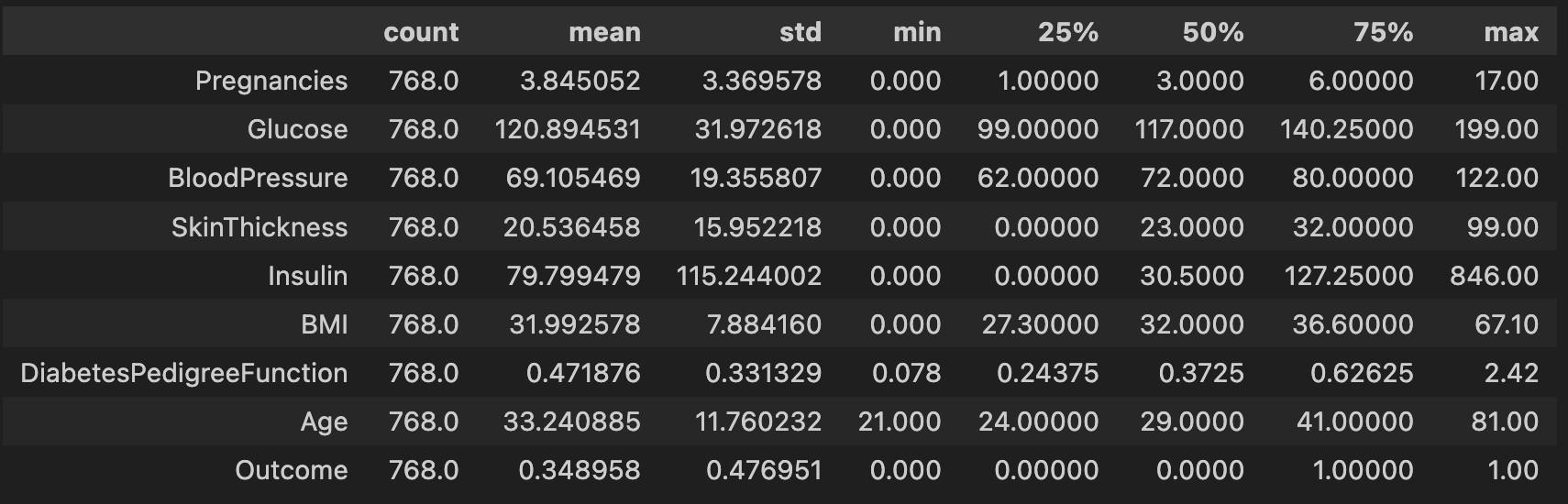
1. To implement the diabetes dataset
2. Hyperparameter tuning to improve the model result

## Objective:

In this experiment, the focus was on implementing an Artificial Neural Network (ANN) using a specific dataset in the field of machine learning. The approach involved systematically investigating various configurations by tweaking parameters like epochs, units, hidden layers, among others, with the goal of improving accuracy. Through careful analysis of how these adjustments impacted accuracy, the aim was to fine-tune the performance of the ANN and gain valuable insights into its behavior. This iterative process sought to deepen comprehension of how different configurations affect the ANN's ability to learn and make predictions, thereby contributing to a more comprehensive understanding of neural network methodologies within the realm of machine learning..

## Description of dataset:

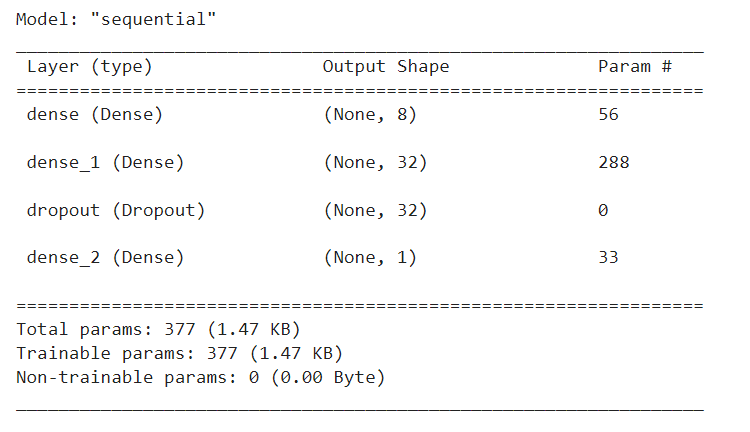
### diabetes dataset:

The dataset, sourced from the National Institute of Diabetes and Digestive and Kidney Diseases, is specialized in forecasting diabetes within a particular demographic—specifically, females of Pima Indian ancestry aged 21 years and above. With nine distinct features, this dataset is designed for binary classification purposes, aiming to ascertain the probability of diabetes occurrence (where the Outcome variable is coded as 0 for absence of diabetes and 1 for presence of diabetes).

The features include essential diagnostic measurements:

* 1. Pregnancies: The number of times a woman has been pregnant.
  2. Glucose: Plasma glucose concentration 2 hours after an oral glucose tolerance test (measured in mg/dL).
  3. Blood Pressure: Diastolic blood pressure in mmHg.
  4. SkinThickness: Triceps skinfold thickness in mm.
  5. Insulin: 2-Hour serum insulin level in micro-units per milliliter (mu U/ml).
  6. BMI (Body Mass Index): Calculated as weight in kilograms divided by the square of height in meters (kg/(m^2)).
  7. Diabetes Pedigree Function: A function assessing the genetic influence of diabetes based on family history.
  8. Age: The age of the individual in years.
  9. Outcome: A binary variable indicating the presence (1) or absence (0) of diabetes.

## Model.summary:



## Method:

### Load the dataset:

Load the dataset from the csv file.

### Preprocess the data:

For both the datasets, we have normalized the data. To improve the accuracy of diabetes data, we removed the column ‘Pregnancies’ for feature selection as this feature was not appropriate for the prediction according to the specific dataset.

### Build the model:

Create a Sequential model using TensorFlow Keras. Add layers to the model such as Dense layers with appropriate activation functions. Define the input shape based on the number of features in the dataset.

### Compile the model:

Compile the model by specifying the loss function, optimizer, and metrics to use during training.

### Train the model:

Train the model on the training data using the fit method. Specify the number of epochs (iterations) and batch size.

### Evaluate the model:

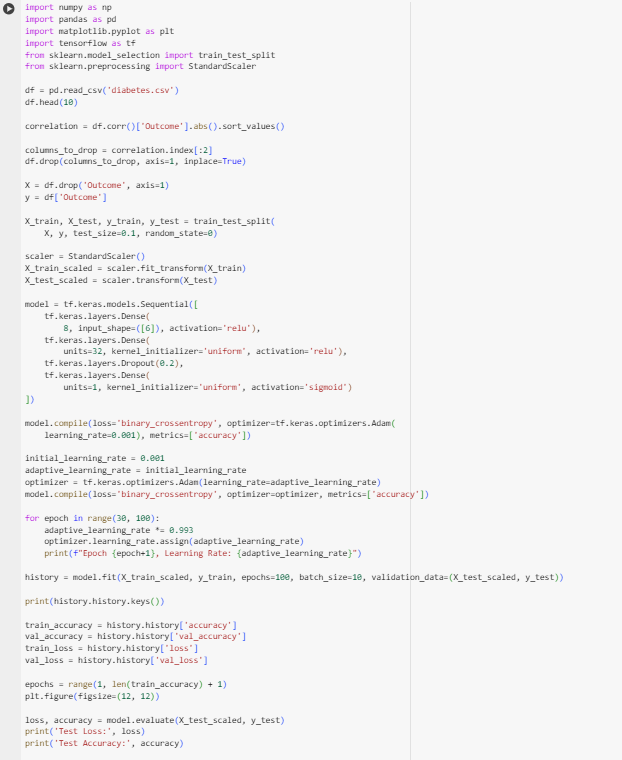
Evaluate the trained model on the test data to assess its performance using metrics such as accuracy, precision, recall, or F1-score.

### Fine-tune the model:

Experiment with different values of learning rate, batch size.

In the code given below, we are mainly focussed on increasing the final accuracy for the diabetes dataset given to us. In order to do this, we have made some modifications to the code by adding another dense layer with relu activation function, increasing the number of units in the existing neural layer, and including dropout regularization and class weights in the code. We have also changed the epoch value, learning rate and batch size. Apart from that, we have added a small condition that keeps the learning rate constant for the first 30 epochs, and then decreases it exponentially after that. This is achieved by multiplying it with 0.98 after each iteration.

**CODE:**



## Evaluation:

