Stamatics
IIT Kanpur

CLASSIFYMEISTER

IMPLEMENTING AND COMPARING LOGISTIC REGRESSION AND KNN ON DIABETES DATASET

Group #10

ADITYA MODI
SANYAM SHIVHARE
JIYA VERMA
AVANTI WASEKAR

OBJECTIVE

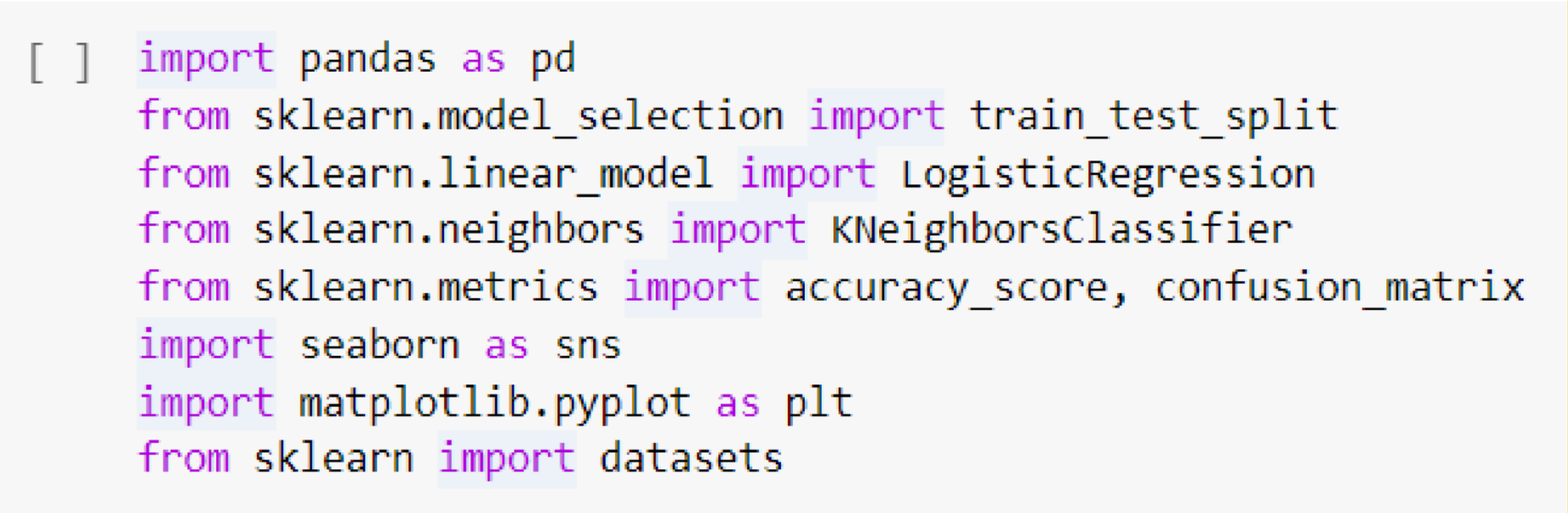
To create a Machine Learning model using Knn algorithm and Logistic Regression to test some given data of patients and see if they are under either category diabetes or non-diabetic. Total number of studied list in this dataset related to diabetic and non-diabetic patients is 768 , which we will manipulate, and scrap these data to use them in our KNN predictive model.

DATASET USED

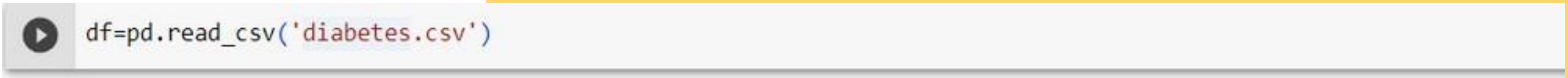
We’ve obtained the subjected dataset from Kaggle , however the dataset was initially presented by the National Institute of Diabetes and Digestive and Kidney Diseases . The data can be downloaded from here. The datasets include data from 768 women with several medical predictor variables and one target variable. The classification goal is to predict whether or not the patients in the dataset have diabetes or not.

METHODOLOGY

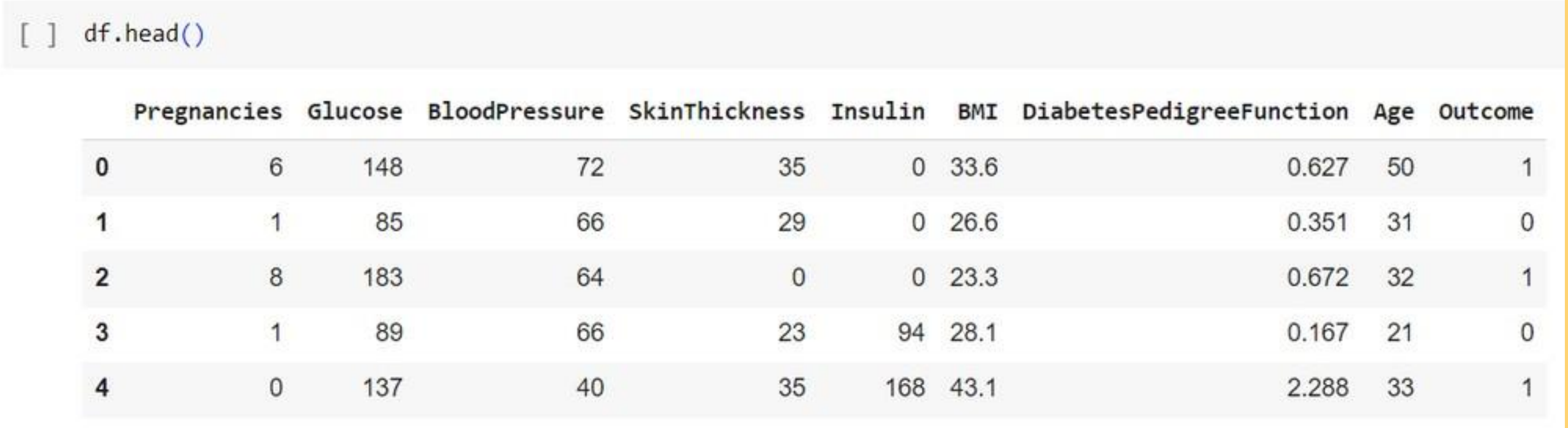
First of all we imported the dataset from the SKlearn databases and converted the data into a dataframe. Then by opening the subjected dataset using pandas syntax csv_read() which read the dataset and transform it to a structured tabular data for us to read. Then we splitted the data into training and testing data and trained the model based on the training data and tested the trained model on the testing data.



```
[ ]  import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.metrics import accuracy_score, confusion_matrix
     import seaborn as sns
     import matplotlib.pyplot as plt
     from sklearn import datasets
```

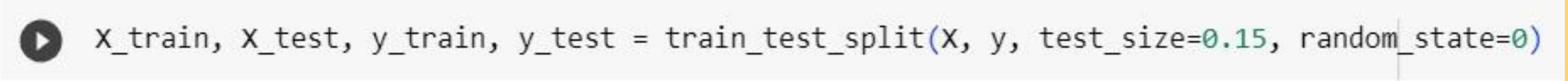


```
df=pd.read_csv('diabetes.csv')
```



```
[ ]  df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
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



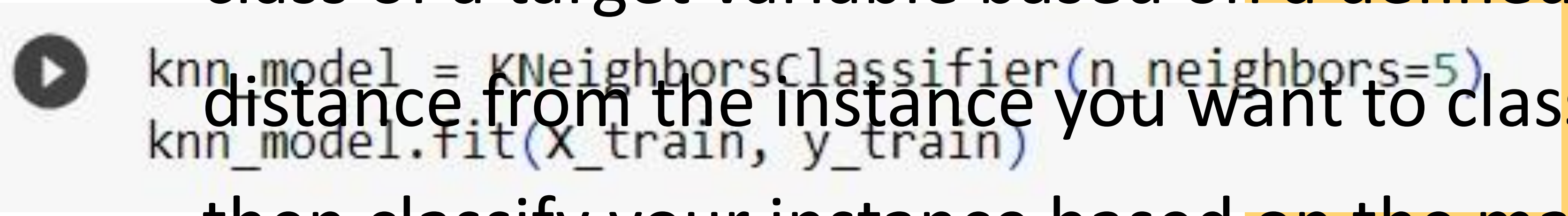
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, random_state=0)
```

TECHNIQUES USED

Firstly we used Logistic Regression model to test the data, the Logistic

Regression model gave us an accuracy of 81.8%. We also tried to use different models. KNN model gave an accuracy of 79.4%. Since the Logistic Regression model gave better accuracy we went with the logistic regression model.

KNN algorithm is a supervised machine learning algorithm that deals with similarity . KNN stands for K-Nearest Neighbors. It’s basically a classification algorithm that will make a prediction of a class of a target variable based on a defined number of nearest neighbors. It will calculate distance from the instance you want to classify to every instance of the training dataset, and then classify your instance based on the majority classes of k nearest instances.

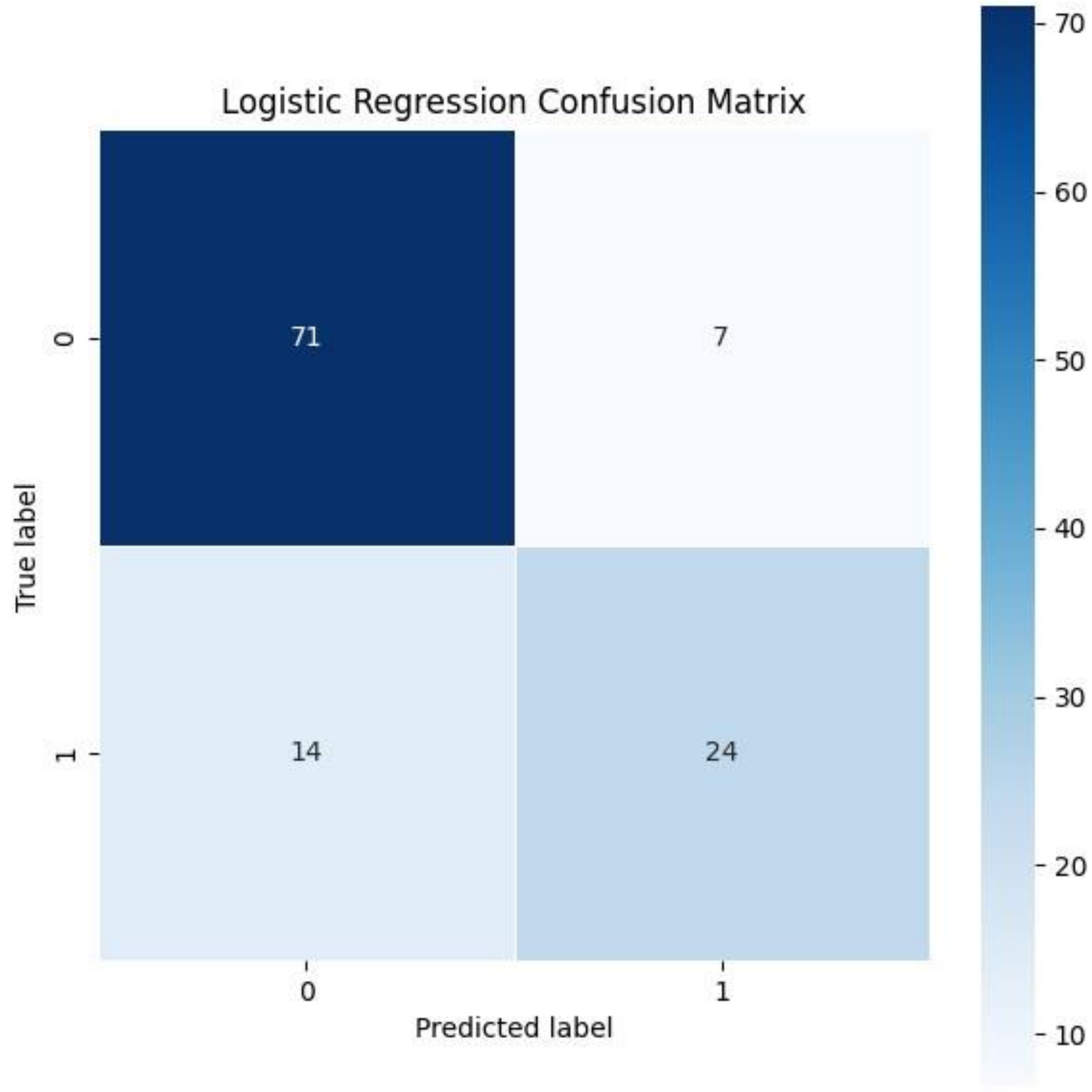


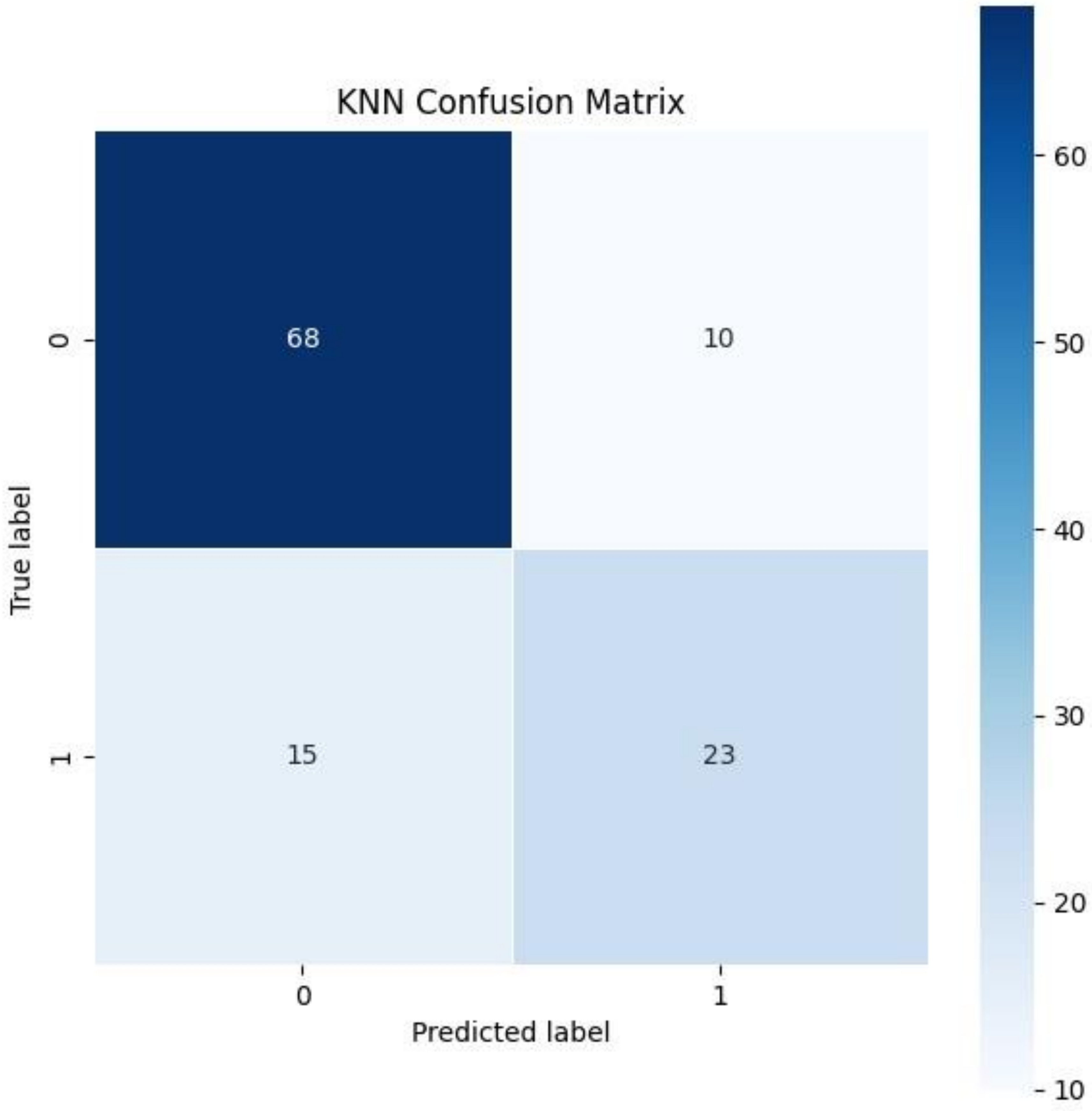
```
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train, y_train)
```

Logistic regression is a binary classification algorithm that predicts the probability of an instance belonging to a certain class. It uses a logistic (or sigmoid) function to transform a linear combination of input features into a probability value between 0 and 1. The model's parameters are estimated through maximum likelihood estimation. Logistic regression is widely used for its simplicity and interpretability in various fields.

Mathematically, logistic regression can be represented as: $p = 1 / (1 + e^{(-z)})$

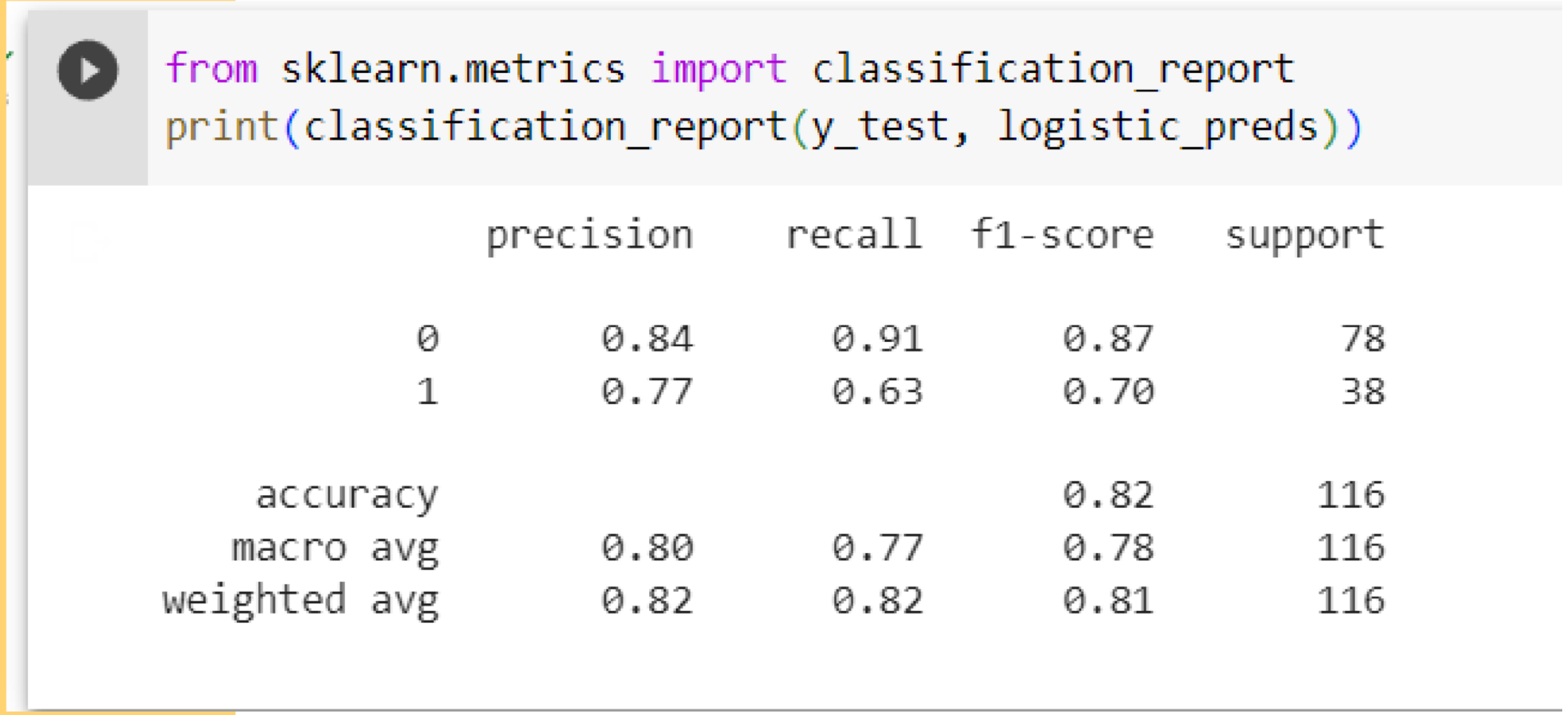
where p is the predicted probability, z is the weighted sum of the input features, and e is the base of the natural logarithm.


Logistic Regression Confusion Matrix
True label
Predicted label
0
1
0
1
71
7
14
24
70
60
50
40
30
20
10


KNN Confusion Matrix
True label
Predicted label
0
1
0
1
68
10
15
23
60
50
40
30
20
10

RESULTS

We used Logistic regression to make the model and obtained an accuracy of 81.8%. The model was able to detect if the patient has diabetes with 81.8% precision. We also implemented the model using KNN. The accuracy we obtained for the KNN model was 78.4%.



```
from sklearn.metrics import classification_report
print(classification_report(y_test, logistic_preds))
```

```
              precision    recall  f1-score   support

           0       0.84      0.91      0.87        78
           1       0.77      0.63      0.70        38

    accuracy                           0.82       116
   macro avg       0.80      0.77      0.78       116
weighted avg       0.82      0.82      0.81       116
```

https://colab.research.google.com/drive/1LHLcnM9ySa_FJ22KcEm2y4bVtVW-ThrL?usp=sharing

CONSLUSION

This model aimed to develop an accurate and efficient system for diagnosing diabetes in patients using machine learning techniques which can be leveraged by doctors to assist in several ways. Different machine learning algorithms, such as logistic regression and KNNs were implemented and analyzed to give the best possible result. Our model was made by using regression techniques and further optimized to achieve the best possible performance.

THANK YOU