Home About Contact

Portofolio

CLASSIFICATION DIABETES DESEASE USING NAIVE BAYES METHOD

By : Muhammad Fikri Riyanto

AUTHOR BIODATA

Profound understanding of programming language and data analyst tools, including versatile SQL Queries, Python, Tableu, Power BI, R studio, IBM SPSS, Minitab, Mathlab, Microsoft Office, Machine Learning Model, and Natural Language Processing Experienced in handling project involving hight - complexity datasets, adept at conducting in depth data analisis, encompassing data collection, cleaning, visualization processes and analyst . Capable of creating solutions and predictions to identify opportunities and contribute to progress of current and future business endeavor.

Home About Contact

ABOUT PORTOFOLIO

This project is designed to predict and identify Diabetes Mellitus by utilizing the Naïve Bayes method, which is known for its simplicity and effectiveness in handling classification problems. The project seeks to analyze the model's performance by calculating key evaluation metrics such as accuracy, recall, and precision. These metrics are used to assess how well the Naïve Bayes method can classify and distinguish between diabetic and non-diabetic cases. By leveraging this method, the project aims to provide a reliable and efficient solution for supporting early detection and diagnosis of diabetes, which is crucial for timely medical intervention and management.

Read More

PREPROCESSING DATA

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

At this stage, data preprocessing will be performed on all variables in the dataset. The first step is to check for missing values, which involves determining whether any data points are missing. After the check, it was found that no variables had missing values, and all of them had a value of 0 for missing data cases.

FEATURE SCALLING

	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

Features that underwent scaling

Feature scaling is used to identify data dimensions that have a range of values with a more significant difference compared to other features. It can be observed that the attributes Glucose, BloodPressure, and SkinThickness have a more prominent range of values, or in other words, higher values compared to other attributes.

Therefore, feature scaling is applied to ensure that the values of each attribute are more balanced and within a comparable range. This step helps in minimizing the bias caused by features with larger value ranges, which could otherwise dominate the model's learning process. After the scaling process, the feature values are transformed to a standardized scale, making them more uniform and comparable with each other. As a result, it can be observed that the values of attributes like Glucose, BloodPressure, and SkinThickness are now on a more equal footing with other features, leading to a more efficient and accurate model training process.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	-0.6	0.767442	1.125	0.666667	-0.333333	1.043011	-0.018767	-0.470588
1	-0.4	-0.813953	-0.250	-0.727273	-0.333333	-1.526882	0.659517	-0.117647
2	-0.2	0.093023	-0.125	0.090909	-0.333333	0.473118	-0.101877	-0.117647
3	0.2	0.651163	0.375	-0.727273	-0.333333	0.655914	0.380697	2.235294
4	-0.6	0.116279	0.000	-0.727273	-0.333333	0.419355	-0.321716	1.352941

Features after scaling

SPLITTING DATA

The result of preprocessing will be divided into two types of dataset splits: the training set and the testing set. The division of the dataset will be done with a proportion of 70:30.

A 70:30 split between the training and testing data is a commonly used choice in evaluating data mining methods. With 70% of the data used to learn the patterns within the dataset, the remaining 30% is allocated for testing, providing a sufficiently large sample to evaluate the model's performance and assess its accuracy regarding behavior on unseen data. The use of a substantial testing dataset allows for stronger model validation, ensuring that the model not only learns patterns specific to the training data but also generalizes well to new data.

The use of 30% of the data for testing ensures that the sample size is large enough to provide a reliable estimate of model performance, reduce variability in estimations, and yield more stable results. Thus, a 70% training data and 30% testing data split achieves a good balance between model performance and time efficiency during the learning or training process.

MODELLING USING NAIVE BAYES

```
# train a Gaussian Naive Bayes classifier on the training set
from sklearn.naive_bayes import GaussianNB


# instantiate the model
gnb = GaussianNB()


# fit the model
gnb.fit(X_train, y_train)
```

```
GaussianNB
GaussianNB()
```

After splitting the data, the next step is to build a classification model using the Naïve Bayes algorithm. During the Exploratory Data Analysis (EDA) phase of this study, no oversampling was performed. In this phase, the model is built using the original data without oversampling, applying the Naïve Bayes algorithm.

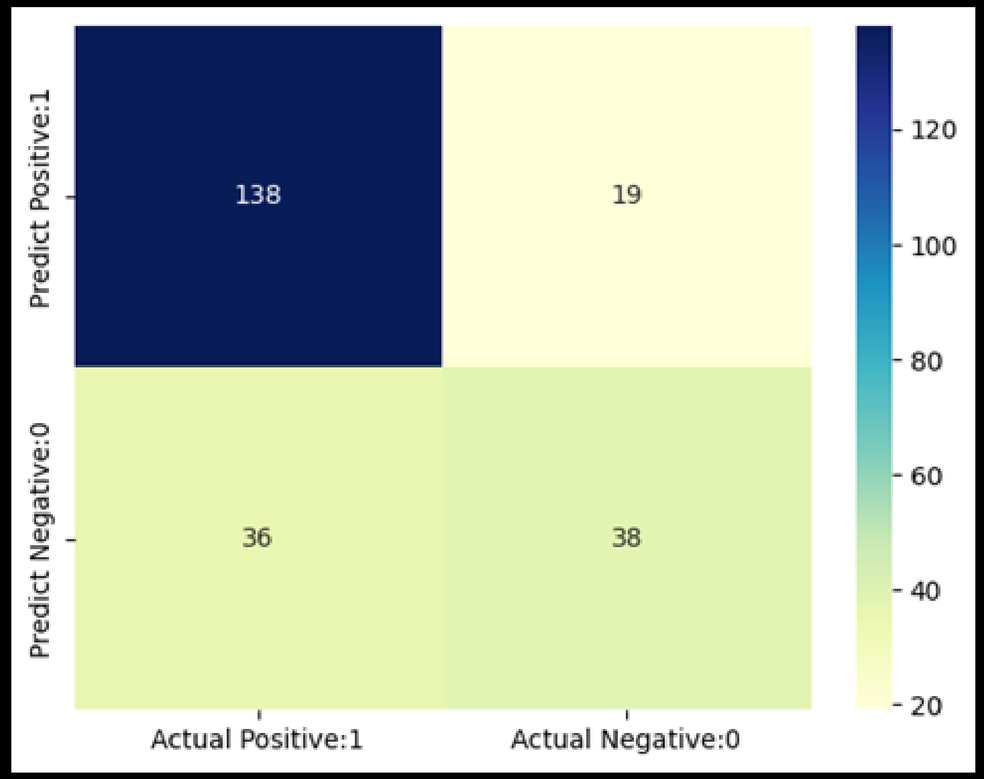
After building the Naïve Bayes classification model, the next step is to make predictions on the test data. In this phase, the previously trained Gaussian Naïve Bayes classification model is used to predict the response (label) for the test data, X_test, and to display the test data, X_test, on the output layer.

```
y_pred = gnb.predict(X_test)

y_pred
```

```
array([1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0])
```

EVALUATION MODELS



76.19%
Accuracy

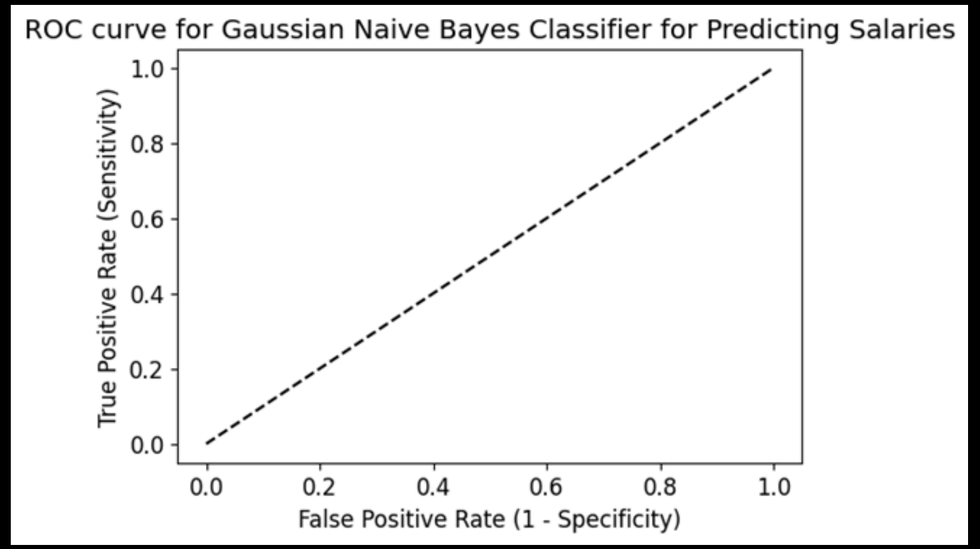
The next step is to calculate the model's accuracy. This function takes two arguments: y_test, which contains the actual labels of the test data, and y_pred, which contains the predicted labels generated by the model. The function will calculate how accurate the model's predictions are compared to the actual labels, resulting in an accuracy value of 76.19%.

From the confusion matrix above, the following values were obtained: true positive (TP) = 198 records, false positive (FP) = 36 records, true negative (TN) = 19 records, and false negative (FN) = 38 records. Accuracy is defined as the degree of closeness between the predicted values and the actual values.

	precision	recall	f1-score	support
0	0.79	0.88	0.83	157
1	0.67	0.51	0.58	74
accuracy			0.76	231
macro avg	0.73	0.70	0.71	231
weighted avg	0.75	0.76	0.75	231

The figure shows that for individuals without diabetes, the precision is 79%, recall is 88%, and the F1-score is 83%. On the other hand, for individuals with diabetes, the precision is 67%, recall is 51%, and the F1-score is 58%.

The accuracy of AUC is considered perfect when the AUC value reaches 1.00, and the accuracy becomes poor if the AUC value is below 0.500. In Figure 9, the AUC value is 0.8039, indicating that the model has a good ability to differentiate between positive and negative classes. This falls under the 'Poor Classification' diagnostic level.



AT RESULT

There are 9 attributes that influence the classification of the diabetes dataset: Pregnancies, Glucose, BloodPressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age, and Outcome. This study uses the Naïve Bayes method for classifying the diabetes dataset. Out of 769 raw data points, after data cleaning, there were no missing values, and through the necessary feature selection for diabetes dataset analysis, it was determined that all 769 data points were clean.

The classification results show that individuals without diabetes achieved a precision of 79%, recall of 88%, and an F1-score of 83%. In contrast, individuals with diabetes had a precision of 67%, recall of 51%, and an F1-score of 58%. Additionally, the Naïve Bayes model applied to the diabetes dataset resulted in 75% precision and 76% recall.

The classification evaluation yielded an accuracy of 75% in classifying the diabetes dataset, meaning the Naïve Bayes model performs reasonably well in terms of accuracy. However, there is room for improvement in accuracy, which could be achieved through ensemble methods or other techniques.

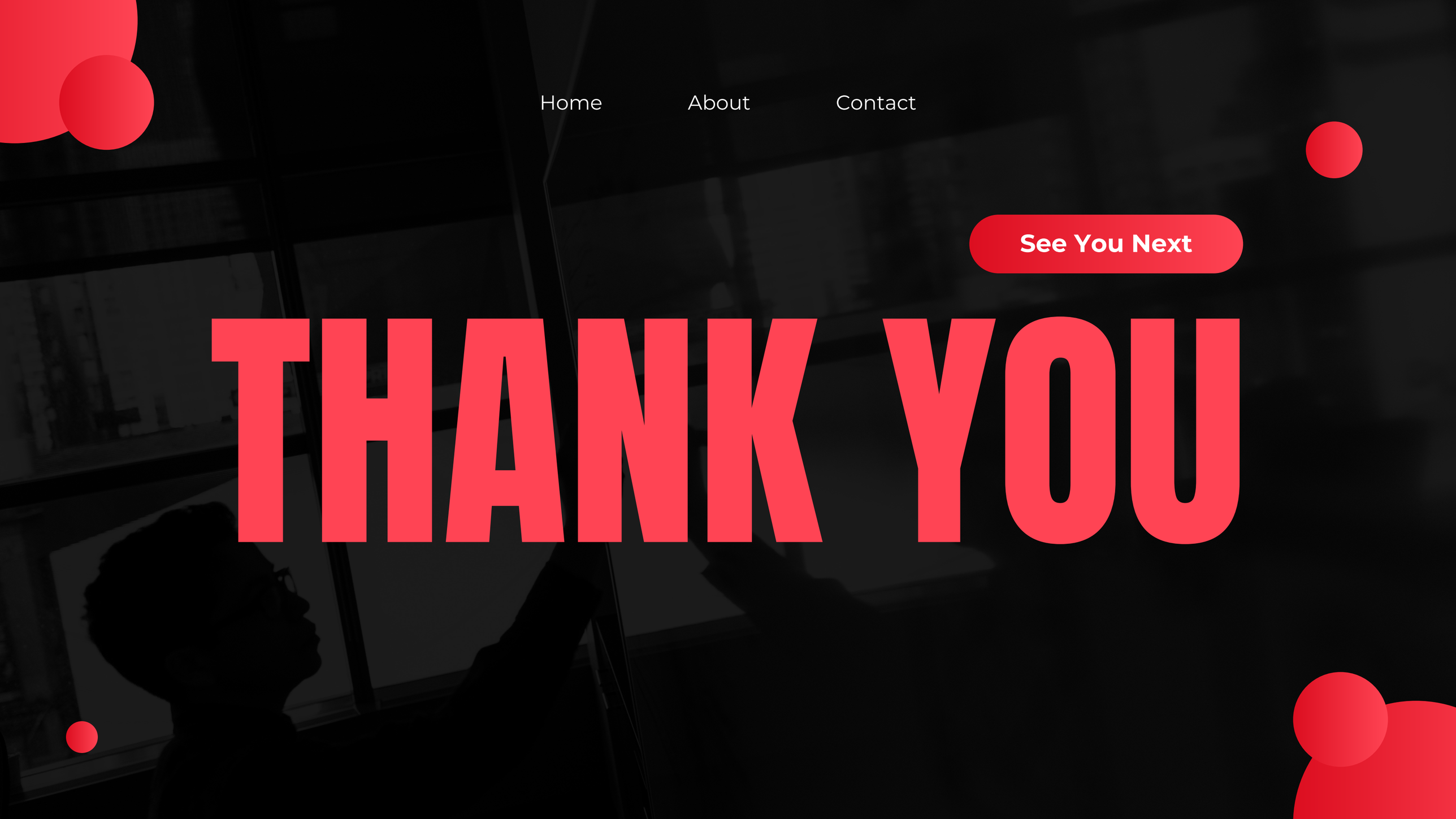
The ROC (Receiver Operating Characteristic) curve for the Naïve Bayes model applied to the diabetes dataset indicates that the Naïve Bayes algorithm has an AUC value of 0.8039, meaning the classification is considered good.

Home About Contact

SOURCE CODE :

Overcoming Common Obstacles

Read More


Home
About
Contact
See You Next
THANK YOU