A2. Regression

## Logistic Regression for the Diabetes Dataset

When selecting a machine learning algorithm for a particular dataset, it's crucial to consider the nature of the data, the problem at hand, and the strengths of various algorithms.

1. Nature of the Problem

The primary task with the diabetes dataset is binary classification: determining whether an individual has diabetes (1) or does not have diabetes (0). Logistic regression is specifically designed for binary classification tasks. It directly estimates the probability of the default class, making it inherently suitable for this kind of problem.

2. Interpretability

Logistic regression is one of the most interpretable machine learning models. The model provides coefficients for each feature, which can be directly interpreted in terms of odds ratios. For instance, you can understand how a unit change in glucose level affects the odds of having diabetes. This interpretability is crucial in medical applications where understanding the influence of different factors is important for both practitioners and patients.

3. Performance

Logistic regression performs well when the relationship between the features and the outcome is approximately linear. Given the nature of the diabetes dataset, which includes features such as glucose level, BMI, and age, there is reason to believe that the relationship between these features and the likelihood of diabetes could be approximated linearly.

In the given results:

The training accuracy was 77.65%, and the test accuracy was 78.79%.

These are respectable performance metrics, indicating that the logistic regression model has learned the underlying patterns in the training data and generalizes reasonably well to the test data.

## Training the Logistic Regression Model

1. Logistic Regression is a statistical model used for binary classification tasks. It estimates the probability that a given input belongs to a certain class. It’s commonly used in scenarios where the outcome is binary, such as predicting whether a person has a disease (yes/no).

First I split the features (X) and targets (Y) of the dataset into training and testing data, I choose a ration 70% of the data for training and 30% of the data for testing.

2. I trained the Logistic Regression Model by importing the model from scikit-learn and used the training sets , both for the features (X) and target (Y) to fit the model.

3. Regression Coefficients:

* Intercept: The intercept is a constant term in the regression equation. It represents the log-odds of the outcome when all predictors are zero.
* Coefficients: These are weights assigned to each feature in the model, indicating how much each feature contributes to the outcome.

For the training data upon fitting the model I received the following intercept and regression coefficients:

* **Intercept: -7.58577284**
* **Coefficients: [[ 0.12628558 0.03362097 -0.01347223 -0.0029405 -0.00112627 0.08693123 0.68107089 0.00703222 ]]**

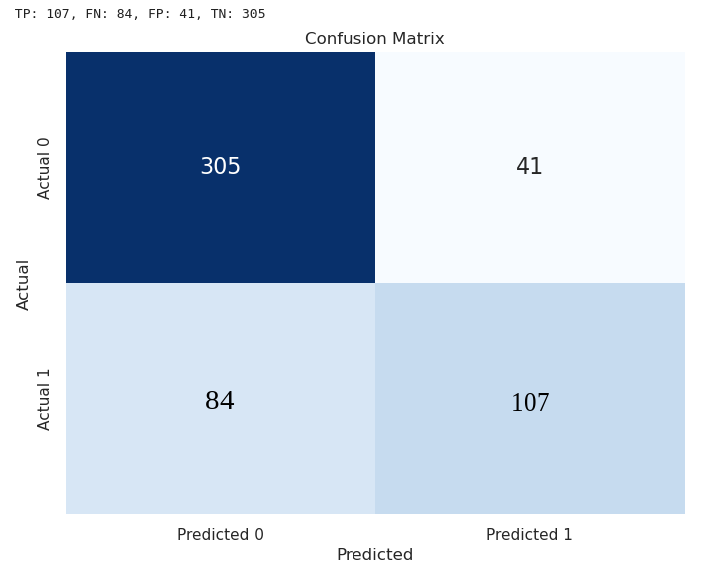
## Model Evaluation

1. Using another scikit-learn module, I found out the accuracy score on the training data.

**Accuracy score of training data: 0.776536312849162 (77%)**

2. I also used a confusion matrix algorithm for evaluating the model’s performance. **Confusion matrix**, an essential tool in evaluating the performance of a classification model, encompasses elements representing True Positive, True Negative, False Positive, and False Negative values. Furthermore, validation techniques ensure the model’s robustness, guard against overfitting, and enhance its generalization capabilities.

**These are the results of the confusion matrix algorithm: TP: 107, FN: 84, FP: 41, TN: 305.**

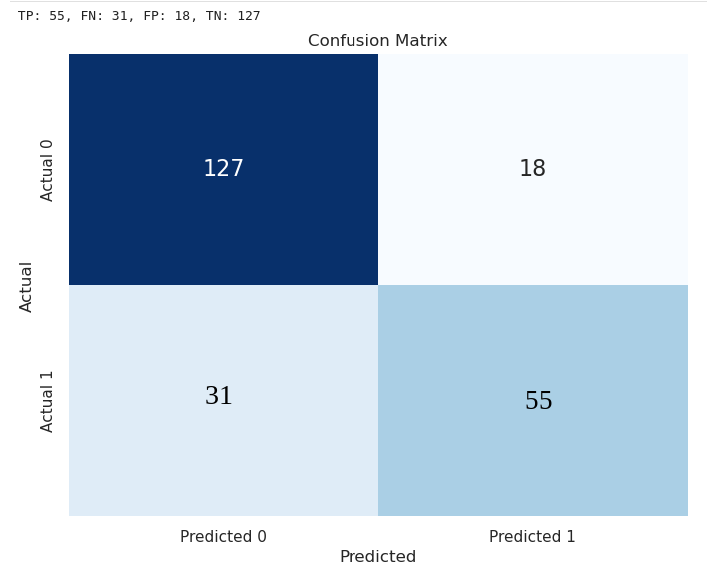


3. I used the predict method on the LogisticRegression model from scikit-learn to predict the target values for the test set. Based on that I got this accuracy score for the test data:

**Accuracy score of test data: 0.7878787878787878 (78%)**

4. Once again I used the confusion matrix algorithm for the model’s performance on the test data.

**These are the results of confusion matrix algorithm: TP: 55, FN: 31, FP: 18, TN: 127.**

5. Precision, the ratio of true positives to the sum of true positives and false positives.

Formula: TP / ( TP + FP )

It indicates the accuracy of positive predictions (the proportion of predicted positives that are actual positives).

**For my trained model the precision is: 0.75**

6. Recall (Sensitivity or True Positive Rate), the ratio of true positives to the sum of true positives and false negatives.

Formula: TP / ( TP + FN )

It measures the model’s ability to correctly identify actual positives.

**For my trained model the recall is: 0.64**

## Model Performance Analysis

**1.Model Accuracy:**

The accuracy of around 77-78% on both the training and test sets indicates that the model is reasonably good at predicting whether an individual has diabetes. However, accuracy alone is not always the best metric, especially in imbalanced datasets.

**2. Confusion Matrix**

The confusion matrix provides a more detailed insight into the performance. There are a considerable number of false negatives (FN) in both training and test sets, indicating that the model is missing some cases of diabetes.

The number of false positives (FP) is relatively lower, which is good because fewer non-diabetic individuals are incorrectly classified as diabetic.

**3. Precision and Recall**

The precision of 75% is quite good, suggesting that when the model predicts diabetes, it is correct 75% of the time

The recall of 64% indicates that the model is able to identify 64% of the actual diabetes cases. This is somewhat lower and suggests that the model might be improved to better capture more of the actual positive cases.

**4. Feature Importance:**

The largest positive coefficient, 0.68107089 for the 7th feature (Diabetes Pedigree Function) suggests that this feature has a strong positive impact on the likelihood of diabetes. In contrast, the negative coefficients suggest a negative relationship with diabetes.

## Jupyter Notebook source code:

<https://github.com/alexban14/DataMining_Diabetes_DS>

## Resources utilized:

https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

https://www.datacamp.com/tutorial/understanding-logistic-regression-python