FEDERAL REPUBLIC OF GERMANY
UNIVERSITY OF EUROPE FOR APPLIED SCIENCES
POTSDAM CAMPUS

BREAST CANCER DATASET ANALYSIS

Sebastian Russo
Career: Software Engineering
Artificial Intelligence
Semester: Summer

Potsdam, May 2024

Analyze the dataset and explain whether this is a Regression or Classification and which algorithm will you apply for this problem. Explain the concept of dependent and independent variables and which is the dependent variable in this dataset.

This particular dataset consists of a Binomial Classification problem because, as stated in the assignment itself, the key challenge against the detection of breast cancer is how to classify tumors from malignant (cancerous) or benign (non-cancerous), meaning, that in this case we require to predict the classification of a data point between 2 predefined classes (M and B) based off other factors called independent variables by using the Logistic Regression Machine Learning model.

The independent variables are those that are manipulated and can exist by themselves whereas the dependent variables depend on the effects of the independent ones and wouldn't exist without them. Therefore, the dependent variable in this dataset would be the "diagnosis" column and it would completely depend on the independent variables (which are all the other columns except "id" which is taken as the index).

IMPORTANT

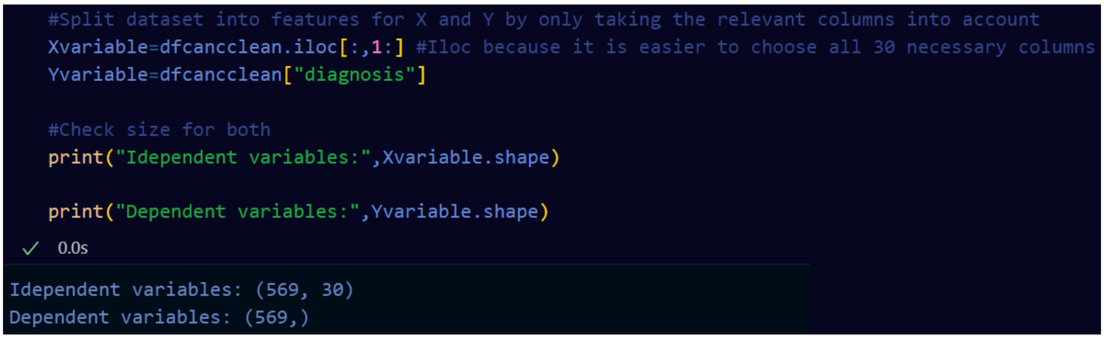
We decided to utilize Feature scaling towards the Xtrain and Xtest to further improve the model performance

Implement the correct algorithm required for this task (Linear or Logistic):

(a) Extract the X feature and Y target (Include explanation in the Word document about X feature and Y target

After importing the necessary libraries, Reading and analyzing the dataset, checking for missing values and cleaning the data as necessary, we can finally start with the implementation of the Logistic Regression model and of course, the first step is to separate independent variables from the dependent ones.

This means that we need to take the dataset and “create 2 sub-datasets”, 1 that has only the column for the dependent variables and the other that has all the columns for the independent variables that affect the dependent ones. In python, we used the simple ‘dataset[“columnname”]’ to pick the “diagnosis” column that has the dependent variables and then we used the ‘dataset.iloc[:,1:]’ to pick all the other columns given that there are 30 of them and it is easier and faster this way instead of manually writing the name of each of them individually.



```
#Split dataset into features for X and Y by only taking the relevant columns into account
Xvariable=dfcancclean.iloc[:,1:] #Iloc because it is easier to choose all 30 necessary columns
Yvariable=dfcancclean["diagnosis"]

#Check size for both
print("Idependent variables:",Xvariable.shape)

print("Dependent variables:",Yvariable.shape)
```

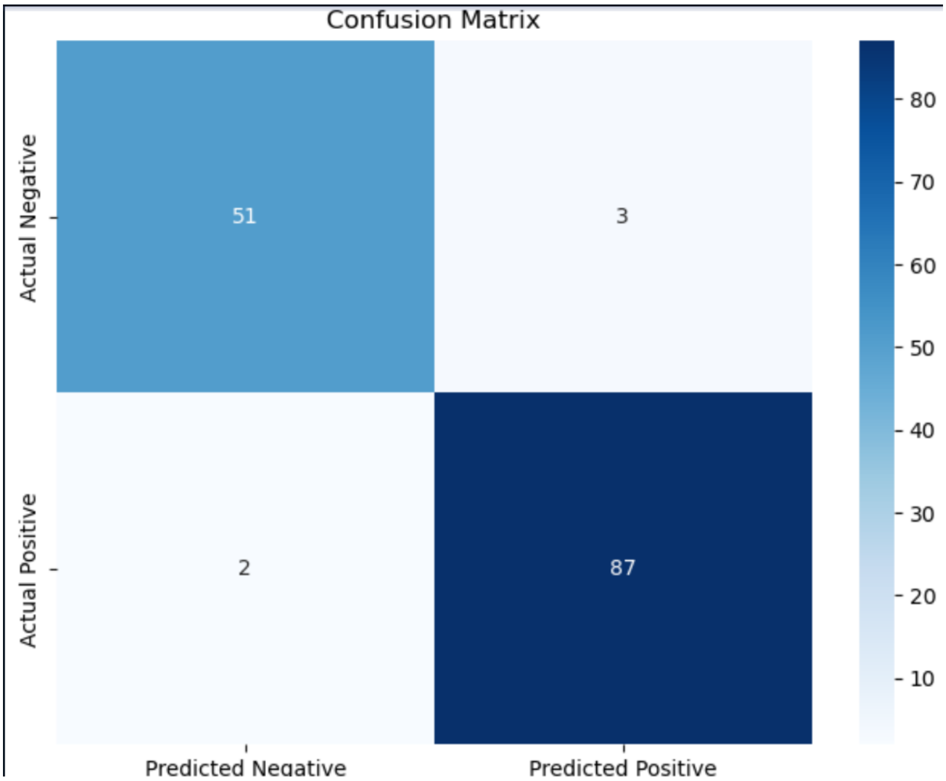
```
Idependent variables: (569, 30)
Dependent variables: (569,)
```

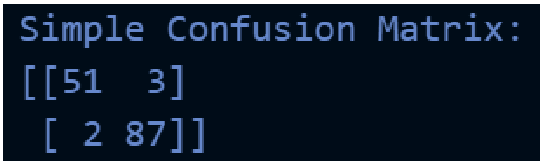
As we can see the split was successful (the “id” column was set as the index_col, that is why we used .iloc[:,1:] instead of .iloc[:,2:])

Result

(a) Show the result with the Confusion Matrix (In your Word document, explain the values obtained in this matrix)

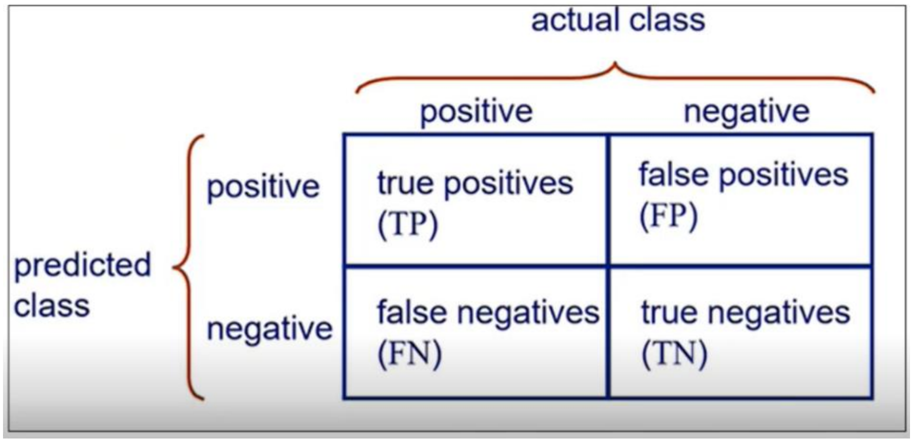
After successfully running the Logic Regression model, we end up with a Confusion Matrix that corroborates that the model is very precise (we took the liberty to use a heatmap to visualize it better):





```
Simple Confusion Matrix:
[[51  3]
 [ 2 87]]
```

Given that a Confusion Matrix is organized like:

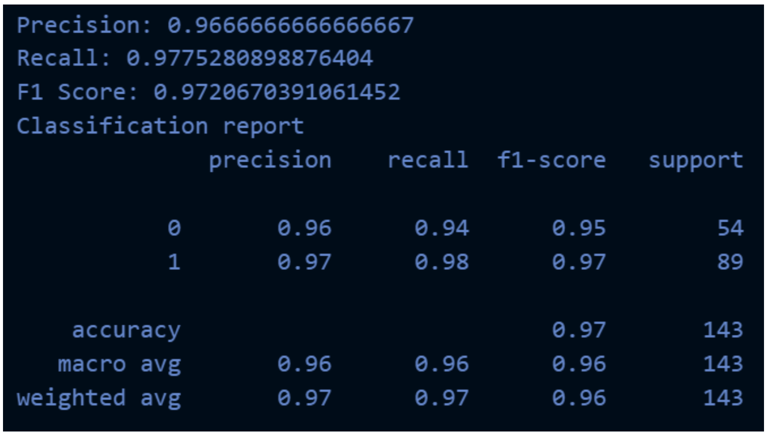


		actual class	
		positive	negative
predicted class	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

Then we can conclude that this model has 51 True Positives (TP), 3 False Positives (FP), 2 False Negatives (FN) and 87 True Negatives (TN). Even without calculating the other parameters that evaluate the model we can infer that is a very good model because it has more correctly predicted results (TP and TN) than incorrectly predicted results (FP and FN), but of course, to know for sure it is necessary to calculate parameters like Precision, Recall and F1 score.

Result

(b) Calculate and print the Precision, Recall and F1 Score (Explanation – 1 Mark)



```
Precision: 0.9666666666666667
Recall: 0.9775280898876404
F1 Score: 0.9720670391061452
Classification report
              precision    recall  f1-score   support

           0       0.96      0.94      0.95        54
           1       0.97      0.98      0.97        89

    accuracy                           0.97       143
   macro avg       0.96      0.96      0.96       143
weighted avg       0.97      0.97      0.96       143
```

- Precision: Given by TP / (TP + FP), this parameter gives a value that tells the proportion of POSITIVE predictions that were actually POSITIVE themselves. In this case, the value for class 0 (Malignant) is 0.96 while for class 1 (Benign) is 0.97, meaning the model was able to correctly predict Positives consistently for both classes.
- Recall: Given by TP / (TP + FN), this parameter gives a value that tells the proportion of actual POSITIVE cases that are correctly predicted as POSITIVE. In this case, the value for class 0 (Malignant) is 0.94 while for class 1 (Benign) is 0.98, meaning the model was able to correctly predict as positive a good amount of actual positive cases for both classes.
- F1 score: Given by 2 * (Precision * Recall) / (Precision + Recall), this parameter gives a value that tells the harmonic mean of precision and recall just to evaluate the performance of the classifier itself. In this case, the value for class 0 (Malignant) is 0.95 while for class 1 (Benign) is 0.97.

With all this in mind, we can say that the model is very good with an Accuracy of 0.9667 and being slightly better at predicting class 1 (Benign) than class 0 (Malignant). This extremely high accuracy might be because of the fact that there are 30 columns of independent variables determining the "diagnosis", especially when comparing it to the previous datasets we have worked with during the lectures because they have significantly less columns than this dataset.

On a different note, in this particular dataset fortunately, the cases of Benign tumors are more than those Malignant, so that is good for the patients.

Conclusion

List your learnings from this assignment and your overall understanding of the ML algorithm you implemented.

1. For all datasets, it is recommended to always have an index column, if there is already a column that meets said requirement just use ‘index_col= x’ to assign it as such and if there is no column that meets said requirement create it
2. The more independent columns (independent variables) that are used for the model, the more accurate and precise it becomes (in this case they are 30)
3. For all datasets (especially for large ones), it is recommended to always check for missing values in the dataset and if there are, to always deal with them so they don’t tamper with the Logistic or Linear Regression model
4. Always convert categorical data into numerical variables so the model can correctly work, whether its int or double
5. Use .iloc[:,:] to select the columns when there are too many to be written manually by manipulating the ‘:’
6. By default, the Logistic Regression model can go through a maximum of 100 iterations, therefore in the case this limit is exceeded, it is recommended to use “max_iter= x)” to increase the limit and avoid the Convergence warning
7. It is recommended to adapt the heatmap to get a better visualization of the Confusion Matrix and therefor it’s easier to analize