Classification Using Machine Learning

Sanidhya Chopde
Department of Computer Science
University at Buffalo
Buffalo, NY 14260
schopde@buffalo.edu

Abstract

The aim is to perform classification using machine learning on the Wisconsin Diagnostic Breast Cancer dataset to determine whether the suspected cells are benign or malign. To achieve this a two-class problem will be implemented using logistic regression as the classifier.

1 Introduction

In this project the task is to perform classification using Machine Learning. I'll be using features which are pre-computed from images of a fine needle aspirate (FNA) to classify suspected FNA cells to Benign (class 0) or Malignant (class 1) using logistic regression as the classifier. The dataset used for training, testing and validation is the Wisconsin Diagnostic Breast Cancer (wdbc.dataset).

2 Dataset Definition

For this project we will be using the Wisconsin Diagnostic Breast Cancer (wdbc.dataset) for training, testing and validation. The dataset contains a total of 569 instances with 32 attributes, of which, 2 are ID and diagnosis (B/M) and the other 30 are real valued input features. The. features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Computed features describe the following characteristics of the cell nuclei present in the image:

1	Radius (mean of distances from center to points on the perimeter)
2	Texture (standard deviation of gray-scale values)
3	Perimeter
4	Area
5	Smoothness (local variation in radius lengths)
6	Compactness (perimeter2/area − 1.0)
7	Concavity (severity of concave portions of the contour)
8	Concave points (number of concave portions of the contour)
9	Symmetry
10	Fractal dimension ("coastline approximation" - 1)

Table 1. Features in the dataset

The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features.

3 Pre-Processing

In the pre-processing phase the dataset is required to be converted into a numpy matrix or a pandas dataframe so as to make the dataset compatible for performing mathematical operations. After converting the dataset into a numpy matrix or a pandas dataframe, the values in the "Diagnosis" column i.e. B/M are mapped to the values 0/1. After this the first 2 columns namely "ID" and "Diagnosis" are dropped from the dataframe or the matrix to obtain a dataset of dimensions 569 * 30. This matrix is further split into training, testing and validation data in the ratio 8:1:1 respectively. On performing the above operations, the pre-processing phase comes to an end for the dataset.

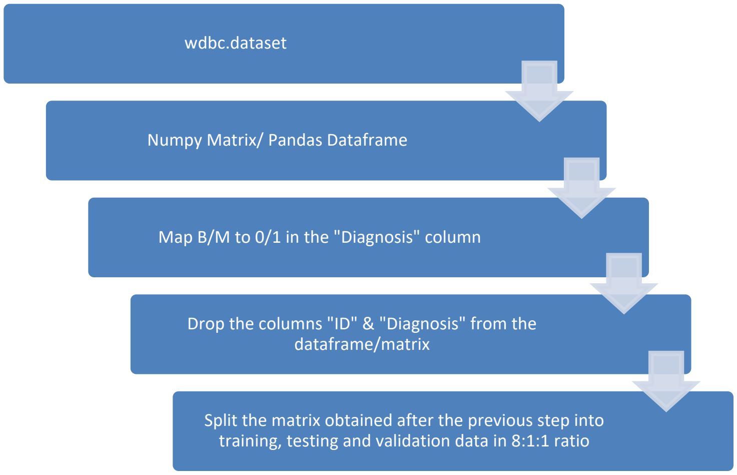
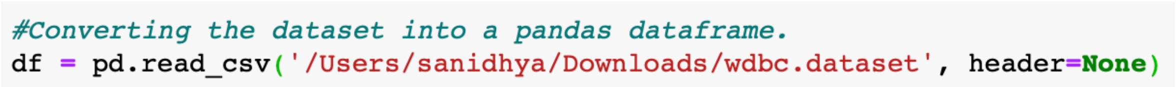

wdbc.dataset
Numpy Matrix/ Pandas Dataframe
Map B/M to 0/1 in the "Diagnosis" column
Drop the columns "ID" & "Diagnosis" from the dataframe/matrix
Split the matrix obtained after the previous step into training, testing and validation data in 8:1:1 ratio

Figure 1. Flowchart for Pre-Processing

4 Architecture

4.1 Pandas Dataframe

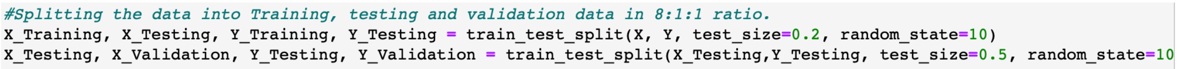
We need to convert the given dataset into a pandas dataframe or a numpy matrix to make the dataset suitable for mathematical calculations. Pandas dataframe has been used in this particular project.



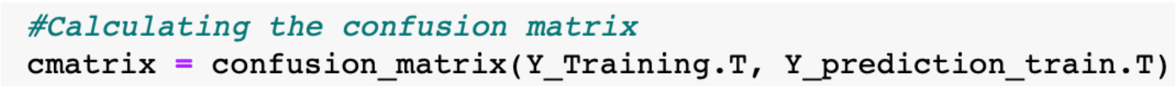
```
#Converting the dataset into a pandas dataframe.
df = pd.read_csv('/Users/sanidhya/Downloads/wdbc.dataset', header=None)
```

4.2 Scikit-learn

The scikit-learn library in python has been used to perform several operations in this project such as splitting the data into training, testing and validation data and also to obtain the confusion matrix.



```
#Splitting the data into Training, testing and validation data in 8:1:1 ratio.
X_Training, X_Testing, Y_Training, Y_Testing = train_test_split(X, Y, test_size=0.2, random_state=10)
X_Testing, X_Validation, Y_Testing, Y_Validation = train_test_split(X_Testing,Y_Testing, test_size=0.5, random_state=10
```



```
#Calculating the confusion matrix
cmatrix = confusion_matrix(Y_Training.T, Y_prediction_train.T)
```

4.3 Logistic Regression

When you apply sigmoid to linear regression, you arrive at logistic regression. Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

4.3.1 Sigmoid function

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$S(z) = \frac{1}{1 + e^{-z}}$$

4.3.2 Cost Function

It is the average of the loss functions of the entire training set. For gradient descent we use a cost function called the cross-entropy cost function.

$$L(p, y) = \ -(ylogp + (1 - y)\log(1 - p))$$

4.3.3 Gradient Descent

To reduce our cost, we use gradient descent. Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model. Parameters refer to coefficients in Linear Regression and weights in neural networks.

4.3.4 Calculating Accuracy, Precision and Recall

Accuracy, precision and recall are calculated using the following formulas respectively:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives

5 Result

When we use different hyperparameters for the cost function we get the following graphs:

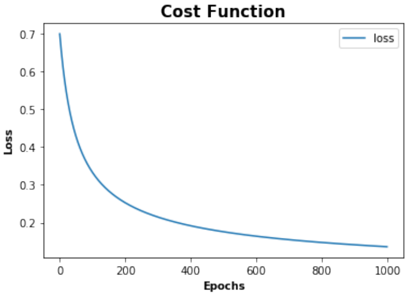


Figure 2. Epochs =1000, Learning rate = 0.005

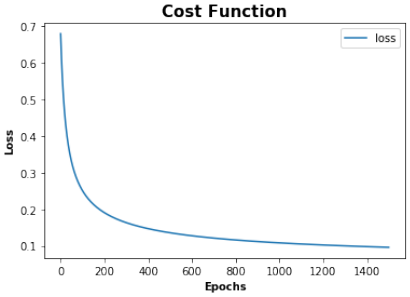


Figure 3. Epochs =1500, Learning rate = 0.01

The following is the graph obtained for training accuracy vs number of epochs:

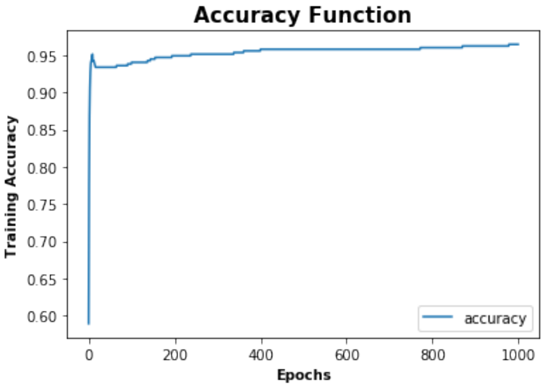


Figure 4. Training accuracy vs Number of epochs

When computed with the test datatset we get the following results:

Testing accuracy:	100.0%
Accuracy:	1.0
Precision:	1.0
Recall	1.0

6 Conclusion

In this project classification using machine learning was performed on the Wisconsin Diagnostic Breast Cancer dataset and it was determined whether the suspected cells were benign or malign. For achieving the result, a two-class problem was implemented using logistic regression as the classifier. Accuracy, precision and recall was calculated for the testing data and a training accuracy graph was plotted using the training data.

Acknowledgements

I am extremely grateful to Professor Sargur Srihari for teaching all the necessary concepts related to logistic regression. I would also like to thank all the TA's of the course for helping me at every step during the course of this project.

References

[1] Professor Sargur Srihari (2019) Linear Regression lecture slides.
[2] Pandas Documentation (https://pandas.pydata.org/pandas-docs/stable/getting_started/dsintro.html)
[3] ML Cheatsheet (https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html)
[4] Internal Pointers(https://www.internalpointers.com/post/cost-function-logistic-regression)
[5]Towards Datascience (https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a)
[6] StackOverflow (https://stackoverflow.com/questions/4702249/is-there-a-way-in-python-to-return-a-value-via-an-output-parameter)