# **Binary Classification Task Evaluation**

Krishna Kumar Veeraputhiran

Grand Canyon University

DSC 540 – O500 – Machine Learning for Data Science

Dr. Aiman Darwiche

September 1,2021

# **Binary Classification Task Evaluation**

Binary classification is the task of classifying the data into two classes. This is a supervised machine-learning algorithm where the target variable is discrete. Evaluating an algorithm is as important as building one. For classification algorithm, there are multiple ways to evaluate the algorithm based on how the classification is done. In this report, we will see two important evaluations methods namely Misclassification Errors and Maximizing the margin between the classes.

**Misclassification Error**

According to Gopal (2019), Misclassification error rate can be given as the ratio of the total number of instances classified incorrectly to that of the total number of instances. Let’s consider a target variable yi (where yi ε [0,1]) that is being predicted by a model h(w,xi). The predicted outcome of the variable can be given as, y̅I (where y̅I ε [0,1]) FOR I= 1,..,N.

A 0% error can be given as yi - y̅I =0 (for all values of i). To simplify Misclassification error can be given as,

**Number of Data points having yi - y̅I <>0 / Total number of data points.**

The misclassification error can be used to calculate the accuracy of the model on how well the classification algorithm has performed. A simple way to get the misclassification is by the use of confusion matrix. For a binary classification, the outcome will be either 0,1 and hence the confusion matrix will look like below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted Category | | |
| Actual Category |  | **0** | **1** |
| **0** | *True Negatives*  Predicted 0  Actual 0 | *False positives*  Predicted 1  Actual 0 |
| **1** | *False Negatives*  Predicted 0  Actual 1 | *True Positives*  Predicted 1  Actual 1 |

The True negatives and True Positives are where the actual and predicted values match, whereas the False negative and False Positive are where they don’t match. The accuracy can be given as,

**Accuracy** = True negative + True Positive / (True Negative + True Positive + False Negative+

False Positive )

**Misclassification Error** = 1 – Accuracy

Misclassification error treats misclassification of all classes equally, which cannot be true. This happens when the training dataset has majority of the data under one class and the other class is rare. For example, let us take the case of banking transactions to identify fraudulent transaction. Let us say there are a total of 100 transaction out of which there is one fraudulent transaction. Now if the classification model predicts all the 99 transactions correctly and misinterprets the fraudulent transaction incorrectly, then the overall accuracy of the model will be 99% with the misclassification error at 1%.Generally a model with 99% accuracy is considered efficient, however in our case there was just one fraudulent transaction and the model failed to predict it accurately. This may lead to overall failure of the model in predicting the transactions. Using the Confusion matrix in this case will help us focus on the negative classes and segregate them appropriately.

**Maximizing the Margin**

Maximizing the margin is the performance criteria used in distinguishing two classes in Support Vector Machine (SVM) algorithm. SVM is a supervised machine learning algorithm that can be used for classification. Under this algorithm each data point (support vectors) is plotted in an n-dimensional (n being the total number of features and the features being the values of the coordinates) space. Classification is performed by finding a hyperplane that differentiates between the classes. SVM algorithm is used to classify the data that are also non-linear in nature. Not all classification can be linear in nature and hence SVM plays a crucial role in classifying non-linear cases. One important point to note is that the SVM algorithm classifies the classes prior to maximizing the margin. In order to understand how the margins can be maximized we will take the case of training data D, that is linearly separable. Here D can be given as,

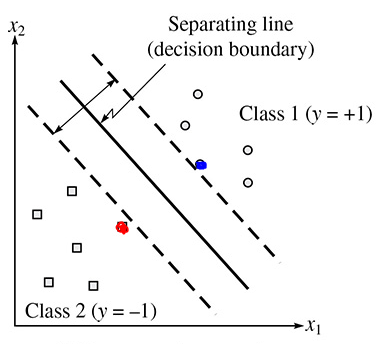
D = { (xi, yi) } for i=1,..n

Here x = [x1,x2,..xn]T is the n-dimensional input vector and y being it’s class label where y ε [+1,-1], where +1 denotes Class 1 and -1 being class 2. According to Gopal (2019), for the above dataset, SVM finds a linear function of the form,

g(x) = wTx + w0

So the input vector xi will be assigned to class 1 if g(x) >0 and class 2 if g(x) < 0. In this case the hyperplane can be given as, **wTx + w0 =0.**

For data that is linearly separable we can identify multiple hyperplanes that can separate the vectors, however the SVM helps us in finding the hyperplane that has the largest margin. Finding the hyperplane with largest margin is crucial to avoid/minimize training errors. Below is a depiction of a Linear SVM on 2 dimension (2 input variables) with a one dimension hyperplane from Gopal (2019),



Here Margin is nothing but the distance between the nearest data point (from each class highlighted in red and blue) to the hyper-plane. Higher margin hyper-plane makes the classification robust and minimizes the Euclidean norm of the weight vector

**Conclusion**

Choosing the binary class evaluation depends on the problem statement and the model we are using to predict the classification. For data that is balanced and the training data that has an equal mix of both the classes, then using the Misclassification error would help us in getting the accuracy of the model. However in case of training data that has an unbalanced mix of the classes can lead to inaccurate predictions which may not reflect in the overall accuracy but would have got the key predictions wrong. These are the cases a contingency matrix might be more helpful than the Misclassification error evaluation. On the other hand SVM (Maximizing the margin) can handle unbalanced data without any issues. Class-weighted SVM is designed to deal with unbalanced data by assigning higher misclassification penalties to training instances of the minority class. Maximizing the Margin is a more robust way to evaluate classification model SVM where classification can be done by creating a hyper-plane in n-dimension for the input vectors. This Hyper-plane can be both linear and non-linear in nature. So whenever we use a model to predict non-linear dataset with n-dimensions using the Maximizing the margin as an evaluation tool would be of greater advantage. From above I can conclude by saying given the data is unbalanced or non-linear using SVM algorithm seems more feasible and maximizing the margin would be the better way to evaluate it.

# **References**

Gopal, M. (2019). *Applied machine learning*. McGraw-Hill Education.

*Support Vector Machine*. Lecture 9: SVM. (n.d.). https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote09.html.

Rebours, P., & Khoshgoftaar, T. M. (n.d.). *Misclassification rate*. Misclassification Rate - an overview | ScienceDirect Topics. https://www.sciencedirect.com/topics/engineering/misclassification-rate.