**Classification**



**WELCOME INTELLIGENT SYSTEMS STUDENTS!**

**I. INTRODUCTION:**

We begin this module by developing a motivation for learning about what is Classification. This lesson will discuss about Linear Classifier, How Binary Classifier works and How to measure the performance of classifier..

**II. OBJECTIVES:**

After completing this course, you will be able to learn about:

* Determine the concept of Classification.

**III. DEVELOPMENT OF THE LESSON**

**COURSE MATERIALS:**

## What is Linear Classifier?

The two most common supervised learning tasks are linear regression and linear classifier. Linear regression predicts a value while the linear classifier predicts a class. This tutorial is focused on Linear Classifier.

Classification problems represent roughly 80 percent of the machine learning task. Classification aims at predicting the probability of each class given a set of inputs. The label (i.e., the dependent variable) is a discrete value, called a class.

1. If the label has only two classes, the learning algorithm is a binary classifier.
2. Multiclass classifier tackles labels with more than two classes.

For instance, a typical binary classification problem is to predict the likelihood a customer makes a second purchase. Predict the type of animal displayed on a picture is multiclass classification problem since there are more than two varieties of animal existing.

The theoretical part of this tutorial puts primary focus on the binary class. You will learn more about the multiclass output function in a future tutorial.

In this tutorial, you will learn

* [What is Linear Classifier?](https://www.guru99.com/linear-classifier-tensorflow.html#1)
* [How Binary classifier works?](https://www.guru99.com/linear-classifier-tensorflow.html#2)
* [How to Measure the performance of Linear Classifier?](https://www.guru99.com/linear-classifier-tensorflow.html#3)
* [Accuracy](https://www.guru99.com/linear-classifier-tensorflow.html#4)
* [Confusion matrix](https://www.guru99.com/linear-classifier-tensorflow.html#5)
* [Precision and Sensitivity](https://www.guru99.com/linear-classifier-tensorflow.html#6)
* [Linear Classifier with TensorFlow](https://www.guru99.com/linear-classifier-tensorflow.html#7)
* [Step 1) Import the data](https://www.guru99.com/linear-classifier-tensorflow.html#8)
* [Step 2) Data Conversion](https://www.guru99.com/linear-classifier-tensorflow.html#9)
* [Step 3) Train the Classifier](https://www.guru99.com/linear-classifier-tensorflow.html#10)
* [Step 4) Improve the model](https://www.guru99.com/linear-classifier-tensorflow.html#11)
* [Step 5) Hyperparameter:Lasso & Ridge](https://www.guru99.com/linear-classifier-tensorflow.html#12)

## How Binary classifier works?

You learned in the previous tutorial that a function is composed of two kind of variables, a dependent variable and a set of features (independent variables). In the linear regression, a dependent variable is a real number without range. The primary objective is to predict its value by minimizing the mean squared error.

For a binary task, the label can have had two possible integer values. In most case, it is either [0,1] or [1,2]. For instance, the objective is to predict whether a customer will buy a product or not. The label is defined as follow:

* Y = 1 (customer purchased the product)
* Y = 0 (customer does not purchase the product)

The model uses the features X to classify each customer in the most likely class he belongs to, namely, potential buyer or not.

The probability of success is computed with **logistic regression**. The algorithm will compute a probability based on the feature X and predicts a success when this probability is above 50 percent. More formally, the probability is calculated as follow:

[https://www.guru99.com/images/tensorflow/082918_1030_LinearClass6.jpg](https://www.guru99.com/images/tensorflow/082918_1030_LinearClass6.jpg)

where 0 is the set of weights, the features and b the bias.

The function can be decomposed into two parts:

* The linear model
* The logistic function

**Linear model**

You are already familiar with the way the weights are computed. Weights are computed using a dot product:[https://www.guru99.com/images/tensorflow/082918_1030_LinearClass7.jpg](https://www.guru99.com/images/tensorflow/082918_1030_LinearClass7.jpg) Y is a linear function of all the features xi. If the model does not have features, the prediction is equal to the bias, b.

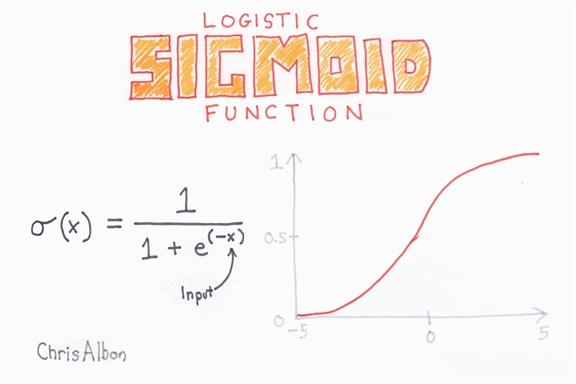
The weights indicate the direction of the correlation between the features xi and the label y. A positive correlation increases the probability of the positive class while a negative correlation leads the probability closer to 0, (i.e., negative class).

The linear model returns only real number, which is inconsistent with the probability measure of range [0,1]. The logistic function is required to convert the linear model output to a probability,

**Logistic function**

The logistic function, or sigmoid function, has an S-shape and the output of this function is always between 0 and 1.

[https://www.guru99.com/images/tensorflow/082918_1030_LinearClass8.jpg](https://www.guru99.com/images/tensorflow/082918_1030_LinearClass8.jpg)



It is easy to substitute the output of the linear regression into the sigmoid function. It results in a new number with a probability between 0 and 1.

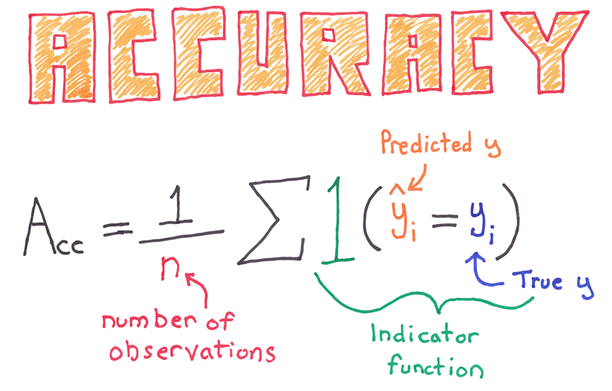
The classifier can transform the probability into a class

* Values between 0 to 0.49 become class 0
* Values between 0.5 to 1 become class 1

## How to Measure the performance of Linear Classifier?

### Accuracy

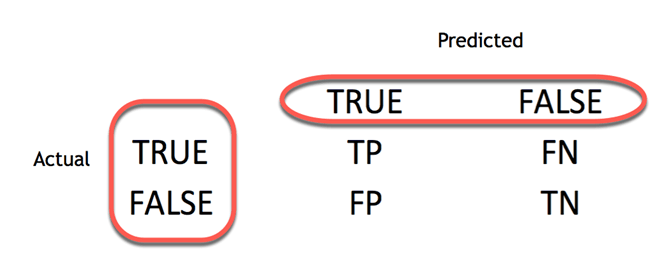
The overall performance of a classifier is measured with the accuracy metric. Accuracy collects all the correct values divided by the total number of observations. For instance, an accuracy value of 80 percent means the model is correct in 80 percent of the cases.



You can note a shortcoming with this metric, especially for imbalance class. An imbalance dataset occurs when the number of observations per group is not equal. Let's say; you try to classify a rare event with a logistic function. Imagine the classifier tries to estimate the death of a patient following a disease. In the data, 5 percent of the patients pass away. You can train a classifier to predict the number of death and use the accuracy metric to evaluate the performances. If the classifier predicts 0 death for the entire dataset, it will be correct in 95 percent of the case.

### Confusion matrix

A better way to assess the performance of a classifier is to look at the confusion matrix.



The confusion matrix visualizes the accuracy of a classifier by comparing the actual and predicted classes. The binary confusion matrix is composed of squares:

* TP: True Positive: Predicted values correctly predicted as actual positive
* FP: Predicted values incorrectly predicted an actual positive. i.e., Negative values predicted as positive
* FN: False Negative: Positive values predicted as negative
* TN: True Negative: Predicted values correctly predicted as actual negative

From the confusion matrix, it is easy to compare the actual class and predicted class.

### Precision and Sensitivity

The confusion matrix provides a good insight into the true positive and false positive. In some case, it is preferable to have a more concise metric.

**Precision**

The precision metric shows the accuracy of the positive class. It measures how likely the prediction of the positive class is correct.

[https://www.guru99.com/images/tensorflow/082918_1030_LinearClass9.jpg](https://www.guru99.com/images/tensorflow/082918_1030_LinearClass9.jpg)

The maximum score is 1 when the classifier perfectly classifies all the positive values. Precision alone is not very helpful because it ignores the negative class. The metric is usually paired with Recall metric. Recall is also called sensitivity or true positive rate.

**Sensitivity**

Sensitivity computes the ratio of positive classes correctly detected. This metric gives how good the model is to recognize a positive class.

[https://www.guru99.com/images/tensorflow/082918_1030_LinearClass10.jpg](https://www.guru99.com/images/tensorflow/082918_1030_LinearClass10.jpg)



We had just finished the discussion on Classification. Let’s move on to the next higher level of activity/ies or exercise/s that demonstrate your potential skills/knowledge of what you have learned.

**IV. ANALYSIS, APPLICATION AND EXPLORATION**

✂

**ACTIVITY 1**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Year & Section: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Choose the correct answer:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Which of the following is **true** related to ‘Satisfiable’ property?   |  |  | | --- | --- | | (a) | A statement is satisfiable if there is some interpretation for which it is false | | (b) | A statement is satisfiable if there is some interpretation for which it is true | | (c) | A statement is satisfiable if there is no interpretation for which it is true | | (d) | A statement is satisfiable if there is no interpretation for which it is false | | (e) | None of the above. | |
| 2. Two literals are complementary if   |  |  | | --- | --- | | (a) | They are equal | | (b) | They are identical and of equal sign | | (c) | They are identical but of opposite sign | | (d) | They are unequal but of equal sign | | (e) | They are unequal but of opposite sign. | | |
| 3. Consider a good system for the representation of knowledge in a particular domain. What property should it possess?   |  |  | | --- | --- | | (a) | Representational Adequacy | | (b) | Inferential Adequacy | | (c) | Inferential Efficiency | | (d) | Acquisitional Efficiency | | (e) | All the above. | | |
| 4. What is Transposition rule?   |  |  | | --- | --- | | (a) | From P → Q, infer ~Q → P | | (b) | From P → Q, infer Q → ~P | | (c) | From P → Q, infer Q → P | | (d) | From P → Q, infer ~Q → ~P | | (e) | None of the above. | | |
| 5. Third component of a planning system is to   |  |  | | --- | --- | | (a) | Detect when a solution has been found | | (b) | Detect when solution will be found | | (c) | Detect whether solution exists or not | | (d) | Detect whether multiple solutions exist | | (e) | Detect a solutionless system. | | |

**Completing this week should take you approximately 3 hours.**

**V. GENERALIZATION**

To train a model, you need to:

* Define the features: Independent variables: X
* Define the label: Dependent variable: y
* Construct a train/test set
* Define the initial weight
* Define the loss function: MSE
* Optimize the model: Gradient descent
* Define:
  + Learning rate
  + Number of epoch
  + Batch size
  + Number of class

**VI. REFERENCES:**

* Nagy (2018), Artificial Intelligence and Machine Learning Fundamentals.
* Kim (2017), MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence.
* Geron (2017), Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques.
* <https://www.udemy.com/course/artificial-intelligence-and-machine-learning-fundamentals/?fbclid=IwAR0CCSgtFGTFlqD08s1wZYRSAsLeewOTl8fwUtojODDgTzrqx_KG4WeRcOU>.
* <https://www.guru99.com/linear-classifier-tensorflow.html>.

**CONGRATULATIONS** on reaching the end of this module!

You may now proceed to the next module.

Don’t forget to submit all the exercises, activities and portfolio

on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**KEEP UP THE GOOD WORK**.

Well Done!!!

