Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Elektro- und
Informationstechnik

Classification

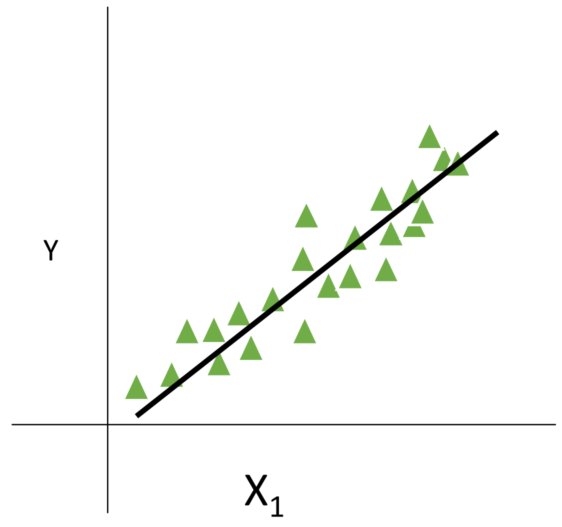
Source: DALL.E

Kawther Aboalam

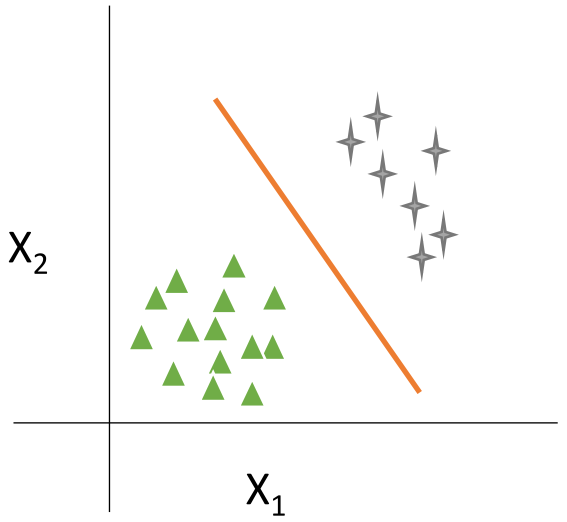
Classification

Introduction

+ Regression models can describe the processes, which have continuous numerical output variable (the target)
+ Classification refers to predictions where the label (the target variable) is qualitative
+ Classification problems occur often, perhaps even more frequently than regression ones.
+ Classification models can answer questions like:
  + Does a product with known characteristics meet the standard specifications?
  + Does a customer with known characteristics buy the offered product?
  + A patient with symptoms that could indicate one of three medical conditions arrives in the emergency room. Which condition does the patient have?
  + Is a chest X-ray normal or abnormal?
  + Which DNA mutations are disease-causing and which ones are not?



Regression

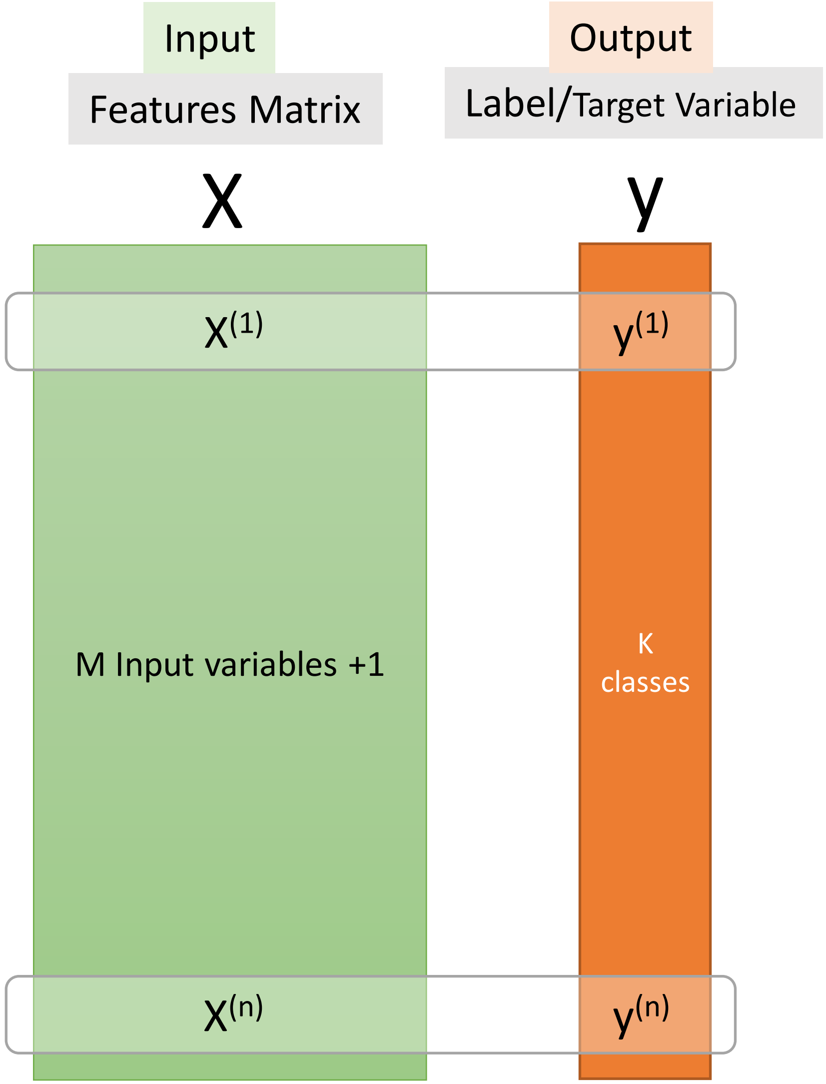


Classification

Classification

Introduction

+ Classification is a supervised learning task
+ Just as in the regression setting, in the classification setting we have a set of training observations $(X^{(1)}, y^{(1)}),\ldots, (X^{(n)}, y^{(n)})$ that can be used to build a classifier.
+ The classifier should perform well, not only on the training data, but also on test observations that were not used to train that classifier (unseen data)
+ The hyperparameters of the classifier should be tuned based on the evaluation over a validation set and not over the test set.
+ Model selection according to the results of a K-fold cross-validation criteria is also valid for the classification problems
+ Hence, a dataset containing observations with K different categories is said to be that dataset has K classes.



Classification

Introduction

+ Types of classification tasks:

Multi-Output Classification

Binary Classification

- Spam
- Not spam

Multiclass Classification

- Dog
- Cat
- Horse
- Fish
- Bird
- ...

Multi-label Classification

- Dog
- Cat
- Horse
- Fish
- Bird
- ...

Possible classes

Target variable (the label)
To be predicted

One of 2 categories

One of 3 or more categories

More than one target each of 2 categories or may be more

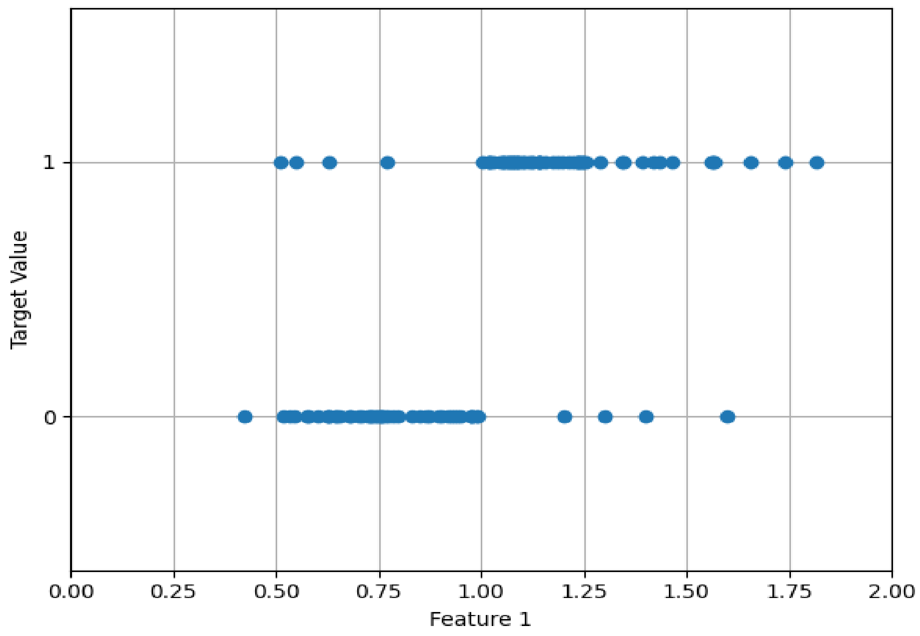
https://www.microsoft.com/en-us/research/uploads/prod/2017/12/40250.jpg

Classification

Binary Classification

+ It deals with two possible categories.
+ The value of y is even ∈ $\{0\,,1\}$ (binary or binominal classification)
+ Example:
  + Sensor Testing (two possible categories : OK or Defective)
  + Encode the target variable as follows:

$$(\text{Target value})\ y = \begin{cases} 1 & \text{if OK;} \\ 0 & \text{if Defective.} \end{cases}$$

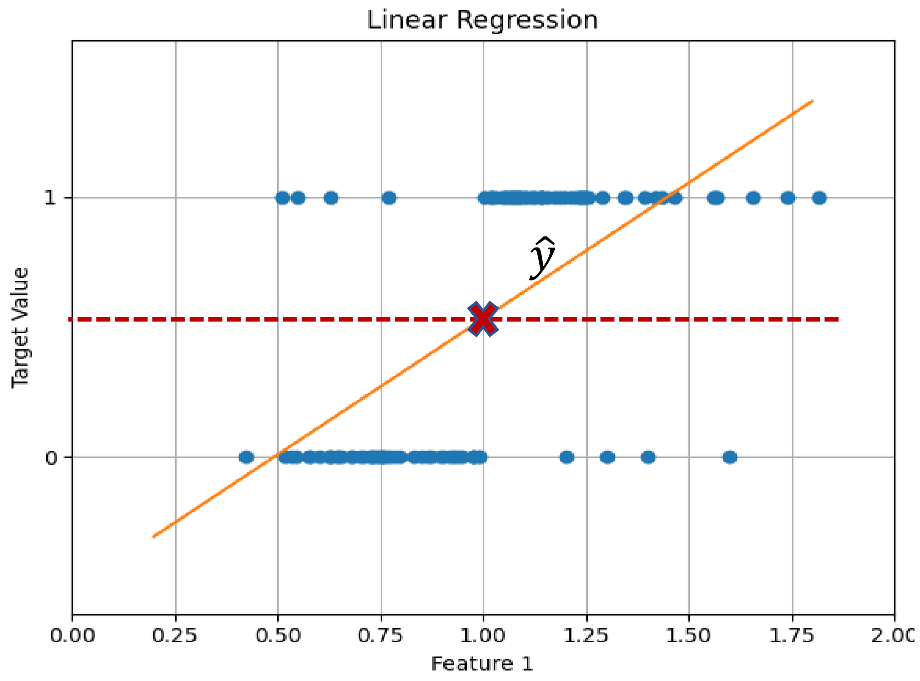


+ What about using a linear regression model to estimate the target values?

Classification

Binary Classification

- What about using linear regression to estimate the target values?
  - A linear regression model ($X\hat{\beta}$) could be fit to this binary target value. Using one feature, the $\hat{y}$ is a straight line with intercept value $\neq 0$
  - Round the values of $\hat{y}$ to 1, if it is ≥ 0.5 and to 0, if it is < 0.5
  - Predict **OK** if $\hat{y} \geq 0.5$ and **Defective** otherwise.
  - 0.5 value is called cut-off threshold value or just the threshold value

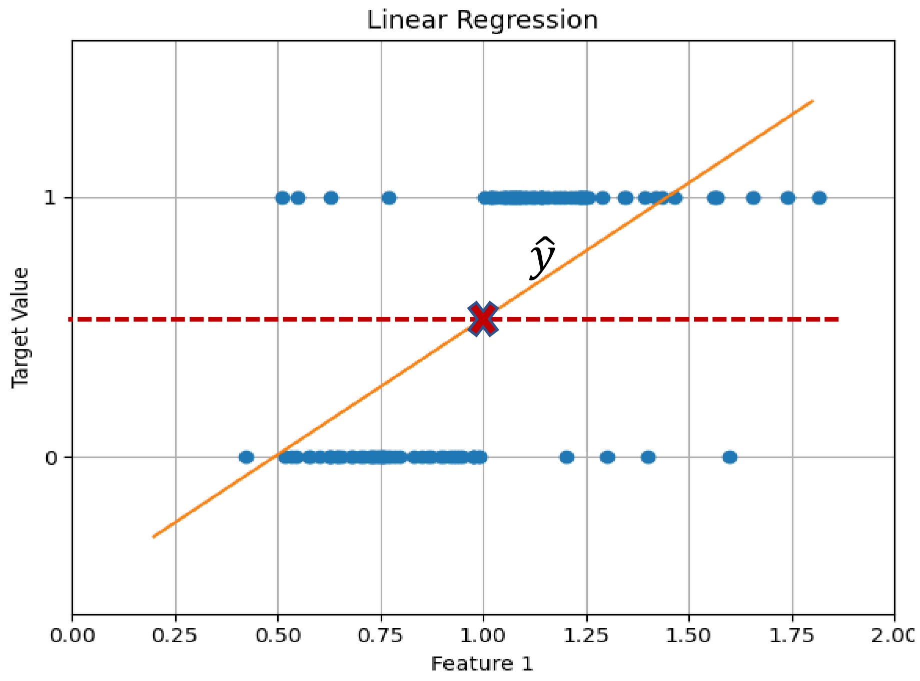


Classification

Binary Classification

+ What about using linear regression to estimate the target values?
  + A linear regression model ($X\hat{\beta}$) could be fit to this binary target value. Using one feature, the $\hat{y}$ is a straight line with intercept value $\neq 0$
  + Round the values of $\hat{y}$ to 1, if it is ≥ 0.5 and to 0, if it is < 0.5
  + Predict **OK** if $\hat{y} \geq 0.5$ and **Defective** otherwise.
  + 0.5 value is called cut-off threshold value or just the threshold value
  + In this case, you can think about $\hat{y}$ as an estimation of the probability of having y=1 with given X and according to $\hat{\beta}$

$$P(y = 1 | X\,; \hat{\beta})$$



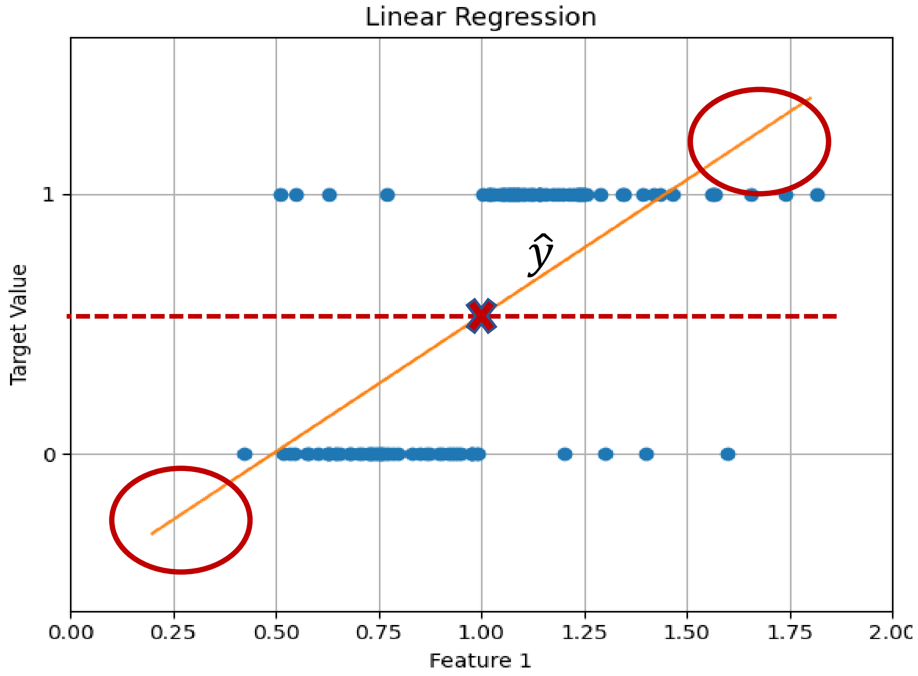
Classification

Binary Classification

- What about using linear regression to estimate the target values?
  - A linear regression model $(X\hat{\beta})$ could be fit to this binary target value. Using one feature, the $\hat{y}$ is a straight line with intercept value $\neq 0$
  - Round the values of $\hat{y}$ to 1, if it is ≥ 0.5 and to 0, if it is < 0.5
  - Predict **OK** if $\hat{y} \geq 0.5$ and **Defective** otherwise.
  - In this case, you can think about $\hat{y}$ as an estimation of the probability of having y=1 with given X and according to $\hat{\beta}$

$$P(y = 1|X\,;\hat{\beta})$$

  - But the estimates $(X\hat{\beta})$ might be outside the [0, 1] interval, making them hard to interpret as probabilities
  - A linear regression method will not provide meaningful estimates of $P(y = 1|X\,;\hat{\beta})$
  - Thus, it is preferred to use a classification method that is truly suited for binary values. That can force the estimations between 0 and 1.



Classification

Binary Classification – Logistic Regression

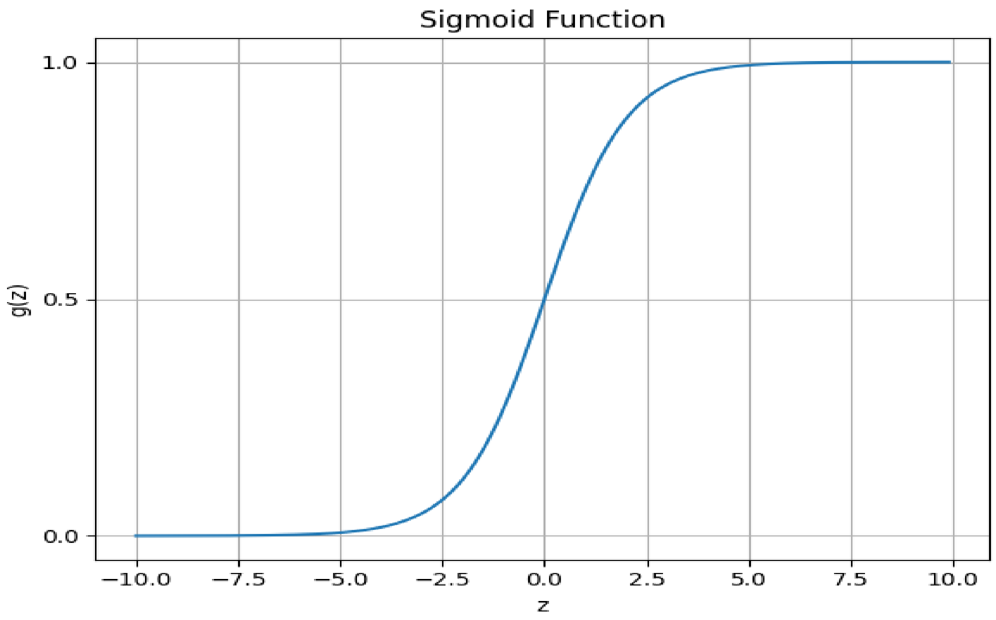
+ Logistic function (sigmoid function)

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

+ Logistic regression method is just to take a linear regression model and pass it through this sigmoid function:

$$z = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_M \cdot x_M$$

+ b_m parameters ($b_0, b_1, \ldots \ldots b_M$) define the shape of the logistic regression function with respect to X



Classification

Binary Classification – Logistic Regression

+ Logistic function (sigmoid function)

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

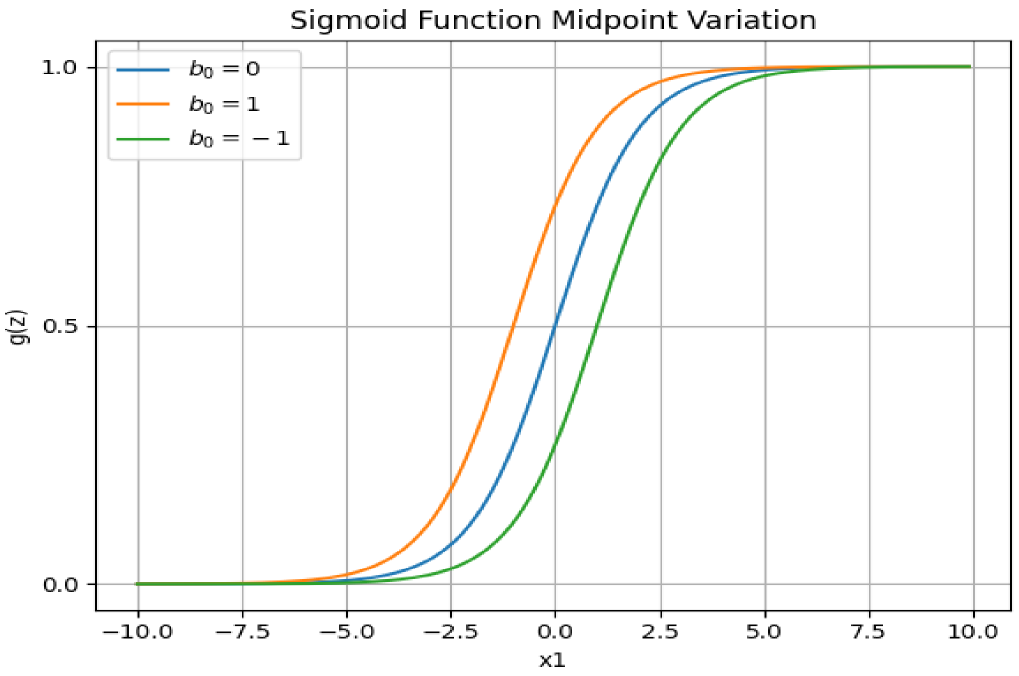
+ Logistic regression method is just to take a linear regression model and pass it through this sigmoid function:

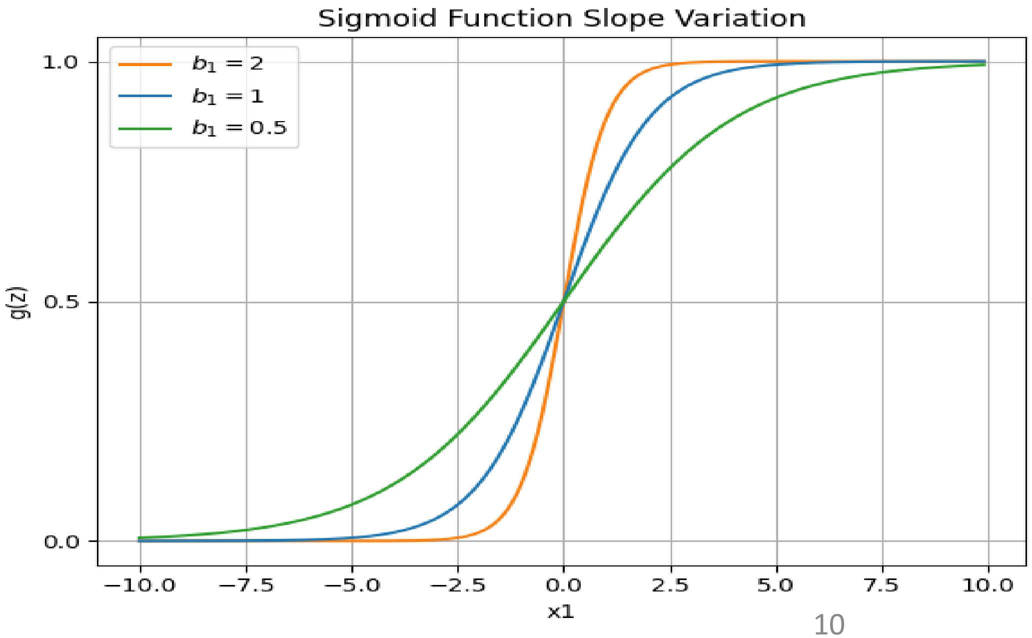
$$z = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_M \cdot x_M$$

+ b_m parameters $(b_0, b_1, \ldots \ldots b_M)$ define the shape of the logistic regression function with respect to X
+ Example: Assume a dataset with only one feature x_1

$$z = b_0 + b_1 \cdot x_1$$

- Plot $g(z)$ w.r.t. x_1 and vary b_0 as b_1 constant and vice versa





Classification

Binary Classification – Logistic Regression

+ Logistic function (sigmoid function)

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

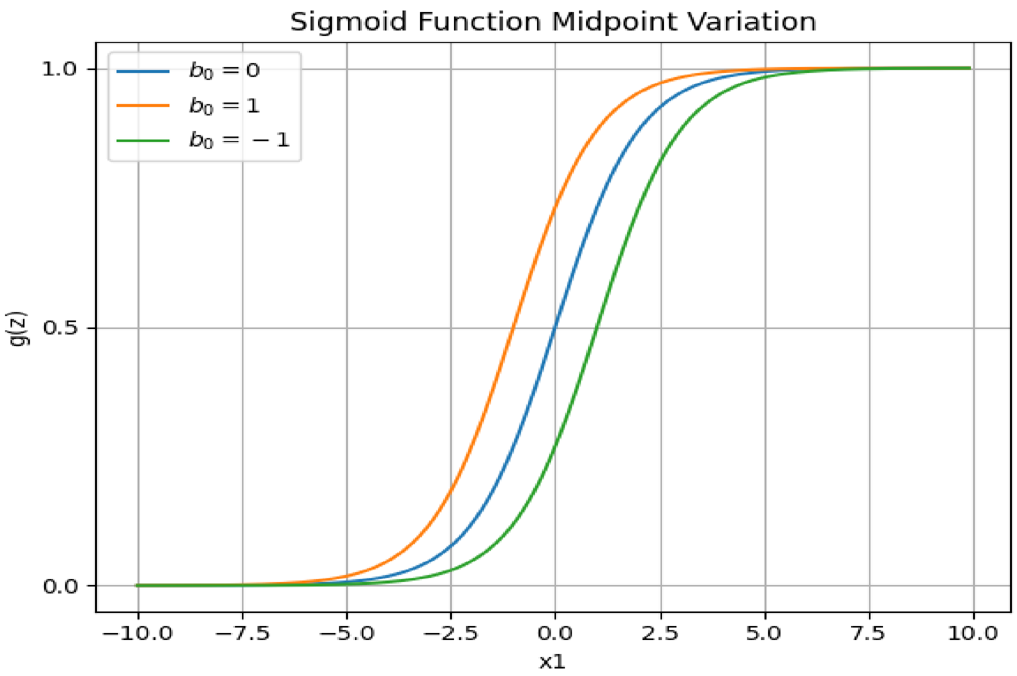
+ Logistic regression method is just to take a linear regression model and pass it through this sigmoid function:

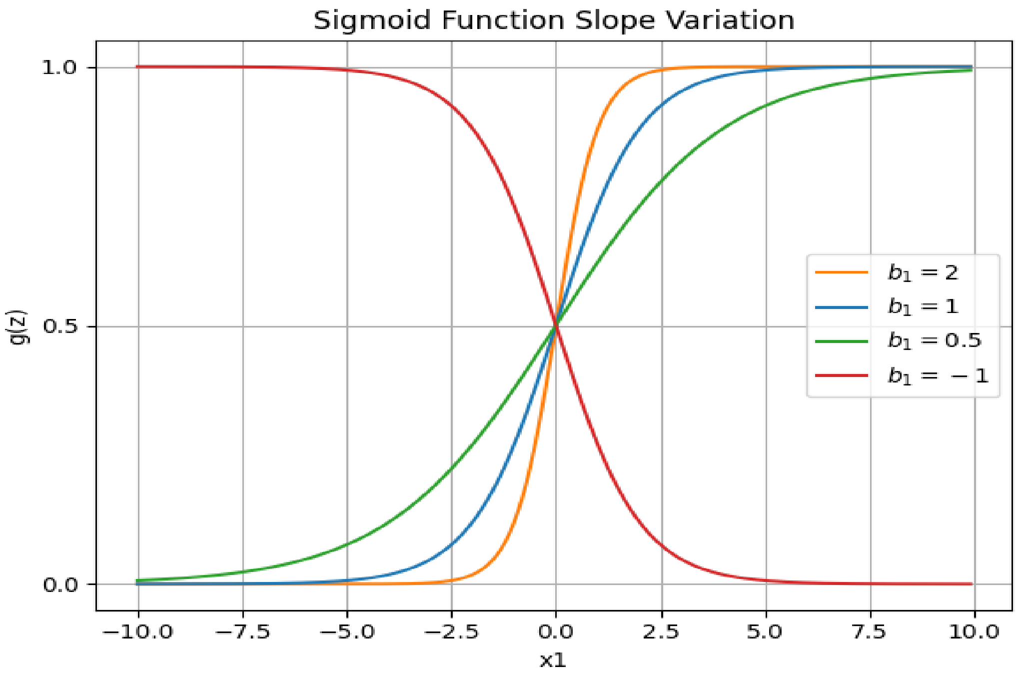
$$z = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_M \cdot x_M$$

+ b_m parameters $(b_0, b_1, \ldots \ldots b_M)$ define the shape of the logistic regression function with respect to X
+ Example: Assume a dataset with only one feature x_1

$$z = b_0 + b_1 \cdot x_1$$

  - Plot $g(z)$ w.r.t. x_1 and vary b_0 as b_1 constant and vice versa
  - If b_1 has a negative value, the slope will be inverted





Classification

Binary Classification – Logistic Regression

+ During training, the model's parameters are determined by minimizing a loss function using the training data.
+ Logistic regression g(z) describes the probability for y_i = 1 under the condition that the values x_{i1}…….. x_{im} are given

$$P(1|z_i) = g(z_i)$$

$z_i = b_{i0} + b_{i1} \cdot x_{i1} + b_{i2} \cdot x_{i2} + \ldots + b_{iM} \cdot x_{iM}$, It is the linear score (logit) for training example i.

+ The probability for the opposite event y_i = 0 under the same condition is:

$$P(0|z_i) = 1 - g(z_i)$$

+ In summary, regardless of y_i value 0 or 1, the two probability equations above can be combined into a single expression:

$$P(y_i|z_i) = \left(g(z_i)\right)^{y_i} \cdot \left(1 - g(z_i)\right)^{(1-y_i)}$$

+ Likelihood across all samples : Overall probability from the product of the probabilities of all the n trainings examples

$$L = \prod_{i=1}^{n} P(y_i|z_i) = \prod_{i=1}^{n} \left(g(z_i)\right)^{y_i} \cdot \left(1 - g(z_i)\right)^{(1-y_i)}$$

Classification

Binary Classification – Logistic Regression

+ Simplify the expression by applying the natural logarithm:

$$\ln(L) = \sum_{i=1}^{n} (y_i \cdot \ln(g(z_i)) + (1 - y_i) \cdot \ln(1 - g(z_i)))$$

+ Optimization by maximizing ln(L) or minimizing (- ln(L))

Loss= - $\sum_{i=1}^{n} (y_i \cdot \ln(g(z_i)) + (1 - y_i) \cdot \ln(1 - g(z_i)))$

+ This loss function is known as log loss or binary cross-entropy.
+ The model's parameter b_m $(b_0, b_1, \dots \dots b_M)$ can be estimated using gradient descent optimization method to minimize the loss function.
+ Gradients are computed using partial derivatives and chain rule as follows:

$$\frac{\partial Loss}{\partial b_m} = \frac{\partial Loss}{\partial g} \cdot \frac{\partial g}{\partial z_i} \cdot \frac{\partial z_i}{\partial b_m}$$

Classification

Binary Classification – Logistic Regression

+ Step-by-step calculation of partial derivatives:

$$\frac{\partial Loss}{\partial g} = \sum_{i=1}^{n} -y_i \cdot \frac{1}{g(z_i)} + (1-y_i) \cdot \frac{1}{1-g(z_i)}$$

$$\frac{\partial g}{\partial z_i} = \frac{\partial\left(\frac{1}{1+e^{-z_i}}\right)}{\partial z_i} = \frac{e^{-z_i}}{(1+e^{-z_i})^2} = \frac{1}{1+e^{-z_i}} \cdot \left(\frac{e^{-z_i}}{1+e^{-z_i}}\right) = \frac{1}{1+e^{-z_i}} \cdot \left(\frac{1+e^{-z_i}-1}{1+e^{-z_i}}\right) = \frac{1}{1+e^{-z_i}} \cdot \left(1 - \frac{1}{1+e^{-z_i}}\right) = g(z_i) \cdot (1-g(z_i))$$

$$\frac{\partial z_i}{\partial b_m} = x_{im}$$

+ The partial derivatives can be displayed as follows:

$$\frac{\partial Loss}{\partial b_m} = \sum_{i=1}^{n} \left(\left(-y_i \cdot \frac{1}{g(z_i)} + (1-y_i) \cdot \frac{1}{1-g(z_i)} \right) \cdot g(z_i) \cdot (1-g(z_i)) \right) \cdot x_{im} = \sum_{i=1}^{n} (-y_i \cdot (1-g(z_i)) + (1-y_i) \cdot g(z_i)) \cdot x_{im}$$

$$= \sum_{i=1}^{n} (-y_i + y_i \cdot g(z_i) + g(z_i) - y_i \cdot g(z_i)) \cdot x_{im} = \sum_{i=1}^{n} (g(z_i) - y_i) \cdot x_{im}$$

The Error Feature value

Classification

Binary Classification – Logistic Regression

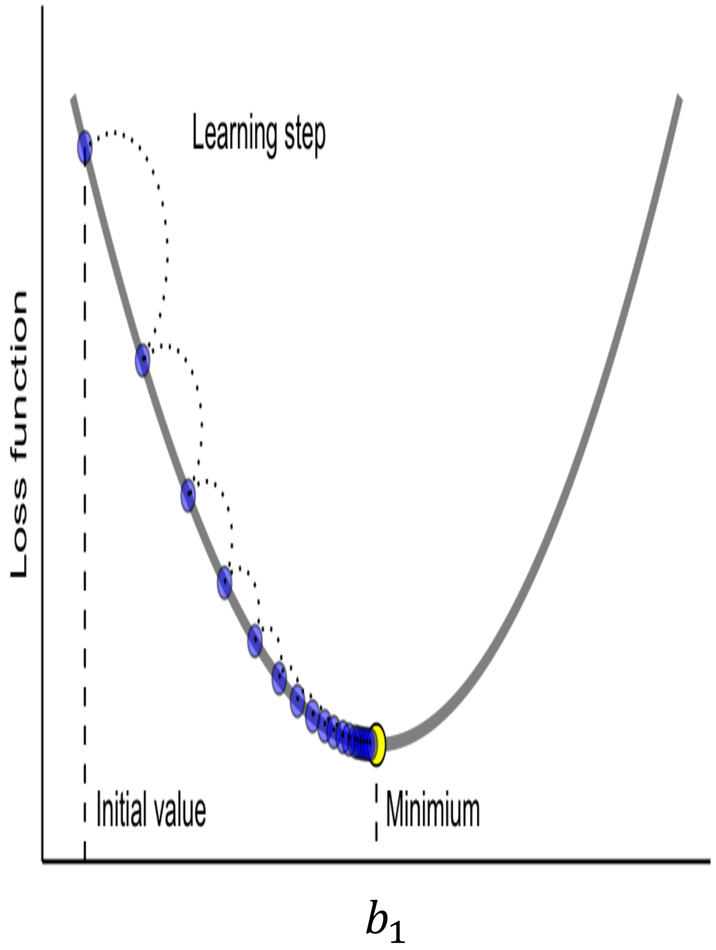
+ Steps of the Gradient Descent optimization algorithm:
  + Initialize random values for the parameters $b_m^{[0]}$ to get started with the iterative process
  + Keep on iterating for k = 0, 1, 2,……. using the update rule of the Gradient Descent

$$b_m^{[k+1]} \coloneqq b_m^{[k]} - \eta \cdot \frac{\partial Loss}{\partial b_m}$$

$$\coloneqq b_m^{[k]} - \eta . \sum_{i=1}^{n} (g(z_i) - y_i) \cdot x_{im}$$

  η is called the learning rate or the learning step size

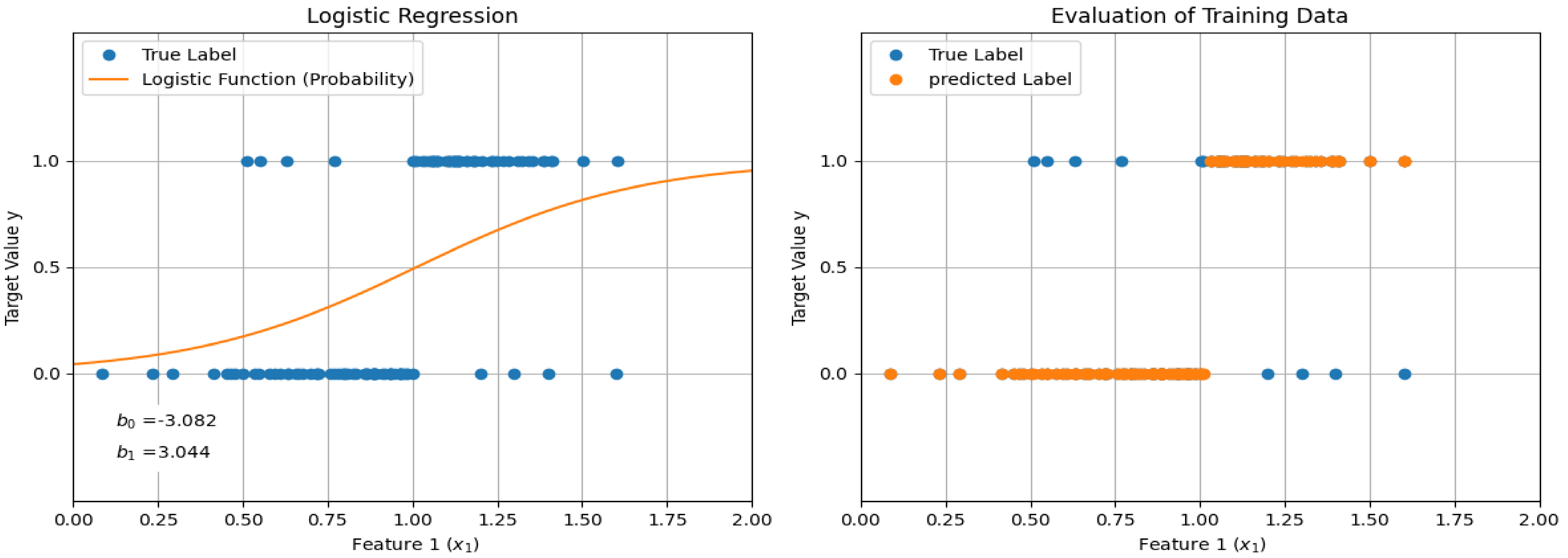
  + At each step, the direction of the negative gradient (steepest descent) is followed to reduce the loss value.
  + Termination criteria for a process can include:
    - Setting a specific number of iterations to be performed (number of epochs)
    - predefine improvement to be obtained in successive iterations
+ The learning rate and the number of epochs can be considered as hyperparameters of this method



Classification

Binary Classification – Logistic Regression

+ Logistic regression for Sensor Testing



Classification

Binary Classification – Confusion Matrix

+ A confusion matrix (also known as an error matrix) is a table that visualizes the performance of a classification algorithm.
+ It shows which predictions are correct or incorrect and identifies the types of classification errors.

	Predicted Label	
True Label	Positive Prediction	Negative Prediction
Positive Class (1)	True Positive (TP)	False Negative (FN)
Negative Class (0)	False Positive (FP)	True Negative (TN)

+ True Positive (TP):

  The model predicted positive, and it's true.
+ True Negative (TN):

  The model predicted negative and it's true.
+ False Positive (FP – Type I Error): A false alarm is raised

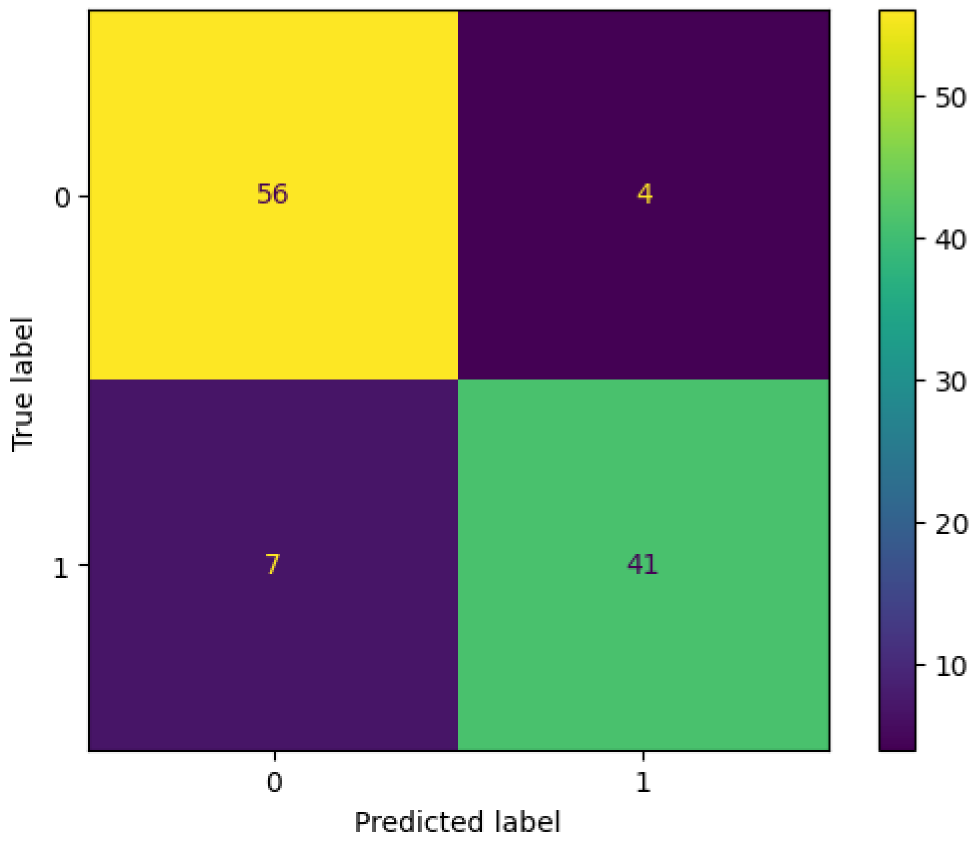
  The model predicted positive, but it's false.
+ False Negative (FN – Type II Error): A true alarm is missed

  The model predicted negative, but it's false.

Classification

Binary Classification – Confusion Matrix

- The confusion matrix from the sensor testing example shows that **41 + 56 = 97** sensors were correctly predicted, while **4 + 7 = 11** sensors were incorrectly predicted.
- True Positive (TP): 41
- True Negative (TN) : 56
- False Positive (FP – Type 1 Error): 4
- False Negative (FN – Type 2 Error): 7



Classification

Binary Classification – Evaluation Metrics From The Confusion Matrix

+ The following performance metrics are derived from the confusion matrix:

+ Classification Accuracy:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{TP+TN}{TP+TN+FP+FN} \in [0,1]$$

+ Classification Error: This is the complement of accuracy — it represents the proportion of incorrect predictions.

$$\text{Error} = \frac{\text{Incorrect Predictions}}{\text{Total Predictions}} = \frac{FP+FN}{TP+TN+FP+FN} \in [0,1]$$

Classification

Binary Classification – Evaluation Metrics From The Confusion Matrix

+ Classification Precision: measures the proportion of predicted positive instances that are truly positive.

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} = \frac{TP}{TP+FP} \in [0,1]$$

+ Classification Recall (also called sensitivity): measures how well the model identifies true positive cases.

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} = \frac{TP}{TP+FN} \in [0,1]$$

+ Precision: Useful when minimizing false positives is important.
+ Recall: Useful when minimizing false negatives is important.

Classification

Binary Classification – Evaluation Metrics – Computed From The Confusion Matrix

+ F1-Score: provides a way to express precision and recall with a single score.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \in [0,1]$$

+ The intuition for F1-measure is that both measures are balanced in importance and that only a good precision and good recall together result in a good F1-score

Classification

Binary Classification – Evaluation Metrics – Computed From The Confusion Matrix

+ sklearn.metrics.classification_report generates a detailed report of precision, recall, F1-score, and support for each class.
+ The classification report of the sensor testing example is shown.
+ Recall for class 1 = $\frac{41}{41+7} = 0.854$
+ Recall for class 0 = $\frac{56}{56+4} = 0.933$
+ Macro Average: It is referred to (unweighted) mean of the metric across all classes.
  + Macro average recall = (0.933+0.854)/2 = 0.89
+ Weighted Average: It takes into account the number of samples in each class to compute a weighted mean. In this example, class 0 has 60 samples and class 1 has 48 samples.
  + Weighted average recall = (60* 0.933 + 48* 0.854)/108 = 0.90

```
#accuracy_score(y_pred, y)
rep = classification_report( y , y_pred)
print(rep)
```

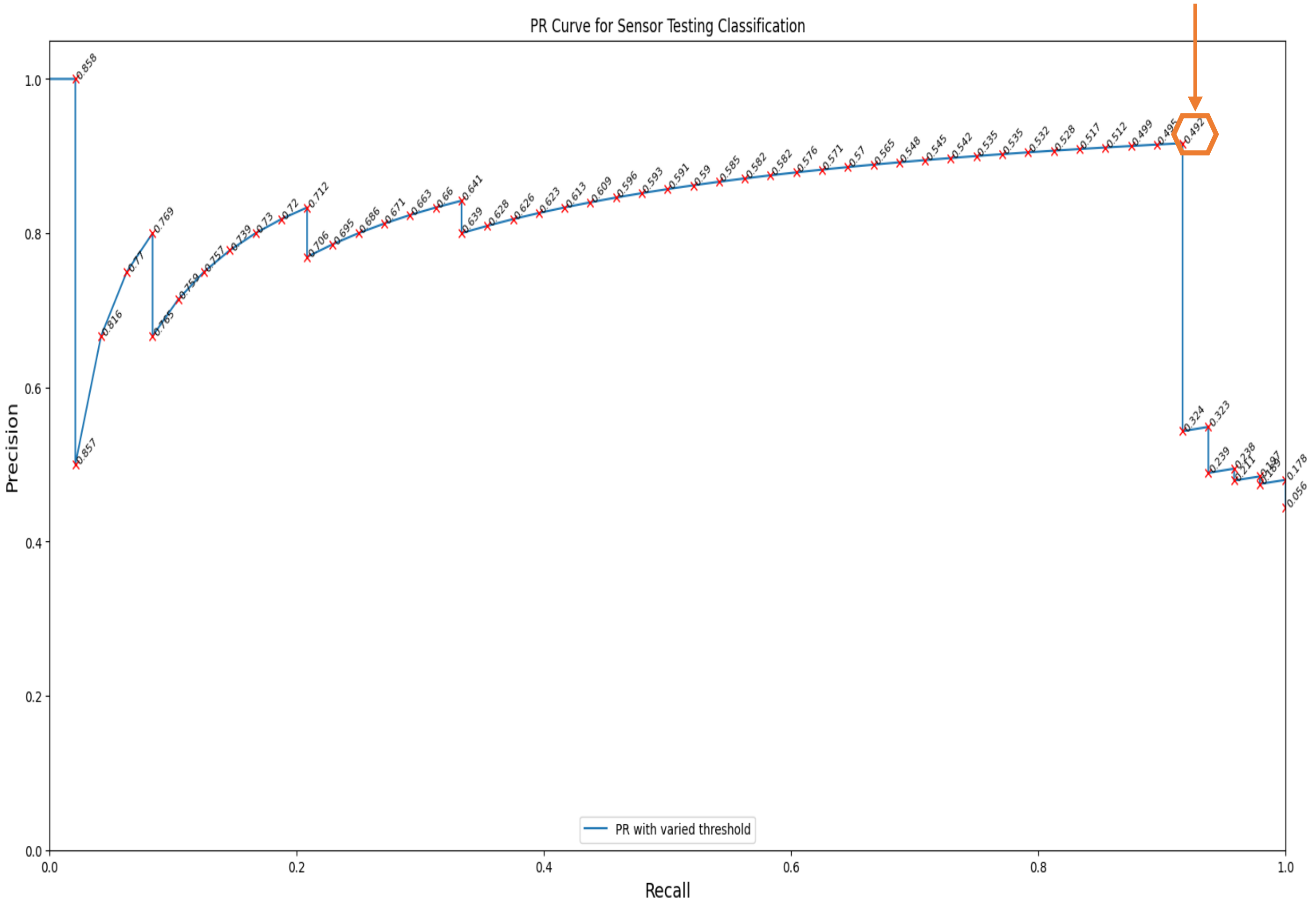
✓ 0.0s

```
              precision    recall  f1-score   support

           0       0.89      0.93      0.91        60
           1       0.91      0.85      0.88        48

    accuracy                           0.90       108
   macro avg       0.90      0.89      0.90       108
weighted avg       0.90      0.90      0.90       108
```

Classification

Binary Classification – Evaluation Metrics – Computed From The Confusion Matrix

- PR Curve: Precision-Recall Curve
  - The precision-recall curve plots parametrically the Precision(T) versus the Recall(T) at varying threshold values (T)
  - Precision = $\frac{TP}{TP+FP}$, Recall $= \frac{TP}{TP+FN}$
  - High precision relates to a low false positive rate, and high recall relates to a low false negative rate.
  - A large area under the curve represents both high recall and high precision
  - The F1-score can be used to pick the optimum point on the precision-recall curve
  - F1-Score = $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$



Classification

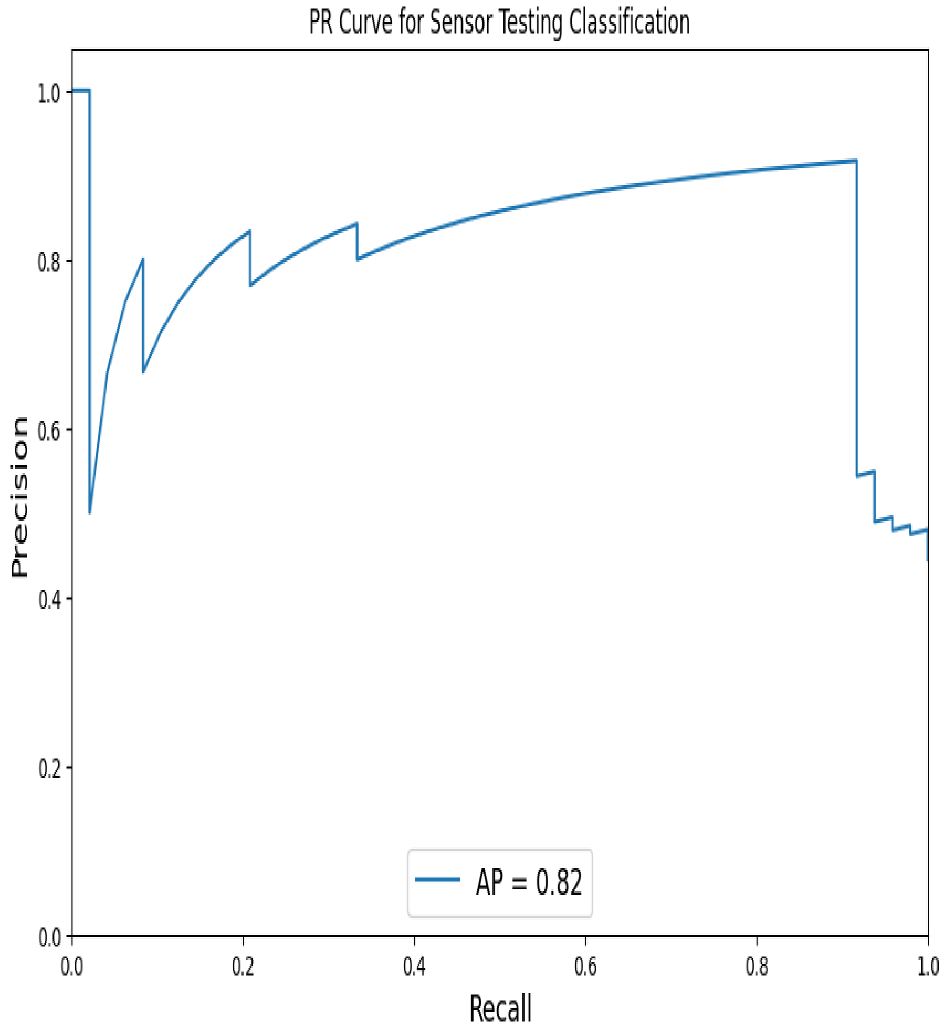
Binary Classification – Evaluation Metrics – Computed From The Confusion Matrix

+ PR Curve: Precision-Recall Curve
  + Area under the curve (PR-AUC) and average precision (AP) metrics are common ways to summarize a precision-recall curve into a single value
  + AP metric is widely used to summarize this curve information
  + AP is the weighted mean of Precision scores achieved at each PR curve threshold, with the increase in Recall from the previous threshold used as the weight

  + AP = $\sum_{k=0}^{k=n-1}[Recall(k) - Recall(k+1)] * Precision(k)$

    n = number of thresholds

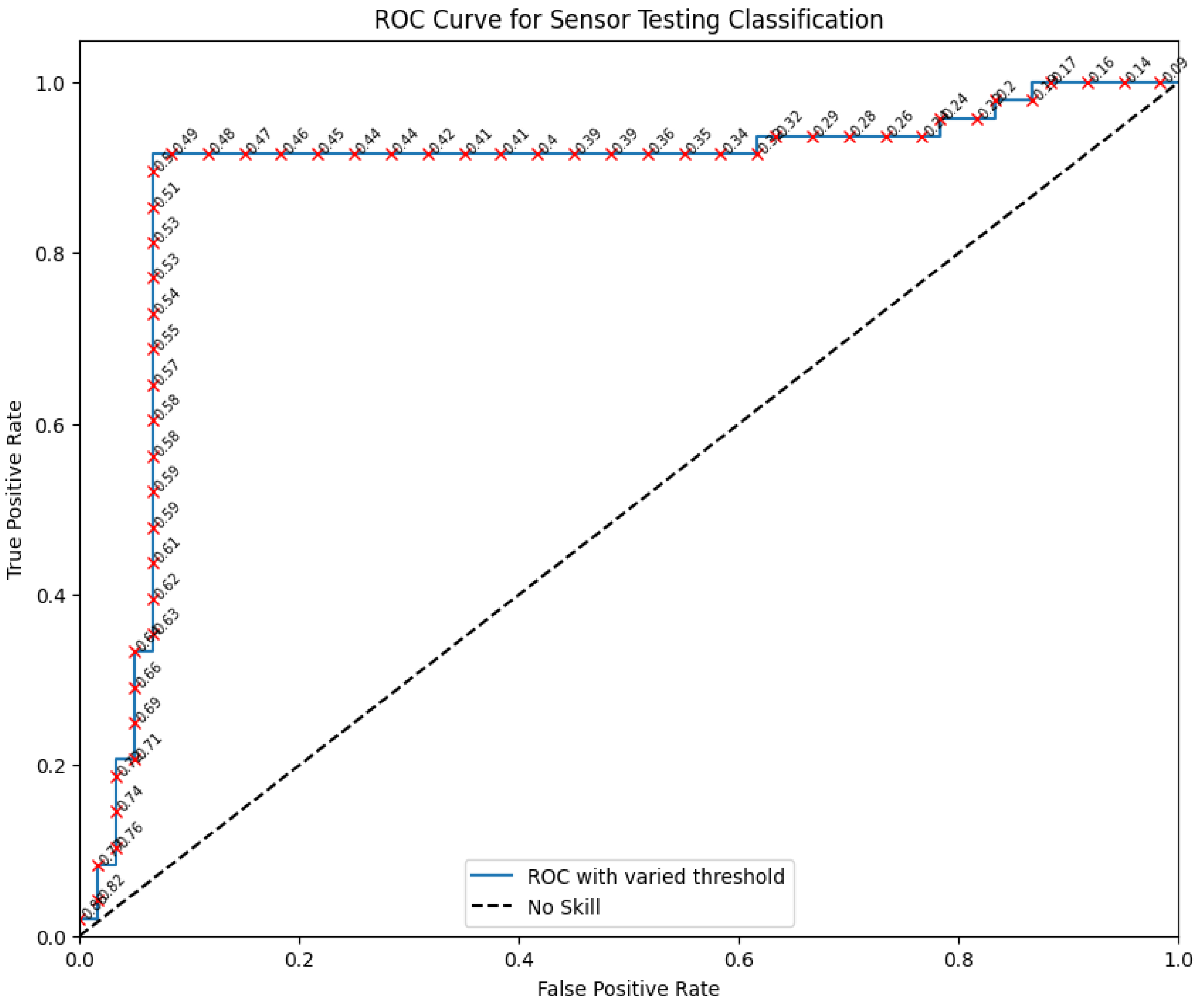
    Recall(n) = 0 and Precision(n) = 1, as T=1
  + The best possible score is 1, as $\mathrm{AP} \in [0,1]$
  + AP is used as an evaluation metric to compare different classifications algorithms.



Classification

Binary Classification – Evaluation Metrics – Computed From The Confusion Matrix

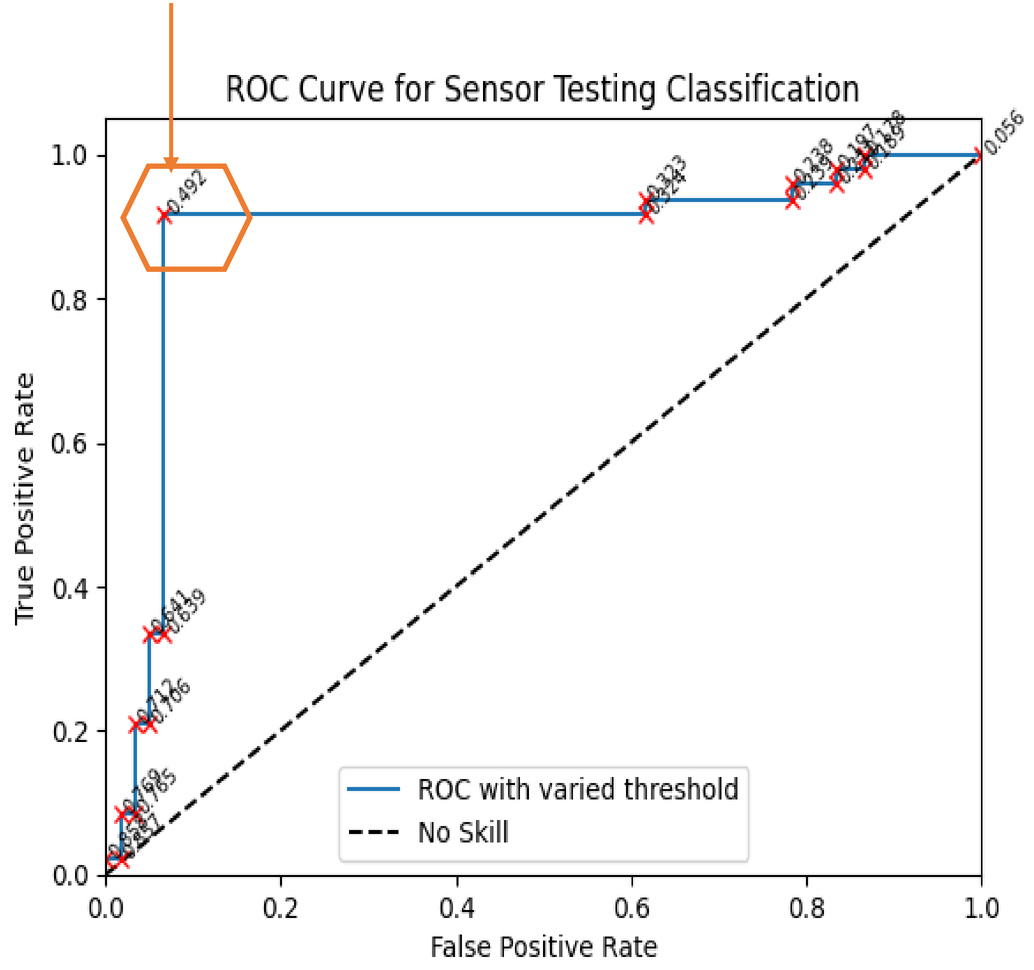
+ ROC Curve: Receiver Operating Characteristic Curve
  + The ROC curve plots parametrically the true positive rate TPR(T) versus the false positive rate FPR(T) at varying threshold values (T)
  + TPR = Sensitivity = Recall $= \frac{TP}{TP+FN}$
  + FPR = 1 - Specificity = $1 - \frac{\mathrm{TN}}{\mathrm{TN+FP}} = \frac{\mathrm{F}P}{\mathrm{F}P+\mathrm{T}N}$
  + FPR represents the false positive alarms rate
  + A large area under the curve represents both high TPR and low false positive alarms (FPR)
  + One measure that can be used for calculating the optimum point on a ROC curve is the point at which, the $TPR–FPR$ is at its maximum. It is the point at the upper left corner.



Classification

Binary Classification – Evaluation Metrics – Computed From The Confusion Matrix

- The optimum point is at T = 0.4921
- Comparing the results at the default threshold as T = 0.5 and the results of the threshold at the optimum point

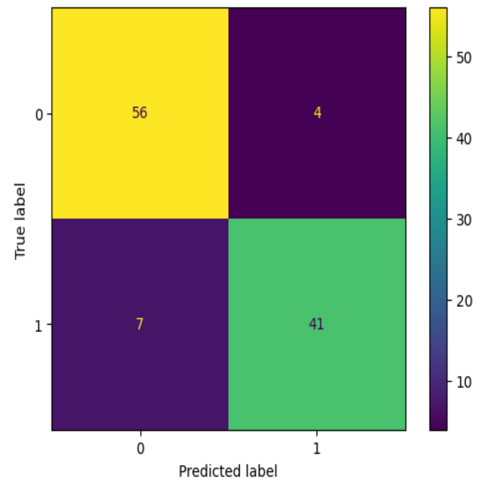


Classification

Binary Classification – Evaluation Metrics – Computed From The Confusion Matrix

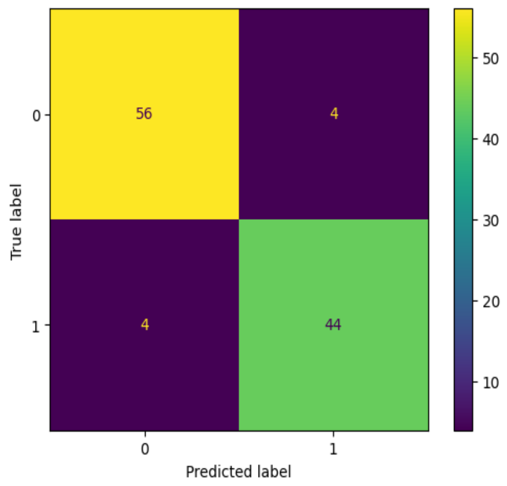
- The optimum point is at T = 0.4921
- Comparing the results at the default threshold, where T = 0.5 and the results of the threshold at the optimum point:
  - The TPR is increased from $\frac{41}{41+7}$ into $\frac{44}{44+4}$
  - The FPR is still as it is $\frac{4}{4+56}$
  - F1-Score is increased from 0.9 to 0.93
  - The accuracy and the other metrics are increased

T = 0.5



	precision	recall	f1-score	support
0	0.89	0.93	0.91	60
1	0.91	0.85	0.88	48
accuracy			0.90	108
macro avg	0.90	0.89	0.90	108
weighted avg	0.90	0.90	0.90	108

T = 0.492

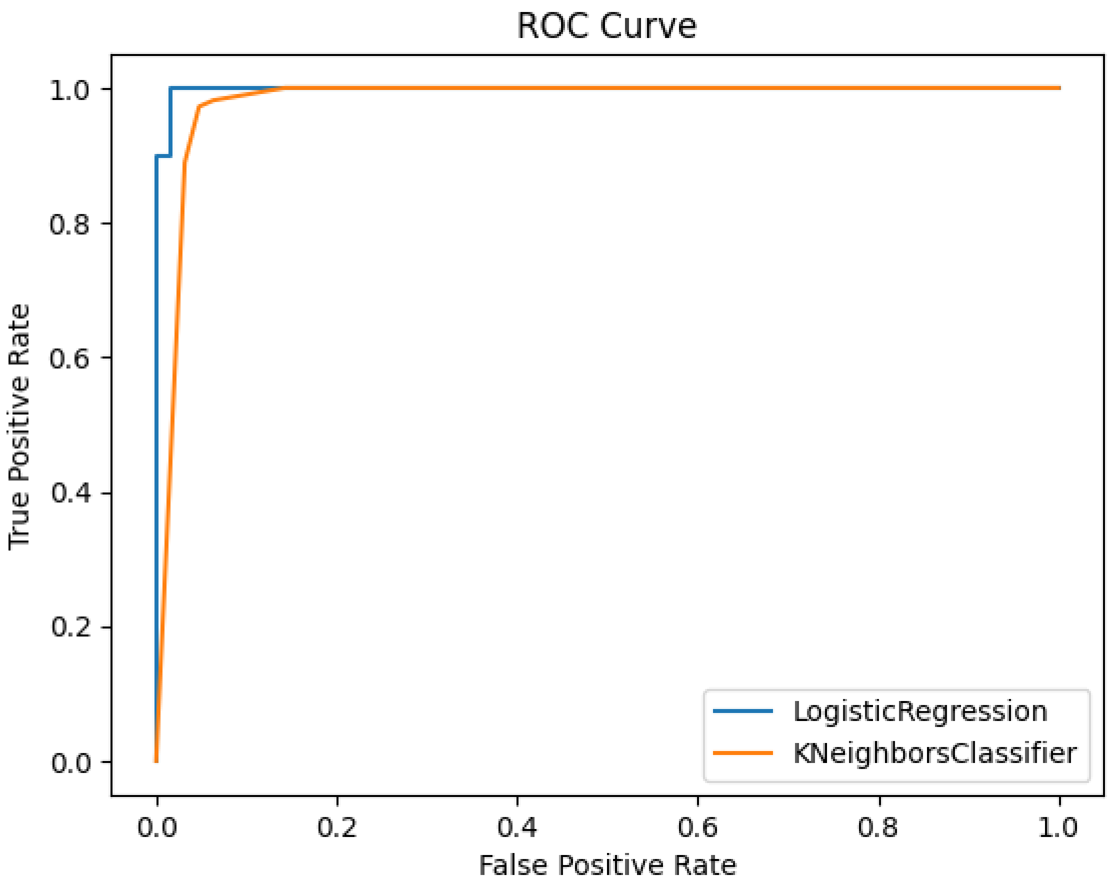


	precision	recall	f1-score	support
0	0.93	0.93	0.93	60
1	0.92	0.92	0.92	48
accuracy			0.93	108
macro avg	0.93	0.93	0.93	108
weighted avg	0.93	0.93	0.93	108

Classification

Binary Classification – Evaluation Metrics – Computed From The Confusion Matrix

+ The area under the ROC curve (ROC-AUC) can summarize the curve information in one number.
+ It is used as an evaluation metric to compare different classifications algorithms.
+ According to the ROC curves at the right side of this slide, the performance of the logistic regression is better than the performance of the K-Neighbors classifier. That is because its ROC-AUC is larger than the ROC-AUC of the K-Neighbors classifier.
+ The precision-recall curve is more informative than the ROC curve when evaluating binary classifiers on imbalanced datasets



https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118432

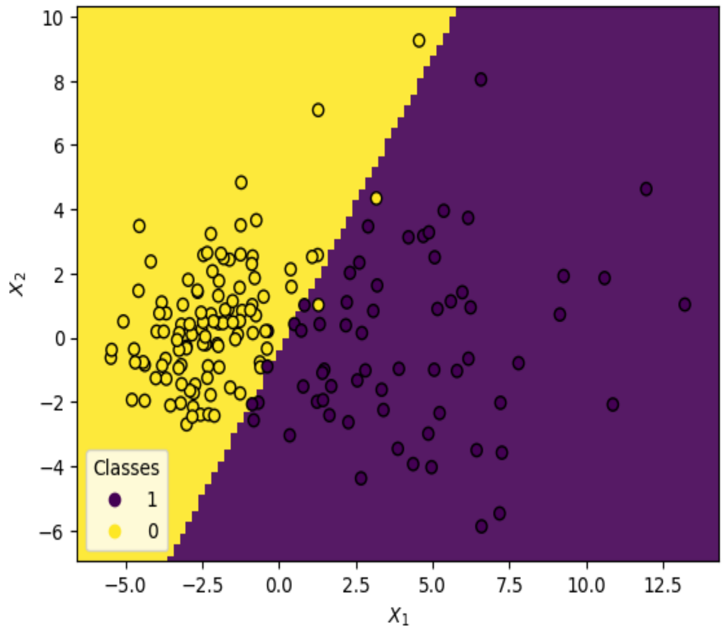
Classification

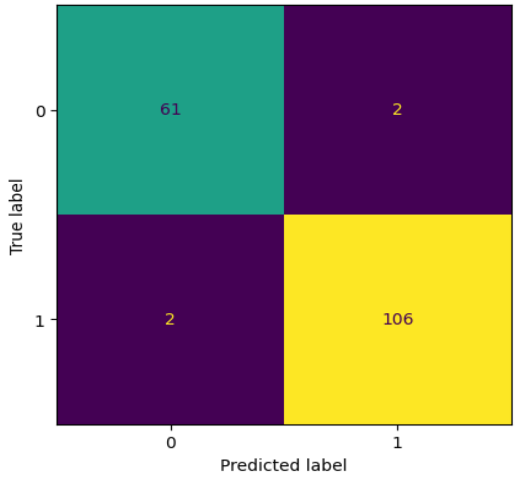
Binary Classification – What is the relationship between the threshold (T) and the decision boundary in logistic regression?

+ Logistic function for two features X_1 and X_2 and at T= 0.5

+ $g(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(b_0+b_1 \cdot x_1+b_2 \cdot x_2)}} = 0.5 = \frac{1}{1+1} = \frac{1}{1+e^{0}}$

$$(b_0 + b_1 \cdot x_1 + b_2 \cdot x_2) = 0$$





+ After training the parameters b_0, b_1 and b_2 are determined and the above equation will represent a line in the space of the two features X_1 and X_2, that is the decision boundary for T=0.5

+ The decision boundary is varied by varying the value of the decision boundary or the values of the determined parameters during the training

Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Elektro- und
Informationstechnik

www.h-ka.de