Methodology

Linear Regression

This chapter is about linear regression, a straightforward approach for supervised learning. Linear regression is useful for predicting quantitative response from a set of predictor variables. Moreover, it determines which variables in particular are significant predictors of the outcome variables and in what way do they *indicated by the beta estimates* impact the outcome variable. These regression estimates are used to explain the relationship between predictor variable X and response variable Y.

Simple Linear Regression

Simple linear regression is an approach for predicting a quantitative response Y on the basis of a single predictor variable X. Assumed there is approximately a linear relationship between X and Y. The linear relationship will be mathematically written as

$$Y \approx \beta_0 + \beta_1 X$$

β_0 and β_1 are unknown constants that designate the *intercept* and *slope* in the linear model and together they are known as the model *parameters* . Then, after we have estimated $\hat{\beta}_0$ and $\hat{\beta}_1$ for the model parameters by using the training data, we can predict the response variable Y by computing

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x,$$

with $\hat{y}$ designates the prediction of Y on basis $X = x$.

Estimating the Parameters

As said before, β_0 and β_1 are unknown. Before we make predictions, we have to use the data to estimate the parameters. Let

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

designate n observation pairs, each of which consists of a measurement of X and Y. The main objective here is to get paramater estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ that fit the linear model, so that

$$\hat{y}_i \approx \hat{\beta}_0 + \hat{\beta}_1 x_i$$
$$i = 1, \ldots, n.$$

Here we want to acquire parameters $\hat{\beta}_0$ and $\hat{\beta}_1$ that will make the regression line as close as possible to the n data points. We will take the *least squares* approach to acquire these parameters. Figure 1 represents the simple linear regression model.

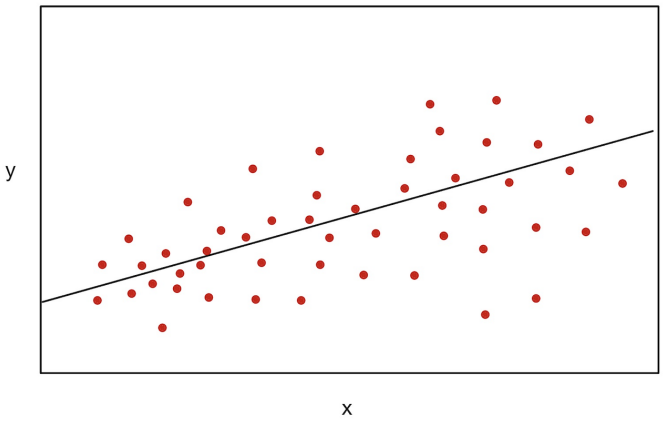

Figure 1: Simple linear regression model, the red dots represent data points

Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for Y based on x_i (ith value of X). Then $e_i = y_i - \hat{y}_i$ designates the ith *residual*. *Residual* is the difference between ith actual response value and the ith predicted response value from our linear model. Next we define the *residual sum of squares* (RSS) as

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2,$$

and furthermore

$$RSS = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \cdots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2.$$

To obtain optimum $\hat{\beta}_0$ and $\hat{\beta}_1$, we have to minimize the RSS. The *least squares parameter estimates* will be written as

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^n (x_i - \overline{x})^2},$$
$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x},$$

with $\overline{y} \equiv \frac{1}{n}\sum_{i=1}^n y_i$ and $\overline{x} \equiv \frac{1}{n}\sum_{i=1}^n x_i$ as the sample means.

R^2 Statistics

The R^2 statistic provides the measure of fit. It is the proportion of variability in Y that can be explained by using X, has the value between 0 and 1, and also independent of the scale of Y. To calculate R^2, we use the formula

$$R^2 = 1 - \frac{RSS}{TSS},$$

where TSS $= \sum(y_i - \overline{y})^2$ is the *total sum of squares*. TSS measures the total variance in Y. R^2 value near 0 indicates that the model did not explain much of the variability in the response, meaning that the regression model could be wrong.

Multiple Linear Regression

As explained before, simple linear regression is a practical approach to predict a response on the basis of a single predictor variable. Unfortunately, in reality we often come up with more than one predictor variable. How do we integrate these extra predictors to make our analysis? One solution would be by making separate simple linear regressions, each of them using different predictor. This solution, however, is not that effective, because we will have different regression equation for each predictor so that a single prediction would be hard to conclude. Another thing is that by using simple linear regression, each of the equation will exclude the other predictors in forming estimates for the parameters. Instead, we will extend the simple linear regression by giving each predictor its own separate slope in a single model. Assumed that we have i predictors:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_i X_i + \epsilon,$$

where X_i designates the ith predictor and β_i is the slope for each ith predictor and is interpreted as the average effect on Y of a one unit increase in X_i, while other predictors fixed. ϵ designates the residual of the regression.

Estimating the Parameters

As we have shown in the simple linear regression, the regression parameters $\beta_0, \beta_1, \ldots, \beta_i$ are also unknown and therefore must be estimated. Given estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_i$, we can make predictions using the formula

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_i x_i.$$

The parameters are going to be estimated by using the least square approach, like it was the case in simple linear regression. We choose $\beta_0, \beta_1, \ldots, \beta_i$ to minimize the sum of squared residuals

$$RSS = \sum_{j=1}^{n} (y_j - \hat{y_j})^2$$

$$= \sum_{j=1}^{n} (y_j - \hat{\beta}_0 - \hat{\beta}_1 x_{j1} - \hat{\beta}_2 x_{j2} - \cdots - \hat{\beta}_i x_{ji})^2$$

Selecting Significant Variables
On many cases, it is possible that only some of the predictors are related with the response. To determine which predictors are related to the response, so that we can make a single model only for those predictors, is called *variable selection*. In our case, we will be using *Akaike Information Criterion*(AIC) to determine which variables are significant.

Logistic Regression

In linear regression model, the response variable Y is assumed quantitative. But in other situations, the response variable is instead qualitative or also referred as categorical. The task is now we predict the probability of each categories of a qualitative variable , as the base for making our final prediction. For example, if someone took a loan, then it's either they *have* or *have not* paid it back. Then we could code these qualitative response as:

$$Y = \begin{cases} 0 & \text{if not default} \\ 1 & \text{if default} \end{cases}$$

After that we could make a linear regression to this binary response, and predict **paid** if $Y > 0.5$ and **unpaid** otherwise. But rather than making a linear model to this response, logistic regression models the probability that Y belongs to a certain category. For example, the probability of default can be written as

$$p(X) = Pr(Y = 1|X)$$

Logistic Model

As mentioned before, the response variable in logistic regression is qualitative. Therefor we cannot model the probability by using linear regression model. One of the reasons is that the predicted probability value would not fall between 0 and 1 if we use linear model. We must then model $p(X)$ using a function that gives results between 0 and 1 for all values of X. In logistic regression, we use the *logistic function*,

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

After a bit of manipulation, we got

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X}.$$

The left-hand side of the equation is called the *odds* and have value between 0 and ∞. Then by giving both sides the logarithm, we will have

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

The left-hand side is now called the *log-odds* or *logit*.

Estimating the Parameters

The parameters β_0 and β_1 are also unknown like the case in linear regression and they need to be estimated based on training data. The preferred approach is the *maximum likelihood*. The idea is that we look for $\hat{\beta}_0$ and $\hat{\beta}_1$ by plugging these estimates into the model for $p(X)$ yields a number close to one for all $Y = 1|X$, and a number close to zero for all $Y = 0|X$. This can be formalized using *likelihood function*:

$$L(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to maximize the likelihood function. For our topic we will not go deep into the mathematical details of maximum likelihood as it can be easily fit using **R** function that we will discuss more later on. Once the parameters have been estimated, we can put them in our model equation:

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 \hat{X}}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 \hat{X}}}.$$

Multiple Logistic Regression

Assumed now that we have multiple predictors. Just like on linear regression, we can extend the formula that we have as follows:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X + \cdots + \beta_p X_p,$$

where $X = (X_1, \ldots, X_p)$ are p predictors. The equation can be rewritten as

$$p(X) = \frac{e^{\beta_0 + \beta_1 X + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X + \cdots + \beta_p X_p}}.$$

We also use the maximum likelihood to estimate $\beta_0, \beta_1, \ldots, \beta_p$.

Support Vector Machines

One of the most commonly used of machine learning algorithms is support vector machine. It was developed in the 1990s by Vladimir Vapnik and colleagues and that has become more popular since then. The support vector machine is a supervised learning approach for classification, which is generally suited for binary classification. There are further extensions of support vector machines to accommodate cases of more than two classes, but it is outside of the scope of our seminar project.

Maximal Margin Classifier

It all started with a simple classifier called the maximal margin classifier. We will firstly define the concept of an hyperplane.

Hyperplane
In a $n-$dimensional space, a flat affine subspace of dimension $n-1$ is called a hyperplane, e.g. a line is a flat one-dimensional subspace of two-dimensional space for that a line is a hyperplane of two-dimensional space. Another example is, in three dimensional space, a plane is a flat two-dimensional subspace that is also called hyperplane.

The definition of a $n-$dimensional hyperplane is defined by the equation:

$$h(X) := \beta_0 + \beta^T X = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n = 0,$$
$$\beta_0 \in \mathbb{R},\ \beta \in \mathbb{R}^n.$$

If a point $X = (X_1, X_2, \ldots, X_p)^T$ satisfies the hyperplane equation, then X lies on the hyperplane.

Suppose that X has the following property:

$$h(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n < 0$$

or

$$h(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n > 0.$$

Since we have inequality instead of equality, it follows that X lies on one side of the hyperplane.

Classification Using a Hyperplane
Given a data set of m training observations in $n-$dimensional space:

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \end{pmatrix}, \ldots, x_m = \begin{pmatrix} x_{m1} \\ \vdots \\ x_{mn} \end{pmatrix},$$

where the dimension of n corresponds to size of features and we define two classes where these observations fall into: $\{-1, 1\} \ni y_1, \ldots, y_m$. One can say that each observation x_i has a property of y_i that is either -1 or 1.

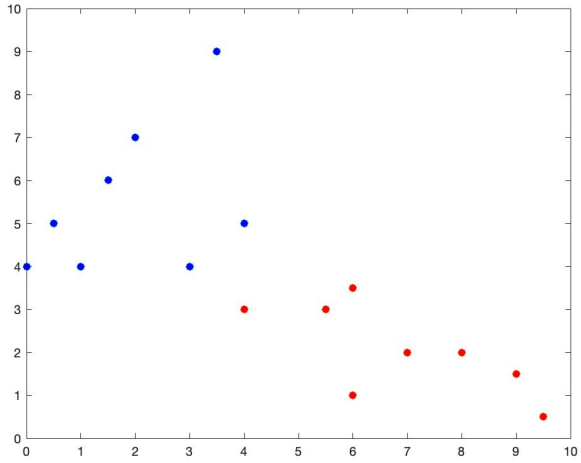

Figure 2: Two classes of observations, shown in blue and in red

We will now construct some hyperplanes that separate the training observations perfectly. Three separating hyperplanes are shown by figure 3.

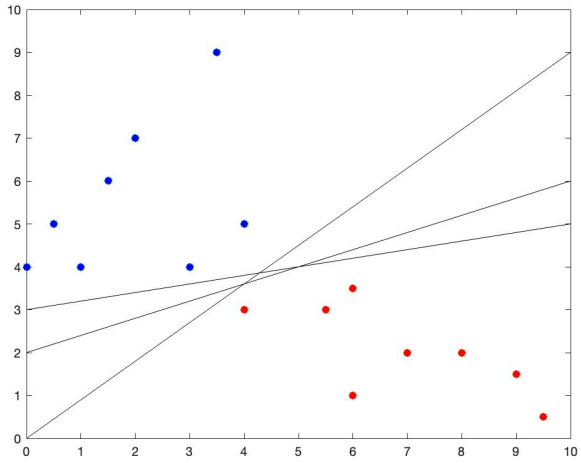

Figure 3: Separating hyperplanes

Suppose the observations from the blue class have the property of $y_i = 1$ and those from the red class $y_i = -1$. Then all separating hyperplanes have the property:

$$h(x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in} < 0 \text{ if } y_i = -1$$

and

$$h(x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in} > 0 \text{ if } y_i = 1.$$

It is equivalent to:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in}) > 0$$

for all $i = 1, \ldots, n$.

Maximal Margin Classifier

If the data sets can be perfectly separated, then there will be an infinite number of such hyperplanes. Imagine we tilt the separating hyperplane by a little bit, there exists many hyperplanes that do not touch the observations. Therefore we have to decide which hyperplane to use. One of the reasonable way is using the maximal margin classifier as method to decide which is the optimal separating hyperplane. Optimal separating hyperplane is a hyperplane that is farthest from the training observations. The smallest distance between the observation to the separating hyperplane is known as the margin and is given by:

$$M = \frac{1}{\|\beta\|_2} + \frac{1}{\|\beta\|_2} = \frac{2}{\|\beta\|_2}.$$

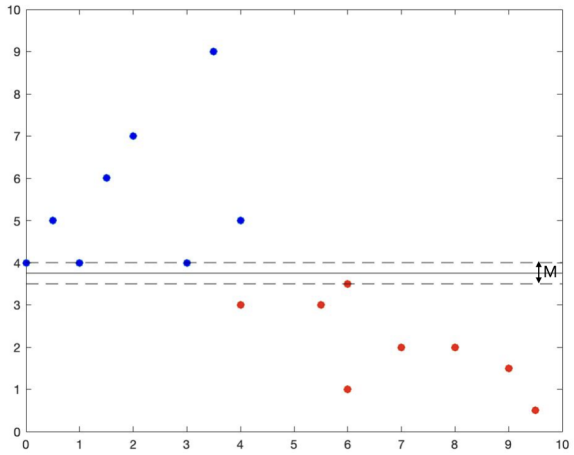

Figure 4: Separating hyperplanes and its margin

Distance between any observation $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T,\ i = 1, \ldots, m$ and the hyperplane $\beta_0 + \beta^T X = 0$ is defined by:

$$d(x_i) := \frac{|\beta_0 + \beta^T X|}{\|\beta\|_2}.$$

Figure 4 shows a separating hyperplan and its margin. However, this hyperplane is not optimal, since we can find another separating hyperplane with greater margin. It could be said that optimal separating hyperplane is the separating hyperplane, which has largest margin.

This can be considered as an optimization problem. The optimal separating hyperplane is the solution of the following optimization problem:

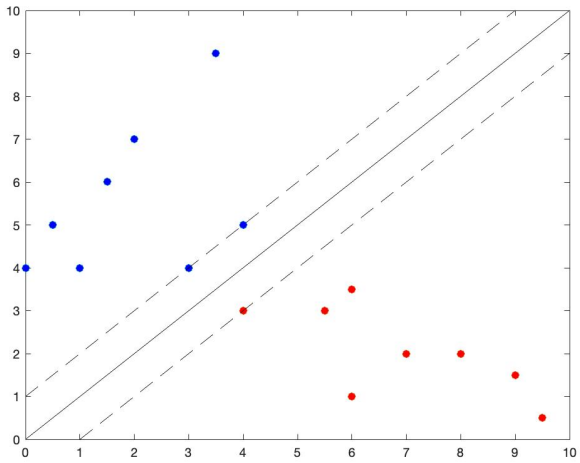

Figure 5: Optimal separating hyperplanes

$$\min_{\beta_0,\beta} \frac{\|\beta\|^2}{2}$$
$$\text{s.t. } y_i(\beta_0 + \beta^T X_i) \geq 1, \ \forall i \in \{1, \ldots, m\}.$$

The maximal margin classifier can be applied when training observations are linearly separable, or it can be said, if a separating hyperplane exists. Unfortunately most data sets are linearly non-separable, that means the maximal margin classifier is not applicable, since the classes should be separable by a linear boundary.

Soft Margin Classifier

Soft Margin Classifier is an extension of maximal margin classifier. The difference between soft margin classifier and maximal margin classifier is, that in soft margin classifier some observations are allowed to be on the incorrect side of the margin or even on the wrong side of hyperplane. We can say, that the margin can be violated by a small subset of observations.

In the optimization problem we introduce slack variables ϵ_i, which measure and penalise the degree of misclasification of x_i. Then the hyperplane is the solution to the optimization problem:

$$\min_{\beta_0,\beta,\epsilon_i} \frac{\|\beta\|^2}{2} + C\sum_{i=1}^{m} \epsilon_i$$
$$\text{s.t. } y_i(\beta_0 + \beta^T X_i) \geq 1 - \epsilon_i \ , \ \epsilon_i \geq 0 \ , \quad \forall i \in \{1, \ldots, m\},$$
$$\sum_{i=1}^{m} \epsilon_i \leq C,$$

where C is a tuning parameter, that is nonnegative. The slack variable ϵ_i describes the position of a given point x_i. If $\epsilon_i = 0$ then there is no misclassification and it follows that x_i is on the correct side. If $\epsilon_i \in (0, 1)$ then x_i lies between the support hyperplane and the separating hyperplane. If $\epsilon_i > 1$ then x_i is on the wrong side of the separating hyperplane. The tuning parameter C has a role as upper bound of the sum of ϵ_i's, so it determines how much we will tolerate.

As the parameter C increases, the margin will widen and it implies that more violation are allowed to the margin. When C is small, the margins are narrow and rarely violated. We generally choose the tuning parameter C via cross-validation. The maximal margin classifier and the soft margin classifier deal only with linear separable data sets. However as we have hinted, most problems are linearly non-separable. That means, there exists no hyperplane that can separate the data sets. Therefore we apply the support vector machines.

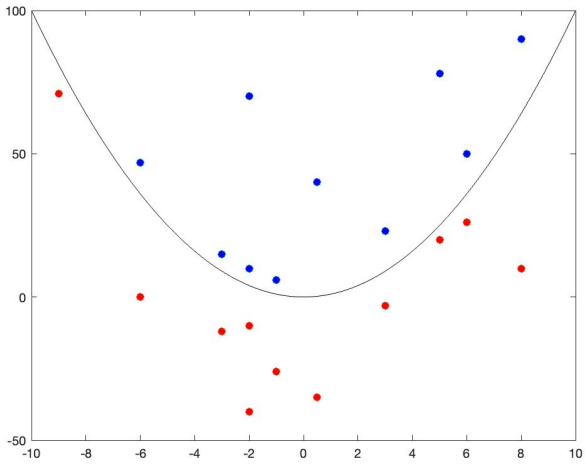

Figure 6: Non-linear problem

Support Vector Machines

Support vector machine is a generalisation of soft margin classifier. Since the problem is not linearly separable in input space, we enlarge the space using kernels. First of all we define the generalized definition of hyperplane. Let V be a vector space with inner product $\Phi : \mathbb{R}^n \to V$, then the hyperplane can be written as:

$$h(X) = \beta_0 + \beta^T \phi(X).$$

The function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$,

$$K(X_i, X_j) = \phi(X_i)^T \phi(X_j),$$

is referred to as kernel function.

Then the modified optimization is defined by:

$$\min_{\beta_0,\beta,\epsilon_i} \frac{\|\beta\|^2}{2} + C\sum_{i=1}^{m}\epsilon_i$$

$$\text{s.t. } y_i(\beta_0 + \beta^T\Phi(X_i)) \geq 1 - \epsilon_i \ , \ \epsilon_i \geq 0 \ , \quad \forall i \in \{1,\ldots,m\},$$

$$\sum_{i=1}^{m}\epsilon_i \leq C.$$

There are some popular kernel functions. For instance, we simply take:

$$K(X_i, X_j) = X_i^T X_j,$$

that is known as a linear kernel. The linear kernel leads us back to the soft margin classifier, because the soft margin classifier is also linear in the input space. Some other kernel functions are:

1. Polynomial kernel of degree d

$$K(X_i, X_j) = (1 + X_i^T X_j)^d,$$

2. Sigmoid kernel

$$K(X_i, X_j) = tanh(\alpha X_i^T X_j + \gamma),$$

3. Gaussian radial basis kernel

$$K(X_i, X_j) = exp(-\gamma\|X_i - X_j\|^2).$$

Confusion Matrix

A confusion matrix is a table used to describe the performance of a classification model. It compares the predictions with the actual value. The table consists of 4 different combinations of predicted and actual value.

- **True Negative (TN)**: We predicted negative and it is actually negative.
- **True Positive (TP)**: We predicted positive and it is actually positive.
- **False Positive (FP)**: We predicted positive but it is actually negative. This is also called as "Type 1 Error"
- **False Negative (FN)**: We predicted negative but it is actually positive. This is also called as "Type 2 Error"

The performance metrics for confusion matrix are *accuracy, sensitivity and specificity*, which are calculated on the basis of classifier above.

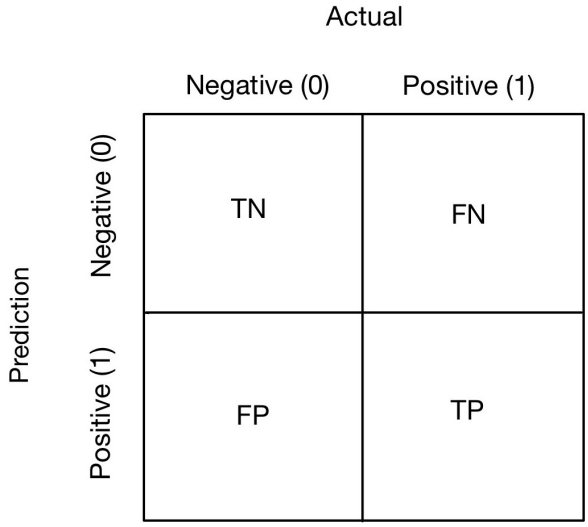

Figure 7: Confusion matrix table

Accuracy represents the ratio of correctly classified points to the total number of points.

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

Sensitivity represents the ratio of correctly predicted positive points to all actual positives.

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

Specificity represents the ratio of correctly predicted negative points to all actual negatives.

$$Specificity = \frac{TN}{TN + FP}$$