DIGITAL CULTURE
ITMO UNIVERSITY

Logistic Regression

High School of Digital Culture
ITMO University
dc@itmo.ru

Contents

1 Logistic Regression

1.1 Introduction. Generative and Discriminative Algorithms

Hello everyone! In this module, we will continue to explore methods applied to classification problems and introduce one more way of algorithm quality estimation, which is called ROC analysis.

We will open our discussion with a review of a binary classification method termed logistic regression. Those who listen carefully may ask what classification has to do with regression considering they are two different problems. Regression predicts a value (in particular, a continuous variable), and classification predicts a class (a variable that usually takes a finite number of values). Well, let's find out what's going on. In fact, a binary logistic regression algorithm outputs the probability of assigning an object to one of the two classes. Probability is a value in the closed interval $[0, 1]$, which means it is a continuous variable and explains why it has 'regression' in its name. At the same time, an observation is assigned a class based on the value of the obtained probability. So, here we also have classification. It turns out there's no inconsistency, though the terms may seem confusing at first.

The second question is, probably, why we need one more classifier when we have studied the naive Bayes classifier (a probabilistic one) in the previous module. The difference will be clear once we approach the development of a probability model.

Let's begin with the formulation of the problem. Let X be a set of objects, and each object is described by p features that are random variables $X_1, X_2, ..., X_p$ and a response Y that takes the values from the set $\{-1, 1\}$. We will think of $X \times Y$ as a probability space with some joint probability distribution. Of course, we are interested in the probability estimator $\mathsf{P}(Y = 1|X_1, X_2, ..., X_p)$ or the estimator of the opposite probability $\mathsf{P}(Y = -1|X_1, X_2, ..., X_p)$. In the case of the binary classification, they are related as follows:

$$\mathsf{P}(Y = 1|X_1, X_2, ..., X_p) + \mathsf{P}(Y = -1|X_1, X_2, ..., X_p) = 1.$$

When we were considering the naive Bayes classifier, we used the Bayes' formula to move to probabilities of a different kind:

$$\mathsf{P}(Y|X_1, X_2, ..., X_p) = \frac{\mathsf{P}(Y, X_1, X_2, ..., X_p)}{\mathsf{P}(X_1, X_2, ..., X_p)} = \frac{\mathsf{P}(X_1, X_2, ..., X_p|Y)\mathsf{P}(Y)}{\mathsf{P}(X_1, X_2, ..., X_p)}.$$

When classifying a new observation, we iterated through all the possible values y of the response Y (mathematicians call it **a maximum a posteriori probability**

(MAP) estimation) to find the maximum of the expression:

$$F(y) = \mathsf{P}(X_1, X_2, ..., X_p|Y = y)\mathsf{P}(Y = y),$$

and we also estimated the probabilities $\mathsf{P}(Y)$ and $\mathsf{P}(X_1, X_2, ..., X_p|Y)$ based on training data and a naive assumption. To put it differently, the numerators written several lines above allow us to conclude that we modeled the joint probability distribution $\mathsf{P}(Y, X_1, X_2, ..., X_p)$ for the classification.

Definition 1.1.1 *Algorithms that model the joint probability distribution* $\mathsf{P}(Y, X_1, X_2, ..., X_p)$ *are often called* ***generative*** *algorithms.*

Opposed to them are so-called discriminative algorithms.

Definition 1.1.2 *Algorithms that model* $\mathsf{P}(Y|X_1, X_2, ..., X_p)$ *are called* ***discriminative*** *algorithms.*

The next logistic regression algorithm we are going to discuss is a discriminative algorithm. It is based on the assumption about the (parametric) distribution of conditional probabilities:

$$\mathsf{P}(Y = 1|X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-(\theta_0+\theta_1 X_1+...\theta_p X_p)}},$$

where the algorithm parameters (the coefficients $\theta_0, \theta_1, ..., \theta_p$) are estimated based on the training dataset.

Let's look at the difference between the generative and discriminative models. Assume that the goal is to learn to distinguish cats from dogs. A discriminative model is trying to create such a boundary in a p-dimensional feature space $\mathbb{R}^p$ so it separates (or almost separates) the training data of different classes (a straight line, a plane, and a manifold). To classify a new observation, it's sufficient to identify on which side of the boundary the test observation lies.

As mentioned earlier, generative models model the distribution for each class separately. In the cat and dog example, we first develop a model describing how cats look like and another model for dogs. Then, to classify a new observation, we simply need to find the model (for cats or dogs) that best matches the observation. That's the difference.

Now that you know the difference between the algorithm types, it's time to describe the logistic regression algorithm. Before we do this, let's see why it's logical to use the conditional probability relation introduced earlier:

$$\mathsf{P}(Y = 1|X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-(\theta_0+\theta_1 X_1+...\theta_p X_p)}}.$$

1.2 Development of a Logistic Regression Model

According to the formulation of the problem, each object with a set of predictors $X_1, X_2, ..., X_p$ should be assigned to one of the two classes: $+1$ (conditionally, a positive class) or -1 (conditionally, a negative class). As has been noted, the probabilities of assigning the object to the positive class

$$\mathsf{P}_+ = \mathsf{P}(Y = 1|X_1, X_2, ..., X_p)$$

and to the negative class

$$\mathsf{P}_- = \mathsf{P}(Y = -1|X_1, X_2, ..., X_p)$$

are determined by the normalization condition, that is, the relation:

$$\mathsf{P}_+ + \mathsf{P}_- = 1.$$

Thus, if we learn to estimate the probability P_+ of assigning the object of interest to the positive class, the probability of assigning the object to the opposite class will be given by the expression $\mathsf{P}_- = 1 - \mathsf{P}_+$. But how do we get these estimators?

Remark 1.2.1 *Note that the values of* P_+ *and* P_- *are functions of* $X_1, X_2, ..., X_p$. *For ease of notation, we suppress the function arguments.*

Now, let's move on from the probabilities in the closed interval $[0, 1]$ to the so-called odds. We are going to consider the value of $\mathsf{P}_+ \in [0, 1]$, which is the probability of assigning the object with the predictors $X_1, X_2, ..., X_p$ to a positive class. Hence, P_- is the probability of the opposite event or the probability of assigning the same object to the negative class.

Definition 1.2.1 *The odds of assigning the object with predictors* $X_1, X_2, ..., X_p$ *to the positive class are the value*

$$\text{odds}_+ = \text{odds}_+(X_1, X_2, ..., X_p) = \begin{cases} \frac{\mathsf{P}_+}{\mathsf{P}_-}, & \mathsf{P}_- \neq 0 \\ +\infty, & \mathsf{P}_- = 0 \end{cases}.$$

For example, if the probability P_+ of assigning the object to the positive class equals 0.8, the odds will be $4 : 1$, since

$$\text{odds}_+ = \frac{0.8}{1 - 0.8} = 4,$$

well, it's quite vivid and intuitive, right? The next question is why odds are better than probability? Here's the thing. Unlike probability, odds take various non-negative values, that is,

$$\text{odds}_+ \in [0, +\infty].$$

Taking the logarithm of the odds, we obtain the value that now can take any values from the (extended) set of real numbers:

$$\ln(\text{odds}_+) = \ln\left(\frac{\mathsf{P}_+}{1-\mathsf{P}_+}\right) \in [-\infty, +\infty].$$

Thus, the straightforward calculations have provided us with a continuous variable that depends on $X_1, X_2, ..., X_p$ and takes any values in the closed interval $[-\infty, +\infty]$. So, our task has turned into a regression problem that we know how to solve. However, here's a catch. Since we don't know the values of P_+, we also don't know $\ln(\text{odds}_+)$. On the other hand, we wouldn't bother with estimating if we knew. Let's ignore this for now and obtain the regression equation:

$$\ln\left(\frac{\mathsf{P}_+}{1-\mathsf{P}_+}\right) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_p X_p.$$

At this point, we don't know the values $\theta_0, \theta_1, ..., \theta_p$.

For convenience, we designate the right-hand side of the expression by Ψ:

$$\Psi = \Psi(X_1, X_2, ..., X_p) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_p X_p$$

From the equality

$$\ln\left(\frac{\mathsf{P}_+}{1-\mathsf{P}_+}\right) = \Psi,$$

we express the probability P_+. Taking the exponents (or exponentiating the equality), we obtain the expression:

$$e^{\Psi} = \frac{\mathsf{P}_+}{1-\mathsf{P}_+},$$

and we rewrite it as follows:

$$(1-\mathsf{P}_+)\cdot e^{\Psi} = \mathsf{P}_+,$$

hence,

$$\mathsf{P}_+ = \frac{e^{\Psi}}{1+e^{\Psi}}.$$

We transform the last fraction, which leads us to the expression:

$$\mathsf{P}_+ = \frac{1}{e^{-\Psi}\cdot(1+e^{\Psi})} = \frac{1}{1+e^{-\Psi}} = \frac{1}{1+e^{-(\theta_0+\theta_1 X_1+\ldots+\theta_p X_p)}}.$$

Note that it is a parametric family mentioned in the introduction.

Remark 1.2.2 *You can also note that since* Ψ *equated to* $\ln(\text{odds}_+)$ *takes values in the closed interval* $[-\infty, +\infty]$*, then* $e^{-\Psi}$ *takes values in the closed interval* $[0, +\infty]$*. Hence,* $\mathsf{P}_+ \in [0, 1]$.

1.3 Logistic Function and Prediction Algorithm

Before we discuss the important question of finding the coefficients $\theta_0, \theta_1, ..., \theta_p$, we need to carefully consider the analytical expression obtained for the probability P_+. It turns out that this expression is closely related to the so-called logistic function or sigmoid function.

Definition 1.3.1 *The function*

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

is called a logistic function or a sigmoid function.

You may notice the similarity between the expressions for P_+ and the logistic function:

$$\mathsf{P}_+ = \frac{1}{1 + e^{-\Psi}} \longleftrightarrow \sigma(x) = \frac{1}{1 + e^{-x}}.$$

So how can we use this similarity? To answer this question, we first need to formulate the main properties of the sigmoid function.

Lemma 1.3.1 *The function*

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

has the following properties:

1. *$\sigma(x)$ is an increasing function.*
2. *$\sigma(x)$ is a continuous function on $\mathbb{R}$.*
3. $$\lim_{x \to +\infty} \sigma(x) = 1, \quad \lim_{x \to -\infty} \sigma(x) = 0.$$
4. *$\sigma(x)$ is constrained by a pair of horizontal asymptotes $y = 0$ and $y = 1$.*

Proof. Although the statements are obvious, we realize that it is necessary to give some explanations. Let's begin with the first one. Since the function e^x is increasing, then e^{-x} is decreasing, the function $1 + e^{-x}$ is also decreasing, and the function $\frac{1}{x}$ is decreasing too, so, the composition $\frac{1}{1+e^{-x}}$ increases.

The continuity of the function follows from the continuity of the composition of elementary functions on its domain.

The asymptote equations are obtained from the limits in the third property:

$$\lim_{x \to +\infty} \sigma(x) = 1, \quad \lim_{x \to -\infty} \sigma(x) = 0.$$

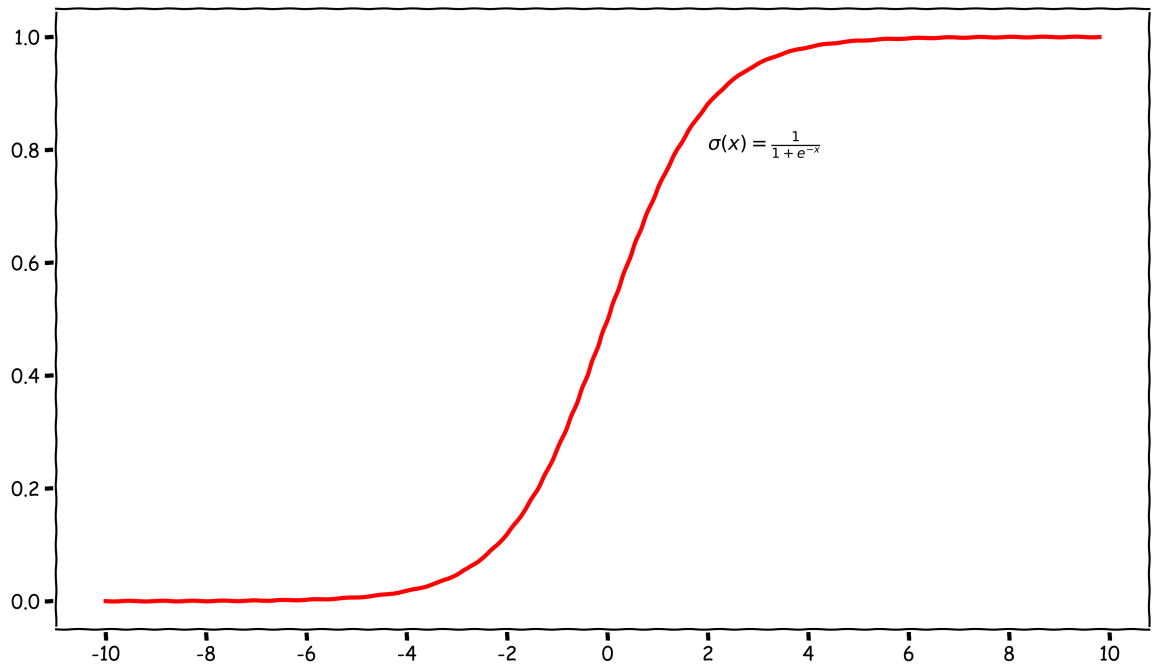

Figure 1: Logistic curve (sigmoid function).

You can verify them using the definition, for example. □
Figure 1 shows the graph of the logistic function. Based on the formulated properties, we conclude that $\sigma(x)$ is a distribution function of a random variable ξ. Thus,

$$\mathsf{P}_+ = \sigma(\Psi) = \mathsf{P}(\xi < \Psi).$$

Remark 1.3.1 *The properties also reveal an interesting fact. The probability cut-off of assigning the object to the class* $+1$ *(or* -1*), that is, the probability* $\mathsf{P}_+ = \mathsf{P}_- = 0.5$ *is the value of the function* $\sigma(x)$ *at point* 0*. In the accepted notation, it is equivalent to the following:*

$$0 = \Psi = \theta_0 + \theta_1 X_1 + \ldots + \theta_p X_p.$$

A set of points satisfying the equality is a hyperplane (a point on the straight line $\mathbb{R}^1$*, a straight line in the plane* $\mathbb{R}^2$*, a plane in the space* $\mathbb{R}^3$*, etc.) in the space* $\mathbb{R}^p$ *divided into two parts by this hyperplane. It makes sense to interpret these parts of the space as follows. One part includes the points for which* $\mathsf{P}_+ > 0.5$*. These points likely fall into the positive class rather than negative. Another part contains the points for which* $\mathsf{P}_+ < 0.5$ *(or what's the same thing,* $\mathsf{P}_- > 0.5$*). These points likely belong to a negative class rather than positive. We say 'likely' because of uncertainty. We will clarify this in a moment.*

When a class boundary is a hyperplane, the classifier is called linear. Hence, a logistic regression algorithm is a linear classifier.

We can formulate the algorithm for predicting the class of the new object z with the predictors $(z_1, z_2, ..., z_p)$ once the coefficients $\theta_0, \theta_1, ..., \theta_p$ are found.

1. Calculate the value Ψ:

$$\Psi = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \ldots + \theta_p z_p.$$

2. Calculate the probability P_+:

$$\mathsf{P}_+ = \frac{1}{1 + e^{-\Psi}}.$$

3. If $\mathsf{P}_+ \geq 0.5$, the object z will fall into the class $+1$ or -1 otherwise.

At this point, we need to pause and ask ourselves why the probability cut-off is the value of 0.5. The choice of this probability is tricky. For example, when we classify emails into two categories of spam and ham, it doesn't make sense to mark the email as junk when the classifier's probability is 0.51. In this case, we apply another rule. If $\mathsf{P}_+ \geq 0.65$, the object z will fall into the class $+1$ (mark as spam) or -1 otherwise (send to the Inbox). Or, for example, we need to determine whether an aircraft is safe to fly. What to do when the classifier's probability prediction about aircraft safety is 0.75? In this case, the probability cut-off should be significantly higher, namely larger than 0.9. In practice, the choice of a probability cut-off is up to a researcher (or other machine learning methods, for example, k-fold cross-validation).

Remark 1.3.2 *The logistic regression algorithm is an algorithm indeed. It is mapping from a feature space to a set of classes, only when the rule of assigning the class label is enforced, or, in other words, if there's the predefined probability starting from which the object is assigned to the class* $+1$ *(or* -1 *by the symmetry). Logistic regression itself (roughly speaking, the function* P_+*) outputs a value in the closed interval* $[0, 1]$*, and it is often called a base algorithm. A rule applied to assign an object to a class based on the predefined probability is called the decision rule. Therefore, a logistic regression algorithm is a composition of a decision rule and a base algorithm of logistic regression.*

Well, let's apply the logistic regression algorithm to specific data. Our data is football statistics. It has three predictors, including shots on target (X_1), possession (X_2), and shots (X_3). The response Y takes only two values. The value 1 corresponds to a win (class $+1$), and the value 0 is a loss or draw (class -1). The training data provides the following values of the model parameters:

$$\theta_0 \approx -0.046, \quad \theta_1 \approx 0.541, \quad \theta_2 \approx -0.014, \quad \theta_3 \approx -0.132.$$

We classify the new object z:

$$z = (1, 40, 3).$$

It's a team that had 1 shot on target, 40 percent of possession, and 3 shots. According to the described algorithm, the probability that the team wins equals:

$$\mathsf{P}_+ = \frac{1}{1 + e^{-(\theta_0 + \theta_1 \cdot 1 + \theta_2 \cdot 40 + \theta_3 \cdot 3)}} \approx 0.38.$$

It means that it will likely lose.

When the model parameters are obtained, it's easy to make a prediction. But how to find these parameters? Well, we need to use maximum likelihood estimation.

1.4 Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation is one of the most powerful statistical methods. It uses a sample to estimate the parameters of probability distribution families. You're probably familiar with this method. Anyway, since it underlies a logistic regression algorithm, we are going to discuss it in detail (within the necessary scope).

1.4.1 Heuristic Arguments

Suppose that n independent trials have been carried out, and each has the probability $p \in (0, 1)$ of success. For example, it can be target shooting or a penalty shoot-out. Given the assumptions, the probability of exactly $k \in \{0, 1, .., n\}$ successes in n trials (the probability of the event $B(n, k)$) is calculated using the Bernoulli formula:

$$\mathsf{P}(B(n, k)) = C_n^k p^k (1 - p)^{n-k},$$

where C_n^k is the number of combinations of n elements taken k elements. It equals:

$$C_n^k = \frac{n!}{k!(n - k)!}$$

For example, an experiment is conducted. The series consists of five shots with the probability p of success (it is the probability of a goal) equal to 0.7 for each shot. The experiment was performed twice (independently!). The first trial produced two successes (two goals), and the second trial produced four (4 goals scored). We have the following sample: $X_1 = 2$, $X_2 = 4$. What is the probability to obtain this sample? Due to independence, the probability equals the product of

the probabilities of the corresponding events, each of which is calculated using the Bernoulli formula:

$$\mathsf{P}\left(X_1 = 2, X_2 = 4\right) = \mathsf{P}(X_1 = 2) \cdot \mathsf{P}(X_2 = 4) =$$

$$= \mathsf{P}(B(5,2)) \cdot \mathsf{P}(B(5,4)) = C_5^2 \cdot 0.7^2 \cdot (1-0.7)^3 \cdot C_5^4 \cdot 0.7^4 \cdot (1-0.7)^1.$$

The calculations show that the probability of the event is low, and it approximately equals:

$$\mathsf{P}\left(X_1 = 2, X_2 = 4\right) \approx 0.048.$$

These calculations were possible only because we knew the probability of scoring a goal for each attempt. However, it's not so easy in practice because we usually observe some regularities related to a random variable (a sample from it), and, if we are lucky, we know which parametric family of probability distributions it obeys. At the same time, we don't know the parameters of these distributions. And now the only thing to do is to estimate the parameters using a sample. How do we do it? We will try to maximize the probability of the observed values.

Let's continue to consider the example. It's logical to assume that a random variable (a number of goals scored in the series of 5 shots) has the binomial distribution with the parameters $n = 5$ (the number of trials) and the unknown parameter p (the probability of success in each trial). In this case, the probability of the event $B(5,k)$ given $k \in \{0,1,2,...,5\}$ (of an event that exactly k goals scored in the series of five shots) can be calculated by the Bernoulli formula:

$$\mathsf{P}(B(5,k)) = C_5^k p^k (1-p)^{5-k}.$$

The value of p that is a probability of scoring a goal for each attempt is unknown. However, we observe the following sample after two independent trials: $X_1 = 2$, $X_2 = 4$. Since the trials are independent, the probability to obtain this sample can be calculated as follows:

$$f\left(X_1 = 2, X_2 = 4, p\right) = \mathsf{P}(X_1 = 2, X_2 = 4) = \mathsf{P}(X_1 = 2) \cdot \mathsf{P}(X_2 = 4) =$$

$$= C_5^2 \cdot p^2 \cdot (1-p)^3 \cdot C_5^4 \cdot p^4 \cdot (1-p)^1.$$

Here's a function of p. Our task is to find such a value of p that maximizes the function over the interval $[0,1]$. This value of p ensures that our sample is the most probable one. It makes sense to interpret the obtained p value as an estimator of a true unknown value.

The ends of the interval are not the values of interest, since the function is zero at the points 0 and 1. We will assume that $p \in (0,1)$. What we are confronting here is a classic problem of mathematical analysis, namely the problem of finding a point at which the function attains its maximum on the given set. To solve this, we apply the following algorithm:

1. Find the first-order derivative.
2. Find the points of the function domain where the derivative is zero or does not exist. All these points are called critical points.
3. Verify that a critical point is a local maximum. To do so, we need a sufficient condition. For example, a critical point is a local maximum if the derivative changes sign from positive to negative at this point.
4. Compare the function values at the local maxima with the values on the boundary of the set (if any). Choose the largest. Identify the point at which the function attains its maximum.

The maximized function $f(X_1 = 2, X_2 = 4, p)$ is a product, which makes it more complicated to find a derivative because we should repeatedly apply the rule of function product differentiation, which is quite cumbersome. To simplify the calculations, we take the logarithm of it and maximize the so-called log-likelihood function:

$$L(X_1 = 2, X_2 = 4, p) = \ln f(X_1 = 2, X_2 = 4, p).$$

Since the logarithm is a monotonic function, the extrema of the function $f(X_1 = 2, X_2 = 4, p)$ will be the extrema of the function $L(X_1 = 2, X_2 = 4, p)$ or vice versa. Thus, after taking the logarithm, the product decomposes into the sum of the logarithms, and we get the following expression:

$$L(X_1 = 2, X_2 = 4, p) = \ln\left(C_5^2 \cdot p^2 \cdot (1-p)^3 \cdot C_5^4 \cdot p^4 \cdot (1-p)^1\right) =$$

$$= \ln C_5^2 + \ln C_5^4 + 6\ln p + 4\ln(1-p).$$

Remark 1.4.1 *Note that, since $p \in (0,1)$, all of the transformations are legitimate.*

So, it boils down to the problem of finding the parameter value $p \in (0,1)$ that maximizes the log-likelihood function:

$$L(X_1 = 2, X_2 = 4, p) = \ln(C_5^2 \cdot C_5^4) + 6\ln p + 4\ln(1-p).$$

To find critical points, we calculate the derivative. It equals:

$$\left(L(X_1 = 2, X_2 = 4, p)\right)'_p = \frac{6}{p} - \frac{4}{1-p}.$$

By equating the derivative to zero, we get the equation:

$$\frac{6}{p} - \frac{4}{1-p} = 0,$$

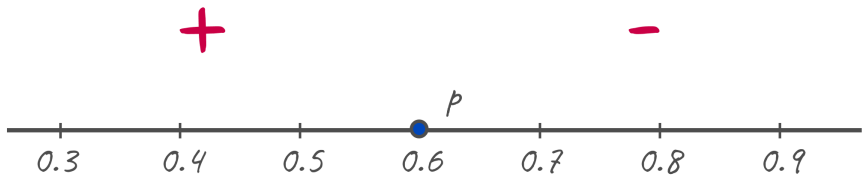

Figure 2: Intervals where the function L is increasing or decreasing.

hence, $p = 0.6$. Let's make sure that the obtained point is a maximum. We check the signs of the derivative of the function on the left and right of the point 0.6. As you can see, the derivative changes its sign from positive to negative. Thus, the obtained value $p = 0.6$ is the maximum. Thus, the probability to score two goals in the first series of five shots and four goals in the second is maximized when $p = 0.6$.

To find the probability value of the event $X_1 = 2, X_2 = 4$ given the obtained $p = 0.6$, we simply calculate:

$$f(X_1 = 2, X_2 = 4, 0.6) = C_5^2 \cdot 0.6^2 \cdot (1 - 0.6)^3 \cdot C_5^4 \cdot 0.6^4 \cdot (1 - 0.6)^1 \approx 0.06.$$

Thus, given the obtained value p, the probability of the event $X_1 = 2, X_2 = 4$ is higher than the previous one. It's not surprising because we've found such a value of p that maximizes our observations (two successes in the first trial and four in the second). The example is clear now, so we can move on to the general description of maximum likelihood estimation.

1.4.2 Maximum Likelihood Estimation

Let X be a sample of size n. The sample elements $X_1, X_2, \ldots, X_n$ are independent, identically distributed, and their distribution $\mathcal{P}_\theta$ depends on the parameter θ. This parameter takes the values from a set Θ, in other words, $\theta \in \Theta$. For example, in the previous example, the family of distributions is a family of binomial distributions $\mathcal{P}_\theta = \mathsf{Bin}(5, p)$ depending on the parameter $\theta = p$, while the set of values Θ of the parameter θ is a closed interval $[0, 1]$.

Maximum likelihood estimation is a method of estimating this parameter. Roughly speaking, as the maximum likelihood value θ, we choose such a value that, on n trials, maximizes the probability of getting the sample $x_1, x_2, \ldots, x_n$ obtained after the experiment. Let's write the function

$$f(X, \theta) = \mathsf{P}_\theta \left(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\right).$$

The function $f(X, \theta)$ shows the probability of the event that the elements of the sample $X_1, X_2, ..., X_n$ are equal to the particular values $x_1, x_2, \ldots, x_n$ respectively. Since the sample elements $X_1, X_2, \ldots, X_n$ are independent, we can move on to

the product of the probabilities:

$$f(X,\theta) = \mathsf{P}_\theta\left(X_1 = x_1\right) \cdot \mathsf{P}_\theta\left(X_2 = x_2\right) \cdot \ldots \cdot \mathsf{P}_\theta\left(X_n = x_n\right).$$

Definition 1.4.1 *The function*

$$f(X,\theta) = \mathsf{P}_\theta\left(X_1 = x_1\right) \cdot \mathsf{P}_\theta\left(X_2 = x_2\right) \cdot \ldots \cdot \mathsf{P}_\theta\left(X_n = x_n\right)$$

is called the likelihood function.

Definition 1.4.2 *A maximum likelihood estimator (MLE) $\widehat{\theta}$ of an unknown parameter θ is a value of $\widehat{\theta} \in \Theta$ that maximizes the likelihood function $f(X,\theta)$.*

To simplify the problem, we rewrite it as follows:

$$\widehat{\theta} = \operatorname{Arg}\max_\theta f(X,\theta),$$

where $f(X,\theta)$ is a likelihood function. As the name suggests, we are trying to find the argument (or arguments) that maximizes the function instead of the value of the function maximum.

Remark 1.4.2 *There may be several arguments of θ for which the function $f(X,\theta)$ attains its maximum. In such a case, the estimator $\widehat{\theta}$ can be any element of the set* $\operatorname{Arg}\max_\theta f(X,\theta)$.

As noted in the example, for ease of calculations, we ignore the likelihood function $f(X,\theta)$ and look at its logarithm, i.e., the so-called log-likelihood function.

Definition 1.4.3 *Let $f(X,\theta)$ be a likelihood function. The function*

$$L(X,\theta) = \ln f(X,\theta) = \ln \mathsf{P}_\theta\left(X_1 = x_1\right) + \ldots + \ln \mathsf{P}_\theta\left(X_n = x_n\right).$$

is called the log-likelihood function.

Since the logarithm is a monotonic function, the extrema of the function $f(X,\theta)$ will be the extrema of the function $\ln f(X,\theta)$ or vice versa. Therefore, we rewrite the problem as follows:

$$\widehat{\theta} = \operatorname{Arg}\max_\theta L(X,\theta).$$

After studying the mathematical essence, we can use the discussed apparatus to train the logistic regression algorithm.

1.5 Finding Model Parameters

Before we solve the problem of finding the unknown model parameters, we need to recollect the initial assumptions. We assume that the probability of assigning the object with the predictors $X_1, X_2, ..., X_p$ to the positive class $+1$ is defined by the expression:

$$\mathsf{P}_+ = \mathsf{P}_+(X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p)}}.$$

We introduce the designation

$$\Psi = \theta_0 + \theta_1 X_1 + ... + \theta_p X_p,$$

and rewrite the probability expression as follows:

$$\mathsf{P}_+ = \mathsf{P}_+(X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-\Psi}} = \sigma[\Psi], \quad \sigma(x) = \frac{1}{1 + e^{-x}}.$$

Thus, the probability P_- of assigning the object to the negative class -1 equals:

$$\mathsf{P}_- = \mathsf{P}_-(X_1, X_2, ..., X_p) = 1 - \mathsf{P}_+ = 1 - \sigma[\Psi].$$

The algebraic transformations lead us to the following conclusion:

$$\mathsf{P}_- = 1 - \mathsf{P}_+ = 1 - \frac{1}{1 + e^{-\Psi}} = \frac{e^{-\Psi}}{1 + e^{-\Psi}} = \frac{1}{1 + e^{\Psi}}.$$

Hence,

$$\mathsf{P}_- = \mathsf{P}_-(X_1, X_2, ..., X_p) = \frac{1}{1 + e^{\Psi}} = \sigma[-\Psi].$$

Therefore, we have a pair of relations:

$$\mathsf{P}_+ = \mathsf{P}_+(X_1, X_2, ..., X_p) = \frac{1}{1 + e^{-\Psi}} = \sigma[\Psi],$$

$$\mathsf{P}_- = \mathsf{P}_-(X_1, X_2, ..., X_p) = \frac{1}{1 + e^{\Psi}} = \sigma[-\Psi].$$

Let $X = \{x_1, x_2, ..., x_n\}$ be a training dataset of size n,

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\},$$

and each object x_i corresponds to the response $y_i \in Y = \{-1, 1\}$. For convenience, we introduce the following designation:

$$M(\theta, x_i) = y_i(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}).$$

If the training object belongs to the class $+1$, then, since $y_i = 1$,

$$\sigma\left[M(\theta, x_i)\right] = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip})}} = \mathsf{P}_+,$$

and if to the class -1, then, since $y_i = -1$,

$$\sigma\left[M(\theta, x_i)\right] = \frac{1}{1 + e^{(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip})}} = \mathsf{P}_-.$$

Given the training dataset with confident responses, all obtained probabilities should be as close to one as possible. We can adjust the proximity by changing the values $\theta_0, \theta_1, ..., \theta_p$. It makes sense to use the maximum likelihood estimation described earlier. In this case, we rewrite the likelihood function as follows:

$$f(X, \theta) = \sigma\left[M(\theta, x_1)\right] \cdot \sigma\left[M(\theta, x_2)\right] \cdot ... \cdot \sigma\left[M(\theta, x_n)\right] = \prod_{i=1}^{n} \sigma\left[M(\theta, x_i)\right].$$

We maximize the log-likelihood function as the likelihood function, and it takes the following form:

$$L(X, \theta) = \sum_{i=1}^{n} \ln\left(\sigma\left[M\left(\theta, x_i\right)\right]\right).$$

We are also interested in such a set of θ values so that

$$\theta = \operatorname{Arg}\max_{\theta} L(X, \theta).$$

On the other hand, we can use the properties of logarithms to rewrite the problem as follows:

$$\operatorname{Arg}\max_{\theta} L(X, \theta) = \operatorname{Arg}\max_{\theta} \sum_{i=1}^{n} \ln\left(\sigma\left[M(\theta, x_i\right]\right) =$$

$$= \operatorname{Arg}\max_{\theta} \sum_{i=1}^{n} \ln\left(1 + e^{-M(\theta, x_i)}\right)^{-1} = \operatorname{Arg}\min_{\theta} \sum_{i=1}^{n} \ln\left(1 + e^{-M(\theta, x_i)}\right),$$

since the maximization of the function

$$\sum_{i=1}^{n} \ln\left(1 + e^{-M(\theta, x_i)}\right)^{-1} = -\sum_{i=1}^{n} \ln\left(1 + e^{-M(\theta, x_i)}\right)$$

is the same as the minimization of the function

$$\text{logloss}(X, \theta) = \sum_{i=1}^{n} \ln\left(1 + e^{-M(\theta, x_i)}\right),$$

to put it differently, of the same function but without the minus sign (formally, of the same function multiplied by -1).

Definition 1.5.1 *The introduced function* logloss *is called the logistic error function (log-loss function).*

Remark 1.5.1 *Note that the log-loss function is a special case of the empirical risk*

$$Q(a, L, x_1, x_2...x_n) = \frac{1}{n}\sum_{i=1}^{n} L(a, x_i),$$

where $L(a, x_i)$ is a loss function discussed earlier. To prove it, we simply consider the following function as a loss function:

$$L(a, x) = n \ln\left(1 + e^{-M(\theta,x)}\right).$$

As mentioned earlier, we are looking for such an algorithm or such parameters $\theta_0, \theta_1, ..., \theta_p$ that minimize the empirical risk.

The loss function is natural because the less the object x agrees with the response, the more the value of the function $1 + e^{-M(\theta,x)}$, and the more valuable the logarithm is for the empirical risk.

The obtained function

$$\text{logloss}(X, \theta) = \sum_{i=1}^{n} \ln\left(1 + e^{-M(\theta,x_i)}\right),$$

is not minimized manually anymore. It is usually done numerically using, for example, the gradient descent, but we will not focus on this today. We can use modeling tools or mathematical packages to solve maximization or minimization problems.

1.6 An Example of the Optimization Problem Formulation

Let's consider the sequence of actions that allow us to formulate the minimization problem described earlier. To begin with, we will consider the subset of the football statistics data. The entire table is provided in the additional materials.

Win or loss	Shots on target	Possession %	Shots
1	7	40	13
0	0	60	6
0	3	43	8
1	4	57	14
...	...	...	...

For clarity, we will assume that the input data is the data in the table. The data in the first column corresponds to responses, and the data in columns 2-4 to predictors. Each row corresponds to its training data item. In the accepted notation, our objects are as follows:

$$x_1 = (7, 40, 13), \quad y_1 = 1,$$

$$x_2 = (0, 60, 6), \quad y_2 = 0,$$
$$x_3 = (3, 43, 8), \quad y_3 = 0,$$
$$x_4 = (4, 57, 14), \quad y_4 = 1.$$

We substitute the values of the class 0 for the values of the class -1 to use the accepted notation. Then,

$$x_1 = (7, 40, 13), \quad y_1 = 1,$$

$$x_2 = (0, 60, 6), \quad y_2 = -1,$$
$$x_3 = (3, 43, 8), \quad y_3 = -1,$$
$$x_4 = (4, 57, 14), \quad y_4 = 1.$$

We rewrite the log-loss function as follows:

$$\text{logloss}(X, \theta) = \sum_{i=1}^{4} \ln\left(1 + e^{-M(\theta, x_i)}\right) = \sum_{i=1}^{4} \ln\left(1 + e^{-y_i(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \theta_3 x_{i3})}\right)$$

It's clear that the expressions on the training data for $M(\theta, x_i)$ are as follows:

$$\begin{aligned} M(\theta, x_1) &= +(\theta_0 + 7 \cdot \theta_1 + 40 \cdot \theta_2 + 13 \cdot \theta_3), \\ M(\theta, x_2) &= -(\theta_0 + 0 \cdot \theta_1 + 60 \cdot \theta_2 + 6 \cdot \theta_3), \\ M(\theta, x_3) &= -(\theta_0 + 3 \cdot \theta_1 + 43 \cdot \theta_2 + 8 \cdot \theta_3), \\ M(\theta, x_4) &= +(\theta_0 + 4 \cdot \theta_1 + 57 \cdot \theta_2 + 14 \cdot \theta_3). \end{aligned}$$

The resultant function $\text{logloss}(X, \theta)$ is a function of the four variables that we will minimize. You can see the example of numerical minimization in the additional materials.

1.7 Margin and Classification Confidence

Let's turn to another example and discuss how to obtain a geometric interpretation of classification using logistic regression. As mentioned, we obtain the equation of a hyperplane once the model is trained:

$$\theta_0 + \theta_1 X_1 + ... + \theta_p X_p = 0.$$

In a sense, it separates the elements of one class from those of another. In two dimensions, a hyperplane is a straight line in a plane, in three dimensions, it's a plane in a space, etc. The next simple example clearly shows that a straight line can accurately split the representatives of different classes into two classes (Figure 3). Having trained the logistic regression model on the provided data, we obtain

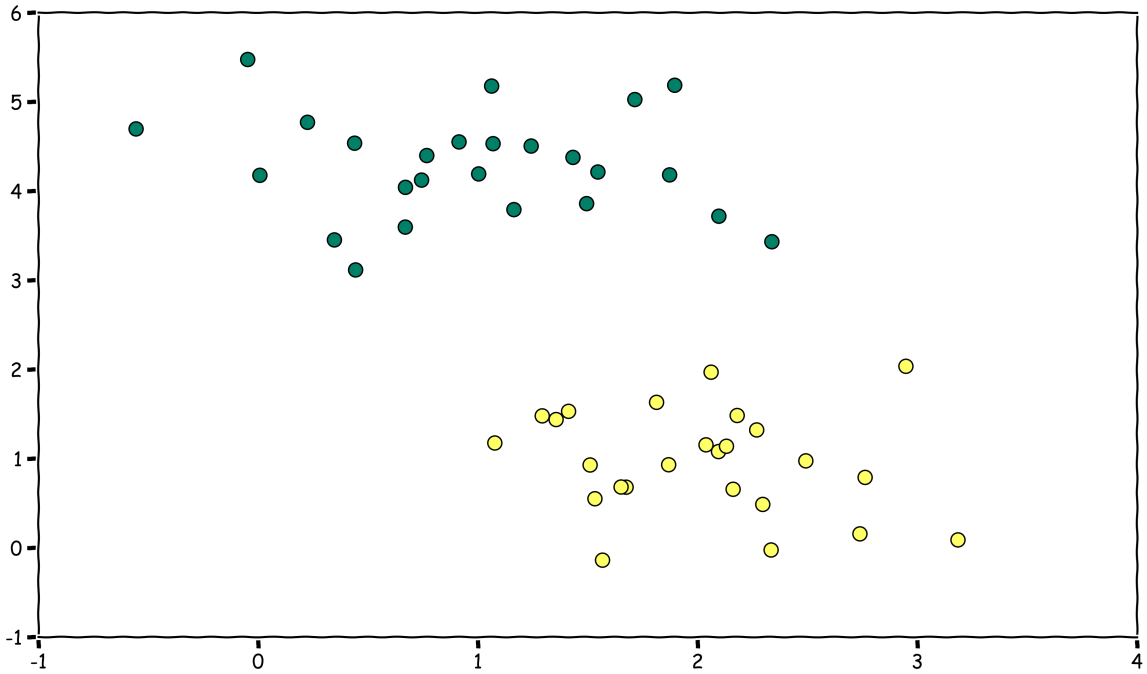

Figure 3: Objects are intuitively separable.

the following hyperplane equation:

$$1.07 + 1.65 \cdot X_1 - 1.55 \cdot X_2 = 0,$$

Figure 4 shows that the constructed straight line has divided the data without errors. Here's an important observation. Despite that we are confident about the

Figure 4: Separation of the objects.

training data (with the probability that is equal to 1), some data items are closer to the separating straight line, while others are further away, and the constructed classifier doesn't output the probability equal to 1 for any item. The classification of the points close to the separating straight line is less confident. The probabilities of assigning to one and another class are near 0.5 (even though one of them prevails because the points don't lie on the separating straight line). On the other hand, the classification of the points further away from the line is more confident.

Definition 1.7.1 *The value*

$$M(\theta, x_i) = y_i(\theta_0 + \theta_1 x_{i1} + ... + \theta_p x_{ip})$$

introduced earlier is called a margin of the object x_i.

In a sense, the margin shows how deeply the object is rooted in the class. The larger the margin, the further away the object is from the separating hyperplane, and the more confident its classification or vice versa.

Remark 1.7.1 *Note that the margin $M(\theta, x_i)$ is negative if and only if the algorithm incorrectly classifies the object. With the training dataset $x_1, x_2, ..., x_n$, we use the margin to write the number of objects that the logistic regression algorithm classified incorrectly:*

$$\widetilde{Q}_{log}(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} \mathsf{I}(M(\theta, x_i) < 0).$$

So, we are looking for such algorithm parameters that will allow us to minimize $\widetilde{Q}_{log}$. However, it's not easy to minimize the written expression. Meanwhile, since

$$\mathsf{I}(M(\theta, x_i) < 0) \leq \log_2 \left(1 + e^{-M(\theta, x_i)}\right),$$

we obtain that

$$\widetilde{Q}_{log}(x_1, x_2, ..., x_n) \leq \sum_{i=1}^{n} \log_2 \left(1 + e^{-M(\theta, x_i)}\right) = \ln 2 \sum_{i=1}^{n} \ln \left(1 + e^{-M(\theta, x_i)}\right) =$$

$$= \ln 2 \cdot \text{logloss}(X, \theta).$$

Thus, by minimizing the logistic loss function, we are trying to reduce the number of errors when classifying the training dataset. It's all connected!

Let us have the following test dataset:

$$A = (0, 2), \quad y_A = 1,$$

$$B = (1, 2), \quad y_B = -1,$$

$$C = (3.5, 4), \quad y_C = 1,$$

$$D = (3, 0), \quad y_D = -1.$$

Let's look at Figure 5 showing the points. Their color corresponds to the initial response (yellow points belong to the class $+1$ and green to -1). As you can see, the object A is classified incorrectly. Its color differs from that of other elements located in the area of this class. We use the margin to understand it analytically:

$$M(\theta, A) = 1 \cdot (1.07 + 1.65 \cdot 0 - 1.55 \cdot 2) = -2.03 < 0.$$

Thus, the classifier makes a mistake on the object A and assigns the test observation to the class (other than a true one).

The objects B and C are close to the separating straight line, and the classifier is not confident about the answer, but the answer is still correct. It's analytically clear:

$$M(\theta, B) = -1 \cdot (1.07 + 1.65 \cdot 1 - 1.55 \cdot 2) = 0.38 > 0,$$

$$M(\theta, C) = 1 \cdot (1.07 + 1.65 \cdot 3.5 - 1.55 \cdot 4) = 0.645 > 0$$

are small, but positive values.

The distance between the object D and the straight line is large, so obviously, it is an outlier. We can prove it analytically:

$$M(\theta, D) = -1 \cdot (1.07 + 1.65 \cdot 3 - 1.55 \cdot 0) = -6.02 < 0.$$

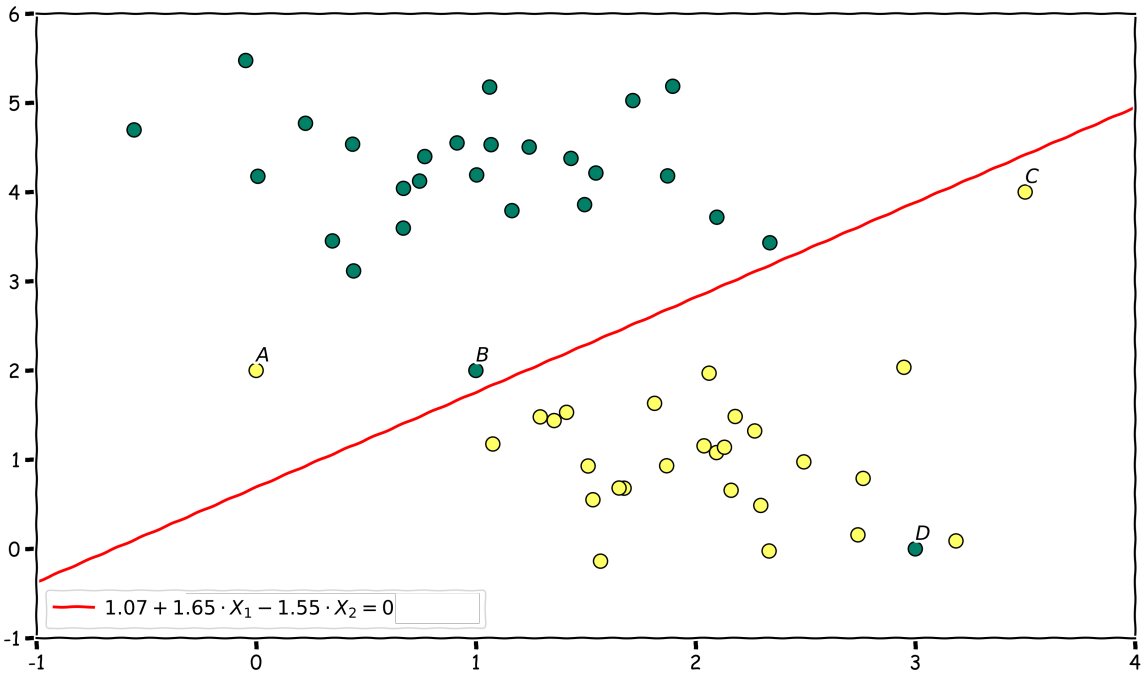

Figure 5: Classification of new objects.

1.8 Linear Regression vs. Logistic Regression

Let's look at some major differences between linear and logistic regression. To do so, we are going to consider the example where the income of a person is a predictor, and loan approval is a response. Figure 6 shows our training data. The income (in thousands of dollars per month) is plotted on the horizontal axis, and the loan approval on the vertical axis (one stands for an approved loan denoted by the yellow points, and zero stands for a declined loan application denoted by the green points). As you can see, an average income doesn't guarantee that the loan application will be approved or declined. When the logistic regression model

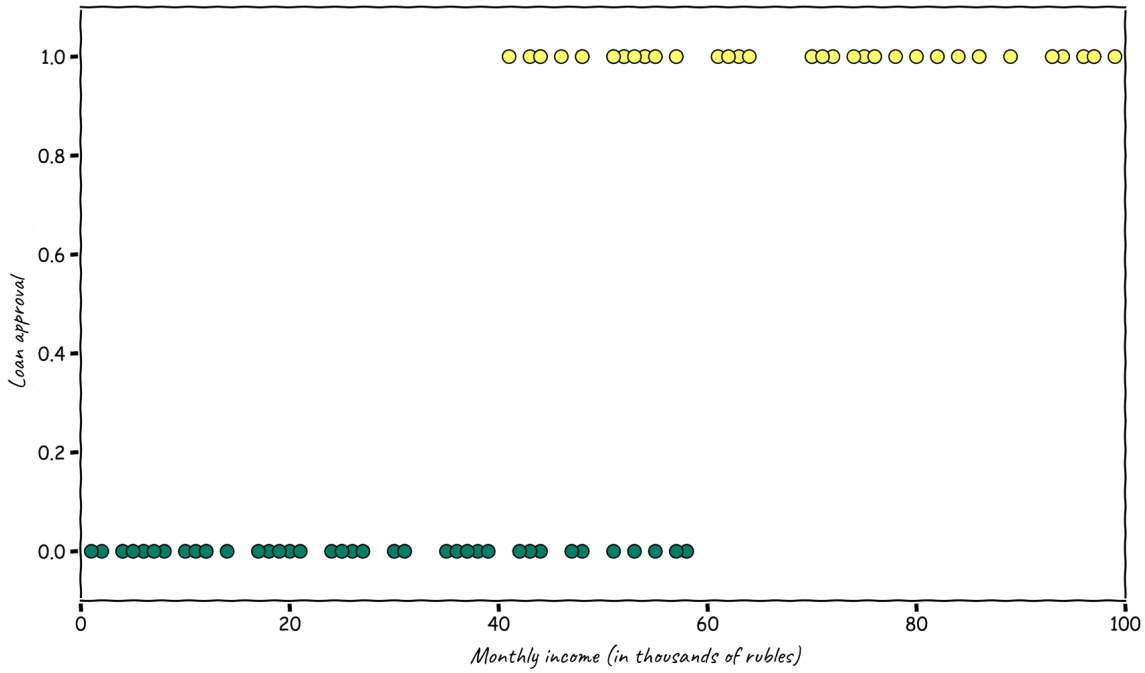

Figure 6: Loan approval training data.

is trained, we obtain the sigmoid function shown in Fig. 7. The graph of the regression line is shown in the figure in blue.

Let's consider a set of new clients with an income of 35, 40, 60, 70, 80 thousand dollars and calculate the probabilities of loan approvals. We obtain $0.11, 0.22, 0.89, 0.98, 0.99$ for the logistic regression. The linear regression provides the probabilities $0.32, 0.39, 0.67, 0.82, 0.96$ (Figures 7-8).

What conclusions can we draw? The classification in the example outputs the same result, which means that both models select the same clients whose applications will be approved and those whose applications will likely be declined (the prediction is less than 0.5). We also see that the models behave differently given average predictors (about 50 thousand dollars in the example). The logistic regression shows the significant change in the probability values due to the changing convexity of the function and its rapid growth given 'average' income values. Meantime, the probability values in the linear model change at the same

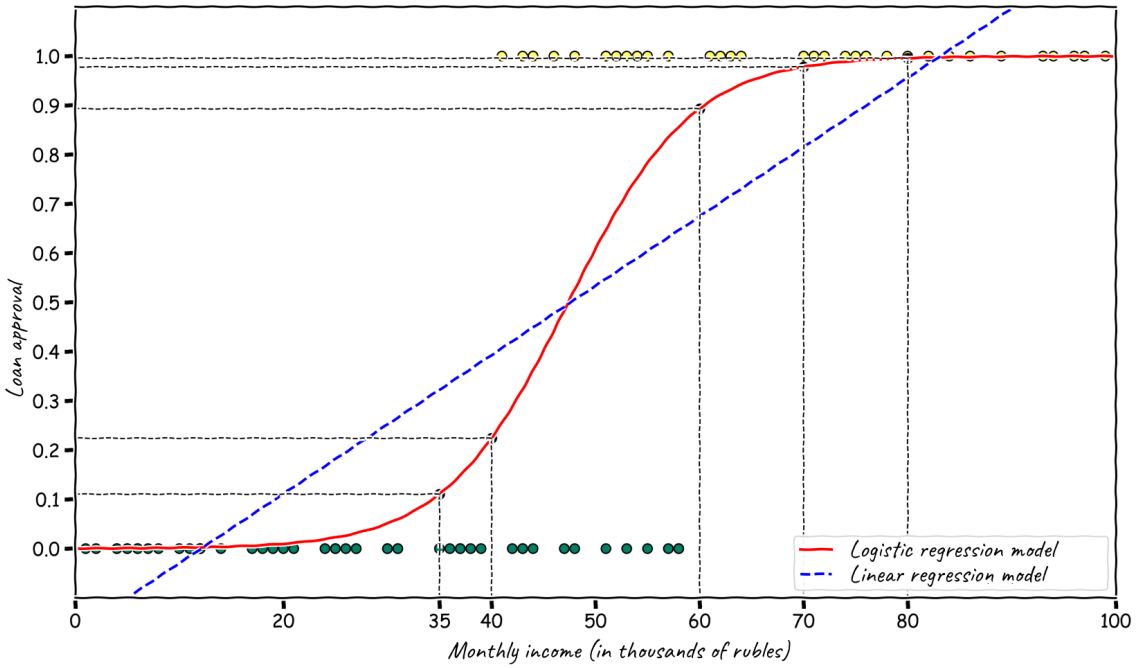

Figure 7: Logistic regression results.

pace within the entire range. These nuances of the models are important for the classification when the number of predictors is high (for example, if we consider such criteria as age, sex, and real estate in possession).

You can also notice that the results of linear regression are odd, even on the training data. There are people for whom it provides the 'result' of less than zero, and those with the 'result' greater than one. How to interpret it? The example once again highlights a problem of linear regression use in classification problems, namely the loss of normalization.

2 Multinomial Logistic Regression

2.1 Model Construction

By now, we've figured out how to solve a binary classification problem using logistic regression. But what if there are more than two classes? It turns out that we can generalize what we discussed. Assume that each object with a set of predictors $X_1, X_2, ..., X_p$ is to be assigned to one of M classes: $Y = \{1, 2, ..., M\}$. We will approximate the so-called relative odds of each class by the corresponding

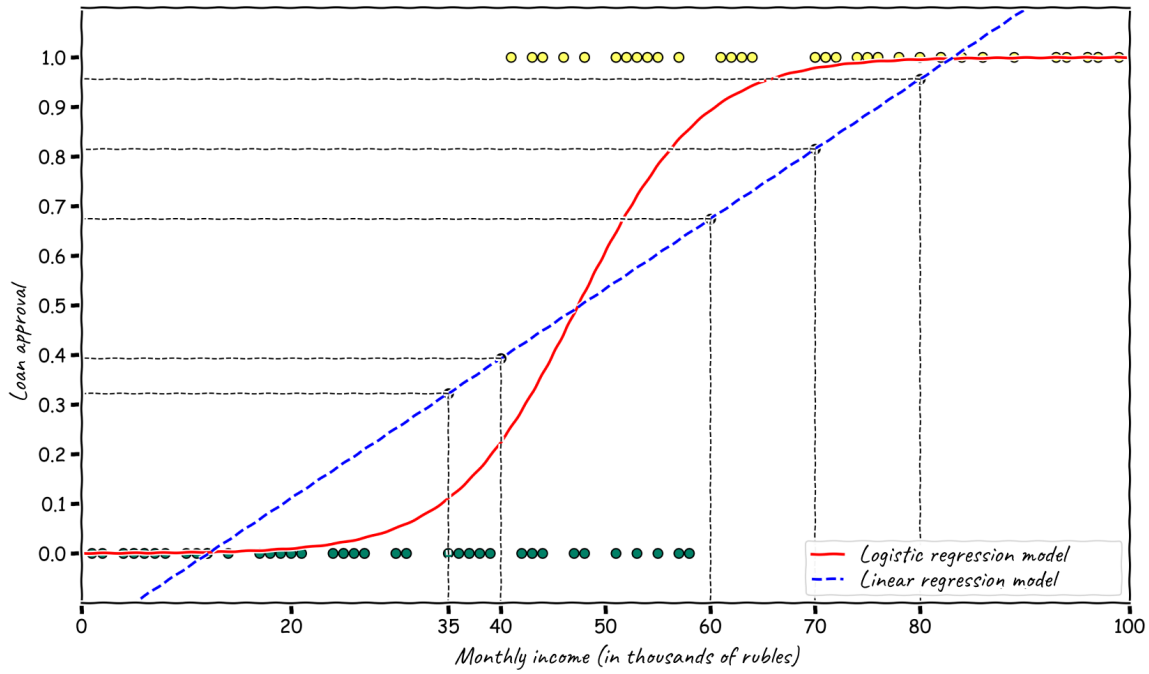

Figure 8: Linear regression results.

line of regression. Thus, we will obtain the following $(M-1)$ equalities:

$$\begin{aligned}
\ln \frac{\mathsf{P}(Y=1|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} &= \theta_0^1 + \theta_1^1 X_1 + ... + \theta_p^1 X_p, \\
\ln \frac{\mathsf{P}(Y=2|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} &= \theta_0^2 + \theta_1^2 X_1 + ... + \theta_p^2 X_p, \\
&\vdots \\
\ln \frac{\mathsf{P}(Y=M-1|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} &= \theta_0^{M-1} + \theta_1^{M-1} X_1 + ... + \theta_p^{M-1} X_p.
\end{aligned}$$

This time, we need to find $(M-1)\cdot p$ unknown parameters. Before we do this, let's obtain the analytical expressions for the probability of assigning a new object to each of the classes dictated by the model. We introduce the following designations

$$\Psi_i = \Psi_i(X_1, X_2, ..., X_p) = \theta_0^i + \theta_1^i X_1 + ... + \theta_p^i X_p, \quad i \in \{1, 2, ..., M-1\},$$

and add the normalization condition to obtain a probability distribution. It leads us to the system of equations:

$$\begin{cases}
\ln \frac{\mathsf{P}(Y=1|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} = \Psi_1, \\
\ln \frac{\mathsf{P}(Y=2|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} = \Psi_2, \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\ln \frac{\mathsf{P}(Y=M-1|X_1,X_2,...,X_p))}{\mathsf{P}(Y=M|X_1,X_2,...,X_p))} = \Psi_{M-1}, \\
\sum\limits_{i=1}^{M} \mathsf{P}(Y=i|X_1, X_2, ..., X_p) = 1.
\end{cases},$$

we solve it to obtain the following analytical expressions for probabilities:

$$\mathsf{P}_k = \mathsf{P}(Y = k|X_1, X_2, ..., X_p) = \frac{e^{\Psi_k}}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i}}, \quad k \in \{1, 2, ..., M-1\},$$

$$\mathsf{P}_M = \mathsf{P}(Y = M|X_1, X_2, ..., X_p) = \frac{1}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i}}.$$

We can formulate the algorithm for predicting the class of the new object z with the predictors $(z_1, z_2, ..., z_p)$ once the coefficients $\theta_0^i, \theta_1^i, ..., \theta_p^i$, $i \in \{1, 2, ..., M-1\}$, are found.

1. Calculate the values of Ψ_k, $k \in \{1, 2, ..., M-1\}$:

$$\Psi_k = \theta_0^k + \theta_1^k z_1 + \theta_2^k z_2 + \ldots + \theta_p^k z_p.$$

2. Calculate the probabilities P_k, $k \in \{1, 2, ..., M-1\}$, based on the relations:

$$\mathsf{P}_k = \frac{e^{\Psi_k}}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i}}, \quad k \in \{1, 2, ..., M-1\}$$

and the probability P_m based on the relations:

$$\mathsf{P}_M = \frac{1}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i}}.$$

3. Assign the test object to any class of the set

$$\underset{y \in \{1,2,...,M\}}{\operatorname{Arg\,max}} \ \mathsf{P}_y$$

Remark 2.1.1 *Note that the last step of the algorithm can be adjusted by a researcher depending on the task. The justification is the same as in binary classification.*

2.2 Finding Model Parameters

Let's briefly describe how to find unknown model parameters. Let $X = \{x_1, x_2, ..., x_n\}$ be a training dataset of size n,

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\},$$

and each object x_i corresponds to the response $y_i \in Y = \{1, 2, ..., M\}$. We introduce the following designations

$$\mathsf{P}_k(x_i) = \frac{e^{\Psi_k(x_{i1}, x_{i2}, ..., x_{ip})}}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i(x_{i1}, x_{i2}, ..., x_{ip})}}, \quad k \in \{1, 2, ..., M-1\}$$

and

$$\mathsf{P}_M(x_i) = \frac{1}{1 + \sum\limits_{i=1}^{M-1} e^{\Psi_i(x_{i1}, x_{i2}, ..., x_{ip})}}.$$

With maximum likelihood estimation, the likelihood function takes the following form:

$$f(X, \theta) = \mathsf{P}_{y_1}(x_1) \cdot \mathsf{P}_{y_2}(x_2) \cdot ... \cdot \mathsf{P}_{y_n}(x_n) = \prod_{i=1}^{n} \mathsf{P}_{y_i}(x_i),$$

and we rewrite the log-likelihood function as follows:

$$L(x, \theta) = \ln f(X, \theta) = \sum_{i=1}^{n} \ln \mathsf{P}_{y_i}(x_i).$$

The written function is maximized by changing the values of the parameters $\theta_0^i, \theta_1^i, ..., \theta_p^i$, $i \in \{1, 2, ..., M-1\}$. We will not discuss it in detail because maximization is performed using numerical analysis, and this subject lies beyond the scope of the course. Lastly, we'd like to note that, similarly to binary classification, the maximization of log-likelihood function leads to the minimization of errors of the algorithm on training data.

2.3 3-Class Classification Example

Let's apply the described algorithm to the data about sweetness and crispness of food. Each object has two predictors X_1 and X_2 and belongs to one of $M = 3$ classes: $Y = \{1, 2, 3\}$, where 1 is fruit, 2 vegetables, and 3 proteins. We know that the data is perfectly separable, so the logistic regression model should do well.

Product	Sweetness	Crispness	Class
banana	10	1	fruit
orange	7	4	fruit
grapes	8	3	fruit
shrimp	2	2	protein
bacon	1	5	protein
nuts	3	3	protein
cheese	2	1	protein
fish	3	2	protein
cucumber	2	8	vegetable
apple	9	8	fruit
carrot	4	10	vegetable
celery	2	9	vegetable
iceberg lettuce	3	7	vegetable
pear	8	7	fruit

We train the multinomial logistic regression algorithm by modeling on the provided data to obtain the following expressions for Ψ_1 and Ψ_2 (with rounded coefficients):

$$\Psi_1 = \Psi_1(X_1, X_2) = -5.561 + 10.786X_1 - 9.976X_2$$

$$\Psi_2 = \Psi_2(X_1, X_2) = -50.441 + 15.765X_1 - 3.961X_2.$$

To find the probability of assigning the object to a class, we use the formulas introduced earlier:

$$\mathsf{P}_k\,(x_i) = \frac{e^{\Psi_k(x_{i1},x_{i2},\ldots,x_{ip})}}{1 + \sum\limits_{i=1}^{2} e^{\Psi_i(x_{i1},x_{i2},\ldots,x_{ip})}}, \quad k \in \{1, 2\},$$

$$\mathsf{P}_3\,(x_i) = \frac{1}{1 + \sum\limits_{i=1}^{2} e^{\Psi_i(x_{i1},x_{i2},\ldots,x_{ip})}}.$$

First, we use the predictors $(6, 9)$ to calculate the values Ψ_k, $k \in \{1, 2\}$ for the object Bell Pepper:

$$\Psi_1 = \Psi_1(6, 9) = -5.561 + 10.786 \cdot 6 - 9.976 \cdot 9 \approx -29.012$$

$$\Psi_2 = \Psi_2(6, 9) = -50.441 + 15.765 \cdot 6 - 3.961 \cdot 9 \approx 8.495.$$

Thus,

$$\mathsf{P}_1 = \frac{e^{\Psi_1}}{1 + e^{\Psi_1} + e^{\Psi_2}} \approx 0,$$

$$\mathsf{P}_2 = \frac{e^{\Psi_2}}{1 + e^{\Psi_1} + e^{\Psi_2}} \approx 1,$$

$$\mathsf{P}_3 = \frac{1}{1 + e^{\Psi_1} + e^{\Psi_2}} \approx 0.$$

The calculations make it clear that the trained algorithm confidently classifies the Bell Pepper as a vegetable, and the probability is close to 1. The figure shows three areas identified by the classifier in the feature space. All points in the upper green area are classified as vegetables, the points to the left in the yellow area are proteins, and the points to the right in the dark green area are fruit. The further the object is from the boundaries, the more confident the classifier is about the classification decision (the higher the probability).

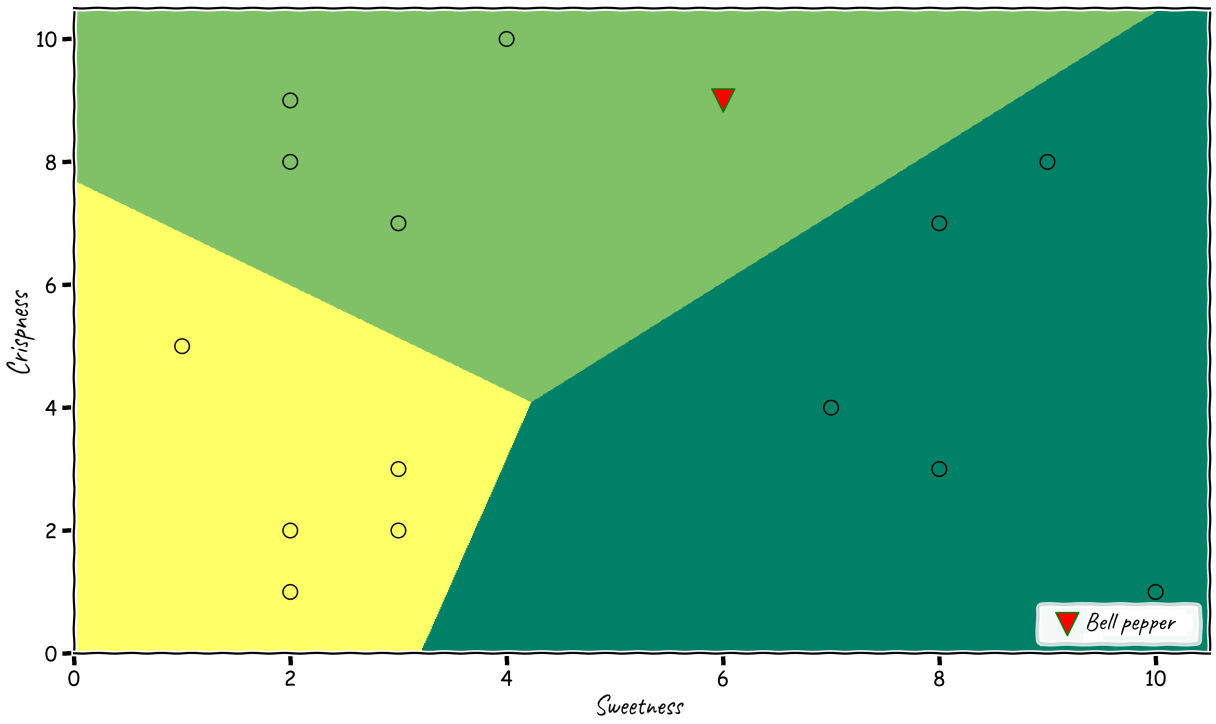

Figure 9: Multivariate logistic regression model.

3 F Score and ROC Analysis

3.1 Confusion Matrix and F Score

ROC curve (also called error curve) is used to estimate the results of binary classification in machine learning. The error curve shows the relationship between the true positive and false positive rate. It is assumed that the classifier has a certain parameter, the change of which affects one or another partition into two classes.

To begin with, we will consider a so-called confusion matrix used to split the objects into 4 groups depending on the combination of a true class and classifier's response:

- TP (True Positives) are correctly classified objects that originally belonged to the class $+1$;
- TN (True Negatives) are correctly classified objects that originally belonged to the class -1;
- FN (False Negatives) are incorrectly classified objects that originally belonged to the class $+1$ (type I error);
- FP (False Positives) are incorrectly classified objects that originally belonged to the class -1 (type II error).

Confusion matrix		Initial class	
		$+$	$-$
Classifier's response	$+$	TP	FP
	$-$	FN	TN

Usually, instead of absolute values, we use relative ones that are called rates and lie between 0 and 1:

- The rate of correct answers of the classifier (sometimes called accuracy):

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

  This value reflects the relationship between the number of correctly classified objects and the total number of classified objects. In other words, it estimates the probability that a random object is correctly classified.

- True Positive Rate (TPR), Sensitivity, or Recall:

$$\mathsf{TPR} = \frac{TP}{TP + FN}.$$

  This value reflects the relationship between the number of correctly classified objects belonging to the class $+1$ and the total number of objects of the class $+1$. To put it differently, it's an estimator of the probability that an object belonging to the class $+1$ will be correctly classified.

- False Positive Rate (FPR):

$$\mathsf{FPR} = \frac{FP}{FP + TN}.$$

This value reflects the relationship between the number of incorrectly classified objects belonging to the class -1 and the total number of objects of the class -1 or estimates the probability that an object belonging to the class -1 will be incorrectly classified.

- Specificity or True Negative Rate (TNR):

$$\mathsf{TNR} = 1 - \mathsf{FPR} = \frac{TN}{TN + FP}.$$

This value reflects the relationship between the number of correctly classified objects belonging to the class -1 and the total number of objects of the class -1 or estimates the probability that an object belonging to the class -1 will be correctly classified.

- Precision:

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

This value reflects the rate of objects that the classifier has assigned to the $+1$ and that actually belong to this class.

These concepts are well explained by an example of the medical application. Assume that patients with a disease belong to a positive class and patients without the disease to negative. A sensitive diagnostic test (with high TPR) is the one that correctly identifies patients with a disease. A test that is 100% sensitive ($\mathsf{TPR} = 1$) correctly identifies all patients who have a disease (which means that all unhealthy patients will find out that they need treatment). However, this test can return a positive result for those who do not have the disease. A sensitive diagnostic test is used to rule out a disease.

A specific diagnostic test (with high TNR) identifies patients without a disease. A test that demonstrates 100% specificity ($\mathsf{TNR} = 1$) will identify patients who do not have the disease. However, it can classify patients with the disease as healthy. This test is done to diagnose a specific disease.

This brings up the question. Is there a common criterion that estimates the quality of the developed model? One of them, so-called F score (F_1 score) is defined by the ratio:

$$F = F_1 = 2 \cdot \frac{\mathsf{Precision} \cdot \mathsf{Recall}}{\mathsf{Precision} + \mathsf{Recall}}.$$

Remark 3.1.1 *F score is the harmonic mean of* Precision *and* Recall *in the closed interval* $[0, 1]$. *The harmonic mean exhibits an important property. It's close to zero if at least one argument is close to zero. That's why it is more preferable than an arithmetic mean. When an algorithm assigns all objects to the positive class, then* $\mathsf{Recall} = 1$, *and* Precision *will likely be low. However, the arithmetic mean that is larger than* 0.5 *won't work.*

Let's return to the football statistics example where we've already trained the model and found the θ values:

$$\theta_0 = -0.046, \theta_1 = 0.541, \theta_2 = -0.014, \theta_3 = -0.132.$$

We use the test dataset in the table to make a confusion matrix.

Win or loss	Shots on target	Possession %	Shots
1	5	60	10
1	2	35	3
0	3	45	6
0	1	53	10
1	7	70	11
1	3	65	12
1	1	30	2
0	2	40	9
1	10	71	15
1	6	54	12
0	7	65	15
0	0	30	3

To begin with, let's find the probability of the team winning for each dataset. The win probability for the team that has 5 shots on target out of 10 and 60 percent possession is:

$$\mathsf{P}_+ = \frac{1}{1 + e^{-(\theta_0 + \theta_1 \cdot 5 + \theta_2 \cdot 60 + \theta_3 \cdot 10)}} \approx 0.588.$$

Which class has the object given the result? If the cut-off value is 0.5, then the class of winners, if 0.6, losers. We find the probabilities of winning the game for all test data and fill out the table with the rounded results. The common practice is not to round the values.

Win Probability
0.588
0.520
0.517
0.186
0.743
0.285
0.446
0.337
0.886
0.671
0.666
0.307

We set 0.5 as the cut-off value and assign the classes.

Win Probability	Win or loss (prediction)
0.588	1
0.520	1
0.517	1
0.186	0
0.743	1
0.285	0
0.446	0
0.337	0
0.886	1
0.671	1
0.666	1
0.307	0

It is easy to link the predicted and initial class. As you can see, they are not always the same, which means that the model is prone to error.

Win or loss	Win Probability	Win or loss (prediction)
1	0.588	1
1	0.520	1
0	0.517	1
0	0.186	0
1	0.743	1
1	0.285	0
1	0.446	0
0	0.337	0
1	0.886	1
1	0.671	1
0	0.666	1
0	0.307	0

Thus, we can make a confusion matrix. We calculate how many teams won the match according to the input data and model prediction and find out there are five of them. Similarly, we calculate the number of teams that lost the game according to the input data and prediction. There are three of them. There are two type I errors and two type II errors. The True Positive Rate (TPR) is:

Confusion matrix		Initial class	
		+	−
Prediction	+	TP=5	FP=2
	−	FN=2	TN=3

$$\mathsf{TPR} = \frac{TP}{TP + FN} = \frac{5}{5+2} \approx 0.71,$$

and the False Positive Rate (FPR) is:

$$\mathsf{FPR} = \frac{FP}{TN + FP} = \frac{2}{3+2} = 0.4.$$

Moreover,

$$F_1 \approx 0.71,$$

which is good actually.

3.2 ROC Curve

Now that you've learned the concepts of accuracy, precision, recall, and F score, you can estimate the quality of the constructed algorithm given a fixed cut-off value. However, the choice of the cut-off value for the developed model

is another important problem. To solve it, we may use such an integral quality metric as *ROC* curve. It is constructed as the relationship between TPR and FPR. To construct it, we need to calculate the corresponding values given different cut-off values.

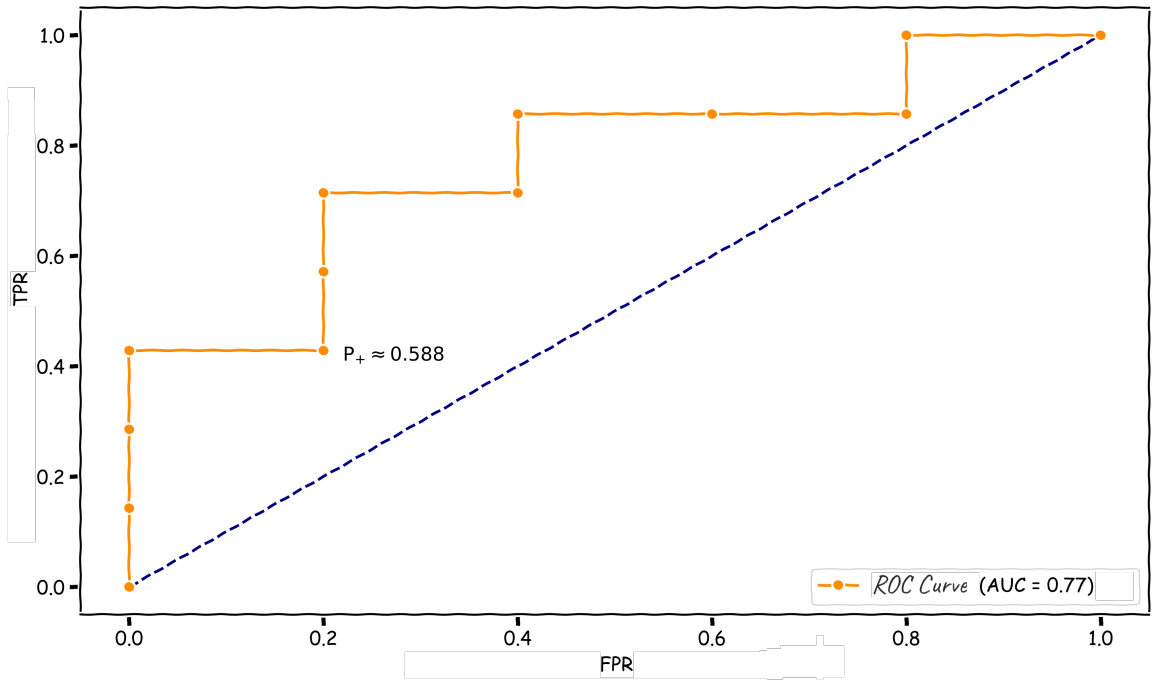

Figure 10: ROC curve.

There are different approaches to change cut-off values. A cut-off value can be raised in increments from zero to one, or its values can be the probabilities sorted in ascending order and obtained by the probability model.

In the latter case, each pair of values of $\mathsf{FPR}_i, \mathsf{TPR}_i$ corresponds to a point in the plane, and the points are connected by the straight-line segments. Since switching to the next probability value changes one classification result, we observe a rapid shift either up or to the right, and the curve becomes stepwise (Figure 10). For example, the probability value of 0.588 corresponds to the True Positive Rate of 0.2 and the False Positive Rate of 0.429. The points having the coordinates $(0, 0)$ and $(1, 1)$ are always marked as the start and end points of the curve.

In a perfect world, the curve passes through the upper left corner where the percentage of truly positive cases is 100%. Therefore, the more the ROC curve arches up, the more accurate the prediction of the model results is. We can obtain an estimator by calculating the area under the *ROC* curve. The parameter is also known as *AUC* (which stays for Area Under Curve), or $AUC - ROC$. We obtained a stepwise curve. Thus, its area can be easily calculated as the sum of the rectangle areas based on the values $\mathsf{TPR}_i, \mathsf{FPR}_i$.

Depending on the values of *AUC*, the model quality is often estimated as excellent if $AUC \in (0.9, 1]$, good if $AUC \in (0.7, 0.9]$, fair if $AUC \in (0.6, 0.7]$,

and poor if $AUC \in (0.5, 0.6]$.

How to find an optimal cut-off value? The intersection of sensitivity and specificity corresponds to the optimal cut-off value. The graph is created similarly to the *ROC* curve case. The only difference is that the relationship between sensitivity and the cut-off value and the relationship between specificity and the cut-off value are plotted in the same plane (Figure 11). In the example, this value approximately equals 0.52. Please note that the probabilities and other values are not rounded in practice.

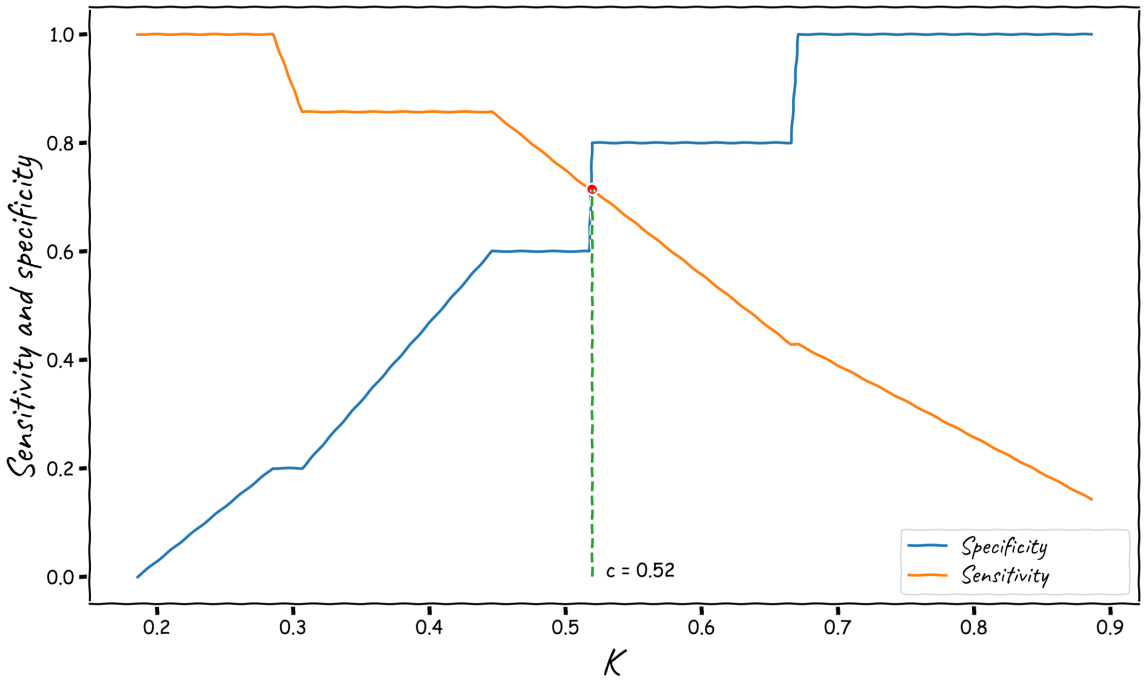

Figure 11: Determining the cut-off value.

The cut-off value slightly differs from the one considered earlier, and the amount of data is small, but the confusion matrix looks different:

Confusion matrix		Initial class	
		+	−
Prediction	+	TP=4	FP=1
	−	FN=3	TN=4

The cut-off value results in the True Positive Rate of 0.571 and the False Positive Rate of 0.2. It means that the model can better cut off negative data due to the increased specificity.

Thus, ROC analysis is aimed to select a cut-off value that allows the model to identify positive or negative outcomes with the greatest accuracy and minimize false positive or false negative errors.

4 Conclusion

This module introduced you to logistic regression, which is one of the probability algorithms of multinomial classification. We also studied different approaches used to estimate the quality of the constructed model. Well, that's all for today! See you soon!