Multiclass Prediction

SoftMax Regression, One-vs-All & One-vs-One for Multi-class Classification

In Multi-class classification, we classify data into multiple class labels. Unlike classification trees and nearest neighbors, the concept of Multi-class classification for linear classifiers is not as straightforward. We can convert logistic regression to Multi-class classification using multinomial logistic regression or SoftMax regression; this is a generalization of logistic regression. SoftMax regression will not work for Support Vector Machines (SVM); One vs. All (One-vs-Rest) and One vs One are two other multi-class classification techniques that can convert most two-class classifiers to a multi-class classifier.

SoftMax Regression

SoftMax regression is similar to logistic regression, the SoftMax function converts the actual distances i.e. dot products of x with each of the parameters θ_i for K classes in the range from 0 to K-1. This is converted to probabilities using the following formula.

$$softmax(x, i) = \frac{k^{-\theta_i^T x}}{\sum_{j=1}^{k} e^{-\theta_i^T x}} \text{ (1)}$$

The training procedure is almost identical to logistic regression using cross-entropy, but the prediction is different . Consider the three-class example where $y \in \{0,1,2\}$ i.e y can equal 0,1,2. We would like to classify x. We can use the SoftMax function to generate a probability of how likely the sample belongs to each class. We then make a prediction using the $argmax$ function:

$$\hat{y} = argmax_i(softmax((x, y)) \text{ (2)}$$

Let's do an example, consider sample $x1$, we will start by creating a table where each column will be the $i{-}th$ values of the SoftMax function. The index of each column is the same as the class.

probability of $\hat{y} = 0$	probability of $\hat{y} = 1$	probability of $\hat{y} = 2$
$softmax(x_1, 0)$	$softmax(x_1, 1)$	$softmax(x_1, 2)$
$i = 0$	$i = 1$	$i = 2$

Table 1. Each column will be the i-th values of the SoftMax function. The index of each column is the same as the class.

Let's add some real probabilities , this is the models estimate of how likely a sample belongs to each class.

0.97	0.02	0.01
$i = 0$	$i = 1$	$i = 2$

Table 2. Table of real probabilities. Each column will be the i-th values of the SoftMax function. The index of each column is the same as the class.

We can represent the probability as a vector [0.97,0.02,0.01]. To get the class we simply apply the $argmax$ function, this returns the index of the largest value.

$$\hat{y} = argmax_i([0.97,0.02,0.01])$$

$$\hat{y} = 0$$

Geometric Interpretation

Each $\theta_i^T x$ is the equation of a hyperplane, we plot the intersection of the three hyperplanes with 0 in fig1 as colored lines, in addition, we can overlay several training samples. We also shade the regions where the value of $\theta_i^T x$ is largest, this also corresponds to the largest probability. This color corresponds to where a sample x would be classified. For example if the input is in the blue region, the sample would be classified $\hat{y} = 0$, If the input is in the red region it would be classified as $\hat{y} = 1$, and in the yellow region $\hat{y} = 2$. We will use this convention going forward.
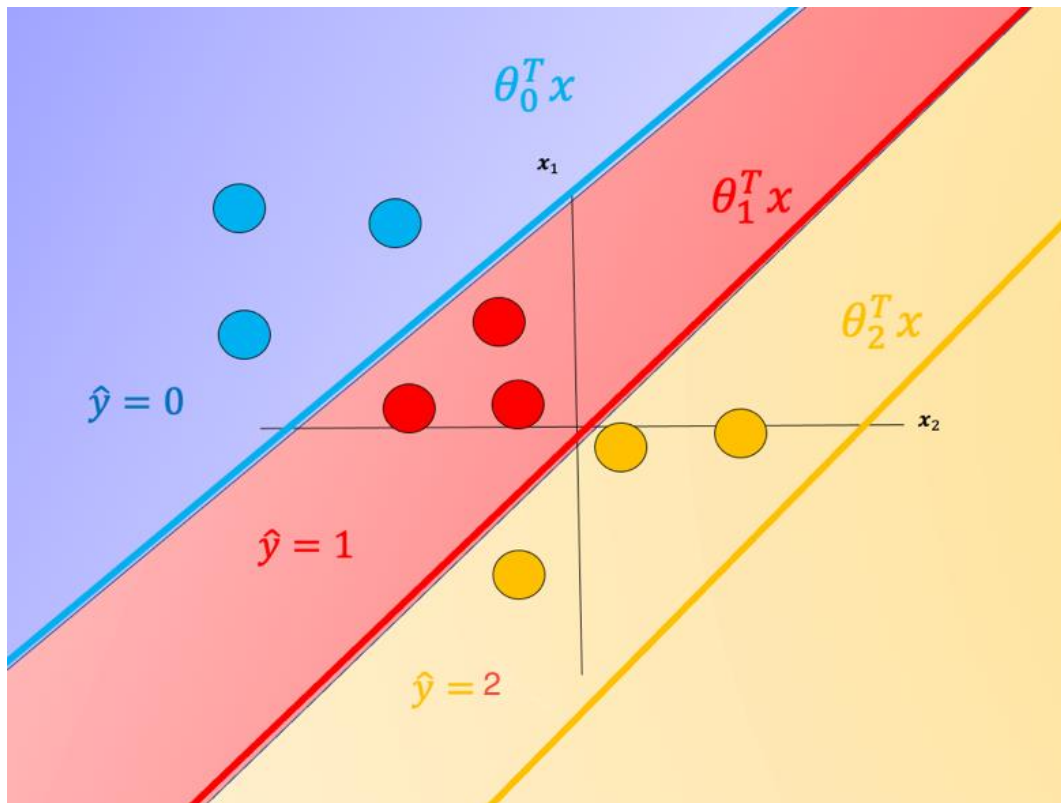

Fig 1. Equation of a hyperplane. We plot the intersection of the three hyperplanes with 0, in addition we can overlay several samples. We also shade the regions where the value of i is largest.

One problem with SoftMax regression with cross-entropy is it cannot be used for SVM and other types of two-class classifiers.

One vs. All (One-vs-Rest)

For one-vs-All classification, if we have K classes, we use K two-class classifier models, the number of class labels present in the dataset is equal to the number of generated classifiers. First, we create an artificial class we will call this "dummy" class. For each classifier, we split the data into two classes. We take the class samples we would like to classify; the rest of the samples will be labelled as a dummy class. We repeat the process for each class. To make a classification, we can use majority vote or use the classifier with the highest probability, disregarding the probabilities generated for the dummy class.

Although classifiers such as logistic regression and SVM class values are $\{0,1\}$ and $\{-1,1\}$ respectively we will use arbitrary class values. Consider the following samples colored according to class $y=0$ for blue, $y=1$ for red, and $y=2$ for yellow:
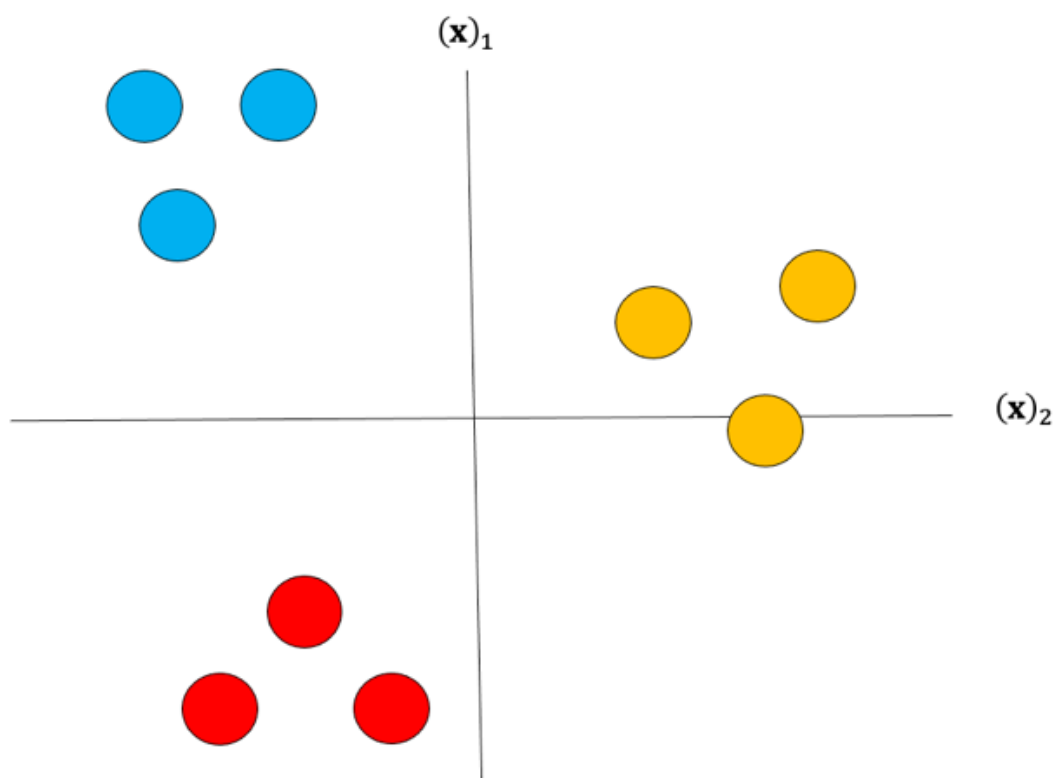

Fig 2. Samples colored according to class.

For each class we take the class samples we would like to classify, and the rest will be labeled as a "dummy" class. For example, to build a classifier for the blue class we simply assign all other labels that are not in the blue class to the Dummy class, we then train the classifier accordingly. The result is shown in fig 3 where the classifier predicts blue $\hat{y} = 0$ and in the purple region where we have our "dummy class" $\hat{y} = dummy$.
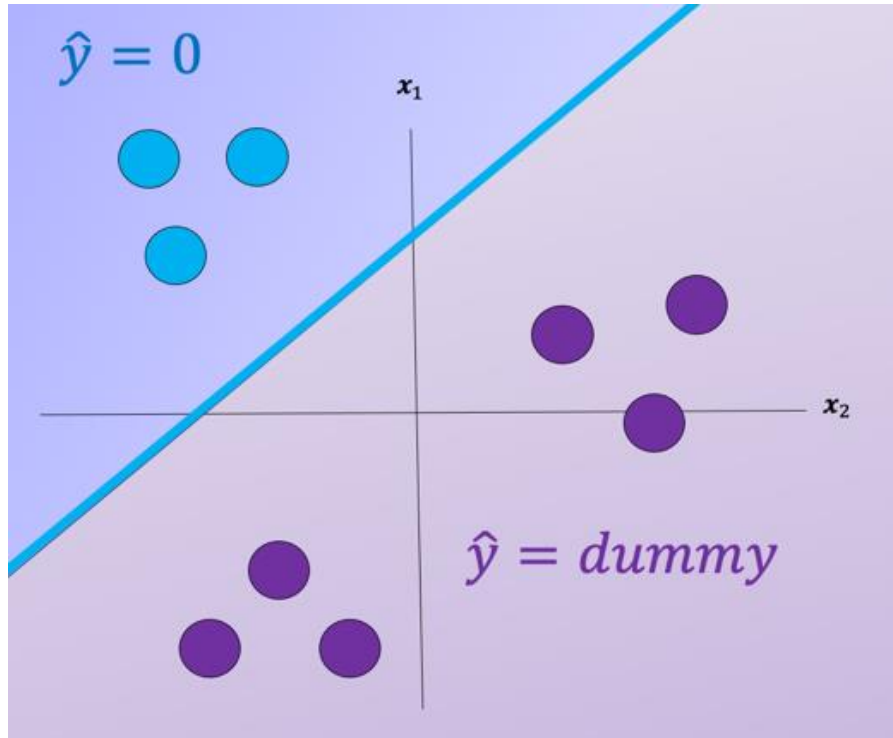
Fig 3. The classifier predicts blue $\hat{y} = 0$ in blue region and dummy class $\hat{y} = dummy$ in purple region.

We repeat the process for each class as shown in Fig 4, the actual class is shown with the same color and the corresponding dummy class is shown in purple. The color of the space is the actual classifier predictions shown in the same manner as above.
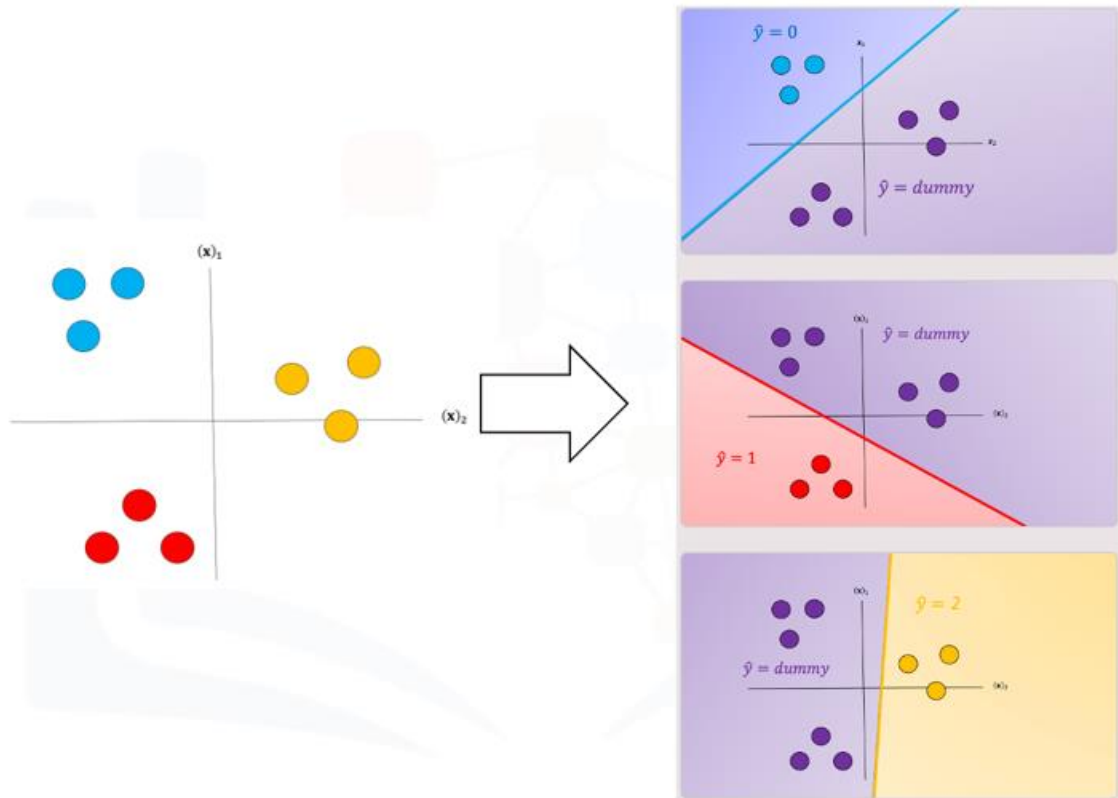

Fig 4. The classifier predicts $\hat{y} = 0, 1, 2$ in blue, red, and yellow region and dummy class $\hat{y} = dummy$ in purple region.

For a sample in the blue region, we would get the following output shown in table 3. You would disregard the dummy classes and select the output $\hat{y}_0 = 0$ in yellow where the subscript is the classifier number.

Classifier 0	Classifier 1	Classifier 2
$\hat{y} = 0$	$\hat{y} = dummy$	$\hat{y} = dummy$

Table 3. Example classification output, 2 of the 3 outputs are dummy; these classifiers would be ignored and the class would be zero.

One issue with one vs all is the ambiguous regions as shown in Fig 5 in purple. In these regions you may get multiple classes for example $\hat{y}_0 = 0$ and $\hat{y}_1 = 1$ or all the outputs will equal "dummy."
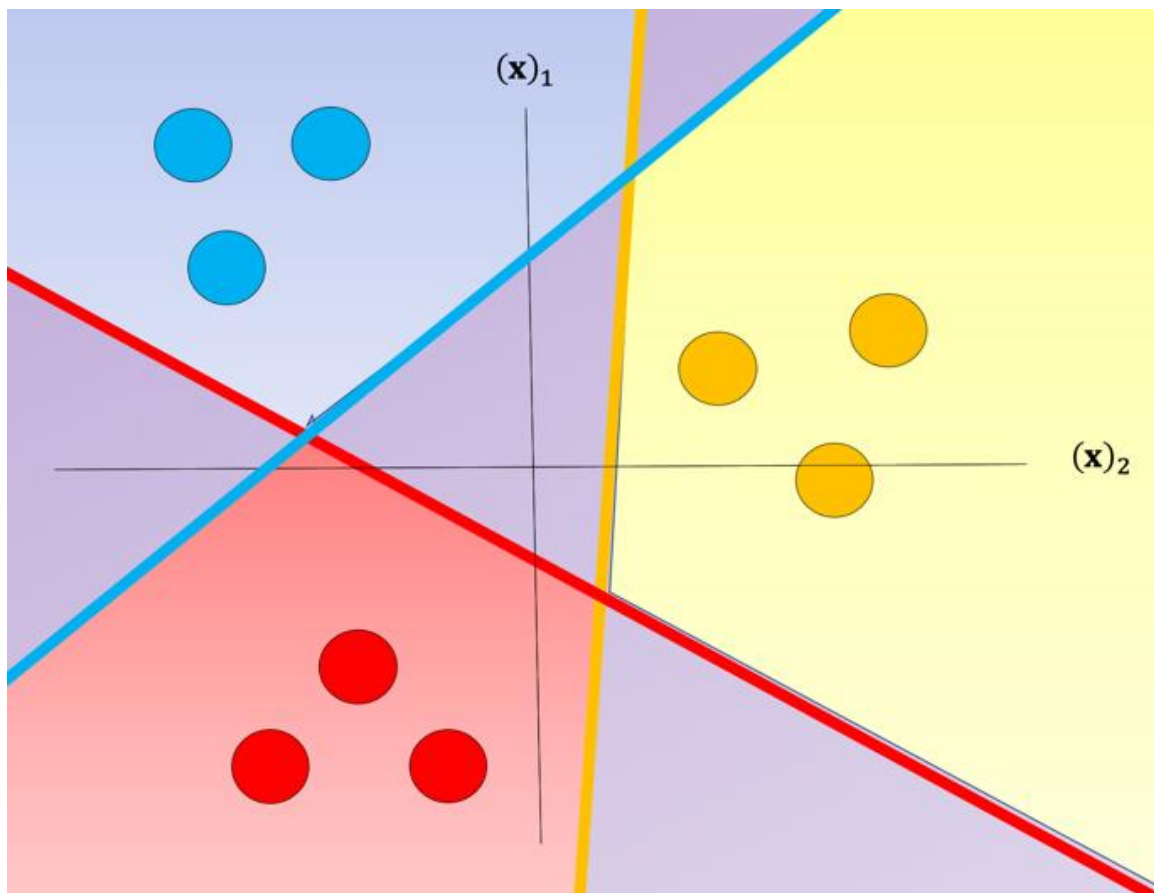

Fig 5. The classifier predicts all outputs $\hat{y}_0$, $\hat{y}_1$, $\hat{y}_2$ will equal "dummy."

There are several ways to reduce this ambiguous region, you can use the output based on the output of the linear function this is called the fusion rule. We can also use the probability of a sample belonging to the actual class as shown in Fig 6, where we select the class with the largest probability in this case $\hat{y}_0 = 0$; we disregard the dummy values. These probabilities are scores, as the probabilities are between the dummy class and the actual class not between classes.

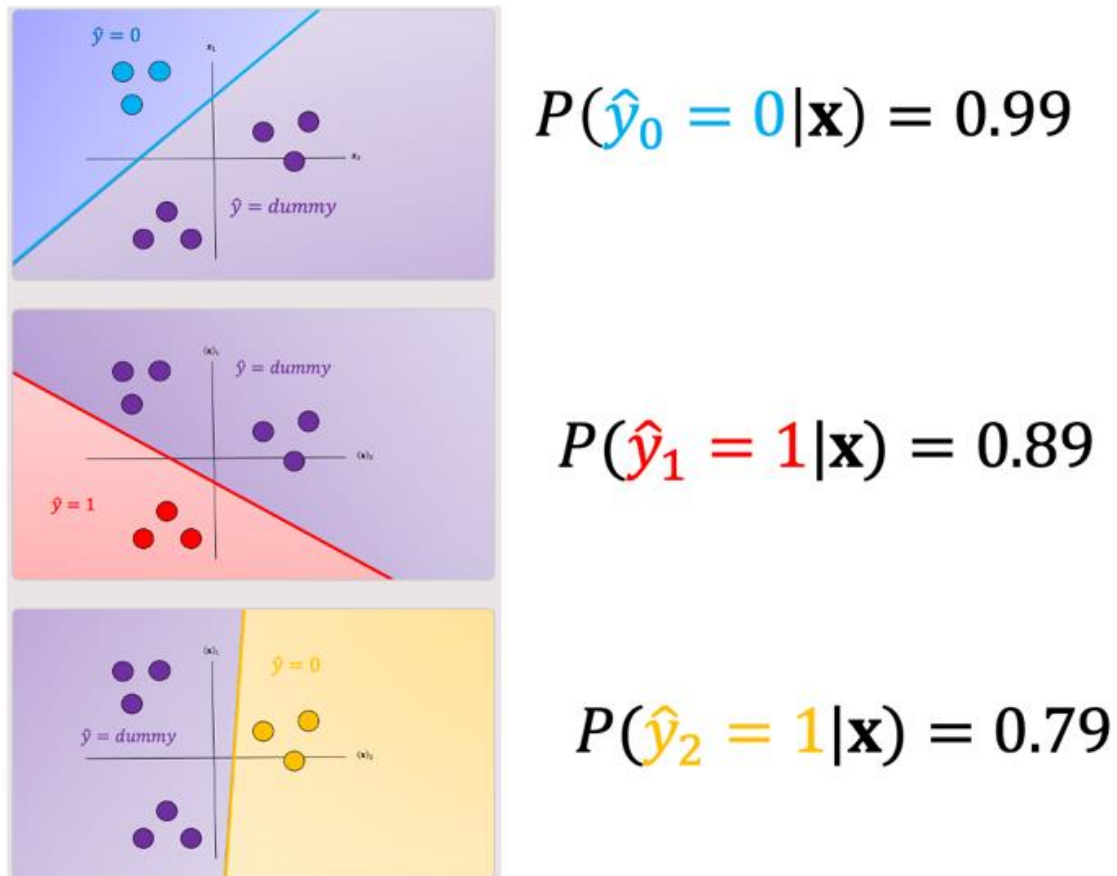Just a note packages like Scikit-learn can output probabilities for SVM.

$$P(\hat{y}_0 = 0|\mathbf{x}) = 0.99$$

$$P(\hat{y}_1 = 1|\mathbf{x}) = 0.89$$

$$P(\hat{y}_2 = 1|\mathbf{x}) = 0.79$$

Fig 6. Probability of a sample belonging to the actual class compared to dummy variable, selects the class with the highest probability.

One-vs-One classification

In One-vs-One classification, we split up the data into each class; we then train a two-class classifier on each pair of classes. For example, if we have class 0,1, and 2, we would train one classifier on the samples that are class 0 and class 1, a second classifier on samples that are of class 0 and class 2, and a final classifier on samples of class 1 and class 2. Fig 7 is an example of class 0 vs class 1, where we drop training samples of class 2. Using the same convention as above where the color of the training samples are based on their class. The separating plane of the classifier is in black. The color represents the output of the classifier for that particular point in space.
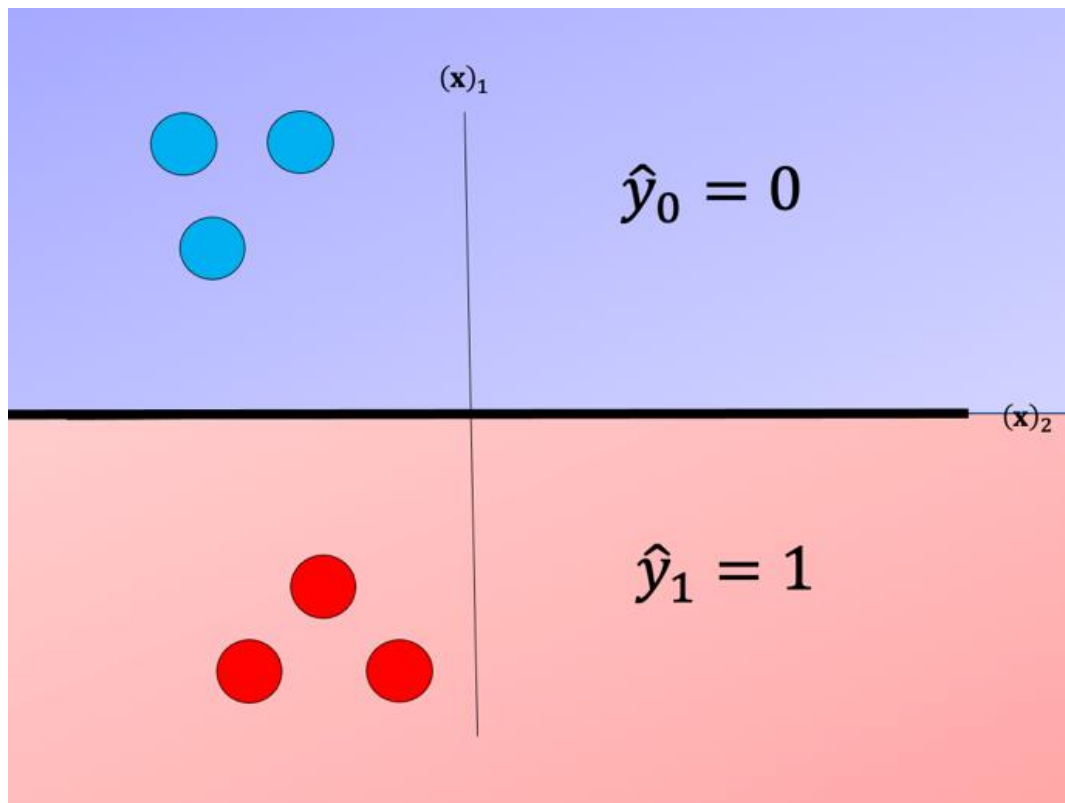
Fig 7. Probability of a sample belonging to the actual class compared to dummy variable, select the class with the highest probability.

We repeat the process for each pair of classes, in Fig 8. For K classes, we have to train $K(K-1)/2$ classifiers. So if $K=3$, we have $(3\times2)/2=3$ classes.
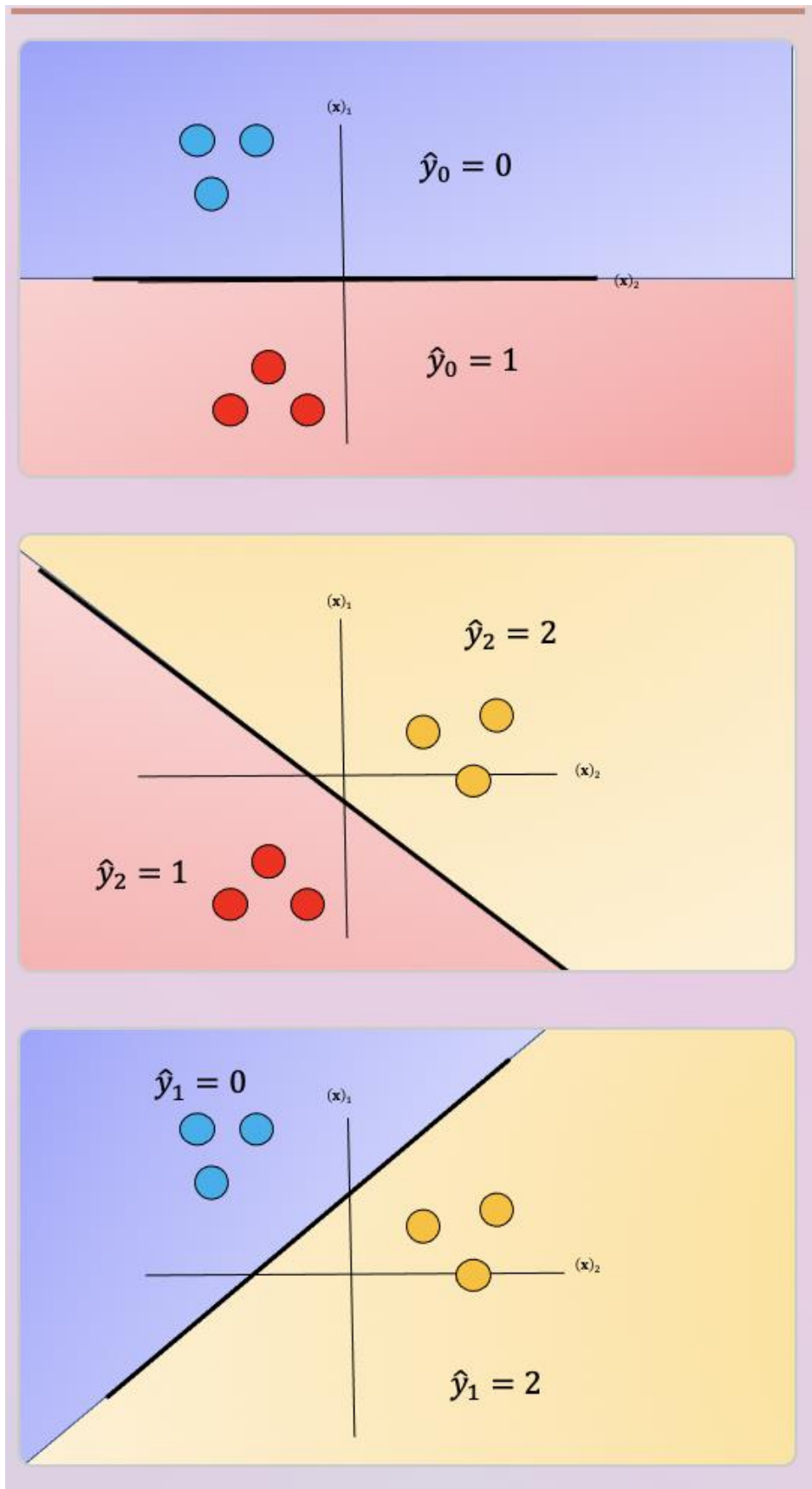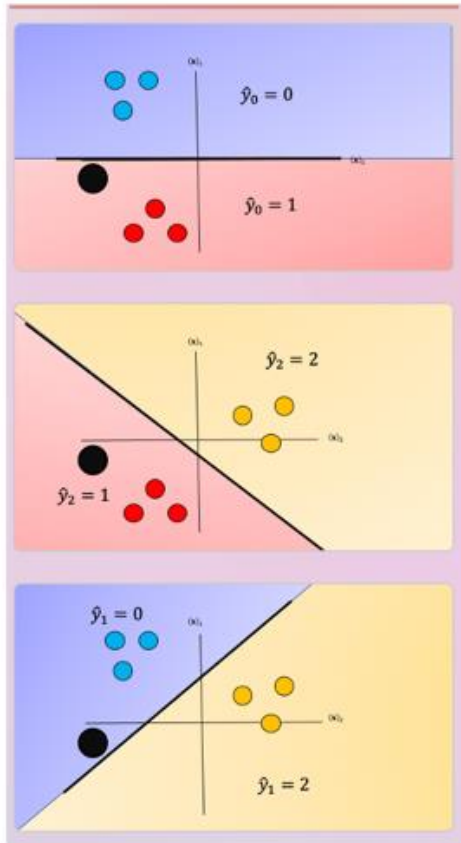
Fig 8. Probability of a sample belonging to the actual class compared to dummy variable, select the class with the highest probability.

To perform Classification on a sample, we perform a majority vote where we select the class with the most predictions. This is shown in Fig 9 where the black point represents a new sample and the output of each classifier is shown in the table. In this case, the final output is one as selected by two of the three classifiers. There is also an ambiguous region but it's smaller, we can also use similar schemes as in One vs all like the fusion rule or using the probability. Check out the labs for more.

$\hat{y}_0$	$\hat{y}_1$	$\hat{y}_2$
1	1	0