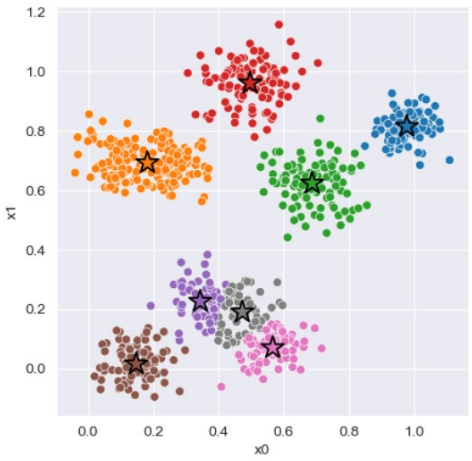
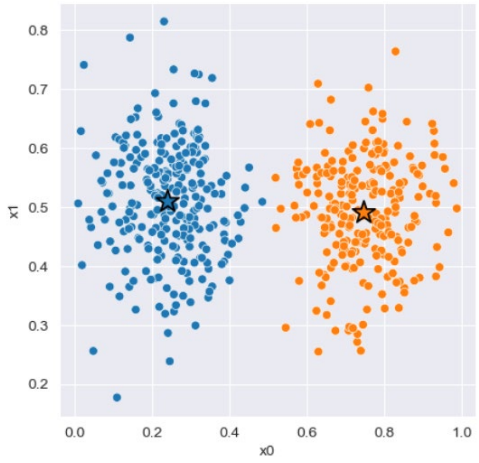
# REPORT

## K-MEANS:

K-means is a method of unsupervised learning that's aims to partition in K cluster in a system with *n* observation. Every observation is putted into a cluster depending on the mean distance from the centroids. The centroids are points into the range of observations that aims to be the center of a cluster.




The result of the first dataset reaches the silhouette and distortion expectation without any preprocessing.

```
Silhouette Score: 0.672
Distortion: 8.837
```

The result of the second dataset needs to be preprocessed, because it's possible to see that the column *x0* has a scale of difference with the column *x1.* To reach the expected result I used this type of preprocessing, for scale-up the data.

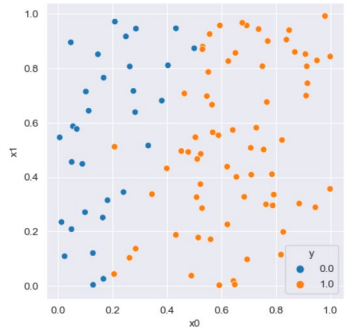
```
Silhouette Score: 0.623
Distortion: 6.663
```

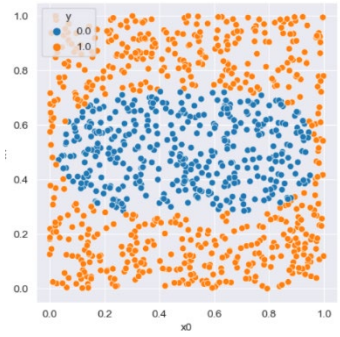
$$X = \frac{X}{[100,10]}$$

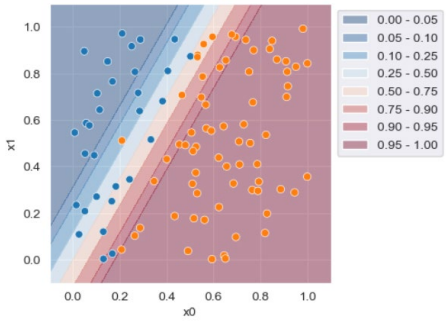
It was necessary to do because the axis *x1* gives more contribution on the calculation of the Euclidean distance, so it will not respect the inductive bias of the feature space weight.

# Logistic Regression:

Logistic regression is a machine learning method used for classification problems. This method estimates the probability of an event occurring. The dependent variable is bounded between 0 and 1. This is made thanks to sigmoid function that made the logistic.



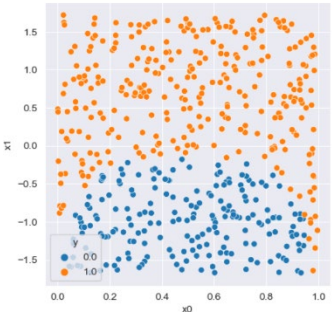

The inductive bias is the linearity between the features and the log-odds and the Independence of errors, that the errors between predictions and the true labels.

For the second dataset I've done a preprocessing to mirror the dataset

filter_data = np.abs(data_2_train['x1'] - 0.5)

data_2_train['x1'] = (filter_data - np.mean(filter_data)) /np.std(filter_data)

This was made because we need a linear feature, but in this case the were not. So, I used this formula to split the two types of data and mirroring them.




In this way I reached a linear dataset so the algorithm should work without any problems and reach the expected value.

This type of preprocessing brings up the result for the second dataset:

```
Train
Accuracy: 0.916
Cross Entropy:  0.213

Test
Accuracy: 0.902
Cross Entropy:  0.292
```

```
Accuracy: 0.910
Cross Entropy: 0.184
```

This result is close to the result achieved in the first dataset.