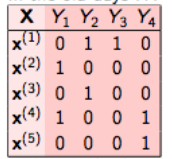
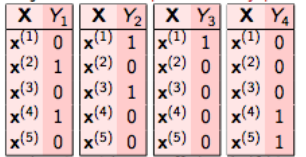
**Binary Relevance**

This is the most basic strategy, in which each label is treated as a distinct single-class classification issue.

Let's look at an example, as given below. We have the following data set, where X is the independent characteristic and Y is the target variable.

This problem is divided into four separate single class classification problems in binary relevance

* BinaryRelevance(GaussianNB()),
* BinaryRelevance(KNeighborsClassifier()),
* BinaryRelevance(RandomForestClassifier(random\_state=42)),

**Classifier Chains**

The initial classifier is trained only on the input data, and each subsequent classifier is trained on the input space as well as all previous classifiers in the chain.



* ClassifierChain (GaussianNB()),
* ClassifierChain (KNeighborsClassifier()),
* ClassifierChain (RandomForestClassifier(random\_state=42)),

**OneVsRestClassifiers**

This technique, often known as one-vs-all, consists of fitting one classifier per class. The class is fitted against all the other classes for each classifier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y1 | Y2 | Y3 | Y4 |
| x1 | 0 | 1 | 1 | 1 |
| x2 | 0 | 0 | 1 | 1 |
| x3 | 1 | 0 | 1 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y1 | Y2 | Y3 | Y4 |
| x1 | 0 | 1 | 1 | 1 |
| x2 | 0 | 0 | 1 | 1 |
| x3 | 1 | 0 | 1 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y1 | Y2 | Y3 | Y4 |
| x1 | 0 | 1 | 1 | 1 |
| x2 | 0 | 0 | 1 | 1 |
| x3 | 1 | 0 | 1 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y1 | Y2 | Y3 | Y4 |
| x1 | 0 | 1 | 1 | 1 |
| x2 | 0 | 0 | 1 | 1 |
| x3 | 1 | 0 | 1 | 0 |

* OneVsRestClassifiers (GaussianNB()),
* OneVsRestClassifiers (KNeighborsClassifier()),
* OneVsRestClassifiers (RandomForestClassifier(random\_state=42)),
* OneVsRestClassifier(SVC())