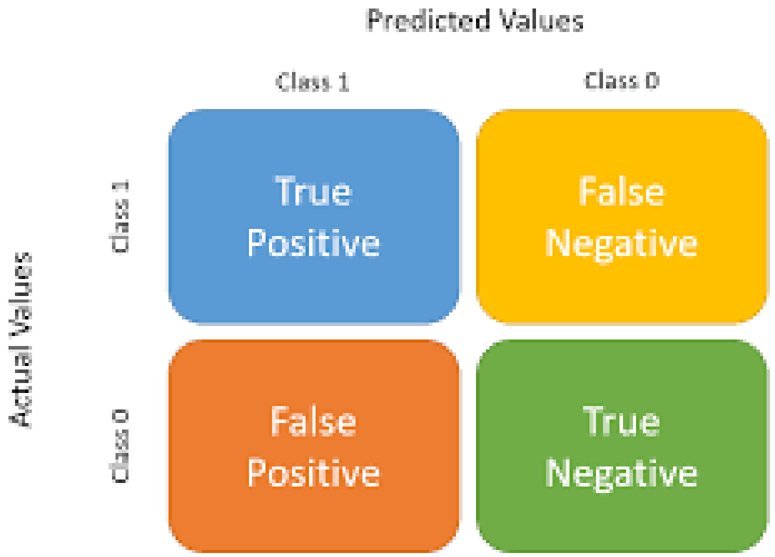
ROC and Precision-recall curve

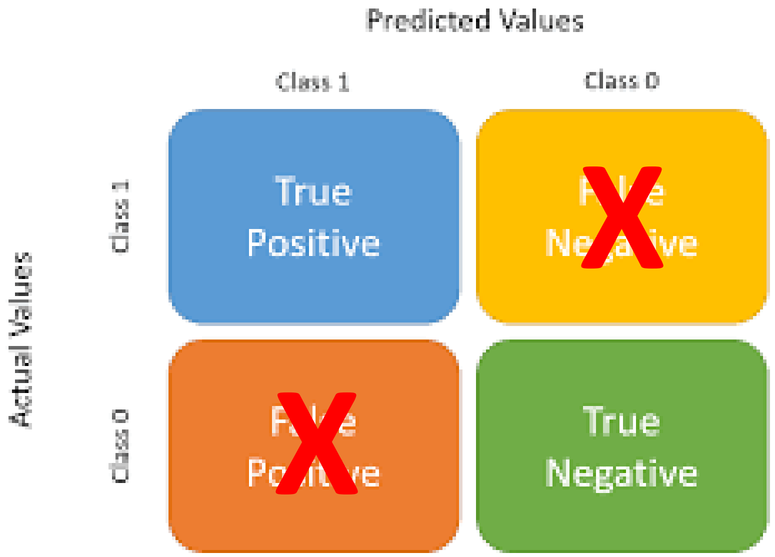
Classification

Meunier Jean Christophe
February the 10th 2021, Becode AI/Data science bootcamp

Confusion matrix



Confusion matrix



Receiver operating characteristic (ROC)

		Predicted value	
		A	B
Actual value	A	45	15
	B	13	27

TPR = TP amongst all REal positive (TP+FN)

Or

Proportion of ‘sicks’ correctly labelled

$$TPR = \frac{TP}{TP + FN} = \frac{45}{45 + 15} = 0{,}75$$

$$FPR = \frac{FP}{FP + TN} = \frac{13}{13 + 27} = 0{,}33$$

FPR = FP amongst all real negative (FP+TN)

Or

Proportion of ‘non-sicks’ incorrectly labelled as sick

Threshold

Set by default at 0,5 in binary hard classification

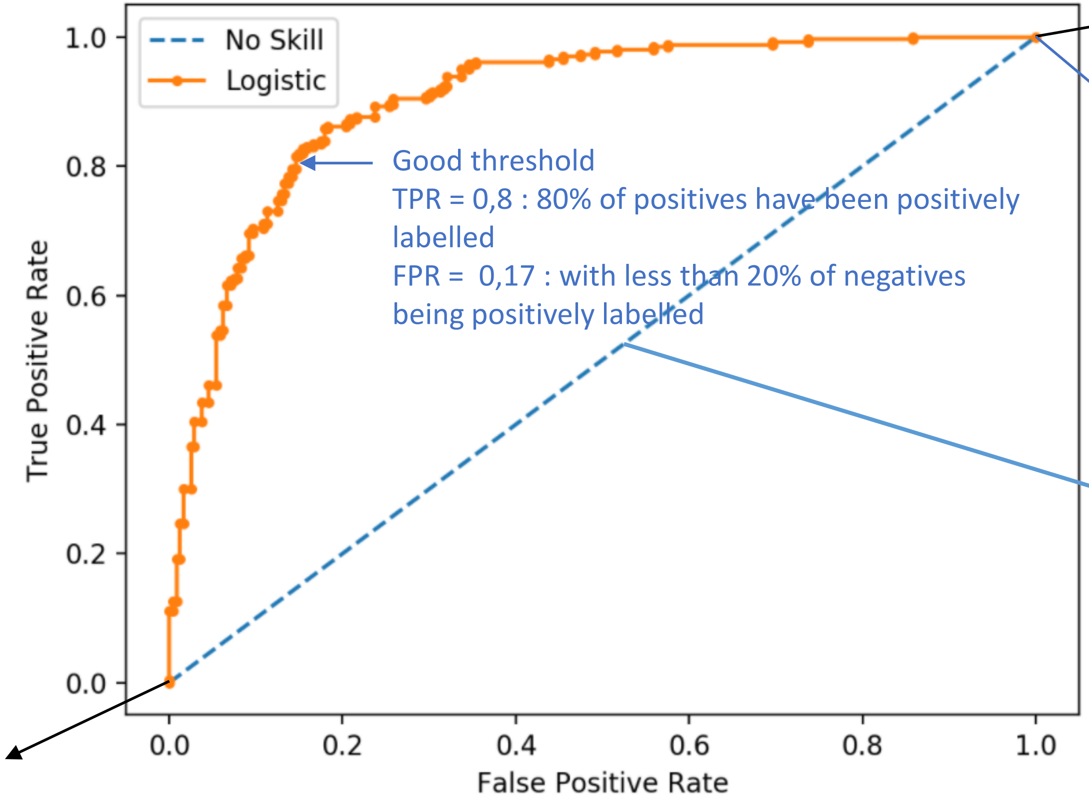
Receiver operating characteristic (ROC)

Threshold = 0,0

Every observation is labelled as positive

TPR = 1,0 : all positive have been positively labelled

FPR = 1,0 : but all negative have also been positively labelled



Good threshold

TPR = 0,8 : 80% of positives have been positively labelled

FPR = 0,17 : with less than 20% of negatives being positively labelled

ROC Curve of a Logistic Regression Model and a No Skill Classifier

Threshold = 1,0

No observation is labelled as positive

No skill's curve

Whatever the real status (sick or not), the chance of being labelled as 'sick' is the same

Rem : based on soft classification i.e. class probabilities

Precision-Recall

		Predicted value	
		A	B
Actual value	A	45	15
	B	13	27

REcall (=TPR) = TP amongst all REal positive (TP+FN)

$$Recall = \frac{TP}{TP + FN} = \frac{45}{45 + 15} = 0{,}75$$

$$Precision = \frac{TP}{TP + FP} = \frac{45}{45 + 13} = 0{,}78$$

PREcision = TP amongst all PREdicted positive (TP+FP)

Threshold

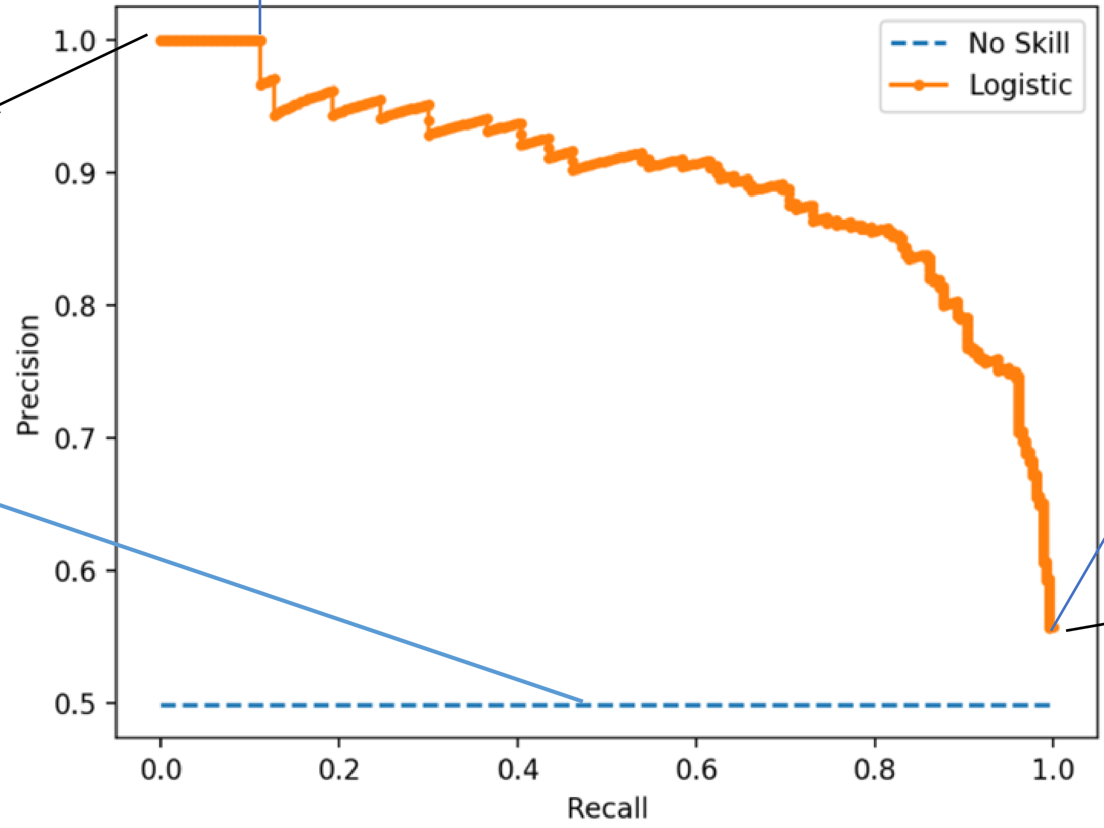
Set by default at 0,5 in binary hard classification

Precision-recall

Precision = 1,0 : all the few positive labels that have been done are TP
Recall = 0,18 : but many other real positive have been negatively labelled

Threshold = 1,0
No observation is labelled as positive

No skill's curve
Whatever the real status (sick or not), the chance of being labelled as 'sick' is the same

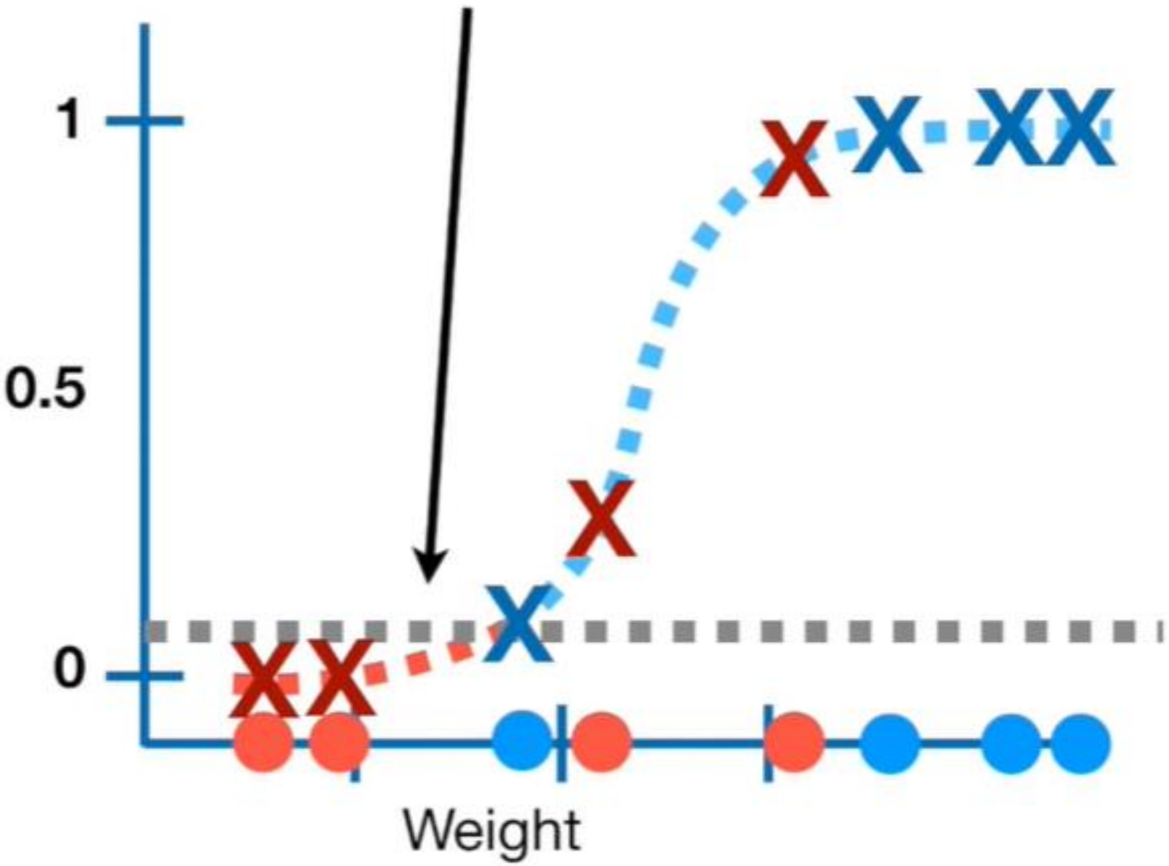


Precision = 0,65 : amongst the positive labels, 45% are FP
Recall = 1,0 : but real positive have been correctly labelled

Threshold = 0,0
Every observation is labelled as positive

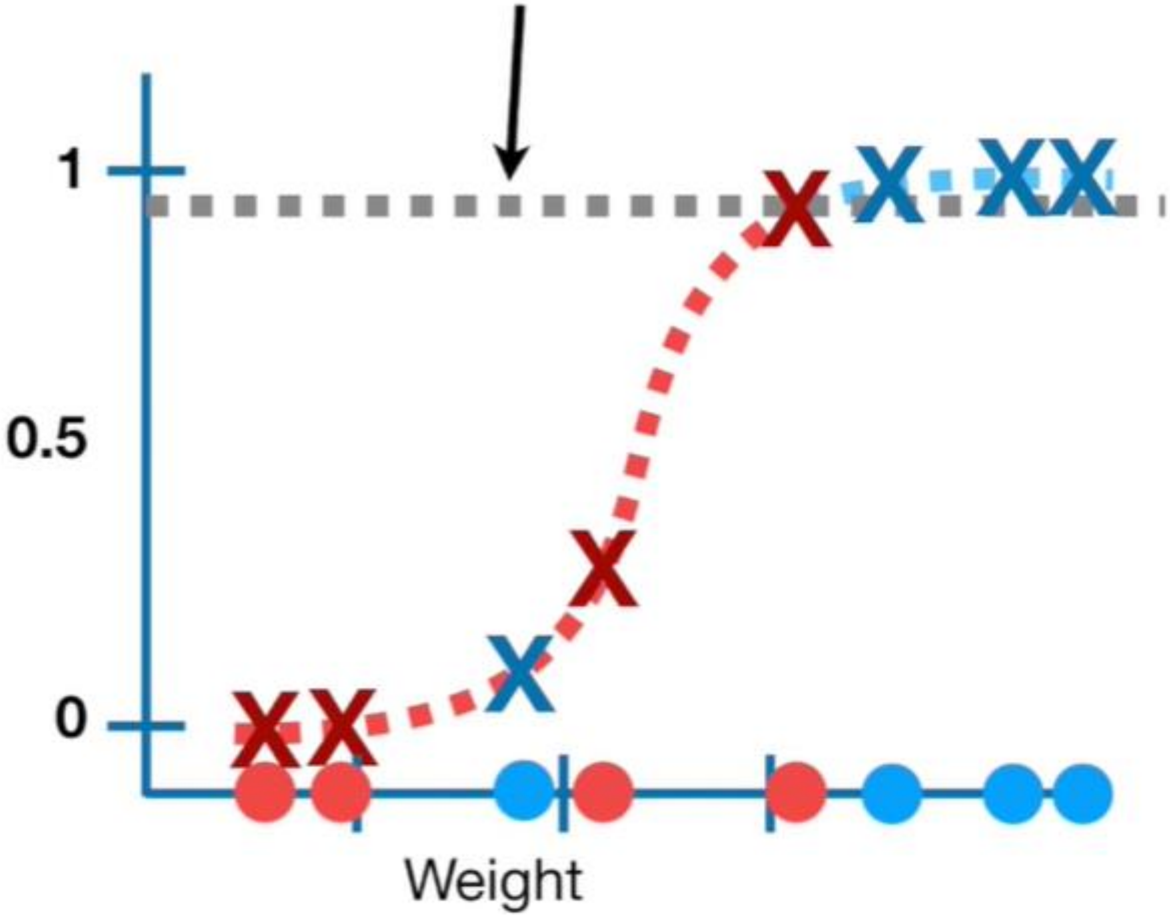
Precision-Recall Curve of a Logistic Regression Model and a No Skill Classifier

Index behavior by threshold (thr = 0,1)



TPR/ Recall = 1,00 : 100% of the real positive have been positively labelled
FPR = 0,50 : 50% of the negative have been positively labelled
Precision = 0,66 : amongst the positive labels, 66% are TP

Index behavior by threshold (thr = 0,9)



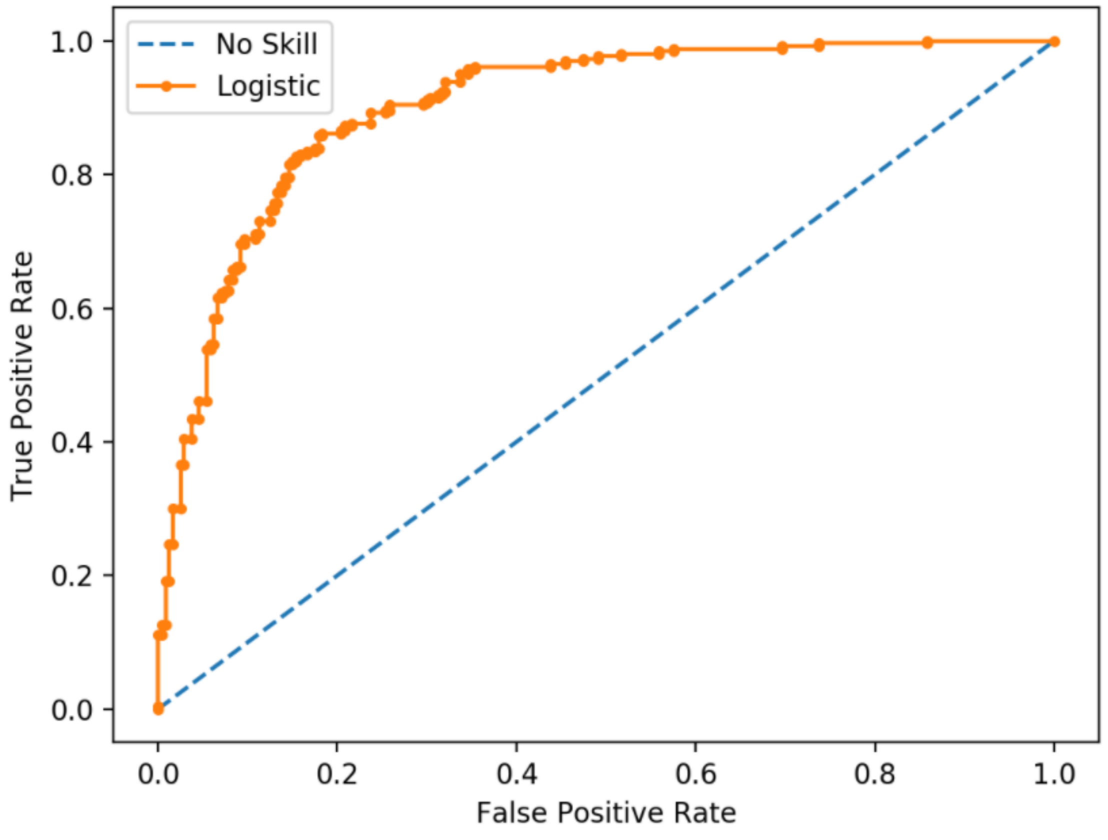
TPR/ Recall = 0,75 : 75% of the real positive have been positively labelled
FPR = 0,20 : 20% of the negative have been positively labelled
Precision = 1,00 : amongst the positive labels, 100% are TP

Balanced vs. unbalanced classes

- ROC curves should be used when there are roughly equal numbers of observations for each class.
- Precision-Recall curves should be used when there is a moderate to large class imbalance.

Balanced classes

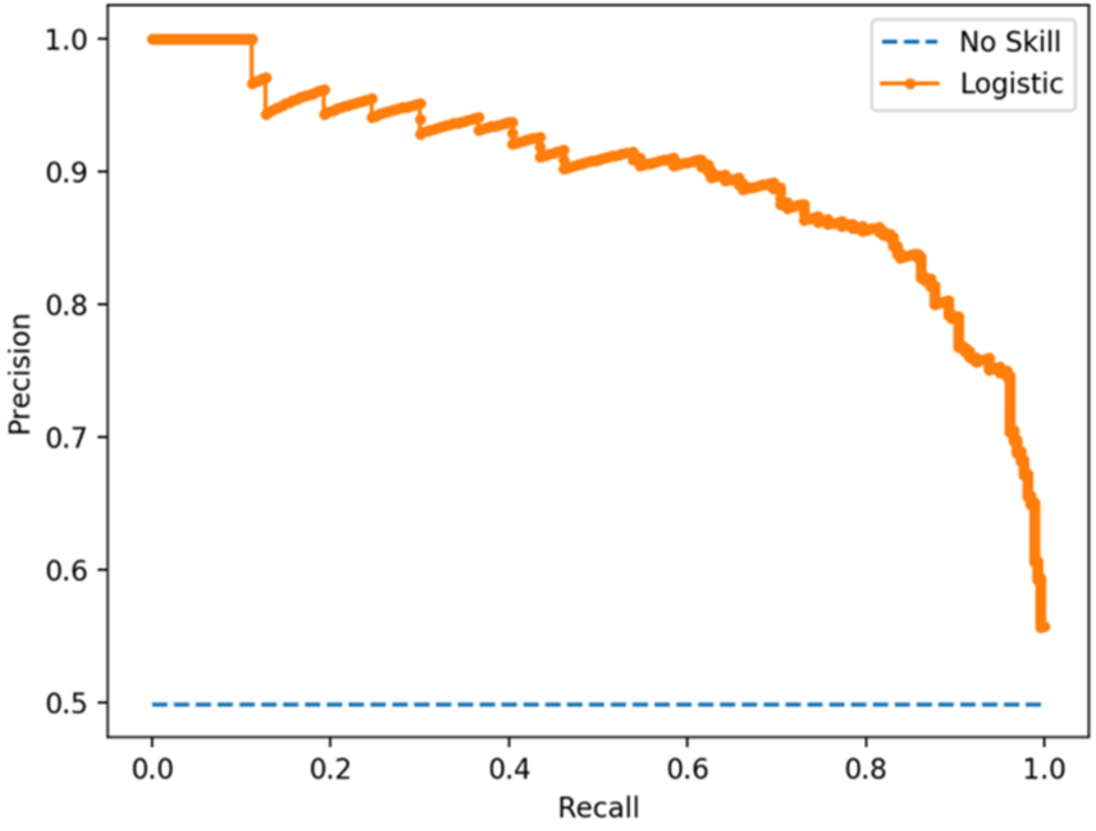
- ROC



ROC Curve of a Logistic Regression Model and a No Skill Classifier

Balanced classes

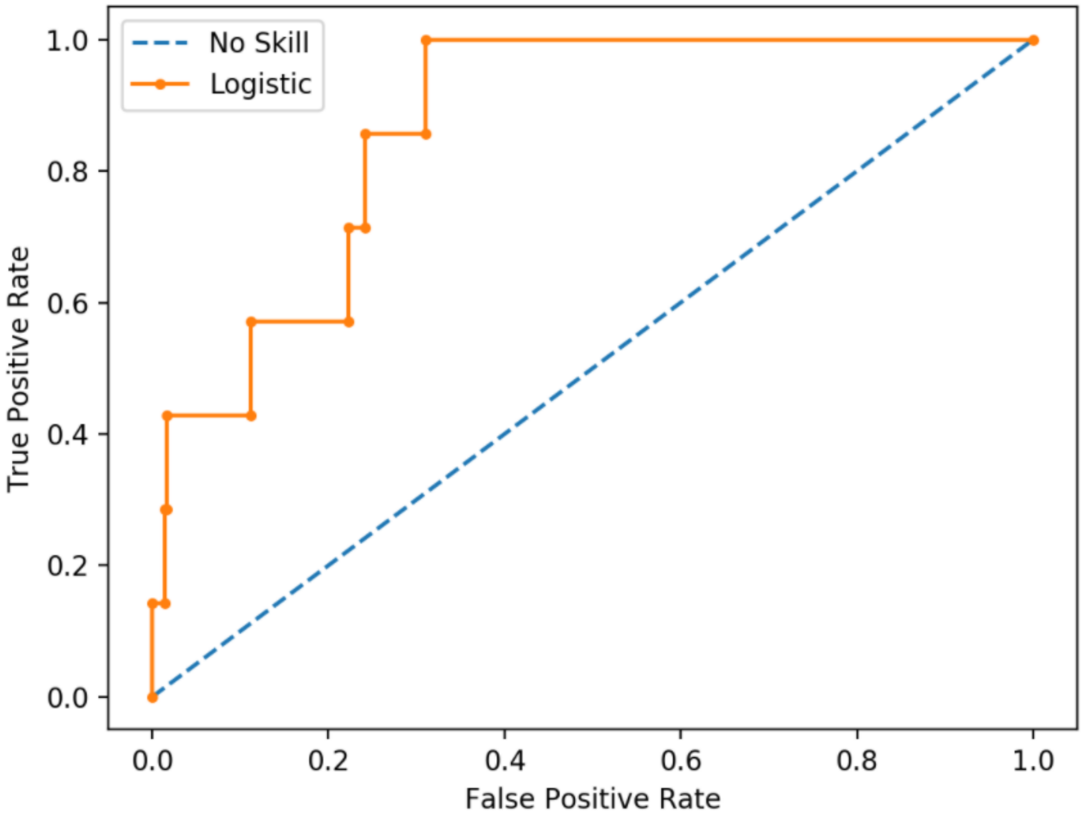
- Precision-recall



Precision-Recall Curve of a Logistic Regression Model and a No Skill Classifier

Unbalanced classes

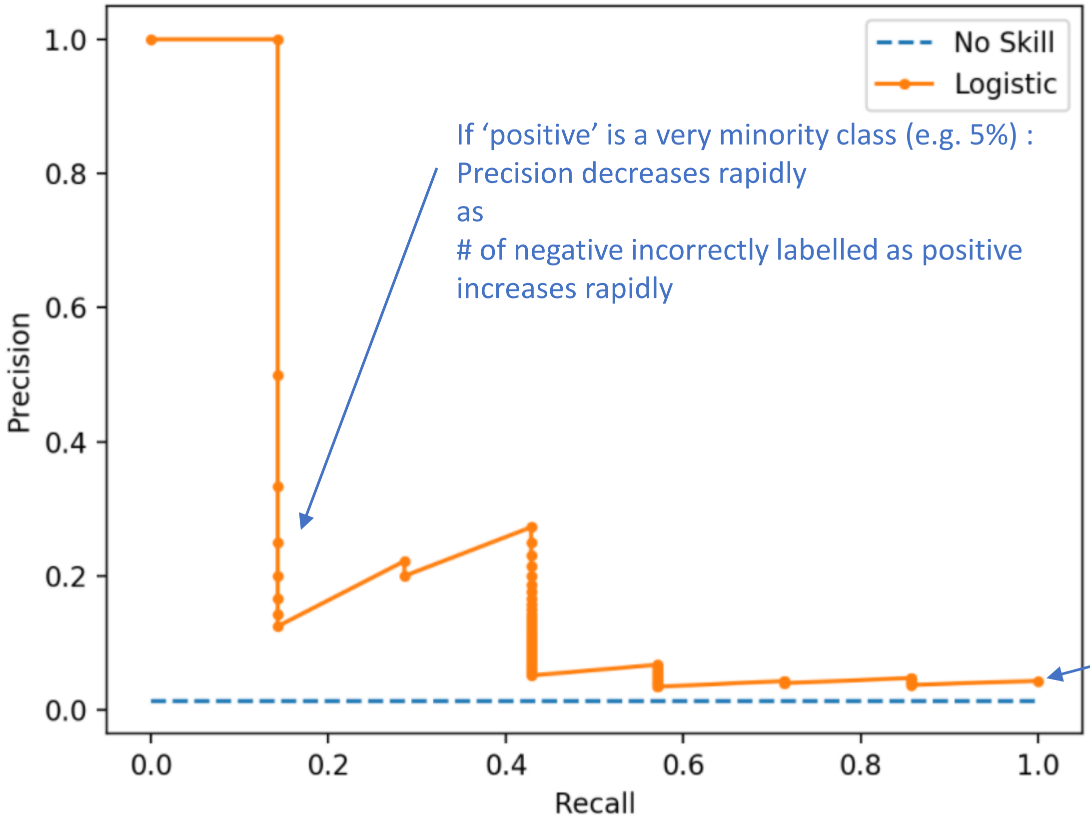
- ROC



Plot of ROC Curve for Logistic Regression on Imbalanced Classification Dataset

Unbalanced classes

- Precision-recall



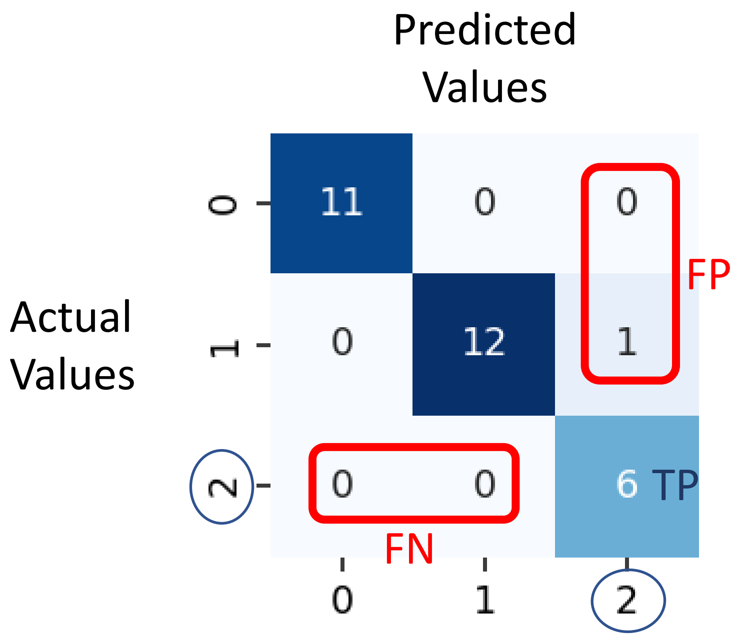
Here all real positive have been taken into account
But many many of the positive labels are in fact negative

Plot of Precision-Recall Curve for Logistic Regression on Imbalanced Classification Dataset

Multiclass classification

```python
from sklearn.metrics import confusion_matrix

conf_matrix = confusion_matrix(y_test, y_pred)
fig = plt.figure (figsize=(2,2))
sns.heatmap(conf_matrix, annot=True, cmap="Blues", cbar=False)
```




```python
print(classification_report(y_test, y_pred, target_names=iris_data.target_names))
```

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	11
versicolor	1.00	0.92	0.96	13
virginica	0.86	1.00	0.92	6
accuracy			0.97	30
macro avg	0.95	0.97	0.96	30
weighted avg	0.97	0.97	0.97	30

Moyenne simple

Moyenne pondérée par taille des cellules

Practical illustration

- See jupyter notebook : https://github.com/jcmeunier77/Data_visualization/blob/master/Visualizing%20ROC%20and%20precision-recall%20curve%20with%20thresholds.ipynb