Supervised Learning Classification

Agenda

- Logistic regression
- Assumptions of LR
- Model evaluation metrics
- Model Performance metrics
- Imbalanced data

Assumptions of Logistic Regression

Assumptions

We have seen that the logistic regression can be linearized, so it has assumptions almost same as that of linear regression

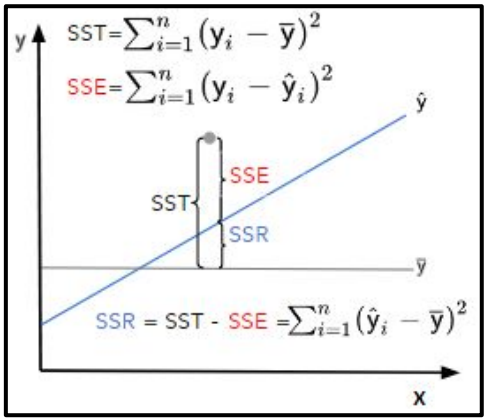
Assumption 1	Independence of error, whereby all sample group outcomes are separate from each other (i.e., there are no duplicate responses)
Assumption 2	Linearity in the logit for any continuous independent variables
Assumption 3	Absence of multicollinearity
Assumption 4	lack of strongly influential outliers

Significance of Coefficients

Significance of coefficients

- In a linear regression model, the significance of a regression coefficient is determined with the help of a t-test
- In a logistic regression, the significance of the coefficients is determined by the wald statistic and by the likelihood ratio test
- To test the significance of the model, the likelihood ratio test is used

For linear regression



Significance of coefficients - Wald test

- For β to be significant, $\beta > 0$.

$H_0 : \beta = 0$ against $H_1 : \beta \neq 0$

- It implies

H_0: The parameter β is not significant

against H_1: The parameter β is significant

- Failing to reject H_0 implies that the parameter β is not significant

Significance of coefficients - Wald test

- The Wald statistic is given by

$$Z_{wald} = \frac{\hat{\beta}}{SE(\hat{\beta})}$$

where $\hat{\beta}$ is the estimated value of β.

- The Wald statistic follows the N(0,1) distribution

- Decision Rule: Reject H_0 if $|Z| > Z_{\alpha/2}$ or if the p-value is less than the α (level of significance)

Significance of coefficients - LRT

- For β to be significant, $\beta > 0$

$H_0 : \beta = 0$ against $H_1 : \beta \neq 0$

- It implies

H_0: The parameter β is not significant

against H_1: The parameter β is significant

- Failing to reject H_0 implies that the parameter β is not significant

Significance of coefficients - LRT

- The likelihood ratio test is given by

$$D = -2\ln\left[\frac{\text{Likelihood of model without predictors}}{\text{Likelihood of model with predictors}}\right]$$

- D follows the chi-squared distribution with one degree of freedom, i.e. χ_1

- Decision Rule: Reject H_0 if $\chi \geq \chi_{\alpha/2}$ or if the p-value is less than the α (level of significance)

Significance of model - LRT

- The hypothesis for testing all coefficient in logistic regression can be extended from the previous test for one coefficient as follows

  $H_0: \beta_1 = \beta_2 = \beta_3 = 0$ against H_1: At least one $\beta_k \neq 0$ (k =1,2,3)

- It implies

  H_0: the logistic model is not significant

  Against H_1: the logistic model is significant

Significance of model - LRT

- The likelihood ratio test is given by

$$D = -2\ln\left[\frac{\text{Likelihood of model without predictors}}{\text{Likelihood of model with predictors}}\right]$$

- D follows the chi-squared distribution with one degree of freedom, i.e. χ_1

- Decision Rule: Reject H_0 if $\chi \geq \chi_{\alpha/2}$ or if the p-value is less than the α (level of significance)

Model Evaluation Metrics

Model evaluation metrics

The model evaluation metrics are

- Deviance
- AIC
- Pseudo R^2

Deviance

Terminologies

- Null model: A model without any predictors
- Saturated model: A model with exactly n samples (n predictors), that fits the data perfectly
- Full model: A model fitted with all the variables in the data
- Fitted model: A model with at least one predictor variable

Deviance

- Deviance is analogous to the sum of squares in the linear regression
- A measure of goodness of fit for a logistic regression
- Given by

$$D = -2\ln\left[\frac{\text{Likelihood of fitted model}}{\text{Likelihood of saturated model}}\right]$$

where saturated model is a model assumed to have the perfect fit

If the saturated model is not available use the fitted model.

Deviance

- Null deviance: The difference between the log likelihood of the null model and saturated model

- Model deviance: The difference between the log likelihood of the null model and fitted model

- Smaller values indicate a better fit

- To check for significance of *k* predictors, subtract the model deviance from the null deviance and access it on χ_k

Deviance

- In linear regression, we have R^2 defined as the ratio of explained variation to the total variation

$$R^2 = \frac{\text{Explained variation}}{\text{Total variation}} = \frac{\text{SSR}}{\text{SST}}$$

- In logistic regression we can consider the deviance similar to the R^2 in linear regression

- Deviance can be thought of as the R2 value such that the denominator is the total variation and the numerator is the variation explained by the fitted model

$$D = -2 \ln \left[\frac{\text{Likelihood of fitted model}}{\text{Likelihood of saturated model}} \right]$$

AIC

- The Akaike Information Criteria (AIC) is a relative measure of model evaluation for a given dataset

- It is given by:

$$AIC = -2 \ln L + 2K$$

L: log-likelihood
K: parameters to be estimated

- The AIC gives a trade-off between the model accuracy and model complexity, i.e. it prevents us from overfitting

Pseudo R^2

- The non-pseudo R^2 or the R^2 in the linear regression framework is the explained variability and the correlation (for simple linear regression)
- An equivalent R^2 statistic does not exist in the logistic regression since the parameters are estimated by the method of maximum likelihood
- However, there are various pseudo R^2s developed which are similar on scale, i.e. on [0,1], and work exactly same with higher values indicating the a better fit

Pseudo R^2

The pseudo R^2 are

- McFadden R^2
- Cox-Snell R^2
- Nagelkerke R^2

McFadden R^2

- It is defined as

$$R^2_{\text{McFadden}} = 1 - \frac{\ln \text{likelihood of full model}}{\ln \text{likelihood of null model}}$$

- If comparing two models on the same data, we consider the model which has higher value is considered to be better

- The pseudo R2 in the python output is the McFadden R^2

Cox-Snell R^2

- It is similar to the McFadden R^2 and is defined as

$$R^2_{\text{Cox-Snell}} = 1 - \left\{ \frac{\ln \text{likelihood of null model}}{\ln \text{likelihood of full model}} \right\}^{\frac{2}{N}}$$

- The likelihood is the product of probability N observations of the dataset. Thus the N^{th} square root of the provides an estimated of each target value

- The $R^2_{\text{Cox-Snell}}$ can be greater than 1

- For a model with likelihood 1, i.e it predictions are perfect, then the denominator becomes 1

Nagelkerke R^2

- It is based on Cox-Snell R^2, it scales the values so that the maximum is 1

$$R^2_{\text{Nagelkerke}} = \frac{1-\left\{\frac{\text{ln likelihood of null model}}{\text{ln likelihood of full model}}\right\}^{\frac{2}{N}}}{1-\{\text{ln likelihood of null model}\}^{\frac{2}{N}}}$$

- If the full model predicts the outcome perfectly, i.e it has likelihood = 1, then $R^2_{\text{Nagelkerke}} = 1$
- Similarly, if likelihood of null model is equal to that of full model

Model Performance Measures

Performance metrics

The following metrics can be used to evaluate the performance of classification models:

- Confusion matrix
- Cross entropy
- ROC

Performance metrics

We shall consider the fish data as described before

BKT kg/ha	Presence of fish
1085.33	0
1210	1
1780.62	1
52.4	0
200	0
2502.67	1
301.33	0
542	0
969.33	1
240.56	0
1640	0
247	0
999.99	0
1220.76	1
150.67	1
160	0
2816	1
760	1
1350	0
1370	1

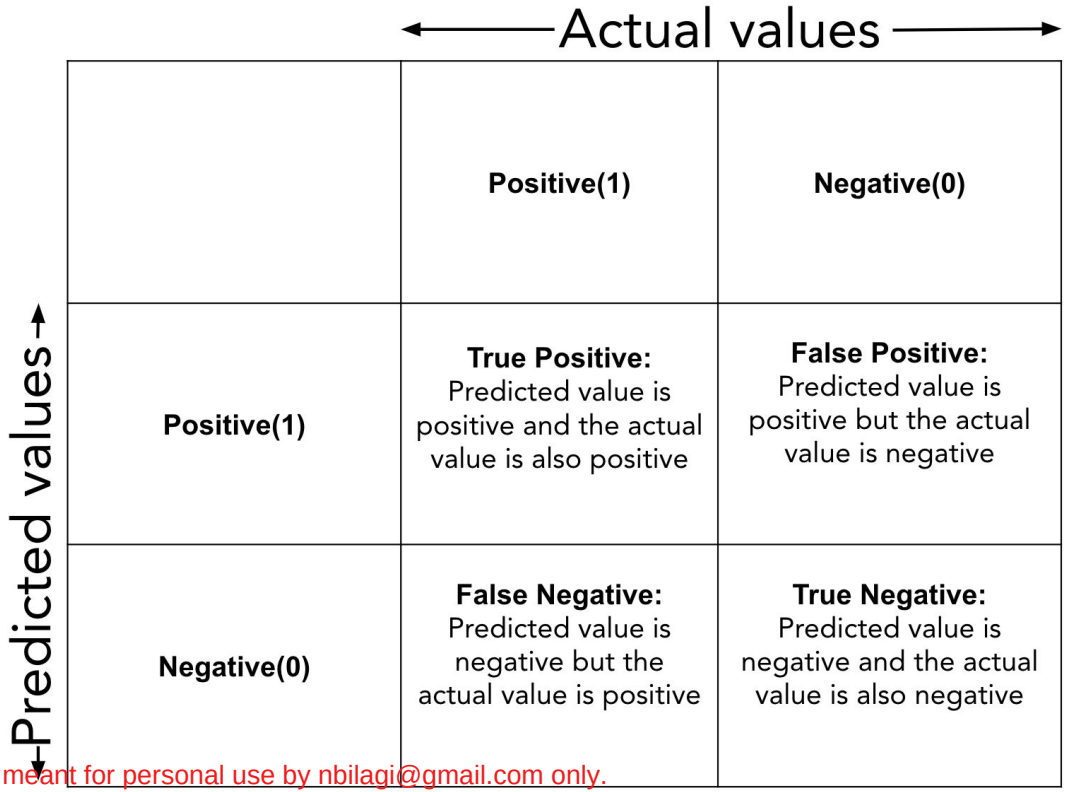
Performance metrics

The following metrics can be used to evaluate the performance of classification models:

- **Confusion matrix**
- Cross entropy
- ROC

Confusion matrix

- Performance measure for classification problem
- It is a table used to compare predicted and actual values of the target variable



Confusion matrix for our example

Confusion matrix for our considered example to predict presence of fish is given as:

Predicted values \ Actual values	Positive(1)	Negative(0)
Positive(1)	True Positive 8	False Positive 3
Negative(0)	False Negative 3	True Negative 6

Actual values

Predicted values

	Positive(1)	Negative(0)
Positive(1)	**True Positive:** Predicted value is positive and the actual value is also positive	**False Positive:** Predicted value is positive but the actual value is negative
Negative(0)	**False Negative:** Predicted value is negative but the actual value is positive	**True Negative:** Predicted value is negative and the actual value is also negative

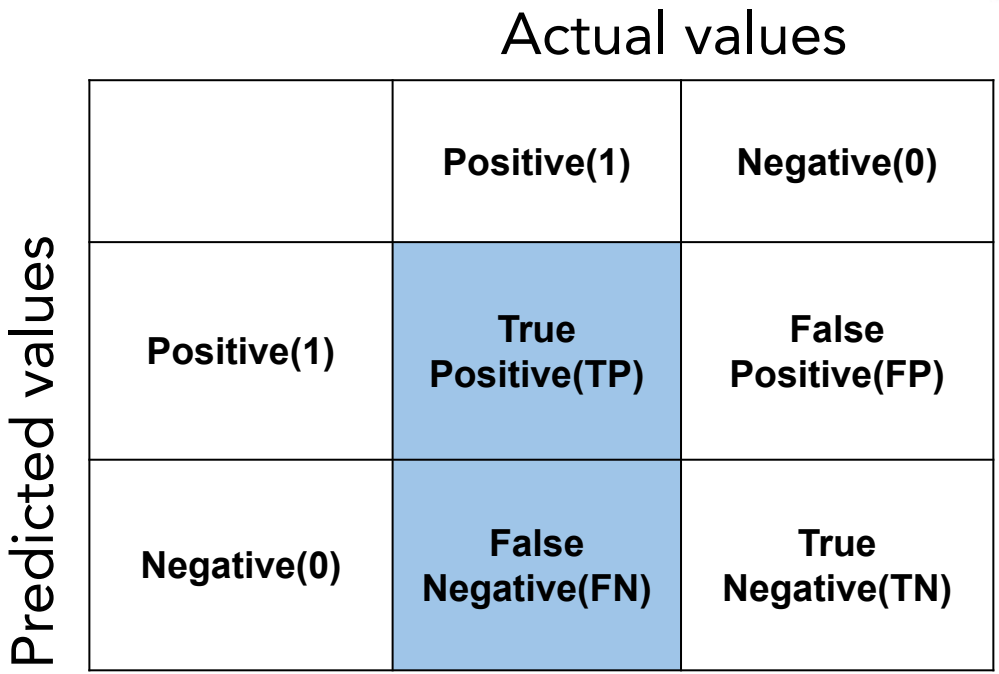
Performance evaluation metrics

Confusion matrix can be used to calculate the following evaluation metrics for a model:

- Accuracy
- Precision
- Recall
- False Positive Rate
- Specificity
- F_1 score
- Kappa

Actual values

Predicted values	Positive(1)	Negative(0)
Positive(1)	**True Positive** 8	**False Positive** 3
Negative(0)	**False Negative** 3	**True Negative** 6



Accuracy

Actual values

Predicted values	Positive(1)	Negative(0)
Positive(1)	True Positive(TP)	False Positive(FP)
Negative(0)	False Negative(FN)	True Negative(TN)

- Accuracy is the fraction of predictions that out model got correct

$$\text{Accuracy} = \frac{\text{number of correctly predicted records}}{\text{Total number of records}}$$

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

- Higher the value of accuracy better is the model

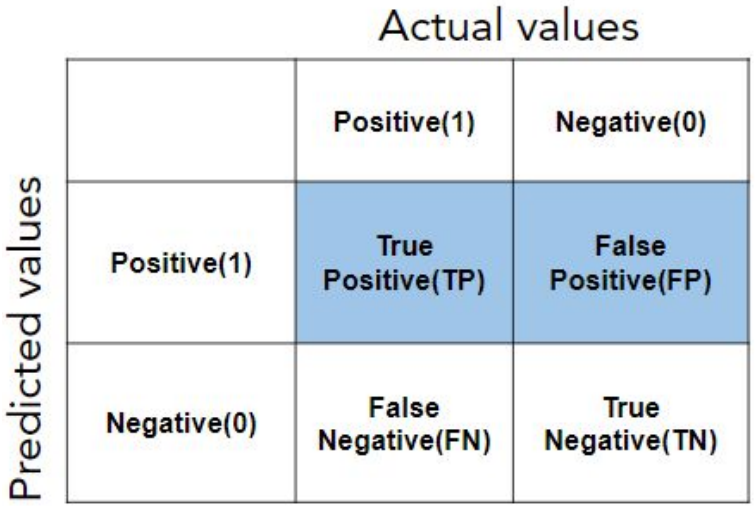
Accuracy is not always a reliable metric

Consider a dataset with information about 1000 patients. 960 of those patients have diabetes and only 40 do not have diabetes.We have a model ‘A’ that classifies every patient as diabetic.

$$\text{Accuracy of model A} = \frac{960}{1000} = 96\,\%$$

Even though the accuracy for model A is high it is not a good model.Since even when it will encounter information about a new patient it will always predict that the patient is diabetic. This scenario when accuracy is not a reliable metric is called the accuracy paradox.

Precision

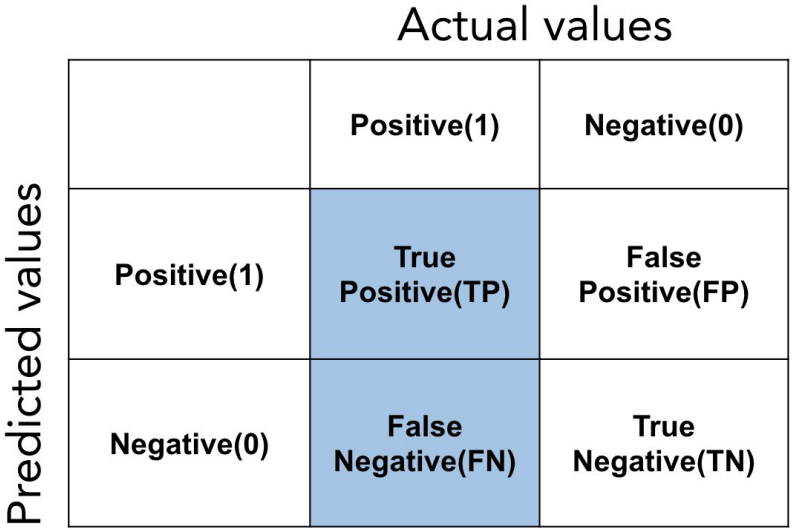


- Precision is proportion of positive cases that were correctly predicted

$$\text{Precision} = \frac{TP}{TP+FP}$$

- Higher is the precision better the model

Recall

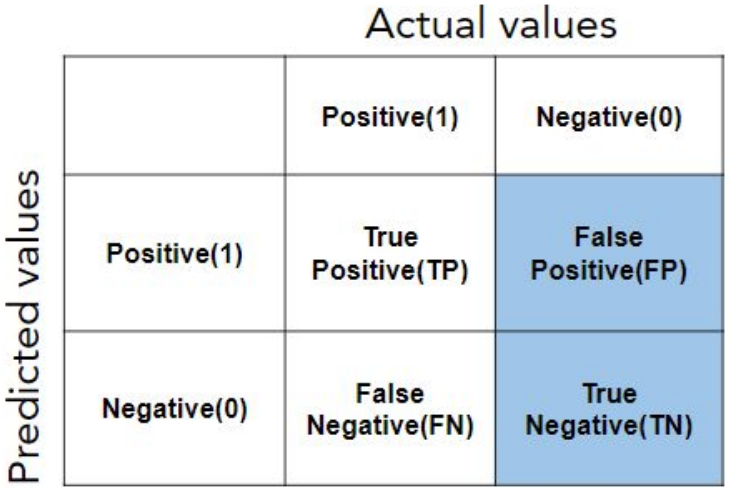


- Recall is the proportion of actual positive cases that were correctly predicted
- Recall is also sometimes called True Positive Rate (TPR) or Sensitivity

$$\text{Recall} = \frac{TP}{TP+FN}$$

- Higher value of TPR implies a better model

False positive rate



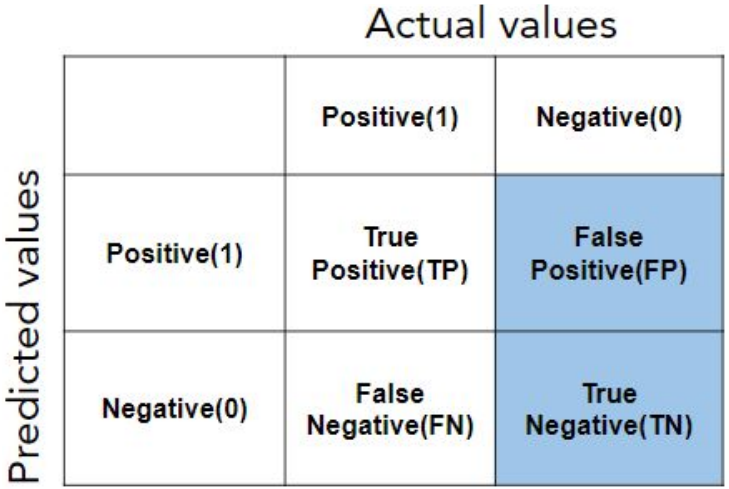
- False Positive Rate (FPR) is the proportion of actual negative cases that were predicted positive (incorrectly)

$$\text{FPR} = \frac{FP}{FP+TN}$$

$$\text{FPR} = 1 - \text{Specificity}$$

- Lower the value of FPR better is the model

Specificity



- Specificity is the proportion of actual negative cases that were correctly predicted

$$\text{Specificity} = \frac{TN}{TN+FP}$$

- Higher the specificity better is the model

F_1 score

- F_1 score is the harmonic mean of precision and recall values for a classification model
- It is good measure if we want to find a balance between precision and recall or if there is uneven distribution of classes (either positive or negative class has way more actual instances than the other)

$$F_1\,\text{score} = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2}\right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- Higher the F_1 score better the model

Performance metrics for our example

Performance Metric	Accuracy	Precision	Recall	Specificity	False positive rate	F_1 score
Formulae	$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$\frac{TN}{TN+FP}$	$1 - \text{Specificity}$	$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$
Value for our model	0.7	0.667	0.667	0.727	0.273	0.667

Performance Metric	Accuracy	Precision	Recall	Specificity	False positive rate	F_1 score
Formulae	$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$\frac{TN}{TN+FP}$	$1 - \text{Specificity}$	$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$
Value for our model	0.7	0.667	0.667	0.727	0.273	0.667

Reliability

- Reliability is the degree to which an assessment tool produces consistent results
- Inter-rater reliability is used to measure the degree to which different raters agree while assessing the same thing
- In case of logistic regression the raters are the actual labels and predicted labels for the categorical target variable
- In logistic regression, the inter-rater reliability is the number of labels that match in both the predicted and actual instances

Reliability - Kappa statistic

The kappa statistics is used to test inter-rater reliability

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

p_o = relative observed agreement between raters

p_e = hypothetical probability of chance agreement

Kappa statistic

- Kappa statistic is a measure of inter-rater reliability or degree of agreement

- Kappa statistic can take values from the range [-1,1]

Kappa	Interpretation
<0	No agreement
0-0.2	Slight agreement
0.2-0.4	Fair agreement
0.4-0.6	Moderate agreement
0.6-0.8	Substantial agreement
0.8-1	Almost perfect agreement

Calculation of kappa

1. Calculate p_o , let A: Actual values and B: Predicted values

2. Calculate $P(A \cap B)_{positive}$

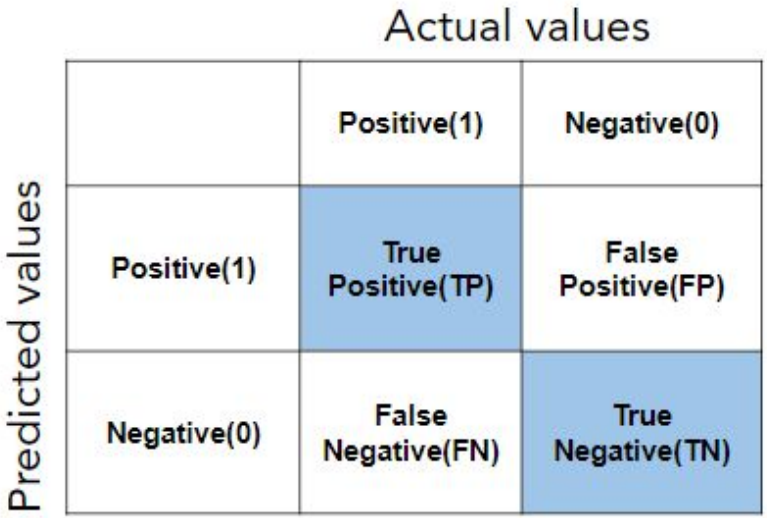
   $P(A \cap B)_{positive}$ = Probability that both actual and predicted values are positive

1. Calculate $P(A \cap B)_{negative}$

   $P(A \cap B)_{negative}$ = Probability that both actual and predicted values are negative

1 $p_o = P(A \cap B)$ + $P(A \cap B)$

Calculation of p_o

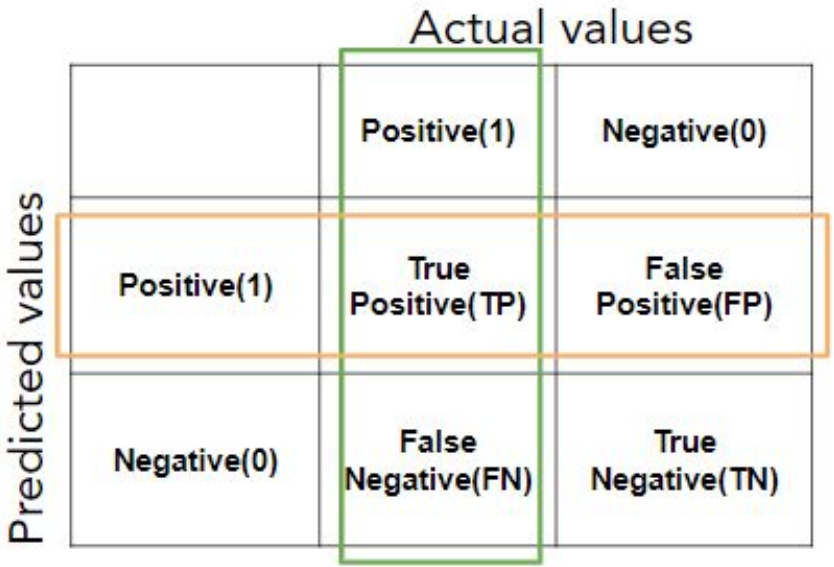


p_o is the observed agreement i.e when the actual and predicted values match

$$p_o = \frac{\text{number of instances in agreement}}{\text{total instances}}$$

$$p_o = \frac{TP+TN}{TP+TN+FP+FN}$$

Calculation of $P(A \cap B)_{positive}$

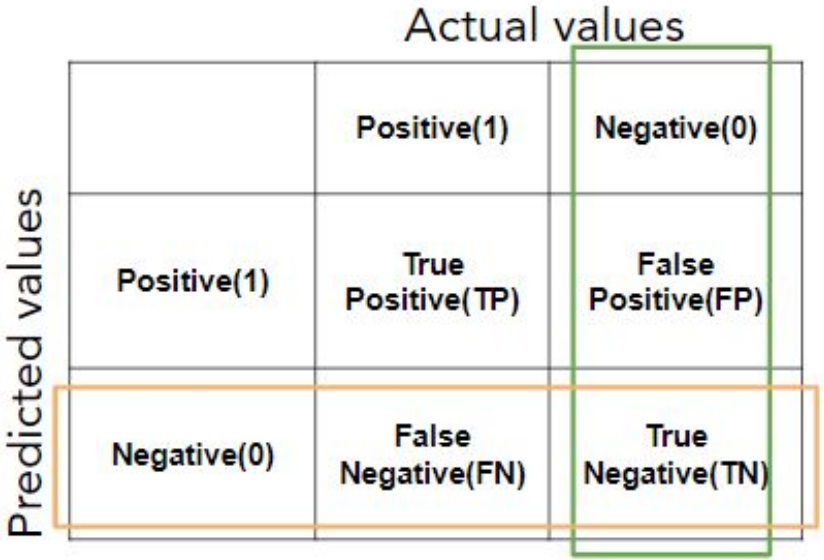


Since A and B are independent events:

$P(A \cap B)_{positive} = P(A)_{positive} * P(B)_{positive}$

$P(A)_{positive}$	$P(B)_{positive}$
$\frac{TP+FN}{TP+TN+FP+FN}$	$\frac{TP+FP}{TP+TN+FP+FN}$

Calculation of $P(A \cap B)_{negative}$



Since A and B are independent events:

$P(A \cap B)_{negative} = P(A)_{negative} \cdot P(B)_{negative}$

$P(A)_{negative}$	$P(B)_{negative}$
$\frac{FP+TN}{TP+TN+FP+FN}$	$\frac{FN+TN}{TP+TN+FP+FN}$

Calculation of p_e

p_e is hypothetical probability of chance agreement i.e when either both actual and predicted values are positive or both are negative

$$p_e = P(A \cap B)_{\text{positive}} + P(A \cap B)_{\text{negative}}$$

Performance metrics

The following metrics can be used to evaluate the performance of classification models:

- Confusion matrix
- Cross entropy
- ROC

Cross Entropy

- Cross entropy is the loss function commonly used in classification problems

- As the prediction goes closer to actual value the cross entropy decreases

$$H(y) = -\sum_i y_{act(i)} \ln\left(y_{pred(i)}\right)$$

i = class (0 or 1)

H(y) = cross entropy

$y_{act(i)}$ = actual probability for class i

$y_{pred(i)}$ = predicted probability for class i

Calculation of cross entropy

Let us consider one observation from our considered example:

	Actual probability	Predicted probability
Fish present	1	0.72
Fish not present	0	0.28

$$H(y) = -\sum_i y_{act(i)} \ln\left(y_{pred(i)}\right)$$

$$= -y_{act(1)} \ln\left(y_{pred(1)}\right) - y_{act(0)} \ln\left(y_{pred(0)}\right)$$

$$= -1 \cdot \ln(0.72) - 0 \cdot \ln(0.28)$$

$$= -(-0.3285) = 0.33$$

Performance metrics

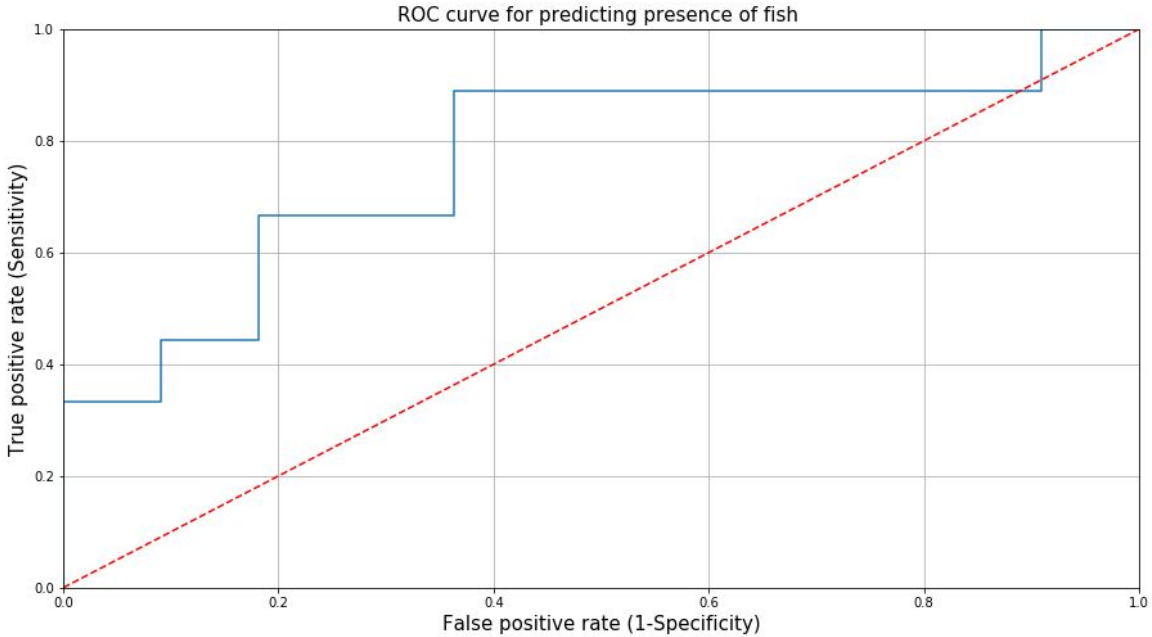
The following metrics can be used to evaluate the performance of classification models:

- Confusion matrix
- Cross entropy
- **ROC**

ROC

- Receiver operating characteristics (ROC) curve is
- The TPR and FPR values change with different threshold values
- ROC curve is the plot of TPR against the FPR values obtained at all possible threshold values

ROC curve for our example



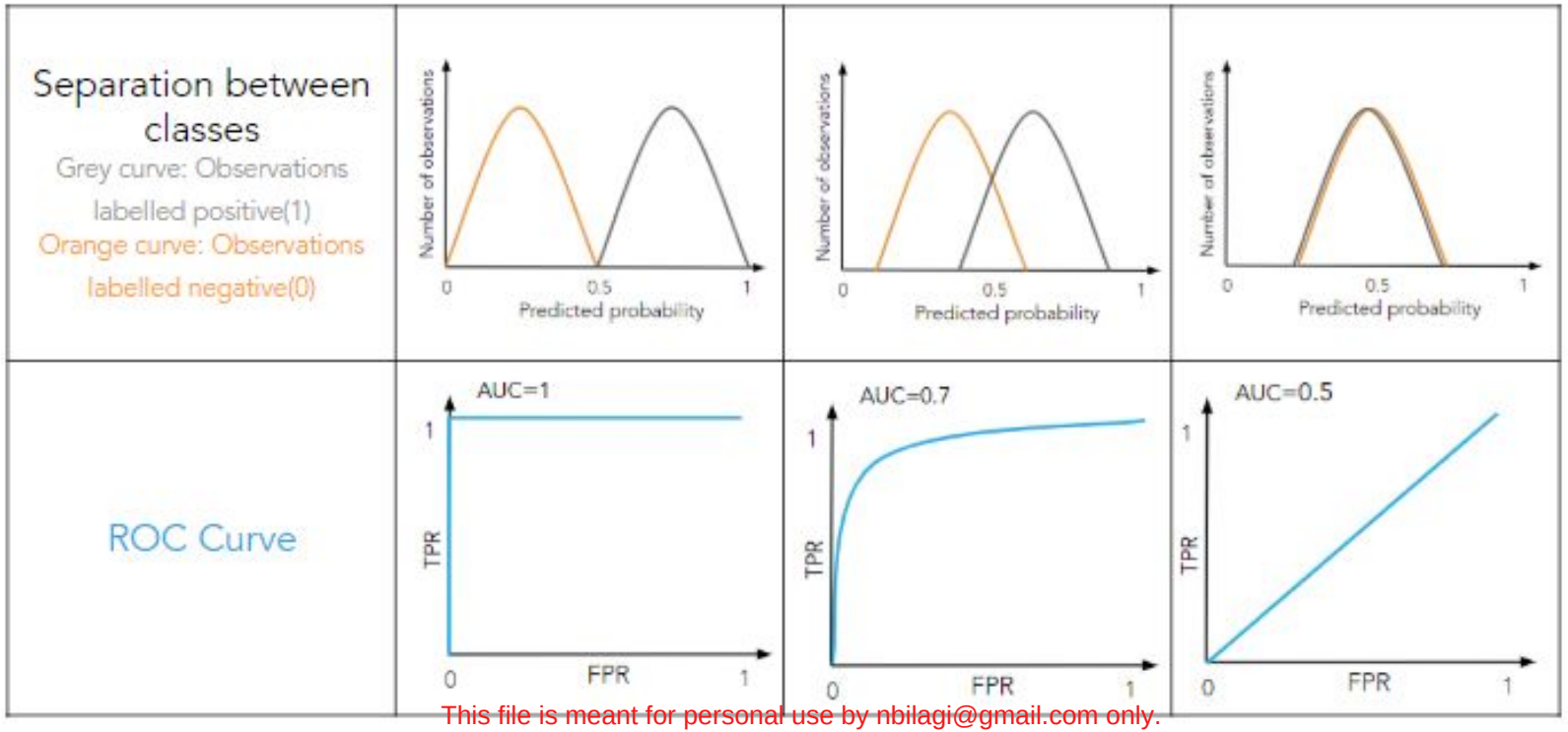
TPR	FPR	Threshold
0.33	0	0.78
0.33	0.09	0.73
0.44	0.091	0.62
0.44	0.18	0.61
0.66	0.18	0.55
0.67	0.36	0.46
0.88	0.36	0.35
0.89	0.91	0.159
1	0.91	0.157

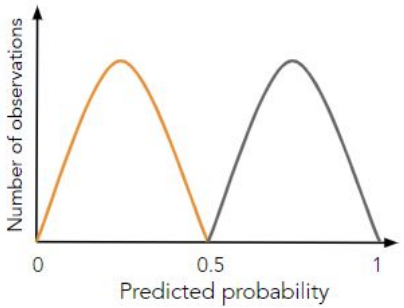
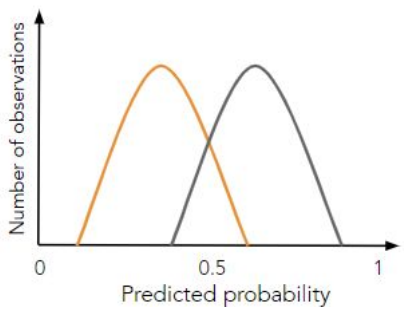
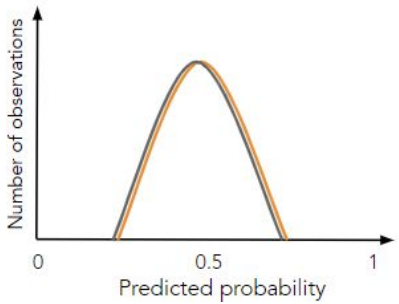
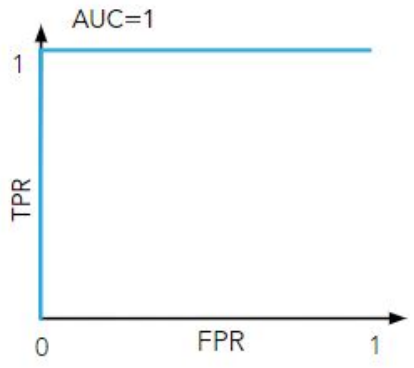
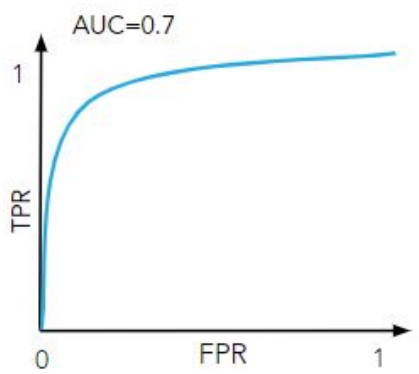
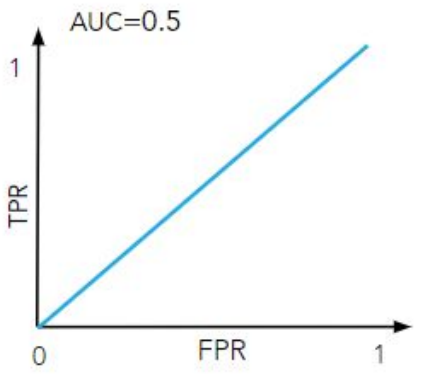
TPR	FPR	Threshold
0.33	0	0.78
0.33	0.09	0.73
0.44	0.091	0.62
0.44	0.18	0.61
0.66	0.18	0.55
0.67	0.36	0.46
0.88	0.36	0.35
0.89	0.91	0.159
1	0.91	0.157

AUC

- Area under the ROC curve (AUC) is the measure of separability between the classes of target variables
- AUC increases as the separation between the classes increases
- Higher the AUC better the model

Effect of separation between classes on ROC



Separation between classes Grey curve: Observations labelled positive(1) Orange curve: Observations labelled negative(0)			
ROC Curve			

Youden’s index

- Sensitivity and specificity represent the total number of correctly identifies samples (true positives and the true negatives)
- Youden’s index is the classification cut-off probability for which the (Sensitivity + Specificity -1) value is maximized
- Higher the value of Youden’s index better the model

$$\text{Youden's Index} = \max\left(\text{Sensitivity} + \text{Specificity} - 1\right) = \max\left(TPR - FPR\right)$$

Imbalanced Data

Imbalanced data

- Data is imbalanced if there are more records of one class compared to other classes
- Imbalanced data may lead to the accuracy paradox
- In reality datasets always have some degree of imbalance

Example of imbalanced data

For example : Consider we have information about 500 patients

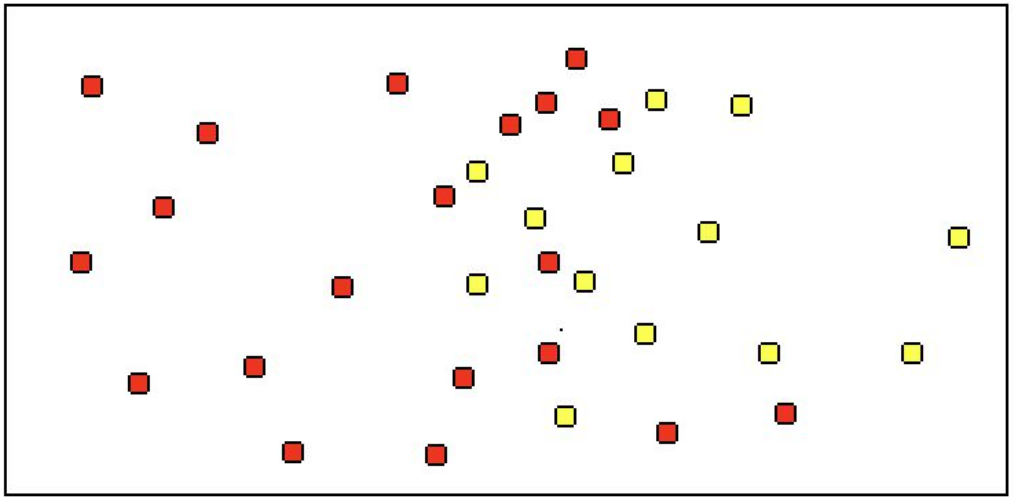
	Diabetic- Yes	Diabetic-No
Number of records	79	421
% of records	15.8	84.2

Handling imbalanced data

- Up sample minority class
- Down sample majority class
- Change the performance metric
- Try synthetic sampling approach
- Use different algorithm

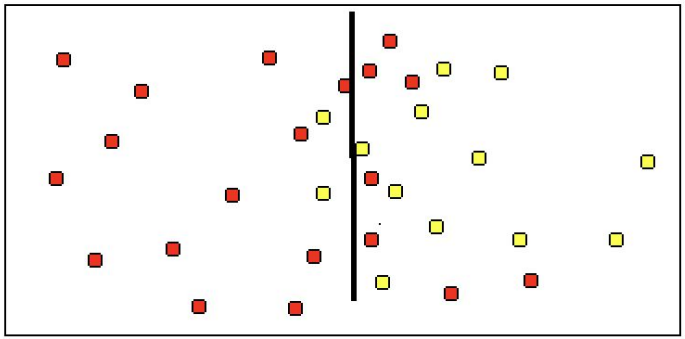
Improper fit

Classify the following data by draw a line or curve

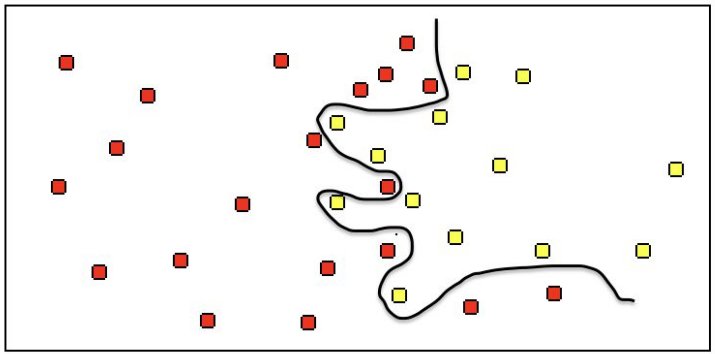


Improper fit

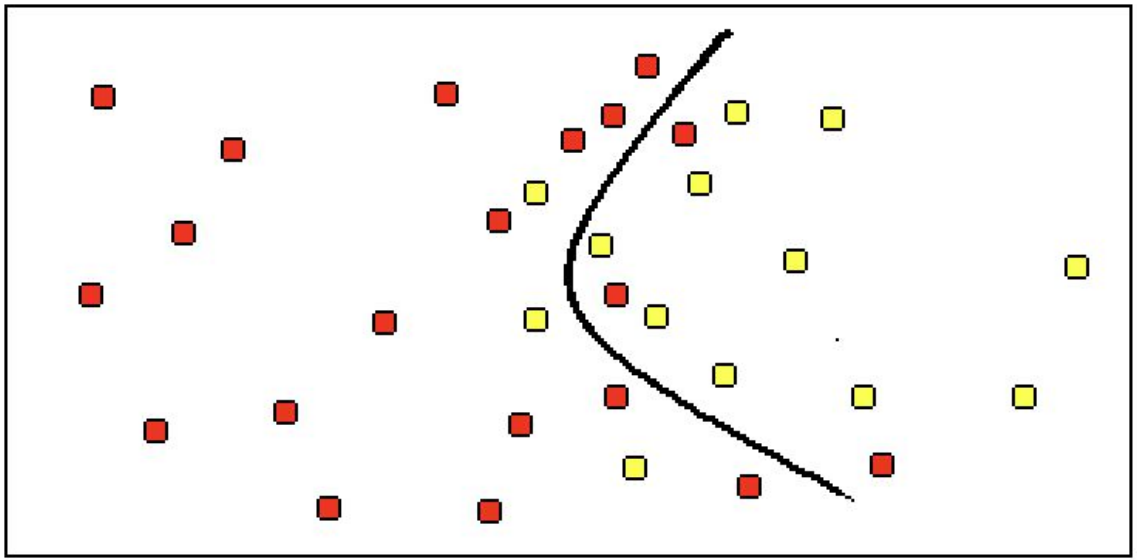
Underfitti



Overfitting



Good fit



Data Imbalance

- If in a dataset there are too few examples of the minority class then it becomes difficult for a model to learn the decision boundary effectively.

- One way to solve this problem is to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model.

Data Imbalance

- An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective.

- SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

SMOTE

- Specifically, a random example from the minority class is first chosen. Then *k* of the nearest neighbors for that example are found (typically *k*=*5*). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space.

Thank You