Supervised Learning
Classification

Logistic Regression

Agenda

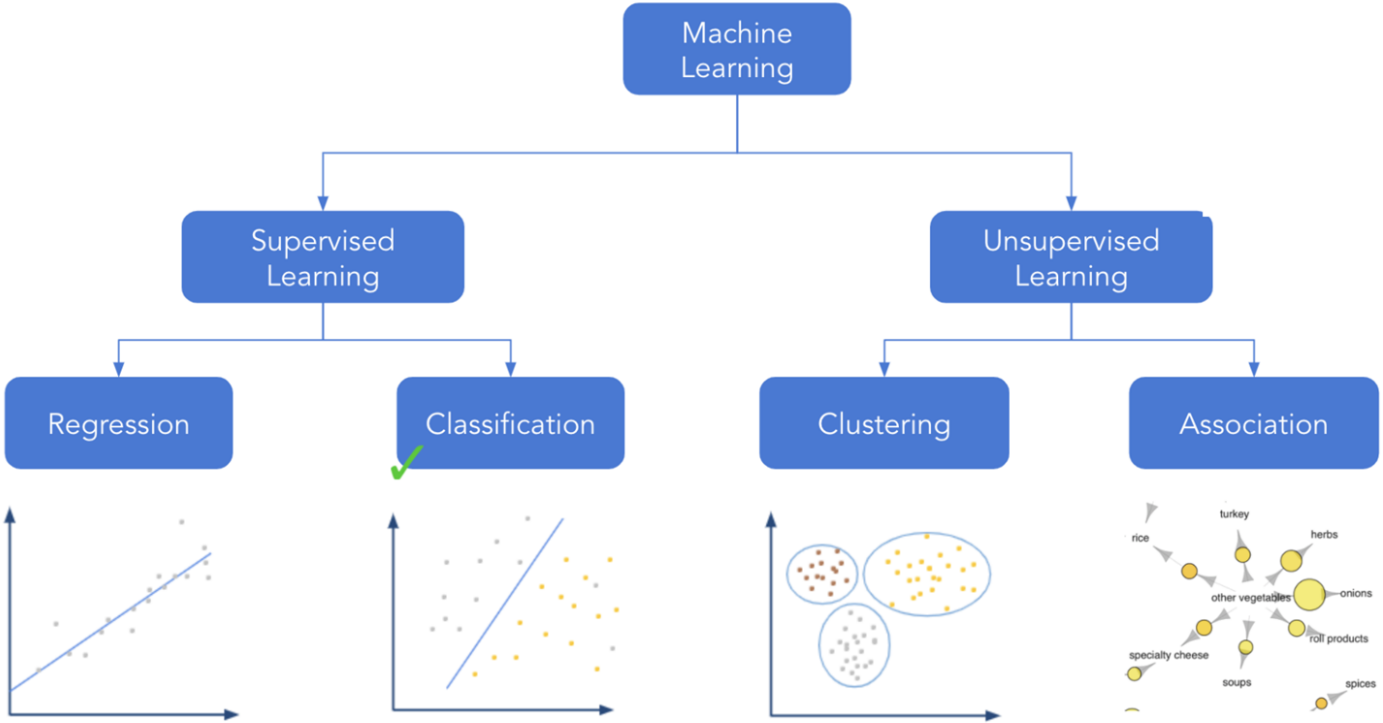
- Types of Supervised Learning
- Standard process for Data Science Project
- Visiting Basics
  - Odds Ratio
- Binomial Logistic Regression
  - Maximum Likelihood Estimation
  - Assumptions of Logistic Regression
  - Significance of Coefficients

Agenda

- Model Evaluation Metrics
  - Deviance
  - AIC
  - Pseudo R^2
- Model Performance Measures
  - Confusion Matrix
  - Cross Entropy
  - ROC - AUC Score
- Imbalanced Data

Machine Learning

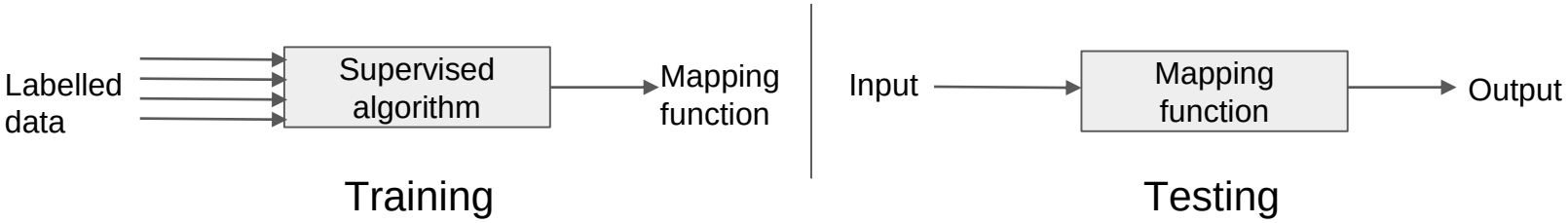
In this session, we shall cover classification



Supervised Learning

Supervised learning

Supervised learning aims at finding a model that maps the output (target) variable to the input (predictor) variables



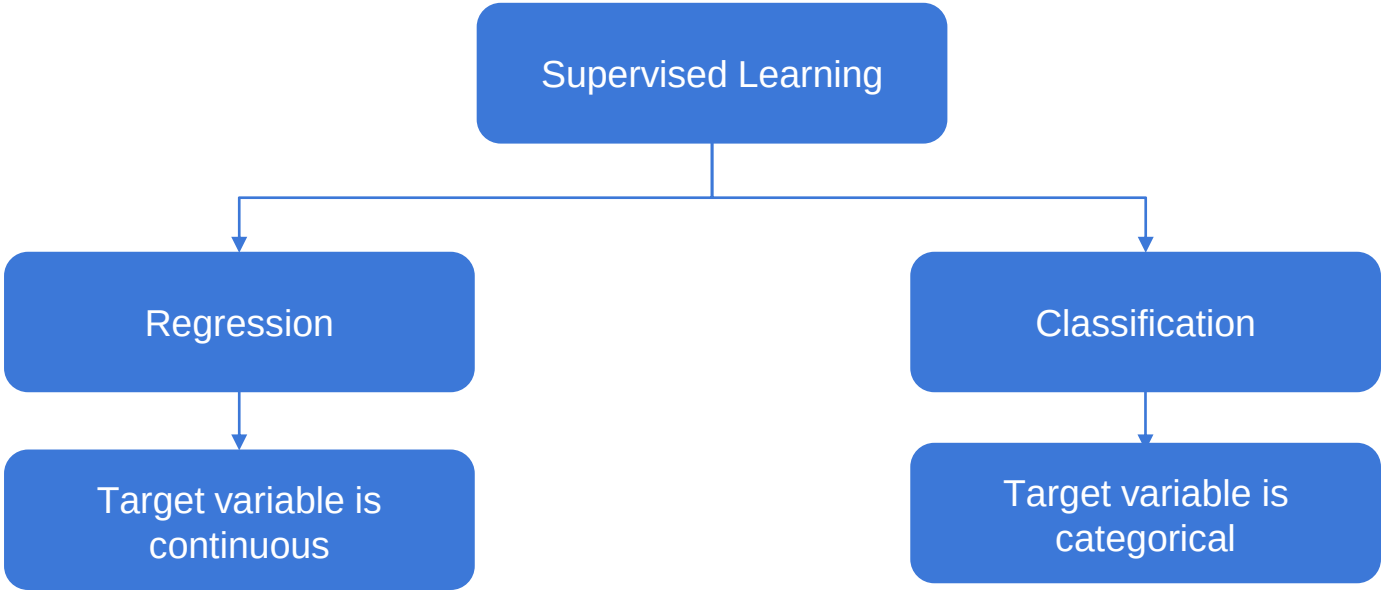
Example: Detection of phishing emails based on certain phrases like ‘You have won million’. More such phrases are prespecified while training the model. So if a new email also contains a similar phrase such emails can directly be tagged as spam.

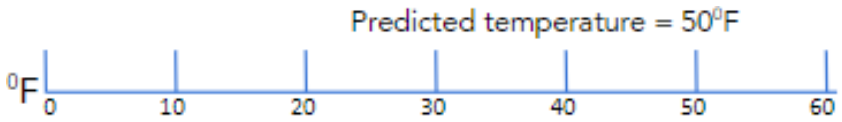
Supervised learning problems

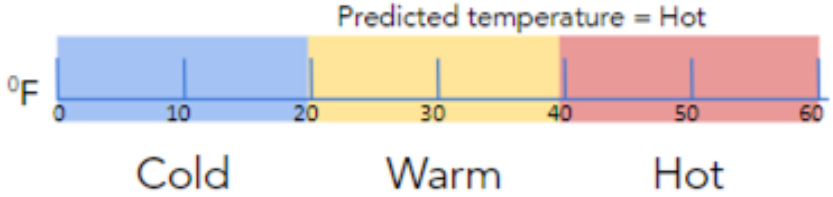
Supervised learning is mainly used for two types of problems:

- Regression problem
- Classification problem

Regression vs classification







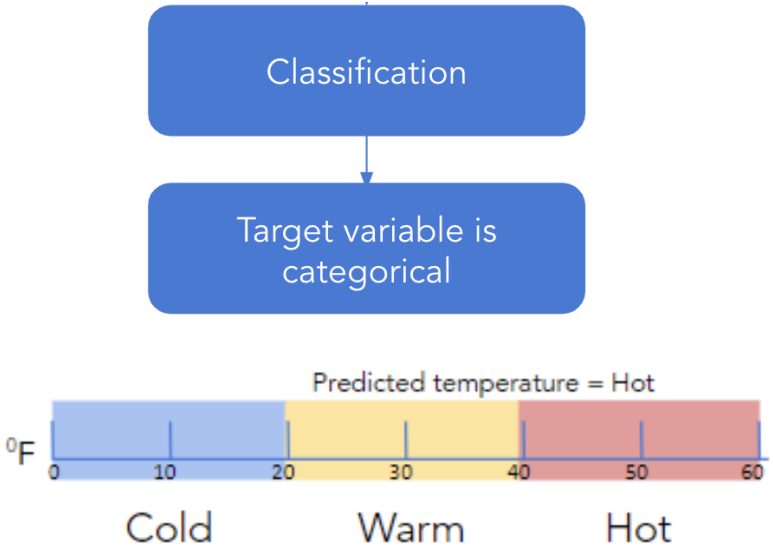
Classification

Class label

For classification, the target variable has categories.

In the example, Cold, Warm and Hot are the categories of the target variable.

These categories are called the class labels.



Classification

- An instance is mapped to one of many available labels
- Labels are the fixed number of values taken by the target variable
- The machine learns the pattern from train data where the labels are known for all instances. Then the learning can be used on new data where labels need to be predicted

Example of classification

Consider the example where we have inventory data for an online retailer which includes the number of orders, type and demand area for each item. The aim is to classify the items based on if they have a high or low demand.

This process can be automated using machine learning algorithms for classification.

Types of classification

- Binary classification:

  Classification with only two class labels

  Example: Emails can be classified into spam and ham

- Multiclass classification:

  Classification with more than two distinct class labels

  Example: Classification of land based on types of soils

Standard Process for Data Science Project

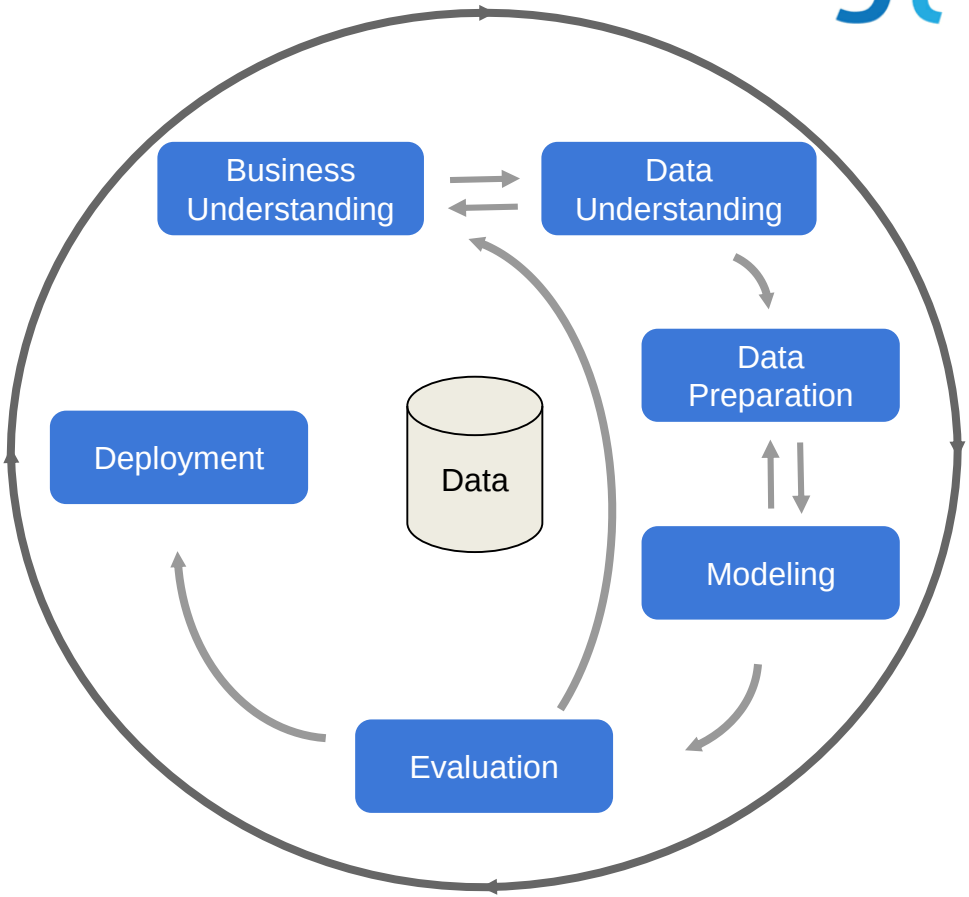
CRISP-DM

Cross Industry Standard Process for Data Mining (CRISP-DM) is a standard process used for data mining

CRISP-DM phases

CRISP-DM breaks data mining into six phases:

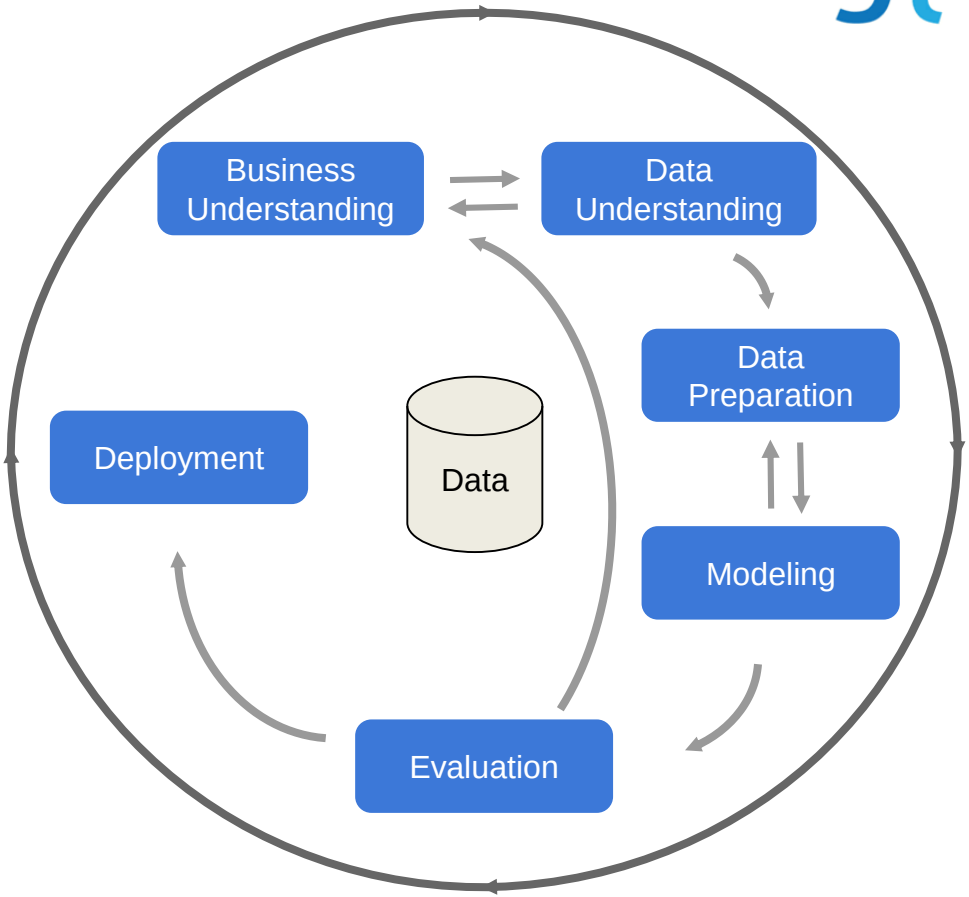
- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment



CRISP-DM phases

CRISP-DM breaks data mining into six phases:

- **Business Understanding**
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment



Business understanding

In this phase, we define what problem we are trying to solve.

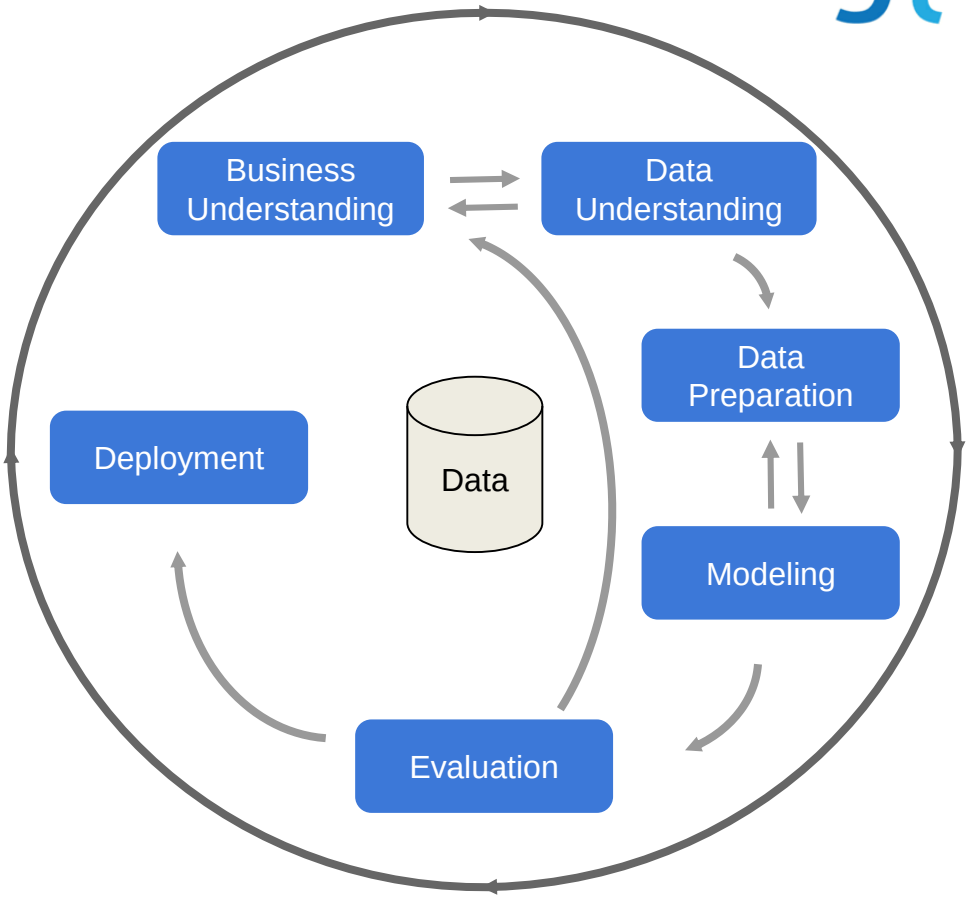
Example: Consider the example where we have inventory data for an online retailer which includes the number of orders, type and demand area for each item. The aim is to classify the items based on if they have a high or low demand. So we can define clear business problems like:

- Is the type of item related to the demand for the item?
- Can attributes in the considered data be used to classify the entire inventory list with reasonable accuracy?

CRISP-DM phases

CRISP-DM breaks data mining into six phases:

- Business Understanding
- **Data Understanding**
- Data Preparation
- Modeling
- Evaluation
- Deployment



Data understanding

This phase involves understanding the data considered for finding the solution.

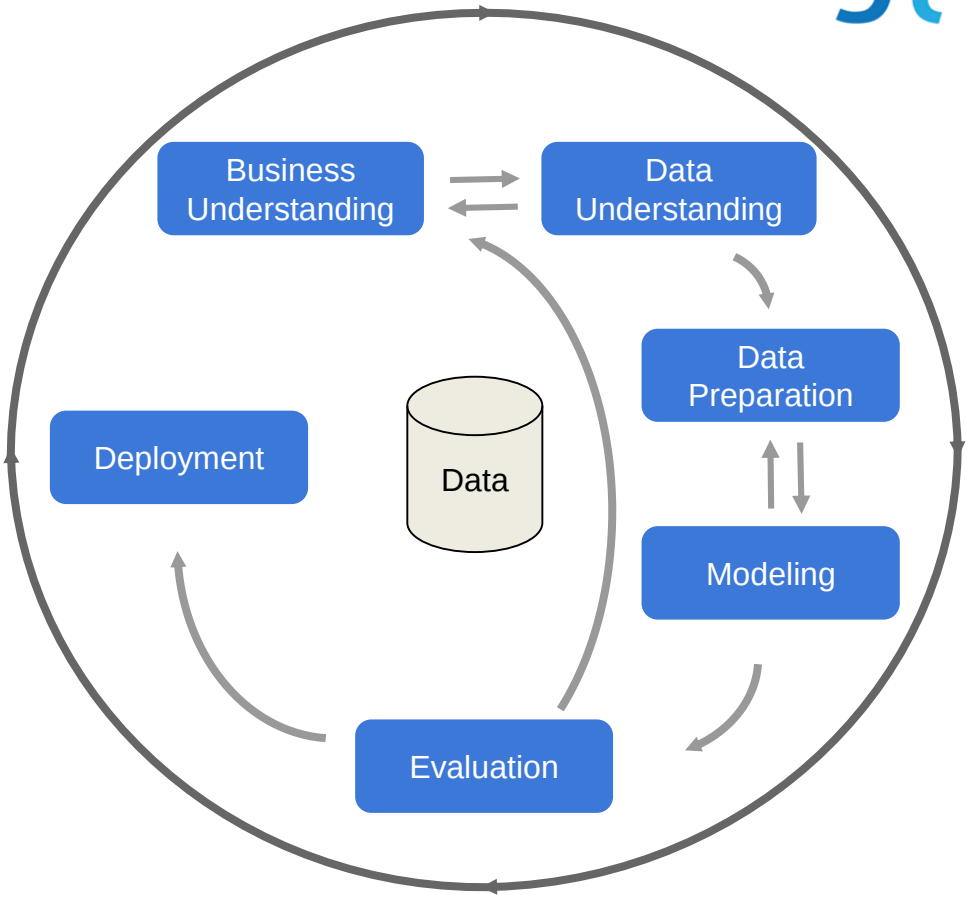
Example: Consider the example where we have inventory data for an online retailer which includes the number of orders, type and demand area for each item. The aim is to classify the items based on if they have a high or low demand.

It is important to know if the items are perishable or non-perishable. For instance, items like dairy, cosmetics, and so on, can not be stocked, and hence adequate inventory should be available to meet the demand.

CRISP-DM phases

CRISP-DM breaks data mining into six phases:

- Business Understanding
- Data Understanding
- **Data Preparation**
- Modeling
- Evaluation
- Deployment



Data preparation

This phase involves cleaning and processing the data to be in a format suitable for the model used to solve the problem.

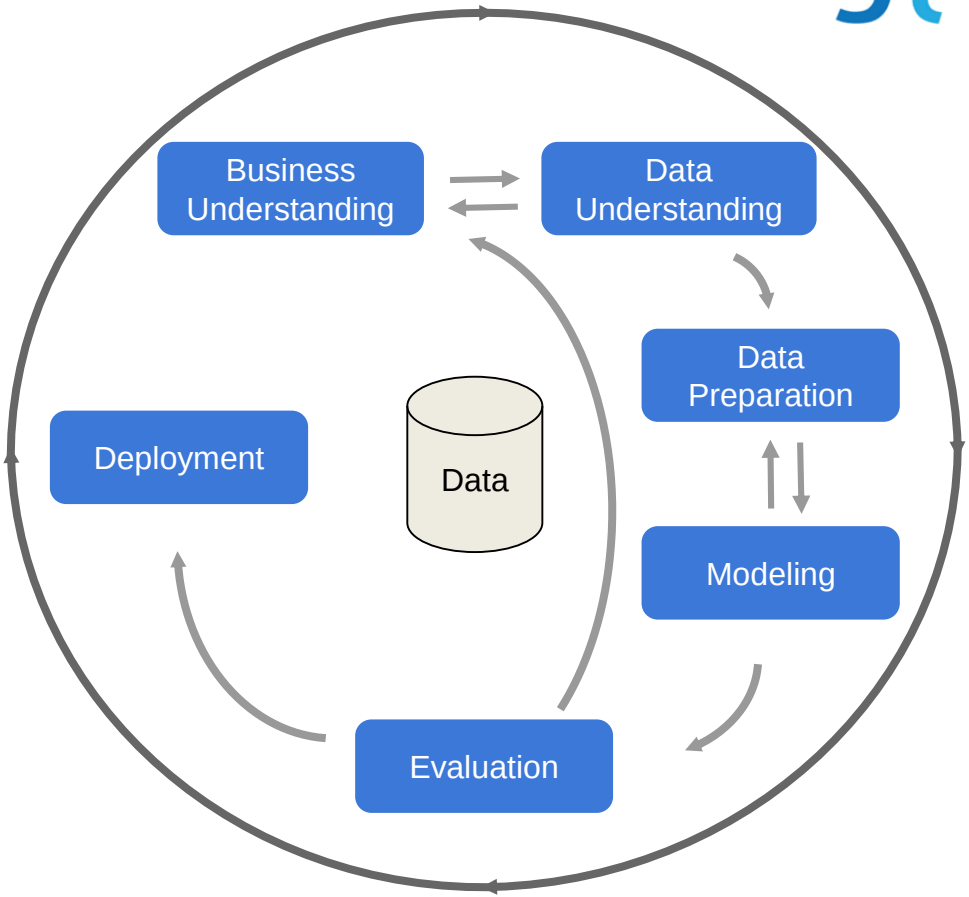
Example: Consider the example to classify the items based on if they have a high or low demand. We can prepare the data as follows:

- Treat the missing values
- The categorical variables need to be dummy encoded
- Check for correlation among variables

CRISP-DM phases

CRISP-DM breaks data mining into six phases:

- Business Understanding
- Data Understanding
- Data Preparation
- **Modeling**
- Evaluation
- Deployment



Modeling

- This phase involves finding the model that captures the solution to the business problem using available data

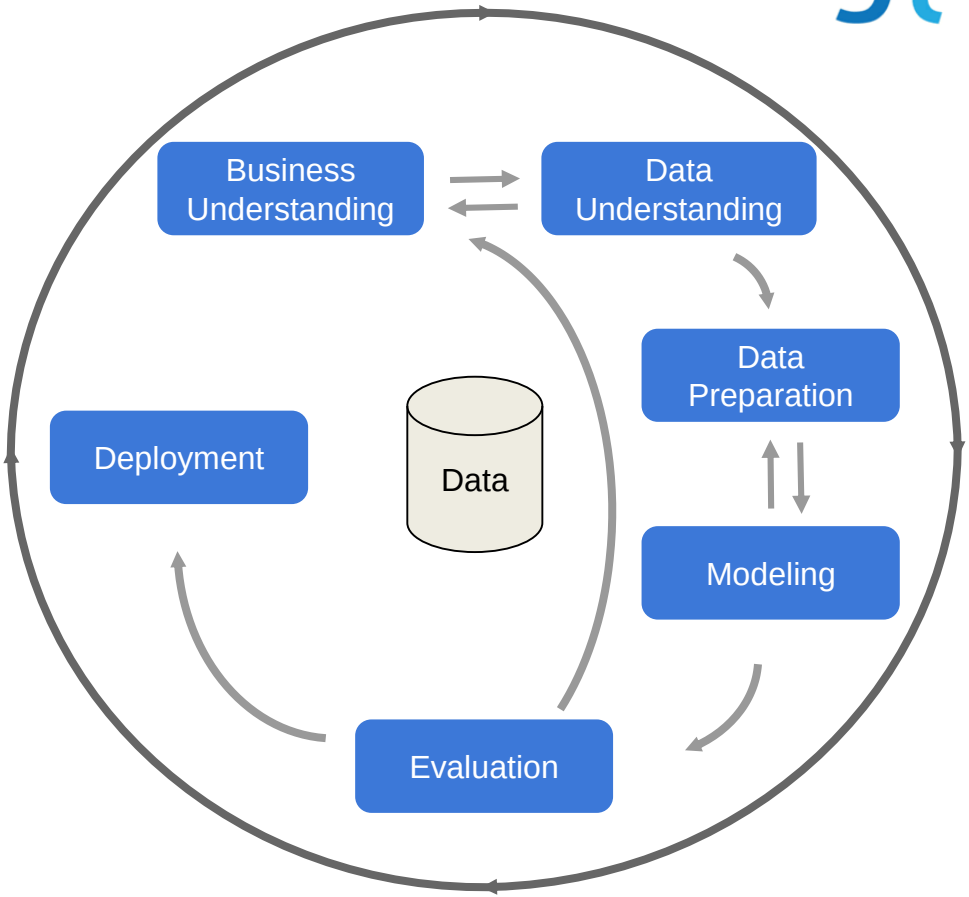
- We may have to try multiple models and go back and forth between data preparation and modelling to choose the correct model

Example: In the modelling phase we try to find a function that maps the attributes like the number of orders, type, etc from the data to demand for the item.

CRISP-DM phases

CRISP-DM breaks data mining into six phases:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- **Evaluation**
- Deployment



Evaluation

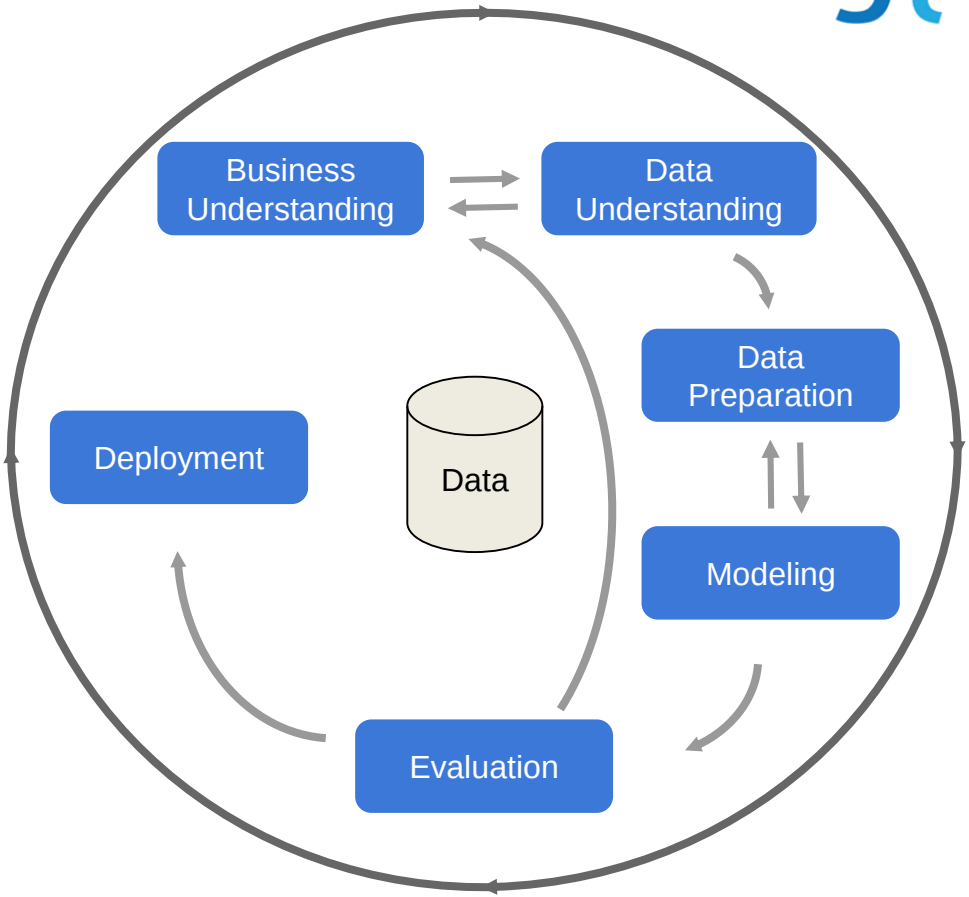
Once the model is built we need to check how good the model performs on unseen data. This process is done during the evaluation phase.

Example: We can check the model performance on data for which we know the actual demand. Using that data we can compare the predicted and actual values and evaluate.

CRISP-DM phases

CRISP-DM breaks data mining into six phases:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- **Deployment**



Deployment

If we are satisfied with the performance of the model from the previous phase we deploy it in the deployment phase

Example: For the considered example of predicting demand for an item, perhaps we could develop an app that takes input as the attribute values for an item and returns the demand for that item to the retailer.

Visiting Basics

Odds vs probability

Odds of an event are the ratio of number of observations in favour of an event to number of observations not in favour of the event

$$\text{odds} = \frac{\text{number of observations in favour of the event}}{\text{number of observations not in favour of the event}}$$

Probability of an event is the ratio of number of observations in favour of an event to all possible observations

$$\text{probability} = \frac{\text{number of observations in favour of the event}}{\text{number of observations}}$$

Odds vs probability

Plasma score	90	90	150	165	115	180	100	170	130	166
Is the patient Diabetic?	No	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes

For the above data, the odds of a patient having diabetes is given by,

$$\text{odds} = \frac{\text{number of patients having diabetes}}{\text{number of patients not having diabetes}} = \frac{6}{4}$$

For the above data,, the probability of a patient having diabetes is given by,

$$\text{probability} = \frac{\text{number of patients having diabetes}}{\text{Total number of patients}} = \frac{6}{10}$$

Log of odds

$\text{odds of having diabetes} = \frac{6}{4}$	$\text{odds of not having diabetes} = \frac{4}{6}$
log(odds of having diabetes) = ln (1.5) = 0.405	$\log(\text{odds of not having diabetes}) = \ln(0.667) = -0.405$

- As we can see if we only consider odds value, the magnitude for each class value taken by variable is very different
- Hence, the log(odds) value is considered; so that no matter whichever the class is, the magnitude remains same
- Log of odds is the logit function used in logistic regression

Relation between odds and probability

If P(A) is probability of event A

$$\text{Odds} = \frac{\text{P(A)}}{\text{1-P(A)}}$$

$$\text{Probability} = \frac{\text{odds}}{\text{1+odds}}$$

$$\log(\text{Odds}) = \ln\left(\frac{\text{P(A)}}{\text{1-P(A)}}\right)$$

Odds ratio

- Odds ratio refers to the ratio of odds
- Odds ratio can be used to determine the impact of a feature on target variable
- For our considered example the odds ratio can be calculated as,

$$\text{odds ratio} = \frac{\text{odds of patient having diabetes}}{\text{odds of patient not having diabetes}} = \frac{\frac{6}{4}}{\frac{4}{6}} = \frac{9}{4}$$

Question:

Patients with high sugar diet are considered more susceptible to diabetes. How can we determine whether sugar content in diet has an impact on possibility of a patient getting diagnosed with diabetes? Consider the following sample data.

Sugar content in diet	High	High	Low	High	Low	High	High	Low	High	Low
Is the patient Diabetic?	Yes	No	Yes	Yes	No	Yes	Yes	No	No	No

Solution:

From the given sample data we can calculate:

1. Odds of a patient having diabetes given he has high sugar diet

$$\frac{\text{number of patients having diabetes given he has high sugar diet}}{\text{number of patients not having diabetes given he has high sugar diet}} = \frac{4}{2}$$

2. Odds of a patient having diabetes given he has low sugar diet

$$\frac{\text{number of patients having diabetes given he has low sugar diet}}{\text{number of patients not having diabetes given he has low sugar diet}} = \frac{1}{3}$$

Solution continued:

From 1 and 2 we can calculate odds ratio:

$$\text{odds ratio} = \frac{\text{odds of a patient having diabetes given he has high sugar diet}}{\text{odds of a patient having diabetes given he has low sugar diet}} = \frac{\frac{4}{2}}{\frac{1}{3}} = 6$$

Thus from the odds ratio we can see that patients with a high sugar diet are 6 times more susceptible to diabetes compared to patients who have a low sugar diet.

Binomial Logistic Regression

Question:

Consider the example below about whether or not a patient has diabetes based on plasma score. Can we use linear regression line to predict the whether the patient is diabetic?

Plasma score	90	90	150	165	115	180	100	170	130	166
Is the patient Diabetic?	No	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes

Answer:

The example about whether or not a patient has diabetes based on plasma score is a classification problem. Moreover, the target variable is categorical. Hence, we can not use linear regression line to predict the whether the patient is diabetic. We classify them as diabetic and non-diabetic.

Plasma score	90	90	150	165	115	180	100	170	130	166
Is the patient Diabetic?	No	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes

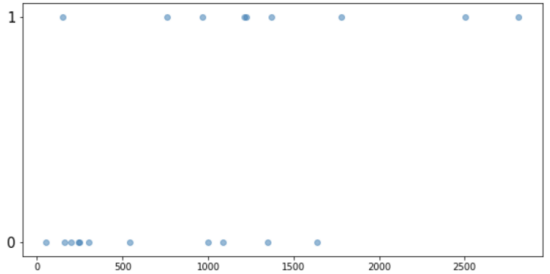
Logistic regression

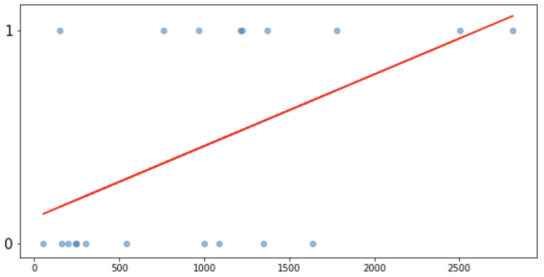
- Here on we shall consider the adjacent data
- The data tells us the presence of one fish depending on the density of other fish in a lake
- If they compete with each other, then the higher density of one may suggest the absence of the other whereas if they are symbiotic, high density of one may promote the other

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

Logistic regression

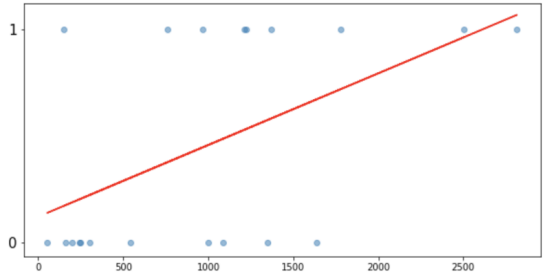
- Consider the scatter plot of the previous data (data on slide 42)
- Fit a linear regression line to it
- Note the line is not a true representative of the data



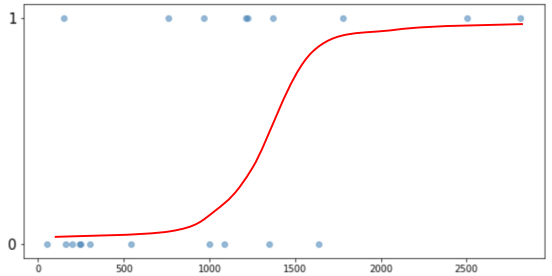


Logistic regression

- A S-shaped curve as in the figure below gives the true relationship



- Such a curve is given be the sigmoid function

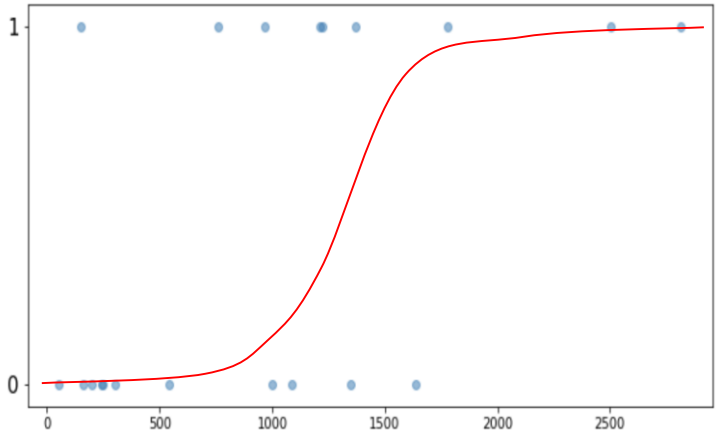


What is a sigmoid function?

- The sigmoid function is a mathematical function which is S-shaped and is given by

$$f(x) = \frac{1}{1-\exp^{-z}}$$

- It exists between 0 to 1



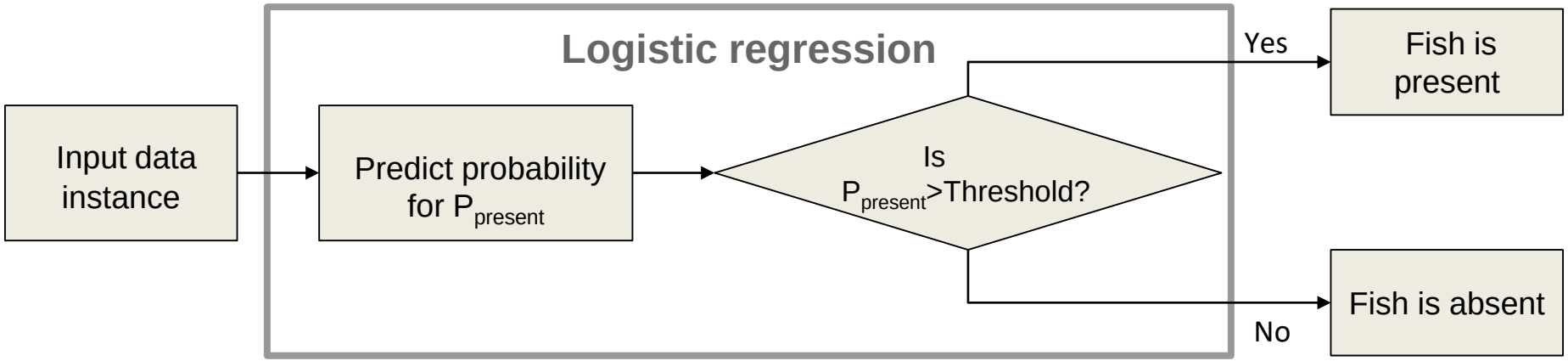
Logistic regression

- Logistic regression is a binary classification algorithm
- It predicts the probability of occurrence of a class label. Based on these probabilities the data points are labelled
- A threshold (or cut-off; commonly a threshold of 0.5 is used) is fixed, then

	Classify as
threshold < probability	Presence of fish
probability > threshold	Absence of fish

Main steps in logistic regression

Consider that logistic regression is used to identify whether or not a patient is suffering from diabetes



Logistic regression

- The logistics regression is given by

$$\pi(x) = \frac{e^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}}{1 + e^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}}$$

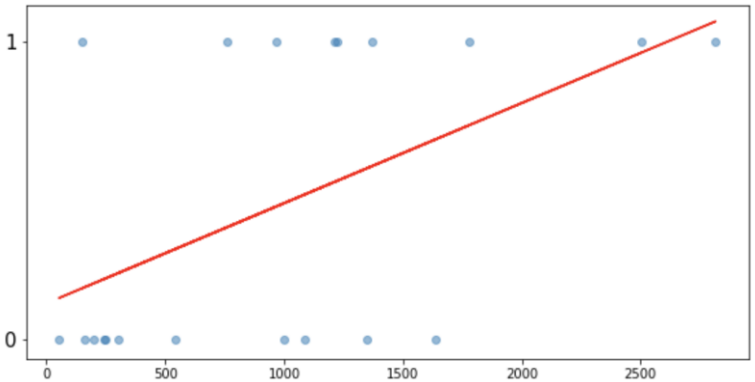
- Here, π(x) is the conditional expectation of the outcome given the values for independent variables, i.e. E(Y|X)

- It predicts the probability of occurrence of a class label by fitting the data to a function called logit function, hence called logit regression

Probability as output of logistic regression

- The logistic regression model is given by $\pi(x) = \frac{e^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}}{1 + e^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}}$
- Taking limits tending to -∞ on both side, $\lim_{x \to -\infty} \frac{e^x}{1+e^x} = 0$
- Taking limits tending to ∞ on both side, $\lim_{x \to \infty} \frac{e^x}{1+e^x} = 1$
- Thus π(x) lies in-between 0 and 1, i.e. π(x) ∈ [0,1] and can be viewed as probability

Logistic regression



- Since we are predicting probabilities values need to be between 0 and 1
- Consider the points A and B for which values for plasma score are 80 and 177 respectively, probability values are out of the expected range 0 to 1
- Hence linear regression cannot be directly used to predict probabilities

Usage

- Classification:

  The ICU in a hospital is assigned on priority to high risk patients. Logistic regression can be used to classify the list of patients into high risk and low risk records.

- Profiling:

  Nuclear fuel companies moderate various factors like pressure, temperature, etc. to produce high yielding and low yielding fuels. Based on different parameters for the current day, it can be predetermined whether the fuel produced will be high yielding or not. The company can then alter the parameters to produce high yielding fuel every day.

Parameter estimation

- The general form of logistic regression model is

$$y_i = E(y_i) + \epsilon_i$$

where the observations y_i are independent bernoulli random variables

- The expected value of y_i's is

$$E(y_i) = \pi_i = \frac{e^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}}{1 + e^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}}$$

Parameter estimation

- The method of least squares fails
- Since, unlike the linear regression, the closed form solution does not exist

where a closed form solution is an exact solution evaluated with a fixed number of operations

- So we use the method of maximum likelihood estimation

Question:

Consider the example below about whether or not a patient has diabetes based on plasma score. Since the logistic regression equation for this example will represent a straight line can we determine the residual values?

Plasma score	90	90	150	165	115	180	100	170
Is the patient Diabetic?	No	No	Yes	Yes	No	Yes	No	Yes

Solution:

For the considered example the equation of logistic regression line is given by,

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \text{Plasma Score}$$

p = probability of a patient having diabetes

$\ln\left(\frac{p}{1-p}\right)$ = logit function

β_0 = Intercept value

β_1 = Coefficient for variable Plasma Score

Plasma Score = Values taken by variable Plasma Score

Solution continued:

The value of logit function i.e the LHS for a patient who has diabetes is given by

$$\ln\left(\frac{p}{1-p}\right) = \ln\left(\frac{1}{0}\right) = \ln(1) - \ln(0) = 0 - (-\infty) = +\infty \quad \ldots(i)$$

The value of logit function i.e the LHS for a patient who has diabetes is given by

$$\ln\left(\frac{p}{1-p}\right) = \ln\left(\frac{0}{1}\right) = \ln(0) - \ln(1) = -\infty - 0 = -\infty \quad \ldots(ii)$$

Solution continued:

From i and ii we can infer that the logistic regression line will approach $+\infty$ and $-\infty$ values.

So the residuals will also take infinite values and hence cannot be determined.

Why we cannot use least squares for logistic regression?

- The residual values for the logistic regression line cannot be determined
- Least squares optimization cannot be performed without determining residuals for the line
- So we cannot use least squares optimization for logistic regression and instead use an optimization approach called maximum likelihood estimation
- Note: least squares for linear regression is a special case of maximum likelihood estimation

Maximum likelihood estimation (MLE)

- It is used to estimate the parameters of the logistic regression
- The method maximizes the likelihood function, which is the joint probability density function of the data
- The model formed by considering the parameter values, obtained by this method is the most likely model that describes the observed data
- i.e. the parameters obtained using MLE, maximizes the probability of getting a particular value of the target variable that we observed on the 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

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
- Use t

```
def deviance(X, y, model):
    return 2*metrics.log_loss(y, model.predict_proba(X), normalize=False)
```

- 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

- 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 R^2 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

```
# build the model on train data (X_train and y_train)
# use fit() to fit the logistic regression model
logreg = sm.Logit(y_train, X_train).fit()

# print the summary of the model
print(logreg.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.241326
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:        Chance of Admit   No. Observations:                  320
Model:                          Logit   Df Residuals:                      312
Method:                           MLE   Df Model:                            7
Date:                Tue, 23 Jun 2020   Pseudo R-squ.:                  0.6486
Time:                        12:35:06   Log-Likelihood:                -77.224
converged:                       True   LL-Null:                       -219.78
Covariance Type:            nonrobust   LLR p-value:                 9.137e-58
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const             -0.7119      0.330     -2.157      0.031      -1.359      -0.065
GRE Score          0.6095      0.447      1.365      0.172      -0.266       1.485
TOEFL Score        0.1989      0.403      0.493      0.622      -0.592       0.990
University Rating  0.5883      0.383      1.535      0.125      -0.163       1.339
SOP                0.1768      0.374      0.473      0.636      -0.555       0.909
LOR                0.5118      0.308      1.662      0.096      -0.092       1.115
CGPA               2.6273      0.544      4.832      0.000       1.562       3.693
Research_1         0.5819      0.465      1.251      0.211      -0.329       1.493
==============================================================================
```

Calculate the AIC (Akaike Information Criterion) value.

It is a relative measure of model evaluation. It gives a trade-off between model accuracy and model complexity.

```
# 'aic' retuns the AIC value for the model
print('AIC:', logreg.aic)
```

```
AIC: 170.44859325107456
```

We can use the AIC value to compare different models created on the same dataset.

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 the scale, i.e. on [0,1], and work exactly the same with higher values indicating 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, 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 $R^2_{\text{Cox-Snell}}$ is less than 1

- For a model with likelihood 1, i.e the 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

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

- 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 then $R^2_{\text{Nagelkerke}} = 0$

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 admission data as described before. Here is the preview of the data using df.hea

```
# load the csv file
# store the data in 'df_admissions'
df_admissions = pd.read_csv('Admission_predict.csv')

# display first five observations using head()
df_admissions.head()
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.500000	4.500000	9.650000	1	1
1	2	324	107	4	4.000000	4.500000	8.870000	1	1
2	3	316	104	3	3.000000	3.500000	8.000000	1	0
3	4	322	110	3	3.500000	2.500000	8.670000	1	1
4	5	314	103	2	2.000000	3.000000	8.210000	0	0

Let us now see the number of variables and observations in the data.

```
# use 'shape' to check the dimension of data
df_admissions.shape
```

```
(400, 9)
```

Interpretation: The data has 400 observations and 9 variables.

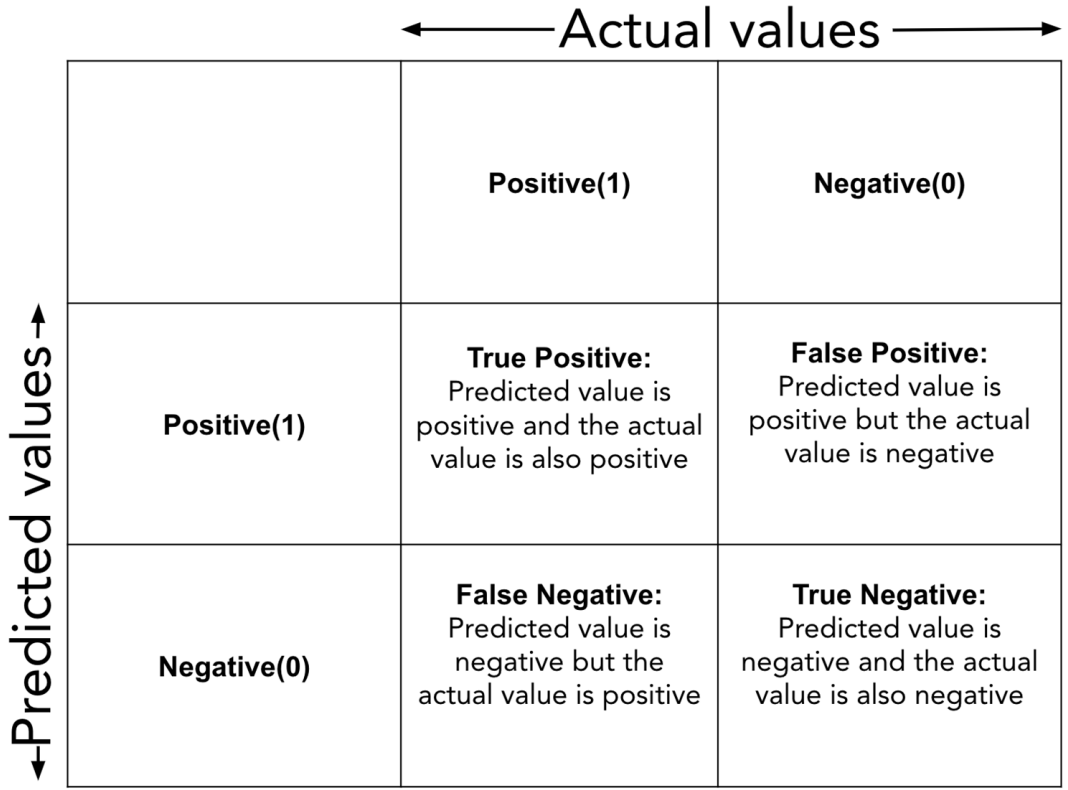
Performance metrics

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

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

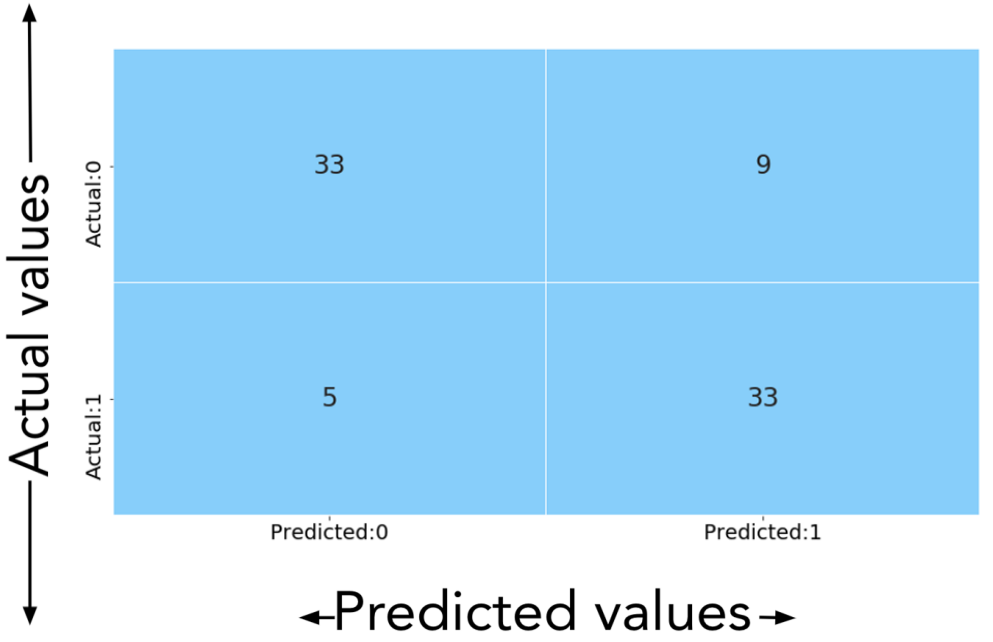
Confusion matrix

- 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 the chances of the admittance in the master program

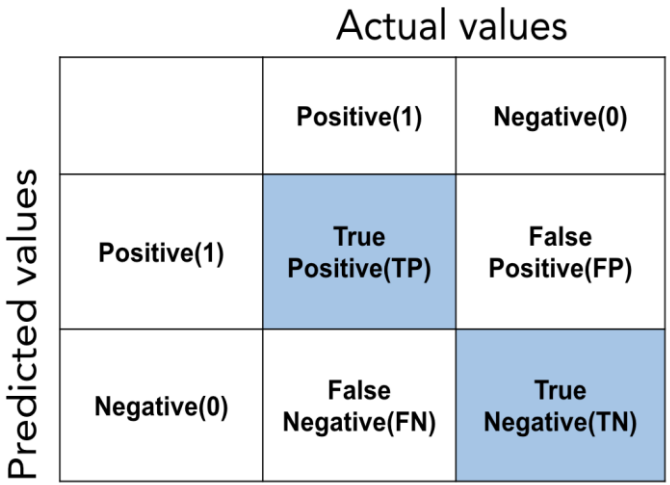


Performance evaluation metrics

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

Accuracy



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

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

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

Accuracy: It is the ratio of correct predictions (i.e. TN+TP) to the total observations. According to the confusion matrix, it is the ratio of the sum of diagonal elements to the sum of all the in the matrix. It is not a very good measure if the dataset is imbalanced.

```
# calculate the accuracy
accuracy = (TN+TP) / (TN+FP+FN+TP)

# print the accuracy
accuracy
```

```
0.825
```

- Higher the value of accuracy better is the model

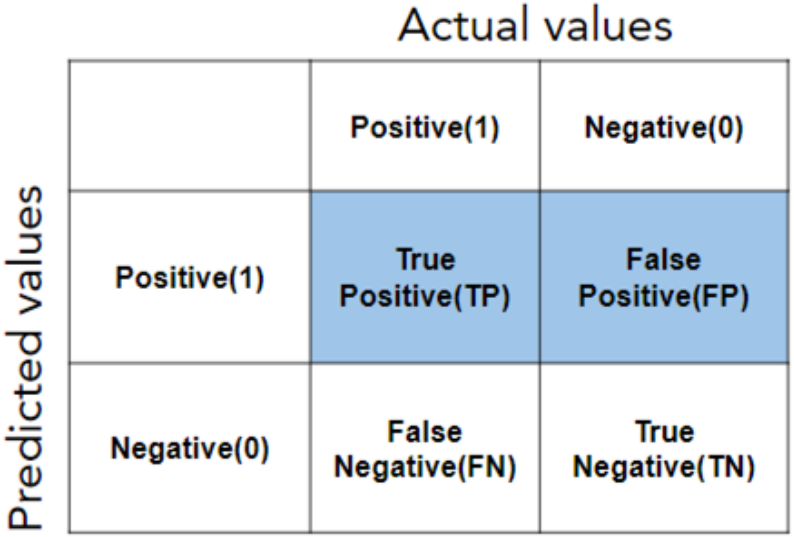
Accuracy is not always a reliable metric

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

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

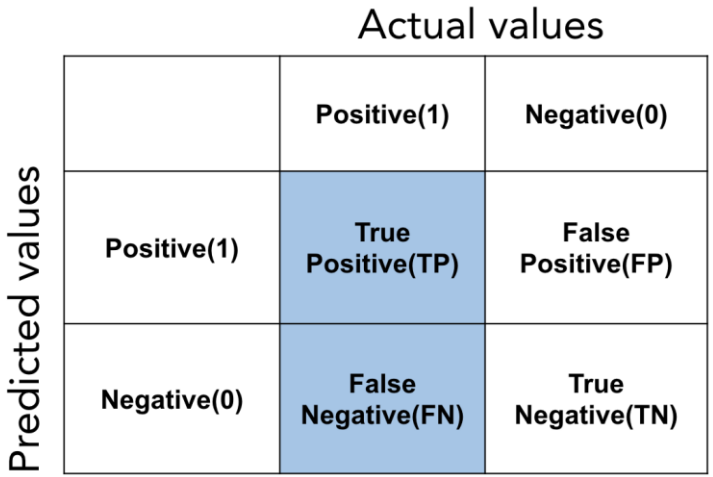
- Higher is the precision better the model

Precision: It is defined as the ratio of true positives to the total positive predictions.

```
# calculate the precision value
precision = TP / (TP+FP)

# print the value
precision
```

```
0.7857142857142857
```

Recall



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

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

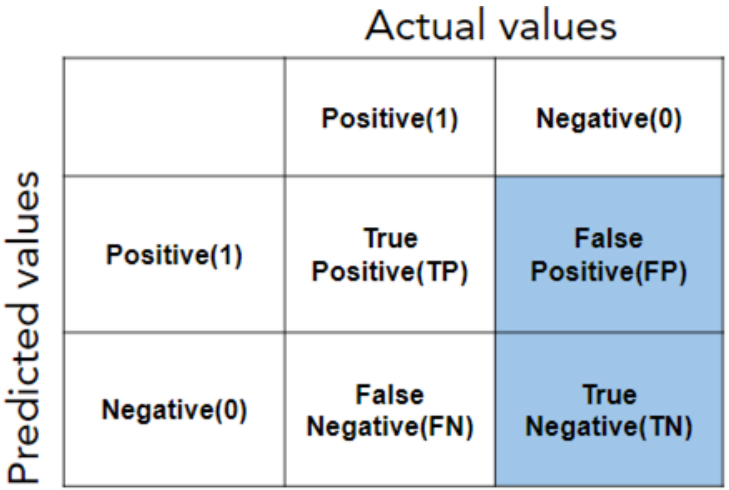
- Higher value of TPR implies a better model

Recall: It is the ratio of true positives to the total actual positive observations. It is also known as, `Sensitivity` or `True Positive Rate`.

```
# calculate the recall value
recall = TP / (TP+FN)

# print the value
recall
```

```
0.868421052631579
```

False positive rate



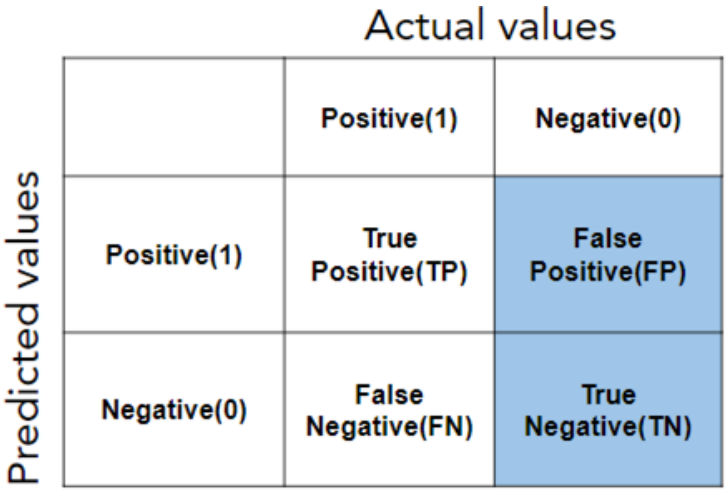
- False Positive Rate (FPR) is the proportion of actual negative cases that were predicted positive (incorrectly)

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

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

- 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

Specificity: It is the ratio of true negatives to the total actual negative observations.

```
# calculate the specificity value
specificity = TN / (TN+FP)

# print the value
specificity
```

```
0.7857142857142857
```

F_1 score

- F_1 score is the harmonic mean of precision and recall values for a classification model
- It is a good measure if we want to find a balance between precision and recall or if there is an 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	0.825	0.785	0.868	0.785	0.215	0.825

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 the 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]

```
# compute the kappa value
kappa = cohen_kappa_score(y_test, y_pred)

# print the kappa value
print('kappa value:',kappa)
```

```
kappa value: 0.6508728179551122
```

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

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

Performance metrics

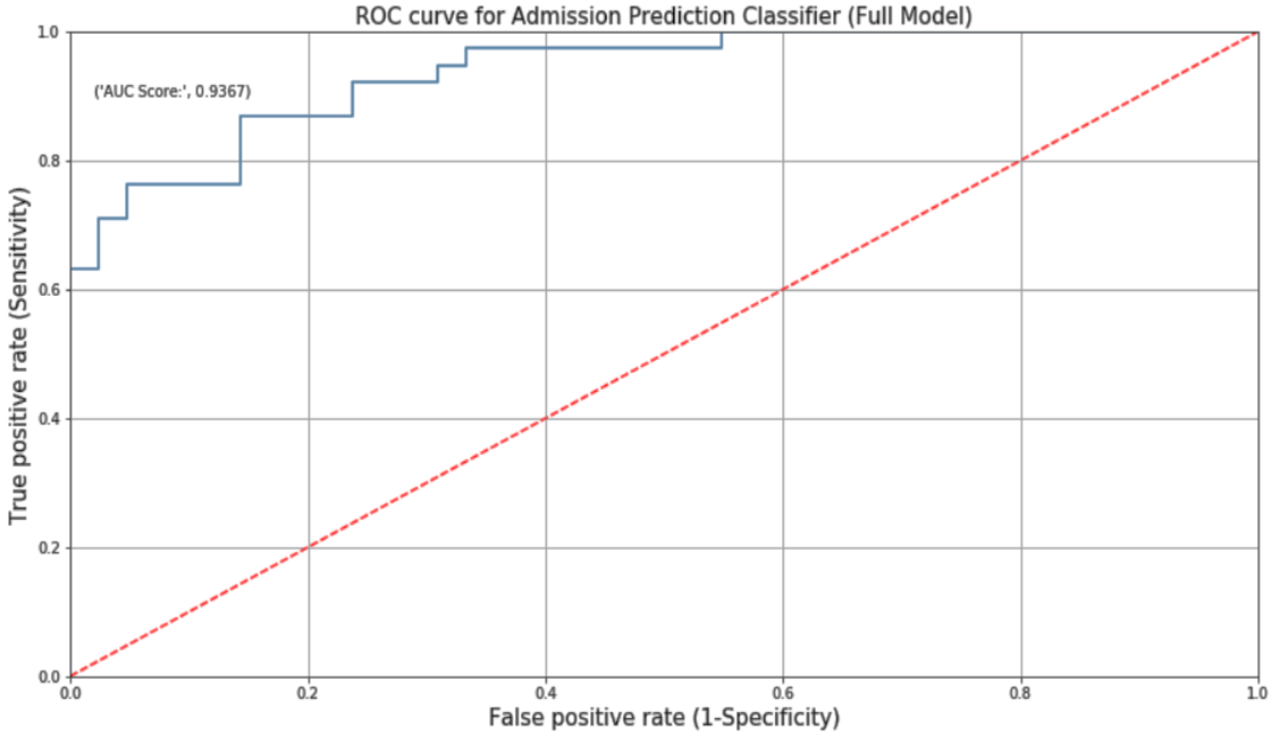
The following metrics can be used to evaluate the performance of classification models:

- 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

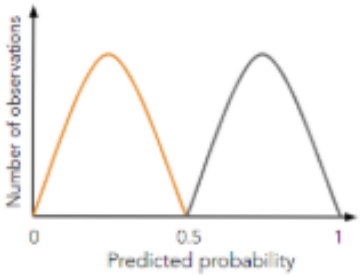
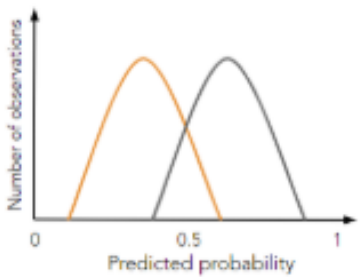
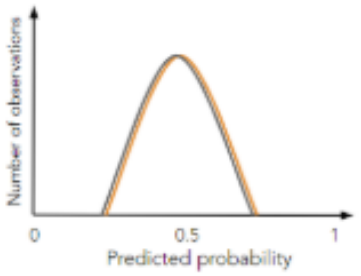
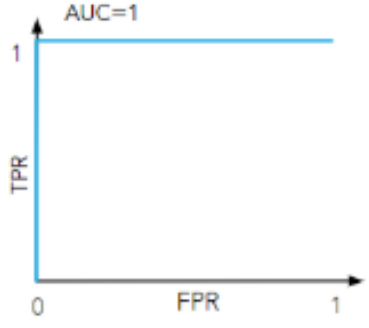
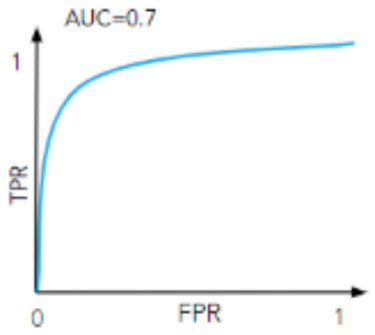
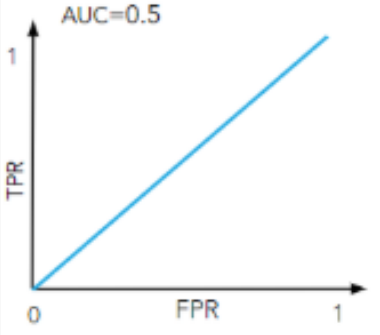


Interpretation: The red dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).
From the above plot, we can see that our classifier (logistic regression) is away from the dotted line; with the AUC score 0.9367.

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

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

Youden’s index working on dataset

3.1.1 Youden's Index

Youden's Index is the classification cut-off probability for which the (Sensitivity + Specificity - 1) is maximized.

Youden's Index = max(Sensitivity + Specificity - 1) = max(TPR + TNR - 1) = max(TPR - FPR)

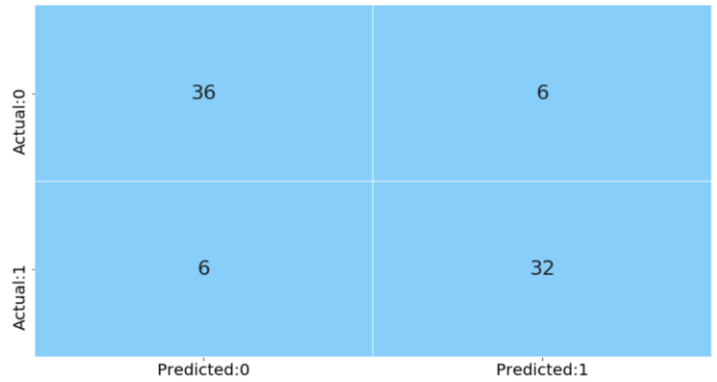
i.e. select the cut-off probability for which the (TPR - FPR) is maximum.

```
# create a dataframe to store the values for false positive rate, true positive rate and threshold
youdens_table = pd.DataFrame({'TPR': tpr,
                              'FPR': fpr,
                              'Threshold': thresholds})

# calculate the difference between TPR and FPR for each threshold and store the values in a new column 'Difference'
youdens_table['Difference'] = youdens_table.TPR - youdens_table.FPR

# sort the dataframe based on the values of difference
# 'ascending = False' sorts the data in descending order
# 'reset_index' resets the index of the dataframe
# 'drop = True' drops the previous index
youdens_table = youdens_table.sort_values('Difference', ascending = False).reset_index(drop = True)

# print the first five observations
youdens_table.head()
```

	TPR	FPR	Threshold	Difference
0	0.868421	0.142857	0.618555	0.725564
1	0.763158	0.047619	0.841912	0.715539
2	0.710526	0.023810	0.860956	0.686717
3	0.921053	0.238095	0.380299	0.682957
4	0.710526	0.047619	0.858806	0.662907

As we can see that the optimal cut-off probability is approximately 0.62. Let us consider this cut-off to predict the target values. i.e. if 'y_pred_prob' is less than 0.62, then consider it to be 0 else consider it to be 1.

```
# convert probabilities to 0 and 1 using 'if_else'
y_pred_youden = [ 0 if x < 0.62 else 1 for x in y_pred_prob]
```




```
# calculate various performance measures
acc_table = classification_report(y_test, y_pred_youden)

# print the table
print(acc_table)
```

```
              precision    recall  f1-score   support

           0       0.86      0.86      0.86        42
           1       0.84      0.84      0.84        38

    accuracy                           0.85        80
   macro avg       0.85      0.85      0.85        80
weighted avg       0.85      0.85      0.85        80
```

Interpretation: From the above output, we can see that the model with cut-off = 0.62, is 85% accurate. The specificity and the sensitivity are nearly balanced.

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

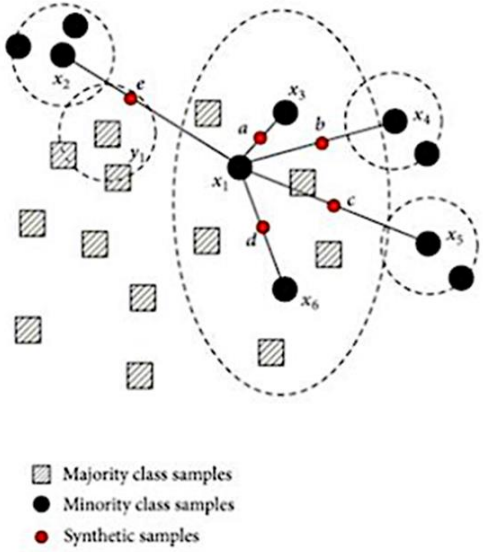
SMOTE

- SMOTE (Synthetic Minority Oversampling Technique) is one of the most used techniques to deal with an imbalanced data
- It generates the synthetic samples for the minority class (a class with fewer observations)
- Python provides a library that provides different techniques to deal with an imbalanced dataset

```
# install imbalanced-learn using jupyter notebook
# !pip install imbalanced-learn

# import SMOTE from imblearn.over_sampling
from imblearn.over_sampling import SMOTE
```

SMOTE Highlights

- SMOTE stands for Synthetic Minority Oversampling Technique. At the heart of SMOTE, the construction of Minority Class Exists.
- Creates Artificial Observations based on feature space similar to minority samples. In simple words, it generates random set of minority class observations to shift the learning bias.
- The SMOTE Samples are the linear combination of two similar samples from the minority class.
- The minority class is sampled and synthetic samples are introduced along with the line segment joining any/all of the k minority class nearest neighbors.
- K is randomly chosen depending on the amount of Over Sampling Required.
- **Doesn't work well with High Dimensional Datasets**. In Low Dimensional Data, the simple under-sampling tends to outperform SMOTE.
- Combination of SMOTE and boosting improves the prediction performance of rare classes.



Imbalanced Data – Under Sampling Vs Over

What Resampling Does?

Balance the classes by increasing minority or decreasing the majority.

What is Random Under-Sampling

1. Randomly Removing majority class observations.
2. Helps Balance the Dataset.
3. Discarded observations could have important information.
4. May Lead to Bias.

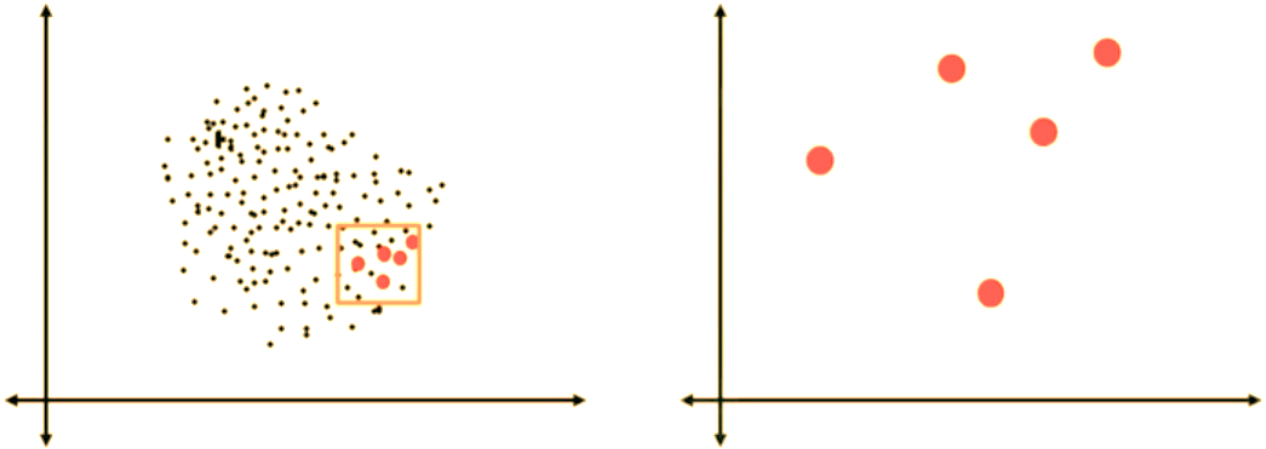
What is Random Over-Sampling

1. Randomly Adds minority class observations.
2. No Information Loss
3. Prone to Overfitting due to copying the same data.

Total Observations	1000
Fraudlent	10 or 1%
Normal Transactions	990
Reduce Normal to	90
Fraudlent Transactions	10 or 10%

Total Observations	1000
Fraudlent	10 or 1%
Normal Transactions	990
Increase fraudlent by	100
Fraudlent Transactions	10%

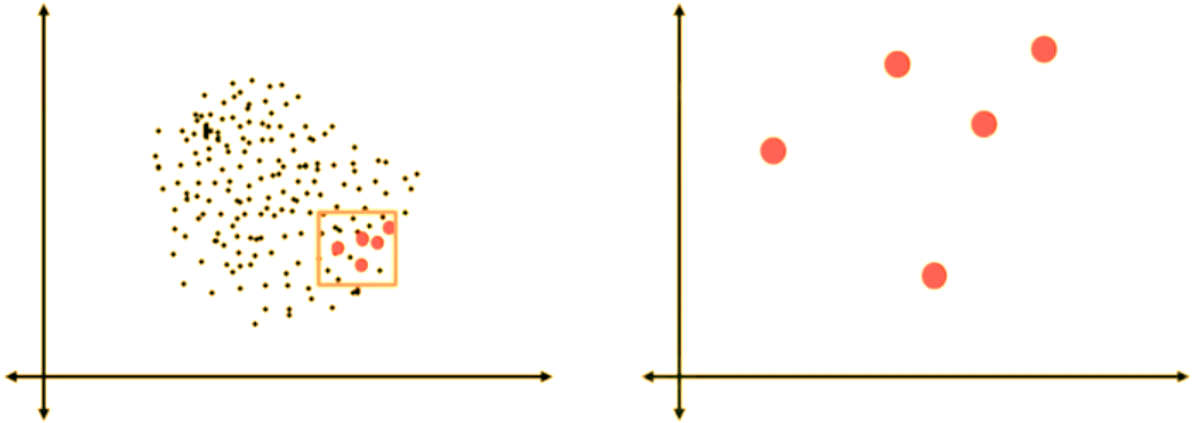
SMOTE Process



- First Step – Plot the Data Points in Two Dimensions and Zoom on the Minority Class

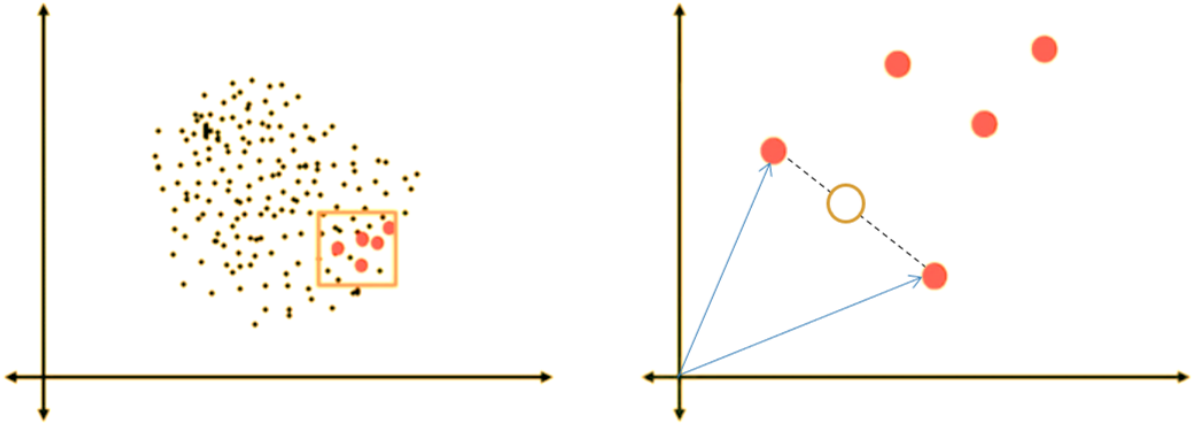
- Identify the Feature Vectors and Its Nearest Neighbors.

SMOTE Process



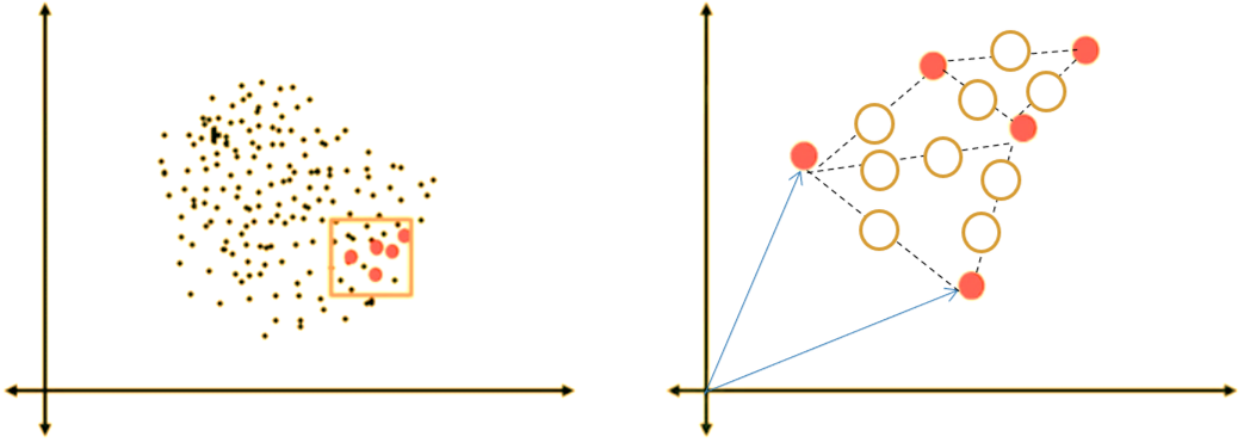
- First Step – Plot the Data Points in Two Dimensions and Zoom on the Minority Class
- Identify the Feature Vectors and Its Nearest Neighbors.
- Let’s Calculate the Linear Distance of these points and multiply it with a random no between 0 & 1

SMOTE Process



- First Step – Plot the Data Points in Two Dimensions and Zoom on the Minority Class
- Identify the Feature Vectors and Its Nearest Neighbors.
- Lets Calculate the Linear Distance of these points and multiply it with a random no between 0 & 1.
- We then plot a new data points (Orange Circle) on this line (rep by orange circle) is our new data point.

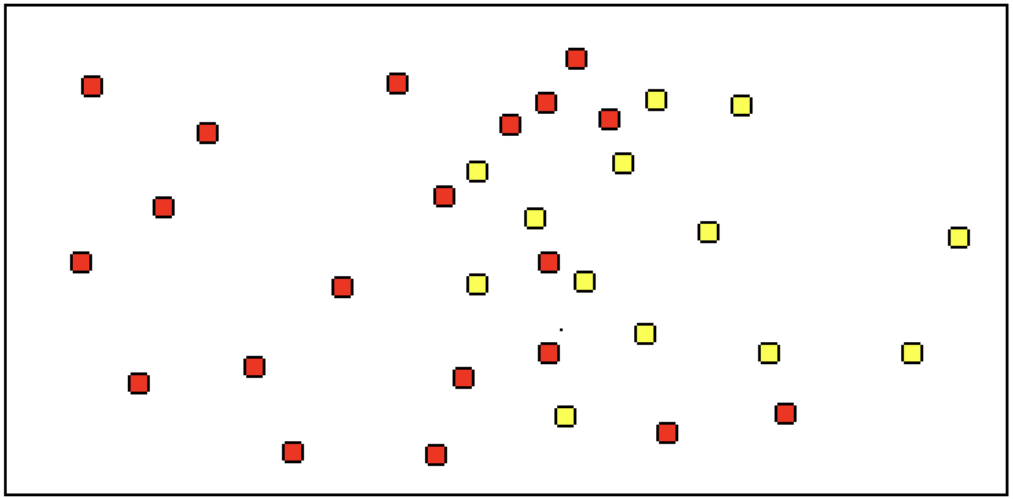
SMOTE Process



- First Step – Plot the Data Points in Two Dimensions and Zoom on the Minority Class
- Identify the Feature Vectors and Its Nearest Neighbors.
- Let's Calculate the Linear Distance of these points and multiply it with a random no between 0 & 1.
- We then plot a new data points (Orange Circle) on this line. This New Point (rep by orange circle) is our new data point.
- We can continue this and keep on adding synthetic (orange circles) points depending on how many data points we are asked to create. We have now managed to create 09 synthetic points from 5 points making total of 14 points.

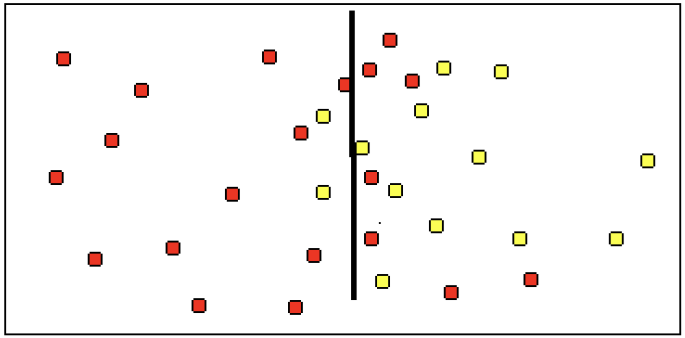
Improper fit

Classify the following data by drawing a line or curve

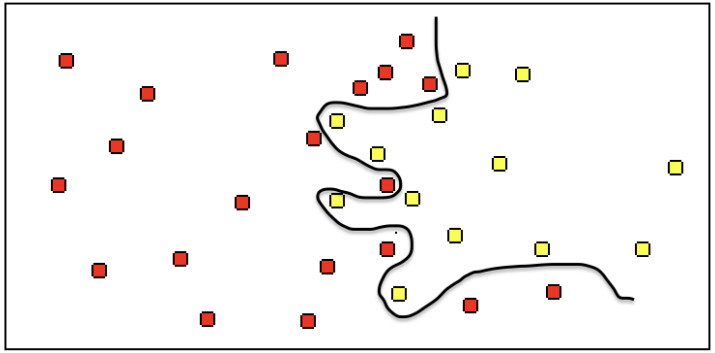


Improper fit

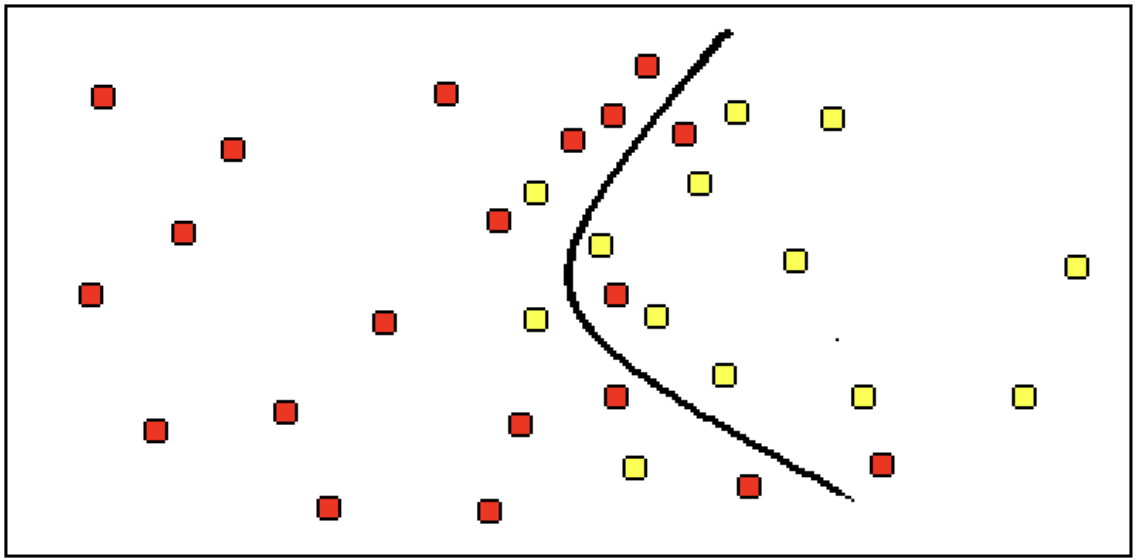
Underfitting



Overfitting



Good fit



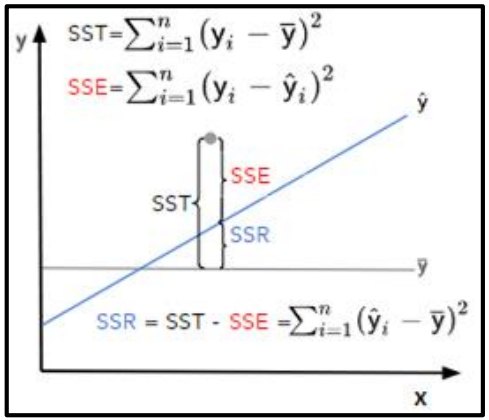
Appendix

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 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, β > 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, β > 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 (LRT) 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)

Logistic regression

- Thus for a binary logistic classification where the target variable takes two values names 0 and 1, we have

Class labels	0	1
P[Y=y | X]	1-π(x)	π(x)

- π(x) denotes the probability that the response is present for the records for some combination of values that the independent variables take, i.e. for X=x

- 1-π(x) denotes the probability that the response is absent for the records for some combination of values that the independent variables take, i.e. for X=x

Linearization

- To estimate the parameter we need to linearize the function. We use the logit transformation

$$\eta = \ln \frac{\pi}{1-\pi} \qquad \pi(x) = \frac{e^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}}{1 + e^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}}$$

- The ratio π/(1-π) is called the odds

- Hence the logit transformation is also known as the log-odds

Linearization

We have,

$$\frac{\pi(x)}{1-\pi(x)} = \frac{\frac{\exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}}{1+\exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}}}{1-\frac{\exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}}{1+\exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}}} = \frac{\frac{\exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}}{1+\exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}}}{\frac{1}{1+\exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}}} = \frac{\exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}}{1}$$

That is,

$$\frac{\pi(x)}{1-\pi(x)} = \exp^{\beta_0+\sum_{i=1}^{n}\beta_i x_i}$$

Linearization

Taking natural log on both sides we have,

$$\ln \frac{\pi(x)}{1-\pi(x)} = \ln \exp^{\beta_0 + \sum_{i=1}^{n} \beta_i x_i}$$

$$\ln \frac{\pi(x)}{1-\pi(x)} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

Thus, we have a linear relationship.

Interpreting the parameter

- The logistic regression is given by

$$\ln \frac{\pi(x)}{1-\pi(x)} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

- In a linear regression model, β_1 gives the average change in Y associated with a one-unit increase in X

- Whereas, in a logistic regression model, increasing X by one unit changes the log odds by β_1, or equivalently it multiplies the odds by $\exp\{\beta_1\}$

Interpreting for our example

- **The logistic regression is given by**

$$\ln \frac{\pi(x)}{1-\pi(x)} = 0.142 + 1.0018 \ \text{BKT kg/ha}$$

- Increasing BKT kg/ha by one unit changes the log odds by 1.0018

OR

- Increasing BKT kg/ha by one unit multiplies the odds by exp{1.0018}

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

Maximum likelihood estimation

- The method of Maximum Likelihood Estimation (MLE) is a method of estimating the parameter of a function by maximizing the likelihood function
- The likelihood function is the joint probability density function of the sample
- For binomial logistic regression the data follows the bernoulli distribution. So the probability distribution is given by

$$f(y_i) = \pi_i^{y_i} (1 - \pi_i)^{1-y_i}$$

where y_i takes value 0 or 1 and π_i is the probability

Maximum likelihood estimation

- The likelihood function (L) is given as

$$L = \prod_{i=1}^{n} f(y_i) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1-y_i}$$

since we have an independent sample, L is the product of all probabilities

- Taking log on both sides, we have

$$\ln L = \ln \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1-y_i}$$

$$\ln L = \sum_{i=1}^{n} y_i \ln \pi_i + \sum_{i=1}^{n} (1 - y_i) \ln(1 - \pi_i)$$

Maximum likelihood estimation

- Taking log on both sides, we have

$$\ln L = \sum_{i=1}^{n} y_i \pi_i + \sum_{i=1}^{n} (1 - y_i)(1 - \pi_i)$$

- So solving, we have

$$\ln L = \sum_{i=1}^{n} \left[y_i \ln \frac{\pi_i}{1-\pi_1} \right] + \sum_{i=1}^{n} \ln(1 - \pi_i)$$

- This equation is further solved by numerical method such as the Newton-Raphson method in order to get the estimates

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

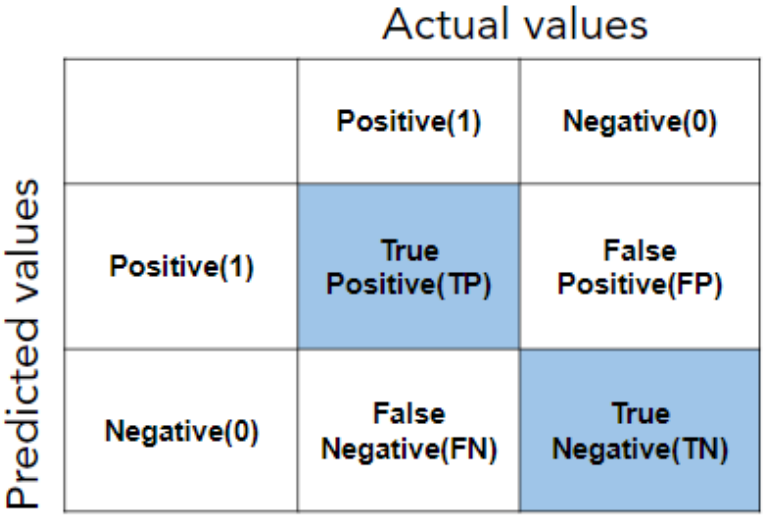
1. Calculate $P(A \cap B)_{negative}$

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

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

2. Calculate kappa statistic from p_o and p_e

Calculation of p_o

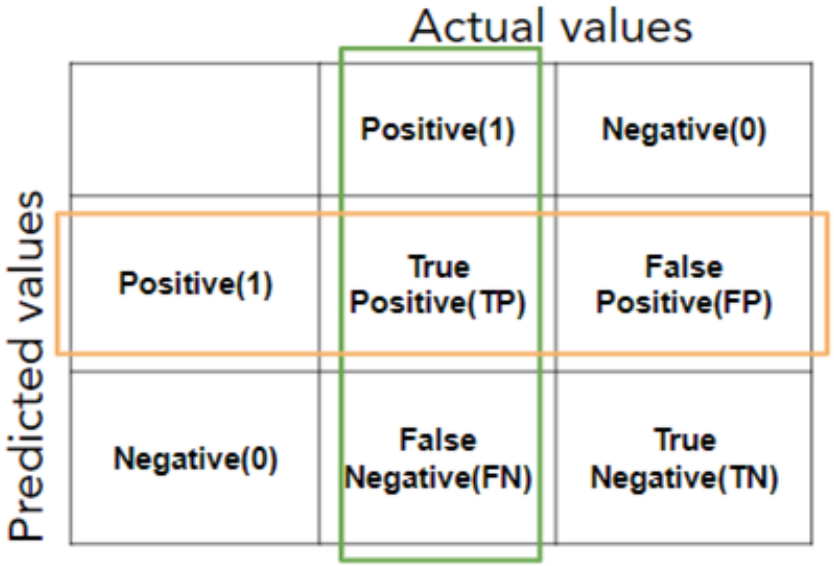


p_o is the observed agreement i.e when the actual and predicted values match

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

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

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

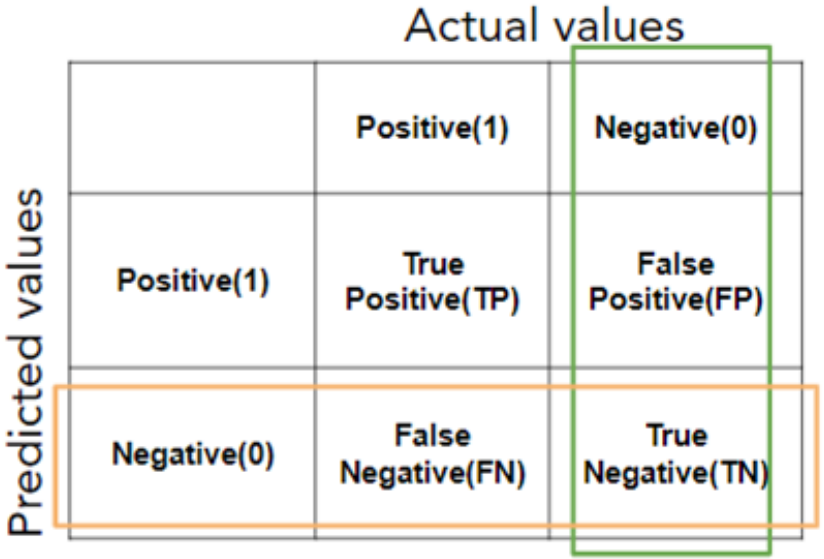


Since A and B are independent events:

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

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

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

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

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

$$
\begin{aligned}
H(y) &= -\sum_i \mathrm{y}_{act(i)} \ln\left(\mathrm{y}_{pred(i)}\right) \\
&= -\mathrm{y}_{act(1)} \ln\left(\mathrm{y}_{pred(1)}\right) - \mathrm{y}_{act(0)} \ln\left(\mathrm{y}_{pred(0)}\right) \\
&= -1 \cdot \ln(0.72) - 0 \cdot \ln(0.28) \\
&= -(-0.3285) = 0.33
\end{aligned}
$$

Thank You