Machine Learning Lecture Sheet

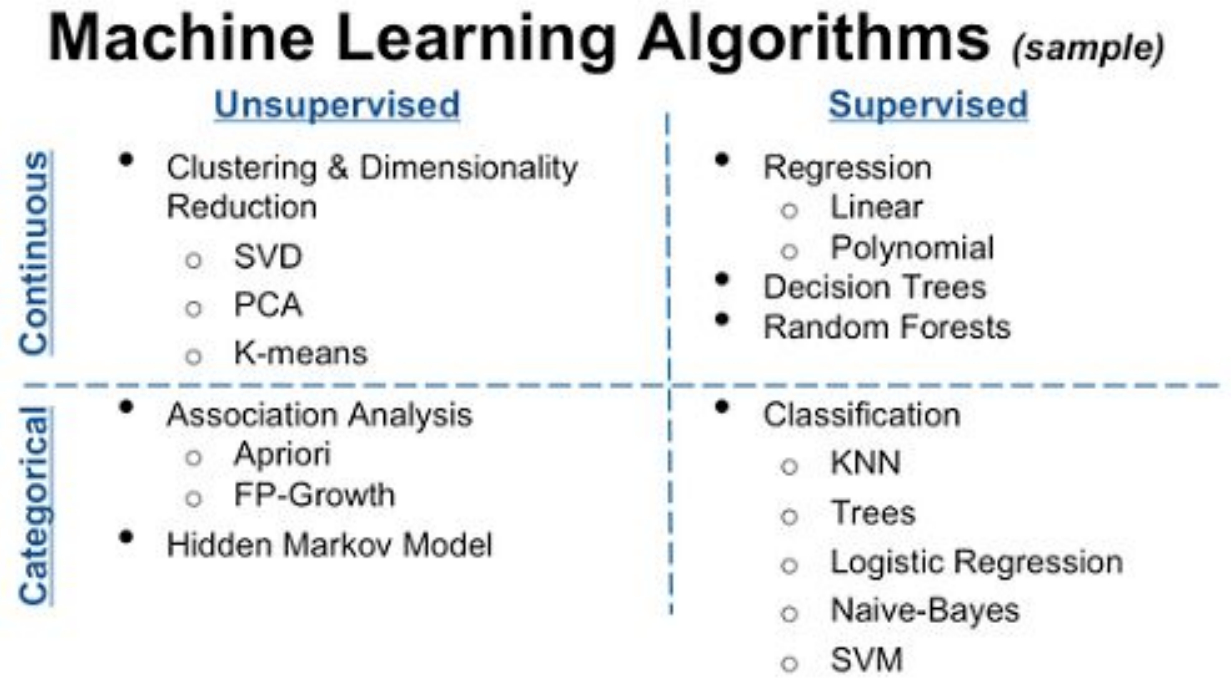
Difference between Machine learning and Artificial Intelligence

Artificial Intelligence and Machine Learning are the terms of computer science. This article discusses some points on the basis of which we can differentiate between these two terms.

Overview

Artificial Intelligence : The word Artificial Intelligence comprises of two words "Artificial" and "Intelligence". Artificial refers to something which is made by human or non natural thing and Intelligence means ability to understand or think. There is a misconception that Artificial Intelligence is a system, but it is not a system .AI is implemented in the system. There can be so many definition of AI, one definition can be *"It is the study of how to train the computers so that computers can do things which at present human can do better."*Therefore It is a intelligence where we want to add all the capabilities to machine that human contain.

Machine Learning : Machine Learning is the learning in which machine can learn by its own without being explicitly programmed. It is an application of AI that provide system the ability to automatically learn and improve from experience. Here we can generate a program by integrating input and output of that program. One of the simple definition of the Machine Learning is *"Machine Learning is said to learn from experience E w.r.t some class of task T and a performance measure P if learners performance at the task in the class as measured by P improves with experiences."*



- **What is Supervised Learning?**

Supervised learning is one of the methods associated with machine learning which involves allocating labeled data so that a certain pattern or function can be deduced from that data. It is worth noting that supervised learning involves allocating an input object, a vector, while at the same time anticipating the most desired output value, which is mostly referred to as the supervisory signal. The bottom line property of supervised learning is that the input data is known and labeled appropriately.

- **What is Unsupervised Learning?**

Unsupervised learning is the second method of machine learning algorithm where inferences are drawn from unlabeled input data. The goal of unsupervised learning is to determine the hidden patterns or grouping in data from unlabeled data. It is mostly used in exploratory data analysis. One of the defining characters of unsupervised learning is that both the input and output are not known.

TABLE SHOWING DIFFERENCES BETWEEN SUPERVISED LEARNING AND UNSUPERVISED LEARNING: COMPARISON CHART

	Supervised Learning	**Unsupervised Learning**
Input Data	Uses Known and Labeled Input Data	Uses Unknown Input Data
Computational Complexity	Very Complex in Computation	Less Computational Complexity
Real Time	Uses off-line analysis	Uses Real Time Analysis of Data
Number of Classes	Number of Classes is Known	Number of Classes is not Known
Accuracy of Results	Accurate and Reliable Results	Moderate Accurate and Reliable Results

Correlation is a measure of association between two variables. The variables are not designated as dependent or independent.

Discrete data can only take particular values. There may potentially be an infinite number of those values, but each is distinct and there's no grey area in between. Discrete data can be numeric -- like numbers of apples -- but it can also be categorical -- like red or blue, or male or female, or good or bad.

Continuous data are not restricted to defined separate values, but can occupy any value over a continuous range. Between any two continuous data values there may be an infinite number of others. Continuous data are always essentially numeric.

Independent and Dependent Variables

Independent variable

A variable whose value does not change by the effect of other variables and is used to manipulate the dependent variable. It is often denoted as **X**.

Dependent variable

A variable whose value change when there is any manipulation in the values of independent variables. It is often denoted as **Y**.

In our example:

Rainfall - Independent variable

Crop yield depends on the amount of rainfall received

Crop yield - Dependent variable

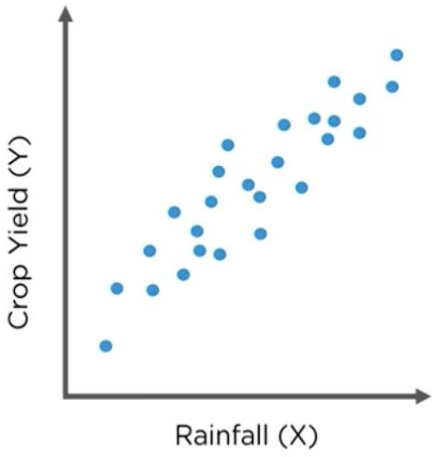
Linear Regression is a statistical model used to predict the relationship between independent and dependent variables.

Examine 2 factors

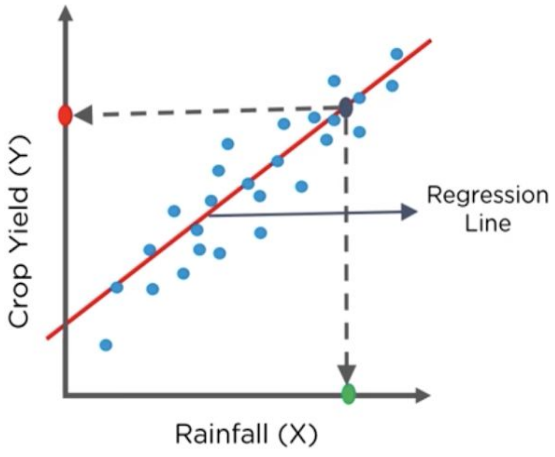
1. Which variables in particular are significant predictors of the outcome variables?

2. How significant is the Regression line to make predictions with highest possible accuracy

Prediction using the Regression line



Plotting the amount of Crop Yield based on the amount of Rainfall

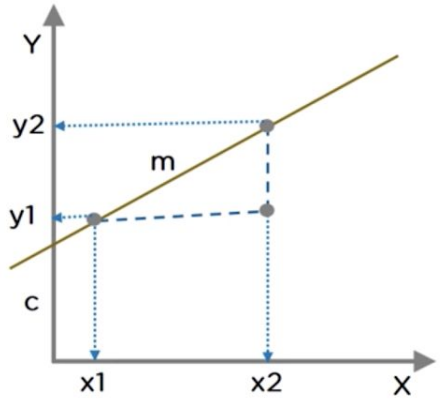


The Red point on the Y axis is the amount of Crop Yield you can expect for some amount of Rainfall (X) represented by Green dot

Regression Equation

The simplest form of a simple linear regression equation with one dependent and one independent variable is represented by:

$$y = m * x + c$$



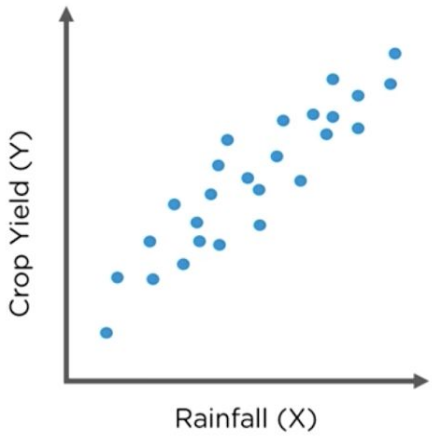
y ---> Dependent Variable

x ---> Independent Variable

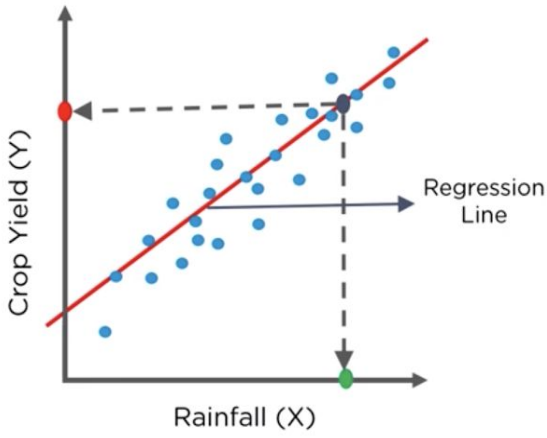
m ---> Slope of the line

$$m = \frac{y2 - y1}{x2 - x1}$$

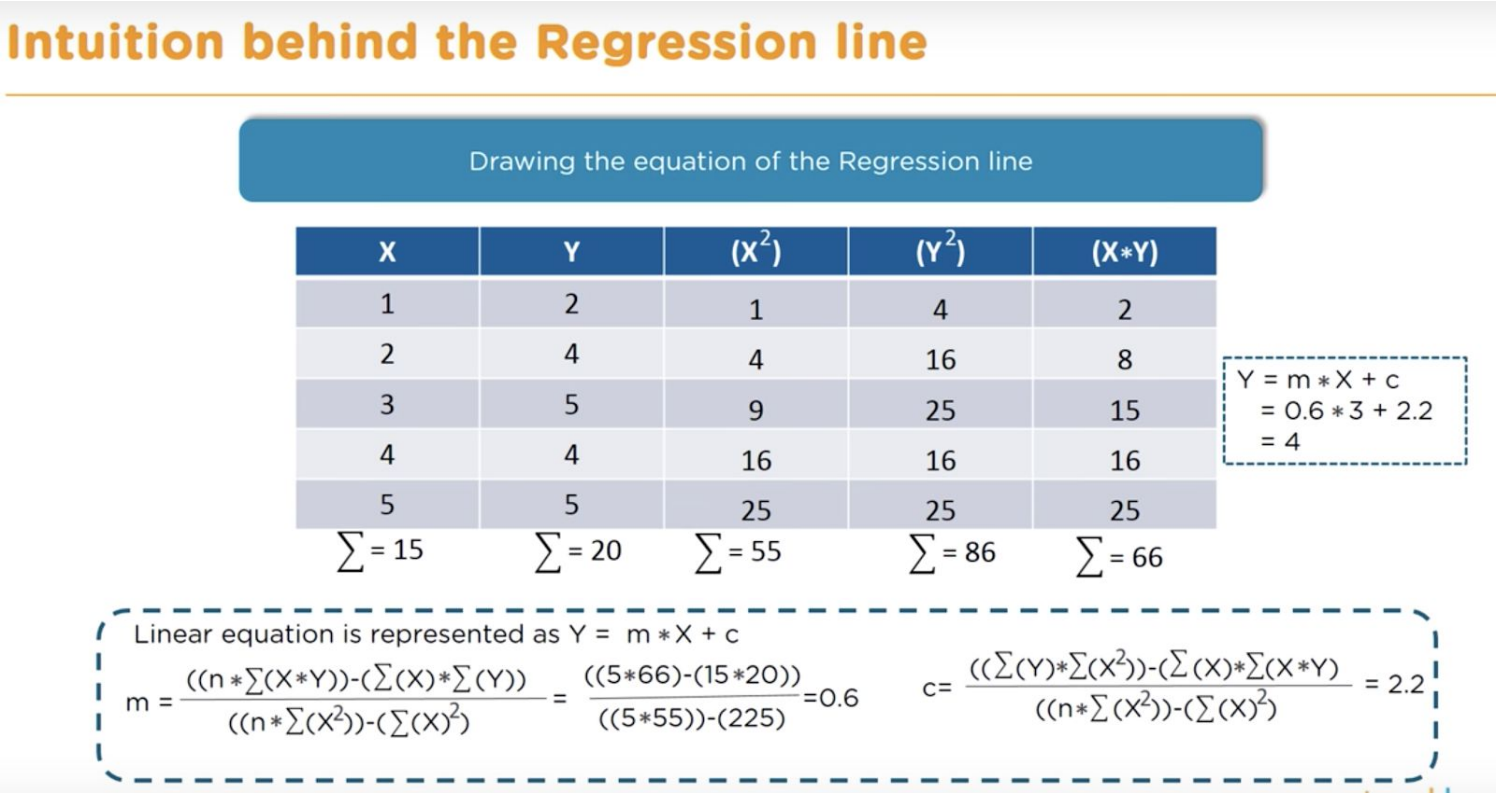
Prediction using the Regression line



Plotting the amount of Crop Yield based on the amount of Rainfall



The Red point on the Y axis is the amount of Crop Yield you can expect for some amount of Rainfall (X) represented by Green dot



Understanding Logistic Regression in Python

Classification techniques are an essential part of machine learning and data mining applications. Approximately 70% of the problems in Data Science are classification problems. There are lots of classification problems that are available, but the logistics regression is common and is a useful regression method for solving the binary classification problem. Another category of classification is Multinomial classification, which handles the issues where multiple classes are present in the target variable. For example, IRIS dataset a very famous example of multi-class classification. Other examples are the classifying article/blog/document category.

Logistic Regression can be used for various classification problems such as spam detection. Diabetes prediction, if a given customer will purchase a particular product or will they churn another competitor, whether the user will click on a given advertisement link or not, and many more examples are in the bucket.

Logistic Regression is one of the most simple and commonly used Machine Learning algorithms for two-class classification. It is easy to implement and can be used as the baseline for any binary classification problem. Its basic fundamental concepts are also constructive in deep learning. Logistic regression describes and estimates the relationship between one dependent binary variable and independent variables.

In this tutorial, you will learn the following things in Logistic Regression:

- Introduction to Logistic Regression
- Linear Regression Vs. Logistic Regression
- Maximum Likelihood Estimation Vs. Ordinary Least Square Method
- How do Logistic Regression works?
- Model building in Scikit-learn
- Model Evaluation using Confusion Matrix.
- Advantages and Disadvantages of Logistic Regression

Logistic Regression

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.
It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.
Linear Regression Equation:

$$y = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots\ldots + \beta nXn$$

Where, y is dependent variable and x1, x2 ... and Xn are explanatory variables.

Sigmoid Function:

$$p = 1/1 + e^{-y}$$

Apply Sigmoid function on linear regression:

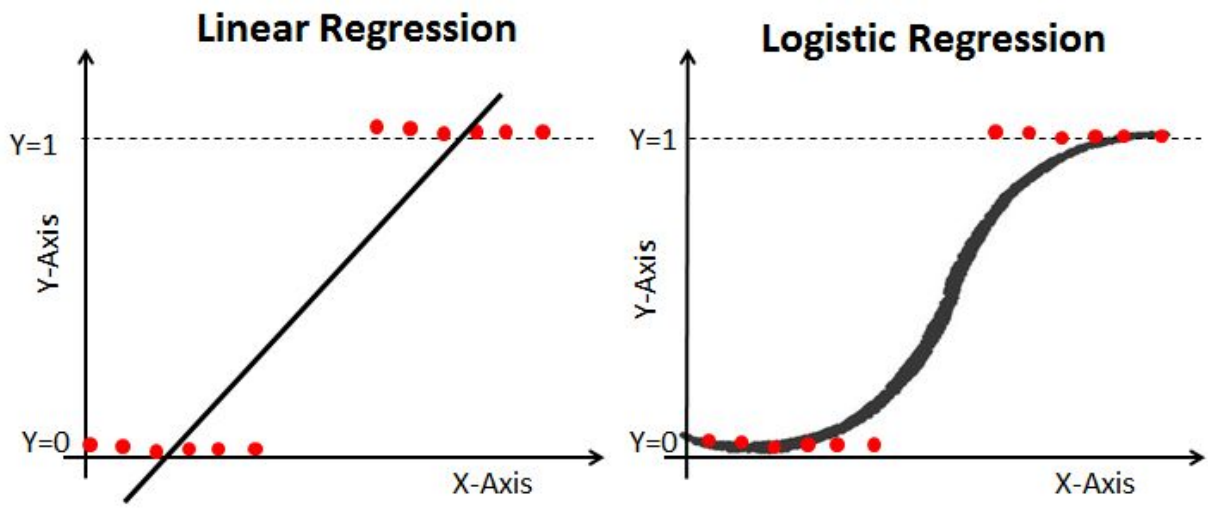
$$p = 1/1 + e^{-(\beta 0 + \beta 1X1 + \beta 2X2 \ldots\ldots \beta nXn)}$$

Properties of Logistic Regression:

- The dependent variable in logistic regression follows Bernoulli Distribution.
- Estimation is done through maximum likelihood.
- No R Square, Model fitness is calculated through Concordance, KS-Statistics.

Linear Regression Vs. Logistic Regression

Linear regression gives you a continuous output, but logistic regression provides a constant output. An example of the continuous output is house price and stock price. Example's of the discrete output is predicting whether a patient has cancer or not, predicting whether the customer will churn. Linear regression is estimated using Ordinary Least Squares (OLS) while logistic regression is estimated using the Maximum Likelihood Estimation (MLE) approach.



Maximum Likelihood Estimation Vs. Least Square Method

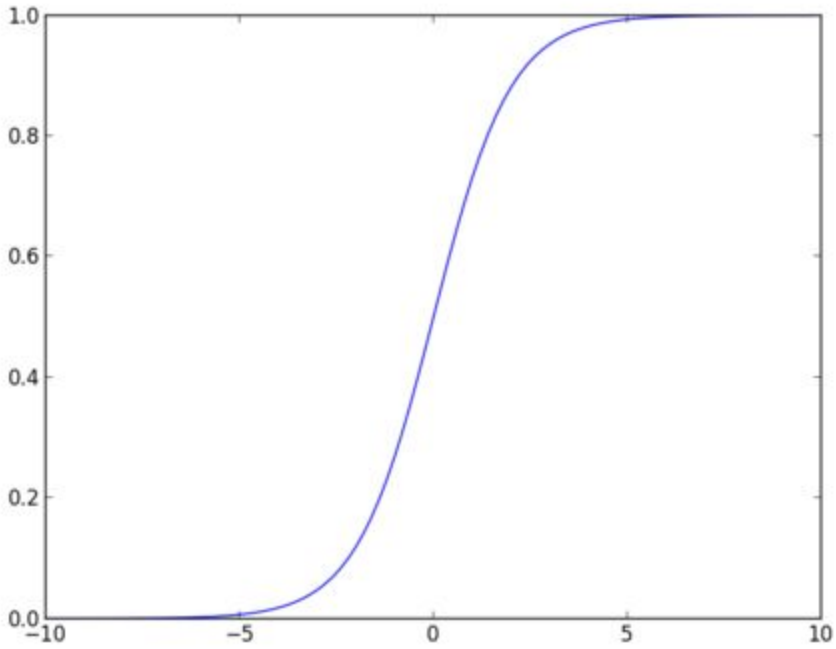
The MLE is a "likelihood" maximization method, while OLS is a distance-minimizing approximation method. Maximizing the likelihood function determines the parameters that are most likely to produce the observed data. From a statistical point of view, MLE sets the mean and variance as parameters in determining the specific parametric values for a given model. This set of parameters can be used for predicting the data needed in a normal distribution.

Ordinary Least squares estimates are computed by fitting a regression line on given data points that has the minimum sum of the squared deviations (least square error). Both are used to estimate the parameters of a linear regression model. MLE assumes a joint probability mass function, while OLS doesn't require any stochastic assumptions for minimizing distance.

Sigmoid Function

The sigmoid function also called the logistic function gives an ‘S’ shaped curve that can take any real-valued number and map it into a value between 0 and 1. If the curve goes to positive infinity, y predicted will become 1, and if the curve goes to negative infinity, y predicted will become 0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it like 0 or NO. The outputcannotFor example: If the output is 0.75, we can say in terms of probability as: There is a 75 percent chance that patient will suffer from cancer.

$$f(x) = \frac{1}{1 + e^{-(x)}}$$



Types of Logistic Regression

Types of Logistic Regression:

- Binary Logistic Regression: The target variable has only two possible outcomes such as Spam or Not Spam, Cancer or No Cancer.
- Multinomial Logistic Regression: The target variable has three or more nominal categories such as predicting the type of Wine.
- Ordinal Logistic Regression: the target variable has three or more ordinal categories such as restaurant or product rating from 1 to 5.

Model building in Scikit-learn

Let's build the diabetes prediction model.
Here, you are going to predict diabetes using Logistic Regression Classifier.
Let's first load the required Pima Indian Diabetes dataset using the pandas' read CSV function. You can download data from the following link:
https://www.kaggle.com/uciml/pima-indians-diabetes-database

Loading Data

```
import pandas as pd
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi',
'pedigree', 'age', 'label']

pima = pd.read_csv("pima-indians-diabetes.csv", header=None,
names=col_names)

pima.head()
```

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Selecting Feature

Here, you need to divide the given columns into two types of variables dependent(or target variable) and independent variable(or feature variables).

```
feature_cols = ['pregnant', 'insulin', 'bmi',
'age','glucose','bp','pedigree']

X = pima[feature_cols] # Features

y = pima.label # Target variable
```

Splitting Data

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.
Let's split dataset by using function train_test_split(). You need to pass 3 parameters features, target, and test_set size. Additionally, you can use random_state to select records randomly.

```
from sklearn.cross_validation import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,r
andom_state=0)
```

Here, the Dataset is broken into two parts in a ratio of 75:25. It means 75% data will be used for model training and 25% for model testing.

Model Development and Prediction

First, import the Logistic Regression module and create a Logistic Regression classifier object using LogisticRegression() function.
Then, fit your model on the train set using fit() and perform prediction on the test set using predict().

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train,y_train)
y_pred=logreg.predict(X_test)
```

Model Evaluation using Confusion Matrix

A confusion matrix is a table that is used to evaluate the performance of a classification model. You can also visualize the performance of an algorithm. The fundamental of a confusion matrix is the number of correct and incorrect predictions are summed up class-wise.

```
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix
```

```
Output:
```

```
array([[119,  11],
       [ 26,  36]])
```

Here, you can see the confusion matrix in the form of the array object. The dimension of this matrix is 2*2 because this model is a binary classification. You have two classes 0 and 1. Diagonal values represent accurate predictions, while non-diagonal elements are inaccurate predictions. In the output, 119 and 36 are actual predictions, and 26 and 11 are incorrect predictions.

Visualizing Confusion Matrix using Heatmap

Let's visualize the results of the model in the form of a confusion matrix using

matplotlib and seaborn.

Here, you will visualize the confusion matrix using Heatmap.

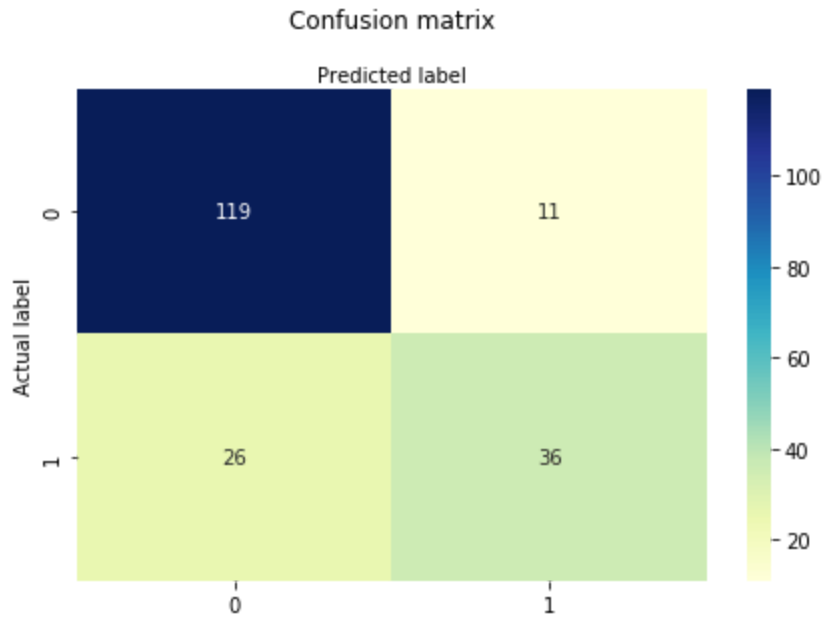
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)

sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu"
,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
```

```
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

Text(0.5,257.44,'Predicted label')
```



Confusion Matrix Evaluation Metrics

Let's evaluate the model using model evaluation metrics such as accuracy, precision, and recall.

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))

Output:
Accuracy: 0.8072916666666666
Precision: 0.7659574468085106
Recall: 0.5806451612903226
```

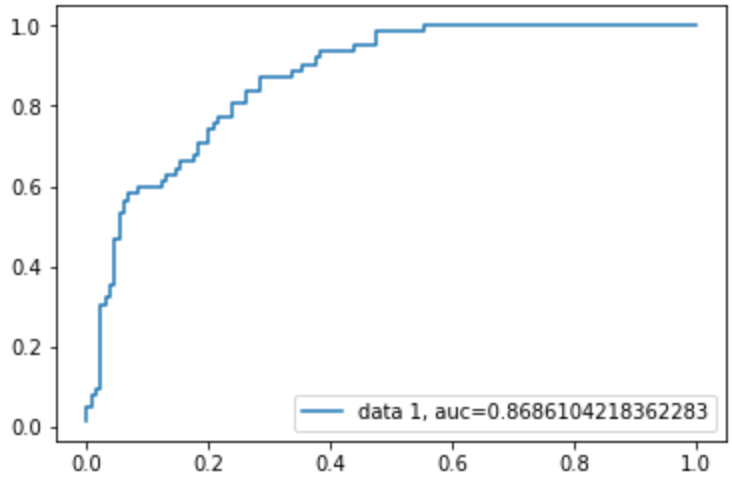
Well, you got a classification rate of 80%, considered as good accuracy.

Precision: Precision is about being precise, i.e., how accurate your model is. In other words, you can say, when a model makes a prediction, how often it is correct. In your prediction case, when your Logistic Regression model predicted patients are going to suffer from diabetes, that patients have 76% of the time.
Recall: If there are patients who have diabetes in the test set and your Logistic Regression model can identify it 58% of the time.

ROC Curve

Receiver Operating Characteristic(ROC) curve is a plot of the true positive rate against the false positive rate. It shows the tradeoff between sensitivity and specificity.

```
y_pred_proba = logreg.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
```



AUC score for the case is 0.86. AUC score 1 represents perfect classifier, and 0.5 represents a worthless classifier.

Advantages

Because of its efficient and straightforward nature, doesn't require high computation power, easy to implement, easily interpretable, used widely by data analyst and scientist. Also, it doesn't require the scaling of features. Logistic regression provides a probability score for observations.

Disadvantages

Logistic regression is not able to handle a large number of categorical features/variables. It is vulnerable to overfitting. Also, can't solve the non-linear problem with the logistic regression that is why it requires a transformation of non-linear features. Logistic regression will not perform well with independent variables that are not correlated to the target variable and are very similar or correlated to each other.

Conclusion

In this tutorial, you covered a lot of details about Logistic Regression. You have learned what the logistic regression is, how to build respective models, how to visualize results and some of the theoretical background information. Also, you covered some basic concepts such as the sigmoid function, maximum likelihood, confusion matrix, ROC curve.

Confusion Matrix: When we get the data, after data cleaning, pre-processing and wrangling, the first step we do is to feed it to an outstanding model and of course, get output in probabilities. But hold on! How in the hell can we measure the effectiveness of our model? Better the effectiveness, better the performance and that are exactly what we want. And it is where the Confusion matrix comes into the limelight. Confusion Matrix is a performance measurement for machine learning classification.

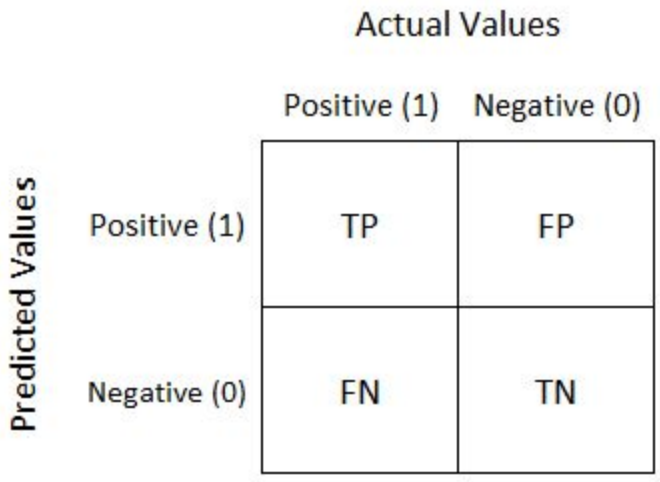
This blog aims to answer the following questions:

1. What the confusion matrix is and why you need it?
2. How to calculate Confusion Matrix for a 2-class classification problem?

Today, let's understand the confusion matrix once and for all.

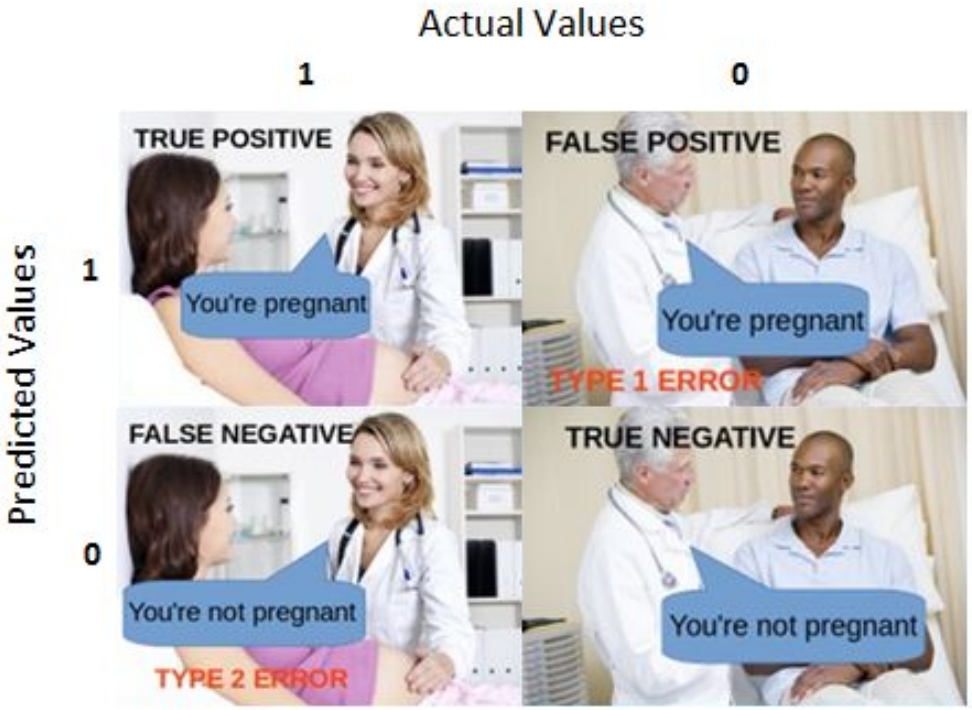
What is the Confusion Matrix and why you need it?

Well, it is a performance measurement for a machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



It is extremely useful for measuring Recall, Precision, Specificity, Accuracy and most importantly AUC-ROC Curve.

Let’s understand TP, FP, FN, TN in terms of pregnancy analogy.



True Positive:

Interpretation: You predicted positive and it’s true.

You predicted that a woman is pregnant and she actually is.

True Negative:

Interpretation: You predicted negative and it’s true.

You predicted that a man is not pregnant and he actually is not.

False Positive: (Type 1 Error)

Interpretation: You predicted positive and it’s false.

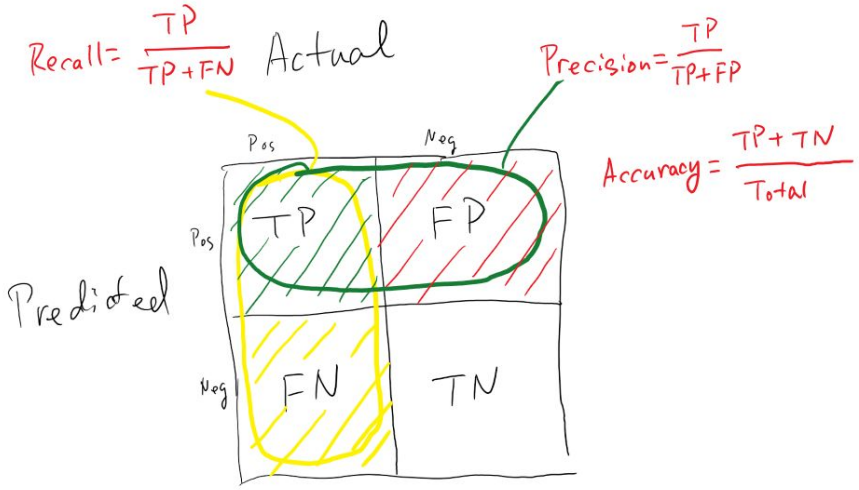
You predicted that a man is pregnant but he actually is not.

False Negative: (Type 2 Error)
Interpretation: You predicted negative and it's false.
You predicted that a woman is not pregnant but she actually is.

Just Remember, We describe predicted values as Positive and Negative and actual values as True and False.

How to Calculate Confusion Matrix for a 2-class classification problem?

y	y pred	output for threshold 0.6	Recall	Precision
0	0.5	0		
1	0.9	1		
0	0.7	1		
1	0.7	1	**1/2**	**4/7**
1	0.3	0		
0	0.4	0		
1	0.5	0		



Let's understand the confusion matrix through math.

Recall

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

Out of all the positive classes, how much we predicted correctly. It should be as high as possible.

Precision

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

Out of all the classes, how much we predicted correctly. It should be as high as possible.

F-measure

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall + Precision}$$

It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score. F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.

I hope I've given you some basic understanding of what exactly is a confusing matrix.