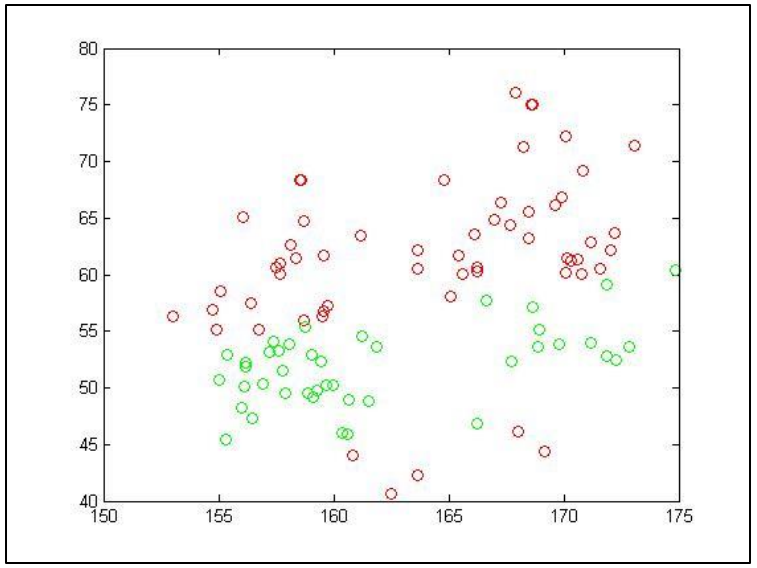
Mahidol University
Wisdom of the Land

BIG DATA PROCESSING

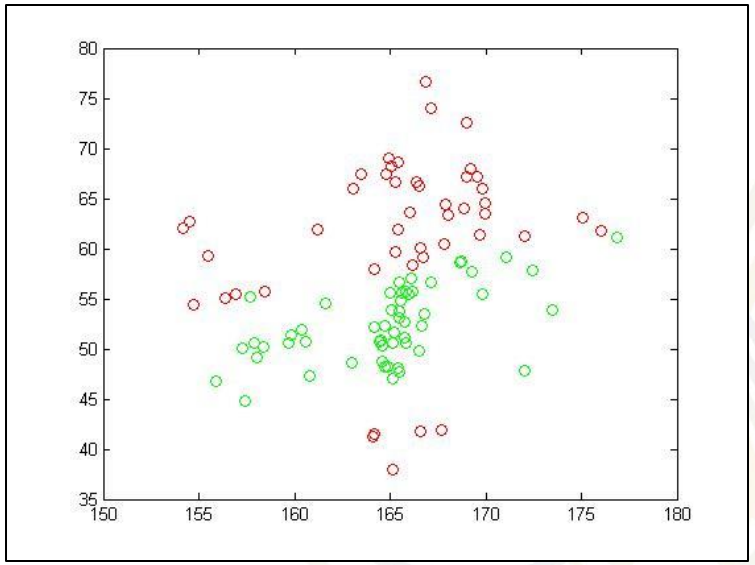
Machine Learning: Evaluations

Evaluation Process

- The evaluation are usually different on training and testing data
- The model that you have created normally will fit the value better
- The test set tell you how the model is actually performed



Train



Test

Mahidol University
Wisdom of the Land

Python for data spilting

- ```
  (trainingData, testData) =
  df.randomSplit([0.8,0.2], seed = 13234 )
  ```
  ```

Mahidol University
Wisdom of the Land

Error Rate for model performance

- **Sum of square Error** (SSE,ESS)
  - Estimate overall error

$$E = \sum (y - t)^2$$

- **Mean square Error** (MSE)
  - To normalised to the number of data

$$E = \frac{1}{N} \sum (y - t)^2$$

- Root mean square Error **(RMS)**
  - To measure the precision more common for regression

$$E = \sqrt{\frac{1}{N} \sum (y - t)^2}$$

Mahidol University
Wisdom of the Land

Regression Evaluators

- Define choice of evaluation

```
from pyspark.ml.evaluation import RegressionEvaluator
evaluator = RegressionEvaluator( labelCol="label",
predictionCol="prediction", metricName="rmse")
```

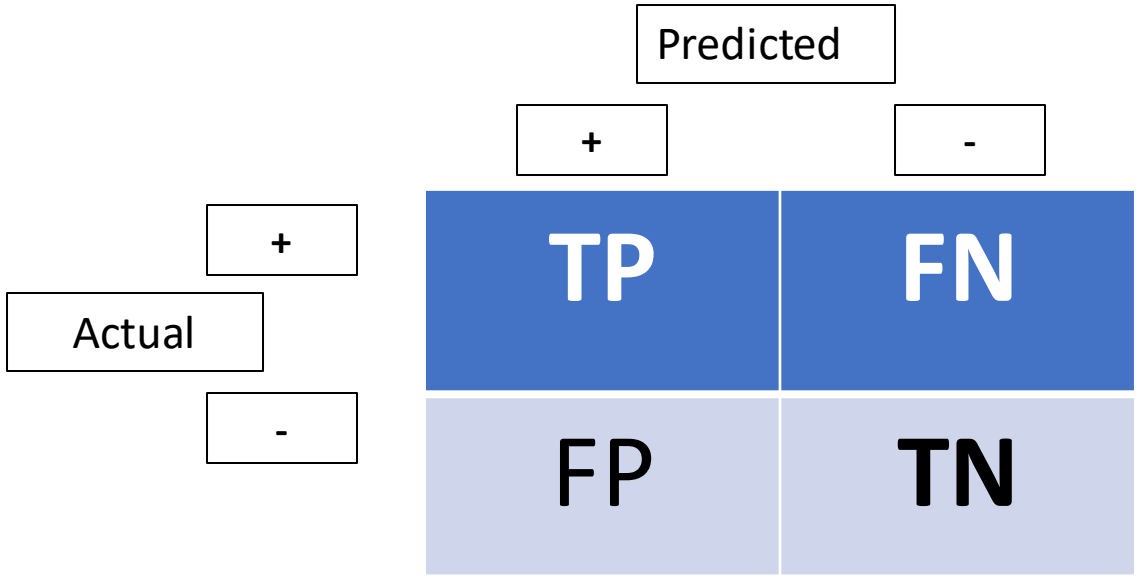
- Display the result from prediction

```
rmse = evaluator.evaluate(predictions)
print("Root Mean Squared Error (RMSE) on test data = %g" % rmse)
```

https://api-docs.databricks.com/python/pyspark/latest/api/pyspark.ml.evaluation.RegressionEvaluator.html

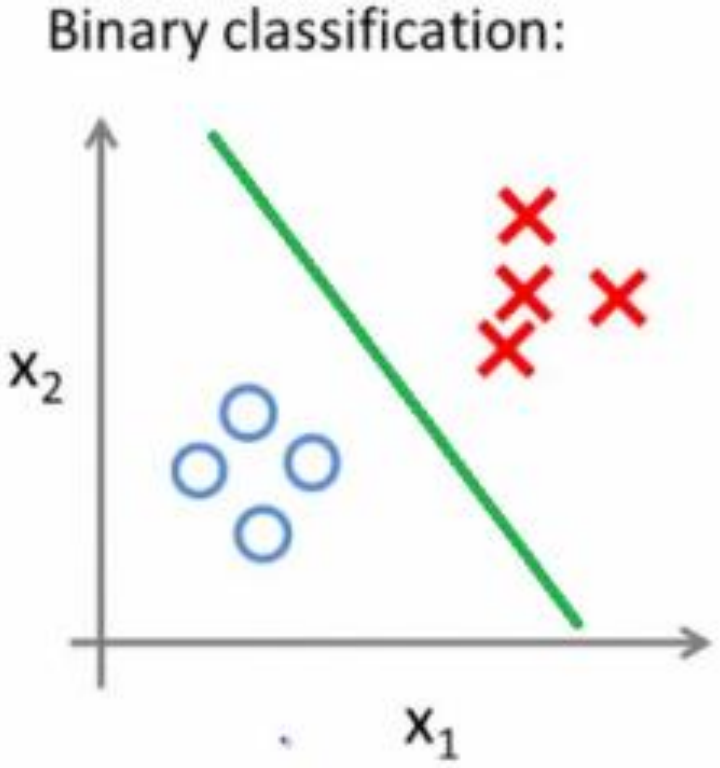
Classification

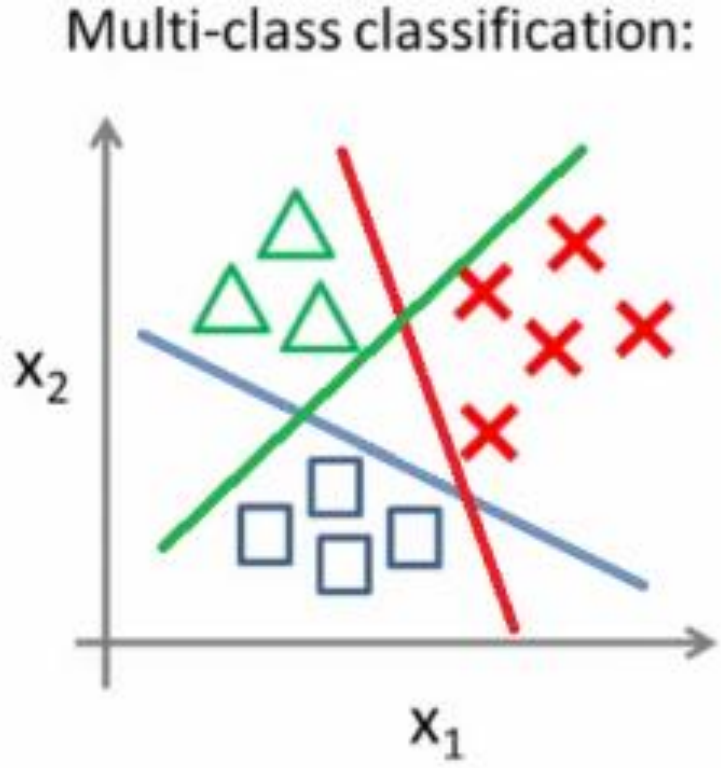
- Classification rate is more common
- Confusion Matrix is usually used



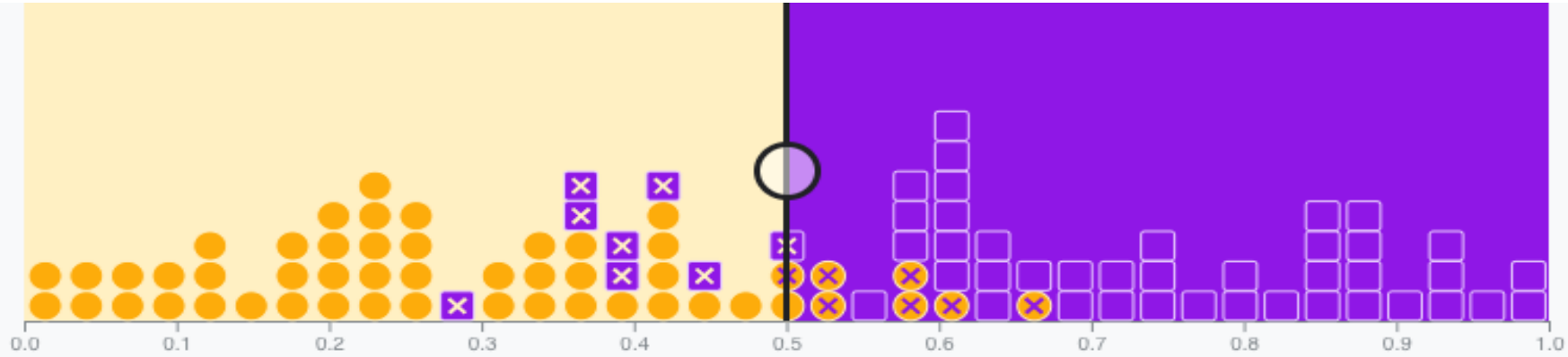
Examples

- Binary Classification
  - Binary output
  - 1 set of parameter output
  - Binary classification
- Multi class Classification
  - Multiclass Output
  - ***C*** sets of paremeter output
  - Covert string -> class





ROC Curve



Threshold: 0.50

Confusion matrix

	Actually positive	Actually negative
Predicted positive	TP=40	FP=7
Predicted negative	FN=8	TN=44

Metrics

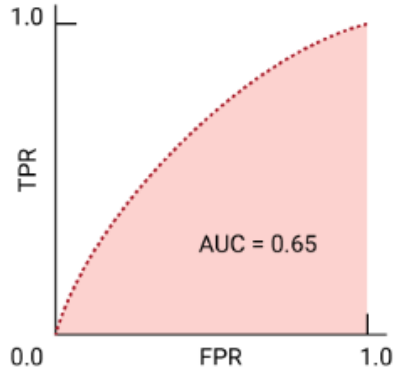
Accuracy	0.85
Precision	0.85
Recall	0.83

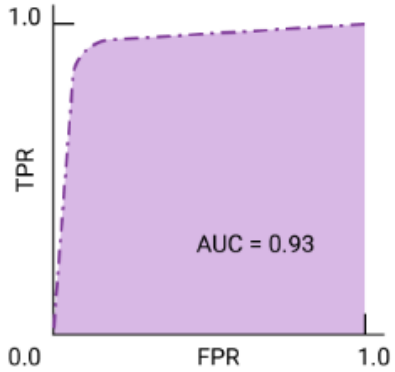
ROC curve

https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Classification: Binary

- Binary Classification
- `from pyspark.ml.evaluation import BinaryClassificationEvaluator`
- `from pyspark.mllib.evaluation import BinaryClassificationMetrics`
- `evaluator = BinaryClassificationEvaluator (rawPredictionCol="rawPrediction", labelCol="Outcome", MetricName="areaUnderROC")`
- `AUC = evaluator.evaluate(predictions)`





https://spark.apache.org/docs/4.0.0/mllib-evaluation-metrics.html#binary-classification

Confusion matrix(expand)

- **Accuracy rate:**

The correct prediction value

- Use only **correctly predicted** data to compute
- From the example:
  (5+6)/20*100=55%

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

Predict / Actual	0	1
0	5	5
1	4	6

Mahidol University
Wisdom of the Land

True Positive/Negative

- **True Positive**
  - The value that is truly positive and being predicted correctly
- **True Positive Rate (sensitivity)**
  - The rate of true positive from all actual positive values
  - **Example** TPR=5 /10 =0.5 or 50% =$\frac{TP}{TP+FN}$

- **True Negative**
  - The value that is truly negative and being predicted correctly
- **True Negative(spacifity)**
  - The rate of true negative from all negative values
  - **Example** TN=6/10 =0.6 or 60%=$\frac{TN}{TN+FP}$

Mahidol University
Wisdom of the Land

False Positive/Negative

- **False Positive**
  - The value that is truly negative and being predicted as positive
- **False Positive Rate**
  - The rate of false positive from all
  - Example FPR=5/10 =0.5 or 50%= $\frac{FP}{FP+TN}$

- **False Negative**
  - The value that is truly positive but being predicted as negative
- **False Negative**
  - The percentage of the incorrectly predicted value as negative
  - Example FN=4/10 =0.4 or 40%=$\frac{FN}{FN+TP}$

Example 2

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Example 3

		Predicted	
Actual		+	-
	+	0	10
	-	100	1000

Mahidol University
Wisdom of the Land

Precision/Recall

- **Precision**
  - Precision is the rate of correctly positive values compared to overall positive prediction
  - **Example** precision (0)=5/9 ,precision (1) =6/11

- **Recall**
  - True positive rates (or true negative rate)
  - **Example** recall(0)=5/10 ,recall(1) =6/10

Average / Weighted average

- Average Precision

  Example precision (0)=5/9 ,precision (1) =6/11

  Average =0.5 * (5/9 + 6/11) =0.5*(0.556+0.5454) =0.55

  Weight Average = (9*0.556+11*0.5454) /20 =0.55

- Average Recall → same as accuracy

  Example recall (0)=5/10 ,recall (1) =6/10

  Average =0.5 * (0.5+0.6) =0.55

  Weight Average = (the same)

Mahidol University
Wisdom of the Land

F-Measure

- F_1 is a harmonic mean between precision and recall
- $F_1 = 2\frac{recall.precision}{recall+precision}$

Mahidol University
Wisdom of the Land

Confusion Matrix 2

Predict / True	0	1	undefined 
0	4	3	4
1	6	4	2

Mahidol University
Wisdom of the Land

Classification: Multiclass

- Binary Classification

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.mllib.evaluation import MulticlassMetrics
evaluator = MulticlassClassificationEvaluator(
labelCol="label", predictionCol="prediction", metricName="accuracy")
        accuracy = evaluator.evaluate(predictions)
print("Accuracy = %g " % (accuracy))
evaluator = MulticlassClassificationEvaluator(metricName="recall")
        recall = evaluator.evaluate(predictions)
print("Recall = %g " % (recall))
```

Mahidol University
Wisdom of the Land

Multi class confusion Matrix

Confusion matrix for multi class problem

		Predicted		
		Cat	**Dog**	**Rabbit**
Actual	Cat	5	2	0
	Dog	3	3	2
	Rabbit	0	1	11

Mahidol University
Wisdom of the Land

Question?

- Accuracy
- Recall
- Precision