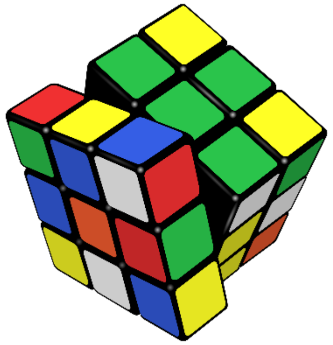


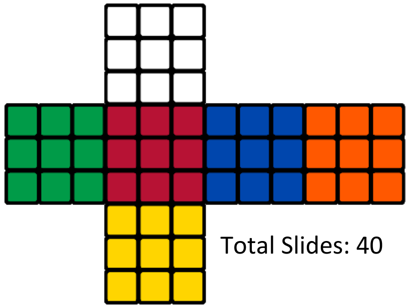
Big Data Engineering For Analytics

Machine Learning on Spark

Dr LIU Fan

(isslf@nus.edu.sg)

NUS-ISS

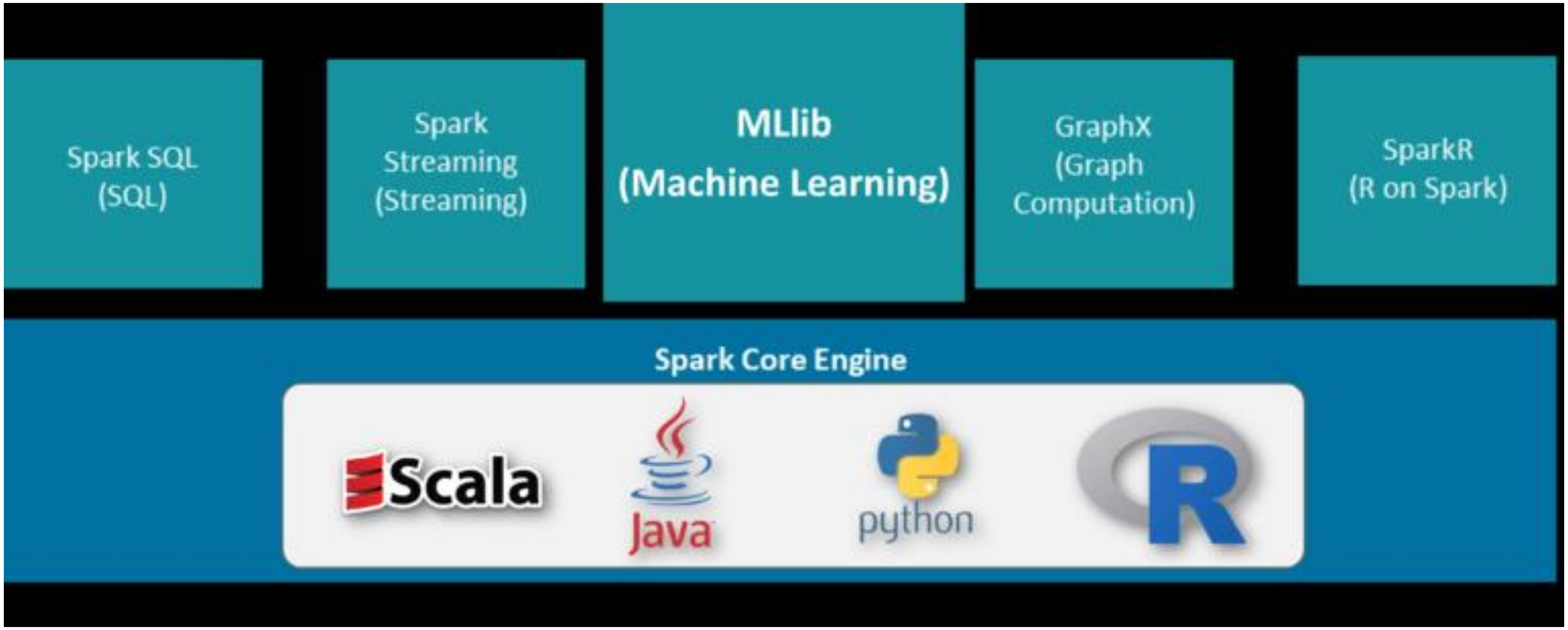


Total Slides: 40

Learning Objectives

- Spark Ecosystem and Machine Learning library on Spark
- Regression
  - ➢Linear Regression
- classification
  - ➢Logistic Regression
- Clustering
  - ➢kmeans

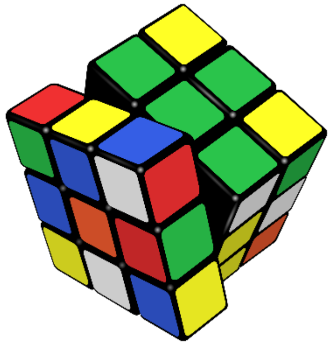
Spark Ecosystem



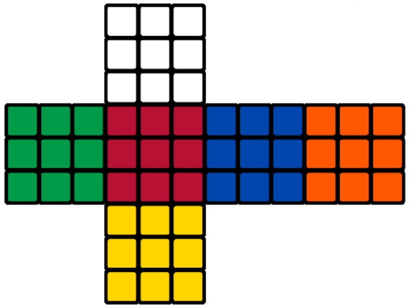
PySpark ML & PySpark MLLib

- PySpark ML
  - ➢DataFrame-based machine learning APIs to let user quickly assemble and configure practical machine learning pipelines.
- PySpark MLlib
  - ➢It is a wrapper over PySpark Core to do data analysis using machine-learning algorithms. It works on distributed systems and is scalable. We can find implementations of classification, clustering, linear regression, and other machine-learning algorithms in PySpark MLlib.

https://www.edureka.co/blog/pyspark-mllib-tutorial/



Big Data Engineering For Analytics



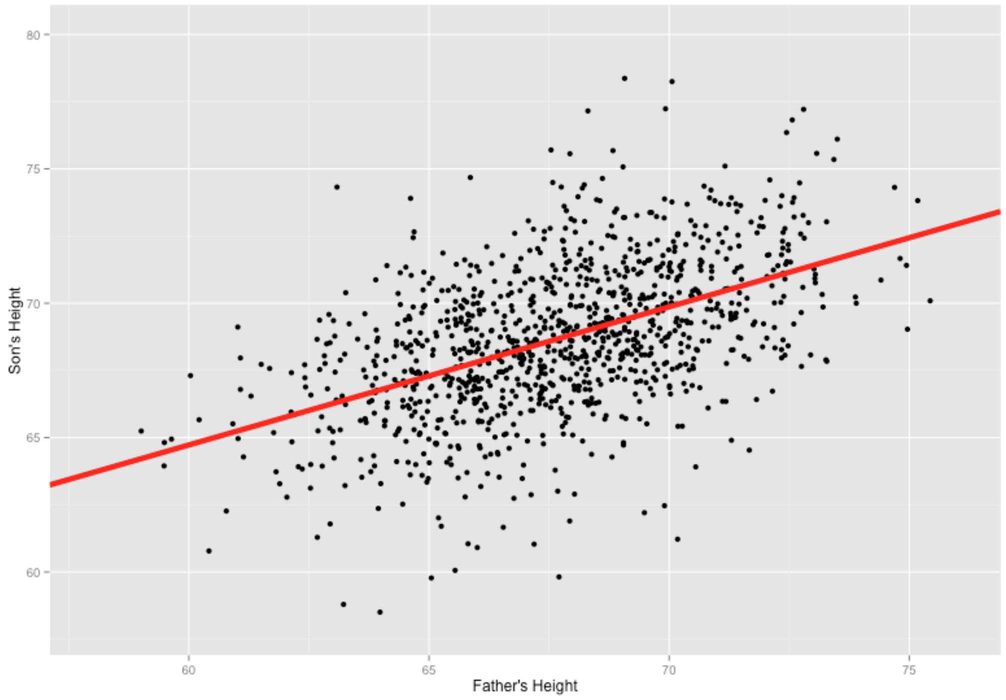
Linear Regression

Example

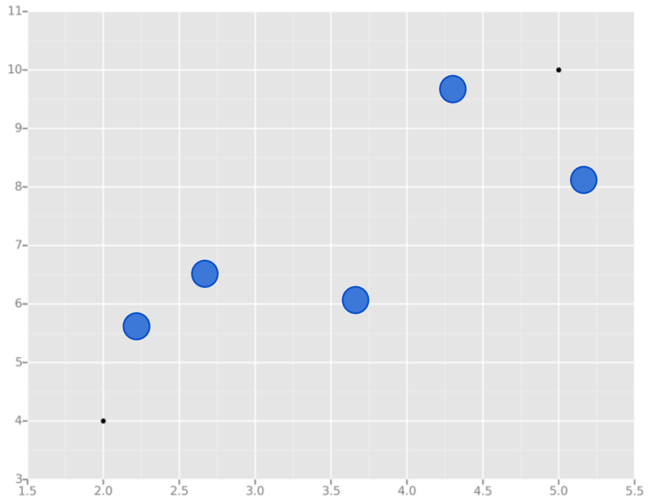
Our goal with linear regression is to **minimize the vertical distance** between all the data points and our line.

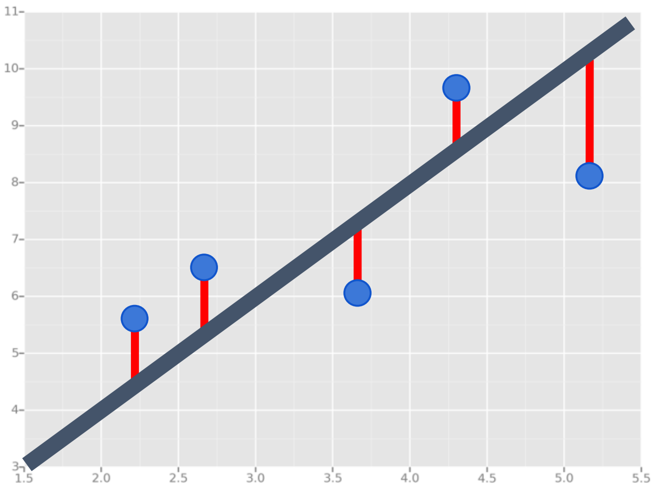
So in determining the **best line**, we are attempting to minimize the distance between **all** the points and their distance to our line.

There are lots of different ways to minimize this, (sum of squared errors, sum of absolute errors, etc), but all these methods have a general goal of minimizing this distance.



Example





For example, one of the most popular methods is the least squares method.

Here we have blue data points along an x and y axis.

We'll use the Least Squares Method, which is fitted by minimizing the ***sum of squares of the residuals.***

The residuals for an observation is the difference between the observation (the y-value) and the fitted line.

Evaluating Regression

- Let’s discuss some of the most common evaluation metrics for regression:
    - Mean Absolute Error
    - Mean Squared Error
    - Root Mean Square Error
    - R Squared Values

Evaluating Regression

- Mean Absolute Error (MAE)
    - This is the mean of the absolute value of errors.
    - Easy to understand, just average error!

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Mean Squared Error (MSE)
    - This is the mean of the squared errors.
    - Larger errors are noted more than with MAE, making MSE more popular.

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Evaluating Regression

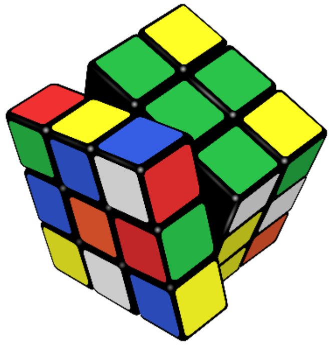
- Root Mean Squared Error (RMSE)
  - This is the root of the mean of the squared errors.
  - Most popular (has same units as y)

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

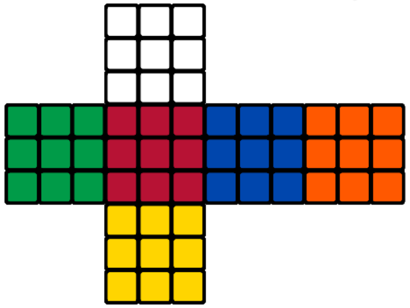
- R Squared Value
  - By itself, R^2, won't tell you the "whole story".
  - In a basic sense it is a measure of how much variance your model accounts for.
  - Between 0-1 (0% to 100%)
  - There are also different ways of obtaining R2 , such as adjusted R squared.

Example

```
from pyspark.ml.regression import LinearRegression

# Load training data
training = spark.read.format("libsvm")\
    .load("sample_linear_regression_data.txt")

lr = LinearRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)

# Fit the model
lrModel = lr.fit(training)

# Print the coefficients and intercept for linear regression
print("Coefficients: %s" % str(lrModel.coefficients))
print("Intercept: %s" % str(lrModel.intercept))

# Summarize the model over the training set and print out some metrics
trainingSummary = lrModel.summary
print("numIterations: %d" % trainingSummary.totalIterations)
print("objectiveHistory: %s" % str(trainingSummary.objectiveHistory))
trainingSummary.residuals.show()
print("RMSE: %f" % trainingSummary.rootMeanSquaredError)
print("r2: %f" % trainingSummary.r2)
```



Big Data Engineering For Analytics



Logistic Regression

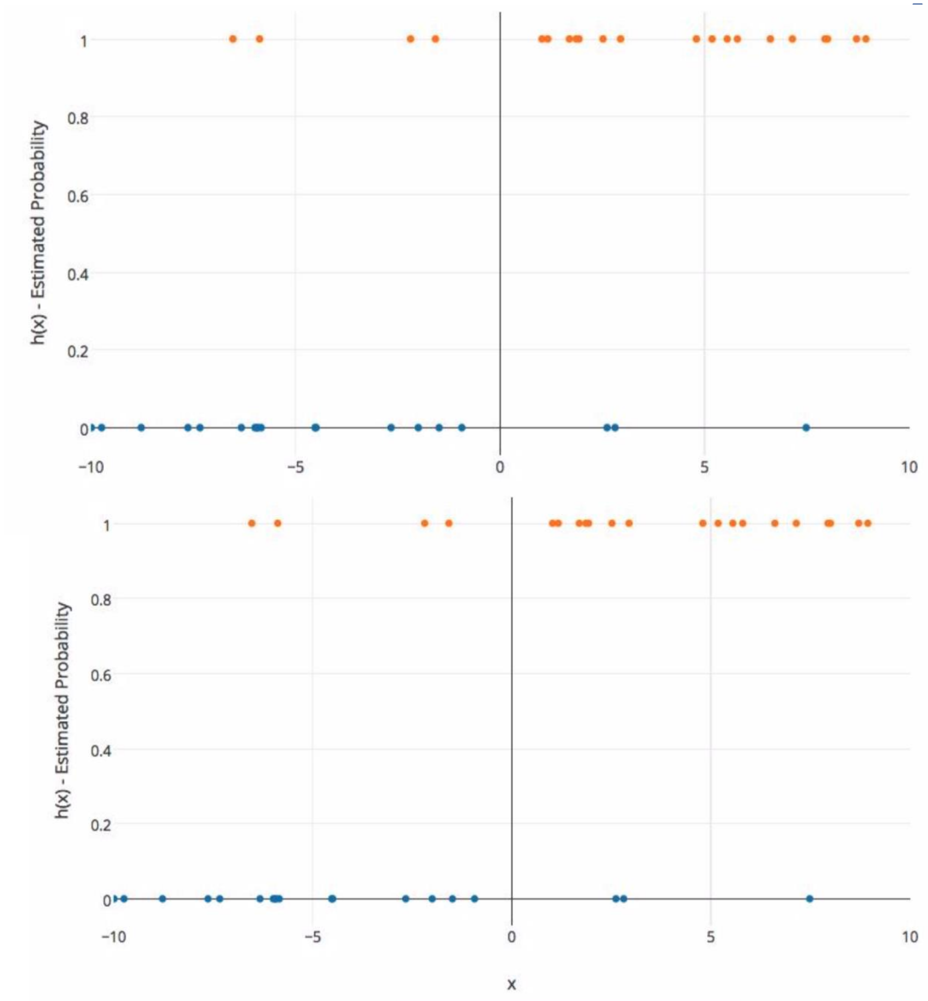
Classification

- Regression problems is to predict a continuous value, for classification problems, it is trying to predict discrete categories.
- We will only cover Binary Classification problems. The convention for binary classification is to have two classes 0 and 1
- Some examples of classification problems:
    - ➢Email spam
    - ➢Loan
    - ➢Disease Diagnosis

Classification

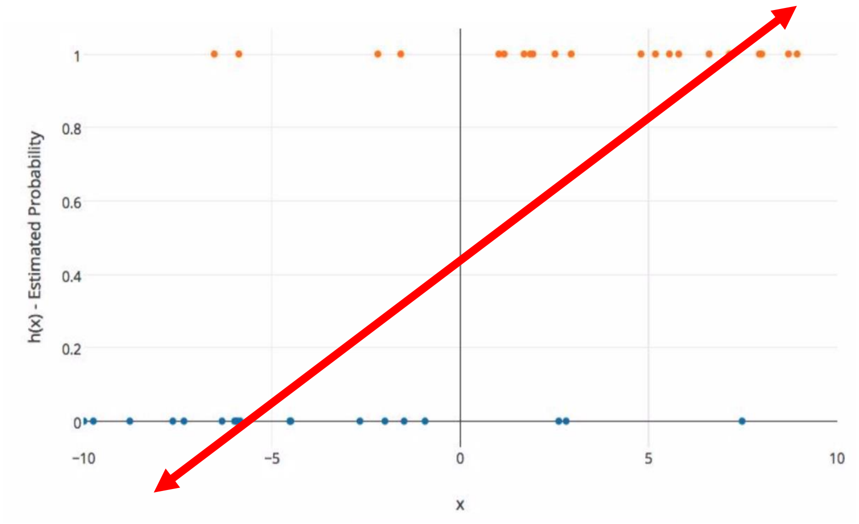
- Imagine we plotted out some categorical data against one feature.

- The X axis represents a feature value and the Y axis represents the probability of belonging to class 1.

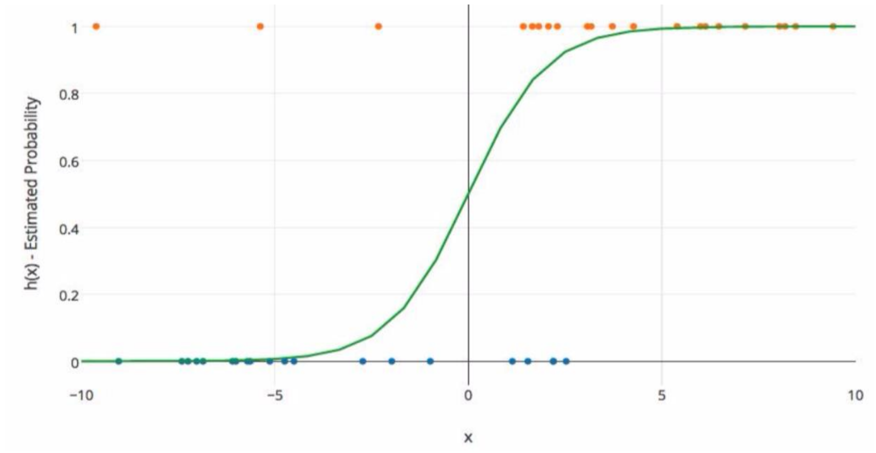


Classification

- We can’t use a normal linear regression model on binary groups. It won’t lead to a good fit:

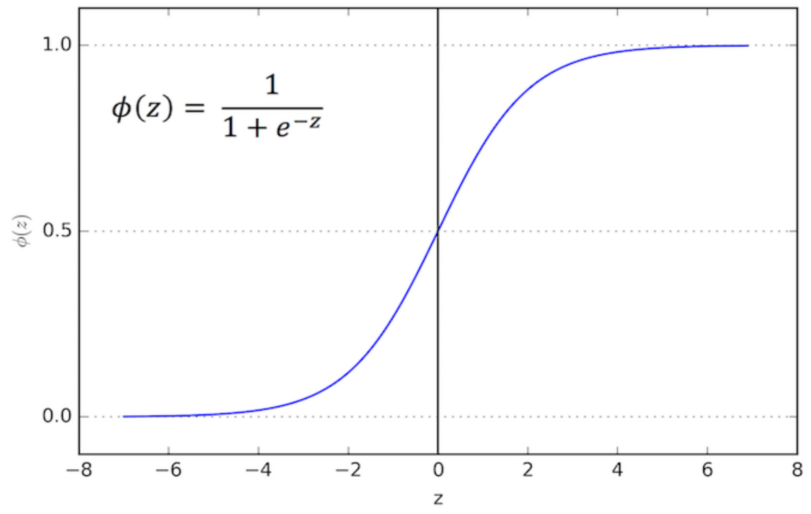


- It would be great if we could find a function with this sort of behavior:



Logistic Regression

- The Sigmoid (aka Logistic) Function takes in any value and outputs it to be between 0 and 1.

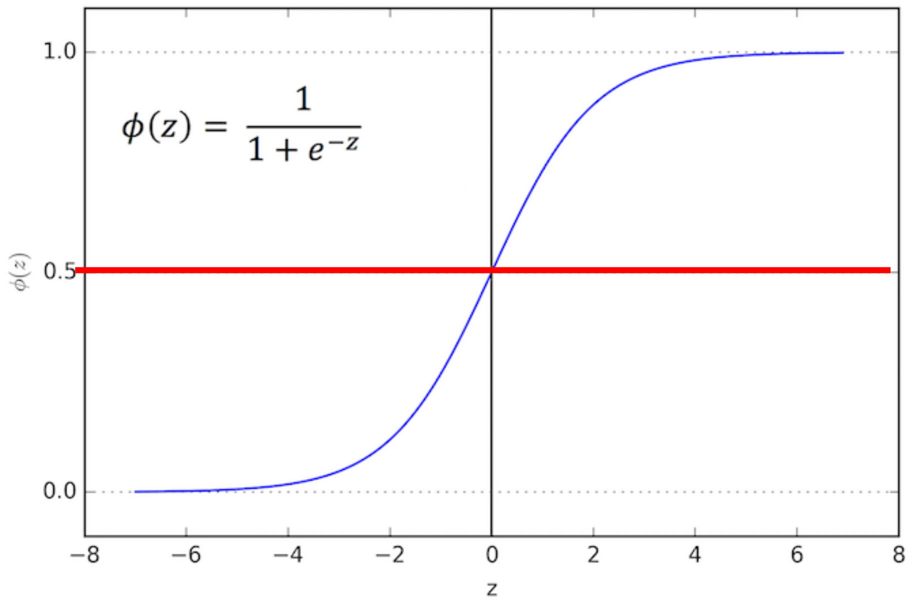


Linear model: $y = b_0 + b_1 x$

Logistic Model: $p = \frac{1}{1+e^{-(b_0+b_1 x)}}$

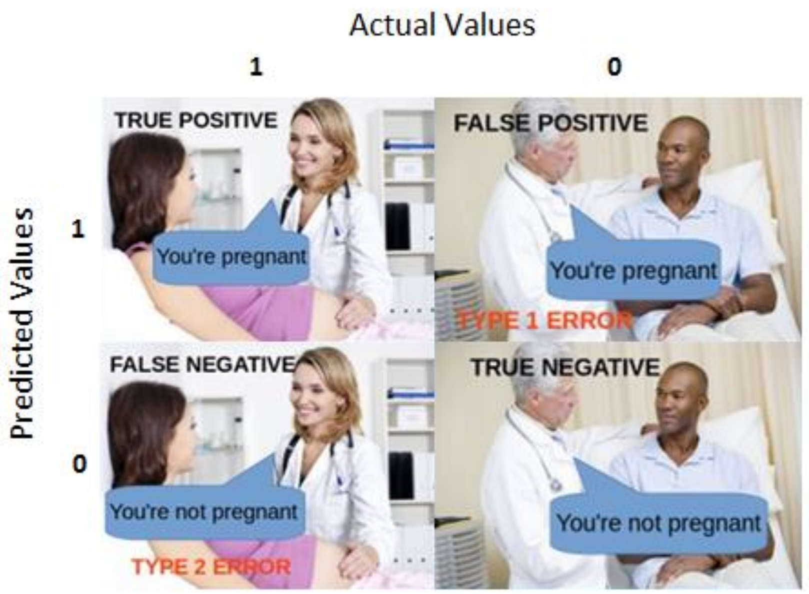
Logistic Regression

- We can set a cutoff point at 0.5, anything below it results in class 0, anything above is class 1.
- We use the logistic function to output a value ranging from 0 to 1. Based off of this probability we assign a class.



Classifier Measurement: Confusion matrix

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive(FP) Type I error
Predicted Negative	False Negative (FN) Type II error	True Negative (TN)



https://miro.medium.com/max/693/1*7EYylA6XlXSGBCF77j_rOA.png

Classification Metrics

- $Accuracy = \frac{TN+TP}{Total\ observations}$
- $Precision = \frac{TP}{TP+FP}$
- $Specificity = \frac{TN}{TN+FP}$
- $Recall/Sensitivity = \frac{TP}{TP+FN}$
- $F1Score = \frac{2*Precision*Recall}{Precision+Recall}$

Group Discussion

- Case 1 - COVID 19 testing result: COVID 19 =1, Healthy = 0
- Case 2 – email spam: email spam = 1, not spam = 0
- Case 3 – bank loan: Bad loan =1, good loan =0
- For each of case, define which metric is better evaluation metric and explain the reason.

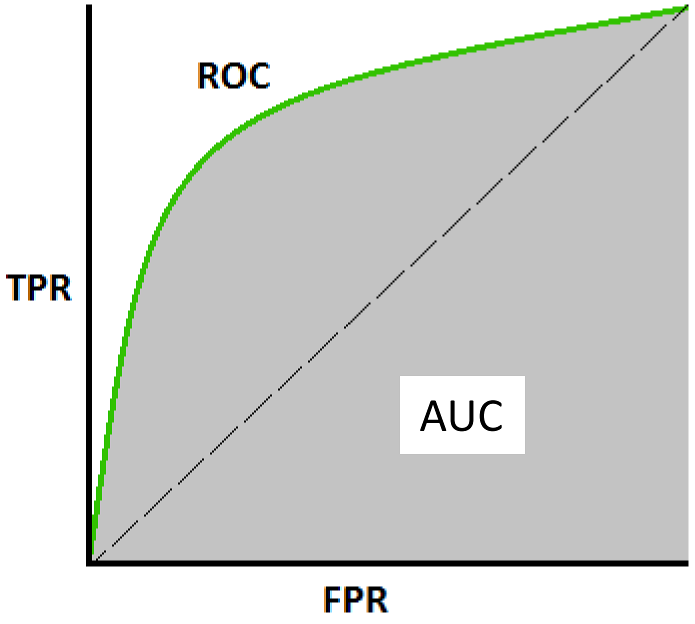
Evaluation – ROC and AUC

- **ROC** curve
- An **ROC** curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:
  - ➢True Positive Rate
  - ➢False Positive Rate

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

AUC stands for "Area under the ROC Curve."

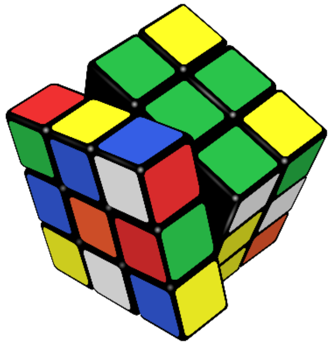
AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

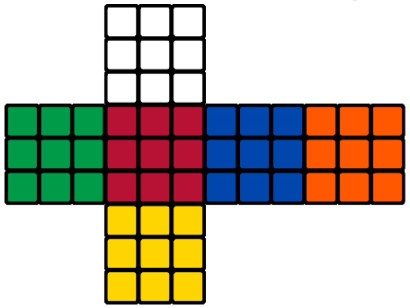


https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Example – Logistic Regression

```
# Initializing Spark Session and import logistic regression module
from pyspark.sql import SparkSession
from pyspark.ml.classification import LogisticRegression
spark = SparkSession.builder.appName(‘lgr').getOrCreate()

# Loading training data
training = spark.read.format("libsvm").load("sample_libsvm_data.txt")
lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)

# Fit the mode and print out the coefficients and intercept
lrModel = lr.fit(training)
print("Coefficients: " + str(lrModel.coefficients))
print("Intercept: " + str(lrModel.intercept))
```





Clustering

Clustering

- For unlabelled data, usually we try to create groups from data, instead of trying to predict classes or values.
- This type of problem is known as **clustering.**
- The input are unlabelled data, and the **unsupervised learning** algorithm returns back possible clusters of the data.
- By the nature of this problem, it can be difficult to evaluate the groups or clusters for "correctness".
- A large part of being able to interpret the clusters assigned comes down to domain knowledge.
- A lot of clustering problems have no 100% correct approach or answer, that is the nature of unsupervised learning

K Means Clustering

- K Means Clustering is unsupervised learning algorithm that will attempt to group similar cluster together.
- Some typical clustering problems:
  - ➢Clustering similar documentation
  - ➢Clustering customers based on features
  - ➢Market segmentation
  - ➢Identify similar physical groups
  - ➢Clustering users group based on there usage pattern (telecom, online purchase)

Clustering with K Means

Theory

The K-Means algorithm iteratively attempts to determine clusters within the test data by minimizing the distance between the mean value of cluster center vectors, and the new candidate cluster member vectors. The following equation assumes data set members that range from **X1** to **Xn**; it also assumes **K** cluster sets that range from **S1** to **Sk** where **K <= n**.

$$\underset{\mathbf{S}}{\mathbf{arg\,min}} \sum_{i=1}^{K} \sum_{x \in \mathbf{S}_i} \left\| x - B_i \right\|^2$$

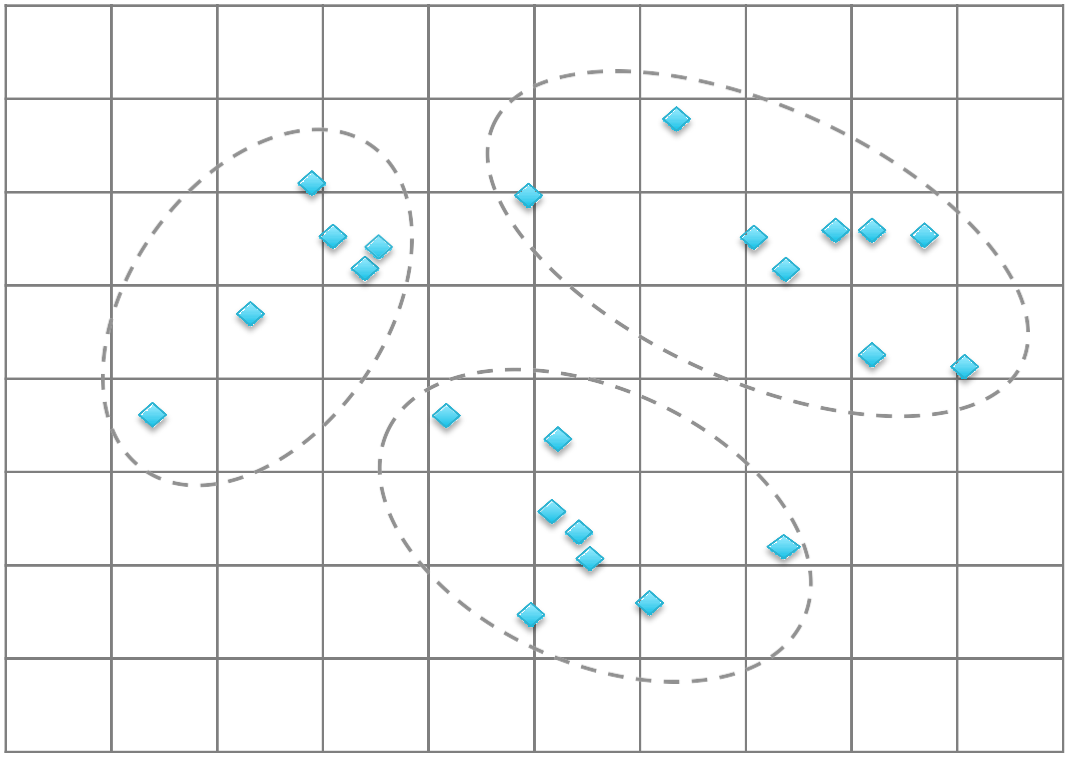
where $\mathbf{B}_i$ is the mean of members of $\mathbf{S}_i$

Clustering with K Means

- The K Means Algorithm
  - ➢Choose a number of Clusters “K”
  - ➢Randomly assign each point to a cluster
  - ➢Until clusters stop changing, repeat the following:
  - ➢For each cluster, compute the cluster centroid by taking the mean vector of points in the cluster
  - ➢Assign each data point to the cluster for which the centroid is the closest

Demo: K Means Clustering

- K Means Clustering
  - ➢A common iterative algorithm used in graph analysis and machine learning



Goal: Find “clusters” of data points

Demo: K Means Clustering

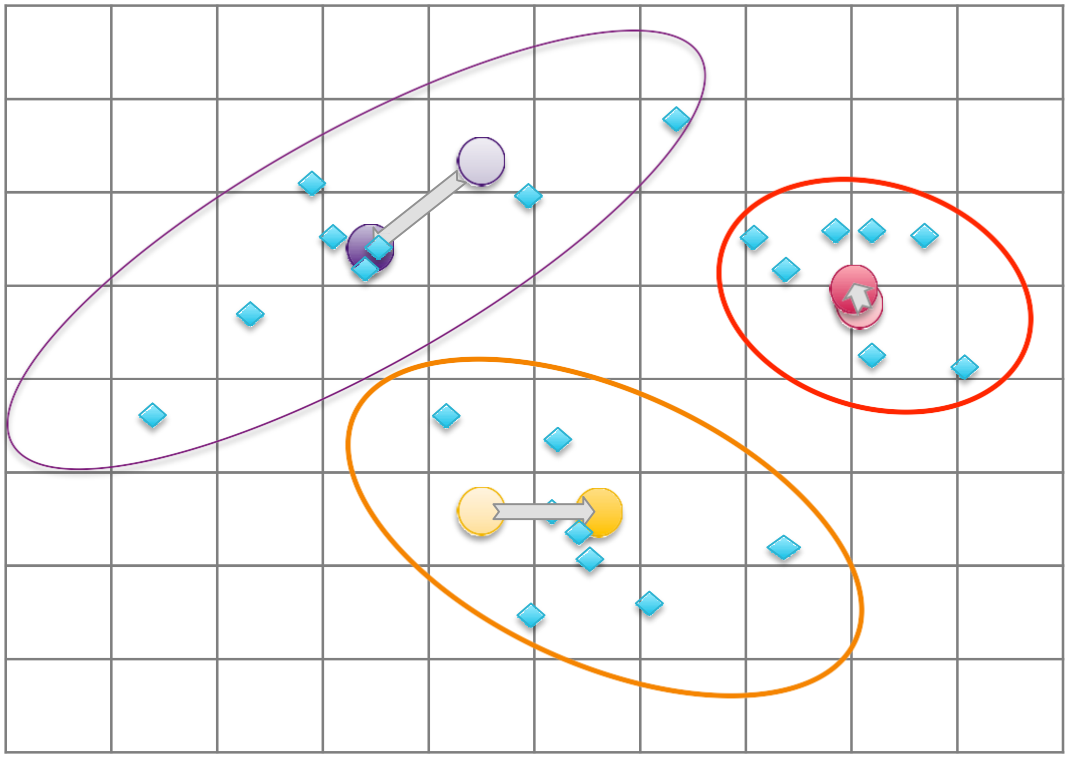
Big Data Engineering For Analytics

Input Column

Param name	Type(s)	Default	Description
featuresCol	Vector	"features"	Feature vector

Output Column

Param name	Type(s)	Default	Description
predictionCol	Int	"prediction"	Predicted cluster center

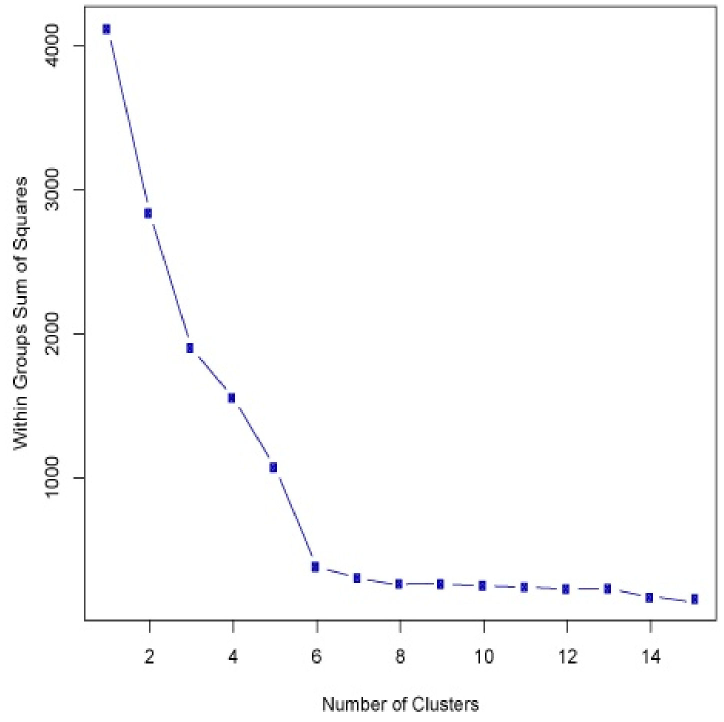


1. Choose K random points as starting centers
2. Find all points closest to each center
3. Find the center (mean) of each cluster
4. If the centers changed by more than c, iterate again

5. Close Enough - Done!

How to choose a K value

- It is not easy to choose a best K value
- One method is the elbow method
  - ➢Compute the sum of the squared error (SSE) for some values of K
  - ➢The SSE is the sum of the squared distance between each member of the cluster and its centroid
  - ➢Plot K against the SSE, you will see that the error decreases as k gets larger; this is because when the number of clusters increases, they should be smaller, so distortion is also smaller.
  - ➢ The idea of the elbow method is to choose the k at which the SSE decreases abruptly.



Evaluation

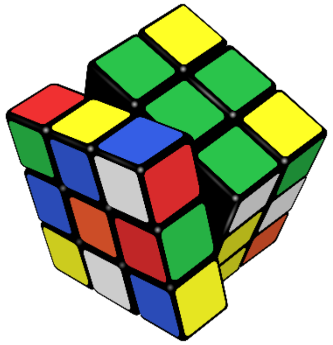
- Silhouette Coefficient
- If the ground truth labels are not known, evaluation must be performed using the model itself. The Silhouette Coefficient is an example of such an evaluation, where a higher Silhouette Coefficient score relates to a model with better defined clusters. The Silhouette Coefficient is defined for each sample and is composed of two scores:
  - a: The mean distance between a sample and all other points in the same cluster.
  - b: The mean distance between a sample and all other points in the next nearest cluster.

  - The Silhouette Coefficient s for a single sample is then given as:

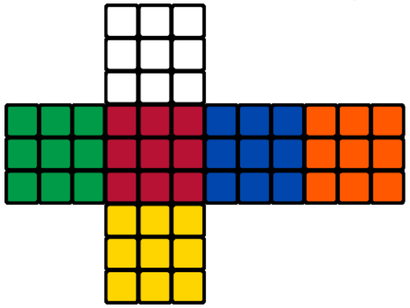
$$s = \frac{b - a}{\max(a, b)}$$

Example - KMeans

```
# Initializing Spark Session
from pyspark.sql import SparkSession
from pyspark.ml.evaluation import ClusteringEvaluator
spark = SparkSession.builder.appName(‘kmeans').getOrCreate()

#Importing Kmeans Library and loading the Dataset
from pyspark.ml.clustering import KMeans
dataset = spark.read.format("libsvm").load("sample_kmeans_data.txt")

# Training a kmeans model
kmeans = KMeans().setK(2).setSeed(1)
model = kmeans.fit(dataset)

# Making predictions
predictions = model.transform(dataset)

# Evaluate clustering by computing silhouette score
evaluator = ClusteringEvaluator()
silhouette = evaluator.evaluate(predictions)
print("Silhouette with squared euclidean distance = " + str(silhouette))

# Print cluster centers
centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
    print(center)
```



Big Data Engineering For Analytics



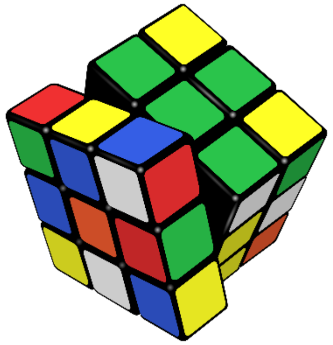
Summary

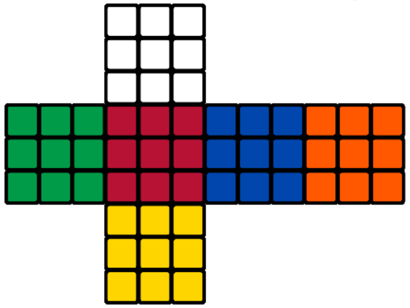
"The real question is, when will we draft an artificial intelligence bill of rights? What will that consist of? And who will get to decide that?"

Gray Scott

In Essence

- Apache Spark project is ideally positioned to help tackle these topics, which range from data ingestion and feature extraction/creation to model building and deployment.
- The three basic steps in feature engineering are feature extraction, feature transformation, and selection. Spark ML provides implementation of several algorithms to make these steps easier.
- Spark ML also provides several **classifications** (logistic regression, decision tree classifier, random forest classifier, and more), **regression** (liner regression, decision tree regression, random forest regression, survival regression, and gradient-boosted tree regression), **decision tree** and **tree ensembles** (random forest and gradient-boosted trees), as well as **clustering** (K-means and LDA) algorithms





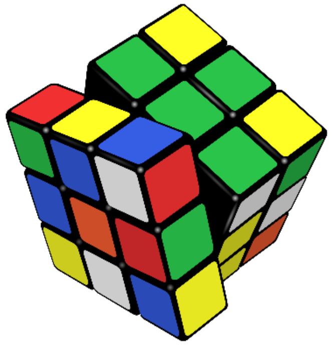
References

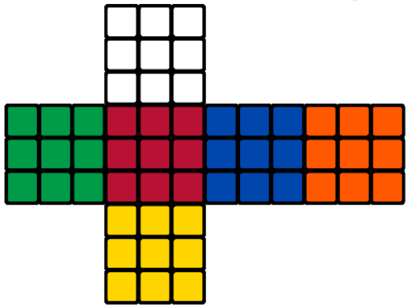
"The key to artificial intelligence has always been the representation."

—Jeff Hawkins

References

- Machine Learning and Security, by David Freeman , Clarence Chio, O'Reilly Media, Inc, February 2018.
- Machine Learning, by Eihab Mohammed Bashier , Muhammad Badruddin Khan , Mohssen Mohammed, CRC Press, August 2016.
- Machine Learning, 2nd Edition by Stephen Marsland, Chapman and Hall/CRC, September 2015
- https://spark.apache.org/docs/2.4.6/api/python/index.html
- https://spark.apache.org/docs/latest/ml-guide.html





Big Data Engineering For Analytics

Appendix

Case Study results

Case 1

COVID 19/ Healthy

Cost of **FN** > Cost of **FP**

Recall

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives}$$

Case 2

Spam/Not Spam

Cost of **FP** > Cost of **FN**

Precision

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

Case 3

Good/Bad loan

Cost of **FN** > Cost of **FP**

Recall

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives}$$

Linear regression - Regularization

- In mathematics, statistics, and computer science, particularly in the fields of machine learning and inverse problems, regularization is a process of introducing additional information in order to solve an ill-posed problem or to prevent overfitting.
- Ridge (L2) – tends to give small but well distributed weights - If you only have a few predictors, and you are confident that all of them should be really relevant for predictions, try Ridge as a good regularized linear regression method.
- Lasso (L1) and elastic net – give sparse weights (most zeros) – If you have a lot of predictors (features), and you suspect that not all of them are that important, Lasso and ElasticNet may be a good choice.

- `LinearRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)`
- regParam >0, elasticNetParam =0 → L2
- regParam >0, elasticNetParam =1 -> L1
- regParam >0, elasticNetParam (0 , 1) -> elasticNet