# CS525: Advanced Topics In Database Systems
# Large-Scale Data Management
# Spring-2013

# Project 4

**Total Points:** 130

**Release Date:** 03/12/2013

**Due Date:** 03/28/2013

**Teams:** Project to be done in teams of two.

## Short Description

In this project, you will write map-reduce jobs that implement data mining and machine learning techniques in Hadoop. More specifically, you will implement the ***K-Means*** clustering technique and ***Naïve Bayes*** classifier.

## Problem 1 (Naïve Bayes Classifier) [50 points]

Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. In general, any classifier technique has two phases: (1) Phase 1: the creation of the classifier, and (2) Phase 2: using the created classifier to classify (label) new objects. In this problem you will implement phase 1 of Naïve Bayes (i.e., the creation of the classifier) using Hadoop.

*Hint: You may reference these links to get some ideas (in addition to the course slides):*

*http://nickjenkin.com/blog/?p=85*
*http://en.wikipedia.org/wiki/Naive_Bayes_classifier (especially the 'Sex Classification Example')*

### Step 1 (Creation a Training Dataset) [10 points]:

- Assume we have 20 class labels, namely {C1, C2, …., C20}, and 50 numeric features, namely {F1, F2, …, F50}. You need to create a training dataset for the classifier that consists of 2M (2 x $10^6$) records, where the first field is the class label, and then the numeric values for the 50 features.
- Make the probability of labels C1 to C5 10% each, from C6 to C10 6% each, and the rest is 2% each (this should some to 1).
- Use a range and distribution of your choice for the values in each feature.
- The values in each record are comma separated.

### Step 2 (Build the Classifier Model) [20 points]:

Write map-reduce job(s) to build the Naïve Bayes classifier in a distributed fashion. The final output file (Refer to Table 1) should have one record for each class label along with its learned probability (the percentage of records having is class label), and the mean and variance of each feature.

- The output file should not include a header line
- Use the appropriate separator (of your choice) between the values.

| Class label | Learned probability | Feature 1 | Feature 2 | …. | Feature 50 |
|---|---|---|---|---|---|
| C1 | % of records with C1 | Mean and variance of F1 values having label C1 | … | | …. |
| C2 | % of records with C2 | Mean and variance of F1 values having label C2 | …. | | … |
| … | | | | | |
| C20 | % of records with C20 | …. | Mean and variance of F2 values having label C20 | … | … |

**Table 1: Classifier Model**

### Step 3 (Classify Unseen Values) [20 points]:

- Create a dataset similar to the training one, but without class labels. Create 500K line, each line consists of the values of the 50 features.
- Write map-reduce job(s) that reads the unseen data records and classifies them, i.e., assigns a label to each record based on the model created in Step 2.
- The classification equation is given in the course slides, and you can also refer to the example in this link: *http://en.wikipedia.org/wiki/Naive_Bayes_classifier* *(the 'Sex Classification Example')*

## Problem 2 (K-Means Clustering) [50 points]

K-Means clustering is a popular algorithm for clustering similar objects into *K* groups (clusters). It starts with an initial seed of K points (randomly chosen) as centers, and then the algorithm iteratively tries to enhance these centers. The algorithm terminates either when two consecutive iterations generate the same K centers, i.e., the centers did not change, or a maximum number of iterations is reached.

*Hint: You may reference these links to get some ideas (in addition to the course slides):*

http://en.wikipedia.org/wiki/K-means_clustering#Standard_algorithm
https://cwiki.apache.org/confluence/display/MAHOUT/K-Means+Clustering

### Step 1 (Creation of Dataset) [10 points]:

- Create a dataset that consists of 2-dimenional points, i.e., each point has (x, y) values. X and Y values each range from 0 to 10,000. Each point is in a separate line.
- Scale the dataset such that its size is around 100MB.
- Create another file that will contain K initial seed points. ***Make the "K" value as a parameter to your program***, so in the demo session, I will give you certain K, your program will generate these K seeds, and then you upload the generated file to the cluster.

### Step 2 (Clustering the Data) [40 points]:

Write map-reduce job(s) that implement the K-Means clustering algorithm as given in the course slides. The algorithm should terminates if either of these two conditions become true:

a) The K centers did not change over two consecutive iterations
b) The maximum number of iterations (make it six (6) iterations) has reached.

- Apply the tricks given in class and in the 2nd link above such as:
  - Use of a combiner
  - Use a single reducer
  - The reducer should indicate in its output file whether centers have changed or not.

Hint: Since the algorithm is iterative, then you need your program that generates the map-reduce jobs to control whether it should start another iteration or not.

## Problem 3 (Use of Mahout) [30 points]

Mahout is a package that implements data mining and machine learning techniques on top of Hadoop including Naïve Bayes and K-Means clustering.

- Choose one of the two techniques above and run it using Mahout.
- You need to understand the data format that Mahout accepts and the parameters that it takes to run either or the two algorithms.

*Hint: You may reference these links to get some ideas (in addition to the course slides):*

**http://mahout.apache.org/**

## What to Submit

You will submit a single zip file containing the java code needed to answer the queries above. Also include a .doc or .pdf report file containing any required documentation.

## How to Submit

Use blackboard system to submit your files.

## Demonstrating Your Code

Each team will schedule an appointment with the instructor to demonstrate the project. Demonstration should be within the week after the due date.