Assignment 2

Due Date: Mentioned in eLearning

Instructions

- This project involves writing code in PySpark for the questions below. The code can be on Databricks notebook or a notebook that can run on other platforms, such as Amazon EMR (Elastic MapReduce) cluster.
- You can submit the public link of your Databricks notebook or a PySpark file that can be run on AWS EMR cluster.
- All instructions for compiling and running your code must be placed in the README file.
- You should use a cover sheet, which can be downloaded from http://www.utdallas.edu/~axn112530/cs6350/CS6350_CoverPage.docx
- You are allowed to work in pairs i.e. a group of two students is allowed. Please write the names of the group members on the cover page.
- **You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After four days have been used up, there will be a penalty of 10% for each late day. The submission for this assignment will be closed 2 days after the due date.**
- Please ask all questions on Piazza, not via email.

1 Friend Recommendation using Mutual Friends

In this part of the assignment, you will use Spark based MapReduce algorithm to generate friend recommendation for users. The recommendations will be based on number of mutual friends. For example, if users X and Y are not yet friends but have a large number of mutual friends, then we may be able to recommend them to each other.

The dataset that you will be given contains data in the following format for each line.

UserID List of friends' UserIDs separated by commas

There is a tab character after the UserID. Also, note that the friendship relationship is undirected i.e. if X is a friend of Y, then Y is also a friend of X.

For a *random subset of 10 users*, you will generate an ordered list of top 10 friend recommendations based on count of mutual friends. The following should be the output format for each user's recommendations:

UserID List of UserIDs of top 10 friend recommendations

There should be a tab character after the UserID and the list of friends' User IDs should be comma separated.

It may not be possible to generate 10 recommendations for some users. In that case, you can output fewer than 10 recommendations.

Below are the requirements of the project:

1. You have to come up with the best algorithm using MapReduce. You are free to use RDDs or DataFrames for implementation in Apache Spark. You should detail your algorithm and its pseudo-code in a separate file.
2. You cannot use any external library that computes these values for you.
3. You are free to use Databricks or other Spark environments. You have to ensure that the TA can run the code.
4. The dataset for the project is at:

   https://an-ml.s3.us-west-1.amazonaws.com/soc-LiveJournal1Adj.txt
5. Remember that you have to display the results for a random subset of 10 users on screen.

2 Implementing Naive Bayes Classifier using Spark MapReduce

In this part, you will implement a Naive Bayes classifier using MapReduce. You will need to apply the classifier on a text based dataset of your choice. Following are the suggested steps:

1. **Data Preprocessing**: The first step would involve pre-processing a large text corpus of your choice using PySpark. This would involve steps such as tokenization, stemming, stop word removal. Remember to use a dataset that corresponds to classification problem.
2. **Splitting the dataset**: Split the preprocessed dataset into a training set and a testing set.
3. **Training the Naive Bayes model**: Implement the Naive Bayes algorithm in PySpark using MapReduce to train the model on the training set.
4. **Testing the model**: Use the trained model to classify the documents in the testing set and evaluate the performance of the model.

Below are some of the requirements of the project:

1. Remember that you have to implement the Naive Bayes algorithm using PySPark and not use a library for it. You will need to study the details of the algorithm and implement it from scratch.
2. You are allowed to use libraries for data loading, parsing, pre-processing, result evaluation, but not for the main model.
3. You have to select a text-based classification dataset of your choice. You will need to host your dataset on a public location.
4. The output of the algorithm should be the accuracy of the model and other details of Naive Bayes such as prior of each class.
5. You should include a report that includes the pseudo-code of the MapReduce and the summary of results.