MIDTERM PROJECT

1) Million Songs data set

As a part of the pre-processing steps, using scala, we created an inline function that gets the data and creates a field with binary values based on the year in each row. That is, the function assigns a binary value '1' when the year is greater than or equal to 1965 and '0' when the year is lesser than 1965. Then, a RDD[LabeledPoint] is created with the newly created binary field as the 'label' and the rest all as the 'features'. Having created a labeled point, as a second step of pre-processing, we used Standard Scaler to normalize the data. Finally, the data is divided into training and testing set using the Randomsplit function.

Now, the training RDD has 90% of the data and the test RDD has 10% of the data. To minimize the number of features to be taken into consideration for the implementation of different classification and regression models, we used PCA and reduced the number of features to 50. With this new RDD, for the classification, Decision tree and SVMwithSGD models were implemented and for regression, the Linear Regression model was used.

In order to evaluate the performance of the models, metrics like test error, mean squared error etc., were chosen. The test error evaluates classification model and the mean squared error evaluates regression models.

Metric Results & Conclusion:

Classification

Model Name	With PCA	Without PCA
SVMwithSGD	0.0140	0.0141
Decision Trees	0.0148	0.0150

Regression

Model Name	With PCA	Without PCA
Linear Regression with SGD	2.84	1.84

Our basic assumption with PCA is that the feature reduction process the model performs basically keeps the features that has a higher correlation with the labels. Logically, with the lesser number of features, the error rate must be higher than the rate without PCA. This logic is evident in the regression model while the classification model shows a slightly lesser error rate with the PCA. In both cases, SVM with SGD model has a better rate. So, this classification model looks good for this case. And, for regression, we used Decision Trees as well. But, the metric evaluation was producing an error rate that was way higher than the result of Linear Regression with SGD model. So, we conclude that the Linear Regression with SGD is a better regression model for this case.

2) Adult Income data set

As a part of the pre-processing steps, using python, we created a data frame that stores the entire data set. From the created data frame, the columns having categorical variables were chosen to perform a categorical to numerical data conversion process. And, for the conversion, we used a concept called 'dummy coding', which takes all the unique categories available in each column and creates a new column for each category and assigns a binary values for them based on the occurrence of those categories. The updated data frame now contains 110 columns and has been written to a new CSV file. This CSV file is then loaded in Spark Context. Now, to create a labeled point with this data set, an inline function is called to get a label with binary value based on the income value. That is, if the Income is >50K then 1 else 0. As the second step of pre-processing, we used Standard Scaler to normalize the data. Finally, the data is divided into training and testing set using the Randomsplit function.

Now, the training RDD has 70% of the data and the test RDD has 30% of the data. To minimize the number of features to be taken into consideration for the implementation of different classification models, we used PCA and reduced the number of features to 50. With this new RDD, for the classification, Decision tree and SVMwithSGD models were implemented.

In order to evaluate the performance of the models, the metric area under ROC was chosen.

Metric Results and Conclusion:

Model Name	With PCA	Without PCA
SVM with SGD	0.614	0.779
Decision Trees	0.759	0.805

For Area under ROC metric, the closer the value of the metric to 1, the better the performance of the model is. In this case, the decision tree model has a better Area under ROC rate. And, in both the classification models, without using PCA has a better result than with using PCA.

3) TV Commercial data sets

As a part of the pre-processing steps, using scala, we used the MLUtils library to load the data set in LIBSVM format and normalized the contents using Standard Scaler. To select the top 30 features from the data set, we used the ChiSqTest model. Now, the RDD has a label and 30 features. This RDD is then divided into training and testing data using the random split function.

Now, the training RDD has 70% of the data and the test RDD has 30% of the data. To further minimize the number of features to be taken into consideration for the implementation of different clustering models, we used PCA and reduced the number of features to 20. With this new RDD, for the clustering, K-Means and Gaussian Mixture models were implemented.

In order to evaluate the performance of the models, the metric WSSSE was used.

Metric Result and Conclusion:

Model Name	With PCA	Without PCA
Kmeans	200.363	208.283

For this case, we used Kmeans and Gaussian Mixture to perform clustering. We could not use a proper metric with the Gaussian Mixture model to evaluate its performance. For Kmeans, we tweaked the number of clusters to see the effect of that over the result. The result of the model with PCA had more error rate than the model without the PCA in almost all situations.

To run the project:

We used IntelliJ IDE to run the scala scripts. We have 3 scala scirpts and all the 3 scripts comes under a project named A1. The Jar file of the project is included in the submitted zip file. Use the jar file to run the scripts in IntelliJ. And, for the second question, you need to run the python script first. The script generates a new CSV file which in turn has to be used as an input file in the scala script. So, you will be needing to change the file path in all the scala scripts to run the projects.