6240 - Parallel Data Processing with Map-Reduce

Section 02, Data Mining Project Report

Team Members:

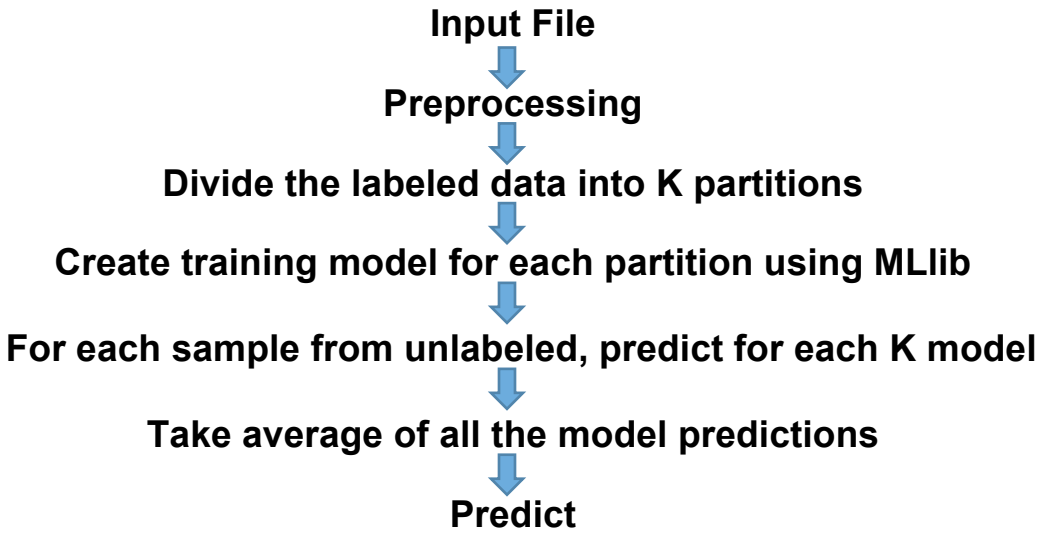
Atindra Mardikar

Nishant Ratnakar Shetty

Technology Used:

- **Spark 2.0.6**
- **Scala**
- **MLlib (RDD-based)**

Program Flow:



Input File:

The input is .bz2 file compressed and consists of .csv files from different tables have been joined and provided to us. The primary challenge was to figure out the number of columns and what each column represents. We did extract few top thousand records from the data set to analyze the different columns and the quality of the data.

Preprocessing:

Selection of attributes:

During our analyses of the given data set, we realized that the input data is highly biased. This affects the model adversely. There were approximately 1.7 million records present and out of those, just 11% of the data had positive outcomes. Hence, we had to do some serious preprocessing to ensure the model generated would not be biased.

Many columns were redundant in the given data set. For instance, sampling event id, country, state province, group id and few other columns in sampling covariates had no correlation what so ever with the presence of the bird specified in the problem statement. Choice of good columns as features was important.

Candidates for selection of attributes for model features are as follows:

No.	Column / Model feature	Reason for choosing
1	LOC_ID	Chances of finding a bird in the same location is higher
2	LATITUDE	Since the bird is a migratory, it is seen in specific locations only given the weather conditions and follows that pattern every year, LATITUDE comes in handy
3	LONGITUDE	Since the bird is a migratory, it is seen in specific locations only given the weather conditions and follows that pattern every year, LONGITUDE comes in handy
4	YEAR	This parameter combined with MONTH, DAY AND TIME forms a good correlation for citing of the bird
5	MONTH	This parameter combined with YEAR, DAY AND TIME forms a good correlation for citing of the bird
6	DAY	This parameter combined with MONTH, YEAR AND TIME forms a good correlation for citing of the bird
7	TIME	This parameter combined with MONTH, DAY

		AND YEAR forms a good correlation for citing of the bird
8	BCR	Chances of finding a bird at a conservation site is higher
9	CAUS_TEMP_AVG	Chances of finding the bird at places experiencing moderate temperatures is high
10	CAUS_PREC	Places where the bird habitats needs to have moderate to high precipitation
11	CAUS_SNOW	The bird migrates to warmer places during winter. In winter, you can find them at crop fields and pastures. This parameter combined with HOUSING_DENSITY, POP00_SQMI and HOUSING_PERCENT_VACANT gives valuable insight about chances of presence of bird.
12	HOUSING_DENSITY	Correlated with attribute No. 11,13 and 14
13	POP00_SQMI	Correlated with attribute No. 11,12 and 14
14	HOUSING_PERCENT_VACANT	Correlated with attribute No. 11,12 and 13
15	ELEV_GT	Habitats places close to wet vegetation. Lower the value better the chances
16	ELEV_NED	Since, data from a different source but emphasis the same. Habitats places close to wet vegetation. Lower the value better the chances
17	Agelaius_xanthomus	Belongs to same family and follows the same migratory pattern
18	Agelaius_tricolor	Belongs to same family and follows the same migratory pattern

Choice of model:

After lot of research, we realized that for a discrete categorical output from predictor variables that are continuous and/or categorical, logical regression model works well.

We are using logistic regression with multiple predictors. All the predictors are entered at the same time which makes the computation really fast.

Since we have zeroed on Spark for programming the project, we decided to use MLlib for providing the logistic regression model. As per documentation online, the input required for the model need to be of the format **LabeledPoint**. This in turn would require that we provide a label and a vector. A vector would be the set of features we give as predicators for the regression model and label would be 0 or 1 based on whether the Red-winged Blackbird was seen or not.

Data cleaning:

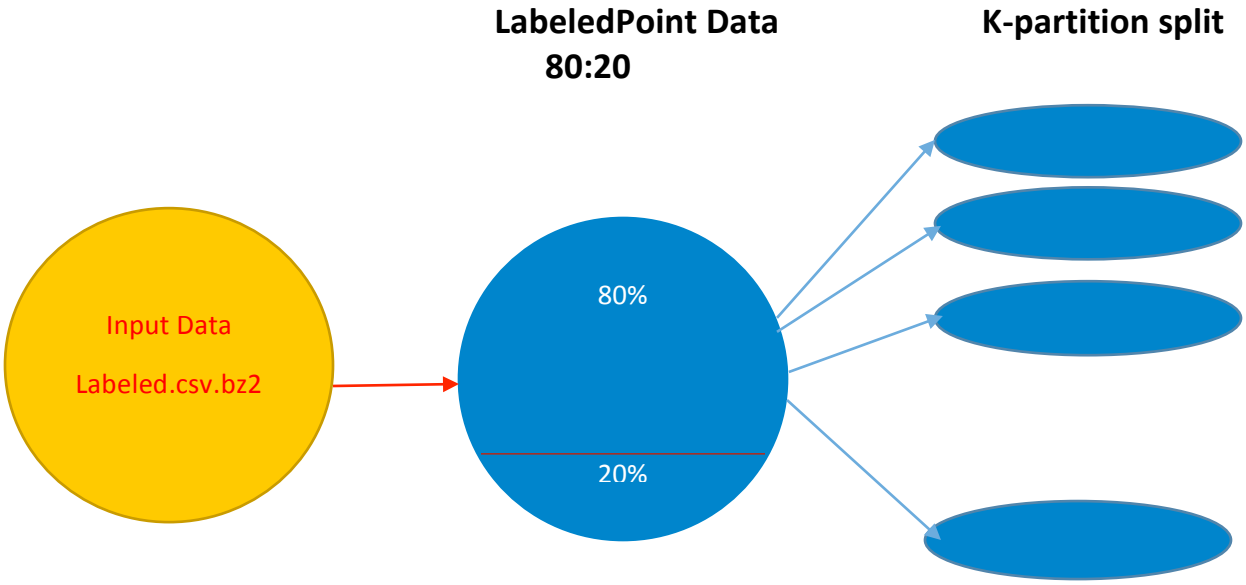
Since the data provided to us had a lot of inconsistent data, we had to make some assumptions regarding replacement values of the invalid data.

1. Most of the records '?' as the value entered. The value of all these records are replaced by 0.
2. For records with bird count as 'X', we assumed that the birder doesn't recollect the number of times he seen the bird. Hence, we have replaced all the 'X' values with 1.

Splitting input data:

Before splitting the input data for training, we first convert data into the labeled format as required by the regression model. This happens in parallel and we create an RDD of LabeledPoint. Once the LabeledPoint RDD of all input data is created, we first randomly split the data into 2 sets: One for training and other for testing. The ration we choose was 80:20.

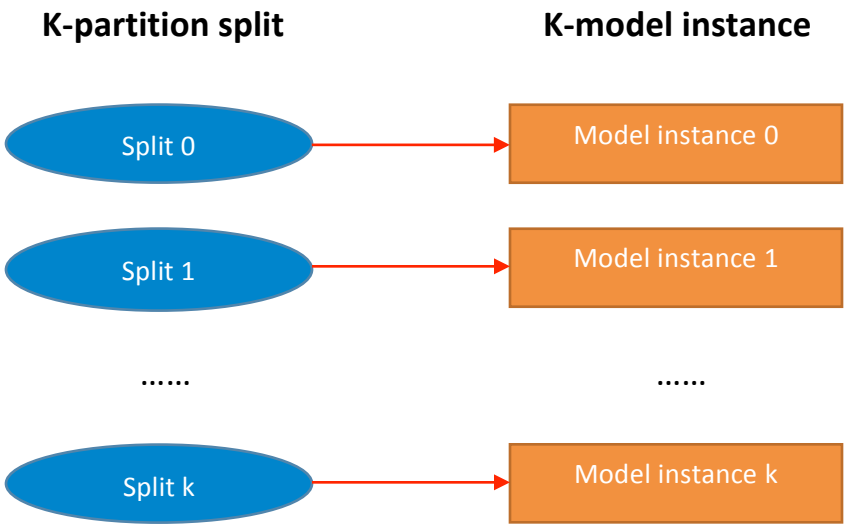
Now out of the 80 percent LabeledPoint data, we randomly split the data into K-different partitions which make inputs for K-regression model.



Pseudo Code:

```
map(...,(label,features)){
//Generate a random number between 1 to K
        int rand = randomNumber (1..K)
        emit( k, (label,features))
 }
```

Training on input data:

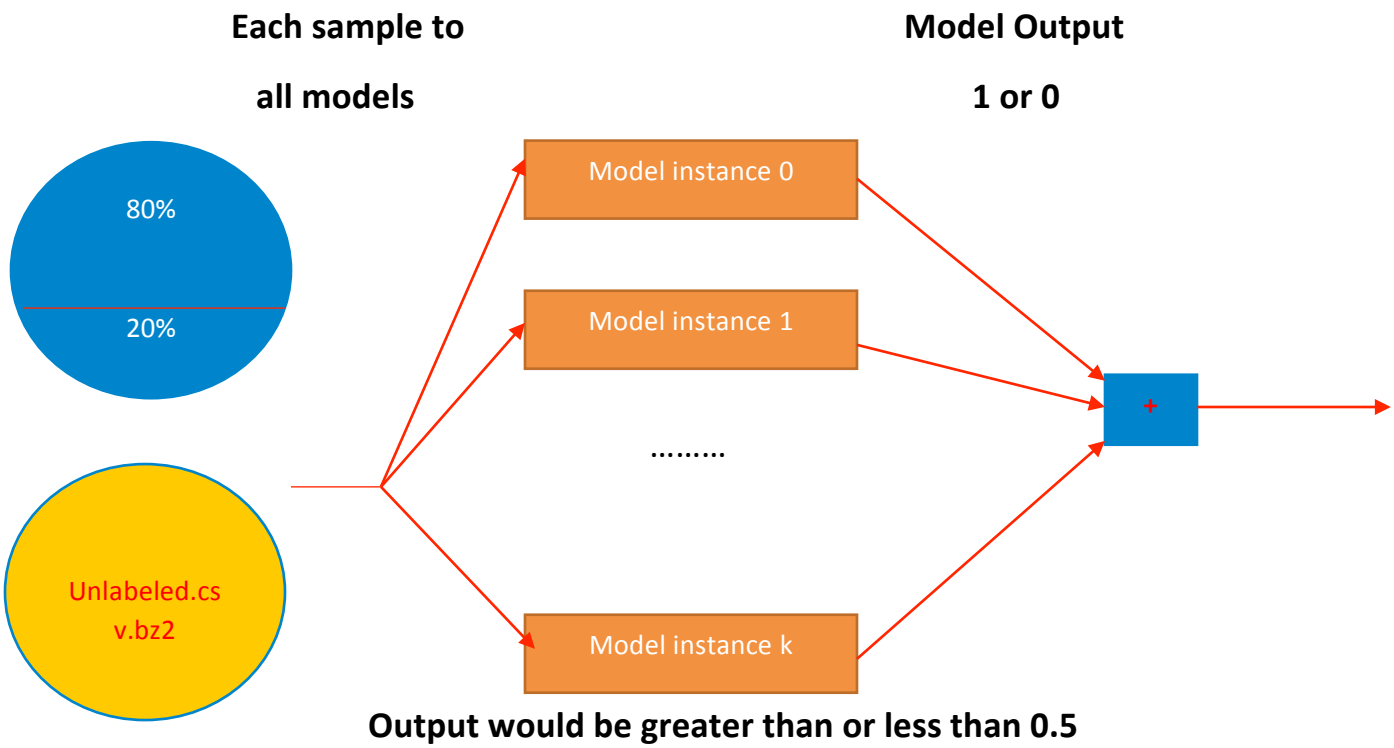


Now for each split of input LabeledPoint RDD split, we are creating an instance of logistic regression model using MLlib library. Hence, the model trains the data in parallel.

Pseudo Code:

```
map(instanceNo,[(label0,features0),(label1,features1)...]){
        //Create model for each partition
            Model model= new RegressionModel
            ([(label0,features0),(label1,features1)...])
            emit(k,model)
}
```

Testing on remaining input data and unlabeled input file:



Output would be greater than or less than 0.5

Testing and Prediction happens in parallel and the process is the same. Each record from remaining LabeledPoint or from unlabeled.csv.bz2 is sent to all the K instance of logistic regression model and the output of each model is either 1 or 0.

Pseudo Code:

```
map(...,(sampleId,(label,features))){
        int positives = 0;
        for(i <- 1 to K){
        prediction= model[i].predict(features)
        if(prediction==1){
            positives++;
```

```
                    }
                }
        if(positives/K > 0.5) emit(sampleID,1)
        else emit(sampleID,0)
            }
```

Prediction:

The output of each instance of model is accumulated and divided by the number of instances. If less than 0.5, more than half the models indicate that the bird is not seen. Hence, label is assigned to 0 else it is 1.

Output:

We have achieved an accuracy of **79.8%** when tested on labeled dataset.

Data Type	Machine	No. of Cores	Training Time	Prediction Time
labeled.csv.bz2	**m4.large**	**11**	**20 mins**	**6 mins**
Unlabeled.csv.bz2	**m4.large**	**11**	**------**	**2 mins**

References:

www.stackoverflow.com

http://spark.apache.org/docs/latest/

https://spark.apache.org/docs/latest/mllib-linear-methods.html

http://www.zeigler-hill.com/uploads/7/7/3/2/7732402/psy_512_logistic_regression.pdf

https://www.allaboutbirds.org/guide/Red-winged_Blackbird/id