DIC Lab3: Data Analytics Pipeline using Apache Spark

Submitted by

1. SAJID KHAN - 50248743
2. VINAY VARDHAMAN – 50248852

Preparation – Word Count Example

This example performs wordcount operation on an article text specified in the path below, all using Apache spark context. It then prints the word frequency to the output screen.
Please find the code in the respective jupyter notebook.

Lab - Titanic Data Analysis

This example uses Apache spark context. It tries to predict label of whether a person survived or not in Titanic ship wreck using Machine Learning modules like MLlib, SQL Dataframe and ML module with and without pipeline in Random forest classification.

It further generates different gauge parameters like Test Error, Accuracy, f1, Weighted precision and Weighted recall.

Analysis with MLlib module

First preprocess the data to transform some variables like Gender, remove empty lines and strings and create labeled points specific to MLlib library.

This labeled data is split into training and test sets in the ratio of 7:3

Random forest classifier is used to model the training data. Please note this is a binary classification.

Classification metrics like Area under Precision recall and ROC curve are computed.

Results:

Random Forest takes 4sec for training the model.
Area under Precision recall curve is 0.68
Area under ROC curve is 0.85

Analysis with SQL data frame and ML module

Then repeat the preprocessing step and drop all the null values.

The same data is again split into training and test data in the ratio of 7:3

Without Pipeline

Use String Indexer to index the labels, adding metadata to the data columns. Encode the labeled columns with one hot encoder and generate features column from the encoded labels using vector assembler.

Train the model with the modified data and validate predictions.

With Pipeline

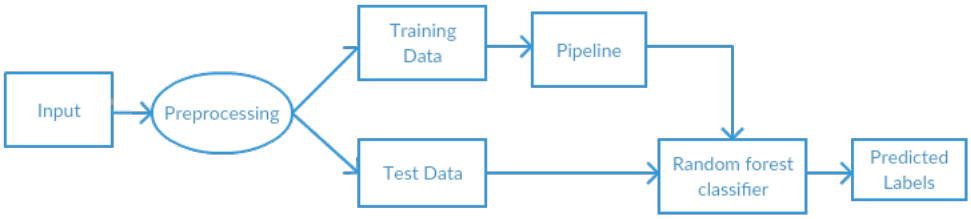
Index, encode and generate features from the dataset like before.

Now, chain the indexers and forest in a pipeline. Train the model with pipeline and make the predictions. Please note that we use a Multi class classifier in this model.

Evaluated the following metrics from Multiclass classifier.

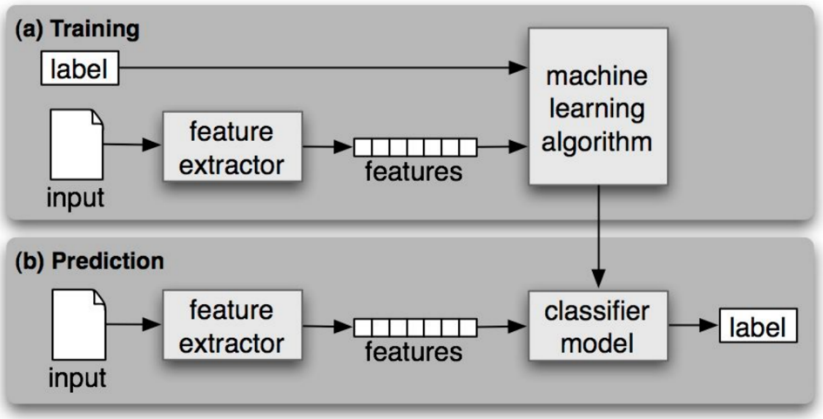
Results:
Test Error is 0.21
Accuracy is 0.79
f1 = 0.78
Weighted Precision is 0.80
Weighted Recall is 0.79.

Block Diagram



Titanic Analysis with Pipeline

Complete pipeline



Lab – Multiclass Article Classification using pyspark

The following part does Multi-class article classification using pyspark. Article text is preprocessed by removing non-alphanumeric characters. Tokenizing, stop words removal and count vector operations are done within a pipeline. The dataset is split into training and test sets to train the model and perform predictions.

Predictions are made the following ways

1. Logistic Regression using Count Vector Features
2. Logistic Regression using TF-IDF Features
3. Cross-Validation
4. Naive Bayes
5. Random Forest

Sample output - Respective Accuracies

1. 0.73
2. 0.76
3. 0.76
4. 0.80
5. 0.78

Analysis

Data collected from different articles is loaded into a data frame using a new spark context.

The data set is processed with a regex tokenizer, stop words remover and count vectorizer by fitting in a pipeline. Data is indexed, encoded and vectorized.

The data set is split into training and test sets in the ratio of 8:2.

Logistic Regression using Count Vectors

A Logistic regression classifier is used to train the model with the following parameters
Number of Iterations is 20
Regularization, lambda is 0.3
Elastic net parameter is 0.

Accuracy obtained for test data = 0.73

Logistic Regression using TF-IDF features

Using the data frame in the example above, Term frequency – TF and Inverse Document Frequency – IDF are calculated using the pyspark ml feature module.

Tokenizing, stop words removal, TF, IDF and string index are fit to the pipeline.

Data set is again split into train and test sets using the same ratio. Same regression parameters are employed.

Accuracy obtained is 0.76

Cross validation

Same processing and pipeline fitting is done. Cross validator is used to transform the test data.

Accuracy obtained is 0.76. Better than both the Logistic regression models above.

Naïve Bayes

Import Naïve Bayes from pyspark ml classification module. Train the model and transform the test data.

Accuracy obtained is 0.80, close to the same accuracy obtained in cross validation.

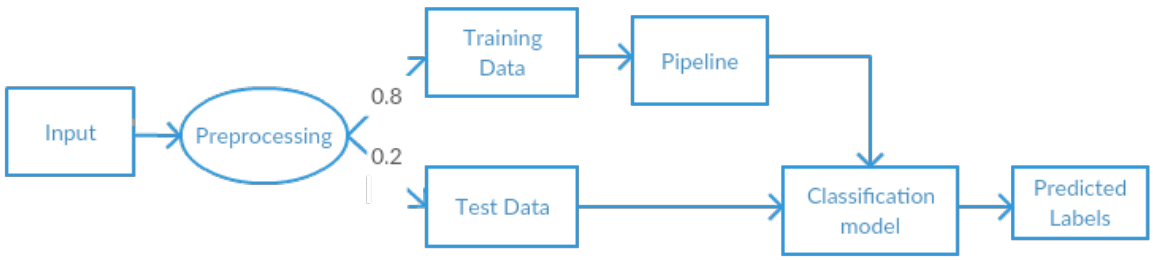
Random forest

Import Random forest classifier form pyspark ml classification module. Use the following parameters in classification.

Number of trees = 100
Maximum depth = 4
Maximum Bins = 32

Please note that Multiclass classification evaluator is used.

Accuracy obtained using this model 0.78.



The block diagram shows various steps involved in multi class classification of the article data.

The classification models we used are Logistic regression with Count vectors, TF – IDF, Cross validation, Naïve Bayes and Random forest.

By far, the highest accuracy is achieved for Naïve Bayes.

Unknown data Analysis

Few articles from each category/ topic are collected with respective category labels.

The same models discussed above are employed on this unknown article data and the accuracies obtained are as follows.

Results:

1. Logistic Regression using Count Vector Features – 0.73
2. Logistic Regression using TF-IDF Features – 0.75
3. Cross-Validation – 0.87
4. Naive Bayes – 0.80
5. Random Forest – 0.93

Conversely, it can observed that accuracy obtained is highest for Random forest classifier.

Note

Keywords used

1. Sports-sports
2. Business – business
3. Politics – politics
4. Technology – blockchain.

References

1. https://creativedata.atlassian.net/wiki/spaces/SAP/pages/83237142/Pyspark+-+Tutorial+based+on+Titanic+Dataset.
2. https://towardsdatascience.com/multi-class-text-classification-with-pyspark-7d78d022ed35
3. https://piazza.com/class/jd1uzvjlrnp4dq?cid=495.