
# PROJECT REPORT
# ”BICYCLE SHARING DEMAND PREDICTION USING PYSPARK AND FLUME WITH STREAMING HTTP REQUESTS”

Vikhyat Dhamija


# PySpark Certification Project Report

## Project Description:

With the spike in pollution levels and the fuel prices, many Bicycle Sharing Programs are running around the world. Bicycle sharing systems are a means of renting bicycles where the process of obtaining membership, rental and bike return is automated via a network of joint locations throughout the city. Using this system people can rent a bike from one location and return it to a different place as and when needed.

In this project , we will be building a Bicycle Sharing demand forecasting service that combines historical usage patterns with weather data to forecast the Bicycle rental demand in real-time. To develop this system, we have to explore the dataset and build a model and once it's done we have to persist the model and then on each request run a Spark job to load the model and make predictions on each Streaming HTTP request.

## Tasks to be Performed in the project :

1. Data Exploration , Analysis & Transformation
2. Model Development
3. Model Implementation and Prediction


# Data Exploration and Transformation And Model Development

## Description :

- Training Dataset in CSV format was pushed into HDFS
- Spark SQL Dataframe was created after loading a training Dataset from the HDFS
- Data Analysis and Transformations were performed over the data stored in a Dataframe.
- One hot Encoding of the categorical variables was performed for ML Model building.
- Features were assembled in the vector in one column in the Dataframe to act as input for the training of the ML model.
- Various Trained Models were created using various Machine Learning Models available in the Spark Mlib Library :
  - Linear regression
  - Decision Tree
  - Random Forests
  - Gradient Boosted Regression

## Observations :

- Based on the performance analysis , it was found that RMSE was around : 150. As per studies a term called scatter index (SI) is used to judge whether RMSE is good or not. SI is RMSE normalised to the measured data mean or SI=RMSE/measured data mean. If SI is less than one, your estimations are acceptable.
- In our case , 150 / 191( mean count) was less than 1 so the results were acceptable
- Gradient Boosted Regression Trained Model was stored/persisited in the HDFS for future use.


# Model Implementation and Predictive service Application development – Demand Prediction

## Description:


## Major Steps Performed :

- Test Dataset in CSV format was pushed into HDFS.
- Dataset was converted into Spark Datframe (which are actually the RDDs beneath in the Spark )
- ML Model was loaded from the HDFS for predictions to be performed.
- Over the test Dataframe , the feature extraction was done using String Indexer , One hot encoding and all the inputs which are to be given to ML model were clubbed/assembled in the “Features” Vector Column in the Dataframe.
- Then the resultant Dataframe was given as an input to our stored/persisted ML Model.
- Result of the above steps with the predicted count was stored in the RDBMS Table named “test_predictions” to be used by WEB Application.


# Model Implementation and Predictive service Application development- Streaming Application

## Description of Architecture :

### Source

HTTP Request generator to the Host running the Flume Agent

curl -X POST -H 'Content-Type: application/json; charset=UTF-8' -d '[{"headers" : {"a":"b"},"body": "{\"season\" : 1 , \"weather\" :2 , \"temp\" : 13.4 , \"atemp\" : 14.6 , \"humidity\" : 84.0 , \"windspeed\" : 18.0}"}] ' http://localhost:9000

### Flume Agent

HTTP Source running at Port 9000

Memory Channel

Custom Sink for spark Streaming to pull Data from this sink

### Spark

Spark Streaming creating DStreams of RDDs , by polling the Data from the Flume Custom sink

Spark Processing the Batch predictions

### RDBMS

Final Predictions are stored in the RDBMS


# Model Implementation and Predictive service Application development- Streaming Application-(Cont…)

## Description :

- Flume was configured with desired configuration as per our architecture .
- Connection was established between the Spark Streaming Context and the Flume Custom Sink *( Note we need to put the hostname of host where our Flume agent is running, in our Spark Streaming application so as to make our Spark Application poll the Custom sink running at that host. Hence in our case , Pull Based Flume Integration with Spark Streaming was performed*)
- ML Model stored in HDFS was loaded by our Spark Application .
- Predictions were performed based on the ML model over the microbatches or RDDs in the Dstreams generated by Spark Streaming. In order to do this , RDDs were first converted to Dataframe and in the same way as described in previous application , the stored model was used to predict the results .
- Feature extraction was done using String Indexer , One hot encoding and then , all the inputs which need to be given to ML model were clubbed/assembled in the “Features” Vector Column in the Dataframe.
- Results were appended to the table in RDBMS(Mysql).
- Finally , Curl requests with post HTTP requests were pushed to the Localhost at port 9000 for generating streaming HTTP requests to our Host which is running Flume agent.( *as the request was generated from the same EC2 instance where the Flume was running , that is why the messages pushed to the local host otherwise Elastic public IP address of the Host running Flume would have been used , to generate requests from any Machine over the web to our Public Host* )


# Project Submission :

## Attachments :

- Jupyter Notebooks with Spark Streaming outputs
- Jupyter Notebooks with comprehensive comments and proper outputs showing the process of model development and Data exploration and the final Demand Prediction application.
- Config File – flume_spark.conf for the configuration of the flume.

## Important Commands :

- Curl Command for generating requests:

```
curl -X POST -H 'Content-Type: application/json; charset=UTF-8' -d '[{"headers" : {"a":"b"},"body": "{\"season\" : 1 , \"weather\" :2 , \"temp\" : 13.4 , \"atemp\" : 14.6 , \"humidity\" : 84.0 , \"windspeed\" : 18.0}"}] ' http://localhost:9000
```

- Run the Flume Agent

Change the directory to flume

```
cd /opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1.p0.4/bin
```

Executing the flume agent with proper configuration:

```
./flume-ng agent --conf conf --conf-file /mnt/home/edureka_960126/flume_spark.conf --name httpagent -Dflume.root.logger=INFO,console
```

*Note : Location of the Config file , Hostname of the Host running the flume needs to be changed in both the flume_spark.conf as well as the Jupyter notebook in the Spark Streaming Code so as to make it connect with the host running the Flume agent*

*Note : Dimesnsions used in setting up the model has one categorical variable "Season" as in the problem statement the exploding od season was demanded and it is possible to encode the weather and include holiday as well which already has 0 , 1 values and club them in the features vector for ML model generation.*