Customer Insight and Predictive analytics Using Apache Storm

Divya Ramadugu [dxr131330@utdallas.edu](mailto:dxr131330@utdallas.edu)

Sathishbabu Rajasekar

sxr136330@utdallas.edu

UpendraReddy Lingareddygari

uxr130130@utdallas.edu

Viswatej Reddirouthu Samhita Muthineni

[vxr131330@utdallas.edu](mailto:vxr131330@utdallas.edu) sxm130331@utdallas.edu

***Abstract* – The proposed project is gain customer insight and predictive analytics on phones based on trending tweets from Twitter using Spark and MLlib. In this inputs are the live tweets from Twitter and with help of Spark we find the tweets relevant to phones matches using hash-tags and tweet text. These tweets are then classified using naïve bayes classifier for sentiment analysis. Upon classification, supporters and non-supporters for different phone models are provided with corresponding recommendations.**

***Index terms*: Spark, hash-tags, ml-lib, Hive**

1. INTRODUCTION

Motivation: In the current era people who support or don’t support particular brands. These topmost activities are the trending topics which the world users are interested in. Since the users want to know the feedback and rating of the products before they purchase phones.

This led us in the building of the project, to show the recommendation URLs after classifying tweets in twitter. Spark is used to collect the live tweets related to phones and is stored on flat files. The flat files are then fed to Naïve Bayes for sentimental analysis. The recommendation URLs are then provided based on the classification from Naïve Bayes. The **Apache Spark** is an open-source cluster computing framework originally developed in the AMPLab at [UC Berkeley](http://en.wikipedia.org/wiki/UC_Berkeley). In contrast to [Hadoop](http://en.wikipedia.org/wiki/Hadoop)'s two-stage disk-based [MapReduce](http://en.wikipedia.org/wiki/MapReduce" \o "MapReduce) paradigm, Spark's in-memory primitives provide performance up to 100 times faster for certain applications.[[1]](http://en.wikipedia.org/wiki/Apache_Spark#cite_note-1) By allowing user programs to load data into a cluster's memory and query it repeatedly, Spark is well suited to machine learning algorithms.[[2]](http://en.wikipedia.org/wiki/Apache_Spark#cite_note-2) Spark can interface with a wide variety of file or storage systems, including [Hadoop DistributedFileSystem(HDFS)](http://en.wikipedia.org/wiki/Apache_Hadoop#Hadoop_distributed_file_system),[]](http://en.wikipedia.org/wiki/Apache_Spark#cite_note-3)[Cassandra](http://en.wikipedia.org/wiki/Apache_Cassandra), [OpenStack Swift](http://en.wikipedia.org/wiki/OpenStack#Object_Storage_.28Swift.29), or [Amazon S3](http://en.wikipedia.org/wiki/Amazon_S3).

Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and perform RDD transformations on those mini-batches of data. This design enables the same set of application code written for batch analytics to be used in streaming analytics, on a single engine.

MLlib is a distributed machine learning framework on top of Spark that because of the distributed memory-based Spark architecture is ten times as fast as Hadoop disk-based [Apache Mahout](http://en.wikipedia.org/wiki/Apache_Mahout) and even scales better than [Vowpal Wabbit](http://en.wikipedia.org/wiki/Vowpal_Wabbit" \o "Vowpal Wabbit). It implements many common machine learning and statistical algorithms to simplify large scale machine learning pipelines, including:

* summary statistics, correlations, stratified sampling, hypothesis testing, random data generation
* classification and regress: SVMs, logistic regression, linear regression, decision trees, naive Bayes
* collaborative filtering: alternating least squares (ALS)
* clustering: k-means
* dimensionality reduction: singular value decomposition (SVD), principal component analysis (PCA)
* feature extraction and transformation
* optimization primitives: stochastic gradient descent, limited-memory BFGS (L-BFGS)

GraphX is a distributed graph processing framework on top of Spark. It provides an API for expressing graph computation that can model the Pregel abstraction. It also provides an optimized runtime for this abstraction. GraphX started initially as a research project at UC Berkeley AMPLab and Databricks, and was later donated to the Spark project

1. DETAILS OF APPROACH
2. Architecture

The architecture of spark is shown in figure 1. From architecture perspective Apache Spark is based on two key concepts; Resilient Distributed Datasets (RDD) and directed acyclic graph (DAG) execution engine. With regards to datasets, Spark supports two types of RDDs: parallelized collections that are based on existing Scala collections and Hadoop datasets that are created from the files stored on HDFS. RDDs support two kinds of operations: transformations and actions. Transformations create new datasets from the input (e.g. map or filter operations are transformations), whereas actions return a value after executing calculations on the dataset (e.g. reduce or count operations are actions).  
The DAG engine helps to eliminate the MapReduce multi-stage execution model and offers significant performance improvements.

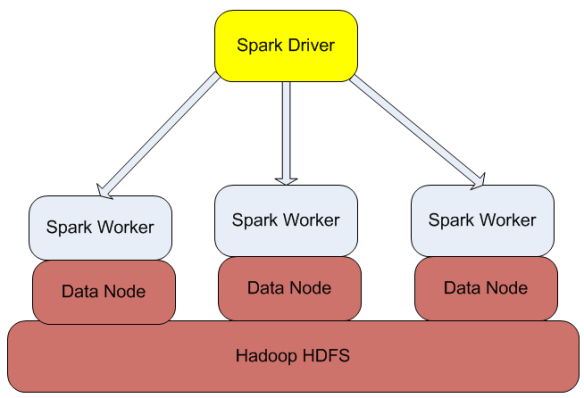


Figure 1: Architecture of Spark



Figure 2: Spark Streaming Architecture

1. Topology

In topology layers: First layer runs a Spark streaming to read the real time input of tweets which the users are tweeting. Second layer picks up these tweets and pre-process them and classify these tweets related to each category defined, in this case vapours types of phones. This filters the data into eight different subcategory. The third layer writes the filtered data to corresponding flat files. Fourth layer takes input from the generated flat files and uses Mlib APIs for multiclass classification. The classified data is stored on flat file and is passed to next layer. The fifth layer identifies obtains the user query on which data he wants to get and recommends the target audience whom to advertise to. This layer also displays the results on HTML page.

1. METHOD
2. Technologies Used:
3. Spark

Apache Spark is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala and Python, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming.

1. Naïve Bayes

[Naive Bayes](http://en.wikipedia.org/wiki/Naive_Bayes_classifier) is a simple multiclass classification algorithm with the assumption of independence between every pair of features. Naive Bayes can be trained very efficiently. Within a single pass to the training data, it computes the conditional probability distribution of each feature given label, and then it applies Bayes’ theorem to compute the conditional probability distribution of label given an observation and use it for prediction.

MLlib supports [multinomial naive Bayes](http://en.wikipedia.org/wiki/Naive_Bayes_classifier#Multinomial_naive_Bayes), which is typically used for [document classification](http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html). Within that context, each observation is a document and each feature represents a term whose value is the frequency of the term. Feature values must be nonnegative to represent term frequencies. [Additive smoothing](http://en.wikipedia.org/wiki/Lidstone_smoothing) can be used by setting the parameter λ (default to 1.0). For document classification, the input feature vectors are usually sparse, and sparse vectors should be supplied as input to take advantage of sparsity. Since the training data is only used once, it is not necessary to cache it.

1. Spark Streaming

Spark Streaming receives live input data streams and divides the data into batches, which are then processed by the Spark engine to generate the final stream of results in batches. Spark Streaming provides a high-level abstraction called discretized stream or DStream, which represents a continuous stream of data. DStreams can be created either from input data stream from sources such as Kafka and Flume, or by applying high-level operations on other DStreams. Internally, a DStream is represented as a sequence of [RDDs](https://spark.apache.org/docs/0.9.1/api/core/index.html#org.apache.spark.rdd.RDD)

1. Twitter4J

Twitter4J is an unofficial Java library for the Twitter API.  
With Twitter4J, java application can be easily integrated with the Twitter service. Developer feature of authenticated twitter account is used to stream tweets using twitter4j client. The Hosebird client is broken down into two modules: twitter4j-core and twitter4j-stream. The twitter4j-core module makes use of a message queue, which the consumer can poll for the raw String messages and hbc-twitter4j module uses the twitter4j listeners and data model on top of the message queue to provide a parsing layer.

1. Setup of Technologies
2. Spark

The installation process is explained in this section. We tried to cover all important steps and troubleshooting steps here in this document based on Ubuntu 14.04 and Hadoop 2.5.0. In the following, we will discuss the steps in details. We assume that you already have Hadoop based on our previous documents.

## Downloading Spark.

Spark can be downloaded from its website with some pre-built options:

<http://spark.apache.org/downloads.html>

We would like to use Hadoop pre-built package. So, select Spark release “1.1.0” and package “Pre-built for Hadoop 2.4” from the menu, and download it from the provided link

## Running on Scala

Open a new terminal and go to “spark-1.1.0-bin-hadoop2.4/bin  
 folder, and run spark shell using the following command:

./spark-shell

1. Naïve Bayes

[Naive Bayes](http://en.wikipedia.org/wiki/Naive_Bayes_classifier) is a simple multiclass classification algorithm with the assumption of independence between every pair of features. Naive Bayes can be trained very efficiently. Within a single pass to the training data, it computes the conditional probability distribution of each feature given label, and then it applies Bayes’ theorem to compute the conditional probability distribution of label given an observation and use it for prediction.

MLlib supports [multinomial naive Bayes](http://en.wikipedia.org/wiki/Naive_Bayes_classifier#Multinomial_naive_Bayes), which is typically used for [document classification](http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html). Within that context, each observation is a document and each feature represents a term whose value is the frequency of the term. Feature values must be nonnegative to represent term frequencies. [Additive smoothing](http://en.wikipedia.org/wiki/Lidstone_smoothing) can be used by setting the parameter λ (default to 1.0). For document classification, the input feature vectors are usually sparse, and sparse vectors should be supplied as input to take advantage of sparsity. Since the training data is only used once, it is not necessary to cache it.

1. Twitter4J

Twitter4J was set up by importing twitter4j-core and twitter4j-stream jars. Several methods of twitter4j were used in development for specifying developer credentials of twitter and fetching near real time data from twitter.

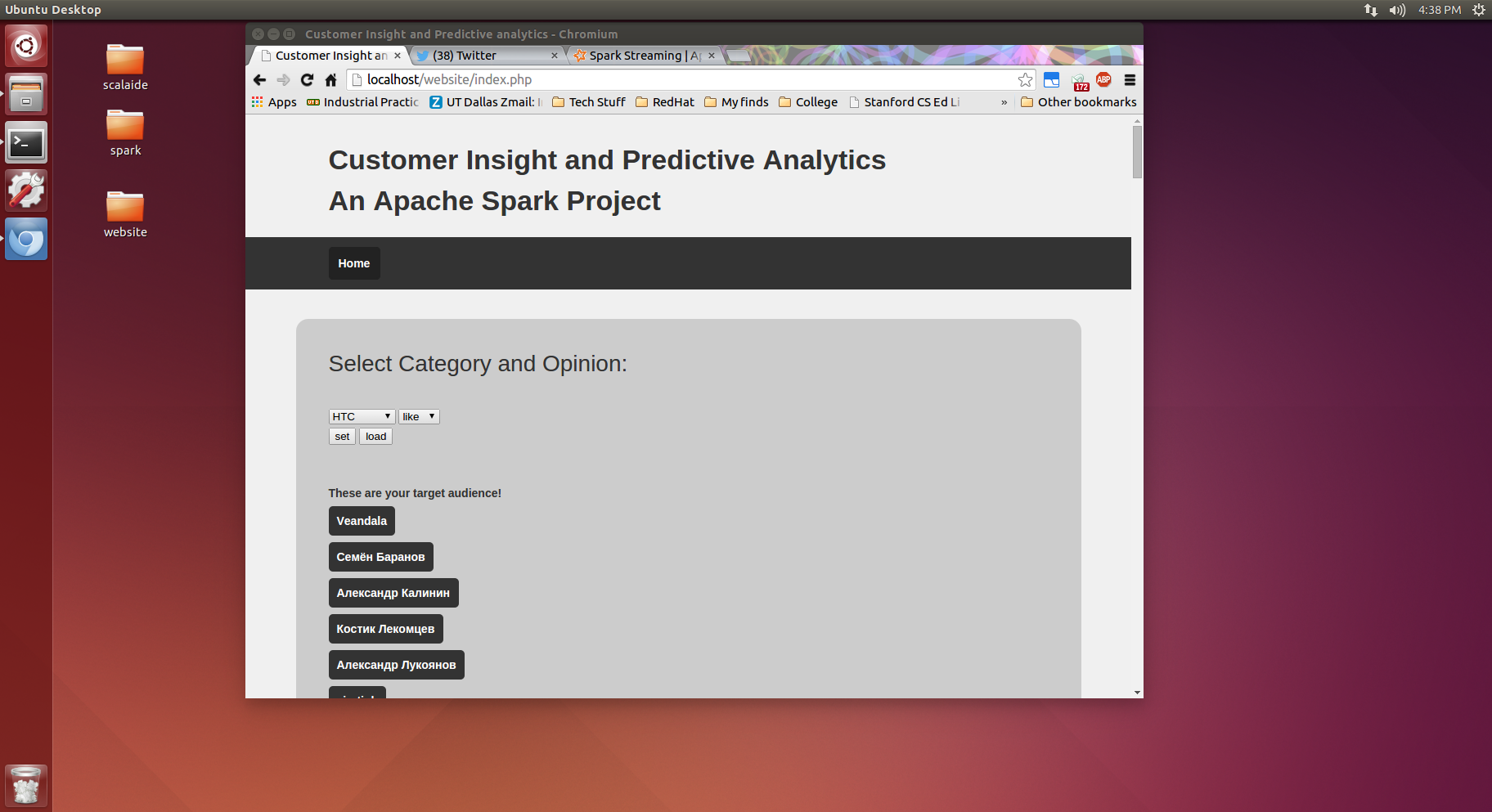
1. PERFORMANCE EVALUATION

Multiclass Naïve Bayes was used to classify tweets in MLIB modelling suite. This language model based classifier gave 75% accuracy in classifying tweets based on their interest towards various brands. We have tried other classifiers such as decision tree, but the accuracy was way less than 75%. Hence we used multinomial naïve Bayes.

1. RESULTS

Tweets are obtained from real-time Twitter using developer credentials. These tweets are used as input by Spark using Spark Streaming and processed using Spark Context to get Phone specific tweets. By using Mlib Naïve Bayes classification model, the selected tweets were classified as eight categories: Phones, Apple, Samsung and HTC each having tow sub categories positive and negative. Based on the results from Naïve Bayes model, the users are recommended with corresponding URLs on HTML page.

Sample result is attached here:



1. CONTRIBUTION

In this project, we have created a new feature of finding out the user’s interest on top of the existing Spark project available on Github. The enhancements were integrating twitter4j API with Spark Streaming project to stream near real-time tweets. We then used Mlib tool to classify selected tweets based on sentiment analysis. These users are recommended with corresponding URLs on HTML page.

1. CONCLUSION

Thus as a part of implementation of this Twitter Spark project we have extended the rudimentary version of Spark starter to a developed version which fetches and streams near real time, classifies Phone tweets based on subcategories, user interest and recommends user with corresponding target audience URLs.

REFERENCES

1-https://spark.apache.org/docs/latest/mllib-guide.html

2 https://spark.apache.org/docs/0.9.1/index.html

3. https://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.package – Spark Api

4- [http://twitter4j.org/en/index.HTML](http://twitter4j.org/en/index.html)

5 –www.youtube.com for video references