ACADGILD

E-COMMERCE

DATA ANALYSIS USING

HADOOP

About ACADGILD

ACADGILD is a technology education startup that aims to create an ecosystem for skill development in which people can learn from mentors and from each other.

We believe that software development requires highly specialized skills that are best learned with guidance from experienced practitioners. Online videos or classroom formats are poor substitutes for building real projects with help from a dedicated mentor. Our mission is to teach hands-on, job-ready software programming skills, globally, in small batches of 8 to 10 students, using industry experts.

ACADGILD offers courses in

Enroll in our programming course & Boost your career

ANDROID DEVELOPMENT

DIGITAL MARKETING

MACHINE LEARNING WITH R

BIG DATA ANALYSIS

JAVA FOR FRESHER

BIG DATA & HADOOP ADMINISTRATION

FULL STACK WEB DEVELOPMENT

NODE JS

CLOUD COMPUTING

FRONT END DEVELOPMENT (WITH ANGULARJS)

Watch this short video to know more about ACADGILD.

Disclaimer

This material is intended only for the learners and is not intended for any commercial purpose. If you are not the intended recipient, then you should not distribute or copy this material. Please notify the sender immediately or click here to contact us.

Published by
ACADGILD,
support@acadgild.com

Table of Contents

Motivation

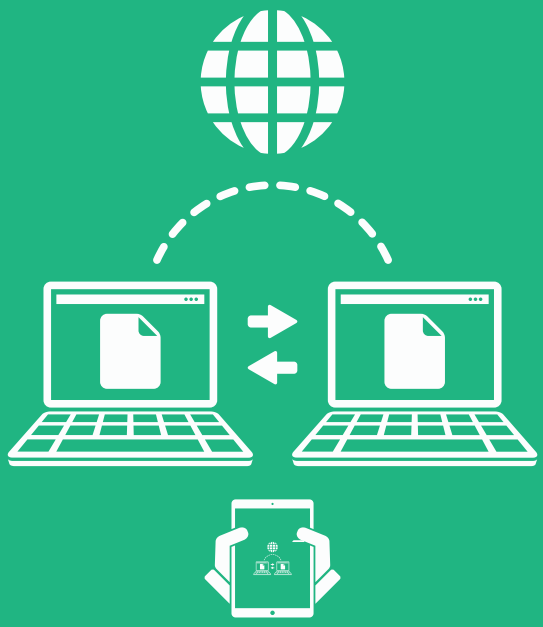
A leading e-commerce **MyCart** company is planning to investigate and analyze products and customer behavior. It is receiving lots of data about different products, registered users, and behavior of users in terms of placing orders and subsequent actions made on the orders. Different products belong to different categories and have different amounts of discounts and profit percentages associated with them. Users are spread across different locations and depending on their behavior, MyCart wants to capture purchase pattern of users and detect for possible fraud activities. It is receiving files on a daily basis and is trying to process them using Big Data tools in order to:

1. Gain competitive benefit
2. Campaign design
3. Possible fraud detection

Creating Hadoop Cluster and Deploying Test Codes

For any Big Data project, a proper environmental setup is mandatory. In real world scenarios, a Hadoop cluster can consist of 40K clusters/nodes. As it is not possible for us to build such a huge setup in an academic lab, we will be going with a minimum of 2 node clusters and simulate the ingestion, cleaning, analysis, and then send the report to the downstream team.

Create a Pseudo-Distributed Node Cluster, and deploy all the test codes using Maven and SBT.



Files Received

MyCart warehouse receives JSON files on a **daily basis** for different types of data.

There are three different types of files:

1. products_info_<timestamp>
2. users_info_<timestamp>
3. user_activity_<timestamp>

products_info_<timestamp

One type of file has a naming convention as **products_info_<timestamp>**

It contains information about different products that are hosted on the e-Commerce websites

The structure looks like:

Field	Description
ID	Unique identifier of every product
Name	Name of the product
Reseller	Name of the reseller who sells that product
Category	Category to which the product belongs to
Price	Selling price of the product
Discount	Discount percentage offered on the product
Profit_percent	Profit percentage to MyCart

Sample Record

```
{"id":"P101", "name":"Men's Trimmer", "reseller":"Philips India",
"category":"Electronic Equipments", "price":2399.00, "discount":10,
"profit_percent":5}
```

users_info_<timestamp>

The second type of file received has naming convention as: **users_info_<timestamp>**

It contains user specific details.

The structure looks like:

Field	Description
ID	Unique identifier of the user
Name	Name of the user
Location	A complex type, including city and state of the user
Age	Age of the user
Price	Selling price of the product
Occupation	Occupation of the user

Sample Record

```
{"id":"U101", "name":"Rakesh", "location":{"city":"MUMBAI","state":
"MAHARASHTRA"}, "age":20, "occupation":"STUDENT"}
```

user_activity_<timestamp>

These files contain different types of activities performed by a user. They contain details of different orders made by different users as well as the cancellation/return(s) if they were performed.

Field	Description
Product_id	Unique identifier of the product
Name	Name of the user
User_id	Unique identifier of the user
Cancellation	boolean value showing whether the order was canceled
Return	boolean value showing whether the order was returned
Cancellation_reason	Reason for cancellation, NA, if the product was not canceled
Return_reason	Reason for return, NA, if the product was not returned
Order_date	Date on which the order was placed (in the format: yyyy-mm-dd)
Shipment_date	Date on which the order was shipped (in the format: yyyy-mm-dd)
Delivery_date	Date on which the order was delivered (in the format yyyy-mm-dd)
Cancellation_date	Date on which the order was cancelled (in the format yyyy-mm-dd)
Return_date	Date on which the order was returned (in the format yyyy-mm-dd)

They have a structure like:

Sample Record


```
{"product_id":"P101", "user_id":"U101", "cancellation":"false", "return":
"true", "cancellation_reason":"NA", "return_reason":"Duplicate product",
"order_date":"2015-09-18", "shipment_date":"2015-09-19", "delivery_-
date":""2015-09-20"", "cancellation_date":"NA", "return_date":""2015-09-25""}
```

Data Preparation

A single directory of HDFS contains all three different types of files. The first challenge is to filter out these records corresponding to the different prefixes that they have.

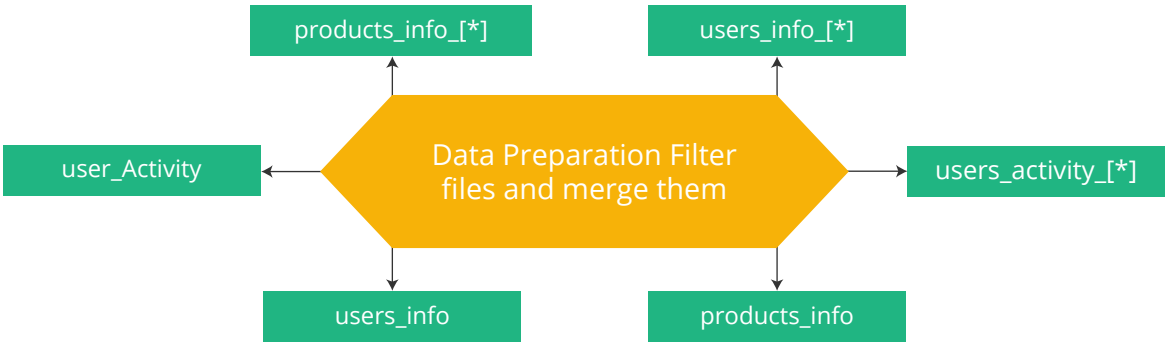
Since many files may be small, it is also required to merge all the small files in a small number of large files, so that subsequent processing speeds up.

Make sure no file is less than 100 MB unless it is a single file of its category (by category, we mean, one of the products_info_*, users_info_* or users_activity_* files).

For example, if we receive three files of the users_info_* category, each of size 50 MB, merge them together to make a single file.

If we receive a single file of products_info_* category with a size of 81 MB, then it cannot be merged and let it be of the same size.

The flow looks as shown below:



Data Enrichment

We are having two HBase lookup tables:

Table name	Row Key	Column Families	Column Qualifiers	Values	Description
Prod _category	Product _id	C	V	Category of the product	For every product, it stores the corres-ponding category.
User _location	User_id	L	V	Location of the user	For every user, it stores the location of the user in the form **'city<tab> state'**

In case, we get NULL for the **product category** or user location in the users _activity_* file, refer to the lookup table and enrich the fields.

Let us assume, we have a value of **C:V** as **'Men Clothes'** for row key **'P110'** in our HBase lookup table **'prod_category'.**

In the **users_activity_*** file, we receive a product **id** as **P110**, but the category is **NULL**, then we can perform a lookup on the basis of product id and retrieve category as **'Men Clothes'**.

Similarly, if we have a user **id** present in **users_activity_*** file, but the **location** is **missing** or is **NULL**, we can retrieve the location by performing a lookup over **user_location** table present in HBase.

Rules for Checking and Validation

In order to increase the trustworthiness of data, there are some rules which every record must adhere to. Below are the rules which every record must follow:

Rule Id	Description
R1	user_id and product_id should not be NULL
R2	order_date should be less than or equal to the shipment_date
R3	age of the user should be a positive number

In case any of the above rules are violated, drop that record from the processing pipeline:

After the validation is complete, identify the number of records failed for each rule.

If the number of invalid records for any of the failures is more than its tolerable threshold, stop the processing and send an e-mail notifying about the same.

Rule	Failure Threshold (in %)
R1	1
R2	2
R3	3

These threshold values are maintained in a separate configuration file.

Data Analysis

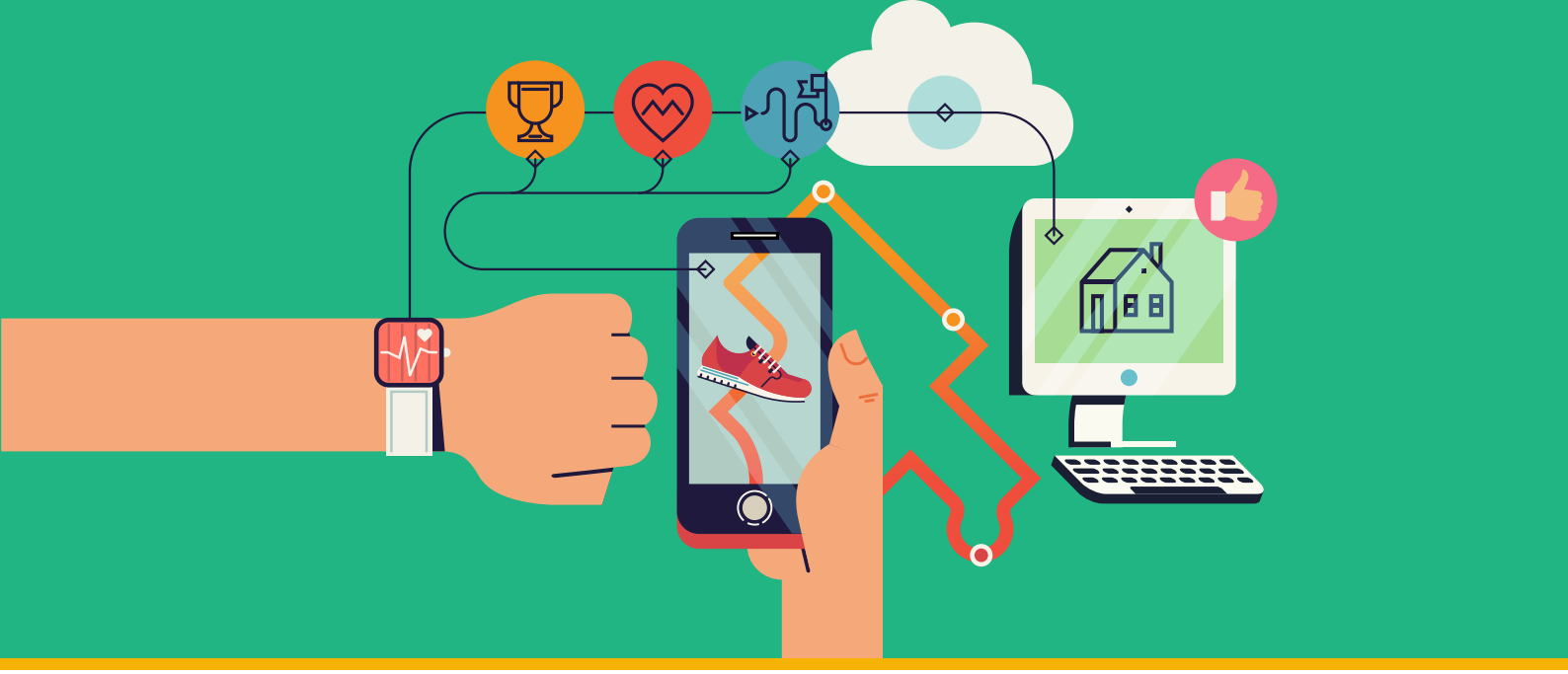
Following details are desired to be known to the management team:

Purchase Pattern Detection

What is the most purchased category for every user? Identify the users with a maximum amount of valid purchase
Which products are generating the maximum profit? [Profit = (Price – Discount)] * profit_precentage)
Which resellers are generating the maximum profit?
Which is most sought-after category corresponding to the very occupation?

Fraud Detection:

Which user has the most amount of returns? What are the valid purchases made by that user?
Which location is getting the most number of cancellations?
Which location is getting the most number of returns?



Data Export

Export to MySQL table

Create a table in MySQL if it does not already exist with the following fields:

Column	Type	Comment
Month	TINYINT	PRIMARY KEY
User_id	VARCHAR(10)	PRIMARY KEY
Category	VARCHAR(20)	Category corresponding to the maximum purchases made by a user
Purchase	DECIMAL(10,2)	Total valid purchases made by a user in the given month

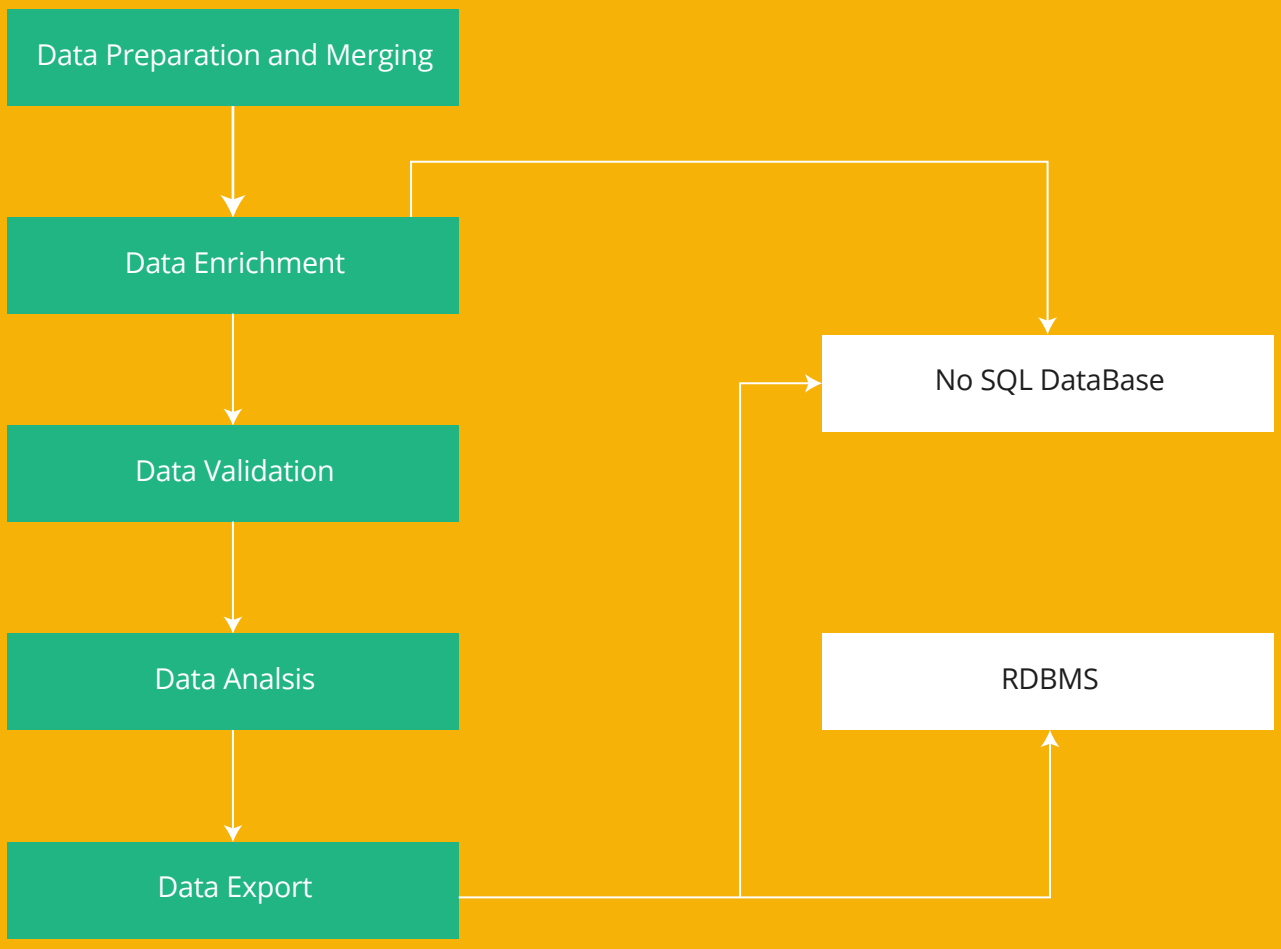
Load final analysis of the result corresponding to the most purchased category in MySQL table.

MySQL table must be incrementally loaded after every month corresponding to the analysis performed on data of the previous month.

This incremental export will take place on the 1st of every month, i.e., on Aug 1, 2016, it will be loaded with the details from July 1, 2016, to July 31, 2016.

Consider using partitioned Hive table or using MapReduce with DBOutputFormat and store the last value of exported months in RDBMS.

Data Flow


Data Preparation and Merging
Data Enrichment
Data Validation
Data Analsis
Data Export
No SQL DataBase
RDBMS

ACADGILD

support@acadgild.com | www.acadgild.com | 8880025025