**Capstone Project 2**

2)**Bank Loan data analysis**:

In this project you are going to analyze the bank loan data set. You need to use HDFS, Hive, Spark and MongoDB.

Part 1:

Get the data to HDFS and use Hive to create the required database.

Part 2: Read the loan table from Hive and create the Dataframe in PySpark and answer the below Questions.

1)Create a dataframe with the Customer IDs having top 5 current loan amounts and save the result in MongoDB?

2)Create a dataframe with the Customer IDs having the lowest 5 current loan amounts and save the result in MongoDB?

3)Create a dataframe of Customer IDs who have taken the Short Term Loan and include their total Current Loan Amount and save the result in MongoDB?

4)Create a dataframe of Customer IDs who have taken the Long Term Loan and include their total Current Loan Amount and save the result in MongoDB?

5)Count how many Bankruptcies are present?

6)Group the data based on Term and find the average monthly debt.

7)Create a dataframe of the customers who have 10 + years experience in their current job. Include their Annual Income

8)Group the data based on Home ownership and Term. Find the aggregated sum of the total current loan.

9)Find the highest credit score for short term and long term customers.

10)Group the data based on years in current job and Home ownership and find the aggregated sum of credit score.

**Solution:**

Here 4 stages require in given project:

satge1:load data in HDFS.

stage2: load into EMR HIVE.

stage3: load into spark.

stage 4: load data into mongodb.

**stage1:HDFS**

To load data into HDFS first we have to create EMR cluster and s3 bucket:

create S3 bucket:

Login to AWS account -->open S3-->create bucket-->create folder-->copy the s3 URL.

Creating Emr cluster:

-->we have to create EMR cluster to connect EMR HIVE and HDFS.

--> opened EMR and created new cluster name"capstonetwo".i choosed default values emr 5.36.0,m5.large,i kept auto termination 4hours.last we have to give keypair. so open EC2

select keypairs-->create keypairs-->'bankkeypair'-->.pem(here automatically .pem file will be download into our pc)-->open puttygen-->load our .pem file-->saveprivatekey-->.ppk file save.

-->choose the keypair name into creating cluster and hit on create cluster.

successfully EMR cluster created.

-->choose master link and edit inbounded rules(SSH& MYIP)save.

-->copy master node path

-->open putty-->paste the master node link-->add-->auth-->choose .ppk file -->open.

by accepting the window it will directly connect to HDFS.

**stage 2:EMR HIVE**

-->hive

create database bank;

use bank;

**Creating external table:**

create external table bank\_loan\_data

(

Loan\_ID string,

Customer\_ID string,

Current\_Loan\_Amount string,

Term string,

Credit\_Score string ,

Annual\_Income string ,

Years\_in\_current\_job string,

Home\_Ownership string,

Purpose string,

Monthly\_Debt string,

Years\_of\_Credit\_History string,

Months\_since\_last\_delinquent string,

Number\_of\_Open\_Accounts string,

Number\_of\_Credit\_Problems string,

Current\_Credit\_Balance string,

Maximum\_Open\_Credit string,

Bankruptcies string,

Tax\_Liens string

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

LOCATION 's3://capstonetwo/data/'

TBLPROPERTIES("skip.header.line.count"="1");

show tables;

select \* from bank\_loan\_data limit 30;

set .hive.cli.data.header=true;

**stage 3: pyspark**

in emr hive -->quit;

sprk-shell

:quit

pyspark

sc

:quit

pyspark

bdf=spark.sql('select \* from bank\_loan\_data')

bdf.show(5)

**1**)**Create a dataframe with the Customer IDs having top 5 current loan amounts and save the result in MongoDB?**

top5loans=spark.sql('select customer\_id from bank\_loan\_data order by curent\_loan\_amount desc limit 6)

op5loans.show()

Here i am getting header as row value so:

rdd=top5loans.rdd

header=rdd.first()

newrdd=rdd.filter(lambda line:line !=header)

top5loans=spark.createDataFrame(newrdd,top5loans.schema)

top5loans.show()

**stage 4:Mongodb**

installed mongodb from google:create a file by using the link

-->open file insert past the file data and :wq for exit the file.

to install:

sudo yum install -y maongodb-org

to start monogodb:

sudo systemctl start mongod

to check mongod:

sudo systemctlstatus mongod

mongosh---------its connect tomongodb

show dbs;

**installation of mongodb compass:**

from google i installed mongodb compass-->downlod&install-->downloads page-->windows-->download

successfully downloaded mongodb compass.

now open mongodb compass:-->new connection-->paste the host link from mongosh-->save-->name-->connet.

here i am getting error . I am unable to connect mongodb compass.

**Pyspark to mongodb:**

quit()

pip install pymongo

pyspark

import pymongo

from pymongo import MongoClient

Client=MongoClient('mongodb://127.0.0.1',27017)

db=client['bank']

bankcollection=db.bank

import numpy as np

top5loans-list=[list(row) for row in top5loans.collect ()]

top5loans\_list

**Converting list to dictionary**

import numpy as np

ar=np.array(top5loans-list)

dict={}

for i, column in enumerate(top5loans.columns):

dict[column]=list (ar[:,i])

dict

**To load into mongodb:**

dict['customer\_id'}

dict['current\_loan\_amount']

for key,value in dict:

dict

2)Create a dataframe with the Customer IDs having the lowest 5 current loan amounts and save the result in MongoDB?

lowest5loans=spark.sql('select customer\_id from bank\_loan\_data order by curent\_loan\_amount limit 5')

3)Create a dataframe of Customer IDs who have taken the Short Term Loan and include their total Current Loan Amount and save the result in MongoDB?

shortterm=spark.sql('select customer\_id,sum(current\_loan\_amount) total\_current\_loan\_amount from bank\_loan\_data where term='short term' group by customer\_id,term')

4)Create a dataframe of Customer IDs who have taken the Long Term Loan and include their total Current Loan Amount and save the result in MongoDB

longterm=spark.sql('select customer\_id,sum(current\_loan\_amount) total\_current\_loan\_amount from bank\_loan\_data where term='long term' group by customer\_id,term')

5)Count how many Bankruptcies are present?

bankrupt=spark.sql('select count(bankruptcies) from bank\_loan\_data')

6)Group the data based on Term and find the average monthly debt.

averagemonthly=spark.sql('select avg(montly\_debt) from bank\_loan\_data group by term')

7)Create a dataframe of the customers who have 10 + years experience in their current job. Include their Annual Income

tenyea=spark.sql('select customer\_id,annual\_income from bank\_loan\_data where years\_in\_current\_job='10+ years'')

8)Group the data based on Home ownership and Term. Find the aggregated sum of the total current loan.

totalcurrentloan=spark.sql('select sum(current\_loan\_amount) from bank\_loan\_data group by home\_ownership,term')

9)Find the highest credit score for short term and long term customers.

creditscore=spark.sql('select customer\_id,credit\_score from bank\_loan\_data where term='short term' and 'long term' order by credit\_score')

10)Group the data based on years in current job and Home ownership and find the aggregated sum of credit score.

sum\_creditscore=spark.sql('select sum(credit\_score)total\_credit\_score from bank\_loan\_data group by home\_ownership and years\_in\_current\_job').