**My Github Link:** <https://github.com/Shiwenn>

**Sample 1:**

Create the external table for the entire data of 2018

CREATE EXTERNAL TABLE tripdata\_sample

(vendorid INT,

tpep\_pickup\_date\_time TIMESTAMP,

tpep\_dropoff\_date\_time TIMESTAMP,

passenger\_count INT,

trip\_distance DOUBLE,

RatecodeID INT,

store\_and\_fwd\_flag CHAR(1),

PULocationID INT,

DOLocationID INT,

payment\_type INT,

fare\_amount DOUBLE,

extra FLOAT,

mta\_tax FLOAT,

tip\_amount DOUBLE,

tolls\_amount DOUBLE,

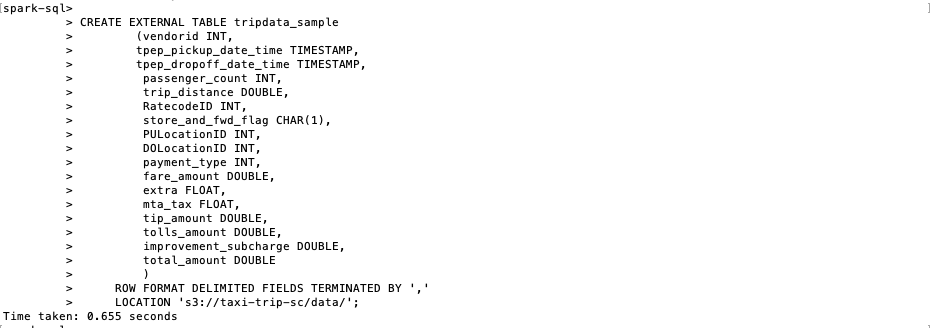
improvement\_subcharge DOUBLE,

total\_amount DOUBLE

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

LOCATION 's3://taxi-trip-sc/data/';



Query 1:

SELECT Month(tpep\_pickup\_date\_time), COUNT(\*)

FROM (

SELECT tpep\_pickup\_date\_time, RatecodeID, passenger\_count, payment\_type

FROM tripdata\_sample

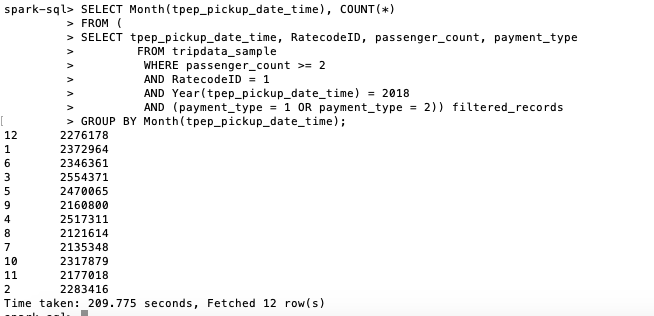
WHERE passenger\_count >= 2

AND RatecodeID = 1

AND Year(tpep\_pickup\_date\_time) = 2018

AND (payment\_type = 1 OR payment\_type = 2)) filtered\_records

GROUP BY Month(tpep\_pickup\_date\_time);



Screenshot of AWS’ Cluster



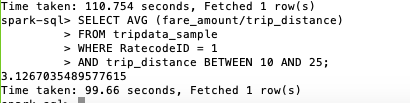
Query 2: Calculate the result as it required

SELECT AVG (fare\_amount/trip\_distance)

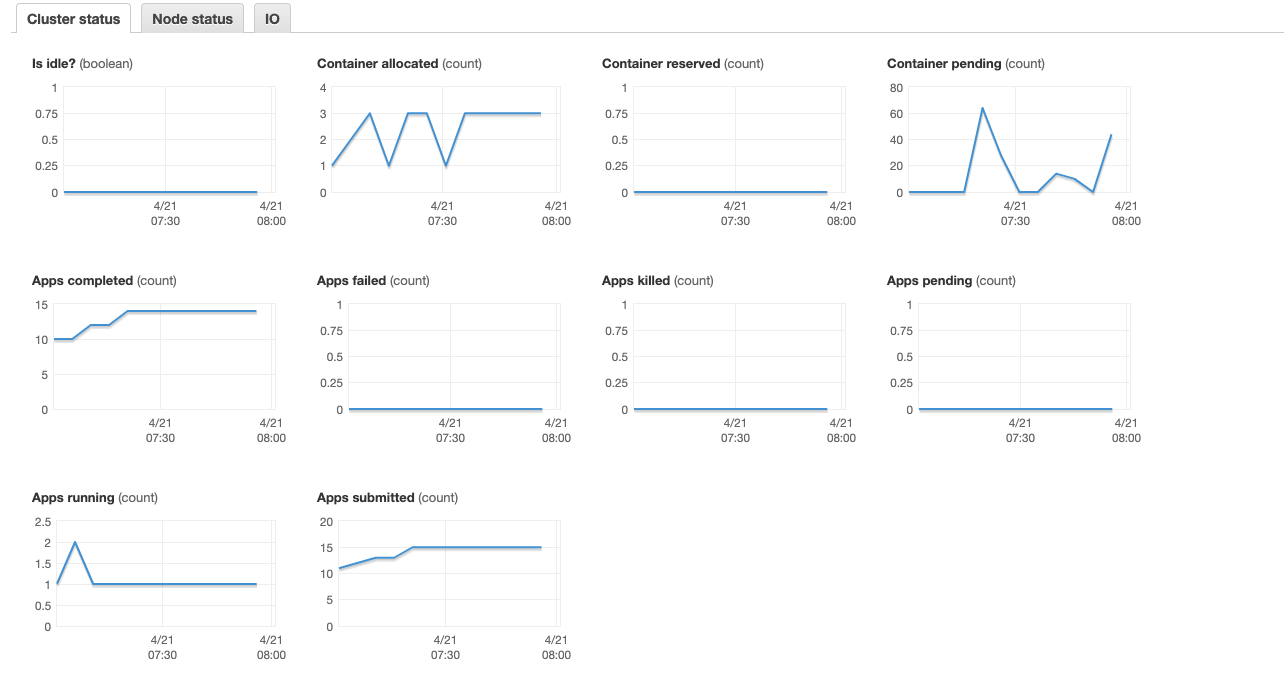
FROM tripdata\_sample

WHERE RatecodeID = 1

AND trip\_distance BETWEEN 10 AND 25;



Screenshot of AWS - Cluster



Query 3:

SELECT day\_of\_week, avg(number\_of\_rides) as avg\_rides

FROM

(SELECT dt, day\_of\_week, count(\*) as number\_of\_rides FROM

(SELECT \*, date\_format(tpep\_pickup\_date\_time, 'y-MM-dd') as dt,

date\_format(tpep\_pickup\_date\_time, 'EEEE') as day\_of\_week from

tripdata\_sample

WHERE passenger\_count = 1) single\_rider

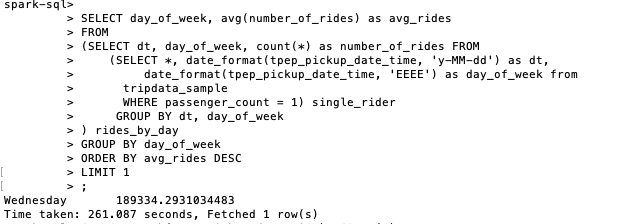
GROUP BY dt, day\_of\_week

) rides\_by\_day

GROUP BY day\_of\_week

ORDER BY avg\_rides DESC

LIMIT 1



Screenshot of AWS - Cluster



**Sample 2:**

> # Display the text output from each decision tree.

> # You're looking for the tree that minimizes xerror.

> # xerror is the relative misclassifica .... [TRUNCATED]

###### Display the text output from each decision tree: ######

> printcp(MyTree)

Classification tree:

rpart(formula = payback ~ age + sex + region + income + married +

children + car + save\_act + current\_act + mortgage, data = trainingSet,

method = "class", control = rpart.control(minsplit = 25,

cp = 0.005))

Variables actually used in tree construction:

[1] age children income married mortgage region save\_act

Root node error: 133/300 = 0.44333

n= 300

CP nsplit rel error xerror xstd

1 0.142857 0 1.00000 1.00000 0.064695

2 0.105263 2 0.71429 0.75940 0.061542

3 0.075188 4 0.50376 0.60902 0.057817

4 0.045113 5 0.42857 0.47368 0.053044

5 0.027569 6 0.38346 0.42857 0.051089

6 0.018797 9 0.30075 0.41353 0.050391

7 0.005000 11 0.26316 0.39098 0.049296

> # Show the result graphically

> plotcp(MyTree, minline = FALSE)

> #######################################################################

> ###### Pruning the tree ######

> .... [TRUNCATED]

> #######################################################################

> ###### Evaluating Classification Accuracy ######

> .... [TRUNCATED]

> predValidation <- predict(MyTree, validationSet, type="class")

> # Generating Confusion Matrices for the traing and validation sets:

> cat("\n###### Confusion Matrix for the training set ######\n")

###### Confusion Matrix for the training set ######

> table(Predicted=predTraining,Observed=trainingSet[, 12] )

Observed

Predicted 0 1

0 153 21

1 14 112

> # trainingSet[, 2] tells R the 2nd column of the dataset has the outcome variable (Survived)

> # If your outcome variable is in a different column, .... [TRUNCATED]

###### Confusion Matrix for the validation set ######

> table(Predicted=predValidation,Observed=validationSet[, 12] )

Observed

Predicted 0 1

0 123 36

1 36 105

> # validationSet[, 2] tells R the 2nd column of the dataset has the outcome variable (Survived)

> # If your outcome variable is in a different column .... [TRUNCATED]

> predRateValidation <- mean(predValidation == validationSet[, 12])

> # This stops R from writing any more to the text output file.

> sink()

**Sample 3:**

Proposal: Targeted Incentives & Premium Conversions - prepared by Gotham Musalytics

We proposed that Phandellafy radio service be upgraded to induce trials of the new service and test its conversion rate to make sure free offerings are appropriate enough to maintain the benefits brought by product upgrades. One thing is that we will upgrade its radio service with a few new features to continuously attract new users to use the free version, helping the product generate traffic and build a larger target for the premium conversion. Another thing is that we will test whether the number of features offered for free is appropriate, which further helps Phandellafy reach the ideal quarterly conversion rate (from freemium to premium) – it can’t be too low like 1% or too high like 50%, it can depend on the Phandellafy’s financial performance and goal.

First, we set a product development goal for Phandellafy’s new service. The premium version will have 12 more features [feature A-L] than the current radio service. (Note: the complications of the features are ranked in ascending order, A<L) And the first 6 features [feature A-F] are ready to be added to upgrade the free Phandellafy to attract new freemium users. Simultaneously, features G-L will convert older users into premium users.

We will plan to roll out a free version with subtle differences in the first three quartiers – Quarter 1(Q1) with features A-D, Q2 with features A-E, Q3 with features A-F, and at the same time, we will collect data generated within the three quarters to calculate the number of new freemium users and the number of new premium users transitioned from old freemium users. By further calculating the traffic generated by free users and quarterly revenue from premium, Phandellafy would decide how many features the radio service should finally add which can create a best combination of new users and the conversion rate, making a best balance to generate ad revenue and stable rise of subscription revenue.

When Phandellafy gets the appropriate conversion rate for its financial vision, we will build a predictive model to offer a one-month free trial of premium incentives to prospective users to maximize the conversion effect. By collecting all the attribute data (such as demographic, level of usage of the service, tenure with the service…) from successfully converted older users in the three quarters, we get to know which attributes are most influential to the conversion. And we will assign a score range for the correlations (degree of influence) to define the total correlation between each old user and conversion (0<the level of attribute A \* the correlation score of attribute A<1). Because we have an understanding of the ideal conversion rate, we will rank all the correlation score of the older users and choose the certain percentage of targets with higher scores to send the free trial offer.