**11676 – Big Data Analytics**

**lneves**

**HW 4 – Cassandra**

For this assignment, I have created a separate static class CassandraCluster to handle the Cassandra requests. I have adapted my code to populate the database from train and test csv files, and for the random forest to read the instances from the database. Also, error values for each experiment were persisted to the database.

I have created a cyber\_sec database with all the features, labels, an id and the type (train or test) for each instance. I have also created a header\_order table, because my code was using this information. With this, I can map the name of the columns to the right features. For the result, I have created a table with the exp name, the type(train or test), the number of trees and the error. Below is the output for my describe method on Cassandra cqlsh. My code is on my git repo.

CREATE KEYSPACE bigdata\_class WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': '1'} AND durable\_writes = true;

CREATE TABLE bigdata\_class.cyber\_sec (

id text PRIMARY KEY,

attack\_cat text,

count text,

ct\_flw\_http\_mthd text,

ct\_state\_ttl text,

diff\_srv\_rate text,

dinpkt text,

djit text,

dload text,

dmean text,

dpkts text,

dst\_bytes text,

dst\_host\_count text,

dst\_host\_diff\_srv\_rate text,

dst\_host\_srv\_diff\_host\_rate text,

dttl text,

duration text,

hot text,

is\_ftp\_login text,

is\_host\_login text,

label text,

land text,

num\_access\_files text,

num\_compromised text,

num\_failed\_logins text,

num\_file\_creations text,

num\_outbound\_cmds text,

num\_shells text,

rate text,

rerror\_rate text,

response\_body\_len text,

root\_shell text,

same\_srv\_rate text,

sinpkt text,

sload text,

smean text,

spkts text,

src\_bytes text,

srv\_diff\_host\_rate text,

sttl text,

synack text,

trans\_depth text,

type text,

urgent text,

wrong\_fragment text

) WITH bloom\_filter\_fp\_chance = 0.01

AND caching = '{"keys":"ALL", "rows\_per\_partition":"NONE"}'

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy'}

AND compression = {'sstable\_compression': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99.0PERCENTILE';

CREATE INDEX cyber\_sec\_type\_idx ON bigdata\_class.cyber\_sec (type);

CREATE TABLE bigdata\_class.result (

experiment text,

type text,

tree\_num int,

error double,

PRIMARY KEY ((experiment, type), tree\_num)

) WITH CLUSTERING ORDER BY (tree\_num ASC)

AND bloom\_filter\_fp\_chance = 0.01

AND caching = '{"keys":"ALL", "rows\_per\_partition":"NONE"}'

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy'}

AND compression = {'sstable\_compression': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99.0PERCENTILE';

CREATE INDEX result\_type\_idx ON bigdata\_class.result (type);

CREATE TABLE bigdata\_class.header\_order (

name text,

ord int,

PRIMARY KEY (name, ord)

) WITH CLUSTERING ORDER BY (ord ASC)

AND bloom\_filter\_fp\_chance = 0.01

AND caching = '{"keys":"ALL", "rows\_per\_partition":"NONE"}'

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy'}

AND compression = {'sstable\_compression': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99.0PERCENTILE';