COVID 19 REPORT

Danhong Tang

(200471292)

Introduction:

This database is developed by Toronto Public Health to report an ongoing COVID-19 outbreak. This data set contains demographic, geographic, and hospitalization information for all cases reported by Toronto Public Health since January 2020. The data are extracted from the provincial Case & Contact Management System (CCM). The data will be completely refreshed and overwritten on a weekly basis.

Background

Ontario health officials are reporting 3,301 new cases of COVID-19 on 18th Dec. The Ontario government announced new COVID-19 restrictions on 17th Dec due to the unprecedented spread of the Omicron variant. The province's hospital system is dealing with the overloaded Omicron patience for bracing for an overwhelming wave of COVID-19 patients with diminished staffing levels and limited medical resource. So a simulator that can predict number of patients that may possibly enter into a hospital and fatalities will help hospitals to prepare and allocate their resources.

Objective

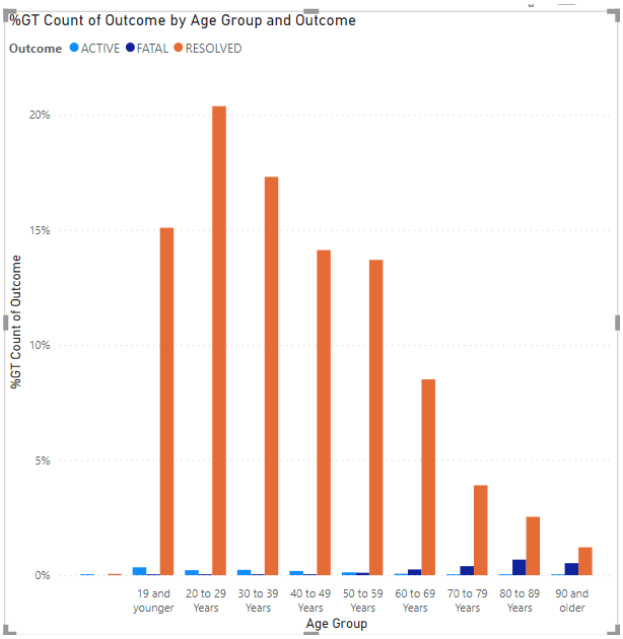
Build up an algorithm model to evaluate and predict number of patients that may possibly enter into a hospital and fatalities of them.

Tool

- Use Spark on Google Cloud Platform to build up evaluation and prediction algorithm model.
- Use PowerBI to finish visualization.

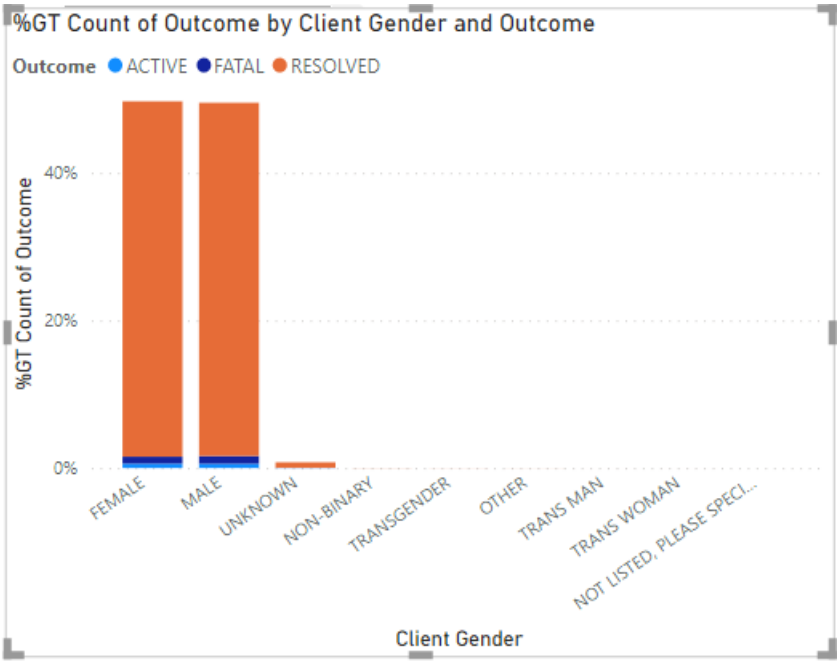
Visualization and Analysis

- Age group & Outcome Chart



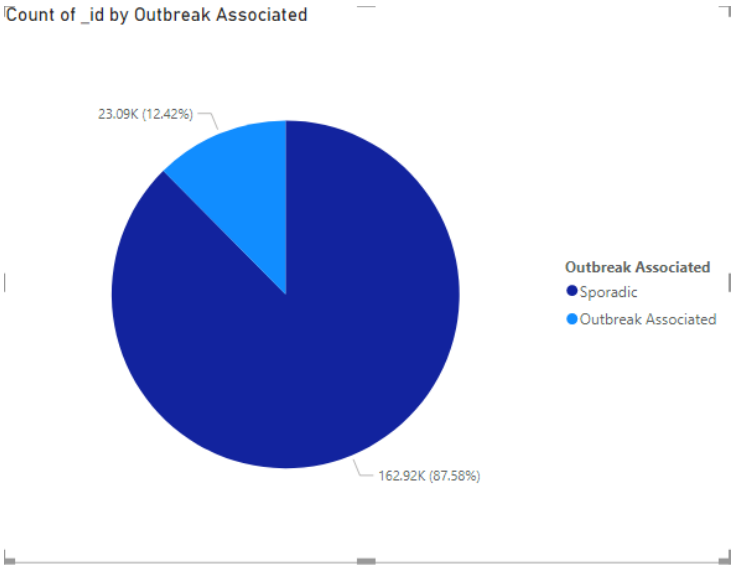
- It can be seen from the chart that age from 20-29 has the highest infection rate among all age group , which might caused by they are working and have frequent social activities.
- The fatal rate generally increase with age and group of 80-89 years has the highest fatal rate 0.68%.

➢ Gentle & Outcome Chart



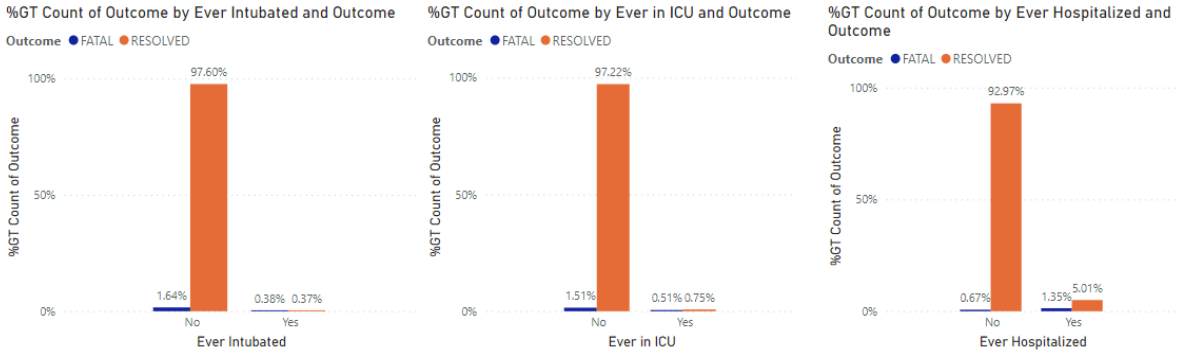
- The male and female patients are almost the same with each other, but male fatal rate 1.04% is slightly higher than female fatal rate 0.93%.

➢ Outbreak Associated Chart



- We can see that sporadic cases (87.58%) is the most common outbreak associated type.

➢ Fatality Rate in 3 Situations Chart



	Among all cases%	Fatal among them%
Hospitalized	6.36%	21.23%
ICU	1.26%	40.48%
Intubated	0.75%	50.67%

- This meaningful comparison tells us that 6.36% cases are hospitalized, among them fatality rate 21.23%; Only 1.26% cases are sent to ICU, but the fatality rate increase to 40.48% ; Even worse, people who were intubated had almost half of them died.
- This analysis result is scary that it told people the best way to protect is to keep social distance and wear mask, and when getting serious symptoms need to be hospitalized, people have to face higher fatality rate.

- On the other side, China had much lower fatality rate among patients who were hospitalized among the same period of time, the reason of that might be there were much lower number of total cases so patients could get more medical resources.

Machine learning algorithm

➢ **Preparation dataset(Appendix 1,2)**

1. Download datasets
2. Index the string variables
3. Upload file to Google cloud
4. Launch Spark shell and import libraries
5. Load dataset
6. Select and cast relative columns into Int, also filter off “AVTIVE” cases in “outcome” that are not meaningful.

➢ **Random Forest machine learning (Appendix 3,4,5)**

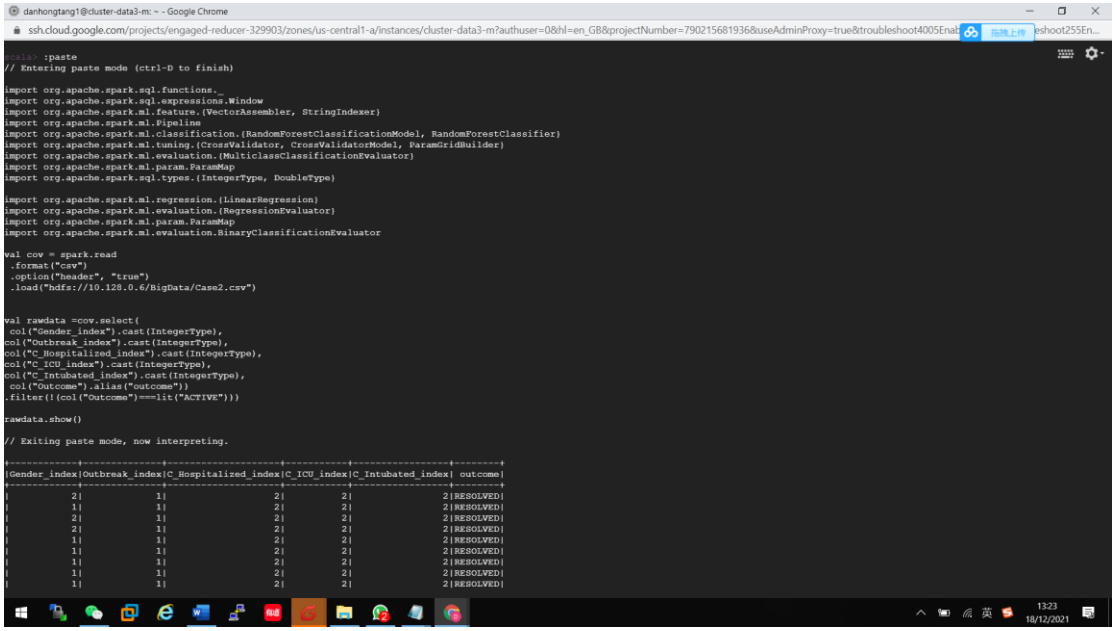
1. Split our dataset into training and test data typical 80 20
2. Assemble features using vector assembler
3. Random forest： Create a new random forest object, give feature as a vector, give the label as "outcome"
4. Set up pipeline
5. Evaluate the model: using MulticlassClassificationEvaluator to compare "outcome" to the prediction column
6. hyper parameters:
   --MaxDepth is an array with two values, 5, 10 which is the limit on how deep we can construct the tree
   --Impurity which we give as entropy
7. Cross validate model
   -- Cross validator will divide the training data set into 3
   -- Each fold is coupled with the paramters for each type: Fold 1 is tried with max depth 3 and entropy and then fold 1 is again tried but this time with max depth 5 and entropy
   -- the best model is picked
8. Training dataset, give the best model
9. Evaluate the model and print accuracy 0.98

Reference:

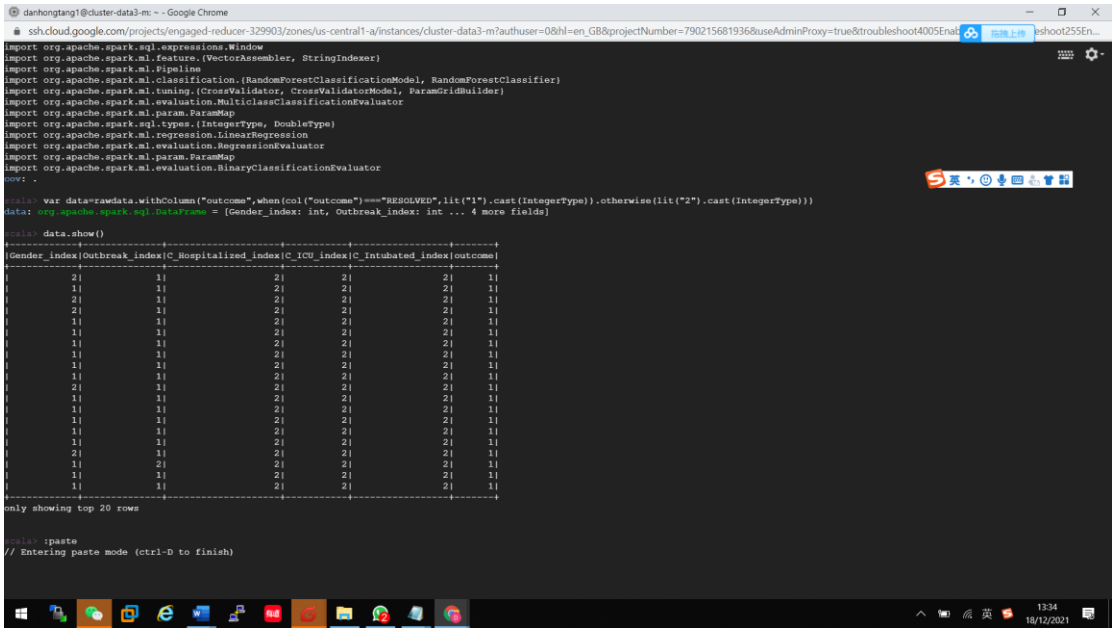
https://toronto.ctvnews.ca/ontario-reports-3-301-new-covid-19-cases-four-additional-deaths-1.5712753

Appendix

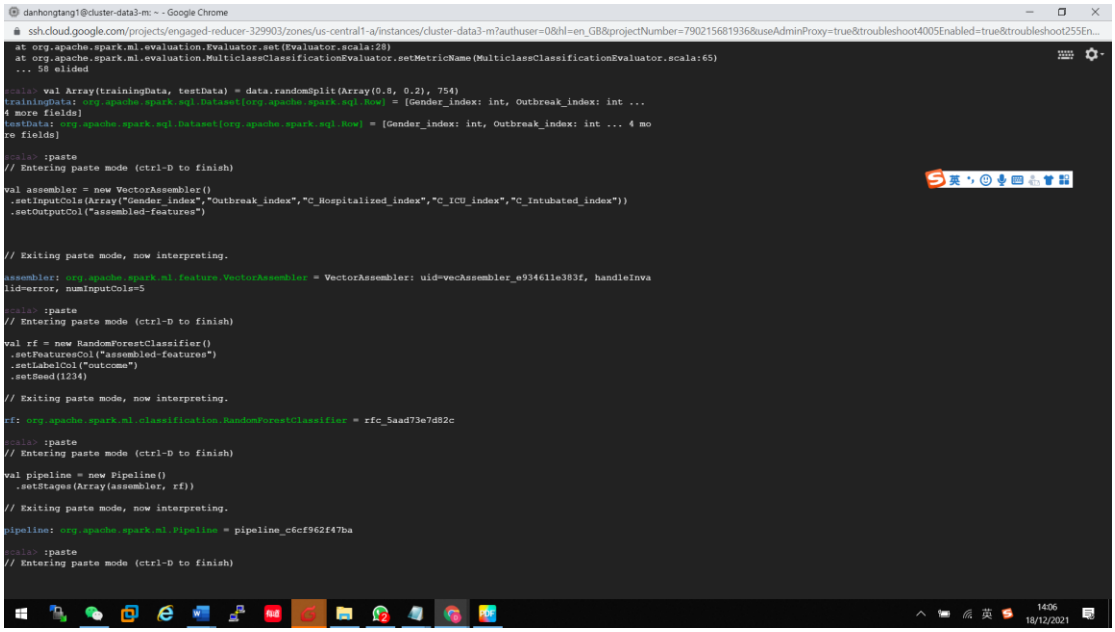
1



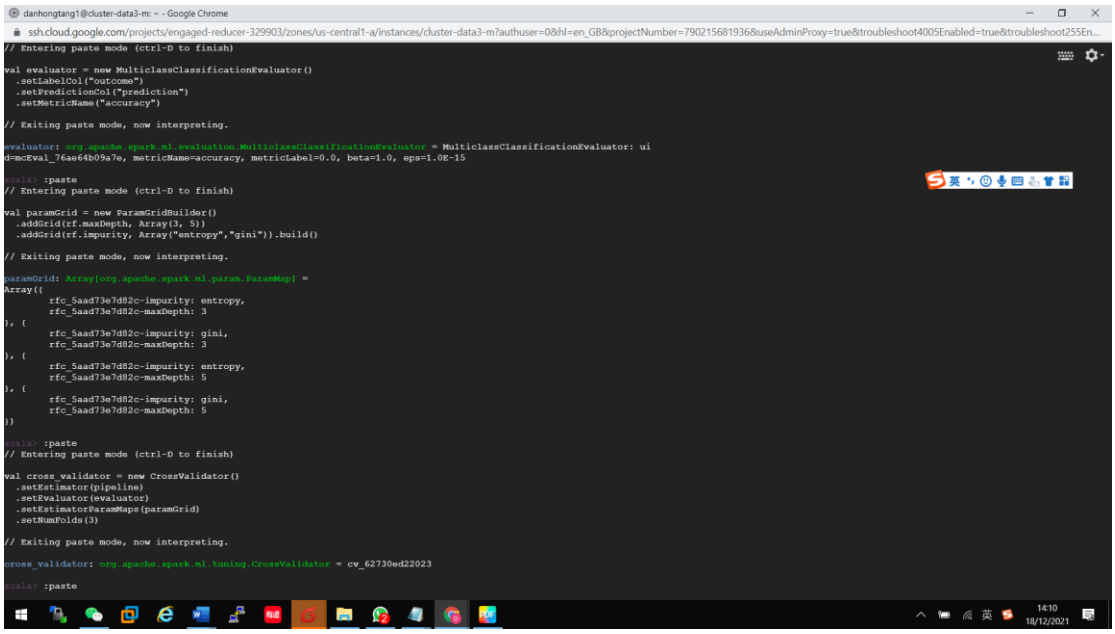
2



3



4



5

```
scala> val Model = cross_validator.fit(trainingData)
Model: org.apache.spark.ml.tuning.CrossValidatorModel = CrossValidatorModel: uid=cv_62730ed22023, bestModel=pipeline_c6cf962f47ba, numFolds=3

scala> val predict = Model.transform(testData)
predict: org.apache.spark.sql.DataFrame = [Gender_index: int, Outbreak_index: int ... 8 more fields]

scala> val accuracy = evaluator.evaluate(predict)
accuracy: Double = 0.9802446933158941

scala> println("Accuracy based on test data = " + accuracy)
Accuracy based on test data = 0.9802446933158941

scala>
```