School of Continuing Studies

YCBS 299 – Data Science Capstone Project
Summer 2021

Predict high fire risk areas in the city of Montreal

Team #02:

Zhongdi Wang
Vy Kiet Le

Table of Contents

1. Problem Statement

Every year, firefighters, for the sake of protecting lives and civilians' properties, have encountered many life-threatening rescuing missions and challenges, and have incurred many injuries, ranging from simple cuts and bruises to burns and pains and even deaths.

Along with many cities in the world, the city of Montreal is tasked with the prediction of high fire risk areas in the city. Using a statistical approach based on machine learning models, our team wants to predict fire risk score for each borough/sector of the city of Montreal for the next months defined on three levels: low, medium, and high.

The prediction of high fire risk areas in the city will help fire departments to focus their attention when determining areas for fire inspections of properties and for resources allocations to provide fire prevention and safety measures, thus reducing in the long run the number of occurrences of fire incidents.

2. Data Sources

The datasets listed in the Table 2 below that we use for our project are directly available on the website of the city of Montreal (https://donnees.montreal.ca/**).**

	Features				Response
Dataset	**Interventions 2015-2021**	**Crime Incidents 2015-2021**	**Property Assessment**	**Administrative limits of Montreal Area**	**Fire incidents 2015-2021**
Fields used	Monthly count of interventions in the previous Month of incident	Monthly count of crimes in the previous month	Number of floors above ground level Number of apartments in the property Average Property Age Total Building Area Total Land Area	Code marmot for each administrative area in Montreal	Monthly count of interventions that resulted in Fire
Records	741,157	196,469	497,101	34	
% records retained	100% (dropped 3 records where area is not provided)	83% (dropped records where location is not provided)	66% (dropped records where building area is not provided)	100%	

Table 2. Datasets used and features retained for modelling

Please note that records of interventions and crimes are used towards counting of each for each month from January 2015 to August 2021. We have created a new variable called "Fire incidents" which are counts of interventions when they resulted in a Fire incident (please refer to the next section on *Data exploration and cleaning* for more details).

The geographical location of each record where available is then used for association to each of the 34 administrative areas in the city of Montreal. This is the field used to merge the various datasets into one output file ready to be used for Python for the modelling stage.

3. Data Exploration and Cleaning

In order to work with adequate data, we have ensured the necessary data preprocessing steps in Alteryx tool.

Fire Incidents 2015-2021:
First off, our goal is to predict a fire risk score for each area in Montreal city for a given month. It is important that we define which types of interventions identified by the fire department are relevant in this context. Among the types classified by the city, we refer to a fire incident when the intervention is described either as "*Incendie*" or "*Autre feu*". A new field was created to indicate whether the intervention resulted in a Fire incident or not, then it's used towards aggregation per area and per month.

When plotting the correlation matrix between the features listed in the previous section, we have seen high correlation between the fire incidents count and the predictor variables such as number of interventions, number of crimes, number of building floors above ground as well as number of accommodations.

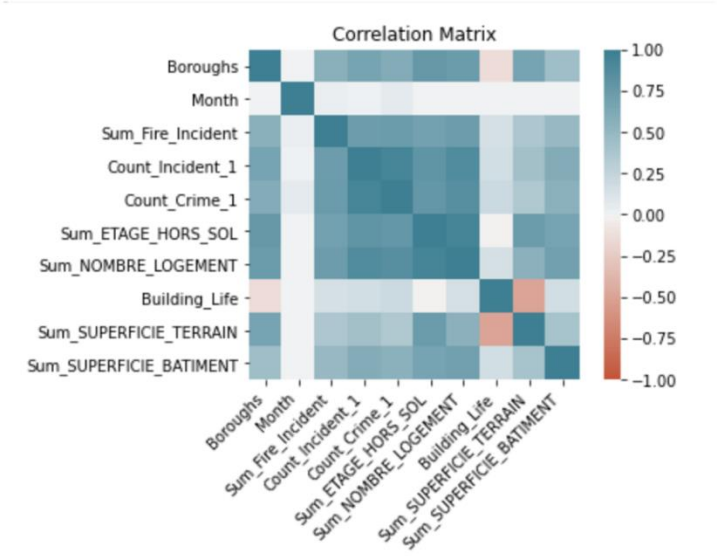

Figure 3 Correlation Matrix

Interventions 2015-2021:
We have removed 3 records where the administrative area was not provided at the time of the intervention. This represents <0.001% of the interventions dataset.

Crime Incidents 2015-2021:
We have included crime incidents in our model because we believe that a fire can be a result of any type of crime incidence. In this dataset, we have removed records where the location (longitude, latitude) was not provided at the time of the incident. This represents 17% of the crime incidents dataset.

Property Assessment:
We have also included details on the properties in the city of Montreal because we believe that fire occurrence is higher in administrative areas with huge property land area and higher number of accommodations. In this dataset, we have removed records where numerical features, such as building area, number of floors above ground, or number of accommodations, were provided as null. This represents 34% of the property assessment dataset.

4. Feature Engineering

Fire Incidents:
As described in the previous section, a new field is created for the monthly count of Fire incidents based on the interventions dataset.

Average Property Age:
For each administrative area in Montreal city, we have calculated the average life of all the buildings and properties within that area.

Lag features (interventions, crimes):
We have also introduced two new variables, which are the counts of interventions in the previous month and the counts of crimes in the previous month. Under the belief that places with a history of interventions and crimes, fire incidents can have a higher probability of occurring in the future. These two new lag features are crucial in our prediction model. In addition, past months' data are readily available for any kind of prediction in the future.

5. Tools and Techniques Used

5.1. Tools

Below is an overview of our solution approach to the business problem depicting the tools used.

- We use various open datasets that are readily and publicly available from the website of the city of Montreal. We employ preprocessing techniques on the data in Alteryx in order to output a merged dataset file.
- This file is then used in Python for generating models. Based on our evaluation of the models' metrics, we will choose the best prediction model for our data.
- We then use Tableau to help visualize the prediction of fire risk for each borough in the city for the next months.

Figure 5.1 Workflow of the solution

5.2. Categorization of the fire risk

Based on the number of fire incidents, we categorized the fire risk into three levels: low, medium, and high. The way how the levels are defined is shown in the Figure 5.1 below, it is defined this way to make sure that the total number of fire incidents under each level is almost the same to avoid imbalanced classification, and the accuracy from the models can be accurate and reliable.

Count of Fire incidents	Fire risk level
[0, 2]	1 (low)
[3, 15]	2 (medium)
>15	3 (high)

Figure 5.2 Categorization of the fire risk

5.3. Simple encoding

In the raw dataset, we have non-numeric values for feature Boroughs. We converted the 34 administrative areas into numeric values (0 to 33) using simple encoding technique so this feature can be used later by the model.

5.4. Data normalization

There are in total 9 features selected in our project and their values don't have the same range. We normalized all 9 features to ensure they are all in a common scale without distorting differences in the ranges of values.

6. Summary of Modelling Techniques Evaluated

We split the dataset into 70% training dataset and 30% testing dataset randomly and they are used in the models explained in the below sections.

6.1. Baseline Model

A baseline is a model that is easy to setup and can have a reasonable chance of providing us a decent result. This model is the one to which we will compare all other models and help us evaluate the performance of other models.
We created the baseline model with Decision Tree algorithm and with the simplest feature selection (Month, Total number of building floors & Total number of accommodations) and parameters.

We selected Accuracy and R-score as the main evaluation metrics and produced the classification report which provide us more metrics which we can use. As we can see in Figure 5.2, the Baseline model has an accuracy of 0.744 and a R-score of 0.528, which is a decent performance.

```
[[195  74   0]
 [ 43 298  18]
 [  0  71 107]]

 Accuracy:  0.7444168734491315

 R Square Score: 0.5283081582154595

 Classification Report:
               precision    recall  f1-score   support

           1       0.82      0.72      0.77       269
           2       0.67      0.83      0.74       359
           3       0.86      0.60      0.71       178

    accuracy                           0.74       806
   macro avg       0.78      0.72      0.74       806
weighted avg       0.76      0.74      0.74       806
```

Figure 6.1 Performance of baseline model

6.2. Random Forest Classifier

We first used all 9 features and a max depth of 15 to train the model; it has an accuracy of 0.784 and a R-score of 0.588. We plotted the feature importance to have an understanding of which features have more importance.

We used **SelectFromModel** function in Python which can select the most important features automatically based on the threshold defined by the function itself. Five features are selected (Life of building, Total number of accommodations, Total number of crimes, Total number of interventions, and Month) and after implementing this technique, the model has a lower accuracy (0.78) and a higher R-score (0.581).

6.3. Support Vector Machine

We first used all 9 features to train the model; it has an accuracy of 0.696 and a R-score of 0.439. We plotted the feature importance to have an understanding of which features have more importance.
We only selected the four most important features (Total number of accommodations, Total number of crimes, Total number of interventions and Total number of building floors) to retrain the model, and it has a slightly higher accuracy (0.701) and the same R-score (0.448).

6.4. K-Nearest Neighbors Classifier

We first used all 9 features with number of neighbors of 8 to train the model, it has an accuracy of 0.777 and a R-score of 0.574.

To select the best parameters to train the model, we gave a range of values to all three parameters: *Leaf_size*, *n_neighbors* and *p*, then used **GridSearchCV** function to select the best combination of parameters, which can provide the best result. With the value of parameters selected (*Leaf_size*=1, *n_neighbors*=11 and *p*=1), it has a slightly higher accuracy (0.784) and R-score (0.588).

6.5. XGBoost Classifier

We first used all 9 features to train the model; it has an accuracy of 0.793 and a R-score of 0.604. We plotted the feature importance to have an understanding of which features have more importance.

We used **SelectFromModel** function, and we trained the model with all possible parameters *threshold* and *norm_order* to get the best parameters. With the best parameters (*threshold* = 0.019 and *norm_order* = 8) and the 7 features selected (Month, Total number of intervention, Average life of building, Total number of crimes, Total number of accommodations, Boroughs and Total number of building floors), the model has a better accuracy (0.805) and a higher R-score (0.627).

The evaluation (Accuracy & R-score) of models' performance is summarized in the Table 6.5 below and we can see that the XGBoost with 7 features has the best accuracy score and a high R-score. This model has a better performance than the baseline model and is used for our prediction for September 2021. For more details about how other evaluation metrics are, please find in Appendix 1-8.

Model	Decision Trees (DT) (Baseline)	Random Forest (RF)	K-Nearest Neighbors (KNN)	Support Vector Machine (SVM)	XGBoost (XGB)
Accuracy	• 74.4% with an R-score of 0.528 for a model based on 3 features	• 78.4% with an R-score of 0.588 for a model based on 5 features	• 78.4% with an R-score of 0.588 for a model based on all 9 features	• 70.1% with an R-score of 0.488 for a model based on 4 features	• 80.5% with an R-score of 0.627 for a model based on 7 features
Advantages	• Generally fast training time and simple tool.	• Generally fast training time and simple tool. • Decorrelate trees to reduce variance of predictions.	• No training period before making predictions • Easy to implement	• Works well if there is clear margin of separation between classes	• XGBoost performs very well on small dataset with subgroups and structured datasets with not too many features
Disadvantages	• Can overfit data and slow down predictions if there is a large number of trees in the algorithm	• Can overfit data and slow down predictions if there is a large number of trees in the algorithm	• Does not work well with large datasets (slow performance) • Does not work well high dimensions (features)	• Not suitable for large data sets • Does not perform well if target classes (fire risk levels 1-3) overlap	• Does not perform so well on sparse and unstructured data

Table 6.5 Summary of model performance

7. Modelling Results

Figure 7.1 shows the fire risk level in September 2020 and Figure 7.2 shows the prediction result of September 2021. We extracted the August 2021 data to predict for September 2021.

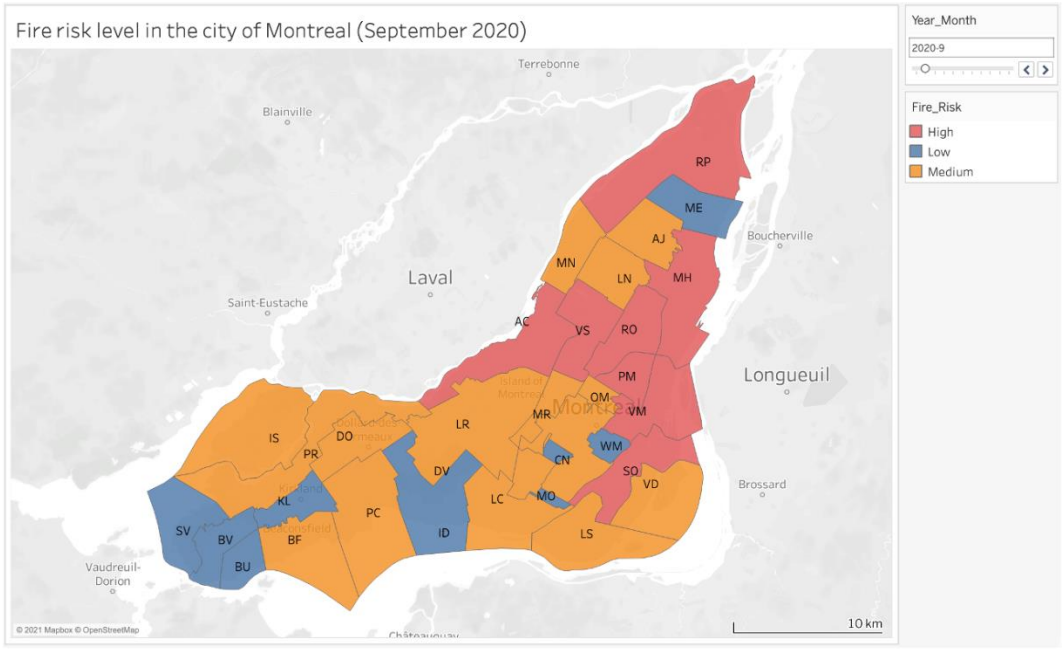

Figure 7.1 Fire risk level in Montreal (September 2020)

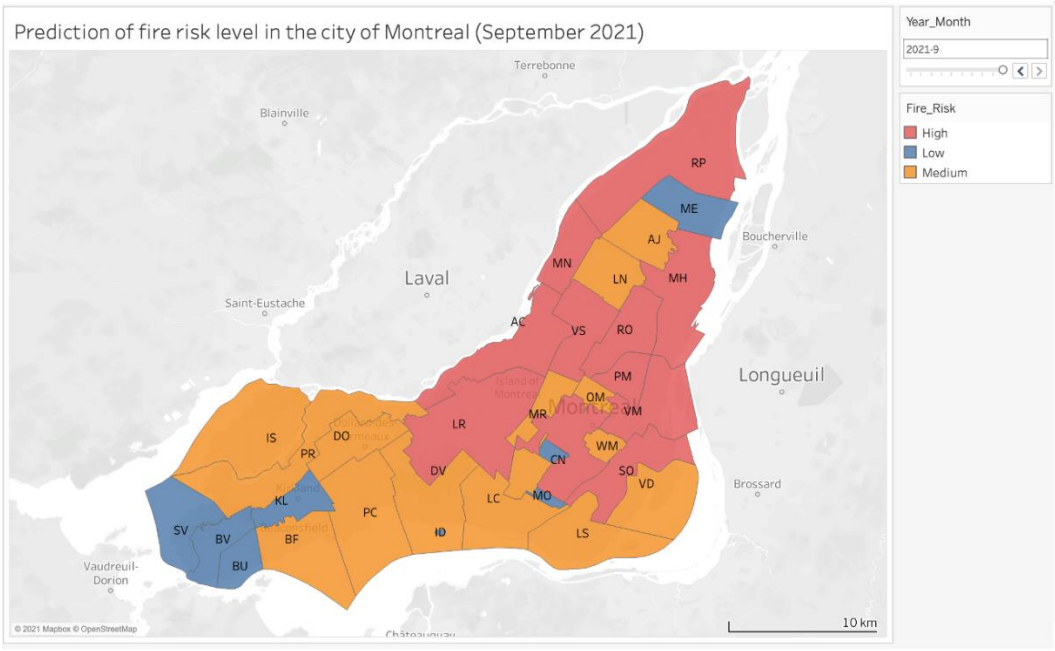

Figure 7.2 Prediction of fire risk level in the city of Montreal (September 2021)

8. Insights and Challenges

As shown in Figures 7.1 and 7.2 above, the high fire risk is more in the east, north and central area of Montreal, and areas in the west have low and medium fire risk level. The high fire risk area is more in downtown area where the population is denser. This month September 2021, compared to last year:

- areas such as Saint-Laurent, Montréal-Nord, Côte-des-Neiges-Notre-Dame-de-Grâce have changed from Medium risk to High risk
- areas such as Dorval and Westmount have changed from Low risk to Medium risk

The full list of fire risk level prediction can be found in Appendix 9.

One of the challenges is the time constraint given that we are a team of two. Below is the list of points which can be improved in the future:

- Explore more features from new datasets which may be more relevant to the fire risk, such as population of borough, education level, etc.
- Spend more time on feature engineering which helps select better feature to improve model performance.
- In the fire risk categorization, consider the size of areas/boroughs to compare the risk as the same number of fire incidents may have a different impact on boroughs with different sizes. For example, consider the dissemination blocks of 1 km^2.
- The severity of the incident may also be taken into consideration in the analysis. An area with low fire risk assigned by the model could have a high severity calculated in terms of significant loss of life and property damages.
- Consider other potential classification models to increase the chance of having a better prediction result.
- Consider the number of fire stations in each borough to help stakeholders better manage their resources.

9. Conclusions

The objective of our project is to predict the high fire risk level in Montreal and this report clearly meets this objective by explaining the problem, data source used, data exploration & cleaning, feature engineering, tools & techniques used, modelling techniques & results, insights & challenges.

Our project predicted the fire risk for September 2021 based on our understanding and limited knowledge. Since no model can have a perfect prediction, our stakeholders should not predict the fire risk and prioritize inspection process depending only on the model, this model should be used as a tool with the stakeholder's expertise and experience to make better decisions.

10. References

- Emmitsburg, Md., "Firefighter Injuries", U.S. Fire Administration, Volume 2, Issue 1, July 2001, Revised 2002
- Carnegie Mellon University., "Predictive Modeling of Building Fire Risk, Metro21: Smart Cities Initiative"
- Vancouver Fire Rescue Service and New Westminster Fire Rescue Service, "A Building Fire Risk Prediction Validation Project", April 2019
- Jason,B. (2016, August 16). Feature Importance and Feature Selection With XGBoost in Python. Machine Learning Mastery. https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/
- Emmanuel,A. (2018, Mar 6). Always start with a stupid model, no exceptions. How to efficiently build Machine Learning powered products. https://blog.insightdatascience.com/always-start-with-a-stupid-model-no-exceptions-3a22314b9aaa
- Steve,M. (2019, Apr 16). Introduction to Machine Learning Model Evaluation. HEARTBEAT. https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f
- Urvashi,J. (2018, Oct 7). Why Data Normalization is necessary for Machine Learning models. https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029
- Jonathan,J. (2017, Jan 26). How we predicted building fires in Baton Rouge, LA -- working version. https://scholar.harvard.edu/jonjay/blog/how-we-predicted-building-fires-baton-rouge-la-working-version
- Noah,C. (2019, May 20). Predicting Fire Risk for New York City Census Tracts. https://medium.com/@nchristiansen/predicting-fire-risk-for-new-york-city-census-tracts-ab5f96825c22
- Drazen,Z. (2019, Apr 15). Better Heatmaps and Correlation Matrix Plots in Python. Towards data science. https://towardsdatascience.com/better-heatmaps-and-correlation-matrix-plots-in-python-41445d0f2bec
- Présentation géographique des limites administratives des arrondissements et villes liées de l'agglomération de Montréal (n.d.) https://qlik.beta.montreal.ca/pub/sense/app/58f0430f-47b6-4038-a9e3-8a9e795834d8/sheet/b133656d-8f01-4a41-a237-fe42dc31b634/state/analysis?fbclid=IwAR00ZKD-C7PVIcXR4dUxOuw44eMU6CHz58dUaekdwsJGzv4LeleZcF0zKHE

11. Appendix

Appendix 1: Random Forest Model performance (All features)

```
[[217  50   2]
 [ 51 273  35]
 [  0  36 142]]

 Accuracy:  0.7841191066997518

 R Square Score: 0.5878420799940909

 Classification Report:
               precision    recall  f1-score   support

           1       0.81      0.81      0.81       269
           2       0.76      0.76      0.76       359
           3       0.79      0.80      0.80       178

    accuracy                           0.78       806
   macro avg       0.79      0.79      0.79       806
weighted avg       0.78      0.78      0.78       806
```

Appendix 2: Random Forest Model performance (Selected features)

```
[[219  48   2]
 [ 56 264  39]
 [  0  32 146]]

 Accuracy:  0.7803970223325062

 R Square Score: 0.5809727813273257

 Classification Report:
               precision    recall  f1-score   support

           1       0.80      0.81      0.81       269
           2       0.77      0.74      0.75       359
           3       0.78      0.82      0.80       178

    accuracy                           0.78       806
   macro avg       0.78      0.79      0.79       806
weighted avg       0.78      0.78      0.78       806
```

Appendix 3: Support Vector Machine Model performance (All features)

```
[[221  48   0]
 [ 94 221  44]
 [  0  59 119]]

 Accuracy:  0.6960297766749379

 R Square Score: 0.4390072755475125

 Classification Report:
               precision    recall  f1-score   support

           1       0.70      0.82      0.76       269
           2       0.67      0.62      0.64       359
           3       0.73      0.67      0.70       178

    accuracy                           0.70       806
   macro avg       0.70      0.70      0.70       806
weighted avg       0.70      0.70      0.69       806
```

Appendix 4: Support Vector Machine Model performance (Selected features)

```
[[243  26   0]
 [110 201  48]
 [  0  57 121]]

 Accuracy:  0.7009925558312655

 R Square Score: 0.4481663404365327

 Classification Report:
               precision    recall  f1-score   support

           1       0.69      0.90      0.78       269
           2       0.71      0.56      0.63       359
           3       0.72      0.68      0.70       178

    accuracy                           0.70       806
   macro avg       0.70      0.71      0.70       806
weighted avg       0.70      0.70      0.69       806
```

Appendix 5: K-Nearest Neighbors Model performance (All features without parameter tuning)

```
[[218  49   2]
 [ 65 271  23]
 [  0  41 137]]

 Accuracy:  0.7766749379652605

 R Square Score: 0.5741034826605605

 Classification Report:
               precision    recall  f1-score   support

           1       0.77      0.81      0.79       269
           2       0.75      0.75      0.75       359
           3       0.85      0.77      0.81       178

    accuracy                           0.78       806
   macro avg       0.79      0.78      0.78       806
weighted avg       0.78      0.78      0.78       806
```

Appendix 6: K-Nearest Neighbors Model performance (All features with parameter tuning)

```
[[209  58   2]
 [ 48 281  30]
 [  0  36 142]]

 Accuracy:  0.7841191066997518

 R Square Score: 0.5878420799940909

 Classification Report:
               precision    recall  f1-score   support

           1       0.81      0.78      0.79       269
           2       0.75      0.78      0.77       359
           3       0.82      0.80      0.81       178

    accuracy                           0.78       806
   macro avg       0.79      0.79      0.79       806
weighted avg       0.79      0.78      0.78       806
```

Appendix 7: XGBoost Model performance (All features without parameter tuning)

```
[[215  52   2]
 [ 53 275  31]
 [  0  29 149]]

 Accuracy:  0.792803970223325

 R Square Score: 0.6038704435498762

 Classification Report:
               precision    recall  f1-score   support

           1       0.80      0.80      0.80       269
           2       0.77      0.77      0.77       359
           3       0.82      0.84      0.83       178

    accuracy                           0.79       806
   macro avg       0.80      0.80      0.80       806
weighted avg       0.79      0.79      0.79       806
```

Appendix 8: XGBoost Model performance (Selected features with parameter tuning)

```
[[215  52   2]
 [ 47 285  27]
 [  0  29 149]]

 Accuracy:  0.8052109181141439

 R Square Score: 0.6267681057724267

 Classification Report:
               precision    recall  f1-score   support

           1       0.82      0.80      0.81       269
           2       0.78      0.79      0.79       359
           3       0.84      0.84      0.84       178

    accuracy                           0.81       806
   macro avg       0.81      0.81      0.81       806
weighted avg       0.81      0.81      0.81       806
```

Appendix 9: Prediction of fire risk level in the city of Montreal (September 2021)

	Boroughs	Year	Month	FIRE_RISK
0	Montréal-Est	2021	9	1
1	Westmount	2021	9	2
2	Montréal-Ouest	2021	9	1
3	Côte-Saint-Luc	2021	9	2
4	Hampstead	2021	9	1
5	Mont-Royal	2021	9	2
6	Dorval	2021	9	2
7	L'Île-Dorval	2021	9	1
8	Pointe-Claire	2021	9	2
9	Kirkland	2021	9	1
10	Beaconsfield	2021	9	2
11	Baie-d'Urfé	2021	9	1
12	Sainte-Anne-de-Bellevue	2021	9	1
13	Senneville	2021	9	1
14	Dollard-des-Ormeaux	2021	9	2
15	Outremont	2021	9	2
16	Anjou	2021	9	2
17	Verdun	2021	9	2
18	Saint-Léonard	2021	9	2
19	Saint-Laurent	2021	9	3
20	Montréal-Nord	2021	9	3
21	LaSalle	2021	9	2
22	Ville-Marie	2021	9	3
23	Le Sud-Ouest	2021	9	3
24	Le Plateau-Mont-Royal	2021	9	3
25	Mercier-Hochelaga-Maisonneuve	2021	9	3
26	Ahuntsic-Cartierville	2021	9	3
27	Rosemont-La Petite-Patrie	2021	9	3
28	Villeray-Saint-Michel-Parc-Extension	2021	9	3
29	Lachine	2021	9	2
30	Pierrefonds-Roxboro	2021	9	2
31	L'Île-Bizard-Sainte-Geneviève	2021	9	2
32	Rivière-des-Prairies-Pointe-aux-Trembles	2021	9	3
33	Côte-des-Neiges-Notre-Dame-de-Grâce	2021	9	3