# Data Science Capstone Project

## The Raiders & Co.

**McGill University**

**July 24th, 2019**

# Table of contents


| | |
|---|---|
| The team | 3 |
| Problem statement and scope | 4 |
| Our approach | 6 |
| Transforming the data | 7 |
| Feature engineering | 9 |
| Analysis and model comparison | 10 |
| Visualizing the analysis | 11 |
| Insights and improvements initiatives | 12 |

# The Team

This project was brought to completion by the following team of data enthousiasts.

**Maria Papadopoulos**

**François St-Amant**

**Stanley Tran**

**Ikram Mecheri**

All team members are part of the Machine Learning track of the course.

# Problem statement and scope

*The objective of this project is to predict Fire Risk and prioritize Fire Inspections using the City of Montreal available open data.*

## THE SITUATION

The City of Montreal has embarked on a transformational journey to revolutionize its Fire Department by embracing new digital technologies and approaches.

The City of Montreal wants to predict Fire Risk and prioritize Fire Inspection using available open data.

## THE CHALLENGE

During his presentation, Martin told us that you have tried to solve the puzzle in silos and it led to multiple challenges across People, Process, Data and Technology domains.

The City of Montreal faces significant challenges in building the right model to predict Fire Risk and manage  managing integrations to and from its current process. The Fire Risk prediction and prioritization is the nerve center for preventing fires and incidents in Montreal.

## THE SOLUTION

The City of Montreal is looking for a trusted partner who can develop an a model and roadmap including a solution options analysis and recommendation.

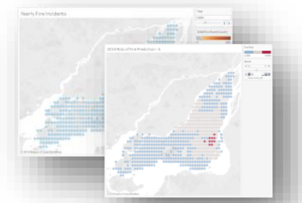
Our results are statistically significant and our state-of-the art AI models will ensure the accuracy of our results and help the Fire Department plan its fire inspection activities based on on

# History and prediction of fire risk in Montreal

**Show Tableau animation**

# Our approach

## Project Architecture

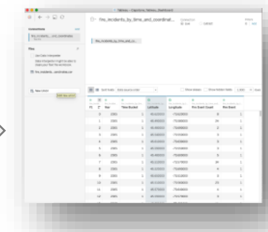
| Data source | Transformation | Raw Zone File | Modelling | Analysis |
|---|---|---|---|---|
| Dataset A - Interventions<br>Dataset B – Fire Stations<br>Dataset C – Property Assessment<br>Dataset D – Crime Intervention Data<br>Dataset E – CUBF Data | python<br>• Manual Exploration of data and variables selection<br>• Data Merging and Cleansing<br>• Features Engineering | Conformed CSV File with Merged Data | python<br>• **Model 1:** Neural Network<br>• **Model 2:** Random Forest<br>• **Model 3:** Gradient Boosting<br>• Hyperparameters fine tuning | tableau<br><br>python<br> |

# Transforming the data

We used a wide array of data cleansing techniques to merge, clean and conform the data. The following illustrates the different data processing methods we used for two of the data sources we used. The first method consisted of cleaning the data directly on Tableau Similar methods were used for the other data sources.

## Property Assessment Units Data Processing Using Tableau

- Group the CODE_UTILISATION into Property Use Code
- CATEGORIE_UEF variable into numeric
- Merge the NO_ARROND_ILE_CUM variable into the list of neighborhood data (Code REM variable)
- Add the names of the municipality in the Property Assessment Units data
- Notice that, once merged, there is a column named ID-uadm. We use those numbers to replace the municipality no. 50 (Montreal) so that we can have a unique Municipality ID
- We can now have a new column with the names of the municipality combined with Nom official variable. We clean this variable so that we can merge easily with the intervention data



**Once the data was cleansed, we extracted the merged data from Tableau directly into a CSV file**

Processing the data using Tableau enabled us to save a lot of time. The easy-friendly platform made the grouping and sorting of the data quicker to execute. Indeed, fewer steps were required from us when using Tableau in comparison to Python

## Data Transformation Using Python

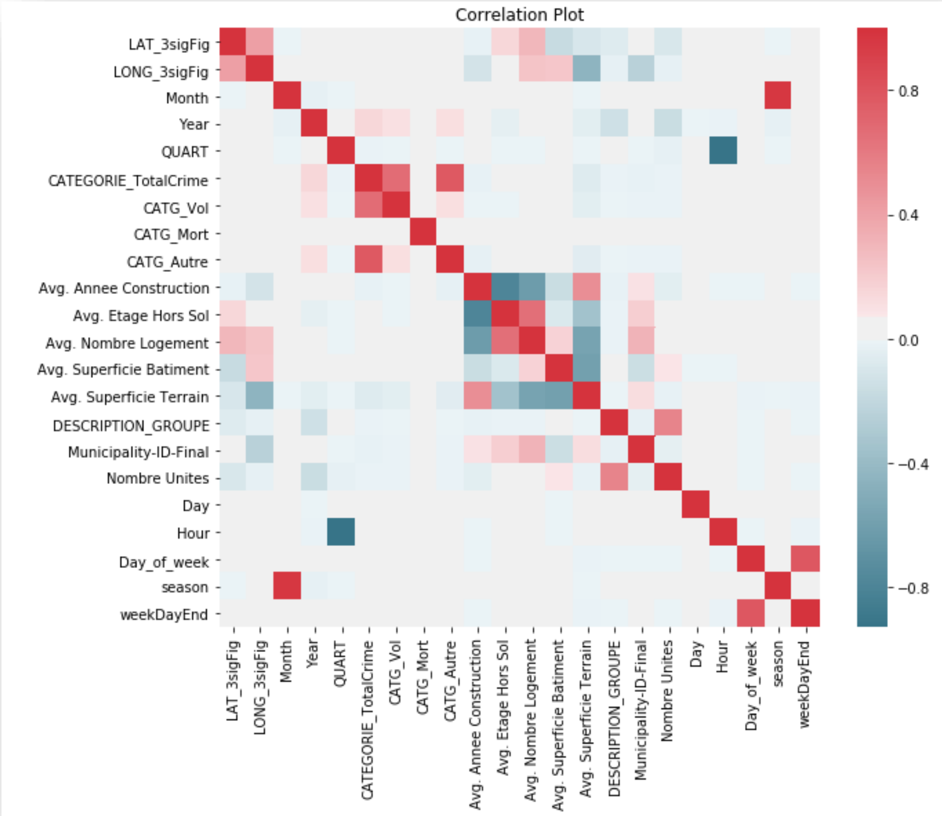
- Removed nulls and undertermined variables
- Drop the useless columns (e.g. ID_UEV, CIVIQUE_DEBUT, CIVIQUE_FIN, NOM_RUE, SUITE_DEBUT, etc.)
- Match the data using LAT and LONG
- Validate the distribution of the variables to predict
- Hot-encode variables and validate the conformity of variables
- Split the data into training and validation

CSV

**Once the data was cleansed, we extracted the merged data from Pythomn directly into a CSV file**

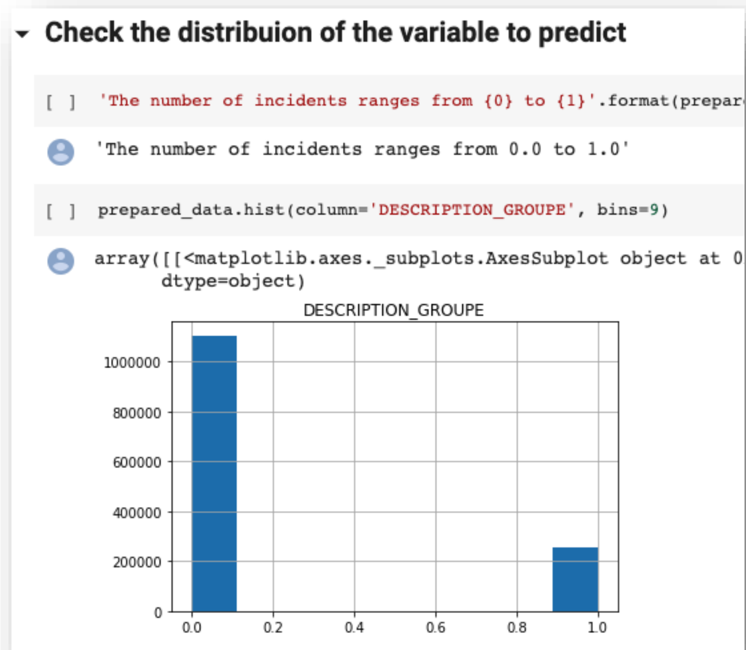
Processing the data on Python was a long process that involved multiple trial and errors. Compared to Tableau, Python offers more options and flexibility. However, given that every step had to be coded, this process consumed a lot of time when done via Python.

# Transforming the data




During the data preparation process, the team used Python to visualize correlation between the features. The correlation matrix on the right  shown here represents – on a scale of 0 to 1 – wether the features are intercorrelated, 1 being correlated.

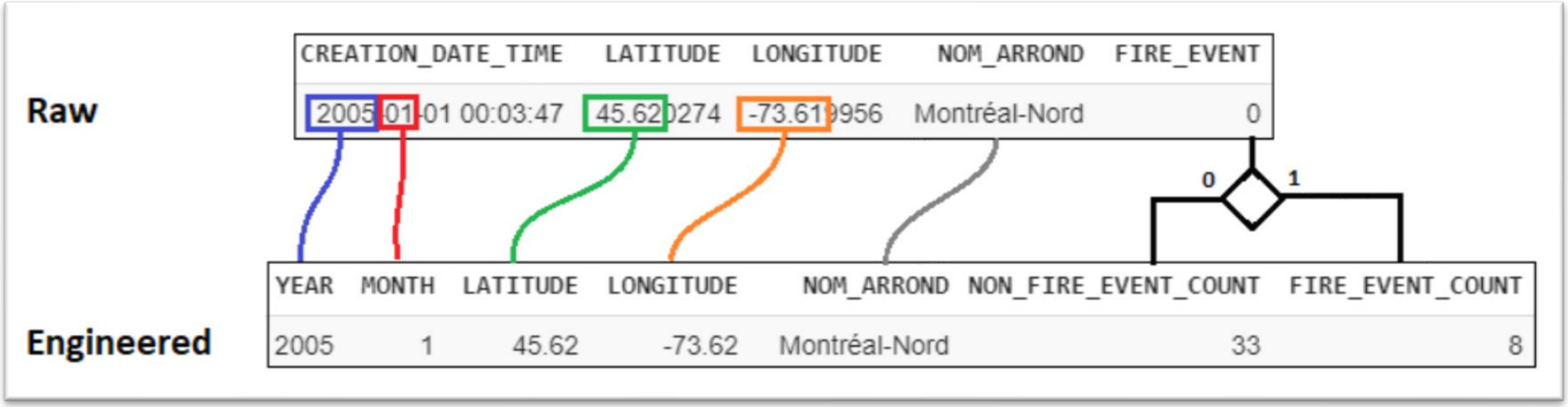
We used this plot, along with other quick visualisations during our data preparation process to analyze our features. This process proved to be extremely




In this case, we used the matplotlib library in Python to visualize the distribution of the variable to predict, which is the likelikhood of a fire, where 1 represents a high risk of fire. This histogram provides a quick and easy to execute validation.

# Features Engineering

After Nathaniel's presentation, we realized that our approach was flawed and not efficient. Our approach was mechanical and that we didn't really understood why our models were behaving the way they were. We didn't have a story line and that we couldn't make predictions beyond the data already available. As per Nathaniel's guidance, we took a step back and decided to get a more thorough and robust understanding of the data we were using. After further research, we decided to use features engineering instead of a simple mechanical data processing approach.



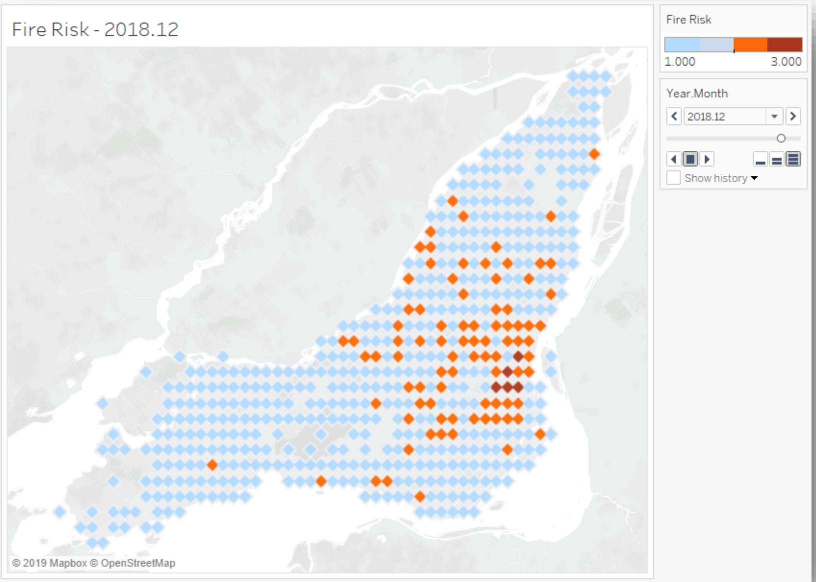

***We used Features Engineering to get a deeper understanding of the data and the hypothesis we were testing.***

# Analysis and model comparison

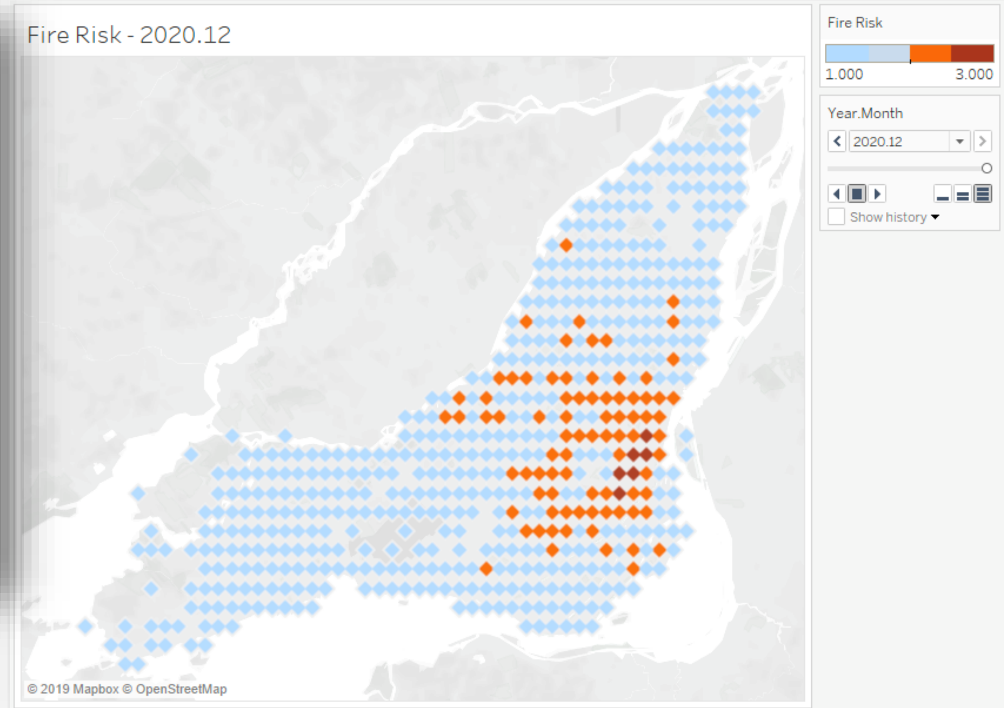
| Model | Neural Net: Adabound Optimizer | Random Forest | Gradient Boosting |
|---|---|---|---|
| **Accuracy** | • 86.4% without feature engineering<br>• 79.7% with feature engineering | • 86% without feature engineering<br>• 86% with feature engineering | • 75.97% without feature engineering<br>• 81.59% with feature engineering |
| **Advantages** | • Fast training speed<br>• Good with large amounts of data | • Can easily interpret data with different classes<br>• Decorrelates trees: important when dealing with multiple features that may be correlation | • Simple<br>• Good for small amounts of data<br>• Goof for unbalanced data sets |
| **Disadvantages** | • Neural nets poorly synthesize the ratio feature. We mitigated this issue by performing additional feature engineering<br>• Susceptible to overfitting | • Prone to overfitting as shown by our results | • Sensitive to overfitting with noisy data<br>• Longer training speed |
| **Description** | • Neural nets are computing systems that mimic the biological neural networks to solve prediction problems. The optimization algorithm is the main approach used in order to minimize errors.<br>• So far, the most common used are Adam and SGD. In recent papers, we discovered that there is a model that is as fast as Adam and as good as SGD, which is called Adabound.<br>• From our understanding, Adabound is an algorithm that employs dynamic bound on learning rate using Adam and gradually transitions into SGD | • Random Forests are deterministic data structures that use decision trees to model decisions rules for a specific classification problem. Each node represents a decision and the the final node gives us the probability for different classes.<br>• By fine tuning the hyperparameters (depth and estimators), we were able to increase the accuracy from 78% to 82%. | • Gradient boosting uses regression trees to predict a future state based on the historical data.<br>• Gradient boosting transforms weak learners into strong learners. This is particularly useful when the data is sparse and comes from disparate sources. |

# Visualizing the analysis – Tableau

**Historical Data**

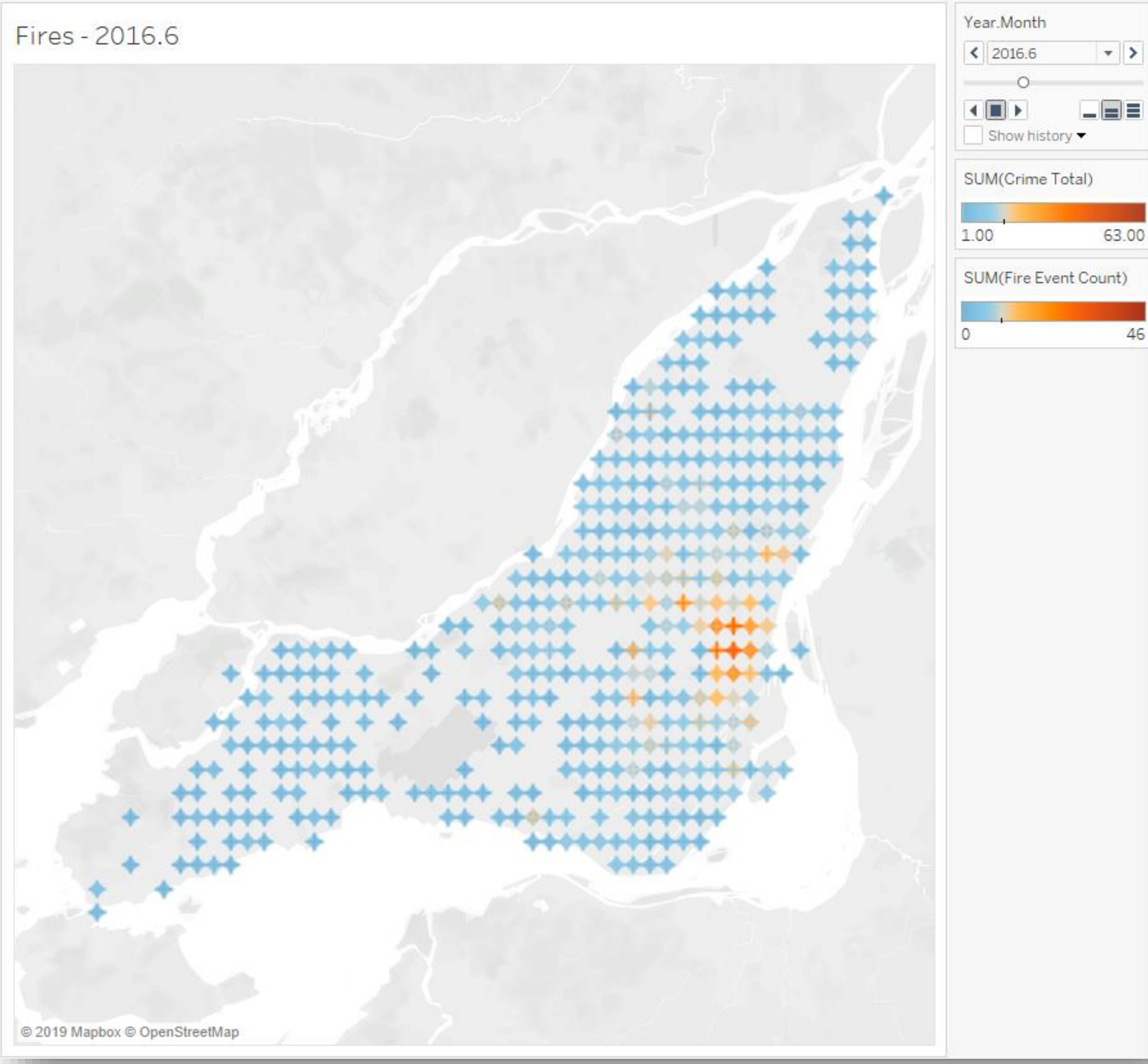



**Predicted Data**




- We also used Tableau to visualize our datasets and our predicted results.
- After fine-tuning the random forest classifier (85% instead of 76%) and ran the predictions for 2019 and 2020, we plotted our data into Tableau. Instead of having a big strip of medium risk dividing the island of Montréal in two, the medium risk are more sparse. That makes the transition from historical data to predicted data much more subtle.
- We can also observe that areas around educational institutions, such as Côte-des-Neiges or Berri-Uqam where a lot of student live (Cégep du Vieux-Montréal, UQAM) are also associated with a higher risk of fire.

# Insights and future improvements process initiatives




- We used Tableau throughout this journey to process and visualize our data. Using visualization tools also enabled us to quickly observe the future improvements that we can bring to our model.
- For this dashboard, we first merged the Fire dataset with the crime dataset. From then, we used Tableau to visualize what the combine risk score would be like.
- The visualization shows that the Crime and Fire data may be related. Further analysis will have to be conducted to investigate this relationship.
- In order to predict Fire risk using the Crime data, we would need to create a model that would also predict the risk of crime and then merge it with the Fire risk data to investigate this relationship.
- In addition, we could also use data from Environment and Climate Change Canada to study the relationship of the climate change with the fire occurrences.

# Conclusion

The Raiders & Co. Team succesfully completed the project within the timeline that was given.

**Did we answer the original question?**

- Yes, the objective of this project is to predict Fire Risk and prioritize Fire Inspections using the City of Montreal available open data which we were effectively able to achieve. We developed various models and kept the three most significant models for our analysis and final predictions. We believe that the various methods used vary in complexity and granularity.

**Were our external data pertinent?**

- Our external data proved to be significantly pertinent as it allowed us to go deeper in our analysis and derive more accurate conclusions.

**Do we understand why we used different models?**

- As data enthusiasts, we understand that different models can lead to different conclusions and accuracy results. In order to minimize bias, false positives and improve our accuracy, we opted for a comparative approach where we selected different models with different level of complexity to analyze our data. Our analysis on these models can be found in the Analysis section.
- We also invested great efforts in features engineering during our data preparation process to improve our predictions. We wanted to have features that could be known with high certainty, that we could estimate with enough precision to feed our prediction model. This process was highly beneficial as it allowed us to improve our overall model prediction accuracy. It also enabled us to better understand the results that the models were giving us, in other words, it gave us the tools to explain the why of our results.

**What kind of visualizations are we using?**

- We used Tableau as well as Python to visualize our data and our results.