Prediction of Extreme Weather Events Using Statistical Learning Methods

by

Zixuan Deng

Supervisor: Xu (Sunny) Wang
Second Reader: David Soave

B.A., Wilfrid Laurier University, 2020

WILFRID LAURIER UNIVERSITY
Winter 2020

Abstract

Weather affects our daily life. It influences how people feel and look at the world. It is hard for people to well-prepare for every day's weather. A severe weather event has more severe consequences than typical weather. It could happen anytime and in any part of the country. As a consequence of global warming, extreme weather events happen more frequently in the last twenty years. Each severe weather event could be a disaster for humans, thus predicting this kind of event becomes meaningful. It could help the government to prevent and control the loss of infrastructure, and help people protect themselves. Many methods including Deep learning methods have been applied to detect patterns of extreme weather events. In this report, some statistical learning methods such as Logistic Regression, Decision Tree will be applied for an extreme weather event forecast. Weather data from Toronto is used in this project.

Keywords: Weather; Extreme weather; Decision Tree; Random forest; Logistic Regression; LDA; QDA; SVM; KNN; Supervised learning

Acknowledgements

I would like to express my special gratitude to my supervisor Dr. Xu Wang for the continuous support of my study and research. She is pretty patient, enthusiasm and knowledgeable. She always provides me with helpful suggestions and leads me to the correct direction. Without her guidance, I cannot continue my project.

Also, Dr. Xu Wang hosted a statistical learning group discussion every Friday, and it gives me an excellent chance to share my ideas and ask questions related to my research. I would also like to thank all the participants in the group discussion. You help me improve my presentation skills and increase my knowledge. It is always good to hear people around you who are going in the same way as you go.

The course "Statistical Learning" in Wilfrid Laurier University provides me with an outstanding experience of statistical learning. Lab Coordinator Sukhjit Singh Sehra helps me a lot with R programming.

Table of Contents

Abstract ii

Acknowledgements iii

Table of Contents iv

List of Figures vi

1 Introduction 1

1.1 Literature Review . . . 2

2 Methodologies 3

2.1 Logistic Regression (LR) . . . 4

2.2 LDA and QDA . . . 5

2.3 K Nearest Neighbours . . . 5

2.4 Support Vector Machine . . . 6

2.5 Decision Trees . . . 8

2.6 Random Forest . . . 8

3 Data 10

3.1 Description . . . 10

3.1.1 Unit of Variables . . . 11

3.2 Data Cleaning . . . 12

3.2.1 Data Processing . . . 12

3.2.2 Missing value problem . . . 13

3.2.3 Imbalance Class . . . 16

3.3 Feature Selection . . . 19

4 Results and Future Research 22

4.1 Results . . . 22

4.2 Future Research 23

Bibliography **25**

Appendix A Code **28**

List of Figures

Figure 2.1 An example of LDA and QDA [15] 6
Figure 2.2 An example of KNN [15] 7
Figure 2.3 An example of support vector machine [8] 7
Figure 2.4 An example of decision tree 9
Figure 2.5 Influence of number of tree [15] 9

Figure 3.1 Weather Description 11
Figure 3.2 Steps in applying multiple imputation to missing data via the "Multivariate Imputation via Chained Equations" approach [16] 14
Figure 3.3 A summary of missing values for Toronto Weather data 15
Figure 3.4 Toronto NA distribution 16
Figure 3.5 Confusion Matrix 17
Figure 3.6 Random Over-Sampling [7] 18
Figure 3.7 Split Dataset [11] 19
Figure 3.8 Summary of best subset models 20
Figure 3.9 Feature Selection 21

Chapter 1

Introduction

Weather affects our daily life. Weather changes quickly. Sometimes it is cloudy then it may rain in 15 minutes. It is hard for people to well-prepare every day's weather, but the day-to-day changes in weather can influence how people look at the world. A severe weather event has more severe consequences than typical weather. It could happen anytime, in any part of the country. Severe weather refers to any dangerous meteorological phenomena with the potential to cause damage, severe social disruption, or loss of human life [2]. Sorts of extreme weather differ, contingent upon the scope, elevation, geology, and environmental conditions. High breezes, hail, unnecessary precipitation, and rapidly spreading fires are types of extreme weather. The radical weather is brought about by rainstorms, downbursts, lightning, tornadoes, waterspouts, violent tropical winds, and extratropical typhoons. Territorial extreme climate marvels incorporate snowstorms, blizzards, ice tempests, and residue storms [2]. Overwhelming precipitation and flooding, warmth and dry spell can likewise make extreme weather events occur. These extraordinary climate occasions can cause a cataclysmic event. In June 2013, Alberta, Canada, experienced substantial precipitation that activated flooding. Unfortunately, five lives were lost, and as much as 6 billion dollars in budgetary misfortunes and property harm were continued crosswise over southern Alberta [24]. Each severe weather event could be a disaster for humans, and it threatens personal safety and brings huge losses to public utilities. Every year, the governments spend billions on repairing the damage caused by severe weather events. Thus predicting these kinds of events become meaningful. Toronto, though it is not the capital of Canada, it is one of the most important cities in Canada. The goal of this project is to predict the severe weather events that may happen in the city of Toronto.

1.1 Literature Review

Climate scientists have used advanced methods including deep learning techniques to analyze climate data. Due to our limited knowledge, there is no much research that has done on predicting severe weather events using statistical learning methods. Prabhat et. al.(2015) have developed TECA (Toolkit for Extreme Climate Analysis). The TECA strategy has been used to assess the presence of different computational models in repeating the insights of extraordinary climate occasions. It also recognizes storms in high-recurrence atmosphere model yield. Prabhat et al(2015) have applied systems to recognize three various classes of tempests: violent tropical winds, climatic streams and extra-tropical violent winds [21].

In the field of Machine Learning, deep learning has been largely applied. Several works proposed fully supervised convolutional neural networks (CNNs) to detect and classify well-know types of extreme climate events with high precision. Yunjie (2016) developed deep Convolutional Neural Network (CNN) classification system and demonstrated the usefulness of Deep Learning technique for tackling climate pattern detection problems. Coupled with Bayesian based hyper-parameter optimization scheme, his deep CNN system achieves 89 % to 99 % of accuracy in detecting extreme events (Tropical Cyclones, Atmospheric Rivers and Weather Fronts) [17].

Ziqi (2016) proposed a new extreme weather recognition method - CNN based on images by using computer vision manners. The experimental results show that the proposed method is able to achieve a high performance with the recognition accuracy rate of 94.5 % and can meet the requirements of some real applications [27]. Evan Racah presented a multichannel spatiotemporal CNN architecture for semi-supervised bounding box prediction and exploratory data analysis. He and his team demonstrated that their approach was able to leverage temporal information and unlabeled data to improve the localization of extreme weather events [23].

Xingjian(2015) explored a convolutional LSTM (Long short-term memory) architecture for future prediction on a local scale using radar echo data [25]. However, those applied deep learning techniques used different detest than this project that it uses normal weather data.

Chapter 2 introduces the methods used in the data analysis. Many classification methods have applied in this project.

Chapter 2

Methodologies

In this chapter, it gives general background of classification methods used in this project. Supervised learning and unsupervised learning are two popular groups of statistical learning methods.

- Supervised Learning: A supervised learning algorithm takes a known set of input dataset and its known responses to the data (output) to learn the regression/-classification model. A learning algorithm then trains a model to generate a prediction for the response to new data or the test dataset[9].
- Unsupervised Learning: An unsupervised learning algorithm is the opposite of supervised learning algorithm. It uses unknown responses to the data to learn clustering/neural networks.

We select statistical learning methods based on whether the response variable is numerical or categorical. If the response variable is numerical, then it's a regression problem. If the response variable is categorical, then it's a classification problem. Regression predictive modeling is the task of approximating a mapping function (f) from input variables (X) to a continuous output variable (y). A continuous output variable is a real-value, such as an integer or floating point value. These are often quantities, such as amounts and sizes. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modelling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). Classification belongs to the category of supervised learning where the targets also are provided with the input data. There are many applications in classification in many domains, such as in credit approval, medical diagnosis, target marketing etc [6]. Weather can be severe (Yes) or normal (No). Predicting severe weather is a classification problem as the output is

yes or no. Here are some classification algorithms in statistical learning. The detailed description of each method will be introduced in the following sections.

- Logistic Regression
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- K Nearest Neighbours
- Support Vector Machine
- Decision Tree
- Random Forest

2.1 Logistic Regression (LR)

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (e.g., yes, success, etc.) or 0 (e.g., no, failure, etc.).

From a mathematical aspect, we consider a model with m variables $x_1, x_2, \cdots, x_m$ and write

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m = \beta_0 + \sum_{i=1}^{m} \beta_i x_i. \tag{2.1}$$

Then the formula for logistic regression can be written as in Equation (2.2).

$$p(X) = \frac{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m)}} = P(Y = 1|X), \tag{2.2}$$

where the probability of X lies within 0 and 1 range. If the threshold is set to 0.5, then every predicted Y above or equal to 0.5 belongs to the first category. The threshold will change based on different problems. If taking a log transformation to Equation (2.2), then we can get Equation (2.3). The right hand side of Equation (2.3) is exactly the same as multivariable regression function.

$$\log \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m \tag{2.3}$$

2.2 LDA and QDA

Assume data $\boldsymbol{X}$ has p features, k classes and, and observations of classes have means $\boldsymbol{\mu_0}, \boldsymbol{\mu_1}, \cdots, \boldsymbol{\mu_k}$, and covariances $\Sigma_i, i = 1, \cdots, k.$. For LDA, we assume Σ_i's are the same. Let $\Sigma_i = \Sigma$. The conditional probability function is:

$$P(\boldsymbol{x}|y = k) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_k})^T\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu_k})) \tag{2.4}$$

Linear disciminant function is:

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu_k} - \frac{1}{2}\boldsymbol{\mu_k}^T\Sigma^{-1}\boldsymbol{\mu_k} + \log\pi_k, \tag{2.5}$$

where π_k is the prior probability that a randomly chosen observation comes from the k^{th} class. Then, the predicted output for LDA is:

$$f(\boldsymbol{x}) = \arg\max_k \delta_k(\boldsymbol{x}) \tag{2.6}$$

Like LDA, the QDA classifier results from assuming that the observations from each class are drawn from a Gaussian distribution. However, QDA assumes that an observation from the k^{th} class is of the form $X \sim N(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$[15].

Quadratic linear discriminant function is:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_k})^T\Sigma_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu_k}) + \log\pi_k \tag{2.7}$$

And the output is:

$$f(x) = \arg\max_k \delta_k(x) \tag{2.8}$$

LDA and QDA have different shapes of decision boundary. Figure 2.1 shows an example of LDA and QDA. The solid line is the decision boundary for QDA, and the dash line is the decision boundary for LDA.

2.3 K Nearest Neighbours

KNN classifies new case based on the classes of its k neighbours in the explanatory variable space. It measures the distance between the new case and its neighbours. When measuring distance, Manhattan Distance and Euclidean Distance are commonly used.

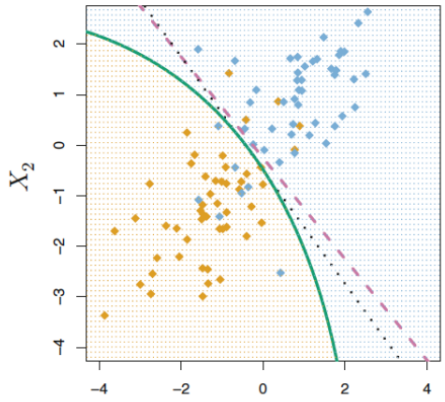

Figure 2.1: An example of LDA and QDA [15]

$$d_(\mathbf{X}, \mathbf{X}') = \|\mathbf{X} - \mathbf{X}'\| = \sum_{i=1}^{m} |x_i - x'_i| \tag{2.9}$$

$$\begin{aligned} d(\mathbf{X}, \mathbf{X}') = d(\mathbf{X}', \mathbf{X}) &= \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \cdots + (x_m - x'_m)^2} \\ &= \sqrt{\sum_{i=1}^{m} (x_i - x'_i)^2}. \end{aligned} \tag{2.10}$$

Equations (2.9) and (2.10) are Manhattan Distance and Euclidean Distance respectively. Figure 2.2 illustrates the situation of 3 nearest neighbours for a test point denotes as **"x"**. When k is small, the algorithm is more flexible, which and has high variance, low bias. In the KNN model, k is the number of neighbours is the hyperparameter in the model and should selected by Cross-Validation.

2.4 Support Vector Machine

An SVM training algorithm builds a model that assigns new observation to one category or the other. The algorithm outputs an optimal hyperplane, which categorizes a new example. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence [8]. Figure 2.3 shows an example of support vector machine. The optimal hyperplane can be a plane or a line based on the number of input features. For example, if the number of input features is 2, then the hyperplane is a line. The algorithm for support vector machine is shown in Equation (2.11). It is an optimization problem. C represents the regularization

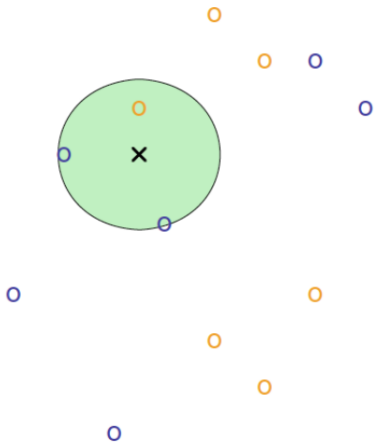

Figure 2.2: An example of KNN [15]

parameter. If C is too large or small, it will cause bias. ϵ is the slack variable, and M is the margin.

$$
\begin{aligned}
\underset{\beta_0,\beta_1,\beta_2,\cdots,\beta_p,\epsilon_1,\cdots,\epsilon n}{\text{maximize}} \quad & M \\
\text{subject to} \quad & \textstyle\sum_{j=1}^{p} {\beta_j}^2 = 1, \\
& y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} \geq M(1 - \epsilon_i) \\
& \epsilon_i \geq 0, \textstyle\sum_{i=1}^{n} \epsilon_i \leq C, \\
& \text{for all } i = 1, \ldots, N.
\end{aligned}
\tag{2.11}
$$

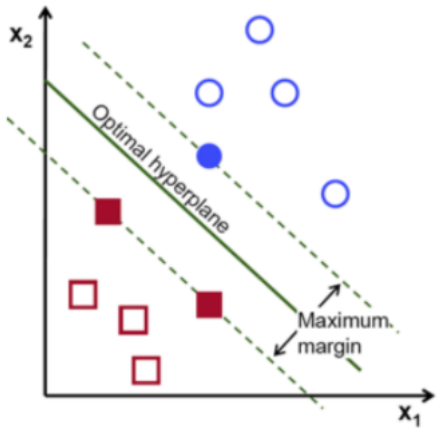

Figure 2.3: An example of support vector machine [8]

2.5 Decision Trees

A decision tree can be seen as a graph representation of a decision algorithm where each node holds a statement that is either true or false. From each node there are two directed edges where one corresponds to true, and one to false. Depending on the value of the statement, the decision algorithm continues along the corresponding edge [18]. An example of this process is shown in Figure 2.4. This tree is the optimal tree model using weather data. When **"Year"** is greater than 0.7, the tree goes to a terminal node. If **"Year"** is less than 0.7 and **"Wind speed"** is greater than 0.24, another terminal node arrives.

In Decision Tree the major challenge is to identification of the attribute for the node in each level. This process is known as attribute selection. There are three popular attribute selection measures.

Suppose a random variable takes C possible values (classes) each with prob p_k $k = 1, \cdots, C$. the entropy for the t^{th} terminal node of this random variable is defined as Equation (2.12).

$$E = -\sum_{k=1}^{C} p_{tk} \log(p_{tk}) \tag{2.12}$$

And information gain is defined as Equation (2.13).

$$IG(T, a) = \mathrm{E}(T) - \mathrm{E}(T|a),, \tag{2.13}$$

where $E(T|a)$ is the conditional entropy of T given the value of attribution a. The last criterion is Gini index and it is popular in machine learning.

$$Gini = 1 - \sum_{i=1}^{C} {p_{tk}}^2 \tag{2.14}$$

2.6 Random Forest

Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees. As in bagging, we build a number of decision trees on bootstrapped training samples. However, when building these decision trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors [15]. A fresh sample of m predictors is taken at each split, and

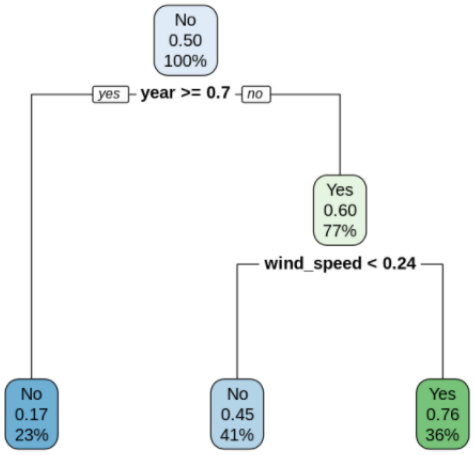

Figure 2.4: An example of decision tree

typically we choose m equals $\sqrt{p}$. That is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors.

Figure 2.5 illustrates the influence of m value.

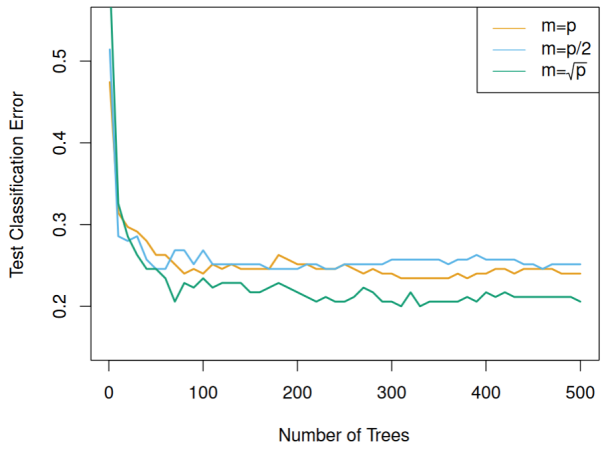

Figure 2.5: Influence of number of tree [15]

This chapter has introduced the background of statistical learning. Regression and classification are two directions of supervised learning. Logistic regression, LDA, QDA, KNN, SVM, decision tree, and random forest have been discussed. Next chapter contains the description and challenges of data.

Chapter 3

Data

When we consider weather, variables such as humidity, air pressure, temperature, wind direction, wind speed, location and population density really affect it. The data set "Historical Hourly Weather Data 2012-2017" created by Selfish Gene in Kaggle.com [10] contains the information of these variables.

3.1 Description

There are eight variables in the dataset, which are Humidity, pressure, temperature, weather type, wind direction, wind speed, DateTime and city. There are 36 cities in this dataset, such as New York City, Los Angeles, etc. Each city does have 45,253 records over 6 years. In this project, city of Toronto data has been used.

Toronto is the provincial capital of Ontario and the most popular city in Canada. It is classified as warm-summer humid continental climate [4]. Its summers are warm to hot, and its winter brings very cold, snowy, windy, and icy weather. The city experiences four distinct seasons, with considerable variance in length. The summer months are characterized by very warm temperature. Spring and autumn are transitional seasons with generally mild or cool temperature with alternating dry and wet periods. Winters are cold with frequent snow.

Explanatory variables

Humidity, air pressure, temperature, wind direction and wind speed can greatly influence weather. They are explanatory variables in this dataset. Humidity tells us the moisture content of the atmosphere, or how much water vapor there is in the air. When the humidity is high it feels oppressive outside because sweat doesn't evaporate and provides cooling [3]. Air pressure is very important as well. When air pressure

is high, it will be cool, dry air and has good weather. When air pressure is low, it is warm, moist air and will be bad weather. For example, when it rains, it is usually low air pressure, and the weather is not good as a sunny day [5]. Water circulatory system is one example of how these factors could influence the weather. Warm temperature and high wind increase sea evaporation while pollution, cloud cover and humidity reduce evaporation.

Response Variable

They are 43 levels in variable weather. Figure 3.1 shows all levels of this response varaible. The highlighted events of weather are the extreme weather events, and our goal is to do a forecast of the extreme weather events. Predicting the weather level is a supervised learning problem as the dataset already has the classes of weather. Each hour's data has its humidity, air pressure, wind speed, wind direction and temperature, and its weather description. To simplify the classification problem, we divide the levels of weather into two groups: Severe (coded as Yes) and non-severe (coded as No).

Broken Clouds	Drizzle	Dust	Few Clouds	Fog	Freezing Rain	Haze	Heavy intensity rain	Heavy intensity snow
Heavy shower snow	Heavy Snow	Light intensity drizzle	Light intensity drizzle rain	Light intensity shower rain	Light rain	Light rain and snow	Light shower sleet	Light shower now
Mist	Moderate rain	Overcast clouds	Proximity shower rain	Proximity thunderstorm	Ragged thunderstorm	Rain and Snow	Sand	Scattered clouds
Shower drizzle	Shower rain	Shower snow	Sky is clear	Sleet	Smoke	Squalls	Thunderst o-rm	Thunderst o-rm with heavy rain
Thunderst orm with light rain	Thunderst orm with rain	Very heavy rain	Volcanic ash					

Figure 3.1: Weather Description

3.1.1 Unit of Variables

Unites of these variables are a little bit different than most weather forecast uses. The relative humidity is the unit of humidity.

The relative humidity is the ratio of the partial pressure of water vapour to the equilibrium vapour pressure of water at a given temperature. It is between 0 % to 100

%. Air pressure is measured in hPa. The unit of temperature is different than what we use. In this dataset, it uses K (Kalvin) to measure the temperature. Kalvins uses absolute zero (-273.15 Celsius) as 0 K [19].

Consequently, a wind blowing from the north has a wind direction of 0° (360°); a wind blowing from the east has a wind direction of 90°; a wind blowing from the south has a wind direction of 180°; and a wind blowing from the west has a wind direction of 270°. In general, wind direction is measured in units from 0° to 360°. Wind speed is the Beaufort wind force scale, usually from 0 to 12. Wind rating is based on the extent to which wind affects ground objects or the surface of the sea [1]. Table 3.1 is the wind scale and its description.

Level	Description	Level	Description
0	Calm	7	Near Gale
1	Light Air	8	Gale
2	Light Breeze	9	Severe Gale
3	Gentle Breeze	10	Storm
4	Monderate Breeze	11	Violent Storm
5	Fresh Breeze	12	Hurricane
6	Strong Breeze		

Table 3.1: Beaufort Wind Force Scale [1]

3.2 Data Cleaning

Data cleaning is an essential part when we are dealing with data in datamining. Further, it is the most time-consuming part. In this stage, we need to think carefully about the variables in the dataset to see if this variable is beneficial to solve our question. Also, we need to look at the missing values and class balance of the data.

3.2.1 Data Processing

When we look at the dataset, we find some challenges. There are six CSV files. Since it is not convenient to look at six files simultaneously, we need to combine the dataset. Also, some missing data are combined in the dataset. We need to wrangle the data first before analyzing it, and we follows the below steps:

- Import packages in R that requires for cleaning dataset.
- Extract city of Toronto weather data from the original CSV files.

- Solve challenges in dataset such as missing value, imbalance class and scale data.
- Extract important information from the dataset.
- Split dataset into train and test datasets.

In this project, R packages **"caret"**, **"e1071"**, **"dplyr"**, **"rsample"** and **"imbanlance"** are used. In original data files, there is one common factor in all files, which is "city". By using R, we could combine all files and get one big dataset. This dataset contains all cities in America. However, we only need City of Toronto data. After extracting the information for city of Toronto, a new dataset is created that includes eight variables (humidity, pressure, temperature, weather, wind direction, wind speed). For convinence, datetime has been splited into four variables that are month, day of month, day of year, week of day and hour.

3.2.2 Missing value problem

A missing data (or missing values) is defined as a value that is not stored for a variable in the observation of interest. The problem of missing data is relatively common in almost all research and can have a significant effect on the conclusions that can be drawn from the data [14].

There are many different ways that we can deal with missing values. By using **"is.na()"** in R, we can identify the missing values. The simplest way is deleting missing values,or impute mean or median value. Using **"omit.na()"** or **"na.rm = TRUE"**, we can delete missing data from the dataset. Maximum likelihood estimation and multiple imputations are popular for solving missing value problems.

Multiple imputations (MI) provides an approach to missing values that are based on repeated simulations. MI is the common method of choice for complex missing value problems. In MI, a set of complete datasets (typically 3 to 10) are generated from an existing dataset containing missing values. Monte Carlo methods are used to fill in the missing data in each of the simulated datasets. Standard statistical methods are applied to each of the simulated datasets, and the outcomes are combined to provide estimated results and confidence intervals that take into account the uncertainty introduced by the missing values [16]. Figure 3.2 shows the complete process of multiple imputation method.

The maximum likelihood estimation is used to analyze the full, incomplete data set. This method does not impute any data, but instead uses each available data

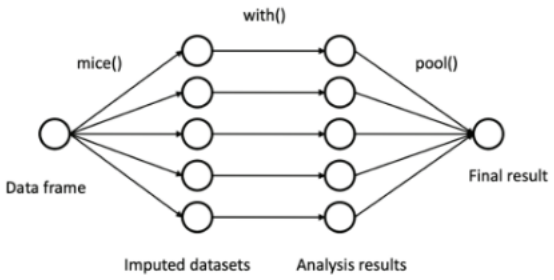

Figure 3.2: Steps in applying multiple imputation to missing data via the "Multivariate Imputation via Chained Equations" approach [16]

to compute maximum likelihood estimates. The maximum likelihood estimate of a parameter is the value of the parameter that is most likely to have resulted in the observed data [12].

Another important issue that needs to get our attention is that we need to see whether the data point is MAR - Missing At Random or MCAR — missing Completely at Random. The propensity for a data point to be missing is completely random. There is no relationship between whether a data point is missing and any values in the data set, missing or observed [13]. MAR is missing at Random A better name would be Missing Conditionally at Random because the missingness is conditional on other values [13].

Missing value analysis

To determine which method will be used to solve incomplete data problem, the first thing needs to do is to analyze these missing data, which we could see if there is any trend or pattern in the missing data. Since the data is time-series data, missing values cannot be directly deleted as time-series data must be continuous. If any data in the middle is deleted, it will cause biased analysis. Figure 3.3 shows percentages of missing values in each variable.

In total, the missing data of Toronto deteset is approximately 10 %. We need to see how missing values are distributed in the whole dataset so that the pattern of missing values could be determined. The missing values' distribution in the whole timeline is show in Figure 3.4. The blue point is the value of each variable in the unit of time. The data is measured in hours, and is from November, 2012 to November, 2017. Thus there are over 40,000 units of time. The black lines are the lines that connect two adjacent blue points. Every time when missing values are detected in the dataset,

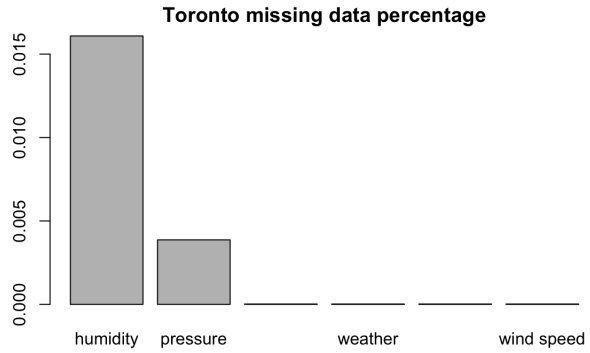

Figure 3.3: A summary of missing values for Toronto Weather data

there is one vertical red line as shown in the graph. By looking at the position of red lines, we can see if the missing data is missing seasonally, or if there is any trend. Variables wind speed, wind direction and temperature do not have too many missing values, and they all appear in the beginning of the time period. However, variables pressure and humidity don't follow similar patterns and have a lot of missing data in the beginning and at the end. It may be because the data is lost or cannot be collected as limited technology. Variable humidity has many missing data, and does not follow any pattern since it almost has missing values every day. The other two sub-datasets have fewer missing values, variables temperature, wind direction, and wind speed are almost complete. However, there are sill missing values in variable humidity and pressure. Most of them are missing at the beginning of the timeline, and some of them are in the middle.

Missing values in Weather

In terms of variable weather, there is one record missing at 2012-10-1 12.00 in Toronto data. As the missing data appear either at the beginning of or the end of the dataset, we can delete them without affecting other data.

Multiple Imputation

There is an interesting phenomenon that if there are missing values in "humidity", there mostly like have missing values in "pressure". Furthermore, for wind direction and wind speed, they have a similar relationship as that between pressure and humidity. Then we say, these variables are missing at random, i.e, the missingness is conditional on another variable.

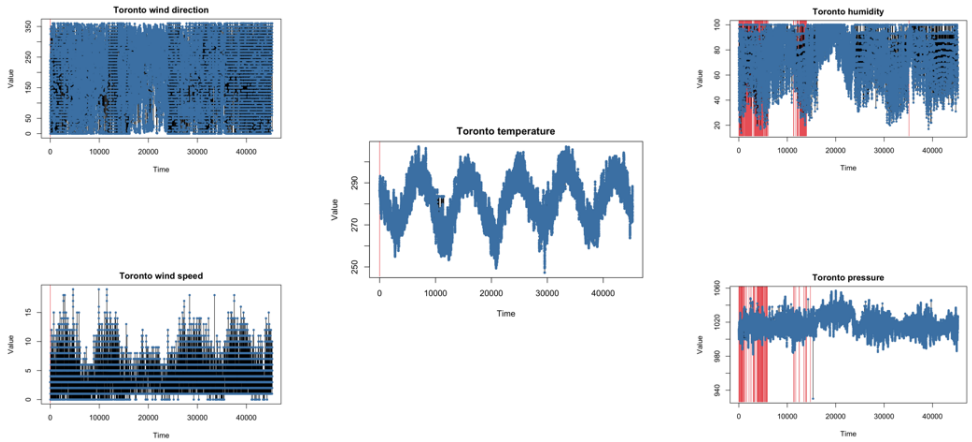

Figure 3.4: Toronto NA distribution

Multiple imputation methods can apply to solve the missing value problem. **"Mice"** is the package in R in which multiple imputations can be processed. The function "mice()" starts with a data frame containing missing data and returns an object containing several complete datasets (the default is 5). Each complete dataset is created by imputing values for the missing data in the original data frame. There is a random component to the imputations, so each complete dataset is slightly different. The "with()" function is then used to apply a statistical model (for example, linear or generalized linear model) to each complete dataset in turn. Finally, the "pool()" function combines the results of these separate analyses into a single set of results. The standard errors and p-values in this final model correctly reflect the uncertainty produced by both the missing values and the multiple imputations.

3.2.3 Imbalance Class

Imbalanced class is another challenge for this project. Severe weather events happen rarely and there is only 1068 cases happened and counting 2.4% to the total observations. This causes a challenge. Assume all observations are labeled as non-severe weather, then the prediction accuracy will still have over 97%. However this makes no sense as we want to predict the rare class, i.e the severe weather.

Handle Imbalanced Class

There are two ways to handle an imbalanced class problem. First one is changing the criterion that measures model performance. Figure 3.5 displays a confusion matrix.

Originally, we use accuracy to measure the performance of model. However for imbalanced class dataset, accuracy cannot be used along as measure criterion. The reason is that the goal of project is to predict the severe weather events. Therefore, precision, recall and F - measure are introduced as measure criteria. Formulas 3.1 and 3.2 are the definition of precision and recall.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.2}$$

Recall means out of all the positive classes, how many we predict correctly. It should be as high as possible. Precision means out of all the positive classes we have predicted correctly, how many are actually positive. Because it is difficult to compare two models with low precision and high recall or vice versa. Therefore, to make them comparable, models will use F-Measure as it helps to measure recall and precision at the same time. Therefore, the model with the highest F-Measure is the best model for predicting severe weather events in this project. And the formula for F-Measure is in formula 3.3.

$$\text{F} - \text{Measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3.3}$$

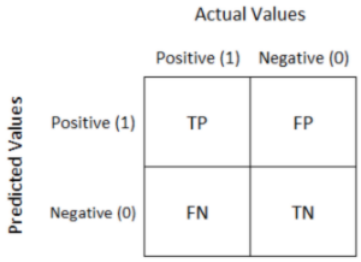

Figure 3.5: Confusion Matrix

The other way is to use resampling techniques to handle imbalanced class data. The main objective of balancing classes is to either increasing the frequency of the minority class or decreasing the frequency of the majority class. This is done in order to obtain approximately the same number of instances for both classes. Figure 3.6 shows random oversampling. Random Over-Sampling increases the number of instances in the minority class by randomly replicating them in order to present a higher representation of the minority class in the sample. There is one important factor needing to be noticed that it needs to use oversampling after training and

test datasets split. Oversampling before splitting the data can allow the exact same observations to be present in both the test and training sets. This can allow models to simply memorize specific data points and cause overfitting and poor generalization to the test data.

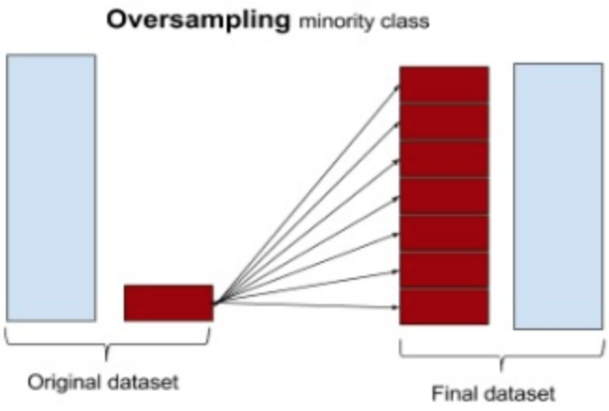

Figure 3.6: Random Over-Sampling [7]

Feature Scaling

The units of each explanatory variables of data are different, therefore if features do not scale , the model will be inaccurate. The formula for min-max normalization is in euqation 3.4.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.4}$$

After the min-max normalization, the values in each variable are scaled between 0 and 1. Min-max normalization has the advantages of presenting all relationships in the data, and no bias will be produced[22]. Normalization is another popular feature scaling method, but it is not applied in this project because there is no special needs for any normal distribution of dataset.

Dataset Split

Figure 3.7 illustrates the frame of dataset split. Before building model, dataset needs to be splited into either two or three parts. Usually, there are three parts, and they are training, validation and test datasets. However, in this project, only training and test datasets are used to build and evaluate the model. The difference between split two and three datasets is that models will be trained and hyparameters in models will be tuned in the training dataset or in training and valiation dataset. Three fold

cross validation is used to tune hyperparameters as the limited of computation. Since there are enough observations in the dataset, the ratio 7:3 will be used to split the dataset. As the Figure 3.7 shows, 70% data is used in training dataset, and the rest is in test dataset.

Figure 3.7: Split Dataset [11]

3.3 Feature Selection

Feature selection is one important step for building models. It selects the most important variables for the model. Subset Selection is used to do feature selection, and it selects the best model out of 2^p possibilities of model construction. The function **"regsubsets(...)"** from **"leaps"** can be used for the task to identify the best subset selection. Figure 3.8 summarizes the best subset models. An asterisk indicates that a given variable is included in the corresponding model.

- For the best subset of size 1, it puts 1 stars next to the variables that are in the best subset of size 1. In this project, **"year"** is the best subset of size 1.
- For the best subset of size 10, all variables except **"Mdat"** is the best subset.
- For the best subset of size 11, all variables are used.

Accuracy Metrics

Mallows Cp, Bayesian information criterion, Residual sum of squares and adjusted R^2 are regression model accuracy metrics. Cp assesses the fit of regression model. Cp is defined as:

$$C_p = \frac{SSE_p}{S^2} - N + 2P, \tag{3.5}$$

RSS measures the the amount of variance in dataset that cannot be explained by the regression model. Formular for RSS is

$$\text{RSS} = \sum_{i=1}^{n} (y_i - f(x_i))^2, \tag{3.6}$$

```
11 Variables  (and intercept)
              Forced in Forced out
year               FALSE      FALSE
Month              FALSE      FALSE
Mdat               FALSE      FALSE
Yday               FALSE      FALSE
Wday               FALSE      FALSE
hour               FALSE      FALSE
humidity           FALSE      FALSE
pressure           FALSE      FALSE
temperature        FALSE      FALSE
wind_direction     FALSE      FALSE
wind_speed         FALSE      FALSE
1 subsets of each size up to 11
Selection Algorithm: exhaustive
          year Month Mdat Yday Wday hour humidity pressure temperature
1  ( 1 )  "*"  " "   " "  " "  " "  " "  " "      " "      " "
2  ( 1 )  "*"  " "   " "  " "  " "  " "  " "      " "      " "
3  ( 1 )  "*"  " "   " "  " "  " "  " "  "*"      " "      " "
4  ( 1 )  "*"  "*"   " "  " "  " "  " "  "*"      " "      " "
5  ( 1 )  "*"  "*"   " "  " "  " "  " "  "*"      " "      "*"
6  ( 1 )  "*"  "*"   " "  " "  " "  " "  "*"      "*"      "*"
7  ( 1 )  "*"  "*"   " "  " "  " "  "*"  "*"      "*"      "*"
8  ( 1 )  "*"  "*"   " "  " "  " "  "*"  "*"      "*"      "*"
9  ( 1 )  "*"  "*"   " "  "*"  " "  "*"  "*"      "*"      "*"
10  ( 1 ) "*"  "*"   " "  "*"  "*"  "*"  "*"      "*"      "*"
11  ( 1 ) "*"  "*"   "*"  "*"  "*"  "*"  "*"      "*"      "*"
          wind_direction wind_speed
1  ( 1 )  " "            " "
2  ( 1 )  " "            "*"
3  ( 1 )  " "            "*"
4  ( 1 )  " "            "*"
5  ( 1 )  " "            "*"
6  ( 1 )  " "            "*"
7  ( 1 )  " "            "*"
8  ( 1 )  "*"            "*"
9  ( 1 )  "*"            "*"
10  ( 1 ) "*"            "*"
11  ( 1 ) "*"            "*"
```

Figure 3.8: Summary of best subset models

where y_i is the i^{th} value of the variable to be predicted, x_i is the i^{th} value of the explanatory variable. And $f(x_i)$ is the predicted value of y_i.

The formula for BIC is

$$\text{BIC} = \ln(n)k - 2\ln(\widehat{L}). \tag{3.7}$$

As R^2 always increases as the number of explanatory variables increases, so Adjusted R^2 is used.

$$\bar{R}^2 = 1 - \frac{SS_{\text{res}}/\text{df}_e}{SS_{\text{tot}}/\text{df}_t} \tag{3.8}$$

Best Subset Selection

To select the best subset size, the model needs to have lowest Cp, BIC, RSS value and highest Adjusted RSq.

Figure 3.9 has plots of subset size and its corresponding criterion. From Figure 3.9, Cp and Adjusted RSq suggests size 11 is better, while BIC indicates size 10 is better. Cp and BIC are the criteria that have a penalty term related to the number of predictors. BIC penalizes more large models, so BIC tends to select a smaller model comparing to Cp.

Based on these evidences, size 11 is the best subset size. And in size 11, variables year, Month, Mdat, Yday, Wday, hour, humidity, pressure, temperature, wind direction and wind speed are the best subset.

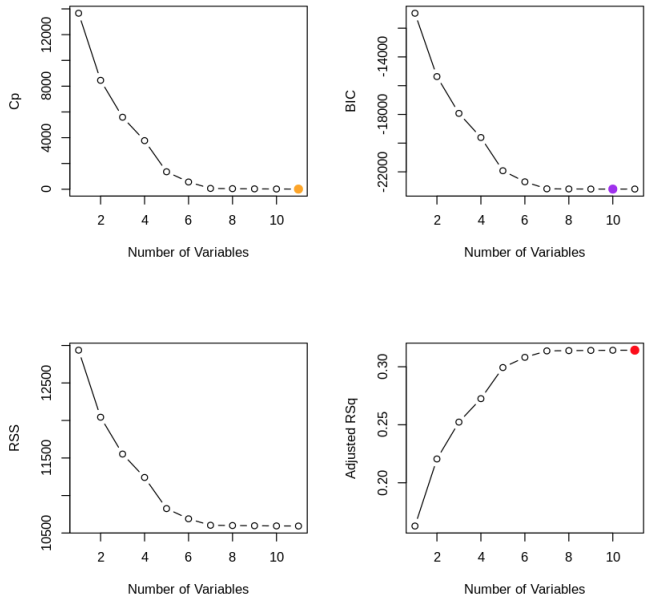

Figure 3.9: Feature Selection

This chapter has discussed the steps of data cleaning. Moreover, we have explored the data very well and overcome some challenges such as missing value, imbalance class. Feature scaling, Dataset split, and feature selection are also the critical steps in this project. After looking at the data, we finally build models. The next chapter will present the results of models and announce the best performance model. Future work of this project is also mention in the last chapter.

Chapter 4

Results and Future Research

4.1 Results

Table 4.1 is the summary of model result. As highlighted, Random Forest gives the best performance. it has the highest accuracy, F measure. When it considers F-measure, other models have very poor F measure. That's because some model either has good prediction on severe weather events or good prediction on non-severe weather events. However, the model that has both good predictions is the best one.

	Accuracy	Recall	Precision	F-Measure
Logistic Regression	0.7371	0.76271	0.06042	0.111970012513212
LDA	0.7289	0.7277	0.0598	0.110517993650794
QDA	0.83729	0.77667	0.07688	0.139910701423467
KNN	0.9515	0.8305	0.2872	0.426804330321195
SVM	0.8873	0.8847	0.1485	0.254312717770035
Decision Tree	0.8227	0.55254	0.06686	0.119285839199225
Random Forest	0.9867	0.6169	0.7309	0.66907880991245

Table 4.1: Models' Results

Tuning Hyper parameters

Table 4.2 gives the summary of tuned optimal hyper parameters used in each model. Due to the limit time for, I simply use the default values for some hyparameter. For logistic regression, 0.5 is threshold; while LDA and QDA have no hypermeter to tune. K nearest neighbour chooses 5 as optimal neighbourhood size. In Support vector

machine, σ and γ are equal. γ in the Radial Basis Function kernel determines the reach of a single training instance. If the value of Gamma is low, then every training instance will have a far reach. Conversely, high values of gamma mean that training instances will have a close reach. The cost parameter **"C"** decides how much an SVM should be allowed to "bend" with the data. For a low cost, you aim for a smooth decision surface and for a higher cost, you aim to classify more points correctly[20]. Complexity parameter in decision tree is the minimum improvement in the model needed at each node.if any split does not increase the overall R^2 of the model by at least Cp (where R^2 is the usual linear-models definition) then that split is decreed to be, a priori, not worth pursuing. The program does not split said branch any further, and saves considerable computational effort[26]. **"mtry"** defines the number of variables randomly sampled as candidates at each split in Random Forest.

Model	Logistic Regression	LDA	QDA	KNN	SVM	Decision Tree	Random Forest
Optimal Hyperparameter	none	none	none	k=5	Cost=1 Sigma=0.06377	Cp=0.044	Mtry=2

Table 4.2: Optimal Hyper parameters

4.2 Future Research

Even this project is finished but there is still some space for improvement.

- The dataset used is not the best. This dataset only includes limited information of weather, and there is no indication ofwhether an severe weather happens. Using a dataset with more information could make the prediction more accurate.
- Time series models such as ARIMA, are popularly used in analyzing time series dataset. This data is also time series. Prediction can be more accurate if combining both time series models and statistical learning models.
- Geographic locations also affect weather. People may find that the weather from different places is similar or different. For example, the weather in Montreal, Canada, is similar to Ha'erbing, China. They are both warm to hot in summers, and winter brings freezing, snowy, windy and icy weather. However, Vancouver's

and Montreal's weather has much difference. Therefore, including geographic locations can also help increase the prediction accuracy.

Bibliography

[1] Beaufort wind scale. https://www.weather.gov/mfl/beaufort.

[2] Extreme Weather. https://www.earlyalert.com/services/extreme-weather/.

[3] Humidity. https://climate.ncsu.edu/edu/Humidity.

[4] Ontario climate. https://digimarconcanada.ca/ontario-climate/.

[5] Weather. https://www.nationalgeographic.org/encyclopedia/weather/.

[6] Sidath Asiri. Machine learning classifiers. https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623, Jun 2018.

[7] Tom Fawcett. Learning from imbalanced classes. https://www.svds.com/learning-imbalanced-classes/.

[8] Rohith Gandhi. Support vector machine - introduction to machine learning algorithms. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47, Jul 2018.

[9] Shanta Rangaswamy Gangadhar Shobha. Chapter 8 machine learning. *Handbook of Statistics*, 38:197–228, 2018.

[10] Selfish Gene. Historical hourly weather data 2012-2017. "https://www.kaggle.com/selfishgene/historical-hourly-weather-data".

[11] Google. Training and test sets: Splitting data. https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data.

[12] Karen Grace-Martin. Two recommended solutions for missing data: Multiple imputation and maximum likelihood. "https://www.theanalysisfactor.com/missing-data-two-recommended-solutions/".

[13] Karen Grace-Martin. What is the difference between mar and mcar missing data? https://www.theanalysisfactor.com/mar-and-mcar-missing-data/.

[14] John W. Graham. Missing data analysis: making it work in the real world. *Annual Review of Psychology*, 60(1):549–576, 2009.

[15] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.

[16] Robert Kabacoff. *R in action: data analysis and graphics with R*. Manning, 2015.

[17] Yunjie Liu, Evan Racah, Mr Prabhat, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner, and William Collins. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. 05 2016.

[18] Emill Lundkvist. *Decision Tree Classification and Forecasting of Pricing Time Series Data*. Master's thesis, KTH Royal Institute of Technology, 2014.

[19] Anne Marie. Temperature conversion formulas. `https://www.thoughtco.com/temperature-conversion-formulas-609324/`, November 2019.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] Prabhat, Surendra Byna, Venkatram Vishwanath, Eli Dart, Michael Wehner, and William D. Collins. Teca: Petascale pattern recognition for climate science. *Computer Analysis of Images and Patterns Lecture Notes in Computer Science*, page 426–436, 2015.

[22] Kevin L. Priddy and Paul E. Keller. *Artificial neural networks: an introduction*. SPIE, 2005.

[23] Evan Racah, Christopher Beckham, Tegan Maharaj, Prabhat, and Christopher J. Pal. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. *CoRR*, abs/1612.02095, 2016.

[24] Water Services. The flood of 2013. `https://www.calgary.ca/UEP/Water/Pages/Flood-Info/Flooding-History-Calgary.aspx`, Oct 2017.

[25] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai Kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. 06 2015.

[26] Terry Therneau and Elizabeth Atkinson. An introduction to recursive partitioning using the rpart routines. *Mayo Clinic*, 61, 01 1997.

[27] Ziqi Zhu, Li Zhuo, Panling Qu, Kailong Zhou, and Jing Zhang. Extreme weather recognition using convolutional neural networks. *2016 IEEE International Symposium on Multimedia (ISM)*, 2016.

Appendix A

Code

lstlisting

```
library(tidyr)
library(dplyr)
library(tidyverse)
library(lubridate)
library(leaps)
library(imbalance)
library(ROSE)
library(dplyr)
library(rsample)
library(rpart.plot)
library(caret)
library(e1071)
library(kernlab)
library(randomForest)
library(rpart.plot)
library(ROCR)

cities<-read.csv("/Users/sunnysmac/Desktop/R code/historical-hourly-weather
View(cities)
humidity <- read.csv("/Users/sunnysmac/Desktop/R code/historical-hourly-we
  gather(2:37,
         key = "city",
         value = "humidity")
pressure <- read.csv("/Users/sunnysmac/Desktop/R code/historical-hourly-we
  gather(2:37,
         key = "city",
         value = "pressure")
```

```
temperature <- read.csv("/Users/sunnysmac/Desktop/R code/historical-hourly-
  gather(2:37,
         key = "city",
         value = "temperature")
weather <- read.csv("/Users/sunnysmac/Desktop/R code/historical-hourly-wea
  gather(2:37,
         key = "city",
         value = "weather")
### Maybe need to set weather as factor.
#View(weather)

wind_direction <- read.csv("/Users/sunnysmac/Desktop/R code/historical-hou
  gather(2:37,
         key = "city",
         value = "wind_direction")
wind_speed <- read.csv("/Users/sunnysmac/Desktop/R code/historical-hourly-
  gather(2:37,
         key = "city",
         value = "wind_speed")

#join tables into single dataset
dat <- humidity %>%
  inner_join(pressure, by = c("datetime", "city")) %>%
  inner_join(temperature,  by = c("datetime", "city")) %>%
  inner_join(weather, by = c("datetime", "city")) %>%
  inner_join(wind_direction, by = c("datetime", "city")) %>%
  inner_join(wind_speed, by = c("datetime", "city")) %>%
  left_join(cities, by = c("city" = "City"))

data_city<-subset(dat,Country=="Canada",select=c(datetime,city,humidity,pr

data_city

# split datatime to 6 variables
data_city$year<-year(data_city$datetime)
data_city$Month<-month(data_city$datetime)
data_city$Mdat<-mday(data_city$datetime)
data_city$Yday<-yday(data_city$datetime)
data_city$Wday<-wday(data_city$datetime)
data_city$hour<-hour(data_city$datetime)


write.csv(data_city,'/Users/sunnysmac/Desktop/R code/historical-hourly-wea
```

```
# Get Toronto data
data_Toronto<-subset(data_city,city=="Toronto",select=c(datetime,year,Montl
View(data_Toronto)

write.csv(data_Toronto,'/Users/sunnysmac/Desktop/R code/historical-hourly-v

# Missing value Analysis
library(imputeTS)

plotNA.distribution(data_Toronto$pressure,main="Toronto pressure")
plotNA.distribution(data_Toronto$humidity,main="Toronto humidity")
plotNA.distribution(data_Toronto$temperature,main="Toronto temperature")

plotNA.distribution(data_Toronto$wind_direction,main="Toronto wind directio
plotNA.distribution(data_Toronto$wind_speed,main="Toronto wind speed")

# Multiple Imputation
library(lattice)
library(nnet)
library(MASS)
library(mice)

imp2<-mice(data = data_Toronto,m=3)
fit2<-with(imp2,lm(temperature~year+Month+Mdat+Yday+Wear+hour+humidity+pres
summary(fit2)
pooled2<-pool(fit2)
summary(pooled2)
result_Toronto=complete(imp2,action = 2)


# Save result to new csv file.
write.csv(result_Toronto,'/Users/sunnysmac/Desktop/R code/historical-hourly

# Get the saved data
c_Toronto<-read.csv("c_Toronto.csv")
str(c_Toronto)


# create the label to extreme weather.
weather<-c_Toronto$weather
c_Toronto$is_extreme_weather<-weather

class(c_Toronto$s_extreme_weather)
```

```
levels(c_Toronto$is_extreme_weather)
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
levels(c_Toronto$is_extreme_weather)[levels(c_Toronto$is_extreme_weather)=
```

```
# do normalization to the dataset.
normalize <- function(x) {
    return ((x - min(x)) / (max(x) - min(x)))
  }

Norm_Toronto <- as.data.frame(lapply(select_if(c_Toronto,is.numeric), norm
new_Toronto<-cbind(Norm_Toronto,c_Toronto[14])

# Split train & test dataset.
set.seed(666)
train_index <- sample(x = nrow(new_Toronto), size = nrow(new_Toronto)*0.7)
train <- new_Toronto[train_index, ]
test <- new_Toronto[-train_index, ]
table(new_Toronto$is_extreme_weather)
table(train$is_extreme_weather)


# Oversampling
over_train <- ROSE::ovun.sample(is_extreme_weather~year+Month+Mdat+Yday+Wd
table(over_train[["data"]][["is_extreme_weather"]])
toronto.train<-data.frame(over_train[["data"]])

# Best subset selection
set.seed(666)
fit_all = regsubsets(is_extreme_weather ~ ., data = toronto.train, nvmax =
# nvmax is a tuning parameter specifying the maximum size of subsets to exa
fit_all_sum<-summary(fit_all)
fit_all_sum
names(fit_all_sum)
fit_all_sum$bic


best.cp=which.min(fit_all_sum$cp)
best.bic=which.min(fit_all_sum$bic)
best.adj2=which.max(fit_all_sum$adjr2)
# Choose 11.

par(mfrow=c(2,2))
#Plotting for Cp
plot(fit_all_sum$cp, xlab = "Number of Variables", ylab = "Cp", type = 'b')
points(best.cp, fit_all_sum$cp[best.cp],
       col = "orange", cex = 2, pch = 20)
```

```
#Plotting for BIC
plot(fit_all_sum$bic, xlab = "Number of Variables", ylab = "BIC", type = 'l
points(best.bic, fit_all_sum$bic[best.bic],
       col = "purple", cex = 2, pch = 20)

#plotting for RSS and Adjusted R-Square.
plot(fit_all_sum$rss, xlab = "Number of Variables", ylab = "RSS", type = "l
plot(fit_all_sum$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq
best_adj_r2 = which.max(fit_all_sum$adjr2) #
points(best.adj2, fit_all_sum$adjr2[best.adj2], col = "red",cex = 2, pch =


coef(fit_all,11)
#par(mfrow=c(3,1))
plot(fit_all, scale='bic')
plot(fit_all,scale='Cp')
plot(fit_all,scale='adjr2')

# Build Model
formular<-is_extreme_weather~year+Month+Mdat+Yday+Wday+hour+humidity+pressu
formular

train.control1 <- trainControl(method = "cv", number = 3)

# Logistic Regression
set.seed(2020)
fit.lr.full<-train(formular,method="glm",family="binomial", data = toronto

# LDA
set.seed(2020)
fit.lda.full <- train(formular, data=toronto.train, method="lda", trContro

# QDA
set.seed(2020)
fit.qda.full <- train(formular, data=toronto.train, method="qda", trContro

# kNN
set.seed(2020)
fit.knn.full <- train(formular, data=toronto.train, method="knn", trContro

# SVM
set.seed(2020)
fit.svm.full<-train(formular, data=toronto.train, method="svmRadial",trCon
```

```
# Random Forest
set.seed(2020)
fit.rf.full<-train(formular, data=toronto.train, method="rf",trControl = tr

# Decision Tree
set.seed(2020)
fit.tree.full<-train(formular, data=toronto.train, method="rpart",trContro

# collect resamples
results <- resamples(list(LR=fit.lr.full, LDA=fit.lda.full, KNN=fit.knn.ful
results

## Optimal Model
fit.lr.full$finalModel
fit.lr.full$bestTune
fit.lda.full$finalModel
fit.qda.full$finalModel
fit.qda.full$bestTune
fit.knn.full$finalModel
fit.knn.full$bestTune
fit.svm.full$finalModel
fit.svm.full$bestTune
fit.svm.full$results
fit.tree.full$finalModel
rpart.plot(fit.tree.full$finalModel)
fit.tree.full$bestTune
fit.rf.full$finalModel
fit.rf.full$bestTune


# Predict Final model
# Logistic Regression
predict.lr<-predict(fit.lr.full,test)
confusionMatrix(predict.lr,test$is_extreme_weather)
2*(0.76271*0.06042)/(0.76271+0.06042)

# LDA
predict.lda<-predict(fit.lda.full,test)
confusionMatrix(predict.lda,test$is_extreme_weather)

# QDA
predict.qda<-predict(fit.qda.full,test)
confusionMatrix(predict.qda,test$is_extreme_weather)
```

```
# Specificity : 0.83729
# Neg Pred Value : 0.07688

predict.knn<-predict(fit.knn.full,test)
confusionMatrix(predict.knn,test$is_extreme_weather)
2*(confusionMatrix(predict.knn,test$is_extreme_weather)$byClass[2] * confus

predict.svm<-predict(fit.svm.full,test)
confusionMatrix(predict.svm,test$is_extreme_weather)
2*(confusionMatrix(predict.svm,test$is_extreme_weather)$byClass[2] * confus


predict.tree<-predict(fit.tree.full,test)
confusionMatrix(predict.tree,test$is_extreme_weather)
2*(0.06686 *0.55254 )/(0.06686+0.55254 )

predict.rf<-predict(fit.rf.full,test)
confusionMatrix(predict.rf,test$is_extreme_weather)
```