CONCORDIA UNIVERSITY

Department of Computer Science and Software Engineering

Predict Montreal weather with classic machine learning method

COMP 432
Project

Zhaoyang Li 27838824

Weichen Wan 40072743

Yufei Li 40165883

Concordia University

December 23, 2022

Contents

1 Goal **1**

2 Dataset Description and Manipulation **2**

2.1 Description 2

2.2 Dataset Manipulation 2

2.2.1 Features Selection 3

2.2.2 Standardization and Scaling 3

2.2.3 Merging 4

3 Analyses **5**

3.1 Evaluation Metric and Cross Validation 5

3.1.1 Evaluation Metric 5

3.1.2 Cross Validation 5

3.2 Logistic Regression 6

3.3 Linear Regression 6

3.4 Classic Neural Network 7

3.5 Recurrent Neural Networks 7

3.5.1 Description 7

3.5.2 Classification 8

3.6 Other Analyses 9

4 Conclusion **10**

4.1 Summaries 10

1 Goal

Weather is the most prevalent topic in small talk across the full world. This project will attempt to answer these questions for the city of Montreal. The first being whether it'll rain/snow and the second being the amount of rain/snow that will fall from the sky in a given day.

A common approach to this type of questiony is to build two separate models. One for classification to answer the first question and one for regression to answer the second. After fitting the two models the prediction process, is as follows:

$$\mathbb{E}[\text{Precipitation}|X] = \begin{cases} 0 & , \mathbb{P}[\text{Precipitation}|X] < \text{Threshold} \\ \text{Regression Prediction} & , \text{Otherwise} \end{cases}$$

In words, first the classification model will be used to probability of precipitation for a given day. If the probability of precipitation is above a threshold (usually 0.4 or 0.5), then the prediction of the amount of precipitation from the regression will be generated. If it is below the threshold, the value of zero will be the prediction. This approach is chosen for the majority of the modelling techniques learned in this course, as well as machine learning classification techniques.

2 Dataset Description and Manipulation

2.1 Description

We concern ourselves with giving day predictions for the weather station located at the Montreal International Airport. Data was chosen between 2013-2018 due to the quality of daily and hourly data being high, when compared to other weather stations.

Here is a list of relevant terminology. For further reference, see weather data source.

Dew Point Temperature (C), Heating degree-days, Cooling degree-days, Gusts,Pressure (kPa), max-tmp, min-tmp, mean-tmp, mean-RH, mean-DP, mean-press, mean-Wind-speed, Heat-Deg-Days

2.2 Dataset Manipulation

To be able to perform cross-validation, the dataset has been divided into two parts. The training set contains 75% observations, the testing set contains 25% observations.

Description of variables that were included on a daily basis (all temperatures are in Celsius):

- `max_tmp` - the maximum temperature
- `min_tmp` - the minimum temperature
- `mean_tmp` - the average temperature
- `mean_RH` - the average relative humidity expressed as a percentage.
- `mean_DP` - the average dew point temperature
- `mean_press` - the average station pressure
- `mean_Wind_speed` - the average wind speed (km/h) usually 10 kilometres above the ground
- `Heat_Deg_Days` - See Definition of Heating Degree-Days
- `Cooling_Deg_Days` - See Definition of Cooling Degree-Days
- `spd_max_gust` - the maximum gust speed (km/h) recorded in a day
- `Precipitation` - the total precipitation (mm) recorded in a day

2.2.1 Features Selection

Correlation Before fitting any models, it would be ideal to check the relation between all features, there are two main reasons,

- If two features are highly linear correlated, the model may have multi-collinearity effect.
- Since we are going to fit some of the models like a neural network which contains a lot of hyper parameters, by reducing the number of input, we may obtain less parameters to be tuning when fitting model and consequently reduce the computational load.

The set up in this analysis of features is to rule out those features that have a correlation coefficient of more than 0.7 in absolute value.
We summarize some of the discoveries in my correlation matrix as follows, here We summarize the result regarding the feature `mean_tmp`, i.e. mean temperature.

Correlation between `mean_tmp` and others

`max_tmp`	`min_tmp`	`mean_RH`
0.990	0.986	0.005
`mean_DP`	`mean_press`	`mean_Wind_speed`
0.971	-0.236	-0.187
`Heat_Deg_days`	`Cold_Deg_days`	`spd_max_gust`
-0.989	0.605	-0.045

As can be seen from the table above, several features are highly linearly correlated, namely `mean_tmp`, `max_tmp`, `mean_DP`, `Heat_Deg_days`. We choose `mean_tmp` as my feature, along with `mean_RH`, `mean_press`, `mean_Wind_speed`, `Cold_Deg_days` and `spd_max_gust`. There are 6 features in total.

2.2.2 Standardization and Scaling

Since our features are expressed in different measurement scales, we standardize or scale the features based on the following: If the feature can take either positive or negative values, then we standardize it. If the feature can only take positive values, then we scale it by dividing the range.

2.2.3 Merging

Since we are trying to predict the o given day (either rain or precipitation) by considering all features of the previous 7 days, it would convenient to rebuild a bigger dataset by stacking all the features in the past 7 days as the features for predicting the 8 th day. By rebuilding the dataset, we have $6 \times 7 + 2 \times 7 = 56$ features for predicting the 8th day.

3 Analyses

In this section, we will provide several methods to perform the analysis. For the machine learning models, we will use some common evaluation metric and cross validation to determine some models.

3.1 Evaluation Metric and Cross Validation

3.1.1 Evaluation Metric

We use the common evaluation metric described as follows,

- For classification, We use *Accuracy*[1] plus *Confusion Matrix*
- For regression, We use *Mean Square Error.*

3.1.2 Cross Validation

Classification We use a sequential cross validation based on month. We split the training data, then we train the model with one of the train set and use the trained model to predict the validation set, we record the number of errors we made. Again, we use data from other train set to train the model and make prediction for the validation set. As the procedure goes on when running all the training data, we obtain total number of error count, and compute the error rate accordingly.
We choose the model with smallest error rate and use the setting of the model to train all data in the training set and use it to predict the data in the testing set and obtain the error rate.

Regression The regression applies similar strategy of sequential cross validation described above for classification but slightly different when training the model. Since we are trying to predict the precipitation, we only use the observations with non-zero precipitation, i.e. the days with rain or snow. Then we make prediction for every single day of the next month. Of course, we are going to set negative precipitation to zero, then we compute the sum of square error. Similarly, we choose the model with least sum of square error and train the whole subset of training data with only raining or snowing days and predict the precipitation in the testing set and obtain MSE.

[1] Accuracy = 1 - Error Rate

3.2 Logistic Regression

The logistic regression is to perform the first step of the prediction, i.e. whether a given day will rain (snow) or not. For such a simple logistic regression, my learning strategy is designed as follows,

- We train logistic regression model with the train data and obtain parameters.
- We obtain predictions of training data from the model, and choose the classification threshold with minimum error rate.
- We use the trained model and the optimized threshold to predict the test data and evaluate the error rate.

After fitting a logistic regression model with the above procedure, my result are summarized in the following table, with optimized threshold be 0.44.

	Train	Test
Accuracy	0.633	0.563

Which seems NOT good, since it is close to 0.5.

3.3 Linear Regression

The linear regression to predict the precipitation is similar We have to sub-setting the days with precipitation and use the subset to train the model.

- We train linear regression model with the train data and obtain parameters.
- We obtain predictions of training data from the model.
- We use the trained model to predict the precipitation in the test data.
- To avoid negative prediction values, take the maximum of 0 and the predicted precipitation.
- We evaluate the prediction by MSE.

Note that, in this case, misclassification of raining day and non-raining day have, practically, no difference, so we do not assign cost of misclassification.

The training MSE and testing MSE are summarized below,

The training MSE and testing MSE are summarized below,

	Train	Test
MSE	58.125	68.413

3.4 Classic Neural Network

Single Layer Neural Network For single layer Neural Network, with consideration of the computational load, we only try hidden unit in the set $S = \{10, 20, \ldots, 150\}$. For each hidden unit in the set, to obtain the error rate, we should run approximately 48 times of a neural net with same level of hidden unit by the cross validation strategy mentioned above.

3.5 Recurrent Neural Networks

3.5.1 Description

Vanilla RNN model In the last section we looked at the performance of the classic Neural Networks. In this part, we will explore the world of recurrent neural networks. RNN is a type of ANN that has recurring feedback to itself. This recurring feedback helps the learning algorithm to understand the effect of the last observation(x_{t-1}) on its training and combine it with the information brought by the new input(x_{t-1}) while predicting the output y_t. The hidden. The hidden layer activations calculated at time $t-1$ are fed in as an input at time ”t”. A visualization of Vanilla RNN can be seen in Figure. 1

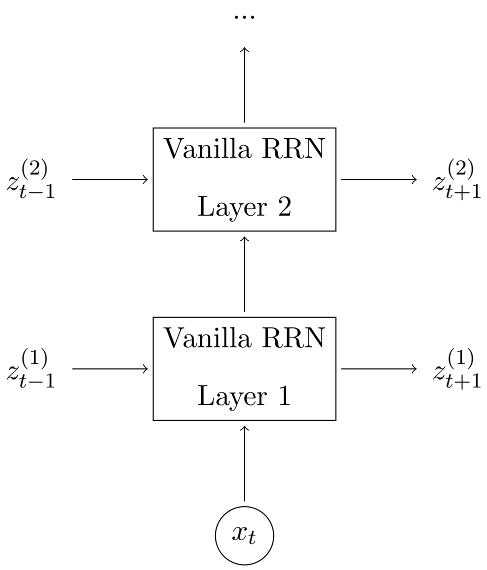

Figure 1: Visualization of the multilayer Vanilla RNN

VRNN Equations The Vanilla RNN works as follows:

Step t = 0 : Initial hidden state z_0 is assumed to be given

step t = 1,...,T the Vanilla RNN equations are:

$$a_t = Wz_{t-1} + Ux_t + b$$

$$z_t = \tanh(a_t)$$

$$y_t = f(Vz_t + c)$$

where $f(x)$ can be chosen based on the problem.

VRNN Flaws Compared to Feed-Forward networks, RNN are complex and use output as inputs, which could lead to noise amplification. It is also hard to draw a line between correlated variables which makes it really hard to analyze.

3.5.2 Classification

Implementation All the implementation is done with Pytorch package. Since RNN is really flexible and there is a lot of days with no precipitation, calibrating the model using the predefined loss functions tend to predict no Rain for almost every day. Therefore, the loss function we are using is BCEWithLogitsLoss()
The better way to define the loss function is to distinguish between the different type of misclassification which is still an challenge problem for us.

Overfitting Due to the high flexibility of the RNN it is important to have some sort of regularization technique to fight over fitting. This is done through dropout in the Pytorch package dropout which randomly zeros out input units of layers to fight over fitting. We have not performed a full cross validation to determine the best dropout rate.

3.6 Other Analyses

We also performed some of the traditional analyses for the classification, such as *Random Forest, Adaptive Boosting* and *K-Nearest Neighbor*

4 Conclusion

4.1 Summaries

We summarize my models and performances in the following tables, note that the confusion matrix will be in the following form,

$$\begin{bmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{bmatrix}$$

where (TP,FP,FN,TN) stands for (True Positive, False Positive, False Negative, True Negative).

Table 1: Summary of Classification Models

Model	Train Accuracy	Test Accuracy	Confusion Matrix
Logistic Regression	0.557	0.574	$\begin{bmatrix} 109 & 46 \\ 89 & 61 \end{bmatrix}$
Feed-forward Neural Net	0.544	0.537	$\begin{bmatrix} 96 & 59 \\ 80 & 70 \end{bmatrix}$
Random forest	0.622	0.554	$\begin{bmatrix} 114 & 41 \\ 74 & 76 \end{bmatrix}$
SVM	0.554	0.548	$\begin{bmatrix} 114 & 41 \\ 74 & 76 \end{bmatrix}$
RNN	0.532	0.522	$\begin{bmatrix} 90 & 38 \\ 71 & 70 \end{bmatrix}$

Table 2: Summary of Regression Models

Model	Training MSE	Testing MSE
Linear Regression	61.125	68.413
Feed-forward Neural Net	162.246	170
adaboost	72.246	78

We performed a series of analyses to give a model to predict the level of precipitation for the upcoming day using the historical weather data provided by the Montreal International Airport Weather Station.

We found that the linear regression outperformed the other classification methods I used. Additionally, with increased time and computational power more parameters and careful comparison of the multi layer and recurrent neural networks could have better performance, as well as incorporating multiple weather stations should serve as a baseline of further research into this problem.

References

[1] Bishop, C.M.: Pattern recognition and machine learning

[2] G. James, D Witten, T.H., Tibshirani, R.: An introduction to statistical learning.

[3] Mueller, A., Guido, S.: Introduction to machine learning with python

[4] Riahi, D.: Lecture and slides