EFC1 - Exercise 1

Rafael Claro Ito

September 2019

1 Theoretical Activities

1.1 Exercise 1

1.1.1 a) Obtain P(X) e P(Y).

$i)\ P_X(0):$
$P_X(0) = \sum_y P_{XY}(0,y)$
$P_X(0) = P_{XY}(0,0) + P_{XY}(0,1)$
$P_X(0) = 1/6 + 3/8$
$\boxed{P_X(0) = 13/24}$

$ii)\ P_X(1):$
$P_X(1) = \sum_y P_{XY}(1,y)$
$P_X(1) = P_{XY}(1,0) + P_{XY}(1,1)$
$P_X(1) = 1/8 + 1/3$
$\boxed{P_X(1) = 11/24}$

$iii)\ P_Y(0):$
$P_Y(0) = \sum_x P_{XY}(x,0)$
$P_Y(0) = P_{XY}(0,0) + P_{XY}(1,0)$
$P_Y(0) = 1/6 + 1/8$
$\boxed{P_Y(0) = 7/24}$

$iv)\ P_Y(1):$
$P_Y(1) = \sum_x P_{XY}(x,1)$
$P_Y(1) = P_{XY}(0,1) + P_{XY}(1,1)$
$P_Y(1) = 3/8 + 1/3$
$\boxed{P_Y(1) = 17/24}$

1.1.2 b) Calculate $\mathbf{P(X = 0|Y = 0)}$.

$$P_{X|Y}(x,y) = \frac{P_{XY}(x,y)}{P_Y(y)}$$
$$P_{X|Y}(0,0) = \frac{P_{XY}(0,0)}{P_Y(0)}$$
$P_{X|Y}(0,0) = \frac{1/6}{7/24}$
$\boxed{P_{X|Y}(0,0) = 4/7}$

1.1.3 c) Calculate E[X] e E[Y].

$i)\ E[X]:$
$E[X] = \sum_k x_k P_X(x_k)$
$E[X] = 0 \cdot P_X(0) + 1 \cdot P_X(1)$
$\boxed{E[X] = 11/24}$

$ii)\ E[Y]:$
$E[Y] = \sum_k y_k P_Y(y_k)$
$E[Y] = 0 \cdot P_Y(0) + 1 \cdot P_Y(1)$
$\boxed{E[Y] = 17/24}$

1.1.4 d) Are the variables independents? Why?

$P_{XY}(0,0) \neq P_X(0)P_Y(0)$
$\frac{1}{6} \neq \frac{13}{24} \cdot \frac{7}{24}$

Since $P_{XY}(0,0) \neq P_X(0)P_Y(0)$, the variables X and Y are not independents.

1.2 Exercise 2

1.2.1 a) Calculate H(X), H(Y) e H(X|Y).

$P_X(0) = P_{XY}(0,0) + P_{XY}(0,1) = 1/4$
$P_X(1) = P_{XY}(1,0) + P_{XY}(1,1) = 3/4$
$P_Y(0) = P_{XY}(0,0) + P_{XY}(1,0) = 3/8$
$P_Y(1) = P_{XY}(0,1) + P_{XY}(1,1) = 5/8$

$i)\ H(X):$
$H(X) = -\sum_x p(x)\log_2[p(x)]$
$H(X) = -P_X(0)\log_2[P_X(0)] - P_X(1)\log_2[P_X(1)]$
$H(X) = -\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{3}{4}\log_2\left(\frac{3}{4}\right)$
$H(X) = 0.5 + 0.3113$
$\boxed{H(X) = 0.8113}$

$ii)\ H(Y):$
$H(Y) = -\sum_y p(y)\log_2[p(y)]$
$H(Y) = -P_Y(0)\log_2[P_Y(0)] - P_Y(1)\log_2[P_Y(1)]$
$H(Y) = -\frac{3}{8}\log_2\left(\frac{3}{8}\right) - \frac{5}{8}\log_2\left(\frac{5}{8}\right)$
$H(Y) = 0.5306 + 0.4238$

$\boxed{H(Y) = 0.9544}$

$iii)\ H(X,Y):$
$H(X,Y) = -\sum_x \sum_y p(x,y)\log_2[p(x,y)]$
$H(X,Y) = -P_{XY}(0,0)\log_2[P_{XY}(0,0)] - P_{XY}(0,1)\log_2[P_{XY}(0,1)] - P_{XY}(1,0)\log_2[P_{XY}(1,0)] - P_{XY}(1,1)\log_2[P_{XY}(1,1)]$
$H(X,Y) = -0\cdot\log_2(0) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{3}{8}\log_2\left(\frac{3}{8}\right) - \frac{3}{8}\log_2\left(\frac{3}{8}\right)$
$H(X,Y) = 0 + 0.5 + 0.5306 + 0.5306$
$\boxed{H(X,Y) = 1.5613}$

1.2.2 b) Calculate H(X|Y) e H(Y|X).

$i)\ H(X|Y):$
$H(X,Y) = H(Y) + H(X|Y)$
$H(X|Y) = H(X,Y) - H(Y)$
$H(X|Y) = 1.5613 - 0.9544$
$\boxed{H(X|Y) = 0.6068}$

$ii)\ H(Y|X):$
$H(X,Y) = H(X) + H(Y|X)$
$H(Y|X) = H(X,Y) - H(X)$
$H(Y|X) = 1.5613 - 0.8113$
$\boxed{H(Y|X) = 0.75}$

1.2.3 c) Calculate I(X,Y).

$I(X,Y) = H(X) - H(X|Y)$
$I(X,Y) = 0.8113 - 0.6068$
$\boxed{I(X,Y) = 0.2044}$

$I(X,Y) = H(Y) - H(Y|X)$
$I(X,Y) = 0.9544 - 0.75$
$\boxed{I(X,Y) = 0.2044}$

1.3 Exercise 3

$C_1 \sim \mathcal{N}(-1,1)$
$\mu_1 = -1$
$\sigma_1 = 1$

$C_2 \sim \mathcal{N}(1,1)$
$\mu_2 = 1$

$\sigma_2 = 1$

1.3.1 a) Under the ML criteria, we have that x belongs to the class C_1 if $p(x|C_1) > p(x|C_2)$:

$$p(x|C_1) > p(x|C_2)$$

$$\frac{1}{\sqrt{2\pi\sigma_1^2}} exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right) > \frac{1}{\sqrt{2\pi\sigma_2^2}} exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)$$

$$\frac{1}{\sqrt{2\pi \cdot 1^2}} exp\left(-\frac{(x-(-1))^2}{2 \cdot 1^2}\right) > \frac{1}{\sqrt{2\pi \cdot 1^2}} exp\left(-\frac{(x-1)^2}{2 \cdot 1^2}\right)$$

$$exp\left(-\frac{(x+1)^2}{2}\right) > exp\left(-\frac{(x-1)^2}{2}\right)$$

$$-(x+1)^2 > -(x-1)^2$$

$$-x^2 - 2x - 1^2 > -x^2 + 2x - 1$$

$$-2x > 2x$$

$$\boxed{x < 0}$$

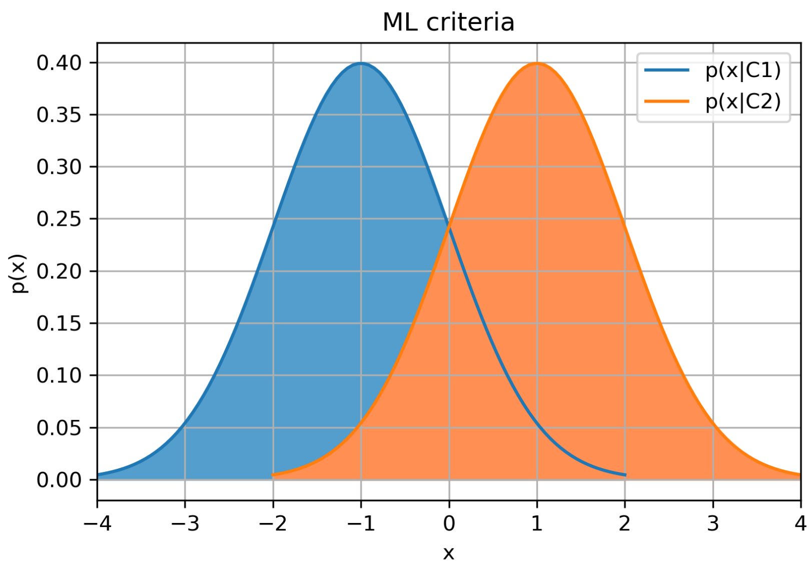


Figure 2: Visualization of C1 and C2 distributions under the ML criteria

As we can see in the figure 2, for x < 0 the class C1 will be chosen.

1.4 b) Under the MAP criteria, we have that x belongs to the class C_1 if $p(C_1|x) > p(C_2|x)$:

$$p(C_1) = 0.7$$

$$p(C_2) = 0.3$$

$$p(C_1|x) > p(C_2|x)$$

Applying Bayes' rule, we have:

$$\frac{p(x|C_1) \cdot p(C_1)}{p(x)} > \frac{p(x|C_2) \cdot p(C_2)}{p(x)}$$

$$\frac{1}{\sqrt{2\pi\sigma_1^2}} exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right) \cdot p(C_1) > \frac{1}{\sqrt{2\pi\sigma_2^2}} exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right) \cdot p(C_2)$$

$$\frac{1}{\sqrt{2\pi \cdot 1^2}} exp\left(-\frac{(x-(-1))^2}{2 \cdot 1^2}\right)(0.7) > \frac{1}{\sqrt{2\pi \cdot 1^2}} exp\left(-\frac{(x-1)^2}{2 \cdot 1^2}\right)(0.3)$$

$$exp\left(-\frac{(x+1)^2}{2}\right)(0.7) > exp\left(-\frac{(x-1)^2}{2}\right)(0.3)$$

Taking the ln from both sides:

$$-\frac{(x+1)^2}{2} + \ln(0.7) > -\frac{(x-1)^2}{2} + \ln(0.3)$$

$$-x^2 - 2x - 1 + 2\ln(0.7) > -x^2 + 2x - 1 + 2\ln(0.3)$$

$$-2x + 2\ln(0.7) > 2x + 2\ln(0.3)$$

$$-x + \ln(0.7) > x + \ln(0.3)$$

$$2x < \ln(0.7) - \ln(0.3)$$

$$x < \frac{\ln(0.7) - \ln(0.3)}{2}$$

$$\boxed{x < 0.4236}$$

As we can see in the figure 3, for x < 0.4236 the class C1 will be chosen.

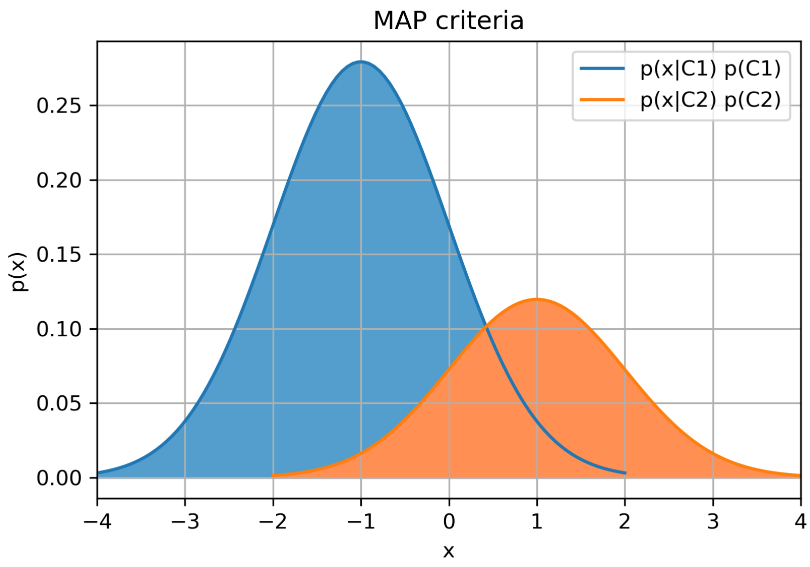


Figure 3: Visualization of C1 and C2 distributions under the MAP criteria

2 Computational Activities

All code presented and all figures showed here can be found at the following GitHub repository:
`https://github.com/ito-rafael/IA006C-MachineLearning`
In this repository, one can found the following files:

- Jupyter Notebook
  - theoretical-ex.ipynb
  - pre-ex1.ipynb
  - pre-ex2.ipynb
  - ex1.ipynb
  - ex2.ipynb
- Python code
  - ex1.py
  - ex2.py
- LaTeX
  - efc1.tex

The notebook ”theoretical-ex.ipynb” is used to generate the figures 2 and 3. The notebook ”pre-ex1” is used only for data visualization. It shows how the dataset is split in order to use the k-fold cross-validation method. The notebook ”pre-ex2” has a similar use than the previous one. It is used to visualize the data, but besides the raw data, it also plots the dataset linearly transformed before and after the application of the hyperbolic tangent function. The two last notebooks, ”ex1” and ”ex2” actually implements all the functions to train the models and that answers the exercises 1 and 2.

The Python codes presented contain the same code of the notebooks. Finally, the LaTeXfile is the source code to generate this pdf.

2.1 Exercise 1

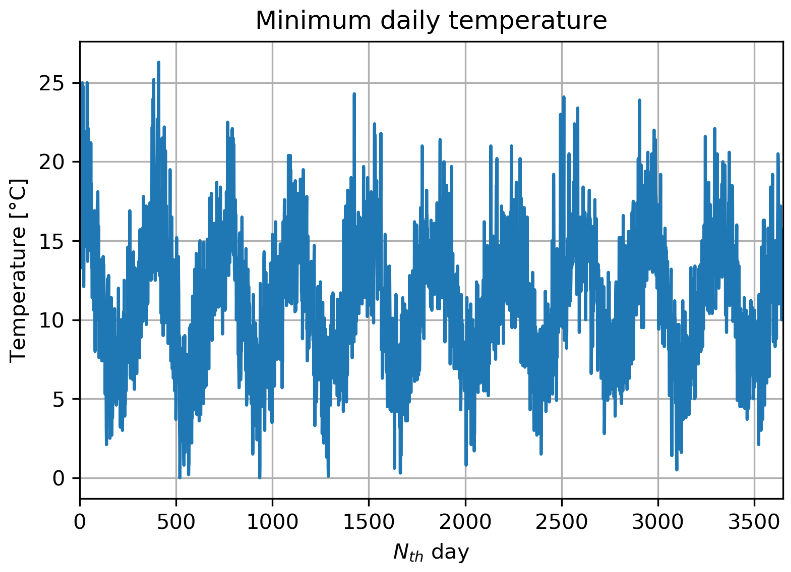


Figure 4: All data

The first thing we did before starting this exercise was to plot all the data, as can be seen in figure 4.

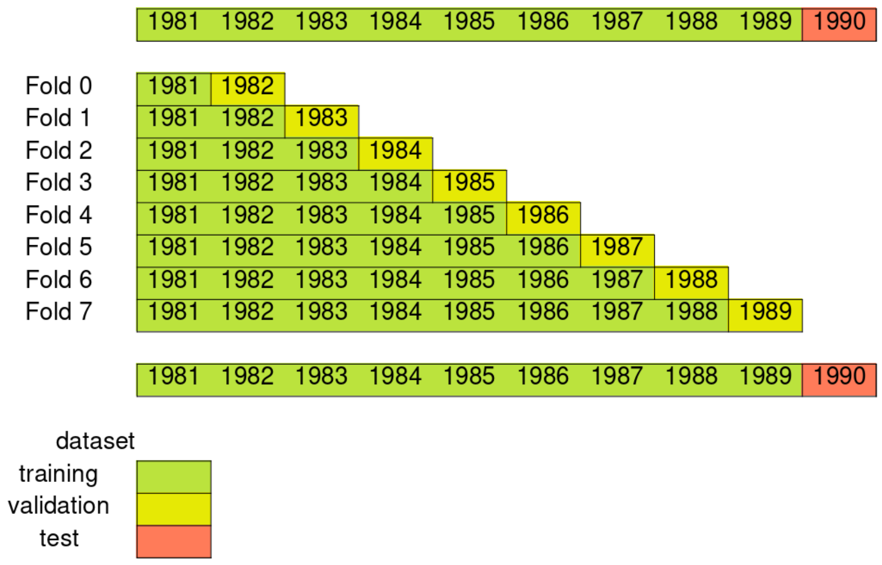
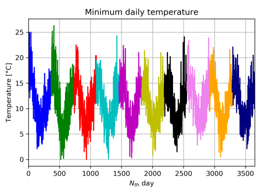


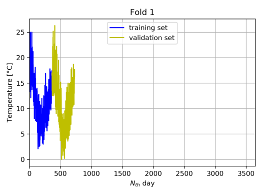
Figure 5: Folds used to cross-validate the model

The next step was to decide the number of folds used in the k-fold cross-validation method. Since we are treating a time series forecasting problem, it was decided to split the dataset equally to the number of the years of the dataset excluding the data used to test the model, so nine folds were used. When creating the folds, a method like the one that is showed in figure 5 was used.

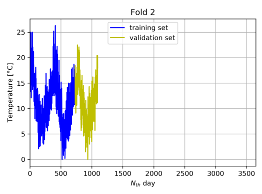
In this method, only past data is used to predict forward-looking data. The first fold is formed by only the first year of the dataset, the year 1981, while the year used at the validation step is the next year, 1982. The second fold uses all the data of the previous fold (train plus validation) as the data to be trained, and the next year, 1983, as the validation data. Doing this for all the folds we get the configuration showed in figure 6.



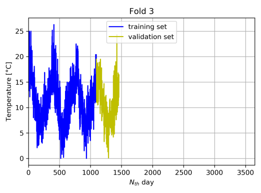
(a) Dataset split



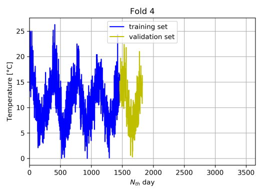
(b) Fold 1



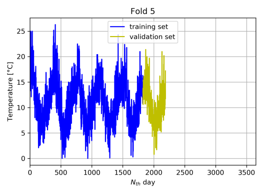
(c) Fold 2



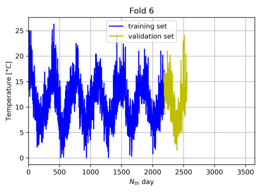
(d) Fold 3



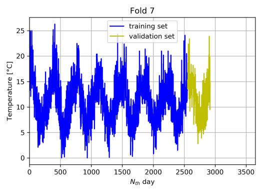
(e) Fold 4



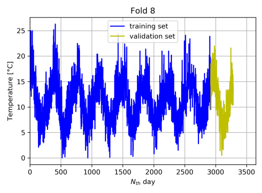
(f) Fold 5



(g) Fold 6



(h) Fold 7



(i) Fold 8

Figure 6: Folds used in cross-validation

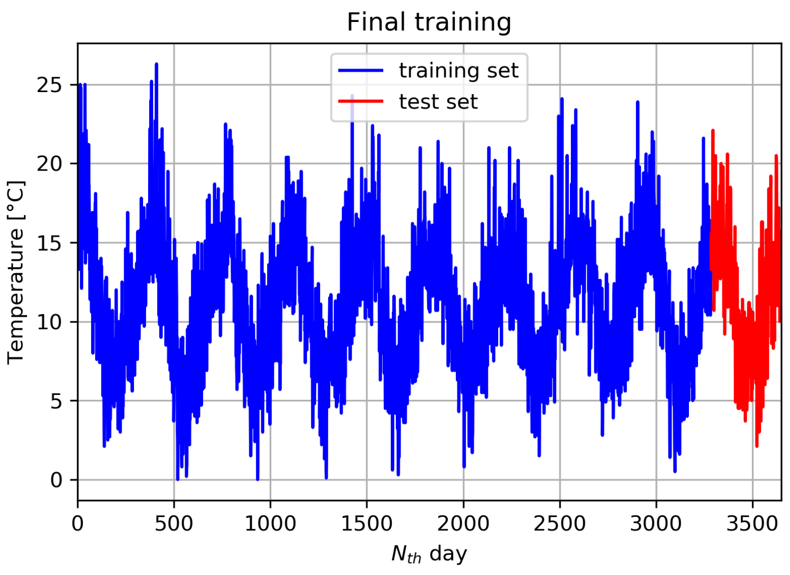


Figure 7

The goal of using this method of cross-validation is to find the best value for the hyperparameter K, the number of previous days used

as input for the model.
After choosing the hyperparameter, we no more use the folds division. Instead, all data before used for training and validation is now used only for training. This can be considered as the last training of the model and the one which will return the weights desired in this problem of linear regression. This can be seen in figure 7.

2.1.1 a) Progression of the RMSE measured in validation set in function of hyperparameter K (from 1 to 31):

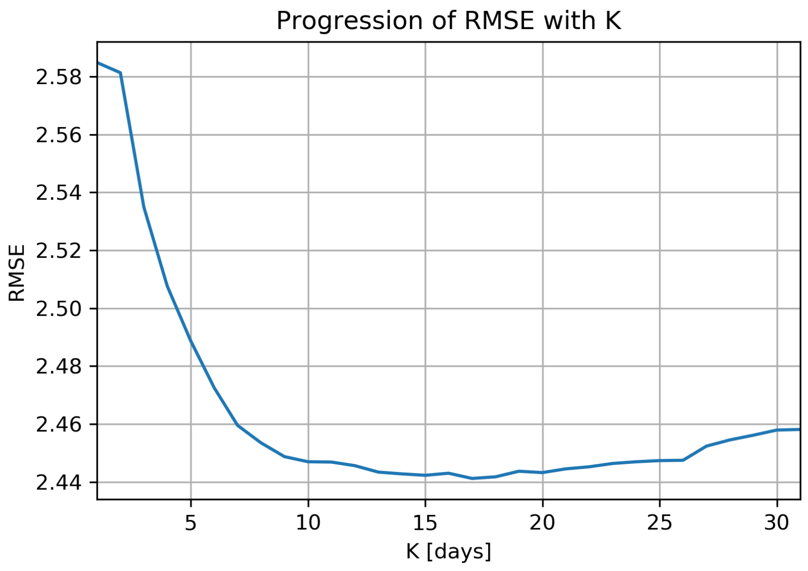


Figure 8

The model was trained for values of the hyperparameter K from 1 to 31 days. The progression of the root mean squared error (RMSE) measured in the validation set can be seen in figure 8.

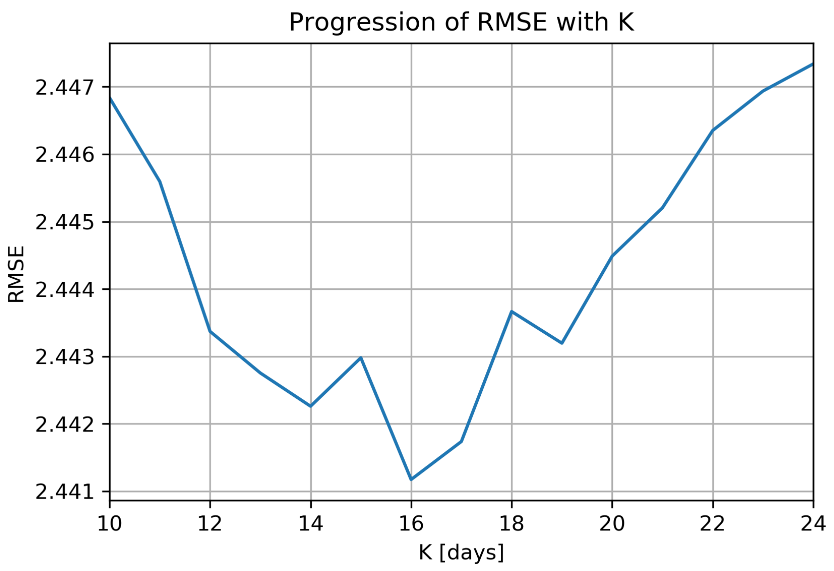


Figure 9

We can see that the error starts with values bigger than 2.5 degrees Celsius and decreases as K increases. After K equal to 20 days, we can see that the error only increases, possibly indicating an overfitting. Figure 9 shows a zoom of the previous curve for values of K from 10 to 24. With this zoomed curve, we can see that the value of K that gives the lowest error is 16.

2.1.2 b) Plot of predicted and real temperatures of test set

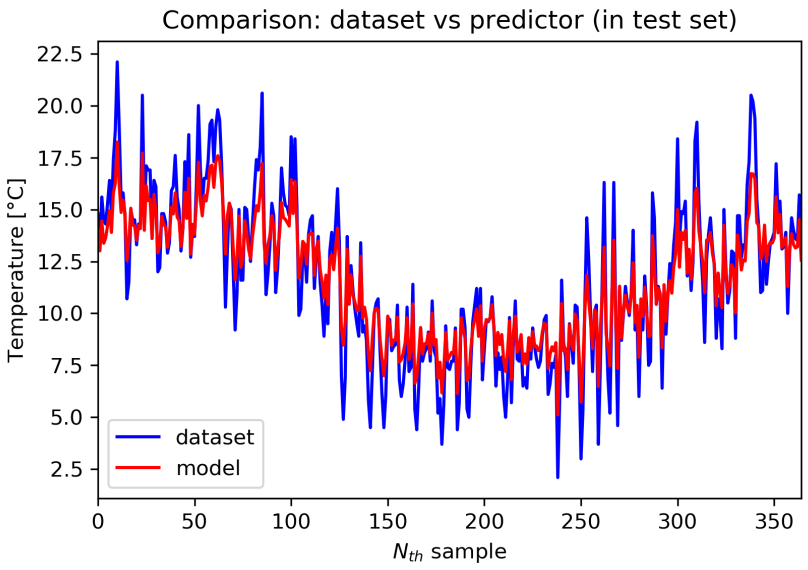


Figure 10

After finding the optimum K using the k-fold cross-validation, we trained the model considering all data. With the value of the weights, we can compute the predicted minimum temperature for the test set (year 1990) and compare with the real temperatures measured. This plot can be seen in figure 10.

After calculating the predicted values for all the inputs corresponding to the year 1990, we can calculate the overall RMSE between the predicted values and the real ones. The error measured with this model is 2.279 degrees Celsius.

2.2 Exercise 2

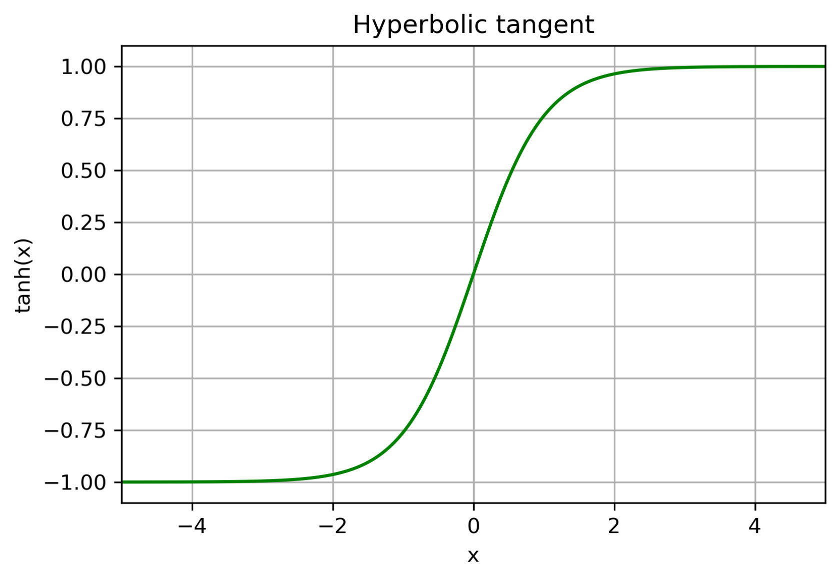


Figure 11

In this exercise, a non-linear transformation applied to the input data is explored in order to try to improve the temperature prediction. The non-linearity suggested is to apply the hyperbolic tangent to the input data. The function of the hyperbolic tangent for values from -5 to 5 can be seen in the figure 11.

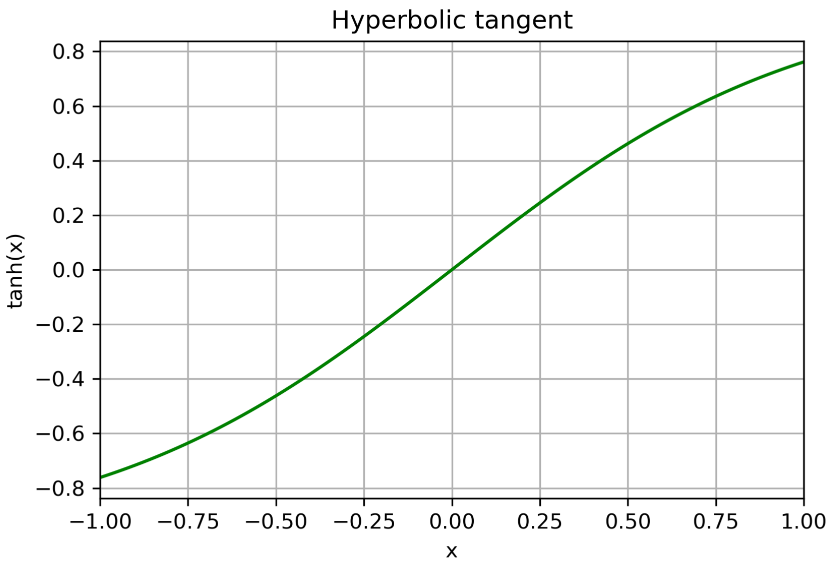


Figure 12

As we can see, the function is practically saturated for values smaller than -2.5 or bigger than 2.5. In fact, tanh(-3) returns -0.995 and tanh(3) returns 0.995. If we apply this function directly to the input data, almost all data would be under the saturated region. In order to avoid that, a linear transformation plus an offset was applied to all data. The goal is to work with the data in a region before the saturation region. It was chosen to work with the data in a range from -1 to 1. The figure 12 shows the behavior of the hyperbolic tangent in this region.

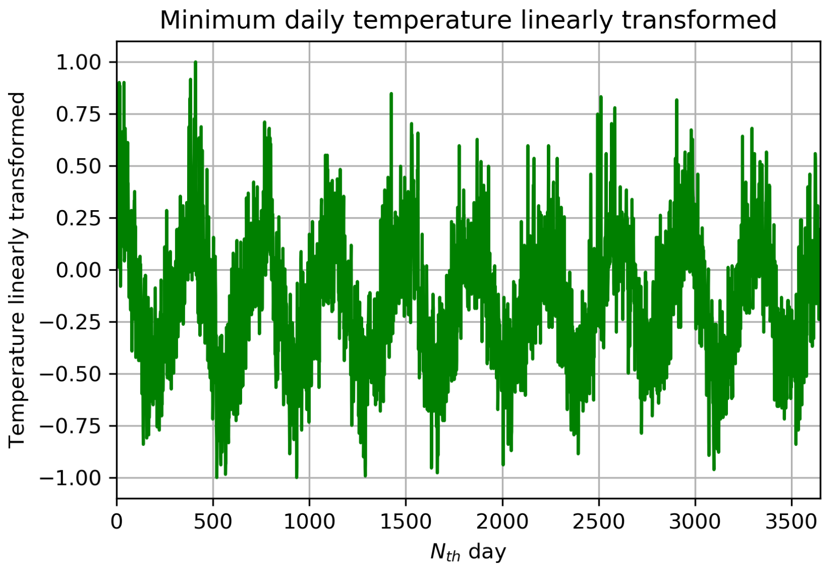


Figure 13

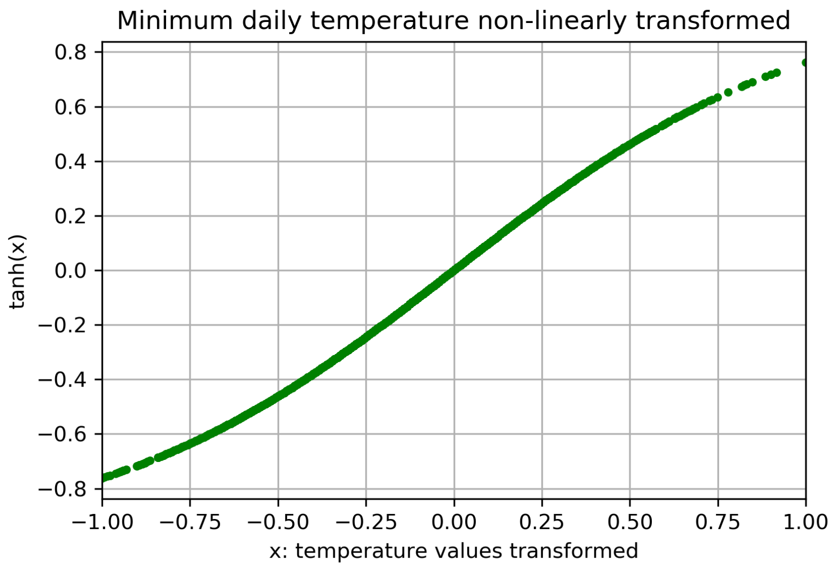


Figure 14

The procedure was to flatten the input data in a region from -1 to 1. To achieve this, we divided all data by the difference of the maximum and minimum input data, than applied a gain of 2 followed by an offset of -1. The first operation attenuated the data to be in the range of 0 to 1, the gain changed the range from 0 to 2 and the offset guaranteed

the data to be in the range from -1 to 1. The figure 13 shows the input data after this feature scaling and figure 14 shows the data transformed after applying the hyperbolic tangent.

2.2.1 a) Progression of the RMSE in function of the number of features T (from 1 to 100) taking K fixed to 5 days.

All functions used in the previous exercise were also utilized here. Additional functions were added in order to perform the feature scaling, to implement a one-dimensional golden-section search method (to find the best regularization coefficient) and a loop to train the model for each value of T.

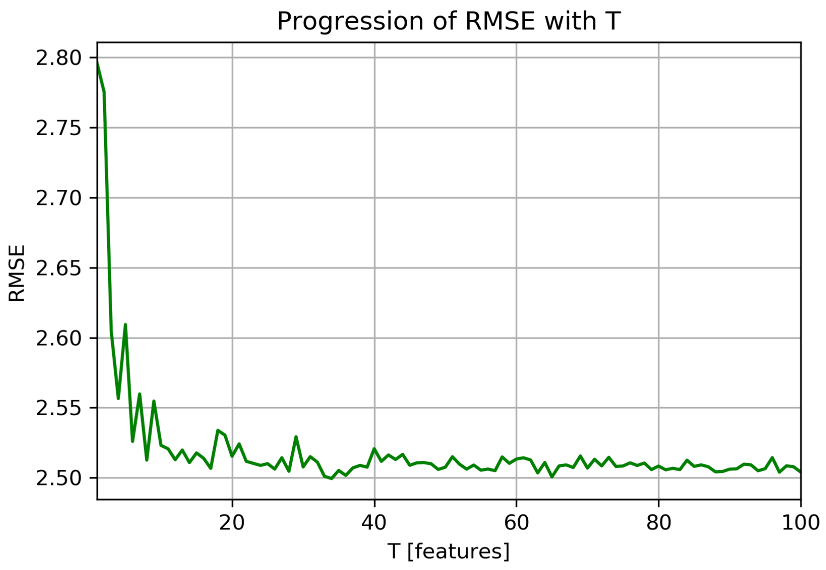


Figure 15

The model was trained for values of T from 1 to 100, and for each T a golden-section search was performed to find the optimum regularization coefficient. For this specific coefficient calculated for each T, the RMSE was measured. The progression of the root mean squared error measured with help of the k-fold cross-validation in the validation set can be seen in figure 15.

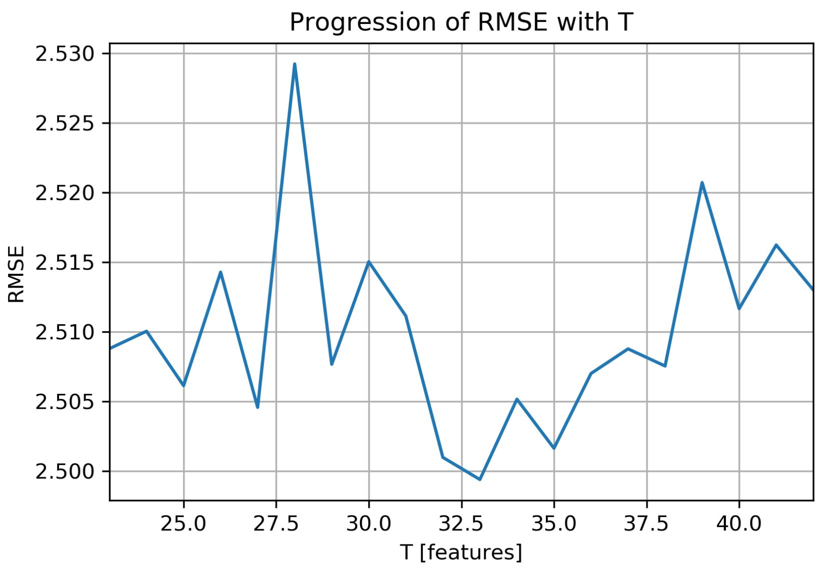


Figure 16

We can see a high value for the error for small T. After T equal to 10, the error seems to be more or less the same with a noisy added. In fact, the model was trained several times, and each time, the T for the smallest error changed. However, the range of this optimum T, was always in the same range: from 15 to 35. Also, the shape of the RMSE curve was also always the same: high error for small T, static with some noisy for T bigger than 20.
For this specific case of training, we can zoom the previous figure and check that the T for the minimum error was 33. The fact of the optimum T keep changing for each model training is that for each training, the matrix wk is generated randomly and this randomness can interfere in the model. The curve for the RMSE zoomed can be seen in the figure 16.

2.2.2 b) The best regularization coefficient for each value of T.

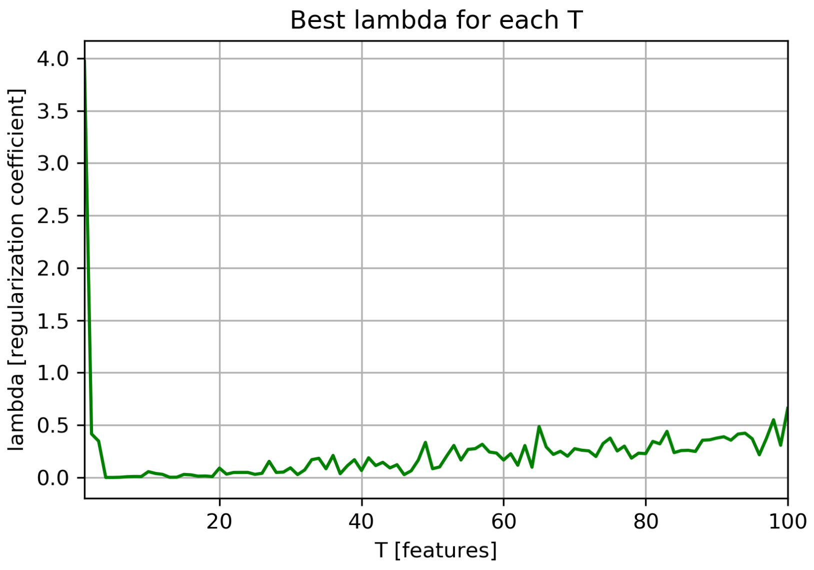


Figure 17

After having trained the model for each value of T, the optimum coefficient was saved to plot the next curve. The figure 17 shows the best lambda coefficient for each value of T. The regularization coefficient was found using a one-dimensional golden-section search method, searching in an interval of 1e-10 to 1e3 with a precision of 1e-12.

We can see that, with except of the very early values of T, when T increase, the optimum regularization coefficient, lambda, also increases. This means that when T increase, the model tends to penalty more and more the norm of weights array in order to try to reduce overfitting. In fact, for small values of T, lambda tends even to be zero, indicating that we are in a underfitting situation, or that trying to reduce the norm of the weights array is not so important.

2.2.3 b) Plot the predicted and real temperatures of test set considering the best T and its corresponding regularization coefficient.

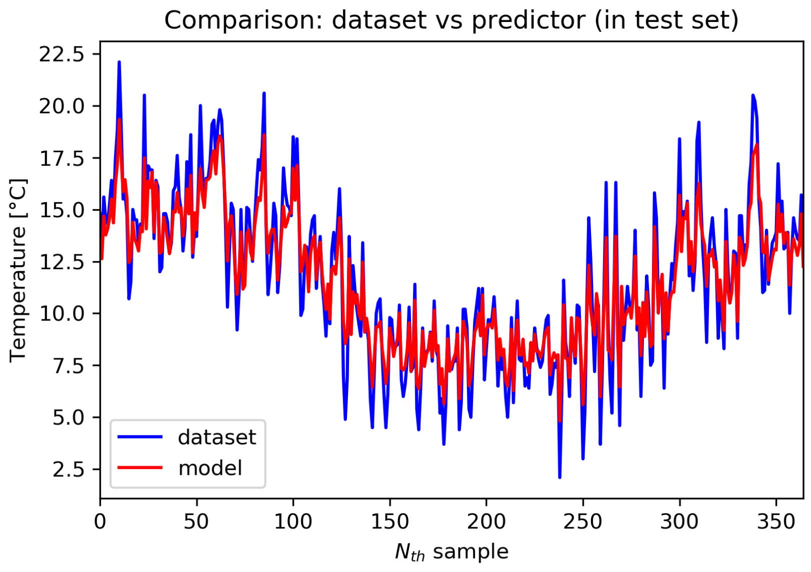


Figure 18

Now that we have the optimum T and also its optimum lambda, the model was trained with all data (i.e. not using cross-validation). The predicted minimum temperature and the real minimum temperature for the year of 1990 (test set) are plotted in the figure 18.

Despite the fact that each model training generates a new random matrix wk and this changes the best T chosen for that training, the RMSE calculated over the test year of 1990 does not change of magnitude. Several trainings were done, and the error always stayed in the range of 1.3 and 1.5. For the previous image, the RMSE calculated was 1.393, indicating that when a non-linear function is used, even though with random values for the wk matrix, a better result can be obtained from the linear regression of exercise 1.