Report
INF8245E (Fall 2021) : Machine Learning - Assignment #1
Amine EL AMERI - Matricule: 2164634

1 Sampling

1- We're only allowed to sample from a uniform distribution over [0, 1], and we have 4 activities, so we're going to divide [0, 1] into 4 intervals whose size depends on the probability of each activity. We'll have [0, 0.2] [0.2, 0.6] [0.6, 0.7] [0.7, 1]. Then we'll sample from U([0.1]), if the random number is in the n-th interval (n in {1, 4}, we choose the n-th activity.

Pseudocode:

```
randomNumber = sample(Uniform([0, 1]))

if randomNumber in [0, 0.2[
    return Movies
else if randomNumber in [0.2, 0.6[
    return INF8245E
else if randomNumber in [0.6, 0.7[
    return Playing
else if randomNumber in [0.7, 1]
    return Studying
```

2- After implementing this algorithm:

For 100 days:

{'frac spent in Movies': **0.23**,
'frac spent in INF8245E': **0.31**,
'frac spent in Playing': **0.1**,
'frac spent in Studying': **0.36**}

For 1000 days:

{'frac spent in Movies': **0.203**,
'frac spent in INF8245E': **0.41**,
'frac spent in Playing': **0.099**,
'frac spent in Studying': **0.288**}

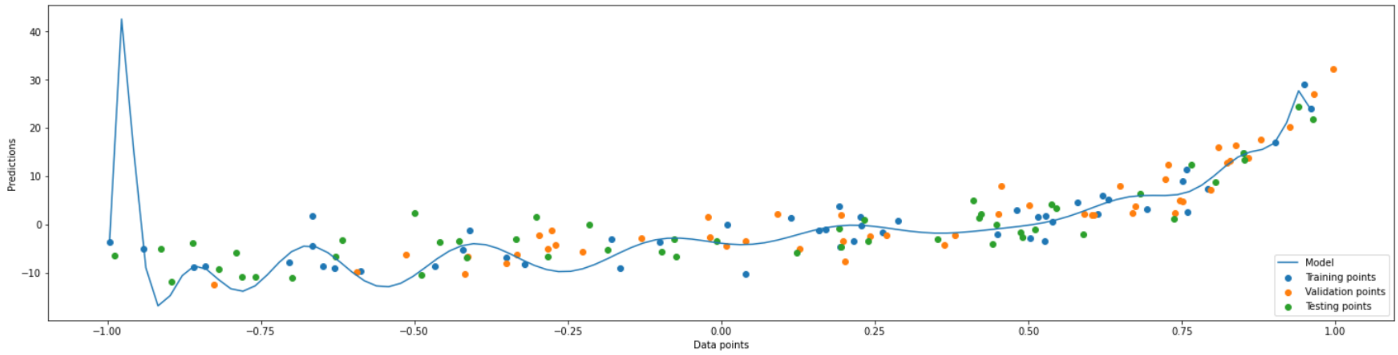
We observe that these fractions are very close to the multinomial distribution, especially those for 1000 days.

2 Model Selection

1-

a) The training RMSE is 2.544695, while the validation RMSE is 37.666521

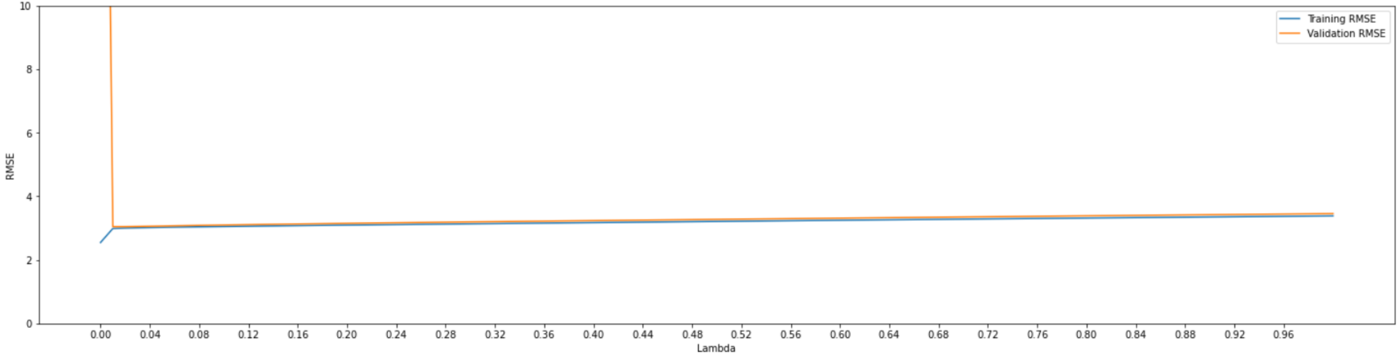
b) Visualization of the fit



c) The model is clearly overfitting, and that's because 20 degree polynomial is more than what is needed to model the data.

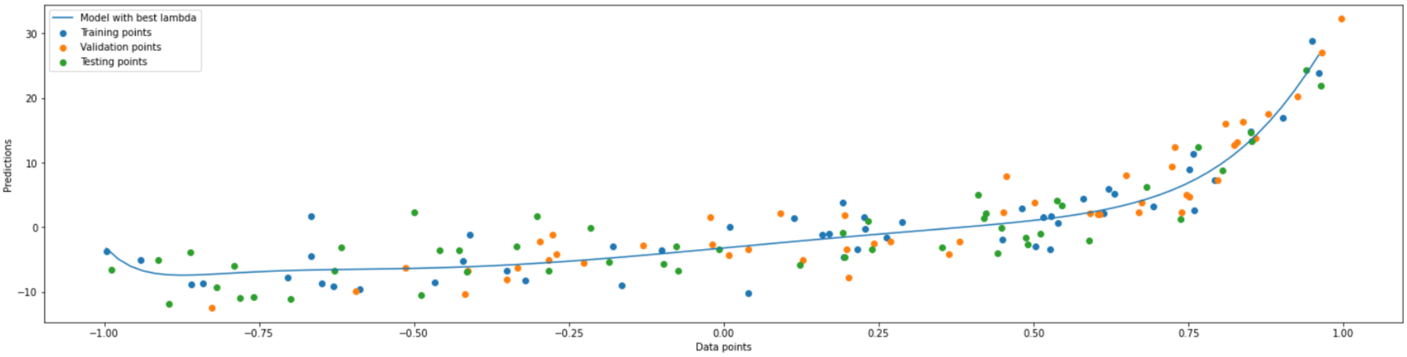
2-

a) For different values of λ, plotting the training and validation RMSE



b) The best value of lambda (in the plot it's the one that minimizes the validation RMSE) is *λ=0.01, and for this λ, the testing RMSE is 3.304821*

c) Visualization of the fit for the model with *λ=0.01*



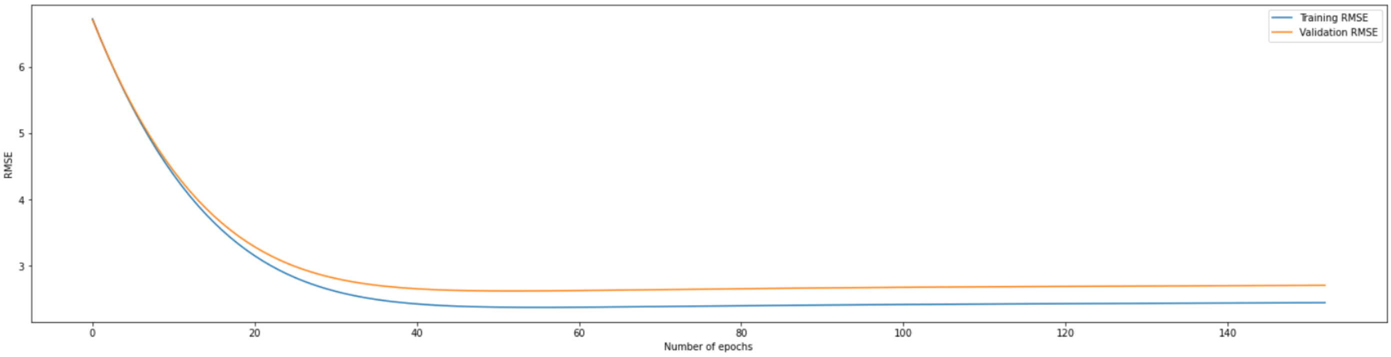
d) The model with the best lambda tends to be a little bit underfitting in regards of the testing data points, but is in general fitting the training and validation data very well.

3- When looking at the shape of model obtained with the best lambda, I think the degree of the source polynomial is 2, because we can have a very close looking model with 2nd degree polynomial.

3 Gradient Descent for Regression

1-

a) Using stochastic gradient descent and a step size of 10^{-4}. The visualization of the training and validation RMSE against the number of epochs.



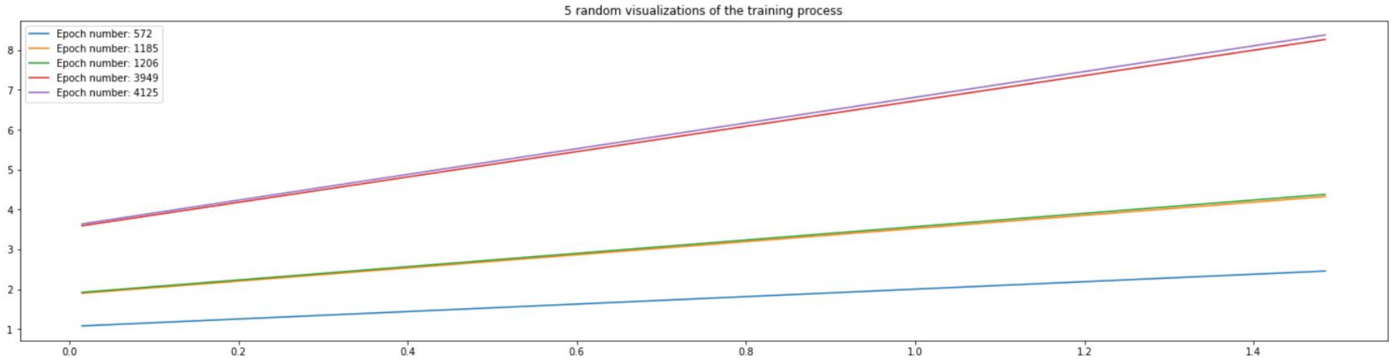
2-

a)

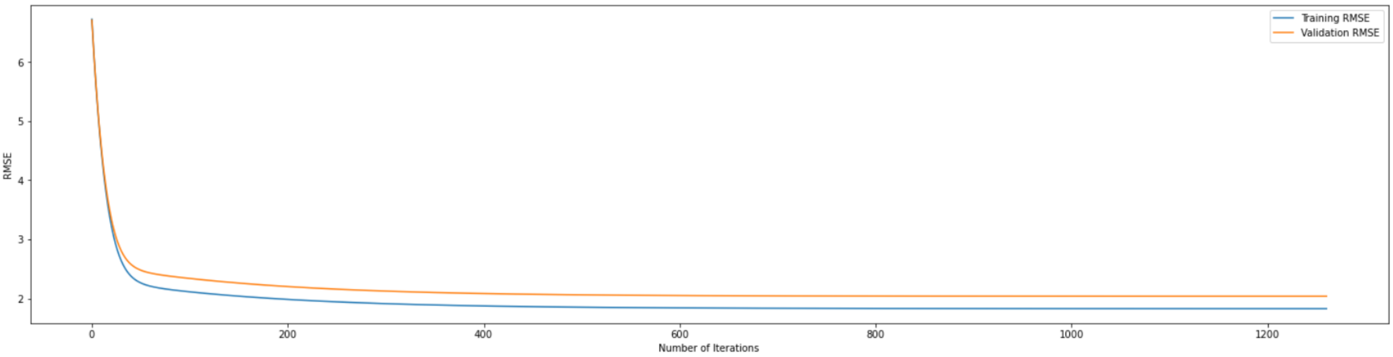
Step-sizes	0.01	0.001	0.0001	1e-05	1e-06	1e-07
Validation RMSE	2.8376	2.8300	2.6925	2.6195	2.6147	5.6178

b) Thus the best step size is 1e-06.
And the test RMSE of the chosen model with the best step-size is: 2.4727

3- 5 different visualizations chosen at random to illustrate how the regression fit evolves during the training process:



4- Repeating part 1 using full-batch gradient descent and a step size of 10^{-4}. The visualization of the training and validation RMSE against the number of epochs.



5- Based on the plots of the questions 1-a and 4 we can see that the full batch gradient descent converges faster (in terms of the number of epochs) than the stochastic gradient descent, and also converges to a smaller RMSE.

4 Real life dataset

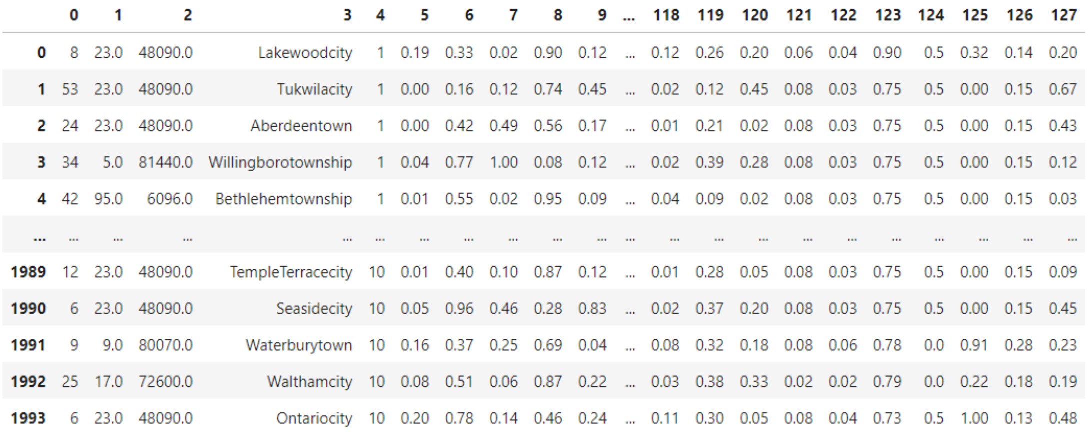
1-

a) Filling the missing attributes with the mean is not a good choice because the mean is not robust to outliers (very big numbers for example), and also because it can have no meaning if the numerical data is discrete (for example if our data is a list of country calling codes).

b) We can use the median of each column instead of the mean to fill the missing attributes.

c) The median is better than the mean because it is robust to outliers like very big and very small numbers.

d)



	0	1	2	3	4	5	6	7	8	9	...	118	119	120	121	122	123	124	125	126	127
0	8	23.0	48090.0	Lakewoodcity	1	0.19	0.33	0.02	0.90	0.12	...	0.12	0.26	0.20	0.06	0.04	0.90	0.5	0.32	0.14	0.20
1	53	23.0	48090.0	Tukwilacity	1	0.00	0.16	0.12	0.74	0.45	...	0.02	0.12	0.45	0.08	0.03	0.75	0.5	0.00	0.15	0.67
2	24	23.0	48090.0	Aberdeentown	1	0.00	0.42	0.49	0.56	0.17	...	0.01	0.21	0.02	0.08	0.03	0.75	0.5	0.00	0.15	0.43
3	34	5.0	81440.0	Willingborotownship	1	0.04	0.77	1.00	0.08	0.12	...	0.02	0.39	0.28	0.08	0.03	0.75	0.5	0.00	0.15	0.12
4	42	95.0	6096.0	Bethlehemtownship	1	0.01	0.55	0.02	0.95	0.09	...	0.04	0.09	0.02	0.08	0.03	0.75	0.5	0.00	0.15	0.03
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1989	12	23.0	48090.0	TempleTerracecity	10	0.01	0.40	0.10	0.87	0.12	...	0.01	0.28	0.05	0.08	0.03	0.75	0.5	0.00	0.15	0.09
1990	6	23.0	48090.0	Seasidecity	10	0.05	0.96	0.46	0.28	0.83	...	0.02	0.37	0.20	0.08	0.03	0.75	0.5	0.00	0.15	0.45
1991	9	9.0	80070.0	Waterburytown	10	0.16	0.37	0.25	0.69	0.04	...	0.08	0.32	0.18	0.08	0.06	0.78	0.0	0.91	0.28	0.23
1992	25	17.0	72600.0	Walthamcity	10	0.08	0.51	0.06	0.87	0.22	...	0.03	0.38	0.33	0.02	0.02	0.79	0.0	0.22	0.18	0.19
1993	6	23.0	48090.0	Ontariocity	10	0.20	0.78	0.14	0.46	0.24	...	0.11	0.30	0.05	0.08	0.04	0.73	0.5	1.00	0.13	0.48

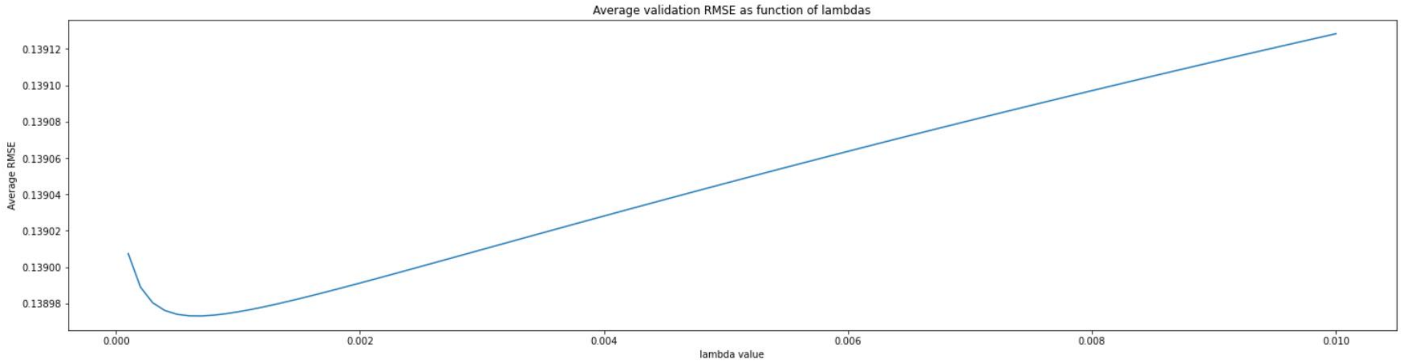
2-

a) 5-fold cross-validation average RMSE: 7.1628

b) Test RMSEs for each of the 5-fold cross-validation : [0.1398, 0.1410, 0.1411, 34.9351, 0.1412]
Thus the average test RMSE: 7.0997

3-

a) Plot of the average RMSE using 5-fold cross validation for various values of λ



If λ = 0 then it's like if there is no regularization at all, so we'll explore values > 0, and we know that lambda is the penalty term that tries to force some coefficients to be close to 0 so that there is no overfitting, but because of this, if we take a big lambda then we'll end up forcing too many coefficients to be close to zero and so our model will be underfitting. I started by trying numbers in the uniform interval [0, 20], and found that once λ > 1 the RMSE only becomes larger, so changed to [0, 1] and same thing happened once λ > 0.1, so after some iterations found the interval [0.0001, 0.01] and this time the RMSE decreased to arrive at its minimum at λ = 0.0006 then started increasing.

b) λ = 0.0006 gives the best fit

c) Test RMSE using λ = 0.0006 is : 0.14106

d) We can take the parameters learned from the training with regularization, and see which ones are the smallest (the closest to zero), these are the ones that has been reduced due to the regularization because they correspond to features that are not very important in the prediction task, and so we can drop these features.

e) Test RMSE of the best fit (λ = 0.0006 was still the best value) with reduced number of features: 0.14233

f) The test RMSE of the model with all features is 0.14106 while for the model with reduced number of features it is 0.14233, and this even though we have retained only 62 from the 127 features. So we can say that even if the RMSE with reduced feature is slightly bigger than the one with all features, the temporal and spatial complexity that has been reduced is worth it, and also that a lot of features didn't have much importance in our prediction task.