Challenge B - Solution key

Oscar Jara - Nicolas Martinez

08/12/2017

The route of the repo is the following: https://github.com/oscarjararamirez/Challenge-B---Oscar-Jara-Nicolas-Martinez

Task 1B - Predicting house prices in Ames, Iowa (continued)

Step 1: The Random Forest algorythm works creating a large collection of correlated decision trees. Is a method that increases classification accurary with unbiassed and less noisy models to create a model with low variance.The term "forest" depicts a situation where a lot of decision trees are used. Particularly, the Random Forest method takes a training sample from which it creates subsamples. From them, this tool also generates random subsamples to produce decision trees. This decision trees represent variations of the the main clasification. With them it is possible to get a ranking of clasifiers. With these ones, it is possible to make predictions.

Step 2: We train the Random Forest model in the training data. First, the data is corrected for the presence of missing data and factor variables. The variables that had a greater significance in the last project are used to asses.

Step 3: Using the model employed in the step 2, we make predictions for the test set (found in test.csv). We export the predictions to the csv file "predictionsRFF.csv". We also run an OLS model and make a prediction of the sale price using its inputs. Comparing both results, we can observe that the one made with the Random Forest technique possess a better fit. The prediction made by the OLS model has a low level of variation in its sale prices, situation that is different from the one showed by the training data. The Random Forest model has a better result capturing this variance.

Task 2B - Overfitting in Machine Learning (continued)

Step 1: First we create the data using the same code employed in challenge A.

Then we proceed to run the local linear model on the training data.

Step 2: We run the high-flexibility local linear model.

Step 3: First we compute the predictions (fitted values) for both models.Then, we plot the different estimations, the true line, amd the scatter plot.

Step 4: We compute the variance for each prediction and compare to each other.

The model with the lowest variance is the low-flexibility local linear model (with a variance equal to 2.3272 versus a variance of 7.09).

To compare the bias of the models, we first compute the absolute value of the bias for each prediction. Then we sum those values for each model. This way we have a way to quantify which model has the biggest bias.

After computing those values in R, we find that the model with the lowest bias is the high-flexibility local linear model.

Step 5: First we proceed to run the models in the test data and to plot them.

We proceed to compare the variance of the two models using the test data. Then, we compare the variance of each type of model under the different datasets.

Comparing the variance of the predictions in the different models, we see that, under both datasets, the low-flexibility local linear model has a lower variance than the high-flexibility local linear model.

We then compare each type of model predictions under the different datasets. We find that for both models, the predictions have a higher variance for the models estimated using the training data. This could be related to the size of the sample, as the test data has less observations than the training one.

On the test data, the high-flexibility model is the one with the highest bias. Comparing estimations witht the two different datasets we find that the models have a higher bias in the training sample.

Step 6: The step is performed and commented in the code.

Step 7 and 8: The steps are performed and commented in the code.

Step 9: We create a vector that stores the information for all the regressions done on the training data. Then we use that vector to compute the predictions of each one of those regressions over the test data. Finally, we use mutate and sappply to create a vector with the MSE for all bandwidth values, for the test database.

Step 10: The graph created summarizes the relationship between the MSE and the bandwidth under both datasets (in orange the test data and in blue the training data). The MSE is an increasing function of the bandwidth for the training data. Under the test data, the MSE reaches a minimum in a bandwidth value of 0.25.

Task 3B - Privacy regulation compliance in France

Step 1: We import the data of companies or organization in France wishing to adopt the regulatory framework of the CNIL. This data is directly imported from the Open Data Portal link page.We employ the command "fread".

Step 2: A table that contains the number of companies/organizations for each department in france is made with the kable and aggregate command. We use a unique count technique because in the CNIL database there firms with more than CIL offircer, so they are in more than two rows.

Step 3: Once downloaded the CNIL dataset, we move forward to the merge process. In the firstplace, we obtain the unique Siren codes from this data set.Then, we import the data from the CNIL and create a loop to extract some portions. We delimit this procedure for some parts of the Siren database. FOr instance, we show the results for the 10% of the data, in order to fully run this entire file.

```
##    user  system elapsed
## 2792.61   95.36 3218.10
```

Step 4: We create a histogram of the size of the firms. We use the number of employees (variable EFENCENT) as a proxy of the size of the organization.

Anexes

Table from Tas 2B - step3

```
fig1
```

Figure 1: Step 3 − Predictions of ll.fit.lowflex and ll.fit.highflex on training data

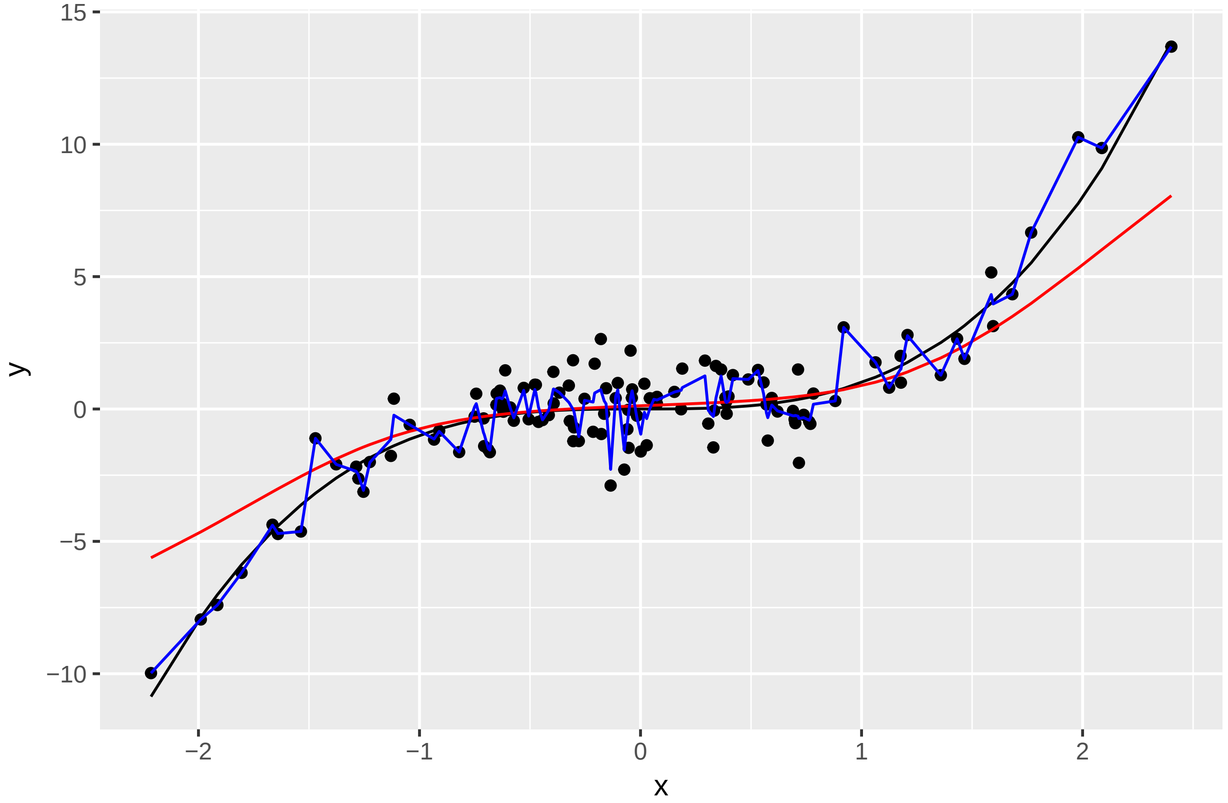


Table from Tas 2B - step5

```
fig2
```

Figure 2: Step 5 − Predictions of ll.fit.lowflex2 and ll.fit.highflex2 on testing dat

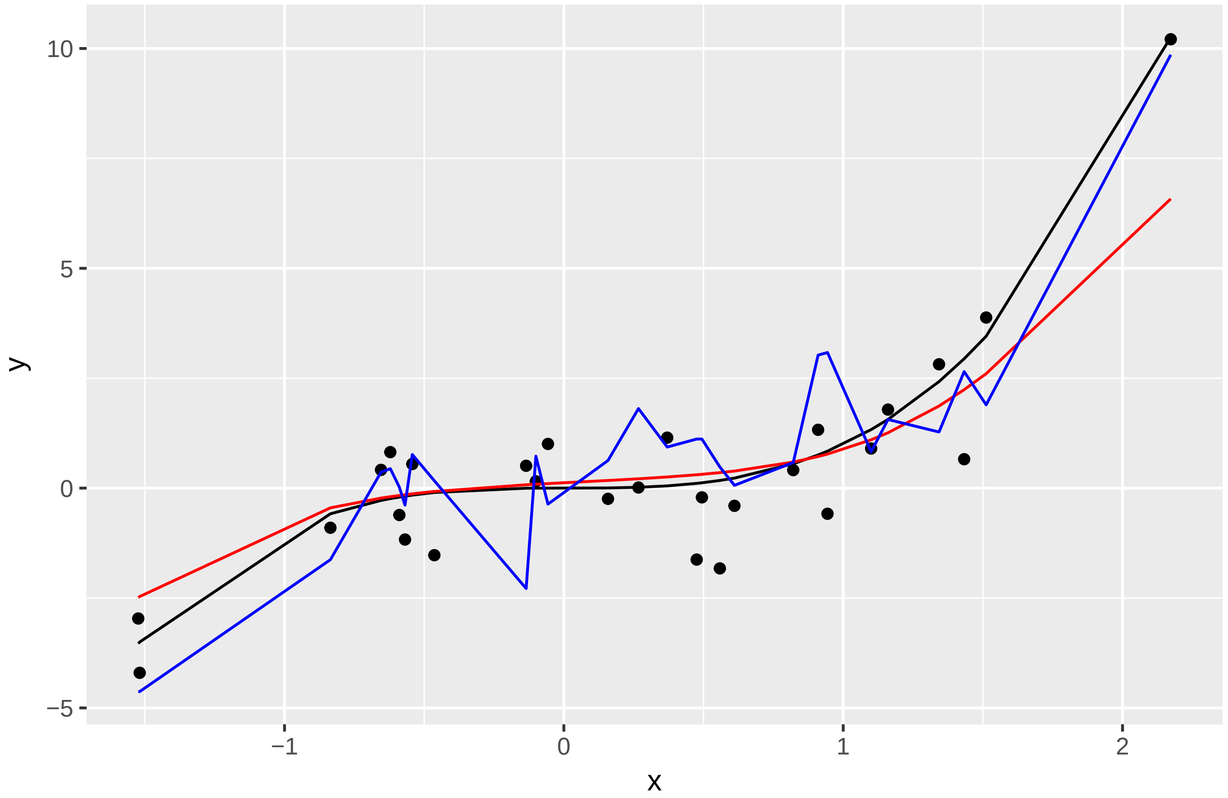


Table from Tas 2B - Step10

```
fig3
```

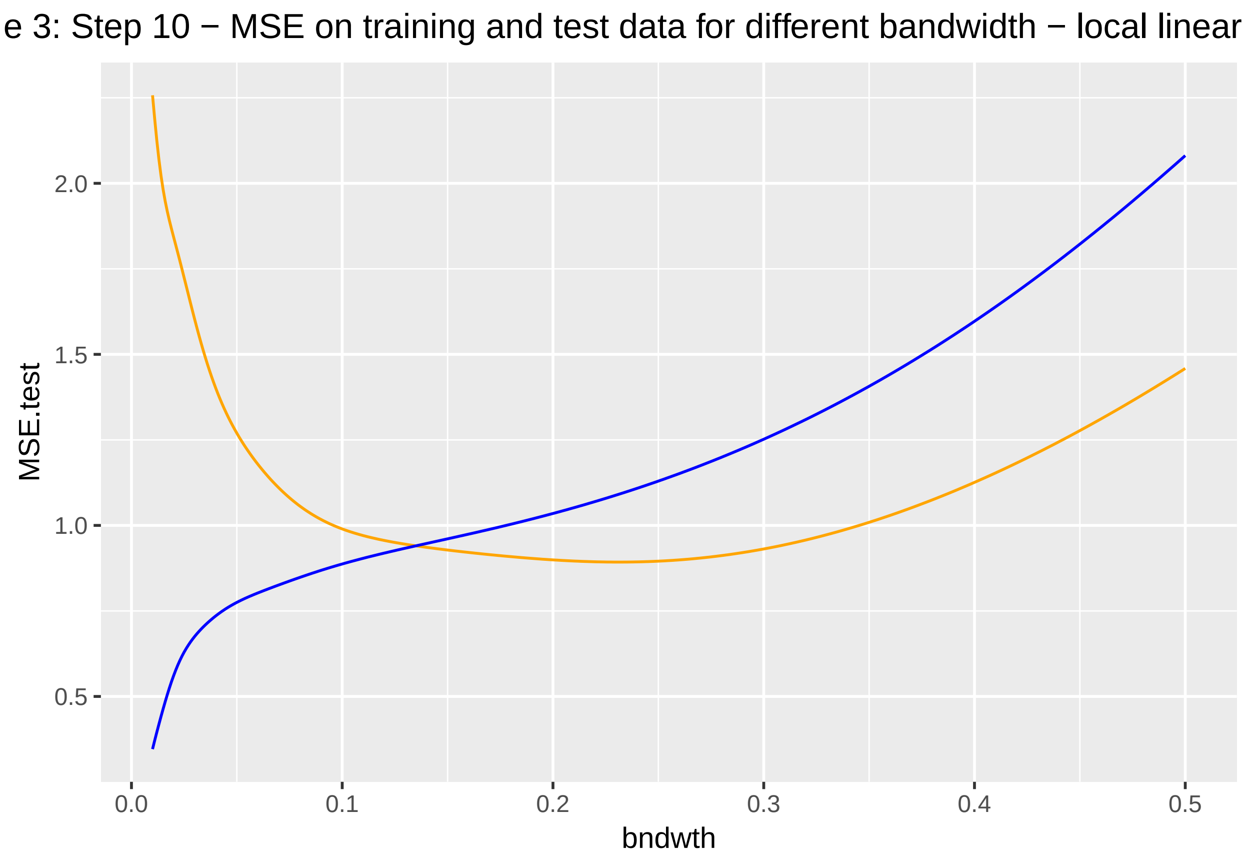


Table from Tas 3B - Step2

```
table3.2
```

Table 1: number of organizations that has nominated a CNIL per department

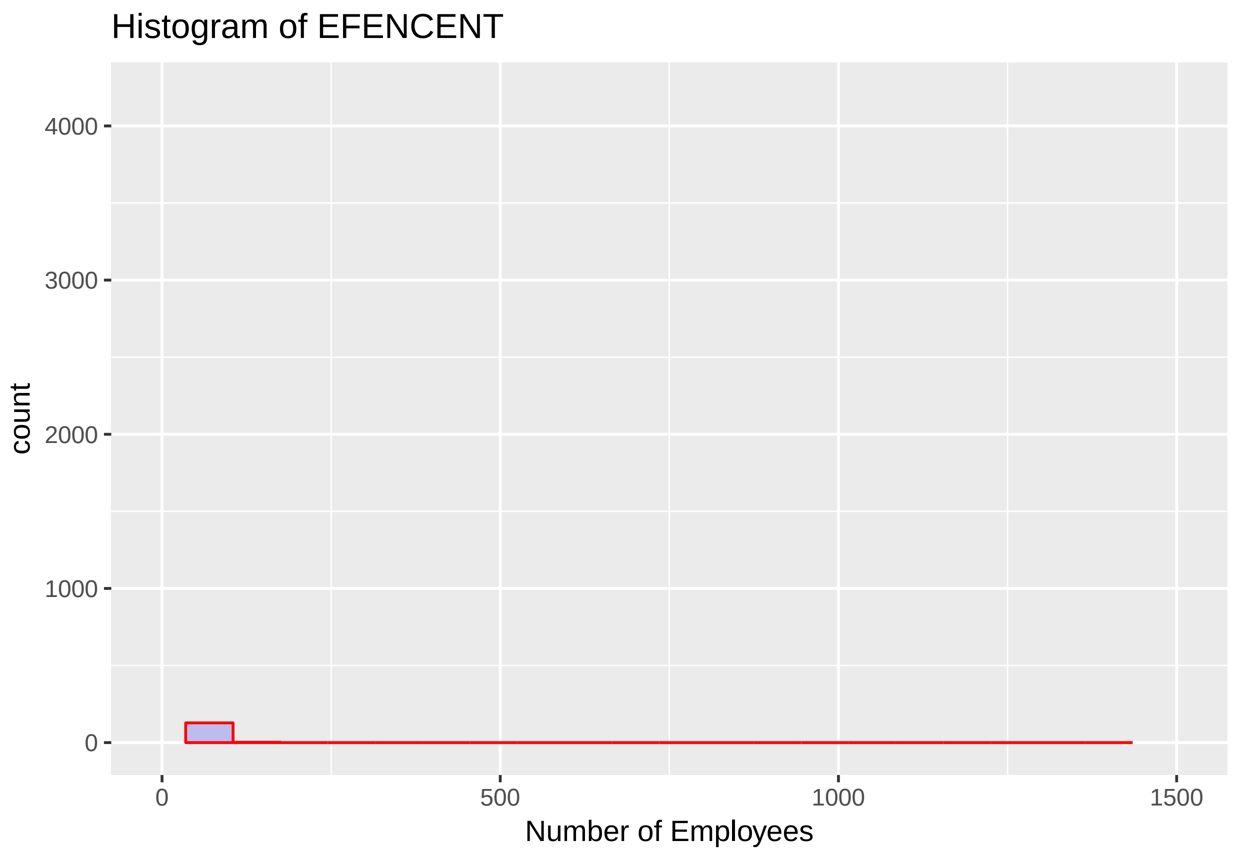
Department	Number of firms
	56
.	1
01	130
02	104
03	68
04	68
05	52
06	253
07	59
08	82
09	18
10	103
11	92
12	83
13	436
14	250
15	30
16	120
17	147

Department	Number of firms
18	78
19	46
20	91
21	143
22	111
23	31
24	79
25	142
26	130
27	109
28	95
29	173
30	132
31	298
32	80
33	358
34	271
35	271
36	52
37	177
38	406
39	66
40	176
41	95
42	210
43	96
44	335
45	176
46	57
47	102
48	11
49	203
50	131
51	167
52	50
53	314
54	189
55	64
56	176
57	238
58	44
59	519
60	207
61	73
62	216
63	138
64	157
65	66
66	108
67	266
68	164
69	572

Department	Number of firms
70	69
71	122
72	128
73	101
74	183
75	1969
76	284
77	222
78	278
79	131
80	153
81	115
82	61
83	195
84	129
85	195
86	152
87	109
88	122
89	90
90	22
91	216
92	918
93	304
94	288
95	174
97	240
98	25
BP	2
CE	1
CS	1
EC	1
F3	1
LI	1
LU	1
PA	2
W1	1
WC	1

Histogram from Task 3B - Step4

```
hist3.4
```


Histogram of EFENCENT
count
4000
3000
2000
1000
0
0
500
1000
1500
Number of Employees