Code in outlier detection part

MLND: Unsupervised Learning
Project: Creating Customer Segments

---

sergei3000 2016-08-14 15:03:48 UTC #1

I need help with understanding this part of the code:
`display(log_data[~((log_data[feature] >= Q1 - step) & (log_data[feature] <= Q3 + step))])`
I understand everything until I get to this operator: ~.
In Python ~True returns -2 and ~False returns -1.
How does all this result in the outliers being printed?

---

ayush_jain 2016-08-14 16:09:02 UTC #2

Hello Sergei,

The ~ operator is the binary ones complement operator. When used inside a list we can use it to extract values from the list where the given condition does not hold true. So for this dataset, the condition supplied is that the points in the log_data are in the IQR (values greater than Q1 and less than Q3). Adding the ~ operator returns all points that are not in this range. As we consider these to be outliers, we can obtain them by this method.

Hope this helps.

---

sergei3000 2016-08-14 17:50:28 UTC #3

> ayush_jain:
>
> When used inside a list we can use it to extract values from the list where the given condition does not hold true.

I'm not sure I understand what you mean by using this operator *inside* a list, but when I use it *with* a list giving python command ~[True,False,True] I get the following return:
Traceback (most recent call last):
File "", line 1, in
~[True,False,True]
TypeError: bad operand type for unary ~: 'list'

---

ayush_jain 2016-08-14 18:00:49 UTC #4

Inside the list as in using the operator with the values in the index. Something like `list[~ (*condition*)]` gives you elements in the list that are false for the condition used in the index.

> sergei3000:
>
> TypeError: bad operand type for unary ~: 'list'

You cannot find the complement of a list and hence you get that error

---

sergei3000 2016-08-14 18:07:44 UTC #5

@ayush_jain
Could you give some very simple example for list[~ (*condition*)]? Can we really do this with lists or is this operation only used with pandas series?

---

ayush_jain 2016-08-14 18:28:24 UTC #6

A good example is the one you have provided.

> sergei3000:
>
> display(log_data[~((log_data[feature] >= Q1 - step) & (log_data[feature] <= Q3 + step))])

this is equiv to display(log_data[~(data points that are in IQR)]) and as I explained previously, it would return all the values in the series `log_data` that do not contain data points that are in IQR. Hence, those points which follow the inverse of the condition.

I have only seen this operator used with pandas series. I can't think of a way to apply this in regular lists and I doubt there is one.

---

jean_marc_1204616771 2016-08-16 20:53:11 UTC #7

The following also works (if you really!! want to get rid off the operator "~"):

```
display(log_data[((log_data[feature] < Q1 - step) | (log_data[feature] > Q3 + step))])
```

It finds the elements that have values smaller than (Q1-step) **or** larger than (Q3+step)
