You got a comment on your Github project

EPFL Extension School <no-reply@extensionschool.ch>

Mar 23.07.2019 23:47

À : dujaj@hotmail.com <dujaj@hotmail.com>

Dear John,

Michael Notter just commented on your Github project.

Michael Notter (commented 10 minutes ago)

Hello,

Thank you for your project solution. You completed all the tasks and did excellent work. Well done!

The following are some of my thoughts, notes and answers that I wrote out while reviewing your project:

Warm up

- You did all three tasks perfectly. Exactly what we expected. I especially liked that you first always thoroughly explored your dataset. Well done! This will be of much use for later projects.
- I saw that you sometimes write multiple lines of codes on one single line, separated by `;`. While the code is correct and executable, it is not recommended. Reading such code is rather difficult and it goes against the suggest code style of python. Try out `import this` for a funny easter egg.

House Prices

- Oh wow! I'm very impressed by all the different visualizations, color scheems etc.

Very well done! (I wrote this comment before going through the notebook or seeing your last request). So, here come the rest of my comments.

- You chose many good visualization approaches. Instead of just plotting the column names, you decide to put them in a 10 by X table, which makes inspection very easy.
- To your question: No, there is no rule of thumb to know the ratio of NaN to drop a feature. It always depends on the dataset and the problem and questions at hand.
- In the `violinplot` during data exploration, certain years have only a single value for `Grvl`. This might be an indication of an issue. Also, if you observe such peculiarities, I recommend to investigate or comment on them.
- The `Check_df()` function is a very nice idea. Well done! You could even add an "outlier" column. For example, you could indicate how many datapoints are above 3 standard deviation, 4 STDEV or 5? Side note: function names should start with a lower case letter, and class names with an upper case letter.
- Your question was: Could you please give an example of code to avoid copy past several time the same filling? - What about using a for loop?

```
for e in ['Bsmt Qual', 'Bsmt Cond', 'Bsmt Exposure', 'BsmtFin Type 1',
          'BsmtFin Type 2', 'Fireplace Qu', 'Garage Type',
          'Garage Finish', 'Garage Qual', 'Garage Cond', 'Mas Vnr Type' ]:
    df_Train[e].fillna(value='NA', inplace=True)
```

- Your question: How to modify that line of code to show only the values that aren't 0 ? - The easiest way would to save the output as a pandas series and then drop non-zero entries.
- About feature engineering: I agree, this is not always easy. But that's where expert and domain knowledge comes into play and where a data scientist can shine as well. I haven't looked into TPOT or Auto-Sklearn, they might be useful. But any automated approach, while perhaps convenient, has also its flaws.
- The density plots under `df_Train.plot(kind='density', subplots=True, layout=(24,3), figsize=(17,75), sharex=False)` are very nice to look at but they make it difficult to detect outliers. I would rather recommend histogram bins, where it is clearer that x-min and x-max represent values that are contained in the dataset, and not just represent the tails of a distribution as for density plots.
- There's no robust or right way to specify a skewness threshold. It always depends on your dataset and value distribution.
- If you get a `RuntimeWarning` because of the log function and zero values. You can either add +1 to all values before transformation or directly use numpy's `np.log1p` function that does exactly that. Keep in mind that this doesn't work if the feature contains negative values or nans
- For the conversion of ordinal variables, yes `OrdinalEncoder` could be a good approach. But for some features a sequential encoding might not be straight forward and you might want to map different values to the same number. In this case a manual approach is probably better.
- There's not really a best way to removing outliers. Z-thresholding is a quick approach and might lead to good enough results as it can easily detect extreme outliers (for example with a threshold above 4 or 5). But then again, doing it manually might make more sense, for certain features where a visual inspection reveals clear cutoffs.
- Yes, it is a valuable approach to remove outliers first and then split the training and test set.
- Your ridge regression curve for the intermediate model seems to be correct. It looks perhaps so smooth because the graph plots up to 10^7, i.e. you loos a lot of detail of the curve around the best alpha because of the high range in y direction. Removing

the y-limit and only exploring till 10^2 helps (a bit) to see the minima.

- And yes, your test curve reaches a minima of RMSLE = 0.0038. If you remove the y-limit and explore alpha in a between 10^-2 and 10^+2, you can see this directly.
- Overall, I recommend to restrict your grid search to a smaller range. Having an alpha range from 10^-20 to 10^+20 for the complex model, doesn't make sense. Covering 40 magnitudes of values is not really possible. The left tail of the test curve suddenly jumps up, which means that we've reached a clear overfitting situation. Also, in this low area we often see spikes which are due to numerical issues on the machine. All in all, try to focus on a few (perhaps max 10 magnitudes of values) and fine tune the parameters there.

Your results for the house prices task were exactly what we expected. You reached very good scores on the test set, very well done! Also your accuracy on the prediction set was very good, with an RMSLE of 0.0547 and an MAE of 14,797. Congratulations!

Final comments & questions

- Doing this all next to working 100% and a family is impressive. And even if this should take you longer than somebody who has more free time, the end result clearly shows that you understood the methods and developed a deep understanding of the thematic. Very well done!
- I've tried to answer almost all open questions. Should I have missed some important ones, don't hesitate to point this out and ask again.
- I can understand that its sometimes difficult to see a clear logical order for when to use which approach. And to be honest, often times there's no clear answer to that. There's never a one-for-all solution, sometimes it makes sense to remove extreme values, sometimes its better to cut them down to a value at 95% and sometimes it might be important to leave them in. The same is true for almost everything else. But this shouldn't make this endeavour desperate but more exciting. More important than knowing that step A is usually followed by step B is that you understand why you replace certain values, or what the advantage is of feature engineering. Knowing the reason behind those steps allows you to decide yourself what to do. And this is the most important lesson in this domain. Rarely is there a clear right or wrong. What is important is that you document all your observations, investigate strange things and justifies any decisions.
- Yes, concatenating the two datasets and adding an additional column with 'pred' and 'train' is a very good idea. Just keep an eye on the fact that any estimations of parameters should always be done on 'train' and that 'pred' should stay as untouched (i.e. unbiased) as possible.
- I should probably take a closer look at tpot. If you want, you can always try to use those approaches that you want to learn more about in your final capstone project.
- Concerning your final request: You really did great work and show an impressive skill set! Very well done! It is clear to us that you follow the course in much details and your project shows that you incorporated everything!
- About finding a pythonista group to get your hands dirty on python, I unfortunately don't know of any myself. Depending on where you live, perhaps look into meetup.com. Alternatively, it is always a great experience to work on your own project, or find a github project of somebody else and help them with their project. But I acknowledge that this doesn't help with meeting other enthusiasts.

So again, very well done! You can now proceed with the next course. Don't hesitate to book a 1-1 video call with me if you have any questions about the course or want to discuss the project results in more details. You can do it directly via this link:
https://calendly.com/s/g7OBj04s

https://calendly.com/s/q7OBl04s

Kind regards,
The EPFL Extension School team