DS420, Fall 2022
Assignment 4: Transformation Pipeline

Posted date: Wed, Oct 26; Due date: Wed, Nov 3, 11:00 PM

Submission instructions:

- Please submit your solution in 1 notebook to Katie. You can submit your code in .py file as well.
- In your code, please include the following lines at the beginning of the file
  # Student Name:
  # Collaborate with (if any):

Problem

We have learnt about creating a ML project with 5 steps:

[1] get data
[2] create train, dev, and test sets
[3] prepare data for training
[4] train models
[5] evaluate the final model

We have already experienced all the above steps (see assignment 2, 3, and midterm exam). The purpose of this assignment is to go deeper in **step #3 with a systematic way**. Doing data preparation step by step in the midterm practice notebook helps you to easily understand the process. Now let's try to wrap every step there in a nicer way.

What you need to do is to prepare the California housing dataset for a linear regression algorithm. Your preparation needs to include the following steps:
[1] (10 points) Handle missing values for both numerical and categorical features
[2] (30 points) Remove outliers
[3] (10 points) Scaling all numerical features
[4] (10 points) Convert categorical features into one-hot-vector features
[5 bonus part] (10 points) add three ratio features to train and test sets: bedrooms_ratio, rooms_per_house and people_per_house. Here,
bedrooms_ratio = total_bedrooms/total_rooms ,
rooms_per_house = total_rooms/households, and
people_per_house = population / households
The score of the assignment is 100 points: 60 points for the above steps and 40 points for making your preparation part work smoothly with all other steps in ML pipelines (5 steps of ML projects).

How to do it

You should start with the note pipeline.train_test_transforming_separately_Californiahousing. You already have the frame of 5 steps of the ML project and the transformation pipeline there. Now you need to add the outlier remover to the pipeline.

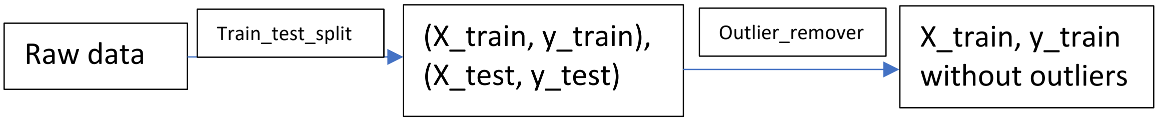
You can add the remover at the beginning of the pipeline or any position you think it works for you. I recommend you add it at the beginning. You should be aware of some difficulties:

1. Remove outliers of the train set. Do NOT remove outliers of the test set.
2. Recall: when you want to add a transformer into a transformation pipeline, you need to make sure the transformer has 2 methods: fit() and transform(). Actually, the fit() method will train a "model" using X_train and y_train. The transform() method then takes the model and make predictions for X_train. Are you confused? Why we need to make predictions for X_train? Well, we are in a need of transforming X_train to something. For example, we need to scale X_train in the range of [0..1]. Here the predictions are exactly X_train transformed into other form.
   So the fit() function will take X_train and y_train as input parameters and transform() will take only X_train as an input.

   When you remove outliers, you need to remove them from X_train and y_train. You need the transform() function to take X_train and y_train as inputs and return new X_train and new y_train (the new ones are the original one minus outliers). Unfortunately, the transform() method of all built-in transformers takes only one input which is X_train. So, you have to code a transformer for the remover yourself.

   You can get a help from the textbook, chapter 2, section "Custom Transformer".

3. If you cannot add the remover to the pipeline, you can remove outliers manually (see the code or removing outliers in the practice midterm notebook). This means you feed the pipeline X_train_ and y_train without outliers. The scheme looks like this



Then you feed X_train and y_train without outliers to the pipeline.