Data Scientist Test - 2022

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/598ad5be-95bf-459e-9491-109954a80c05/sample.csv

Demand Forecasting

We have divided the normal time series forecasting problem into different steps and provided some sample transportation volume data. Please follow the steps below to complete the tasks.

Please make sure your model is versatile enough to handle time-series data that has different mean/variance. (i.e. do not hardcode the hyper-parameters.). The expected programming language for the assignment is python.

Steps

Data Engineering (Mandatory)

In this step, please write a function which takes in the data at `sample.csv` and returns a standardised `pandas DataFrame` the prediction engine could use

Input specifications:

- The input contains two rows, first row contains the headers and the second row contains the values.
- Each of the column contains the total transportation volume of the week indicated by the header.
- For example, `201501` means the first week of 2015 and goes from `2014-12-29` till `2015-01-04`. Please refer to ISO 8601 Week Number for detailed definition.

Output specifications:

- a `pandas` dataframe of two columns `ds` and `y`.
- `ds` contains `datetime` objects indicating the last day (Sunday) of the corresponding week.
- `y` contains the corresponding total transportation volume of the week.

Seasonality and Trend Detection (Mandatory)

In this step, please write a function which takes in the output from the data engineering step and leverages the `Gaussian process` to extract seasonality and trend.

Input specifications:

- A positional input which takes in the output from the previous step.
- An optional input which is an integer indication how many data points to predict, defaults to `6`.

Output specifications:

- This function returns two `DataFrame` objects of the same format as the input. The first `DataFrame` object indicates the seasonality and trend detected. The second `DataFrame` object contains the residue of subtracting the seasonality and trend from the input.
- Both outputs would have `n` number of extra rows in the end where `n` is the same as the integer indicated by the optional input.
- For the first output, the `n` extra rows contains predicted seasonality and trend of the next `n` weeks .
- For the second output, the `ds` column should be filled for the `n` extra rows but the `y` column should be left as `numpy.nan`. They will be filled during the next step.

Requirements:

- The process of hyper-parameter tuning must be automated.
- Set the second parameter to `6`.

Anomaly Detection (optional)

Find if there are any anomalies in the data using **Extreme Studentized Deviate (ESD)** test.

Requirements:

- Choose an appropriate way to find anomalies using the test mentioned above, highlight anomalies in the plot. Can you find any reasoning/pattern behind the anomalies? Can you think of any other approach that would be more useful here?

Prediction (optional)

The final goal is the fill the `numpy.nan` slots of the residue with appropriate predicted values (6 weeks ahead prediction). You should choose the best method for residual predictions based on proper cross-validation. 1 week ahead prediction could see different performance than 6 weeks ahead predictions, you should take this into account.

Input specifications:

- A residue `DataFrame` object with the `y` column of the last `n` rows left as `numpy.nan` (`n=6` in this case)

Output specification:

- The residue `DataFrame` object with the `y` column of the last `n` rows filled with appropriate predicted values.

Resources to learn more about Gaussian Process:

- https://www.jgoertler.com/visual-exploration-gaussian-processes/#GaussianProcesses
- http://katbailey.github.io/post/gaussian-processes-for-dummies/
- https://nbviewer.jupyter.org/github/adamian/adamian.github.io/blob/master/talks/Brown2016.ipynb