Report

What is outlier

(I did not visit last lab session and don't know all the details)
- In my opinion outlier is a sample with outstanding weight. The outlier obtains too high value after all iterations, which means that it did not obtain correct classification side, algorithm penalizes more and more this sample without effect.
- I measure **outstanding** weight by its deviation from mean value of the sample.

Approach

What to do with nan?

1) Simpliest solution - drop all nan values. That leads to lose of ~40 countries from entire dataset, it is quite huge (~20%)
2) Another simple solution - assign 0 to nans. `df.fillna(0)` This solution affects mean & deviation values, don't like it.
3) My solution - set average values based on belonging to specific region. `Literacy (%)[Guernsey] = avg(Literacy (%), Europe)`

How to get sample weights?

In order to obtain sample weights I inherited from AdaBoostClassifier and store sample weights in additional container.
`self.sample_container = []`

How do I measure outlier?

1) Standartize the values of RV (last iteration's sample weights) `X~N(μ,σ2)`
2) Apply formula (W - μ) / σ *W - vector of weights*
3) Interpret outliers as values that are larger than two standard deviations, link to the article
- Outliers = { val ∈ W | val > (μ + kσ)} - A value of `k` is a hyperparameter, that should be chosen carefully

Output examples

This combination provides best improvement of accuracy. K=[2], ESTIMATORS=[25], algorithm='SAMME'

Score with outliers equals = [0.927]
Score w/o outliers equals = [0.984]

Found [9] outliers
- Albania

- Ireland
- Italy
- Japan
- Lesotho
- Rwanda
- San Marino
- Turkmenistan
- Uruguay

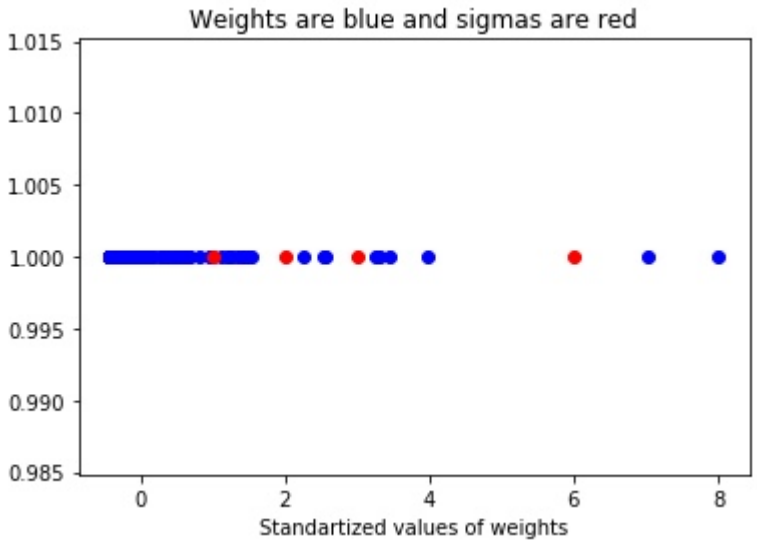
Not enough time for the model. K=[3], ESTIMATORS=[10], algorithm='SAMME'

Score with outliers equals = [0.913] - **as you can see here, the model did not have enough time to train**
Score w/o outliers equals = [0.940] - that leads to 3% improvement, but not the best

Found [6] outliers
- Ireland
- Japan
- Lesotho
- Rwanda
- San Marino
- Uruguay

Justification of the results:

This plot represents standartized sample weights (blue) and sigma values (red) [1, 2, 3, 6]
We can see the distribution and find sample outliers that lay below 3 or 6 sigmas.



Post Scriptum

- Test fraction equals to 0.3
- Only default method was used in AdaBoostClassifier *(Decision Tree)*
- With K=0 model (identification of outliers). The purified model proposes best results with 100% accuracy, but it is silly. Model loses ~50 countries. We will not treat it as a winner.
- Cases with K=6 identify astonishing outliers and increase the accuracy of the model, but only if #estimators is big enough (not equal to 10).
- It is also possible to measure results by altering learning rate, but we are analyzing **outliers**, not AdaBoost performance.