- We are interested in how you approach and break down the problem

● Make a clear choice for the type of model you use, and optimize for that specific type

● Clearly explain why and how you constructed your features

● Clearly explain what tuning/optimization you applied and why

● Supply us with some evaluation metrics you think are relevant to the performance of  your model and explain how we should interpret them

1. **Assumption for problem statement**

I made a binary classification problem. So, variables like Robot':'NHT', 'Robot Mobile':'NHT i.e bot traffic. Rest are non-bot. I was not sure what “special” was in classification so, I dropped it. My aim is to build an optimal model that can help us to identify bot traffic (NHT) and human traffic (HT). My focus is more towards bot traffic as negative bot traffic might create fuss for my company.

Bot=1, non-bot=0

1. **Choice of features/EDA**

Key features as were in train examples are considered.

* I did check epoch\_ms. It didn’t give any interesting outcome.
* My initial assumption was that “**visitor\_recognition\_type**” contains information about ANONYMOUS and they might relate to bot traffic. I was wrong in that as it was distributed for both classes in our target variable.
* Feature of interest=[labels','visitor\_recognition\_type','country\_by\_ip\_address','region\_by\_ip\_address']

1. **Model selection**

There are two options to select an optimal model

* Run models and choose based on accuracy which one performs the best
* Another way is to see what type of data we have. Will it suit linear, non-linear, ensemble, boosting or neural network.

1. **Evaluation**

* KNN vs my model(DT)
* Special attention given to evaluation methods. K-fold used to check cross-validation technique and to find out which is the most suitable most.
* Hyper-parameter tuning is done using randomizedsearchCV method. All best parameters were found and model (DT) was again fit. Performed as good as NN.
* Special attention to ***over-fitting and under-fitting***
* Change in threshold to get even more desired result for bot traffic i.e 1.
* Accuracy: For initial analysis.
* Sensitivity (Recall): For our positive i.e bot traffic
* Precision(for non-bot traffic)
* Specify
* Roc\_auc(good for binary). Also we can change default threshold values.

Takeaway

Model (DT) is optimal solution. It trains fast so computation and time efficient. If new feature then we might test with all models again.