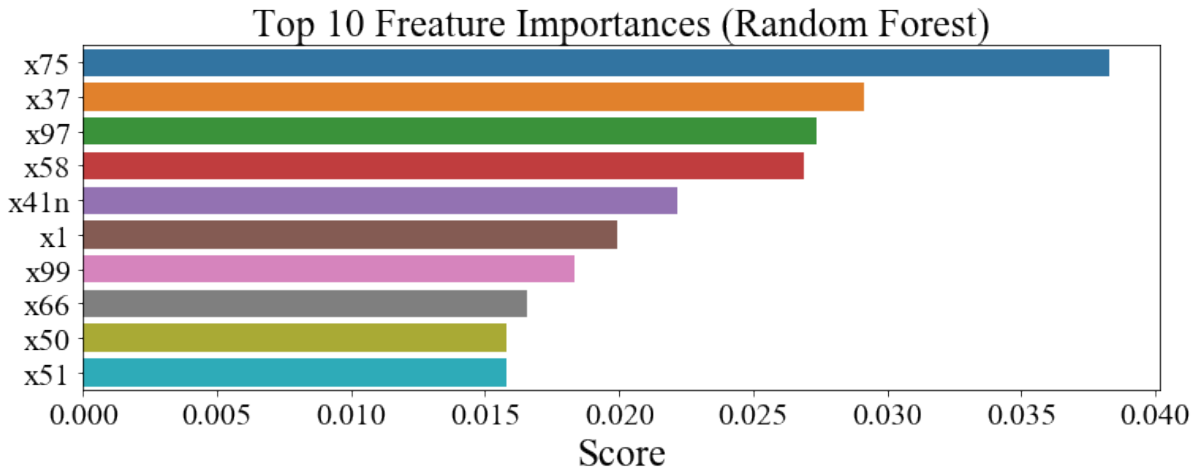
Sunday, 4 August 2019

Compare the two model approaches

The StateFarm Code Screen by Sungryong Hong



I have applied two ML models to the train set, (1) **LightGBM** : a gradient boosted tree model from mmlspark (Microsoft ML for Spark) (2) **RandomForest** : a random forest model from sparkml (Vanilla ML from Apache Spark). In my parameter tunings, both show close-to-overfit performances for the train sample.

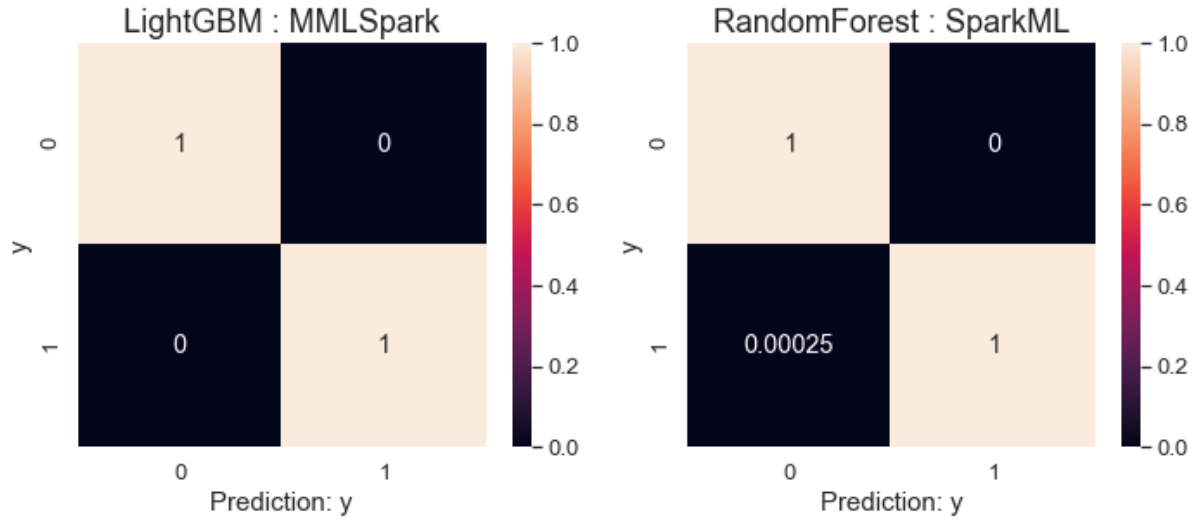
The *'result1.csv'* is produced by the **LightGBM** and the *'result2.csv'* by the **RandomForestClassifier**.

The LightGBM shows good prediction performances in most parameter tunings, while the RandomForest works well only for certain parameter combinations.

Hence, LightGBM can fit faster with low computation costs. But this also means that it can fall easily into a wrong local minimum; hence, overfitting data easily in a wrong way.

The RandomForestClassifier takes more time to find a proper fit than the LightGBM. However, due to its random ensemble, this model has a lower risk in overfitting issue.

If we need to make a quick and rough decision, LightGBM is a good choice. If we have more time to work on, we can try various methods including RandomForest or "Deeper" models like Neural Networks.



If I had more time to work on this problem, I would apply **multilayer perceptrons (MLPs)** to this training set for investigating how Deep Learnings can make predictions for this problem.