Machine Learning Nanodegree - Capstone Proposal

by Sven Görres

Domain Background

For my capstone project I want to use the data provided by Mercedes Benz on kaggle[1] ("Mercedes-Benz Greener Manufacturing"). The original idea of this kaggle project is to identify potential for using test stations more efficiently thus reducing emissions (since in this case it's car test stations) and subsequently costs by increasing capacity. This is a general issue in many manufacturing industrial fields since every test station has a relatively high initial cost and even being able to test 5 more parts a day on a single test station can lead to more than one thousand parts tested more per year thus the total amount of deliverable product does increase.

Problem statement

Based on testing data of a product (in this case a car) predict the assumed testing time for this specific feature combination.

Datasets and Inputs

The dataset used for this poject is obtained from kaggle[2]. It consists of a training data set with 4209 entries including resulting time and a testing data set with again 4209 entries but without the resulting time. I plan to use the training set and split it randomly into a training and a testing set. The data itselft contains of a unique test ID (incrementally count), the time consumed for the test (unit unknown) and 382 features unique to this test ID. Every feature could stand for e.g. colors or different sales components of the car (e.g. does it have bluetooth connectivity?). The data appears to be complete without any feature missing for any entry.

Solution Statement

My idea is to use a decision tree based model for this. Since the features are in mostly binary (1 or 0) it should be possible to create a (though very deep and wide) tree that should be able to predict the overall time rather well. Also the non binary features should be easily includable.

Benchmark Model

As a benchmark I will look at a publicly available solution to the original problem at kaggle based on deep learning[3]. Since decision trees and deep learning are techniques from different fields of machine learning it will be interesting to observe how decision trees will perform in this situation.

Evaluation Metrics

As a measuring instrument I want to use the coefficient of determination (R2[4]) as intended by Mercedes-Benz in the first place. This is also applicable to the benchmark model.

---

[1] https://www.kaggle.com/c/mercedes-benz-greener-manufacturing
[2] https://www.kaggle.com/c/mercedes-benz-greener-manufacturing
[3] https://www.kaggle.com/umbertogriffo/deep-learning/code
[4] https://en.wikipedia.org/wiki/Coefficient_of_determination

Project Design

My strategy to tackle this project will consist of several steps:

1. Analyze the data even further:
   E.g. are all features relevant or are some even constant over all samples?
   Are some features only appearing together?
2. Split the (train-)data into random sets for training and testing.
3. Use AdaBoostRegressor based on a DecisionTreeRegressor to fit the training data
4. Use GridCV to optimize on the different parameters of the DecisionTreeRegressor and the AdaBoostRegressor
5. Collect the R2 metrics on the seperated test data
6. Get the predictions for the same data from the benchmark model and compare R2 scores

To be honest I expect some issues here:

- I have no real experience on how good or bad the calculation performance for AdaBoost and DecisionTrees scale for this many features. I would expect it being not that good in performance.
- I am pretty sure I can not get a better score than the benchmark model (which seems to be pretty high based on the official kaggle leaderboard for this problem)
  However I think this will be a great learning experience nonetheless.