Machine Learning Engineer Nanodegree

Capstone Proposal - Mercedes Benz Greener Manufacturing

Kimpster @ Github (Ryan Chan) 4th Aug 2017

Domain Background

This project is a kaggle competition that was previously presented. The link is here :https://www.kaggle.com/c/mercedes-benz-greener-manufacturing

The competition is hosted by Daimler Mercedes Benz they are an automotive company. As their cars have a huge variety in models and specification it takes different times to test all the features and options. In this challenge they would like Kagglers to predict the time it would take on the test bench for given features and options. This will "contribute to speedier testing, resulting in lower carbon dioxide emissions without reducing Daimler's standards."

Problem Statement

The problem presented as mentioned earlier is to reduce the time cars spend on the test bench therefore resulting in faster testing and lower carbon dioxide emissions. There are many different features presented each affecting the time on the test bench, this is due the the many selections of options available for each car presented.

The main thing here is the prediction of time on the test bench given the features selected by the customers. Given this problem we would use regression techniques for this problem as the prediction 'y' is time values which is continuous data.

Datasets and Inputs

I will use information provided by SRK, a kaggler, who did a simple exploration of the datasets given. The link to the notebook is here :https://www.kaggle.com/sudalairajkumar/simple-exploration-notebook-mercedes

Three different data sets were given by Mercedes-Benz here: sample_submission.csv , test.csv and train.csv

I have excluded the sample_submission.csv as it only gives the submission format relating to how to submit on kaggle. Which has the ID column and y column, ID being the ID of the test set and 'y' being the time taken in seconds.

For the train data set we see 4209 rows and 378 columns and Test data set has 4209 rows and 377 columns. Test set has one column missing which is the 'y' column giving that it is what we are trying to predict makes sense.

Taking a look at the train dataset we notice that excluding ID and 'y' we are given all these features,

the first 9 columns are categorical whilst the rest is numerical. 'y' is a float given it is time in seconds. Every single of these features that are integers are 0 and 1, which we can assume is whether the specific has or does not have those specific features.

SRK analyzed the dataset fully and there are no missing values in the train set. He also analyzed columns that contain unique values. There were 12 columns with only unique values of 0 meaning they contributed nothing to the dataset in general or it was consistently applied to every output which means we can still ignore this.

Solution Statement

I will attempt to use Deep Learning techniques to attempt to predict the 'y' values of the test set, and submit onto kaggle. Given that there is late submission, although no prizes but it is still useful to see where our prediction model will place. I have decided to attempt using Deep Learning as the course only explored categorical model using deep learning but not regression and I believe this would be well suited to try since the data is simple and clean. Also majority of models used in this kaggle competition used other machine learning techniques such as XGboost and Random Forest and many Cross Fold validation techniques, I would like to see how Deep Learning compares in a regression problem with so many features. In this project I will use MLPs with different layers and architectures as my prediction models.

Benchmark Model & Evaluation Metrics

I am going to include a benchmark test which takes the mean of all the 'y' values in the train dataset and apply to the submission. I received a -0.00039 score. The scoring mechanism uses the R Squared value, this is obviously a much lower score than anything achieve. By using R^2 value is essentially a goodness of fit function. It is a measure of how close the data fits to the regression model. This is a good metric given that we are using a regression model.

For comparisons sake we could use one of the top models presented which has a score of 0.55425 on the private leaderboard by DDgg. Link to their kernel is here : https://www.kaggle.com/tobikaggle/stacked-then-averaged-models-0-5697

Essentially their model uses different CV techniques to feature engineer and then use an XGboost model to reach their value. This model was actually forked from Hakeem @ https://www.kaggle.com/hakeem/stacked-then-averaged-models-0-5697/code. He achieved a score of around 0.5697 on the public leaderboards.

I will try to get as close as possible to their score using Deep Learning techniques or even try to achieve a better score. Currently the submission only allows the calculation of the 19% of the data, which will be my metric, although I should be aware of overfitting problems.

Project Design

This project will mainly consist of using Keras and Tensorflow for the backend, essentially I will use a MLPs to predict the 'y' values. I would probably attempt this without any dimensional reduction at

first, and later apply PCA and ICA and compare which resulted in a better prediction model. This is due to the fact that there are so many features and the baseline discussed above actually used many dimension reduction techniques before applying it to XGboost model.

First I will remove all the irrelevant features, i.e. those which do not contain any additional information all 0s.

Then I will create different MLP architectures and check them to see which provides the highest accuracy

I would compare all my findings and optimize base on which creates the best accuracy. I believe if these model are unable to perform well or takes too much time to train I would first do dimensionality reduction and then move on to these models.
