BLG 454E

Learning From Data

Term Project

Team Name: Colorless green ideas

Students

Mertcan Yasakçı Burak Buğrul
150140051 150140015

Kaggle Names

mcanyasakci burakbugrul

Date of Delivery

31.5.2017

1. Introduction

The project chosen as our term project is Otto Group Product Classification Challenge from kaggle.com. The Otto Group is one of the e-commerce companies which sells millions of products in every day. Since this company has a great range of products. That’s why this challenge is mainly a product classification problem.

For the given challenge, a dataset with 93 features is supplied. The aim of this challenge is to construct a model that classifies more than 200000 products.

2. Dataset Description

In the dataset given for training has class labels. We are wanted to construct our model using training dataset. Dataset contains 93 feature for every product in the set. Also, every product has an ID number. After constructing our model with training data, we move on to test data to test our model.

Efficiency of a model is determined by the multi-class logarithmic loss function.

$$logloss =- \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(p_{ij})$$

3. Methods Used

We used quite few different methods like neural network, KNN, random forest etc. in this project. In addition, we used python with libraries numpy, sklearn and csv.

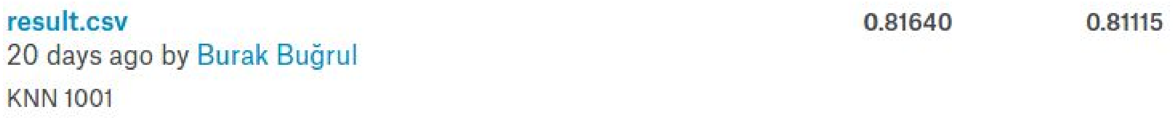
4. Experiment Results

In the beginning we just read the csv and analysed the data and found the closed interval of every feature individially. Then we applied this information the test data by simply counting matches of our intervals and features of data. This gave us a score of **2.18835**.

result.csv
23 days ago by Burak Buğrul
Analysing intervals

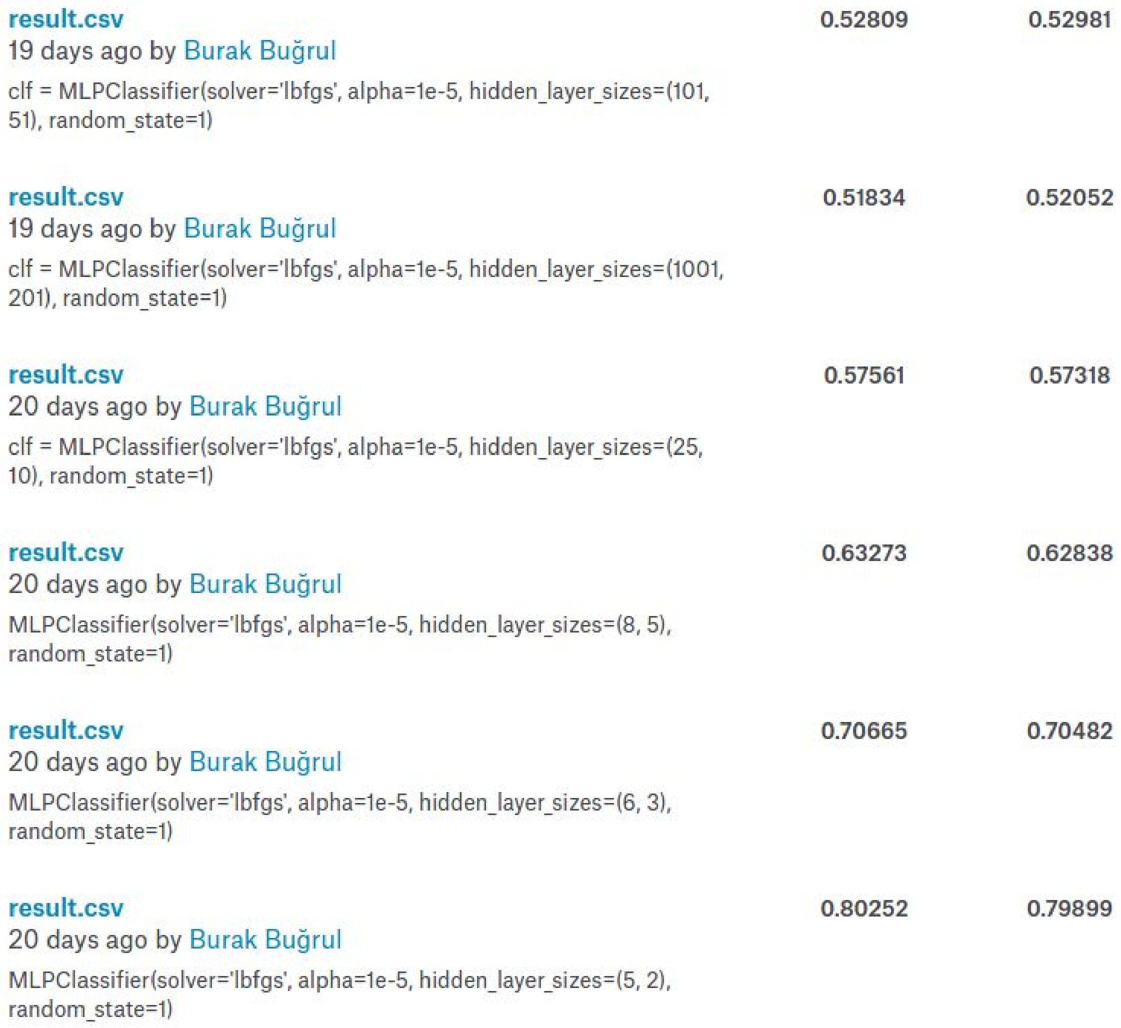
2.18835 2.18838 □

After that we performed KNN on dataset. Best results we got was **0.81640** with the KNN-1001.



result.csv 0.81640 0.81115
20 days ago by Burak Buğrul
KNN 1001

Then we tried neural network. In the beginning it gave a little better score than KNN-1001. But score got better as we increased the number of layers. Best one was **0.51834** points with 1001 layers.



result.csv 0.52809 0.52981
19 days ago by Burak Buğrul
clf = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(101, 51), random_state=1)

result.csv 0.51834 0.52052
19 days ago by Burak Buğrul
clf = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(1001, 201), random_state=1)

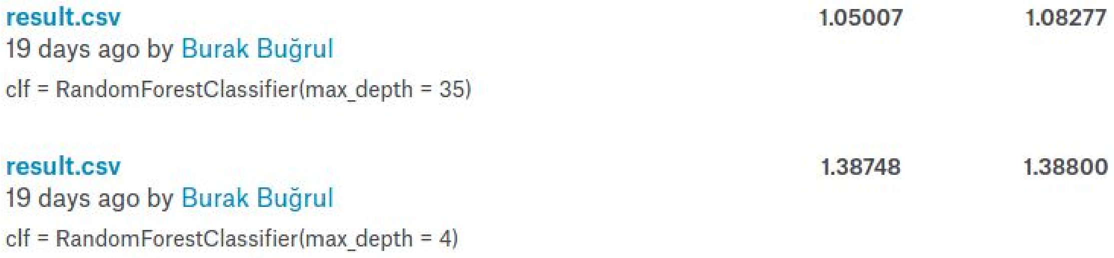
result.csv 0.57561 0.57318
20 days ago by Burak Buğrul
clf = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(25, 10), random_state=1)

result.csv 0.63273 0.62838
20 days ago by Burak Buğrul
MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(8, 5), random_state=1)

result.csv 0.70665 0.70482
20 days ago by Burak Buğrul
MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(6, 3), random_state=1)

result.csv 0.80252 0.79899
20 days ago by Burak Buğrul
MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1)

We tried random forest as well, but it was not as good as KNN and MLP. It gave best score of **1.05007**.

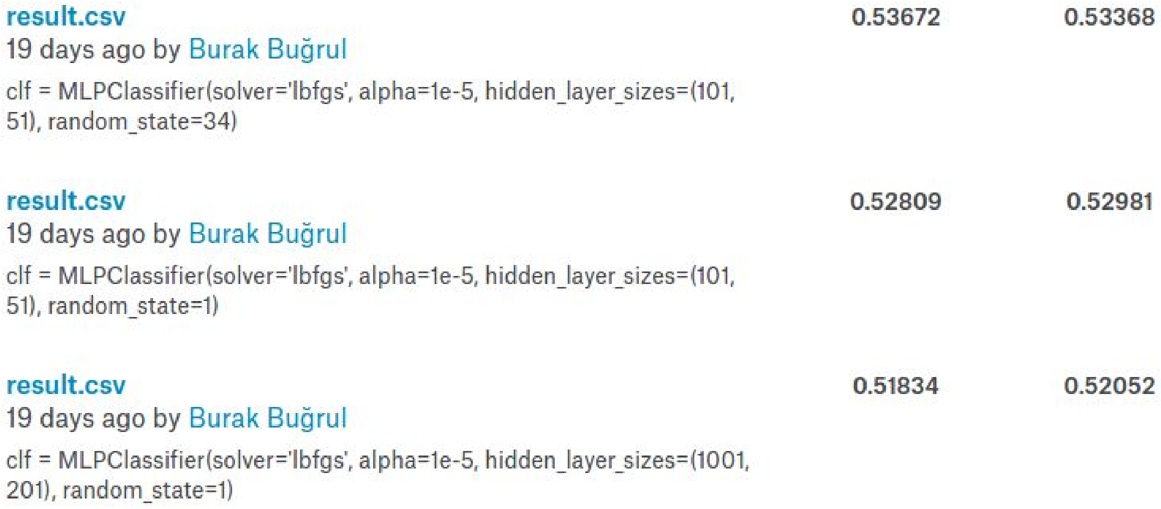


result.csv 1.05007 1.08277
19 days ago by Burak Buğrul
clf = RandomForestClassifier(max_depth = 35)

result.csv 1.38748 1.38800
19 days ago by Burak Buğrul
clf = RandomForestClassifier(max_depth = 4)

5. Discussion

We saw that results of direct using of classifiers may change a lot. At first we discussed implementing KNN with KD-Tree by using C++, but we decided not to do it because it will only boost time complexty of the algorithm not our score. Also we saw that after a point number of layers in MLP can not boost score anymore.



result.csv 0.53672 0.53368
19 days ago by Burak Buğrul
clf = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(101, 51), random_state=34)

result.csv 0.52809 0.52981
19 days ago by Burak Buğrul
clf = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(101, 51), random_state=1)

result.csv 0.51834 0.52052
19 days ago by Burak Buğrul
clf = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(1001, 201), random_state=1)

We can state that difference of scores of MLPs with 101 layers and 1001 layers do not differs as others.