MACHINE LEARNING

Task 3: Supervised Learning

Luis Santiyán García and Alejandro Ruiz Aranda[†]

Computation, UCLM, Paseo de la Universidad, Ciudad Real, 13004, Ciudad Real, España.

Contributing authors: luis.santiyan@alu.uclm.es; alejandro.ruiz13@alu.uclm.es;
[†]These authors contributed equally to this work.

Abstract

This problem is about regression of insurances policies and it consists of prediction of the final cost of each policy.

Keywords: knn, decision tree, random forest, hyperparameter optimization.

Contents

1 Problem description

The problem given is a regression problem whose objective is using the fields of the given dataset to predict UltimateIncurredClaimCost, the final cost of an insurance.

2 Dataset

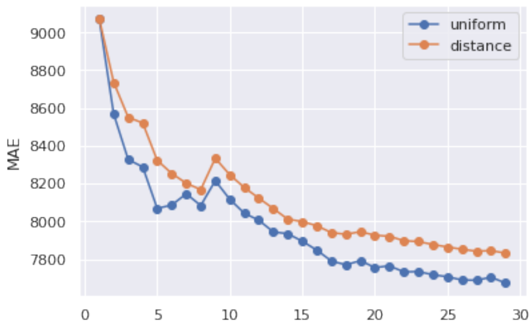
The training set consists of a 30000 insurances policies which have an id, date of accident, date of report, age of applicant, marital status, dependent children, others dependent, weekly wages, part time/full time(boolean P or F), hours of work per week, days worked per week, claim description, initial incurred claim cost and ultimate incurred claim cost.

3 Algorithms

In this section the different algorithms used for resolving the problem will be briefly explained.

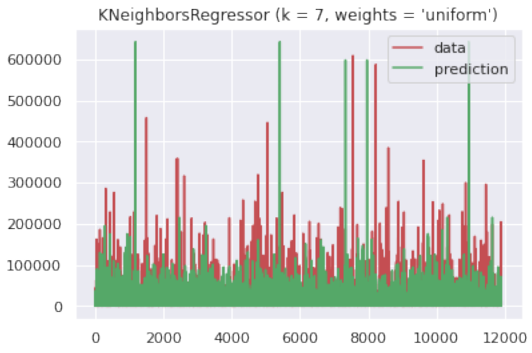
3.1 KNN

The KNN algorithm uses feature similarity to predict the values of any new data points. This means that the new point is assigned a value based on its similarity to points in the training set. Therefore, in this case we will try to predict the value of the 'Ultimate Incurred Claim Cost'



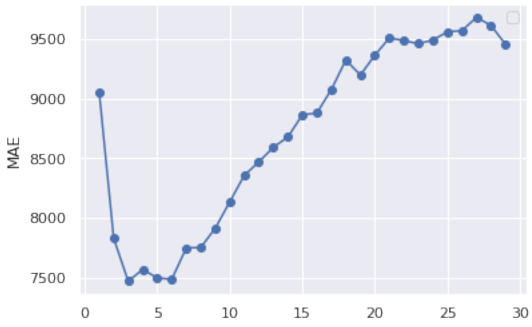
Since this is a regression problem, we can observe the error results through the MAE coefficient. We obtain different minimum error values between distance and uniform.
Min Value uniform : 7674.204899549508
Min Value distance : 7832.176673565171
Therefore we have a lower error coefficient in using the uniform parameter.

In the following graph we can see the comparison of the actual data and the predicted data. We can see that it is a acceptable prediction although it could be improved.



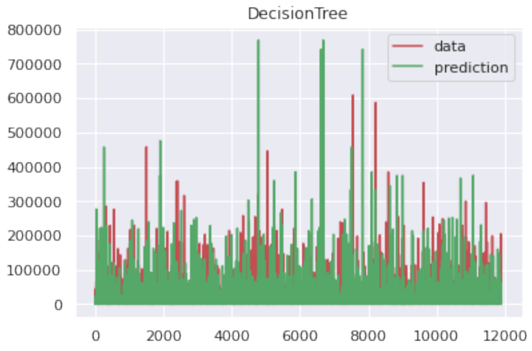
3.2 DecisionTrees

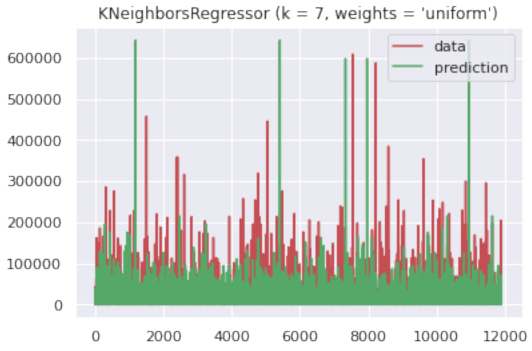
Another regression algorithm that can be useful in our prediction are the decision trees since they allow us to predict the value of a variable as a function of other given variables. We have made models of 1, 2 and 3 features, obtaining the best results in the model of 3 feautures.
Let's look at the MAE:



Min Value :7468.5437736132935
We obtain the lowest error coefficient so far so decision trees are the best option at the moment.

This can be seen in the prediction, where a significant improvement can be seen in comparison to the previous prediction made by knn.

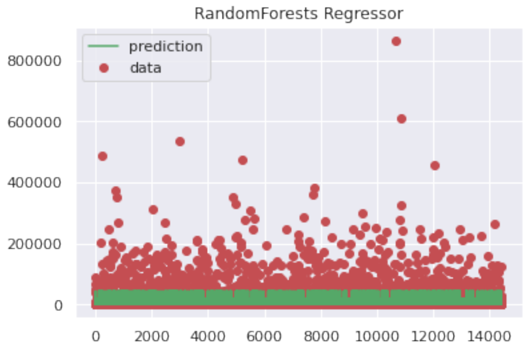




3.3 Random forests

Another algorithm used was the random forest algorithm, in this case we obtained an MAE of 6126.417761099787, which was the lowest of all.

In the following graph we can observe the prediction and we can see that we are clearly facing an over-fit model.



3.4 Hyperparameter Optimization of Ada Boosting

Hyperparameter optimization consists on setting hyperparameters for getting the best parameters for ada boosting (in this case).

This have been done via GridSearch to do a quicksearch rather than discovering new hyperparameters values.

This have been done using first hyperparameters:

```
search_grid={
'n_estimators':[500,1000,2000],
'learning_rate':[.001,0.01,.1],
'random_state':[1]}
```

We had to use this configuration because the others took a long time, yet this one also takes a long time, so we discarded this algorithm directly for this task because of its execution time.

The mae obtained with this configuration was: 8221.691677509963

Too high a value that does not pay off over time and proves that there are better alternatives such as decision trees.

4 Conclusions

This is due to the fact that although it is not the algorithm with the lowest error coefficient, its accuracy is quite acceptable, therefore, we obtain well-balanced models. It is also important to mention that its execution time is not very high compared to the other algorithms.