A comparison of KNN and Random Forest applied to the car evaluation dataset

Shengqiang Fan and Dipendra Tamang

Brief description and motivation

When people tend to buy a new car, lots of aspects play an important role to influence people's decision, such as car price, appearance, safety, fuel consumption etc. Car evaluation dataset [1] was derived from a simple hierarchical decision model originally developed by Bohanec and Rajkovic [2], which is a good example for us to learn the relationship between different aspects and different level of acceptable from buyers. Two machine learning methods (k-nearest neighbors and random forest) are applied to classify good acceptable car from the unacceptable ones given different features. Meanwhile, we try to figure out the importance of different aspects of cars. Finally, pros and cons of two ML methods will be discussed in this project.

Initial analysis of the data set including basic statistics

- Dataset: Car Evaluation dataset from UCI [1]
- Original dataset has 1728 observations, including 6 attributes as buying (vhigh, high, med, low), maint (vhigh, high, med, low), doors (2, 3, 4, 5more), persons (2, 4, more), lug_boot (small, med, big), safety (low, med, high) with class values (unacc, acc, good, vgood). No missing value was obtained in this dataset.
- Distribution of all 4 classes is plotted (figure 1), which shows the dataset is positive skewed from 'unacc' (unacceptable) to 'vgood'(acceptable) with majority in 'unacc'(unacceptable) and 'acc' (acceptable).
- The relationships between 6 attributes are explored by using correlation matrix (fig 2), which demonstrate that each attribute is not highly correlated to each other. Therefore, these 6 attributes can be treated individually for classification prediction. Meanwhile, We encode each attribute into numerical numbers as 1,2,3,4 rather than using dummy variables. Because we think price of buying/maint and safety have its increasing/decreasing relationship among different categories in each attribute.

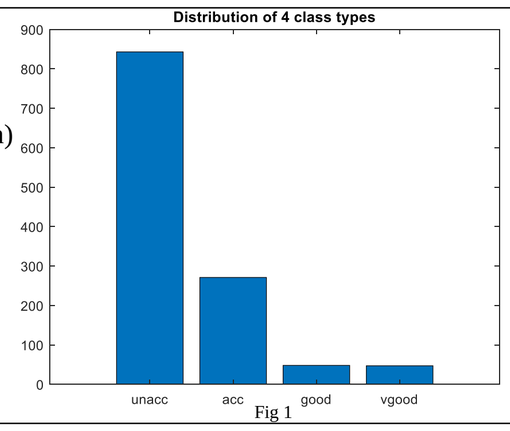

Fig 1

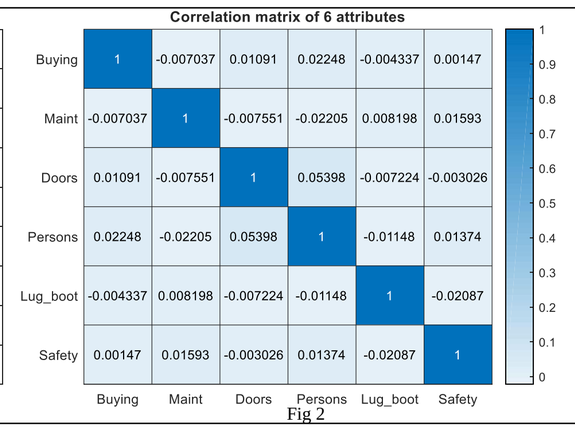

Fig 2

Brief summary of the two ML models with pros and cons

k-nearest neighbors algorithm (k-NN)

- A non-parametric and supervised method based on feature similarity, Euclidean, Manhattan and Minkowski distance for continuous variables and Hamming distance for categorical variables [3].
- Can be used for both classification regression, however, it is more widely used in classification problems.
- In k-NN classification, an target is classified by a majority vote of its k numbers of neighbors while if regression task, prediction will be the average values of its k nearest neighbors.
- Little or no prior knowledge about the distribution data is needed when using k-NN.

Pros

- It is not required to establish predictive model before classification. Hence, it can be a quick and simple way to begin machine learning datasets [5].
- It is capable to classify the events with low probability [5][6].
- It is easy to understand and implement, no (re)training phase required.

Cons

- KNN is more like a black box, which cannot provide the classification formula.
- Accuracy of KNN is highly affected by parameter k, however, it is time consuming and memory utilization for finding optimal k value
- It is a lazy learning which will store all training data for prediction. Therefore, high memory is required and prediction stage might be slow when dealing with big dataset [7]

Random Forest (RF)

- Random Forest is also a supervised learning algorithm introduced by Breiman [4], which is an ensemble of decision trees used for classification or regression.
- Bootstrapping is used to select several subsets from training data to train several decision trees.
- In RF classification task, an target is classified by a majority vote from all trees while if regression task, prediction will be the average prediction.

Pros

- Random forests can correct overfitting problem of decision trees by using bootstrapping [8].
- Prototypes (decision trees) are computed that explain the relation between the attributes and the classification.
- The importance of attributes can be generated after training phase.

Cons

- Overfitting problem still occurs when dataset is very noisy [9].
- It cannot predict beyond the range in the training data when used in regression task.
- In terms of categorical attribute with different numbers of levels, RF is biased to the attributed with more numbers of levels [10]. Hence, it's importance of attribute is not reliable in this case.

Hypothesis Statement

- We expect both ML models will produce similar and accurate classification prediction.
- Because a study [11] indicated that Random Forests algorithm has the similar performance as KNN on low dimensional datasets because both algorithm uses the weighted neighborhoods schemes.
- Based on common sense, attributes such as number of doors and lug_boot (the size of luggage boot) may not be important considerations compared to price and safety. Thus, we expect attributes as doors and lug_boot have less importance than safety and price in our ML models.

Description of choice of training and evaluation methodology

- We split original dataset into training set and test set, which consists of 70% and 30% original dataset respectively and made training set into random order.
- 10-fold cross validation was used in k-NN for finding hyperparameter of k value.
- The evaluation criteria used was classification error and precision. Confusion matrix was produced to show test error for both ML models.

Choice of parameters and experimental results

k-nearest neighbors algorithm (k-NN)

Parameter

- Normalization of each attribute was necessary. Otherwise, classification will mainly biased to attribute with large values.
- Grid search was used to find the optimal hyperparameter in k-NN, which is number of neighbors (k). Evaluation criteria was cross validation accuracy.

Result

- Optimal hyperparameter k was found to locate between 4-8 based on fig 3. We choose 5 as our model's parameter. In addition, when k increasing from 8, the accuracy on test set was not improved a lot (even decreased), which may due to overfitting problem.
- Normalization of each attribute was necessary to improve the accuracy.
- Cross validation accuracy of k-NN was 94.62%, test set accuracy was 95.38%.
- In order to figure out importance of each attribute in k-NN, each attributes was dropped and trained individually to monitor the accuracy change after dropping one attributes. Results are shown in table 1.

Random Forest (RF)

Parameter

- Grid search was used to find the optimal hyperparameter in RF, which are number of trees in the forest, number of attributes to sample at each split and depth of each tree (maximum splits) one by one. Evaluation criteria was cross validation error.

Result

- From grid search (fig 4-6), our chosen optimal hyperparameters are 50 trees, 4 attributes to sample at each split and 50 splits in maximum.
- Prediction accuracy on test set from random forest was 96.15%.
- One typical decision tree from random forest is shown in figure 9 (1,2,3,4 represents for 'unacc', 'acc', 'good', 'vgood' class), which can demonstrate every decision on each split.
- Importance of attributes were also generated from random forest as shown in figure6.

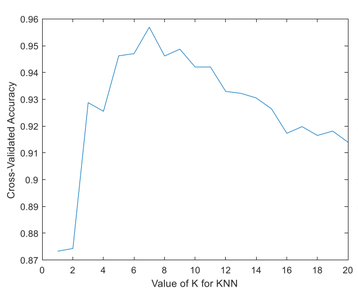

Fig 3

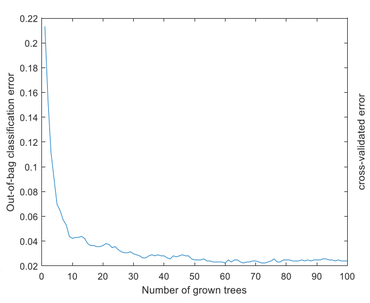

Fig 4

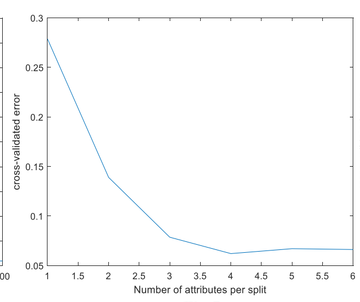

Fig 5

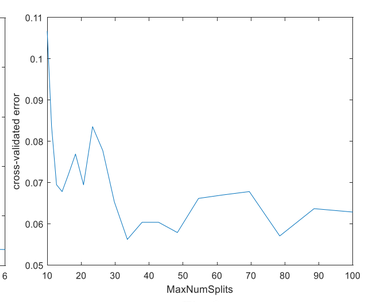

Fig 6

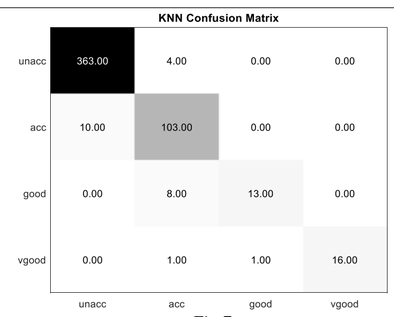

Fig 7

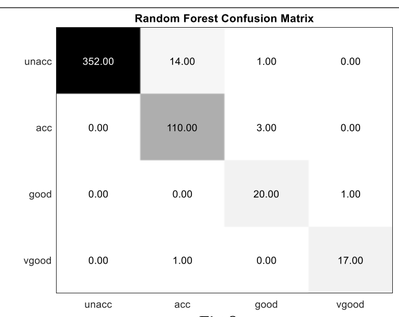

Fig 8

Analysis and critical evaluation of results

- Based on test set accuracy, the prediction performance of k-NN and random forest were quiet accurate and similar (95.38% and 96.15%), which meets our expectation in our hypothesis.
- Fig 8&9 are confusion matrix from two ML model. Precision of each class can be computed from confusion matrix. For k-NN, precision of 'unacc', 'acc', 'good', 'vgood' was 98.9%, 91.1%, 61.9% and 88.8% respectively. Similarly for random forest, precision of 'unacc', 'acc', 'good', 'vgood' was 96.2%, 97.3%, 95.2% and 94.4%. From two groups of precision, it can be noticed that although overall accuracy was similar from k-NN and RF, precision of 'good' and 'vgood' is much lower in k-NN. One explainable reason is that the distribution of original dataset was skewed as shown in fig 1. In other word, sample size of class 'good' and 'vgood' is too small to give a accurate perdition by k-NN, but random forest can overcome this problem.
- From table 1, it can be concluded that when dropping 'doors' attribute, test set accuracy only decreased to 90%, which is still a good prediction. It demonstrates that 'doors' attribute has low impact compared to other attributes in k-NN model. In the meanwhile, lower accuracy was produced when dropping 'persons' and 'safety' (68% and 66% respectively). It demonstrates that the importance of 'persons' and 'safety' are very high in k-NN. Importance of each attributes was also pointed out from random forest. The importance relationship from random forest is similar to k-NN, where 'persons' and 'safety' have highest values and 'doors' is lowest. This result is not totally under our hypothesis. In our hypothesis, we guessed 'buying price' & 'safety' might be dominant factors and 'doors' & 'lug_boot' have lowest importance. Therefore, 'persons' & 'safety' are first consideration when people purchasing a car in this dataset and 'doors' value does not affect people's choice a lot .

Attributes dropped	Buying	Maint	Doors
Test set accuracy	77%	83%	90%
Attibutes dropped	Persons	Lug	Safety
Test set accuracy	68%	83%	66%

Table 1

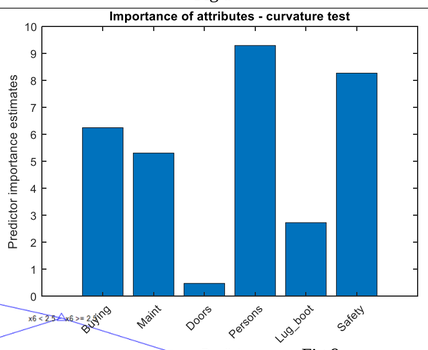

Fig 9

Fig 10

Lessons learned and future work

- Small size of certain class in training dataset (skewed dataset) has big influence on k-NN model but random forest can overcome this problem.
- Different evaluation technique can give different views of results. In this case though overall accuracy is similar for both ML models, precision of each class gives more detailed difference of performance.
- Future works: 1. Computational time can be compared of both ML models, especially for large dataset. 2. Compared these ML models on dataset with more attributes. 3. For k-NN, allocate different weight to each attribute to see how accuracy and precision will be improved. 4. For RF, compare performance of different method for each split, Gini impurity against information gain.

[1] UCI Machine Learning Group [online] http://archive.ics.uci.edu/ml/datasets/Car+Evaluation.
[2] M. Bohanec and V. Rajkovič. "DEX: An expert system shell for decision support", Sistemica, 1(1), pp.145-157. 1990.
[3] N. S. Altman. "An introduction to kernel and nearest-neighbor nonparametric regression", The American Statistician. 46 (3): 175–185. 1992.
[4] L. Breiman. Random forests. Machine learning, 45(1), pp.5-32. 2001.
[5] I. C. Yeh, and C. H. Lien, C.H. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients", Expert Systems with Applications, 36(2), pp.2473-2480. 2009.
[6] A. Starzacher and B. Rinner. "Evaluating KNN, LDA and QDA classification for embedded online feature fusion", In Intelligent Sensors, Sensor Networks and Information. International Conference On (pp. 85-90). IEEE. 2008, December.
[7] X. Hu and B. Wu. "Classification and summarization of pros and cons for customer reviews". In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03 (pp. 73-76). IEEE Computer Society. 2009, September.
[8] T. Hastie, R. Tibshirani and J. Friedman. The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5. 2008.
[9] J. Ali, R. Khan, R and N. Ahmad. "Random forests and decision trees", International Journal of Computer Science Issues (IJCSI), 9(5), p.272. 2012.
[10] A. Statnikov, L. Wang and C. F. Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC bioinformatics, 9(1), p.319. 2008.
[11] R.A. Nugrahaeni and K. Mutijarsa. "Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification", In Technology of Information and Communication (ISemantic), International Seminar on Application for (pp. 163-168). IEEE. 2016, August.