Analysis of car Dataset with Machine Learning

Darshankumar Kapadiya
Institute of Technology
Nirma University
Ahmedabad, India
21mcec02@nirmauni.ac.in

Yagnesh Bhadiyadra
Institute of Technology
Nirma University
Ahmedabad, India
https://orcid.org/0000-0002-8017-9535

***Abstract*—Cars are one of the most important luxury in the history of mankind which is now turning into a necessity for most of the people. Various car manufacturers are putting new models of cars everyday, and a user can be very confused seeing the data and selecting whether to buy a particular car or not. The machine learning algorithms can be helpful for a user to decide whether to select a car or not based on available features. However, many machine learning algorithms are there which can classify whether to buy a car or not based on given features of the car. We here in this paper propose a study of various machine learning algorithms which are applied to car dataset provided by University of California Irvine (UCI) dataset. The paper is having detailed analysis of all the algorithms and precision, accuracy, and f1-score parameters are discussed. We also discuss various techniques for handling imbalanced datasets.**

***Index Terms*—ML, Linear regression, Random Forest, KNN, XGBoost, SMOTE, ADASYN, car evaluation dataset.**

I. Introduction

Machine Learning is nowadays a compulsory tool for any kind of decision making exercise. Various Regression and Classification algorithms are developed every day to improve the performance of the prediction.

Here in this paper we propose a comparative study of various algorithm for the car dataset. The dataset is provided in [1]. We are providing a complete analysis and inference derivation in this paper. Section II describes related work on this dataset. In section III, We do Exploratory Data Analysis and then we apply classification models to that and we evaluate that model based on accuracy, precision etc. in Section IV. It described in detail the algorithms and their results along with analysis. Section V concludes the paper showing brief comparison all the algorithms.

II. Related Work

Awwalu et al. [3] analyzed the same dataset in 2014 and applied 3 algorithms : Decision Tree, ANN, and Naive Bayes and recored time to run these algorithms, and identified how much precentage of data was classified correctly. Their work comprised of testing the data on various combinations of train and test ratio and recording results for the same. They also performed clustering algorithms on this dataset and evaluted performance. However, their analysis lacked resampling of data, because data is not uniformly distributed among various classes. They also lacked other analysis based on F1 score which is considered to be one of the important parameters for classification algorithm evaluation when classes are imbalanced [2].

Table I
Class Distribution

CLASS	**PROPORTION**
Unacceptable ”unacc”	1210 (69.85 %)
Acceptable ”acc”	384 (22.28 %)
Good ”good”	69 (4.05 %)
Very Good ”vgood”	65 (3.82 %)
TOTAL	1728 (100 %)

Rehman et al. [4] did extend the work of Awwalu et al. and included Multilayered Perceptron Neural Network to the already done work. The author compared various algorithms based on correctly classified classes and recorded the runtime of the algorithm.

However both the algorithm achieved maximum accuracy of 93 % and 95 % respectively. We here present the classification using XGBoost algorithm and along with that we also present imbalance-learn algorithm to preprocess the data before classification. We also show Shuffled KFold validation for each algorithm to validate the model. We perform GridSearchCV on XGBoost algorithm for optimizing F1 score which is a good parameter to evaluate the model for imbalanced class data as mentioned in [2].

III. Exploratory Data Analysis

We first perform EDA (Exploratory Data Analysis) on the data. There are total 6 attributes/features in the data, and 4 output classes are there, namely acceptable, unacceptable, good and very good. We first see that whether data has any null values or not. The car dataset has no null values, or not any missing values as well. The data is highly imbalanced in terms of distribution of class. The imbalance is shown in the table I.

IV. Analysis of Various Algorithms

We have implemented different machine learning techniques for the car dataset which are as below.

1) Logistic Regression
2) Random Forest
3) K nearest neighbour
4) XGBoost Algorithm
5) XGBoost with ADASYN over sampling

Table II
CLASS DISTRIBUTION OF MANUALLY SAMPLED DATA

CLASS	PROPORTION
Unacceptable ”unacc”	1210 (55.12 %)
Acceptable ”acc”	384 (17.49 %)
Good ”good”	276 (12.57 %)
Very Good ”vgood”	325 (14.81 %)
TOTAL	2195 (100 %)

6) XGBoost with SMOTE over sampling
7) XGBoost with optimized parameters using Grid search

Above ML algorithms are first run on car dataset [5] then the same is run on modified and manually over-sampled dataset and results are show in table 1 and 2 respectively. Modified dataset contains 2195 instances compared to original 1728. In modified dataset we have added duplicates of minority classes ’good’ and ’very good’. here,we have replicated the class ’good’ three times and class ’very good’ four times.

In table III and IV classes 1, 2, 3, and 4 represents the classes ’unacc’, ’acc’, ’good’, and ’vgood’ respectively.

Here, from table III we can see that accuracy of randomized shuffled data for Logistic Regression and KNN are very low about 80.8 and 89.5 percentages respectively. also f1-score for minority classes 3 and 4 are significantly lower. For the random forest algorithm accuracy may seem acceptable but precision and f1-score are still low.

In Logistic Regression, precision of a certain class is very low, 0.29 and 0.27 respectively. Logistic Regression works by updating weights of the hypothesis function according to the data one by one. Now as the data consists of most of the values as ”unacc”, we get the precision about good and verygood low. The classifier for the class good and verygood is not trained properly. So, number of exampled that are classified as class 3 and class 4 have very less chances of being correct, which is not the case with class 1 and 2. F1-score is weighted average of precision and recall. We can see that LR (Logistic Regression) is also not able to classify the overall samples of class 3 and 4 properly. That’s why F1 score is not good. Had recall been great, F1 score would improve and it would be different from accuracy.

The RF (Random Forest) is an ensemble-learning algorithm. It works on a concept of combining multiple weak learners coming together to form a strong learner. Random Forests are best algorithms considering there is enough randomness in sub-trees that combine to form a classifier. As we can see from the table, RF performs significantly better than LR because of the inherent randomness it provides. We see that precision for the first class is 1, that is no false positives are there. Still we are not getting a good precision, as 20% (precision for class 3 and class 4 is around 80 %) of the positively identified good/very good card mislead users.

KNN works by selecting K neighbours and the other thing important here is distance measure. If we try to visualize the data, then most of the data will belong to a single class, so it will kind of classify the points which belong to class other than the majority class to the majority class and this will lead to a decrease in precision, and accuracy. In the last class, we can see that the precision is 57 % but as the recall is not so good, F1-score falls down to 50 % down from that.

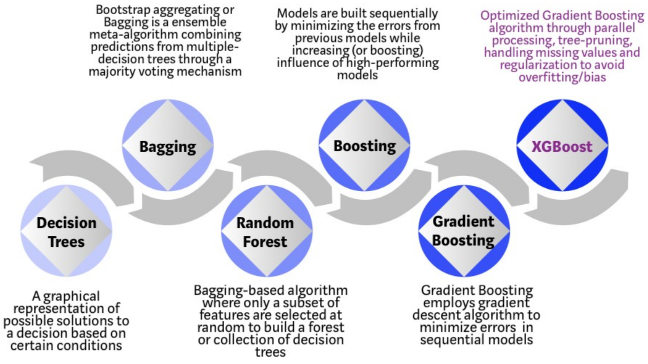
Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism
Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias
Bagging
Boosting
XGBoost
Decision Trees
Random Forest
Gradient Boosting
A graphical representation of possible solutions to a decision based on certain conditions
Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees
Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

Figure 1. Idea of XGBoost

Now, after implementing several standard algorithms, we went to XGBoost algorithm. The idea of XGBoost [6] is depicted Figure 1. XGBoost is a decision tree based algorithm only, but having inbuilt functionalities like Regularization, In-built tree pruning, parallelization for faster convergence, and many more. Now Regularization can kind of reduce the effects of overfitting which maybe classifying more samples to ”unacc” class which is having highest number of instances.We see that XGBoost performs beautifully with accuracy 0.995, and precision for minority classes increasing significantly (Table IV) from previous algorithms, although precision and f1-score still has space for improvement.

To improve minority class Precision and f1-score we use sampling methods ADASYN and SMOTE for XGBoost algorithm, here we get improvement of evaluation parameters like precision and f1-score at the minor loss of overall accuracy.

ADASYN stands for Adasptive Synthetic. It is an algorithm that generates harder to learn examples. Oversampling and under-sampling can cause other problems like, over-fitting, and deletion of some other classes respectively. The ADASYN algorithm utilizes K-nearest Neighbors. [7].

SMOTE stands for Synthetic Minority Oversampling Technique. It works same like ADASYN but with a little less accuracy. ADASYN adds a random value at the end after doing KNN which adds more reality to the original data.

At the end we run all the algorithms for manually sampled data as shown in table IV and the results are phenomenal. If we exclude Logistic Regression, then we get equal or improved performance for every other algorithm.

Random Forest algorithm gives phenomenal accuracy results, we also get the near-perfect precision and f1-score for it. Here, because we added duplicates of minority classes, those tuples are having more weight in decision tree of random forest. so, we are getting better classification for minority classes which improves overall accuracy.

In KNN, since we added duplicates of minority-class instances, instances of minority classes 3 and 4 got the impact

Table III
DATA FOR EXPERIMENT ON ORIGINAL CAR DATASET SHOWN IN TABLE I

Algorithm	Accuracy	Class	Precision	F1-score
Logistic Regression	0.808	1	0.88	0.91
		2	0.60	0.58
		3	0.29	0.24
		4	0.27	0.15
Random Forest	0.940	1	1.00	0.97
		2	0.82	0.89
		3	0.77	0.83
		4	0.82	0.72
KNN	0.895	1	0.90	0.94
		2	0.75	0.72
		3	0.79	0.67
		4	0.57	0.23
XGBoost	0.995	1	1.00	1.00
		2	0.98	0.97
		3	1.00	1.00
		4	0.85	0.92
XGBoost with ADASYN over sampling	0.978	1	0.99	0.99
		2	0.97	0.96
		3	0.98	0.99
		4	0.95	0.96
XGBoost with SMOTE over sampling	0.977	1	1.00	0.99
		2	0.97	0.97
		3	0.99	1.00
		4	0.96	0.97
XGBoost classifier with parameters optimized using Grid search	0.993	1	1.00	1.00
		2	0.97	0.99
		3	1.00	1.00
		4	1.00	1.00

factor of 4x and 5x compared to instances of classes 1 or 2. which leads to better classification of minority classes, and better classification evaluation parameters.

XGBoost, XGBoost with ADASYN oversampling, and XGBoost with SMOTE oversampling shows accuracy of 99.4%, 99.7%, and 99.9% respectively, but we get the best results with perfect overall accuracy, precision and f1-score for all 4 classes using XGBoost with parameters optimized using grid search. We here used GridSearchCV algorithm provided by sklearn.

V. CONCLUSION

Comparative study of the models used in this study shows that the XGBoost in both the cases shows best results in terms of overall accuracy compared to Logistics Regression, Random Forest, and KNN. Although, different variations of XGBoost leads to different precision values for the four classes. Sampling and Grid search on parameters leads to significant improvement in precision and f1-score of minority classes. We get the perfect classification of data with 100 % accuracy and 100 % precision as well as f1-score for all four classes when XGBoost classifier is used with optimized parameters obtained using grid search on manually sampled dataset. In future, imbalanced dataset can be improved using some better techniques and the traditional algorithms can perform well too. However, modern algorithms like XGBoost can be of a great help when data is highly imbalanced.

Table IV
DATA FOR EXPERIMENT ON MANUALLY SAMPLED CAR DATASET SHOWN IN TABLE II

Algorithm	Accuracy	Class	Precision	F1-score
Random Forest	0.990	1	1.00	1.00
		2	1.00	0.98
		3	0.99	0.99
		4	0.99	1.00
KNN	0.959	1	0.98	0.97
		2	0.87	0.93
		3	1.00	0.97
		4	0.91	0.91
XGBoost	0.994	1	1.00	1.00
		2	0.99	1.00
		3	1.00	1.00
		4	1.00	1.00
XGBoost with ADASYN over sampling	0.997	1	0.99	0.99
		2	0.99	1.00
		3	1.00	1.00
		4	0.99	0.99
XGBoost with SMOTE over sampling	0.999	1	1.00	1.00
		2	0.99	1.00
		3	1.00	1.00
		4	1.00	1.00
XGBoost classifier with parameters optimized using Grid search	1.000	1	1.00	1.00
		2	1.00	1.00
		3	1.00	1.00
		4	1.00	1.00

REFERENCES

[1] M. Bohanec and B. Zupan, ”UCI Machine Learning Repository: Car Evaluation Data Set”, *Archive.ics.uci.edu*, 1997. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Car+Evaluation. [Accessed: 30- Nov- 2021]

[2] P. HuiGol, ”Accuracy vs. F1-Score”, *Medium*, 2021. [Online]. Available: https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2. [Accessed: 30- Nov- 2021]

[3] Awwalu, J., Ghazvini, A., Bakar, A. A. (2014). Performance Comparison of Data Mining Algorithms: A Case Study on Car Evaluation Dataset. *International Journal of Computer Trends and Technology (IJCTT)*, 13(2).

[4] Rehman, Z. U., Fayyaz, H., Shah, A. A., Aslam, N., Hanif, M., & Abbas, S. (2018). Performance evaluation of MLPNN and NB: a comparative study on Car Evaluation Dataset. *International Journal of Computer Science and Network Security*, 18(9), 144-147.

[5] Bohanec, M., amp; Zupan, B. (n.d.). Car Evaluation Data Set. UCI Machine Learning Repository: Car Evaluation Data Set. Retrieved November 30, 2021, from https://archive.ics.uci.edu/ml/datasets/car+evaluation.

[6] V. Morde, ”XGBoost Algorithm: Long May She Reign!”, *Medium*, 2021. [Online]. Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d. [Accessed: 02- Dec- 2021]

[7] R. Nian, ”An Introduction to ADASYN (with code!)”, *Medium*, 2021. [Online]. Available: https://medium.com/@ruinian/an-introduction-to-adasyn-with-code-1383a5ece7aa. [Accessed: 02- Dec- 2021]