Project Report: Car Prices

Dhanashri Phalake

Abstract

The focus of this project is developing machine learning models that can accurately predict the class or type of a car based on its features. We implement and evaluate various learning methods on a dataset consisting of the different properties of class. Our results show that Gradient Boosting Classifier give the best results.

Dataset

For this project, we are using dataset of given 'car_class.csv' file.

Description of attributes

- Comp: Compactness
- Circ: Circularity
- D.Circ: Distance Circularity
- Rad.Ra: Radius ratio
- Pr.Axis.Ra: pr.axis aspect ratio
- Max.L.Ra: max.length aspect ratio
- Scat.Ra: scatter ratio
- Elong: elongatedness
- Pr.Axis.Rect: pr.axis rectangularity
- Max.L.Rect: max.length rectangularity
- Sc.Var.Maxis: scaled variance along major axis
- Sc.Var.maxis: scaled variance along minor axis
- Ra.Gyr: scaled radius of gyration
- Skew.Maxis: skewness about major axis
- Skew.maxis: skewness about minor axis
- Kurt.maxis: kurtosis about minor axis
- Kurt.Maxis: kurtosis about major axis
- Holl.Ra: hollows ratio

Pre-processing

In order to get a better understanding of the data, we plotted a boxplot of the data. We noticed that the dataset had many outliers. For preprocessing the data we used standardize StandardScaler which is offered by python sklearn library. We initially created an object of the StandardScaler() function. Further, we use fit_transform() along with the assigned object to transform the data and standardize it.

Methodology

We utilized several classifiers, including KNeighborsClassifier, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, GaussianNB, with a 80% - 20% split for the training and test data. For most of the model implementations, the open-source Scikit-Learn package was used.

GradientBoostingClassifier

This algorithm builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n_classes_ regression trees fit on the negative gradient of the loss function, e.g. binary or multiclass log loss. Binary classification is a special case where only a single regression tree is induced.
sklearn.ensemble.HistGradientBoostingClassifier is a much faster variant of this algorithm for intermediate datasets (n_samples >= 10_000).

Results

Learning Algorithm	Best Score on Test Data	Best Score on Training Data
GradientBoostingClassifier	0.7530631908086095	0.8055555555555556

Future Work

For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset.