**Report**

* Dataset source: https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data
* Number of instances in dataset: 1728
* Number of attributes in dataset: 6
* Number of fold cross-validation performed: 10

**Language used:** Python( sklearn )

**Preprocessing:**

We have used map function to preprocess the data made it into numerical.

There are no null values in the dataset taken.

**evaluation metric used**: Accuracy and F-score

**Results Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Best Parameters Used | Accuracy | F-Score |
| Decision Trees | max\_leaf\_nodes = 50  max\_depth = 10  max\_features = None | 96.78 | 96.69 |
| Perceptron | n\_iter = 15  alpha = 0.01 | 78.11 | 78.11 |
| Neural Net | hidden\_layer\_sizes = (100,100)  max\_iter = 100  alpha = 0.3  momentum = 0.7 | 93.55 | 93.22 |
| Deep Learning | hidden\_layer\_sizes = (100,100,100)  max\_iter = 100  alpha = 0.3  momentum = 0.7 | 93.47 | 94.05 |
| SVM | C = 100  Kernel = rbf | 98.43 | 98.43 |
| naïve Bayes | Alpha = 1 | 71.57 | 71.57 |
| Logistic Regression | C = 10 max\_iter = 10 | 82.19 | 82.19 |
| k-Nearest Neighbors | n\_neighbors = 5 leaf\_size = 30 | 95.28 | 95.28 |
| Bagging | n\_estimators = 30  max\_samples = 1000 | 97.36 | 97.44 |
| Random Forests | n\_estimators = 100 max\_depth = 10 | 97.60 | 97.52 |
| AdaBoost | n\_estimators = 10  learning\_rate = 1 | 86.78 | 86.78 |
| Gradient Boosting | n\_estimators = 30  learning\_rate = 1 | 99.17 | 99.17 |

**Analysis:**

* All the classifiers are applied on the car dataset from UCI Repository to detect whether the car is unacc, acc, good, vgood. Based on the tests on this assignment we have observed that Gradient Boosting method perfoms better than any other classifier in terms of Accuracy and F-Score.
* In case of our dataset, the samples are balanced across target classes hence the accuracy and the F1-score are almost equal
* In Gradient Boosting , To get the best value of the parameters we have set the parameters, n\_extimators to 30 and learning\_rate to 1
* Of all the classifiers, naïve bayes classifier has the least performance.
* We found that all the classifiers have many attributes associated with them but only the attributes we included in our log files were effecting the performance of the classifiers. We have not included the parameters that have no effect on the accuracy