Report

# Data Set Name: balance-scale

# URL : http://archive.ics.uci.edu/ml/machine-learning-databases/balance-scale/balance-scale.data

**Number of Instances:** 625 observations

**Number of Attributes:**5(including class label)

**Number of cross validation folds:** 10 folds

The dataset had a variety and I had good understanding of what data was doing. I was able to understand and analyze the data, hence it was pretty interesting assignment.

**Preprocessing:**

Data is numerical and class is categorical.

There is no missing value.

Numerical data has been scaled by mean value.

**Best Parameter for Classifiers:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Classifier** | **Best Parameters Used** | **Accuracy** | **F1- Weighted** |
|  |
| Decision Tree | DecisionTreeClassifier( max\_leaf\_nodes=100) | 76.8279569892% | 77.084448419% |
| Perceptron | Perceptron(penalty='l2', alpha=0.0001) | 86.8996415771% | 84.39148287% |
| Neural Network | MLPClassifier(hidden\_layer\_sizes=(30,10),activation='tanh',learning\_rate\_init=0.001,solver='lbfgs',max\_iter=100) | 96.4823348694% | 97.0836166% |
| Deep Learning | MLPClassifier(hidden\_layer\_sizes=(20,20,20,20,20,20,20),activation='relu',learning\_rate\_init=0.01,solver='lbfgs') | 98.0875576037% | 97.15228843% |
| SVM | SVC(C=1,kernel='linear') | 91.6871479775% | 92.40353088% |
| Gaussian NB | GaussianNB() | 87.8469022017% | 85.23358756% |
| Logistic Regression | LogisticRegression(penalty='l2', C=0.5, solver='sag', max\_iter=100) | 85.7680491551% | 82.89161337% |
| KNeighbors  Classifier | KNeighborsClassifier(n\_neighbors=15,algorithm='kd\_tree',weights='distance',leaf\_size=15) | 87.7035330261% | 84.947902118% |
| Bagging Classifier | BaggingClassifier(n\_estimators=50,bootstrap=True,bootstrap\_features=False) | 79.5366103431% | 80.37115447% |
| RandomForest Classifier | RandomForestClassifier(n\_estimators=30,bootstrap=True,max\_features=2,max\_depth =10) | 82.096774193% | 80.516394937% |
| AdaBoost Classifier | AdaBoostClassifier(n\_estimators=100,algorithm='SAMME.R') | 91.523297491% | 92.866036565% |
| GradientBoosting Classifier | GradientBoostingClassifier(n\_estimators=200,max\_depth=3) | 87.7035330261% | 85.5879477% |

**Out Put :**

| Decision Tree | Accuracy : 76.9867% | F1-Weighted : 77.26694% |

| Perceptron | Accuracy : 86.8996% | F1-Weighted : 84.39148% |

| Neuron Network | Accuracy : 95.0256% | F1-Weighted : 96.12438% |

| Deep Learning | Accuracy : 97.9186% | F1-Weighted : 96.94875% |

| SVM | Accuracy : 91.6871% | F1-Weighted : 92.40353% |

| Gaussian NB | Accuracy : 87.8469% | F1-Weighted : 85.23359% |

| LogisticRegression | Accuracy : 85.7680% | F1-Weighted : 82.89161% |

| KNeighborsClassifier | Accuracy : 87.7035% | F1-Weighted : 84.94790% |

| BaggingClassifier | Accuracy : 79.7056% | F1-Weighted : 81.57084% |

| RandomForestClassifier | Accuracy : 81.1418% | F1-Weighted : 81.19001% |

| AdaBoostClassifier | Accuracy : 91.2007% | F1-Weighted : 92.39975% |

| GradientBoostingClassifier | Accuracy : 87.7035% | F1-Weighted : 85.70886% |

* If we look at the above table, accuracy value of Deep Learning is higher than rest all other classifier and Decision Tree is least. The accuracy of Decision Tree classifier is little unpredictable because changing the values of max\_leaf\_nodes varies the result significantly.
* Training accuracy is highest for random forest it performed good on training data but our aim is to perform better in test data which was not fulfilled because it led to overfitting.
* I have computed another evaluation metric – F1-weighted, I feel that F1-weighted is more better measure for this dataset because it is easy to compare F1-weighted value and identify the best and weak classifier.