CS 6375.004 MACHINE LEARNING

Assignment – 5

Lakshmi Priyanka Parimi (lxp160730)

Jaya Padma Sri Maddi (jxm166230)

- Chosen Dataset – Haberman
- URL - https://archive.ics.uci.edu/ml/datasets/Haberman's+Survival

Details of Dataset:

The chosen dataset is multivariate, has more than 100 instances and the associated task is classification.

- Number of Instances: 306
- Number of Attributes: 4 (including the class attribute)
- Attribute Information:
  - Age of patient at the time of operation.(numerical)
  - Patient's year of operation (numerical)
  - Number of positive nodes detected (numerical)
  - Survival status (class attribute)
- Missing Attribute Values: None

Pseudo Code:

- Install packages.
- Read the CSV file.
- Remove Duplicates and null values from the dataset.
- Scale the data as the attribute values vary significantly.
- Sample the data into n- Folds (We used 10-fold cross validation in our assignment)
- Create a list for n-Fold
- Set all the classifier's accuracy and precision values to 0.
- Run a for loop for n-fold times
- All the classifiers are applied to the dataset for all n folds.
- For every iteration, for each model accuracy is calculated and summed up
- After this, calculate the average of all the accuracies obtained in each fold for each classifier and display it as the accuracy of the classifier. Also, calculate average precision for each classifier.

Pre-processing- The data has no missing values and the duplicates are removed initially. Then the data is scaled since the range of the values of the attributes vary tremendously. The data is then normalized to be given as input for the neural net, perceptron and deep learning.

Evaluation Metric Used: We used precision for evaluation.

Results:

Classifier	**Best Parameters Used**	**Accuracy**	**Evaluation Metric (Precision)**
Decision Tree	method = "class",control=rpart.control(maxdepth=30), parms=list(split="information")	72.05	60.02
Perceptron	hidden=0, threshold=0.5,lifesign = 'minimal', err.fct="sse"	70.12	65.65
Neural Net	hidden=c(c(4),c(3)), rep=5, threshold=0.1, learningrate = 0.2, act.fct = "logistic" ,lifesign = 'minimal'	68.50	65.70
Deep Learning	hidden=c(5,8,12,9,15), rep=5, threshold=0.1, learningrate = 0.2, act.fct = "logistic" ,lifesign = 'minimal'	66.49	64.48
SVM	cost=100, gamma=0.5, kernel='linear' , type="C-classification"	72.13	50.51
Naive Bayes	type="raw",laplace=1	74.39	57.55
Logistic Regression	method = "glm.fit",threshold=0.4	74.26	68.84
K-nearest Neighbors	cl=factor(trainData$class),k=10	75.43	60.55
Bagging	mfinal= 3, boos = TRUE,rpart.control(maxdepth=7))	71.50	70.67
Random Forest	importance = TRUE, ntree=50,maxnodes=20	71.20	56.64
Adaboost	mfinal = 5, boos= FALSE, control=rpart.control(minsplit=3,maxdepth = 10)	72.21	57.14
Gradient Boosting	max.depth=3,nrounds = 2	73.78	63.20

Analysis:

From the above mentioned results, based on the accuracy and evaluation metric(precision), we can clearly see that K-Nearest Neighbours classifier has got the highest accuracy and precision comparatively. This is because it has given consistent performance even when the parameters have been heavily altered. The next best classifier is Naive Bayes. Third best classifier is Logistic Regression. The best set of parameters and their accuracies along with precession have been tabulated.