Implementing classifiers on the dataset and Parameter Tuning (Results)

1) Random Forest

Experiment	Cross Validation Fold	Step	Outcome (Accuracy) %	Outcome (ROC curve area)
1	10	0.1	99.254	0.732
2	10	0.3	99.255	0.734
3	10	0.5	99.246	0.703
4	10	0.7	99.214	0.719
5	10	0.9	99.232	0.729
6	10	1.5	99.242	0.732
7	15	0.3	99.252	0.739
8	20	0.3	99.249	0.738
9	25	0.3	99.243	0.737

Parameter Tuning :

1) First we tried with minimum value for step with 10 as a crossfold validation.
2) Then we increased step value from 0.1 to 1.5 to check its effect on accuracy and ROC curve area.
3) As step here is used with parameter scale = “log” which decides the rate with which it removes less significant features.It starts with all the features.
4) Here we can observe that there is no significant difference and proportional change in Accuracy as well as ROC.
5) So we picket 0.3 where we got the best accuracy among those experiments.
6) From that keeping 0.3 constant we increased k fold that means increased records in training data so the accuracy and area under ROC increased with increasing k.

2) Logistic Regression

Experiment	Train/Test Ratio	MaxIt	Step Size	Outcome (Accuracy) %	Precision(%)
1	80:20	3	0.1	58.03	92
2	80:20	3	0.3	57.37	90
3	80:20	3	0.5	55.90	90.1
4	80:20	3	0.7	58.95	81.57
5	80:20	3	1	58.68	89.18
6	80:20	3	1.5	62.13	97.56
7	80:20	5	1.5	56.22	86.84
8	80:20	10	1.5	59.34	77.27
9	95:05	3	1.5	68.52	77.78

Parameter Tuning :

1) First we tried with minimum values for step with 80:20 split for training testing data.
2) As we increased value from 0.1 to 1.5 we observed that accuracy increased overall but the increase in accuracy was not much. We kept Maxit constant = 3
3) We observed highest accuracy and precision on stepsize = 1.5
4) Then we started increasing maxit and we observed that accuracy and precision both went down.
5) So far the best Output metric we got was on maxit = 3 and step size = 1.5
6) Now we increase the train test split to 95:05 , so the accuracy went up significantly as expected.

3) KNN

Experiments	Cross Validation Fold	K	Tune Length	Outcome (Accuracy) %	Outcome (Precision)
1	10	2	5	78.9	89.1
2	10	3	5	73.9	87.9
3	10	5	5	71.6	85.4
4	10	7	5	71	85
5	10	7	3	67.3	80.3
6	10	9	3	65.3	79.4
7	10	9	7	64.4	78.1
8	20	9	7	64.7	78.5
9	25	4	5	73.2	85.6
10	10	4	5	76.2	87.2

Parameter Tuning:

1) First observation led the best Accuracy and precision outcome.
2) First of all, with the increase in 'K' the bot the Accuracy and Precision outcomes decreased, which can be observed in the experiments [1,2,3,4]
3) The decrease or increase in the "Tune Length" parameter keeping other parameters constant led to the decrease in accuracy and precision outcomes. Observed in experiments [(4,5) & (6,7)]
4) We believe, the ideal value for the "Tune Length" is 5, which gives us the best outcome as observed in Experiment 1.

4 & 5) Bagging & Boosting

Bagging:

Experiments	Cross Validation Fold	mfinal	Cp (Complexity parameter)	Outcome (Accuracy) %	Outcome (Precision)
1	10	10	0.03	57.9	77.8
2	10	15	0.03	57.7	77.4

3	10	20	0.03	57.4	76.9
4	10	25	0.03	57.1	76.9
5	10	100	0.03	57.09	77.1
6	10	10	0.01	59.4	78.5
7	10	10	0.05	58	78.4
8	10	10	0.07	55.7	75.8
9	25	10	0.03	57.7	78.4
10	50	10	0.03	57.1	77.6

Parameter Tuning:

1) Firstly, we started varying the "mfinal" parameter and as observed in Experiments [1,2,3,4,5], there is minimal affect on the Accuracy and Precision outcomes.
2) The best Accuracy and Precision outcomes is observed in Experiment 6, where "mfinal" = 10 and cp = 0.01.
3) Secondly, Increasing the parameter "cp" has marginally more affect on the Accuracy and Precision outcomes which observed a decrease. This can be seen in Experiments [6,7,8]
4) Samples size, that is the Cross Validation Fold doesn't affect the outcomes. Observed in Experiments [9,10]

Boosting:

Experiments	Cross Validation Fold	mfinal	Cp (Complexity parameter)	Outcome (Accuracy) %	Outcome (Precision)
1	10	10	0.03	58.4	78.7
2	10	15	0.03	58.1	78.1
3	10	20	0.03	57.9	77.9
4	10	25	0.03	57.8	77.6
5	10	100	0.03	57.6	77.5
6	10	10	0.01	60.5	79.6
7	10	10	0.05	58.9	78.9
8	10	10	0.07	57.1	77.3
9	25	10	0.03	57.9	78.7
10	50	10	0.03	57.6	77.9

Parameter Tuning:

1) As observed, it follows almost the same pattern of observations as the bagging with sightly better Accuracy and Precision outcomes.
2) Similar to bagging, the best Accuracy and Precision outcomes is observed in Experiment 6, where "mfinal" = 10 and cp = 0.01.

Analysis:

Number of instances in dataset: 303

Number of attributes in dataset: 13
How many fold cross validation performed: 10, 20, 25, 50 etc

Best results of the classifiers:

Classifier	Validation Technique	Accuracy	ROC	Precision
Logistic Regression	Sampling and taking average	68.52	-	77.78
KNN	10-fold cross validation	78.9	-	89.1
Bagging	10-fold cross validation	59.4	-	78.5
Random Forest	10-fold cross validation	99.25	0.739	-
Boosting	10-fold cross validation	60.5	-	79.6

Performance justification for each classifier.

Random forest worked out to be the best of all classifier because it's really easy to build RF model and only parameter to tune (Number of features to use)
Also FR model is more robust to overfitting so even if we give 95% data as a training data, the model does not overfit and performs well. Also we have noticed that Random forest builds model faster than other classifiers used in this task.

KNN performed the second best. It performs good even when there's less correlation between the attributes. In the dataset that we are using some features influence class label more than some other features, but KNN does not take that into consideration while calculating distance between two data points. This may be the reason we got less accuracy than random forest.

Logistic regression did not perform well. From what we have read in the sources it requires scaled data, But When we tried using scaled data it could not built model because of multiple errors.
We built a model on the non-scaled data, so we think that LR accuracy is less.

Relatively lesser accuracy observed in bagging & boosting cases. Because, we believe, as the values in the dataset are not clean and scaling has been avoided due to constant errors while designing the model

https://www.quora.com/When-would-one-use-Random-Forests-over-Gradient-Boosted-Machines-GBMs

https://www.quora.com/Classification-machine-learning-When-should-I-use-a-K-NN-classifier-over-a-Naive-Bayes-classifier
https://www.quora.com/topic/Logistic-Regression