CSC6515 – Machine Learning for Big Data

Assignment 1

-Shakti Singh

B00779881

Naïve Bayes:

	Confusion mat. (elk=1)	Confusion mat. (cattle=1)	Confusion mat. (deer =1)	Accuracy
Train	[[152 1689] [81 1929]]	[[2936 136] [678 101]]	[[383 2406] [66 996]]	0.5623
Test	[[55 571] [28 630]]	[[972 47] [228 37]]	[[129 794] [22 339]]	0.5612

Accuracy of the Naïve Bayes classifier on training data and test data do not differ that much. And they both are pretty low too. This is due to the fact that firstly, the data is unbalanced and its splitting into training and testing sets does not contain the desires amount of observations of each class, necessary for learning. This can be easily seen by the confusion matrices, high number true negatives in (is_cattle?) and false positives in (is_deer?) set. The testing data also shows the same pattern in different proportion.

Logistic Regression:

	Confusion mat. (elk=1)	Confusion mat. (cattle=1)	Confusion mat. (deer =1)	Accuracy
Train	[[1262 579] [581 1429]]	[[3012 60] [689 90]]	[[2656 133] [929 133]]	0.7428
Test	[[411 215] [204 454]]	[[1002 17] [226 39]]	[[901 22] [316 45]]	0.7403

Logistic regression performs well on binary classes, that's what is reflected in the accuracies of both training and testing data sets. It performs pretty the same on training and testing datasets because the predictions are performed along the log

curve, all the values lying behind a specified threshold are classified negative and others as positive. Be it the training observations or testing observations, predictions are performed along that curve only, thus giving close values of accuracy. By looking at the false negatives in the confusion matrix, we can conclude that Cattle classification is not addressed properly in this model.

Decision tree:

	Confusion mat. (elk=1)	**Confusion mat. (cattle=1)**	**Confusion mat. (deer =1)**	**Accuracy**
Train	[[1841 0] [0 2010]]	[[3072 0] [0 779]]	[[2789 0] [0 1062]]	1.0
Test	[[402 224] [217 441]]	[[865 154] [151 114]]	[[750 173] [211 150]]	0.7066

Decision trees are coming up with 100% accuracy with the training data set because of the overfitting that is happening on train data. Decision trees function by making and storing the rule trees, so, whenever a tree is learned over a data set and that data itself is used for testing, Decision tress return full accuracy. However, when we run the classifier on testing data, the true accuracy of the classifier comes forth. The extent of learning is demonstrated when a classifier is used to predict the testing data. Having a close look at the confusion matrix for (is_deer?), a large number of false negatives can be noted, signifying poor performance on this class.

Random Forest:

	Confusion mat. (elk=1)	**Confusion mat. (cattle=1)**	**Confusion mat. (deer =1)**	**Accuracy**
Train	[[1830 11] [35 1975]]	[[3072 0] [53 726]]	[[2788 1] [65 997]]	0.9857

Test	[[471 155] [213 445]]	[[980 39] [182 83]]	[[849 74] [249 112]]	0.7632

Similar to the decision trees, Random forest classifier also works on trees thus we can see that the accuracy of training data prediction is almost perfect. It is not perfect 1.00 because of the bagging technique that is utilized in it. It forms a bag of attributes and observations, constructing trees and learning in a random fashion for a large number of bags(samples). The accuracy on testing dataset is maximum of all the classifiers compared. However, noting down the (is_deer?) confusion matrix with large number of false negatives, one can conclude that it is not performing well on this class.

10-fold Cross-Validation results:

Classifier	**Mean (+/- Standard Deviation)**
Random Forest	0.75 (+/-0.04)
Decision Tree	0.70 (+/-0.06)
Logistic Regression	0.74 (+/-0.06)
Naïve Bayes	0.55 (+/-0.18)

Using 10-fold cross-validation on the classifiers, we got the results mentioned above. We can clearly see that the best performing classifier is Random forest. It has the maximum value of mean accuracies and minimum standard deviation. The closest accuracy to the Random forest is Logistic regression having almost same the mean accuracy but large standard deviation. Random forest has come out to be the winner because of the bagging technique that is utilized in the construction of the model. Random forest has a tendency to overfit the data that is given for training and thus it shows the above-mentioned results. Logistic regression is also said to perform with good amount of accuracy in a binary classification problem. Decision tree is also performing pretty decently in this classification task, with a fair amount of accuracy. While, Naïve Bayes classifier is not performing so good because of the fact that it works on likelihoods and prior probabilities in order to find out posterior probabilities. In this data, the classes are unbalanced thus making the task of prediction difficult.

Student's t-test results:

	t-statistic	**p-value**
RF vs DT	8.3851	3.05311
RF vs LR	0.3632	0.7190
RF vs NB	6.3619	5.9305

Here we can see that p-value for the t-test between the cross validated accuracies of Random Forest classifier and Decision trees classifier is 3.05311, which is very large than the significance threshold ($\alpha = 0.05$). thus, rejecting the null hypothesis that Means of both groups are the same. This can be noted from their mean accuracies, which are quite a bit different.

```
Student's t test results :
RF vs DT :  Ttest_relResult(statistic=8.3851113385647302, pvalue=3.0531140770328337e-09)
RF vs LR :  Ttest_relResult(statistic=0.36321503978456898, pvalue=0.71907851337571227)
RF vs NB :  Ttest_relResult(statistic=6.3619415195517117, pvalue=5.9305279110856924e-07)
```

On the contrary, looking at the p-value for t-test between the Random Forest and Logistic regression accuracies, we cannot reject the null hypothesis because it is quite high than the significance threshold of 0.05. which means that there is a 71.9% chance that their means are same. We can conclude that accuracies of these two classifiers are quite close to one another.

Lastly, seeing the t-test results of classifier Random forest and Naïve Bayes, we can right away conclude that mean of their accuracies are quite different from each other. (p-value >>> α; unrealistic).

Testing Random forest classifier for different number of trees:

Number of trees (n_estimators)	**Accuracy (10-fold CVed)**
10	0.75(+/-0.06)
20	0.75 (+/-0.06)

50	0.76 (+/-0.05)
100	0.76 (+/-0.05)

As we can see in the results of tuning the Random forest classifier for number of trees, there is no significant change in the accuracy of the classifier even if we increased the number of trees by 5 times. Theoretically, when we increase the number of trees in an Random forest classifier, the accuracy of the model increases. But, when the number of trees grow too much as compared to the sampling size, accuracy of the classifier turns constant w.r.t the number of trees. Moreover, higher number of trees result in an increase in computational costs.

Out of these results, I choose the classifier with 50 trees. This is because the accuracy of the classifier is a bit better than the classifiers with 10 and 20 trees. Also, it has the same accuracy as that of the classifier having twice as many the trees i.e. better results for less computational costs.

```
Student's t test results for comparision with Random Forest with 50 trees:
RF vs DT :  Ttest_relResult(statistic=7.9185709774313979, pvalue=9.8499304418605617e-09)
RF vs LR :  Ttest_relResult(statistic=2.2170001360128544, pvalue=0.034620675710677333)
RF vs NB :  Ttest_relResult(statistic=7.4880377975627743, pvalue=2.9750626257393437e-08)
```

These Student's t-test results show that the Random forest classifier with 50 trees shares most similarity with the results of the Logistic regression classifier (close to the $\alpha = 0.05$). All the other classifier have a really high p-value, making any significance negligible.