

HW 4 :

1. Weka - Weka Accepts CSV file. So I have written the script to convert input(tsv) to csv.

a. Experimentations

- i. J48 -C 0.25 -M 2

Time taken to build model: 6.83 seconds

Correctly Classified Instances 40026 86.0608 %

Incorrectly Classified Instances 6483 13.9392 %

=== Confusion Matrix ===

a b <-- classified as

33271 2080 | a = <=50K

4403 6755 | b = >50K

- ii. SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K

Time taken to build model: 1142.78 seconds

Correctly Classified Instances 39509 84.9491 %

Incorrectly Classified Instances 7000 15.0509 %

=== Confusion Matrix ===

a b <-- classified as

33172 2179 | a = <=50K

4821 6337 | b = >50K

- iii. Logistic -R 1.0E-8 -M -1

Time taken to build model: 28.92 seconds

Correctly Classified Instances 39580 85.1018 %

Incorrectly Classified Instances 6929 14.8982 %

=== Confusion Matrix ===

a b <-- classified as

32924 2427 | a = <=50K

4502 6656 | b = >50K

b. Discussions

- i. One of the major reasons for about 3 percent better results for Weka Tool is primarily in handling continuous variables. For continuous variables , I just took the mean and made the binary classifier based on

mean values. However, the Weka Tool estimates the threshold based on some sophisticated methods and can do binary as well as Multi way classification of continuous variables based data at hand. Another reason for better performance is that it uses probability based method for estimating values in case of missing feature attribute values.

ii. I chose Logistic Regression. As we used Decision Trees here so I will compare and contrast Decision Trees with Logistic regression providing its advantages and disadvantages over Decision trees.

1. Both algorithms are really fast. There isn't much to distinguish them in terms of run-time.
2. Logistic regression will work better if there's a **single** decision boundary, not necessarily parallel to the axis.
3. Decision trees can be applied to situations where there's not just one underlying decision boundary, but many, and will work best if the class labels roughly lie in hyper-rectangular regions.
4. Logistic regression is intrinsically simple, it has low variance and so is less prone to over-fitting. Decision trees can be scaled up to be very complex, are more liable to over-fit. Pruning is applied to avoid this.

iii. Amongst the three, on the given dataset, Decision Tree clearly outperforms both Logistic Regression and SVM in terms of time to build model and accuracy. Dataset is highly skewed ($\leq 50K$ more than 75 percent). If we look into the Confusion Matrix, we observe that for label ' $\leq 50K$ ', SVM almost performs as good as the Decision Tree, but performs really poorly for the other label. Whereas logistic Regression performs better than SVM in ' $> 50K$ ', but performs poorly in ' $\leq 50K$ '. Time wise both Logistic Regression and Decision Trees are fast but SVM is slow.

In order to improve the accuracy further, I applied the Resample Filter Under Unsupervised for the instance in the Preprocess Step on the data. It improved Decision Tree Accuracy to 88.78% and to 85.6% for Logistic Regression. Main intuition for using this filter was because the data was quite skewed (lot of data of one label as compared to other), so normalise the skewed data was bound to give better results.