Akanksha Bansal	**Assignment 4**
Data Mining	**15,April 2014**

Problem 1 Multinomial logistic regression model is given by

$p(y = c|x, w) = \frac{\exp\{w_c{}^T\phi(x)\}}{\sum_{c`=1}^{C} \exp\{w_{c`}{}^T\phi(x)\}} = Softmax_w(x, c)$

In order for the above expression to be identifiable, the equation should result in different models when the weights are changed. Thus the model for $p(y = c|x, w)$ and $p(y = c|x, w + k)$ should be different.

$p(y = c|x, w + k) = \frac{\exp\{(w_c+k)^T\phi(x)\}}{\sum_{c`=1}^{C} \exp\{(w_{c`}+k)^T\phi(x)\}}$

$= \frac{\exp\{w_c{}^T\phi(x)\}.\exp\{k^T\phi(x)\}}{\exp\{k^T\phi(x)\}.\sum_{c`=1}^{C} \exp\{w_{c`}{}^T\phi(x)\}}$

$= \frac{\exp\{w_c{}^T\phi(x)\}}{\sum_{c`=1}^{C} \exp\{w_{c`}{}^T\phi(x)\}}$

$= p(y = c|x, w)$

Part b)

Likelihood of a generic data set D as a function of w

$P(X, Y|w) = P(Y|X, w)P(X|w)$

As X is independent of w and assuming iid, therefore

$P(X, Y|w) = P(X)\prod_{1\le i\le N} P(y_i|x_i, w)$

Let us define another variable $k_{ij} = 1$ when y_i is equal to c_j and 0 for the rest of the $C - 1$ classes. Then, the likelihood expression can be written as

$P(X, Y|w) = P(X)\prod_i \prod_j (P(y_i = c_j|x_i, w))^{k_{ij}}$

The negative log-likelihood with respect to w :

$P(X, Y|w) = P(X)\prod_i \prod_j (P(y_i = c_j|x_i, w))^{k_{ij}}$

$-\log P(X, Y|w) = -\log P(X) - \sum_i \sum_j k_{ij} \log(P(y_i = c_j|x_i, w))$

Maximize the expression $P(w|X, Y)$ by minimizing the negative logarithm of the left hand side from below equation. The Value P(X,Y) will be dropped because the value is independent of w.

$P(w|X, Y)P(X, Y) = P(w)P(X, Y|w)$

$-\log P(w) - \log P(X, Y|w)$

$= \frac{\alpha}{2}w^T w - \log P(X) - \sum_i \sum_j k_{ij} \log(P(y_i = c_j|x_i, w))$

$= \frac{\alpha}{2}w^T w - \sum_i^N \sum_j k_{ij} \log(P(y_i = c_j|x_i, w))$

$= \frac{\alpha}{2}w^T w - \sum_i^N \sum_j k_{ij}(\{w_{c_j}{}^T\phi(x_i)\} - \log(\sum_{c_i=1}^{C} \exp\{w_{c_i}{}^T\phi(x_i)\}))$

Part c)

The derivative of the regularized negative log conditional likelihood with respect to a particular

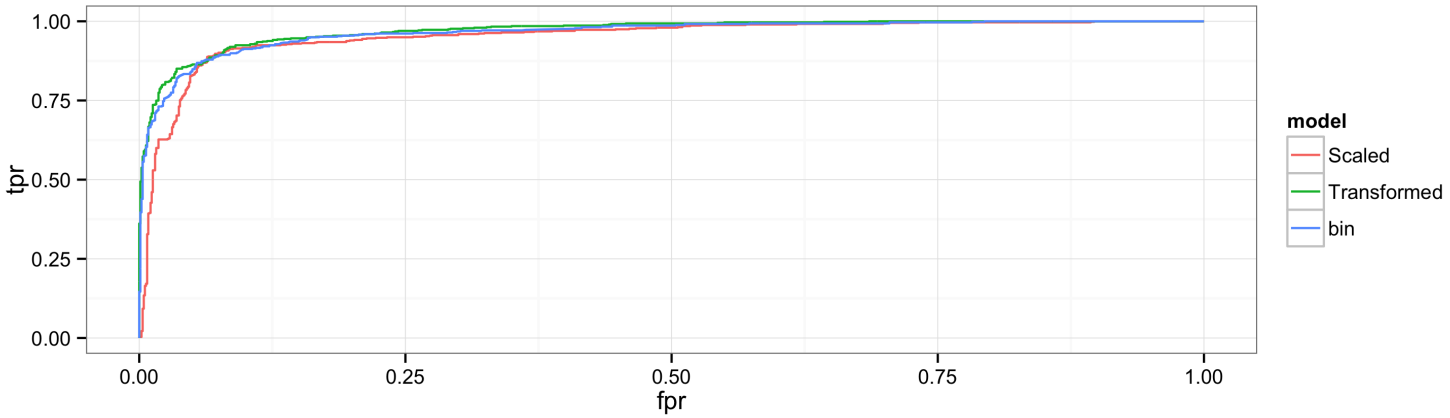
feature weightw_{c_k}, as derived from pervious expression:

$$= 2w_{c_k}\frac{\alpha}{2} - \sum_{i}^{N} k_{ic_k}(\{\phi(x)_i\} - \frac{\exp\{{w_{c_k}}^T\phi(x_i)\}}{(\sum_{c=1}^{C}\exp\{{w_c}^T\phi(x_i)\})})$$

Problem 2 Part A)

MER values	
Model	MER
Standardized Train data	0.091354
Standardized Train data	0.091354
Standardized Test data	0.09570312
Log Trans. Train data	0.07406199
Log Trans. Test data	0.08658854
Binary Train data	08515498
Binary Test data	0.09049479

Part B)



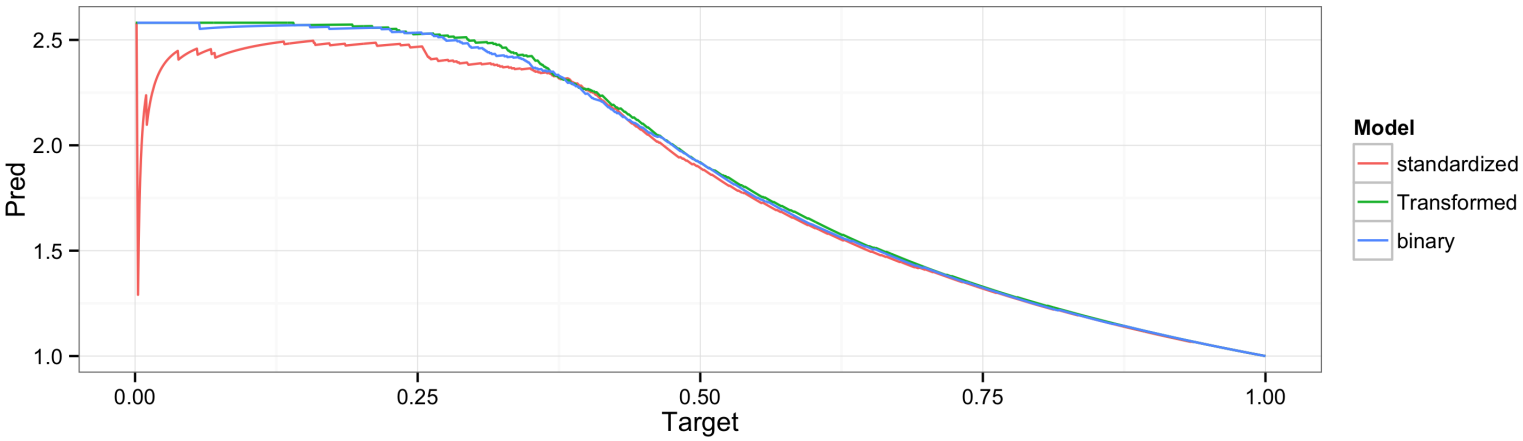
ROC Curve

AUC Values :
Standardized : 0.9518079
Log Transformed : 0.9718153
Binarised : 0.9652274

Area under ROC curve is often used as a measure of quality of a probabilistic classifier. It captures the discriminative ability of the classifier. The closer the AUC value to 1, the better the model. The AUC value does not capture the trade-offs that happen in the performance of the models at various accuracy levels, nor is it able to provide a comparison between the various systems as to in what scenarios it might be beneficial to use which of the model. The given AUC values indicate that the performance of the three models is almost the same. However, the ROC curves show the accuracy levels where the difference in the performance of the three data transformed models actually effect the accuracy of the classifier. As can be seen, the Standardized data set has a lower true positive

rate than the rest, for cases where the false positive rate is low. However, for higher values of for, the Standardized data set behaves in same maker as the other two.

Part C)



Lift Curve

Lift measures the degree to which the predictions made by the model are better than the one generated by a model that randomly predicts. From the graph we can see that the Standardized data set has less of prediction value than the other two models.
Standardized model correctly classifies 60% (0.25*2.4)of the spam mails as spam
Log transformed model correctly classifies 65% (0.25*2.6)of the spam mails as spam
Binarized model correctly classifies 65% (0.25*2.6)of the spam mails as spam

Problem 3 Part A)

Mean Error Rate Train: 0.08543742
Mean Error Rate Test : 0.08813291
Given that in this question we have experimented with Ensembles, we would expect the Mean Error Rate values to reduce. The Mean Error rate is slightly better than what we got in the previous question. The change in the values is not very drastic, which is expected because the errors of all the models would be somewhat co related because the models are inherently the same, but have been trained of different data set. However the variance int he output seems to have reduced because of introduction of bagging in the model.

Part B)
Here the nine models which we have used to create the ensemble are very diverse and we would expect the error values to reduce by a huge factor.
Mean Error Rate Train : 0.05579119
Mean Error Rate Test : 0.06510417

The values have reduced by a third for the ensemble model. However in certain runs, the Mean Error Rate value had even reduced by half. The error value of ensemble is better than the Error values obtained individually from the three types of data set in the previous question. This indicates the power of ensemble models to reduce the overall error rate. As can be seen from the output of the two ensemble approaches, we can say that rather than bagging(introducing randomness in the

models), ensemble of the diverse models returns better results.
However, the Mean Error Rate values are so small that not much judgment can be made on the improvements percentage of the accuracy of the models by introducing ensembles. It seems that this is a simple classification problem and all the models are able to perform very well on the data set. Another reason could be the fact that because of random sampling being done at cross validation in all the cases, we cannot be sure about the percentage improvement in results when the error magnitude is so less.

Problem 4 Part A)

Parameter C (penalty factor) : It is critical here, as in any regularization scheme, that a proper value is chosen for C. If it is too large, we have a high penalty for nonseparable points and we may store many support vectors and overfit(low bias, and high variance). If it is too small, we may have underfitting(high bias, and low variance). The value of C was varied from 0.001 to 1000 in steps and MER values were calculated for them individually. C with minimum MER was chosen : C = 400

Value of C : 0.001
MER for Standardized: 0.1191406
MER for Log Trans. : 0.1014465

Value of C : 100
MER for Standardized: 0.07096354
MER for Log Trans. : 0.0562782

Value of C : 400
MER for Standardized: 0.06835938
MER for Log Trans. : 0.05598958

Value of C : 1000
MER for Standardized: 0.1145833
MER for Log Trans. : 0.0967926

We can also manipulate the class weights so as to decide the ratio of penalty when the a mis classification takes place. If this value is not set, the model tunes the class weights based on the cross validation results. In case they are provided the value of these parameters is not changed.

Part B)
Gaussian radial basis kernel can be tuned with another parameter sigma(the parzen window parameter). Usually the value of this parameter is set to 0.5. However, if no value for this parameter is provided, the ksvm library is able to tune the value of this parameter.
The values chosen was :
Sigma for Standardized model : 0.0303449247393635
Sigma for Log Transformed model 0.0294760226442188

Mean Error Rate values when C is optimized and optimized Sigma value if used.
Mean Error Rate For Standardized data : 0.07552083 (C = 100)
Mean Error Rate for Log transformed data : 0.05403646 (C = 100)

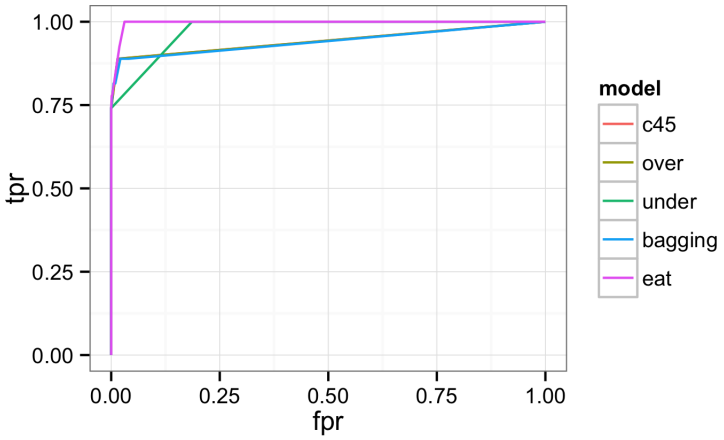
Part C)

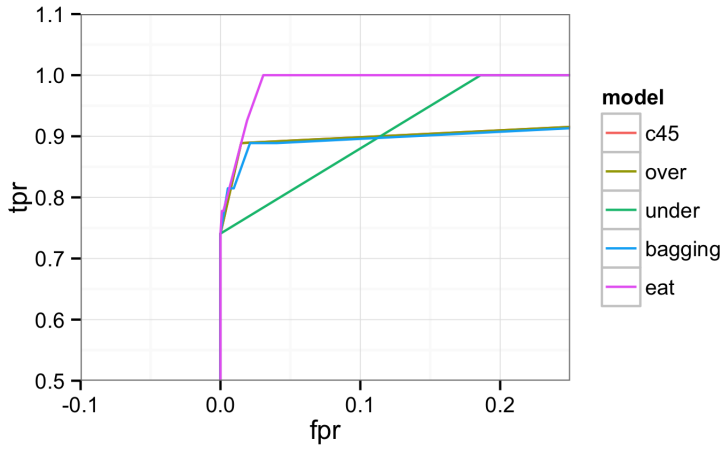
MER values	
Model	MER
Ridge on Standardized	0.09570312
Ridge on Log Transformed	0.08658854
Ridge on Binary	0.09049479
Ensembel Bagging	0.08813291
Ensembel Mix	0.06510417
SVM Standardized	0.06835938
SVM Standardized(RBF)	0.07552083
SVM Log Transformed	0.05598958
SVM Log Transformed(RBF)	0.05403646

Conclusions from the values obtained in the above questions:

From the table, we can make the following conclusions:
Ensembels are always able to achieve better results than individual models.
Ensembles of diverse models, models which are different are able to return better performance than bagging on same model. However, if bagging is performed on models which are diverse, we might be able to get even better results.
SVM's are able to perform as well as ensembles on diverse set of models. However SVM are designed to work for two class problems only.
In this specific problem, changing the kernel did have a significant change in the performance. Which implies that performance of SVM is very much dependent on the choice of kernel.
Every model has performed very well on this problem with Mean Error Rate values always less that 0.1.

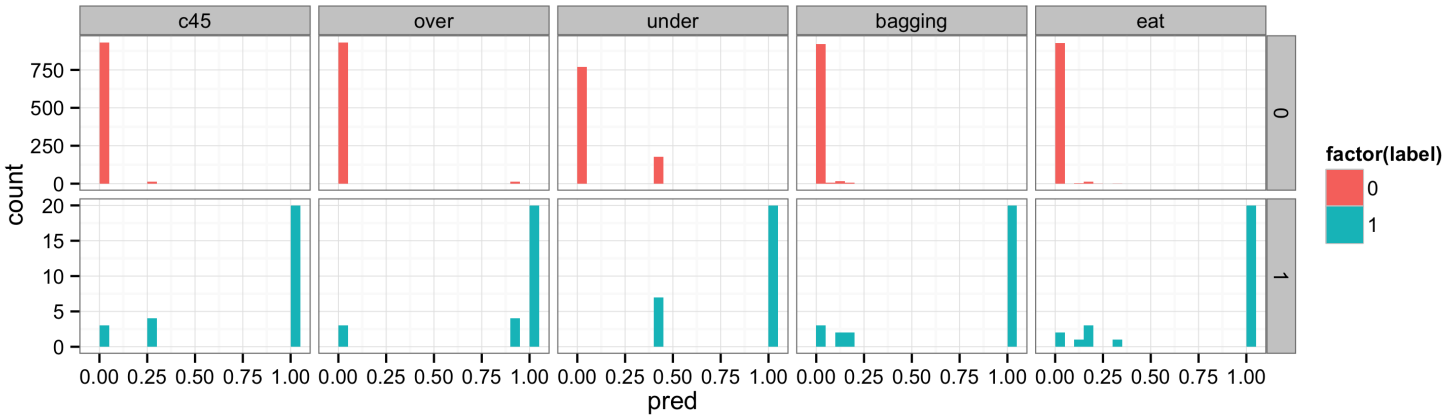
Problem 5 I have made use of ROSE package to achieve under and over sampling. These two operations are performed so as to balance the training data in the models. The idea behind over sampling is to increase the instances of the negative class so as to match the no of instance of the positive class. However the inverse is done in undersampling. Only some of the data points are picked from the positive class training sample so as to match the no of points corresponding to the negative class count.





ROC Curve Zoomed Bagging performs better than EAT. uptil a certain point and then EAT performs better. This could be because of the variation in the alpha tress in EAT help in reduction of bias. In bagging, there is no diversity in the trees, and therefore the errors do net get reduced (correlated errors). Undersampled data is not able to perform as well for the values which belong to the middle range. The AUC value for Under sampled model are one of the best indicating that is very efficiently able to classify the values which are easy to classify. However, for values, which would have needed a bit more complex decision tree, under sampled model is not able to perform well on those data points.

Part c)



Histogram
AUC Values :
C45 : 0.942526
Over sampled : 0.942526
Under sampled : 0.9758829
Bagging : 0.9408817
Eat : 0.9965743

The histogram of Under sampled data indicates lower accuracy because of less no of samples to train on the performance of the model has decreased. When we look at the alpha tree for the under

sample model, we can see that the tree is of mush shorter length. Thus indicating the effect of less training data on the tree that got generated. Undersampled data is not able to perform as well for the values which belong to the middle range. The AUC value for Under sampled model are one of the best indicating that is very efficiently able to classify the values which can be classified in small decisions. This indicates that a smaller tree is better in cases where the co relation between the classes are not much. However, for cases, which would have needed a bit more complex decision tree, under sampled model is not able to perform well on those data points.