Name: Tanay Mukherjee

STA 9891

---

1. Imbalanced data refers to a classification problem where the number of observations per class is not equally distributed. In this question we subsample the IMDB data to create imbalanced data. To subsample the data, keep all the negative observations, but only keep the first 4000 (out of the 12500) of the positive observations. Do this separately for both train and test. We end with 12500 +4000 = 16500 observations separately for training and testing. Use the $p = 2500$ most frequent words as predictors.

   (a) Fit a LASSO logistic regression model to the training data and use 10-fold cross-validation using the AUC as a measure of error to tune λ. For the optimal λ answer the following questions.

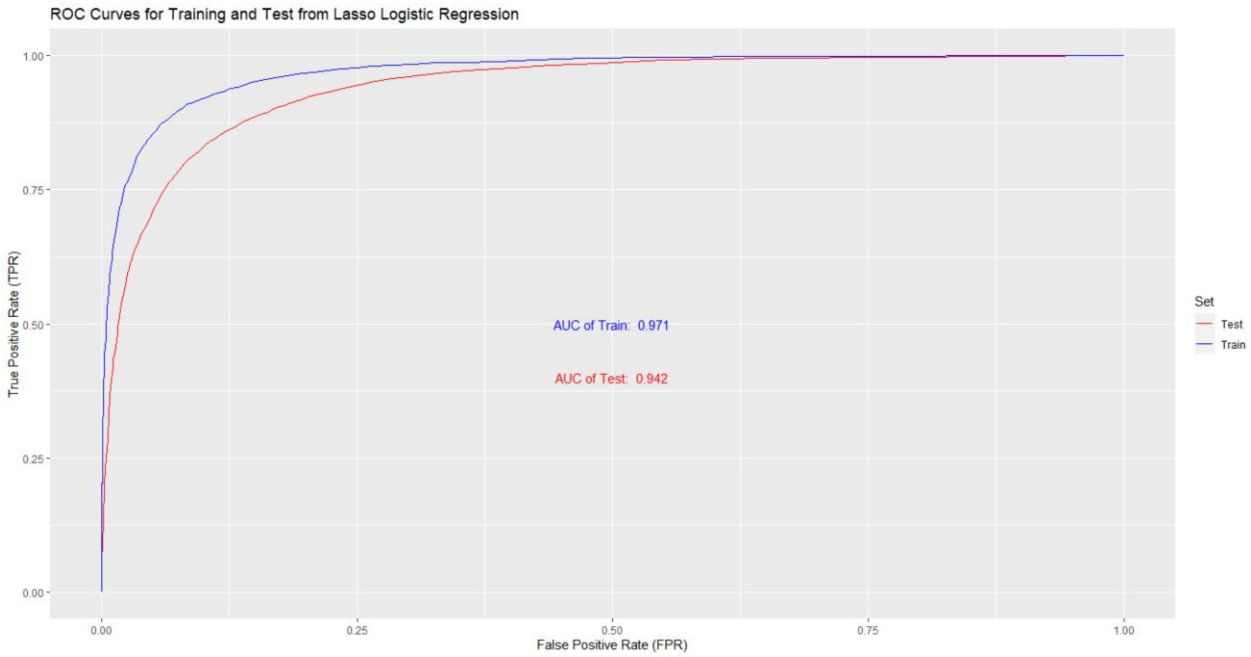
(i) Top 5 words associated with positive reviews are:

noir, wonderfully, 7, excellent, perfect

(ii) Top 5 words associated with positive reviews are:

worst, waste, poorly, badly, awful

(iii) ROC of train and test set:



(iv) Type I and Type II error for theta = 0.5:

1. **For theta = 0.5, the Type I and Type II errors for training set are:**
   a. "Type 1 Error for Training Set: 0.03"
   b. "Type 2 Error for Training Set: 0.21"

2. **For theta = 0.5, the Type I and Type II errors for test set are:**
   a. "Type 1 Error for Test Set: 0.05"
   b. "Type 2 Error for Test Set: 0.28"

(v) Theta for which the type I error is equal (as much as possible) to the type II error:

Set	Theta	Type I Error	Type II Error
Train	0.30	0.09	0.09
Test	0.30	0.11	0.15

(b) Fit a ridge logistic regression model to the training data and use 10-fold cross-validation using the AUC as a measure of error to tune λ. For the optimal λ answer the same questions (i., ii., ...) as asked for LASSO (8 points).

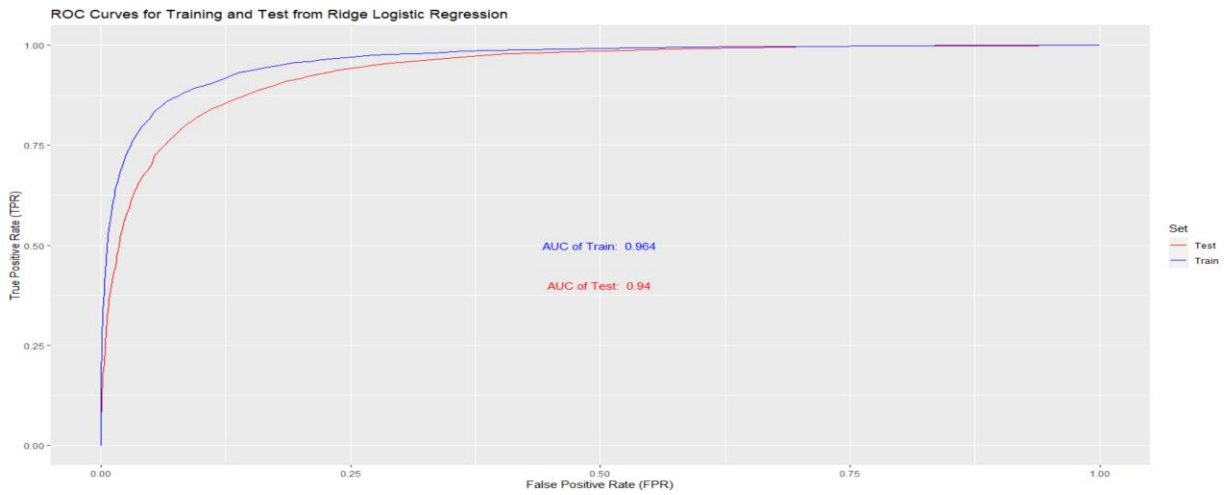
(i) Top 5 words associated with positive reviews are:

excellent, great, perfect, amazing, 7

(ii) Top 5 words associated with positive reviews are:

worst, bad, waste, awful, poor

(iii) ROC of train and test set:



(iv) Type I and Type II error for theta = 0.5:

1. **For theta = 0.5, the Type I and Type II errors for training set are:**
   c. "Type 1 Error for Training Set: 0.02"
   d. "Type 2 Error for Training Set: 0.29"

2. **For theta = 0.5, the Type I and Type II errors for test set are:**
   c. "Type 1 Error for Test Set: 0.04"
   d. "Type 2 Error for Test Set: 0.36"

(v) Theta for which the type I error is equal (as much as possible) to the type II error:

Set	Theta	Type I Error	Type II Error
Train	0.30	0.10	0.10
Test	0.30	0.11	0.16

(c) Fit an Elastic-net logistic regression model to the training data and use 10-fold cross-validation using the AUC as a measure of error to tune λ. For the optimal λ answer the same questions (i., ii., ...) as asked for LASSO (8 points).

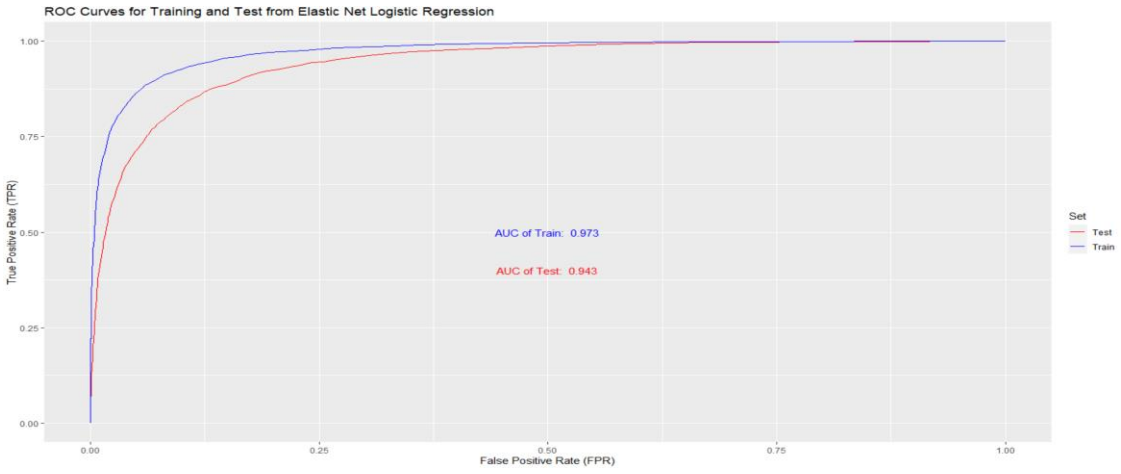
(i) Top 5 words associated with positive reviews are:

noir, excellent, 7, perfect, wonderfully

(ii) Top 5 words associated with positive reviews are:

worst, waste, poorly, awful, badly

(iii) ROC of train and test set:



(iv) Type I and Type II error for theta = 0.5:

1. **For theta = 0.5, the Type I and Type II errors for training set are:**
   e. "Type 1 Error for Training Set: 0.03"
   f. "Type 2 Error for Training Set: 0.20"

2. **For theta = 0.5, the Type I and Type II errors for test set are:**
   e. "Type 1 Error for Test Set: 0.05"
   f. "Type 2 Error for Test Set: 0.28"

(v) Theta for which the type I error is equal (as much as possible) to the type II error:

Set	Theta	Type I Error	Type II Error
Train	0.30	0.09	0.08
Test	0.30	0.11	0.15

The best lambda for each logistic regression model is below:

Model	Lambda
Lasso	0.0002583783
Ridge	0.004310009
Elastic Net	0.0004290

 # Appendix:

I am not sharing the entire code because of the page limit. So, I include only the part which was new to this exercise. Rest of the codes are shared earlier in HW2 and HW3 respectively.

1. For Lasso Logistic Regression

```
> lasso = cv.glmnet(x = X.train, y=y.train, family = "binomial", alpha = 1, i
ntercept = TRUE, standardize = FALSE, nfolds = 10, type.measure="auc")

> lasso_fit = glmnet(x = X.train, y=y.train, lambda = lasso$lambda.min, famil
y = "binomial", alpha = 1, intercept = TRUE, standardize = FALSE)

> beta0.hat = lasso_fit$a0

> beta.hat = as.vector(lasso_fit$beta)
```

2. For Ridge Logistic Regression

```
> ridge = cv.glmnet(x = X.train, y=y.train, family = "binomial", alpha = 0, i
ntercept = TRUE, standardize = FALSE, nfolds = 10, type.measure="auc")

> ridge_fit = glmnet(x = X.train, y=y.train, lambda = ridge$lambda[which.max(
ridge$cvm)], family = "binomial", alpha = 0, intercept = TRUE, standardize =
FALSE)

> beta0.hat = ridge_fit$a0

> beta.hat = as.vector(ridge_fit$beta)
```

3. For Elastic Net Logistic Regression

```
> enet = cv.glmnet(x = X.train, y=y.train, family = "binomial", alpha = 0.5,
intercept = TRUE, standardize = FALSE, nfolds = 10, type.measure="auc")

> enet_fit = glmnet(x = X.train, y=y.train, lambda = enet$lambda[which.max(en
et$cvm)], family = "binomial", alpha = 0.5, intercept = TRUE, standardize = F
ALSE)

> beta0.hat = enet_fit$a0

> beta.hat = as.vector(enet_fit$beta)
```

Rest of the code to fetch the top 5 positive and negative words, to plot the ROC curve, and the Type I and Type II error for different thresholds has been included in previous assignments. There is no change to that code.