Module 2: Supervised learning

Split data into train and test

```
dim(train_x)
```

```
## [1] 106 587
```

```
length(train_y)
```

```
## [1] 106
```

```
dim(test_x)
```

```
## [1]  75 587
```

```
length(test_y)
```

```
## [1] 75
```

LASSO logistic regression

```r
# Choose best lambda using CV
beta_lasso <- lasso_fit(
  x = log(train_x + 1), y = train_y,
  tuning = "cv", family = "binomial"
)
```

```r
# Features Selected
names(beta_lasso[abs(beta_lasso) > 0])[-1]
```

```
##  [1] "COD2"     "COD10"    "NLP1"     "NLP17"    "NLP56"    "NLP82"
##  [7] "NLP93"    "NLP104"   "NLP118"   "NLP130"   "NLP144"   "NLP164"
## [13] "NLP172"   "NLP193"   "NLP199"   "NLP222"   "NLP231"   "NLP265"
## [19] "NLP274"   "NLP280"   "NLP297"   "NLP299"   "NLP346"   "NLP362"
## [25] "NLP375"   "NLP382"   "NLP396"   "NLP401"   "NLP409"   "NLP435"
## [31] "NLP451"   "NLP462"   "NLP488"   "NLP533"   "NLP536"   "NLP552"
## [37] "NLP568"   "main_NLP"
```

ALASSO logistic regression

```
# Fit Adaptive LASSO
beta_alasso <- adaptive_lasso_fit(
  x = log(train_x + 1), y = train_y,
  tuning = "cv", family = "binomial"
)
```

```
# ALASSO features selected
beta_alasso[!beta_alasso == 0][-1]
```

```
##      NLP56      NLP93     NLP104     NLP118     NLP222     NLP231     NLP265
##  0.1966447 -1.0538342 -1.7011315 -1.5489010 -2.0758094  0.3598780 -0.9584738
##     NLP280     NLP297     NLP299     NLP409     NLP536   main_NLP
##  0.6256635 -0.2093127  1.0106695  0.4019735  0.1038460  1.4248803
```

```
# LASSO features selected
beta_lasso[!beta_lasso==0][-1]
```

```
##        COD2       COD10        NLP1       NLP17       NLP56       NLP82
## -0.07891435 -0.07964064 -0.15656996 -0.10698323  0.43476973 -0.14774013
##       NLP93      NLP104      NLP118      NLP130      NLP144      NLP164
## -0.95721897 -1.14198338 -0.83985826 -0.02971022 -0.39607669 -0.13824534
##      NLP172      NLP193      NLP199      NLP222      NLP231      NLP265
##  0.11876041  0.11493486 -0.16297872 -2.01541309  0.40654328 -0.84088955
##      NLP274      NLP280      NLP297      NLP299      NLP346      NLP362
## -0.17839805  0.62463549 -0.54371389  0.86087307 -0.40862069  0.17883546
##      NLP375      NLP382      NLP396      NLP401      NLP409      NLP435
##  0.79214450 -0.47973944 -0.08726960 -0.17450935  0.53175298  0.20241840
##      NLP451      NLP462      NLP488      NLP533      NLP536      NLP552
##  0.61949264 -0.24987822  0.46166193 -0.37801422  0.53979607  0.04623370
##      NLP568    main_NLP
##  0.40970337  1.28008994
```

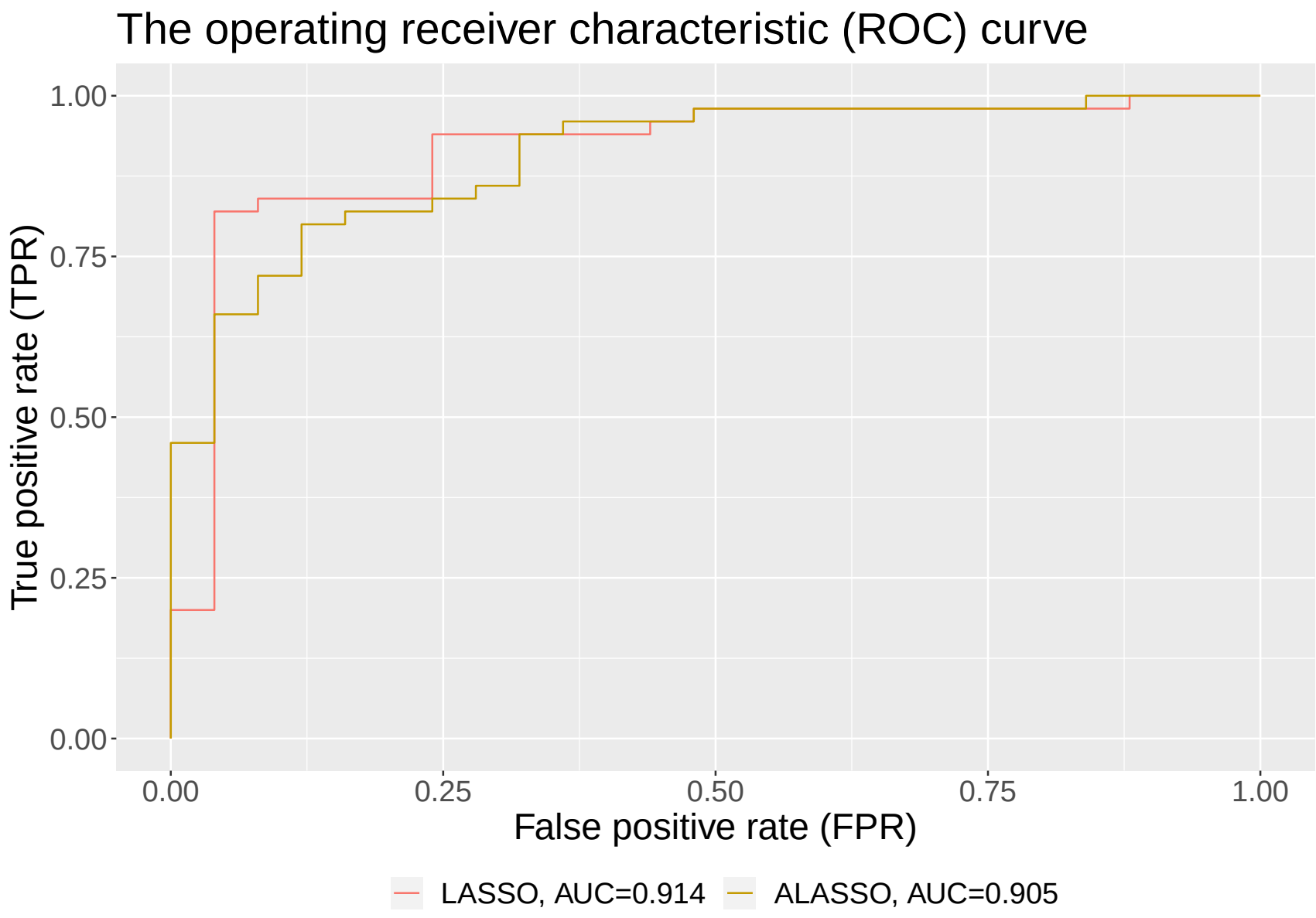
Get model predictions + ROC curve

```r
# Prediction on testing set (LASSO)
y_hat_lasso <- linear_model_predict(
  beta = beta_lasso, x = log(test_x + 1),
  probability = TRUE
)
```

```r
# Prediction on testing set (ALASSO)
y_hat_alasso <- linear_model_predict(
  beta = beta_alasso, x = log(test_x + 1),
  probability = TRUE
)
```

```r
roc_lasso <- roc(test_y, y_hat_lasso)
roc_alasso <- roc(test_y, y_hat_alasso)
# as expected alasso selects less features
```

LASSO vs. ALASSO



LASSO vs. ALASSO at FPR = 0.10

```
roc_full_lasso <- get_roc(y_true = test_y, y_score = y_hat_lasso) %>% data.frame()
get_roc_parameter(0.1, roc_full_lasso)
```

```
##      cutoff  pos.rate FPR  TPR       PPV       NPV        F1
## 1 0.8573308 0.5933333 0.1 0.84 0.9438202 0.7377049 0.8888889
```

```
roc_full_alasso <- get_roc(y_true = test_y, y_score = y_hat_alasso) %>% data.frame()
get_roc_parameter(0.1, roc_full_alasso)
```

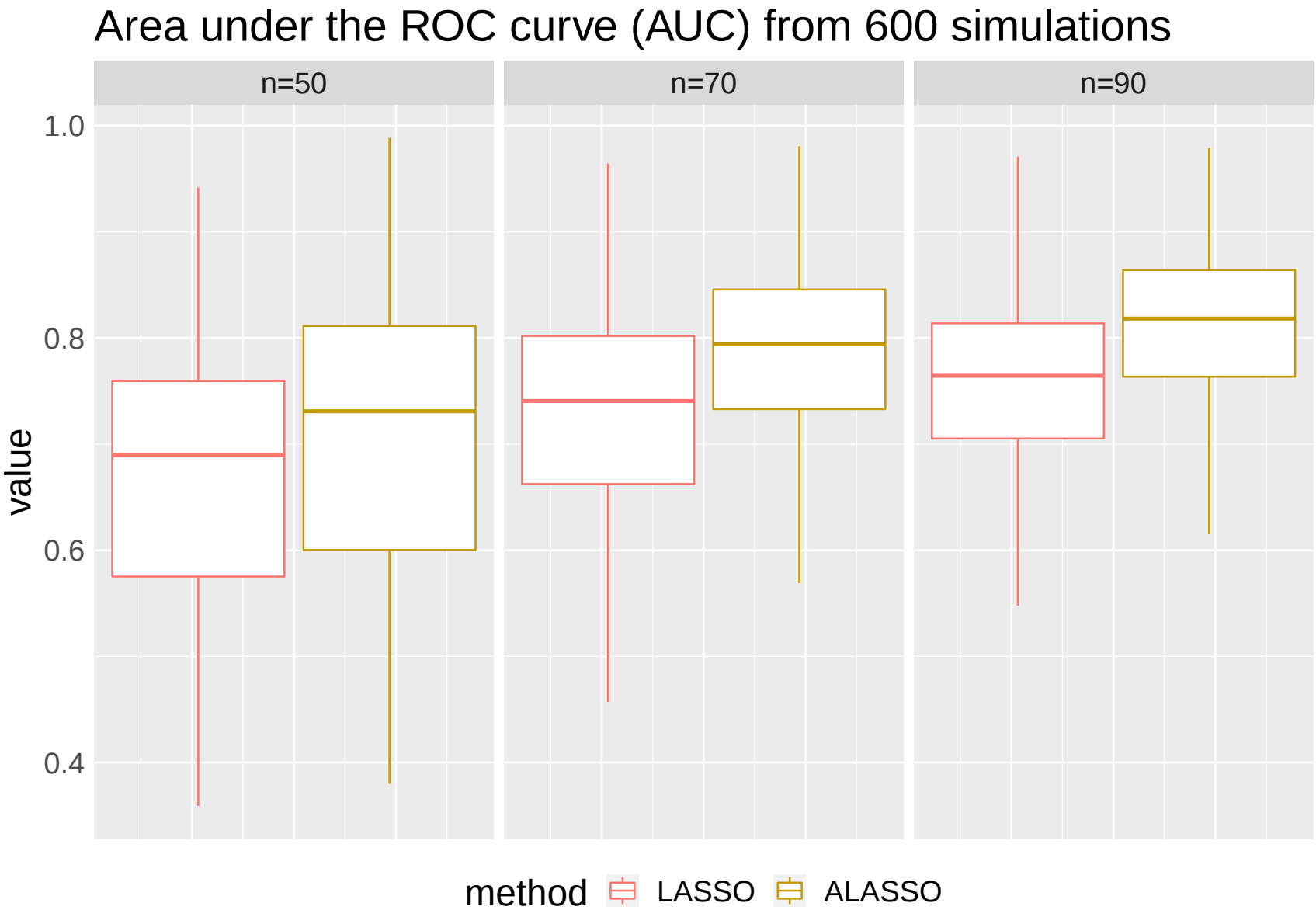
```
##      cutoff  pos.rate FPR  TPR       PPV       NPV        F1
## 1 0.9228109 0.5133333 0.1 0.72 0.9350649 0.6164384 0.8135593
```

LASSO vs. ALASSO with different training set size

- Randomly sample training size $= 50, 70, 90$
- Use the remaining data as the test set
- Repeat 600 times

```
auc_supervised <- validate_supervised(
  dat = labeled_data, nsim = 600,
  n.train = c(50, 70, 90)
)
```

LASSO vs. ALASSO with different training set size

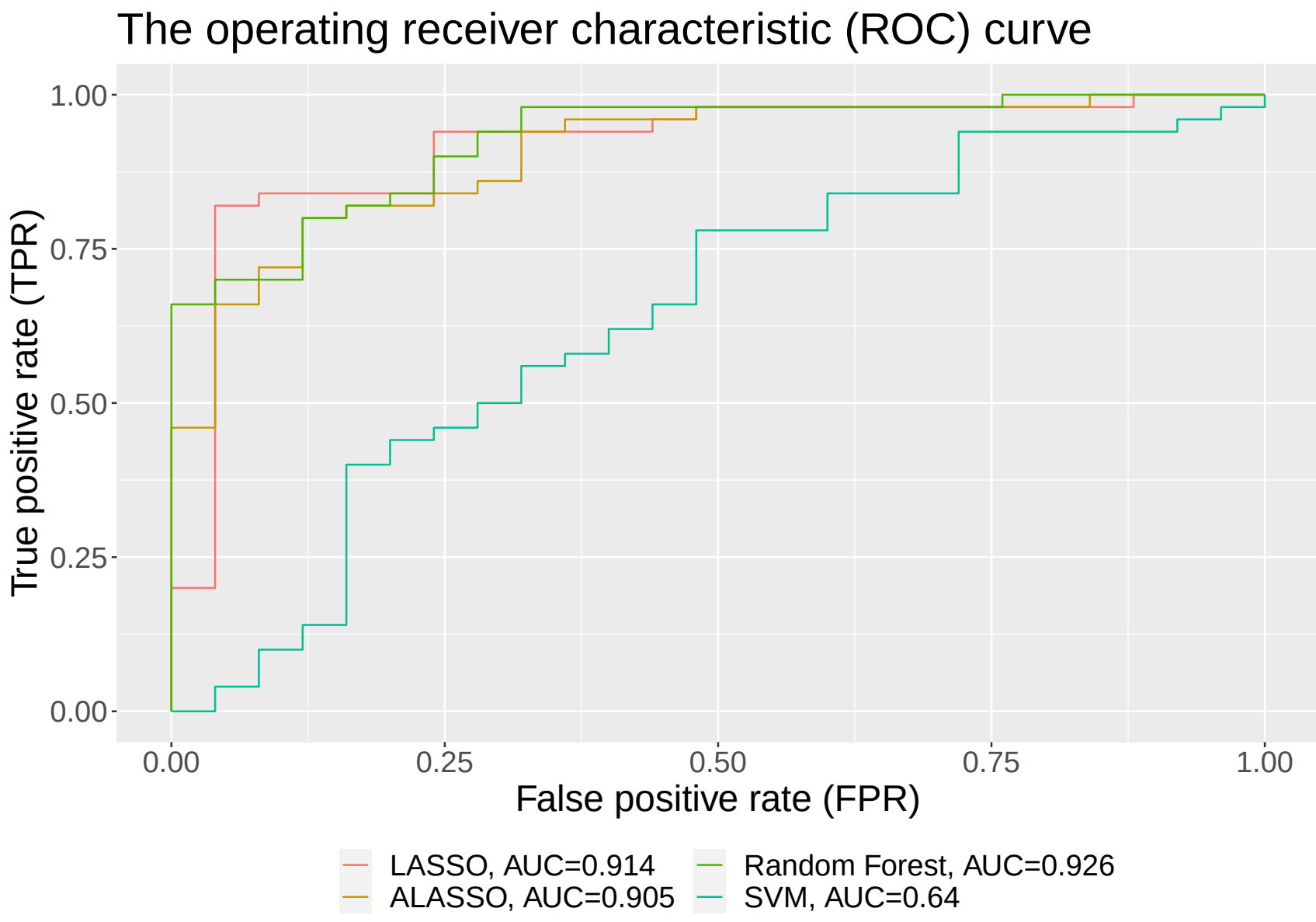


Random Forest and SVM

```r
# Random forest
model_rf <- rfsrc(y ~ ., data = data.frame(y = train_y, x = train_x))
y_hat_rf <- predict(model_rf,
  newdata = data.frame(x = test_x)
)$predicted
roc_rf <- roc(test_y, y_hat_rf)
```

```r
# SVM
model_svm <- SVMMaj::svmmaj(X = train_x, y = train_y)
y_hat_svm <- predict(model_svm, test_x)
roc_svm <- roc(test_y, y_hat_svm)
```

ROC curves


The operating receiver characteristic (ROC) curve
True positive rate (TPR)
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
False positive rate (FPR)
LASSO, AUC=0.914
ALASSO, AUC=0.905
Random Forest, AUC=0.926
SVM, AUC=0.64