Develop a Logistic Regression model to classify the LeaveOrNot event from the other variables.

a) Create a logistic regression model and explain the significant odds ratios in terms of LeaveOrNot.

b) Create a confusion matrix and explain how well the model is classifying the Leaves Company in 2 years events.

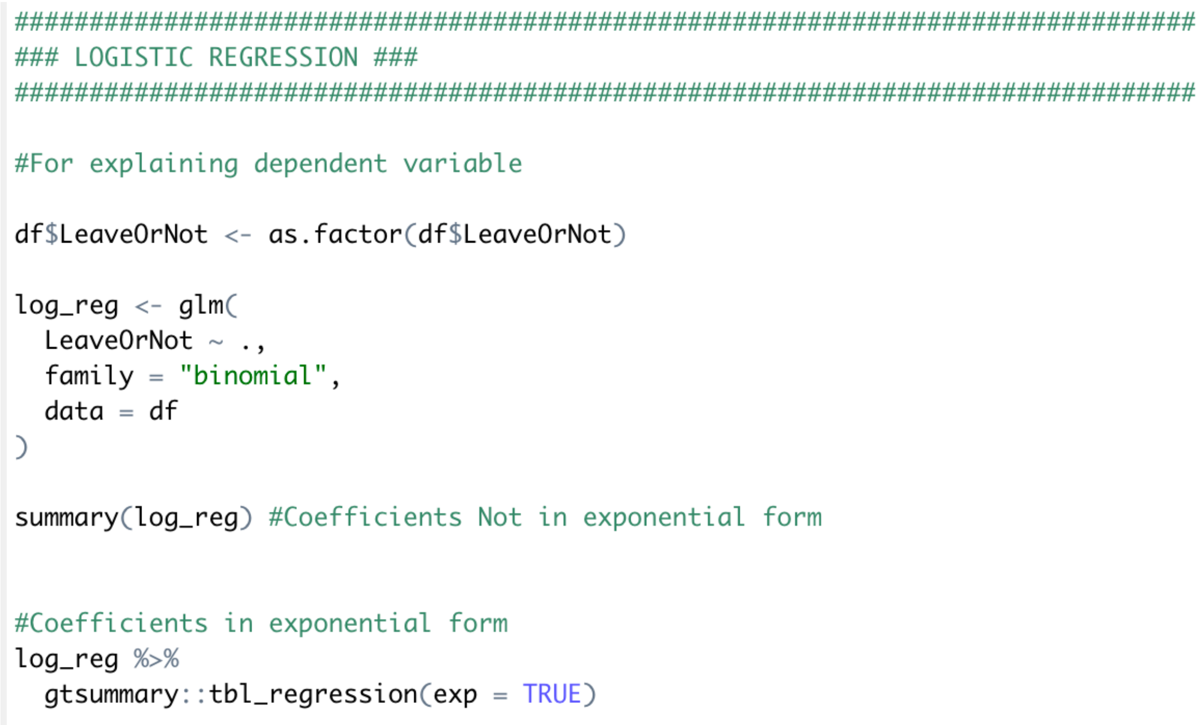
c) Create an ROC curve and calculate the c-statistic (auc). What does this mean about the model?

d) What are the differences between the information in part a and part b?

e) How does this model differ from the linear discriminate analysis you ran in Assignment 7?

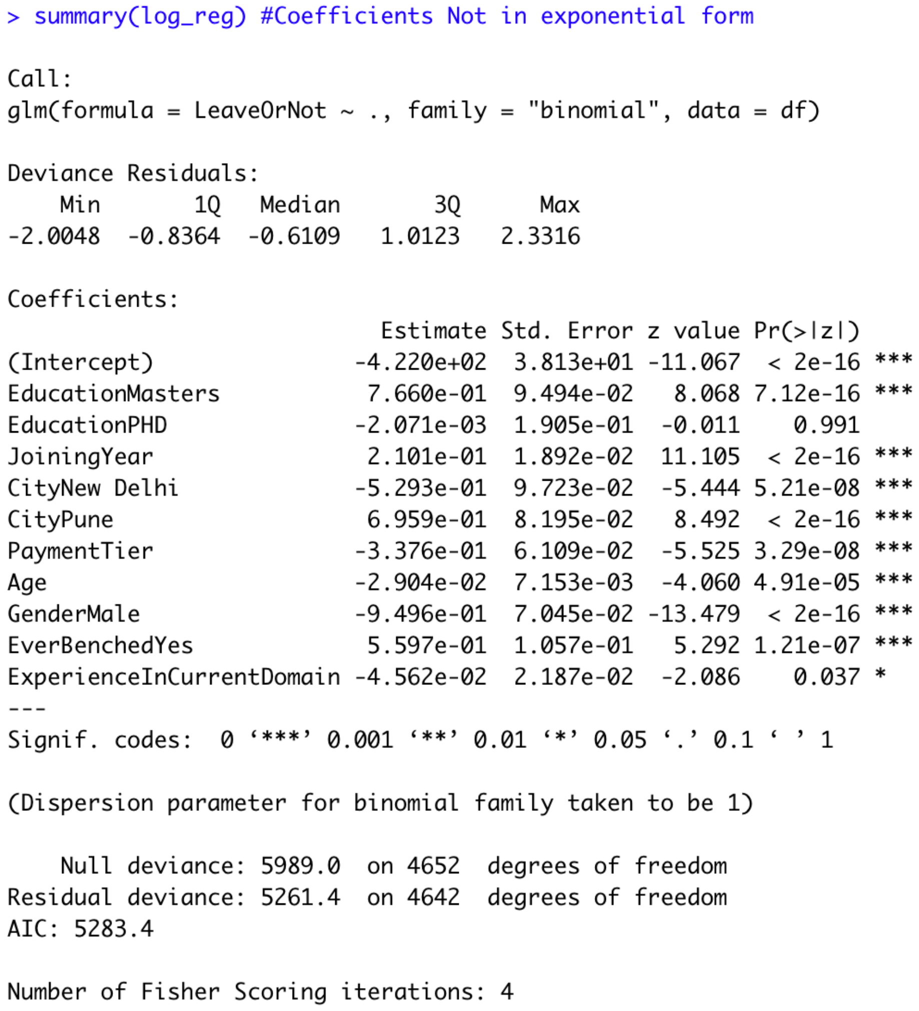
Answers:

a. **Building the logistic regression model**

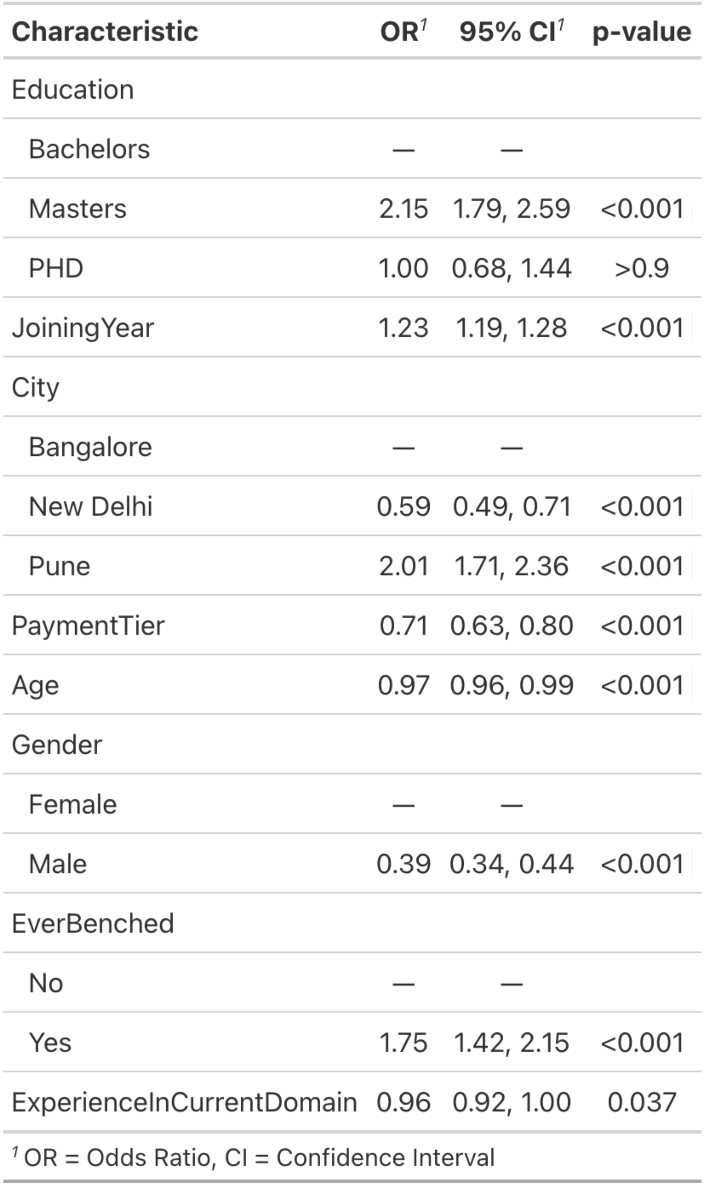


```
###############################################################################
### LOGISTIC REGRESSION ###
###############################################################################

#For explaining dependent variable

df$LeaveOrNot <- as.factor(df$LeaveOrNot)

log_reg <- glm(
  LeaveOrNot ~ .,
  family = "binomial",
  data = df
)

summary(log_reg) #Coefficients Not in exponential form


#Coefficients in exponential form
log_reg %>%
  gtsummary::tbl_regression(exp = TRUE)
```

Output:



```
> summary(log_reg) #Coefficients Not in exponential form

Call:
glm(formula = LeaveOrNot ~ ., family = "binomial", data = df)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.0048  -0.8364  -0.6109   1.0123   2.3316  

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)    
(Intercept)               -4.220e+02  3.813e+01 -11.067  < 2e-16 ***
EducationMasters           7.660e-01  9.494e-02   8.068 7.12e-16 ***
EducationPHD              -2.071e-03  1.905e-01  -0.011    0.991    
JoiningYear                2.101e-01  1.892e-02  11.105  < 2e-16 ***
CityNew Delhi             -5.293e-01  9.723e-02  -5.444 5.21e-08 ***
CityPune                   6.959e-01  8.195e-02   8.492  < 2e-16 ***
PaymentTier               -3.376e-01  6.109e-02  -5.525 3.29e-08 ***
Age                       -2.904e-02  7.153e-03  -4.060 4.91e-05 ***
GenderMale                -9.496e-01  7.045e-02 -13.479  < 2e-16 ***
EverBenchedYes             5.597e-01  1.057e-01   5.292 1.21e-07 ***
ExperienceInCurrentDomain -4.562e-02  2.187e-02  -2.086    0.037 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5989.0  on 4652  degrees of freedom
Residual deviance: 5261.4  on 4642  degrees of freedom
AIC: 5283.4

Number of Fisher Scoring iterations: 4
```



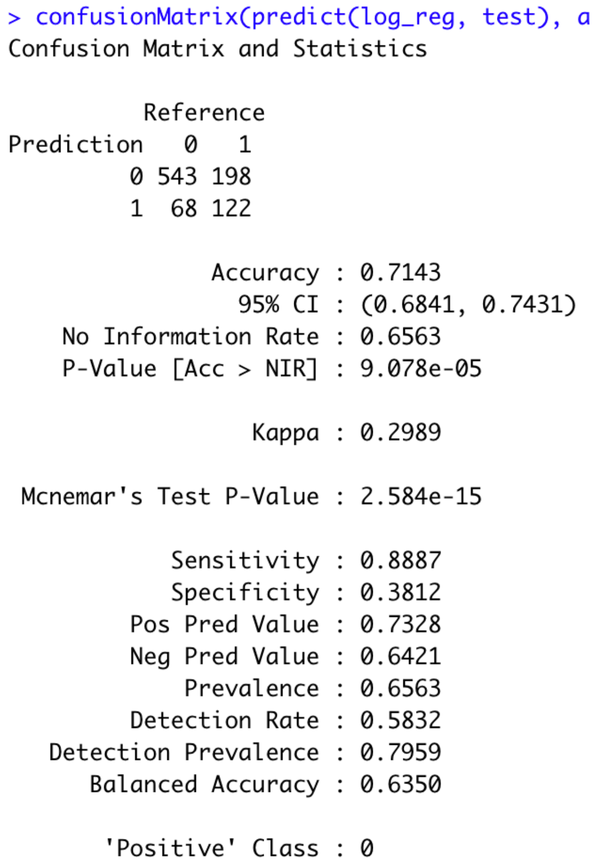
Characteristic	OR[1]	95% CI[1]	p-value
Education			
Bachelors	—	—	
Masters	2.15	1.79, 2.59	<0.001
PHD	1.00	0.68, 1.44	>0.9
JoiningYear	1.23	1.19, 1.28	<0.001
City			
Bangalore	—	—	
New Delhi	0.59	0.49, 0.71	<0.001
Pune	2.01	1.71, 2.36	<0.001
PaymentTier	0.71	0.63, 0.80	<0.001
Age	0.97	0.96, 0.99	<0.001
Gender			
Female	—	—	
Male	0.39	0.34, 0.44	<0.001
EverBenched			
No	—	—	
Yes	1.75	1.42, 2.15	<0.001
ExperienceInCurrentDomain	0.96	0.92, 1.00	0.037

[1] OR = Odds Ratio, CI = Confidence Interval

Explaining odds ratio in terms of LeaveOrNot:

- A person with a Masters degree has a greater odd of Leaving the company (2.15)
- A person with PhD is non significant to the problem (significance is >0.9)
- With every one unit increase in JoiningYear, the odds of Leaving the company is higher (1.23).
- A person from New Delhi has the odds of not leaving is 0.59
- A person from Pune has the odds of leaving as 2.01 (high).
- With every one unit increase in PaymentTier, the odds of not leaving is 0.71
- With every one unit increase in age, the odds of not leaving is 0.97
- Male are associated with increased chances of not leaving
- People who have been benched have a greater odds of leaving (1.75).
- People with experience in the domain have higher chances of not leaving

b. Confusion Matrix:

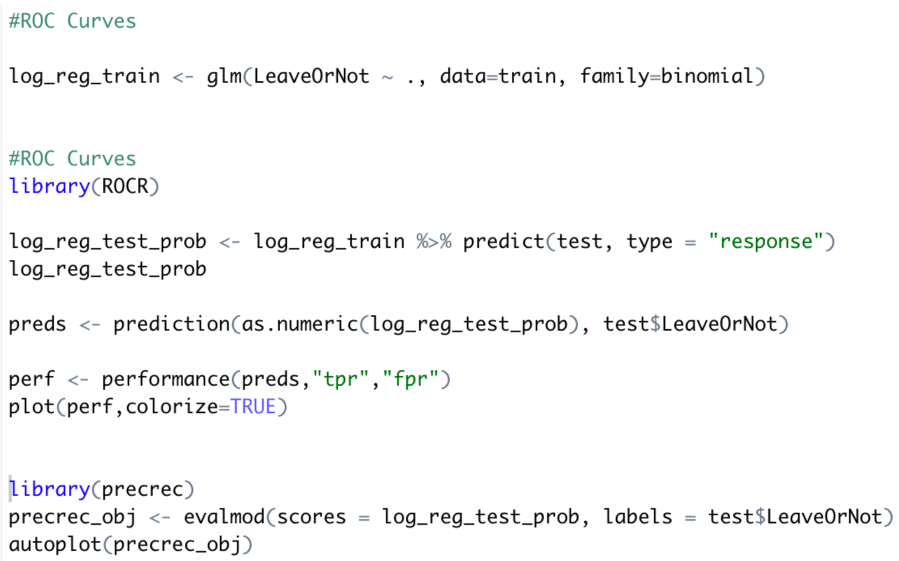


```
> confusionMatrix(predict(log_reg, test), as.factor(test$LeaveOrNot))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 543 198
         1  68 122

               Accuracy : 0.7143
                 95% CI : (0.6841, 0.7431)
    No Information Rate : 0.6563
    P-Value [Acc > NIR] : 9.078e-05

                  Kappa : 0.2989

 Mcnemar's Test P-Value : 2.584e-15

            Sensitivity : 0.8887
            Specificity : 0.3812
         Pos Pred Value : 0.7328
         Neg Pred Value : 0.6421
             Prevalence : 0.6563
         Detection Rate : 0.5832
   Detection Prevalence : 0.7959
      Balanced Accuracy : 0.6350

       'Positive' Class : 0
```

543 samples of “not leaving” have been correctly predicted and 68 samples have been wrongly classified as “leaving” out of a total (534+68) = 602 samples of “not leaving” (0). 122 samples of “leaving” have been correctly classified and 198 samples of “leaving” have been predicted as “not leaving”. Accuracy of the model comes to 71.43% on the test set.

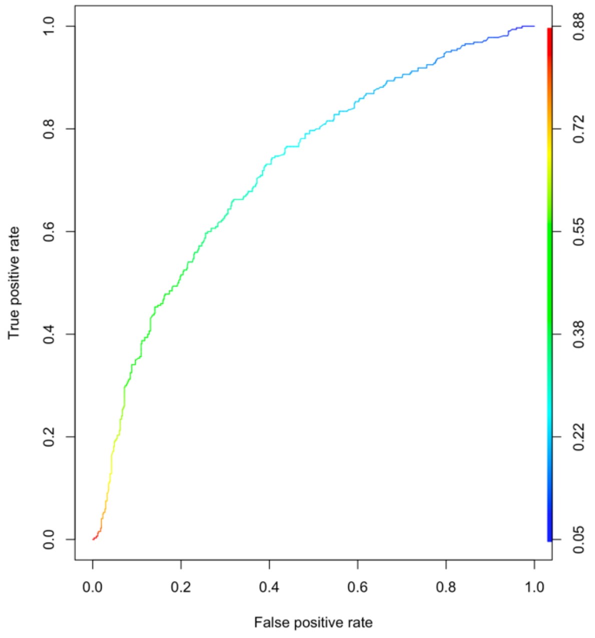
c. ROC Curve and AUC

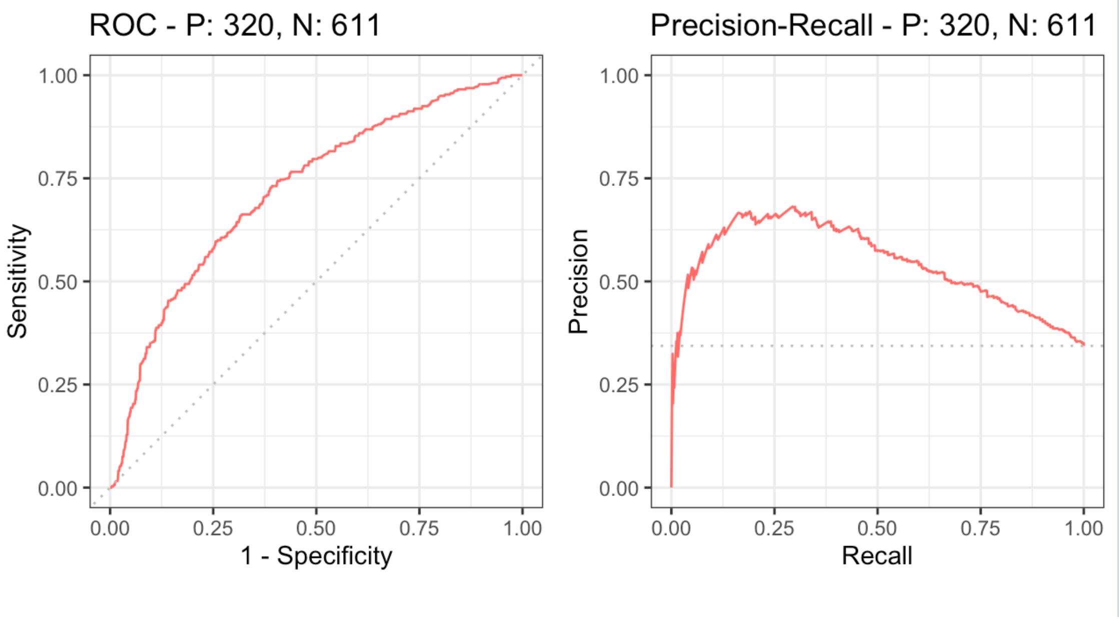
Code:



```
#ROC Curves

log_reg_train <- glm(LeaveOrNot ~ ., data=train, family=binomial)


#ROC Curves
library(ROCR)

log_reg_test_prob <- log_reg_train %>% predict(test, type = "response")
log_reg_test_prob

preds <- prediction(as.numeric(log_reg_test_prob), test$LeaveOrNot)

perf <- performance(preds,"tpr","fpr")
plot(perf,colorize=TRUE)


library(precrec)
precrec_obj <- evalmod(scores = log_reg_test_prob, labels = test$LeaveOrNot)
autoplot(precrec_obj)
```

Output:

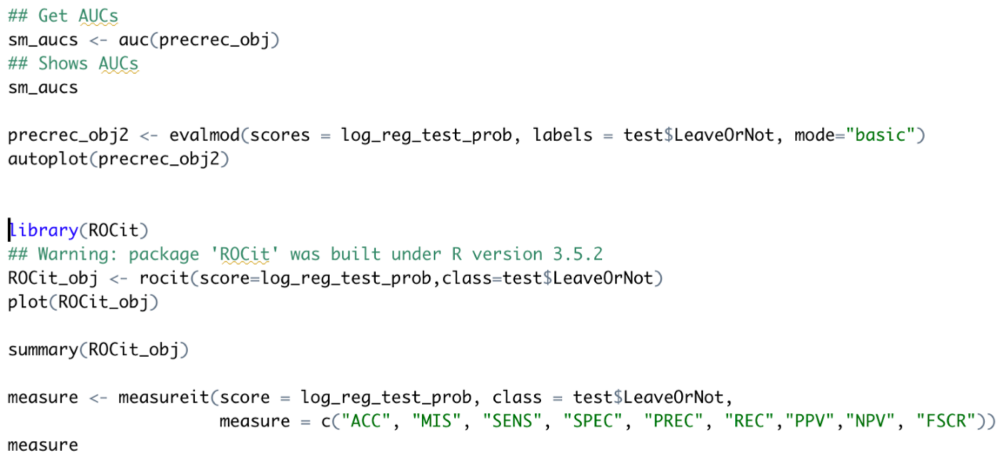




Interpreting the ROC Curve: these curves show the tradeoff between sensitivity (true positive rate) and specificity (1 - false positive rate). If the curve is closer to the top left (1.00) it indicates better performance. We can say this classifier doesn't perform too well based on that, since we're looking for a classifier that has high true positive rate while having low false positive rate. But here we can see that the false positive rate is high, hence we can say the model is not performing too well.

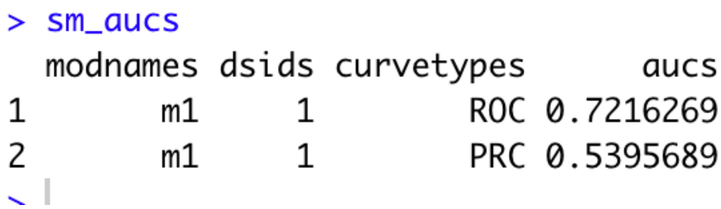
AUC Curves:

Code:



```
## Get AUCs
sm_aucs <- auc(precrec_obj)
## Shows AUCs
sm_aucs

precrec_obj2 <- evalmod(scores = log_reg_test_prob, labels = test$LeaveOrNot, mode="basic")
autoplot(precrec_obj2)


library(ROCit)
## Warning: package 'ROCit' was built under R version 3.5.2
ROCit_obj <- rocit(score=log_reg_test_prob,class=test$LeaveOrNot)
plot(ROCit_obj)

summary(ROCit_obj)

measure <- measureit(score = log_reg_test_prob, class = test$LeaveOrNot,
                     measure = c("ACC", "MIS", "SENS", "SPEC", "PREC", "REC","PPV","NPV", "FSCR"))
measure
```

Output:



```
> sm_aucs
  modnames dsids curvetypes      aucs
1       m1     1        ROC 0.7216269
2       m1     1        PRC 0.5395689
>
```

- AUC stands for Area Under the Curve. Generally, higher the AUC better the performance of the classifier.
- AUC for the ROC curve is 0.7216. In general we can say that if AUC falls between 0.5 and 1 the classifier is doing better than a classifier that just randomly guesses (which will have AUC < 0.5). A perfect classifier will have AUC = 1.0. An AUC of 0.7216 is closer to 0.5 than to 1, and so we can say the classifier is not too good.
- AUC for the Precision-Recall Curve (PRC) comes to 0.5395, which is closer to 0.5 than to 1. Based on these two values, we can say our classifier is not performing well on the test dataset.

d. What are the differences between the information in part a and part b?

Part a talks about the model performance and how each independent variable in the model affects the dependent variable (LeaveOrNot) – if it's in a positive or negative way, whether it's significant and also it's magnitude. Part a outputs the most significant variables that influence the independent variables values and also gives the odds ratio.

Part b talks about how well the model generalizes on unseen data (the test set) by giving us a how many samples are classified right and how many are misclassified. **Part a** focusses on results based only on the training data and Part b talks about model performance based on test/unseen data. Part b gives us information about model performance on new data – including false positive rate, false negative, true positive and true negative rates.

e. How does this model differ from the linear discriminate analysis you ran in Assignment 7?

Both of them – Binary Logistic Regressor and a Linear Discriminant Analyzer – are classifiers (in this case, the focus is on explaining dependent LeaveOrNot in terms of the independent variables). The way both of them work is different. Compared to the Logistic Regression model, the LDA model performed better with an accuracy of ~95% on the test set compared to ~72% accuracy with the logistic regressor.

Logistic Regressions dependent variable is binary(0/1) but in LDA can have more than 2 levels. LDA assumes that the data levels follow multivariate normal distributions where as logistic regression assumes linearity in the data. The parameters in logistic regression are estimated via maximum-likelihood. LDA maximizes the component axes for better class separation.