1 Data Background

Customer Churn defines the rate at which customers stop using a company's services and move on to other options. It has been proven that acquiring new customers is more expensive to a company than retaining existing customers. Therefore, it must be every company's goal to prevent churn. In our analysis, we look at data from a telecommunications company and aim to understand the factors that influence the customer's churning behavior the most.

2 Exploratory Data Analysis

- The data consists of 7043 observations of 21 variables
- Data type: *Churn* is the response variable with 2 levels "No" and "Yes".
  MonthlyCharges, *TotalCharges* are of type double.
  tenure is of type integer.
  Other variables are categorical with 2-4 levels.
  customerID has a unique character string for each 7043 observations.
- Missing values: Number of missing values in the dataset are 11. During data exploration it was discovered that all 11 values occur in the *TotalCharges* column.
- Correlations between numerical variables was checked. *TotalCharges* and *tenure* had a high positive correlation (0.82) suggestive of collinearity.

Barplots of categorical variables plotted against the frequency of *Churn* to visualize if different factor levels have any effect on *Churn*

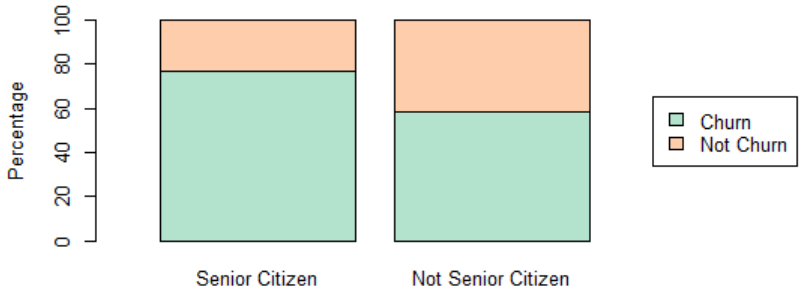

Figure 1: Senior citizens are expected churn more than non-senior citizens.

From looking at barplots of some predictors it becomes evident that marginally their levels may not have a significant effect on probability of *Churn*.

The distribution of some 3 level categorical variables was almost identical. For a parsimonious model, one predictor could be chosen to be included in the model instead of all of them.

After estimating the coefficients, we can check back with the above empirical observations based on the barplots.
Online Security and *Tech Support* seem to have an impact on churning behaviour as also indicated by Lasso. However, barplots for *Streaming TV* and *Streaming Movies* do not justify why these two predictors were picked by lasso.

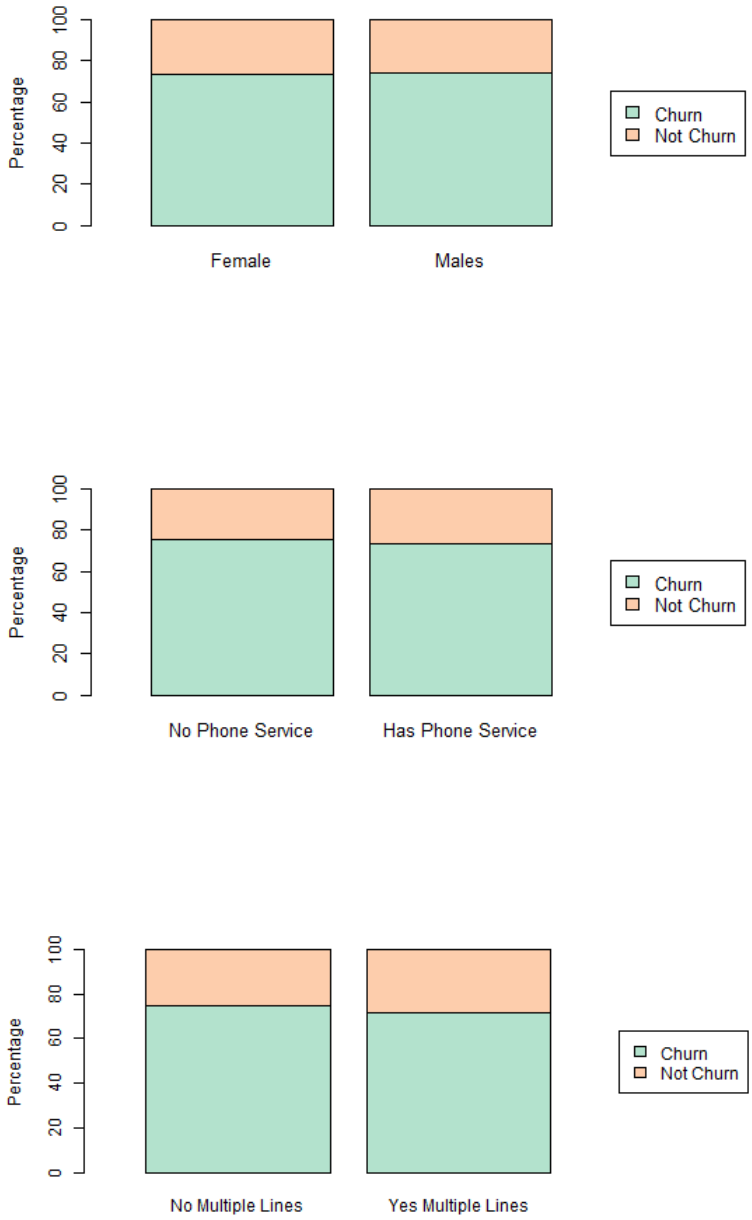

Figure 2: Predictors that marginally do not appear to affect the Churn occurrences: gender, phone service and type of service (multiple vs. single).

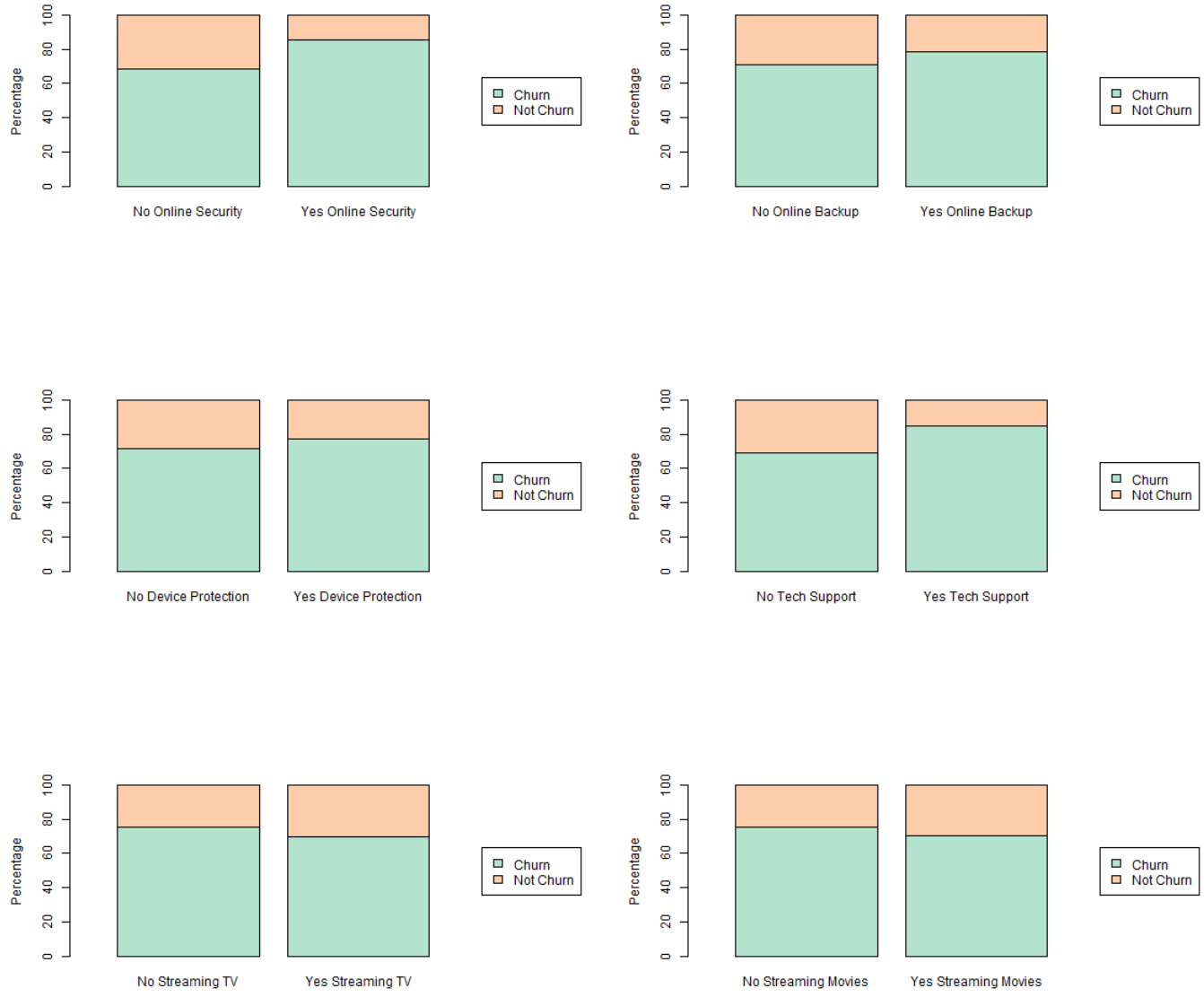

Figure 3: Predictors with three levels that could be reduced to two levels.

Figure 4: Categorical variables which seem to influence churn significantly: Fiber optic group has the least chance of churning; E-check have least churn occurrences; paperless billing have less churn occurrences.

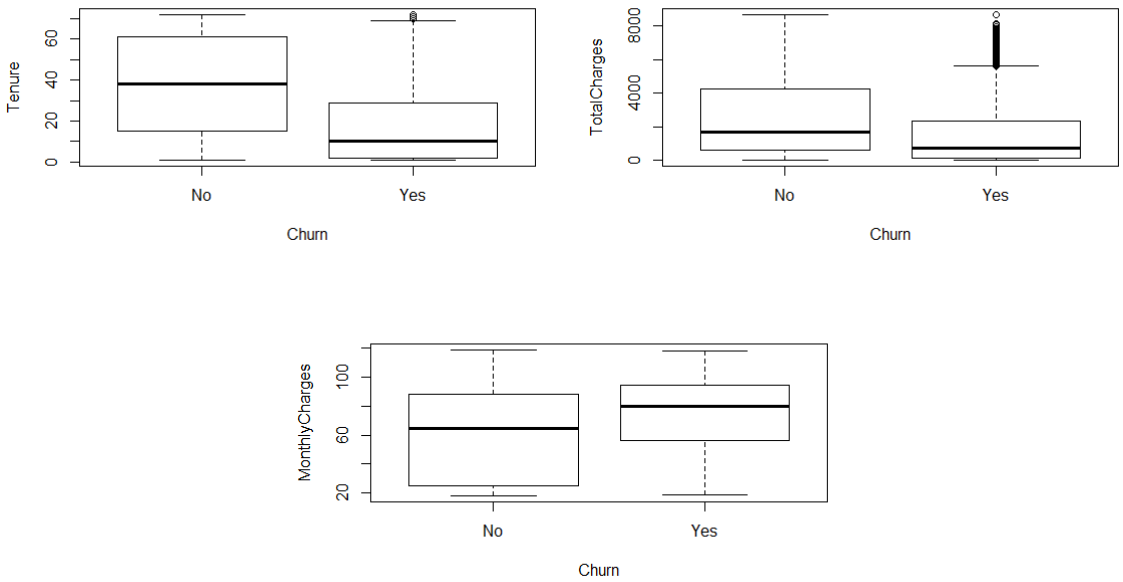

Figure 5: The predictor tenure seems to significantly affect the churn occurrences (larger tenure leads to less churn); monthly charges also seem to affect churn (higher monthly charges leads to larger churn rates).

- Rows containing missing values were removed as it only amounts to about 0.15% missing out of 7043 subjects.
- To use the glmnet() function, x needed to be a matrix and not a dataframe. Therefore, the function model.matrix() was used to create a matrix of predictors. The model.matrix() function automatically transformed qualitative variables into dummy variables, keeping all encoded levels but one from each variable to avoid the dummy variable trap.
- *customerID* was excluded from the x matrix because it was unique for every entry and hence not a predictor of churn.
- Did not perform standardization as glmnet() scales variables automatically. This is useful in comparing the regression coefficient estimates.
- The model was trained on 80% of the dataset and tested on the rest. 10-fold cross validation was also performed multiple times to measure miss-classification error.

3 Results

- The penalized logistic regression with elastic-net penalty function was used for simultaneous parameter estimation and model selection. The Lasso forces some coefficients to be exactly zero and essentially facilitates variable selection. For alpha = 0.95, the variables with non-zero coefficients are:
  SeniorCitizen, DependentsYes, tenure, MultipleLinesYes, InternetServiceFiber.optic, InternetServiceNo, OnlineSecurityYes, TechSupportYes, StreamingTVYes, StreamingMoviesYes, ContractOne.year, ContractTwo.year, PaperlessBillingYes, PaymentMethodElectronic.check
- The prediction accuracy calculated in terms of the misclassification error came out to be 0.81449. Hence, the model correctly predicted the churning behaviour 81.45% of the time.
- The optimal value of the tuning parameter "by one standard error rule" is 0.01518.
- Logistic regression was performed on the subset of variables selected by the elastic net regression. 10 fold cross validation was performed. For a threshold of 0.5, the prediction accuracy calculated in terms of the misclassification error rate came out to be 0.80304.

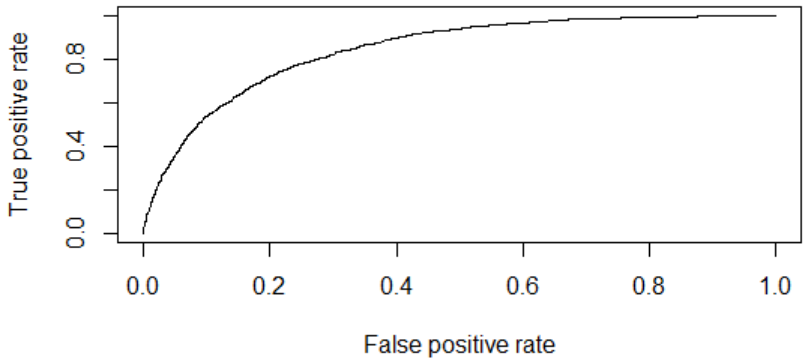

Figure 6: ROC Curve

- The optimal AUC upon varying the threshold was 0.8448.

4 Cost Analysis

The cost of acquiring a new customer in the telecommunication industry is approximately $315. It is estimated that acquiring a customer is almost 5 times costlier than retaining one. By this logic, a company would have to pay $63 to retain an existing customer.
The logistic regression model was used to predict probabilities of churning for each customer. For ease of analysis, these probabilities were transformed into the same factor levels as the response variable, "Churn" and "Not Churn" for a chosen threshold. Since our goal was to reduce the *cost per customer* to the company, optimal threshold would be the one that minimizes cost.
A contingency table for every threshold was constructed using predicted responses and the actual churn response. The four distinct possibility in each contingency table were:

1. True Negative (TN): Model predicts customer would not churn and they actually do not. The cost in this case is zero.

2. True Positive (TP): Model predicts customer to churn and they actually do. The cost in this case is $63.

3. False Positive (FP): Model predicts customer to churn and they actually do not. The cost in this case is $63.

4. False Negative (FN): Model predicts customer would not churn when actually they do. The cost in this case is $315.

$$\text{Cost Function} = 0(\text{TN})+63(\text{TP}+\text{FP})+315(\text{FN})$$

The cost function was evaluated for different thresholds and the optimal value that minimized cost was 0.2.

Cost savings per customer using optimized threshold (0.2) compared to standard threshold (0.5) was $10.34. For a customer base of 1 million, total savings would be $10,374,573.

5 Survival Analysis

Survival analysis is used to analyze data in which the time until an event is of interest. Unlike ordinary least squares regression, survival analysis takes into account information about censoring. Our aim was to model how long customers stay with the company using the variable *tenure* and censoring variable *Churn*. *tenure* served as the "event time" when customer churn did not take place (*Churn*="No") and

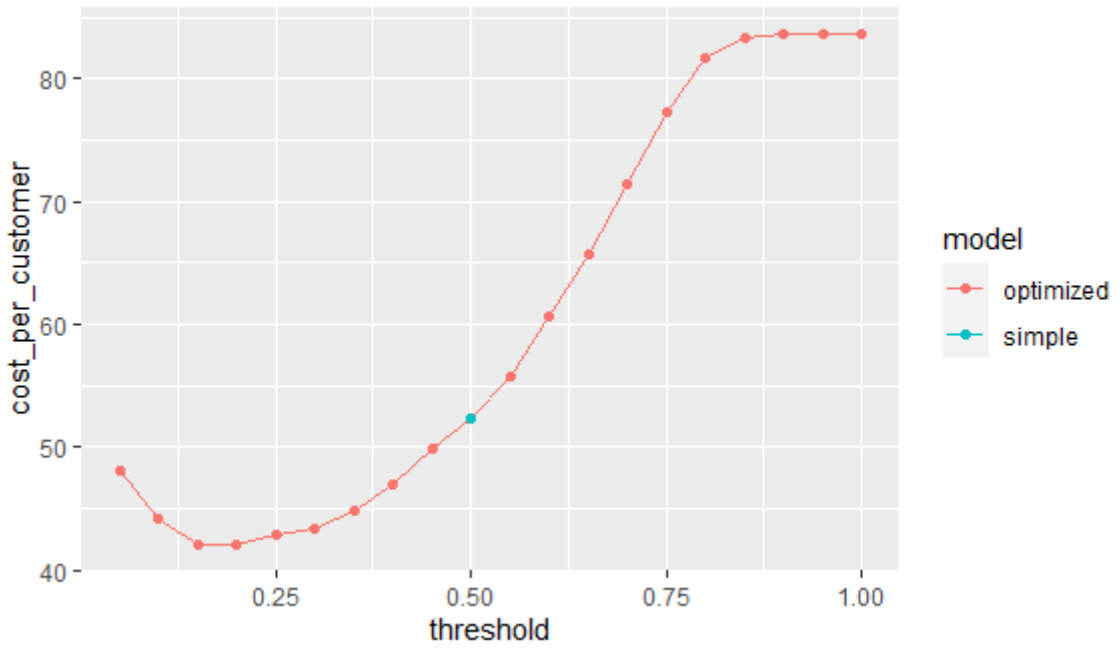

Figure 7: Caption

was right censored time when customer churn took place (*Churn*="Yes"). A Cox Proportional Hazard Model was used to analyze the data. The Cox PH model fits the data to a hazard function of the form:

$$h(t, x_i) = h_0 exp^(\beta^T x_i) \tag{1}$$

The model was fit using the glmnet function and alpha = 0.95. k fold cross validation was performed using cv.glmnet to find the optimal value of lambda.

genderMale	.
SeniorCitizen	.
PartnerYes	-0.158520687
DependentsYes	-0.051732426
PhoneServiceYes	0.003684166
MultipleLinesYes	.
InternetServiceFiber optic	0.709112871
InternetServiceNo	-1.130269955
OnlineSecurityYes	-0.194520772
OnlineBackupYes	-0.046151976
DeviceProtectionYes	.
TechSupportYes	-0.054002495
StreamingTVYes	0.051968531
StreamingMoviesYes	0.048667913
ContractOne year	-0.960245913
ContractTwo year	-2.583566694
PaperlessBillingYes	0.096551356
PaymentMethodCredit card (automatic)	-0.035628318
PaymentMethodElectronic check	0.281542476
PaymentMethodMailed check	0.188619416
MonthlyCharges	0.019896628
TotalCharges	-0.001119960

After the coefficients were estimated, the quantities $exp(b_i)$ known as hazard ratios were estimated. Hazard ratios greater than 1 are positively associated with greater hazard and therefore lower survival. *PhoneServiceYes, InternetServiceFiberOptic, StreamingTVYes, StreamingMoviesYes, Paperless-*

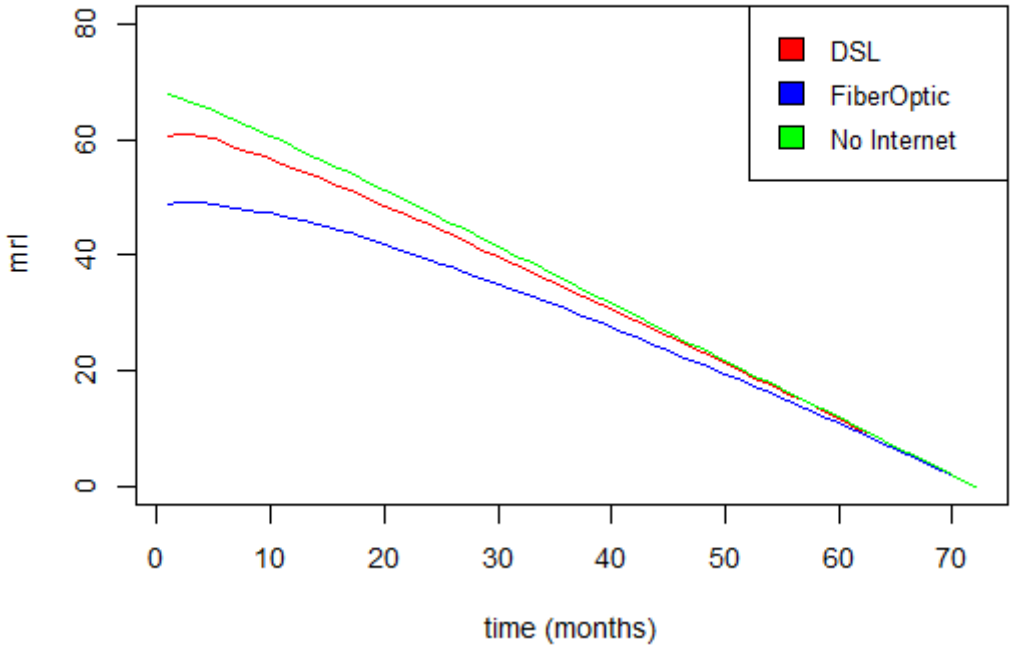

Figure 8: Estimated mean residual life function for different types of *InternetService*

BillingYes, PaymentMethod ElectronicCheck, PaymentMethod MailedCheck and *MonthlyCharges* had greater than 1 hazard ratios.

The survival function S(t) gives the probability of a subject's survival time T going beyond t units of time i.e.,

$$S(t) = Pr(T >= t) \tag{2}$$

In terms of interpretation, a better metric in survival analysis is the mean residual life (mrl) function. The mrl function gives the expected remaining life of a subject given they have survived upto t units of time. If we have a correct estimate of the survival function, the mrl function M(t) can be calculated from the survival function S(t) as follows,

$$M(t) = \frac{\int_t^\infty S(u)du}{S(t)} \tag{3}$$

We used the *survfit* function from the survival library to get the survival probabilities. The mrl function was estimated at different thresholds for all the treatment levels. One such result is illustrated in Fig. 8.

Appendix

R codes used for fitting the models

```
# Reading data
tel.obj=read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
x = model.matrix(Churn~.-customerID-TotalCharges,tel.obj)[,-1]
y = tel.obj$Churn

#Changing "no phone service" to "no"
tel.obj$MultipleLines <- as.factor(mapvalues(tel.obj$MultipleLines,
from=c("No phone service"),to=c("No")))

# Model fit and prediction
log.mod = glmnet(x.train, y[train], alpha = 0.95, lambda = grid, family = "binomial")

cvfit = cv.glmnet(x.train,y[train], family = "binomial",type.measure = "class")
bestlam = cvfit$lambda.1se

log.pred = predict(log.mod, newx = x.test, type = "class", s=bestlam)

# Coefficient Estimates using glmnet
log.full = glmnet(as.matrix(x), y, alpha = 0.95, lambda = bestlam, family = "binomial")
lasso.pred = predict(log.full, type = "coefficients", s=bestlam)[1:23,]
coef(log.full,s=bestlambda) #print the coefficients

#Using glm() function to find ROC and AUC
glm.fit = glm(Churn~SeniorCitizen+Dependents+tenure+MultipleLines+InternetService+OnlineSecurity+Tec
glm.prob = predict(glm.fit, type = "response")
pr <- prediction(glm.prob, tel.obj$Churn)
prcurve <- performance(pr, measure = "tpr",x.measure = "fpr")
plot(prcurve)
auc <- performance(pr, measure = "auc")
auc1 <- auc@y.values

# K fold cross validation
library(caret)
cntrl <- trainControl(method = "cv", number = 10)
model <- train(Churn~.-customerID-Partner-gender-OnlineBackup-DeviceProtection-TotalCharges,
data = tel.obj, trControl = cntrl, method = "glm", family = binomial())
probs = predict(model, type = "prob")
probs$No<-NULL
preds = rep("No",7032)
preds[probs>0.5]="Yes"
mean(preds==tel.obj$Churn)      #0.8030432

#Cost function
# threshold vector
thresh <- seq(0.05,1, length = 20)
#cost vector
cost = rep(0,length(thresh))
# cost as a function of threshold
for (i in 1:length(thresh)){
  glm.pred = rep("No", length(glm.prob))
  glm.pred[glm.prob > thresh[i]] = "Yes"
  glm.pred= as.factor(glm.pred)
  x1 <- confusionMatrix(glm.pred, tel.obj$Churn, positive = "Yes")
```

```
  TN <- x1$table[1]/7032
  FP <- x1$table[2]/7032
  FN <- x1$table[3]/7032
  TP <- x1$table[4]/7032
  cost[i] = FN*315 + TP*63 + FP*63 + TN*0
}

# standard model - threshold 0.5
glm.pred = rep("No", length(churn.probs))
glm.pred[churn.probs > 0.5] = "Yes"
glm.pred <- as.factor(glm.pred)

x1 <- confusionMatrix(glm.pred, tel.obj$Churn, positive = "Yes")
TN <- x1$table[1]/7032
FP <- x1$table[2]/7032
FN <- x1$table[3]/7032
TP <- x1$table[4]/7032
cost_simple = FN*315 + TP*63 + FP*63 + TN*0

savings_per_customer = cost_simple - min(cost)
total_savings = 1000000*savings_per_customer
total_savings #10,374,573

SURVIVAL ANALYSIS: COX MODEL
#Changing levels of Churn from "No" and "Yes" to "1" and "0" respectively
tel.obj$Churn <- as.numeric(tel.obj$Churn)
tel.obj$Churn <- ifelse(tel.obj$Churn==1,1,0)


#Creating x matrix with subset of predictors
x = model.matrix(tenure~.-Churn-TotalCharges-MonthlyCharges-DeviceProtection-OnlineBackup-PhoneServi

#Creating y matrix using Surv function
y = Surv(time = tel.obj$tenure, event = tel.obj$Churn)

#Fit model
mod.cox = glmnet(x, y, family = "cox", alpha = 0.95)
#Cross validation for optimal tuning parameter
cv.cox = cv.glmnet(x ,y, family = "cox", alpha = 0.95)

best.lam = cv.cox$lambda.1se

f = coef(mod.cox, s = best.lam)
#Calculating Hazard Ratios
HR = list()
HR = exp(f@x)

#Hazard Regression Model using hare function
hr = hare(data = tel.obj$tenure, delta = tel.obj$Churn, cov = x)



CODE FOR ESTIMATING MRL FUNCTION:

#Use survfit function to get survival probabilities
y = Surv(time = tel.obj$tenure, event = tel.obj$Churn)
fit = survfit(y~tel.obj$InternetService, data = tel.obj)
###Only change the variable name in the line above, all the other code remains the same###
```

```
summary(fit)

i=1
n_levels = count(fit$strata)$freq
size = count(fit$strata)$x           #Equal to max(tenure)= 72
lim = n_levels*size
counter = size

while(counter<=lim){
prob = fit$surv[i:counter]          #Survival probabilities
xaxi = list()                #List for storing coordinates
yaxi = list()
i=1
#Loop for 1 level out of 3 or 4
while(i<=size) {
  thresh = i          #Threshold has to go from 1-72 for all levels
  den = prob[thresh]  #Denominator storing S(t) at threshold t
  value = thresh
  num = 0             #Numerator of MRL
#At every threshold sum the probabilities after t
while (value< size) {
  value= value+1
  num = num + prob[value]
  }
  mrl = num/den
  yaxi = append(yaxi, mrl)
  xaxi = append(xaxi, thresh)
  i= i+1
}
if(counter==size){
  par(mar = c(5,5,4,2),cex=0.9)
  plot(xaxi, yaxi, xlab ="time (months)", ylab = "mrl", type = "l", col="red", ylim = c(0,80))
} else if(counter==(size*2)){
    lines(xaxi, yaxi, col="blue")
} else if(counter==(size*3)) {
    lines(xaxi, yaxi, col="green")
} else {
  lines(xaxi, yaxi, col="black")
}
i= (counter+1)
counter = counter + size
}
```