In this blog we will evaluate how OLS and GLM models can help us give more accurate info. I will be using Titanic data to demonstrate this. This is a collection of both categorical and continuous variable.

Data View

```
dt_train <- titanic::titanic_train
dt_train[dt_train==" "]= NA

dt_train$Age[which(is.na(dt_train$Age))]
##   [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
NA NA
## [176] NA NA
# paste("Number of NA in each column ")
# sapply(dt_train, function(x) sum(is.na(x)))
dt_train_sub <- subset(dt_train[which(!is.na(dt_train$Age)),],select=c(2,3,5,
6,7,8,10,12))


dt_train_sub$Sex[which(dt_train_sub$Sex=="male")] <- 1
dt_train_sub$Sex[which(dt_train_sub$Sex=="female")] <- 0
# dt_train_sub$Sex<- as.factor(dt_train_sub$Sex)
# dt_train_sub$Survived <- as.factor(dt_train_sub$Survived)

paste(" Glimpse Of Data: ")
## [1] " Glimpse Of Data: "
head(dt_train_sub)
##   Survived Pclass Sex Age SibSp Parch    Fare Embarked
## 1        0      3   1  22     1     0  7.2500        S
## 2        1      1   0  38     1     0 71.2833        C
## 3        1      3   0  26     0     0  7.9250        S
## 4        1      1   0  35     1     0 53.1000        S
## 5        0      3   1  35     0     0  8.0500        S
## 7        0      1   1  54     0     0 51.8625        S
str(dt_train_sub)
## 'data.frame':    714 obs. of  8 variables:
##  $ Survived: int  0 1 1 1 0 0 0 1 1 1 ...
##  $ Pclass  : int  3 1 3 1 3 1 3 3 2 3 ...
##  $ Sex     : chr  "1" "0" "0" "0" ...
##  $ Age     : num  22 38 26 35 35 54 2 27 14 4 ...
##  $ SibSp   : int  1 1 0 1 0 0 3 0 1 1 ...
##  $ Parch   : int  0 0 0 0 0 0 1 2 0 1 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: chr  "S" "C" "S" "S" ...
```

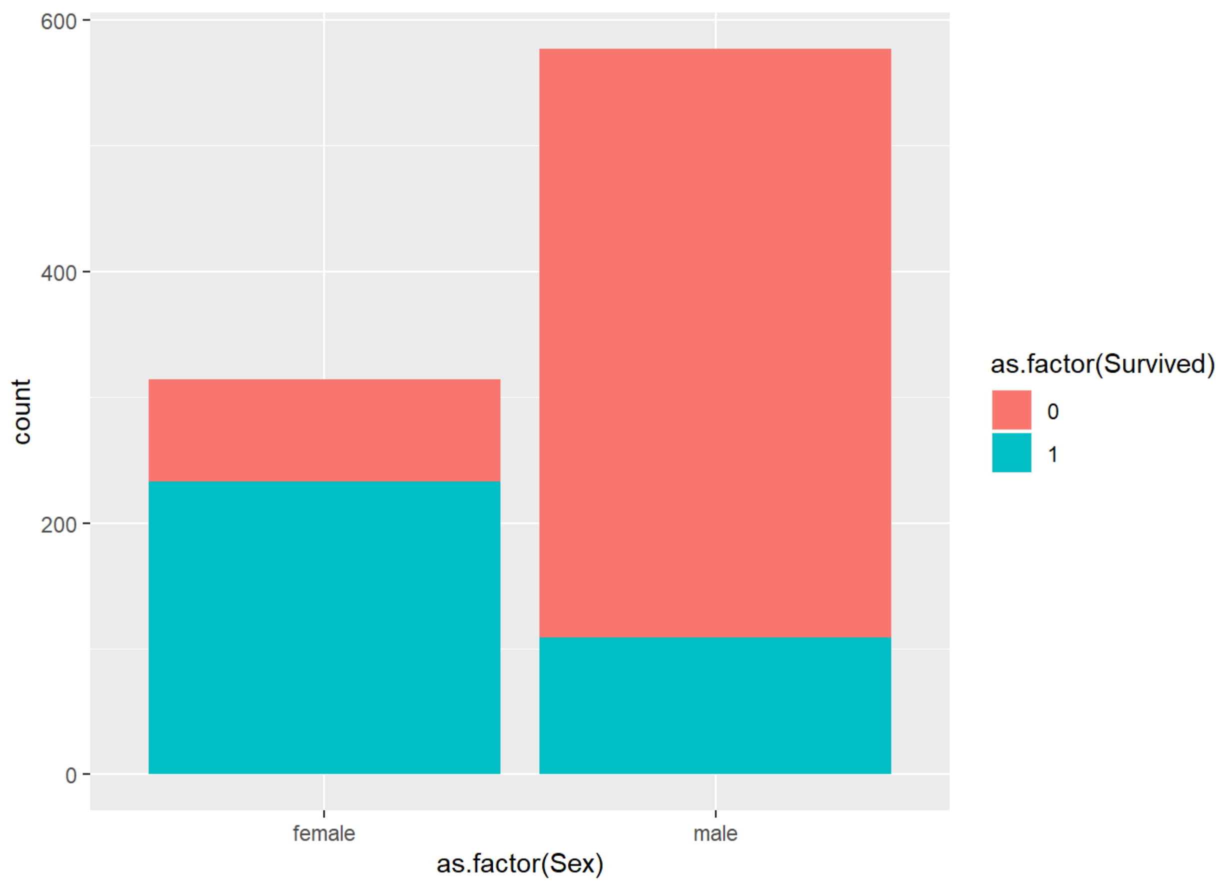
summary of subsetted data:

```
    Survived          Pclass          Sex                  Age            SibSp            Parch       
 Min.   :0.0000   Min.   :1.000   Length:714         Min.   : 0.42   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:1.000   Class :character   1st Qu.:20.12   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :0.0000   Median :2.000   Mode  :character   Median :28.00   Median :0.0000   Median :0.0000  
 Mean   :0.4062   Mean   :2.237                      Mean   :29.70   Mean   :0.5126   Mean   :0.4314  
 3rd Qu.:1.0000   3rd Qu.:3.000                      3rd Qu.:38.00   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :3.000                      Max.   :80.00   Max.   :5.0000   Max.   :6.0000  
      Fare          Embarked        
 Min.   :  0.00   Length:714        
 1st Qu.:  8.05   Class :character  
 Median : 15.74   Mode  :character  
 Mean   : 34.69                     
 3rd Qu.: 33.38                     
 Max.   :512.33                     
```

Eye View

0 => passengers that did not survive, and 1 => passengers who survived

```
ggplot(dt_train, aes(x=as.factor(Sex),fill=as.factor(Survived))) + geom_bar()
```

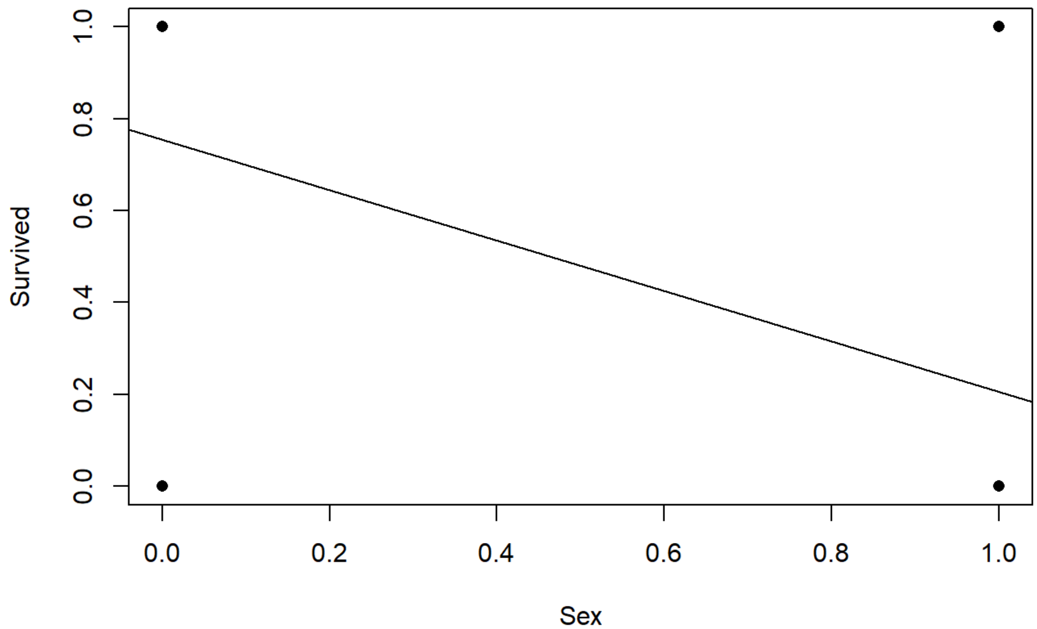


OLS Model

ordinary least squares (OLS) is a type of linear least squares method for estimating the unknown parameters in a linear regression model. OLS chooses the parameters of a linear function of a set of explanatory variables by the principle of least squares: minimizing the sum of the squares of the differences between the observed dependent variable (values of the variable being observed) in the given dataset and those predicted by the linear function

Lets check how we can predict “Survived” by “Sex” from Titanic data. We will see how different models shows different result by model.

```
model_ols <- lm(Survived ~ Sex,data=dt_train_sub)
model_ols_sa <- lm(Survived ~ Age,data=dt_train_sub)
summary(model_ols)
##
## Call:
## lm(formula = Survived ~ Sex, data = dt_train_sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7548 -0.2053 -0.2053  0.2452  0.7947
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.75479    0.02564   29.43   <2e-16 ***
## Sex1        -0.54949    0.03220  -17.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4143 on 712 degrees of freedom
## Multiple R-squared:  0.2903, Adjusted R-squared:  0.2893
## F-statistic: 291.3 on 1 and 712 DF,  p-value: < 2.2e-16
plot( Survived~Sex,dt_train_sub,col = NULL,bg = rgb(0, 0, 0, 0.5), pch = 21)
abline(model_ols)
```



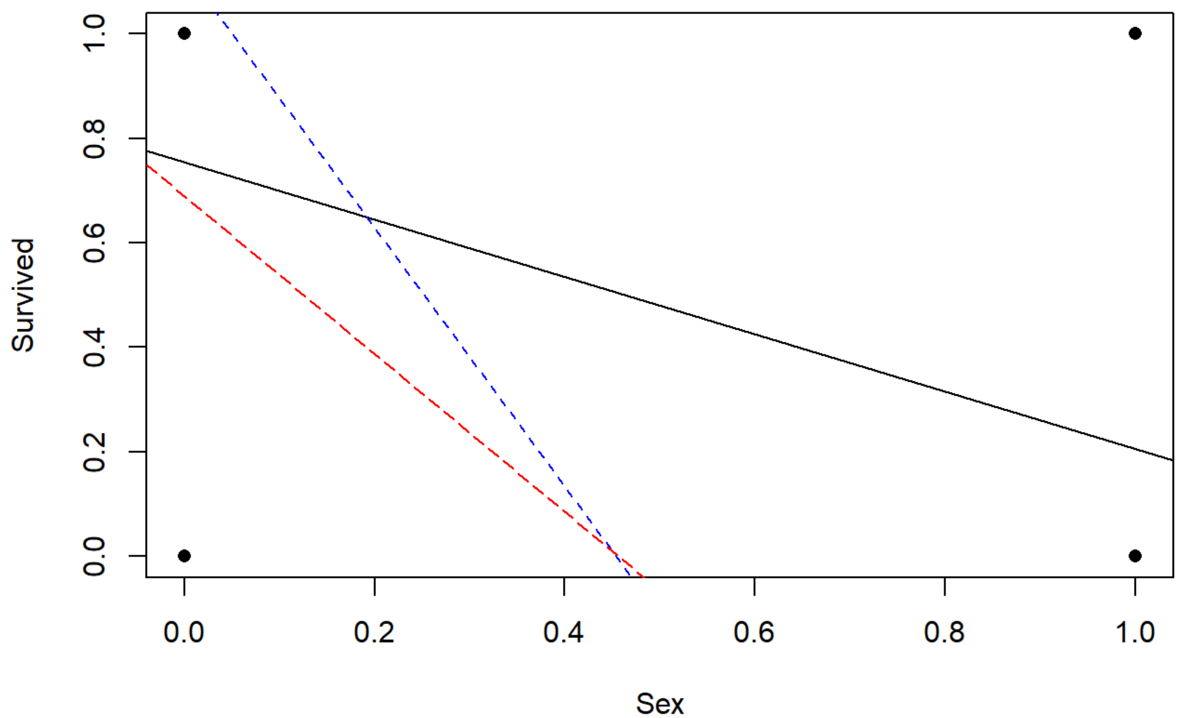
GLM Model

```
# Now let's perform a logistic regression
model_glm <- glm(Survived ~ Sex,family=binomial(link='logit'),data=dt_train_s
ub)
model_glm_sa <- glm(Survived ~ Age,family=binomial(link='logit'),data=dt_trai
n_sub)
model_glm_sal <- glm(Survived ~ .,family=binomial(link='logit'),data=dt_train
_sub)
summary(model_glm)
```

```
##
## Call:
## glm(formula = Survived ~ Sex, family = binomial(link = "logit"),
##     data = dt_train_sub)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6767  -0.6779  -0.6779   0.7501   1.7795
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.1243     0.1439   7.814 5.52e-15 ***
## Sex1         -2.4778     0.1850 -13.392  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 964.52  on 713  degrees of freedom
## Residual deviance: 750.70  on 712  degrees of freedom
## AIC: 754.7
##
## Number of Fisher Scoring iterations: 4
model_glm_prob <- glm(Survived ~ Sex, family = binomial(link = "probit"),data
=dt_train_sub)

plot( Survived~Sex,dt_train_sub,col = NULL,bg = rgb(0, 0, 0, 0.5), pch = 21)
 abline(model_ols)
 abline(model_glm,col="blue",lty=2)
 abline(model_glm_prob , lty=5,col="red")
```



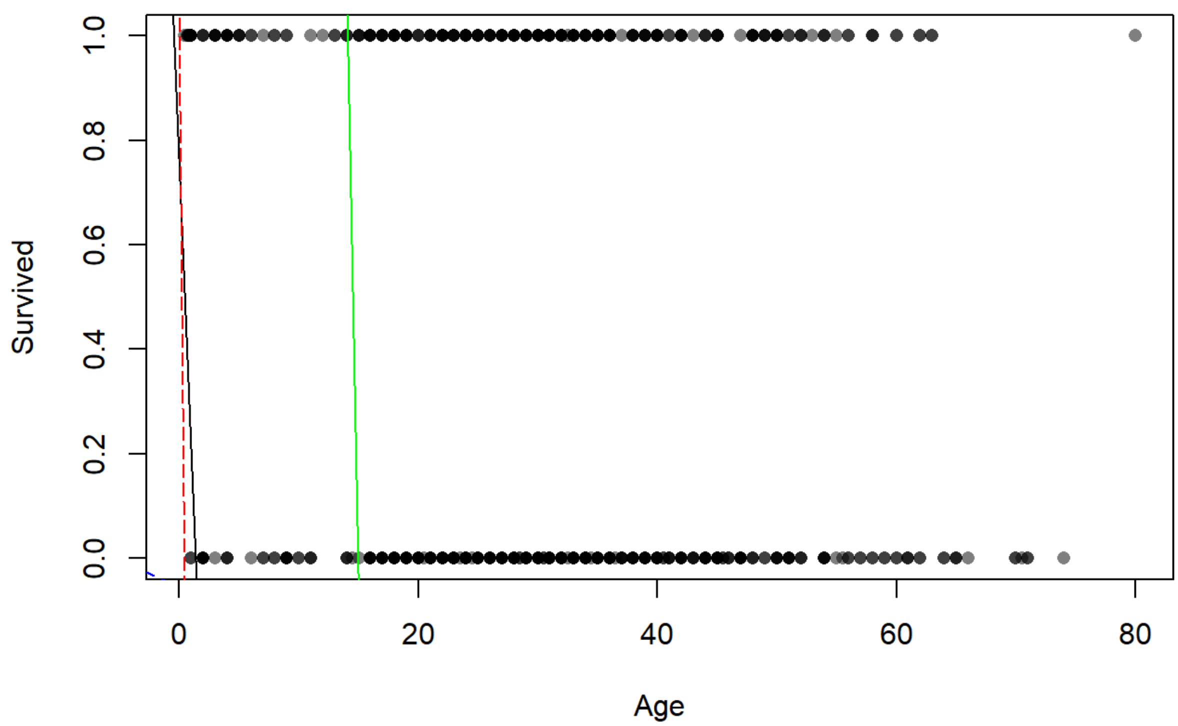

```
bbmle::AICctab(model_ols,model_glm,model_glm_prob)
##                dAICc df
## model_glm       0.0  2
## model_glm_prob  0.0  2
## model_ols      17.3  3
```

As we can see the “Blue” line is based on Logit link function of GLM regression line, does an adequate job predicting Surviver based on sex.

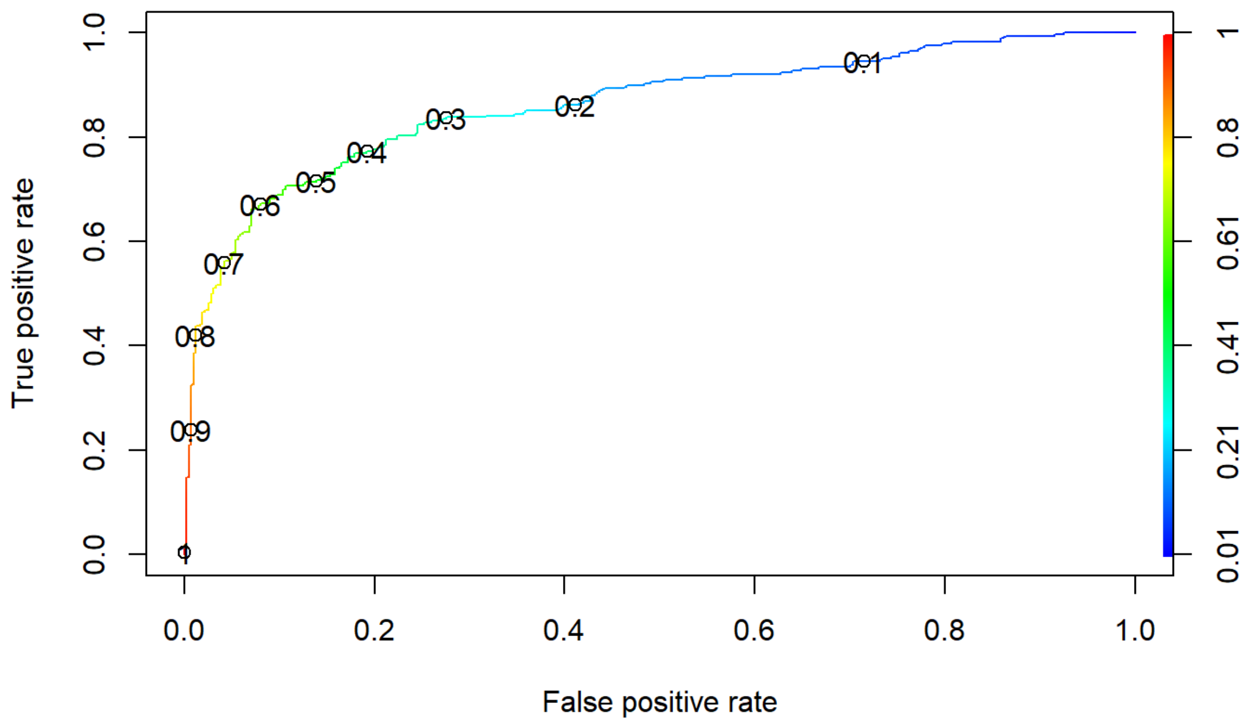
ROC Curve

```r
library(ROCR)
## Warning: package 'ROCR' was built under R version 3.5.3
## Loading required package: gplots
## Warning: package 'gplots' was built under R version 3.5.3
##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##     lowess
 #----(Survived ~ Age)

 plot(Survived ~ Age,dt_train_sub,col = NULL,bg = rgb(0, 0, 0, 0.5), pch = 21
)
  abline(model_ols)
  abline(model_glm_sal,col="green")
## Warning in abline(model_glm_sal, col = "green"): only using the first two
of 10
## regression coefficients
  abline(model_glm_sa,col="blue",lty=2)
  abline(model_glm , lty=5,col="red")
```




```r
res <- predict(model_glm_sal,dt_train_sub,type="response")
ROCRPred <- prediction(res,dt_train_sub$Survived)
ROCSPRef <- performance(ROCRPred,"tpr","fpr")
plot(ROCSPRef,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))
```




```
# 1 if the passenger survived or 0 if they did not
(table (Actaulvalue= as.factor(dt_train_sub$Survived),predictedvalue = res >
0.5))
##              predictedvalue
## Actaulvalue FALSE TRUE
##           0   365   59
##           1    83  207
(table (Actaulvalue= as.factor(dt_train_sub$Survived),predictedvalue = res >
0.6))
##              predictedvalue
## Actaulvalue FALSE TRUE
##           0   390   34
##           1    96  194
(table (Actaulvalue= as.factor(dt_train_sub$Survived),predictedvalue = res >
0.3))
##              predictedvalue
## Actaulvalue FALSE TRUE
##           0   307  117
##           1    48  242
(table (Actaulvalue= as.factor(dt_train_sub$Survived),predictedvalue = res >
0.2))
##              predictedvalue
## Actaulvalue FALSE TRUE
##           0   250  174
##           1    40  250
```

```
(table (Actaulvalue= as.factor(dt_train_sub$Survived),predictedvalue = res >
0.7))
##            predictedvalue
## Actaulvalue FALSE TRUE
##           0   406   18
##           1   129  161
```

```
With threshold >0.7
##            predictedvalue
## Actaulvalue FALSE TRUE
##           0   406   18
##           1   129  161
      WE are saying that Model is predicitng 18 people that they will not sur
vive, so it means that 129 False Negative , i.e. 129 people would not survive
even if our model say that with threshold = .7
```

With threshold >0.3

```
##            predictedvalue
## Actaulvalue FALSE TRUE
##           0   307  117
##           1    48  242
```

WE are saying that Model is predicting 117 people that they will not survive, so it means that 48 False Negative , i.e. 48 people would not survive even if our model say that with threshold = 3.