Alexis Casas: Titanic: Machine Learning from Disaster Competition

The competition is simple: use machine learning to create a model that predicts which passengers survived the Titanic shipwreck.

In this challenge, we ask you to build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (ie name, age, gender, socio-economic class, etc).

In [177]:
```
#1. read the data
training_set <- read.csv('train.csv')
test_set <- read.csv('test.csv')
```

In [178]:
```
#View data
head(training_set)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
	<int>	<int>	<int>	<fct>	<fct>	<dbl>	<int>	<int>	<fct>	<dbl>	<fct>	<fct>
1	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500		S
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250		S
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
5	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500		S
6	6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.4583		Q

In [179]:
```
test_set[test_set==""]  <- NA
training_set[training_set==""]  <- NA
```

In [180]:
```
#Checking Columns for Missing values: Training Set
library(questionr)
missingvaluestable <- freq.na(training_set)
missingvaluestable
```

	missing	%

Cabin	687	77
Age	177	20
Embarked	2	0
PassengerId	0	0
Survived	0	0
Pclass	0	0
Name	0	0
Sex	0	0
SibSp	0	0
Parch	0	0
Ticket	0	0
Fare	0	0

In [181]:
```
#Checking Columns for Missing values: Test Set
missingvaluestable <- freq.na(test_set)
missingvaluestable
```

	missing	**%**
Cabin	327	78
Age	86	21
Fare	1	0
PassengerId	0	0
Pclass	0	0
Name	0	0
Sex	0	0
SibSp	0	0
Parch	0	0
Ticket	0	0
Embarked	0	0

Cabin has about 78% of the data missing. I will delete this column for both sets. Both datasets are missing about 20% of their data in Age. Because Age seems important in predicting who would survive, I would rather not delete this. Will investigate more about this colum before making any decisions

In [182]:
```
head(training_set)#cabin is column 11
head(test_set) #cabin is column 10
```

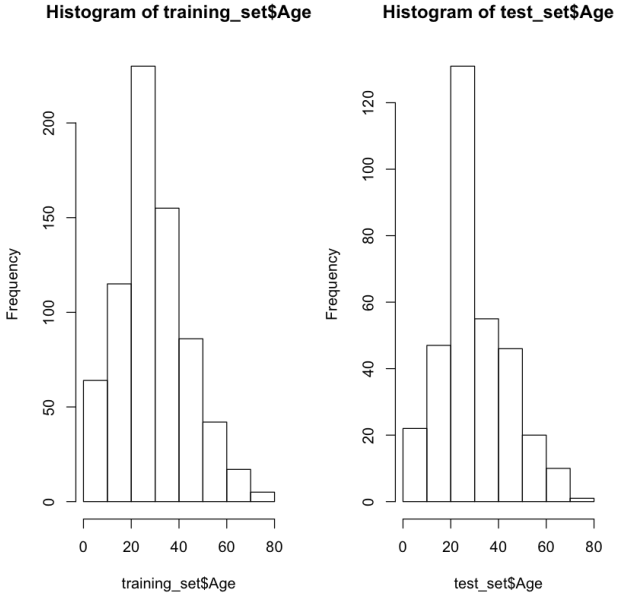
	PassengerId	**Survived**	**Pclass**	**Name**	**Sex**	**Age**	**SibSp**	**Parch**	**Ticket**	**Fare**	**Cabin**	**Embarked**
	<int>	**<int>**	**<int>**	**<fct>**	**<fct>**	**<dbl>**	**<int>**	**<int>**	**<fct>**	**<dbl>**	**<fct>**	**<fct>**
1	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	NA	S
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250	NA	S
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S

5	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500	NA	S
6	6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.4583	NA	Q

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
	<int>	<int>	<fct>	<fct>	<dbl>	<int>	<int>	<fct>	<dbl>	<fct>	<fct>
1	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NA	Q
2	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NA	S
3	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NA	Q
4	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NA	S
5	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NA	S
6	897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250	NA	S

In [183]:
```
training_set <- training_set[,-11]
test_set <- test_set[,-10]
```

In [184]:
```
par(mfrow=c(1,2))
hist(training_set$Age)
hist(test_set$Age)
```



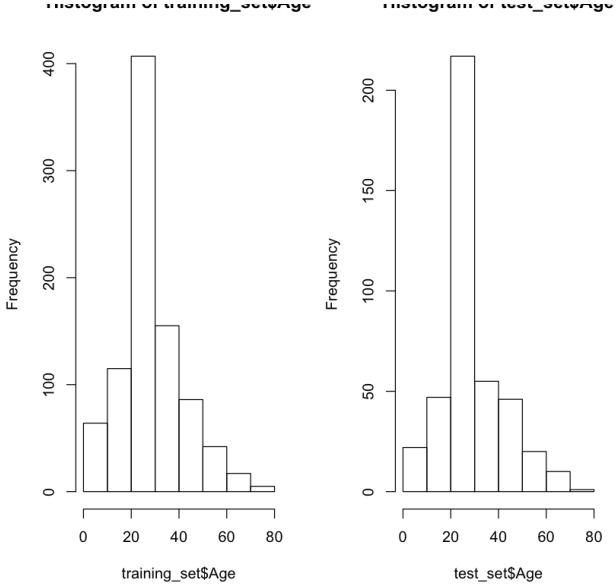
In [185]:
```
#what is the meadian and mean age?
median(training_set$Age,na.rm = T) #about 28 years old
mean(training_set$Age,na.rm = T)
#Because the data is skewed i will impute based on the median.
```

28

29.6991176470588

In [186]:
```
training_set$Age[is.na(training_set$Age)] <- median(training_set$Age,na.rm = T)
test_set$Age[is.na(test_set$Age)] <- median(test_set$Age,na.rm = T)
```

In [187]:
```
par(mfrow=c(1,2))
hist(training_set$Age)
hist(test_set$Age)
```



In [188]:
```
#checking if missing values are gone
missingvaluestable <- freq.na(training_set)
missingvaluestable #two values in embarked column
```

	missing	%
Embarked	2	0
PassengerId	0	0
Survived	0	0
Pclass	0	0
Name	0	0
Sex	0	0
Age	0	0
SibSp	0	0
Parch	0	0
Ticket	0	0
Fare	0	0

In [189]:
```
missingvaluestable <- freq.na(test_set)
missingvaluestable #one missing value for fare
```

	missing	%
Fare	1	0
PassengerId	0	0
Pclass	0	0
Name	0	0
Sex	0	0
Age	0	0
SibSp	0	0
Parch	0	0
Ticket	0	0
Embarked	0	0

In [190]:
```
training_set <- training_set[-which(is.na(training_set$Embarked)),]
test_set <- test_set[-which(is.na(test_set$Fare)),]
```

In [191]:
```
missingvaluestable <- freq.na(test_set)
missingvaluestable #All Clean!
```

	missing	%
PassengerId	0	0
Pclass	0	0
Name	0	0
Sex	0	0
Age	0	0
SibSp	0	0
Parch	0	0
Ticket	0	0
Fare	0	0
Embarked	0	0

Next, I believe that it may be nice to know someone's age group rather than their specific age for future predictions.

In [192]:
```
training_set$AgeGroup <- training_set$Age
training_set$AgeGroup[training_set$Age>=0 & training_set$Age<10] <- 1
training_set$AgeGroup[training_set$Age>=10 & training_set$Age<20] <- 2
training_set$AgeGroup[training_set$Age>=20 & training_set$Age<30] <- 3
training_set$AgeGroup[training_set$Age>=30 & training_set$Age<40] <- 4
training_set$AgeGroup[training_set$Age>=40 & training_set$Age<50] <- 5
training_set$AgeGroup[training_set$Age>=50 & training_set$Age<60] <- 6
training_set$AgeGroup[training_set$Age>=60 & training_set$Age<70] <- 7
training_set$AgeGroup[training_set$Age>=70 & training_set$Age<80] <- 8
```

In [193]:
```
test_set$AgeGroup <- test_set$Age
test_set$AgeGroup[test_set$Age>=0 & test_set$Age<10] <- 1
test_set$AgeGroup[test_set$Age>=10 & test_set$Age<20] <- 2
test_set$AgeGroup[test_set$Age>=20 & test_set$Age<30] <- 3
test_set$AgeGroup[test_set$Age>=30 & test_set$Age<40] <- 4
test_set$AgeGroup[test_set$Age>=40 & test_set$Age<50] <- 5
test_set$AgeGroup[test_set$Age>=50 & test_set$Age<60] <- 6
test_set$AgeGroup[test_set$Age>=60 & test_set$Age<70] <- 7
test_set$AgeGroup[test_set$Age>=70 & test_set$Age<80] <- 8
```

In [194]:
```
#Next I will examine the structure and encode variables.
str(training_set)
str(test_set)
```

```
'data.frame':   889 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417
581 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 28 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 .
..
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
 $ AgeGroup   : num  3 4 3 4 4 3 6 1 3 2 ...
'data.frame':   417 obs. of  11 variables:
```

```
 $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
 $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
 $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182 370 8
5 58 5 104 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
 $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
 $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
 $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
 $ Ticket     : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101 270 .
..
 $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
 $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
 $ AgeGroup   : num  4 5 7 3 3 2 4 3 2 3 ...
```

In [195]:
```
#First off we want to delete, name and ticket
training_set <- training_set[,-c(4,9)]
test_set <- test_set[,-c(3,8)]
```

In [196]:
```
str(training_set)
str(test_set)
```

```
'data.frame':   889 obs. of  10 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 28 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
 $ AgeGroup   : num  3 4 3 4 4 3 6 1 3 2 ...
'data.frame':   417 obs. of  9 variables:
 $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
 $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
 $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
 $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
 $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
 $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
 $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
 $ AgeGroup   : num  4 5 7 3 3 2 4 3 2 3 ...
```

In [197]:
```
#Next, encoding of categorical variables
training_set$Sex <- factor(training_set$Sex, levels=c('male','female'),1:2) #for Sex
training_set$Embarked <- factor(training_set$Embarked, levels=c('S','C','Q'),1:3) #for Embarked

test_set$Sex <- factor(test_set$Sex, levels=c('male','female'),1:2) #for Sex
test_set$Embarked <- factor(test_set$Embarked, levels=c('S','C','Q'),1:3) #for Embarked
```

In [198]:
```
str(training_set)
str(test_set)
head(test_set)
```

```
'data.frame':   889 obs. of  10 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Sex        : Factor w/ 2 levels "1","2": 1 2 2 2 1 1 1 1 2 2 ...
 $ Age        : num  22 38 26 35 35 28 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Embarked   : Factor w/ 3 levels "1","2","3": 1 2 1 1 1 3 1 1 1 2 ...
 $ AgeGroup   : num  3 4 3 4 4 3 6 1 3 2 ...
'data.frame':   417 obs. of  9 variables:
 $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
 $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
 $ Sex        : Factor w/ 2 levels "1","2": 1 2 1 1 2 1 2 1 2 1 ...
 $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
```

```
$ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
$ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
$ Fare       : num  7.83 7 9.69 8.66 12.29 ...
$ Embarked   : Factor w/ 3 levels "1","2","3": 3 1 3 1 1 1 3 1 2 1 ...
$ AgeGroup   : num  4 5 7 3 3 2 4 3 2 3 ...
```

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	AgeGroup
	<int>	<int>	<fct>	<dbl>	<int>	<int>	<dbl>	<fct>	<dbl>
1	892	3	1	34.5	0	0	7.8292	3	4
2	893	3	2	47.0	1	0	7.0000	1	5
3	894	2	1	62.0	0	0	9.6875	3	7
4	895	3	1	27.0	0	0	8.6625	1	3
5	896	3	2	22.0	1	1	12.2875	1	3
6	897	3	1	14.0	0	0	9.2250	1	2

In [199]:
```
#deleting age since we have age group
training_age <- training_set$Age
test_age <- test_set$Age
training_set <- training_set[,-5]
test_set <- test_set[,-4]
```

In [200]:
```
#The response variable is Survived since it is the dependent variable we are trying to predict in
our model
#Fitting Logistic onto the training set (glm- generalied linear models)
classifier <- glm(formula = Survived ~ .,
                  family = binomial,
                  data = training_set[,-1])
```

In [201]:
```
#Finding out the probability of prediction
prob_predictions = predict(classifier, type='response', test_set[,-1]) #test without passenger id
y_predict <- ifelse(prob_predictions>=0.5,1,0)
y_predict #scorecard of the predictions
```

```
1
 0
2
 1
3
 0
4
 0
5
 1
6
 0
7
 1
8
 0
9
 1
10
 0
11
 0
12
 0
13
 1
14
```

0
15
1
16
1
17
0
18
0
19
1
20
1
21
0
22
0
23
1
24
1
25
1
26
0
27
1
28
0
29
0
30
0
31
0
32
0
33
1
34
1
35
0
36
0
37
1
38
1
39
0
40
0
41
0
42
0
43
0
44
1

.
45
1
46
0
47
0
48
0
49
1
50
1
51
0
52
0
53
1
54
1
55
0
56
0
57
0
58
0
59
0
60
1
61
0
62
0
63
0
64
1
65
0
66
1
67
1
68
0
69
1
70
1
71
1
72
0
73
1
74
1

```
75
  1
76
  1
77
  0
78
  1
79
  0
80
  1
81
  0
82
  0
83
  0
84
  0
85
  0
86
  0
87
  1
88
  1
89
  1
90
  0
91
  1
92
  0
93
  1
94
  0
95
  1
96
  0
97
  1
98
  0
99
  1
100
  0
101
  1
102
  0
103
  0
104
  0
```

105
1
106
0
107
0
108
0
109
0
110
0
111
0
112
1
113
1
114
1
115
1
116
0
117
0
118
1
119
1
120
1
121
1
122
0
123
1
124
0
125
0
126
1
127
0
128
1
129
0
130
0
131
0
132
0
133
1
134
0

135
0
136
0
137
0
138
0
139
1
140
0
141
0
142
1
143
1
144
0
145
0
146
0
147
0
148
0
149
0
150
0
151
1
152
0
154
1
155
0
156
0
157
1
158
1
159
0
160
1
161
1
162
0
163
1
164
0
165
0
166

1
167
1
168
0
169
1
170
1
171
0
172
0
173
0
174
0
175
0
176
1
177
1
178
0
179
1
180
1
181
0
182
0
183
1
184
0
185
1
186
0
187
1
188
0
189
0
190
0
191
0
192
0
193
0
194
0
195
0
196

0
197
1
198
1
199
0
200
1
201
1
202
…
203
1
204
0
205
1
206
0
207
1
208
0
209
1
210
1
211
0
212
1
213
0
214
0
215
0
216
1
217
0
218
0
219
0
220
0
221
0
222
0
223
1
224
1
225
1
226

1
227
0
228
0
229
0
230
0
231
1
232
0
233
1
234
1
235
1
236
0
237
1
238
0
239
0
240
0
241
0
242
0
243
1
244
0
245
0
246
0
247
1
248
1
249
0
250
0
251
0
252
0
253
1
254
0
255
1
256

0
257
1
258
1
259
0
260
1
261
0
262
0
263
0
264
0
265
1
266
0
267
1
268
1
269
1
270
0
271
0
272
0
273
0
274
0
275
0
276
1
277
0
278
0
279
0
280
0
281
1
282
0
283
0
284
0
285
0
286
0

287
0
288
0
289
1
290
1
291
0
292
0
293
0
294
1
295
0
296
0
297
0
298
1
299
1
300
1
301
1
302
0
303
0
304
0
305
0
306
0
307
0
308
0
309
1
310
0
311
1
312
1
313
0
314
0
315
1
316
1

317
0
318
1
319
0
320
0
321
0
322
0
323
0
324
0
325
0
326
0
327
0
328
1
329
0
330
1
331
0
332
1
333
0
334
1
335
1
336
0
337
0
338
0
339
1
340
0
341
1
342
0
343
0
344
1
345
0
346
1

347
1
348
0
349
1
350
0
351
0
352
1
353
1
354
0
355
0
356
1
357
0
358
0
359
1
360
1
361
1
362
0
363
0
364
0
365
0
366
0
367
1
368
1
369
0
370
1
371
0
372
0
373
0
374
0
375
0
376
1

```
377
 0
378
 0
379
 0
380
 1
381
 0
382
 1
383
 0
384
 0
385
 1
386
 0
387
 1
388
 0
389
 0
390
 0
391
 0
392
 1
393
 1
394
 1
395
 1
396
 1
397
 1
398
 0
399
 1
400
 0
401
 0
402
 0
```

In [202]:
```
summary(classifier)
```

```
Call:
glm(formula = Survived ~ ., family = binomial, data = training_set[,
    -1])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.3499  -0.6709  -0.4569   0.6726   2.5427

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.414036   0.378946   1.093   0.2746
Pclass      -0.854969   0.133033  -6.427  1.3e-10 ***
Sex2         2.722762   0.198198  13.738  < 2e-16 ***
SibSp       -0.237855   0.101562  -2.342   0.0192 *
Parch       -0.073129   0.114325  -0.640   0.5224
Fare         0.002393   0.002366   1.011   0.3120
Embarked2    0.458376   0.230079   1.992   0.0463 *
Embarked3    0.263279   0.322801   0.816   0.4147
AgeGroup    -0.018856   0.034097  -0.553   0.5803
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.82  on 888  degrees of freedom
Residual deviance:  811.66  on 880  degrees of freedom
AIC: 829.66

Number of Fisher Scoring iterations: 5
```

It seems like ther eis a strong correlation between Pclass, Sex factor 2, Sibsp, and Embarked factor 2.

Now lets try and make some real predictions:

In [208]:
```
str(y_predict)
```

```
 Named num [1:417] 0 1 0 0 1 0 1 0 1 0 ...
 - attr(*, "names")= chr [1:417] "1" "2" "3" "4" ...
```

In [209]:
```
test_set$Survived <- y_predict
```

In [244]:
```
submission_file <- test_set[,c(1,9)]
submission_file
write.csv(submission_file,"gender_submission.csv", row.names = FALSE)
```

	PassengerId	Survived
	<int>	<dbl>
1	892	0
2	893	1
3	894	0
4	895	0
5	896	1
6	897	0
7	898	1
8	899	0
9	900	1
10	901	0
11	902	0
12	903	0
13	904	1
14	905	0
15	906	1
16	907	1

17	908	0
18	909	0
19	910	1
20	911	1
21	912	0
22	913	0
23	914	1
24	915	1
25	916	1
26	917	0
27	918	1
28	919	0
29	920	0
30	921	0
⋮	⋮	⋮
389	1280	0
390	1281	0
391	1282	0
392	1283	1
393	1284	0
394	1285	0
395	1286	0
396	1287	1
397	1288	0
398	1289	1
399	1290	0
400	1291	0
401	1292	1
402	1293	0
403	1294	1
404	1295	0
405	1296	0
406	1297	0
407	1298	0
408	1299	1
409	1300	1
410	1301	1
411	1302	1
412	1303	1
413	1304	1
414	1305	0
415	1306	1

416	1307	0
417	1308	0
418	1309	0

In [204]:
```
library(dplyr)
survived_training <- filter(training_set, Survived == 1)
```

In [205]:
```
#Lets take a look at a scorecard of individuals who did survive. What criteria was in place.
#Common Themes:
#Sex=2 (Female)
#Age Group <=4 (people under the age of 40)
#PClass:
#It looks like if they are in class 3 and dtraveling with no SibSp they can survive,
#if they are any other class and travelining with SibSp they also will
head(survived_training)
```

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked	AgeGroup
	<int>	<int>	<int>	<fct>	<int>	<int>	<dbl>	<fct>	<dbl>
1	2	1	1	2	1	0	71.2833	2	4
2	3	1	3	2	0	0	7.9250	1	3
3	4	1	1	2	1	0	53.1000	1	4
4	9	1	3	2	0	2	11.1333	1	3
5	10	1	2	2	1	0	30.0708	2	2
6	11	1	3	2	1	1	16.7000	1	1

In [216]:
```
#Lets try and predict who will surve for females with an AgeGroup<5
test2 <- filter(test_set, AgeGroup <= 4)
test2 <- filter(test2, Sex==2)
```

In [218]:
```
#My guess is that the model will predict survived for females who are in an AgeGroup<=4.
prob_predictions = predict(classifier, type='response', test2[,-1])
y_predict <- ifelse(prob_predictions>=0.5,1,0)
y_predict #Looks like the model does predict this survived for most of these cases.
```

1
1
2
1
3
1
4
1
5
1
6
1
7
1
8
1
9
1
10
1
11
1
12
1

13
1
14
1
15
1
16
1
17
1
18
1
19
1
20
1
21
1
22
1
23
1
24
1
25
1
26
1
27
1
28
1
29
1
30
1
31
1
32
1
33
1
34
1
35
1
36
1
37
1
38
1
39
1
40
1
41
1
42
1

43
0
44
1
45
1
46
1
47
1
48
1
49
1
50
1
51
1
52
1
53
1
54
1
55
1
56
1
57
1
58
1
59
1
60
1
61
0
62
1
63
1
64
1
65
1
66
1
67
1
68
0
69
1
70
1
71
1
72
1

73
1
74
1
75
1
76
1
77
1
78
1
79
1
80
1
81
1
82
1
83
1
84
1
85
1
86
1
87
1
88
1
89
1
90
1
91
1
92
1
93
1
94
1
95
1
96
1
97
1
98
1
99
1
100
1
101
1
102
1
103

```
1
104
1
105
1
106
1
107
0
108
1
109
1
110
1
111
1
112
1
113
1
114
1
115
1
116
1
117
1
118
1
119
1
120
1
121
1
122
1
```

In [233]:
```
#What about women of any age in first class?
test2 <- filter(test_set, Pclass==1)
test2 <- filter(test2, Sex==2)
prob_predictions = predict(classifier, type='response', test2[,-1])
y_predict <- ifelse(prob_predictions>=0.5,1,0)
y_predict #Women of any age survived
```

```
1
1
2
1
3
1
4
1
5
1
6
1
7
```

1
8
1
9
1
10
1
11
1
12
1
13
1
14
1
15
1
16
1
17
1
18
1
19
1
20
1
21
1
22
1
23
1
24
1
25
1
26
1
27
1
28
1
29
1
30
1
31
1
32
1
33
1
34
1
35
1
36
1
37
1

```
38
 1
39
 1
40
 1
41
 1
42
 1
43
 1
44
 1
45
 1
46
 1
47
 1
48
 1
49
 1
50
 1
```

In [219]:
```
#What about for men in that age group?
test3 <- filter(test_set, AgeGroup <= 4)
test3 <- filter(test3, Sex==1)
```

In [220]:
```
prob_predictions = predict(classifier, type='response', test3[,-1])
y_predict <- ifelse(prob_predictions>=0.5,1,0)
y_predict #yes, they are most likely not to survive!
```

```
1
 0
2
 0
3
 0
4
 0
5
 0
6
 0
7
 0
8
 0
9
 0
10
 1
11
 0
12
 0
13
```

0
14
0
15
0
16
0
17
0
18
0
19
0
20
0
21
0
22
0
23
0
24
0
25
0
26
0
27
0
28
0
29
0
30
0
31
0
32
0
33
1
34
0
35
1
36
1
37
0
38
0
39
0
40
0
41
0
42
0
43

0
44
0
45
0
46
1
47
0
48
0
49
0
50
0
51
0
52
0
53
0
54
0
55
0
56
0
57
0
58
0
59
0
60
1
61
0
62
0
63
0
64
0
65
0
66
0
67
0
68
0
69
0
70
0
71
0
72
0
73

0
74
0
75
0
76
0
77
0
78
0
79
0
80
0
81
0
82
0
83
0
84
0
85
0
86
0
87
0
88
0
89
0
90
0
91
0
92
0
93
0
94
0
95
0
96
1
97
0
98
0
99
0
100
1
101
0
102
0
103
0

104
0
105
0
106
0
107
0
108
0
109
0
110
0
111
0
112
0
113
0
114
0
115
0
116
0
117
0
118
0
119
0
120
1
121
0
122
0
123
0
124
0
125
0
126
0
127
0
128
0
129
0
130
0
131
0
132
0
133
0

134
0
135
0
136
0
137
0
138
0
139
0
140
0
141
0
142
0
143
0
144
0
145
0
146
0
147
0
148
0
149
0
150
0
151
0
152
0
153
0
154
0
155
0
156
0
157
0
158
0
159
0
160
0
161
0
162
0
163
0

164
0
165
0
166
0
167
0
168
0
169
0
170
1
171
0
172
0
173
0
174
0
175
0
176
0
177
0
178
0
179
0
180
0
181
0
182
0
183
0
184
0
185
0
186
0
187
0
188
0
189
0
190
0
191
0
192
0
193
0

194
0
195
0
196
0
197
0
198
0
199
0
200
0
201
0
202
0
203
0
204
0
205
0
206
0
207
0
208
0
209
0
210
0
211
0
212
0
213
0
214
0

In [221]:
```
#What about old men?
test4 <- filter(test_set, AgeGroup > 4)
test4 <- filter(test3, Sex==1)
prob_predictions = predict(classifier, type='response', test4[,-1])
y_predict <- ifelse(prob_predictions>=0.5,1,0)
y_predict #Sorry men
```

1
0
2
0
3
0
4
0
5
0
6

```
    0
7
    0
8
    0
9
    0
10
    1
11
    0
12
    0
13
    0
14
    0
15
    0
16
    0
17
    0
18
    0
19
    0
20
    0
21
    0
22
    0
23
    0
24
    0
25
    0
26
    0
27
    0
28
    0
29
    0
30
    0
31
    0
32
    0
33
    1
34
    0
35
    1
36
```

1
37
0
38
0
39
0
40
0
41
0
42
0
43
0
44
0
45
0
46
1
47
0
48
0
49
0
50
0
51
0
52
0
53
0
54
0
55
0
56
0
57
0
58
0
59
0
60
1
61
0
62
0
63
0
64
0
65
0
66

0
67
0
68
0
69
0
70
0
71
0
72
0
73
0
74
0
75
0
76
0
77
0
78
0
79
0
80
0
81
0
82
0
83
0
84
0
85
0
86
0
87
0
88
0
89
0
90
0
91
0
92
0
93
0
94
0
95
0
96
1

97
0
98
0
99
0
100
1
101
0
102
0
103
0
104
0
105
0
106
0
107
0
108
0
109
0
110
0
111
0
112
0
113
0
114
0
115
0
116
0
117
0
118
0
119
0
120
1
121
0
122
0
123
0
124
0
125
0
126
0

127
0
128
0
129
0
130
0
131
0
132
0
133
0
134
0
135
0
136
0
137
0
138
0
139
0
140
0
141
0
142
0
143
0
144
0
145
0
146
0
147
0
148
0
149
0
150
0
151
0
152
0
153
0
154
0
155
0
156
0

157
0
158
0
159
0
160
0
161
0
162
0
163
0
164
0
165
0
166
0
167
0
168
0
169
0
170
1
171
0
172
0
173
0
174
0
175
0
176
0
177
0
178
0
179
0
180
0
181
0
182
0
183
0
184
0
185
0
186
0

187
0
188
0
189
0
190
0
191
0
192
0
193
0
194
0
195
0
196
0
197
0
198
0
199
0
200
0
201
0
202
0
203
0
204
0
205
0
206
0
207
0
208
0
209
0
210
0
211
0
212
0
213
0
214
0

In [226]:
```
#What about first class men?
test5 <- filter(test_set, Pclass==1)
test5 <- filter(test5, Sex==1)
```

```
prob_predictions = predict(classifier, type='response', test5[,-1])
y_predict <- ifelse(prob_predictions>=0.5,1,0)
y_predict
```

1
0
2
0
3
1
4
0
5
0
6
0
7
0
8
0
9
0
10
0
11
1
12
1
13
1
14
0
15
0
16
1
17
1
18
0
19
1
20
0
21
0
22
0
23
0
24
1
25
0
26
0
27
0
28
1
29

```
 1
30
 1
31
 0
32
 0
33
 0
34
 0
35
 0
36
 0
37
 1
38
 0
39
 1
40
 0
41
 0
42
 0
43
 0
44
 0
45
 0
46
 1
47
 0
48
 1
49
 1
50
 0
51
 0
52
 0
53
 0
54
 0
55
 0
56
 0
57
 1
```

In [229]:
```
#What about first class men and traveling with more than one Parchild?
test5 <- filter(test_set, Pclass==1)
```

```
test5 <- filter(test5, Sex==1)
test5 <- filter(test5, Parch>1)
prob_predictions = predict(classifier, type='response', test5[,-1])
y_predict <- ifelse(prob_predictions>=0.5,1,0)
y_predict #the model predicts about half of these will survive
```

1
0
2
1
3
1
4
0

In [231]:
```
#What about first class men and traveling with more than one SibSp?
test5 <- filter(test_set, Pclass==1)
test5 <- filter(test5, Sex==1)
test5 <- filter(test5, SibSp>1)
prob_predictions = predict(classifier, type='response', test5[,-1])
y_predict <- ifelse(prob_predictions>=0.5,1,0)
y_predict #the model predicts SibSp is not more important for men than Parchild
```

1: 0

Logistic Regression Conclusions:

The most important factors in determining who survived were:

Sex (Females were most likely)

Parchild, or SibSp: Whether or not the individual was traveling with others

Class: First class passengers were most likely to survive

Here are some senarios and the likeliness of survival:

Women of any age & in first class were very likely to survive.

Women under 40 years old had 100% survival rate.

Older men were very likely not to survive

Men traveling in first class with a parent or child had a 50% chance of survival

In [ ]: