

# Trilha\_6\_Regressao\_Logistica\_Titanic

André Gustavo Silva Lovo

26/11/2020

## Regressao Logistica - Base de dados Titanic

### Manipulando os dados

```
#install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

### Importando a base de dados treino

```
treino <- read.csv("titanic_train.csv")
```

### Importando a base de dados teste

```
teste <- read.csv("titanic_test.csv")
```

### Analizando o Banco de dados treino e suas variáveis

```
str(treino)
```

```
## 'data.frame':   891 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       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
"Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
## $ Sex        : chr  "male" "female" "female" "female" ...
## $ Age        : num  22 38 26 35 35 NA 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     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin      : chr  "" "C85" "" "C123" ...
## $ Embarked   : chr  "S" "C" "S" "S" ...
```

# Vocabulário do banco de dados

PassengerId: int - idpassageiro

\$ Survived : int - sobrevivente - (0 = No, 1 = Yes)

\$ Pclass : int - classe passageiro - (1 = 1st, 2 = 2nd, 3 = 3rd)

\$ Name : chr - nome

\$ Sex : chr - sexo

\$ Age : num - idade

\$ SibSp : int - numero de irmaos e conjuges a bordo

\$ Parch : int - numero de pais e filhos a bordo

\$ Ticket : chr - numero do bilhete

\$ Fare : num - tarifa do passageiro

\$ Cabin : chr - numero da cabine

\$ Embarked : chr - porto de embarque - (C = Cherbourg, Q = Queenstown, S = Southampton)

## Separando as colunas que queremos trabalhar

```
treino <- treino[ ,c('PassengerId' , 'Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked')]
```

## Verificando as linhas iniciais do meu banco de dados - head e tail

```
head(as.matrix(treino))
```

```
##      PassengerId Survived Pclass Sex      Age      SibSp Parch Fare
## [1,] "  1"          "0"      "3"  "male"  "22.00" "1"    "0"    "  7.2500"
## [2,] "  2"          "1"      "1"  "female" "38.00" "1"    "0"    " 71.2833"
## [3,] "  3"          "1"      "3"  "female" "26.00" "0"    "0"    "  7.9250"
## [4,] "  4"          "1"      "1"  "female" "35.00" "1"    "0"    " 53.1000"
## [5,] "  5"          "0"      "3"  "male"   "35.00" "0"    "0"    "  8.0500"
## [6,] "  6"          "0"      "3"  "male"   NA      "0"    "0"    "  8.4583"
##      Embarked
## [1,] "S"
## [2,] "C"
## [3,] "S"
## [4,] "S"
## [5,] "S"
## [6,] "Q"
```

## Verificando as linhas finais do meu banco de dados - head e tail

```
tail(as.matrix(treino))
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
## [886,]	"886"	"0"	"3"	"female"	"39.00"	"0"	"5"	" 29.1250"
## [887,]	"887"	"0"	"2"	"male"	"27.00"	"0"	"0"	" 13.0000"
## [888,]	"888"	"1"	"1"	"female"	"19.00"	"0"	"0"	" 30.0000"
## [889,]	"889"	"0"	"3"	"female"	NA	"1"	"2"	" 23.4500"
## [890,]	"890"	"1"	"1"	"male"	"26.00"	"0"	"0"	" 30.0000"
## [891,]	"891"	"0"	"3"	"male"	"32.00"	"0"	"0"	" 7.7500"
##	Embarked							
## [886,]	"Q"							
## [887,]	"S"							
## [888,]	"S"							
## [889,]	"S"							
## [890,]	"C"							
## [891,]	"Q"							

## Analizando o BD teste e suas variáveis

```
str(teste)
```

```
## 'data.frame':   418 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        : chr   "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis" "Wirz, Mr. Albert" ...
## $ Sex         : chr   "male" "female" "male" "male" ...
## $ 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      : chr   "330911" "363272" "240276" "315154" ...
## $ Fare        : num    7.83 7 9.69 8.66 12.29 ...
## $ Cabin       : chr    "" "" "" "" ...
## $ Embarked    : chr   "Q" "S" "Q" "S" ...
```

Queremos criar e preencher a coluna Survived (sobreviventes) `a partir do meu modelo de predicao no BD teste. Mas antes eu preciso treinar o modelo à partir do banco de dados treino.

## Separando as colunas que queremos trabalhar no BD teste

```
teste <- treino[,c('PassengerId', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked')]
```

Trabalhando com a coluna Age - procurando o na e substituindo pelo valor medio dos valores os valores que sao na

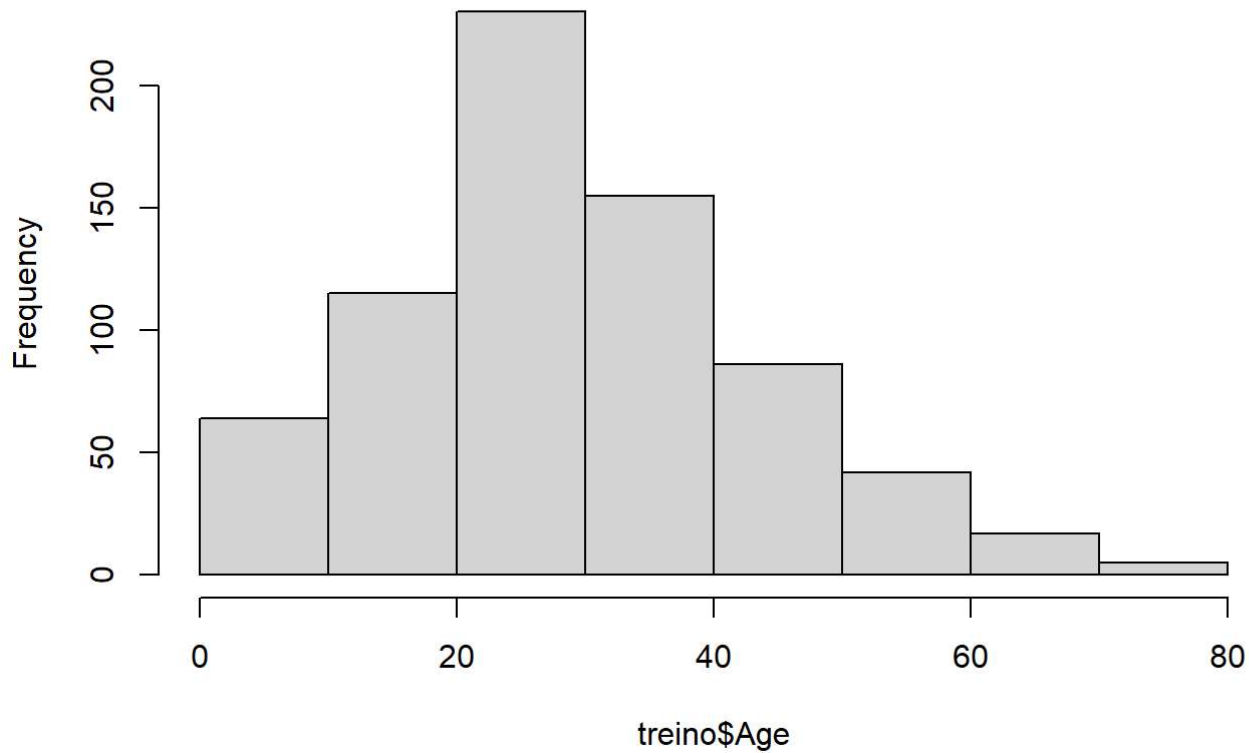
```
head(treino$Age, 10)
```

```
## [1] 22 38 26 35 35 NA 54 2 27 14
```

## Análise1

```
hist(treino$Age)
```

## Histogram of treino\$Age



```
mean(treino$Age)
```

```
## [1] NA
```

```
median(treino$Age)
```

```
## [1] NA
```

```
# Nao consigo medir as medias por causa dos NAs
```

## Analise2

```
mean(treino$Age)
```

```
## [1] NA
```

```
# Nao conseguimos calcular por causa dos NAs  
mean(is.na(treino$Age)==F)
```

```
## [1] 0.8013468
```

```
median(is.na(treino$Age)==F)
```

```
## [1] TRUE
```

```
#Acertando o valor da media das idades
```

```
mean(treino$Age, na.rm = TRUE)
```

```
## [1] 29.69912
```

```
#Acertando o valor da mediana das idades
```

```
median(treino$Age, na.rm = TRUE)
```

```
## [1] 28
```

```
#Comparando com o summary (Calcula a media já sem os NAs)
```

```
summary(treino$Age)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##      0.42   20.12   28.00   29.70   38.00   80.00   177
```

## Acertando os valores de na

```
# Calcular a soma da qtd de na na coluna Age
```

```
sum(is.na(treino$Age))
```

```
## [1] 177
```

```
sum(is.na(treino$Parch))
```

```
## [1] 0
```

```
# Calcular a qtd total de na no bd em todas as colunas
```

```
colSums(is.na(treino))
```

```
## PassengerId   Survived    Pclass         Sex       Age      SibSp  
##           0           0           0           0       177           0  
##      Parch         Fare   Embarked  
##           0           0           0
```

```
# Podemos omitir os valores na do bd ( ele mostra as linhas que foram omitidas - mas nao vamos fazer)  
# na.omit(treino$Age)
```

## Atribuindo o valor médio das idades para os valores NAs da coluna Age (idade)

```
#treino$Age[which(is.na(treino$Age))] <- 29.7  
#View(treino)
```

```
treino$Age[which(is.na(treino$Age))] <- mean(treino$Age,na.rm = TRUE)  
View(treino)
```

Não existe outros NAs além da coluna Age, conforme visto acima.

# Criando o modelo

Vamos usar a Regressão Logística (pois o modelo não é linear)

```
regressao1 <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked, family=binomial, data=treino)

summary(regressao1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##      Fare + Embarked, family = binomial, data = treino)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6446  -0.5907  -0.4219   0.6210   2.4431
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  17.572941  610.227586   0.029   0.9770
## Pclass       -1.100058   0.143529  -7.664 1.80e-14 ***
## Sexmale      -2.718695   0.200783 -13.540 < 2e-16 ***
## Age          -0.039901   0.007854  -5.080 3.77e-07 ***
## SibSp        -0.325777   0.109384  -2.978  0.0029 **
## Parch        -0.092602   0.118708  -0.780   0.4353
## Fare          0.001918   0.002376   0.807   0.4194
## EmbarkedC    -12.287753  610.227400  -0.020   0.9839
## EmbarkedQ    -12.321829  610.227451  -0.020   0.9839
## EmbarkedS    -12.706570  610.227384  -0.021   0.9834
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  784.19  on 881  degrees of freedom
## AIC: 804.19
##
## Number of Fisher Scoring iterations: 13
```

## Buscando os resultados da coluna Survived (sobreviventes)

```
resultado1 <- predict(regressao1, newdata = teste)
resultado1
```

##	1	2	3	4	5	6
##	-2.342183171	2.479865636	0.543980532	2.145870754	-2.533581079	NA
##	7	8	9	10	11	12
##	-1.007534876	-2.261804655	0.325029974	2.258368495	1.020249892	1.503001726
##	13	14	15	16	17	18
##	-1.935070570	-3.437419585	1.022653126	0.502408188	-2.187397995	NA
##	19	20	21	22	23	24
##	0.038026788	NA	-1.399090395	-1.384126933	1.367829475	-0.001504013
##	25	26	27	28	29	30
##	0.217486227	-0.678607995	NA	-0.368530005	NA	NA
##	31	32	33	34	35	36
##	-0.076417979	NA	NA	-2.665744975	0.181062493	-0.854239311
##	37	38	39	40	41	42
##	NA	-1.974971271	0.230959295	1.122191479	-0.337432591	1.303442449
##	43	44	45	46	47	48
##	NA	2.534148004	1.207938936	NA	NA	NA
##	49	50	51	52	53	54
##	NA	0.556352246	-2.751381347	-1.975450833	2.051404522	1.233232294
##	55	56	57	58	59	60
##	-1.100821435	NA	1.848481637	-1.856984140	2.009001867	-3.315527466
##	61	62	63	64	65	66
##	-1.597629586	15.110116313	-0.913564520	-2.421116094	NA	NA
##	67	68	69	70	71	72
##	1.529276033	-1.894962123	-0.585223743	-2.824853048	-1.309121155	-0.796335883
##	73	74	75	76	77	78
##	-0.749363752	-2.069149639	-2.320947862	-2.135341373	NA	NA
##	79	80	81	82	83	84
##	-0.215132885	0.393105763	-2.013049634	-2.291395414	NA	0.020747677
##	85	86	87	88	89	90
##	2.008084440	-0.697452051	-2.328552724	NA	2.190562279	-2.094673372
##	91	92	93	94	95	96
##	-2.294176875	-1.935446163	-0.996242177	-2.661429480	-3.492732492	NA
##	97	98	99	100	101	102
##	-1.300039708	0.577653446	1.261148556	-1.684966295	0.464123118	NA
##	103	104	105	106	107	108
##	0.265357640	-2.452620671	-3.265175463	-2.254571969	0.742956516	NA
##	109	110	111	112	113	114
##	-2.653578974	NA	-0.727966213	1.108403504	-2.014871971	0.461252808
##	115	116	117	118	119	120
##	1.334436218	-1.975211052	-3.565890068	-1.495054038	0.891022305	0.058077883
##	121	122	123	124	125	126
##	-1.400916954	NA	-1.198489552	1.394419204	-1.051365479	-1.516702207
##	127	128	129	130	131	132
##	NA	-2.096415718	NA	-2.934650202	-2.035258587	-1.936988819
##	133	134	135	136	137	138
##	-0.607098293	1.233232294	-1.025020628	-0.522477989	2.873413276	-0.652625734
##	139	140	141	142	143	144
##	-1.773229746	0.660743161	NA	0.703247640	0.313207456	-1.512922238
##	145	146	147	148	149	150
##	-0.748593098	-1.158436696	-2.214863093	0.436272754	-1.644145622	-1.703332538
##	151	152	153	154	155	156
##	-2.063350011	2.690476225	-3.351545440	-2.925866401	NA	-0.543362576
##	157	158	159	160	161	162
##	1.327361165	-2.334077576	NA	NA	-2.969847566	1.100439135
##	163	164	165	166	167	168
##	-2.175002292	-1.814193541	-2.511977143	-1.657436882	NA	-0.872000705
##	169	170	171	172	173	174
##	NA	-2.161345060	-1.322063630	-2.267199396	1.129273676	-1.975211052
##	175	176	177	178	179	180
##	-0.709122397	-2.274023451	NA	2.245172429	-1.224524131	-2.588923685
##	181	182	183	184	185	186

##	NA	NA	-2.939706304	-0.761684626	1.263639081	NA
##	187	188	189	190	191	192
##	NA	-0.696984252	-2.752430958	-2.573777573	1.414369554	-0.785616425
##	193	194	195	196	197	198
##	0.497373023	-0.540646666	2.482674305	2.151952496	NA	-2.904808627
##	199	200	201	202	203	204
##	NA	1.733575159	-2.251494713	NA	-2.496661721	-2.535304106
##	205	206	207	208	209	210
##	-1.855269169	1.413862815	-2.724693234	-1.735060689	1.327393200	-0.070127657
##	211	212	213	214	215	216
##	-2.096591621	1.310013445	-2.016406570	-1.224524131	NA	2.839721083
##	217	218	219	220	221	222
##	0.504079831	-2.002253651	3.054653799	-1.229319754	-1.775467768	-1.104822029
##	223	224	225	226	227	228
##	-3.171992288	NA	-0.621743041	-2.012378247	-0.790412048	-1.956555520
##	229	230	231	232	233	234
##	-0.745715724	NA	2.204137572	-2.294704394	-2.380685323	-0.061408415
##	235	236	237	238	239	240
##	-0.989915551	NA	-2.083973301	2.212198993	-0.790412048	-1.345616963
##	241	242	243	244	245	246
##	NA	NA	-1.189419053	-2.016646352	-1.916843247	-0.802182489
##	247	248	249	250	251	252
##	0.583593495	1.551248356	-0.746274802	-2.482980307	NA	0.010767298
##	253	254	255	256	257	258
##	-1.375296162	-2.644412271	-0.216163666	0.671934256	NA	2.735220379
##	259	260	261	262	263	264
##	3.771380262	0.628492095	NA	-2.700302100	-1.292808815	-0.548410264
##	265	266	267	268	269	270
##	NA	-1.468723957	-3.110487652	-2.460878192	1.653849434	2.629966787
##	271	272	273	274	275	276
##	NA	-2.150015979	0.975129780	-0.045521367	NA	1.076335551
##	277	278	279	280	281	282
##	-0.214468473	NA	-2.386901498	-0.209862042	-3.346436215	-2.254651768
##	283	284	285	286	287	288
##	-1.772686306	-1.895169869	NA	-2.033787866	-2.331296114	-2.015167765
##	289	290	291	292	293	294
##	-1.703332538	1.087988996	2.880148575	3.275952415	-1.045351232	0.625556313
##	295	296	297	298	299	300
##	-2.094969166	NA	-1.657480637	3.466241325	NA	2.572299177
##	301	302	303	304	305	306
##	NA	NA	-1.910611775	NA	NA	0.790638996
##	307	308	309	310	311	312
##	NA	3.389938551	-1.110383107	3.097313101	3.387031253	3.133460361
##	313	314	315	316	317	318
##	1.260332307	-2.254571969	-2.136195126	0.543844720	1.432735797	-2.180222696
##	319	320	321	322	323	324
##	2.660442358	2.428727528	-2.016406570	-2.214671268	1.877665449	1.425689857
##	325	326	327	328	329	330
##	NA	3.008882971	-3.574476120	1.254766752	-0.049731721	3.565334983
##	331	332	333	334	335	336
##	NA	-0.713194017	-0.266831641	-2.407934390	NA	NA
##	337	338	339	340	341	342
##	-0.307523766	2.807205516	-2.932588084	-0.679815922	-0.500745965	2.150661579
##	343	344	345	346	347	348
##	-1.144722730	-1.025020628	-1.463928335	1.733575159	1.095163949	NA
##	349	350	351	352	353	354
##	-1.660079093	-2.811711055	-2.052518729	NA	-1.736703371	-2.441647745
##	355	356	357	358	359	360
##	NA	-2.251494713	2.901399047	1.174965351	NA	NA
##	361	362	363	364	365	366
##	-3.391192289	-1.063344985	0.124606649	-2.535499328	NA	-2.335612175
##	367	368	369	370	371	372



##	1.609659342	NA	NA	3.360447581	0.249491521	-2.184027113
##	373	374	375	376	377	378
##	-1.895169869	0.848797692	0.416989730	NA	0.702288516	0.609621246
##	379	380	381	382	383	384
##	-1.523998617	-1.895697388	2.945749943	1.790105132	-2.414118758	2.143760680
##	385	386	387	388	389	390
##	NA	-0.629661651	-2.916520460	1.254766752	NA	2.429778697
##	391	392	393	394	395	396
##	-0.669598341	-1.975458889	-2.906069158	3.158926687	0.455410393	-2.015359590
##	397	398	399	400	401	402
##	0.344341217	-1.837998101	-0.950014850	1.573300969	-2.693423662	-2.174474773
##	403	404	405	406	407	408
##	0.421352107	-2.565090432	0.784799444	-1.694557541	-3.172567762	-0.554553972
##	409	410	411	412	413	414
##	-1.975498789	NA	NA	NA	2.681196904	NA
##	415	416	417	418	419	420
##	-2.892927165	NA	0.953595322	1.787775186	-1.224524131	1.028311157
##	421	422	423	424	425	426
##	NA	-1.590837424	-2.294512569	0.058221105	-2.250317152	NA
##	427	428	429	430	431	432
##	1.273132994	1.958015900	NA	-2.413878977	-0.018672343	NA
##	433	434	435	436	437	438
##	0.714523186	-1.817142849	-1.165963744	2.926912158	-0.042535653	0.815245625
##	439	440	441	442	443	444
##	-1.697712504	-1.269220455	0.502698559	-1.932289109	-2.460878192	1.573972356
##	445	446	447	448	449	450
##	NA	0.859835395	2.092349396	-0.258076546	1.078296931	-0.968712072
##	451	452	453	454	455	456
##	-1.946614936	NA	0.322645039	-0.643552231	NA	-1.875655785
##	457	458	459	460	461	462
##	-1.494998264	NA	0.691361321	NA	-0.816686354	-2.493680378
##	463	464	465	466	467	468
##	-0.753862576	-1.942736741	NA	-2.655201429	NA	-1.135891959
##	469	470	471	472	473	474
##	NA	1.247874908	NA	-2.652108253	0.891782251	2.193811421
##	475	476	477	478	479	480
##	0.707251985	NA	-1.694557541	-2.621879782	-2.015887109	1.417363619
##	481	482	483	484	485	486
##	-3.235726065	NA	-3.132091587	-0.929156300	0.317853125	NA
##	487	488	489	490	491	492
##	2.216654147	-0.790833991	-2.334077576	-1.899483297	NA	-1.976505870
##	493	494	495	496	497	498
##	-1.088414174	-1.271553709	-1.974971271	NA	1.854850328	NA
##	499	500	501	502	503	504
##	2.548525212	-2.095160991	-1.814193541	1.127889697	NA	0.108261915
##	505	506	507	508	509	510
##	3.293830187	0.631342764	1.214201916	NA	-2.226509518	-2.081543659
##	511	512	513	514	515	516
##	-1.910010994	NA	-0.338381488	1.818659296	-2.095736466	-0.762454798
##	517	518	519	520	521	522
##	1.329772530	NA	0.953927390	-2.414174771	2.748648123	-2.015167765
##	523	524	525	526	527	528
##	NA	2.448115367	NA	-2.368869051	0.691361321	NA
##	529	530	531	532	533	534
##	-2.693423662	-1.692251891	2.217949121	NA	-1.816504772	NA
##	535	536	537	538	539	540
##	0.385792438	2.252099693	-0.696984252	3.192258377	NA	3.217063473
##	541	542	543	544	545	546
##	2.280879138	-0.221227021	-0.301028422	-1.605164894	-0.650227321	-1.456152600
##	547	548	549	550	551	552
##	1.632239299	NA	-2.848228208	-0.719528990	0.815620353	-1.079884790
##	553	554	555	556	557	558

##	NA	-1.597637643	0.703295597	-1.375296162	2.020082167	NA
##	559	560	561	562	563	564
##	1.944595378	-0.162627664	NA	-2.733380375	-1.143763605	NA
##	565	566	567	568	569	570
##	NA	-2.715342763	-1.895465663	0.079095054	NA	-2.414254570
##	571	572	573	574	575	576
##	-2.506142172	1.098772444	-0.338189663	NA	-1.775467768	-1.882797162
##	577	578	579	580	581	582
##	1.334568153	1.991639049	NA	-2.414118758	1.307906005	2.423325513
##	583	584	585	586	587	588
##	-2.157203706	0.106979169	NA	3.015684603	-1.898999542	-1.194060748
##	589	590	591	592	593	594
##	-2.014871971	NA	-2.535355459	1.934651729	-3.013924085	NA
##	595	596	597	598	599	600
##	-1.804668397	-2.960976657	NA	-3.107632792	NA	-0.705271897
##	601	602	603	604	605	606
##	1.016275357	NA	NA	-2.892687384	0.120839637	-2.884871511
##	607	608	609	610	611	612
##	-2.334373370	0.028805442	1.776034693	2.464664132	-0.718724499	NA
##	613	614	615	616	617	618
##	NA	NA	-2.533581079	1.322343337	-2.899878185	0.233885618
##	619	620	621	622	623	624
##	1.837308358	-1.069716952	-2.109050340	-0.853176217	-1.919877778	-1.975346864
##	625	626	627	628	629	630
##	-1.959529365	-1.324325629	-1.918348553	3.077941575	-2.174770567	NA
##	631	632	633	634	635	636
##	-2.086890813	-3.173902480	0.248118823	NA	0.098075489	1.573972356
##	637	638	639	640	641	642
##	-2.414118758	-1.657386719	-0.456612028	NA	-1.935446163	3.360447581
##	643	644	645	646	647	648
##	0.377380393	NA	1.247874908	-0.627389863	-1.895465663	-0.699906745
##	649	650	651	652	653	654
##	NA	0.662963290	NA	1.899559765	-1.974236006	NA
##	655	656	657	658	659	660
##	1.245673549	-1.520619056	NA	0.285469732	-0.945219227	-0.815720494
##	661	662	663	664	665	666
##	-1.342596473	-2.315850253	-0.778631968	-2.574544873	-2.261086952	-1.839824660
##	667	668	669	670	671	672
##	-1.025020628	NA	-2.852786683	NA	0.726659738	-0.415331604
##	673	674	675	676	677	678
##	-2.825347777	-1.264424832	NA	-1.855796687	-2.114623723	0.866862844
##	679	680	681	682	683	684
##	-0.940956746	0.920182387	NA	0.536301450	-1.932816627	-3.435229568
##	685	686	687	688	689	690
##	-2.790049359	-1.062362495	-3.030686251	-1.891101647	-1.855756788	3.480598228
##	691	692	693	694	695	696
##	-0.405740359	1.758545189	NA	-1.717339744	-1.295494761	-2.101380419
##	697	698	699	700	701	702
##	-2.892687384	NA	-0.694376578	-2.813653282	3.577590155	-0.298480787
##	703	704	705	706	707	708
##	1.201925565	-1.750424113	-2.500626968	-1.558693197	0.896619571	-0.577785691
##	709	710	711	712	713	714
##	3.179208091	NA	3.322474305	NA	-1.093643514	-2.291427449
##	715	716	717	718	719	720
##	-2.102339544	-1.895937169	3.105352745	1.609077434	NA	-2.454307196
##	721	722	723	724	725	726
##	2.397550664	-2.143055262	-1.384126933	-2.022538143	-0.253618728	-1.933895642
##	727	728	729	730	731	732
##	0.532187146	NA	-1.325859990	0.258104631	3.014590508	-1.136550181
##	733	734	735	736	737	738
##	NA	-0.945219227	-0.945219227	-2.258784619	-0.886680056	1.052685175
##	739	740	741	742	743	744

##	NA	NA	NA	-0.563330118	3.013758260	-2.405008068
##	745	746	747	748	749	750
##	-2.374218057	-2.027614281	-2.170443817	1.494170955	0.065586877	-1.989812395
##	751	752	753	754	755	756
##	2.132392973	-1.460574598	-2.450998216	-2.055068466	0.364726524	-0.469737899
##	757	758	759	760	761	762
##	-2.254763793	-0.748593098	-2.493680378	2.615518277	NA	-2.774759662
##	763	764	765	766	767	768
##	-1.517828185	2.049096745	-1.775995286	1.555143957	NA	0.748833041
##	769	770	771	772	773	774
##	NA	-2.413279524	-2.091891911	-3.052665779	0.412056417	NA
##	775	776	777	778	779	780
##	-0.047846232	-1.855844644	NA	1.390623277	NA	2.363378612
##	781	782	783	784	785	786
##	1.480171805	2.871564536	-0.051955084	NA	-2.136492322	-2.136108672
##	787	788	789	790	791	792
##	0.862362824	-2.426802198	-1.663911965	-0.217072251	NA	-0.640977084
##	793	794	795	796	797	798
##	NA	NA	-2.134869867	-1.583630436	1.860917183	0.345931637
##	799	800	801	802	803	804
##	-1.916835191	-0.002877367	-1.384126933	1.061308367	0.327919173	-0.826704810
##	805	806	807	808	809	810
##	-2.216437592	-2.374505795	-0.508509564	0.862898399	-1.583630436	2.225672155
##	811	812	813	814	815	816
##	-2.174786489	-2.662300070	-1.428823257	-0.101524919	-2.354027926	NA
##	817	818	819	820	821	822
##	0.663682633	-1.217940600	-2.855855882	-2.660520297	1.452454022	-2.213200546
##	823	824	825	826	827	828
##	-0.468608863	0.420205777	-2.551877844	NA	NA	0.212254930
##	829	830	831	832	833	834
##	NA	14.152499499	1.088453153	-0.467969452	NA	-2.055148265
##	835	836	837	838	839	840
##	-1.854789607	2.370142056	-1.973796343	NA	-2.320947862	NA
##	841	842	843	844	845	846
##	-1.935310351	-0.670709946	3.047574435	-2.097907021	-1.814193541	-2.813845107
##	847	848	849	850	851	852
##	NA	-2.115059989	-1.198959835	NA	-2.740418604	-4.090235919
##	853	854	855	856	857	858
##	1.236773755	3.110878564	0.634721785	0.773317553	1.868658037	-0.936388456
##	859	860	861	862	863	864
##	0.786532646	NA	-3.412917155	-1.194071801	1.900817883	NA
##	865	866	867	868	869	870
##	-0.985119928	1.015362548	1.708559773	-0.092440434	NA	-1.709123512
##	871	872	873	874	875	876
##	-2.174770567	1.573413278	-0.259514115	-3.010567149	1.688113380	1.400362348
##	877	878	879	880	881	882
##	-1.931625778	-1.895465663	NA	2.017606747	1.626009609	-2.454075471
##	883	884	885	886	887	888
##	0.708554860	-1.149518353	-2.136492322	-0.012330778	-1.104822029	3.065747008
##	889	890	891			
##	NA	0.486563902	-2.029713096			

Criando a variavel Survived para a separacao entre 0 e 1

```
teste$Survived <-ifelse(resultado1 <=0.5, 0, 1)
```

```
head(resultado1,10)
```

```
##          1          2          3          4          5          6          7
## -2.3421832  2.4798656  0.5439805  2.1458708 -2.5335811      NA -1.0075349
##          8          9         10
## -2.2618047  0.3250300  2.2583685
```

```
head(teste$Survived,10)
```

```
## [1]  0  1  1  1  0 NA  0  0  0  1
```

## Criando o arquivo csv

```
write.csv(teste, 'predicao1.csv')
```