01.survRandomForest

July 6, 2021

1 Random Survival Forest

Customer churn/attrition, a.k.a the percentage of customers that stop using paying services, is one of the most important metrics for a business, as it usually costs more to acquire new customers than it does to retain existing ones. Indeed, according to a study by Bain & Company, existing customers tend to buy more from a company over time, thus reducing the operating costs of the business and may refer the products they use to others. For example, in financial services, a 5% increase in customer retention produces more than a 25% increase in profit. By using Survival Analysis, not only companies can predict if customers are likely to stop doing business but also when that event might happen.

1.1 Methods

In this study, we adopt random survival forests which have never been used in understanding factors affecting membership in a sport club using existing data in a Sport Club. The analysis is based on the use of random survival forests in the presence of covariates that do not necessarily satisfy the PH assumption. Random Survival Forests does not make the proportional hazards assumption (Ehrlinger, 2016) and has the flexibility to model survivor curves that are of dissimilar shapes for contrastinggroups of subjects. Random Survival Forest is an extension of Random Forest allowing efficient non-parametric analysis of time to event data (Breiman, 2001). This characteristcs allow us to surpass the Cox Regression limitation of the proportional hazard assumption, requiring to exclude variables which not fullfill the model assumption. It was shown by (Breiman, 2001) that ensemble learning can be further improved by injecting randomization into the base learning process - a method called Random Forests.
The random survival forest was developed using the package PySurvival (Fotso & Others, 2019) The most relevant variables predicting the dropout are analysed using the log-rank test. The metric variables are transformed to categorical using the quartiles to provide a statistical comparison of groups. The survival analysis was conducted using the package Lifelines (Davidson-Pilon et al., 2017).

1.2 Results

The initial model has a c-index of 0.92. After removing estadoCivil_outro and ano c-index improved to 0.94. Without idade improved to 0.95. The most relevant variables predicting the dropout are: - mesesUP - valorTotal - anoUltimoPagamento - quotaMensal - escaloesTotalJogos_ate 1 - jogosEpoca - escaloesTotalJogos_56 a 105 - estadoCivil_solteiro - escaloesTotalJogos_21 a 56 - escaloesTotalJogos_mais 105 - sexo_M - estadoCivil_nao definido

TODO: COLOCAR ESTA DESCRIÇÃO PARA TODAS AS VARIÁVEIS: There were identified significative differences between the gender groups (2=194.63, p < .005), wrenew two or more contracts, the survival probability for 12 months is 85.49%

1.3 Methods bibliography

- Ehrlinger, J. (2016). ggRandomForests: Exploring Random Forest Survival. ArXiv:1612.08974 [Stat]. http://arxiv.org/abs/1612.08974
- Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32. https://doi.org/10/d8zjwq
- Fotso, S., & others. (2019). PySurvival: Open source package for Survival Analysis modeling. https://www.pysurvival.io/

2 Dataset

Considering the sport club policies all the customers with payments less than 24 months where considered active:
- dt['abandonou'] = 0 - dt.loc[dt['mesesUP']>=24,'abandonou']=1

The variables extracted from the software correspond to the time interval of becoming a customer until the end of observation (censoring on 31 Maio 2019) or the end of the customer relationship (dropout). The survival time in the dataset is represented by the number of years the customer begin affilated.

We extracted records of 25316 customers (male n=17246, female n=8070) from a sport club; data corresponded to the time period between October 1, 1944 and May 31, 2019.

[51]:
```
from IPython.display import HTML
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
import datetime
import seaborn as sns

df = pd.read_excel('../data/membershipData.xlsx',index_col=0)
```

3 Check file

[52]:
```
df.columns
```

[52]:
```
Index(['dataAdesao', 'ano', 'dataNascimento', 'idade', 'sexo', 'estadoCivil',
       'categoria', 'quotaMensal', 'profissao', 'codPostal', 'ultimaQuota',
       'ultimoPagamento', 'valorTotal', 'totalJogos', 'jogosEpoca',
       'diasUltimoPagamento', 'mesesUP', 'abandonou', 'anosSocio',
       'idaEstadio', 'escaloesTotalJogos', 'mes'],
      dtype='object')
```

[53]:
```
df.columns = ['dtInscription','year','birthDate','age','sex','maritalStatus',
           ␣
↪'category','monthlyFee','occupation','zipCode','dtLastInvoice','dtLastPayment','totalAmount
           ␣
↪'totalMatches','seasonMatches','daysSinceLastPayment','monthsSinceLastPayment','dropout',
           ␣
↪'yearsMembership','stadiumAccess','quartStadiumEntries','inscriptionMonth']
```

3.1 Description

Variables:

- 'dtInscription': Inscription Date
- 'inscriptionYear': Inscription year
- 'birthDate': Birth date of the member
- 'age': age in years
- 'sex': male or female
- 'maritalStatus': single, undefined, married, other
- 'category': student, male, female, under_14, athlete, retired, other
- 'monthlyFee': value of the monthly fee in euros (1€, 2.5€, 5€, 6€, 10€)
- 'occupation': student, retired, businessman, teacher, …
- 'zipCode': Zip Code
- 'dtLastInvoice': Last invoice date
- 'dtLastPayment': Last payment date
- 'totalAmount': sum of invoice values
- 'totalMatches': number of matches attended in the stadium
- 'seasonMatches': number of matches attended in the current season
- 'daysSinceLastPayment': number of days since the last payment
- 'monthsSinceLastPayment': number of months since the last payment
- 'dropout': dropout yes (1) no (0) - censured data
- 'yearsMembership': number of years membership
- 'stadiumAccess': the member go to the stadium: yes (1) and no (0)
- 'quartStadiumEntries': quartiles number of access to the stadium
- 'inscriptionMonth': inscription month

Ei is the event indicator such that Ei=1, if an event happens and Ei=0 in case of censoring

As variáveis categóricas foram transformadas em dummies: - sexo - estadoCivil - escaloesTotalJogos

[54]:
```
df.sex.value_counts()
```

[54]:
```
M    17246
F     8070
Name: sex, dtype: int64
```

[55]:
```
df.describe().T
```

[55]:

```
                        count         mean         std     min          25%  \
year                  25316.0  2007.048033   10.937818  1944.0  2004.000000
age                   25316.0    27.262996   20.087078   -70.0    13.000000
monthlyFee            25316.0     4.356099    3.550837     0.0     1.000000
totalAmount           25316.0   316.037984  493.971528     0.0     5.000000
totalMatches          25316.0    26.535946   45.812996     0.0     0.000000
seasonMatches         25316.0     2.171631    4.076356     0.0     0.000000
daysSinceLastPayment  25316.0   586.277033  990.398069     0.0    83.994934
monthsSinceLastPayment  25316.0  18.814110  32.498248     0.0     2.000000
dropout               25316.0     0.221638    0.415357     0.0     0.000000
yearsMembership       25316.0    11.264339   10.908777     0.0     5.000000
stadiumAccess         25316.0     0.401367    0.490185     0.0     0.000000
inscriptionMonth      25316.0     6.875454    3.391117     1.0     4.000000

                              50%        75%          max
year                  2010.000000  2014.00000  2019.000000
age                     19.000000    41.00000   118.000000
monthlyFee               2.500000     6.00000    10.000000
totalAmount             53.000000   448.25000  2602.000000
totalMatches             0.000000    36.00000   197.000000
seasonMatches            0.000000     2.00000    16.000000
daysSinceLastPayment   122.113031   534.98242  4778.034828
monthsSinceLastPayment   4.000000    17.00000   156.000000
dropout                  0.000000     0.00000     1.000000
yearsMembership          8.000000    14.00000    74.000000
stadiumAccess            0.000000     1.00000     1.000000
inscriptionMonth         8.000000     9.00000    12.000000
```

[56]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 25316 entries, 1 to 25316
Data columns (total 22 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   dtInscription           25316 non-null  datetime64[ns]
 1   year                    25316 non-null  int64
 2   birthDate               25316 non-null  object
 3   age                     25316 non-null  int64
 4   sex                     25316 non-null  object
 5   maritalStatus           25316 non-null  object
 6   category                25316 non-null  object
 7   monthlyFee              25316 non-null  float64
 8   occupation              25316 non-null  object
 9   zipCode                 25316 non-null  object
 10  dtLastInvoice           25316 non-null  object
 11  dtLastPayment           25316 non-null  object
```

```
 12  totalAmount            25316 non-null  float64
 13  totalMatches           25316 non-null  int64
 14  seasonMatches          25316 non-null  int64
 15  daysSinceLastPayment   25316 non-null  float64
 16  monthsSinceLastPayment 25316 non-null  int64
 17  dropout                25316 non-null  int64
 18  yearsMembership        25316 non-null  int64
 19  stadiumAccess          25316 non-null  int64
 20  quartStadiumEntries    25316 non-null  object
 21  inscriptionMonth       25316 non-null  int64
dtypes: datetime64[ns](1), float64(3), int64(9), object(9)
memory usage: 4.4+ MB
```

[57]:
```
df.head().T
```

[57]:
```
Sócio                                        1  \
dtInscription              1944-10-01 00:00:00
year                                      1944
birthDate                  1935-05-11 00:00:00
age                                         83
sex                                          M
maritalStatus                           casado
category                                 homem
monthlyFee                                10.0
occupation                              MEDICO
zipCode                               4715-196
dtLastInvoice                          2019-12
dtLastPayment       2019-01-21 10:45:33.540000
totalAmount                             1906.0
totalMatches                                 0
seasonMatches                                0
daysSinceLastPayment                103.308984
monthsSinceLastPayment                       3
dropout                                      0
yearsMembership                             74
stadiumAccess                                0
quartStadiumEntries                      ate 1
inscriptionMonth                            10

Sócio                                        2  \
dtInscription              1944-10-01 00:00:00
year                                      1944
birthDate                  1930-09-29 00:00:00
age                                         88
sex                                          M
maritalStatus                         solteiro
category                                 homem
```

```
monthlyFee                               10.0
occupation                             MEDICO
zipCode                              4715-196
dtLastInvoice                         2019-12
dtLastPayment      2019-01-11 16:49:24.640000
totalAmount                            1906.0
totalMatches                                0
seasonMatches                               0
daysSinceLastPayment               113.056309
monthsSinceLastPayment                      3
dropout                                     0
yearsMembership                            74
stadiumAccess                               0
quartStadiumEntries                     ate 1
inscriptionMonth                           10

Sócio                                       3  \
dtInscription             1945-08-24 00:00:00
year                                     1945
birthDate                 1945-08-24 00:00:00
age                                        73
sex                                         M
maritalStatus                    nao definido
category                                homem
monthlyFee                               10.0
occupation                 GERENTE INDUSTRIAL
zipCode                            4700 - 699
dtLastInvoice                         2016-12
dtLastPayment      2016-04-29 17:25:33.810000
totalAmount                            1553.0
totalMatches                                0
seasonMatches                               0
daysSinceLastPayment              1100.031203
monthsSinceLastPayment                     36
dropout                                     1
yearsMembership                            73
stadiumAccess                               0
quartStadiumEntries                     ate 1
inscriptionMonth                            8

Sócio                                       4                            5
dtInscription             1945-09-01 00:00:00          1945-09-01 00:00:00
year                                     1945                         1945
birthDate                 1921-05-27 00:00:00          1921-03-08 00:00:00
age                                        97                           97
sex                                         M                            M
maritalStatus                          casado                        outro
```

```
category                                       reformado                       homem
monthlyFee                                           5.0                        10.0
occupation                                     REFORMADO                   REFORMADO
zipCode                                         4740-033                    4700-055
dtLastInvoice                                    2018-12                     2016-04
dtLastPayment                 2018-08-12 19:28:16.463000  2016-05-09 19:32:00.657000
totalAmount                                        790.0                      1466.0
totalMatches                                           0                           0
seasonMatches                                          0                           0
daysSinceLastPayment                          264.945987                 1089.943393
monthsSinceLastPayment                                 8                          35
dropout                                                0                           1
yearsMembership                                       73                          73
stadiumAccess                                          0                           0
quartStadiumEntries                                ate 1                       ate 1
inscriptionMonth                                       9                           9
```

[58]:
```
df.tail().T
```

[58]:
```
Sócio                                              25312  \
dtInscription                        2019-02-21 00:00:00
year                                                2019
birthDate                            2011-04-14 00:00:00
age                                                    7
sex                                                    M
maritalStatus                                   solteiro
category                                           sub14
monthlyFee                                           1.0
occupation                                             0
zipCode                                         4710-411
dtLastInvoice                                    2020-01
dtLastPayment                 2019-02-21 11:03:14.367000
totalAmount                                         17.0
totalMatches                                           0
seasonMatches                                          0
daysSinceLastPayment                           72.296706
monthsSinceLastPayment                                 2
dropout                                                0
yearsMembership                                        0
stadiumAccess                                          0
quartStadiumEntries                                ate 1
inscriptionMonth                                       2

Sócio                                              25313  \
dtInscription                        2019-02-21 00:00:00
year                                                2019
birthDate                            2010-05-26 00:00:00
```

```
age                                             8
sex                                             M
maritalStatus                            solteiro
category                                   atleta
monthlyFee                                    1.0
occupation                              ESTUDANTE
zipCode                                  4715-404
dtLastInvoice                             2020-01
dtLastPayment          2019-02-21 11:41:59.797000
totalAmount                                  12.0
totalMatches                                    0
seasonMatches                                   0
daysSinceLastPayment                    72.269791
monthsSinceLastPayment                          2
dropout                                         0
yearsMembership                                 0
stadiumAccess                                   0
quartStadiumEntries                         ate 1
inscriptionMonth                                2

Sócio                                       25314  \
dtInscription                 2019-02-21 00:00:00
year                                         2019
birthDate                     2016-11-24 00:00:00
age                                             2
sex                                             M
maritalStatus                            solteiro
category                                    sub14
monthlyFee                                    1.0
occupation                              ESTUDANTE
zipCode                                4715 - 586
dtLastInvoice                             2020-01
dtLastPayment          2019-02-21 15:53:35.253000
totalAmount                                  17.0
totalMatches                                    0
seasonMatches                                   0
daysSinceLastPayment                    72.095076
monthsSinceLastPayment                          2
dropout                                         0
yearsMembership                                 0
stadiumAccess                                   0
quartStadiumEntries                         ate 1
inscriptionMonth                                2

Sócio                                       25315                         25316
dtInscription                 2019-02-21 00:00:00           2019-02-21 00:00:00
year                                         2019                          2019
```

```
birthDate                  2004-06-30 00:00:00         1990-05-23 00:00:00
age                                         14                          28
sex                                          M                           M
maritalStatus                         solteiro                    solteiro
category                                 sub14                       homem
monthlyFee                                 1.0                        10.0
occupation                           ESTUDANTE                ADJ. COZINHA
zipCode                              4715 -028                    4700-277
dtLastInvoice                          2020-01                     2019-04
dtLastPayment       2019-02-21 18:55:03.097000  2019-02-21 18:58:32.137000
totalAmount                               17.0                         0.0
totalMatches                                 0                           0
seasonMatches                                0                           0
daysSinceLastPayment                 71.969059                   71.966639
monthsSinceLastPayment                       2                           2
dropout                                      0                           0
yearsMembership                              0                           0
stadiumAccess                                0                           0
quartStadiumEntries                      ate 1                       ate 1
inscriptionMonth                             2                           2
```

3.2 Convert dateLastPayment to date data type

[59]:
```
df['dtLastPayment'] = pd.to_datetime(df['dtLastPayment'],format='%Y-%m-%d %H:
 ↪%M', errors='coerce')
```

[60]:
```
df['yearLastPayment']=df['dtLastPayment'].apply(lambda x: x.year)
```

[61]:
```
df.yearLastPayment.unique()
```

[61]:
```
array([2019., 2016., 2018., 2015., 2017., 2014., 2008.,   nan, 2009.,
       2007., 2010., 2011., 2012., 2013., 2006.])
```

[62]:
```
df.yearLastPayment=df.yearLastPayment.fillna(0)
```

[63]:
```
df['yearLastPayment']=df.yearLastPayment.astype(int)
```

[64]:
```
df.columns
```

[64]:
```
Index(['dtInscription', 'year', 'birthDate', 'age', 'sex', 'maritalStatus',
       'category', 'monthlyFee', 'occupation', 'zipCode', 'dtLastInvoice',
       'dtLastPayment', 'totalAmount', 'totalMatches', 'seasonMatches',
       'daysSinceLastPayment', 'monthsSinceLastPayment', 'dropout',
       'yearsMembership', 'stadiumAccess', 'quartStadiumEntries',
       'inscriptionMonth', 'yearLastPayment'],
      dtype='object')
```

[65]:
```
df.
↪drop(columns=['dtInscription','dtLastPayment','birthDate','inscriptionMonth','occupation',␣
↪'zipCode','category','dtLastInvoice','daysSinceLastPayment'],inplace=True)
```

[66]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 25316 entries, 1 to 25316
Data columns (total 14 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   year                   25316 non-null  int64
 1   age                    25316 non-null  int64
 2   sex                    25316 non-null  object
 3   maritalStatus          25316 non-null  object
 4   monthlyFee             25316 non-null  float64
 5   totalAmount            25316 non-null  float64
 6   totalMatches           25316 non-null  int64
 7   seasonMatches          25316 non-null  int64
 8   monthsSinceLastPayment 25316 non-null  int64
 9   dropout                25316 non-null  int64
 10  yearsMembership        25316 non-null  int64
 11  stadiumAccess          25316 non-null  int64
 12  quartStadiumEntries    25316 non-null  object
 13  yearLastPayment        25316 non-null  int64
dtypes: float64(2), int64(9), object(3)
memory usage: 2.9+ MB
```

[67]:
```
df.maritalStatus.value_counts()
```

[67]:
```
solteiro        12065
nao definido     7667
casado           5085
outro             499
Name: maritalStatus, dtype: int64
```

4 Dropout event

Ei (event of interest - Dropout) is the event indicator such that Ei=1, if an event happens and Ei=0 in case of censoring

4.1 Converting from categorical to numerical

There are several categorical features that need to be encoded into one-hot vectors:

- sex
- maritalStatus

- quartStadiumEntries

[68]:
```
dfCurvas = df.copy()
df = pd.get_dummies(df,␣
↪columns=['sex','maritalStatus','quartStadiumEntries'],drop_first=True)
```

[69]:
```
# Creating the time and event columns
time_column = 'yearsMembership'
event_column = 'dropout'

# Extracting the features
features = np.setdiff1d(df.columns, [time_column, event_column] ).tolist()
```

4.2 check null values and duplicates

[70]:
```
# Checking for null values
N_null = sum(df[features].isnull().sum())
print(f"The raw_dataset contains {N_null} null values") #0 null values
```

```
The raw_dataset contains 0 null values
```

Change this to fstring pyton

[71]:
```
# Removing duplicates if there exist
N_dupli = sum(df.duplicated(keep='first'))
df = df.drop_duplicates(keep='first').reset_index(drop=True)
print(f"The raw_dataset contains {N_dupli} duplicates")

# Number of samples in the dataset
N = df.shape[0]
```

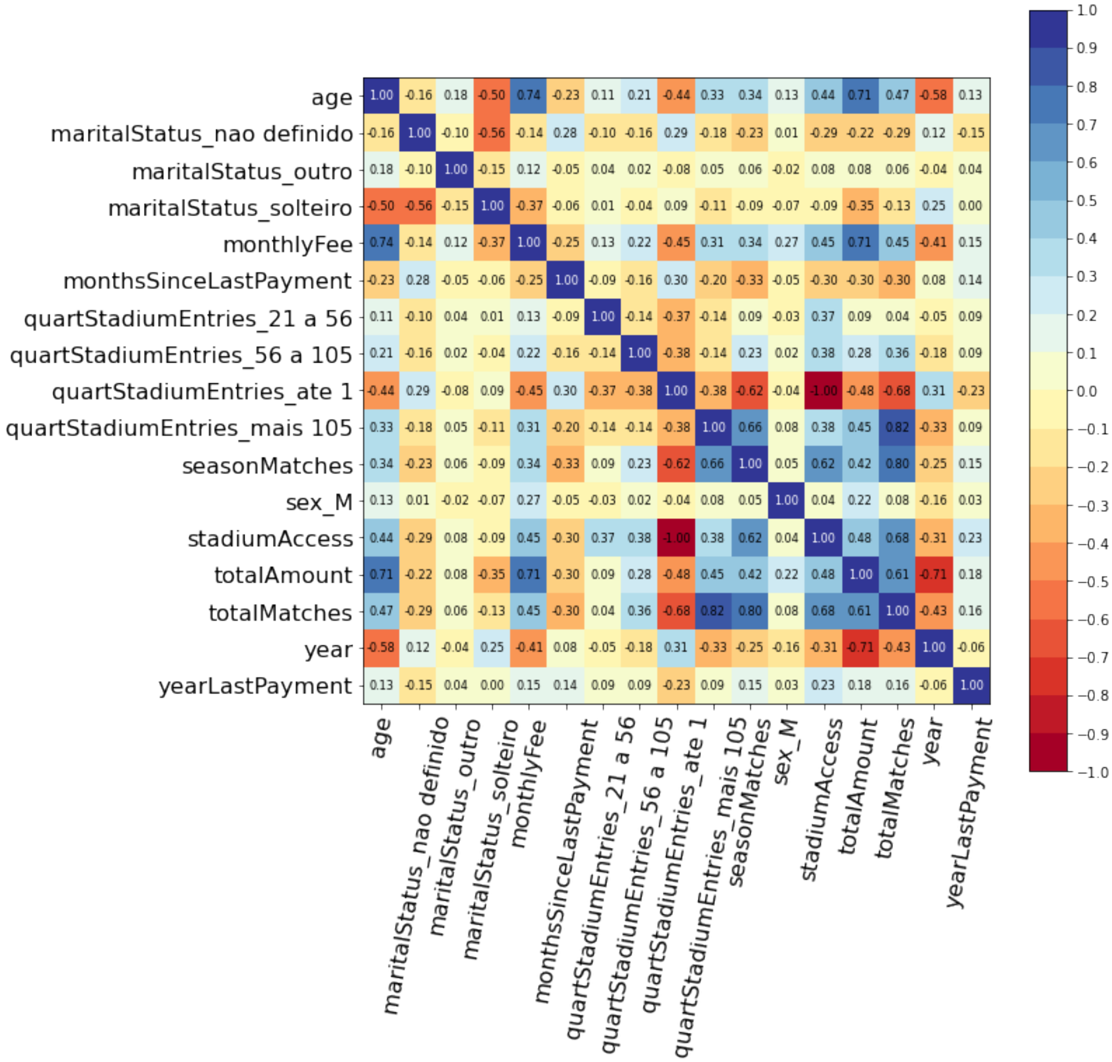
```
The raw_dataset contains 4928 duplicates
```

5 Exploratory Data Analysis

[72]:
```
df.columns
```

[72]:
```
Index(['year', 'age', 'monthlyFee', 'totalAmount', 'totalMatches',
       'seasonMatches', 'monthsSinceLastPayment', 'dropout', 'yearsMembership',
       'stadiumAccess', 'yearLastPayment', 'sex_M',
       'maritalStatus_nao definido', 'maritalStatus_outro',
       'maritalStatus_solteiro', 'quartStadiumEntries_21 a 56',
       'quartStadiumEntries_56 a 105', 'quartStadiumEntries_ate 1',
       'quartStadiumEntries_mais 105'],
      dtype='object')
```

```
[73]: from pysurvival.utils.display import correlation_matrix
      correlation_matrix(df[features], figure_size=(10,10), text_fontsize=8)
```



Vamos remover as variáveis com correlações maiores

```
[74]: #to_remove = ['totalJogos', 'idaEstadio']
      #features = np.setdiff1d(features, to_remove).tolist()
```

6 Modeling

So as to perform cross-validation later on and assess the performances of the model, let's split the dataset into training and testing sets.

6.1 Building the model

[75]:
```
# Building training and testing sets
from sklearn.model_selection import train_test_split
index_train, index_test = train_test_split( range(N), test_size = 0.4)
data_train = df.loc[index_train].reset_index( drop = True )
data_test  = df.loc[index_test].reset_index( drop = True )

# Creating the X, T and E inputs
X_train, X_test = df[features], data_test[features]
T_train, T_test = df[time_column], data_test[time_column]
E_train, E_test = df[event_column], data_test[event_column]
```

[76]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20388 entries, 0 to 20387
Data columns (total 19 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   year                          20388 non-null  int64
 1   age                           20388 non-null  int64
 2   monthlyFee                    20388 non-null  float64
 3   totalAmount                   20388 non-null  float64
 4   totalMatches                  20388 non-null  int64
 5   seasonMatches                 20388 non-null  int64
 6   monthsSinceLastPayment        20388 non-null  int64
 7   dropout                       20388 non-null  int64
 8   yearsMembership               20388 non-null  int64
 9   stadiumAccess                 20388 non-null  int64
 10  yearLastPayment               20388 non-null  int64
 11  sex_M                         20388 non-null  uint8
 12  maritalStatus_nao definido    20388 non-null  uint8
 13  maritalStatus_outro           20388 non-null  uint8
 14  maritalStatus_solteiro        20388 non-null  uint8
 15  quartStadiumEntries_21 a 56   20388 non-null  uint8
 16  quartStadiumEntries_56 a 105  20388 non-null  uint8
 17  quartStadiumEntries_ate 1     20388 non-null  uint8
 18  quartStadiumEntries_mais 105  20388 non-null  uint8
dtypes: float64(2), int64(9), uint8(8)
memory usage: 1.9 MB
```

[77]:
```
#from pysurvival.models.survival_forest import ConditionalSurvivalForestModel
from pysurvival.models.survival_forest import RandomSurvivalForestModel
# Fitting the model
csf = RandomSurvivalForestModel(num_trees=200)
csf.fit(X_train, T_train, E_train, max_features='sqrt',
```

```
        max_depth=5, min_node_size=20)
```

[77]: RandomSurvivalForestModel

6.1.1 Features importance

[78]:
```
csf.variable_importance_table
```

[78]:

	feature	importance	pct_importance
0	monthsSinceLastPayment	26.789716	0.352602
1	yearLastPayment	12.012238	0.158103
2	totalAmount	8.330971	0.109651
3	seasonMatches	5.155454	0.067855
4	totalMatches	4.137202	0.054453
5	monthlyFee	4.020849	0.052922
6	sex_M	3.703763	0.048748
7	maritalStatus_solteiro	3.537542	0.046561
8	quartStadiumEntries_ate 1	3.110066	0.040934
9	quartStadiumEntries_56 a 105	2.383092	0.031366
10	stadiumAccess	2.108755	0.027755
11	quartStadiumEntries_21 a 56	0.392324	0.005164
12	age	0.295135	0.003885
13	maritalStatus_nao definido	-0.130668	0.000000
14	quartStadiumEntries_mais 105	-0.131286	0.000000
15	maritalStatus_outro	-1.072368	0.000000
16	year	-13.855385	0.000000

A negative number means that the model performs better without estadoCivil_Outro and ano: https://stackoverflow.com/questions/27918320/what-does-negative-incmse-in-randomforest-package-mean

The variable monthsSinceLastPayment explains the survival 26.7%%, year last payment 12.%, totalAmount 8%, number of games 5.1%….

6.1.2 Model performance

We are going to access the model performance using the training and test set. Previously defined. C-index close to 1, the model has an powerfull discriminatory; but if it is close to 0.5, it has no ability to discriminate between low and high risk subjects.

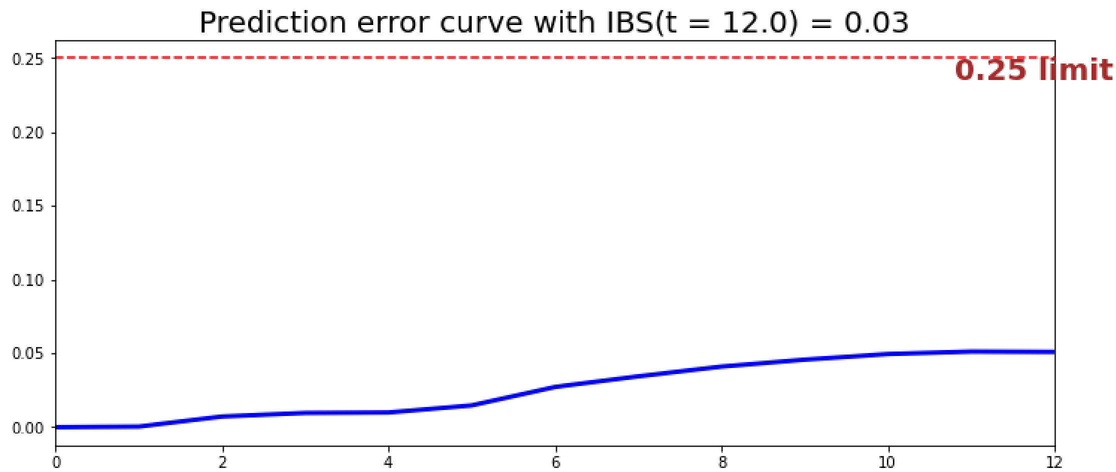
6.1.3 C-index

[79]:
```
from pysurvival.utils.metrics import concordance_index
c_index = concordance_index(csf, X_test, T_test, E_test)
print('C-index: {:.2f}'.format(c_index)) #0.83
```

```
C-index: 0.91
```

6.1.4 Brier Score

[80]:
```
from pysurvival.utils.display import integrated_brier_score
ibs = integrated_brier_score(csf, X_test, T_test, E_test, t_max=12,
    figure_size=(12,5))
print('IBS: {:.2f}'.format(ibs))
```




```
IBS: 0.03
```

The IBS is equal to 0.03 on the entire model time axis. This indicates that the model will have very good predictive abilities.

6.2 Building model without maritalStatus_outro, maritalStatus_nao definido, quartStadiumEntries_mais 105, year

[81]:
```
df.columns
```

[81]:
```
Index(['year', 'age', 'monthlyFee', 'totalAmount', 'totalMatches',
       'seasonMatches', 'monthsSinceLastPayment', 'dropout', 'yearsMembership',
       'stadiumAccess', 'yearLastPayment', 'sex_M',
       'maritalStatus_nao definido', 'maritalStatus_outro',
       'maritalStatus_solteiro', 'quartStadiumEntries_21 a 56',
       'quartStadiumEntries_56 a 105', 'quartStadiumEntries_ate 1',
       'quartStadiumEntries_mais 105'],
      dtype='object')
```

[82]:
```
to_remove = ['maritalStatus_outro', 'maritalStatus_nao definido',␣
 ↪'quartStadiumEntries_mais 105', 'year']
features = np.setdiff1d(features, to_remove).tolist()
```

[83]:

```
# Creating the X, T and E inputs
X_train, X_test = df[features], data_test[features]
T_train, T_test = df[time_column], data_test[time_column]
E_train, E_test = df[event_column], data_test[event_column]
```

[84]:

```
csf = RandomSurvivalForestModel(num_trees=200)
csf.fit(X_train, T_train, E_train, max_features='sqrt',
        max_depth=5, min_node_size=20)
```

[84]:

```
RandomSurvivalForestModel
```

6.2.1 Features importance

[85]:

```
csf.variable_importance_table
```

[85]:

	feature	importance	pct_importance
0	monthsSinceLastPayment	24.876900	0.288813
1	yearLastPayment	11.783575	0.136804
2	totalAmount	10.772272	0.125063
3	totalMatches	6.971886	0.080941
4	seasonMatches	6.933513	0.080496
5	maritalStatus_solteiro	5.183270	0.060176
6	stadiumAccess	3.992370	0.046350
7	monthlyFee	3.736315	0.043377
8	quartStadiumEntries_ate 1	3.495499	0.040582
9	quartStadiumEntries_56 a 105	3.233783	0.037543
10	quartStadiumEntries_21 a 56	2.401079	0.027876
11	sex_M	1.912730	0.022206
12	age	0.841816	0.009773

6.2.2 Model performance

We are going to access the model performance using the training and test set. Previously defined. C-index close to 1, the model has an powerfull discriminatory; but if it is close to 0.5, it has no ability to discriminate between low and high risk subjects.

6.2.3 C-index

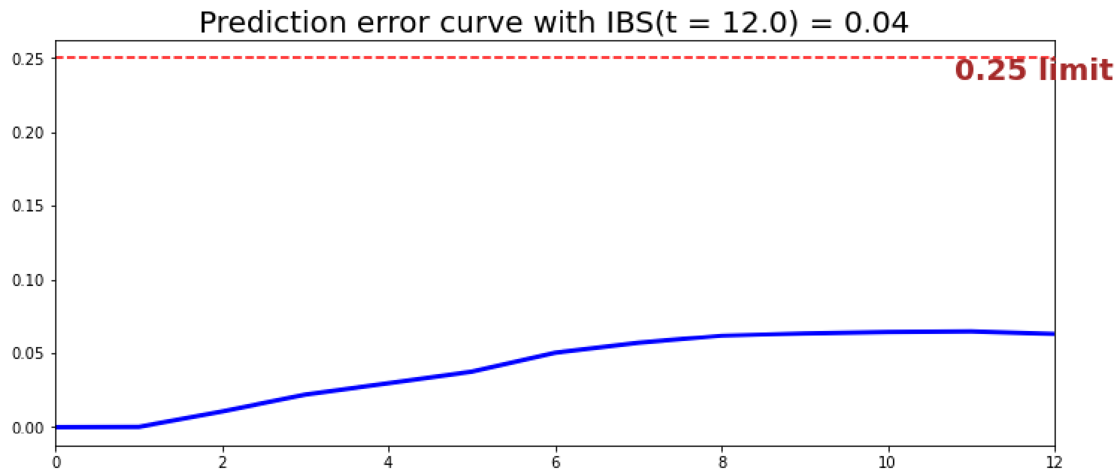
[86]:

```
from pysurvival.utils.metrics import concordance_index
c_index = concordance_index(csf, X_test, T_test, E_test)
print('C-index: {:.2f}'.format(c_index)) #0.83
```

```
C-index: 0.94
```

6.2.4 Brier Score

[87]:
```
from pysurvival.utils.display import integrated_brier_score
ibs = integrated_brier_score(csf, X_test, T_test, E_test, t_max=12,
    figure_size=(12,5))
print('IBS: {:.2f}'.format(ibs))
```

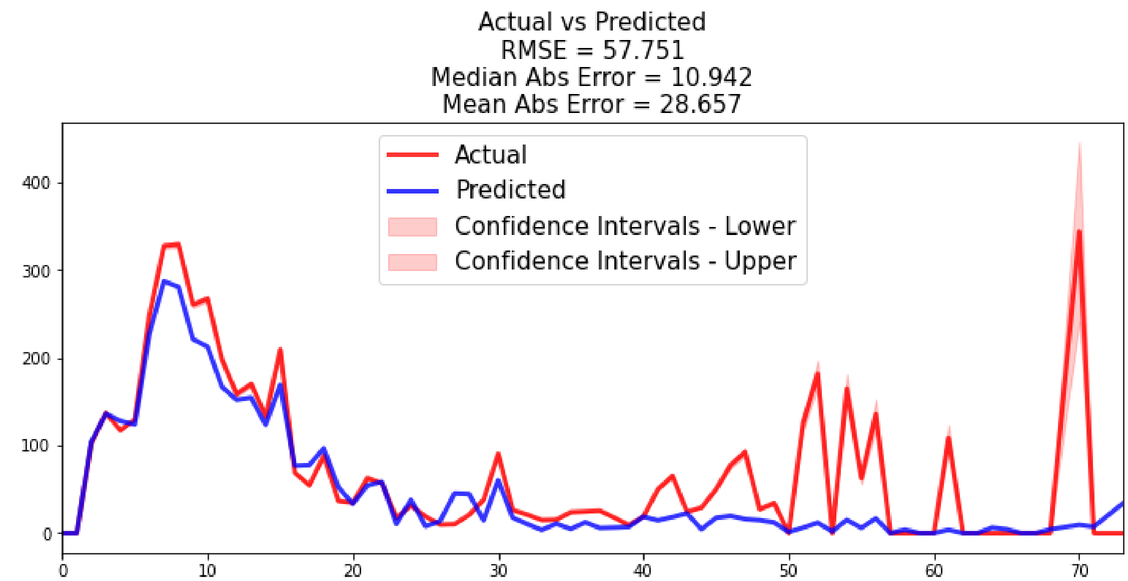



```
IBS: 0.04
```

7 Predictions

Lets compare the timeseries of actual and predicted customers who leave for each time t.

[89]:
```
from pysurvival.utils.display import compare_to_actual
results = compare_to_actual(csf, X_test, T_test, E_test, is_at_risk = False,
figure_size=(12, 5), metrics = ['rmse', 'mean', 'median'])
```

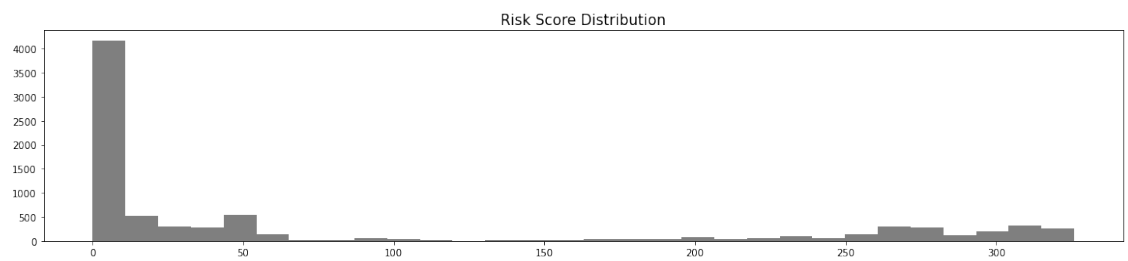


The model only makes an average absolute error of ~33 customers.

7.1 Individual predictions

Compute the probability of remaining a customer for all times t

[90]:
```
from pysurvival.utils.display import create_risk_groups

risk_groups = create_risk_groups(model=csf, X=X_test,
    use_log = False, num_bins=30, figure_size=(20, 4))
```



8 Survival Curves

[91]:
```
def curvaSobrevivencia(dados,coluna):
    ax = plt.subplot(111)
    plt.rcParams['figure.figsize'] = [12, 5]
    for item in dados[coluna].unique():
```

```
        ix = dados[coluna] == item
        kmf.fit(T.loc[ix], C.loc[ix], label=str(item))
        ax = kmf.plot(ax=ax)
```

8.1 Kaplan-Meier main curve

[92]:
```
from lifelines import KaplanMeierFitter
from lifelines.statistics import multivariate_logrank_test
from lifelines.statistics import pairwise_logrank_test

kmf = KaplanMeierFitter()
T = dfCurvas['yearsMembership']
C = dfCurvas['dropout']
kmf.fit(T,C,label="Membership dropout");
```

[93]:
```
tabela=pd.concat([kmf.event_table.reset_index(),
           kmf.conditional_time_to_event_.reset_index(),
           kmf.survival_function_.reset_index()],axis=1)
```

[94]:
```
tabela.columns = ['event_at', 'removed', 'observed', 'censored', 'entrance',␣
 ↪'at_risk','timeline',
                  'median duration remaining to event','timeline', 'Membership␣
 ↪dropout']
```

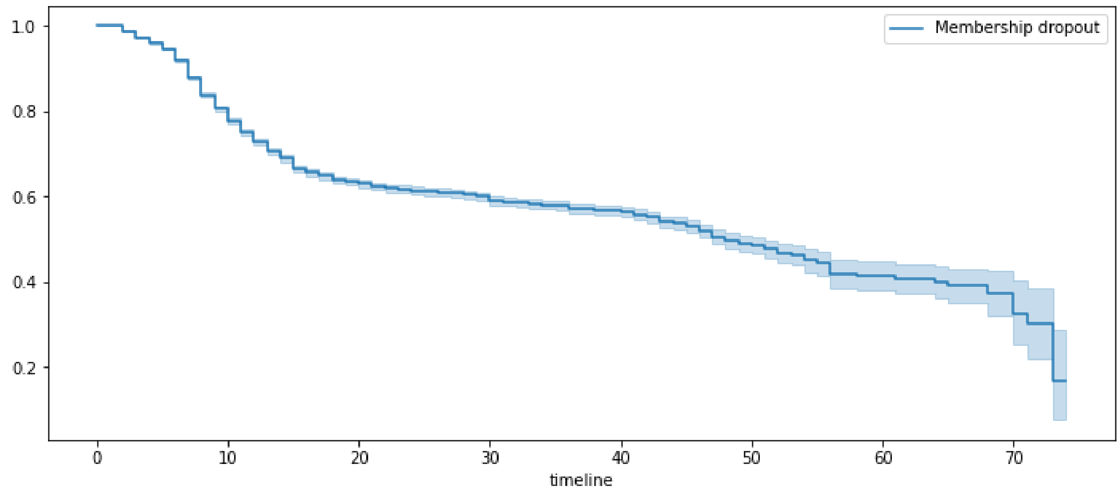
[95]:
```
tabela.head(12)
```

[95]:

```
    event_at  removed  observed  censored  entrance  at_risk  timeline  \
0          0     1595         0      1595     25316    25316       0.0
1          1     1809         0      1809         0    23721       1.0
2          2     1132       261       871         0    21912       2.0
3          3     1019       318       701         0    20780       3.0
4          4      630       260       370         0    19761       4.0
5          5      827       264       563         0    19131       5.0
6          6     2111       534      1577         0    18304       6.0
7          7     1988       719      1269         0    16193       7.0
8          8     1942       652      1290         0    14205       8.0
9          9     1241       459       782         0    12263       9.0
10        10     1946       397      1549         0    11022      10.0
11        11      978       310       668         0     9076      11.0

    median duration remaining to event  timeline  Membership dropout
0                                 48.0       0.0            1.000000
1                                 47.0       1.0            1.000000
2                                 47.0       2.0            0.988089
3                                 48.0       3.0            0.972968
4                                 47.0       4.0            0.960166
```

5	47.0	5.0	0.946916
6	48.0	6.0	0.919291
7	49.0	7.0	0.878473
8	48.0	8.0	0.838151
9	55.0	9.0	0.806780
10	58.0	10.0	0.777720
11	57.0	11.0	0.751157

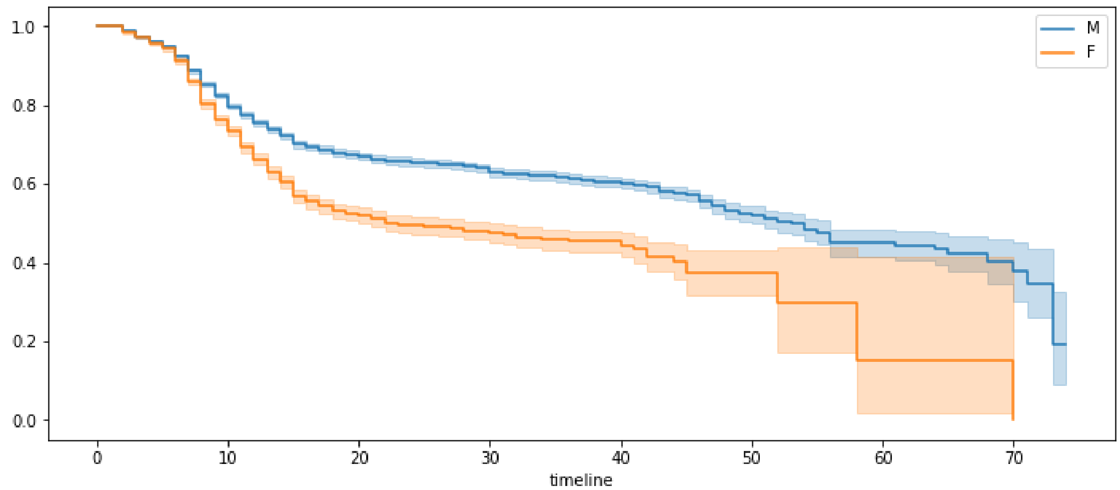
[96]:
```
plt.rcParams['figure.figsize'] = [12, 5]

kmf.plot();
```



8.2 By gender

[98]:
```
print(dfCurvas.sex.value_counts())
curvaSobrevivencia(dfCurvas,'sex')
```

```
M    17246
F     8070
Name: sex, dtype: int64
```



[99]:
```
results=multivariate_logrank_test(event_durations=T,groups=dfCurvas.
 ↪sex,event_observed=C)
results.print_summary()
```

	test_statistic	p	-log2(p)
0	194.625277	3.110248e-44	144.527807

[100]:
```
results=pairwise_logrank_test(event_durations=T,groups=dfCurvas.
 ↪sex,event_observed=C)
results.print_summary()
```

		test_statistic	p	-log2(p)
F	M	194.625277	3.110248e-44	144.527807

8.3 MesesUP

[102]:
```
dfCurvas.monthsSinceLastPayment.describe()
```

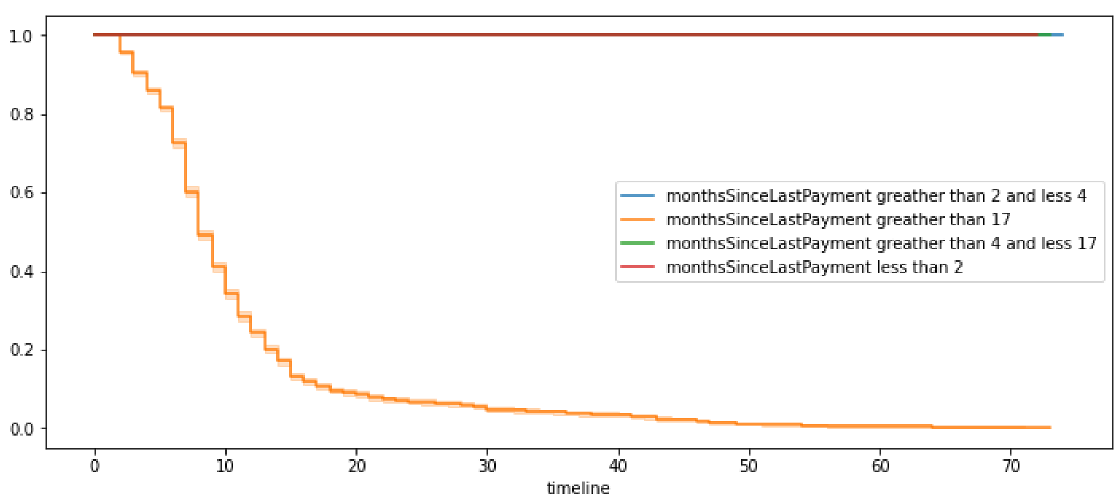
[102]:
```
count    25316.000000
mean        18.814110
std         32.498248
min          0.000000
25%          2.000000
50%          4.000000
75%         17.000000
max        156.000000
Name: monthsSinceLastPayment, dtype: float64
```

```
[103]: var='monthsSinceLastPayment'
varEscalao='escMesesUP'
dfCurvas[varEscalao]=''
for index, cliente in dfCurvas.iterrows():
    #se a variável tiver o valor 1 colocar na nova variável a descrição da␣
 ↪atividade
    if cliente[var] <= 2:
        dfCurvas.at[index,varEscalao]=var+' less than 2'
    elif (cliente[var] > 2) & (cliente[var] <= 4):
        dfCurvas.at[index,varEscalao]=var+' greather than 2 and less 4'
    elif (cliente[var] > 4) & (cliente[var] <= 17):
        dfCurvas.at[index,varEscalao]=var + ' greather than 4 and less 17'
    elif (cliente[var] > 17):
        dfCurvas.at[index,varEscalao]=var + ' greather than 17'
```

```
[104]: dfCurvas.monthsSinceLastPayment.value_counts()
```

```
[104]: 0      4767
2      4047
3      3557
4      1396
13      834
       …
150       6
154       5
149       4
151       4
155       1
Name: monthsSinceLastPayment, Length: 156, dtype: int64
```

```
[105]: curvaSobrevivencia(dfCurvas,varEscalao)
```



[106]:
```
results=multivariate_logrank_test(event_durations=T,groups=dfCurvas[varEscalao],event_observed
results.print_summary()
```

	test_statistic	p	-log2(p)
0	19998.897283	0.0	inf

[107]:
```
results=pairwise_logrank_test(event_durations=T,groups=dfCurvas[varEscalao],event_observed=C)
results.print_summary()
```

monthsSinceLastPayment greather than 17	monthsSinceLastPayment greather than 2 and less 4
	monthsSinceLastPayment greather than 4 and less 17
	monthsSinceLastPayment less than 2
monthsSinceLastPayment greather than 2 and less 4	monthsSinceLastPayment greather than 4 and less 17
	monthsSinceLastPayment less than 2
monthsSinceLastPayment greather than 4 and less 17	monthsSinceLastPayment less than 2

8.4 ValorTotal

[108]:
```
dfCurvas.totalAmount.describe()
```

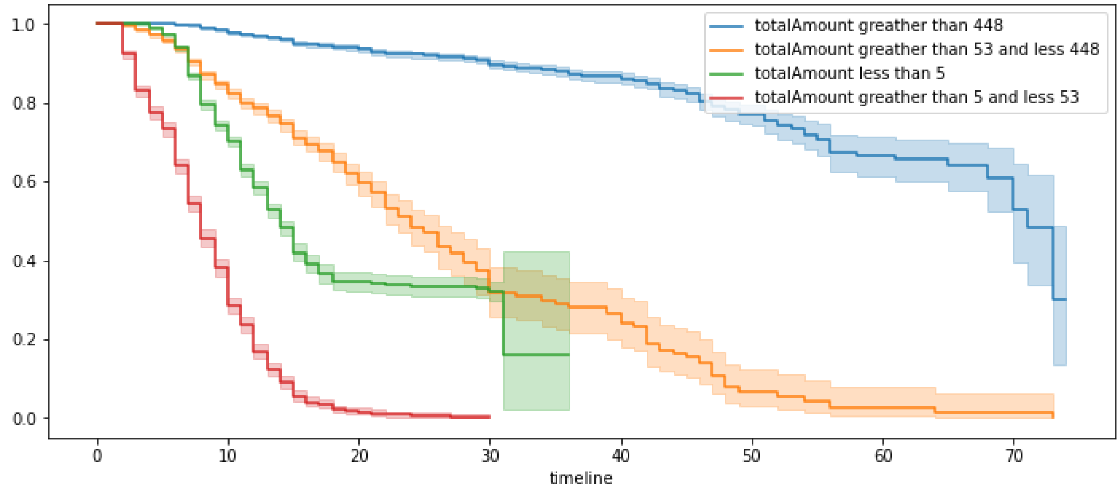
[108]:
```
count    25316.000000
mean       316.037984
std        493.971528
min          0.000000
25%          5.000000
50%         53.000000
75%        448.250000
max       2602.000000
Name: totalAmount, dtype: float64
```

[109]:
```
var='totalAmount'
varEscalao='escValorTotal'
dfCurvas[varEscalao]=''
for index, cliente in dfCurvas.iterrows():
    #se a variável tiver o valor 1 colocar na nova variável a descrição da␣
 ↪atividade
    if cliente[var] <= 5:
        dfCurvas.at[index,varEscalao]=var+' less than 5'
    elif (cliente[var] > 5) & (cliente[var] <= 53):
        dfCurvas.at[index,varEscalao]=var+' greather than 5 and less 53'
    elif (cliente[var] > 53) & (cliente[var] <= 448):
        dfCurvas.at[index,varEscalao]=var + ' greather than 53 and less 448'
    elif (cliente[var] > 448):
        dfCurvas.at[index,varEscalao]=var + ' greather than 448'
```

[110]:
```
dfCurvas[varEscalao].value_counts()
```

[110]:
```
totalAmount less than 5                          7060
totalAmount greather than 448                    6329
totalAmount greather than 53 and less 448        6280
totalAmount greather than 5 and less 53          5647
Name: escValorTotal, dtype: int64
```

[111]:
```
curvaSobrevivencia(dfCurvas,varEscalao)
```



[112]:
```
results=multivariate_logrank_test(event_durations=T,groups=dfCurvas[varEscalao],event_observed
results.print_summary()
```

	test_statistic	p	-log2(p)
0	9517.829603	0.0	inf

[113]:
```
results=pairwise_logrank_test(event_durations=T,groups=dfCurvas[varEscalao],event_observed=C)
results.print_summary()
```

		test_statistic	
totalAmount greather than 448	totalAmount greather than 5 and less 53	8318.461705	0.000
	totalAmount greather than 53 and less 448	1527.147254	0.000
	totalAmount less than 5	3177.425216	0.000
totalAmount greather than 5 and less 53	totalAmount greather than 53 and less 448	2997.582565	0.000
	totalAmount less than 5	2005.351350	0.000
totalAmount greather than 53 and less 448	totalAmount less than 5	274.052087	1.48

8.5 MonthlyFee

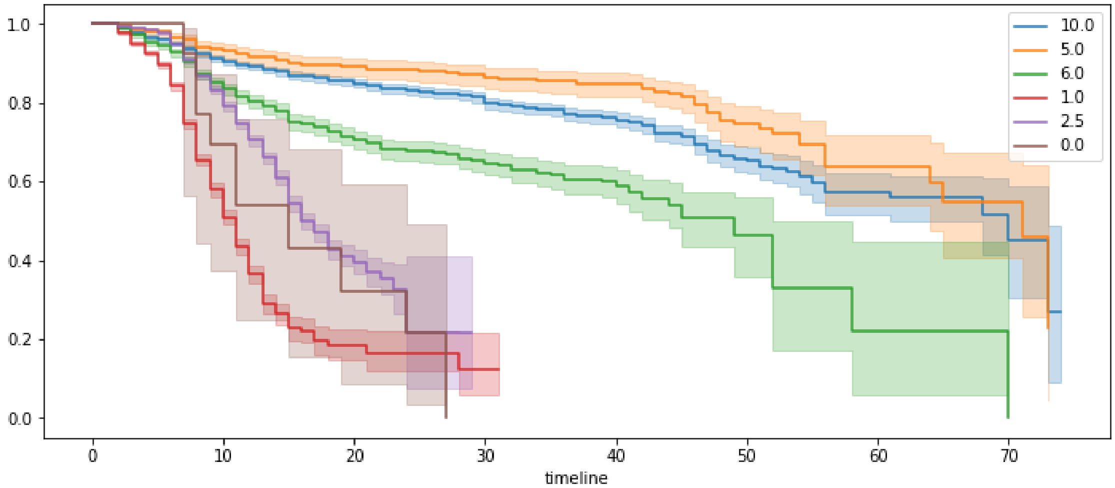
[115]:
```
varEscalao='monthlyFee'
dfCurvas[varEscalao].describe()
```

[115]:
```
count    25316.000000
mean         4.356099
std          3.550837
min          0.000000
25%          1.000000
50%          2.500000
75%          6.000000
max         10.000000
Name: monthlyFee, dtype: float64
```

[116]:
```
dfCurvas[varEscalao].value_counts()
```

[116]:
```
1.0     8016
2.5     7168
10.0    6126
6.0     3123
5.0      869
0.0       14
Name: monthlyFee, dtype: int64
```

[117]:
```
curvaSobrevivencia(dfCurvas,varEscalao)
```



[118]:
```
results=multivariate_logrank_test(event_durations=T,groups=dfCurvas[varEscalao],event_observed
results.print_summary()
```

	test_statistic	p	-log2(p)
0	3373.682348	0.0	inf

[119]:
```
results=pairwise_logrank_test(event_durations=T,groups=dfCurvas[varEscalao],event_observed=C)
results.print_summary()
```

		test_statistic	p	-log2(p)
0.0	1.0	0.837357	3.601539e-01	1.473315
	2.5	0.727220	3.937856e-01	1.344518
	5.0	76.734174	1.955894e-18	58.826878
	6.0	19.484981	1.013938e-05	16.589671
	10.0	53.939692	2.067388e-13	42.137256
1.0	2.5	1031.075028	3.160656e-226	749.095525
	5.0	653.856846	3.238374e-144	476.662376
	6.0	846.919933	3.397667e-186	616.114081
	10.0	2167.783976	0.000000e+00	inf
2.5	5.0	340.212952	5.734471e-76	249.946875
	6.0	178.329728	1.122322e-40	132.710637
	10.0	957.333303	3.378086e-210	695.848694
5.0	6.0	109.345324	1.363389e-25	82.601005
	10.0	16.291769	5.429930e-05	14.168707
6.0	10.0	159.387160	1.540097e-36	118.966390

8.6 Season Matches

[120]:
```
var='seasonMatches'
varEscalao='escJogosEpoca'
dfCurvas[var].describe()
```

[120]:
```
count    25316.000000
mean         2.171631
std          4.076356
min          0.000000
25%          0.000000
50%          0.000000
75%          2.000000
max         16.000000
Name: seasonMatches, dtype: float64
```

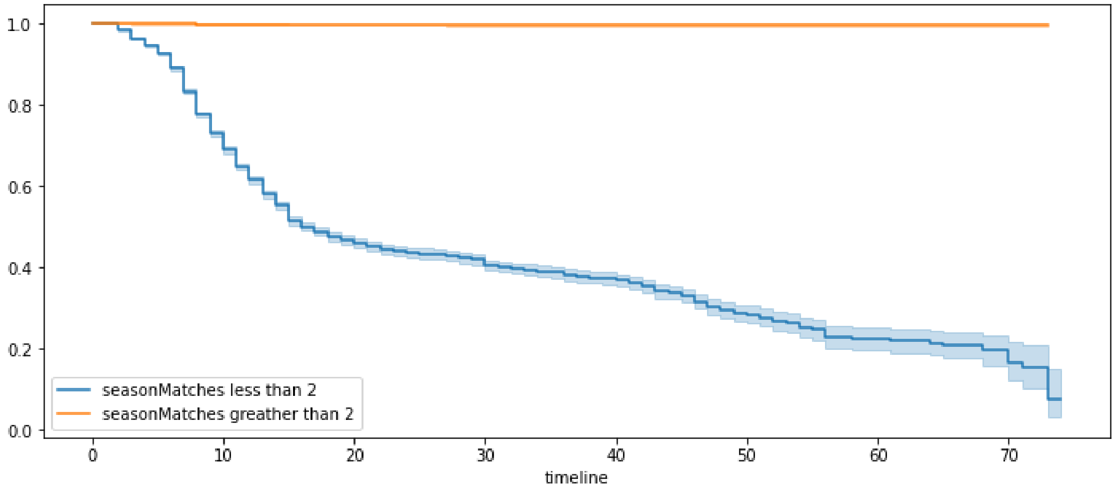
[121]:
```
dfCurvas[varEscalao]=''
for index, cliente in dfCurvas.iterrows():
    #se a variável tiver o valor 1 colocar na nova variável a descrição da␣
↪atividade
    if cliente[var] <= 2:
        dfCurvas.at[index,varEscalao]=var+' less than 2'
    elif (cliente[var] > 2):
```

```
        dfCurvas.at[index,varEscalao]=var + ' greather than 2'
```

[122]:
```
dfCurvas[varEscalao].value_counts()
```

[122]:
```
seasonMatches less than 2        19015
seasonMatches greather than 2     6301
Name: escJogosEpoca, dtype: int64
```

[123]:
```
curvaSobrevivencia(dfCurvas,varEscalao)
```



[124]:
```
results=multivariate_logrank_test(event_durations=T,groups=dfCurvas[varEscalao],event_observed
results.print_summary()
```

	test_statistic	p	-log2(p)
0	3270.332736	0.0	inf

[125]:
```
results=pairwise_logrank_test(event_durations=T,groups=dfCurvas[varEscalao],event_observed=C)
results.print_summary()
```

		test_statistic	p	-log2(p)
seasonMatches greather than 2	seasonMatches less than 2	3270.332736	0.0	inf

8.7 escalaoTotalJogos

[128]:
```
var='escJogosEpoca'
dfCurvas[var].describe()
```

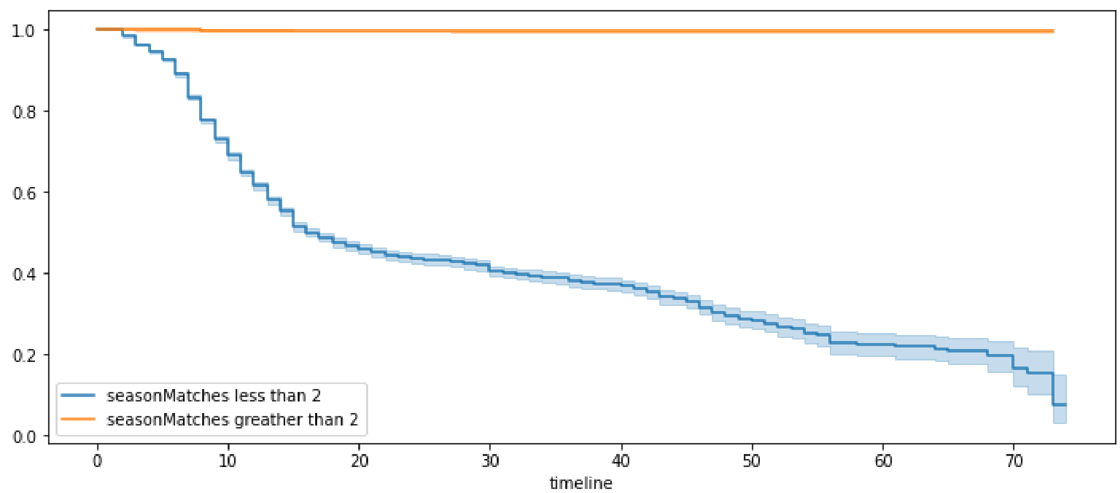
[128]:
```
count                     25316
unique                        2
```

```
top       seasonMatches less than 2
freq                          19015
Name: escJogosEpoca, dtype: object
```

[129]:
```
dfCurvas[var].value_counts()
```

[129]:
```
seasonMatches less than 2        19015
seasonMatches greather than 2     6301
Name: escJogosEpoca, dtype: int64
```

[130]:
```
curvaSobrevivencia(dfCurvas,var)
```



[82]:
```
results=multivariate_logrank_test(event_durations=T,groups=dfCurvas[var],event_observed=C)
results.print_summary()
```

	test_statistic	p	-log2(p)
0	3147.499074	0.0	inf

[83]:
```
results=pairwise_logrank_test(event_durations=T,groups=dfCurvas[var],event_observed=C)
results.print_summary()
```

		test_statistic	p	-log2(p)
1 a 21	21 a 56	159.947442	1.161801e-36	119.373049
	56 a 105	816.389467	1.474826e-179	594.064585
	ate 1	11.448851	7.153824e-04	10.448998
	mais 105	1574.871681	0.000000e+00	inf
21 a 56	56 a 105	279.881955	7.967468e-63	206.287349
	ate 1	361.252035	1.502998e-80	265.166405
	mais 105	851.984446	2.692429e-187	619.771645
56 a 105	ate 1	1204.146733	7.657725e-264	874.052101
	mais 105	217.912019	2.581281e-49	161.406390
ate 1	mais 105	1894.318290	0.000000e+00	inf

8.8 Marital Status

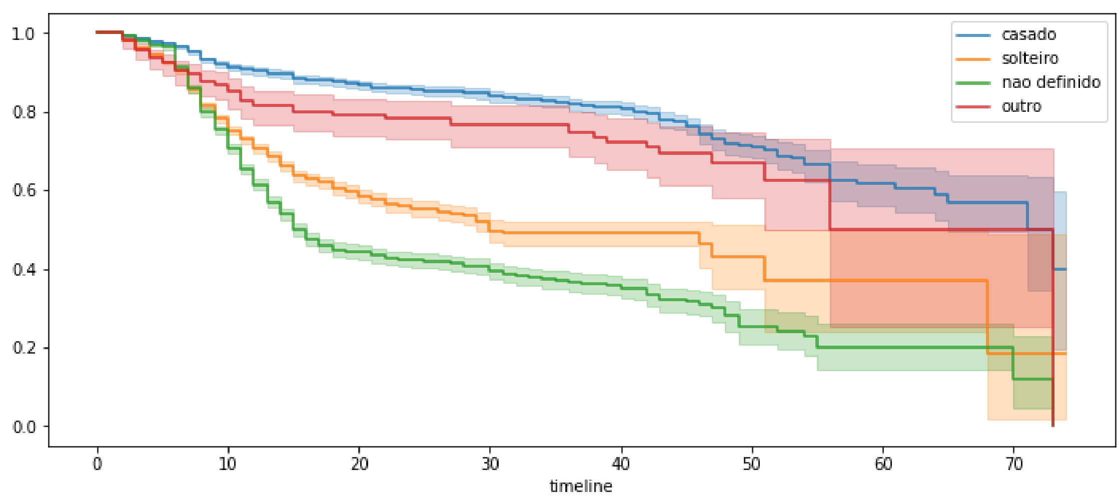
[131]:
```
var='maritalStatus'
dfCurvas[var].describe()
```

[131]:
```
count        25316
unique           4
top      solteiro
freq         12065
Name: maritalStatus, dtype: object
```

[132]:
```
dfCurvas[var].value_counts()
```

[132]:
```
solteiro        12065
nao definido     7667
casado           5085
outro             499
Name: maritalStatus, dtype: int64
```

[133]:
```
curvaSobrevivencia(dfCurvas,var)
```



[134]:
```
results=multivariate_logrank_test(event_durations=T,groups=dfCurvas[var],event_observed=C)
results.print_summary
```

[134]:
```
<bound method StatisticalResult.print_summary of <lifelines.StatisticalResult:
multivariate_logrank_test>
               t_0 = -1
 null_distribution = chi squared
degrees_of_freedom = 3
         test_name = multivariate_logrank_test

---
 test_statistic      p  -log2(p)
        1350.15 <0.005    969.05>
```

[135]:
```
results=pairwise_logrank_test(event_durations=T,groups=dfCurvas[var],event_observed=C)
results.print_summary()
```

		test_statistic	p	-log2(p)
casado	nao definido	1387.589094	1.045572e-303	1006.479921
	outro	16.817280	4.115683e-05	14.568509
	solteiro	763.130292	5.603198e-168	555.597669
nao definido	outro	84.704006	3.465475e-20	64.645509
	solteiro	86.339240	1.515709e-20	65.838569
outro	solteiro	35.301134	2.824676e-09	28.399268