TCC_MARCIO_Fontes

March 12, 2020

0.0.1 CUSTOMER CHURN EM EMPRESAS DE SEGUROS DE AUTOMÓVEL - 03/2020

0.0.2 MARCIO DE LIMA

Trabalho de Conclusão de Curso para a obtenção do ttulo de Especialista em Big Data, Data Science e Data Analytics, pelo curso de Pós-Graduação Lato Sensu em Big data, Data science e Data Analytics da Universidade do Vale do Rio dos Sinos – UNISINOS. Orientadora: Prof. Dra. Josiane B. Porto

0.1 Dicionário de Dados

0.1.1 O DataSet contém 48 colunas / variáveis.

0.1.2 Cada variável possui uma descrição relacionada ao negócio e possuem valores / códigos

0.1.3 relacionados ao seu significado.

0.1.4 Variável => Tipo => Valores / Domínios => Descrição

anosFidelidadeCliente => Variável numérica => Representa quantos anos o cliente está na CIA.

codigoEstadoCivil => Variável Categória numérica, representada por código / domínios (1 - Solteiro, 2 - Casado, 3 - Viúvo, 4 - Separado judicialmente, 5 - Divorciado). Representa o estado civil do Segurado.

sexoSegurado => Variável Categória numérica, representada por código / domínios (0 - masculino, 1 - Feminino). Representa o sexo do Segurado.

tipoPessoaSegurado => Variável Categória numérica, representada por código / domínios (0 - Juridica, 1 - Fisica). Representa se o segurado é pessoa física ou pessoa jurídica (empresa).

resgatePontos => Variável Categória numérica, representada por código / domínios (0 - Não resgatou, 1 - resgatou). Flag que informa se o segurado utilizou seus pontos do cartão de crédito para a renovação do seguro.

numeroSegmentoCliente => Variável Categória numérica, representada por código / domínios (1 - Tradicional, 2 - Auto-Premium). Representa a classificação do cliente na CIA.

codigoCategoriaTarifaria => Variável Categória numérica, representada por código / domínios (10 - Passeio, 11 - Importado, 30 - Moto, 80 - taxi, etc.) São dezenas de códigos que representam uma categoria do veículo.

quantidadePortasVeiculo => Variável numérica => Representa a quantidade de portas do veiculo (2, 3 4 ou 5 portas).

codigoCombustivelVeiculo => Variável Categória numérica, representada por código / domínios (8 - Gasolina, 9 - Alcool, 10 - Flex, 11 - Diesel, 12 - Gas). Representa o tipo do combustivel do veículo.

codigoUsoVeiculo => Variável Categória numérica, representada por código / domínios (1 - Particular, 2 - Comercial). Representa o uso que o segurado faz do veículo.

codigoFamiliaVeiculo => Variável Categória numérica, representada por código / domínios (1 - Veiculos Leves, 2 - Veiculos pesados, 3 - Vans e onibus, etc.). Representa um agrupamento de categorias de veiculo.

especieVeiculo => Variável Categória numérica, representada por código / domínios (1 - Passageiro, 2 - Carga, 3 - Especial, etc.). Representa a espécie do veiculo.

codigoTipoVeiculo => Variável Categória numérica, representada por código / domínios (1 - SUV, 2 - Hatch, 3 - Sedan, etc.). Representa o tipo do veiculo.

codigoMarcaVeiculo => Variável Categória numérica, representada por código / domínios (1 - VW, 2 - Fiat, 3 - Chevrolet, 4 - Nissan, 5 - Pegueot, etc.). Representa o fabricante do veiculo.

numeroSegmentoDocumento => Variável Categória numérica, representada por código / domínios (1 - Premium, 2 - Auto-Jovem, 3 - Moto, 4 - Idoso, 5 - Caminhão, 6 - Mulher, 7 - Juridica). Representa a classificação da apólice na CIA.

codigoSucursal => Variável Categória numérica, representada por código / domínios (1 - Sucursal Centro SP, 2 - Sucursal Rio de Janeiro, 3 - Sucursal São Miguel Pta SP, 4 - Sucursal Campinas SP, etc.). Representa um ID para cada sucursal (unidade de atendimento da CIA para segurados e corretores) nas cidades brasileiras.

premioLiquidoPagoApolice => Variável numérica => Representa o valor em reais do valor pago para a contratação da apólice anterior. Apólice antiga.

valorFranquia => Variável numérica => Representa o valor em reais da franquia do veículo.

valorPremioFinal => Variável numérica => Representa o valor em reais da apólice a ser contratada.

valorDiferencaPremioAnual => Variável numérica => Representa o valor em reais do prêmio anual dividido pelo valor do prêmio anterior pago pelo segurado.

quantidadeUtilizacaoEstapar => Variável numérica => Representa a quantidade de uso do serviço da Estapar efetuados pelo segurado no período.

codigoClasseLocalizacao => Variável Categória numérica, representada por código / domínios (10 - Zona Leste de São Paulo, 11 - Centro de São Paulo, 2 - Rio de Janeiro, etc.). São dezenas de códigos que representam macro-região dos estados brasileiros.

origemProposta => Variável Categória numérica, representada por código / domínios (20 - Online, 12 - Renovação, 1 - Legado). Representa o sistema de origem da proposta de renovação.

codigoAgravamentoPremio => Variável Categória numérica, representada por código / domínios (0 - Sem Agravo, 22 - Agravo de 5%, 2 - Agravo de 10%, etc.). Representa se houve um aumento do preço devido há alguma questão.

quantidadeParcelas => Variável numérica => Representa a quantidade de parcelas que a apólice anterior foi paga.

quantidadeSinistroAuto => Variável numérica => Representa a quantidade de sinistros de automóvel (roubo/furto ou colisão) que o veículo teve no período de 5 anos.

quantidadeTotalSinistros => Variável numérica => Representa a quantidade total de sinistros de qualquer espécie que o veículo teve no período de 5 anos.

codigoClasseBonus => Variável numérica => Representa a classe de bônus da apólice. A cada renovação, a classe aumenta.

codigoFormaPagamento => Variável Categória numérica, representada por código / domínios (52 - Boleto, 97 - Cartão, etc.). Representa a forma de pagamento da apólice anterior.

Mes => Variável numérica => Representa o mês da renovação.

diaSemana => Variável numérica => Representa o dia da Semana da renovação.

estacaoAno => Variável Categória numérica, representada por código / domínios (1 - Verão, 2 - Outono, 3 - Inverno, 4 - Primavera). Representa a estação do ano da renovação.

codigoIBGEUF => Variável Categória numérica, representada por código / domínios (11-RO, 12-AC, 13-AM, 14-RR, 15-PA, 16-AP, 17-TO, 21-MA, 22-PI, 23-CE, 24-RN, 25-PB, 26-PE, 27-AL, 28-SE, 29-BA, 31-MG, 32-ES, 33-RJ, 35-SP, 41-PR, 42-SC, 43-RS, 50-MS, 51-MT, 52-GO, 53-DF). Representa a UF de acordo com o IBGE.

IDHM => Variável numérica => Índice de Desenvolvimento Humano Municipal. Média geométrica dos índices das dimensões Renda, Educação e Longevidade, com pesos iguais. Referente a UF da apólice.

IDHM_E => Variável numérica => Índice de Desenvolvimento Humano Municipal - Dimensão Educação. Referente a UF da apólice.

IDHM_L => Variável numérica => Índice de Desenvolvimento Humano Municipal - Dimensão Longevidade da UF. Referente a UF da apólice.

IDHM_R => Variável numérica => Índice de Desenvolvimento Humano Municipal - Dimensão Renda da UF. Referente a UF da apólice.

I_ESCOLARIDADE => Variável numérica => Subíndice de escolaridade fundamental da população adulta - IDHM Educação da UF. Referente a UF da apólice.

valorPremioPagoAtual => Variável numérica => Representa o valor em reais do prêmio atual da apólice que vai renovar.

correntista => Variável Categória numérica, representada por código / domínios (0-Nao, 1-Sim). Informa se o segurado é cliente Itaú.

faixaIdade => Variável Categória numérica, representada por código / domínios (1- De 18 a 24 anos, 2- De 25 a 60, 3- Acima de 60). Faixa de idade do segurado.

faixaPrecoLiquido => Variável Categória numérica, representada por código / domínios (1- De 0 ao 1 quartil(25%), 2- Do 1 quartil(25%) a média, 3- Acima da média. Faixa de preços líquidos.

faixaPrecoFranquia => Variável Categória numérica, representada por código / domínios (1- De 0 ao 1 quartil(25%), 2- Do 1 quartil(25%) a média, 3- Acima da média. Faixa de preços de franquia.

faixaPrecoPremioFinal => Variável Categória numérica, representada por código / domínios (1- De 0 ao 1 quartil(25%), 2- Do 1 quartil(25%) a média, 3- Acima da média. Faixa de preços de prêmios finais.

diferencaPremioAntNovo => Variável numérica => Representa o valor em reais do prêmio atual menos o prêmio anterior pago pelo segurado. Pode ser positivo ou negativo.

faixaDiferencaValores => Variável Categória numérica, representada por código / domínios (1- De 0 ao 1 quartil(25%), 2- Do 1 quartil(25%) a média, 3- Acima da média. Faixa de preços da diferença entre os prêmios.

renovou => Variável Categória numérica, representada por código / domínios (0-Apolice não-renovada (CHURN), 1-Apolice renovada). Nossa variável TARGET.

gerouSinistro => Variável Categória numérica, representada por código / domínios (0-Nao, 1-Sim). Flag de controle, caso exista qualquer sinistro na apólice durante o período de vigncia.

0.2 Importação

```
In [1]: import string
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

```
import warnings

%matplotlib inline
warnings.filterwarnings("ignore")

from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import auc
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.multiclass import OneVsRestClassifier
from sklearn.ensemble import ExtraTreesClassifier

from sklearn.model_selection import RandomizedSearchCV, GridSearchCV

#!pip install lightgbm
from lightgbm import LGBMClassifier

#!pip install xgboost
from xgboost import XGBClassifier

#!pip install joblib
from sklearn.externals import joblib
```

```
/home/marcio/anaconda3/lib/python3.7/site-packages/sklearn/externals/joblib/__init__.py:15: Dep
  warnings.warn(msg, category=DeprecationWarning)
```

In [2]:
```
#Carregando o dataset
df = pd.read_csv('dados/dataset_original.csv', sep=",")
df.head(10)
```

Out[2]:

	anosFidelidadeCliente	codigoEstadoCivil	sexoSegurado	tipoPessoaSegurado	\
0	11	2	1	1	
1	11	2	1	1	

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

   resgatePontos  numeroSegmentoCliente  codigoCategoriaTarifaria  \
0              1                      1                        10
1              0                      1                        10
2              0                      1                        10
3              0                      1                        10
4              0                      1                        11
5              0                      1                        30
6              0                      1                        14
7              0                      1                        30
8              0                      1                        10
9              0                      1                        22

   quantidadePortasVeiculo  codigoCombustivelVeiculo  codigoUsoVeiculo  ...  \
0                        5                         8                 1  ...
1                        5                         8                 1  ...
2                        5                         8                 1  ...
3                        5                         8                 1  ...
4                        2                         2                 1  ...
5                        0                         8                 1  ...
6                        3                         8                 1  ...
7                        0                         8                 1  ...
8                        4                         8                 1  ...
9                        5                         8                 1  ...

   valorPremioPagoAtual  correntista  faixaIdade  faixaPrecoLiquido  \
0           2152.436533            1           2                  3
1           2152.436533            1           3                  2
2           2152.436533            1           3                  2
3           2131.880000            1           3                  3
4           1495.110000            0           3                  2
5           1328.220000            0           2                  2
6           2152.436533            0           3                  2
7           2152.436533            0           2                  1
8           2152.436533            0           2                  2
9           8150.710000            0           2                  3

   faixaPrecoFranquia  faixaPrecoPremioFinal  diferencaPremioAntNovo  \
0                   2                      1             -288.523467
1                   3                      3              386.676533
```

```
2                    3                    3              386.676533
3                    1                    3               59.540000
4                    3                    2               45.400000
5                    2                    2              -42.150000
6                    2                    1              561.556533
7                    1                    2             1092.336533
8                    1                    2              574.456533
9                    3                    3             4546.420000

   faixaDiferencaValores  renovou  gerouSinistro
0                      1        0              0
1                      3        0              0
2                      3        0              0
3                      2        0              0
4                      2        0              0
5                      2        0              0
6                      3        0              0
7                      3        0              0
8                      3        0              0
9                      3        0              0

[10 rows x 48 columns]
```

1 Análise Exploratória dos Dados

```
In [3]: # Mostrando os dados
        df.shape
```

```
Out[3]: (74570, 48)
```

```
In [4]: # Mostrando as estruturas do Dataset
        df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74570 entries, 0 to 74569
Data columns (total 48 columns):
anosFidelidadeCliente          74570 non-null int64
codigoEstadoCivil              74570 non-null int64
sexoSegurado                   74570 non-null int64
tipoPessoaSegurado             74570 non-null int64
resgatePontos                  74570 non-null int64
numeroSegmentoCliente          74570 non-null int64
codigoCategoriaTarifaria       74570 non-null int64
quantidadePortasVeiculo        74570 non-null int64
codigoCombustivelVeiculo       74570 non-null int64
codigoUsoVeiculo               74570 non-null int64
codigoFamiliaVeiculo           74570 non-null int64
especieVeiculo                 74570 non-null int64
```

```
codigoTipoVeiculo              74570 non-null int64
codigoMarcaVeiculo             74570 non-null int64
numeroSegmentoDocumento        74570 non-null int64
codigoSucursal                 74570 non-null int64
premioLiquidoPagoApolice       74570 non-null float64
valorFranquia                  74570 non-null float64
valorPremioFinal               74570 non-null float64
valorDiferencaPremioAnual      74570 non-null float64
quantidadeUtilizacaoEstapar    74570 non-null int64
codigoClasseLocalizacao        74570 non-null int64
origemProposta                 74570 non-null int64
codigoAgravamentoPremio        74570 non-null int64
quantidadeParcelas             74570 non-null int64
quantidadeSinistroAuto         74570 non-null int64
quantidadeTotalSinistros       74570 non-null int64
codigoClasseBonus              74570 non-null int64
codigoFormaPagamento           74570 non-null int64
Mes                            74570 non-null int64
diaSemana                      74570 non-null int64
estacaoAno                     74570 non-null int64
codigoIBGEUF                   74570 non-null int64
IDHM                           74570 non-null float64
IDHM_E                         74570 non-null float64
IDHM_L                         74570 non-null float64
IDHM_R                         74570 non-null float64
I_ESCOLARIDADE                 74570 non-null float64
valorPremioPagoAtual           74570 non-null float64
correntista                    74570 non-null int64
faixaIdade                     74570 non-null int64
faixaPrecoLiquido              74570 non-null int64
faixaPrecoFranquia             74570 non-null int64
faixaPrecoPremioFinal          74570 non-null int64
diferencaPremioAntNovo         74570 non-null float64
faixaDiferencaValores          74570 non-null int64
renovou                        74570 non-null int64
gerouSinistro                  74570 non-null int64
dtypes: float64(11), int64(37)
memory usage: 27.3 MB
```

```
In [5]: # Dados Estatisticos - Analise descritiva das colunas Numéricas
        # Arquivo com 48 colunas, todas numéricas, total de Registros: 74.570 linhas
        # Variavel Target (renovou) possue 0 e 1, olhando a média já vemos que o dataset está
        # Possíveis outlier na coluna valorPremioPagoAtual e diferencaPremioAntNovo.
        # Desvio padrão quase o mesmo para todas as colunas.

        df.describe()
```

```
Out[5]:        anosFidelidadeCliente  codigoEstadoCivil  sexoSegurado  \
```

```
              74570.000000       74570.000000  74570.000000
mean              5.275003           1.940445      0.418064
std               5.062921           0.869902      0.493244
min               0.000000           0.000000      0.000000
25%               2.000000           2.000000      0.000000
50%               4.000000           2.000000      0.000000
75%               7.000000           2.000000      1.000000
max              24.000000           6.000000      1.000000

       tipoPessoaSegurado  resgatePontos  numeroSegmentoCliente  \
count        74570.000000   74570.000000                74570.0
mean             0.981816       0.269049                    1.0
std              0.133618       0.443469                    0.0
min              0.000000       0.000000                    1.0
25%              1.000000       0.000000                    1.0
50%              1.000000       0.000000                    1.0
75%              1.000000       1.000000                    1.0
max              1.000000       1.000000                    1.0

       codigoCategoriaTarifaria  quantidadePortasVeiculo  \
count              74570.000000             74570.000000
mean                  14.746560                 4.042349
std                    7.702437                 1.559515
min                   10.000000                 0.000000
25%                   10.000000                 4.000000
50%                   10.000000                 5.000000
75%                   22.000000                 5.000000
max                   98.000000                 5.000000

       codigoCombustivelVeiculo  codigoUsoVeiculo  ...  valorPremioPagoAtual  \
count              74570.000000      74570.000000  ...          74570.000000
mean                   6.624326          1.063538  ...           2260.690956
std                    2.482504          1.137616  ...           2627.574709
min                    1.000000          1.000000  ...            279.270000
25%                    8.000000          1.000000  ...           1467.592500
50%                    8.000000          1.000000  ...           2104.510000
75%                    8.000000          1.000000  ...           2315.072500
max                   11.000000         30.000000  ...         106166.360000

        correntista    faixaIdade  faixaPrecoLiquido  faixaPrecoFranquia  \
count  74570.000000  74570.000000       74570.000000        74570.000000
mean       0.448974      2.276478           2.117661            2.111184
std        0.497393      0.470236           0.766156            0.770829
min        0.000000      1.000000           1.000000            1.000000
25%        0.000000      2.000000           2.000000            2.000000
50%        0.000000      2.000000           2.000000            2.000000
75%        1.000000      3.000000           3.000000            3.000000
max        1.000000      3.000000           3.000000            3.000000
```

```
       faixaPrecoPremioFinal  diferencaPremioAntNovo  faixaDiferencaValores  \
count           74570.000000            74570.000000           74570.000000
mean                2.159970              163.627763               2.175298
std                 0.780438             2361.560243               0.819463
min                 1.000000           -32548.331070               1.000000
25%                 2.000000             -135.392500               1.000000
50%                 2.000000               49.076904               2.000000
75%                 3.000000              424.855428               3.000000
max                 3.000000            97820.780000               3.000000

             renovou  gerouSinistro
count  74570.000000   74570.000000
mean       0.499933       0.036985
std        0.500003       0.188727
min        0.000000       0.000000
25%        0.000000       0.000000
50%        0.000000       0.000000
75%        1.000000       0.000000
max        1.000000       1.000000

[8 rows x 48 columns]
```

In [6]:
```
#Checando valores NA nos dados
df.isna().any()[lambda x: x]
```

Out[6]:
```
Series([], dtype: bool)
```

In [7]:
```
#Checando valores Null nos dados
df.isnull().any()[lambda x: x]
```

Out[7]:
```
Series([], dtype: bool)
```

1.0.1 Comentário: Dados sem valores NA e sem valores NULL

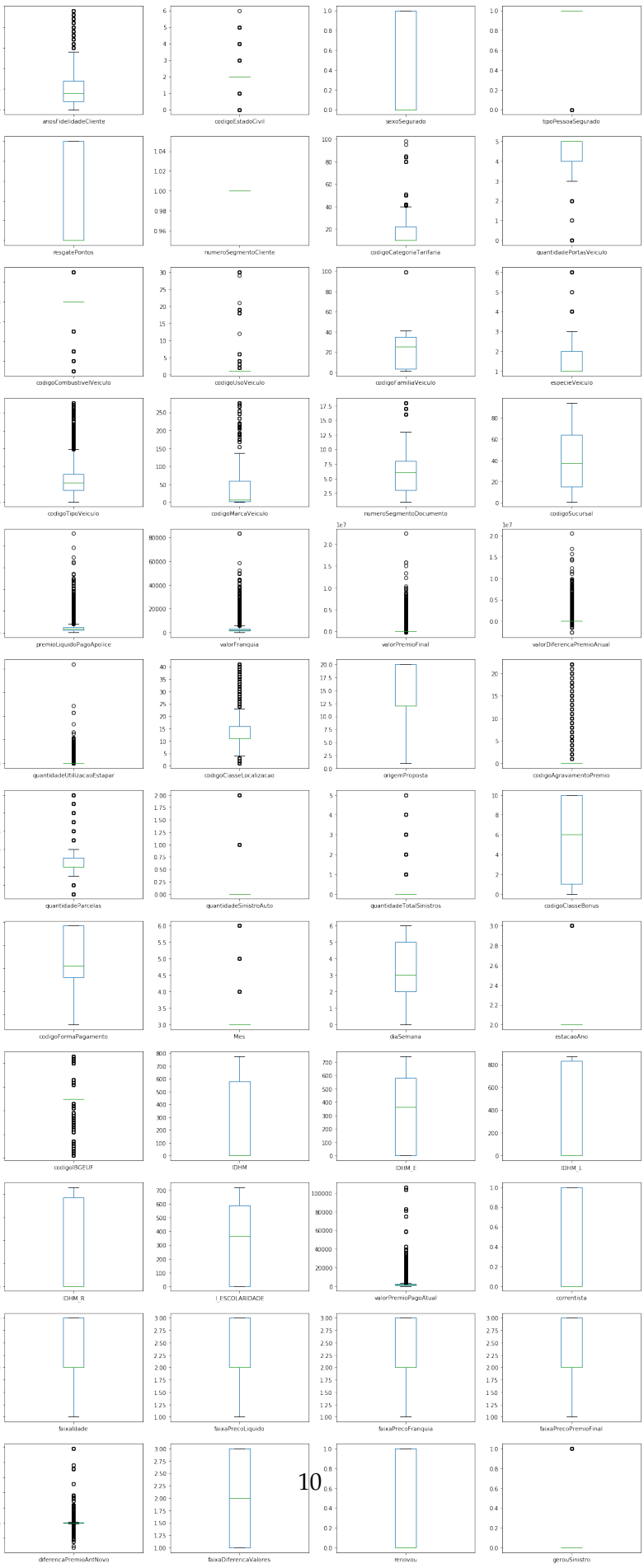
In [8]:
```
# Distribuição da variável TARGET
df.groupby('renovou').size()
```

Out[8]:
```
renovou
0    37290
1    37280
dtype: int64
```

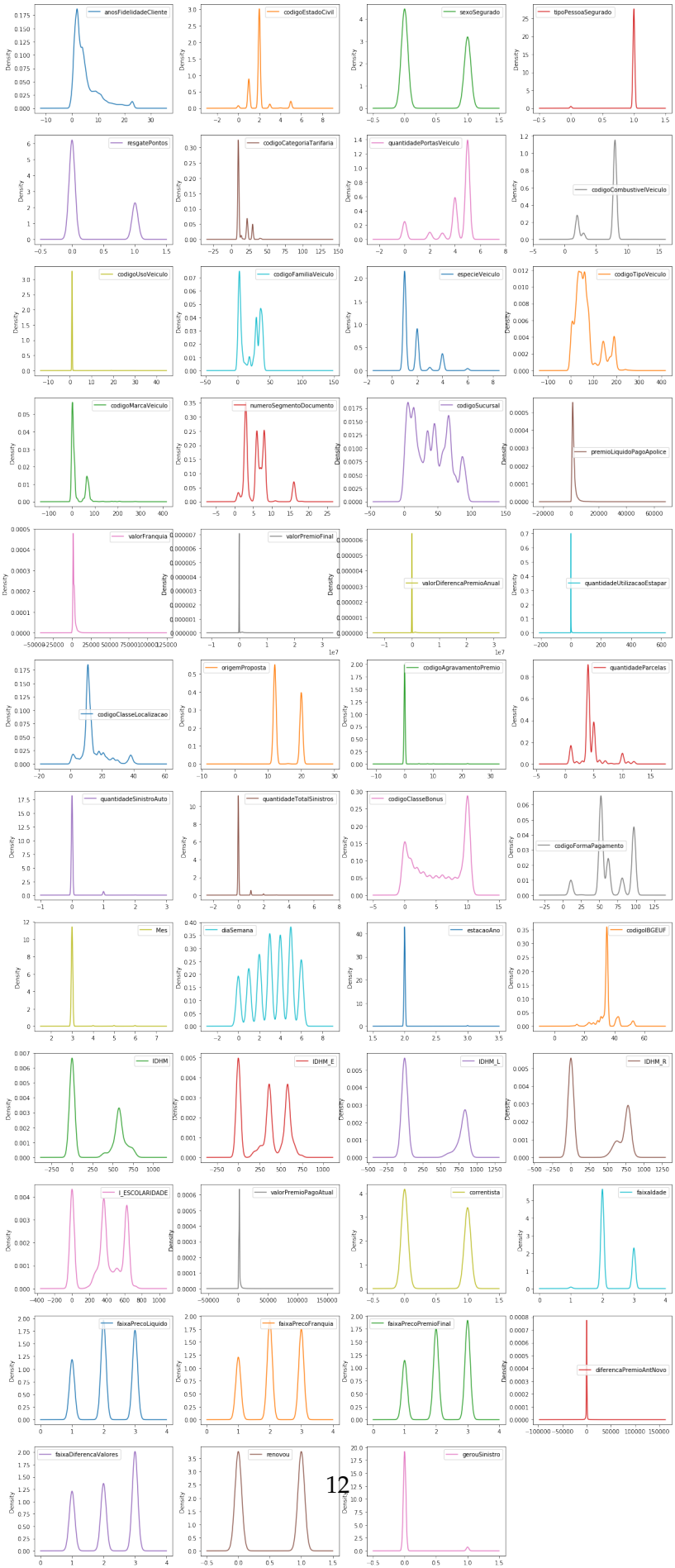
1.0.2 Comentário: Balanceamento da variável Target OK. 50% dos dados (37.290 clientes) que renovaram e 50% de dados (37.280 clientes) que não renovaram (Churn).

1.1 Gráficos

In [9]:
```
#Box-Plots
df.plot(kind = 'box', subplots = True, layout = (12,4), sharex = False, sharey = False
plt.show()
```


anosFidelidadeCliente
codigoEstadoCivil
sexoSegurado
tipoPessoaSegurado
resgatePontos
numeroSegmentoCliente
codigoCategoriaTarifaria
quantidadePortasVeiculo
codigoCombustivelVeiculo
codigoUsoVeiculo
codigoFamiliaVeiculo
especieVeiculo
codigoTipoVeiculo
codigoMarcaVeiculo
numeroSegmentoDocumento
codigoSucursal
premioLiquidoPagoApolice
valorFranquia
valorPremioFinal
valorDiferencaPremioAnual
quantidadeUtilizacaoEstapar
codigoClasseLocalizacao
origemProposta
codigoAgravamentoPremio
quantidadeParcelas
quantidadeSinistroAuto
quantidadeTotalSinistros
codigoClasseBonus
codigoFormaPagamento
Mes
diaSemana
estacaoAno
codigoIBGEUF
IDHM
IDHM_E
IDHM_L
IDHM_R
I_ESCOLARIDADE
valorPremioPagoAtual
correntista
faixaIdade
faixaPrecoLiquido
faixaPrecoFranquia
faixaPrecoPremioFinal
diferencaPremioAntNovo
faixaDiferencaValores
renovou
gerouSinistro

In [10]:
```
#Gráfico de Densidade - Retirado do gráfico a coluna - numeroSegmentoCliente
df.drop("numeroSegmentoCliente", axis=1).plot(kind = 'density', subplots = True, layou
plt.show()
```


anosFidelidadeCliente
codigoEstadoCivil
sexoSegurado
tipoPessoaSegurado
resgatePontos
codigoCategoriaTarifaria
quantidadePortasVeiculo
codigoCombustivelVeiculo
codigoUsoVeiculo
codigoFamiliaVeiculo
especieVeiculo
codigoTipoVeiculo
codigoMarcaVeiculo
numeroSegmentoDocumento
codigoSucursal
premioLiquidoPagoApolice
valorFranquia
valorPremioFinal
valorDiferencaPremioAnual
quantidadeUtilizacaoEstapar
codigoClasseLocalizacao
origemProposta
codigoAgravamentoPremio
quantidadeParcelas
quantidadeSinistroAuto
quantidadeTotalSinistros
codigoClasseBonus
codigoFormaPagamento
Mes
diaSemana
estacaoAno
codigoIBGEUF
IDHM
IDHM_E
IDHM_L
IDHM_R
I_ESCOLARIDADE
valorPremioPagoAtual
correntista
faixaIdade
faixaPrecoLiquido
faixaPrecoFranquia
faixaPrecoPremioFinal
diferencaPremioAntNovo
faixaDiferencaValores
renovou
gerouSinistro
Density

1.1.1 Comentários: Analisando os gráficos acima podemos observar os seguintes pontos:

1) numeroSegmentoCliente => 100% com valor 1, desta forma, sua importância para o modelo preditivo é baixa. Será retirado do dataset; O mesmo acontece com a coluna tipoPessoaSegurado e estacaoAno.

2) Possível OutLiers nas variáveis valorPremioPagoAtual e diferencaPremioAntNovo;

3) Visualmente, algumas colunas possuem valores muito acima e abaixo da média, mas não são outliers pois seus valores condizem com o negócio. Vide dicionários de dados;

4) Grande parte das variáveis são categóricas (qualitativas), estão representadas por números devido a anonimação dos dados. Vide dicionários de dados;

5) Os dados estão em uma distribuição normal.

```
In [11]: #Dados de Correlação entre as variáveis, multiplicado por 100 para melhorar a visuali
         df.corr(method = 'pearson')*100
```

Out[11]:

	anosFidelidadeCliente	codigoEstadoCivil	\
anosFidelidadeCliente	100.000000	5.014704	
codigoEstadoCivil	5.014704	100.000000	
sexoSegurado	2.675238	10.356486	
tipoPessoaSegurado	-0.571099	30.357564	
resgatePontos	5.024082	2.182543	
numeroSegmentoCliente	NaN	NaN	
codigoCategoriaTarifaria	-9.121247	-9.543389	
quantidadePortasVeiculo	11.121668	5.091899	
codigoCombustivelVeiculo	-2.530743	3.671659	
codigoUsoVeiculo	-1.506190	-1.936227	
codigoFamiliaVeiculo	3.750868	-0.005754	
especieVeiculo	-8.674420	-9.273762	
codigoTipoVeiculo	-1.922181	0.881401	
codigoMarcaVeiculo	1.045784	-0.685322	
numeroSegmentoDocumento	-3.820306	-1.923351	
codigoSucursal	7.797603	1.661593	
premioLiquidoPagoApolice	2.513011	-8.946433	
valorFranquia	0.355673	-6.668240	
valorPremioFinal	2.785441	2.171884	
valorDiferencaPremioAnual	2.317634	2.068943	
quantidadeUtilizacaoEstapar	3.541974	-0.261765	
codigoClasseLocalizacao	-6.794267	-1.296245	
origemProposta	-43.205964	-3.448883	
codigoAgravamentoPremio	5.812066	-1.465773	
quantidadeParcelas	-3.794365	-0.307011	
quantidadeSinistroAuto	-0.853850	-0.216035	
quantidadeTotalSinistros	-0.419818	-0.502328	

codigoClasseBonus	44.715513	7.094400
codigoFormaPagamento	5.669700	0.373342
Mes	-6.245340	-0.402711
diaSemana	-5.830016	-0.528743
estacaoAno	-3.973294	-0.127183
codigoIBGEUF	1.352719	1.053490
IDHM	0.521052	-0.595931
IDHM_E	0.126490	-0.024040
IDHM_L	0.101319	0.708686
IDHM_R	-0.202286	0.461335
I_ESCOLARIDADE	0.365951	0.414300
valorPremioPagoAtual	2.412327	-4.852129
correntista	17.829983	4.431756
faixaIdade	22.032856	8.129791
faixaPrecoLiquido	4.481598	-5.119776
faixaPrecoFranquia	1.983720	-4.622289
faixaPrecoPremioFinal	3.812561	6.513415
diferencaPremioAntNovo	1.066042	0.361527
faixaDiferencaValores	-7.608133	0.260551
renovou	26.483523	3.903927
gerouSinistro	-0.795014	-0.251161

	sexoSegurado	tipoPessoaSegurado	resgatePontos	\
anosFidelidadeCliente	2.675238	-0.571099	5.024082	
codigoEstadoCivil	10.356486	30.357564	2.182543	
sexoSegurado	100.000000	11.534962	2.350491	
tipoPessoaSegurado	11.534962	100.000000	8.188769	
resgatePontos	2.350491	8.188769	100.000000	
numeroSegmentoCliente	NaN	NaN	NaN	
codigoCategoriaTarifaria	-15.993235	-18.800697	-4.450219	
quantidadePortasVeiculo	18.951782	5.498711	3.066305	
codigoCombustivelVeiculo	13.559156	13.533738	3.959055	
codigoUsoVeiculo	-2.088281	-1.171977	2.190985	
codigoFamiliaVeiculo	-3.934052	0.295404	-1.220767	
especieVeiculo	-16.790698	-19.801066	-3.865018	
codigoTipoVeiculo	4.150675	5.951926	2.748799	
codigoMarcaVeiculo	1.167854	-3.269756	0.653392	
numeroSegmentoDocumento	-42.483766	1.645860	0.205379	
codigoSucursal	-0.597320	-2.191491	5.296891	
premioLiquidoPagoApolice	-6.392228	-25.051501	1.867021	
valorFranquia	-8.038148	-17.406144	0.738209	
valorPremioFinal	-0.214505	2.947652	25.941298	
valorDiferencaPremioAnual	-0.588302	2.867895	24.881676	
quantidadeUtilizacaoEstapar	1.716809	-0.611377	-0.529140	
codigoClasseLocalizacao	3.015778	1.199660	-2.046817	
origemProposta	-3.337218	0.982317	-8.300879	
codigoAgravamentoPremio	0.123001	0.013902	-0.185269	
quantidadeParcelas	-1.421184	2.658262	16.007280	

quantidadeSinistroAuto	1.663347	0.906558	0.583141
quantidadeTotalSinistros	1.922862	-0.497841	0.959177
codigoClasseBonus	6.724867	0.994011	-5.766567
codigoFormaPagamento	1.870472	2.709135	58.530182
Mes	-1.924887	1.745896	-6.900401
diaSemana	-0.345251	-0.444542	0.490087
estacaoAno	-1.484137	1.076863	-4.202259
codigoIBGEUF	1.589399	0.539149	-0.434564
IDHM	0.129641	0.019604	-0.711077
IDHM_E	0.308663	-0.106957	-0.334836
IDHM_L	0.183739	-0.140376	0.032811
IDHM_R	-0.098760	-0.178840	0.064484
I_ESCOLARIDADE	-0.026012	0.064393	0.505342
valorPremioPagoAtual	-4.366709	-14.520100	1.032243
correntista	4.560023	12.284498	60.846646
faixaIdade	-9.200113	-20.939736	-9.787551
faixaPrecoLiquido	-1.437520	-11.690775	6.908591
faixaPrecoFranquia	-3.905120	-7.958402	5.842611
faixaPrecoPremioFinal	-2.289075	-6.636740	4.578031
diferencaPremioAntNovo	-0.742915	-0.026130	-0.053575
faixaDiferencaValores	0.789880	3.437925	1.236175
renovou	2.542567	3.109424	-0.418242
gerouSinistro	1.714044	0.912139	0.768500

	numeroSegmentoCliente	codigoCategoriaTarifaria	\
anosFidelidadeCliente	NaN	-9.121247	
codigoEstadoCivil	NaN	-9.543389	
sexoSegurado	NaN	-15.993235	
tipoPessoaSegurado	NaN	-18.800697	
resgatePontos	NaN	-4.450219	
numeroSegmentoCliente	NaN	NaN	
codigoCategoriaTarifaria	NaN	100.000000	
quantidadePortasVeiculo	NaN	-65.357658	
codigoCombustivelVeiculo	NaN	-46.032813	
codigoUsoVeiculo	NaN	1.064700	
codigoFamiliaVeiculo	NaN	11.350283	
especieVeiculo	NaN	88.979932	
codigoTipoVeiculo	NaN	-23.498522	
codigoMarcaVeiculo	NaN	18.084065	
numeroSegmentoDocumento	NaN	17.940755	
codigoSucursal	NaN	-4.326731	
premioLiquidoPagoApolice	NaN	18.624714	
valorFranquia	NaN	22.375163	
valorPremioFinal	NaN	0.689842	
valorDiferencaPremioAnual	NaN	0.778188	
quantidadeUtilizacaoEstapar	NaN	-1.614309	
codigoClasseLocalizacao	NaN	6.438009	
origemProposta	NaN	12.128212	

codigoAgravamentoPremio	NaN	-0.467411
quantidadeParcelas	NaN	4.607296
quantidadeSinistroAuto	NaN	-3.666920
quantidadeTotalSinistros	NaN	-2.858798
codigoClasseBonus	NaN	-21.664730
codigoFormaPagamento	NaN	-0.488573
Mes	NaN	3.689242
diaSemana	NaN	2.935936
estacaoAno	NaN	2.442993
codigoIBGEUF	NaN	-8.374955
IDHM	NaN	0.086686
IDHM_E	NaN	0.048056
IDHM_L	NaN	0.536299
IDHM_R	NaN	-0.106188
I_ESCOLARIDADE	NaN	0.205235
valorPremioPagoAtual	NaN	13.536719
correntista	NaN	-8.086148
faixaIdade	NaN	-9.768301
faixaPrecoLiquido	NaN	5.271353
faixaPrecoFranquia	NaN	10.548571
faixaPrecoPremioFinal	NaN	1.705121
diferencaPremioAntNovo	NaN	3.069904
faixaDiferencaValores	NaN	-2.634130
renovou	NaN	-6.406974
gerouSinistro	NaN	-3.678128

	quantidadePortasVeiculo \
anosFidelidadeCliente	11.121668
codigoEstadoCivil	5.091899
sexoSegurado	18.951782
tipoPessoaSegurado	5.498711
resgatePontos	3.066305
numeroSegmentoCliente	NaN
codigoCategoriaTarifaria	-65.357658
quantidadePortasVeiculo	100.000000
codigoCombustivelVeiculo	29.447915
codigoUsoVeiculo	0.824935
codigoFamiliaVeiculo	0.341808
especieVeiculo	-76.512058
codigoTipoVeiculo	14.649839
codigoMarcaVeiculo	0.620061
numeroSegmentoDocumento	-16.592902
codigoSucursal	6.056085
premioLiquidoPagoApolice	14.729199
valorFranquia	6.669353
valorPremioFinal	2.377215
valorDiferencaPremioAnual	2.249878
quantidadeUtilizacaoEstapar	6.585435

```
codigoClasseLocalizacao                  -8.129127
origemProposta                          -11.967068
codigoAgravamentoPremio                   2.677438
quantidadeParcelas                       -8.681431
quantidadeSinistroAuto                    3.102220
quantidadeTotalSinistros                  4.377685
codigoClasseBonus                        30.232011
codigoFormaPagamento                      0.949222
Mes                                      -4.515209
diaSemana                                -2.060014
estacaoAno                               -2.829217
codigoIBGEUF                             10.874067
IDHM                                     -0.304340
IDHM_E                                   -0.073721
IDHM_L                                   -0.182040
IDHM_R                                    0.065034
I_ESCOLARIDADE                           -0.619630
valorPremioPagoAtual                      3.717201
correntista                               7.454962
faixaIdade                               11.269899
faixaPrecoLiquido                        26.309885
faixaPrecoFranquia                       19.007954
faixaPrecoPremioFinal                    12.643274
diferencaPremioAntNovo                   -5.347567
faixaDiferencaValores                   -10.032469
renovou                                   7.576106
gerouSinistro                             3.117460

                         codigoCombustivelVeiculo  codigoUsoVeiculo  ...  \
anosFidelidadeCliente                   -2.530743         -1.506190  ...
codigoEstadoCivil                        3.671659         -1.936227  ...
sexoSegurado                            13.559156         -2.088281  ...
tipoPessoaSegurado                      13.533738         -1.171977  ...
resgatePontos                            3.959055          2.190985  ...
numeroSegmentoCliente                         NaN               NaN  ...
codigoCategoriaTarifaria               -46.032813          1.064700  ...
quantidadePortasVeiculo                 29.447915          0.824935  ...
codigoCombustivelVeiculo               100.000000         -0.150080  ...
codigoUsoVeiculo                        -0.150080        100.000000  ...
codigoFamiliaVeiculo                   -12.187842          0.137502  ...
especieVeiculo                         -37.325019         -0.223635  ...
codigoTipoVeiculo                       24.624695          0.675453  ...
codigoMarcaVeiculo                     -21.698029          0.843952  ...
numeroSegmentoDocumento                 -8.612959          6.917107  ...
codigoSucursal                          -0.603828          0.592393  ...
premioLiquidoPagoApolice               -27.020401          3.896704  ...
valorFranquia                          -32.463549          2.375455  ...
valorPremioFinal                        -1.740283          0.744527  ...
```

valorDiferencaPremioAnual	-1.940635	1.880010	...
quantidadeUtilizacaoEstapar	-0.881448	-0.692218	...
codigoClasseLocalizacao	5.137257	-0.545622	...
origemProposta	-0.454773	-0.662513	...
codigoAgravamentoPremio	-0.522813	-0.606272	...
quantidadeParcelas	-0.353745	2.480651	...
quantidadeSinistroAuto	3.906573	-0.287823	...
quantidadeTotalSinistros	2.876980	0.121038	...
codigoClasseBonus	1.682848	-2.269999	...
codigoFormaPagamento	2.134678	1.907193	...
Mes	-1.409303	0.119934	...
diaSemana	-0.746168	0.228993	...
estacaoAno	-0.966885	0.337751	...
codigoIBGEUF	-1.543988	0.760991	...
IDHM	-0.511133	0.202326	...
IDHM_E	0.043392	0.287404	...
IDHM_L	-0.034343	-0.440798	...
IDHM_R	-0.050096	-0.321936	...
I_ESCOLARIDADE	-0.517358	-0.218473	...
valorPremioPagoAtual	-15.240482	1.701806	...
correntista	5.158288	1.419057	...
faixaIdade	-2.564911	-1.305961	...
faixaPrecoLiquido	-11.618793	4.105815	...
faixaPrecoFranquia	-19.382215	2.737739	...
faixaPrecoPremioFinal	-7.987368	-0.016525	...
diferencaPremioAntNovo	0.440031	-0.615413	...
faixaDiferencaValores	5.976254	-1.252326	...
renovou	-1.378441	-1.692019	...
gerouSinistro	3.844398	-0.376247	...

	valorPremioPagoAtual	correntista	faixaIdade	\
anosFidelidadeCliente	2.412327	17.829983	22.032856	
codigoEstadoCivil	-4.852129	4.431756	8.129791	
sexoSegurado	-4.366709	4.560023	-9.200113	
tipoPessoaSegurado	-14.520100	12.284498	-20.939736	
resgatePontos	1.032243	60.846646	-9.787551	
numeroSegmentoCliente	NaN	NaN	NaN	
codigoCategoriaTarifaria	13.536719	-8.086148	-9.768301	
quantidadePortasVeiculo	3.717201	7.454962	11.269899	
codigoCombustivelVeiculo	-15.240482	5.158288	-2.564911	
codigoUsoVeiculo	1.701806	1.419057	-1.305961	
codigoFamiliaVeiculo	5.597756	-0.946420	-2.826872	
especieVeiculo	8.509016	-7.834309	-9.025478	
codigoTipoVeiculo	-4.710164	3.333868	-1.551627	
codigoMarcaVeiculo	8.038359	0.990639	-3.168296	
numeroSegmentoDocumento	-3.939276	-1.402269	-1.487401	
codigoSucursal	1.081345	8.612936	4.002384	
premioLiquidoPagoApolice	45.543163	-0.014423	1.597808	

valorFranquia	34.116800	0.149932	-3.543795
valorPremioFinal	4.287594	24.337940	-3.526225
valorDiferencaPremioAnual	4.976943	23.288383	-3.462042
quantidadeUtilizacaoEstapar	2.464465	0.168138	-0.248878
codigoClasseLocalizacao	-2.209592	-3.905073	-3.979736
origemProposta	1.780514	-17.663902	-11.867296
codigoAgravamentoPremio	1.850305	0.994738	1.038062
quantidadeParcelas	2.608057	13.019580	-8.536438
quantidadeSinistroAuto	1.191077	-0.158976	-0.170739
quantidadeTotalSinistros	2.531253	0.651703	-0.387265
codigoClasseBonus	-1.926470	5.011623	23.764775
codigoFormaPagamento	1.955782	55.171674	-8.994006
Mes	2.423439	-9.624945	-2.664052
diaSemana	1.660611	-0.462621	-1.674559
estacaoAno	1.395280	-5.892514	-1.149350
codigoIBGEUF	1.551038	-0.242606	1.627966
IDHM	0.050426	-0.755709	-0.088996
IDHM_E	-0.257273	-0.300914	0.155944
IDHM_L	0.073487	0.090388	0.283942
IDHM_R	0.104356	-0.134358	-0.058364
I_ESCOLARIDADE	0.218977	0.270824	0.135577
valorPremioPagoAtual	100.000000	-0.571654	1.135652
correntista	-0.571654	100.000000	-7.387720
faixaIdade	1.135652	-7.387720	100.000000
faixaPrecoLiquido	28.692358	6.160797	-1.689251
faixaPrecoFranquia	19.885733	6.025006	-12.110204
faixaPrecoPremioFinal	17.595199	3.914765	0.587911
diferencaPremioAntNovo	81.941102	-0.626761	0.234817
faixaDiferencaValores	10.975082	-0.736955	-0.936598
renovou	-6.782029	4.746491	9.370425
gerouSinistro	1.175294	-0.061010	-0.204415

	faixaPrecoLiquido	faixaPrecoFranquia	\
anosFidelidadeCliente	4.481598	1.983720	
codigoEstadoCivil	-5.119776	-4.622289	
sexoSegurado	-1.437520	-3.905120	
tipoPessoaSegurado	-11.690775	-7.958402	
resgatePontos	6.908591	5.842611	
numeroSegmentoCliente	NaN	NaN	
codigoCategoriaTarifaria	5.271353	10.548571	
quantidadePortasVeiculo	26.309885	19.007954	
codigoCombustivelVeiculo	-11.618793	-19.382215	
codigoUsoVeiculo	4.105815	2.737739	
codigoFamiliaVeiculo	16.582501	18.068302	
especieVeiculo	-5.712972	0.505925	
codigoTipoVeiculo	-2.973019	-9.690709	
codigoMarcaVeiculo	18.865623	21.486508	
numeroSegmentoDocumento	1.898068	4.717543	

codigoSucursal	2.109919	5.157054
premioLiquidoPagoApolice	63.539328	41.979184
valorFranquia	39.867214	60.103153
valorPremioFinal	10.427649	7.475123
valorDiferencaPremioAnual	10.299887	7.417918
quantidadeUtilizacaoEstapar	9.159141	6.383312
codigoClasseLocalizacao	-2.630879	-8.956007
origemProposta	-0.884607	-0.292429
codigoAgravamentoPremio	3.564931	0.858010
quantidadeParcelas	1.673433	1.686203
quantidadeSinistroAuto	2.693506	-3.988548
quantidadeTotalSinistros	4.429357	-0.980746
codigoClasseBonus	-1.898221	1.465092
codigoFormaPagamento	7.338274	5.374771
Mes	0.147437	2.132965
diaSemana	-0.340060	-0.949458
estacaoAno	-0.097270	1.177542
codigoIBGEUF	6.586344	5.972877
IDHM	0.065611	-0.220788
IDHM_E	0.031877	-0.028036
IDHM_L	-0.463415	-0.077794
IDHM_R	-0.155010	-0.134947
I_ESCOLARIDADE	-0.530178	-0.368951
valorPremioPagoAtual	28.692358	19.885733
correntista	6.160797	6.025006
faixaIdade	-1.689251	-12.110204
faixaPrecoLiquido	100.000000	47.711494
faixaPrecoFranquia	47.711494	100.000000
faixaPrecoPremioFinal	51.663318	33.133862
diferencaPremioAntNovo	-8.985828	-4.902828
faixaDiferencaValores	-33.443038	-15.577466
renovou	-5.721541	-1.040162
gerouSinistro	2.638502	-3.969809

	faixaPrecoPremioFinal	diferencaPremioAntNovo	\
anosFidelidadeCliente	3.812561	1.066042	
codigoEstadoCivil	6.513415	0.361527	
sexoSegurado	-2.289075	-0.742915	
tipoPessoaSegurado	-6.636740	-0.026130	
resgatePontos	4.578031	-0.053575	
numeroSegmentoCliente	NaN	NaN	
codigoCategoriaTarifaria	1.705121	3.069904	
quantidadePortasVeiculo	12.643274	-5.347567	
codigoCombustivelVeiculo	-7.987368	0.440031	
codigoUsoVeiculo	-0.016525	-0.615413	
codigoFamiliaVeiculo	8.388400	-1.148700	
especieVeiculo	-2.805597	6.082687	
codigoTipoVeiculo	-2.334200	-0.870226	

codigoMarcaVeiculo	10.477410	-2.766159
numeroSegmentoDocumento	0.925745	-0.393130
codigoSucursal	2.008155	-0.321737
premioLiquidoPagoApolice	37.819186	-13.712321
valorFranquia	25.865543	-5.715024
valorPremioFinal	13.505270	-3.176669
valorDiferencaPremioAnual	14.016309	-2.159140
quantidadeUtilizacaoEstapar	3.559282	-2.038582
codigoClasseLocalizacao	-0.883249	-1.840456
origemProposta	-1.338993	0.594747
codigoAgravamentoPremio	-2.961660	-0.402380
quantidadeParcelas	2.612309	3.342773
quantidadeSinistroAuto	0.423331	0.477429
quantidadeTotalSinistros	1.256092	0.736637
codigoClasseBonus	-4.867843	1.054792
codigoFormaPagamento	4.204056	-0.119144
Mes	1.227845	2.253672
diaSemana	-0.945209	1.043548
estacaoAno	0.753724	1.449135
codigoIBGEUF	1.826959	-0.872122
IDHM	-0.087268	-0.118532
IDHM_E	0.124001	0.247176
IDHM_L	0.388331	-0.064277
IDHM_R	-0.064735	-0.305158
I_ESCOLARIDADE	-0.191541	0.123294
valorPremioPagoAtual	17.595199	81.941102
correntista	3.914765	-0.626761
faixaIdade	0.587911	0.234817
faixaPrecoLiquido	51.663318	-8.985828
faixaPrecoFranquia	33.133862	-4.902828
faixaPrecoPremioFinal	100.000000	-4.772935
diferencaPremioAntNovo	-4.772935	100.000000
faixaDiferencaValores	-20.048584	30.504503
renovou	-2.287753	-4.789050
gerouSinistro	0.435221	0.496925

	faixaDiferencaValores	renovou	gerouSinistro
anosFidelidadeCliente	-7.608133	26.483523	-0.795014
codigoEstadoCivil	0.260551	3.903927	-0.251161
sexoSegurado	0.789880	2.542567	1.714044
tipoPessoaSegurado	3.437925	3.109424	0.912139
resgatePontos	1.236175	-0.418242	0.768500
numeroSegmentoCliente	NaN	NaN	NaN
codigoCategoriaTarifaria	-2.634130	-6.406974	-3.678128
quantidadePortasVeiculo	-10.032469	7.576106	3.117460
codigoCombustivelVeiculo	5.976254	-1.378441	3.844398
codigoUsoVeiculo	-1.252326	-1.692019	-0.376247
codigoFamiliaVeiculo	-5.610817	2.033133	-1.105055

```
especieVeiculo                    2.158603   -6.421152      -3.654501
codigoTipoVeiculo                 1.897415   -0.296729       2.713073
codigoMarcaVeiculo               -5.908591    0.711142      -0.813698
numeroSegmentoDocumento          -2.102004   -2.245693      -1.149111
codigoSucursal                   -0.316910    3.218144      -0.755100
premioLiquidoPagoApolice        -28.411852   -4.281904       1.259223
valorFranquia                   -14.869507   -0.765086      -3.183193
valorPremioFinal                 -4.004577    0.008023       0.380107
valorDiferencaPremioAnual        -4.133593    0.072314       0.335540
quantidadeUtilizacaoEstapar      -3.431996   -0.690578      -0.166905
codigoClasseLocalizacao          -1.417140   -7.557477       1.142499
origemProposta                    4.740992  -22.397376       1.238976
codigoAgravamentoPremio          -1.212809    3.986527       0.032272
quantidadeParcelas               10.210964   -4.270177      -0.234374
quantidadeSinistroAuto            5.773433   -0.962291      98.366474
quantidadeTotalSinistros          6.193473   -0.807332      80.666636
codigoClasseBonus                -4.645868   22.387871      -1.140064
codigoFormaPagamento              1.031287   -0.851513       0.630807
Mes                               3.992257    0.285012      -0.201276
diaSemana                        -1.497312    0.811329      -0.242676
estacaoAno                        2.756192   -0.307310      -0.327542
codigoIBGEUF                     -2.963118    3.237789       1.008509
IDHM                              0.061701    0.237241      -0.397240
IDHM_E                           -0.715785    0.556927       0.225123
IDHM_L                            0.605651    0.293922      -0.073902
IDHM_R                            0.548921   -0.342850      -0.190801
I_ESCOLARIDADE                    0.806432    0.133432      -0.019571
valorPremioPagoAtual             10.975082   -6.782029       1.175294
correntista                      -0.736955    4.746491      -0.061010
faixaIdade                       -0.936598    9.370425      -0.204415
faixaPrecoLiquido                -33.443038   -5.721541       2.638502
faixaPrecoFranquia               -15.577466   -1.040162      -3.969809
faixaPrecoPremioFinal            -20.048584   -2.287753       0.435221
diferencaPremioAntNovo            30.504503   -4.789050       0.496925
faixaDiferencaValores            100.000000  -10.140009       5.874958
renovou                          -10.140009  100.000000      -0.821628
gerouSinistro                      5.874958   -0.821628     100.000000

[48 rows x 48 columns]
```

In [12]:
```
# Correlação em gráfico de HeatMap
f, ax = plt.subplots(figsize=(15, 40))
sns.heatmap(df.corr(method = 'pearson'),linewidths=.5, ax=ax)
```

Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7effb1b88150>
```


anosFidelidadeCliente
codigoEstadoCivil
sexoSegurado
tipoPessoaSegurado
resgatePontos
numeroSegmentoCliente
codigoCategoriaTarifaria
quantidadePortasVeiculo
codigoCombustivelVeiculo
codigoUsoVeiculo
codigoFamiliaVeiculo
especieVeiculo
codigoTipoVeiculo
codigoMarcaVeiculo
numeroSegmentoDocumento
codigoSucursal
premioLiquidoPagoApolice
valorFranquia
valorPremioFinal
valorDiferencaPremioAnual
quantidadeUtilizacaoEstapar
codigoClasseLocalizacao
origemProposta
codigoAgravamentoPremio
quantidadeParcelas
quantidadeSinistroAuto
quantidadeTotalSinistros
codigoClasseBonus
codigoFormaPagamento
Mes
diaSemana
estacaoAno
codigoIBGEUF
IDHM
IDHM_E
IDHM_L
IDHM_R
I_ESCOLARIDADE
valorPremioPagoAtual
correntista
faixaIdade
faixaPrecoLiquido
faixaPrecoFranquia
faixaPrecoPremioFinal
faixaPrecoPremioNovo
diferencaPremioAntNovo
faixaDiferencaValores
renovou
gerouSinistro
0.9
0.6
0.3
0.0
−0.3
−0.6

In [13]:
```
# Correlação com a variável Target ordenado, feita a multiplicacao por 100 para melho
df.drop("renovou", axis=1).apply(lambda x: x.corr(df.renovou) * 100).sort_values()
```

Out[13]:
```
origemProposta               -22.397376
faixaDiferencaValores        -10.140009
codigoClasseLocalizacao       -7.557477
valorPremioPagoAtual          -6.782029
especieVeiculo                -6.421152
codigoCategoriaTarifaria      -6.406974
faixaPrecoLiquido             -5.721541
diferencaPremioAntNovo        -4.789050
premioLiquidoPagoApolice      -4.281904
quantidadeParcelas            -4.270177
faixaPrecoPremioFinal         -2.287753
numeroSegmentoDocumento       -2.245693
codigoUsoVeiculo              -1.692019
codigoCombustivelVeiculo      -1.378441
faixaPrecoFranquia            -1.040162
quantidadeSinistroAuto        -0.962291
codigoFormaPagamento          -0.851513
gerouSinistro                 -0.821628
quantidadeTotalSinistros      -0.807332
valorFranquia                 -0.765086
quantidadeUtilizacaoEstapar   -0.690578
resgatePontos                 -0.418242
IDHM_R                        -0.342850
estacaoAno                    -0.307310
codigoTipoVeiculo             -0.296729
valorPremioFinal               0.008023
valorDiferencaPremioAnual      0.072314
I_ESCOLARIDADE                 0.133432
IDHM                           0.237241
Mes                            0.285012
IDHM_L                         0.293922
IDHM_E                         0.556927
codigoMarcaVeiculo             0.711142
diaSemana                      0.811329
codigoFamiliaVeiculo           2.033133
sexoSegurado                   2.542567
tipoPessoaSegurado             3.109424
codigoSucursal                 3.218144
codigoIBGEUF                   3.237789
codigoEstadoCivil              3.903927
codigoAgravamentoPremio        3.986527
correntista                    4.746491
```

```
quantidadePortasVeiculo          7.576106
faixaIdade                       9.370425
codigoClasseBonus               22.387871
anosFidelidadeCliente           26.483523
numeroSegmentoCliente                 NaN
dtype: float64
```

1.1.2 Comentários: Pela análise de correlação, temos como principais fatores que influenciam no Churn as variáveis com valores maiores que 4 (positivo e negativo) abaixo:

anosFidelidadeCliente, codigoClasseBonus, faixaIdade, quantidadePortasVeiculo, correntista, origemProposta, faixaDiferencaValores, codigoClasseLocalizacao, valorPremioPagoAtual, especieVeiculo, codigoCategoriaTarifaria, faixaPrecoLiquido, diferencaPremioAntNovo e premioLiquidoPagoApolice

1.1.3 Iremos utilizar a técnica de Features Selection para selecionar as melhores features para o Churn, pois é um método mais moderno e efetivo.

In [14]:
```
# Tratamento de identificação de colunas com alta colinearidade.
# Se a correlação de duas variáveis é maior de 0.90, é um indicativo de correlação fo
# Isso é um índicio de problema para o modelo preditivo.
corr_matrix = df.corr().abs()
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(np.bool))

# Achando os indexes das colunas com correlação maior que 0.90
resultado = [column for column in upper.columns if any(upper[column] > 0.90)]
resultado
```

Out[14]:
```
['gerouSinistro']
```

1.1.4 Comentário: As variáveis 'valorDiferencaPremioAnual', 'IDHM_E', 'IDHM_L', 'I_ESCOLARIDADE' e 'gerouSinistro' apresentam alta colinearidade (Acima de 90%). Desta forma, serão descartadas na construção do modelo preditivo.

2 Criando o modelo Preditivo

In [15]:
```
# Retirando as variaveis problemáticas e com alta colinearidade através das análises
# Total de 8 colunas
df = df.drop(['valorDiferencaPremioAnual',
 'IDHM_E',
 'IDHM_L',
 'I_ESCOLARIDADE',
 'gerouSinistro',
'numeroSegmentoCliente',
'tipoPessoaSegurado',
'estacaoAno'], axis=1)

df.shape
```

```
Out[15]: (74570, 40)

In [16]: df.columns

Out[16]: Index(['anosFidelidadeCliente', 'codigoEstadoCivil', 'sexoSegurado',
                'resgatePontos', 'codigoCategoriaTarifaria', 'quantidadePortasVeiculo',
                'codigoCombustivelVeiculo', 'codigoUsoVeiculo', 'codigoFamiliaVeiculo',
                'especieVeiculo', 'codigoTipoVeiculo', 'codigoMarcaVeiculo',
                'numeroSegmentoDocumento', 'codigoSucursal', 'premioLiquidoPagoApolice',
                'valorFranquia', 'valorPremioFinal', 'quantidadeUtilizacaoEstapar',
                'codigoClasseLocalizacao', 'origemProposta', 'codigoAgravamentoPremio',
                'quantidadeParcelas', 'quantidadeSinistroAuto',
                'quantidadeTotalSinistros', 'codigoClasseBonus', 'codigoFormaPagamento',
                'Mes', 'diaSemana', 'codigoIBGEUF', 'IDHM', 'IDHM_R',
                'valorPremioPagoAtual', 'correntista', 'faixaIdade',
                'faixaPrecoLiquido', 'faixaPrecoFranquia', 'faixaPrecoPremioFinal',
                'diferencaPremioAntNovo', 'faixaDiferencaValores', 'renovou'],
               dtype='object')

In [17]: # Separandos os dados de predição e de target
         array = df.values
         X = array[:,0:39]
         Y = array[:,39]

         seed = 1313
```

2.0.1 Normalizando e padronizando os dados do dataframe

```
In [18]: # Gerando a nova escala (normalizando os dados)
         scaler = MinMaxScaler(feature_range = (0, 1))
         rescaledX = scaler.fit_transform(X)

         #Gerando padronização dos valores
         scalerP = StandardScaler().fit(rescaledX)
         standardX = scalerP.transform(rescaledX)
```

2.0.2 Features Selection

```
In [19]: # Seleção das melhores variáveis para construir o modelo pretidivo. Iremos utilizar o

         from sklearn.ensemble import RandomForestClassifier
         from sklearn.feature_selection import SelectFromModel

         #Com Normalizacao e Padronizacao
         clf_normalizado = RandomForestClassifier(random_state=seed)
         selector_normalizado = clf_normalizado.fit(standardX, Y)
         fs_normalizado = SelectFromModel(selector_normalizado, prefit=True)

         #Dados não normalizados e padronizados
```

```
clf = RandomForestClassifier(random_state=seed)
selector = clf.fit(X, Y)
fs = SelectFromModel(selector, prefit=True)

standardX_new = fs.transform(X)
standardX_norm = fs_normalizado.transform(standardX)
standardX_new.shape
```

Out[19]: (74570, 10)

In [20]:
```
# Montando nova estrutura de dados com as colunas selecionadas
mask = fs.get_support()
colunas = df.columns
new_features = []
for bool, feature in zip(mask, colunas):
    if bool:
        new_features.append(feature)


df_selection = pd.DataFrame(standardX_new, columns=new_features)
df_selection['renovou'] = Y

df_selection_norm = pd.DataFrame(standardX_norm, columns=new_features)
df_selection_norm['renovou'] = Y

df_selection_norm.head(10)
```

Out[20]:

	anosFidelidadeCliente	codigoSucursal	premioLiquidoPagoApolice	\
0	1.130777	0.170723	0.226174	
1	1.130777	-0.131435	-0.217892	
2	1.130777	-0.131435	-0.217892	
3	1.328293	-0.131435	-0.016260	
4	-0.054318	1.077198	-0.425752	
5	-0.449349	-0.811291	-0.477932	
6	-0.844381	-1.340067	-0.332907	
7	-0.844381	-0.131435	-0.681990	
8	-0.844381	1.039428	-0.341391	
9	-0.646865	1.945902	0.991274	

	valorFranquia	valorPremioFinal	codigoClasseBonus	IDHM	IDHM_R	\
0	-0.092357	-0.222461	1.169820	1.003527	-0.902894	
1	0.205334	-0.220200	1.169820	1.003527	-0.902894	
2	0.205334	-0.220200	1.169820	-0.929167	-0.902821	
3	-0.868368	-0.223022	1.169820	-0.928896	1.237630	
4	0.061530	-0.220659	-0.883880	-0.929167	-0.902821	
5	-0.218623	-0.220374	-1.140592	1.003527	-0.902894	
6	-0.469402	-0.222213	1.169820	1.003527	-0.902894	
7	-0.810496	-0.221225	-1.397305	-0.929167	-0.902821	

```
8      -0.605313         -0.220692          -1.397305 -0.928896  1.237630
9       1.482026         -0.216794           1.169820  1.003527 -0.902894

   valorPremioPagoAtual  diferencaPremioAntNovo  renovou
0             -0.041200               -0.191464      0.0
1             -0.041200                0.094450      0.0
2             -0.041200                0.094450      0.0
3             -0.049023               -0.044076      0.0
4             -0.291366               -0.050064      0.0
5             -0.354881               -0.087137      0.0
6             -0.041200                0.168504      0.0
7             -0.041200                0.393263      0.0
8             -0.041200                0.173966      0.0
9              2.241633                1.855901      0.0
```

In [21]:
```
# Separando os dados predição e target
array = df_selection_norm.values
X = array[:,0:10]
Y = array[:,10]

array = df_selection.values
X_comum = array[:,0:10]
Y_comum = array[:,10]

X.shape, Y.shape
```

Out[21]: ((74570, 10), (74570,))

3 Criando os modelos

In [22]:
```
#!pip install tensorflow
#!pip install keras
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from sklearn.model_selection import StratifiedKFold
```

```
Using TensorFlow backend.
```

In [23]:
```
#Função de Construção do Modelo com Redes Neurais - KERAS

# Definindo os valores para o número de folds
num_folds = 10

#Criando a Rede Neural - 2 camadas
def build_model():
    model = Sequential()
```

```
        model.add(Dense(20, activation='relu', input_dim=10))
        model.add(Dropout(0.5))
        model.add(Dense(20, activation='relu'))
        model.add(Dropout(0.5))
        model.add(Dense(20, activation='relu'))
        model.add(Dense(units=1, activation='sigmoid'))
        model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['accuracy
        return model

    #Cross-Validation de Redes Neurais
    def k_fold_train(model, x_train, y_train):
        kfold = StratifiedKFold(n_splits=num_folds, shuffle=True, random_state=seed)
        k_fold_accuracies = []

        for k_train, k_test in kfold.split(x_train,y_train):
            history = model.fit(
                x_train[k_train],
                y_train[k_train],
                epochs=10,
                batch_size=20,
                verbose=0,
                validation_data=(x_train[k_test], y_train[k_test]))

            #Acuraria
            score = history.history['val_accuracy']
            k_fold_accuracies.append(score)

        return k_fold_accuracies
```

In [24]:
```
# Avaliação dos modelos e Considerações
def montarBoxPlot(resultados, nomes):

    sns.boxplot( y=nomes, x=resultados);
    plt.title('Comparação entre os Algoritmos')
    plt.show()
```

In [25]:
```
def montarMatrixConfusion(y_test, y_pred, title):

    cm = confusion_matrix(y_test, y_pred)
    print("\n Tabela - %s" % title)
    print(cm)
    print("\n")
    #sns.heatmap(cm, annot=True)
    sns.heatmap(cm/np.sum(cm), annot=True, fmt='.2%', cmap='Blues')
    plt.title(title)
    plt.show()
```

```
            #Curva ROC do modelo
            print("\n")
            montarCurveROC(y_test, y_pred, 'Curva ROC do Modelo')

In [26]: def montarCurveROC(y_test, y_pred, title):

            fpr, tpr, threshold = roc_curve(y_test, y_pred)
            roc_auc = roc_auc_score(y_test, y_pred)

            plt.title(title)
            plt.plot(fpr, tpr, 'b', label='AUC = %0.2f' % roc_auc)
            plt.legend(loc='lower right')
            plt.plot([0, 1], [0, 1], 'r--')
            plt.xlim([0, 1])
            plt.ylim([0, 1])
            plt.xlabel('Falso Positivo')
            plt.ylabel('Verdadeiro Positivo')
            plt.show()

In [46]: # Preparando a lista de modelos com Normalização e Padronização dos dados
         def executar(X, Y, titulo, executaSVC):
            modelos = []
            modelos.append(('LR', LogisticRegression()))
            modelos.append(('LDA', LinearDiscriminantAnalysis()))
            modelos.append(('NB', GaussianNB()))
            modelos.append(('KNN', KNeighborsClassifier()))
            modelos.append(('CART', DecisionTreeClassifier()))
            modelos.append(('EXTREE', ExtraTreesClassifier()))
            modelos.append(('SVC', OneVsRestClassifier(SVC(kernel='linear', probability=True,
            modelos.append(('RAFOR', RandomForestClassifier(random_state=seed)))
            modelos.append(('LGB', LGBMClassifier(num_leaves=500, learning_rate=0.15, n_estima
            modelos.append(('XGB', XGBClassifier(n_estimators=150)))
            modelos.append(('KERAS', build_model()))

            # Avaliando cada modelo e exibindo os resultados
            resultados = []
            nomes = []

            kfold = KFold(n_splits = num_folds, random_state = seed, shuffle=True)

            for nome, modelo in modelos:
                if nome != 'KERAS':
                    if nome == 'SVC':

                        if executaSVC == True:
                            # Gerando uma amostra menor ao modelo, devido ao tempo de process
                            X_treinoSVC, X_testeSVC, y_treinoSVC, y_testeSVC = train_test_spli
```

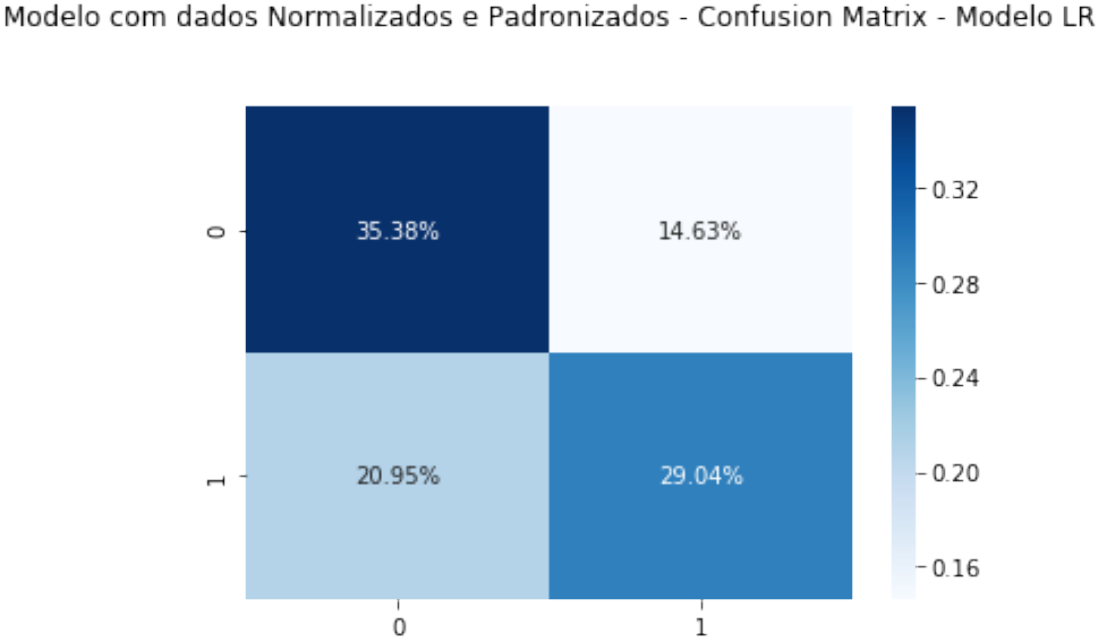
```
                cv_results = cross_val_predict(modelo, X_treinoSVC, y_treinoSVC, 
                cv_results_score = cross_val_score(modelo, X_treinoSVC, y_treinoS

                #Confusion Matrix
                montarMatrixConfusion(y_treinoSVC, cv_results, titulo + " - Confus
                msg = "Modelo: %s => %.2f%s (Acuraria) - %s" % (nome, cv_results_

            else:
                cv_results_score = [0.50,0.50,0.50,0.50,0.50,0.50,0.50,0.50,0.50,
                msg = "Modelo: %s => %.2f%s (Acuraria) - %s" % (nome, 0.50 * 100,

        else:
            cv_results = cross_val_predict(modelo, X, Y, cv = kfold)
            cv_results_score = cross_val_score(modelo, X, Y, cv = kfold, scoring =

            #Confusion Matrix
            montarMatrixConfusion(Y, cv_results, titulo + " - Confusion Matrix - M
            msg = "Modelo: %s => %.2f%s (Acuraria) - %s" % (nome, cv_results_score


        resultados.append(cv_results_score)
        nomes.append(nome)

    else:
        mean_acc = k_fold_train(modelo, X, Y)
        resultados.append(np.array(mean_acc[0]))
        nomes.append(nome)
        msg = "Modelo: %s => %.2f%s (Acuraria) - %s" % (nome, np.mean(mean_acc) *

    print("\n")
    print(msg)

print("\n")
print("\n")
montarBoxPlot(resultados, nomes)
return resultados
```

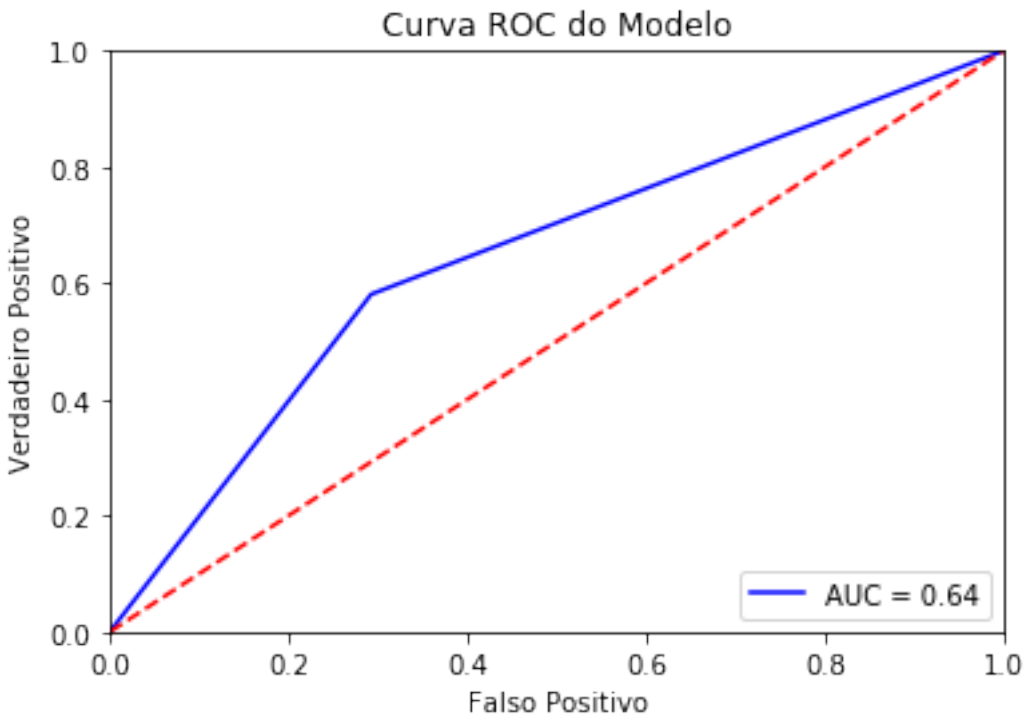
In [28]:
```
# Modelos com os dados Normalizados
```

In [29]:
```
# Devido ao tempo muito alto de processamento (mais de 4hrs, o SVC foi criado e treina
```

In [30]:
```
resultados = executar(X, Y, 'Modelo com dados Normalizados e Padronizados', True)
```

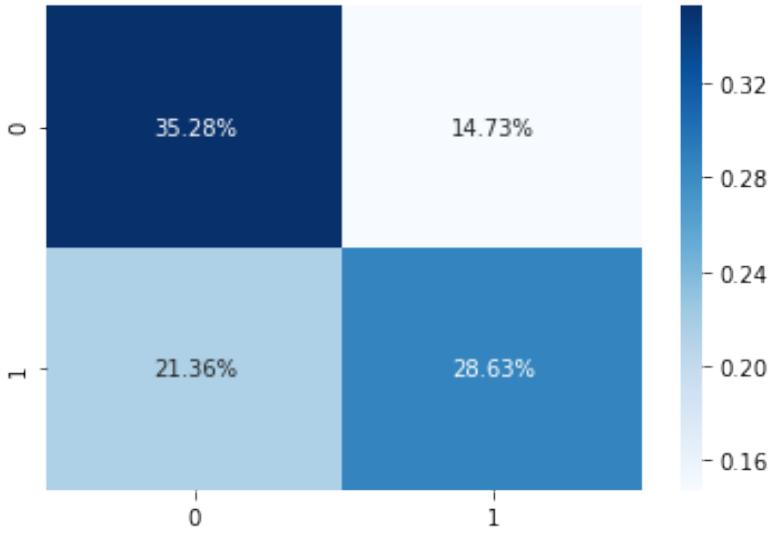
```
 Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo LR


[[26382 10908]
 [15623 21657]]
```

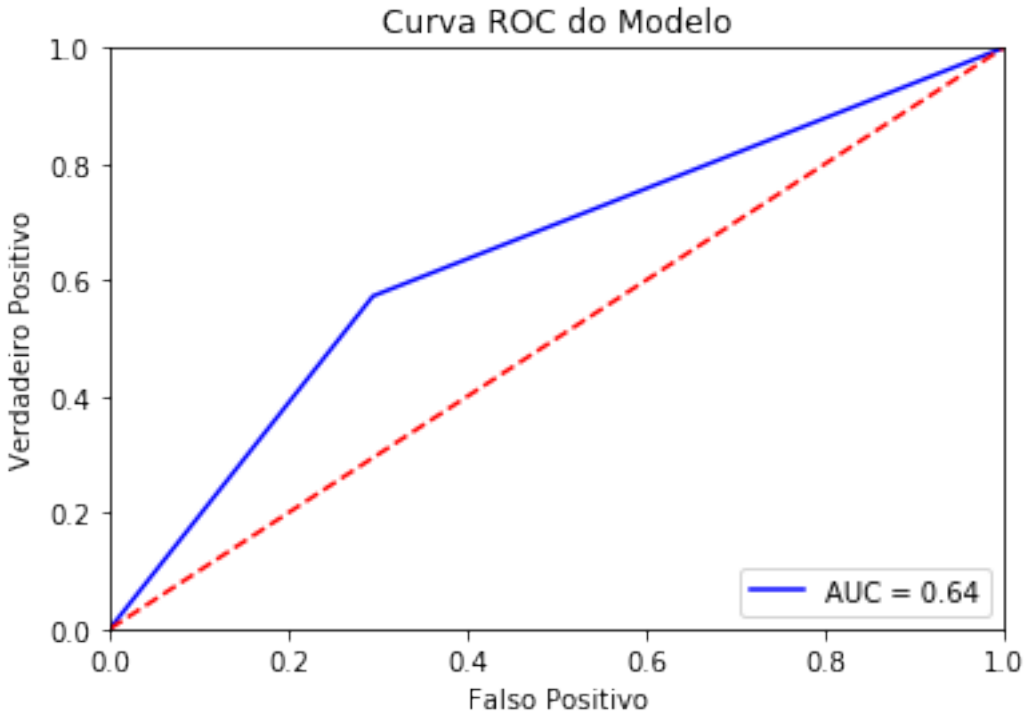

Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo LR
35.38%
14.63%
20.95%
29.04%
0
1
0
1
0.32
0.28
0.24
0.20
0.16



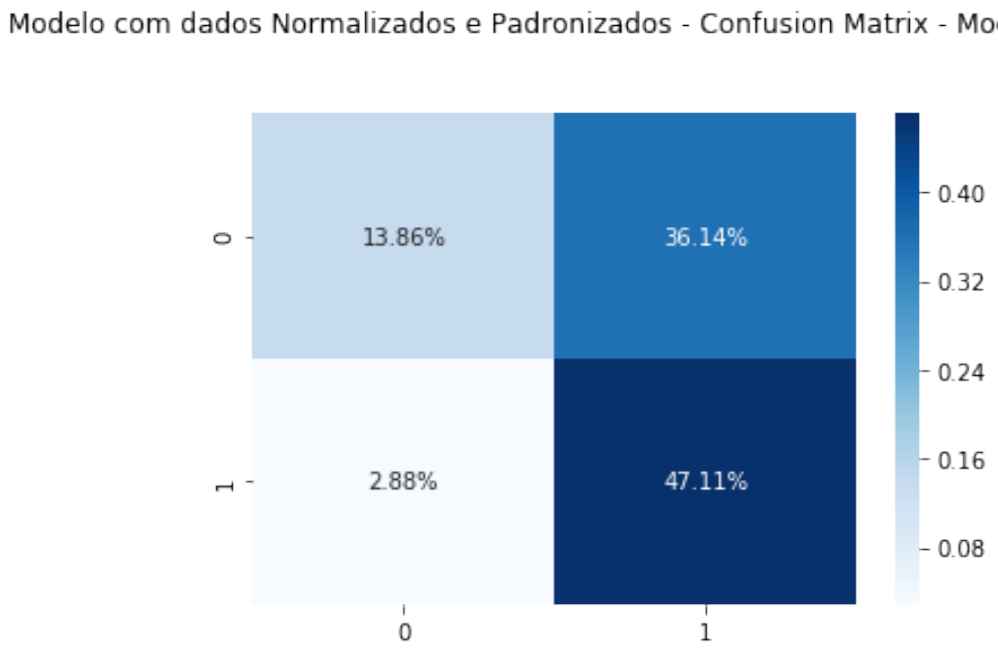

```
Modelo: LR => 64.42% (Acuraria) - Modelo com dados Normalizados e Padronizados

 Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo LDA


[[26309 10981]
 [15930 21350]]
```

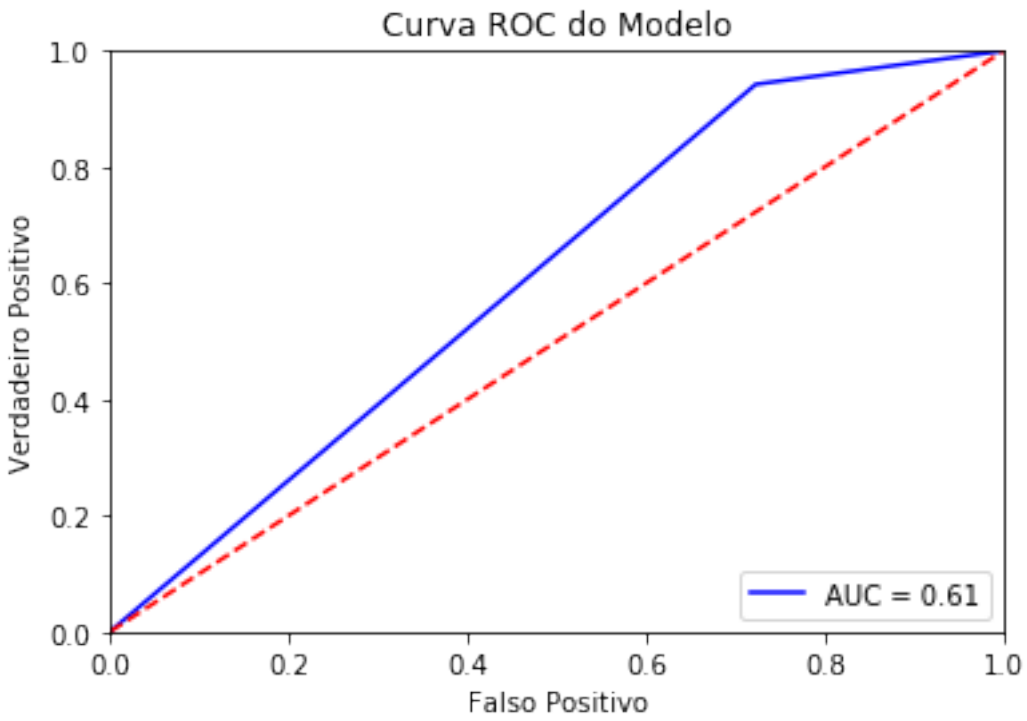
Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo LDA





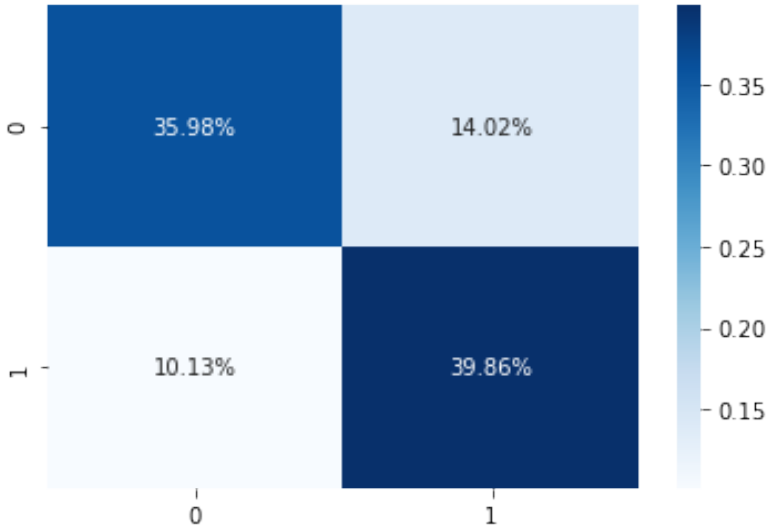
```
Modelo: LDA => 63.91% (Acuraria) - Modelo com dados Normalizados e Padronizados

 Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo NB


[[10337 26953]
 [ 2151 35129]]
```

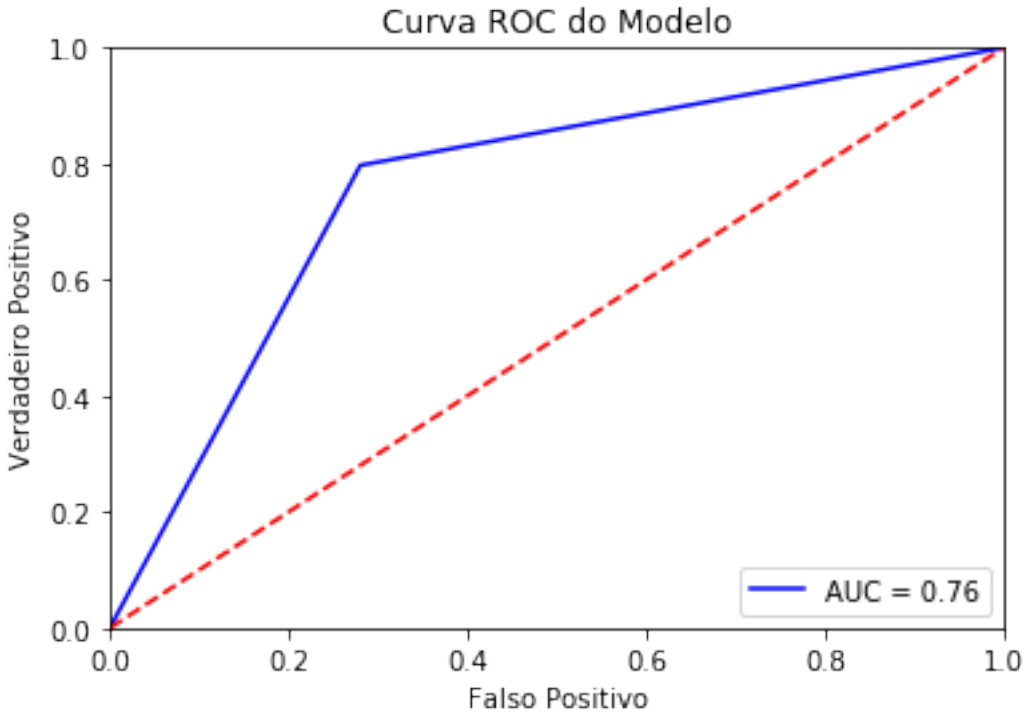




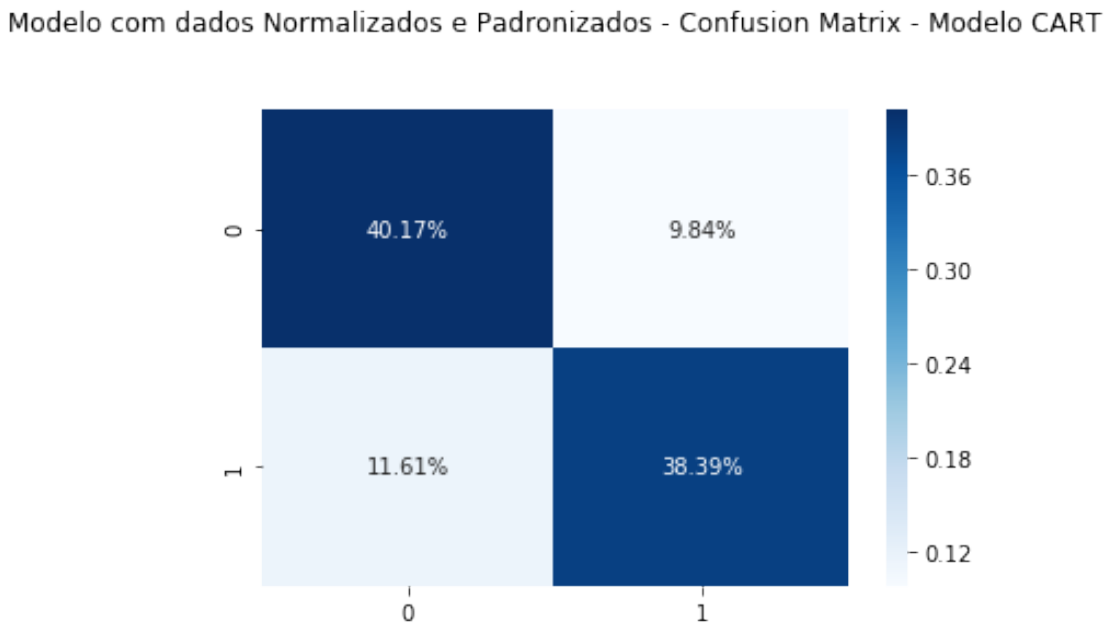

```
Modelo: NB => 60.97% (Acuraria) - Modelo com dados Normalizados e Padronizados

 Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo KNN


[[26833 10457]
 [ 7557 29723]]
```

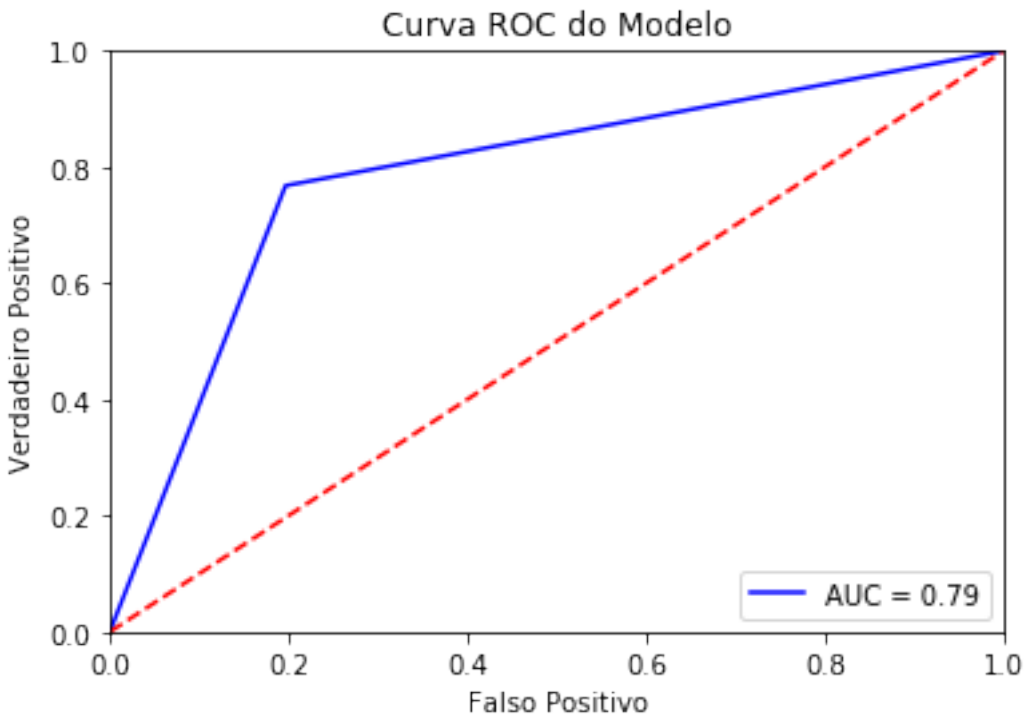
Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo KNN





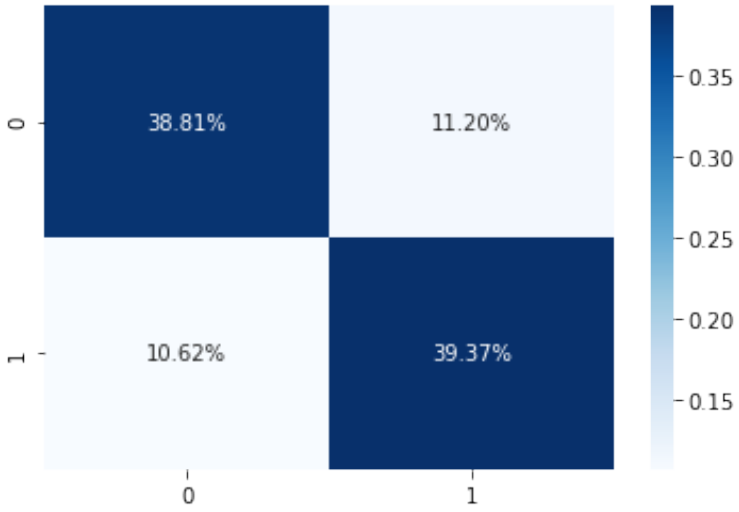
```
Modelo: KNN => 75.84% (Acuraria) - Modelo com dados Normalizados e Padronizados

 Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo CART


[[29954  7336]
 [ 8654 28626]]
```

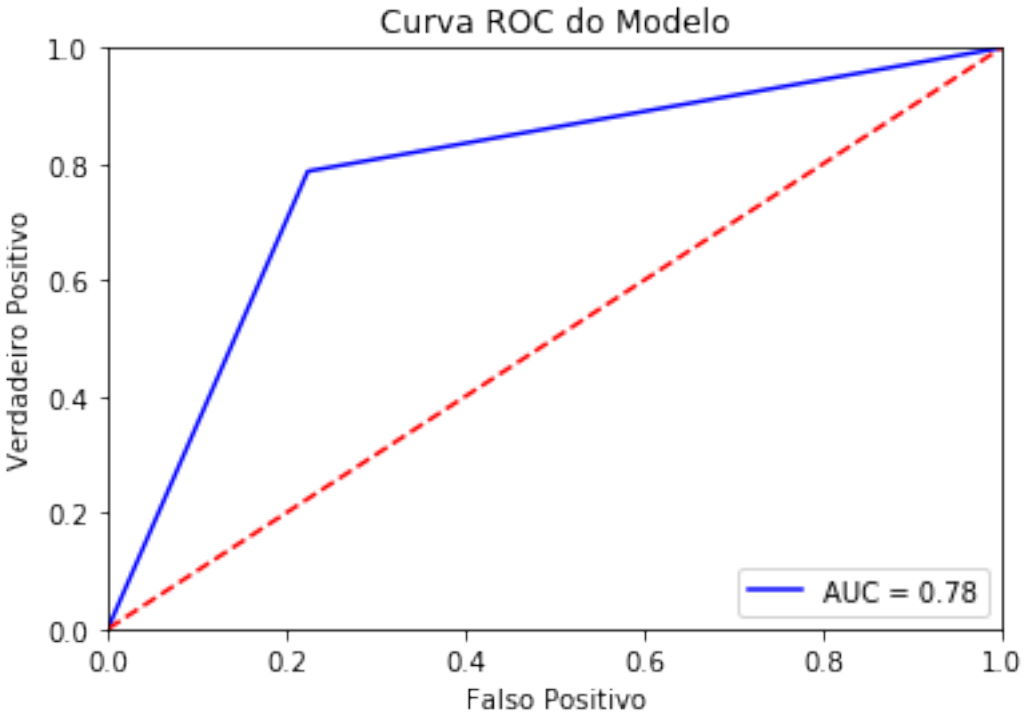




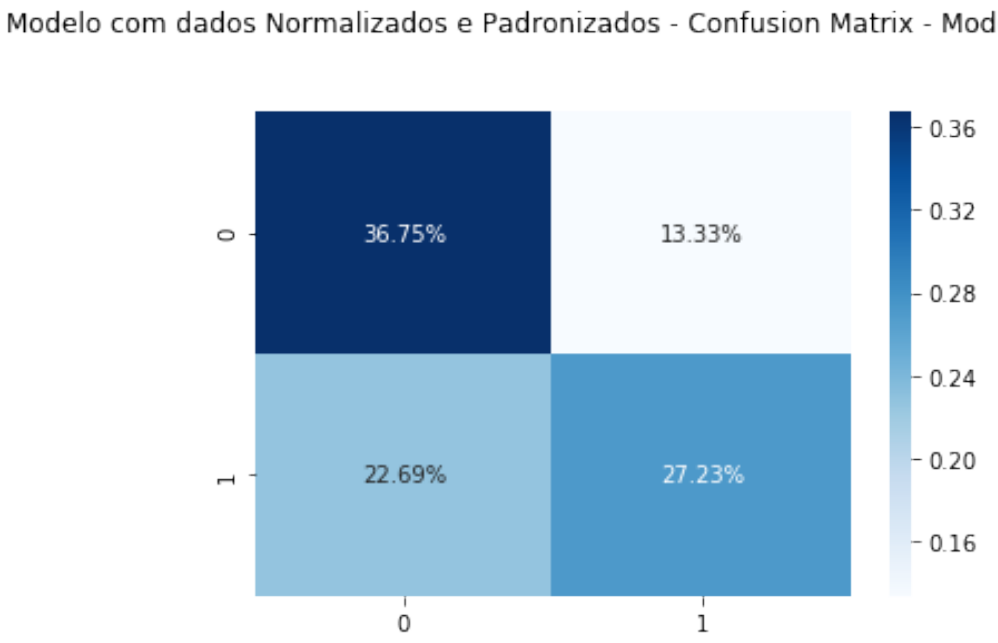

```
Modelo: CART => 78.60% (Acuraria) - Modelo com dados Normalizados e Padronizados

 Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo EXTREE


[[28941  8349]
 [ 7923 29357]]
```

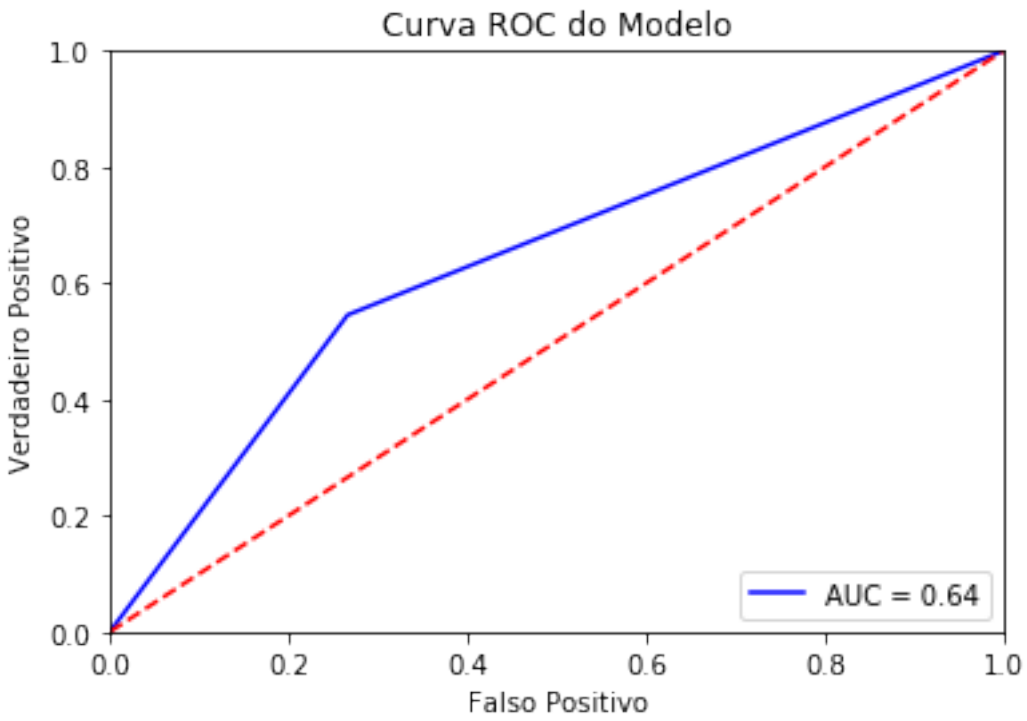
Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo EXTREE





```
Modelo: EXTREE => 78.28% (Acuraria) - Modelo com dados Normalizados e Padronizados

 Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo SVC


[[1370  497]
 [ 846 1015]]
```



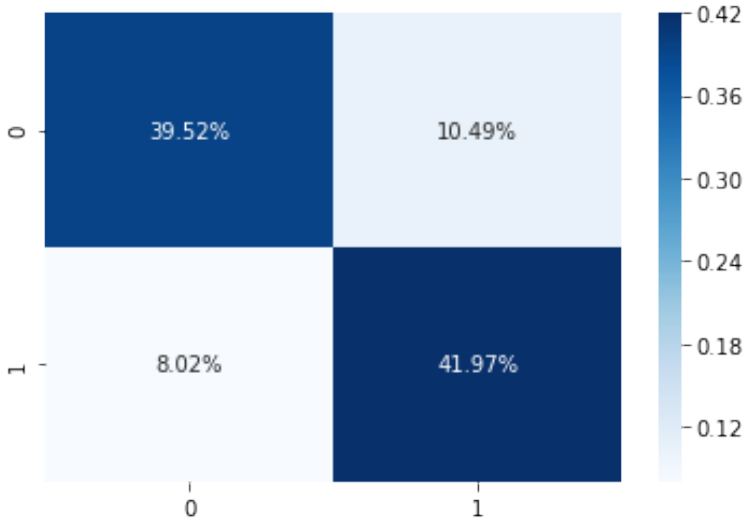


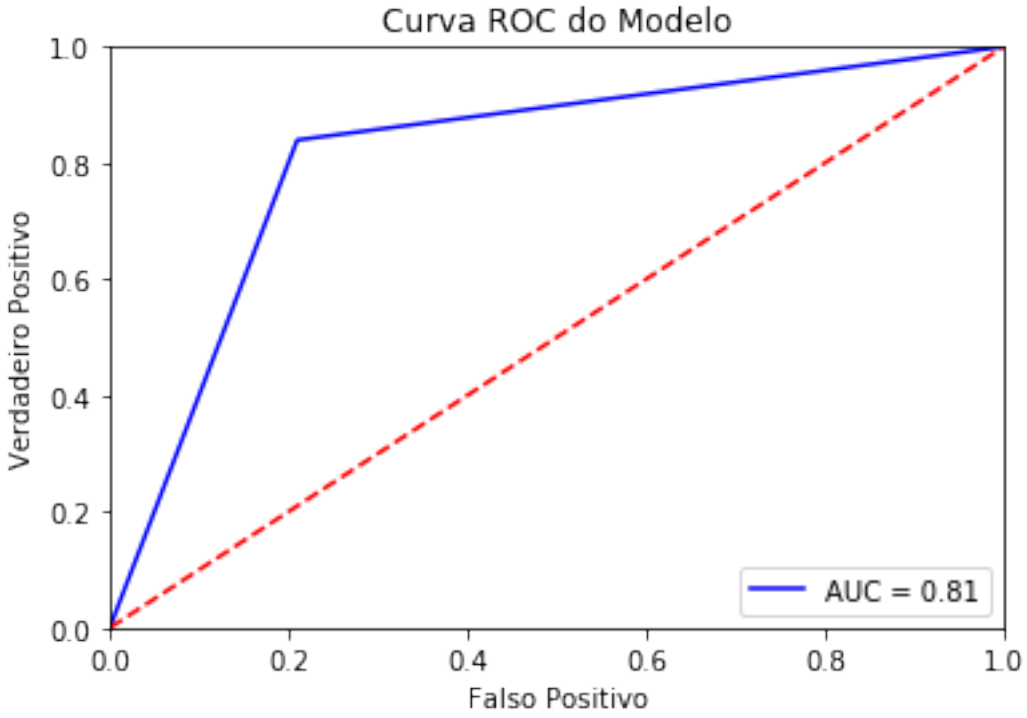
Modelo: SVC => 63.98% (Acuraria) - Modelo com dados Normalizados e Padronizados

Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo RAFOR

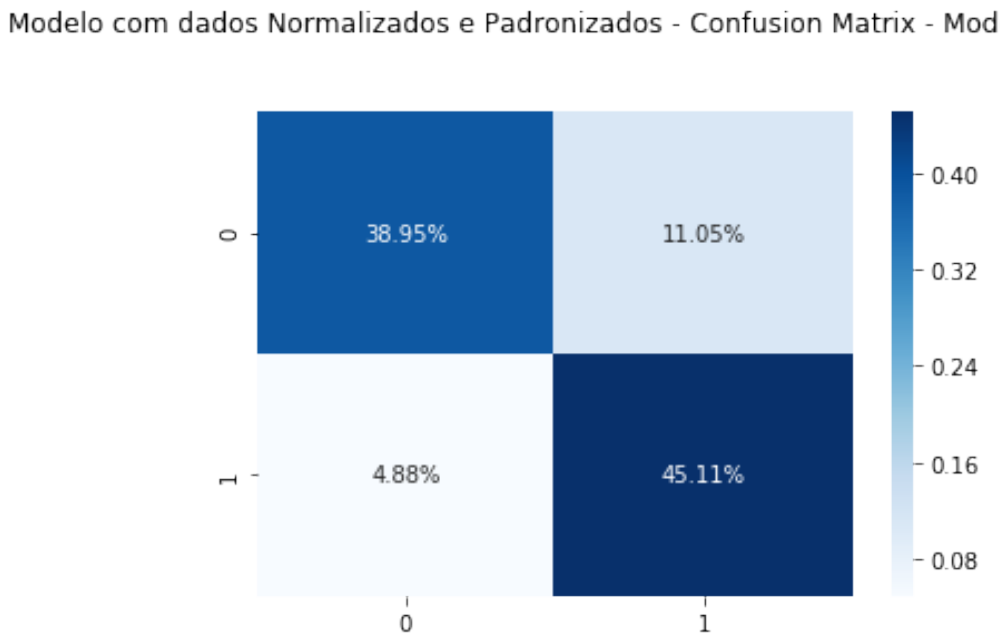
```
[[29467  7823]
 [ 5982 31298]]
```

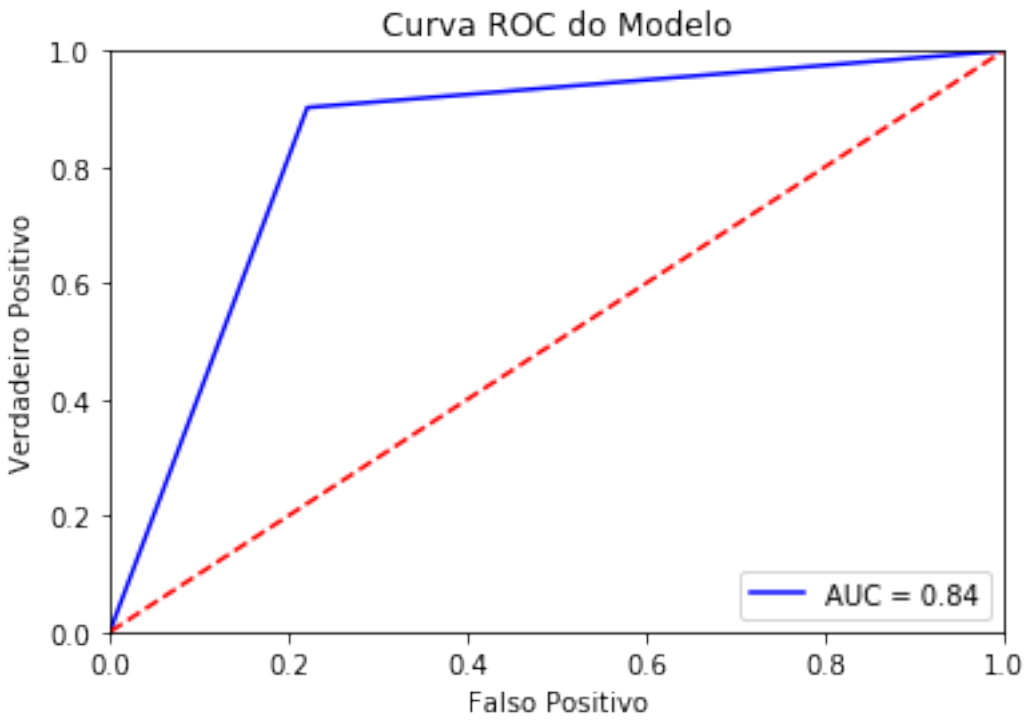
Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo RAFOR





```
Modelo: RAFOR => 81.49% (Acuraria) - Modelo com dados Normalizados e Padronizados

 Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo LGB


[[29047  8243]
 [ 3640 33640]]
```



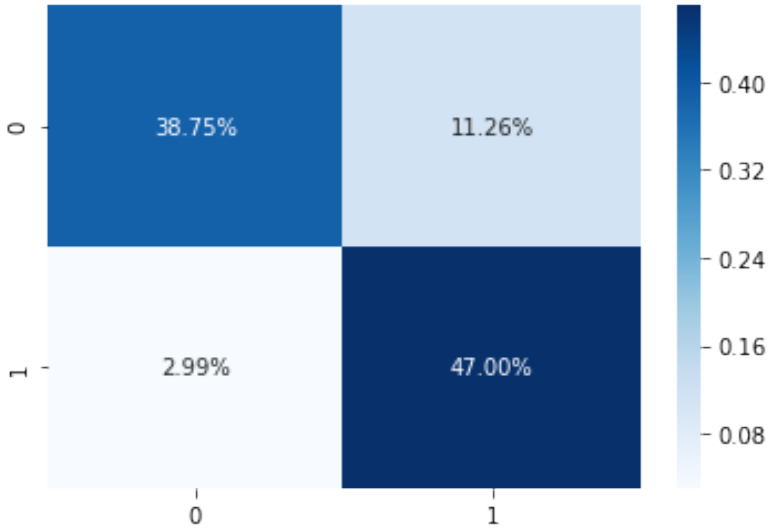


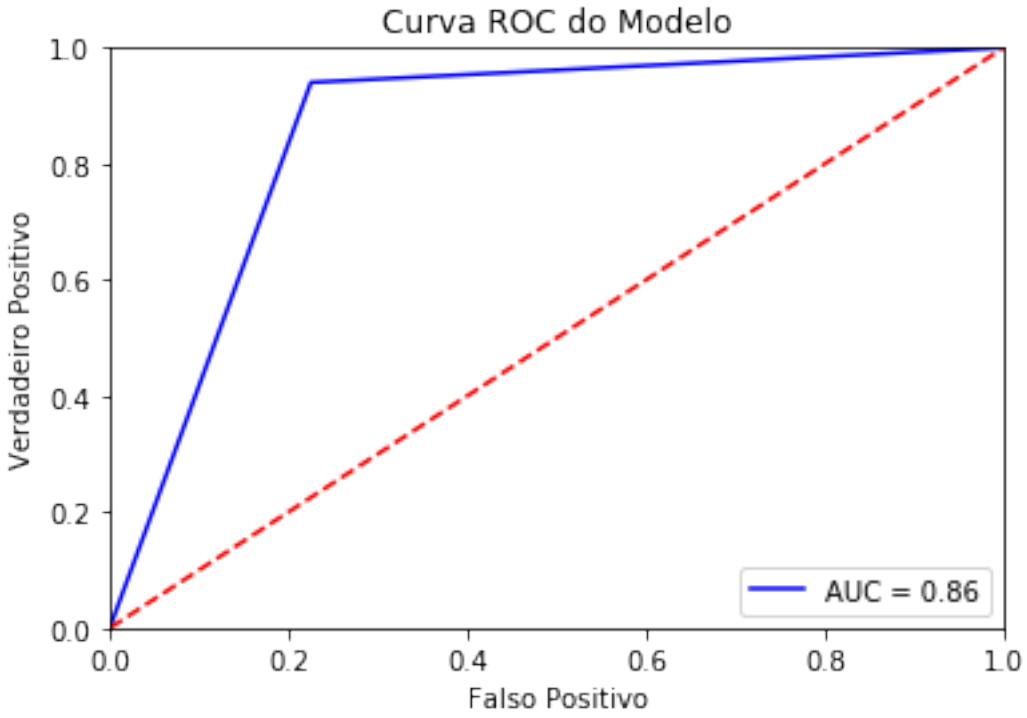
Modelo: LGB => 84.06% (Acuraria) - Modelo com dados Normalizados e Padronizados

Tabela - Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo XGB

```
[[28894  8396]
 [ 2231 35049]]
```

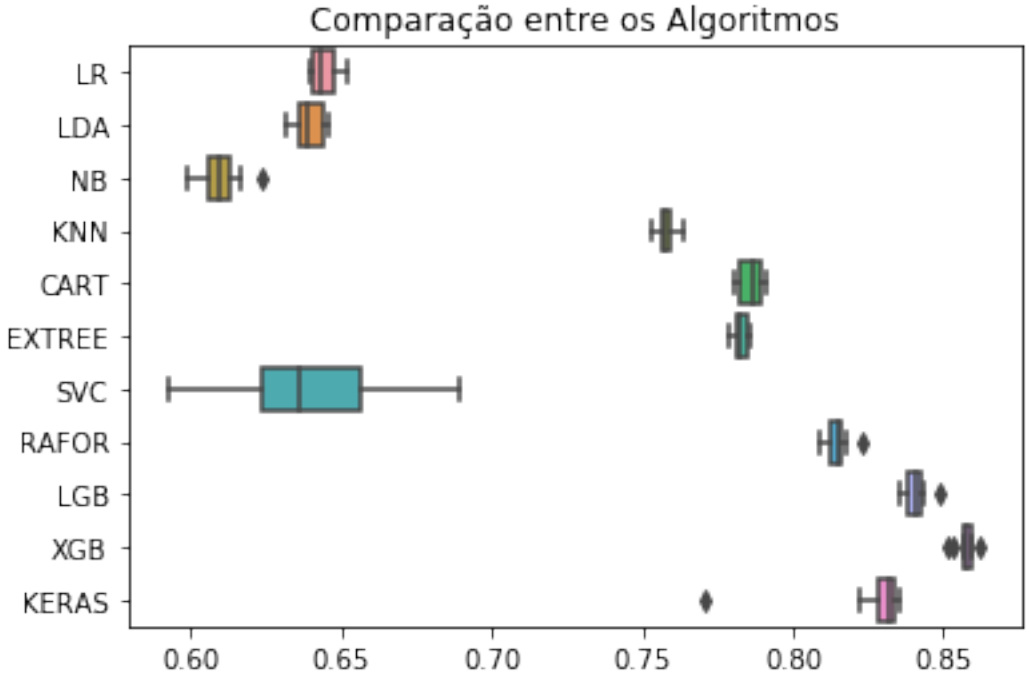
Modelo com dados Normalizados e Padronizados - Confusion Matrix - Modelo XGB





```
Modelo: XGB => 85.75% (Acuraria) - Modelo com dados Normalizados e Padronizados
```

```
Modelo: KERAS => 82.16% (Acuraria) - Modelo com dados Normalizados e Padronizados
```

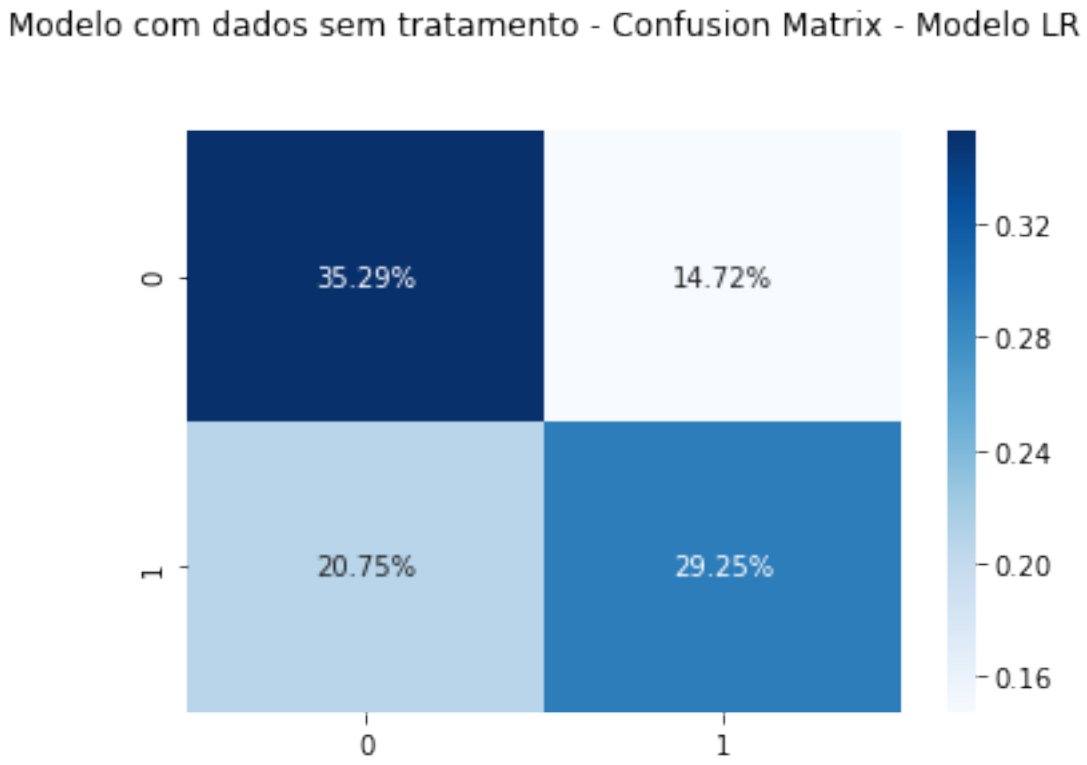


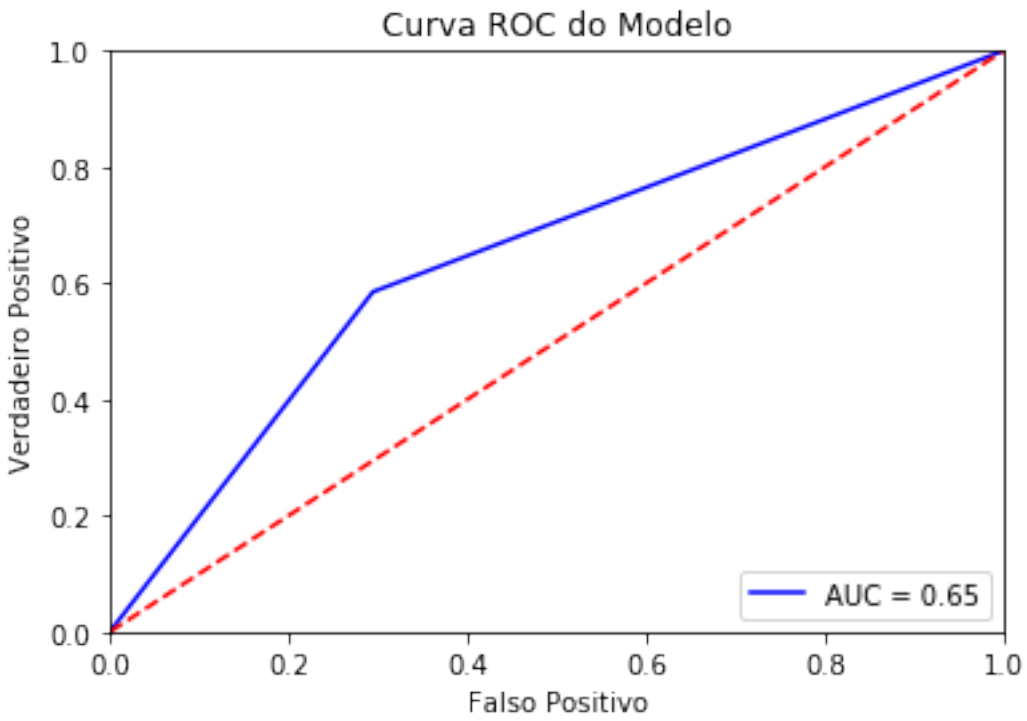

```
In [31]: # Modelos com os dados comuns, sem tratamento
```

```
In [47]: resultados1 = executar(X_comum, Y_comum, 'Modelo com dados sem tratamento', False)
```

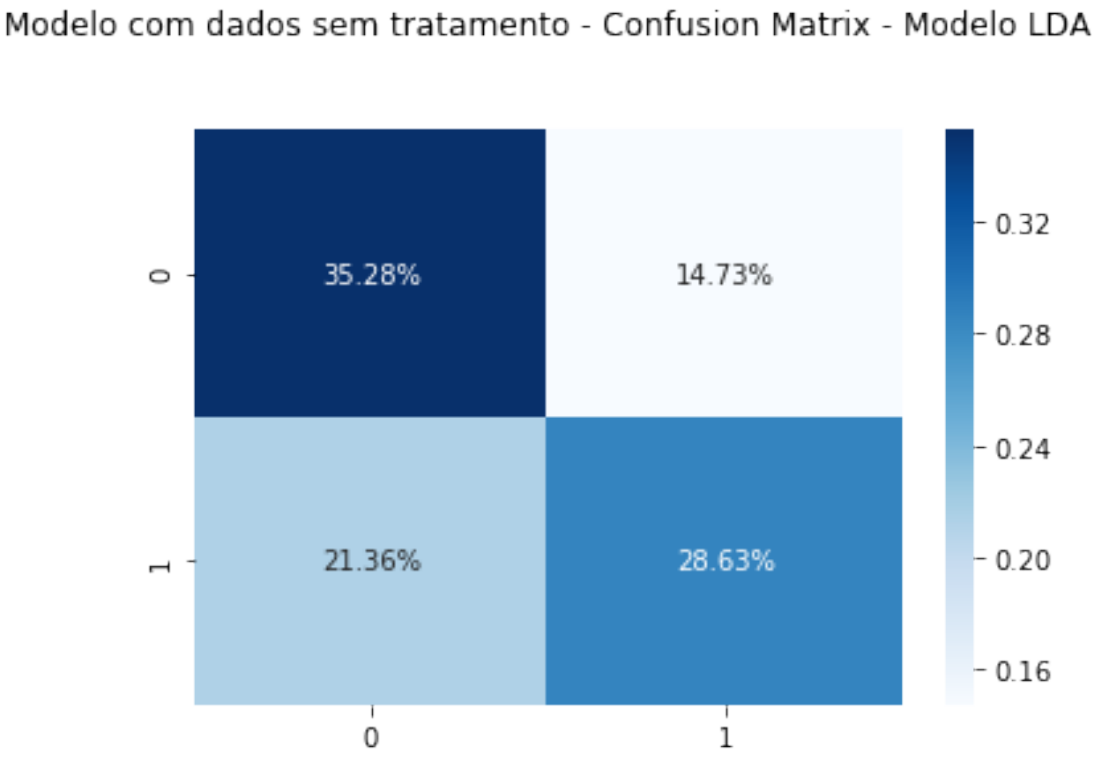
```
 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo LR
```

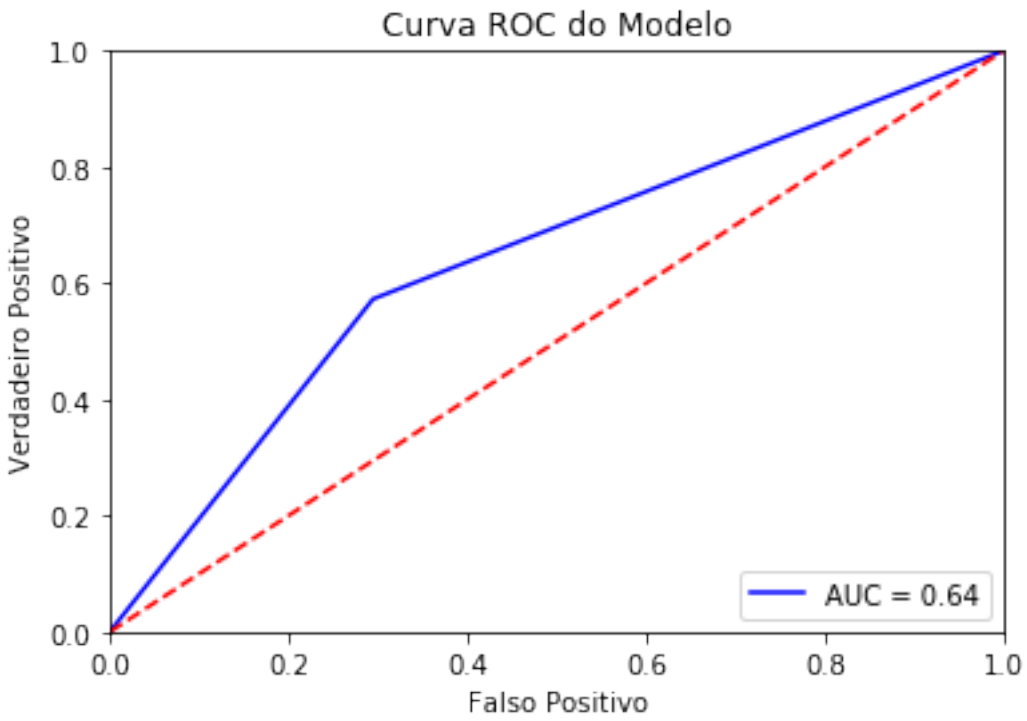
```
[[26313 10977]
 [15470 21810]]
```


Modelo com dados sem tratamento - Confusion Matrix - Modelo LR
35.29%
14.72%
20.75%
29.25%
0
1
0
1
0.32
0.28
0.24
0.20
0.16

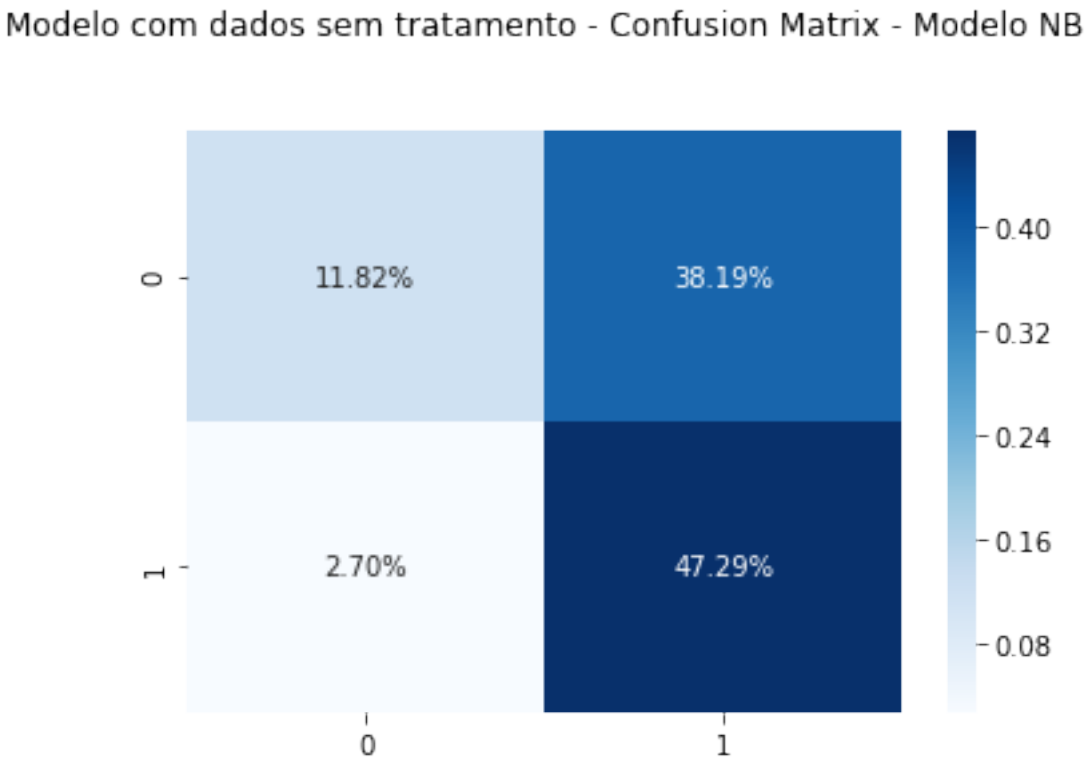


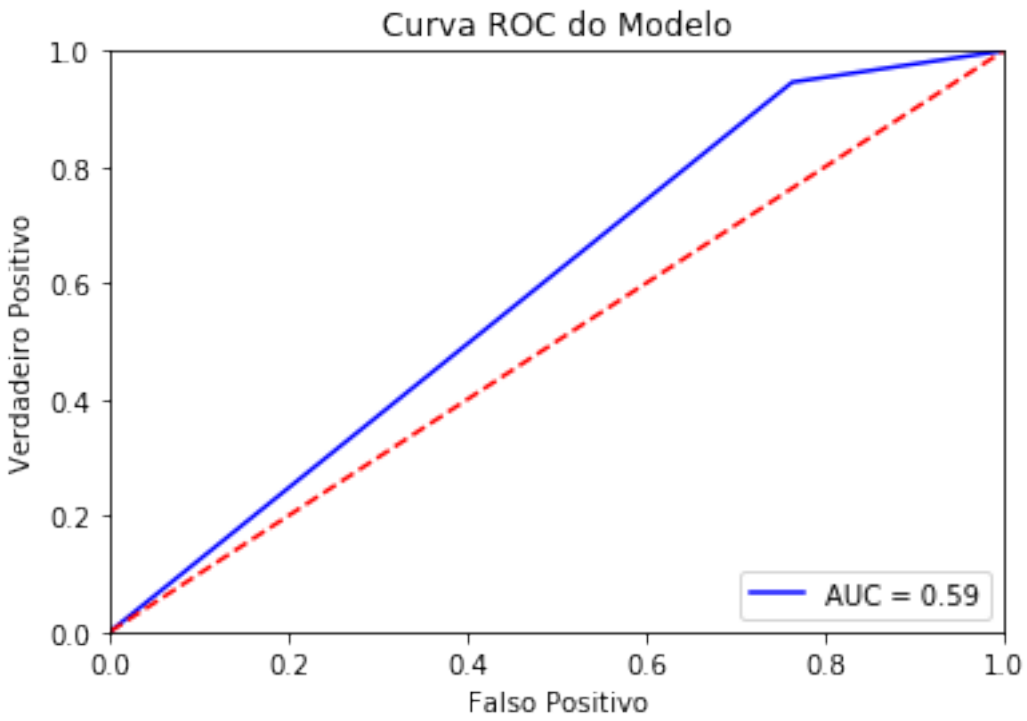

```
Modelo: LR => 64.53% (Acuraria) - Modelo com dados sem tratamento

 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo LDA


[[26309 10981]
 [15930 21350]]
```


Modelo com dados sem tratamento - Confusion Matrix - Modelo LDA
35.28%
14.73%
21.36%
28.63%
0
1
0
1
0.32
0.28
0.24
0.20
0.16

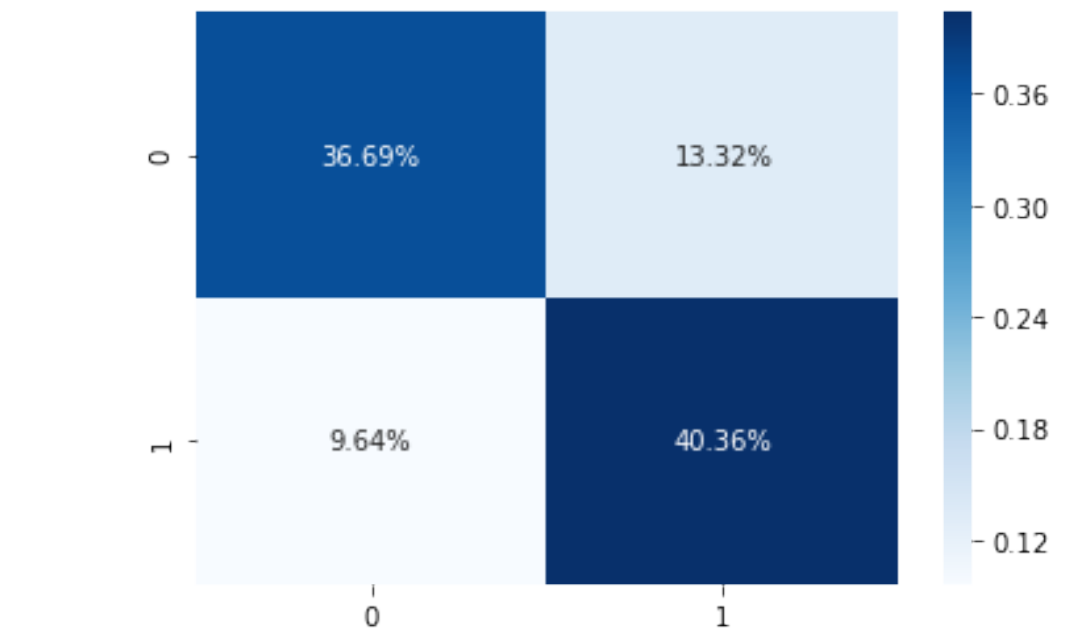


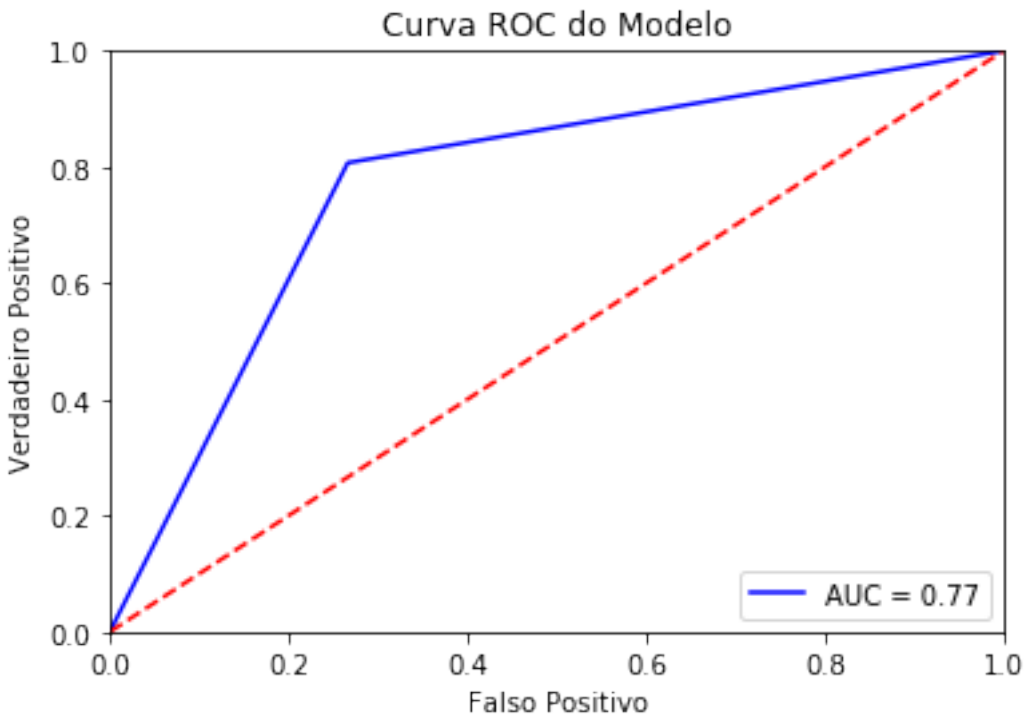

```
Modelo: LDA => 63.91% (Acuraria) - Modelo com dados sem tratamento

 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo NB


[[ 8814 28476]
 [ 2013 35267]]
```


Modelo com dados sem tratamento - Confusion Matrix - Modelo NB
11.82%
38.19%
2.70%
47.29%
0
1
0
1
0.40
0.32
0.24
0.16
0.08

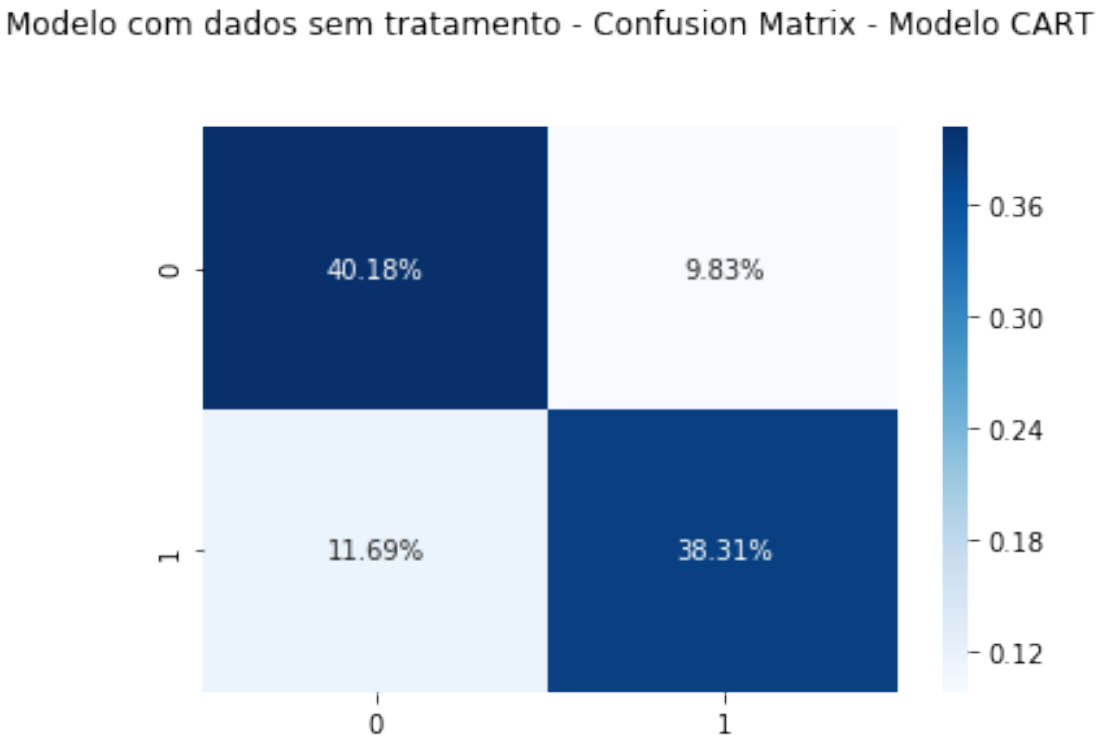


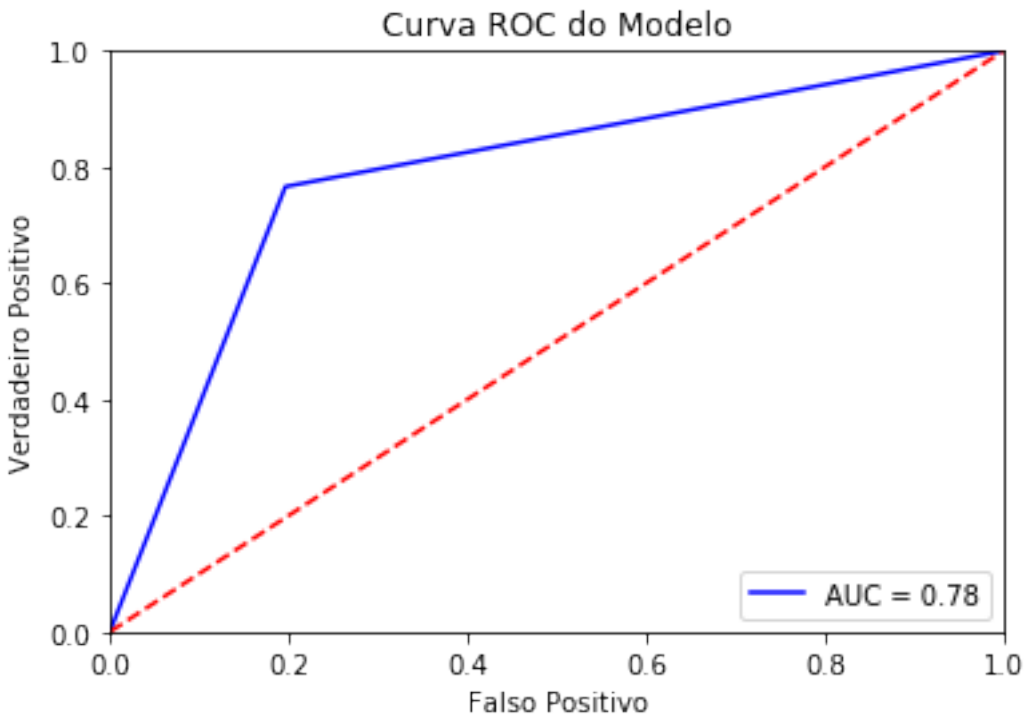

```
Modelo: NB => 59.11% (Acuraria) - Modelo com dados sem tratamento

 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo KNN


[[27357  9933]
 [ 7187 30093]]
```


Modelo com dados sem tratamento - Confusion Matrix - Modelo KNN
36.69%
13.32%
9.64%
40.36%
0
1
0
1
0.36
0.30
0.24
0.18
0.12

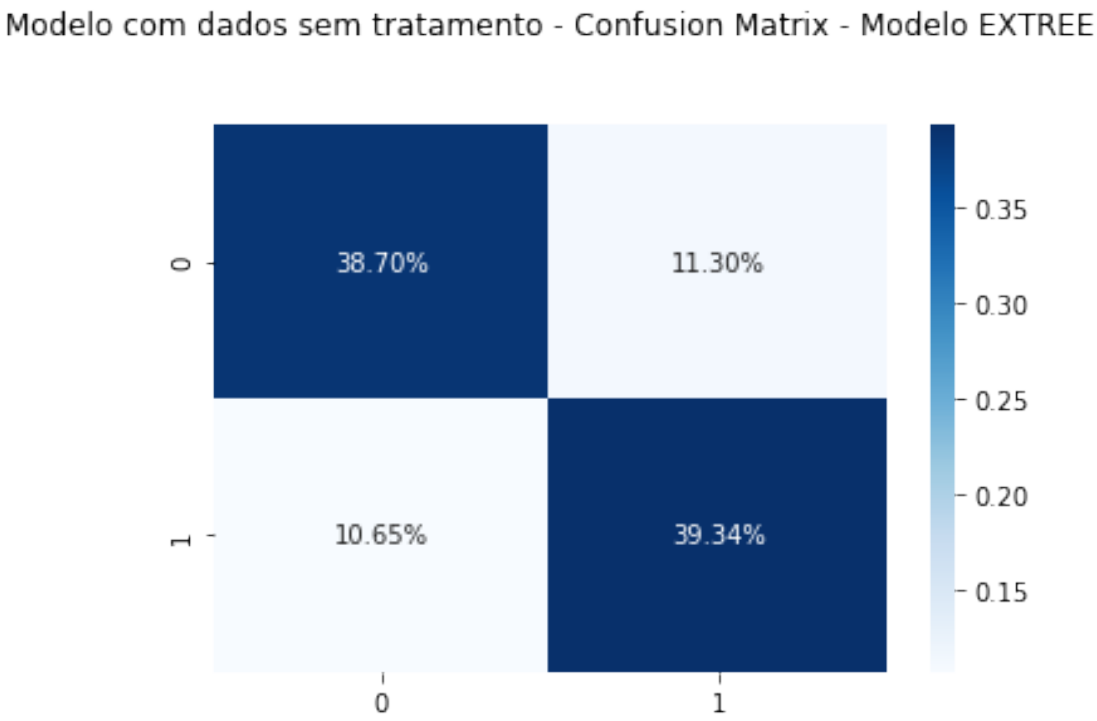


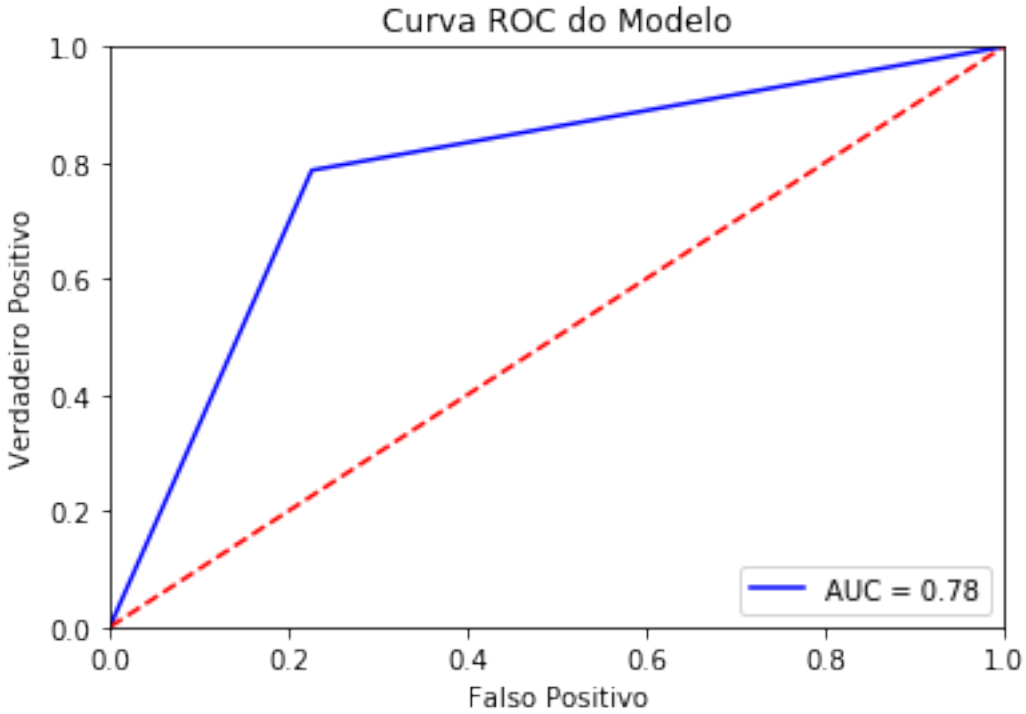

```
Modelo: KNN => 77.04% (Acuraria) - Modelo com dados sem tratamento

 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo CART


[[29959  7331]
 [ 8715 28565]]
```


Modelo com dados sem tratamento - Confusion Matrix - Modelo CART
40.18%
9.83%
11.69%
38.31%
0
1
0
1
0.36
0.30
0.24
0.18
0.12

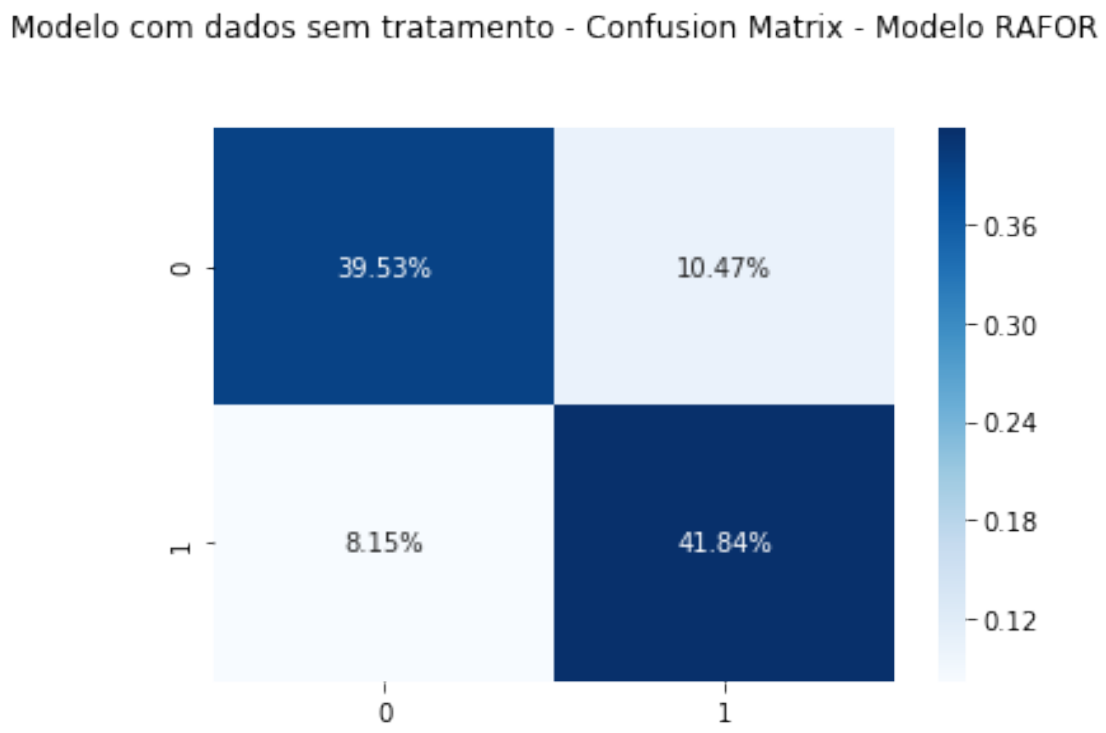


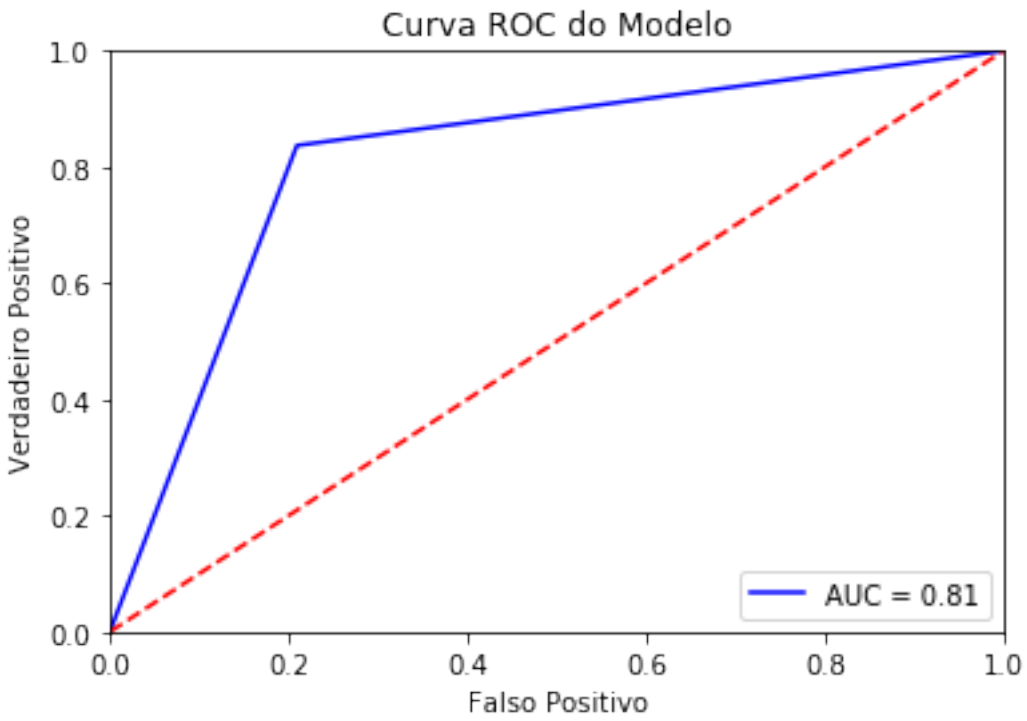

```
Modelo: CART => 78.54% (Acuraria) - Modelo com dados sem tratamento

 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo EXTREE


[[28861  8429]
 [ 7941 29339]]
```


Modelo com dados sem tratamento - Confusion Matrix - Modelo EXTREE
38.70%
11.30%
10.65%
39.34%
0
1
0
1
0.35
0.30
0.25
0.20
0.15

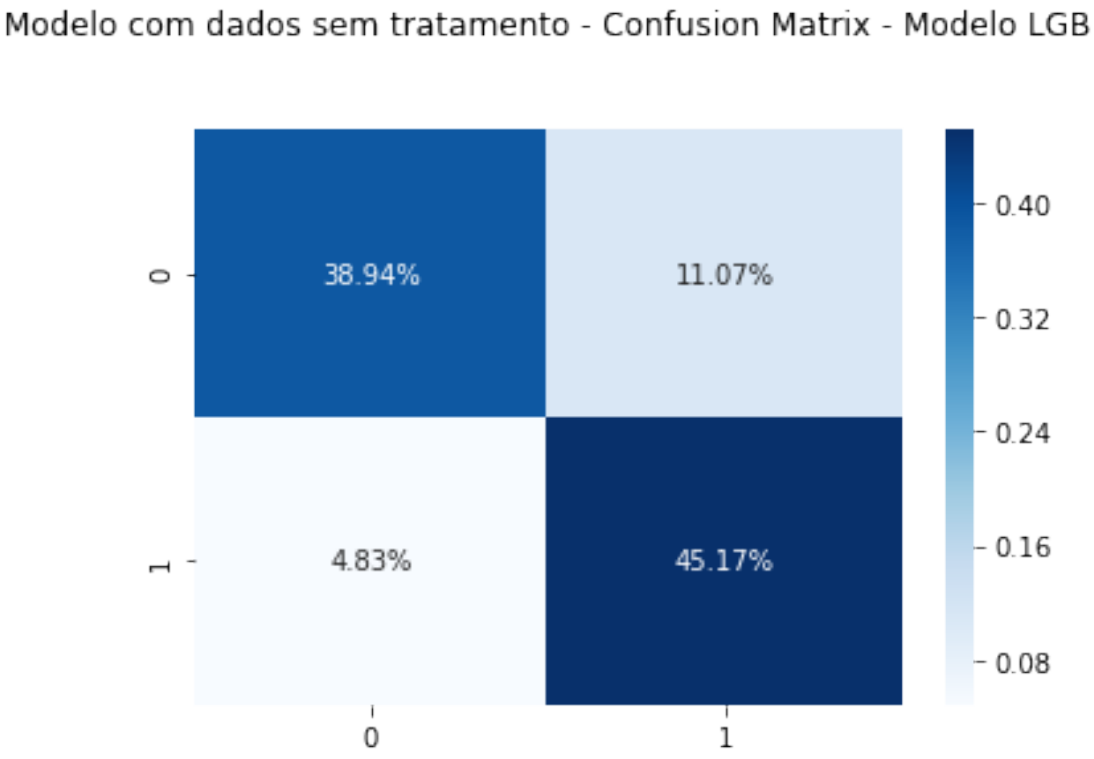

Curva ROC do Modelo
Verdadeiro Positivo
Falso Positivo
AUC = 0.78
0.0
0.2
0.4
0.6
0.8
1.0

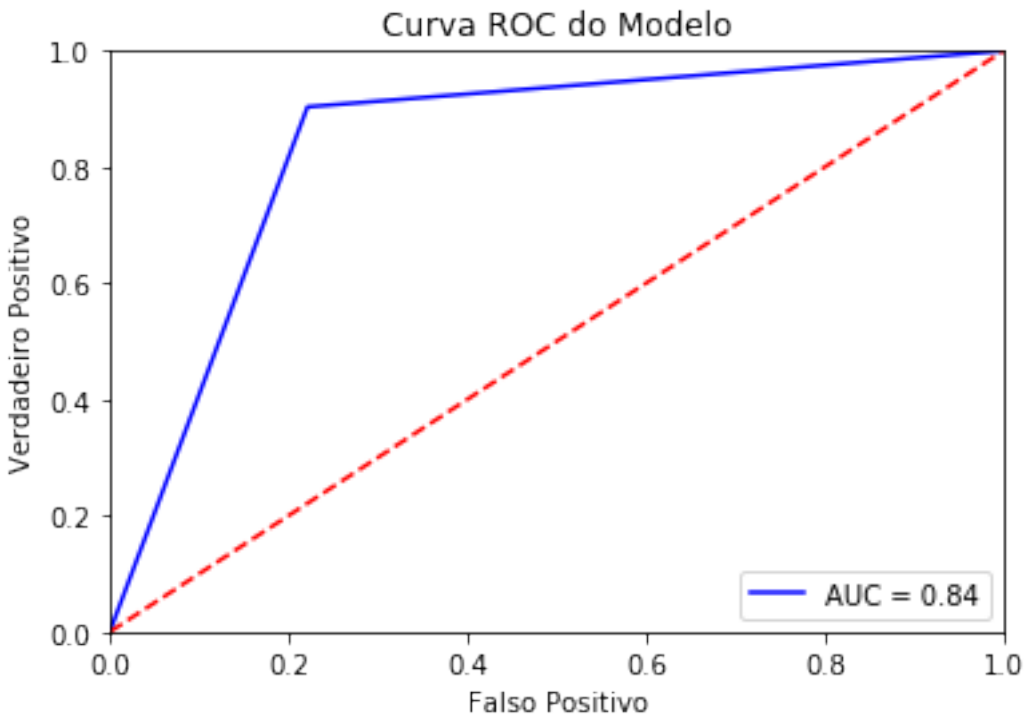
```
Modelo: EXTREE => 78.08% (Acuraria) - Modelo com dados sem tratamento


Modelo: SVC => 50.00% (Acuraria) - Modelo com dados sem tratamento

 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo RAFOR


[[29481  7809]
 [ 6077 31203]]
```



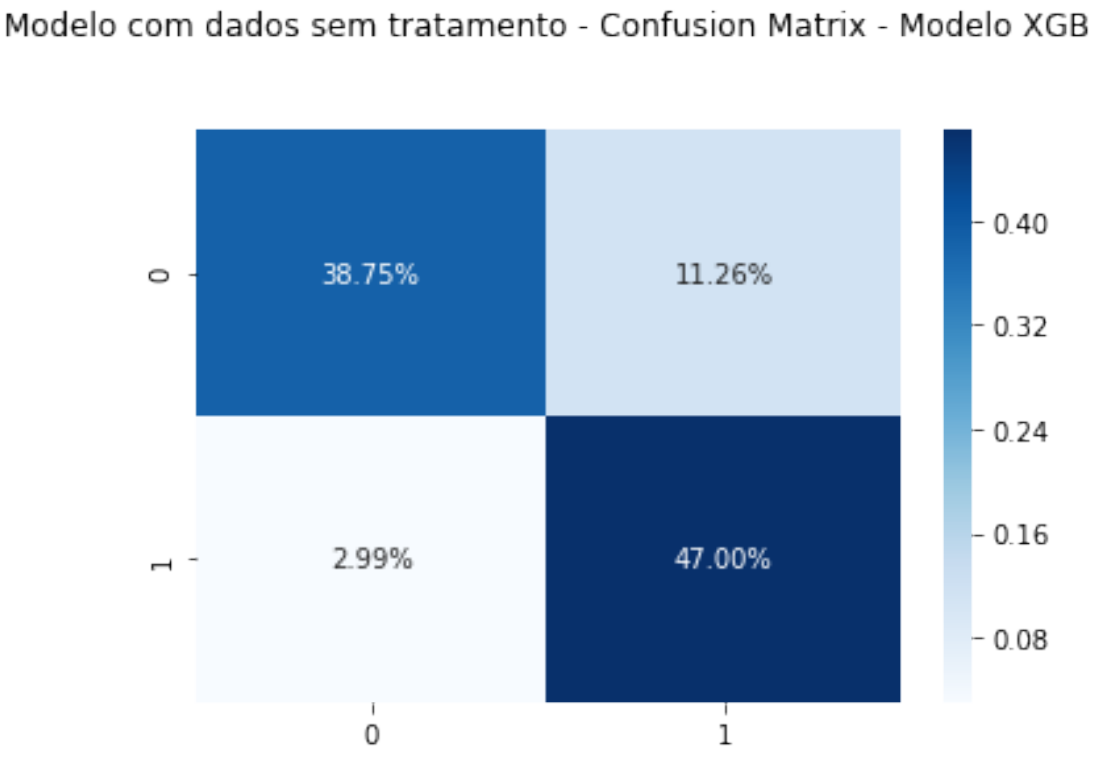


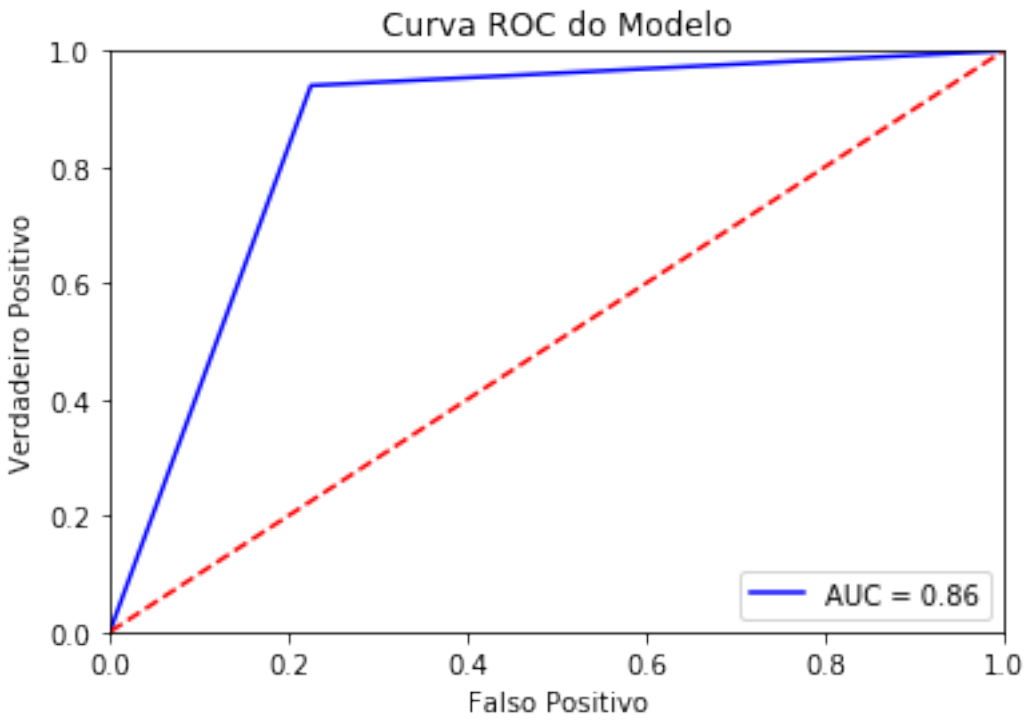

```
Modelo: RAFOR => 81.38% (Acuraria) - Modelo com dados sem tratamento

 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo LGB


[[29038  8252]
 [ 3600 33680]]
```


Modelo com dados sem tratamento - Confusion Matrix - Modelo LGB
38.94%
11.07%
4.83%
45.17%
0
1
0
1
0.40
0.32
0.24
0.16
0.08



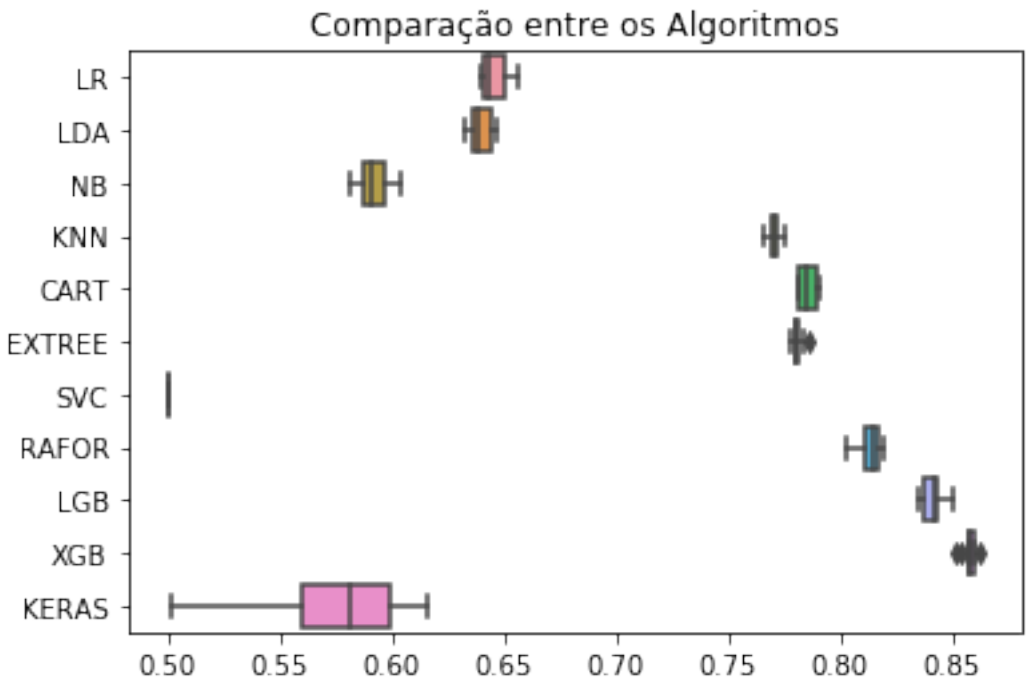

```
Modelo: LGB => 84.11% (Acuraria) - Modelo com dados sem tratamento

 Tabela - Modelo com dados sem tratamento - Confusion Matrix - Modelo XGB


[[28894  8396]
 [ 2231 35049]]
```


Modelo com dados sem tratamento - Confusion Matrix - Modelo XGB
38.75%
11.26%
2.99%
47.00%
0
1
0
1
0.40
0.32
0.24
0.16
0.08




```
Modelo: XGB => 85.75% (Acuraria) - Modelo com dados sem tratamento
```

```
Modelo: KERAS => 63.19% (Acuraria) - Modelo com dados sem tratamento
```



3.1 Comentário: Dados com Normalização tiveram melhor acurária em comparação com os não normalizados para a maioria dos algoritmos. O XGBClassifier apresenta o mesmo desempenho.

3.2 Modelo Escolhido => XGBClassifier - Modo Normalizado e Padronizado

3.2.1 Otimizando o Modelo para aumentar a acurácia.

```
In [33]: # Utilizando RandomizedSearchCV
         from sklearn.model_selection import RandomizedSearchCV
         from scipy import stats
         from scipy.stats import randint

         modelo_inicial = XGBClassifier(objective = 'binary:logistic')
         param_dist = {'n_estimators': stats.randint(500, 1500),
                       'learning_rate': stats.uniform(0.05, 0.20),
                       'subsample': stats.uniform(0.1, 0.9),
                       'max_depth': [3, 4, 5, 6, 7, 8, 9],
                       'min_child_weight': [1, 2, 3, 4]
                      }

         numFolds = 10
         kfold = KFold(n_splits = num_folds, random_state = seed, shuffle=True)

         rsearch = RandomizedSearchCV(modelo_inicial,
```

```
                                    param_distributions = param_dist,
                                    cv = kfold,
                                    n_iter = 5,
                                    scoring = 'roc_auc',
                                    error_score = 0,
                                    verbose = 3,
                                    n_jobs = -1)

        rsearch.fit(X, Y)

        # Print dos resultados
        print("Acurácia: %.2f" % (rsearch.best_score_ * 100))
        print("Melhores Parâmetros do Modelo:\n", rsearch.best_estimator_)
```

```
Fitting 10 folds for each of 5 candidates, totalling 50 fits


[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  24 tasks      | elapsed: 16.1min
[Parallel(n_jobs=-1)]: Done  50 out of  50 | elapsed: 22.0min finished


Acurácia: 91.01
Melhores Parâmetros do Modelo:
 XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.12221039053782455, max_delta_step=0, max_depth=4,
              min_child_weight=3, missing=None, n_estimators=744, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=0.9244632953849392, verbosity=1)
```

```
In [34]: #Gerando dados de Treino e de Teste para os modelos - 70% para o treino, 30% para Tes
         X_treino, X_teste, y_treino, y_teste = train_test_split(X, Y, test_size = 0.30)

         X_treino.shape, X_teste.shape, y_treino.shape, y_teste.shape
```

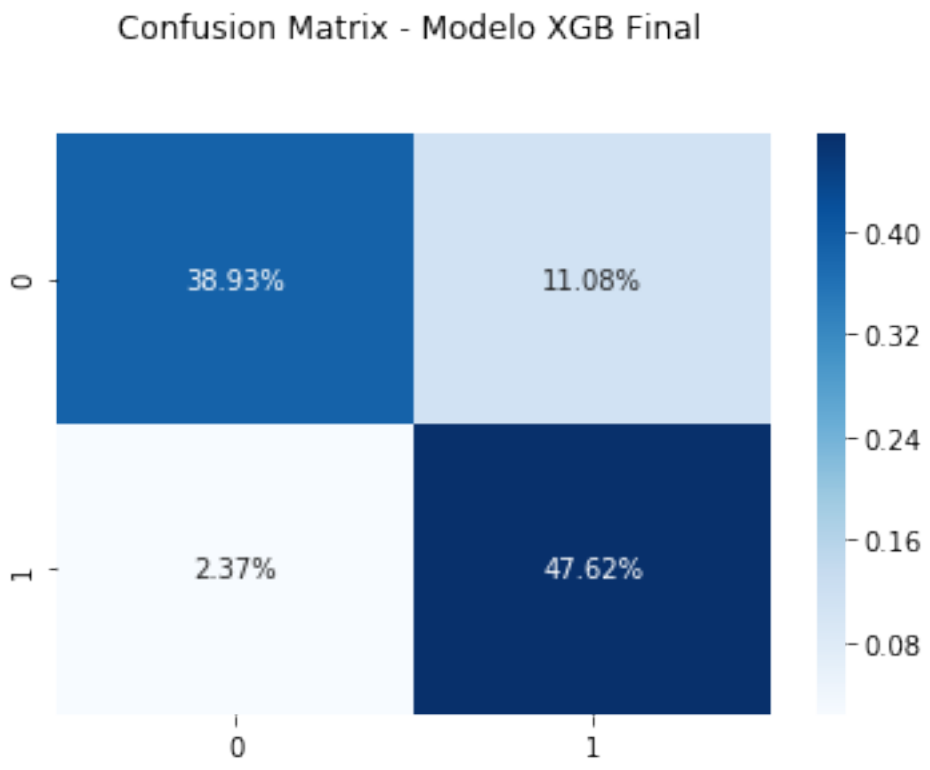
```
Out[34]: ((52199, 10), (22371, 10), (52199,), (22371,))
```

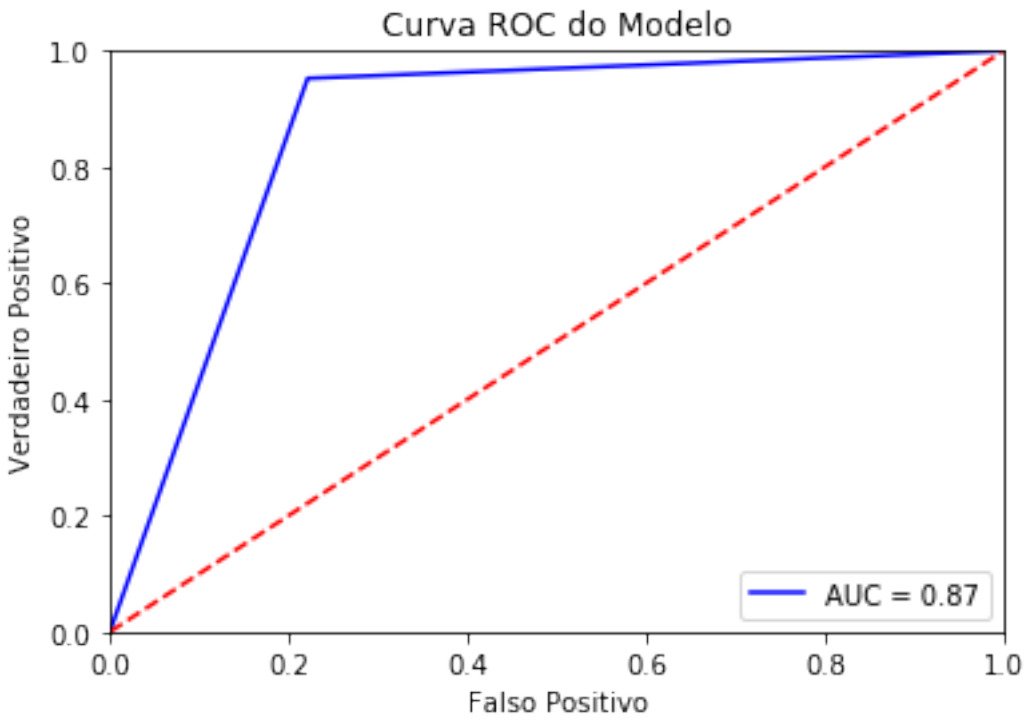
3.2.2 Gerando o modelo final com a otimização

```
In [35]: # Criando o modelo
         modelo_xgb_final = XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=
                            colsample_bynode=1, colsample_bytree=1, gamma=0,
                            learning_rate=0.11660371295014214, max_delta_step=0, max_depth=3,
                            min_child_weight=2, missing=None, n_estimators=735, n_jobs=1,
                            nthread=None, objective='binary:logistic', random_state=0,
                            reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
```

```
                      silent=None, subsample=0.6020533055981342, verbosity=1)

        kfold = KFold(n_splits = num_folds, random_state = seed, shuffle=True)
        cv_results = cross_val_predict(modelo_xgb_final, X, Y, cv = kfold)
        cv_results_score = cross_val_score(modelo_xgb_final, X, Y, cv = kfold, scoring = 'accu
        print("Modelo: %s => %.2f%s (Acuraria) - %s" % ('XGB Final', cv_results_score.mean() *

        #Confusion Matrix
        montarMatrixConfusion(Y, cv_results, "Confusion Matrix - Modelo XGB Final \n\n")
```

```
Modelo: XGB Final => 86.55% (Acuraria) - Modelo Otimizado

 Tabela - Confusion Matrix - Modelo XGB Final


[[29031  8259]
 [ 1770 35510]]
```





3.2.3 Otimização manual, baseada em minha experiência

```
In [36]: modelo_xgb_final1 = XGBClassifier(base_score=0.7, booster='gbtree', learning_rate=0.1

         # Treinando o model
         modelo_xgb_final1.fit(X_treino, y_treino)
         # Fazendo previsões
         y_pred_xgb = modelo_xgb_final1.predict(X_teste)
         previsoes_xgb = [round(value) for value in y_pred_xgb]

         # Avaliando as previsões
         accuracy = accuracy_score(y_teste, previsoes_xgb)
         print("Acurácia XGB Final - Otimização Manual: %.2f%%" % (accuracy * 100.0))
```

```
Acurácia XGB Final - Otimização Manual: 86.58%
```

```
In [37]: # Salvando o modelo
         joblib.dump(modelo_xgb_final1, 'modelo/modelo_xgb_final.joblib')
         print("Modelo salvo!")
```

```
Modelo salvo!
```

3.2.4 Previsões

Temos 10 casos reais num dataset separado para estimular e prever o Churn O Objetivo desse trecho é demonstrar o algoritmo funcionando e mostrando os resultados de Churn Sim ou Não e a probabilidade disso. Conforme objetivos específicos desse TCC.

```
In [38]: #Load do modelo - Churn
         modelo_churn = joblib.load('modelo/modelo_xgb_final.joblib')
         modelo_churn
```

```
Out[38]: XGBClassifier(base_score=0.7, booster='gbtree', colsample_bylevel=1,
                       colsample_bynode=1, colsample_bytree=1, gamma=0,
                       learning_rate=0.11, max_delta_step=0, max_depth=3,
                       min_child_weight=1, missing=nan, n_estimators=750, n_jobs=1,
                       nthread=None, objective='binary:logistic', random_state=0,
                       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                       silent=None, subsample=1, verbosity=1)
```

```
In [39]: #Carregando o dataset - Predicao
         df = pd.read_csv('dados/dataset_previsao.csv', sep=",")
         df.head(10)
```

```
Out[39]:    anosFidelidadeCliente  codigoEstadoCivil  sexoSegurado  tipoPessoaSegurado  \
         0                      3                  2             0                   1
         1                      5                  2             0                   1
         2                      4                  5             1                   1
         3                      3                  5             1                   1
         4                      2                  2             0                   1
         5                      4                  2             0                   1
         6                      0                  2             0                   1
         7                     12                  2             1                   1
         8                      2                  1             1                   1
         9                      4                  2             1                   1

            resgatePontos  numeroSegmentoCliente  codigoCategoriaTarifaria  \
         0              0                      1                        10
         1              0                      1                        23
         2              1                      1                        22
         3              0                      1                        10
         4              0                      1                        10
         5              0                      1                        10
         6              0                      1                        10
         7              1                      1                        10
         8              0                      1                        10
         9              0                      1                        10

            quantidadePortasVeiculo  codigoCombustivelVeiculo  codigoUsoVeiculo  ...  \
         0                        4                         8                 1  ...
         1                        5                         2                 1  ...
```

```
2                       5                       8                1  ...
3                       4                       8                1  ...
4                       5                       8                1  ...
5                       5                       8                1  ...
6                       4                       2                1  ...
7                       5                       8                1  ...
8                       3                       8                1  ...
9                       5                       8                1  ...

   I_ESCOLARIDADE  valorPremioPagoAtual  correntista  faixaIdade  \
0         647.000               1300.18            0           2
1         365.000               6506.66            0           2
2         365.000               2120.00            1           2
3         251.000               1643.34            1           2
4         629.000               1217.53            0           2
5         647.000                620.29            0           3
6           0.487                897.45            0           3
7         273.000                846.21            1           2
8         326.000                833.20            0           2
9         419.000                881.31            1           2

   faixaPrecoLiquido  faixaPrecoFranquia  faixaPrecoPremioFinal  \
0                  2                   2                      2
1                  3                   3                      3
2                  3                   3                      1
3                  2                   2                      2
4                  2                   2                      2
5                  1                   1                      1
6                  1                   3                      2
7                  1                   2                      2
8                  1                   2                      1
9                  1                   1                      1

   diferencaPremioAntNovo  faixaDiferencaValores  gerouSinistro
0                   76.61                      2              0
1                  171.14                      3              0
2                 -677.50                      1              0
3                 -151.24                      1              0
4                   -7.94                      2              0
5                 -123.64                      1              0
6                  -25.02                      2              0
7                  -47.01                      2              0
8                  -95.13                      2              0
9                   23.19                      2              0

[10 rows x 47 columns]
```

In [40]: *# Tratamento de Dados - Pré-Processamento do Modelo*

```
        df = df.drop(['valorDiferencaPremioAnual',
         'IDHM_E',
         'IDHM_L',
         'I_ESCOLARIDADE',
         'gerouSinistro',
        'numeroSegmentoCliente',
        'tipoPessoaSegurado',
        'estacaoAno'], axis=1)
```

In [41]:
```
        #Pegando as colunas no Features Selection
        mask = fs.get_support()
        colunas = df.columns
        new_features = []
        for bool, feature in zip(mask, colunas):
            if bool:
                new_features.append(feature)

        df_selection = pd.DataFrame(df, columns=new_features)

        # Gerando a nova escala (normalizando os dados)
        scaler = MinMaxScaler(feature_range = (0, 1))
        rescaledX = scaler.fit_transform(df_selection.values)

        #Gerando padronização dos valores
        scalerP = StandardScaler().fit(rescaledX)
        standardX = scalerP.transform(rescaledX)
```

In [42]:
```
        # Previsões
        Y_churn_prob = modelo_churn.predict_proba(standardX)
        Y_churn = modelo_churn.predict(standardX)
```

In [48]:
```
        df_resultado = df_selection
        df_resultado['Churn'] = Y_churn
        df_resultado['Probabilidade_NoChurn'] = (Y_churn_prob[:,1] * 100)
        df_resultado['Probabilidade_Churn'] = (Y_churn_prob[:,0] * 100)

        df_resultado['Churn'] = ['Sim' if x == 0.0 else  'Nao' for x in df_resultado['Churn']]
```

In [49]:
```
        df_resultado[['Churn','Probabilidade_NoChurn','Probabilidade_Churn']].head(10)
```

Out[49]:

	Churn	Probabilidade_NoChurn	Probabilidade_Churn
0	Nao	55.099304	44.900696
1	Sim	19.033230	80.966774
2	Nao	79.348831	20.651167
3	Nao	84.593994	15.406007
4	Nao	72.657394	27.342606
5	Sim	43.794579	56.205421
6	Nao	88.420441	11.579561
7	Nao	90.939590	9.060407

```
8   Nao              87.992477            12.007523
9   Sim              49.968468            50.031532
```

```
In [45]: #Salvando o trabalho
         df_resultado[['Churn','Probabilidade_NoChurn','Probabilidade_Churn']].to_excel('result
```

```
In [ ]:
```

3.3 FIM

3.4 OBRIGADO

```
In [ ]:
```