analiseFraude_modelo

January 26, 2020

0.0.1 Análise de Fraude - 01/2020

0.0.2 MARCIO DE LIMA

0.0.3 DADOS Fornecidos - desafio_fraude.csv

0.0.4 ARQUIVO CSV ANONIMIZADO com 31 colunas.

0.0.5 Estrutura do arquivo

Ocorrencia float64
PP1 float64
PP2 float64
PP3 float64
PP4 float64
PP5 float64
PP6 float64
PP7 float64
PP8 float64
PP9 float64
PP10 float64
PP11 float64
PP12 float64
PP13 float64
PP14 float64
PP15 float64
PP16 float64
PP17 float64
PP18 float64
PP19 float64
PP20 float64
PP21 float64
PP22 float64
PP23 float64
PP24 float64
PP25 float64
PP26 float64
PP27 float64
PP28 float64

Sacado float64
Fraude int64 => VARIAVEL TARGET (0 => OK, 1 => Fraude)
Comentário: Não foi fornecido um dicionário de dados nem os nomes das colunas perante ao negócio, provavelmente devido a anonimização, mas isso prejudica e dificulta a análise estatística.
Comentário: Não foi fornecido no arquivo algum campo de data, dados de sexo, faixa de idade, localidade, faixa de renda, faixa de score Serasa, etc.; que seriam interessantes para a análise e para melhorar a deteção de fraude, na minha opinião.

```
In [1]: # Importando as bibliotecas
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import warnings

        %matplotlib inline
        warnings.filterwarnings("ignore")
```

```
In [120]: # Importando o arquivo
          df = pd.read_csv('desafio_fraude.csv')
```

```
In [121]: #df.dtypes
          df.head(5)
```

```
Out[121]:    Ocorrencia       PP1       PP2       PP3       PP4       PP5       PP6  \
          0    -44299.0 -1.239996  0.985194 -1.005080  0.251323  0.872854 -1.677811
          1    -44300.0 -0.472690  1.869177 -0.277741  1.122846  1.526166  0.262325
          2    -44301.0  0.277314  3.455314 -0.722444 -0.428284  2.512025 -0.540760
          3    -44301.0 -1.061770 -0.105481 -0.226711 -0.929524 -0.100625 -0.300173
          4    -44302.0  4.622715  2.621667  0.872085  0.374010  1.456021 -1.531875

                  PP7       PP8       PP9  ...      PP21      PP22      PP23      PP24  \
          0  1.451311 -0.478908 -0.009459  ...  0.387768  0.286200  0.128686  1.280392
          1  0.242333 -0.006108 -1.659659  ... -0.387745 -0.434629  0.512801 -0.110994
          2  0.345111 -0.013655 -0.233508  ... -0.630255 -0.388096  0.697177 -0.523084
          3  0.029912 -0.205934  0.233190  ... -0.147422 -0.426827  0.070413  0.283090
          4 -0.162837 -1.331547 -0.340639  ...  0.221196  0.804017  1.309062  1.505088

                 PP25      PP26      PP27      PP28  Sacado  Fraude
          0 -0.301116 -0.673309 -0.069611 -0.009597  -28.38       0
          1 -0.350975 -0.073826  0.035071 -0.080140 -407.00       0
          2 -0.069830  0.196482  0.052145 -0.166683 -800.00       0
          3 -0.487739  0.288220 -0.035644 -0.007305  -31.28       0
          4  0.260178 -0.861611 -0.130562  1.023781 -522.16       0

          [5 rows x 31 columns]
```

0.1 Análise Exploratória

In [12]:
```
# Mostrando os dados
df.shape
```

Out[12]: (150000, 31)

In [13]:
```
# Mostrando as estruturas do Dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150000 entries, 0 to 149999
Data columns (total 31 columns):
Ocorrencia    150000 non-null float64
PP1           150000 non-null float64
PP2           150000 non-null float64
PP3           150000 non-null float64
PP4           150000 non-null float64
PP5           150000 non-null float64
PP6           150000 non-null float64
PP7           150000 non-null float64
PP8           150000 non-null float64
PP9           150000 non-null float64
PP10          150000 non-null float64
PP11          150000 non-null float64
PP12          150000 non-null float64
PP13          150000 non-null float64
PP14          150000 non-null float64
PP15          150000 non-null float64
PP16          150000 non-null float64
PP17          150000 non-null float64
PP18          150000 non-null float64
PP19          150000 non-null float64
PP20          150000 non-null float64
PP21          150000 non-null float64
PP22          150000 non-null float64
PP23          150000 non-null float64
PP24          150000 non-null float64
PP25          150000 non-null float64
PP26          150000 non-null float64
PP27          150000 non-null float64
PP28          150000 non-null float64
Sacado        150000 non-null float64
Fraude        150000 non-null int64
dtypes: float64(30), int64(1)
memory usage: 35.5 MB
```

In [38]:
```
# Dados Estatisticos - Analise descritiva das colunas Numéricos
# Arquivo com 31 colunas, todas numéricas, total de Registros: 150.000 linhas
```

```
# Variavel Target possue o 0 e 1, mas só olhando a média já vemos que temos poucos re
# o que é característico em bases de análise de fraude (Fraud Analytics). Veremos mai
# Possíveis outlier na coluna Sacado.
# Desvio padrão quase o mesmo para todas as colunas.

df.describe()
```

Out[38]:

```
         Ocorrencia            PP1            PP2            PP3  \
count  150000.000000  150000.000000  150000.000000  150000.000000
mean   -84550.214580       0.058999      -0.000790      -0.192183
std     27710.748503       1.894453       1.623712       1.406053
min   -133236.000000      -2.454930     -22.057729      -9.382558
25%   -115169.000000      -1.243456      -0.802149      -1.138473
50%    -77502.500000       0.042647      -0.082193      -0.359076
75%    -61713.750000       0.952018       0.588600       0.555060
max    -44299.000000      36.802320      63.344698      33.680984

                 PP4            PP5            PP6            PP7  \
count  150000.000000  150000.000000  150000.000000  150000.000000
mean       -0.037416       0.061588      -0.025715       0.026695
std         1.397615       1.341265       1.310820       1.194923
min       -16.875344     -32.911462     -21.307738     -31.527244
25%        -0.812624      -0.526469      -0.424574      -0.527260
50%        -0.039549       0.124219       0.245177      -0.013129
75%         0.816575       0.751890       0.734024       0.564334
max         5.683171      31.356750      21.929312      43.557242

                 PP8            PP9  ...           PP21           PP22  \
count  150000.000000  150000.000000  ...  150000.000000  150000.000000
mean       -0.004257       0.028148  ...       0.009957       0.027398
std         1.205874       1.106154  ...       0.739429       0.707714
min       -16.635979     -15.594995  ...     -27.202839     -10.503090
25%        -0.340863      -0.565387  ...      -0.165038      -0.466423
50%        -0.037083       0.095975  ...       0.033794       0.014600
75%         0.193112       0.678488  ...       0.225362       0.540801
max        73.216718      13.434066  ...      34.830382      10.933144

                PP23           PP24           PP25           PP26  \
count  150000.000000  150000.000000  150000.000000  150000.000000
mean        0.007275      -0.002739      -0.035211      -0.001127
std         0.622620       0.606964       0.506130       0.483787
min       -19.002942      -4.022866      -7.519589      -3.220178
25%        -0.128298      -0.431560      -0.369398      -0.247606
50%         0.020008      -0.049357      -0.071030       0.057265
75%         0.164620       0.348762       0.274183       0.331361
max        44.807735       2.824849      10.295397       2.604551

                PP27           PP28         Sacado         Fraude
```

```
count   150000.000000  150000.000000  150000.000000  150000.000000
mean        -0.000535      -0.001028     -88.602261       0.001580
std          0.397662       0.307684     247.302373       0.039718
min        -12.152401     -22.620072  -19656.530000       0.000000
25%         -0.090965      -0.078861     -77.662500       0.000000
50%         -0.004792      -0.016759     -22.040000       0.000000
75%          0.068544       0.048427      -5.410000       0.000000
max         22.565679      11.710896      -0.000000       1.000000

[8 rows x 31 columns]
```

In [122]:
```python
#Checando valores NA nos dados
df.isna().any()[lambda x: x]
```

Out[122]:
```
Series([], dtype: bool)
```

In [123]:
```python
#Checando valores Null nos dados
df.isnull().any()[lambda x: x]
```

Out[123]:
```
Series([], dtype: bool)
```

Comentário: Dados sem valores NA e sem valores NULL

In [124]:
```python
# Distribuição da variável TARGET
df.groupby('Fraude').size()
```

Out[124]:
```
Fraude
0    149763
1       237
dtype: int64
```

Comentário: Conforme números acima, temos somente 0,15% de registros de Fraude. Desta forma, os dados estão totalmente desbalanceados, isso é um problema, pois o modelo de ML tenderá a ter alta acurácia somente do campo Não Fraude.

In [125]:
```python
# Verificando dados duplicados no DataSet

duplicateRowsDF = df[df.duplicated(keep='last')]
print("Linhas duplicadas: ")
print(duplicateRowsDF)

duplicateRowsDF_col = df[df.duplicated(['Ocorrencia'])]
print("Linhas duplicadas por Ocorrencia somente: ")
print(duplicateRowsDF_col)

duplicateRowsDF_col_target = df[df.duplicated(['Ocorrencia', 'Fraude'])]
print("Linhas duplicadas por Ocorrencia e Fraude somente: ")
print(duplicateRowsDF_col_target)
```

```
Linhas duplicadas:
        Ocorrencia       PP1       PP2       PP3       PP4       PP5  \
828       -44667.0  2.447845 -1.979710 -0.900147 -1.594519  1.818622
830       -44667.0  2.451616 -1.973770 -0.902784 -1.595978  1.805211
918       -44706.0  1.582883 -1.531487 -0.667022 -0.158470  0.985789
920       -44706.0  1.567447 -1.555799 -0.656228 -0.152497  1.040683
927       -44708.0  4.678386 -3.451893  1.946664 -0.277297  3.050028
929       -44708.0  4.686662 -3.284455  1.260573 -0.283085  3.713478
1749      -45102.0  0.940684 -0.903703 -0.978734  0.437718 -0.876943
1751      -45102.0  0.960468 -0.872543 -0.992569  0.430062 -0.947298
2020      -45242.0  2.843254 -2.163409  0.155632  1.535508  0.527553
2022      -45242.0  2.848328 -2.155417  0.152084  1.533544  0.509508
2481      -45473.0  1.874503 -1.947798 -0.098195  1.482192  0.665688
2483      -45473.0  1.224211 -2.971995  0.356549  1.733830  2.978190
3426      -45971.0 -1.289497 -0.107352 -0.198329 -0.637300  0.080349
3428      -45971.0 -1.289478 -0.107320 -0.198318 -0.637309  0.080359
4311      -46387.0  2.525765 -1.934136 -0.803184 -1.571913  1.621559
4312      -46387.0  2.525765 -1.934136 -0.803184 -1.571913  1.621559
4313      -46387.0  2.525765 -1.934136 -0.803184 -1.571913  1.621559
6055      -47252.0  1.878658 -0.752120 -1.514593 -0.658044  0.394475
6057      -47252.0  1.913817 -0.696747 -1.539178 -0.671649  0.269448
8289      -48321.0 -1.196288 -0.090215 -0.536234 -0.847762  0.488749
8290      -48321.0 -1.196288 -0.090215 -0.536234 -0.847762  0.488749
8292      -48321.0 -1.212806 -0.117509 -0.545821 -0.840817  0.480317
8293      -48321.0 -1.212806 -0.117509 -0.545821 -0.840817  0.480317
8295      -48321.0 -1.203301 -0.101803 -0.540304 -0.844813  0.485169
8296      -48321.0 -1.203301 -0.101803 -0.540304 -0.844813  0.485169
9292      -48780.0 -1.068566  0.052581 -1.389764 -1.783097  0.761890
9294      -48780.0 -0.827041  0.451685 -1.249578 -1.884651  0.885191
10911     -49561.0  1.543198 -1.590305 -0.650783  1.229780  0.659293
10913     -49561.0  0.703708 -2.912485 -0.063735  1.554630  3.644603
11670     -49931.0  3.699370 -2.412031  0.681610  0.112343  1.006368
...            ...       ...       ...       ...       ...       ...
140500   -128860.0  9.031135 -4.602548  5.709994  0.388750  5.213013
140502   -128860.0  9.070453 -4.540622  5.682499  0.373536  5.073193
140503   -128860.0  9.070453 -4.540622  5.682499  0.373536  5.073193
140504   -128860.0  9.070453 -4.540622  5.682499  0.373536  5.073193
140506   -128860.0  9.053350 -4.567559  5.694459  0.380154  5.134014
140507   -128860.0  9.053350 -4.567559  5.694459  0.380154  5.134014
140508   -128860.0  9.053350 -4.567559  5.694459  0.380154  5.134014
140510   -128860.0  9.070817 -4.540050  5.682245  0.373395  5.071901
140511   -128860.0  9.070817 -4.540050  5.682245  0.373395  5.071901
140512   -128860.0  9.070817 -4.540050  5.682245  0.373395  5.071901
140865   -129013.0  1.760329 -2.052557 -0.574398 -2.768605 -0.890918
140866   -129013.0  1.760329 -2.052557 -0.574398 -2.768605 -0.890918
140867   -129013.0  1.760329 -2.052557 -0.574398 -2.768605 -0.890918
141828   -129422.0 -1.855776  0.666473  0.618532 -0.087655 -0.114717
141830   -129422.0 -1.862730  0.654980  0.614496 -0.084731 -0.118268
```

142167	-129572.0	-1.946754	0.581398	0.233863	-0.426496	0.948617
142169	-129572.0	-1.928150	0.612140	0.244661	-0.434319	0.958114
142310	-129632.0	0.484665	1.145539	-0.522770	0.614029	-1.952371
142312	-129632.0	0.493060	1.158761	-0.528641	0.610780	-1.982224
145345	-131011.0	0.791645	-1.479846	1.278378	1.185195	-0.373919
145347	-131011.0	0.077452	-2.604686	1.777807	1.461560	2.165823
146085	-131356.0	-2.050015	-1.246896	5.789343	-0.906735	-4.491831
146087	-131356.0	-2.091936	-0.148705	3.738184	-0.159139	-3.386352
147876	-132220.0	1.424878	-1.797959	0.731765	1.026684	0.267218
147878	-132220.0	0.698155	-2.942533	1.239957	1.307898	2.851517
147912	-132234.0	1.619943	-2.606157	-0.435562	-3.477395	0.828810
148892	-132699.0	-0.600186	2.654198	2.542308	-0.456329	-0.369346
148894	-132699.0	-0.597762	2.715289	1.697999	-0.421503	0.101838
149074	-132796.0	1.645271	-1.626502	-1.207925	-2.873895	0.330077
149076	-132796.0	1.649243	-1.620246	-1.210703	-2.875432	0.315952

	PP6	PP7	PP8	PP9	...	PP21	PP22	\
828	-0.850377	1.584894	-2.243695	-0.094686	...	0.149312	0.461226	
830	-0.841257	1.597407	-2.246228	-0.095427	...	0.147537	0.462306	
918	0.527557	0.223909	-0.928606	0.173164	...	0.065562	0.275115	
920	0.490230	0.172693	-0.918241	0.176195	...	0.072826	0.270694	
927	0.865884	2.385780	-3.357111	0.523805	...	-0.198531	0.418373	
929	0.364052	2.859323	-3.536301	0.077482	...	-0.320571	-0.016018	
1749	-1.047219	-0.859711	0.240886	-0.607975	...	0.307423	0.314469	
1751	-0.999378	-0.794069	0.227603	-0.611860	...	0.298113	0.320135	
2020	1.077263	-0.261374	-0.535435	-0.475024	...	0.281940	0.523384	
2022	1.089534	-0.244537	-0.538842	-0.476020	...	0.279552	0.524838	
2481	0.994484	-0.314571	-0.739119	-0.023230	...	0.226122	0.623880	
2483	-0.577974	-2.472135	-0.302505	0.104450	...	0.532133	0.437621	
3426	0.213012	-0.011484	0.123796	-0.411247	...	0.278204	0.581097	
3428	0.213008	-0.011497	0.123798	-0.411246	...	0.278198	0.581103	
4311	-0.981111	1.486179	-2.180808	0.069379	...	0.148983	0.410374	
4312	-0.981111	1.486179	-2.180808	0.069379	...	0.148983	0.410374	
4313	-0.981111	1.486179	-2.180808	0.069379	...	0.148983	0.410374	
6055	-1.473919	0.435243	-0.957995	-0.461980	...	0.085277	-0.038925	
6057	-1.388904	0.551893	-0.981601	-0.468883	...	0.068732	-0.028854	
8289	0.686884	-0.070069	0.197166	-0.316271	...	0.209673	0.416199	
8290	0.686884	-0.070069	0.197166	-0.316271	...	0.209673	0.416199	
8292	0.689961	-0.058723	0.194947	-0.317278	...	0.214853	0.411687	
8293	0.689961	-0.058723	0.194947	-0.317278	...	0.214853	0.411687	
8295	0.688190	-0.065252	0.196224	-0.316698	...	0.211872	0.414283	
8296	0.688190	-0.065252	0.196224	-0.316698	...	0.211872	0.414283	
9292	-0.437976	0.459556	-0.202723	-1.063006	...	0.384900	0.566345	
9294	-0.482966	0.293657	-0.170273	-1.048281	...	0.309147	0.632329	
10911	0.987309	-0.250606	-0.785127	0.503223	...	0.067575	0.269103	
10913	-1.042646	-3.035901	-0.221483	0.668052	...	0.462617	0.028653	
11670	-1.544806	1.454448	-2.614846	0.475832	...	-0.195860	-0.290994	
...	...	...	...	...	...	...	...	

140500	-0.930310	5.379218	-7.033614	0.977720	...	-1.013577	-0.840098
140502	-0.835235	5.509670	-7.060012	0.970000	...	-1.032079	-0.828836
140503	-0.835235	5.509670	-7.060012	0.970000	...	-1.032079	-0.828836
140504	-0.835235	5.509670	-7.060012	0.970000	...	-1.032079	-0.828836
140506	-0.876592	5.452924	-7.048529	0.973358	...	-1.024031	-0.833735
140507	-0.876592	5.452924	-7.048529	0.973358	...	-1.024031	-0.833735
140508	-0.876592	5.452924	-7.048529	0.973358	...	-1.024031	-0.833735
140510	-0.834357	5.510875	-7.060256	0.969929	...	-1.032250	-0.828732
140511	-0.834357	5.510875	-7.060256	0.969929	...	-1.032250	-0.828732
140512	-0.834357	5.510875	-7.060256	0.969929	...	-1.032250	-0.828732
140865	-0.683093	-0.536781	-0.224398	1.254269	...	0.364658	1.311868
140866	-0.683093	-0.536781	-0.224398	1.254269	...	0.364658	1.311868
140867	-0.683093	-0.536781	-0.224398	1.254269	...	0.364658	1.311868
141828	-1.491618	0.952031	-0.528168	-1.010252	...	-0.289160	-1.126544
141830	-1.490322	0.956808	-0.529103	-1.010676	...	-0.286979	-1.128444
142167	0.407090	0.812064	-0.146576	-1.194853	...	-0.234114	-0.804711
142169	0.403624	0.799285	-0.144076	-1.193718	...	-0.239950	-0.799629
142310	-1.381680	0.330775	-0.251166	-0.896257	...	-0.546275	-2.107344
142312	-1.361380	0.358628	-0.256802	-0.897906	...	-0.550226	-2.104940
145345	0.335704	-0.417306	-0.811472	0.348181	...	0.074344	0.238701
145347	-1.391272	-2.786885	-0.331954	0.488408	...	0.410425	0.034139
146085	-1.994031	-0.765272	-0.432772	0.653751	...	-0.002492	-0.150827
146087	-2.869366	-0.462437	-0.467049	0.437896	...	-0.299464	-0.955414
147876	1.183675	-0.768201	-0.514182	-0.001425	...	0.042513	0.197442
147878	-0.573600	-3.179351	-0.026251	0.141262	...	0.384490	-0.010710
147912	-2.602139	4.647154	10.212223	2.024113	...	4.792316	-2.342932
148892	-1.677936	-0.666785	-0.210393	-0.484479	...	-0.443321	0.041683
148894	-2.035217	-0.116527	-0.437828	-0.852160	...	-0.050966	1.194228
149074	-0.328809	0.368182	-0.990243	1.170417	...	-0.450746	-0.886486
149076	-0.319205	0.381360	-0.992910	1.169637	...	-0.452615	-0.885348

	PP23	PP24	PP25	PP26	PP27	PP28	Sacado	\
828	-0.153280	-0.207583	0.016627	0.409590	0.227248	0.023966	-7.00	
830	-0.152206	-0.207392	0.017109	0.409981	0.230704	0.022505	-3.99	
918	0.015844	-0.542599	-0.053804	-0.197270	0.156040	0.024100	-18.73	
920	0.011447	-0.543381	-0.055775	-0.198870	0.141895	0.030078	-31.05	
927	-0.241571	-0.536152	-0.228046	-0.160962	1.251586	0.306001	-12.23	
929	-0.271662	-0.544310	-0.036979	-0.212491	1.176372	0.306610	-42.72	
1749	0.266982	1.283902	-0.129588	-0.241267	0.495507	0.304586	-26.88	
1751	0.272617	1.284905	-0.127062	-0.239217	0.513636	0.296925	-11.09	
2020	0.179738	0.063810	0.163243	-0.537724	1.085632	0.842339	-19.00	
2022	0.181184	0.064067	0.163890	-0.537198	1.090282	0.840374	-14.95	
2481	-0.024829	0.093113	0.043570	-0.705951	-0.340573	-0.209147	-40.00	
2483	-0.210068	0.060139	-0.039457	-0.773333	-0.936464	0.042673	-559.00	
3426	0.100286	0.390937	-0.590368	-0.318255	0.018919	-0.006222	-7.98	
3428	0.100295	0.390937	-0.590364	-0.318254	0.018921	-0.006224	-7.99	
4311	-0.095628	-0.024774	-0.047707	0.402202	0.213922	0.042551	-9.90	
4312	-0.095628	-0.024774	-0.047707	0.402202	0.213922	0.042551	-9.90	

4313	-0.095628	-0.024774	-0.047707	0.402202	0.213922	0.042551	-9.90
6055	0.215722	1.195828	-0.297348	0.250901	0.583929	0.405957	-53.80
6057	0.225737	1.197611	-0.292859	0.254545	0.616146	0.392342	-25.74
8289	0.016219	-0.439244	-0.480025	-0.264399	0.019545	-0.018169	-25.77
8290	0.016219	-0.439244	-0.480025	-0.264399	0.019545	-0.018169	-25.77
8292	0.008683	-0.438805	-0.483164	-0.265311	0.017929	-0.016680	-17.22
8293	0.008683	-0.438805	-0.483164	-0.265311	0.017929	-0.016680	-17.22
8295	0.013019	-0.439057	-0.481358	-0.264786	0.018859	-0.017537	-22.14
8296	0.013019	-0.439057	-0.481358	-0.264786	0.018859	-0.017537	-22.14
9292	-0.116626	-0.437554	-0.421707	0.533001	-0.100001	-0.033441	-4.98
9294	-0.006434	-0.443969	-0.375803	0.546334	-0.076379	-0.055209	-130.00
10911	-0.146288	-0.456151	0.306232	-0.671955	-0.134498	-0.103593	-29.00
10913	-0.385422	-0.498719	0.199048	-0.758942	-0.903759	0.221492	-699.00
11670	-0.174667	0.991617	0.295017	-0.230921	1.402923	0.608549	-21.13
...	...	...	...	...	...	...	...
140500	-0.234780	1.352273	0.018903	-0.611472	2.802598	0.918644	-35.46
140502	-0.223580	1.354267	0.023923	-0.607398	2.838626	0.903418	-4.08
140503	-0.223580	1.354267	0.023923	-0.607398	2.838626	0.903418	-4.08
140504	-0.223580	1.354267	0.023923	-0.607398	2.838626	0.903418	-4.08
140506	-0.228452	1.353400	0.021739	-0.609170	2.822954	0.910041	-17.73
140507	-0.228452	1.353400	0.021739	-0.609170	2.822954	0.910041	-17.73
140508	-0.228452	1.353400	0.021739	-0.609170	2.822954	0.910041	-17.73
140510	-0.223476	1.354285	0.023969	-0.607360	2.838959	0.903277	-3.79
140511	-0.223476	1.354285	0.023969	-0.607360	2.838959	0.903277	-3.79
140512	-0.223476	1.354285	0.023969	-0.607360	2.838959	0.903277	-3.79
140865	-0.046937	1.094539	0.754971	0.659055	1.432986	0.261143	-1.51
140866	-0.046937	1.094539	0.754971	0.659055	1.432986	0.261143	-1.51
140867	-0.046937	1.094539	0.754971	0.659055	1.432986	0.261143	-1.51
141828	-0.121292	1.565666	0.385410	-0.724869	-0.021312	0.075236	-15.15
141830	-0.124465	1.565851	0.384088	-0.725253	-0.021992	0.075863	-11.55
142167	-0.151925	-0.102158	0.315001	-0.593372	0.035681	0.061749	-7.03
142169	-0.143437	-0.102652	0.318537	-0.592345	0.037500	0.060072	-16.66
142310	-0.397686	1.608215	2.431705	-0.319148	0.112576	-0.084515	-26.45
142312	-0.395294	1.608641	2.432777	-0.318278	0.120269	-0.087766	-19.75
145345	0.011724	-0.202351	-0.235518	0.125195	-0.061553	0.059248	-29.00
145347	-0.191718	-0.238566	-0.326705	0.051191	-0.715999	0.335813	-599.00
146085	0.268852	-0.286542	-0.921104	0.171582	-0.031088	-0.033623	-1.00
146087	0.182722	-0.751730	-0.917921	0.174132	0.034289	0.085717	-1.00
147876	0.066629	0.124411	0.036346	0.125909	0.039955	-0.125407	-49.00
147878	-0.140383	0.087561	-0.056440	0.050607	-0.625973	0.156010	-629.00
147912	-0.518027	0.077433	-0.315683	-0.165119	-0.049758	-0.395228	-19.27
148892	0.466008	0.858110	0.373725	-0.654709	0.177613	-0.028672	-704.51
148894	0.080988	0.943411	1.023305	-0.156422	0.132204	-0.048220	-677.40
149074	0.137708	0.040727	0.383486	-0.000354	0.503884	-0.076193	-11.83
149076	0.138840	0.040928	0.383993	0.000058	0.507524	-0.077731	-8.66

	Fraude
828	0

830	0
918	0
920	0
927	0
929	0
1749	0
1751	0
2020	0
2022	0
2481	0
2483	0
3426	0
3428	0
4311	0
4312	0
4313	0
6055	0
6057	0
8289	0
8290	0
8292	0
8293	0
8295	0
8296	0
9292	0
9294	0
10911	0
10913	0
11670	0
...	...
140500	0
140502	0
140503	0
140504	0
140506	0
140507	0
140508	0
140510	0
140511	0
140512	0
140865	0
140866	0
140867	0
141828	0
141830	0
142167	0
142169	0
142310	0

```
142312       0
145345       0
145347       0
146085       0
146087       0
147876       0
147878       0
147912       0
148892       0
148894       0
149074       0
149076       0

[496 rows x 31 columns]
Linhas duplicadas por Ocorrencia somente:
        Ocorrencia       PP1       PP2       PP3       PP4       PP5  \
3         -44301.0 -1.061770 -0.105481 -0.226711 -0.929524 -0.100625
5         -44302.0 -1.226897 -0.057421 -0.645474 -1.209587  0.508127
6         -44302.0 -1.187076 -0.283468 -0.160228 -0.871200 -0.157315
7         -44302.0  3.532498  4.777980 -0.899279  0.341794 -2.695645
8         -44302.0 -1.273200  0.430986 -0.374847  1.429721  0.803957
11        -44304.0 -1.118176  0.113425 -1.341694 -1.211844  1.128599
15        -44307.0 -1.347450  0.874492 -0.873481  1.993250  1.829640
16        -44307.0 -1.215099 -0.042094  0.247738 -0.774207 -0.161264
18        -44308.0 -1.113277  0.124191 -0.506976 -0.762110  0.574466
19        -44308.0 -0.804932  0.448302 -0.212507 -1.356755  0.174896
22        -44310.0 -1.082875  0.463526  1.086728  1.485478 -0.659555
26        -44313.0 -1.107192  0.094741  0.325628 -0.177363  0.244454
27        -44313.0 -0.975078  0.315845 -0.846517 -1.449360  0.687804
29        -44314.0 -1.132055  0.996445 -0.894827  0.795648  1.146411
31        -44315.0 -1.047878  0.516899 -1.109663 -0.109822  0.916704
32        -44315.0  0.115026 -1.408212 -1.384819 -2.400055 -0.901880
35        -44320.0  2.307957 -0.868635  0.304017  0.438678  0.029155
37        -44321.0 -1.182032 -0.326653  0.267051 -1.001421 -0.387597
38        -44321.0  1.221357 -0.597587 -1.601322 -2.083818 -2.421130
39        -44321.0 -1.162529  1.141016 -0.814359  0.515110  1.486830
40        -44321.0  4.637122 -3.919194  0.614419  2.432366  0.784109
41        -44321.0 -1.240021  1.772805  1.058088  2.382751 -0.655955
44        -44323.0 -1.220683 -0.412753  0.393457 -0.785179  0.162242
46        -44324.0  0.492596 -0.976523 -1.664666 -0.001562  0.062942
48        -44325.0 -0.506519  0.819796  0.201278 -1.412227  0.097519
49        -44325.0 -1.258000 -0.356130 -0.306815 -0.692962  0.372462
51        -44326.0 -1.181022  0.637220 -0.593523  1.214785  1.045497
54        -44328.0  2.782534  1.478002 -1.878356  0.218030  0.355298
56        -44329.0  0.275138  4.286284 -0.555130  0.220299  3.032724
57        -44329.0  3.575393 -2.263615  1.183434  0.701136 -0.362412
...            ...       ...       ...       ...       ...       ...
149943   -133207.0  4.179500  4.359499  1.542454  0.558873  4.585864
```

149944	-133207.0	-0.844281	3.330230	1.705140	2.401643	1.077727
149947	-133209.0	-2.048629	0.367489	2.544065	0.728472	-2.387380
149948	-133209.0	-0.187998	-1.123733	-0.120336	-2.073112	-0.808504
149951	-133211.0	0.481853	-1.271337	0.583532	0.716726	-1.109235
149952	-133211.0	-1.863803	0.654008	0.412362	-0.093165	0.653742
149954	-133212.0	-0.034006	-0.922148	-0.320239	0.588479	-0.509390
149955	-133212.0	1.804966	-1.417304	3.649482	1.027651	3.775529
149960	-133216.0	-1.974117	1.296857	1.789526	3.026517	-1.168127
149962	-133217.0	-1.807015	0.698912	0.260289	-0.190686	0.847228
149964	-133219.0	1.100771	-1.648408	-0.044837	1.260544	-1.499065
149965	-133219.0	-1.597321	1.513308	1.165231	1.720620	0.286431
149967	-133220.0	1.139332	-0.578320	-1.530765	0.700564	0.126485
149969	-133221.0	-1.722507	0.788653	1.446628	0.072066	-0.503136
149971	-133222.0	-1.468190	0.573515	2.272459	-1.017709	-1.333551
149972	-133222.0	0.612879	0.587153	1.182241	-0.007825	-0.308115
149974	-133223.0	0.514182	-0.293052	-1.615246	0.800298	-0.525206
149977	-133225.0	-2.010648	0.276660	1.098537	-0.579355	0.240295
149979	-133226.0	2.017282	1.568706	0.035778	1.976914	-1.422737
149980	-133226.0	-0.081466	-0.995292	0.330903	0.637603	-0.959816
149985	-133230.0	-0.076588	-1.045554	1.078269	-0.027139	-0.175636
149986	-133230.0	-1.163582	1.814206	0.144765	-1.069574	1.395047
149987	-133230.0	-0.180614	-1.100063	1.022411	-0.003125	-0.208371
149989	-133231.0	-1.852532	0.390781	1.000306	-0.521646	0.309452
149991	-133232.0	0.725038	-0.483498	-0.244706	-1.308022	0.562931
149992	-133232.0	-2.005222	0.129549	2.097779	-0.145306	-0.596579
149994	-133233.0	-2.041642	1.517956	1.757935	1.815668	-1.089298
149997	-133236.0	1.943282	-0.898549	0.402690	0.179340	-0.301361
149998	-133236.0	1.103962	-0.940976	-1.652220	0.365992	0.299453
149999	-133236.0	-2.320998	0.496412	2.593376	0.962050	-0.323330

	PP6	PP7	PP8	PP9	...	PP21	PP22	\
3	-0.300173	0.029912	-0.205934	0.233190	...	-0.147422	-0.426827	
5	0.231503	0.256178	-0.073481	-0.805029	...	0.392138	1.018474	
6	-0.019609	-0.042348	-0.061183	0.292538	...	-0.086133	-0.313357	
7	0.814919	1.299022	-0.491621	1.140031	...	-0.236773	0.487309	
8	0.597754	0.377943	-0.021881	-1.524914	...	0.325682	0.717795	
11	0.133668	0.745899	-0.283339	-0.707395	...	-0.038199	-0.167271	
15	1.297897	0.844260	0.206649	-0.433855	...	0.315272	0.324280	
16	0.004911	-0.014432	-0.081928	-0.071755	...	-0.002786	0.141817	
18	0.388953	0.099294	-0.022572	-0.261967	...	0.161383	0.426340	
19	-0.631969	0.027822	-0.319939	-0.235690	...	-0.046426	0.022834	
22	-0.195302	-0.463953	0.087227	-0.628466	...	0.284468	1.019828	
26	1.359825	-0.628739	0.463639	0.261384	...	-0.027217	0.118274	
27	-0.235910	0.331947	-0.158167	-0.905233	...	-0.027697	-0.230044	
29	-0.556060	1.213977	-0.351772	0.658131	...	-0.523305	-1.362715	
31	-0.438309	0.717068	-0.276455	-0.800787	...	0.119687	0.132351	
32	0.247150	-1.409374	0.664603	1.400895	...	-0.080735	-0.569349	
35	-1.179519	0.404545	-1.753573	0.795754	...	-0.339147	-0.656587	

37	0.190439	-0.343358	0.011200	0.490704	...	-0.061203	-0.166268
38	-3.242185	-0.182718	-1.029669	1.574973	...	0.106676	0.405349
39	0.174976	0.992741	-0.000120	0.095633	...	-0.348447	-0.820483
40	0.376329	-0.155145	-0.599167	-3.179263	...	0.693890	0.262201
41	-3.404521	1.434898	-0.776598	1.902277	...	0.321623	1.096284
44	1.250604	-0.164204	0.071801	0.134876	...	0.095586	0.352837
46	0.614897	-0.637429	-0.057768	0.402765	...	0.176875	0.431812
48	-0.499301	-0.433725	-0.138058	0.167849	...	-0.255514	-0.196741
49	1.074096	-0.084462	0.202683	-0.035094	...	0.288326	0.830769
51	0.459895	0.553235	-0.030493	-1.930857	...	-0.121836	-0.661536
54	0.242499	-0.251224	0.312750	0.548804	...	0.386752	0.034173
56	-1.030975	0.864361	-0.269361	0.616115	...	-0.110326	0.765596
57	-3.095939	4.047788	13.552401	1.687846	...	6.573153	-2.287873
...	...	...	...	...	...	...	...
149943	-4.712300	-6.708490	0.969044	-0.155478	...	0.362579	-1.332317
149944	-0.967115	0.232603	-0.170216	-1.155433	...	0.238345	1.172368
149947	-3.318407	0.477692	-0.791276	-0.403246	...	0.292703	0.835377
149948	0.539494	-1.153551	0.372012	1.070275	...	-0.172820	-0.796442
149951	0.811334	-1.143664	0.214196	-0.060885	...	0.279833	0.540523
149952	0.137965	0.609632	-0.029443	-1.148072	...	-0.254206	-0.869341
149954	1.121632	-1.061309	0.261381	0.190169	...	0.248098	0.491085
149955	-3.265869	-1.528084	5.072494	2.890832	...	-4.564125	-0.461332
149960	-3.421960	1.473329	-1.009339	-2.730104	...	-0.110774	-0.458175
149962	0.285520	0.616249	-0.038492	-1.148734	...	-0.290345	-0.945275
149964	0.465066	-2.229448	1.511768	-2.283193	...	0.202802	-1.415563
149965	-0.930616	0.726403	-0.385000	-1.993243	...	-0.407446	-1.078183
149967	-0.199015	0.150824	-0.393944	-0.861519	...	0.035418	-0.274308
149969	-1.083225	0.217950	-0.331119	-0.925990	...	0.019606	-0.029901
149971	-1.101036	-0.617678	-0.204555	0.352642	...	-0.389520	-0.866921
149972	-0.091757	-0.635118	-0.138615	-0.464194	...	-0.095908	-0.448965
149974	-1.109187	-0.205817	-0.346199	-0.698304	...	-0.228521	-1.164362
149977	0.876971	-0.001639	0.213264	-1.208393	...	-0.091508	-0.554219
149979	0.576726	-0.017808	0.006264	1.005563	...	0.049830	0.095234
149980	0.487422	-0.890990	-0.035733	0.364656	...	0.270907	0.651510
149985	1.303074	-0.936198	-0.004175	0.388199	...	-0.387979	-1.051819
149986	-0.335362	0.497708	-0.173726	-1.454900	...	-0.088837	0.244182
149987	1.445858	-0.602641	-0.118823	0.112330	...	-0.331292	-0.920748
149989	0.951259	-0.136549	0.298178	-0.856320	...	-0.072687	-0.239713
149991	-0.862232	-0.850322	-0.379935	-0.039274	...	0.389262	0.974373
149992	0.883900	-0.594168	0.328442	0.080784	...	-0.167419	-0.513273
149994	-3.858948	1.768771	-1.011176	-0.181492	...	-0.458184	-1.245442
149997	0.134193	-1.288226	0.478686	-0.308260	...	0.341853	0.157110
149998	-0.489269	0.106126	-0.832525	-0.349337	...	0.357995	0.842373
149999	1.454052	-0.329897	0.520266	0.860366	...	-0.452442	-1.325093

	PP23	PP24	PP25	PP26	PP27	PP28	Sacado	\
3	0.070413	0.283090	-0.487739	0.288220	-0.035644	-0.007305	-31.28	
5	-0.068629	0.190145	-0.393703	0.603708	-0.039229	-0.023810	-2.27	

6	0.122671	0.289686	-0.637261	0.296274	-0.028010	-0.003439	-1.50
7	-1.403715	1.384792	0.056282	-0.339581	0.121420	-0.227391	-384.99
8	-0.046891	-0.024916	-0.360409	0.281505	-0.020939	-0.005926	-1.14
11	-0.031323	-0.489935	-0.286082	0.424303	-0.057538	-0.026498	-1.18
15	-0.112219	-0.721698	-0.168070	0.095688	-0.060231	-0.038529	-20.20
16	0.264780	0.872962	-0.739584	0.271126	0.009964	-0.000534	-39.16
18	0.029600	-0.352060	-0.395698	-0.249226	0.036142	-0.008575	-45.91
19	0.149676	0.299685	-0.456418	0.310663	-0.016464	-0.023769	-149.00
22	0.262184	1.705515	-0.677687	0.932712	-0.019508	-0.016686	-148.75
26	0.191624	-0.504522	-0.532177	-1.055915	0.130419	-0.011916	-119.00
27	0.148829	-0.108917	-0.565031	0.254591	-0.058481	-0.036939	-86.08
29	0.104192	0.230618	-0.225120	0.013736	-0.055299	-0.018883	-69.99
31	-0.050226	-0.118862	-0.049064	-0.947783	0.022701	-0.009664	-48.95
32	0.159677	-0.430686	0.265303	-0.050320	0.522341	0.389265	-1.93
35	0.132531	0.975890	0.427614	-0.444930	0.408354	0.239485	-78.26
37	0.171095	0.304925	-0.760560	0.247426	0.003928	0.004698	-15.80
38	0.075924	2.822684	0.016500	0.125639	-0.097517	-0.072360	-36.70
39	0.271623	0.052875	-0.483774	0.028408	-0.027706	-0.041484	-129.00
40	-0.092857	0.324404	-0.437814	-0.635853	-0.216020	0.796954	-0.77
41	0.002027	-0.955708	-0.233007	0.412516	-0.014788	-0.051849	-177.70
44	0.042760	-0.394919	-0.447453	-0.345204	0.050391	-0.024311	-0.76
46	-0.062611	-0.376714	0.312587	-0.080973	-0.279350	-0.121121	-7.78
48	0.367721	0.269935	-0.528363	0.250220	0.023867	-0.049912	-315.34
49	-0.130364	-0.344871	-0.215840	-0.094554	0.023012	-0.030633	-0.89
51	0.146284	-0.177014	-0.544788	-0.183451	-0.048469	-0.018666	-31.88
54	0.958122	-0.357559	-0.307712	0.458944	0.927209	0.485878	-153.50
56	0.607361	-0.240450	0.108608	0.294692	0.052681	-0.157986	-835.00
57	-0.567235	1.328727	0.539571	-0.225416	0.156753	0.273579	-178.95
...	...	...	...	...	...	...	...
149943	-2.656684	1.500171	-0.133468	-1.468283	-0.659448	0.163891	-1600.00
149944	0.060700	1.451734	0.771245	1.215379	-0.034226	-0.052391	-617.58
149947	-0.380780	-0.698235	0.265199	-0.219159	0.052337	0.066750	-1.98
149948	0.166619	-0.047432	0.347892	0.186813	-0.259572	-0.213359	-22.09
149951	-0.133492	-0.500217	1.024053	0.001681	-0.431338	-0.391238	-24.99
149952	-0.024408	0.266204	0.149939	-0.116657	-0.006760	0.043565	-64.95
149954	-0.059348	-0.066118	0.457325	-0.134528	-0.253374	-0.099857	-1.79
149955	-0.449869	0.927111	0.851773	-0.681479	-0.660276	0.467920	-716.82
149960	-0.221161	-0.711916	0.182071	0.896612	-0.116210	0.027784	-34.36
149962	-0.074708	-0.082614	0.229517	-0.096358	-0.007232	0.038728	-74.95
149964	0.228434	-0.769656	0.566237	0.433385	-0.227002	0.775267	-2.99
149965	0.036816	1.585237	0.357038	0.059149	-0.039820	0.040355	-183.02
149967	0.171536	0.494049	0.435037	-0.194310	-0.644629	-0.350960	-27.00
149969	-0.126310	1.624082	0.276920	-0.014188	0.001124	0.066662	-105.00
149971	0.111977	1.687658	-0.078804	0.375491	0.006849	0.057055	-209.03
149972	-0.203246	-0.328908	2.177730	0.336409	-0.170936	0.187118	-193.60
149974	0.145832	0.993183	0.599722	-0.623886	-0.254156	0.014830	-11.50
149977	-0.017820	0.058939	-0.240699	0.134838	0.003559	0.058344	-13.73
149979	0.525687	1.430365	-1.098772	0.072544	-0.236500	0.010514	-230.45

```
149980 -0.107843 -0.609232  0.470817 -0.091213 -0.222871 -0.075734    -5.99
149985 -0.071051 -0.003717  0.194947  0.159311  0.086168  0.038301   -85.25
149986 -0.006485 -0.054011  0.643924 -0.062500  0.050965 -0.017235  -378.80
149987  0.059538  0.033873  0.277221  0.151729  0.074708  0.046507   -21.28
149989 -0.080927 -0.007950  0.024803  0.348099  0.008373  0.032704   -93.65
149991 -0.516036 -0.555578  0.581815  1.032067 -0.013116  0.027044  -220.17
149992  0.034386 -0.852609 -0.316635 -0.767426  0.139487  0.082271   -45.36
149994 -0.074987 -0.764075  0.086074 -0.039620 -0.034757  0.044239   -59.78
149997  0.050649  0.425485  0.590081 -0.282905  0.730029  0.155896  -108.21
149998  0.254363  0.512391 -0.456296  0.414252 -0.202450 -0.054214    -1.00
149999  0.168678 -0.738673 -0.734526 -0.278194  0.104142  0.091720    -1.79

        Fraude
3            0
5            0
6            0
7            0
8            0
11           0
15           0
16           0
18           0
19           0
22           0
26           0
27           0
29           0
31           0
32           0
35           0
37           0
38           0
39           0
40           0
41           0
44           0
46           0
48           0
49           0
51           0
54           0
56           0
57           0
...        ...
149943       0
149944       0
149947       0
149948       0
```

```
149951       0
149952       0
149954       0
149955       0
149960       0
149962       0
149964       0
149965       0
149967       0
149969       0
149971       0
149972       0
149974       0
149977       0
149979       0
149980       0
149985       0
149986       0
149987       0
149989       0
149991       0
149992       0
149994       0
149997       0
149998       0
149999       0

[85042 rows x 31 columns]
Linhas duplicadas por Ocorrencia e Fraude somente:
        Ocorrencia       PP1       PP2       PP3       PP4       PP5  \
3         -44301.0 -1.061770 -0.105481 -0.226711 -0.929524 -0.100625
5         -44302.0 -1.226897 -0.057421 -0.645474 -1.209587  0.508127
6         -44302.0 -1.187076 -0.283468 -0.160228 -0.871200 -0.157315
7         -44302.0  3.532498  4.777980 -0.899279  0.341794 -2.695645
8         -44302.0 -1.273200  0.430986 -0.374847  1.429721  0.803957
11        -44304.0 -1.118176  0.113425 -1.341694 -1.211844  1.128599
15        -44307.0 -1.347450  0.874492 -0.873481  1.993250  1.829640
16        -44307.0 -1.215099 -0.042094  0.247738 -0.774207 -0.161264
18        -44308.0 -1.113277  0.124191 -0.506976 -0.762110  0.574466
19        -44308.0 -0.804932  0.448302 -0.212507 -1.356755  0.174896
22        -44310.0 -1.082875  0.463526  1.086728  1.485478 -0.659555
26        -44313.0 -1.107192  0.094741  0.325628 -0.177363  0.244454
27        -44313.0 -0.975078  0.315845 -0.846517 -1.449360  0.687804
29        -44314.0 -1.132055  0.996445 -0.894827  0.795648  1.146411
31        -44315.0 -1.047878  0.516899 -1.109663 -0.109822  0.916704
32        -44315.0  0.115026 -1.408212 -1.384819 -2.400055 -0.901880
35        -44320.0  2.307957 -0.868635  0.304017  0.438678  0.029155
37        -44321.0 -1.182032 -0.326653  0.267051 -1.001421 -0.387597
```

```
38        -44321.0  1.221357 -0.597587 -1.601322 -2.083818 -2.421130
39        -44321.0 -1.162529  1.141016 -0.814359  0.515110  1.486830
40        -44321.0  4.637122 -3.919194  0.614419  2.432366  0.784109
41        -44321.0 -1.240021  1.772805  1.058088  2.382751 -0.655955
44        -44323.0 -1.220683 -0.412753  0.393457 -0.785179  0.162242
46        -44324.0  0.492596 -0.976523 -1.664666 -0.001562  0.062942
48        -44325.0 -0.506519  0.819796  0.201278 -1.412227  0.097519
49        -44325.0 -1.258000 -0.356130 -0.306815 -0.692962  0.372462
51        -44326.0 -1.181022  0.637220 -0.593523  1.214785  1.045497
54        -44328.0  2.782534  1.478002 -1.878356  0.218030  0.355298
56        -44329.0  0.275138  4.286284 -0.555130  0.220299  3.032724
57        -44329.0  3.575393 -2.263615  1.183434  0.701136 -0.362412
...            ...       ...       ...       ...       ...       ...
149943   -133207.0  4.179500  4.359499  1.542454  0.558873  4.585864
149944   -133207.0 -0.844281  3.330230  1.705140  2.401643  1.077727
149947   -133209.0 -2.048629  0.367489  2.544065  0.728472 -2.387380
149948   -133209.0 -0.187998 -1.123733 -0.120336 -2.073112 -0.808504
149951   -133211.0  0.481853 -1.271337  0.583532  0.716726 -1.109235
149952   -133211.0 -1.863803  0.654008  0.412362 -0.093165  0.653742
149954   -133212.0 -0.034006 -0.922148 -0.320239  0.588479 -0.509390
149955   -133212.0  1.804966 -1.417304  3.649482  1.027651  3.775529
149960   -133216.0 -1.974117  1.296857  1.789526  3.026517 -1.168127
149962   -133217.0 -1.807015  0.698912  0.260289 -0.190686  0.847228
149964   -133219.0  1.100771 -1.648408 -0.044837  1.260544 -1.499065
149965   -133219.0 -1.597321  1.513308  1.165231  1.720620  0.286431
149967   -133220.0  1.139332 -0.578320 -1.530765  0.700564  0.126485
149969   -133221.0 -1.722507  0.788653  1.446628  0.072066 -0.503136
149971   -133222.0 -1.468190  0.573515  2.272459 -1.017709 -1.333551
149972   -133222.0  0.612879  0.587153  1.182241 -0.007825 -0.308115
149974   -133223.0  0.514182 -0.293052 -1.615246  0.800298 -0.525206
149977   -133225.0 -2.010648  0.276660  1.098537 -0.579355  0.240295
149979   -133226.0  2.017282  1.568706  0.035778  1.976914 -1.422737
149980   -133226.0 -0.081466 -0.995292  0.330903  0.637603 -0.959816
149985   -133230.0 -0.076588 -1.045554  1.078269 -0.027139 -0.175636
149986   -133230.0 -1.163582  1.814206  0.144765 -1.069574  1.395047
149987   -133230.0 -0.180614 -1.100063  1.022411 -0.003125 -0.208371
149989   -133231.0 -1.852532  0.390781  1.000306 -0.521646  0.309452
149991   -133232.0  0.725038 -0.483498 -0.244706 -1.308022  0.562931
149992   -133232.0 -2.005222  0.129549  2.097779 -0.145306 -0.596579
149994   -133233.0 -2.041642  1.517956  1.757935  1.815668 -1.089298
149997   -133236.0  1.943282 -0.898549  0.402690  0.179340 -0.301361
149998   -133236.0  1.103962 -0.940976 -1.652220  0.365992  0.299453
149999   -133236.0 -2.320998  0.496412  2.593376  0.962050 -0.323330

             PP6       PP7        PP8       PP9  ...      PP21      PP22  \
3      -0.300173  0.029912  -0.205934  0.233190  ... -0.147422 -0.426827
5       0.231503  0.256178  -0.073481 -0.805029  ...  0.392138  1.018474
6      -0.019609 -0.042348  -0.061183  0.292538  ... -0.086133 -0.313357
```

7	0.814919	1.299022	-0.491621	1.140031	...	-0.236773	0.487309
8	0.597754	0.377943	-0.021881	-1.524914	...	0.325682	0.717795
11	0.133668	0.745899	-0.283339	-0.707395	...	-0.038199	-0.167271
15	1.297897	0.844260	0.206649	-0.433855	...	0.315272	0.324280
16	0.004911	-0.014432	-0.081928	-0.071755	...	-0.002786	0.141817
18	0.388953	0.099294	-0.022572	-0.261967	...	0.161383	0.426340
19	-0.631969	0.027822	-0.319939	-0.235690	...	-0.046426	0.022834
22	-0.195302	-0.463953	0.087227	-0.628466	...	0.284468	1.019828
26	1.359825	-0.628739	0.463639	0.261384	...	-0.027217	0.118274
27	-0.235910	0.331947	-0.158167	-0.905233	...	-0.027697	-0.230044
29	-0.556060	1.213977	-0.351772	0.658131	...	-0.523305	-1.362715
31	-0.438309	0.717068	-0.276455	-0.800787	...	0.119687	0.132351
32	0.247150	-1.409374	0.664603	1.400895	...	-0.080735	-0.569349
35	-1.179519	0.404545	-1.753573	0.795754	...	-0.339147	-0.656587
37	0.190439	-0.343358	0.011200	0.490704	...	-0.061203	-0.166268
38	-3.242185	-0.182718	-1.029669	1.574973	...	0.106676	0.405349
39	0.174976	0.992741	-0.000120	0.095633	...	-0.348447	-0.820483
40	0.376329	-0.155145	-0.599167	-3.179263	...	0.693890	0.262201
41	-3.404521	1.434898	-0.776598	1.902277	...	0.321623	1.096284
44	1.250604	-0.164204	0.071801	0.134876	...	0.095586	0.352837
46	0.614897	-0.637429	-0.057768	0.402765	...	0.176875	0.431812
48	-0.499301	-0.433725	-0.138058	0.167849	...	-0.255514	-0.196741
49	1.074096	-0.084462	0.202683	-0.035094	...	0.288326	0.830769
51	0.459895	0.553235	-0.030493	-1.930857	...	-0.121836	-0.661536
54	0.242499	-0.251224	0.312750	0.548804	...	0.386752	0.034173
56	-1.030975	0.864361	-0.269361	0.616115	...	-0.110326	0.765596
57	-3.095939	4.047788	13.552401	1.687846	...	6.573153	-2.287873
...	...	...	...	...	...	...	...
149943	-4.712300	-6.708490	0.969044	-0.155478	...	0.362579	-1.332317
149944	-0.967115	0.232603	-0.170216	-1.155433	...	0.238345	1.172368
149947	-3.318407	0.477692	-0.791276	-0.403246	...	0.292703	0.835377
149948	0.539494	-1.153551	0.372012	1.070275	...	-0.172820	-0.796442
149951	0.811334	-1.143664	0.214196	-0.060885	...	0.279833	0.540523
149952	0.137965	0.609632	-0.029443	-1.148072	...	-0.254206	-0.869341
149954	1.121632	-1.061309	0.261381	0.190169	...	0.248098	0.491085
149955	-3.265869	-1.528084	5.072494	2.890832	...	-4.564125	-0.461332
149960	-3.421960	1.473329	-1.009339	-2.730104	...	-0.110774	-0.458175
149962	0.285520	0.616249	-0.038492	-1.148734	...	-0.290345	-0.945275
149964	0.465066	-2.229448	1.511768	-2.283193	...	0.202802	-1.415563
149965	-0.930616	0.726403	-0.385000	-1.993243	...	-0.407446	-1.078183
149967	-0.199015	0.150824	-0.393944	-0.861519	...	0.035418	-0.274308
149969	-1.083225	0.217950	-0.331119	-0.925990	...	0.019606	-0.029901
149971	-1.101036	-0.617678	-0.204555	0.352642	...	-0.389520	-0.866921
149972	-0.091757	-0.635118	-0.138615	-0.464194	...	-0.095908	-0.448965
149974	-1.109187	-0.205817	-0.346199	-0.698304	...	-0.228521	-1.164362
149977	0.876971	-0.001639	0.213264	-1.208393	...	-0.091508	-0.554219
149979	0.576726	-0.017808	0.006264	1.005563	...	0.049830	0.095234
149980	0.487422	-0.890990	-0.035733	0.364656	...	0.270907	0.651510

```
149985  1.303074 -0.936198  -0.004175  0.388199  ... -0.387979 -1.051819
149986 -0.335362  0.497708  -0.173726 -1.454900  ... -0.088837  0.244182
149987  1.445858 -0.602641  -0.118823  0.112330  ... -0.331292 -0.920748
149989  0.951259 -0.136549   0.298178 -0.856320  ... -0.072687 -0.239713
149991 -0.862232 -0.850322  -0.379935 -0.039274  ...  0.389262  0.974373
149992  0.883900 -0.594168   0.328442  0.080784  ... -0.167419 -0.513273
149994 -3.858948  1.768771  -1.011176 -0.181492  ... -0.458184 -1.245442
149997  0.134193 -1.288226   0.478686 -0.308260  ...  0.341853  0.157110
149998 -0.489269  0.106126  -0.832525 -0.349337  ...  0.357995  0.842373
149999  1.454052 -0.329897   0.520266  0.860366  ... -0.452442 -1.325093

            PP23      PP24      PP25      PP26      PP27      PP28   Sacado  \
3       0.070413  0.283090 -0.487739  0.288220 -0.035644 -0.007305   -31.28
5      -0.068629  0.190145 -0.393703  0.603708 -0.039229 -0.023810    -2.27
6       0.122671  0.289686 -0.637261  0.296274 -0.028010 -0.003439    -1.50
7      -1.403715  1.384792  0.056282 -0.339581  0.121420 -0.227391  -384.99
8      -0.046891 -0.024916 -0.360409  0.281505 -0.020939 -0.005926    -1.14
11     -0.031323 -0.489935 -0.286082  0.424303 -0.057538 -0.026498    -1.18
15     -0.112219 -0.721698 -0.168070  0.095688 -0.060231 -0.038529   -20.20
16      0.264780  0.872962 -0.739584  0.271126  0.009964 -0.000534   -39.16
18      0.029600 -0.352060 -0.395698 -0.249226  0.036142 -0.008575   -45.91
19      0.149676  0.299685 -0.456418  0.310663 -0.016464 -0.023769  -149.00
22      0.262184  1.705515 -0.677687  0.932712 -0.019508 -0.016686  -148.75
26      0.191624 -0.504522 -0.532177 -1.055915  0.130419 -0.011916  -119.00
27      0.148829 -0.108917 -0.565031  0.254591 -0.058481 -0.036939   -86.08
29      0.104192  0.230618 -0.225120  0.013736 -0.055299 -0.018883   -69.99
31     -0.050226 -0.118862 -0.049064 -0.947783  0.022701 -0.009664   -48.95
32      0.159677 -0.430686  0.265303 -0.050320  0.522341  0.389265    -1.93
35      0.132531  0.975890  0.427614 -0.444930  0.408354  0.239485   -78.26
37      0.171095  0.304925 -0.760560  0.247426  0.003928  0.004698   -15.80
38      0.075924  2.822684  0.016500  0.125639 -0.097517 -0.072360   -36.70
39      0.271623  0.052875 -0.483774  0.028408 -0.027706 -0.041484  -129.00
40     -0.092857  0.324404 -0.437814 -0.635853 -0.216020  0.796954    -0.77
41      0.002027 -0.955708 -0.233007  0.412516 -0.014788 -0.051849  -177.70
44      0.042760 -0.394919 -0.447453 -0.345204  0.050391 -0.024311    -0.76
46     -0.062611 -0.376714  0.312587 -0.080973 -0.279350 -0.121121    -7.78
48      0.367721  0.269935 -0.528363  0.250220  0.023867 -0.049912  -315.34
49     -0.130364 -0.344871 -0.215840 -0.094554  0.023012 -0.030633    -0.89
51      0.146284 -0.177014 -0.544788 -0.183451 -0.048469 -0.018666   -31.88
54      0.958122 -0.357559 -0.307712  0.458944  0.927209  0.485878  -153.50
56      0.607361 -0.240450  0.108608  0.294692  0.052681 -0.157986  -835.00
57     -0.567235  1.328727  0.539571 -0.225416  0.156753  0.273579  -178.95
...          ...       ...       ...       ...       ...       ...      ...
149943 -2.656684  1.500171 -0.133468 -1.468283 -0.659448  0.163891 -1600.00
149944  0.060700  1.451734  0.771245  1.215379 -0.034226 -0.052391  -617.58
149947 -0.380780 -0.698235  0.265199 -0.219159  0.052337  0.066750    -1.98
149948  0.166619 -0.047432  0.347892  0.186813 -0.259572 -0.213359   -22.09
149951 -0.133492 -0.500217  1.024053  0.001681 -0.431338 -0.391238   -24.99
```

149952	-0.024408	0.266204	0.149939	-0.116657	-0.006760	0.043565	-64.95
149954	-0.059348	-0.066118	0.457325	-0.134528	-0.253374	-0.099857	-1.79
149955	-0.449869	0.927111	0.851773	-0.681479	-0.660276	0.467920	-716.82
149960	-0.221161	-0.711916	0.182071	0.896612	-0.116210	0.027784	-34.36
149962	-0.074708	-0.082614	0.229517	-0.096358	-0.007232	0.038728	-74.95
149964	0.228434	-0.769656	0.566237	0.433385	-0.227002	0.775267	-2.99
149965	0.036816	1.585237	0.357038	0.059149	-0.039820	0.040355	-183.02
149967	0.171536	0.494049	0.435037	-0.194310	-0.644629	-0.350960	-27.00
149969	-0.126310	1.624082	0.276920	-0.014188	0.001124	0.066662	-105.00
149971	0.111977	1.687658	-0.078804	0.375491	0.006849	0.057055	-209.03
149972	-0.203246	-0.328908	2.177730	0.336409	-0.170936	0.187118	-193.60
149974	0.145832	0.993183	0.599722	-0.623886	-0.254156	0.014830	-11.50
149977	-0.017820	0.058939	-0.240699	0.134838	0.003559	0.058344	-13.73
149979	0.525687	1.430365	-1.098772	0.072544	-0.236500	0.010514	-230.45
149980	-0.107843	-0.609232	0.470817	-0.091213	-0.222871	-0.075734	-5.99
149985	-0.071051	-0.003717	0.194947	0.159311	0.086168	0.038301	-85.25
149986	-0.006485	-0.054011	0.643924	-0.062500	0.050965	-0.017235	-378.80
149987	0.059538	0.033873	0.277221	0.151729	0.074708	0.046507	-21.28
149989	-0.080927	-0.007950	0.024803	0.348099	0.008373	0.032704	-93.65
149991	-0.516036	-0.555578	0.581815	1.032067	-0.013116	0.027044	-220.17
149992	0.034386	-0.852609	-0.316635	-0.767426	0.139487	0.082271	-45.36
149994	-0.074987	-0.764075	0.086074	-0.039620	-0.034757	0.044239	-59.78
149997	0.050649	0.425485	0.590081	-0.282905	0.730029	0.155896	-108.21
149998	0.254363	0.512391	-0.456296	0.414252	-0.202450	-0.054214	-1.00
149999	0.168678	-0.738673	-0.734526	-0.278194	0.104142	0.091720	-1.79

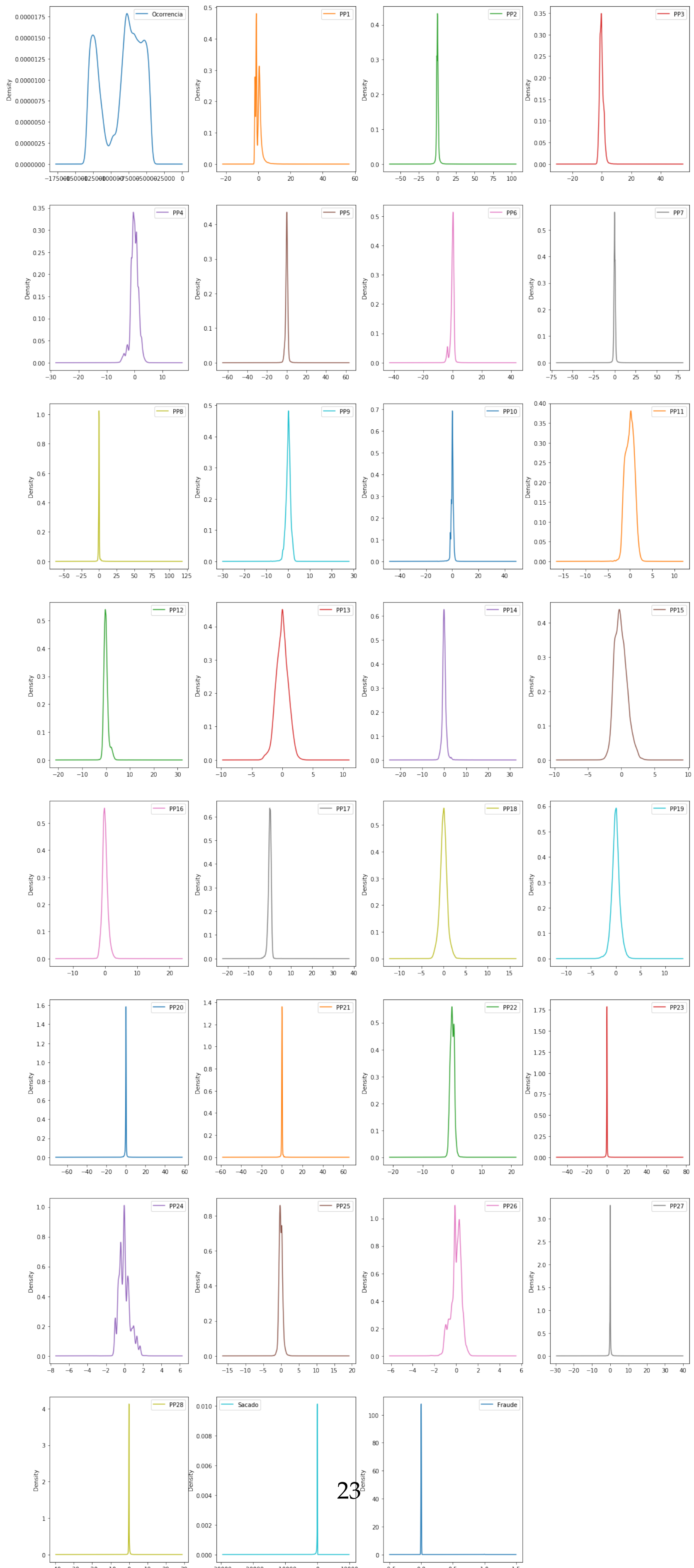
	Fraude
3	0
5	0
6	0
7	0
8	0
11	0
15	0
16	0
18	0
19	0
22	0
26	0
27	0
29	0
31	0
32	0
35	0
37	0
38	0
39	0
40	0

```
41          0
44          0
46          0
48          0
49          0
51          0
54          0
56          0
57          0
...        ...
149943      0
149944      0
149947      0
149948      0
149951      0
149952      0
149954      0
149955      0
149960      0
149962      0
149964      0
149965      0
149967      0
149969      0
149971      0
149972      0
149974      0
149977      0
149979      0
149980      0
149985      0
149986      0
149987      0
149989      0
149991      0
149992      0
149994      0
149997      0
149998      0
149999      0

[84884 rows x 31 columns]
```

Comentário: Não temos linhas duplicadas no dataset, mas temos registros da mesma ocorrência. Verificado que isso não é um problema, pois a mesma ocorrência não possui status de Fraude Sim ou Não. Desta forma, decidi não efetuar a eliminação dessas linhas.

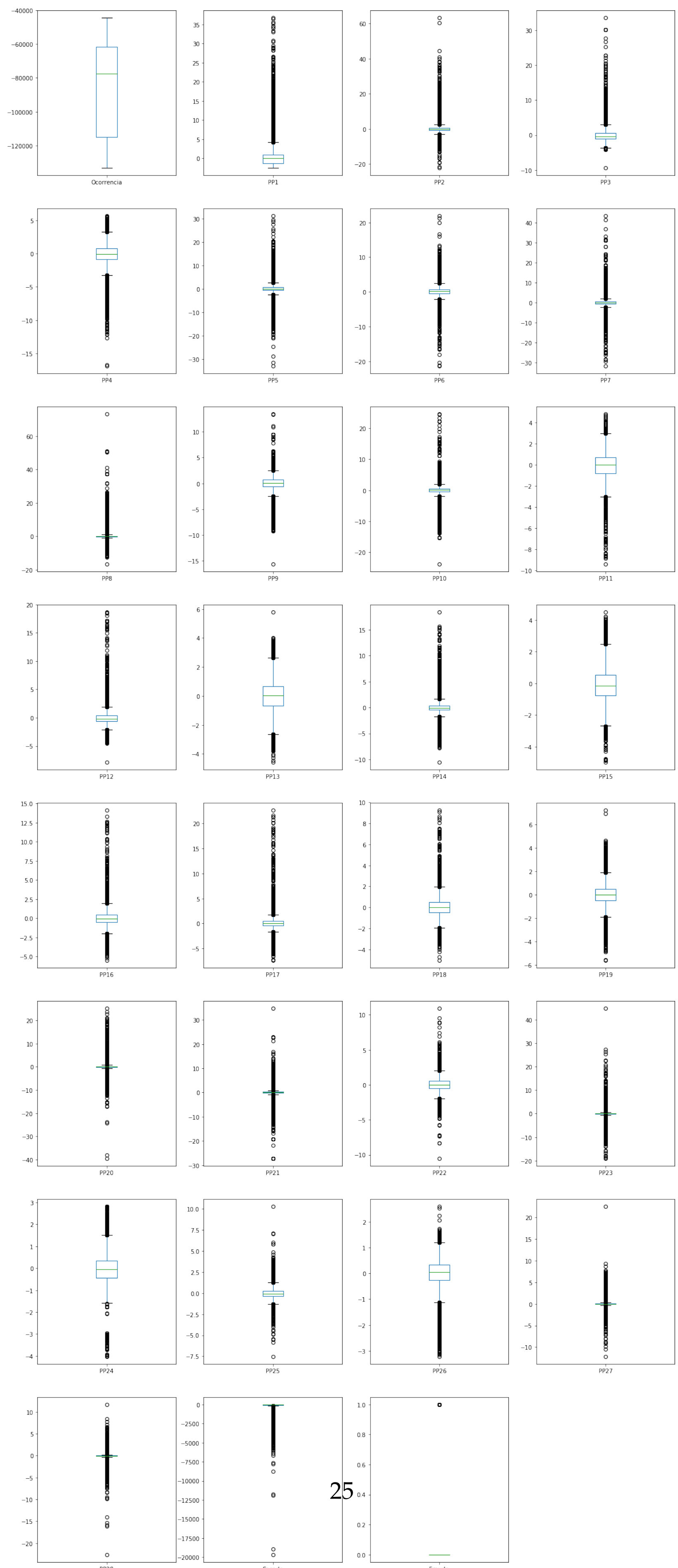
0.2 Gráficos

```
In [19]: #Gráfico de Densidade
         df.plot(kind = 'density', subplots = True, layout = (8,4), sharex = False, figsize=(20
         plt.show()
```


Ocorrencia
PP1
PP2
PP3
PP4
PP5
PP6
PP7
PP8
PP9
PP10
PP11
PP12
PP13
PP14
PP15
PP16
PP17
PP18
PP19
PP20
PP21
PP22
PP23
PP24
PP25
PP26
PP27
PP28
Sacado
Fraude
Density

Comentário: Os dados estão em distribuição normal, como praticamente todas as colunas estão assim, acredito que os dados foram submetidos a técnica de padronização. Isso também prejudica a análise exploratória, o ideal seria ter os dados brutos e após as análises, aplicar as técnicas para criar o modelo de ML.

In [18]:
```
#Box-Plots
df.plot(kind = 'box', subplots = True, layout = (8,4), sharex = False, sharey = False
plt.show()
```


Ocorrencia
PP1
PP2
PP3
PP4
PP5
PP6
PP7
PP8
PP9
PP10
PP11
PP12
PP13
PP14
PP15
PP16
PP17
PP18
PP19
PP20
PP21
PP22
PP23
PP24
PP25
PP26
PP27
PP28
Sacado
Fraude

Comentário: Conforme imagens acima, tudo parece outliers, mas na verdade isso é causado pela normalização dos dados, mesmo assim, conseguimos visualizar algumas colunas com possíveis anomalias, são elas: PP2, PP4, PP8, PP13, PP19, PP20, PP21, PP23, PP25, PP27, PP28, Sacado

In [37]:
```
#Dados de Correlação entre as variáveis, multiplicado por 100 para melhorar a visuali
df.corr(method = 'pearson')*100
```

Out[37]:

	Ocorrencia	PP1	PP2	PP3	PP4	\
Ocorrencia	100.000000	13.067820	-2.572325	-40.544231	-8.026139	
PP1	13.067820	100.000000	1.173272	-3.030647	0.802767	
PP2	-2.572325	1.173272	100.000000	1.667202	0.726745	
PP3	-40.544231	-3.030647	1.667202	100.000000	1.241917	
PP4	-8.026139	0.802767	0.726745	1.241917	100.000000	
PP5	19.598616	-2.613793	-0.240376	-3.747855	2.081759	
PP6	-3.761042	0.234826	1.914754	-3.563867	-0.148190	
PP7	8.558175	1.247073	1.345375	0.991878	0.256024	
PP8	-5.744129	4.480802	-4.356064	6.203969	-1.334295	
PP9	12.647566	0.437889	0.113823	-4.476653	0.146879	
PP10	2.253735	-2.088208	1.977236	-0.548994	-0.245808	
PP11	-13.211170	1.439144	0.370965	-2.136770	-2.355842	
PP12	-14.667502	-3.738015	1.065760	7.907898	3.202273	
PP13	6.839664	1.659506	1.179891	-5.924284	-0.609928	
PP14	1.734045	0.708601	1.781970	-8.229749	2.154841	
PP15	-23.038307	0.344181	2.357420	0.741965	-0.381126	
PP16	0.253492	-0.182399	2.199384	-1.982990	-0.380892	
PP17	-2.378883	0.027837	-1.106038	-2.326712	1.516422	
PP18	8.470933	-0.085209	-0.823344	-1.750222	0.041325	
PP19	1.516400	0.293764	0.287575	0.874008	0.634218	
PP20	-4.826736	-2.313096	-1.801338	-3.718517	-0.008724	
PP21	5.664646	1.016753	-2.246106	1.423048	0.523730	
PP22	16.652607	-1.093254	-0.122071	1.530278	2.200356	
PP23	4.202739	-4.113476	-0.704367	-1.209861	1.277257	
PP24	-2.187605	-0.237412	0.003097	0.607901	0.411375	
PP25	-21.550630	4.181609	-1.782152	-1.404172	-0.762939	
PP26	-6.234330	-1.111493	-0.633893	1.176296	-1.667290	
PP27	0.374262	0.210341	-3.462527	1.894299	-0.924048	
PP28	-0.208598	5.928828	3.905098	0.637131	-1.010220	
Sacado	-0.461526	-23.145794	-54.914261	-21.816828	9.043935	
Fraude	0.254854	9.248126	-8.368848	17.689876	-11.336734	

	PP5	PP6	PP7	PP8	PP9	...	\
Ocorrencia	19.598616	-3.761042	8.558175	-5.744129	12.647566	...	
PP1	-2.613793	0.234826	1.247073	4.480802	0.437889	...	
PP2	-0.240376	1.914754	1.345375	-4.356064	0.113823	...	
PP3	-3.747855	-3.563867	0.991878	6.203969	-4.476653	...	
PP4	2.081759	-0.148190	0.256024	-1.334295	0.146879	...	

PP5	100.000000	2.249919	1.898054	2.078840	3.052313	...
PP6	2.249919	100.000000	-3.260929	0.026700	0.237227	...
PP7	1.898054	-3.260929	100.000000	8.262808	2.383295	...
PP8	2.078840	0.026700	8.262808	100.000000	0.702179	...
PP9	3.052313	0.237227	2.383295	0.702179	100.000000	...
PP10	0.004796	-1.371321	1.353688	3.365958	-0.569145	...
PP11	1.631287	-2.489269	2.626832	-1.204086	-1.809472	...
PP12	-6.118168	1.157007	-3.544052	3.685632	-2.439747	...
PP13	4.342058	-1.773695	2.903592	-0.307129	-1.053506	...
PP14	1.102271	-1.516454	-1.160400	0.765656	-1.558080	...
PP15	0.626852	-2.685122	1.096678	0.218302	-4.752044	...
PP16	2.116417	-1.772958	-1.047258	1.079471	-2.397408	...
PP17	-3.842120	-1.148160	-2.473741	2.964921	-2.541060	...
PP18	0.218825	1.162469	-0.165235	0.658230	-0.368044	...
PP19	1.010855	2.213769	-0.652447	0.742669	1.303960	...
PP20	-5.374920	3.110098	3.680845	-2.011872	0.084727	...
PP21	-0.647679	1.271251	4.796492	0.914486	2.227166	...
PP22	-1.145190	1.292403	-2.562452	0.826322	3.119177	...
PP23	0.365812	-0.226060	0.913256	-1.433278	-0.508558	...
PP24	-0.427903	-1.413964	-0.116141	0.384491	0.074397	...
PP25	-0.374673	0.437165	-1.622634	0.772028	0.239217	...
PP26	-2.037261	0.387136	-1.326974	1.447366	0.809328	...
PP27	5.168961	-2.036120	-1.436795	1.260036	1.396891	...
PP28	-2.487260	1.561270	0.774663	-0.390277	-1.826636	...
Sacado	-36.705345	19.689885	36.954343	-9.498692	-3.626822	...
Fraude	9.002515	2.705548	18.960769	4.048326	7.975801	...

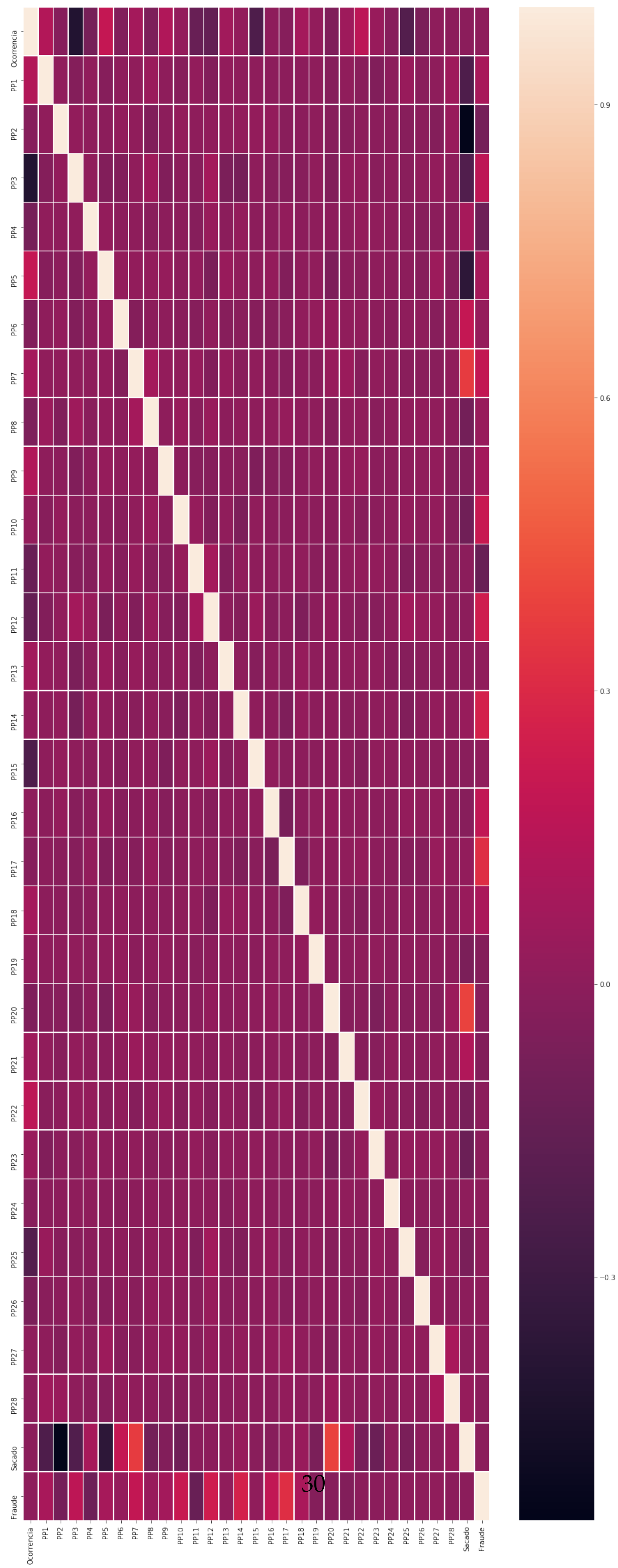
	PP21	PP22	PP23	PP24	PP25	\
Ocorrencia	5.664646	16.652607	4.202739	-2.187605	-21.550630	
PP1	1.016753	-1.093254	-4.113476	-0.237412	4.181609	
PP2	-2.246106	-0.122071	-0.704367	0.003097	-1.782152	
PP3	1.423048	1.530278	-1.209861	0.607901	-1.404172	
PP4	0.523730	2.200356	1.277257	0.411375	-0.762939	
PP5	-0.647679	-1.145190	0.365812	-0.427903	-0.374673	
PP6	1.271251	1.292403	-0.226060	-1.413964	0.437165	
PP7	4.796492	-2.562452	0.913256	-0.116141	-1.622634	
PP8	0.914486	0.826322	-1.433278	0.384491	0.772028	
PP9	2.227166	3.119177	-0.508558	0.074397	0.239217	
PP10	1.187392	-2.092959	-0.723866	0.296538	1.490055	
PP11	0.796999	1.775896	1.886321	0.559964	-4.343651	
PP12	-0.786925	-2.629243	-1.720648	0.750194	6.871799	
PP13	0.802656	0.877592	0.811973	-1.693585	-2.513807	
PP14	-1.084480	-0.537774	0.210562	0.232611	-4.525530	
PP15	-0.303949	-3.275332	0.863680	0.528318	0.671867	
PP16	0.359525	0.260356	-0.240533	-0.406965	2.299358	
PP17	1.359062	2.419932	-0.090111	-0.876355	-2.570389	
PP18	-1.206279	-3.184595	-0.668196	-0.263260	0.166619	
PP19	-0.122583	0.705358	0.898488	-0.446262	0.247676	

```
PP20         -1.792823   -1.154457   -5.674154    0.244934   -2.235963
PP21        100.000000   -2.063154   -1.701877    1.008823   -1.050433
PP22         -2.063154  100.000000    2.140415    0.006823   -1.450116
PP23         -1.701877    2.140415  100.000000    0.245229    2.363738
PP24          1.008823    0.006823    0.245229  100.000000   -0.230022
PP25         -1.050433   -1.450116    2.363738   -0.230022  100.000000
PP26         -0.879770   -3.468213    1.569491   -0.395557   -2.863803
PP27          0.901639   -0.851283    2.163282   -0.796837    2.554276
PP28          1.195937   -0.480740    0.927053   -0.029700    1.017615
Sacado       12.430812   -7.569978  -12.261850    0.372231   -6.547628
Fraude       -3.987550   -0.742457   -0.615783    0.421241    0.584267

                  PP26        PP27        PP28      Sacado      Fraude
Ocorrencia   -6.234330    0.374262   -0.208598   -0.461526    0.254854
PP1          -1.111493    0.210341    5.928828  -23.145794    9.248126
PP2          -0.633893   -3.462527    3.905098  -54.914261   -8.368848
PP3           1.176296    1.894299    0.637131  -21.816828   17.689876
PP4          -1.667290   -0.924048   -1.010220    9.043935  -11.336734
PP5          -2.037261    5.168961   -2.487260  -36.705345    9.002515
PP6           0.387136   -2.036120    1.561270   19.689885    2.705548
PP7          -1.326974   -1.436795    0.774663   36.954343   18.960769
PP8           1.447366    1.260036   -0.390277   -9.498692    4.048326
PP9           0.809328    1.396891   -1.826636   -3.626822    7.975801
PP10          0.416648    1.953034   -1.839924  -10.630216   20.873338
PP11         -1.644519   -1.159104   -1.182398   -0.542591  -13.285001
PP12          3.466660    2.258045   -0.133800   -0.310821   23.865133
PP13         -1.166483   -0.201681   -0.493229    0.083541    1.088100
PP14         -0.884263    1.514736    0.604957    3.394939   26.499645
PP15         -0.119602    0.149468   -0.245867   -1.484415    1.302661
PP16          1.197527    2.086586    0.013925   -1.726129   18.383480
PP17         -1.784553    3.722385    1.454492    1.734242   31.661802
PP18         -0.934918    1.249689    0.736421    4.067615   10.105426
PP19          0.472194   -0.382550   -0.423083   -6.214709   -3.400318
PP20         -0.189203   -2.842810    4.680753   39.261615   -1.723442
PP21         -0.879770    0.901639    1.195937   12.430812   -3.987550
PP22         -3.468213   -0.851283   -0.480740   -7.569978   -0.742457
PP23          1.569491    2.163282    0.927053  -12.261850   -0.615783
PP24         -0.395557   -0.796837   -0.029700    0.372231    0.421241
PP25         -2.863803    2.554276    1.017615   -6.547628    0.584267
PP26        100.000000    0.764819    0.121191   -0.139791    0.313836
PP27          0.764819  100.000000    9.718450   -0.115842    0.896319
PP28          0.121191    9.718450  100.000000    2.831210   -0.943733
Sacado       -0.139791   -0.115842    2.831210  100.000000   -0.751516
Fraude        0.313836    0.896319   -0.943733   -0.751516  100.000000

[31 rows x 31 columns]
```

```
In [36]: # Correlação em gráfico de HeatMap
```

```python
f, ax = plt.subplots(figsize=(15, 40))
sns.heatmap(df.corr(method = 'pearson'),linewidths=.5, ax=ax)
```

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37dbc2310>


Ocorrencia
PP1
PP2
PP3
PP4
PP5
PP6
PP7
PP8
PP9
PP10
PP11
PP12
PP13
PP14
PP15
PP16
PP17
PP18
PP19
PP20
PP21
PP22
PP23
PP24
PP25
PP26
PP27
PP28
Sacado
Fraude
0.9
0.6
0.3
0.0
−0.3

```
In [31]: # Correlação com a variável Target ordenado, feita a multiplicacao por 100 para melho
         df.drop("Fraude", axis=1).apply(lambda x: x.corr(df.Fraude) * 100).sort_values()

Out[31]: PP11          -13.285001
         PP4           -11.336734
         PP2            -8.368848
         PP21           -3.987550
         PP19           -3.400318
         PP20           -1.723442
         PP28           -0.943733
         Sacado         -0.751516
         PP22           -0.742457
         PP23           -0.615783
         Ocorrencia      0.254854
         PP26            0.313836
         PP24            0.421241
         PP25            0.584267
         PP27            0.896319
         PP13            1.088100
         PP15            1.302661
         PP6             2.705548
         PP8             4.048326
         PP9             7.975801
         PP5             9.002515
         PP1             9.248126
         PP18           10.105426
         PP3            17.689876
         PP16           18.383480
         PP7            18.960769
         PP10           20.873338
         PP12           23.865133
         PP14           26.499645
         PP17           31.661802
         dtype: float64
```

Comentário: Feita a análise de correlação entre as colunas e a análise com a variável Target, as com maiores e menores valores serão utilizados no modelo de ML, as com valores próximos de 0 serão desconsiderados. Ainda iremos analisar a colinearidade para fecharmos a relação de variáveis.

```
In [46]: # Tratamento de identificação de colunas com alta colinearidade
         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
```

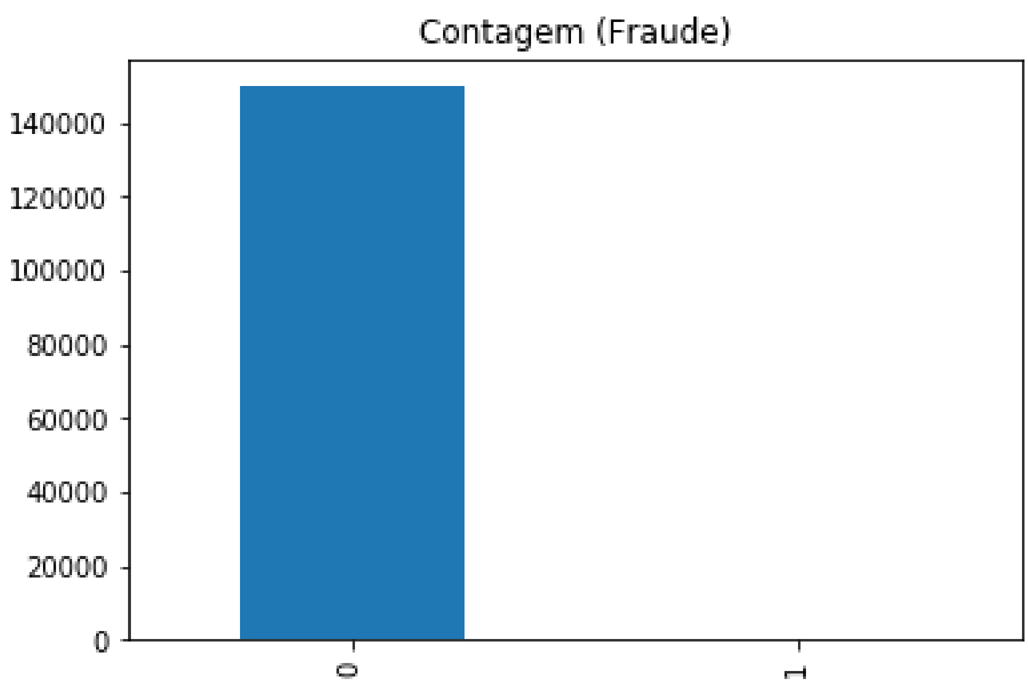
Out[46]: []

Comentário: Não foi identificado alta colinearidade entre as colunas, desta forma, iremos seguir selecionando as colunas pela correlação através da técnica de Feature selection com Random Forest.

1 Problema da base desbalanceada

In [49]:
```
target_count = df.Fraude.value_counts()
print('Não Fraude:', target_count[0])
print('Fraude:', target_count[1])
print('Proporção:', round(target_count[0] / target_count[1], 2), ': 1')
target_count.plot(kind='bar', title='Contagem (Fraude)',color = ['#1F77B4', '#FF7F0E']
```

```
Não Fraude: 149763
Fraude: 237
Proporção: 631.91 : 1
```



Comentário: Devido a base desbalanceada, iremos utilizar o SMOTE do pacte imblearn

In [126]:
```
#!pip install imblearn
import collections
```

```
from imblearn.over_sampling import SMOTE, ADASYN
y = df.Fraude
del df['Fraude']
data_o, target_o = SMOTE().fit_sample(df, y)
```

In [81]:
```
data_o.shape
```

Out[81]: (299526, 30)

In [82]:
```
target_o.shape
```

Out[82]: (299526,)

In [83]:
```
collections.Counter(target_o)
```

Out[83]: Counter({0: 149763, 1: 149763})

2 Feature selection

In [127]:
```
seed = 1313

# Feature selection com Random Forest
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectFromModel

clf = RandomForestClassifier(random_state=seed)
selector = clf.fit(data_o, target_o)
fs = SelectFromModel(selector, prefit=True)

data_o_new = fs.transform(data_o)


print(data_o_new.shape, target_o.shape)
```

```
(299526, 6) (299526,)
```

In [86]:
```
# 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(data_o_new, columns=new_features)
df_selection['Fraude'] = target_o

df_selection.head(10)
```

```
Out[86]:         PP4      PP10      PP11      PP12      PP14      PP16      PP17  \
         0  0.251323 -0.521274  0.357440 -1.229859  1.034054  1.322059  0.203699
         1  1.122846  1.176820 -1.005574 -1.315100 -0.038456  0.076187  0.434745
         2 -0.428284  0.138069  0.340559 -0.600446  1.186472 -0.294058 -1.185046
         3 -0.929524 -0.093204 -1.695905 -0.858794 -0.721720  0.324972  0.092153
         4  0.374010  1.774507  1.130069 -0.529619 -0.539129 -1.296754  0.275466
         5 -1.209587  0.144356  1.371141  0.337040 -0.121332 -0.041093  0.216084
         6 -0.871200 -0.134853 -1.272085 -0.975338 -0.548332 -0.037385  0.530802
         7  0.341794 -0.104135  0.261161 -0.787753  0.236741  1.002734  0.706106
         8  1.429721  1.045694 -0.977857 -1.426133 -0.114068  0.541163  0.146142
         9  1.592340  0.015361 -1.018212 -0.045338 -0.180483  0.890010  0.487856

            Fraude
         0       0
         1       0
         2       0
         3       0
         4       0
         5       0
         6       0
         7       0
         8       0
         9       0
```

Comentário: As melhores colunas são as mesmas com maior correlação positiva ou negativa demonstrado acima.

3 Dados de Treino e Teste

```
In [128]: #Gerando dados de Treino e de Teste para os modelos, 70% para treinamento e 30% para
          array = df_selection.values
          X = array[:,0:6]
          Y = df_selection.Fraude.values

          from sklearn.model_selection import train_test_split

          X_treino, X_teste, y_treino, y_teste = train_test_split(X, Y, test_size = 0.30, rand
```

```
In [90]: X_treino.shape , X_teste.shape , y_treino.shape , y_teste.shape
```

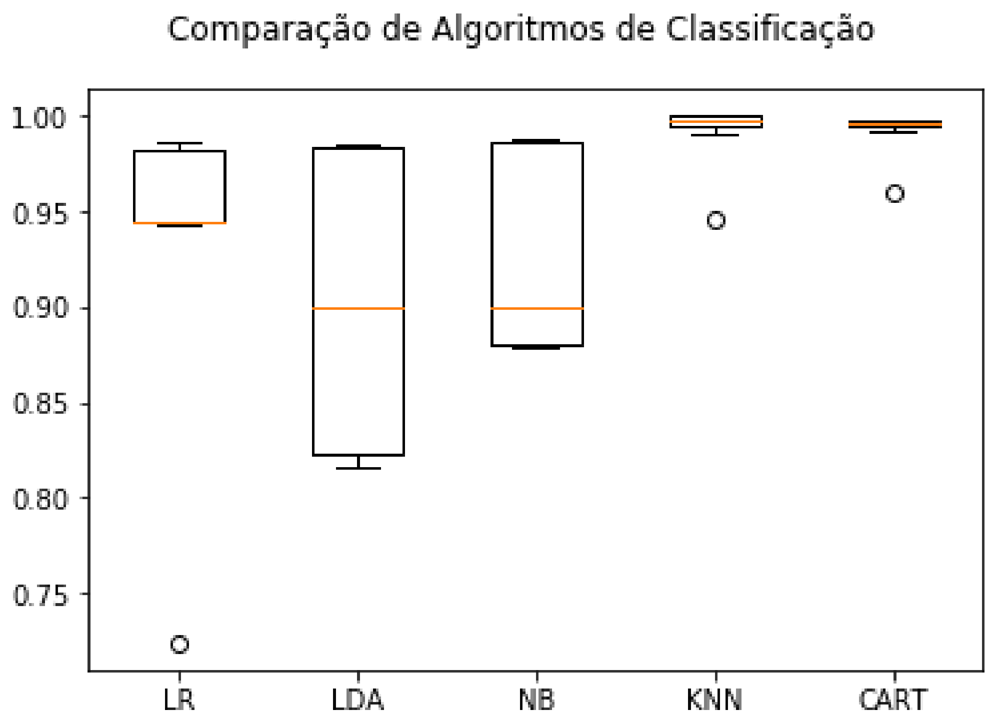
```
Out[90]: ((209668, 6), (89858, 6), (209668,), (89858,))
```

4 Escolhendo os modelos de Classificação

```
In [92]: from sklearn.model_selection import KFold
         from sklearn.model_selection import cross_val_score
         from sklearn.linear_model import LogisticRegression
```

```python
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

# Preparando a lista de modelos
modelos = []
modelos.append(('LR', LogisticRegression()))
modelos.append(('LDA', LinearDiscriminantAnalysis()))
modelos.append(('NB', GaussianNB()))
modelos.append(('KNN', KNeighborsClassifier()))
modelos.append(('CART', DecisionTreeClassifier()))

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

# Avaliando cada modelo em um loop
resultados = []
nomes = []

for nome, modelo in modelos:
    kfold = KFold(n_splits = num_folds, random_state = seed)
    cv_results = cross_val_score(modelo, X, Y, cv = kfold, scoring = 'accuracy')
    resultados.append(cv_results)
    nomes.append(nome)
    msg = "%s: %f (%f)" % (nome, cv_results.mean(), cv_results.std())
    print(msg)

# Boxplot para comparar os algoritmos
fig = plt.figure()
fig.suptitle('Comparação de Algoritmos de Classificação')
ax = fig.add_subplot(111)
plt.boxplot(resultados)
ax.set_xticklabels(nomes)
plt.show()
```

```
LR: 0.937401 (0.073535)
LDA: 0.901527 (0.080682)
NB: 0.926393 (0.050433)
KNN: 0.992081 (0.015867)
CART: 0.992535 (0.011235)
```



Comentário: A acurácia de mais de 99% indica possível overfiting, irei utilizar os métodos Ensemble para evitá-lo.

5 Métodos com Ensemble

```
In [94]: from sklearn.ensemble import GradientBoostingClassifier

         # Definindo o número de trees
         num_trees = 100

         # Separando os dados em folds
         kfold = KFold(num_folds, True, random_state = seed)

         # Criando o modelo
         modelo_gradiente = GradientBoostingClassifier(n_estimators = num_trees, random_state =

         # Cross Validation
         resultado_gradiente = cross_val_score(modelo_gradiente, X, Y, cv = kfold)

         # Print do resultado
         print("Acurácia GradientBoosting: %.3f" % (resultado_gradiente.mean() * 100))
```

```
Acurácia: 97.919
```

In [101]:
```
#!pip install xgboost
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score

# Criando o modelo
modelo_xgb = XGBClassifier()
# Treinando o modelo
modelo_xgb.fit(X_treino, y_treino)
# Pront do modelo
print(modelo_xgb)
# Fazendo previsões
y_pred_xgb = modelo_xgb.predict(X_teste)
y_pred_xgb_prob = modelo_xgb.predict_proba(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: %.2f%%" % (accuracy * 100.0))
```

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=None, n_estimators=100, 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)
Acurácia XGB: 97.80%
```

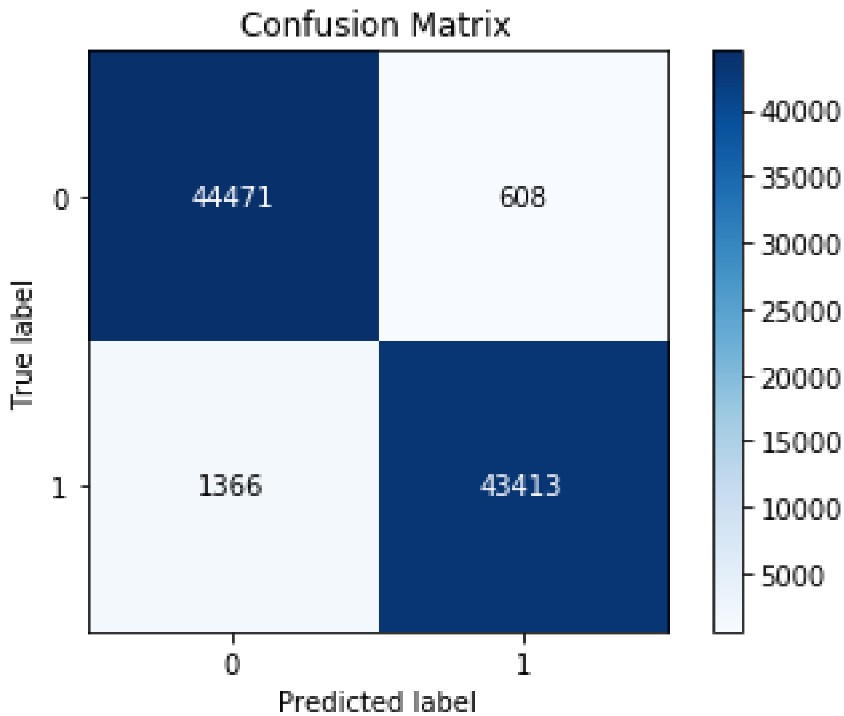
Comentário: o Modelo XGBClassifier obteve a melhor acurária => 97.80%, decidi utilizá-lo e apresentá-lo como modelo final.

5.1 Confusion Matrix e Curva ROC do modelo XGBClassifier

In [109]:
```
#!pip install scikit-plot
from sklearn.metrics import classification_report, confusion_matrix
import scikitplot as skplt

print(confusion_matrix(y_teste, previsoes_xgb))
print(classification_report(y_teste, previsoes_xgb))

skplt.metrics.plot_confusion_matrix(y_true=y_teste, y_pred=previsoes_xgb)
plt.show()
```

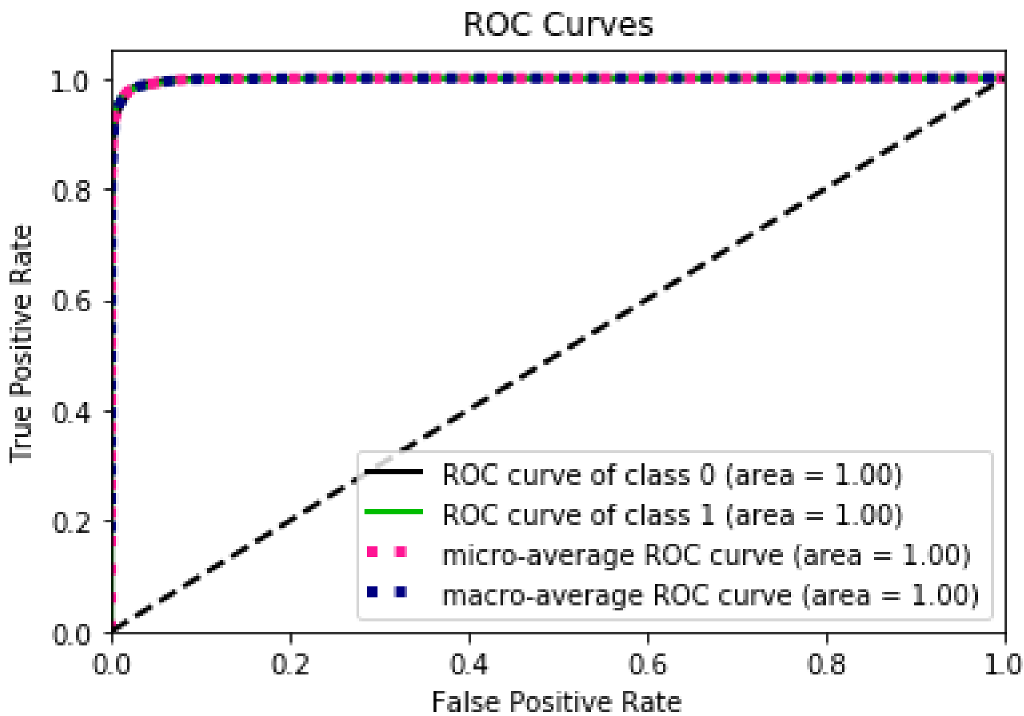
```
[[44471   608]
 [ 1366 43413]]
              precision    recall  f1-score   support
```

```
           0       0.97      0.99      0.98     45079
           1       0.99      0.97      0.98     44779

    accuracy                           0.98     89858
   macro avg       0.98      0.98      0.98     89858
weighted avg       0.98      0.98      0.98     89858
```



Comentário: Através da Matriz de Confusão conseguimos visualizar melhor que o modelo alcançou um resultado de 98% de acerto nas Fraudes apresentadas nesse dataset e que os 98% ficaram balanceados nas categorias de Não Fraude e Fraude.

```
In [110]: skplt.metrics.plot_roc(y_teste, y_pred_xgb_prob)
          plt.show()
```



6 Melhorando o modelo - HyperParametros

In [131]:
```
# 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(150, 1000),
              'learning_rate': stats.uniform(0.01, 0.6),
              'subsample': stats.uniform(0.3, 0.9),
              'max_depth': [3, 4, 5, 6, 7, 8, 9],
              'min_child_weight': [1, 2, 3, 4]
             }

numFolds = 5
kfold = KFold(num_folds, True, random_state = seed)

rsearch = RandomizedSearchCV(modelo_inicial,
                             param_distributions = param_dist,
                             cv = kfold,
                             n_iter = 2,
                             scoring = 'roc_auc',
```

```
                                error_score = 0,
                                verbose = 3,
                                n_jobs = -1)

rsearch.fit(X, Y)

# Print dos resultados
print("Acurácia: %.3f" % (rsearch.best_score_ * 100))
print("Melhores Parâmetros do Modelo:\n", rsearch.best_estimator_)
```

```
Fitting 10 folds for each of 2 candidates, totalling 20 fits


[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  20 out of  20 | elapsed: 27.1min remaining:    0.0s
[Parallel(n_jobs=-1)]: Done  20 out of  20 | elapsed: 27.1min finished


Acurácia: 99.991
Melhores Parâmetros do Modelo:
 XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.13447815907134084, max_delta_step=0, max_depth=5,
              min_child_weight=3, missing=None, n_estimators=522, 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=0.9651070749316022, verbosity=1)
```

7 Gerando o modelo final com a otimização

In [141]:
```
# 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=0.05, max_delta_step=0, max_depth=5,
              min_child_weight=3, missing=None, n_estimators=500, 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)

# Treinando o modelo
modelo_xgb_final.fit(X_treino, y_treino)
# Fazendo previsões
y_pred_xgb = modelo_xgb.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: %.2f%%" % (accuracy * 100.0))
```

```
Acurácia XGB Final: 97.80%
```

Comentário: Mesmo com a otimização, a acurária foi a mesma, 97,80%

```
In [142]: import pickle
          # Salvando o modelo
          arquivo = 'modelo_analiseFraude_final.sav'
          pickle.dump(modelo_xgb_final, open(arquivo, 'wb'))
          print("Modelo salvo!")
```

```
Modelo salvo!
```

7.1 FIM

7.2 OBRIGADO