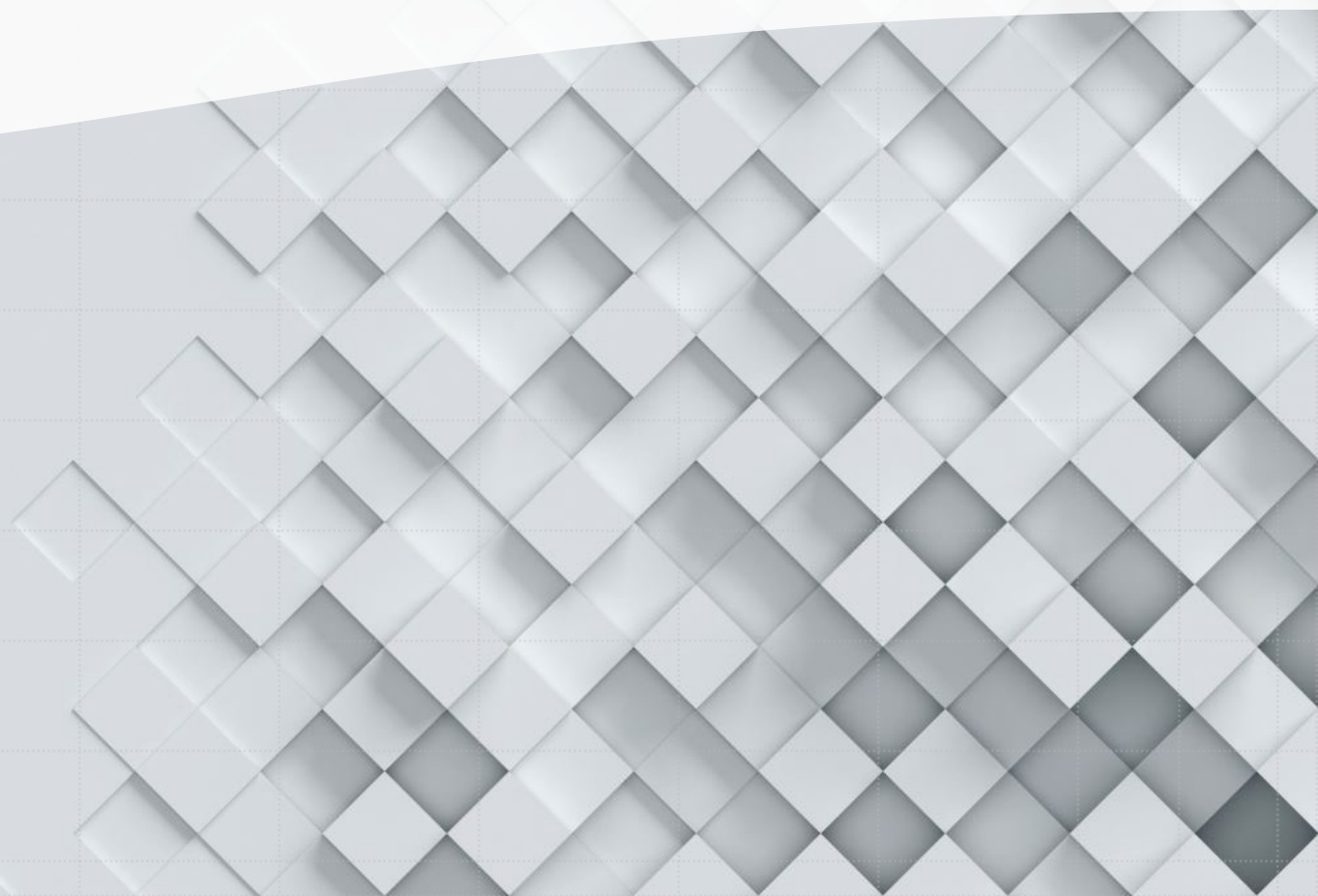


FRAUDES EM TRANSAÇÕES DE CARTÃO DE CRÉDITO

Trabalho Final CDD 2021-2

Bruno Eduardo Farias



RELEMBRANDO: O DATASET

A	B	C	D	E	F	G	H	I	J	K	L
Time	V1	V2	V3	V4	...	V25	V26	V27	V28	Amount	Class
0	-1,36E+13	-0.07278117	2,54E+14	1,38E+14	...	0.1285393	-0.1891148	0.1335583	-0.0210530	149.62	0
0	1,19E+14	0.266150712	0.1664801	0.4481540	...	0.1671704	0.1258945	-0.0089830	0.0147241	2.69	0
1	-1,36E+14	-1,34E+14	1,77E+14	0.3797795	...	-0.3276418	-0.1390965	-0.0553527	-0.0597518	378.66	0
1	-0.96627171157	-0.18522600	1,79E+14	-0.8632912	...	0.6473760	-0.2219288	0.0627228	0.0614576	123.5	0
2	-1,16E+14	0.877736754	1,55E+12	0.4030339	...	-0.2060095	0.5022922	0.2194222	0.2151531	69.99	0
2	-0.42596588441	0.960523044	1,14E+14	-0.1682520	...	-0.2327938	0.1059147	0.2538442	0.0810802	3.67	0
4	1,23E+14	0.141003507	0.0453707	1,20E+14	...	0.7501369	-0.2572368	0.0345074	0.0051677	4.99	0
7	-0.64426944234	1,42E+14	1,07E+13	-0.4921990	...	-0.4152665	-0.0516342	-1,21E+14	-1,09E+14	40.8	0
7	-0.89428608220	0.286157196	-0.1131922	-0.2715265	...	0.3732046	-0.3841573	0.0117473	0.1424043	93.2	0
9	-0.33826175242	1,12E+14	1,04E+14	-0.2221872	...	-0.0697330	0.0941988	0.2462193	0.0830756	3.68	0
10	1,45E+14	-1,18E+14	0.9138598	-1,38E+14	...	0.2513673	-0.1294775	0.0428498	0.0162532	7.8	0
10	0.384978215180	0.616109459	-0.8742997	-0.0940186	...	-0.7673148	-0.4922082	0.0424724	-0.0543373	9.99	0
10	1,25E+12	-1,22E+14	0.3839301	-1,23E+14	...	0.1611345	-0.3549900	0.0264155	0.0424220	121.5	0
11	1,07E+13	0.287722129	0.8286127	2,71E+14	...	0.5482647	0.1040941	0.0214910	0.0212933	27.5	0

- Dataset com transações de cartão de crédito capturadas durante dois dias na Europa em 2013
- CSV

RELEMBRANDO: O DATASET

A	B	C	D	E	F	G	H	I	J	K	L
Time	V1	V2	V3	V4	...	V25	V26	V27	V28	Amount	Class
0	-1,36E+13	-0.07278117	2,54E+14	1,38E+14	...	0.1285393	-0.1891148	0.1335583	-0.0210530	149.62	0
0	1,19E+14	0.266150712	0.1664801	0.4481540	...	0.1671704	0.1258945	-0.0089830	0.0147241	2.69	0
1	-1,36E+14	-1,34E+14	1,77E+14	0.3797795	...	-0.3276418	-0.1390965	-0.0553527	-0.0597518	378.66	0
1	-0.96627171157	-0.18522600	1,79E+14	-0.8632912	...	0.6473760	-0.2219288	0.0627228	0.0614576	123.5	0
2	-1,16E+14	0.877736754	1,55E+12	0.4030339	...	-0.2060095	0.5022922	0.2194222	0.2151531	69.99	0
2	-0.42596588441	0.960523044	1,14E+14	-0.1682520	...	-0.2327938	0.1059147	0.2538442	0.0810802	3.67	0
4	1,23E+14	0.141003507	0.0453707	1,20E+14	...	0.7501369	-0.2572368	0.0345074	0.0051677	4.99	0
7	-0.64426944234	1,42E+14	1,07E+13	-0.4921990	...	-0.4152665	-0.0516342	-1,21E+14	-1,09E+14	40.8	0
7	-0.89428608220	0.286157196	-0.1131922	-0.2715265	...	0.3732046	-0.3841573	0.0117473	0.1424043	93.2	0
9	-0.33826175242	1,12E+14	1,04E+14	-0.2221872	...	-0.0697330	0.0941988	0.2462193	0.0830756	3.68	0
10	1,45E+14	-1,18E+14	0.9138598	-1,38E+14	...	0.2513673	-0.1294775	0.0428498	0.0162532	7.8	0
10	0.384978215180	0.616109459	-0.8742997	-0.0940186	...	-0.7673148	-0.4922082	0.0424724	-0.0543373	9.99	0
10	1,25E+12	-1,22E+14	0.3839301	-1,23E+14	...	0.1611345	-0.3549900	0.0264155	0.0424220	121.5	0
11	1,07E+13	0.287722129	0.8286127	2,71E+14	...	0.5482647	0.1040941	0.0214910	0.0212933	27.5	0

- Numéricos
- 284.807 transações, 492 fraudes – (0,172%) – desbalanceado

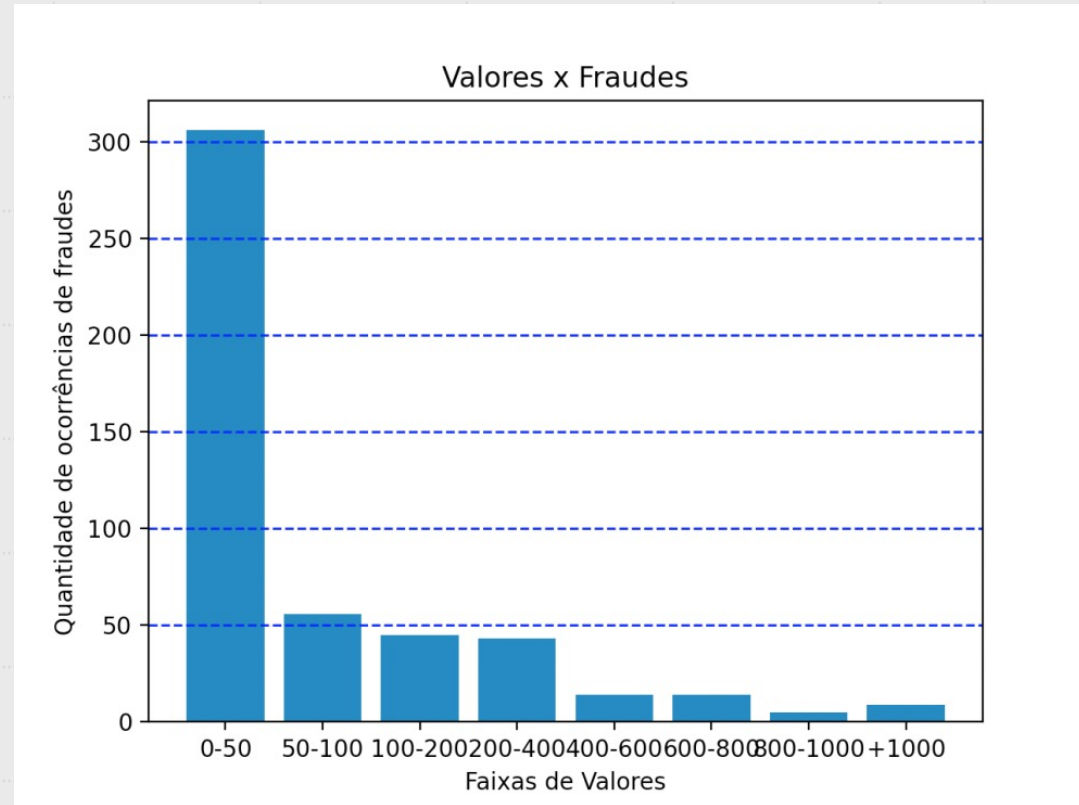
RELEMBRANDO: O DATASET

A	B	C	D	E	F	G	H	I	J	K	L
Time	V1	V2	V3	V4	...	V25	V26	V27	V28	Amount	Class
0	-1,36E+13	-0.07278117	2,54E+14	1,38E+14	...	0.1285393	-0.1891148	0.1335583	-0.0210530	149.62	0
0	1,19E+14	0.266150712	0.1664801	0.4481540	...	0.1671704	0.1258945	-0.0089830	0.0147241	2.69	0
1	-1,36E+14	-1,34E+14	1,77E+14	0.3797795	...	-0.3276418	-0.1390965	-0.0553527	-0.0597518	378.66	0
1	-0.96627171157	-0.18522600	1,79E+14	-0.8632912	...	0.6473760	-0.2219288	0.0627228	0.0614576	123.5	0
2	-1,16E+14	0.877736754	1,55E+12	0.4030339	...	-0.2060095	0.5022922	0.2194222	0.2151531	69.99	0
2	-0.42596588441	0.960523044	1,14E+14	-0.1682520	...	-0.2327938	0.1059147	0.2538442	0.0810802	3.67	0
4	1,23E+14	0.141003507	0.0453707	1,20E+14	...	0.7501369	-0.2572368	0.0345074	0.0051677	4.99	0
7	-0.64426944234	1,42E+14	1,07E+13	-0.4921990	...	-0.4152665	-0.0516347	-1,21E+14	-1,09E+14	40.8	0
7	-0.89428608220	0.286157196	-0.1131925	-0.2715265	...	0.3732046	-0.3841573	0.0117473	0.1424043	93.2	0
9	-0.33826175242	1,12E+14	1,04E+14	-0.2221872	...	-0.0697330	0.0941988	0.2462193	0.0830756	3.68	0
10	1,45E+14	-1,18E+14	0.9138598	-1,38E+14	...	0.2513673	-0.1294779	0.0428498	0.0162532	7.8	0
10	0.384978215180	0.616109459	-0.8742997	-0.0940186	...	-0.7673148	-0.4922082	0.0424724	-0.0543373	9.99	0
10	1,25E+12	-1,22E+14	0.3839301	-1,23E+14	...	0.1611345	-0.3549900	0.0264155	0.0424220	121.5	0
11	1,07E+13	0.287722129	0.8286127	2,71E+14	...	0.5482647	0.1040941	0.0214910	0.0212933	27.5	0

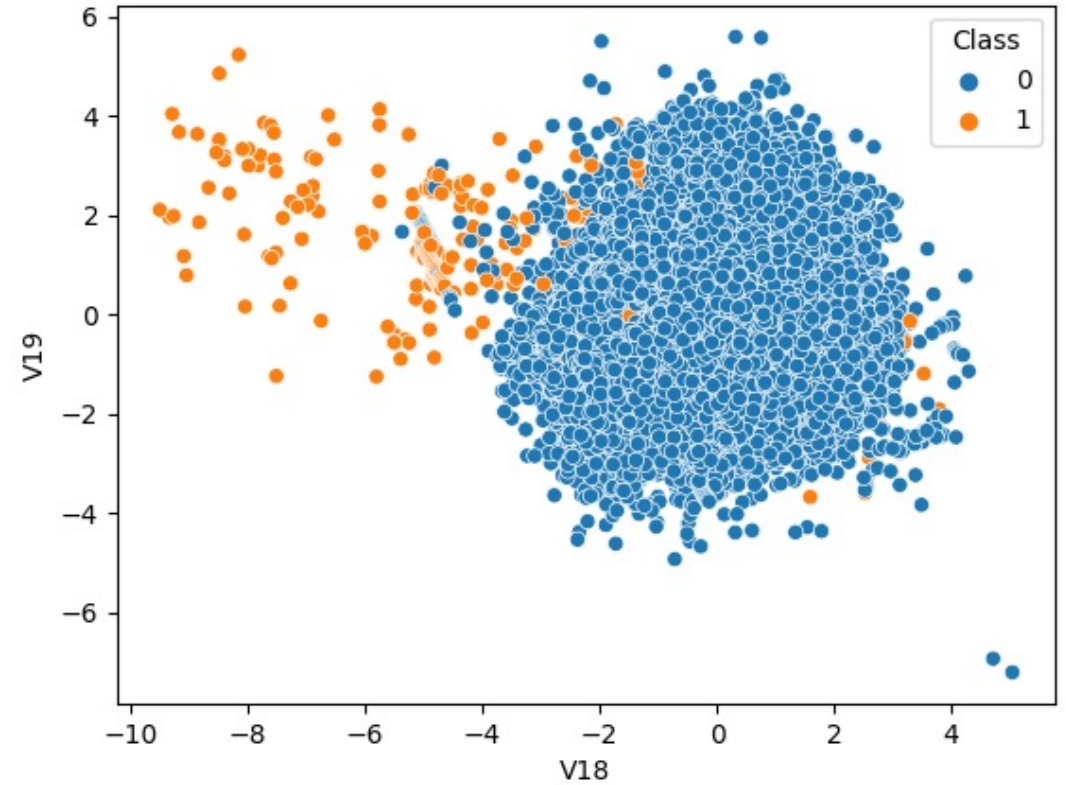
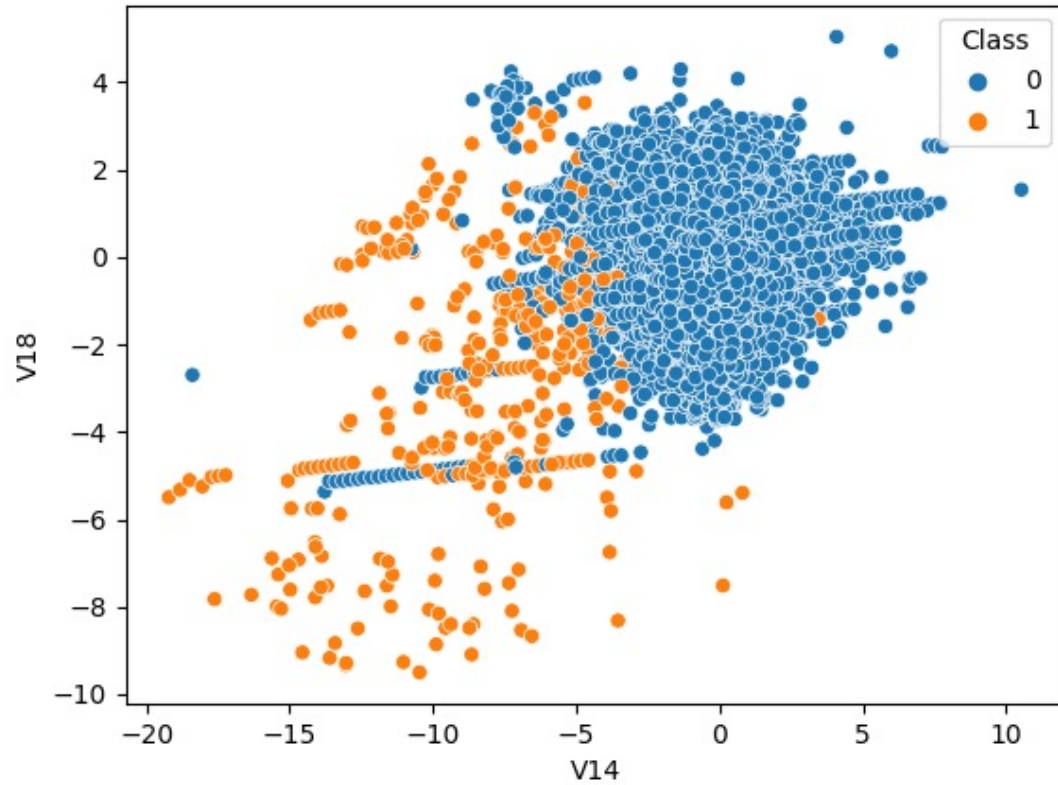
- Time: tempo passado desde a primeira transação capturada (segundos)
- V1-V28: não informado por segurança; uso de PVA
- Amount: valor da transação
- Class: label 1 indica fraude

RELEMBRANDO: O DATASET

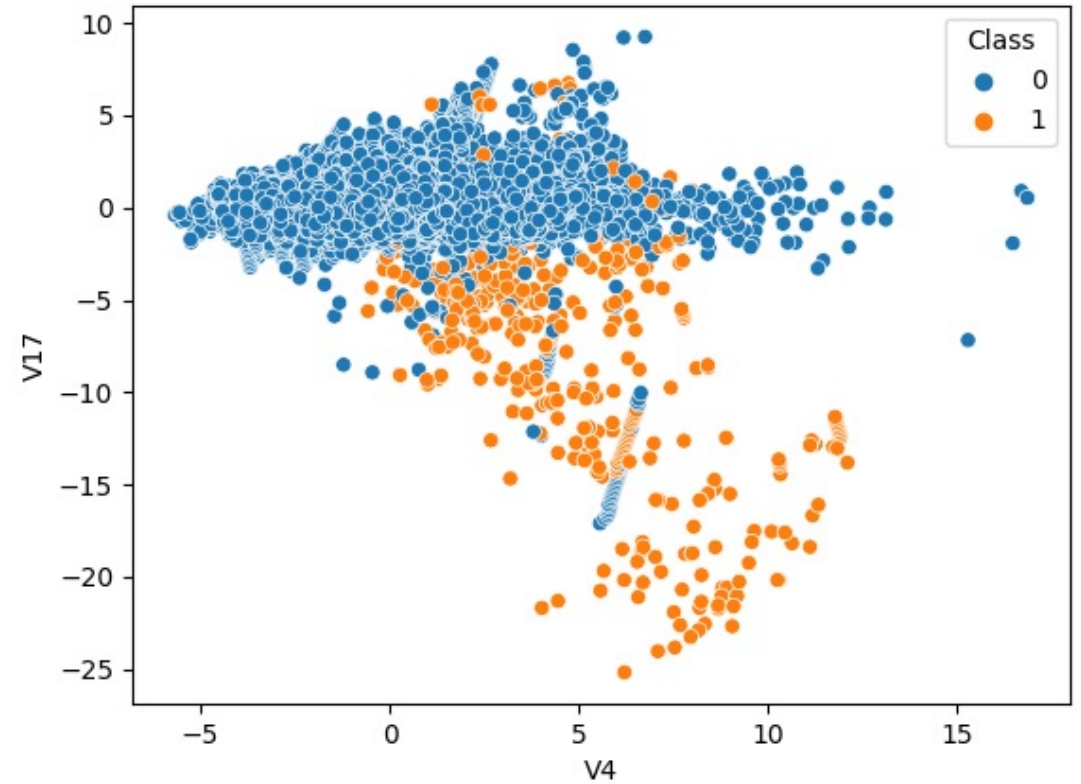
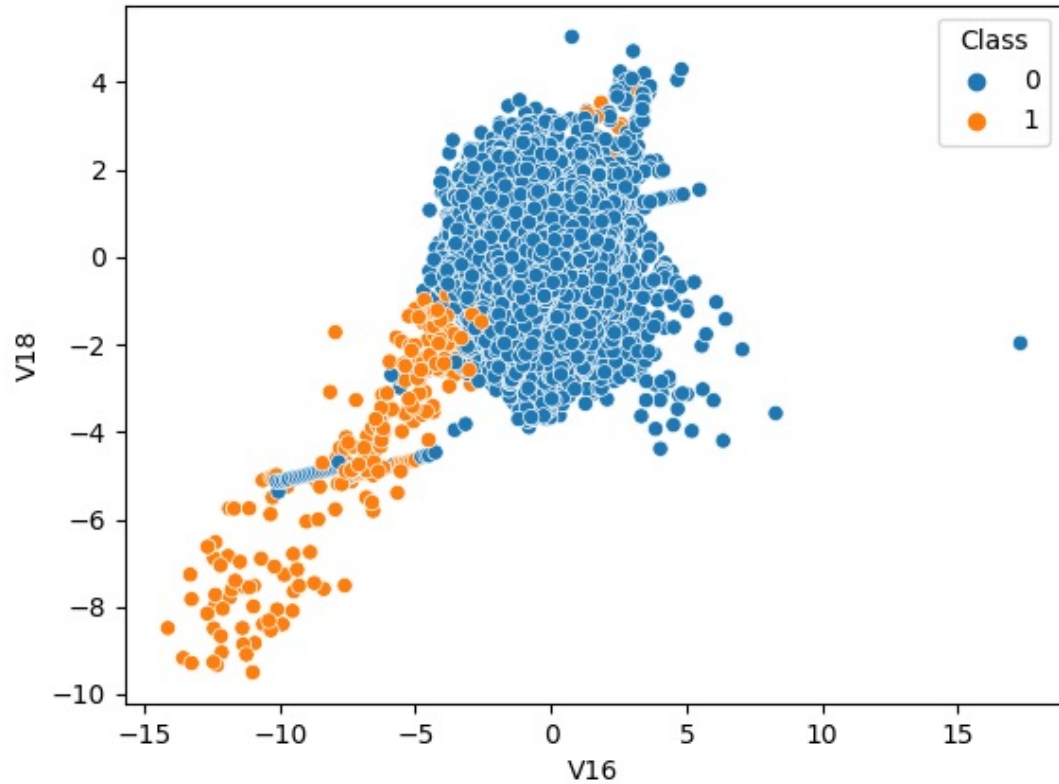
- Menor valor com fraude: 0
- Maior valor com fraude: 2125.87
- Valores comuns nas fraudes: 1.0 (113), 0.77 (10), 0.76 (17), 99.99 (28), 0.01 (5), etc...



RELEMBRANDO: SCATTERPLOTS

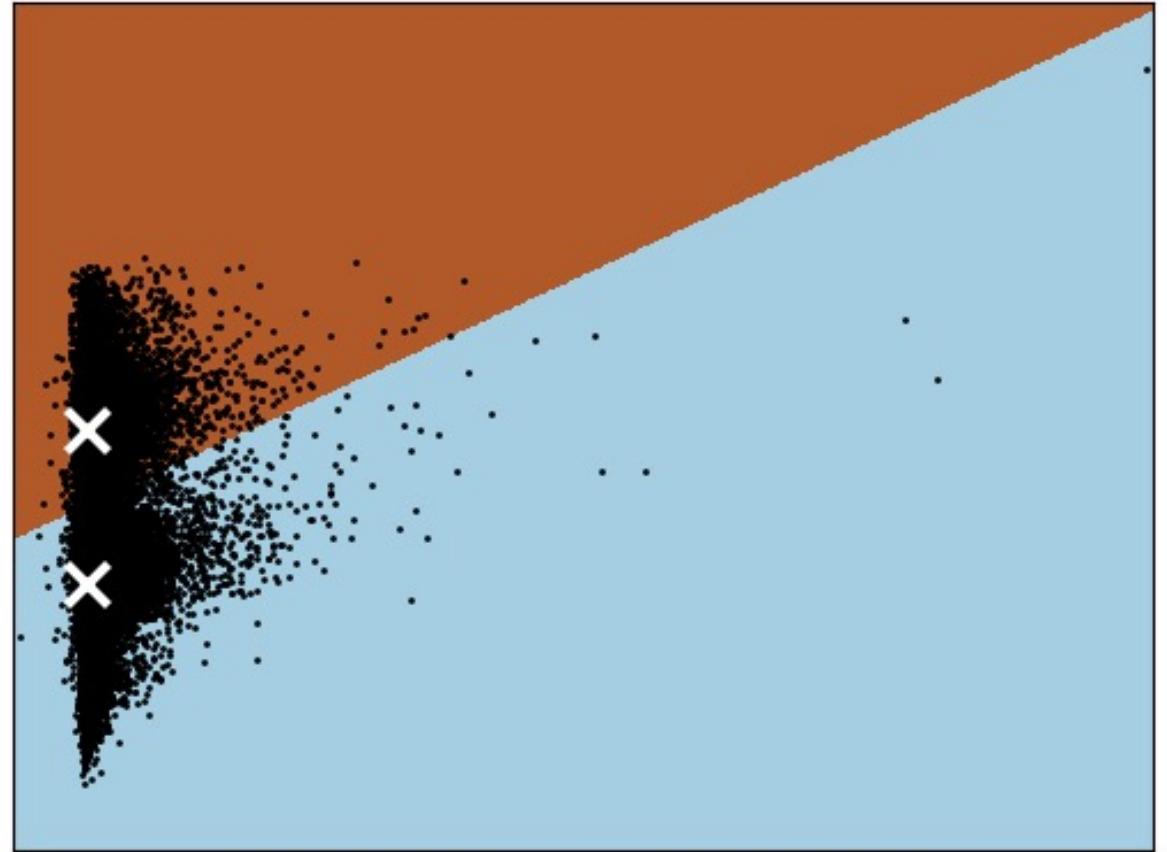


RELEMBRANDO: SCATTERPLOTS



RELEMBRANDO: CLUSTERING

- Kmeans com 2 clusters
- PCA 2





CLASSIFICADORES



1º Passo

Divisão 80-20

Dividir o dataset original em 80% para
treino e seleção dos classificadores e
20% para futuros testes

DIVISÃO 80-20

20%

56.962
amostras

V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
4,00064E+16	5,14868E+15	-1,33376E+16	3,63301E+14	1,36996E+16	9,22784E+14	4,2784E+14	-3,0477E+14	7,68232E+15	5,50667E+14	-6,20484E+13	2,142E+14	193115	0
-2,09566E+14	-9,78939E+13	1,22794E+16	1,79664E+16	2,01934E+16	5,39678E+14	-2,94339E+14	-4,2243E+14	9,11863E+15	-5,49807E+14	-2,63641E+14	-1,39366E+14	150	0
-8,06705E+14	1,61993E+16	2,53761E+14	-6,78089E+14	3,15466E+14	1,20626E+16	1,19775E+13	-3,01625E+14	-1,94477E+14	2,19802E+14	-5,08428E+14	-5,67286E+14	579	0
-4,70435E+16	6,41792E+14	8,41954E+15	2,85741E+16	1,20934E+14	9,80113E+15	-7,58594E+15	3,9722E+14	7,4111E+15	5,20429E+16	-1,03124E+16	-8,17433E+15	10	0
-1,05335E+14	-4,87692E+16	7,93033E+14	-1,00361E+16	3,43785E+14	9,9884E+15	1,05675E+16	-3,91592E+16	-1,93368E+14	-7,75114E+14	1,61091E+14	-6,47951E+14	110	0
-7,81724E+15	-3,01451E+14	8,62865E+15	8,7841E+14	-4,13376E+13	-1,35127E+16	4,82015E+14	-4,72404E+14	1,81792E+14	6,49924E+14	-1,10835E+14	-5,98969E+12	5193	0
1,25943E+14	-6,73457E+14	2,66407E+16	-1,14703E+16	-5,80831E+15	-1,18163E+14	-7,09135E+14	-8,54545E+15	4,50985E+14	-5,01518E+14	4,13832E+14	2,63727E+16	9509	0
-6,76848E+14	1,36351E+14	2,93361E+14	2,28362E+14	3,8556E+16	1,3814E+16	-5,69972E+14	2,62695E+14	-1,22311E+14	3,4129E+14	2,31323E+16	2,61094E+16	184	0
3,82141E+14	7,71637E+15	-6,26341E+14	7,07067E+14	-1,89636E+14	-6,6661E+15	-2,53419E+16	6,55828E+14	1,98955E+16	1,07348E+13	-1,1786E+16	3,67922E+14	31496	0
1,27576E+16	-6,49275E+14	-1,62935E+14	2,00577E+14	3,20389E+14	9,30146E+15	-2,90108E+14	-2,47866E+14	1,09315E+13	6,84158E+14	-2,41502E+14	-7,69783E+14	818	0
8,31443E+14	-4,73347E+14	1,19012E+16	2,73799E+14	-2,60547E+14	-2,95255E+14	-1,80459E+14	-4,36539E+14	4,94649E+14	-2,83738E+13	-1,12828E+14	3,50745E+14	9801	1
-5,56856E+15	-1,00308E+14	-2,53745E+16	-4,28862E+16	1,15618E+14	2,71229E+16	6,16235E+14	3,60607E+16	1,02929E+16	-1,41973E+14	-1,57603E+14	1,19509E+16	1979	0
2,06678E+13	-4,95167E+14	5,60369E+15	2,70288E+13	-6,87624E+14	-1,66313E+16	-2,83847E+14	5,70383E+14	2,65259E+14	2,18134E+14	1,77476E+14	9,12343E+14	103	0
1,24841E+14	8,10886E+15	-1,06494E+14	8,16317E+14	-2,31173E+16	-5,22804E+14	-2,81426E+14	1,82525E+16	-1,70198E+16	3,74603E+14	2,93738E+16	8,53156E+14	10	0
-8,86349E+15	2,52585E+14	-1,69205E+14	2,60403E+14	-7,86905E+14	-2,65088E+16	-2,18412E+14	3,442E+16	4,88777E+14	7,76578E+14	-2,34619E+14	-1,82201E+16	2218	0
2,53589E+16	2,06171E+11	2,51299E+14	4,04191E+14	-1,04992E+14	-2,21332E+16	-5,74507E+16	-1,69227E+14	-3,62014E+16	-3,02863E+16	1,42979E+12	-9,18626E+14	534	0


DIVISÃO 80-20

80%

227.845
amostras

V17	V18	V19	V20	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
-4,04233E+10	-2,17367E+10	-2,03664E+10	-3,80067E+10	-6,36033E+14	-4,46343E+13	1,90914E+14	-6,3011E+14	-2,31833E+10	-3,42319E+14	-3,19631E+14	2,96109E+14	134	0	
5,23202E+14	6,52786E+14	-4,94819E+12	-6,42768E+14	2,22626E+14	7,1556E+14	-1,4044E+16	-4,37645E+15	2,35577E+14	6,87368E+14	-4,36962E+14	-4,39763E+14	1890	0	
-5,66394E+14	-3,22615E+16	-4,69843E+14	-1,12555E+16	-2,09707E+16	-6,51683E+15	2,37007E+13	-8,27364E+14	3,08303E+14	-5,26639E+15	4,36947E+14	2,63167E+14	300	0	
-5,92347E+14	3,41083E+16	4,63545E+14	-2,01489E+14	-1,1039E+14	-2,31772E+14	2,10765E+16	2,80546E+14	2,28621E+14	-4,93849E+15	9,77409E+14	-3,68969E+13	8871	0	
-5,27594E+14	-1,38437E+14	4,24767E+14	-5,94598E+14	-1,2294E+16	-1,32005E+14	-2,63183E+14	-1,31078E+14	8,37781E+14	-2,49314E+14	3,75614E+14	-4,27693E+14	1995	0	
8,29723E+15	-1,05894E+14	-9,10833E+14	-1,59973E+14	-1,69375E+16	1,56748E+14	-1,39629E+14	4,41071E+14	-2,05752E+16	8,06209E+14	-5,91101E+14	3,97723E+13	462	0	
-3,27743E+14	1,62876E+16	3,16942E+14	-1,04322E+16	2,0602E+16	7,91915E+14	-2,8342E+13	-3,23245E+16	-1,25954E+14	4,12063E+16	2,52142E+14	1,4448E+14	1	0	
-5,71252E+14	-3,85161E+16	-1,41828E+14	-7,79231E+14	-1,4115E+16	-2,87671E+16	3,21025E+16	-9,69528E+14	-4,13205E+14	-6,22817E+14	3,73994E+14	-2,23221E+14	490	0	
-5,16841E+14	6,10526E+14	2,3693E+16	-4,61563E+14	-6,12365E+13	-2,64392E+14	9,66874E+13	5,75846E+14	3,93478E+16	8,74062E+15	-8,04975E+14	3,82997E+16	220	0	
1,61818E+16	-3,2479E+16	-3,86241E+16	-4,11328E+16	6,85528E+14	-3,7181E+14	2,76925E+16	1,07327E+14	-1,33032E+14	-3,53837E+14	-7,78793E+14	-1,62042E+14	77	0	
6,47638E+15	8,96776E+14	1,78687E+14	4,316E+14	-5,28938E+14	3,55698E+14	-4,85381E+16	5,05537E+14	3,39837E+16	6,98662E+14	5,65866E+15	1,9498E+16	2979	0	
-5,16269E+14	5,58002E+16	-1,21938E+14	-1,00149E+14	2,71944E+14	3,93594E+14	-2,56605E+14	-8,06563E+14	6,93175E+14	-7,61794E+15	1,19072E+16	-4,10578E+14	10	0	
1,68874E+16	-4,29294E+14	8,90627E+15	-6,12618E+14	-3,19873E+16	-1,3131E+16	1,23446E+14	4,48463E+16	-3,82686E+14	-5,03465E+14	-4,14104E+14	9,29971E+14	37568	0	
-4,71037E+16	-2,63519E+14	1,8803E+13	-5,81608E+14	-1,23167E+16	-3,75392E+16	-3,26952E+14	-6,52575E+14	3,05148E+16	9,79024E+14	-8,55468E+14	6,86253E+15	150	0	
-3,70071E+14	-1,99141E+16	7,79951E+15	6,92537E+14	1,63967E+14	1,24565E+16	-2,69241E+14	5,37102E+14	-2,20757E+16	-5,95548E+14	4,6071E+14	-3,35506E+14	20	1	
3,47125E+14	-9,32103E+14	-3,16671E+14	-9,13146E+14	-2,54802E+13	-5,47934E+14	1,67845E+16	-1,2493E+14	-1,26232E+16	7,60898E+15	-8,05258E+13	-2,12658E+14	2416	0	
4,5467E+13	-2,75036E+14	-1,01286E+16	4,1863E+14	-3,02238E+14	-6,07232E+15	4,16538E+14	3,63606E+14	-1,06909E+13	-1,038E+14	-5,52502E+14	-4,47605E+16	3599	0	
6,95572E+14	-2,19633E+14	1,0219E+14	-1,36299E+14	9,3296E+15	9,16355E+14	-4,42714E+14	5,02831E+15	-3,11195E+14	-1,94452E+14	-7,32163E+14	-4,48469E+14	10	0	
-1,14716E+16	-9,05938E+14	3,39605E+14	-1,00109E+16	1,92985E+16	5,12019E+15	1,41055E+14	4,24154E+16	-9,19276E+14	-3,40981E+14	-1,43849E+14	-3,70757E+14	515	0	

Obs.: este foi o DATASET utilizado para treinamento dos modelos



2º Passo Decisões

1. Meta: recall score -> classificador que melhor consegue encontrar as fraudes
2. Classe “positiva”: fraude
3. DATASET desbalanceado: criar 3 conjuntos com reamostragem sendo 1 amostra da menor classe (fraude) pra cada 5 amostras da maior classe -> aumentar o número de amostras da classe fraude
4. Máquina: i5-7400 3GHz 4 núcleos, 16GB RAM, Ubuntu 18.04.6 LTS

DECISÕES

1. 3 conjuntos (melhor amostragem) de 272.936 amostras
2. 227.447 amostras “não fraude”
3. 45.489 amostras “fraude”

```
#upsample a classe de minoria -> 1 amostra fraude para cada 5 amostras não fraude
df_1_upsampled = resample(df_1, n_samples=int(len(other_df)/5))

#concatena o dataframe upsampled
df_1_upsampled = pd.concat([df_1_upsampled, other_df])

#reseta index
df_1_upsampled = df_1_upsampled.reset_index()
```




3º Passo

Ajuste de limiar

Execução do GridSearch para:

- RandomForest (50 e 100 árvores)
- KNN (1, 3 e 5 vizinhos)
- SVM (1000 e 5000 épocas)

AJUSTES DE LIMIAR - RF

Na sequência: árvores, min_samples_split, max_depth, max_features

1. C-1: 50, 3, 15, 8
2. C-2: 50, 5, 15, 8
3. C-3: 100, 3, 15, 8

```
param_grid = {
    'min_samples_split': [3, 5], #número mínimo de amostras necessárias para dividir um nó interno
    'n_estimators': [50, 100], #número de árvores
    'max_depth': [3, 15], #altura máxima da árvore
    'max_features': [5, 8] #número de atributos a considerar quando procurar a melhor divisão
}

scorers = {
    'precision_score': make_scorer(precision_score), #habilidade de não classificar como positivo uma amostra que é negativa
    'recall_score': make_scorer(recall_score), #habilidade de encontrar todas as amostras positivas
    'accuracy_score': make_scorer(accuracy_score) # fração das previsões que o modelo acertou
}

inicioGridSearch = datetime.datetime.now();

logging.info('Rodando GridSearchCV...')
grid_search_clf = grid_search_wrapper(clf, param_grid, scorers, X_train, X_test, y_train, y_test, refit_score='recall_score')

best_n_estimators = grid_search_clf.best_params_['n_estimators']
logging.info('Melhor n_estimators: {}'.format(best_n_estimators))
```

AJUSTES DE LIMIAR - RF

CONJUNTO 1	PRED_NEG	PRED_POS
NEG	56.854	8
POS	85	11.287

CONJUNTO 2	PRED_NEG	PRED_POS
NEG	56.854	8
POS	179	11.193

CONJUNTO 3	PRED_NEG	PRED_POS
NEG	56.854	8
POS	192	11.180

Média de 50min por conjunto

AJUSTES DE LIMIAR - KNN

Número de vizinhos (1, 3 ou 5)

1. C-1: 1
2. C-2: 1
3. C-3: 1

```
param_grid = {  
    'n_neighbors': [1, 3, 5], #número de vizinhos  
}  
  
scorers = {  
    'precision_score': make_scorer(precision_score), #habilidade de não classificar como positivo uma amostra que é negativa  
    'recall_score': make_scorer(recall_score), #habilidade de encontrar todas as amostras positivas  
    'accuracy_score': make_scorer(accuracy_score) # fração das previsões que o modelo acertou  
}  
  
inicioGridSearch = datetime.datetime.now();  
  
logging.info('Rodando GridSearchCV...')  
grid_search_clf = grid_search_wrapper(clf, param_grid, scorers, X_train, X_test, y_train, y_test, refit_score='recall_score')
```

AJUSTES DE LIMIAR - KNN

CONJUNTO 1	PRED_NEG	PRED_POS
NEG	56.846	16
POS	0	11.372

CONJUNTO 2	PRED_NEG	PRED_POS
NEG	56.837	25
POS	0	11.372

CONJUNTO 3	PRED_NEG	PRED_POS
NEG	56.848	14
POS	0	11.372

Média de 3h30min por conjunto

AJUSTES DE LIMIAR - SVM

Na sequência: max_iter, cache_size

1. C-1: 5000, 200
2. C-2: 5000, 200
3. C-2: 5000, 200

```
param_grid = {  
    'max_iter': [1000, 5000], #número de iterações  
    'cache_size': [200, 500] #tamanho da cache do kernel em MB  
}  
  
scorers = {  
    'precision_score': make_scorer(precision_score), #habilidade de não classificar como positivo uma amostra que é negativa  
    'recall_score': make_scorer(recall_score), #habilidade de encontrar todas as amostras positivas  
    'accuracy_score': make_scorer(accuracy_score) # fração das previsões que o modelo acertou  
}  
  
inicioGridSearch = datetime.datetime.now();  
  
logging.info('Rodando GridSearchCV...')  
grid_search_clf = grid_search_wrapper(clf, param_grid, scorers, X_train, X_test, y_train, y_test, refit_score='recall_score')
```


AJUSTES DE LIMIAR - SVM

CONJUNTO 1	PRED_NEG	PRED_POS
NEG	56.791	71
POS	449	10.923

CONJUNTO 2	PRED_NEG	PRED_POS
NEG	56.782	80
POS	397	10.975

CONJUNTO 3	PRED_NEG	PRED_POS
NEG	56.788	74
POS	507	10.865

Média de 2h25min por conjunto

Observações até aqui

RF

CONJUNTO 1	PRED_NEG	PRED_POS
NEG	56.854	8
POS	85	11.287

- Classificou bem as fraudes
- Número considerável de amostras positivas classificadas como negativas

KNN

CONJUNTO 1	PRED_NEG	PRED_POS
NEG	56.846	16
POS	0	11.372

- Classificou MUITO bem as fraudes
- Nenhuma amostra positiva classificada como negativa
- Aumentou (nem tanto) o número de amostras negativas classificadas como positivas

SVM

DESCARTAR?		
CONJUNTO 1	PRED_NEG	PRED_POS
NEG	56.791	71
POS	449	10.923

- MUITAS amostras positivas classificadas como negativas
- Aumentou consideravelmente as neg. classificadas como pos.
- Amostras pos. classificadas como pos. caiu



4º Passo

Treinar modelos

- Execução dos algoritmos com os melhores parâmetros definidos no ajuste de limiar para 50/50 e 80/20
- Curvas ROC com KFold = 5
- Gravar modelos (pickle)
- Marcar tempo de treinamento

Separação 50/50 e 80/20

- Para cada um dos 3 conjuntos gerados no reamostramento

```
#Treine modelos usando separação do DATASET em 50/50 e 80/20;
logging.info('4. Separando os conjuntos resampled do DATASET em 50/50 e 80/20...')
X_train11, X_test11, y_train11, y_test11 = train_test_split(data_scaled1, targets1, test_size = 0.5) #50/50
X_train12, X_test12, y_train12, y_test12 = train_test_split(data_scaled2, targets2, test_size = 0.5) #50/50
X_train13, X_test13, y_train13, y_test13 = train_test_split(data_scaled3, targets3, test_size = 0.5) #50/50

X_train21, X_test21, y_train21, y_test21 = train_test_split(data_scaled1, targets1, test_size = 0.2) #80/20
X_train22, X_test22, y_train22, y_test22 = train_test_split(data_scaled2, targets2, test_size = 0.2) #80/20
X_train23, X_test23, y_train23, y_test23 = train_test_split(data_scaled3, targets3, test_size = 0.2) #80/20
```


Treinando RF com os conjuntos

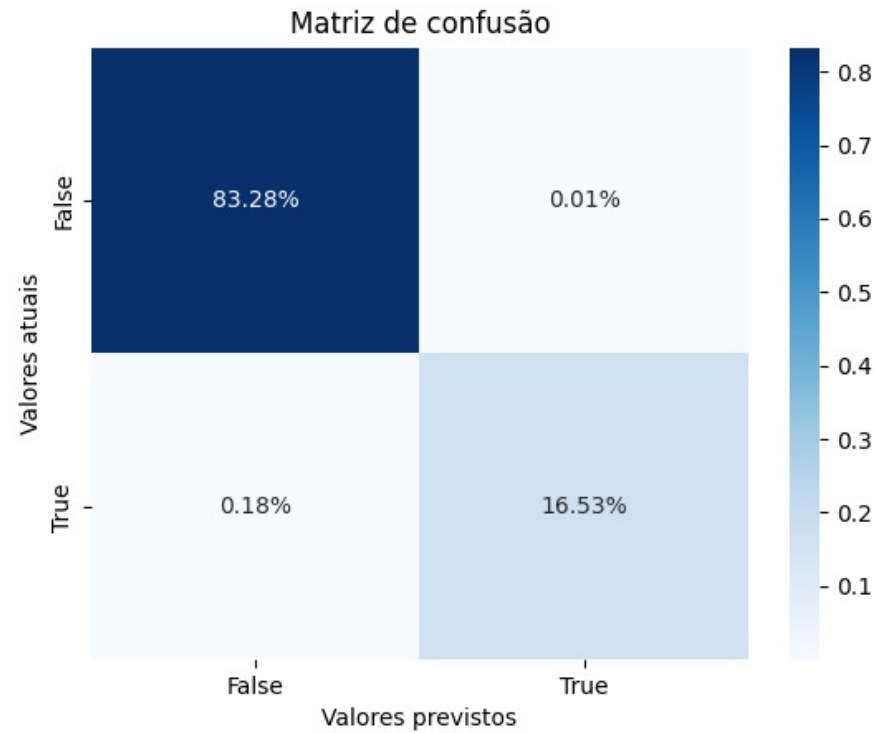
- Para cada um dos 3 conjuntos gerados no reamostramento

```
logging.info('5.1 Fazendo RandomForest com 50/50 - conjunto 1: ')\nrandomForest2(data_scaled1, targets1, X_train11, X_test11, y_train11, y_test11, '50-50-c1', params_rf_c1)\nlogging.info('5.2 Fazendo RandomForest com 50/50 - conjunto 2: ')\nrandomForest2(data_scaled2, targets2, X_train12, X_test12, y_train12, y_test12, '50-50-c2', params_rf_c2)\nlogging.info('5.3 Fazendo RandomForest com 50/50 - conjunto 3: ')\nrandomForest2(data_scaled3, targets3, X_train13, X_test13, y_train13, y_test13, '50-50-c3', params_rf_c3)\n\nlogging.info('5.4 Fazendo RandomForest com 80/20 - conjunto 1: ')\nrandomForest2(data_scaled1, targets1, X_train21, X_test21, y_train21, y_test21, '80-20-c1', params_rf_c1)\nlogging.info('5.5 Fazendo RandomForest com 80/20 - conjunto 2: ')\nrandomForest2(data_scaled2, targets2, X_train22, X_test22, y_train22, y_test22, '80-20-c2', params_rf_c2)\nlogging.info('5.6 Fazendo RandomForest com 80/20 - conjunto 3: ')\nrandomForest2(data_scaled3, targets3, X_train23, X_test23, y_train23, y_test23, '80-20-c3', params_rf_c3)
```

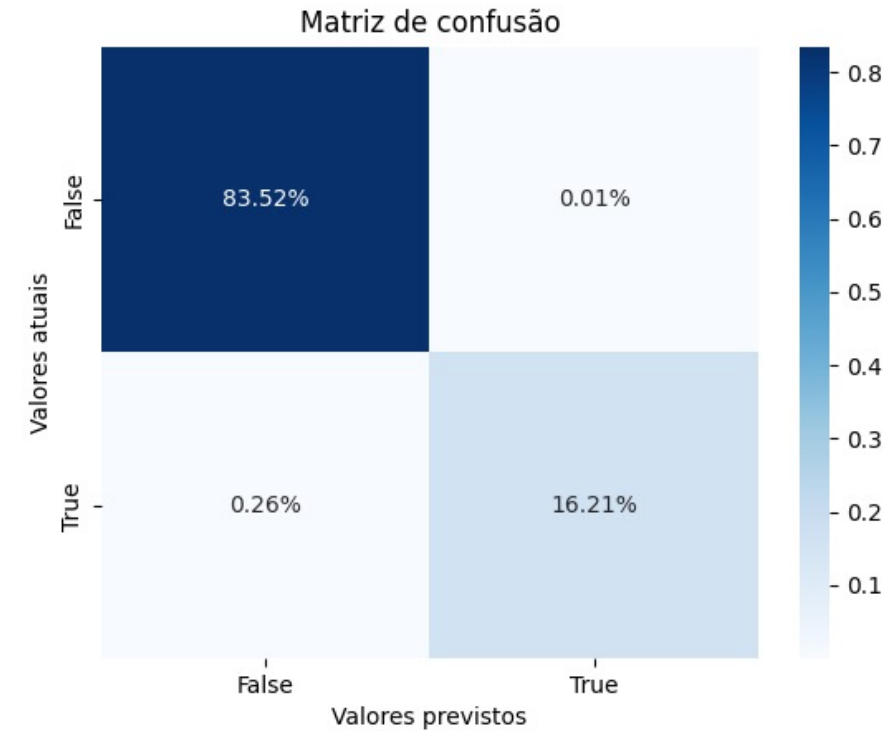
Treinando RF com o conjunto 1

50/50

80/20



- Precision: 0.999
- Accuracy: 0.998
- Recall: 0.983
- F1_Score: 0.994
- Tempo de Treinamento: 16s
- Tempo de Previsão: menos de 1s

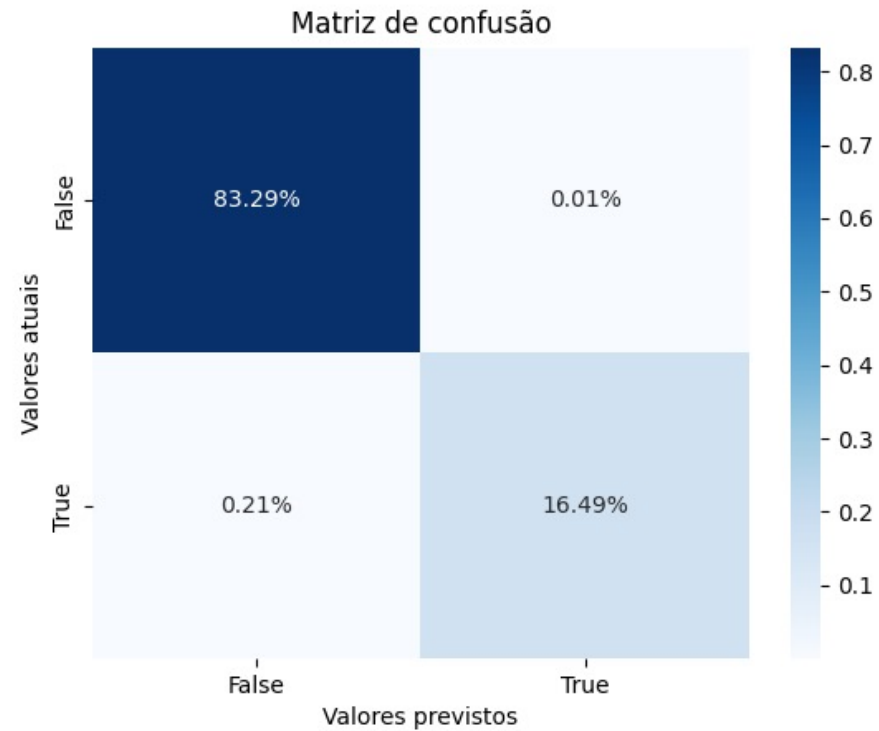


- Precision: 0.999
- Accuracy: 0.997
- Recall: 0.984
- F1_Score: 0.991
- Tempo de Treinamento: 28s
- Tempo de Previsão: menos de 1s

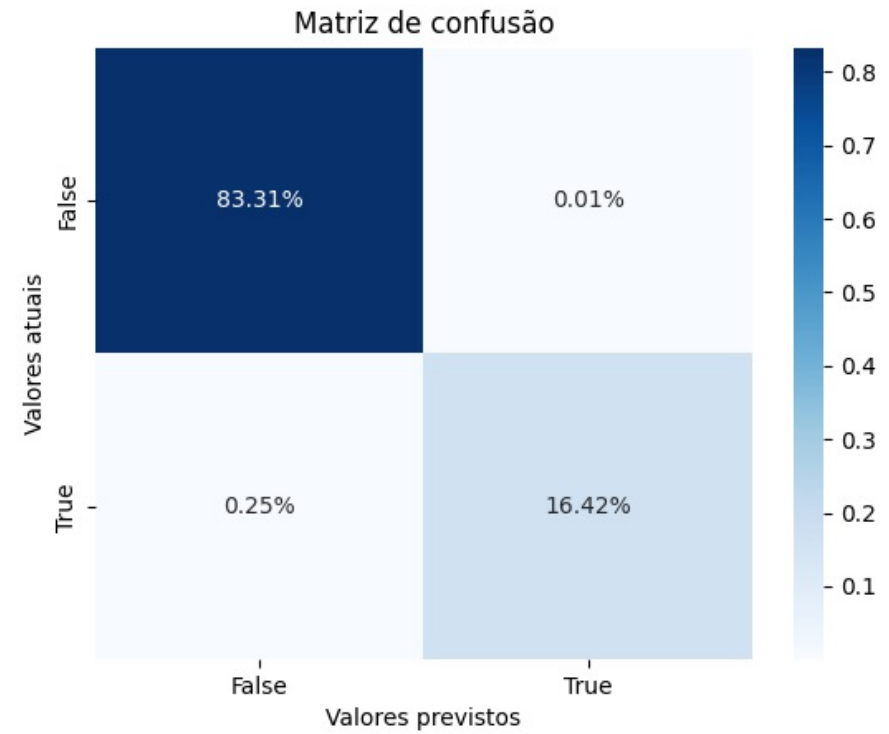
Treinando RF com o conjunto 2

50/50

80/20



- Precision: 0.999
- Accuracy: 0.997
- Recall: 0.987
- F1_Score: 0.987
- Tempo de Treinamento: 16s
- Tempo de Previsão: menos de 1s

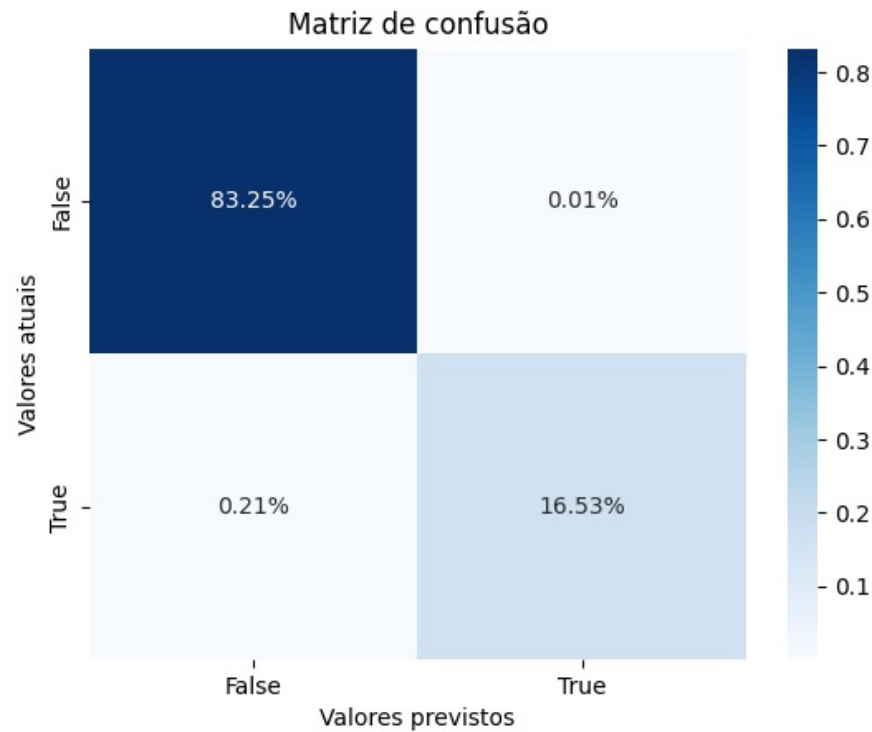


- Precision: 0.999
- Accuracy: 0.997
- Recall: 0.984
- F1_Score: 0.991
- Tempo de Treinamento: 28s
- Tempo de Previsão: menos de 1s

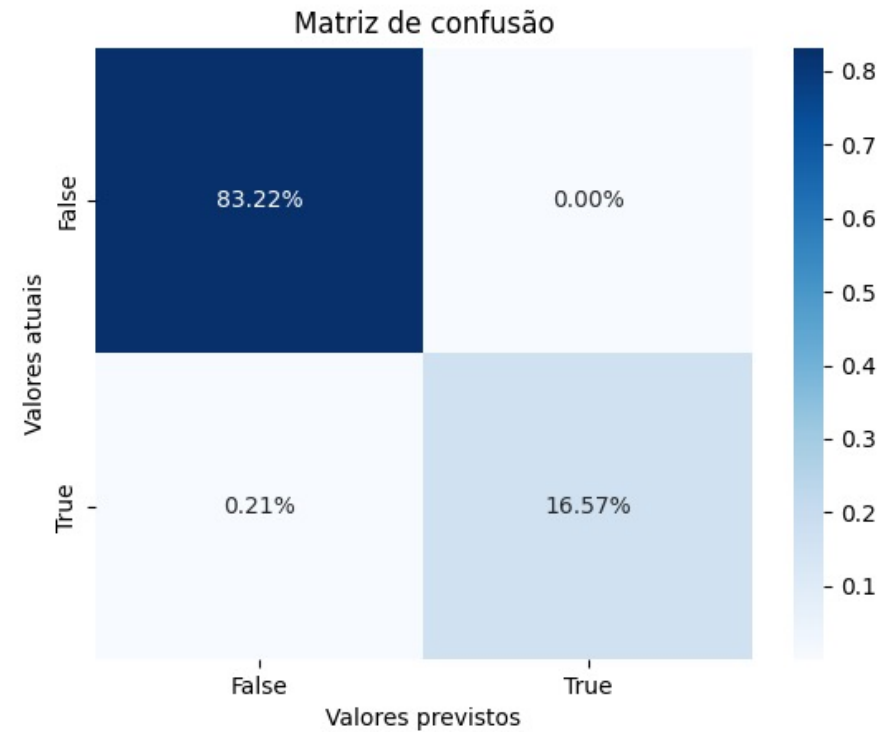
Treinando RF com o conjunto 3

50/50

80/20



- Precision: 0.999
- Accuracy: 0.997
- Recall: 0.987
- F1_Score: 0.983
- Tempo de Treinamento: 31s
- Tempo de Previsão: menos de 1s



- Precision: 0.999
- Accuracy: 0.997
- Recall: 0.987
- F1_Score: 0.993
- Tempo de Treinamento: 55s
- Tempo de Previsão: menos de 1s

Treinando KNN com os conjuntos

- Para cada um dos 3 conjuntos gerados no reamostramento

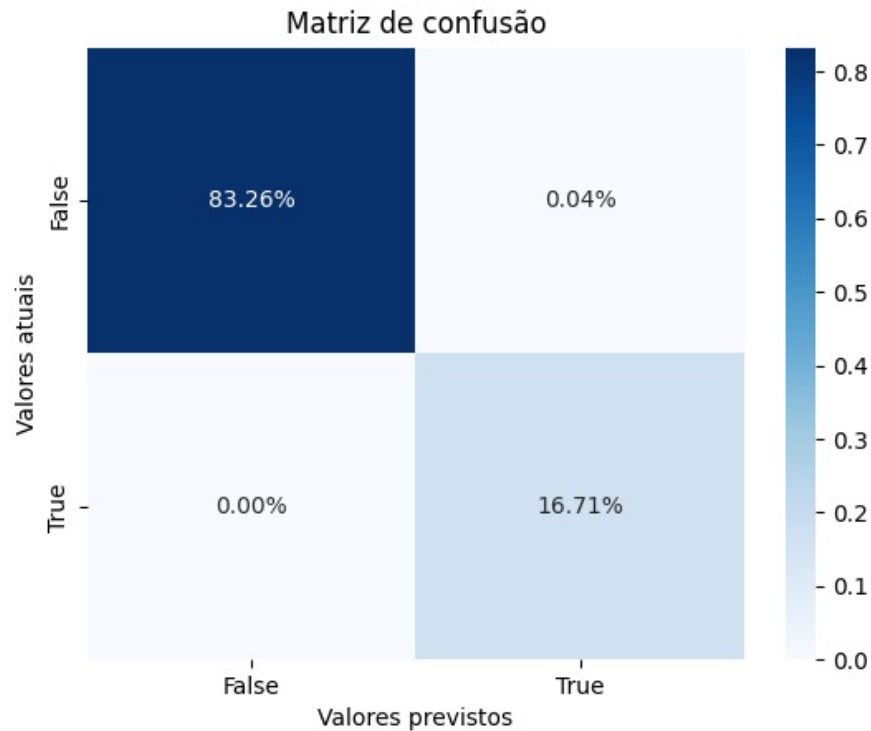
```
logging.info('6.1 Fazendo KNN com 50/50 - conjunto 1: ')
knn(data_scaled1, targets1, X_train11, X_test11, y_train11, y_test11, '50-50-c1', params_knn_c1)
logging.info('6.2 Fazendo KNN com 50/50 - conjunto 2: ')
knn(data_scaled2, targets2, X_train12, X_test12, y_train12, y_test12, '50-50-c2', params_knn_c2)
logging.info('6.3 Fazendo KNN com 50/50 - conjunto 3: ')
knn(data_scaled3, targets3, X_train13, X_test13, y_train13, y_test13, '50-50-c3', params_knn_c3)

logging.info('6.4 Fazendo KNN com 80/20 - conjunto 1: ')
knn(data_scaled1, targets1, X_train21, X_test21, y_train21, y_test21, '80-20-c1', params_knn_c1)
logging.info('6.5 Fazendo KNN com 80/20 - conjunto 2: ')
knn(data_scaled2, targets2, X_train22, X_test22, y_train22, y_test22, '80-20-c2', params_knn_c2)
logging.info('6.6 Fazendo KNN com 80/20 - conjunto 3: ')
knn(data_scaled3, targets3, X_train23, X_test23, y_train23, y_test23, '80-20-c2', params_knn_c2)
```

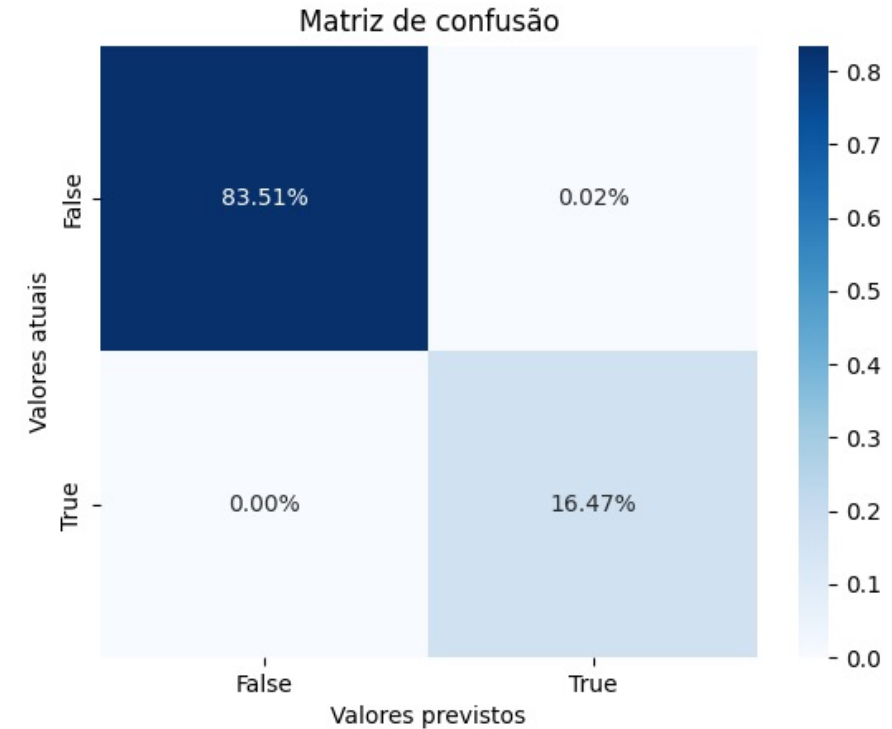
Treinando KNN com o conjunto 1

50/50

80/20



- Precision: 0.997
- Accuracy: 0.999
- Recall: 1.0
- F1_Score: 0.998
- Tempo de Treinamento: menos de 1s
- Tempo de Previsão: 2min41s

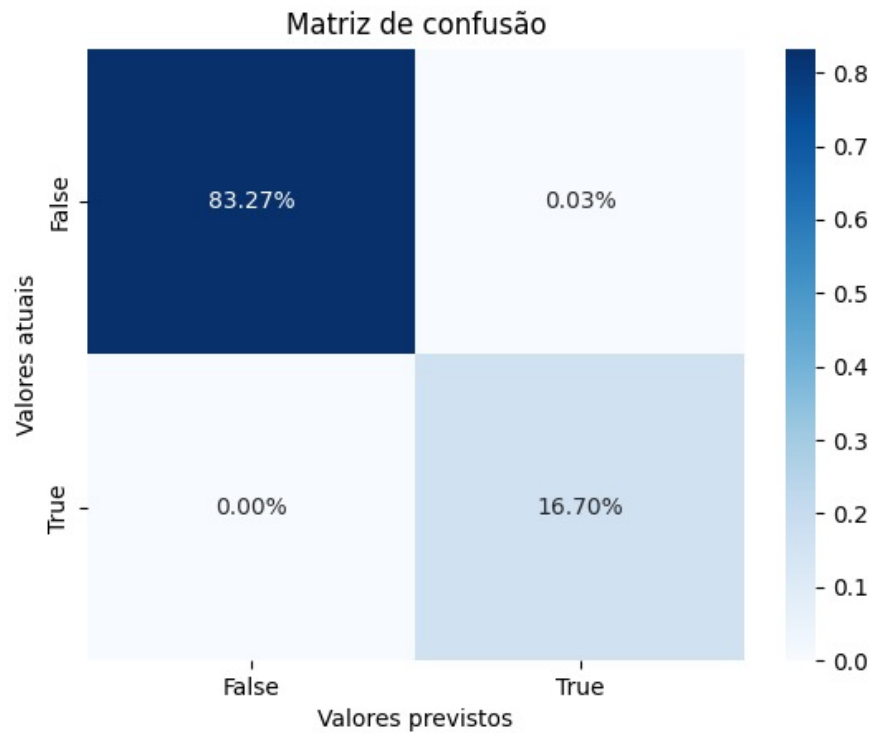


- Precision: 0.998
- Accuracy: 0.999
- Recall: 1.0
- F1_Score: 0.999
- Tempo de Treinamento: menos de 1s
- Tempo de Previsão: 01m43s

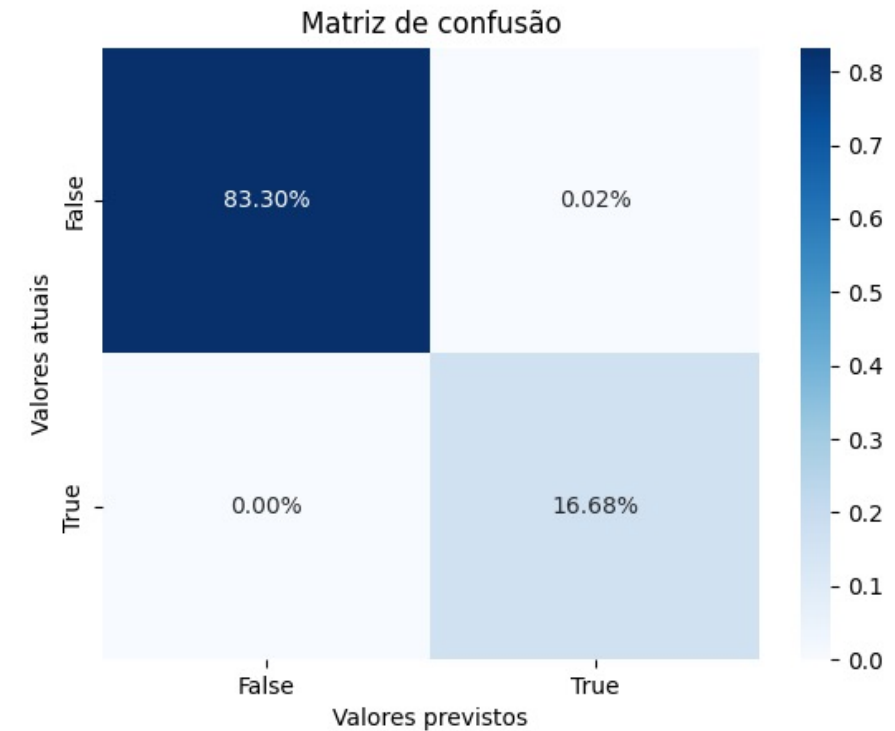
Treinando KNN com o conjunto 2

50/50

80/20



- Precision: 0.998
- Accuracy: 0.999
- Recall: 1.0
- F1_Score: 0.999
- Tempo de Treinamento: menos de 1s
- Tempo de Previsão: 2min41s

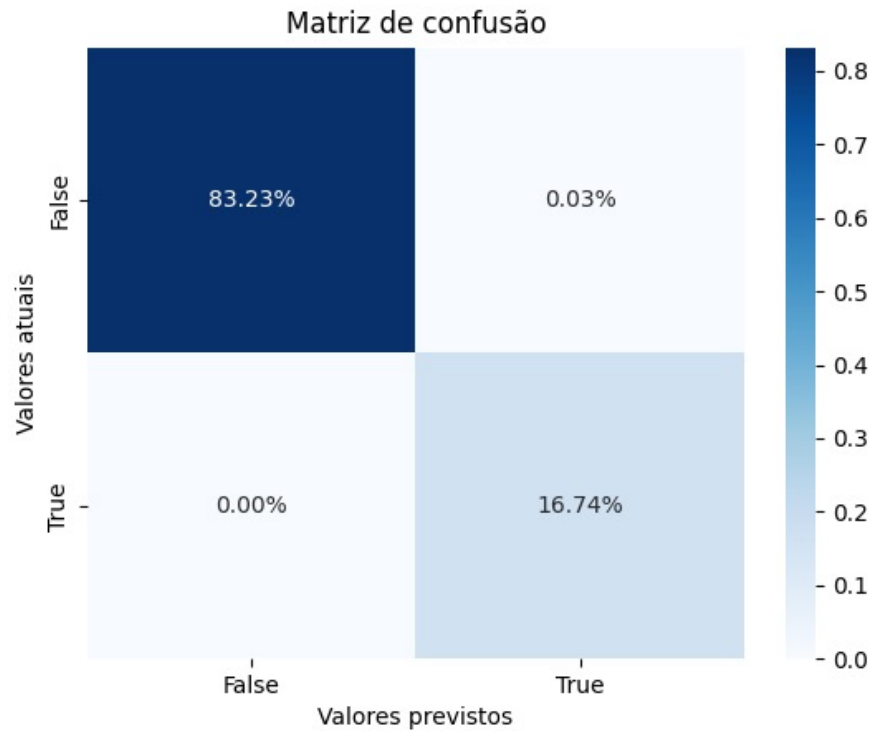


- Precision: 0.998
- Accuracy: 0.999
- Recall: 1.0
- F1_Score: 0.999
- Tempo de Treinamento: menos de 1s
- Tempo de Previsão: 01m43s

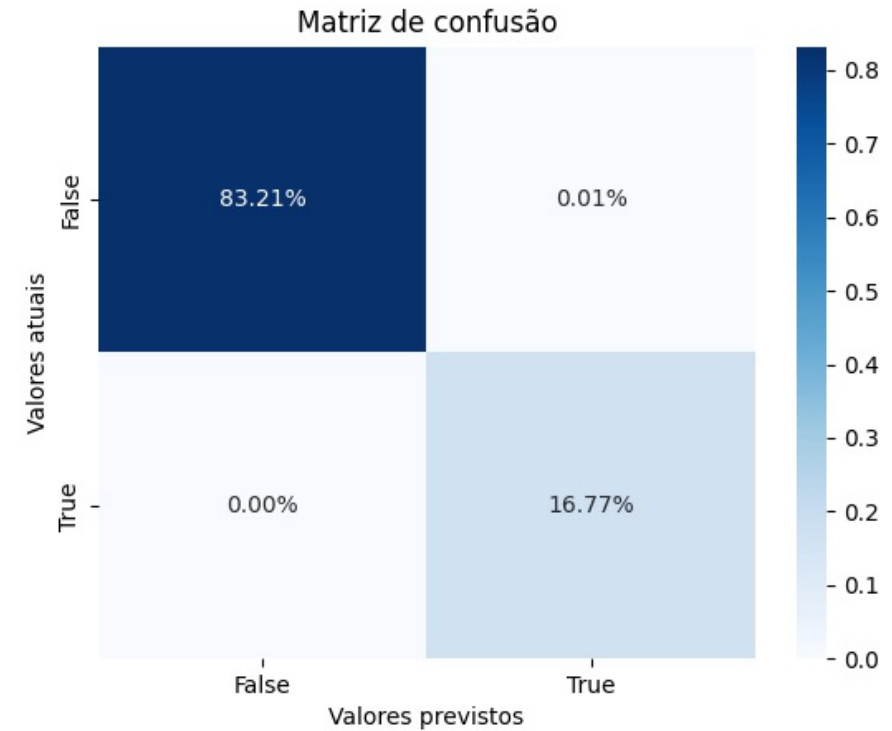
Treinando KNN com o conjunto 3

50/50

80/20



- Precision: 0.998
- Accuracy: 0.997
- Recall: 1.0
- F1_Score: 0.999
- Tempo de Treinamento: menos de 1s
- Tempo de Previsão: 2min41s



- Precision: 0.999
- Accuracy: 0.999
- Recall: 1.0
- F1_Score: 0.999
- Tempo de Treinamento: menos de 1s
- Tempo de Previsão: 01m43s

Treinando SVM com os conjuntos

- Para cada um dos 3 conjuntos gerados no reamostramento

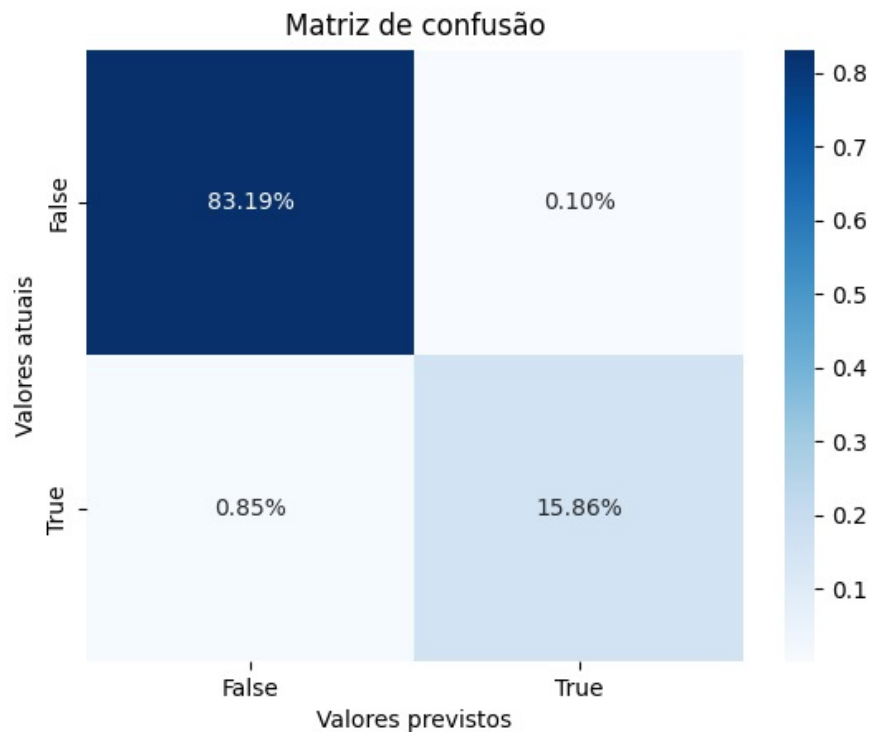
```
logging.info('8. Fazendo SVM...')
logging.info('8.1 Fazendo SVM com 50/50 - conjunto 1: ')
svm_f(data_scaled1, targets1, X_train1, X_test1, y_train1, y_test1, '50-50-c1', params_svm_c1)
logging.info('8.2 Fazendo SVM com 50/50 - conjunto 2: ')
svm_f(data_scaled2, targets2, X_train2, X_test2, y_train2, y_test2, '50-50-c2', params_svm_c2)
logging.info('8.3 Fazendo SVM com 50/50 - conjunto 3: ')
svm_f(data_scaled3, targets3, X_train3, X_test3, y_train3, y_test3, '50-50-c3', params_svm_c3)

logging.info('8.4 Fazendo SVM com 80/20 - conjunto 1: ')
svm_f(data_scaled1, targets1, X_train21, X_test21, y_train21, y_test21, '80-20-c1', params_svm_c1)
logging.info('8.4 Fazendo SVM com 80/20 - conjunto 2: ')
svm_f(data_scaled2, targets2, X_train22, X_test22, y_train22, y_test22, '80-20-c2', params_svm_c2)
logging.info('8.4 Fazendo SVM com 80/20 - conjunto 3: ')
svm_f(data_scaled3, targets3, X_train23, X_test23, y_train23, y_test23, '80-20-c3', params_svm_c3)
```

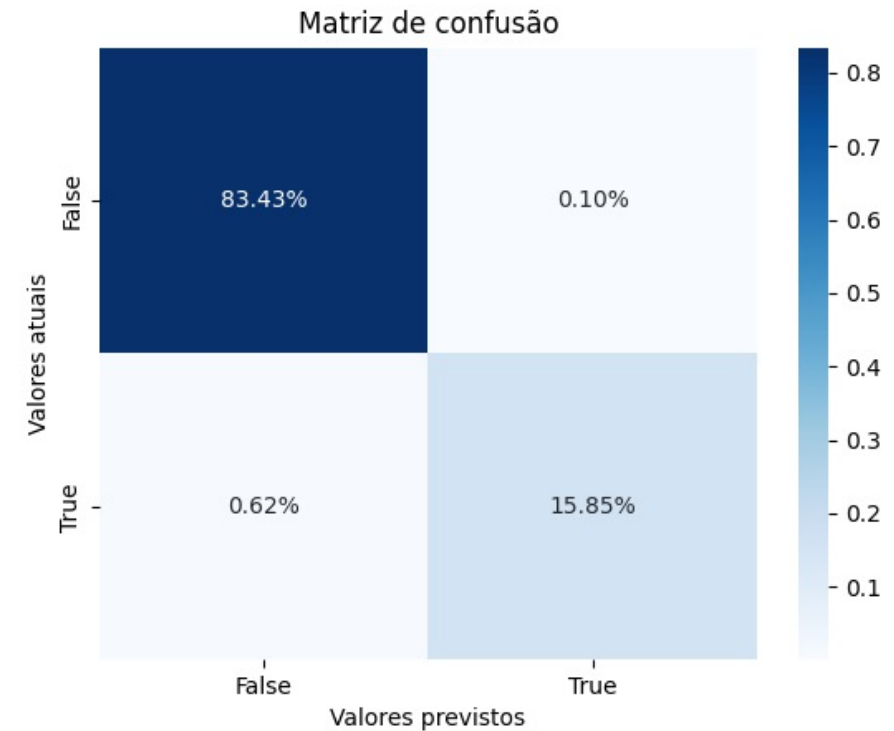
Treinando SVM com o conjunto 1

50/50

80/20



- Precision: 0.993
- Accuracy: 0.990
- Recall: 0.949
- F1_Score: 0.970
- Tempo de Treinamento: 8m11s
- Tempo de Previsão: 54s

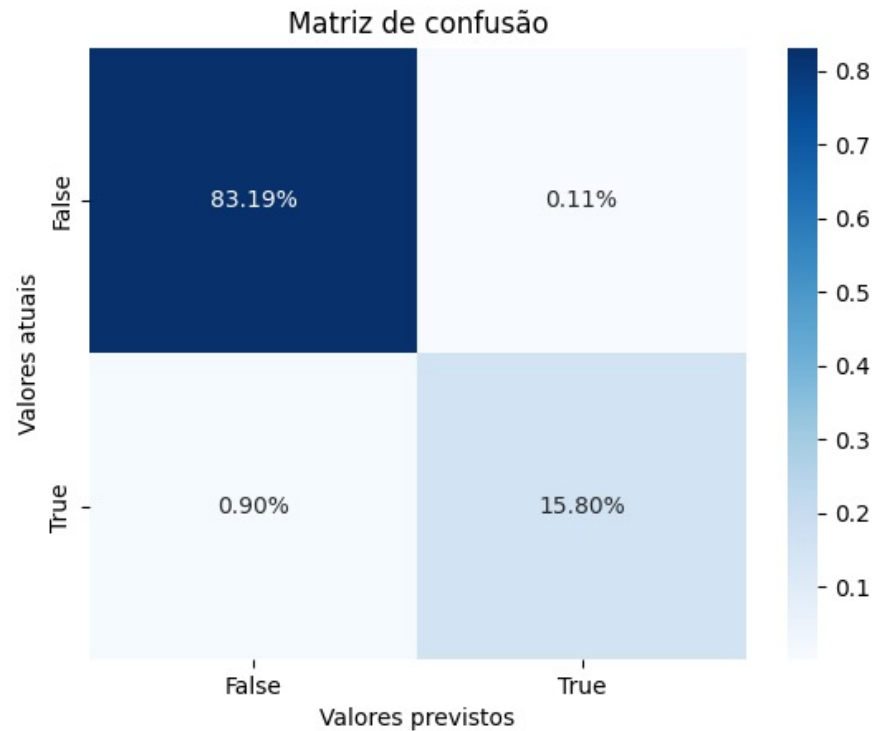


- Precision: 0.993
- Accuracy: 0.992
- Recall: 0.96
- F1_Score: 0.977
- Tempo de Treinamento: 18m11s
- Tempo de Previsão: 29s

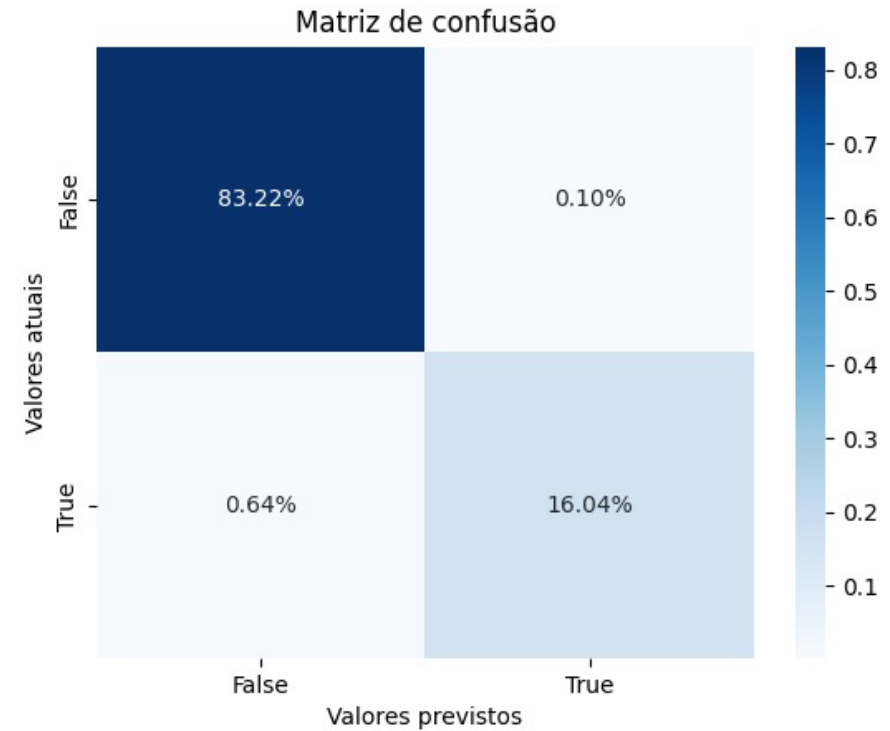
Treinando SVM com o conjunto 2

50/50

80/20



- Precision: 0.993
- Accuracy: 0.989
- Recall: 0.946
- F1_Score: 0.969
- Tempo de Treinamento: 8m10s
- Tempo de Previsão: 54s

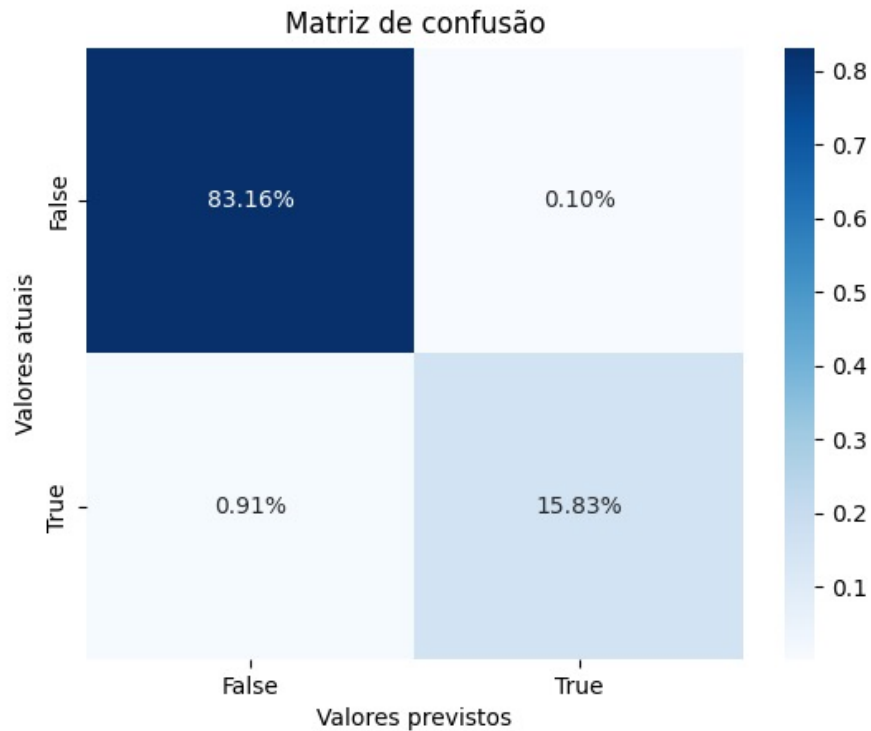


- Precision: 0.993
- Accuracy: 0.992
- Recall: 0.961
- F1_Score: 0.977
- Tempo de Treinamento: 18m05s
- Tempo de Previsão: 29s

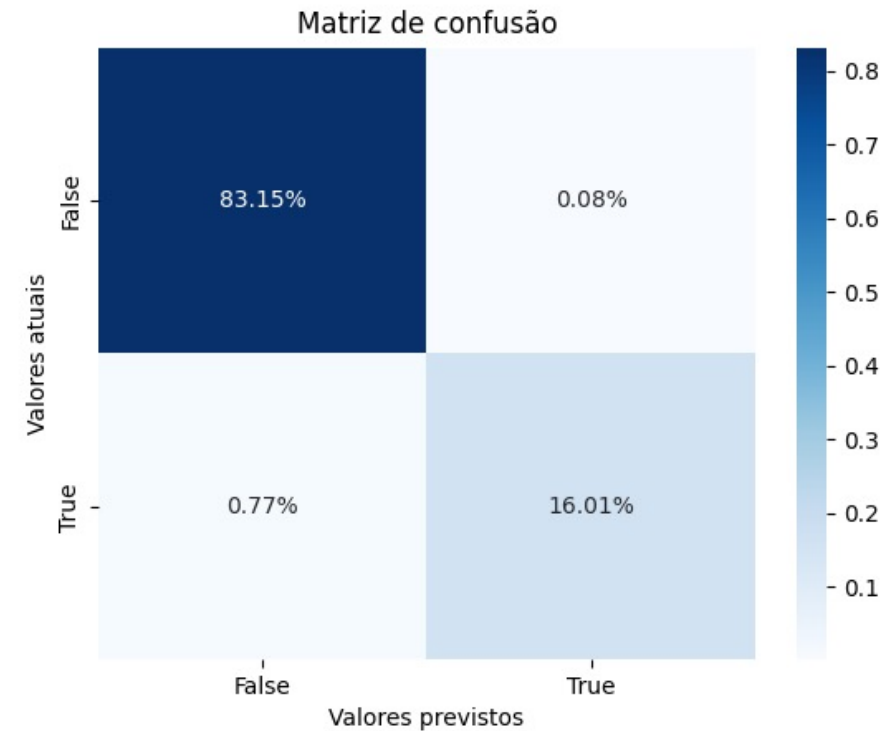
Treinando SVM com o conjunto 3

50/50

80/20



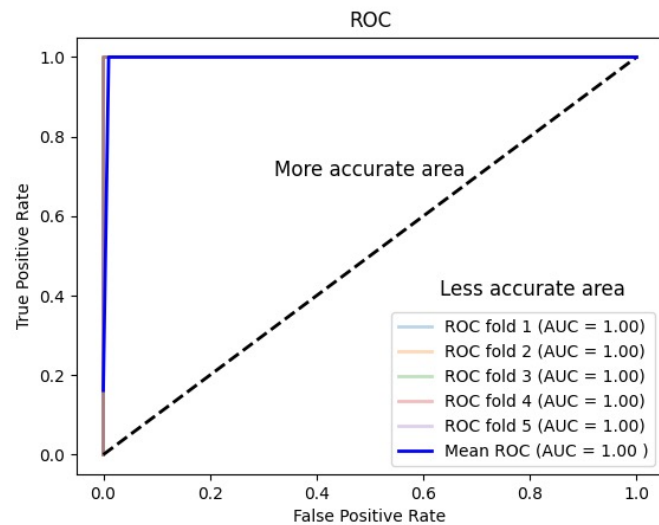
- Precision: 0.993
- Accuracy: 0.989
- Recall: 0.945
- F1_Score: 0.969
- Tempo de Treinamento: 8m09s
- Tempo de Previsão: 54s



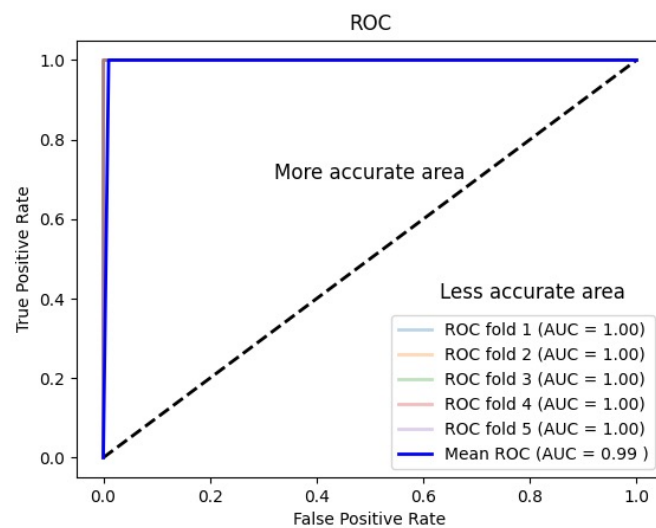
- Precision: 0.995
- Accuracy: 0.991
- Recall: 0.954
- F1_Score: 0.974
- Tempo de Treinamento: 18m02s
- Tempo de Previsão: 29s

CURVAS ROC

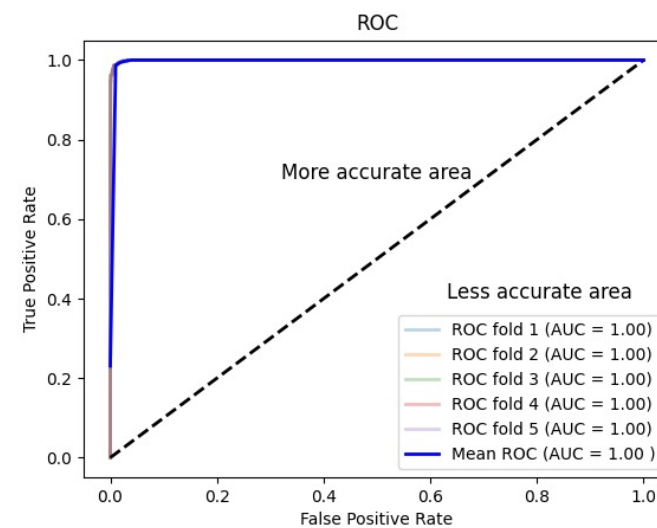
NÃO AJUDARAM MUITO...



RF



KNN



SVM

Até esse ponto, qual escolher?

- DEPENDE! O TEMPO INTERESSA “MAIS” DO QUE A CLASSIFICAÇÃO CORRETA?
- SE SIM: RF (MENOS DE 1s)
- A CLASSIFICAÇÃO CORRETA É MAIS IMPORTANTE?
- SE SIM: KNN (100% DE ACERTO PARA ESSES CONJUNTOS)
- PROVAVELMENTE, A VERIFICAÇÃO SE É FRAUDE NÃO OCORRE NO EXATO MOMENTO DA TRANSAÇÃO: PODE SER EXECUTADA EM ROTINAS PERIÓDICAS
- SE FOR ASSIM: KNN É MELHOR



Mais amostras


- Busque por amostras do mundo real para demonstração posterior: DIFÍCIL
- Solução: gerar mais amostras com resamples variados

Novos “datasets”

1. 20%: creditCard-20.csv
2. 1 fraude para cada 4 não fraude: dataset_1_4.csv (71078 fraudes e 284315 não)
3. 1 fraude para cada 3 não fraude: dataset_1_3.csv (94771 fraudes e 284315 não)
4. 1 fraude para cada 5 e random_state = 42 -> para gerador de números aleatórios: dataset_42.csv (56863 fraudes e 248315 não)
5. Dataset original: creditcard_original.csv (492 fraudes e 284.807 não)

Testes com o KNN 80/20 (conjunto 3), RF 50/50 (conjunto 3), SVM 80/20 (conjunto 2) e 50/50 para testes/treino

DATASET – 20%



RF	CONJUNTO 1	PRED_NEG	PRED_POS
	NEG	28.246	184
	POS	15	36

Recall: 0.705

KNN	CONJUNTO 2	PRED_NEG	PRED_POS
	NEG	28.388	42
	POS	11	40

Recall: 0.784

SVM	CONJUNTO 3	PRED_NEG	PRED_POS
	NEG	25.594	2.836
	POS	46	5

Recall: 0.098


DATASET - 1 PARA 3

RF	CONJUNTO 1	PRED_NEG	PRED_POS	Recall: 0.790
	NEG	141.901	26	
	POS	9.975	37.641	

KNN	CONJUNTO 2	PRED_NEG	PRED_POS	Recall: 0.940
	NEG	141.915	12	
	POS	2.836	44.780	

SVM	CONJUNTO 3	PRED_NEG	PRED_POS	Recall: 0.915
	NEG	141.856	71	
	POS	4.015	43.601	

DATASET - 1 PARA 4




RF	CONJUNTO 1	PRED_NEG	PRED_POS	Recall: 0.851
	NEG	142.000	11	
	POS	5.311	30.375	

KNN	CONJUNTO 2	PRED_NEG	PRED_POS	Recall: 0.952
	NEG	142.005	6	
	POS	1.697	33.989	

SVM	CONJUNTO 3	PRED_NEG	PRED_POS	Recall: 0.931
	NEG	141.901	110	
	POS	2.428	33.258	

DATASET - 1 PARA 5 e random_state = 42



RF	CONJUNTO 1	PRED_NEG	PRED_POS	Recall: 0.917
	NEG	141.965	10	
	POS	2.359	26.255	

KNN	CONJUNTO 2	PRED_NEG	PRED_POS	Recall: 0.961
	NEG	141.967	8	
	POS	1.094	27.520	

SVM	CONJUNTO 3	PRED_NEG	PRED_POS	Recall: 0.940
	NEG	141.825	150	
	POS	1.701	26.913	

DATASET - ORIGINAL

RF	CONJUNTO 1	PRED_NEG	PRED_POS	Recall: 0.722
	NEG	141.127	1.025	
	POS	70	182	

KNN	CONJUNTO 2	PRED_NEG	PRED_POS	Recall: 0.865
	NEG	142.010	142	
	POS	34	218	

SVM	CONJUNTO 3	PRED_NEG	PRED_POS	Recall: 0.194
	NEG	127.569	14.583	
	POS	203	49	



5º Passo

Conclusões Finais



5º Passo

Conclusões Finais

- KNN parece mesmo ser a melhor opção (com base no recall)
- SVM se comportou “bem” para os datasets reamostrados mas não no original
- RF age melhor quando temos mais amostras de fraude



OBRIGADO!