"Default of Credit Card Clients"

Gustavo Isidio dos Santos Filho

Universidade Federal de Pernambuco

Recife, 2019

Universidade Federal de Pernambuco
Centro de Informática
Graduação em Engenharia da Computação

Gustavo Isidio dos Santos Filho

"Default of Credit Card Clients"

Relatório de projeto da disciplina Tópicos Avançados em Sistemas da Informação 6 (IF1015), ministrada pelo professor Fernando Maciano de Paula Neto, para alunos do Centro de Informática da Universidade Federal de Pernambuco..

Recife, 2019

Índice

1.0 Descrição do Problema

O problema tratado aborda inadimplência com cartões de crédito. Nos EUA, por exemplo, cerca de 43% dos usuários, acabam carregando dívidas no cartão de crédito de um mês para o outro. Desses, a maior parte pertence à Geração X (1961-1981) [Fonte: Urban Institute]. Apenas 57% utilizam cartões de crédito para conveniências e ficam fora do vermelho. O impacto é tão grande que essa inadimplência alcançou $1.027 trilhões em março de 2018 [Fonte: Federal Reserve].

No Brasil, o quadro não é tão diferente. A Inadimplência das famílias paulistanas atingiu um altíssimo índice de 20,1% em março deste ano e já é a marca mais alta desde outubro do ano passado. Mais de 70% desse saldo negativo é relativo a compras no cartão de crédito.

2.0 Conteúdo da Base

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6
1	1	20000	2	2	1	24	2	2	-1	-1	...	0	0	0
2	2	120000	2	2	2	26	-1	2	0	0	...	3272	3455	3261
3	3	90000	2	2	2	34	0	0	0	0	...	14331	14948	15549
4	4	50000	2	2	1	37	0	0	0	0	...	28314	28959	29547
5	5	50000	1	2	1	57	-1	0	-1	0	...	20940	19146	19131

5 rows × 25 columns

A base escolhida, "Default of Credit Card Clients", é da UCI e tem foco nessa inadimplência. Ela contém cerca de 30k instâncias e 25 atributos. Dentre esses, alguns são categóricos, outros são numéricos e há algumas anomalias que precisam ser tratadas. Esses 25 atributos estão divididos da seguinte forma:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5
count	30000	30000	30000	30000	30000	30000	30000	30000	30000	30000	...	30000	30000
unique	30000	81	2	7	4	56	11	11	11	11	...	21548	21010
top	15146	50000	2	2	2	29	0	0	0	0	...	0	0
freq	1	3365	18112	14030	15964	1605	14737	15730	15764	16455	...	3195	3506

4 rows × 25 columns

- **ID**: Um ID referente a cada cliente que, posteriormente, será ignorado uma vez que é irrelevante para o treino.
- **LIMIT_BAL**: A quntidade máxima em dólar de crédito concedido incluindo a parcela individual e para a família.
- **SEX**: Caso o indivíduo em questão possuir sexo masculino, esse atributo será 1. Caso seja feminino, será 2.

- **EDUCATION**: Referente ao grau de escolaridade do indivíduo. Caso seja 1, trata-se de alguém com pós-graduação. Sendo 2, alguém que possui graduação. Caso tenha estudado até o colegial, irá constar 3 nesse atributo. Para qualquer outro nível de escolaridade, será 4. Quando 5 e 6, não se sabe a escolaridade do indivíduo.
- **MARRIAGE**: Referente ao status de relacionamento. Caso 1, trata-se de um indivíduo casado. Sendo 2, solteiro. Para outro tipo de relacionamento, nesse atributo constará o número 3. Caso o status do indivíduo seja desconhecido, estará 0.
- **AGE**: A idade do indivíduo em anos.
- **PAY_0**: Pagamento referente a Setembro de 2005. Caso 1, o pagamento foi feito devidamente. Caso entre 2 e 8, indica que o pagamento foi feito com atraso sendo o número referente a quantidade de meses desse atraso. Caso seja 9, significa que o pagamento foi feito com 9 ou mais meses de atraso. Para os demais atributos de PAY_2 a PAY_6, a mesma lógica de valores aplica-se.
- **PAY_2**: Pagamento referente a Agosto de 2005.
- **PAY_3**: Pagamento referente a Julho de 2005.
- **PAY_4**: Pagamento referente a Junho de 2005.
- **PAY_5**: Pagamento referente a Maio de 2005.
- **PAY_6**: Pagamento referente a Abril de 2005.
- **BILL_AMT1**: Valor da fatura em Setembro de 2005, em Dólar
- **BILL_AMT2**: Valor da fatura em Agosto de 2005, em Dólar
- **BILL_AMT3**: Valor da fatura em Julho de 2005, em Dólar
- **BILL_AMT4**: Valor da fatura em Junho de 2005, em Dólar
- **BILL_AMT5**: Valor da fatura em Maio de 2005, em Dólar
- **BILL_AMT6**: Valor da fatura em Abril de 2005, em Dólar
- **PAY_AMT1**: Quantia de pagamento anterior em Setembro de 2005, em dólar
- **PAY_AMT2**: Quantia de pagamento anterior em Agosto de 2005, em dólar
- **PAY_AMT3**: Quantia de pagamento anterior em Julho de 2005, em dólar
- **PAY_AMT4**: Quantia de pagamento anterior em Junho de 2005, em dólar
- **PAY_AMT5**: Quantia de pagamento anterior em Maio de 2005, em dólar
- **PAY_AMT6**: Quantia de pagamento anterior em Abril de 2005, em dólar
- **default.payment.next.month**: Referente a inadimplência do indivíduo no próximo mês. Com 1 representando a inadimplência e 0, a adimplência.

3.0 Pré-processamento

A priori, é notável que não há nenhum valor ausente. Estando, portanto, todos os atributos preenchidos para todos os itens.

	default.payment.next.month	PAY_6	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	...	BILL_
Total	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Percent	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

2 rows × 25 columns

No entanto, foi constatado que algumas modificações são necessárias. Alguns atributos possuem valores muito altos e, para esses, será necessário aplicar uma técnica de normalização. Outros atributos possuem números negativos e deverão ter esses valores trabalhados com uma estratégia de binarização, por exemplo. Além disso, há alguns atributos que possuem em torno de 9 categorias e vão precisar sofrer normalização.

3.1 Atributo Marriage

Como trata-se de um atributo categórico em que 0 signfica desconhecido, para os que não se sabe o status de relacionamento, serão substituídos por 3, que significa "outros". Para que tenhamos apenas valores binários, serão criadas duas colunas para cada tipo de status Status_married, Status_single, Status_others.

```
df['EDUCATION'] = df['EDUCATION'].apply(lambda x: 4 if (x==5 or x==6) else x)
```

```
#  EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others)
EDU_graduateschool = df['EDUCATION'].astype(int).apply(lambda x: 1 if x == 1 else 0)
EDU_university = df['EDUCATION'].astype(int).apply(lambda x: 1 if x == 2 else 0)
EDU_highschool = df['EDUCATION'].astype(int).apply(lambda x: 1 if x == 3 else 0)
EDU_others = df['EDUCATION'].astype(int).apply(lambda x: 1 if x == 4 else 0)
df = df.drop(['EDUCATION'], axis=1)
df['EDU_graduateschool'] = EDU_graduateschool
df['EDU_university'] = EDU_university
df['EDU_highschool'] = EDU_highschool
df['EDU_others'] = EDU_others
```

3.2 Atributo Education

Como trata-se de um atributo categórico em que 5 e 6 significam desconhecido, nos casos em que que não se sabe o grau de escolaridade, serão substituídos por 4 que significa "outros". Será feita uma normalização transformando os atributos categóricos em binários, criando 4 colunas referentes a educação.

3.3 Tratamento de valores muito grandes

	ID	LIMIT_BAL	SEX	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	...	PAY_AMT5	PAY_AMT6	default.payment.next.mo
1	1	20000	2	24	2	2	-1	-1	-2	-2	...	0	0	
2	2	120000	2	26	-1	2	0	0	0	2	...	0	2000	
3	3	90000	2	34	0	0	0	0	0	0	...	1000	5000	
4	4	50000	2	37	0	0	0	0	0	0	...	1069	1000	
5	5	50000	1	57	-1	0	-1	0	0	0	...	689	679	

5 rows × 30 columns

Para não gerar nenhuma anomalia no resultado, os atributos com valores muito grandes serão normalizados utilizando a técnica MinMaxScaler.

$$X_{new} = \frac{X_i - min(X)}{max(x) - min(X)}$$

Sofrem essa normalização os seguintes atributos: LIMIT_BAL, AGE, BILL_AMT1, BILL_AMT2', BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5 e PAY_AMT6. Abaixo, é possível perceber que os valores desses atributos ficam entre 0 e 1.

	ID	LIMIT_BAL	SEX	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	...	PAY_AMT5	PAY_AMT6	default.payment.ne
1	1	0.010101	2	0.051724	2	2	-1	-1	-2	-2	...	0.000000	0.000000	
2	2	0.111111	2	0.086207	-1	2	0	0	0	2	...	0.000000	0.003783	
3	3	0.080808	2	0.224138	0	0	0	0	0	0	...	0.002345	0.009458	
4	4	0.040404	2	0.275862	0	0	0	0	0	0	...	0.002506	0.001892	
5	5	0.040404	1	0.620690	-1	0	-1	0	0	0	...	0.001615	0.001284	

5 rows × 30 columns

3.4 Atributo SEX

Na tentativa de deixar o atributo binário, para todos que são 2, significando "feminino", será posto 0, ficando com 1 para masculino e 0 para feminino.

Antes

```
1    2
2    2
3    2
4    2
5    1
Name: SEX, dtype: object
```

Depois

```
1    0
2    0
3    0
4    0
5    1
Name: SEX, dtype: int64
```

3.5 Conteúdo da base pré-processada

Após efetuar todas as etapas de pré-processamento destacadas acima, a base passa a conter 84 atributos, sendo um deles a classe:

- **ID**: ID of each client
- **LIMIT_BAL**: Amount of given credit in NT dollars (includes individual and family/supplementary credit
- **SEX**: Gender (1 = male, 0 = female)
- **EDU_graduateschool**: EDUCATION is graduate school (1 = yes, 0 = no)
- **EDU_university**: EDUCATION is university (1 = yes, 0 = no)

- **EDU_highschool**: EDUCATION is high school (1 = yes, 0 = no)
- **EDU_others**: EDUCATION is others (1 = yes, 0 = no)
- **AGE**: Age in years

Repayment status in September, 2005

- **PAY_0_duly**: Pay duly
- **PAY_0_D1**: Payment delay for one month (1 = yes, 0 = no)
- **PAY_0_D2**: Payment delay for two months (1 = yes, 0 = no)
- **PAY_0_D3**: Payment delay for three months (1 = yes, 0 = no)
- **PAY_0_D4**: Payment delay for four months (1 = yes, 0 = no)
- **PAY_0_D5**: Payment delay for five months (1 = yes, 0 = no)
- **PAY_0_D6**: Payment delay for six months (1 = yes, 0 = no)
- **PAY_0_D7**: Payment delay for seven months (1 = yes, 0 = no)
- **PAY_0_D8**: Payment delay for eight months (1 = yes, 0 = no)
- **PAY_0_D9**: Payment delay for nine months and above) (1 = yes, 0 = no)

Repayment status in August, 2005

- **PAY_2_duly**: Pay duly (1 = yes, 0 = no)
- **PAY_2_D1**: Payment delay for one month (1 = yes, 0 = no)
- **PAY_2_D2**: Payment delay for two months (1 = yes, 0 = no)
- **PAY_2_D3**: Payment delay for three months (1 = yes, 0 = no)
- **PAY_2_D4**: Payment delay for four months (1 = yes, 0 = no)
- **PAY_2_D5**: Payment delay for five months (1 = yes, 0 = no)
- **PAY_2_D6**: Payment delay for six months (1 = yes, 0 = no)
- **PAY_2_D7**: Payment delay for seven months (1 = yes, 0 = no)
- **PAY_2_D8**: Payment delay for eight months (1 = yes, 0 = no)
- **PAY_2_D9**: Payment delay for nine months and above) (1 = yes, 0 = no)

Repayment status in July, 2005

- **PAY_3_duly**: Pay duly (1 = yes, 0 = no)
- **PAY_3_D1**: Payment delay for one month (1 = yes, 0 = no)
- **PAY_3_D2**: Payment delay for two months (1 = yes, 0 = no)
- **PAY_3_D3**: Payment delay for three months (1 = yes, 0 = no)
- **PAY_3_D4**: Payment delay for four months (1 = yes, 0 = no)
- **PAY_3_D5**: Payment delay for five months (1 = yes, 0 = no)
- **PAY_3_D6**: Payment delay for six months (1 = yes, 0 = no)
- **PAY_3_D7**: Payment delay for seven months (1 = yes, 0 = no)
- **PAY_3_D8**: Payment delay for eight months (1 = yes, 0 = no)
- **PAY_3_D9**: Payment delay for nine months and above) (1 = yes, 0 = no)

Repayment status in June, 2005

- **PAY_4_duly**: Pay duly (1 = yes, 0 = no)
- **PAY_4_D1**: Payment delay for one month (1 = yes, 0 = no)
- **PAY_4_D2**: Payment delay for two months (1 = yes, 0 = no)
- **PAY_4_D3**: Payment delay for three months (1 = yes, 0 = no)
- **PAY_4_D4**: Payment delay for four months (1 = yes, 0 = no)
- **PAY_4_D5**: Payment delay for five months (1 = yes, 0 = no)
- **PAY_4_D6**: Payment delay for six months (1 = yes, 0 = no)
- **PAY_4_D7**: Payment delay for seven months (1 = yes, 0 = no)
- **PAY_4_D8**: Payment delay for eight months (1 = yes, 0 = no)
- **PAY_4_D9**: Payment delay for nine months and above) (1 = yes, 0 = no)

Repayment status in May, 2005

- **PAY_5_duly**: Pay duly (1 = yes, 0 = no)
- **PAY_5_D1**: Payment delay for one month (1 = yes, 0 = no)
- **PAY_5_D2**: Payment delay for two months (1 = yes, 0 = no)
- **PAY_5_D3**: Payment delay for three months (1 = yes, 0 = no)
- **PAY_5_D4**: Payment delay for four months (1 = yes, 0 = no)
- **PAY_5_D5**: Payment delay for five months (1 = yes, 0 = no)
- **PAY_5_D6**: Payment delay for six months (1 = yes, 0 = no)
- **PAY_5_D7**: Payment delay for seven months (1 = yes, 0 = no)
- **PAY_5_D8**: Payment delay for eight months (1 = yes, 0 = no)
- **PAY_5_D9**: Payment delay for nine months and above) (1 = yes, 0 = no)

Repayment status in April, 2005

- **PAY_6_duly**: Pay duly (1 = yes, 0 = no)
- **PAY_6_D1**: Payment delay for one month (1 = yes, 0 = no)
- **PAY_6_D2**: Payment delay for two months (1 = yes, 0 = no)
- **PAY_6_D3**: Payment delay for three months (1 = yes, 0 = no)
- **PAY_6_D4**: Payment delay for four months (1 = yes, 0 = no)
- **PAY_6_D5**: Payment delay for five months (1 = yes, 0 = no)
- **PAY_6_D6**: Payment delay for six months (1 = yes, 0 = no)
- **PAY_6_D7**: Payment delay for seven months (1 = yes, 0 = no)
- **PAY_6_D8**: Payment delay for eight months (1 = yes, 0 = no)
- **PAY_6_D9**: Payment delay for nine months and above) (1 = yes, 0 = no)
- **BILL_AMT1**: Amount of bill statement in September, 2005 (Scaled between -1 and 1)
- **BILL_AMT2**: Amount of bill statement in August, 2005 (Scaled between -1 and 1)
- **BILL_AMT3**: Amount of bill statement in July, 2005 (Scaled between -1 and 1)
- **BILL_AMT4**: Amount of bill statement in June, 2005 (Scaled between -1 and 1)
- **BILL_AMT5**: Amount of bill statement in May, 2005 (Scaled between -1 and 1)

- **BILL_AMT6**: Amount of bill statement in April, 2005 (Scaled between -1 and 1)
- **PAY_AMT1**: Amount of previous payment in September, 2005 (Scaled between -1 and 1)
- **PAY_AMT2**: Amount of previous payment in August, 2005 (Scaled between -1 and 1)
- **PAY_AMT3**: Amount of previous payment in July, 2005 (Scaled between -1 and 1)
- **PAY_AMT4**: Amount of previous payment in June, 2005 (Scaled between -1 and 1)
- **PAY_AMT5**: Amount of previous payment in May, 2005 (Scaled between -1 and 1)
- **PAY_AMT6**: Amount of previous payment in April, 2005 (Scaled between -1 and 1)

Class

- **Class**: Default payment next month (1 = yes, 0 = no)

4.0 Aplicando técnicas de IA

Foram escolhidas 4 técnicas. A primeira, kNN, separa 30% da base para testes e efetua o treinamento com o restante. Ao final, 21000 foram treinados e 9000 testes foram feitos. Desses, 7293 (81.03%) tiveram resultado correto e 1707 falharam (18.97%).

A segunda técnica, utiliza o Keras para criar uma rede neural de duas camadas com k-fold. As duas primeiras, possuem 46 neurônios cada e a função de ativação utilizada foi a relu. A terceira, de saída, possui apenas 1 neurônios e sua função de ativação é sigmoid. Foi feito um k-fold utilizando 7 folds.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 46)                3818
_________________________________________________________________
dense_5 (Dense)              (None, 46)                2162
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 47
=================================================================
Total params: 6,027
Trainable params: 6,027
Non-trainable params: 0
_________________________________________________________________
```

```
Fold  0
Loss: 0.435255450897555 | Accuracy: 0.8161455904052349

Fold  1
Loss: 0.4333191811621217 | Accuracy: 0.8217452168186524

Fold  2
Loss: 0.4464238942698569 | Accuracy: 0.8180121326079393

Fold  3
Loss: 0.4433815672861545 | Accuracy: 0.8175454970781234

Fold  4
Loss: 0.4374575694059024 | Accuracy: 0.822678487961725

Fold  5
Loss: 0.4351353461364047 | Accuracy: 0.8210035007364413

Fold  6
Loss: 0.4405726691805774 | Accuracy: 0.8175029170833085
```

A terceira técnica escolhida foi K-means. Nela, 30% da base foi teste e o treinamento foi efetuado com o restante. Ao final, 21000 foram treinados e 9000 testes foram feitos. Desses, 7014 (77.93%) tiveram resultado correto e 1986 (22.07%) falharam.

A quarta técnica escolhida foi a Random Forest utilizando k-fold com 4 folds. Os resultados de cada fold podem ser vistos abaixo:

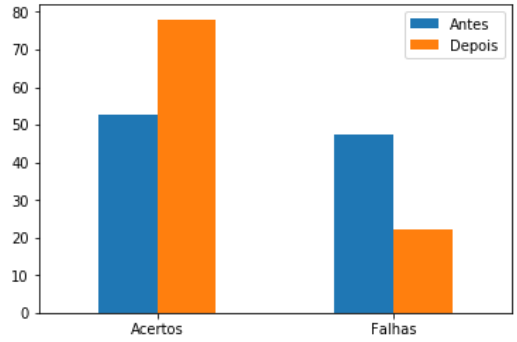
```
Fold  0
Resultado: 18.39 falharam e 81.61 acertaram

Fold  1
Resultado: 18.35 falharam e 81.65 acertaram

Fold  2
Resultado: 18.81 falharam e 81.19 acertaram

Fold  3
Resultado: 18.25 falharam e 81.75 acertaram
```

5.0 Comparando resultados

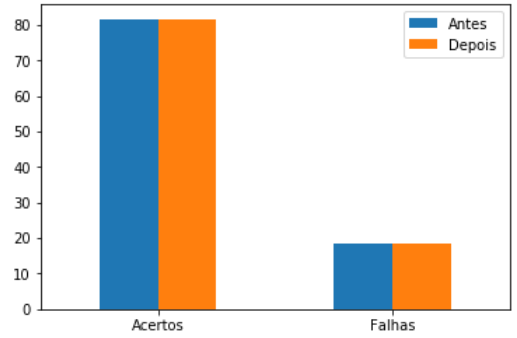
5.1 Impacto do pré-processamento

Durante o experimento, após aplicar as técnicas de IA antes de depois do pré-processamento, ficou claro o impacto causado por ele para o caso de algumas técnicas e sua irrelevância para o caso de outras. No caso do k-means, por exemplo, o impacto foi enorme, representando uma diferença de aproximadamente 20% no número de acertos. Já com a kNN, o impacto representou menos de 10% e a Random Forest não sofreu com a ausência do pré-processamento.

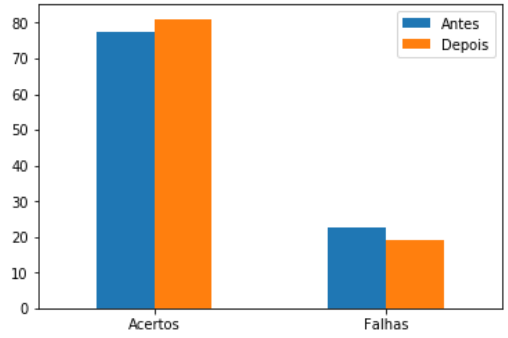
K-Means



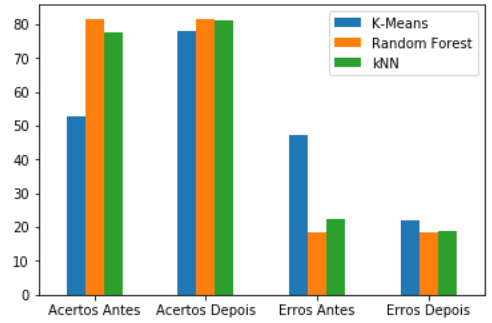
Random Forest



kNN

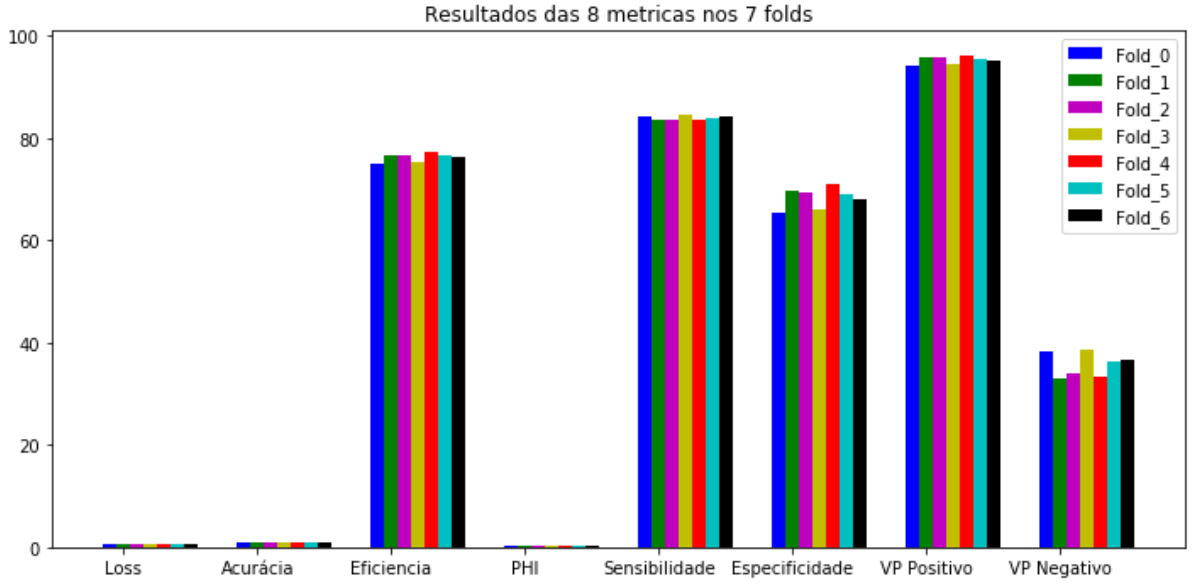


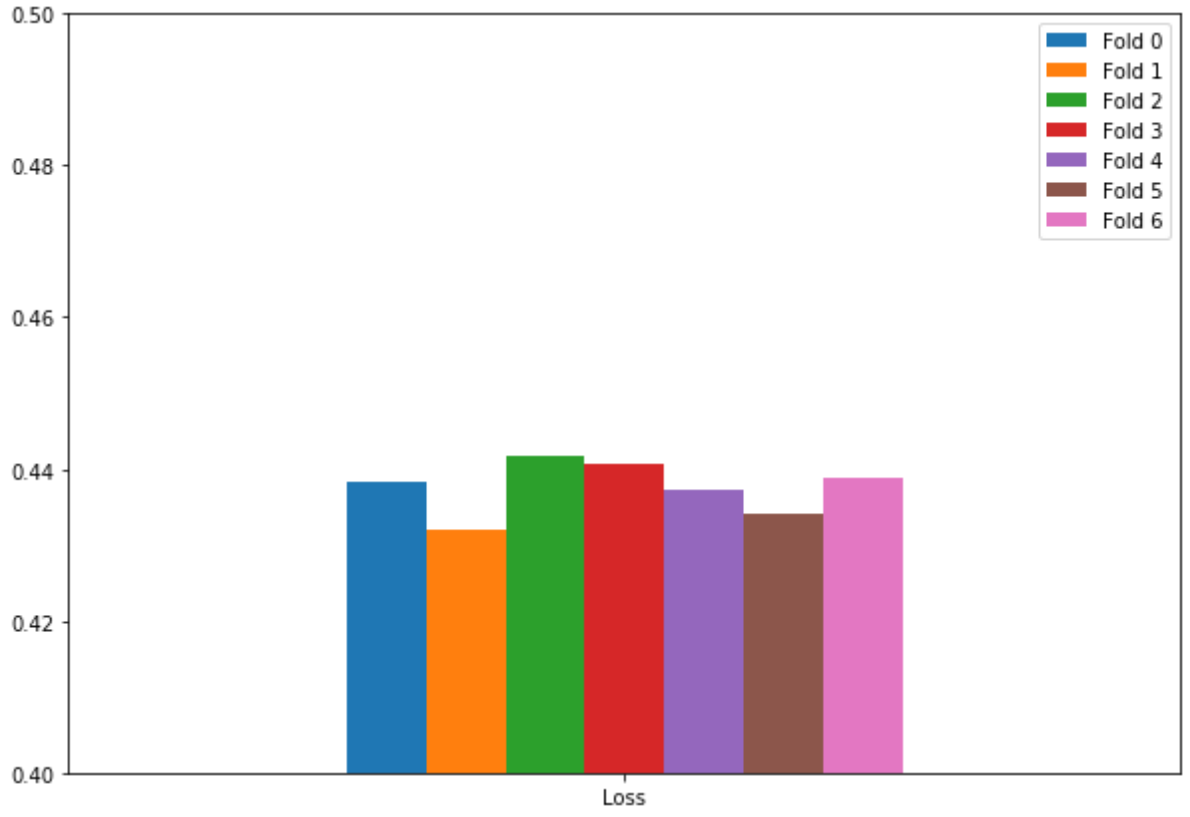
Geral

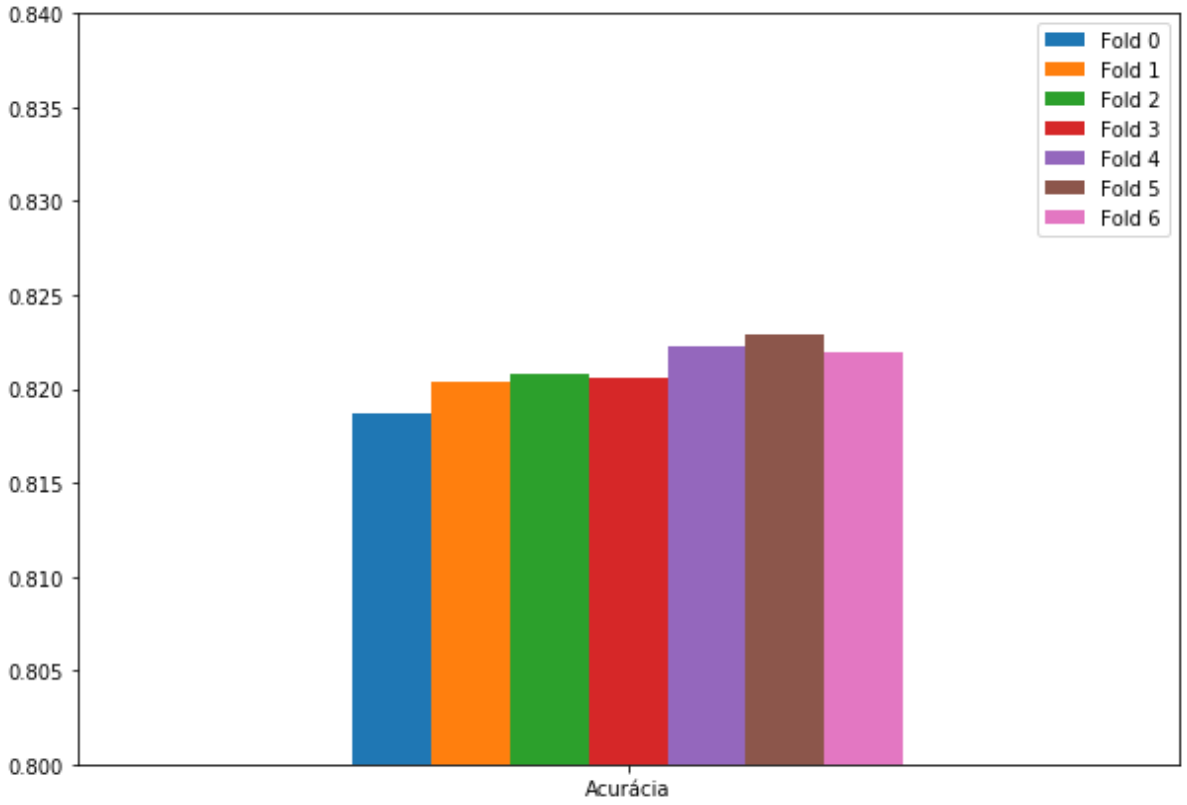


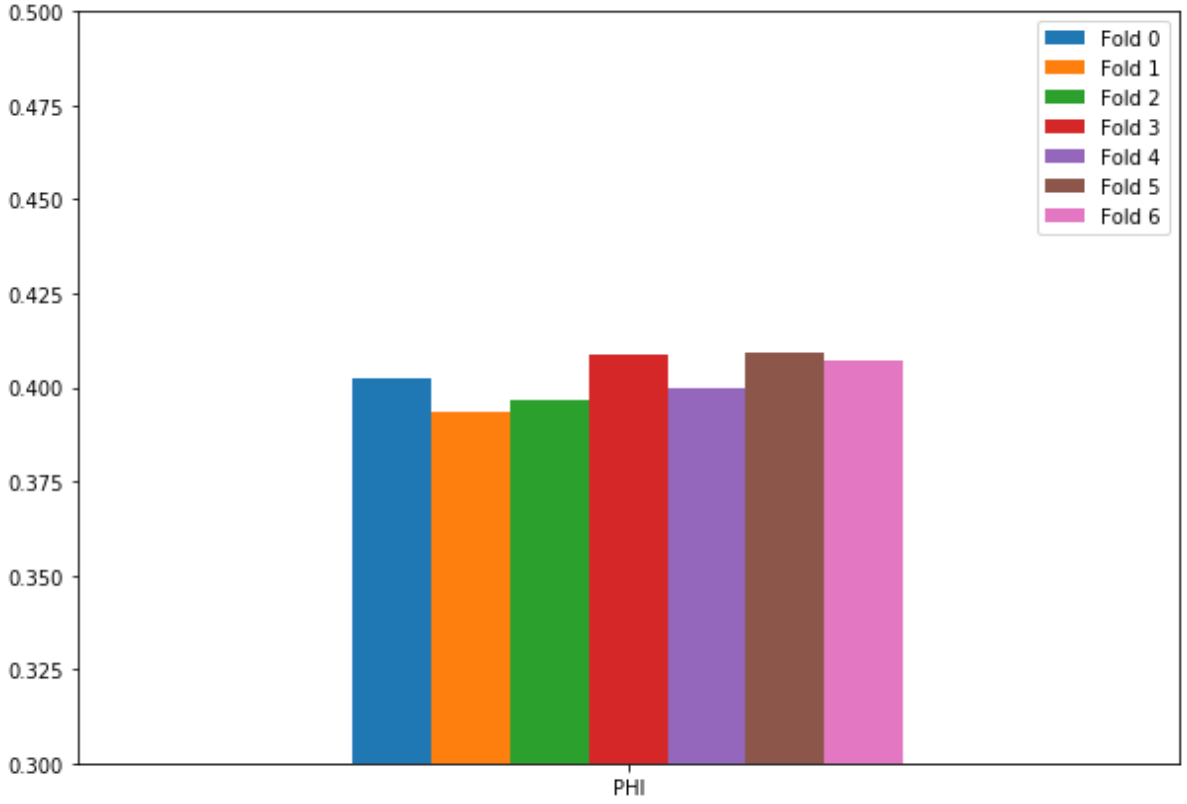
5.2 Rede Neural - Keras

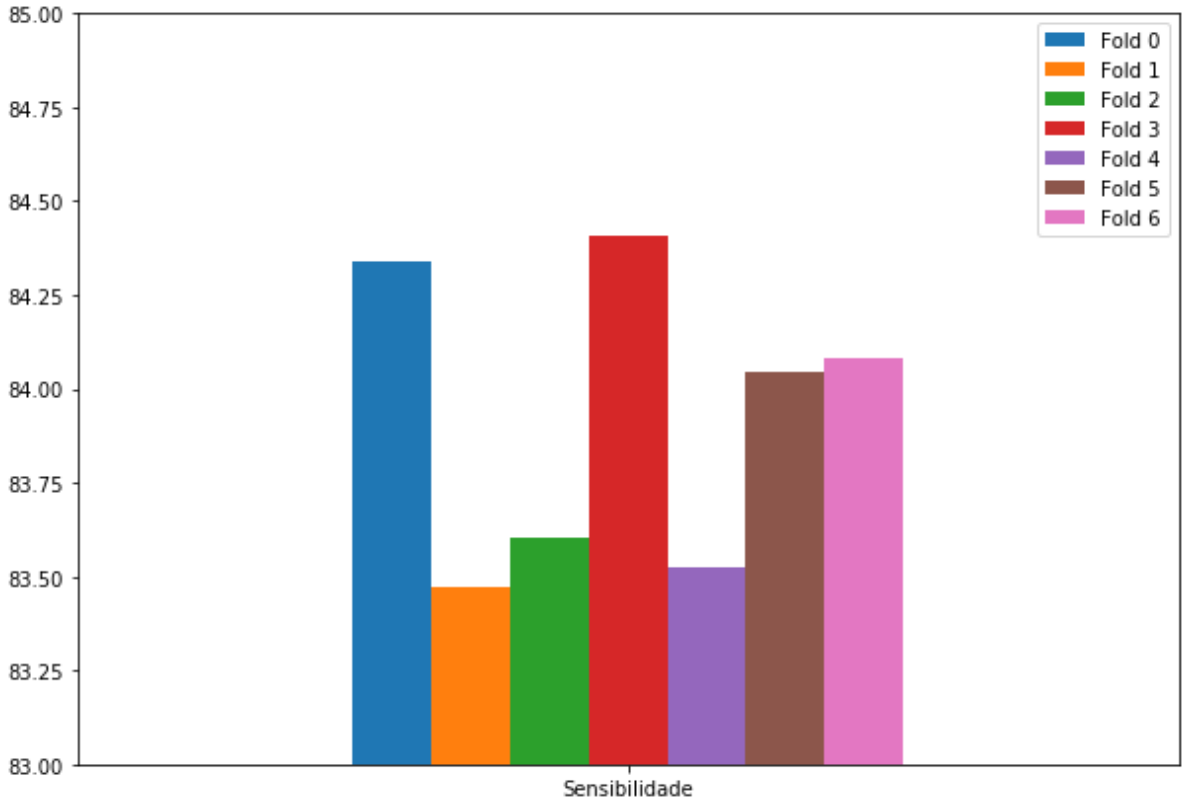
Abaixo é possível ver o quanto a sensibilidade permanece estável em cada um dos folds enquanto que a especificidade varia um pouco.

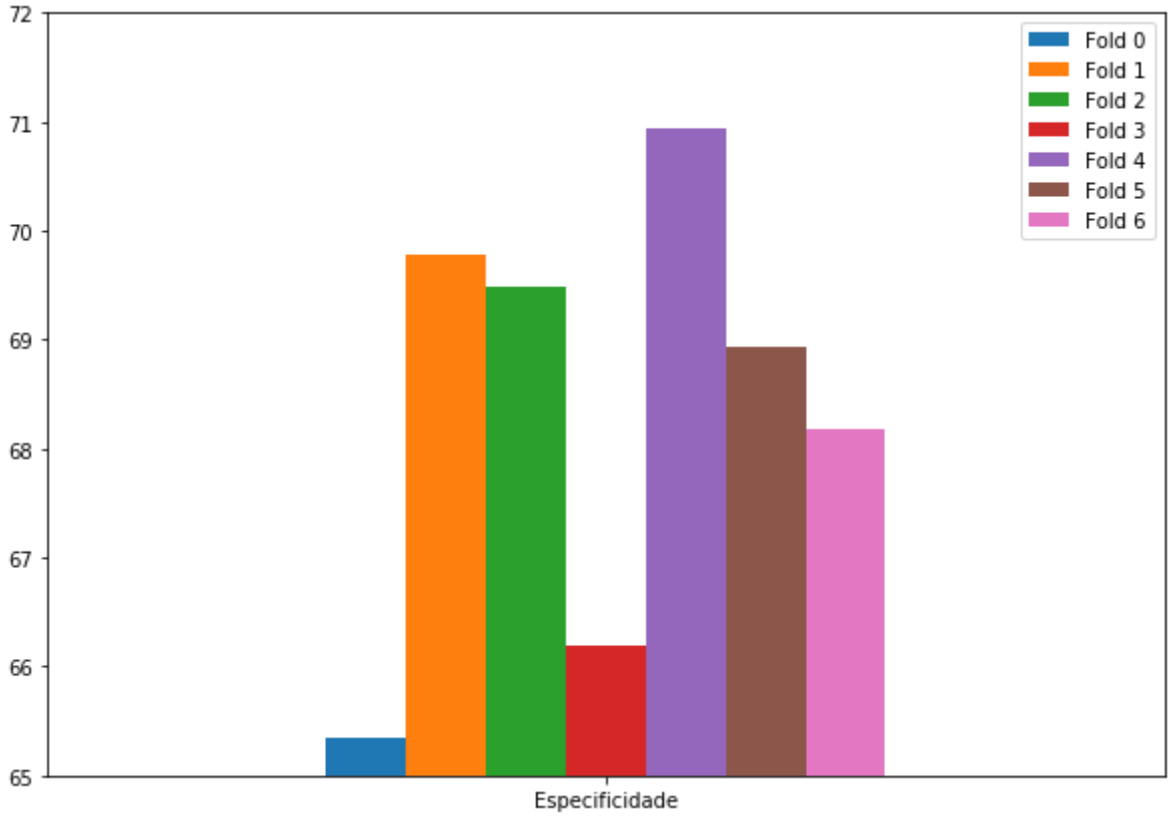

Resultados das 8 metricas nos 7 folds
100
80
60
40
20
0
Fold_0
Fold_1
Fold_2
Fold_3
Fold_4
Fold_5
Fold_6
Loss
Acurácia
Eficiencia
PHI
Sensibilidade
Especificidade
VP Positivo
VP Negativo

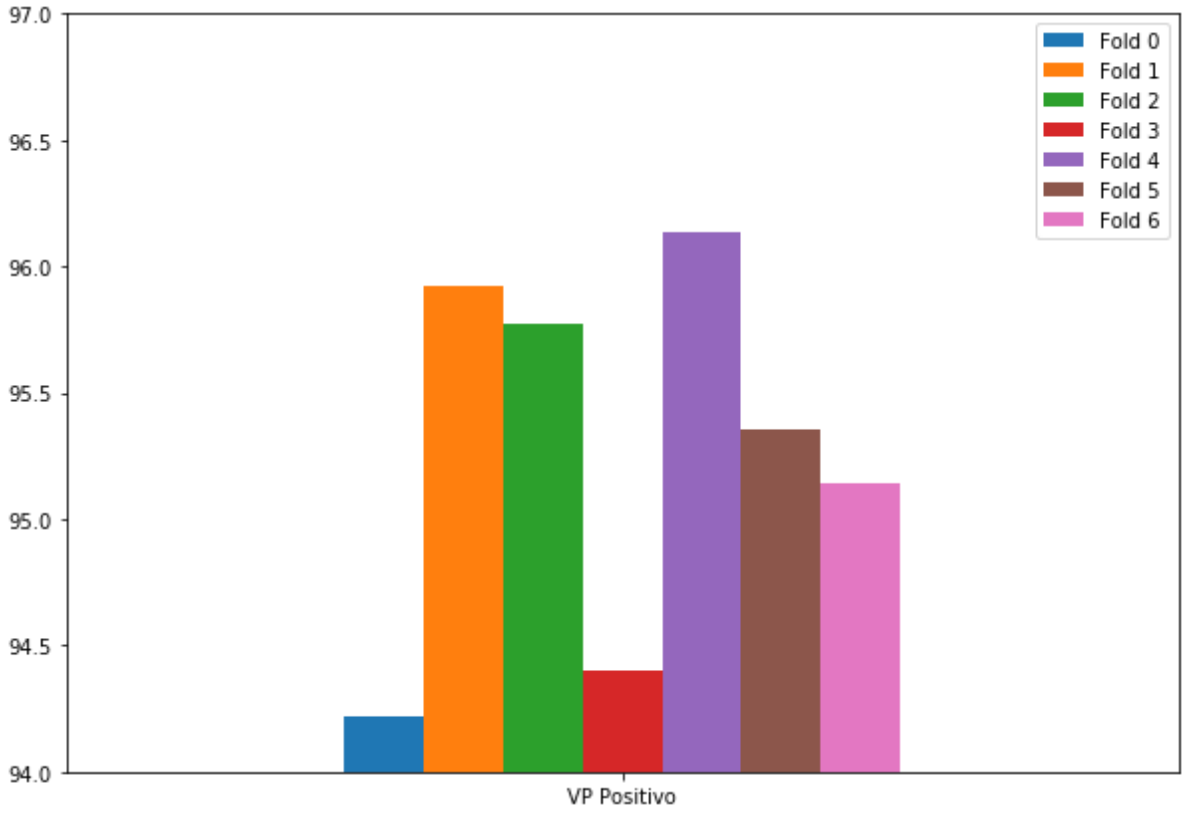

0.50
0.48
0.46
0.44
0.42
0.40
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Loss

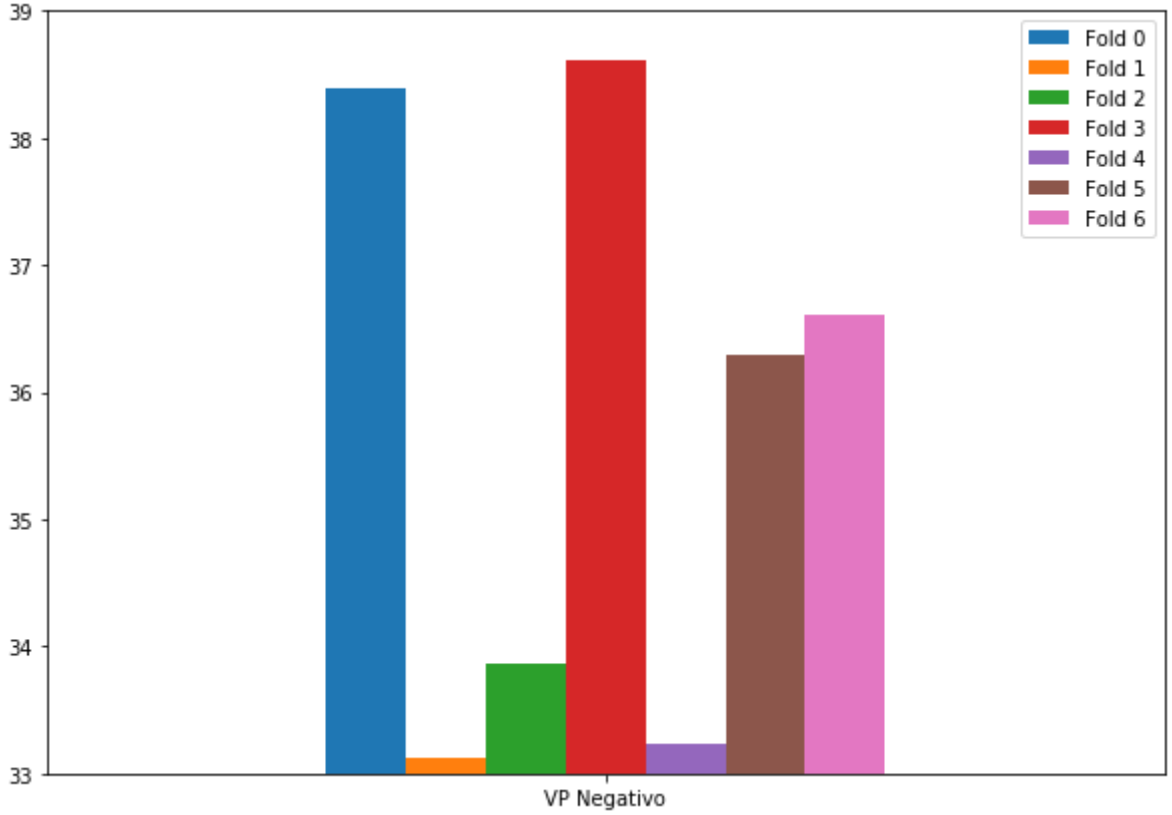

0.840
0.835
0.830
0.825
0.820
0.815
0.810
0.805
0.800
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Acurácia

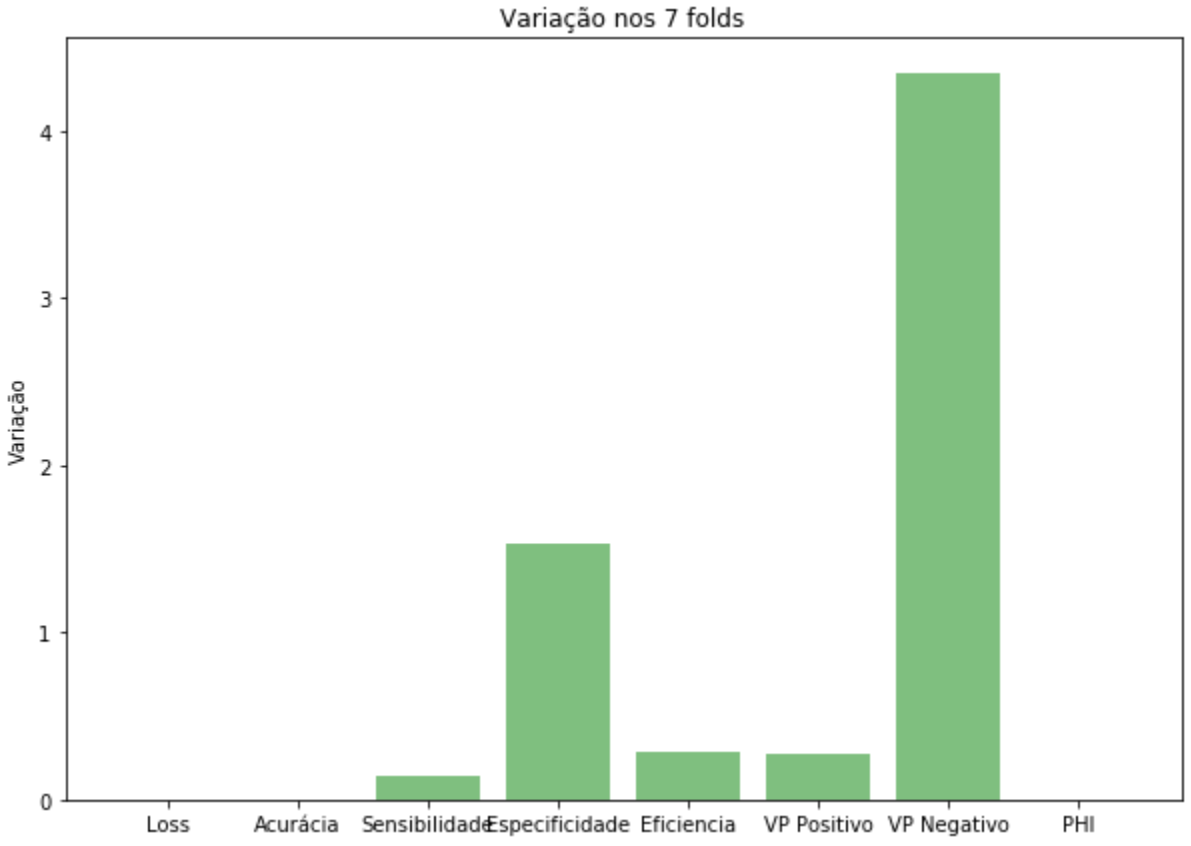

0.500
0.475
0.450
0.425
0.400
0.375
0.350
0.325
0.300
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
PHI


85.00
84.75
84.50
84.25
84.00
83.75
83.50
83.25
83.00
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Sensibilidade


72
71
70
69
68
67
66
65
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Especificidade


97.0
96.5
96.0
95.5
95.0
94.5
94.0
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
VP Positivo

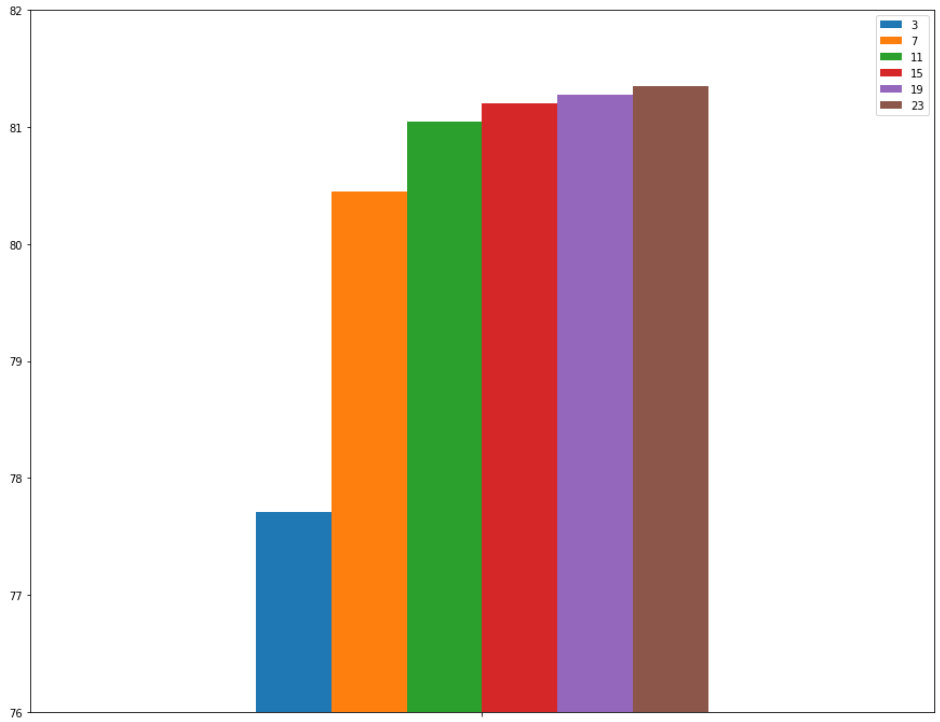

39
38
37
36
35
34
33
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
VP Negativo


Variação nos 7 folds
Variação
0
1
2
3
4
Loss
Acurácia
Sensibilidade
Especificidade
Eficiencia
VP Positivo
VP Negativo
PHI

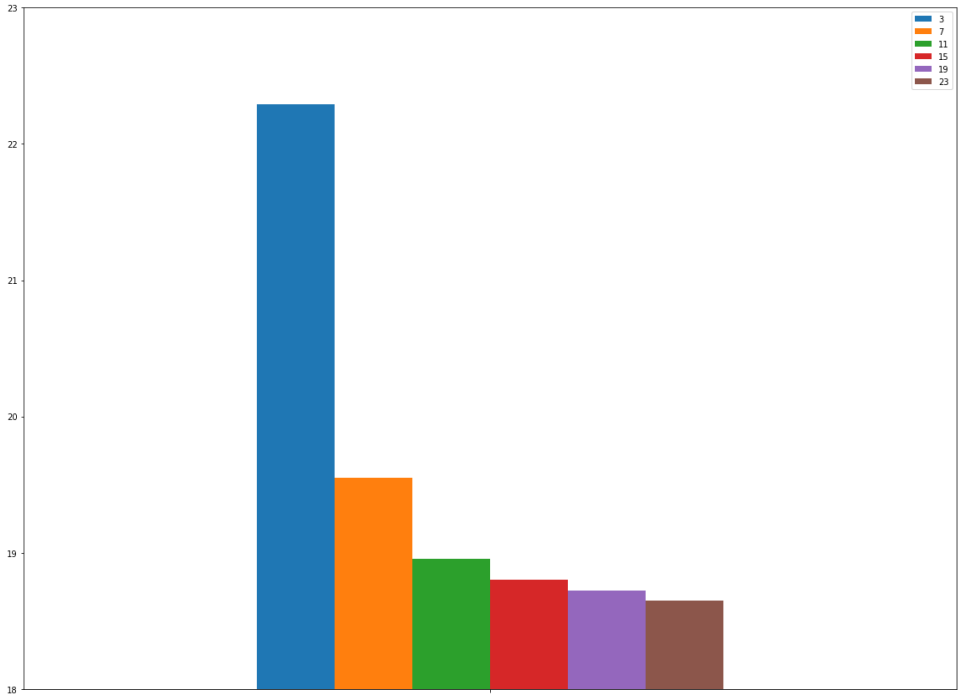
5.3 kNN

Neighbors de 3 a 23 variando de 4 em 4 com pesos uniformes

Acertos

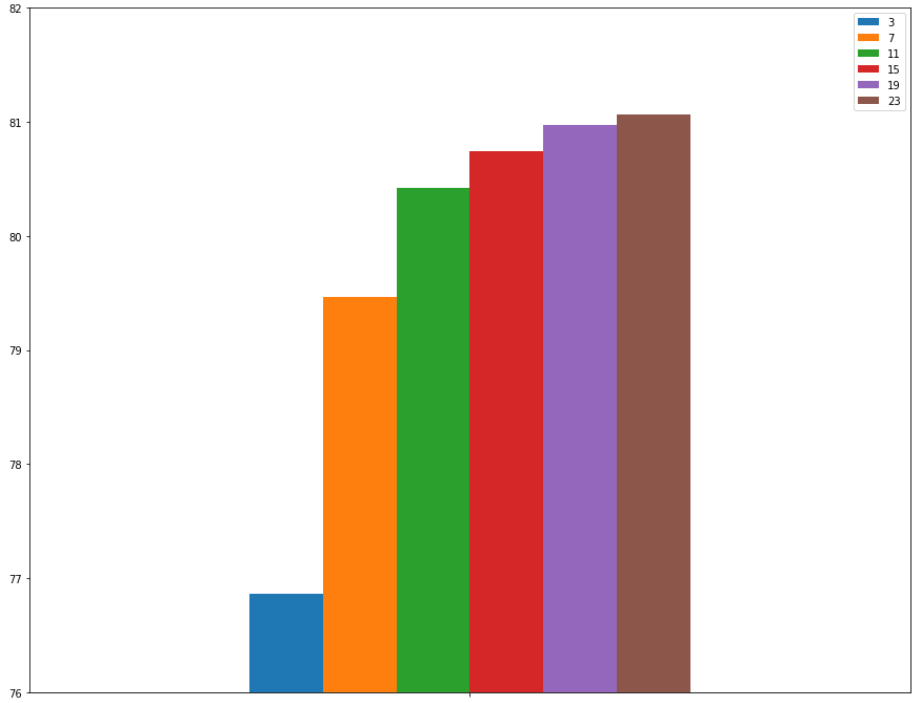


Erros

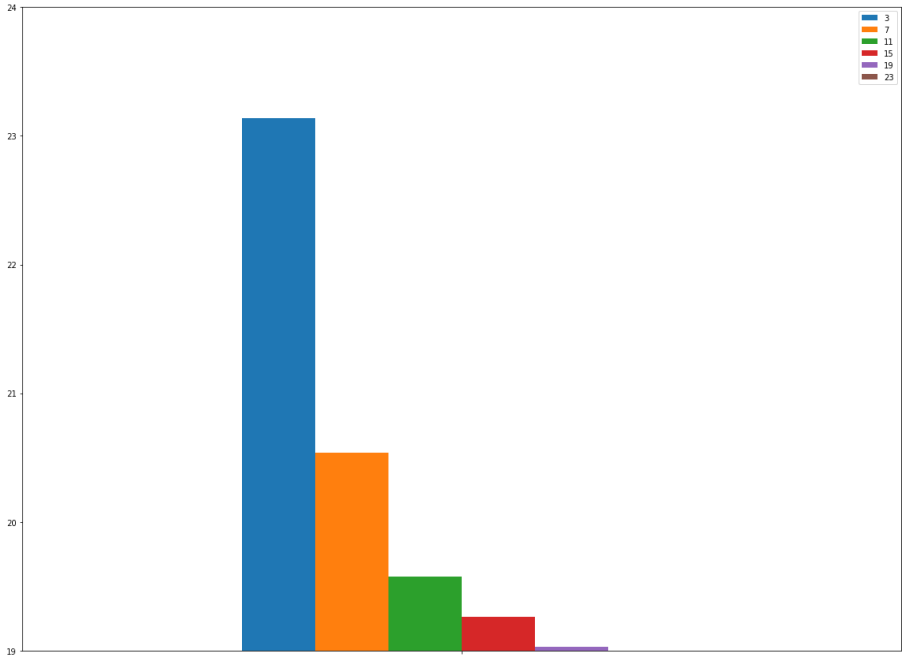


Neighbors de 3 a 23 variando de 4 em 4 com pesos baseados na distância

Acertos

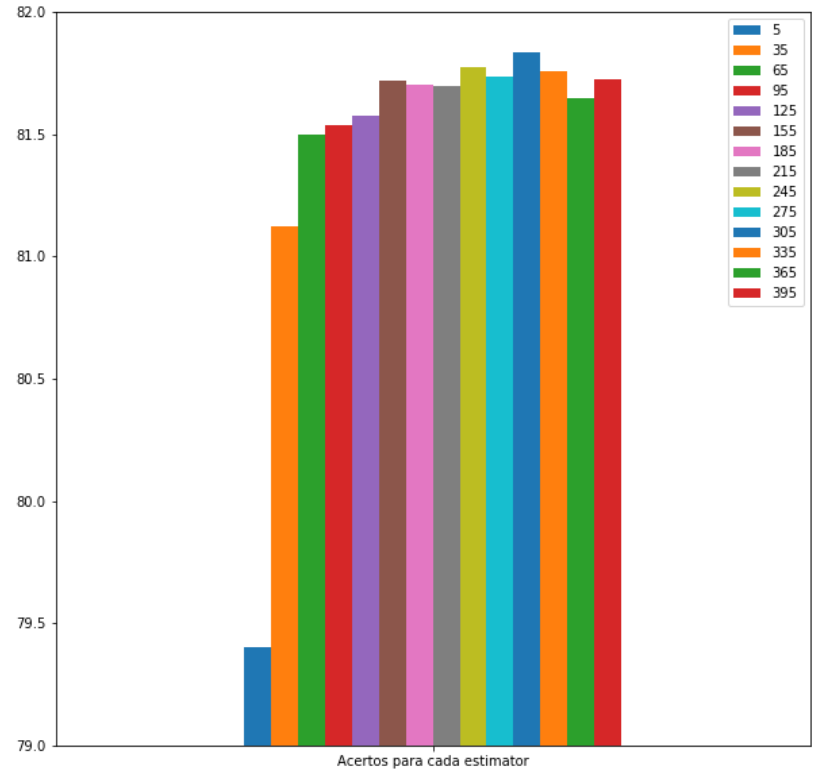


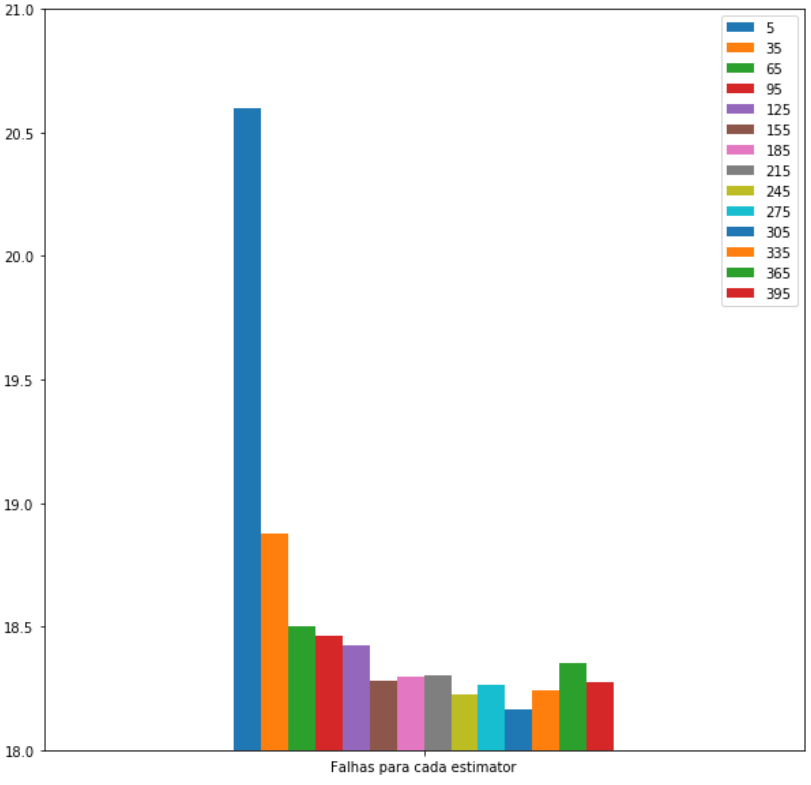
Erros



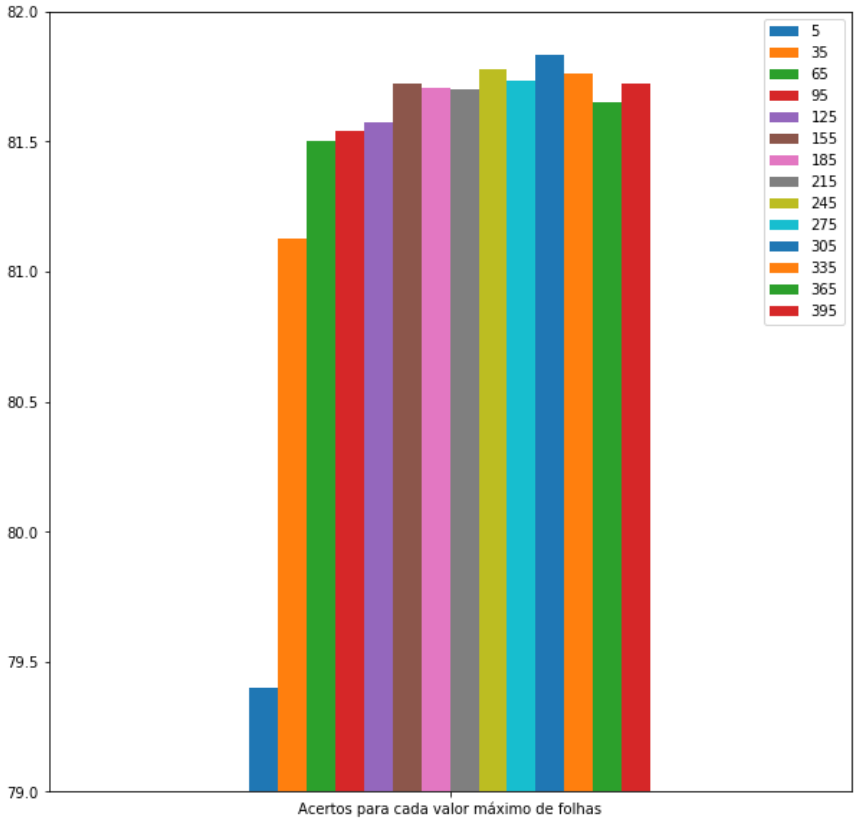
5.4 Random Forest

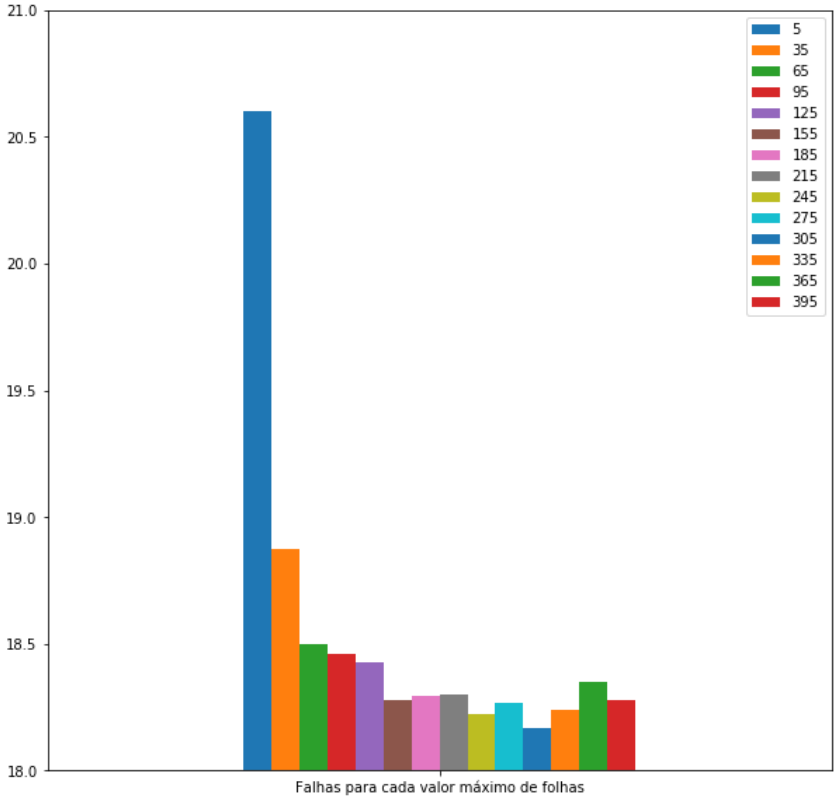
Estimators variando de 5 a 400 de 30 em 30





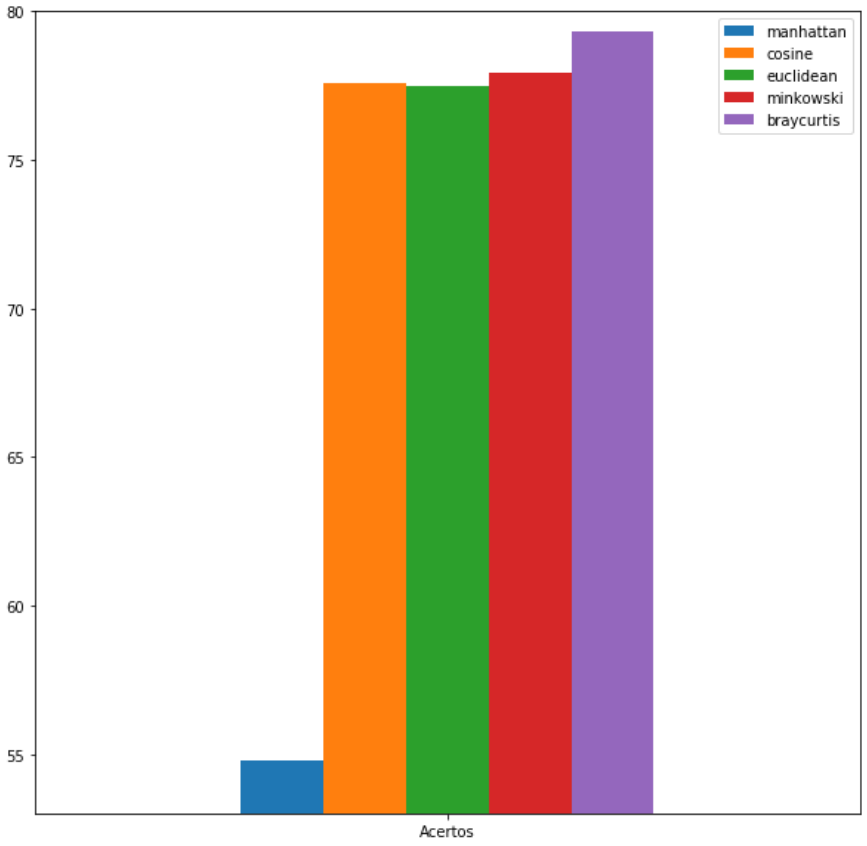
Máximo de folhas variando de 5 a 400 de 30 em 30

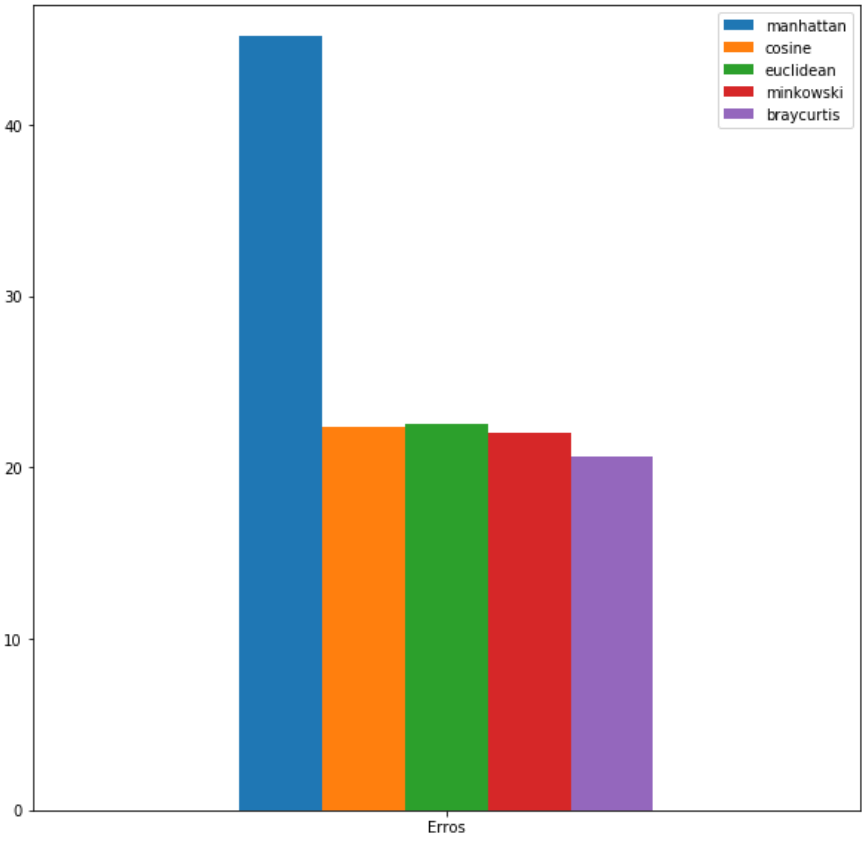

82.0
81.5
81.0
80.5
80.0
79.5
79.0
5
35
65
95
125
155
185
215
245
275
305
335
365
395
Acertos para cada valor máximo de folhas


21.0
20.5
20.0
19.5
19.0
18.5
18.0
5
35
65
95
125
155
185
215
245
275
305
335
365
395
Falhas para cada valor máximo de folhas

5.5 K-Means

Variando as métricas de medida de distância





6.0 Conclusão

6.1 K-Means

Ao final de toda análise, foi possível perceber que, de todas as técnicas utilizadas, a k-means foi a que mais precisou do pré-processamento. Ainda sobre ela, foi possível comparar várias técnicas para medir distância além da euclidiana. Dessas técnicas, a braycurtis foi a que se saiu melhor e a manhattan expressou o pior resultado.

6.2 Random Forest

A Random Forest foi a técnica menos reativas a ausência do pré-processamento. Além disso, ficou claro que não é ideal utilizar 5 estimators nem um máximo de 5 folhas. A medida em que os estimators e número máximo de folhas cresceu, o resultado melhorou um pouco, mas logo voltou a cair para valores acima de 305.

6.3 kNN

Assim como na Random Forest, a kNN não reage bem a apenas 5 neighbors. A medida em que o número de neighbors cresceu, o resultado foi melhor, mas acabou caindo. Com respeito a forma de calcular os pesos, calculando uniformemente alcançou um maior número de acertos. O melhor resultado apareceu com 23 neighbors, com um total de 81,35% de acertos e passou a cair, chegando a pouco menos de 80% em alguns casos.

6.4 Keras

A partir da matriz de confusão gerada com o keras, foi possível extrair algumas métricas. Além disso, foi possível avaliar cada um dos 7 folds por meio dessas métricas. Dentre elas, a especificidade no fold 4 ficou um pouco diferente dos demais. Outro valor que varia mais que os outros é o VP Negativo.