



Desempenho de classificadores binários em conjunto de dados multi classe

P. De N. Vasconcellos *J. Wainer*

Relatório Técnico - IC-PFG-17-01

Projeto Final de Graduação

2017 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Desempenho de classificadores binários em conjunto de dados multi classe

Pedro De N. Vasconcellos Jacques Wainer*

6 de julho de 2017

Resumo

Através deste trabalho, procuramos entender melhor o problema da classificação de dados multi classe. Buscamos compreender o desempenho, tanto em termos de acurácia quanto de tempo, de algoritmos inherentemente binários, explorando novas abordagens como a busca interna de hiper parâmetros. Nos atemos em explorar as diferenças entre a busca interna e externa de hiper parâmetros para as modelagens *One vs. One* e *One vs. Rest*, as quais utilizam algoritmos binários para a classificação. Ficamos contentes com os resultados obtidos, uma vez que eles permitiram um melhor entendimento a respeito deste problema e de outras possíveis maneiras que podemos solucioná-lo. Foram realizadas comparações envolvendo tempo de execução e acurácia dos classificadores e pudermos concluir que existem sim diferenças significativas entre a busca interna e externa de hiper parâmetros, assim como existem diferenças entre os próprios algoritmos binários utilizados nos classificadores *OvO* e *OvR*.

Sumário

1	Introdução	1
2	Metodologia	2
2.1	Conjunto de dados e algoritmos	3
2.2	Pseudo código das implementações <i>OvO</i> e <i>OvR</i>	4
2.3	Análise Estatística	6
3	Teste Realizados e Resultados	7
4	Conclusão	9

1 Introdução

A classificação de conjuntos de dados é um problema relevante para se estudar atualmente, vide a magnitude de aplicações que passam a considerar aprendizado de máquina como

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

parte fundamental da solução. Diante deste acelerado desenvolvimento e da impressionante quantidade de informação gerada pelo nosso estilo de vida, os próprios conjuntos de dados utilizados para treinar estes algoritmos passam, naturalmente, a representar informações intrinsecamente complexas; complexidade esta, que nem sempre possibilita modelar um problema de maneira binária.

Nosso trabalho procura justamente tratar destes conjuntos de dados que não são binários, mas multi classe, isto é: conjuntos de dados que como um todo, representam mais de duas possibilidades.

Existem algoritmos inherentemente multi classe, tais como NAIVE BAYES, LDA, QDA, DECISION TREES, RANDOM FORESTS e NEAREST NEIGHBOURS; o nosso projeto não foca nestes algoritmos, mas em algoritmos que são binários por natureza, ou seja, conseguem resolver problemas de classificação no qual ou um dado pertence à uma determinada classe (1) ou não pertence (0). Assim, no âmbito de modelar problemas multi classe com algoritmos binários, existem basicamente duas abordagens: **One vs. One**, (*OvO, para futuras referências*) e **One vs. Rest**, (*OvR*). Uma implementação *OvR* envolve treinar N sub-classificadores binários para N classes; cada sub-classificador é treinando para uma classe específica (1 para a classe que ele representa, 0 para todas as outras). No caso de *OvO*, $\frac{N*(N-1)}{2}$ sub-classificadores binários são treinados somente entre duas classes específicas. No que se diz respeito à implementação dos algoritmos binários da biblioteca *sklearn* [1], a busca por hiper parâmetros é uma busca externa, isto é, todos os sub-classificadores treinados (sejam os N do *OvR* ou os $\frac{N*(N-1)}{2}$ do *OvO*) compartilham os mesmos hiper parâmetros; por padrão, a abordagem utilizada por estes algoritmos é *OvR*.

Exceto pelo algoritmo SVC – que implementa nativamente *OvO* — temos duas abordagens que não são utilizadas por padrão (*OvO* e a busca interna por hiper parâmetros), sendo que a busca interna não é implementada na biblioteca *sklearn* de forma alguma. Buscamos explorar, através deste trabalho, os seguintes questionamentos:

- As diferenças dos classificadores *OvO* e *OvR*, para os algoritmos: SVC com KERNEL RBF [3], LOGISTIC REGRESSION [5] e LINEAR SVC [4].
- A diferença entre a seleção de hiper parâmetros interna ou externa para os classificadores *OvO* e *OvR*

Desta forma, o nosso trabalho propõe-se a entender as diferenças e possíveis benefícios de modelar através de classificadores *OvO* assim como utilizar busca interna de hiper parâmetros.

2 Metodologia

Nesta sessão, são explicadas as decisões tomadas durante o desenvolvimento do projeto, assim como ele foi realizado.

abalone	annealing	balance-scale
breast-tissue	car	cardiotocography-10clases
cardiotocography-3clases	chess-krvk	conn-bench-vowel-deterding
contrac	dermatology	energy-y1
energy-y2	flags	glass
hayes-roth	heart-cleveland	heart-switzerland
heart-va	image-segmentation	iris
led-display	lenses	letter
lung-cancer	molec-biol-splice	oocytes-merlucius-states-2f
oocytes-trisopterus-states-5b	optical	page-blocks
pendigits	pittsburg-bridges-MATERIAL	pittsburg-bridges-REL-L
pittsburg-bridges-SPAN	pittsburg-bridges-TYPE	plant-margin
plant-shape	plant-texture	primary-tumor
seeds	semeion	soybean
statlog-image	statlog-landsat	statlog-shuttle
statlog-vehicle	steel-plates	synthetic-control
teaching	thyroid	vertebral-column-3clases
wall-following	waveform	waveform-noise
wine	wine-quality-red	wine-quality-white
yeast	zoo	

Tabela 1: Nome dos conjuntos de dados utilizados no projeto

Acrônimo	Algoritmo
SVC	SUPPORT VECTOR CLASSIFICATION
LINEAR SVC	LINEAR SUPPORT VECTOR CLASSIFICATION
LR	LOGISTIC REGRESSION

Tabela 2: Algoritmos do *sklearn* [1] utilizados no projeto

2.1 Conjunto de dados e algoritmos

Todos os 60 conjuntos de dados da Tabela 1 foram disponibilizados pelo professor Jacques. Estes dados contém um número variado de classes — 3 até 10 —, garantia de um conjunto suficientemente heterogêneo para analisar problemas de classificação multi classe.

A decisão que acarretou na utilização dos 3 algoritmos da Tabela 2, deriva justamente de todos serem, necessariamente, binários. Esta é uma decisão importante, pois, como pretendemos explorar a implementação *OvO*, os algoritmos utilizados devem ser binários. A escolha destes três em particular cai no fato de que tanto o SVC quanto o LINEAR SVC representam algoritmos mais robustos e modernos em comparação com o LR, porém, também quisemos utilizá-lo para compararmos seu desempenho.

Como foi brevemente explicado, a biblioteca *sklearn* [1] não implementa busca interna por hiper parâmetros e nem todos os algoritmos binários são capazes da classificação *OvO*.

Assim, a fim de garantir uma comparação equiparável entre todos os nossos classificadores, foi utilizada uma camada de abstração da própria biblioteca [6] para o caso da busca externa, na qual cria, com base em um algoritmo qualquer da biblioteca, um classificador genérico *OvO* ou *OvR*. Esta camada de abstração garante uma busca por hiper parâmetros externa, ou seja, todos os sub-classificadores terão os mesmos hiper parâmetros — no caso da busca interna, nós tivemos que criar programaticamente todos sub-classificadores eliminando a necessidade da camada de abstração.

A busca interna por hiper parâmetros teve de ser implementada e constituiu desenvolvimento de código a fim de garantir que o funcionamento fosse similar àquele proporcionado pela camada de abstração. Desta forma, dependendo da abordagem (*OvO* ou *OvR*), tivemos a necessidade de criar o número correto de sub-classificadores assim como o método que classifica uma previsão como verdadeira ou falsa.

- Utilizaremos 60 bancos de dados multi classe, nos quais serão executados os algoritmos.
- Utilizaremos 3 algoritmos binários (garantindo a consistência de implementação): SVC, LINEAR SVC e LR.
- Para a busca externa de hiper parâmetros, será utilizada a camada de abstração providenciada pela biblioteca *sklearn*, tanto para o classificador *OvO* quanto para o *OvR*.
- Para a busca interna de hiper parâmetros, será implementado tanto a criação dos classificadores filhos quanto os métodos que calculam quantas previsões foram corretas, para ambos os classificadores (*OvO* e *OvR*).

2.2 Pseudo código das implementações *OvO* e *OvR*

A fim de facilitar a leitura dos pseudo códigos de ambas implementações, explicaremos, brevemente, o que é um FOLD EXTERNO e um FOLD INTERNO. Para um conjunto de dados X, um FOLD EXTERNO é uma amostra estratificada deste conjunto X, isto é, uma partição deste conjunto na qual é preservada a proporção de classes do conjunto X. Um FOLD INTERNO segue a mesma lógica, contudo é uma amostra estratificada de um FOLD EXTERNO. Assim, ambas soluções — *OvO* e *OvR* — abordadas no nosso projeto utilizam 5 FOLDS EXTERNOS e 3 FOLDS INTERNOS.

No caso, não há indicação dos FOLDS INTERNOS nos pseudo códigos, contudo, existe a função *Grid Search*, implementada pela biblioteca *sklearn*. Esta função cria os FOLDS INTERNOS e realiza uma busca de hiper parâmetros nestes FOLDS INTERNOS, tal que ela retorna o melhor conjunto de hiper parâmetros. No nosso caso, o nosso *Grid Search* utiliza

3 FOLDS INTERNOS.

```

input      : Arquivo CSV referente ao conjunto de dados
1 X ← CSV;
2 foldsExternos ← StratifiedKFold (K=5, data=X);
3 Tempo 1;
4 previsoesTotalNegativas ← 0;
5 previsoesTotalPositivas ← 0;
6 for i ← 1 to 5 do
7   // Melhores hiper parâmetros, melhor acurácia
8   resultadosGridSearch, previsoesPositivas, previsoesNegativas ← GridSearch
     (foldsExternos [i]);
9   foldsExternos [i].csv ← resultadosGridSearch;
10  previsoesTotalPositivas ← previsoesTotalPositivas + previsoesPositivas;
11  previsoesTotalNegativas ← previsoesTotalNegativas + previsoesNegativas;
12 end
13 Tempo 2;
14 resultadosGridSearch, previsoesPositivas, previsoesNegativas ← GridSearch (X);
15 Tempo 3;
16 resultados_agregados.csv ← resultadosGridSearch, previsoesTotalPositivas,
  previsoesTotalNegativas;

```

Algorithm 1: Fluxo de execução do algoritmo de classificação com busca externa de hiper parâmetros

```

input      : Conjunto de dados X
output    : classificadoresInfo
1 numeroClassificadores ← numeroClassificadores (X);
2 classificadoresInfo = {};
3 for i ← 1 to numeroClassificadores do
4   resultadosGridSearch, previsoesPositivas, previsoesNegativas ← GridSearch (X);
5   classificadoresInfo [i] = (resultadosGridSearch, previsoesPositivas,
     previsoesNegativas);
6 end

```

Algorithm 2: Função Auxiliar SubClassificadores

De acordo com os algoritmos 1 e 3, cada uma dos testes executados gera 6 arquivos CSV com informações relevantes, tais como: algoritmos utilizado, número de previsões verdadeiras e falsas, tempos coletados, melhores hiper parâmetros, etc...

Com todos estes arquivos, nos concentrarmos em como realizar análise estatísticas que pudessem responder nossos questionamentos. Decidimos, por fim, que focaríamos em duas métricas:

- Tempo de treinamento e previsão dos classificadores (ou seja, Tempo 3 - Tempo 1)
- Acurácia dos FOLDS EXTERNOS

```

input      : Arquivo CSV referente ao conjunto de dados
1 X ← CSV;
2 foldsExternos ← StratifiedKFold (K=5, data=X);
3 Tempo 1;
4 previsoesTotalNegativas ← 0;
5 previsoesTotalPositivas ← 0;
6 for i ← 1 to 5 do
7   classificadoresInfo ← SubClassificadores (foldsExternos [i]);
8   foldsExternos [i].csv ← classificadoresInfo;
9   previsoesTotalPositivas ← previsoesTotalPositivas + classificadoresInfo
10  [previsoesPositivas];
10  previsoesTotalNegativas ← previsoesTotalNegativas + classificadoresInfo
11  [previsoesNegativas];
11 end
12 Tempo 2;
13 classificadoresInfo ← SubClassificadores (X);
14 Tempo 3;
15 resultados_agregados.csv ← classificadoresInfo;

```

Algorithm 3: Fluxo de execução do algoritmo de classificação com busca interna de hiper parâmetros

Foi escolhida a acurácia do FOLD EXTERNO, pois no caso da busca interna, cada sub-classificador tinha uma acurácia, o que tornava a comparação incompatível com o caso da busca externa, que só tinha um hiper parâmetro. Porém, ambas soluções apresentam uma acurácia para o FOLD EXTERNO, sendo esta a utilizada como métrica.

2.3 Análise Estatística

Temos 60 conjuntos de dados, 3 algoritmos binários, cada um com duas abordagens implementadas (*OvO* e *OvR*), ocasionando em 360 testes. Como ainda resta a busca interna e externa por hiper parâmetros, temos ao todos 720 testes para comparar e realizar medidas estatísticas.

Para fins dos testes estatísticos, utilizamos a biblioteca SCIPY [2] e o teste dos postos sinalizados de Wilcoxon — teste de hipóteses não paramétrico utilizado quando se deseja comparar duas amostras relacionadas a fim de avaliar se os postos médios populacionais diferem. Esse teste não avalia o valor das amostras, mas quanto a subtração das amostras é simétrica em relação ao zero.

O teste de Wilcoxon retorna dois valores: P VALUE e Z STATISTIC; no entanto, nos atemos ao valor de P VALUE. Utilizando um cutoff α de 0,05, ou seja, obtendo um valor menor que o cutoff podemos rejeitar a hipótese nula de que os pares de amostras pertencem ao mesmo grupo e, portanto, existe uma diferença significativa entre as amostras e, por conseguinte, existe uma diferença significativa no que estamos considerando em cada teste.

Mesmo que o teste de Wilcoxon não leve em consideração o valor das amostras em

consideração, elas serão analisadas a fim de prover conclusões mais fundamentadas.

3 Teste Realizados e Resultados

A Tabela 3 contém, quais testes foram realizados, isto é, o que estávamos comparando.

Nº	Testes
1	Compara a acurácia entre <i>OvO</i> e <i>OvR</i> com busca externa de hiper parâmetros
2	Compara a acurácia entre <i>OvO</i> e <i>OvR</i> com busca interna de hiper parâmetros
3	Compara o tempo entre <i>OvO</i> e <i>OvR</i> com busca externa de hiper parâmetros
4	Compara o tempo entre <i>OvO</i> e <i>OvR</i> com busca interna de hiper parâmetros
5	Compara a acurácia entre os 3 algoritmos com busca externa de hiper parâmetros
6	Compara a acurácia entre os 3 algoritmos com busca interna de hiper parâmetros
7	Compara o tempo entre os 3 algoritmos com busca externa de hiper parâmetros
8	Compara o tempo entre os 3 algoritmos com busca interna de hiper parâmetros
9	Compara a acurácia de <i>OvO</i> e <i>OvR</i> entre busca externa e busca interna de hiper parâmetros
10	Compara o tempo de <i>OvO</i> e <i>OvR</i> entre busca externa e busca interna de hiper parâmetros
11	Compara a acurácia de <i>OvO</i> e <i>OvR</i> entre busca externa e busca interna de hiper parâmetros, divididos em <i>OvO</i> e <i>OvR</i>
12	Compara o tempo de <i>OvO</i> e <i>OvR</i> entre busca externa e busca interna de hiper parâmetros, divididos em <i>OvO</i> e <i>OvR</i>

Tabela 3: Testes Realizados

A Tabela 4 contém os resultados obtidos para cada um dos testes.

Nº	P Value	Mediana das amostras
1	$2,06 \times 10^{-7}$	$OvO = 83,19 \%$ $OvR = 80,86 \%$
2	$6,01 \times 10^{-27}$	$OvO = 82,86 \%$ $OvR = 72,62 \%$
3	$6,11 \times 10^{-2}$	$OvO = 245,43 \text{ s}$ $OvR = 237,17 \text{ s}$
4	$9,79 \times 10^{-12}$	$OvO = 61,77 \text{ s}$ $OvR = 28,07 \text{ s}$
5	$SVC = 2,14 \times 10^{-1}$ $LINEAR SVC = 1,26 \times 10^{-4}$ $LR = 4,22 \times 10^{-4}$	$SVC\ OvO = 84,57 \%$ $SVC\ OvR = 84,57 \%$ $Linear\ SVC\ OvO = 82,41 \%$ $Linear\ SVC\ OvR = 79,22 \%$ $LR\ OvO = 82,60 \%$ $LR\ OvR = 79,86 \%$
6	$SVC = 8,51 \times 10^{-11}$ $LINEAR SVC = 4,07 \times 10^{-8}$ $LR = 2,84 \times 10^{-11}$	$SVC\ OvO = 83,54 \%$ $SVC\ OvR = 83,54 \%$ $Linear\ SVC\ OvO = 79,93 \%$ $Linear\ SVC\ OvR = 79,60 \%$ $LR\ OvO = 83,26 \%$ $LR\ OvR = 69,59 \%$
7	$SVC = 1,41 \times 10^{-1}$ $LINEAR SVC = 5,56 \times 10^{-1}$ $LR = 2,21 \times 10^{-2}$	$SVC\ OvO = 273,29 \text{ s}$ $SVC\ OvR = 246,85 \text{ s}$ $Linear\ SVC\ OvO = 215,37 \text{ s}$ $Linear\ SVC\ OvR = 228,74 \text{ s}$ $LR\ OvO = 265,02 \text{ s}$ $LR\ OvR = 237,18 \text{ s}$
8	$SVC = 4,60 \times 10^{-2}$ $LINEAR SVC = 2,60 \times 10^{-3}$ $LR = 1,80 \times 10^{-11}$	$SVC\ OvO = 62,68 \text{ s}$ $SVC\ OvR = 25,38 \text{ s}$ $Linear\ SVC\ OvO = 60,53 \text{ s}$ $Linear\ SVC\ OvR = 34,32 \text{ s}$ $LR\ OvO = 61,77 \text{ s}$ $LR\ OvR = 25,48 \text{ s}$
9	$7,43 \times 10^{-32}$	$Busca\ Interna = 78,29 \%$ $Busca\ Externa = 82,42 \%$
10	$3,70 \times 10^{-35}$	$Busca\ Interna = 41,69 \text{ s}$ $Busca\ Externa = 241,58 \text{ s}$
11	$OvO = 2,70 \times 10^{-4}$ $OvR = 6,49 \times 10^{-27}$	$Busca\ Interna\ OvO = 82,86 \%$ $Busca\ Externa\ OvO = 83,19 \%$ $Busca\ Interna\ OvR = 72,62 \%$ $Busca\ Externa\ OvR = 80,85 \%$
12	$OvO = 4,46 \times 10^{-16}$ $OvR = 6,91 \times 10^{-21}$	$Busca\ Interna\ OvO = 61,77 \text{ s}$ $Busca\ Externa\ OvO = 245,43 \text{ s}$ $Busca\ Interna\ OvR = 28,07 \text{ s}$ $Busca\ Externa\ OvR = 237,17 \text{ s}$

Tabela 4: P Value e mediana dos testes realizados

4 Conclusão

Lembrando que temos um cutoff de $\alpha = 5 \times 10^{-2}$, estas são as seguintes conclusões que podemos tirar do conjunto de testes realizados:

- O resultado do primeiro teste nos afirma que existe sim uma diferença significativa entre os classificadores *OvO* e *OvR* utilizando a busca externa por hiper parâmetro, contudo, ao analisar a mediana da acurácia, percebe-se que a diferença não é tão importante. De qualquer forma, a abordagem *OvO* resulta, na maioria das execuções, em acurácia maiores. O segundo resultado corrobora com a conclusão do primeiro, *OvO* resulta em acurácia maiores (mesmo na busca interna de hiper parâmetros), contudo, vale a ressalva de que neste teste, as diferenças de acurácia eram mais representativas — variando de 5% até 40%.
- Já o resultado do terceiro teste deixa claro, que para a busca externa de hiper parâmetros, não existe uma diferença significativa na duração, o que fortalece a ideia de utilizar a abordagem *OvO* sempre que possível — previsões com mais acurácia sem custo adicional de tempo.
- O quarto teste, apesar de indicar que existe uma diferença significativa, a conclusão não é tão clara, pois ambas abordagens apresentam variações inconsistentes no tempo (em segundos), como por exemplo:
 - thyroid_SVC, *OvO* = 142,013, *OvR* = 178,05
 - chess-krvk_SVC, *OvO* = 10729,68, *OvR* = 48092,78
 - wine-quality-white_LR, *OvO* = 3883,82, *OvR* = 41,86
 - cardiotocography-10clases_LR, *OvO* = 515,19, *OvR* = 99,34
 - cardiotocography-10clases_Linear-SVC, *OvO* = 353,73, *OvR* = 107,08
 - cardiotocography-10clases_SVC, *OvO* = 372,43, *OvR* = 200.20
 - cardiotocography-3clases_LR, *OvO* = 46,21, *OvR* = 29.30
- O resultado do quinto teste nos demonstra que, com busca externa de hiper parâmetros, não existe uma diferença significativa na acurácia entre *OvO* e *OvR* ao considerar separadamente os algoritmos, ou seja, o resultado do primeiro teste é, em grande parte, resultado da diferença dos algoritmos LINEAR SVC e LR.
- O resultado do sexto teste confirma as conclusões do segundo teste.
- O resultado do sétimo teste também confirma que não existe diferença significativa no tempo de execução com busca externa de hiper parâmetros (corrobora com a conclusão do terceiro teste)
- O resultado do oitavo teste nos mostra que o algoritmo SVC apresenta diferença significativa, mas não tanto quanto os outros algoritmos. Assim, fica claro que o SVC apresenta uma consistência no tempo de execução em ambas abordagens (busca interna e externa de hiper parâmetros)

- Os resultados do nono e décimo testes são possivelmente os mais interessantes. Basicamente, fica evidente, pelo nono teste, que a busca interna por hiper parâmetro apresenta uma diferença significativa em relação a busca externa. O resultado interessante é que é a busca externa que apresenta, na maioria das vezes, acurácia maior nas execuções. Já o resultado do décimo teste demonstra que existe uma diferença significativa também no tempo entre as duas abordagens de busca, contudo, desta vez, a busca interna apresenta, na maioria das vezes, tempos menores de execução.
- Os dois últimos testes confirmam o resultado estatístico do nono e do décimo, apenas mostrando quais abordagens apresentam maior diferença significativa.

Analizando juntamente as conclusões que cada teste nos proporcionou, podemos responder as perguntas que motivaram a existência deste projeto.

Existem diferenças, tanto em acurácia quanto no tempo de execução, entre a busca externa e interna por hiper parâmetros. Primeiramente, para ambas, existe uma diferença significativa na abordagem da modelagem — *OvO* ou *OvR* — visto que *OvO* resulta, na maioria das execuções, em acurácia maior. Na métrica de tempo, a busca externa entre *OvO* e *OvR* não apresenta diferença significativa, o que reforça o uso de *OvO*. Este mesmo resultado não se repete para a busca interna, na qual apresenta tempos de execução significativamente diferentes entre *OvO* e *OvR*.

Ao comparar individualmente os algoritmos, percebe-se que com a busca externa, não existe diferença significativa em relação ao tempo entre *OvO* e *OvR*. Quando se trata do algoritmo SVC, percebe-se que a acurácia não é significativamente diferente, evidenciando uma descoberta interessantíssima; este fato não ocorre com os outros dois algoritmos. Ainda mais, em relação à métrica tempo, o algoritmo SVC não apresenta diferença significativa entre a busca interna e externa. Assim, as principais diferenças de tempo e acurácia ao comparar todos os algoritmos juntos, derivam, principalmente, dos outros dois algoritmos.

Comparando, como um todo, a busca interna com a busca externa, fica claro que existem um comprometimento que deve ser levado em conta: enquanto que a busca externa apresenta, na maioria das vezes, acurácia maior, a busca interna apresenta menores tempo de execução.

Percebe-se que através das soluções propostas e conclusões que este problema não é, de forma, alguma intuitivo. Ficamos contentes com os resultados obtidos, pois permitem ampliar o conhecimento científico e computacional a respeito da classificação de conjunto de dados multi classe.

Referências

- [1] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, **Volume 12**, (2011)

- [2] Eric J. and Travis O. and Pearu P., *SciPy: Open source scientific tools for Python*, (2001)
- [3] Documentação SVC
- [4] Documentação Linear SCV
- [5] Documentação LR
- [6] Documentação Abstração Multi Classe