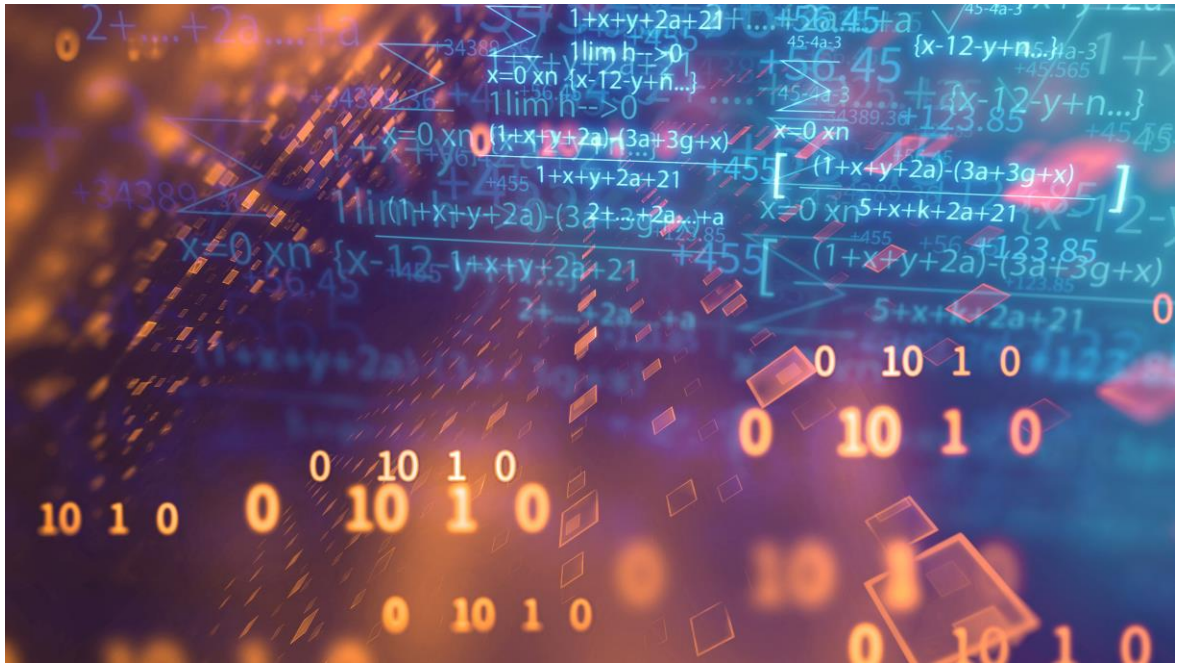
Projeto Aprendizagem Automática



Turma: LEIM-51D

Trabalho realizado por: Miguel Távora

Docente: Gonçalo Marques

Data: 8/2/2021

Introdução

- O objetivo deste trabalho é prever a classificação de críticas de cinemas em classificação entre os valores de 1-4 e 7-10 e classificação se a crítica foi positiva ou não.

- Para isso foram utilizados metodologias de treino e teste para treinar um classificador e posteriormente classificar de modo a obter o melhor resultado possível.

- Para obter os melhores parâmetros para um classificador foram feitos diversos testes, tentando assim obter o melhor resultado possível para a classificação dos dados de teste.

Funções de conversão

Existem 5 funções fundamentais para o desenvolvimento do projeto. Essas funções são nomeadamente:

- A função designada de *convert_indexes_into_binary* onde recebe como argumento as classificações entre 1 - 4 e 7 - 10 e retorna os índices entre 1 - 4 como 0 e os índices 7 - 10 como 1.

- A função *clear_data* recebe como argumento os dados de texto e aplica uma expressão regular nos mesmos obtendo desta forma textos somente com letras, números e caracteres acentuados. A função retorna os dados limpos pela expressão regular. Esta função recebe também outro argumento, caso seja *True* as *strings* recebidas são convertidas de *strings* binárias em *strings* Unicode.

- A função *stemming* que serve para aplicar o stemming nos textos, onde converte vários palavras relacionadas todas numa única palavra. A função recebe os textos e um argumento do tipo inteiro que dita qual o tipo de stemming realizado. O valor 1 é para o PorterStemmer, o valor 2 é para o LancasterStemmer e outro valor é para o SnowballStemmer.

- A função *convert_str_to_tf_idf* que retorna um objeto do tipo tf-idf, esta representação serve para poder posteriormente classificar os textos. A função recebe como argumento o min_df que representa o número de vezes que a palavra aparece num determinado número de corpus, o token_pattern que dita o tamanho mínimo da palavra para ser adicionada como token e os n_grams que adiciona mais tokens com mais do que uma palavra por entrada. Contudo os n_grams só é possível utilizar até aos bi-gramas os tri-gramas ocupam demasiada memória.

- A função *convert_to_sparce_matrix* que recebe o objeto tf-idf e os dados e retorna uma matriz esparsa com os dados. Esta função não foi incorporada na anterior porque por vezes é necessário saber o tamanho dos tokens na representação tf-idf.

Funções de teste

As funções de teste são decompostas em 3 funções principais e algumas funções complementares. Essas funções são:

- A função *test_logistic_regression* recebe como argumentos o X e y do treino e teste e também outros parâmetros úteis para melhorar o classificador. Esta função utiliza outras funções iniciadas por dois under-scores para através de números inteiros ser possível chamar os parâmetros pretendidos. Esta função cria um objeto do discriminante logístico e imprime na consola o resultado do score de treino e teste.

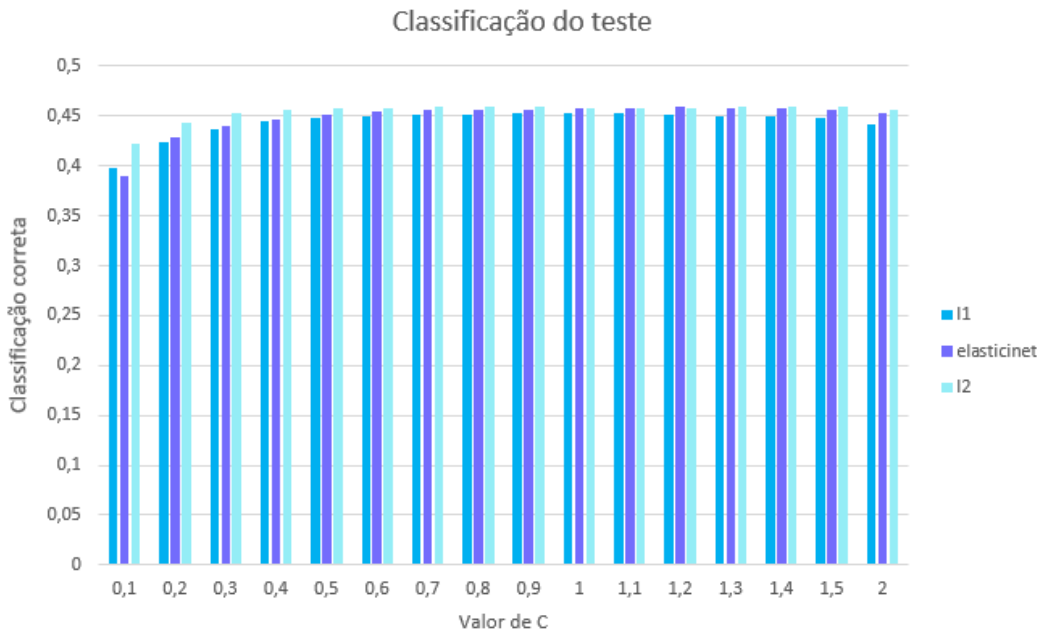
- A função *test_logistic_regression_coef* realiza o mesmo que a função anterior, mas imprime a quantidade de coeficientes utilizados. A necessidade de saber os coeficientes é importante quando se utiliza a penalização l1, porque esta penalização não utiliza todos os coeficiente para classificar.

- A função *test_ridge* que imprime o score da classificação e a quantidade de coeficientes utilizados na classificação.

Teste classificação multi-classe - discriminante logístico

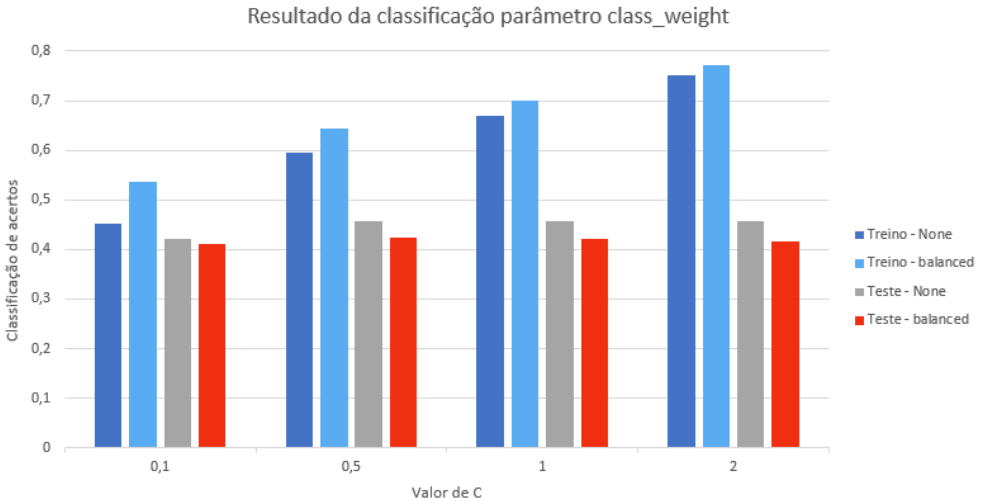
- O primeiro teste realizado foi a penalização l1, elasticinet com l1_ratio de 0.5 e l2 para diferentes valores de C.
- Este teste é realizado com separação em treino e teste, apesar de ser um teste muito específico o objetivo é observar a evolução dos diferentes penalizadores para varias regularizações na classificação nas mesmas condições.

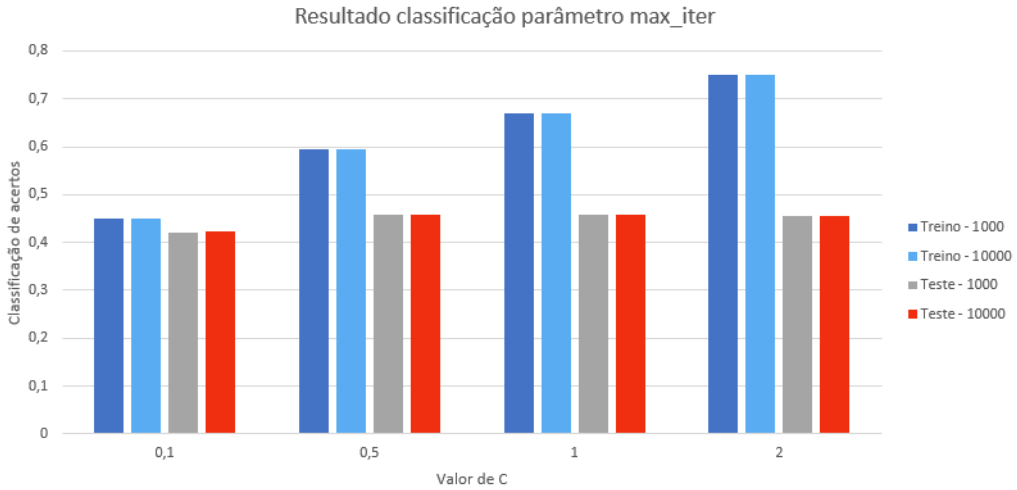
	penalização: l1			penalização:elasticnet l1_ratio=,5			penalização: l2		
Valor de C	Treino	Teste	Coeficientes utilizados	Treino	Teste	Coeficientes utilizados	Treino	Teste	Coeficientes utilizados
0,1	0,4017	0,3987	204	0,3933	0,3903	896	0,4512	0,4216	193752
0,2	0,4408	0,4244	550	0,4362	0,4286	2012	0,5061	0,4432	193752
0,3	0,4577	0,4367	880	0,4607	0,4394	3131	0,5436	0,453	193752
0,4	0,4704	0,4447	1281	0,4801	0,4461	4414	0,5717	0,4562	193752
0,5	0,4816	0,4474	1719	0,4968	0,4515	5916	0,5953	0,4581	193752
0,6	0,4916	0,4494	2160	0,5109	0,454	7531	0,6137	0,458	193752
0,7	0,4997	0,4509	2684	0,5253	0,456	9323	0,6295	0,4587	193752
0,8	0,5101	0,4516	3210	0,5367	0,4563	11112	0,6438	0,4589	193752
0,9	0,5187	0,4534	3817	0,5489	0,4569	13026	0,6575	0,4593	193752
1	0,5288	0,4524	4414	0,5606	0,458	14916	0,67	0,4585	193752
1,1	0,5394	0,4524	5078	0,5718	0,4584	16805	0,6797	0,4583	193752
1,2	0,5499	0,4514	5750	0,584	0,4588	18732	0,6901	0,4586	193752
1,3	0,5604	0,4504	6445	0,5945	0,4582	20658	0,6995	0,4594	193752
1,4	0,5715	0,4495	7100	0,6056	0,4573	22606	0,7083	0,4597	193752
1,5	0,5816	0,4483	7808	0,6164	0,4562	24550	0,7164	0,4594	193752
2	0,6322	0,4416	11448	0,6633	0,4524	33548	0,7506	0,4563	193752
	Máximo:	Máximo:	Máximo:	Máximo:	Máximo:	Máximo:	Máximo:	Máximo:	
	0,6322	0,4534	11448	0,6633	0,4588	33548	0,7506	0,4597	

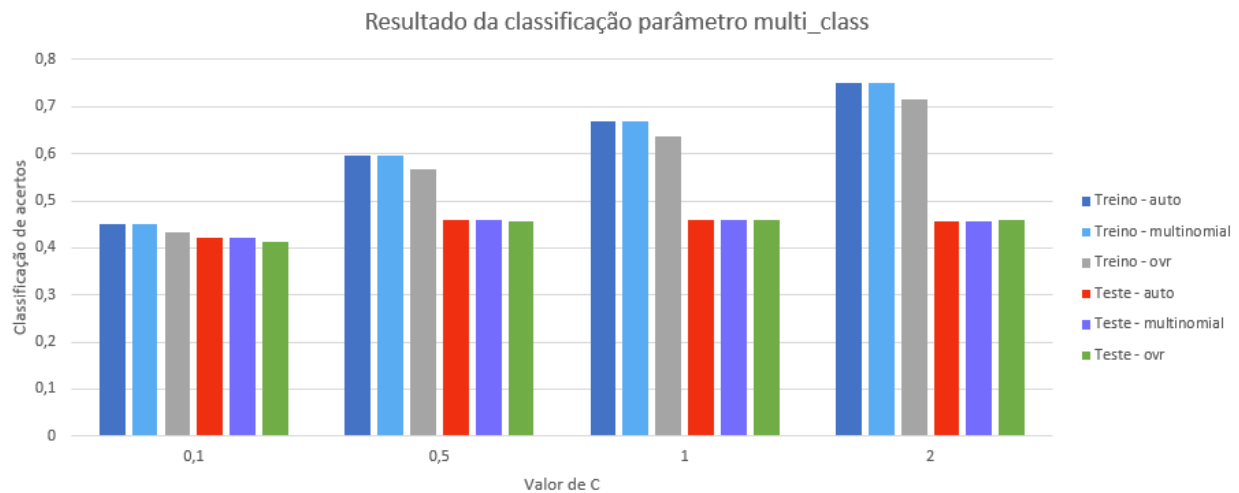


- Perante os resultados obtidos é possível verificar que os melhores valores para o treino estão concentrados nos maiores valores de C, isto deve-se a ter uma regularização mais baixa e por isso decora mais os dados.
- Em relação ao teste os melhores resultados estão entre os valores de 1 e 1,5 de C.
- Desta forma os valores da regularização posteriormente mais utilizados serão entre 1 e 1,5.

- De seguida foram testados diversos outros parâmetros do discriminante logístico. Nomeadamente o parâmetro *class_weight*, *multi_class* e *max_iter*.
- Todos os testes foram realizados nas mesmas condições para saber se era de facto benéfico utilizar um parâmetro em relação a outro.
- Também foram testados outros parâmetros como o *solver* contudo como a diferença estava somente no tempo de processamento, por isso não foram incluídos.

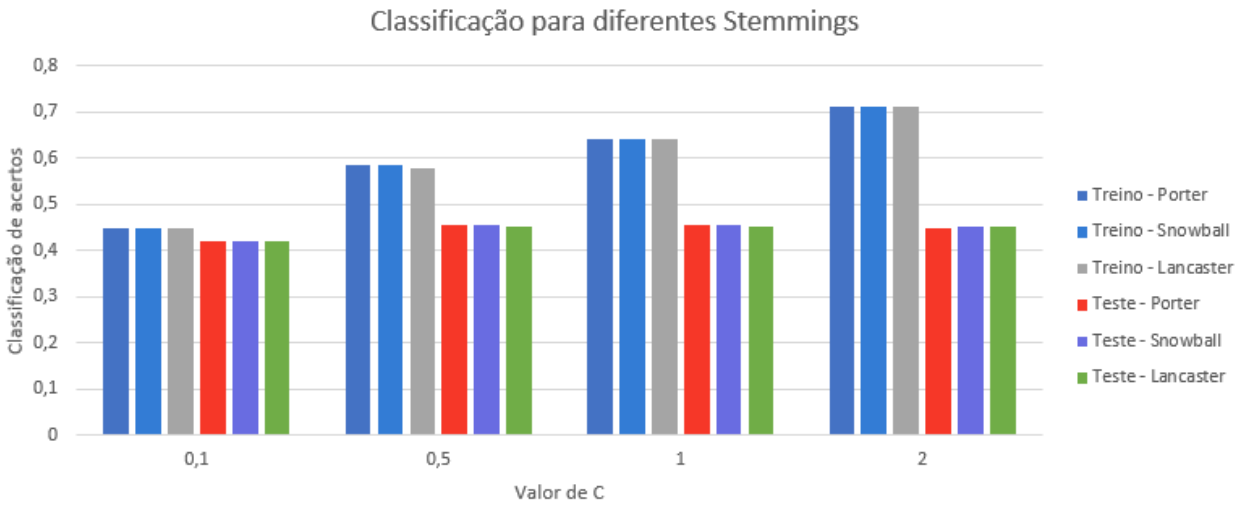






- Perante os resultados conclui-se que o parâmetro *class-weight* melhora o treino, mas piora o teste pelo que não será utilizado.
- O parâmetro *max_iter* os resultados variavam cerca de 0.01% e a forma como varia é bastante irregular, pelo que será utilizado o valor 1000.
- Em relação ao parâmetro *multi_class* os resultados são bastante inconclusivos pelo que foram sempre testados todos os parâmetros. Contudo com mais alguns testes conclui-se que o parâmetro auto em geral era sempre inferior aos restantes.

Testes de stemming e tf-idf



Tamanho minimo da palavra (min_df)	Aparecimento da palavra nos textos (token_pattern)	Treino	Teste	Tamanho tokens (n-grams -> 2)	Treino	Teste	Tamanho tokens (n-grams -> 1)
1	1	0,5033	0,4684	1277898	0,5265	0,4518	53673
1	2	0,4624	0,4307	1329250	0,5044	0,4266	53632
1	3	0,4587	0,427	1428136	0,5051	0,426	52967
1	4	0,4514	0,4196	1495784	0,5032	0,4175	50359
1	5	0,4338	0,3934	1169413	0,4879	0,4055	44277
2	1	0,5103	0,4706	407593	0,5279	0,452	30486
2	2	0,4725	0,4299	414140	0,5064	0,4274	30447
2	3	0,4688	0,4289	405353	0,5075	0,4266	29932
2	4	0,4654	0,4206	374830	0,5054	0,4185	28280
2	5	0,451	0,3992	247655	0,4916	0,4068	24280
3	1	0,5136	0,471	254128	0,5287	0,4526	24219
3	2	0,4757	0,4312	255492	0,5071	0,4265	24181
3	3	0,4739	0,4292	239330	0,5082	0,4256	23740
3	4	0,4718	0,4215	206230	0,5068	0,4197	22389
3	5	0,4593	0,4033	125870	0,4928	0,4056	19020
4	1	0,5146	0,471	189587	0,5293	0,4527	21014
4	2	0,4785	0,4311	189224	0,5079	0,4267	20976
4	3	0,477	0,4294	173170	0,5083	0,4258	20572
4	4	0,4757	0,4242	143008	0,5073	0,4188	19397
4	5	0,4649	0,4046	83037	0,494	0,4044	16378
5	1	0,5169	0,4714	152842	0,5303	0,4525	18752
5	2	0,4803	0,4313	151901	0,5081	0,4264	18715
5	3	0,4784	0,4306	136527	0,5091	0,4251	18349
5	4	0,478	0,4229	109213	0,508	0,4187	17283
5	5	0,4694	0,4058	61435	0,495	0,4045	14521
		Máximo:	Máximo:	Máximo:	Máximo:	Máximo:	Máximo:
		0,5169	0,4714	1495784	0,5303	0,4527	53673

- Em relação aos resultados dos diferentes stemmings, conclui-se que a diferença entre eles é muito pequena e por isso foi decidido utilizar o Snowball Stemming para toda a classificação. Não foram realizados sempre todos os stemmings por ser um processo muito demoroso e com pouco impacto.
- Em termos da comparação entre a utilização de uni-gramas e bi-gramas é possível verificar que a utilização de bi-gramas produz melhores resultados na classificação. Contudo foram realizados também sempre testes com uni-gramas.
- No que diz respeito ao argumento *min_df* e ao *token_pattern* foram realizados mais testes e o valor do *min_df* melhor era variável entre o valor 1, 3 e 5. Em relação ao argumento *token_pattern* em geral tamanho 1 ou mais é o que produz melhores resultados.
- Como estes resultados podem ser devido á disposição dos dados foram sempre feitos testes com diversos valores de *min_df* e *token_pattern*.

Teste parâmetros múltiplos

min_df	3	5										
token_pattern	r'\b\w+\b'	r'\b\w\w\w+\b'	r'\b\w\w\w\w\w+\b'									
ngrams	1,1	1,2										
penalty	l1	l2										
C	0.5	1	1.1	1.2	1.5	1.6	1.7	1.8	1.9	2	2.1	2.2
Multi-classe	multinomial	ovr										

Parâmetros utilizados

Classificação: multiclasse							
Min_df	token_pattern	ngrams	penalty	C	Multi-classe	Treino	Teste
3	r'\b\w+\b'	1,2	l1	2	ovr	0,5959	0,4822
3	r'\b\w+\b'	1,2	l1	2	ovr	0,5959	0,4822
3	r'\b\w+\b'	1,2	l1	1,9	ovr	0,5864	0,4829
3	r'\b\w+\b'	1,2	l1	1,9	ovr	0,5864	0,4829
5	r'\b\w+\b'	1,2	l1	1,9	ovr	0,5974	0,4782
3	r'\b\w+\b'	1,2	l1	2	ovr	0.5952	0,4835

- Os resultados nas primeiras quatro linhas foram todos com a mesma semente aleatória e foram realizados com diversos parâmetros, até obter o melhor resultado.
- Perante os primeiros quatro resultados conclui-se que: o melhor *min_df* foi o 3, o token_pattern foi r'\b\w+\b', utilizar bi-gramas, penalização l1, C de 1.9 e *multi_class* ovr.
- Os dois últimos testes foram realizados com sementes aleatórias para verificar a veracidade dos parâmetros para outras classificações, observa-se que num dos testes o min_df melhor foi o 5 os restantes parâmetros variam comparado aos parâmetros anteriores.
- Não foram realizados mais testes pois os testes eram muito demorados.

Teste para classificação binária - discriminante logístico

Tamanho minimo da palavra (min_df)	Aparecimento da palavra nos textos (token_pattern)	Treino	Teste	Tamanho tokens (n-grams -> 2)	Treino	Teste	Tamanho tokens (n-grams -> 1)
1	1	0,8956	0,8813	1277898	0,9029	0,8821	53673
1	2	0,8906	0,8785	1329250	0,9005	0,8827	53632
1	3	0,8872	0,8777	1428136	0,898	0,8817	52967
1	4	0,8799	0,8687	1495784	0,8952	0,8722	50359
1	5	0,8591	0,8458	1169413	0,8787	0,8543	44277
2	1	0,8989	0,8823	407593	0,9032	0,8819	30486
2	2	0,8948	0,8814	414140	0,9009	0,8825	30447
2	3	0,8913	0,8803	405353	0,899	0,8813	29932
2	4	0,885	0,8713	374830	0,8955	0,8727	28280
2	5	0,8672	0,8496	247655	0,8794	0,8551	24280
3	1	0,9003	0,8839	254128	0,9034	0,8822	24219
3	2	0,8958	0,8834	255492	0,9011	0,8826	24181
3	3	0,893	0,8806	239330	0,899	0,8811	23740
3	4	0,8871	0,8724	206230	0,8956	0,873	22389
3	5	0,8695	0,8512	125870	0,8796	0,8558	19020
4	1	0,9014	0,8846	189587	0,9033	0,8822	21014
4	2	0,897	0,885	189224	0,9014	0,8827	20976
4	3	0,8934	0,8812	173170	0,8992	0,8813	20572
4	4	0,8889	0,8741	143008	0,8958	0,8733	19397
4	5	0,8721	0,8522	83037	0,8795	0,856	16378
5	1	0,902	0,885	152842	0,9034	0,8817	18752
5	2	0,8979	0,8856	151901	0,9016	0,8827	18715
5	3	0,8945	0,8819	136527	0,8993	0,8814	18349
5	4	0,8895	0,8744	109213	0,896	0,8736	17283
5	5	0,8739	0,8539	61435	0,8801	0,8564	14521
		Máximo:	Máximo:	Máximo:	Máximo:	Máximo:	Máximo:
		0,902	0,8856	1495784	0,9034	0,8827	53673

Parâmetros:	Valor:
penalização	l1
max_iter	1000
multi_class	multinomial
stemming	snowball
C	1
classificação	binário

- Em termos gerais os melhores resultados são para *min_df* do valor 5, e o *token_pattern* para o valor 2 no teste. Contudo o valor de *token_pattern* com mais testes foi verificado que era melhor de dimensão 1.
- Em relação aos uni-gramas e os bi-gramas, os bi-gramas possuem melhores resultados.
- A dimensão que melhora os resultados são os textos com menor tamanho (*min_df* mais alto) mas o *token_pattern* é melhor para valores mais baixos.

Teste parâmetros múltiplos

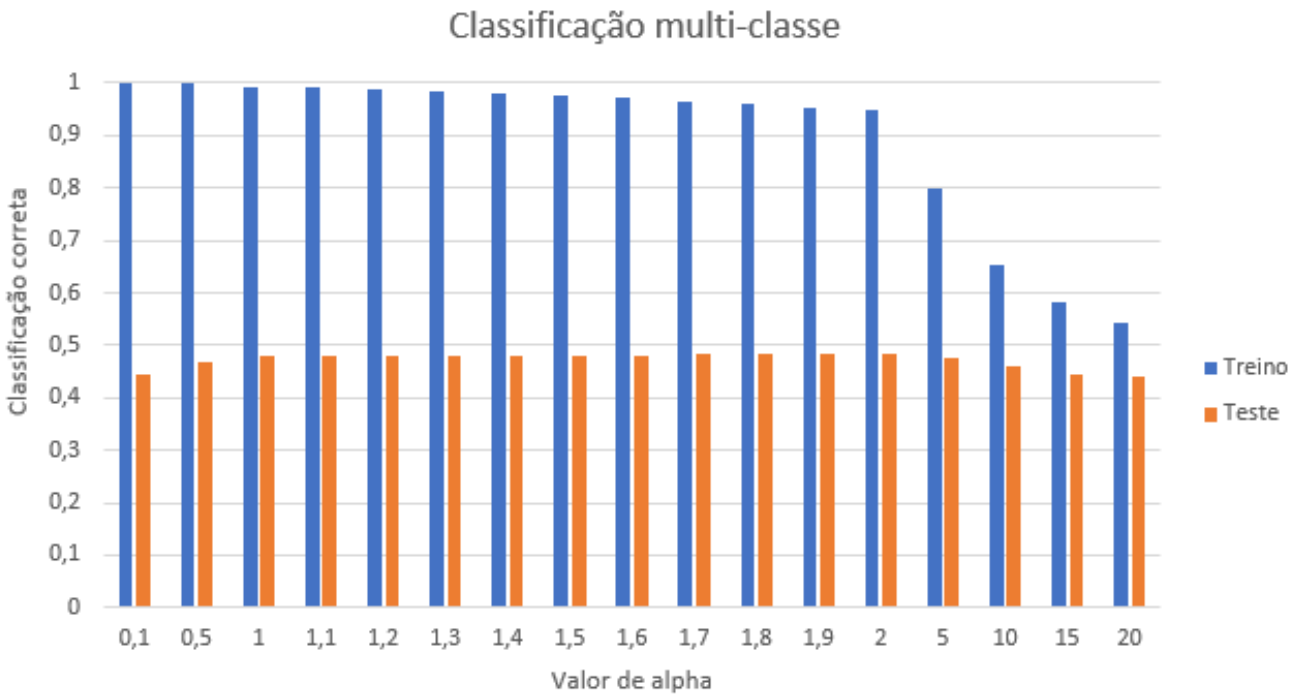
min_df	3	5						
token_pattern	r'\b\w+\b'	r'\b\w\w\w+\b'						
ngrams	1,1	1,2						
penalty	l1	l2						
C	0.5	1	1.5	1.7	1.8	1.9	2	2.1
Multi-classe	multinomial	ovr						

Parâmetros utilizados

Classificação: binário							
Min_df	token_pattern	ngrams	penalty	C	Multi-classe	Treino	Teste
3	r'\b\w+\b'	1,2	l2	2	multinomial	0,9885	0,9143
5	r'\b\w+\b'	1,2	l2	2	multinomial	0,9863	0,914
5	r'\b\w+\b'	1,2	l2	2,1	multinomial	0,9868	0,912
5	r'\b\w+\b'	1,2	l2	2,1	multinomial	0,9866	0,9156

- Os resultados foram todos feitos com sementes diferentes. Os parâmetros foram variando durante os testes para reduzir no tempo de classificação.
- Os melhores parâmetros são: *min_df* de 5, *token_pattern* é r'\b\w+\b', utilizar bi-gramas, penalização l2, valor de C de 2 ou 2.1 e *multi_class* é multinominal.
- Em comparação com a classificação multi-classe a classificação binária é mais rápida e por isso foram realizados mais testes, contudo também é bastante demorada.

Teste classificação multi-classe – classificador ridge



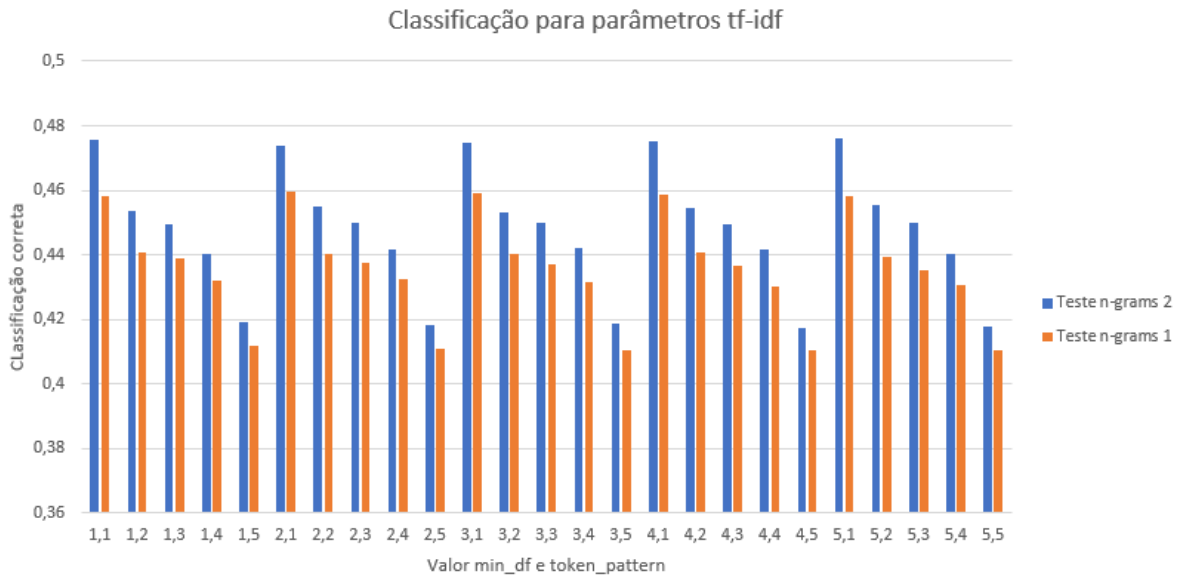
	Classificador: Ridge classificação: multiclasse		
Valor de Alpha	Treino	Teste	Coeficientes utilizados
0,1	1	0,444	2033024
0,5	0,9997	0,4692	2033024
1	0,9929	0,4797	2033024
1,1	0,9903	0,4807	2033024
1,2	0,9875	0,4804	2033024
1,3	0,9838	0,4799	2033024
1,4	0,9796	0,4806	2033024
1,5	0,9757	0,4808	2033024
1,6	0,9704	0,4812	2033024
1,7	0,9644	0,4825	2033024
1,8	0,9594	0,4832	2033024
1,9	0,9535	0,4823	2033024
2	0,9479	0,4826	2033024
5	0,7998	0,4748	2033024
10	0,6536	0,4619	2033024
15	0,583	0,4463	2033024
20	0,5412	0,4387	2033024
	Máximo:	Máximo:	Máximo:
	1	0,4832	2033024

Parâmetros:	Valor:
fit_intercept	True
normalize	False
stemming	snowball
n-grams	2

- Perante os resultados é possível verificar que para valores de alfa baixos o classificador decora muito os dados e possui uma má classificação. Contudo o melhor resultado de classificação foi para um valor de alfa onde o classificador possui uma grande diferença entre treino e teste e por isso ainda está a decorar muito os dados.
- Quanto maior o valor alfa menor é a diferença entre o treino e o teste, mas valores alfa muito altos acabam por também prejudicar o classificação dos dados de teste.

Testes de tf-idf

Tamanho minimo da palavra (min_df)	Aparecimento da palavra nos textos (token_pattern)	Treino	Teste	Tamanho tokens (n-grams -> 2)	Treino	Teste	Tamanho tokens (n-grams -> 1)
1	1	0,8292	0,4757	1277898	0,6551	0,4584	53673
1	2	0,8251	0,4534	1329250	0,6426	0,4406	53632
1	3	0,8291	0,4495	1428136	0,6446	0,4391	52967
1	4	0,8407	0,4405	1495784	0,6478	0,4322	50359
1	5	0,8325	0,419	1169413	0,6351	0,4119	44277
2	1	0,8097	0,4736	407593	0,6468	0,4594	30486
2	2	0,8018	0,4548	414140	0,6337	0,4403	30447
2	3	0,8012	0,45	405353	0,6362	0,4373	29932
2	4	0,8034	0,4415	374830	0,64	0,4325	28280
2	5	0,7768	0,4184	247655	0,6228	0,4111	24280
3	1	0,7998	0,4748	254128	0,6426	0,459	24219
3	2	0,7913	0,4533	255492	0,6288	0,4402	24181
3	3	0,7877	0,4499	239330	0,6317	0,4369	23740
3	4	0,7858	0,4422	206230	0,6337	0,4315	22389
3	5	0,7507	0,4186	125870	0,6161	0,4105	19020
4	1	0,7929	0,4754	189587	0,6389	0,4585	21014
4	2	0,7838	0,4543	189224	0,625	0,4407	20976
4	3	0,7793	0,4496	173170	0,6275	0,4364	20572
4	4	0,7736	0,4418	143008	0,6294	0,4303	19397
4	5	0,7309	0,4175	83037	0,6118	0,4106	16378
5	1	0,7862	0,4762	152842	0,6361	0,4581	18752
5	2	0,7773	0,4554	151901	0,6222	0,4395	18715
5	3	0,7727	0,4499	136527	0,6235	0,4354	18349
5	4	0,7635	0,4403	109213	0,6255	0,4308	17283
5	5	0,7158	0,4178	61435	0,6069	0,4102	14521
		Máximo:	Máximo:	Máximo:	Máximo:	Máximo:	Máximo:
		0,8407	0,4762	1495784	0,6551	0,4594	53673



Parâmetros:	Valor:
fit_intercept	True
normalize	False
stemming	snowball
alpha	5
classificação	multiclasse

- Em relação á classificação com a utilização de bi-gramas o melhor valor de *min_df* é o valor 5 enquanto uni-gramas é 2. Em ambos os casos o melhor valor de *token_pattern* é 1.
- A utilização de bi-gramas produz melhores resultados do que a classificação com uni-gramas.
- Em relação aos parâmetros *fit_intercept* e *normalize* disponibilizados pelo classificador estes produziam piores resultados do que deixar os parâmetros por omissão e por isso não foram utilizados.
- O classificador funciona melhor para uma quantidade de *tokens* mais reduzida.

Testes parâmetros múltiplos

min_df	3	5															
token_pattern	r'\b\w+\b'	r'\b\w\w\w+\b'	r'\b\w\w\w\w\w+\b'														
ngrams	1,1	1,2															
alpha	0,5	1	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	2	2,1	2,2	2,3	2,4	2,5

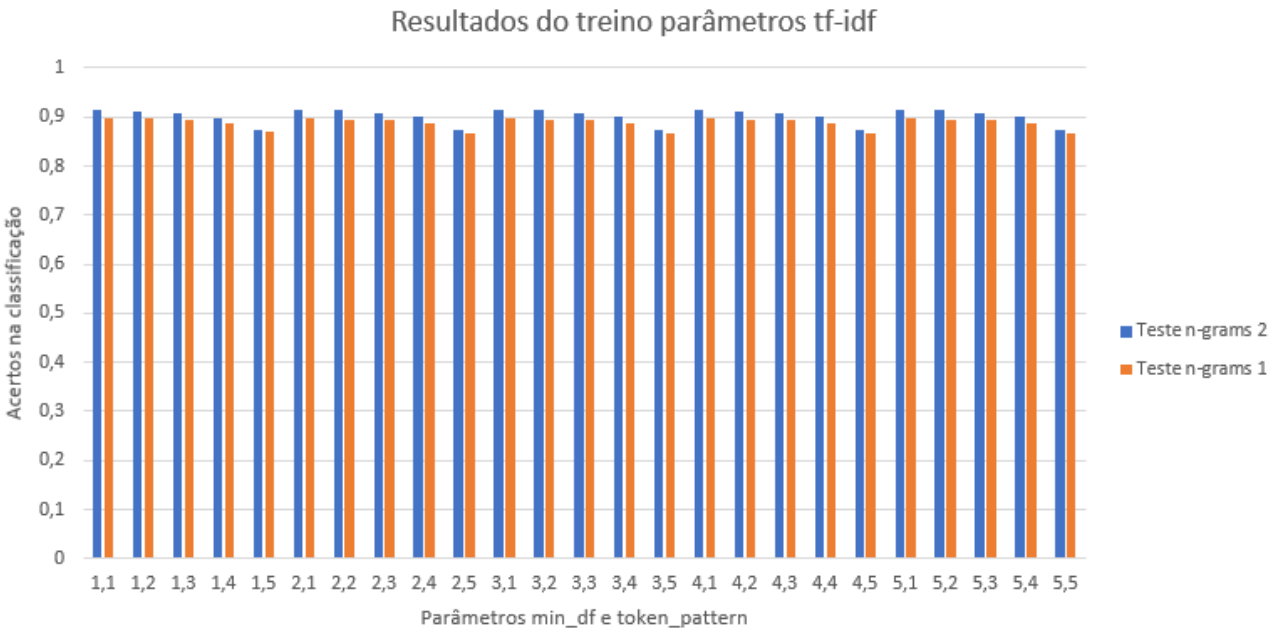
Parâmetros utilizados

Classificação: multiclasse					
Min_df	token_pattern	ngrams	alpha	Treino	Teste
3	r'\b\w+\b'	1,2	2,2	0,9377	0,4829
3	r'\b\w+\b'	1,2	1,5	0,975	0,4855
3	r'\b\w+\b'	1,2	1,9	0,9541	0,4874
2	r'\b\w+\b'	1,2	1,7	0,971	0,4815
1	r'\b\w+\b'	1,2	1,5	0,9873	0,4794

- Todos os testes foram realizados com sementes distintas.
- Em termos dos resultados estes variam muito no que diz respeito ao melhor valor de alfa e do *min_df*.
- Os melhores parâmetros são: r'\b\w+\b' para o *token_pattern* e utilizar bi-gramas.
- Em relação ao valor *min_df,* os primeiros 3 testes darem o valor 3 pode dever-se ao facto de não terem sido testados com os valores 1 e 2, por isso assume-se que o melhor valor é o 2. Em relação ao valor alfa este está disperso entre os valores 1,5 e 2,2. Por isso assume-se que o melhor valor é 1,5 pelo facto de os outros valores estarem mais próximos de 1,5 do que 2,2.
- No que diz respeito ao tempo de processamento o classificador ridge é mais rápido do que o discriminante logístico e produz resultados semelhantes para a classificação multi-classe.

Teste para classificação binária – classificador ridge

Tamanho minimo da palavra (min_df)	Aparecimento da palavra nos textos (token_pattern)	Treino	Teste	Tamanho tokens (n-grams -> 2)	Treino	Teste	Tamanho tokens (n-grams -> 1)
1	1	0,9902	0,9145	1277898	0,9542	0,8987	53673
1	2	0,99	0,9109	1329250	0,9533	0,8964	53632
1	3	0,9902	0,9085	1428136	0,9527	0,8951	52967
1	4	0,9903	0,8969	1495784	0,951	0,8886	50359
1	5	0,9871	0,8722	1169413	0,9406	0,8699	44277
2	1	0,9872	0,915	407593	0,9514	0,8973	30486
2	2	0,9872	0,913	414140	0,9509	0,8954	30447
2	3	0,9867	0,9084	405353	0,9499	0,8941	29932
2	4	0,985	0,9006	374830	0,9474	0,888	28280
2	5	0,9774	0,8732	247655	0,9355	0,8678	24280
3	1	0,9858	0,9148	254128	0,9502	0,8976	24219
3	2	0,9857	0,9131	255492	0,9497	0,8949	24181
3	3	0,9848	0,9079	239330	0,9482	0,8936	23740
3	4	0,9825	0,9006	206230	0,9457	0,8872	22389
3	5	0,9715	0,8739	125870	0,9333	0,8676	19020
4	1	0,9846	0,9158	189587	0,9489	0,8971	21014
4	2	0,9848	0,9122	189224	0,9483	0,8954	20976
4	3	0,9832	0,9091	173170	0,9471	0,8928	20572
4	4	0,9803	0,9001	143008	0,9438	0,8865	19397
4	5	0,9673	0,875	83037	0,931	0,8666	16378
5	1	0,984	0,9145	152842	0,948	0,8961	18752
5	2	0,9839	0,9128	151901	0,9471	0,8951	18715
5	3	0,9823	0,9085	136527	0,9461	0,8929	18349
5	4	0,9785	0,9002	109213	0,9426	0,8869	17283
5	5	0,9635	0,8737	61435	0,9292	0,8664	14521
		Máximo:	Máximo:	Máximo:	Máximo:	Máximo:	Máximo:
		0,9903	0,9158	1495784	0,9542	0,8987	53673



Parâmetros:	Valor:
fit_intercept	True
normalize	False
stemming	snowball
alpha	2
classificação	binário

- Assim como a classificação multi-classe, também a classificação binária é melhor com bi-gramas. Os bi-gramas funcionam melhor para valores altos de *min_df* e uni-gramas para valores baixos de *min_df*. Em ambos os casos o *token_pattern* é melhor o menor valor.
- A classificação com bi-gramas é melhor do que a classificação sem bi-gramas.

Testes parâmetros múltiplos

min_df	1	2	3	5													
token_pattern	r'\b\w+\b'	r'\b\w\w\w+\b'	r'\b\w\w\w\w\w+\b'														
ngrams	1,1	1,2															
alpha	0,5	1	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	2	2,1	2,2	2,3	2,4	2,5

Parâmetros utilizados

Classificação: binário					
Min_df	token_pattern	ngrams	alpha	Treino	Teste
3	r'\b\w+\b'	1,2	1	0,9965	0,9175
3	r'\b\w+\b'	1,2	1	0,9964	0,9162
2	r'\b\w+\b'	1,2	0,5	0,9999	0,9134
2	r'\b\w+\b'	1,2	0,5	1	0,9152
2	r'\b\w+\b'	1,2	1	0,997	0,9165
1	r'\b\w+\b'	1,2	0,5	1	0,9142
1	r'\b\w+\b'	1,2	0,5	1	0,9158

- Em termos gerais o melhor *token_pattern* é r'\b\w+\b' e a utilização de bi-gramas produz melhores resultados.
- Nos primeiros testes não foi incluído o valor 1 e 2 para o *min_df* e por isso é assumido que o min_df 2 é o melhor valor.
- Em relação ao valor de alfa como o número 0,5 é o que aparece mais vezes assume-se como sendo o melhor valor de alfa.

Utilização de PCA no processo classificação

min_df	3	5			
token_pattern	r'\b\w+\b'	r'\b\w\w+\b'			
ngrams	1,2				
penalty	l1				
C	0.5	1	1,5	2	
Multi-classe	multinomial	ovr			
n_components	5	10	20	50	100

Classificação: multiclasse com PCA								
Min_df	token_pattern	ngrams	penalty	C	Multi-classe	n_components	Treino	Teste
5	r'\b\w+\b'	1,2	l1	2	multinomial	100	0,4429	0,4339
5	r'\b\w+\b'	1,2	l1	2	multinomial	110	0,4448	0,4346

Treino	Teste
0,5959	0,4822
0,5959	0,4822
0,5864	0,4829
0,5864	0,4829
0,5974	0,4782
0.5952	0,4835

Classificação: binário com PCA								
Min_df	token_pattern	ngrams	penalty	C	Multi-classe	n_components	Treino	Teste
5	r'\b\w+\b'	1,2	l1	2	ovr	50	0,8554	0,8568

Treino	Teste
0,9885	0,9143
0,9863	0,914
0,9868	0,912
0,9866	0,9156

- Em relação á classificação com PCA foram testados primeiramente diversos valores de n_components e no segundo exemplo foi testado somente com 110.
- Conclui-se então que quanto maior o número de componentes melhor a classificação. Contudo quanto mais o número de componentes mais demorado é a execução do código. Foi testado com 200 componentes mas deu erro de memória e com 150 demorou muito tempo a execução do código e não foi incluído o teste.
- Em comparação com os resultados sem a utilização de PCA estes são bastantes inferiores aos resultados sem utilização de PCA.
- No que diz respeito ao número ótimo de componentes principais como a execução dos testes é muito demorada e os resultados de classificação têm vindo sempre a subir assume-se como sendo 110.

Conclusão

- No presente trabalho foi possível obter uma visão realista de como testar um classificador até obter os melhores parâmetros.

- O processo de obtenção dos melhores parâmetros de um classificador pode ser um processo muito demoroso e, por isso muitas vezes é utilizado os parâmetros para um resultado subótimo.