Aprendizagem Automática
Projeto Final

Classificação de críticas de cinema

Gonçalo Adolfo nº43802
Frederico Costa nº44094

Sumário

- Construção do vocabulário
- Classificadores abordados
- Estimação melhores parâmetros
- Scores por dataset/classificador
- Desempenho multiclasse/binário para um Discriminante Logístico
- Desempenho multiclasse/binário para um SVM Linear
- Desempenho multiclasse/binário para um classificador Distância ao Centróide
- Investigação dimensão do dicionário
- Métodos de agrupamento(clustering)
- Funções da aplicação e tomadas de decisão

Descrição Geral

- O projeto final de Aprendizagem Automática consiste no estudo de diferentes classificadores para classificação de críticas de cinema. Esta classificação pode ser relativa a um caso binário(negativa ou positiva) ou um problema multiclasse(pontuações 1-4 e 7-10)
- Engloba temáticas como a construção de diferentes vocabulários com diferentes tipos de limpeza e stemming; avaliação de classificadores com métricas de desempenho adequadas; tópicos de investigação como a dimensão do dicionário e clustering
- Por fim, conclui-se o dataset a ser utilizado assim como os classificadores para os casos binário e multiclasse nas funções da aplicação(*text2vector*, *binClassify*, *multiClassify*)

Construção dicionários

- Foram efetuados diferentes graus de limpeza e Stemming ao texto das críticas antes da sua representação tf-idf

- A limpeza do texto consistiu em retirar '
' que representa um break em html, retirar caracteres não alfa-numéricos, aplicar o stemming e por fim a representação tf-idf

- As variações em termos de graus de limpeza foram:
  - Token_pattern – extração de sequências de caracteres compostas por 2 ou mais letras ou números (\w) e que estão separadas por caracteres de pontuação ou espaços (\b).
  - Min_df = n – ignora termos que aparecem em menos de *n* documentos
  - Diferentes algoritmos de Stemming
  - Inclusão de n-gramas

Construção dicionários

Dataset	Tempo	Dimensão Final
1	260.995	18309
2	311.721	22268
3	221.103	21934
4	215.179	35411
5	221.729	27809
6	339.731	202342

- Dataset 1:
  - Min_df = 5
  - Pattern = r"\b\w\w+\b"
  - LancasterStemmer
  - Sem n-Gramas
- Dataset 2:
  - Min_df = 5
  - Pattern = r"\b\w\w+\b"
  - PorterStemmer
  - Sem n-Gramas
- Dataset 3:
  - Min_df = 5
  - Pattern = r"\b\w\w+\b"
  - SnowballStemmer
  - Sem n-Gramas
- Dataset 4:
  - Min_df = 2
  - Pattern = r"\b\w\w\w+\b"
  - SnowballStemmer
  - Sem n-Gramas
- Dataset 5:
  - Min_df = 3
  - Pattern = r"\b\w\w\w+\b"
  - SnowballStemmer
  - Sem n-Gramas
- Dataset 6:
  - Min_df = 5
  - Pattern = r"\b\w\w+\b"
  - LancasterStemmer
  - 2 n-Gramas

Classificadores abordados

- Neste projeto avaliou-se desempenho para o problema multiclasse/binário com 3 classificadores:
  - Discriminante Logístico
  - SVM Linear
  - Distância ao centróide

Escolha dos parâmetros

- Dada a dimensão dos dicionários não é possível efetuar uma pesquisa em grelha com grelhas com elevadas combinações. Para tal, estimou-se apenas hiper-parâmetros considerados de maior impacto como o C(termo de regularização) e no caso da distância ao centroide a métrica

- Ao correr o programa *híper_parâmetros.py* conclui-se que o melhor termo de regularização assim como a métrica se mantêm independentemente do dataset utilizado para treinar e classificar

- Efetua-se uma divisão treino-teste e estima-se os melhores parâmetros com uma validação cruzada no conjunto de treino de modo a dividir em treino-validação. O desempenho é mais fidedigno em outros módulos

Escolha dos parâmetros

- Grelha 'C': [0.01, 0.1, 1, 10, 100]
- Grelha 'metric': ['manhattan', 'euclidean', 'cityblock', 'cosine']
- Os restantes parâmetros foram sendo ajustados verificando o seu impacto

```
#### DATASET:  datasets/dataset1.p

Hiper parametros DL:
Score no conjunto de treino(caso multiclasse):  0.6375666666666666
Score no conjunto de teste(caso multiclasse):  0.4292
Score no conjunto de treino(caso binário):  0.9183666666666667
Score no conjunto de teste(caso binário):  0.8801
Melhores parâmetros: {'C': 1}
Hiper parametros Linear SVM:
Score no conjunto de treino(caso multiclasse):  0.6163333333333333
Score no conjunto de teste(caso multiclasse):  0.4305
Score no conjunto de treino(caso binário):  0.9235
Score no conjunto de teste(caso binário):  0.8803
Melhores parâmetros: {'C': 0.1}
Hiper parametros NC:
Score no conjunto de treino(caso multiclasse):  0.4071666666666667
Score no conjunto de teste(caso multiclasse):  0.3368
Score no conjunto de treino(caso binário):  0.819
Score no conjunto de teste(caso binário):  0.8118
Melhores parâmetros: {'metric': 'cosine'}
```

Scores por dataset/classificador

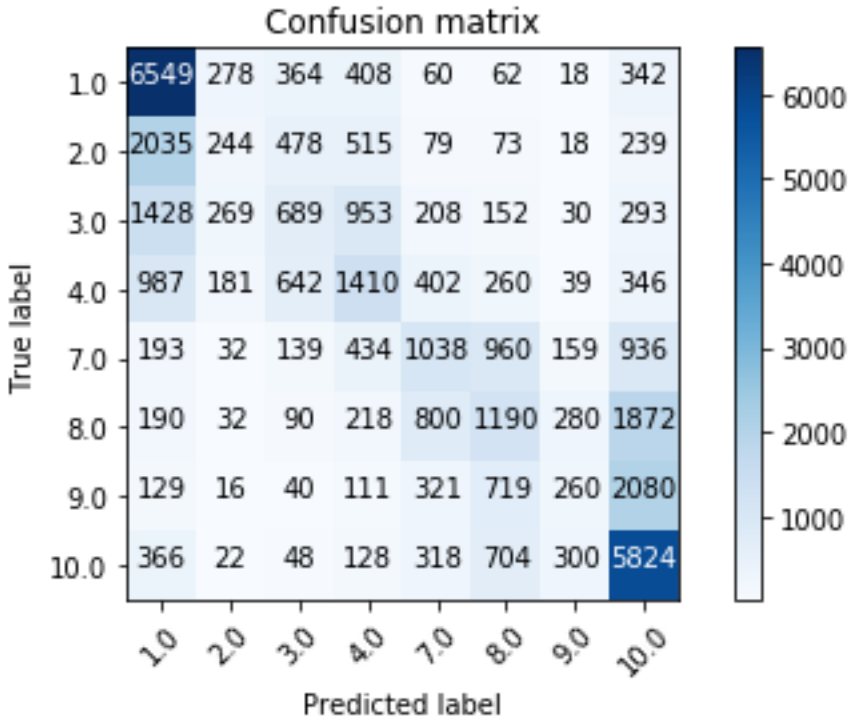
- Desenvolveu-se um método que permite avaliar com uma validação cruzada os scores para o problema multiclasse/binário para cada dataset para cada classificador abordado
- Permite ter uma ideia de qual o dataset a aprofundar de forma a avaliar os classificadores com mais medidas de desempenho assim de como o dataset a utilizar para treinar os classificadores na aplicação

MC	DL	SVM L	DC
Dataset1	0.432	0.430	0.334
Dataset2	0.433	0.433	0.333
Dataset3	0.432	0.434	0.335
Dataset4	0.434	0.434	0.336
Dataset5	0.434	0.431	0.338
Dataset6	0.442	0.440	0.352

Bin	DL	SVM L	DC
Dataset1	0.879	0.878	0.789
Dataset2	0.883	0.880	0.790
Dataset3	0.884	0.881	0.791
Dataset4	0.884	0.883	0.791
Dataset5	0.880	0.879	0.793
Dataset6	0.890	0.893	0.815

Desempenho problema multiclasse – Discriminante Logístico

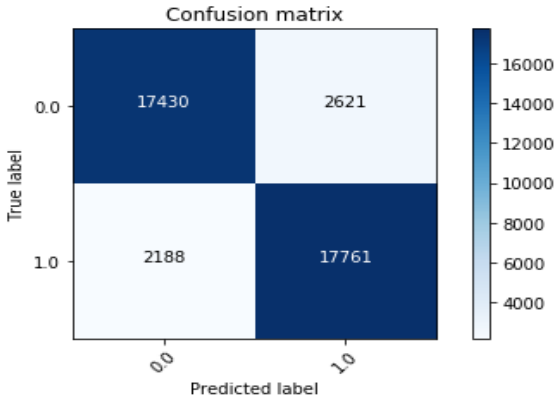
- Probabilidade total de acerto: 0.4301



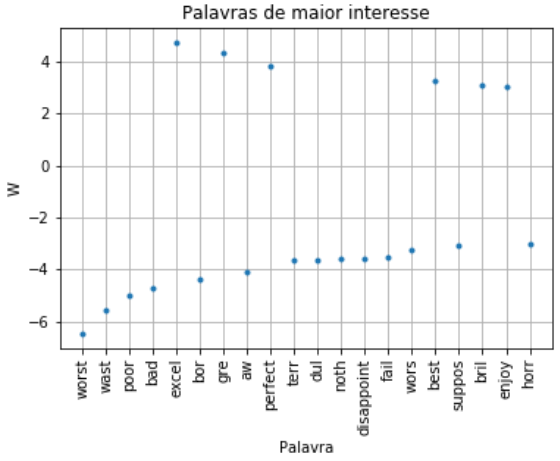
- Apesar da percentagem de acerto ser baixa, pode-se observar pela matriz de confusão que os erros obtidos situam-se maioritariamente nas pontuações ao lado da pontuação verdadeira. Note-se que a pontuação é uma questão subjetiva mesma na perspetiva dos humanos.

Desempenho problema binário – Discriminante Logístico

- Probabilidade total de acerto: 0.880



- Os erros de críticas estão relativamente bem distribuídos entre os dois tipos de críticas

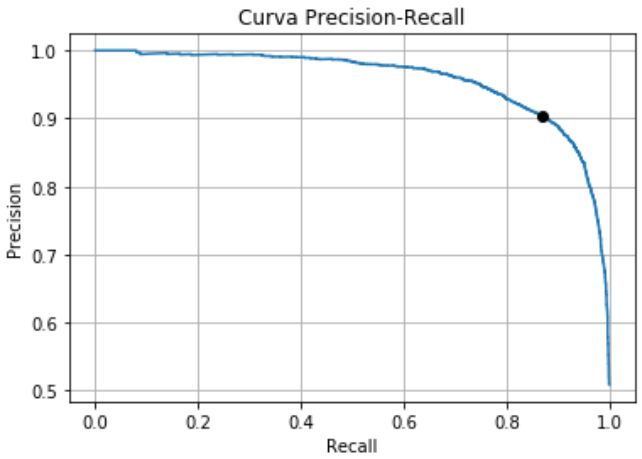


- Palavras de maior peso: excel, gre, perfect,…
- Palavras de menor peso: worst, wast, poor,…

Desempenho problema binário – Discriminante Logístico

- Probabilidade total de acerto:
- Apenas o valor da precision ou apenas o valor do recall não são suficientes para avaliar o classificador. A métrica f-score é a média harmónica entre o precision e o recall

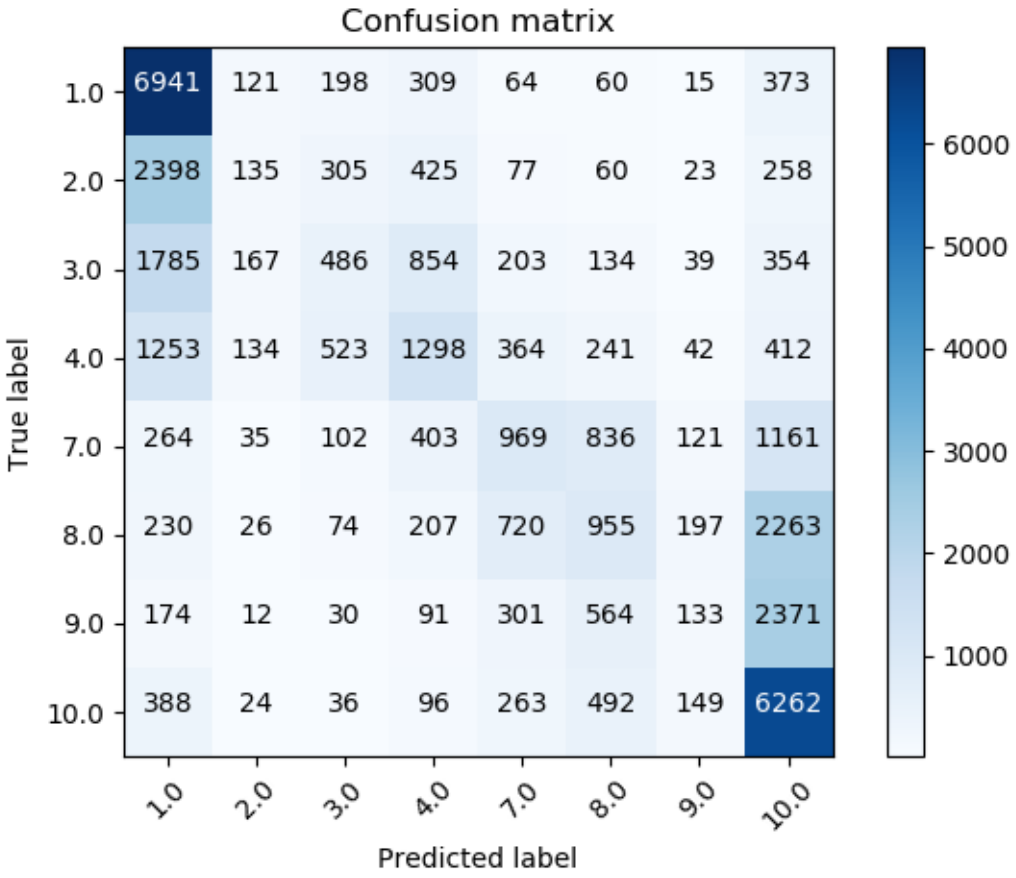
	precision	recall	f-score	support
C.negativa	0.89	0.87	0.88	20051
C.positiva	0.87	0.89	0.88	19949



- A curva precision-recall permite visualizar a precision e o recall para diferentes limiares de modo a afetuar a calibração ou comparar classificadores pela área debaixo da curva
- Área debaixo da curva: 0.958

Desempenho problema multiclasse – SVM Linear

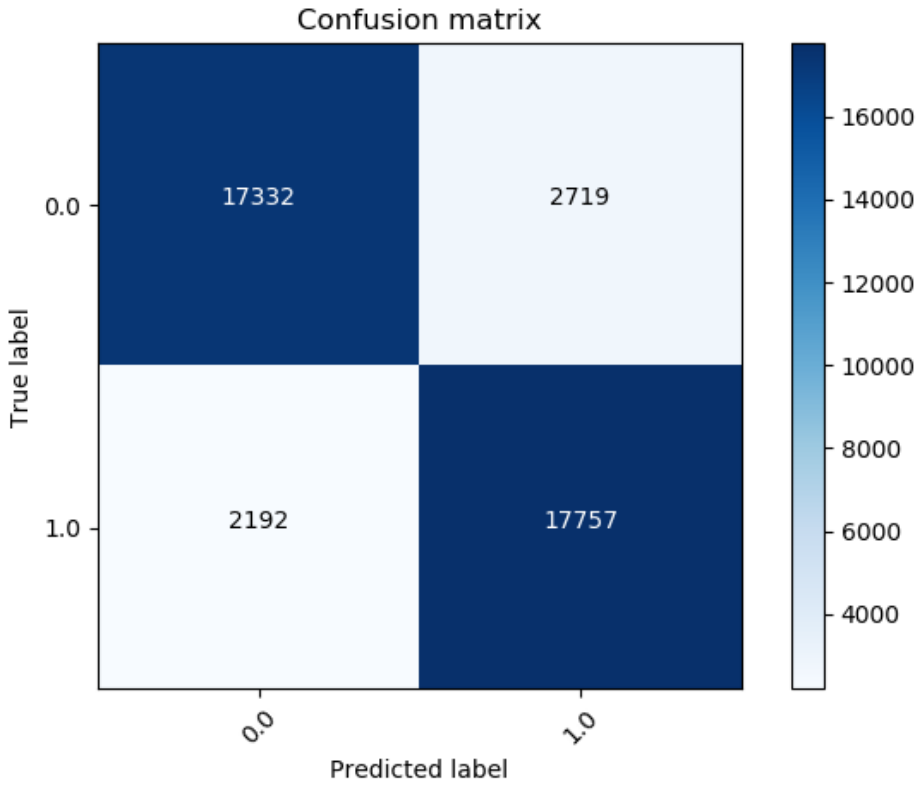
- Probabilidade de acerto : 0.429



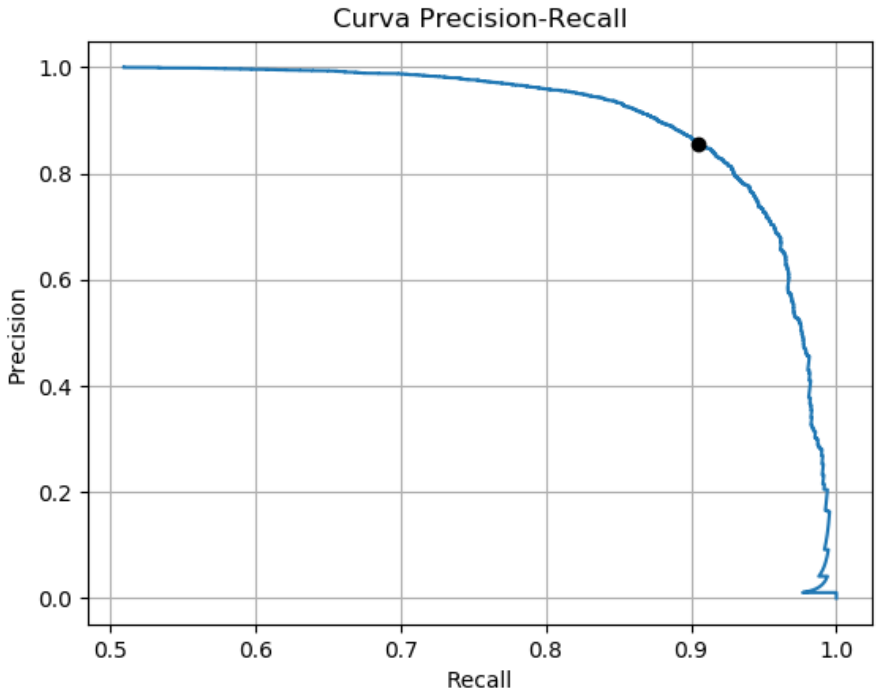
- Observou-se que a matriz de confusão é semelhante à originada pelo discriminante logístico
- Mais uma vez os erros que ocorrem estão perto da pontuação verdadeira
- A quantidade de acertos é exponencial nas pontuações mínimas e máximas (1.0 e 10.0)

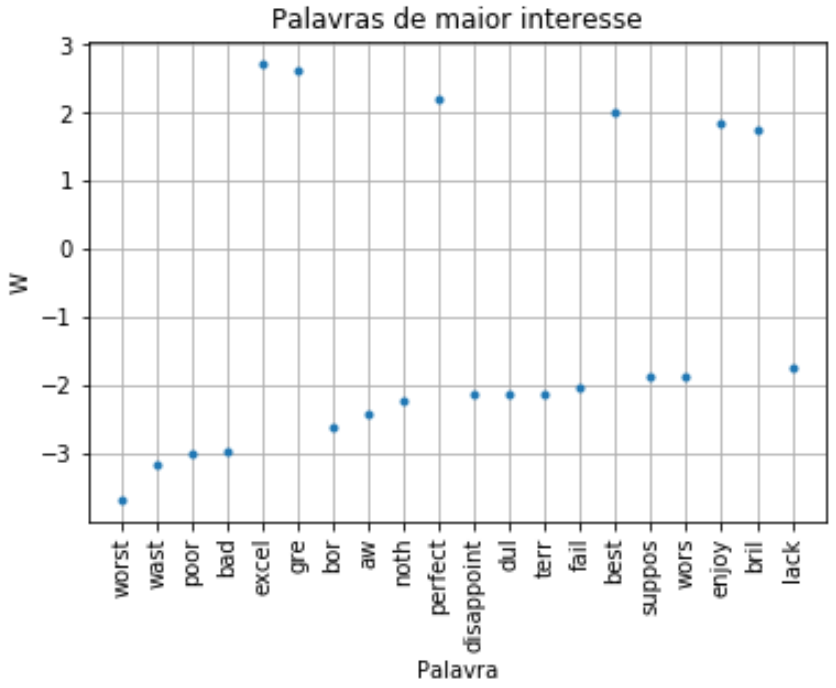
Desempenho problema binário – SVM Linear

- Probabilidade de acerto : 0.876



Desempenho problema binário – SVM Linear

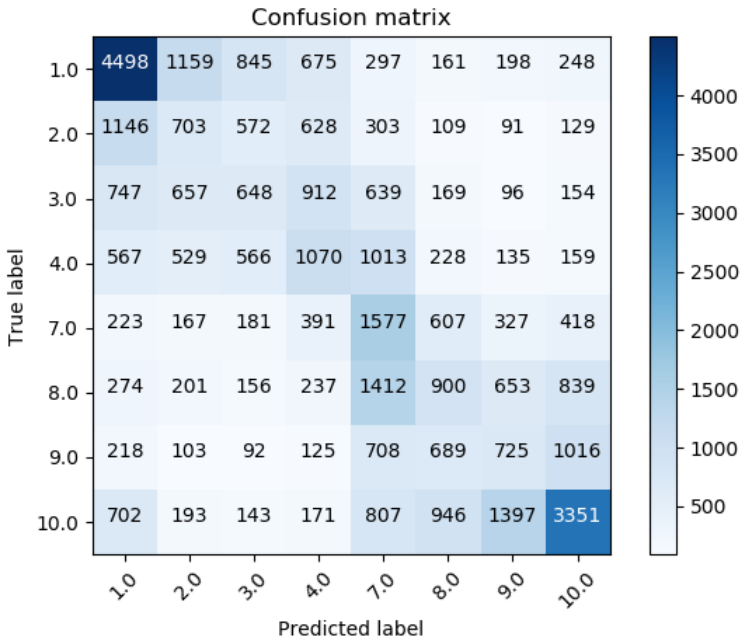




	precision	recall	f1-score	support
C.negativas	0.89	0.86	0.87	20051
C.positivas	0.87	0.89	0.88	19949

Desempenho problema multiclasse – Distância ao centróide

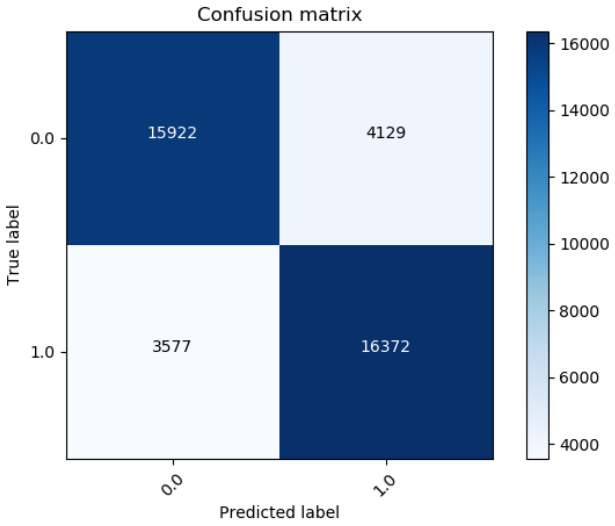
- Probabilidade total de acerto: 0.337



- Comparativamente ao DL, os erros situam-se melhores distribuídos nas pontuações em redor da pontuação verdadeira
- Não apresenta nenhuma função de não linearidade como o DL, apenas atribui a classe do centróide mais próximo

Desempenho problema binário– Distância ao centróide

- Probabilidade total de acerto: 0.807
- Erros bem distribuídos entre as classes
- Maior taxa de erros
- Valores menores de precision/recall e consequentemente f-score



	precision	recall	f-score	support
C.negativa	0.82	0.79	0.81	20051
C.positiva	0.80	0.82	0.81	19949

Dimensão do dicionário

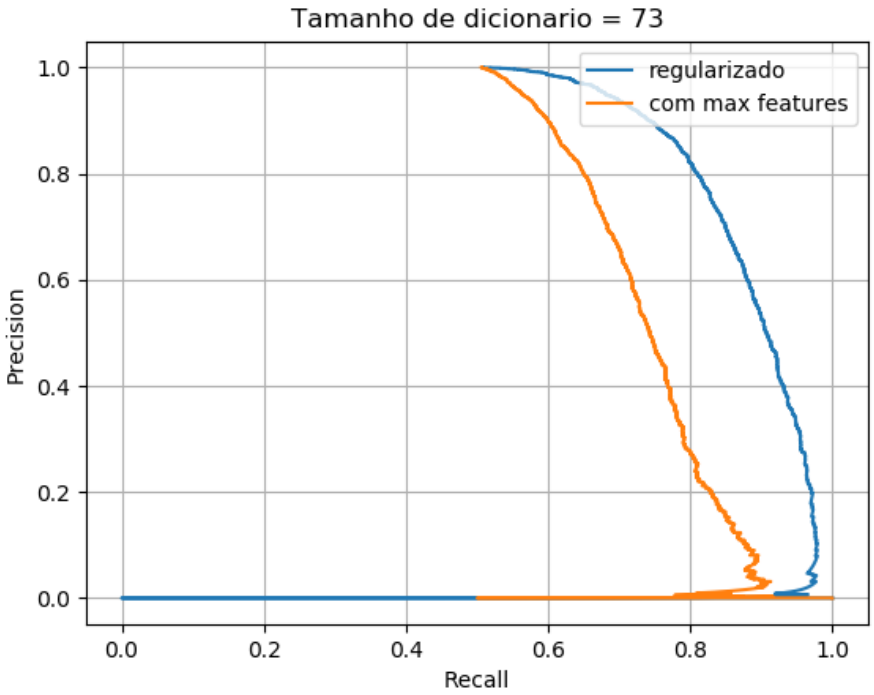
- O desempenho entre o classificador logístico com regularização Lasso e o classificador logístico com parâmetro **max_features** torna-se semelhante a partir de uma certa dimensão.

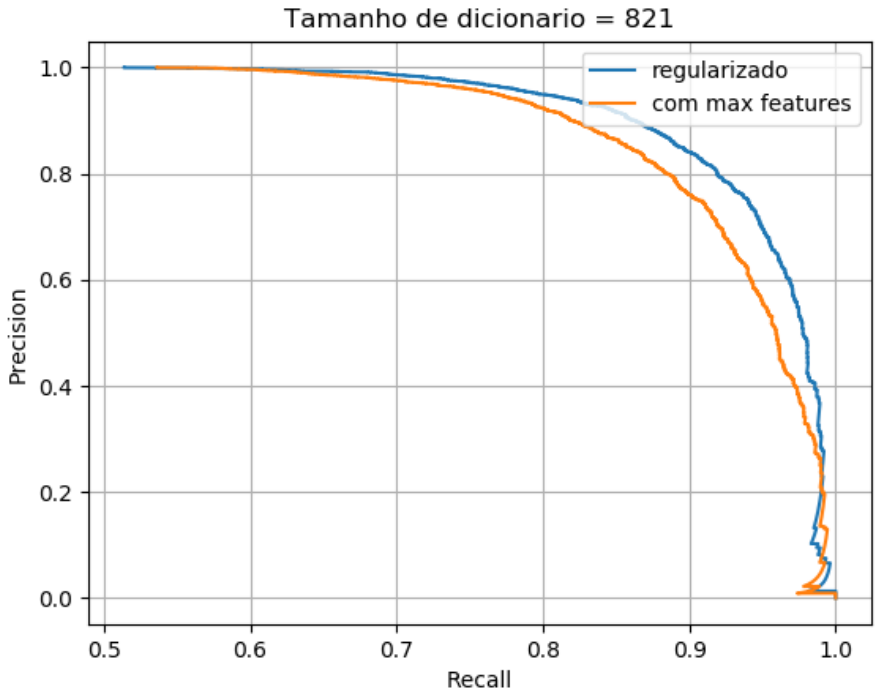
- O valor de **max_features** foi originado a partir do classificador logístico com regularização Lasso, onde se variou o parâmetro **C**, daí obtemos os pesos regularizados para 0 (coef_) e calculámos o tamanho final do vocabulário.

- Foi possível reduzir o tamanho do vocabulário para 6539, uma diferença exponencial em relação ao dataset com melhores desempenhos.

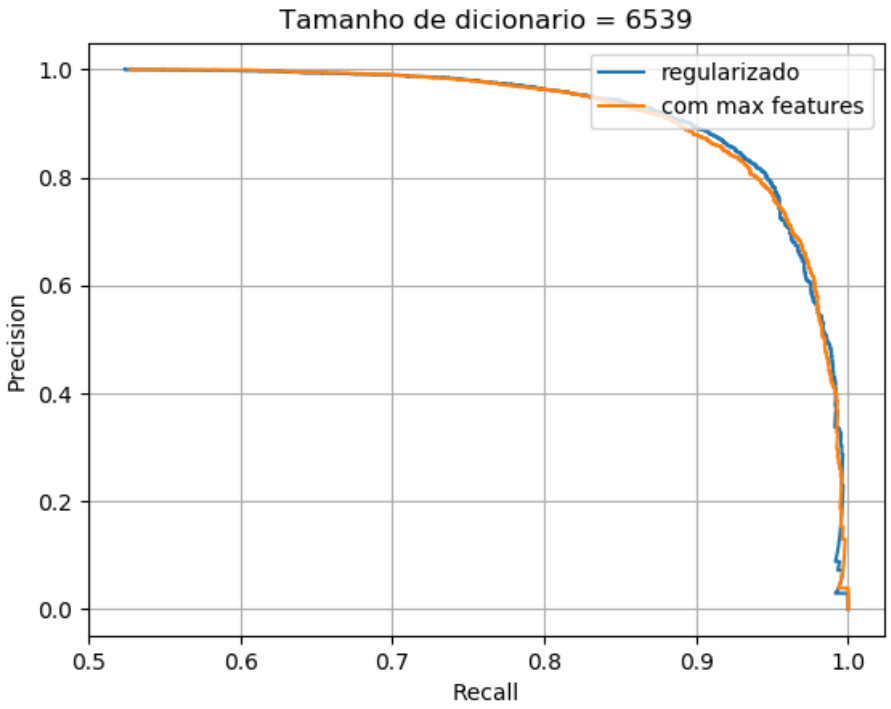
- Para representar esta semelhança de desempenho a partir de uma dimensão referida criámos gráficos da relação precision-recall, representados posteriormente.

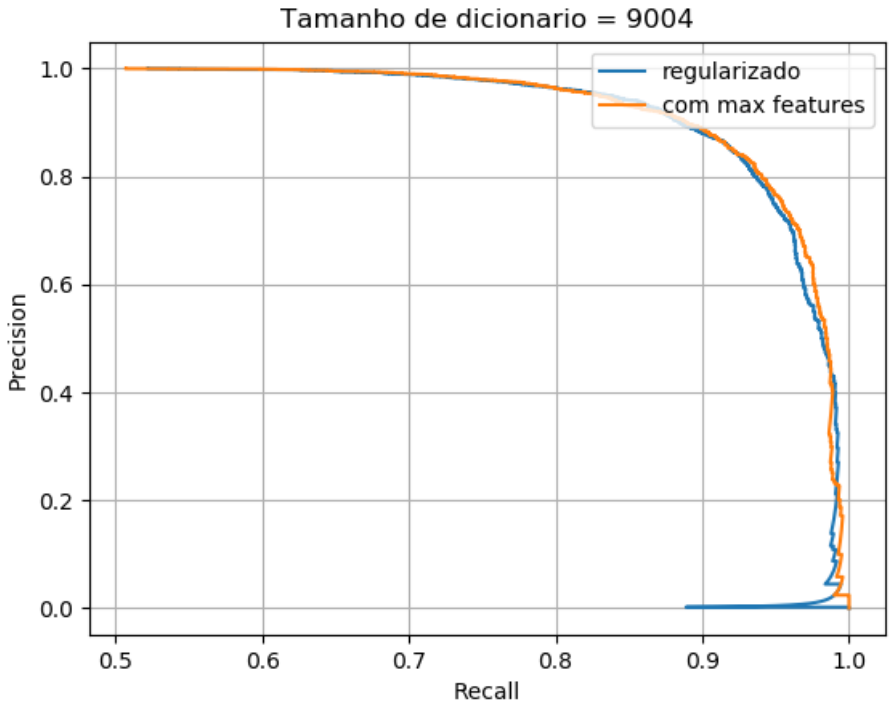
Dimensão do dicionário



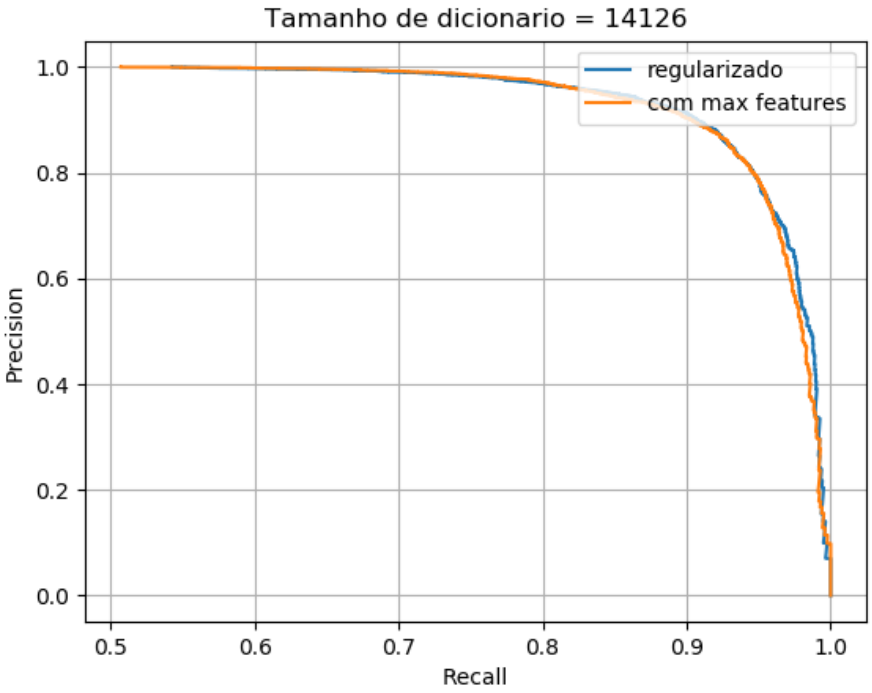


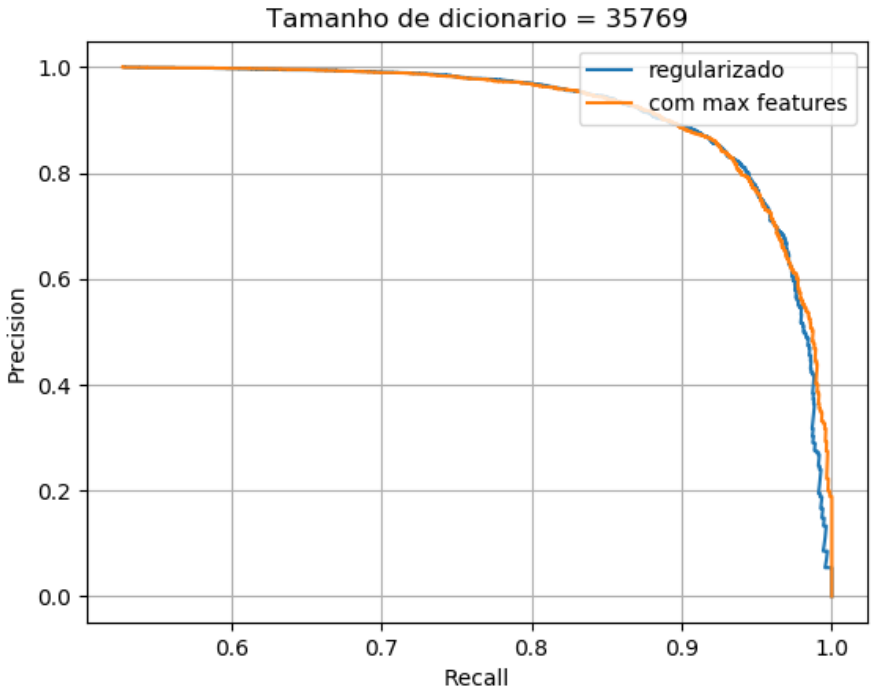
Dimensão do dicionário



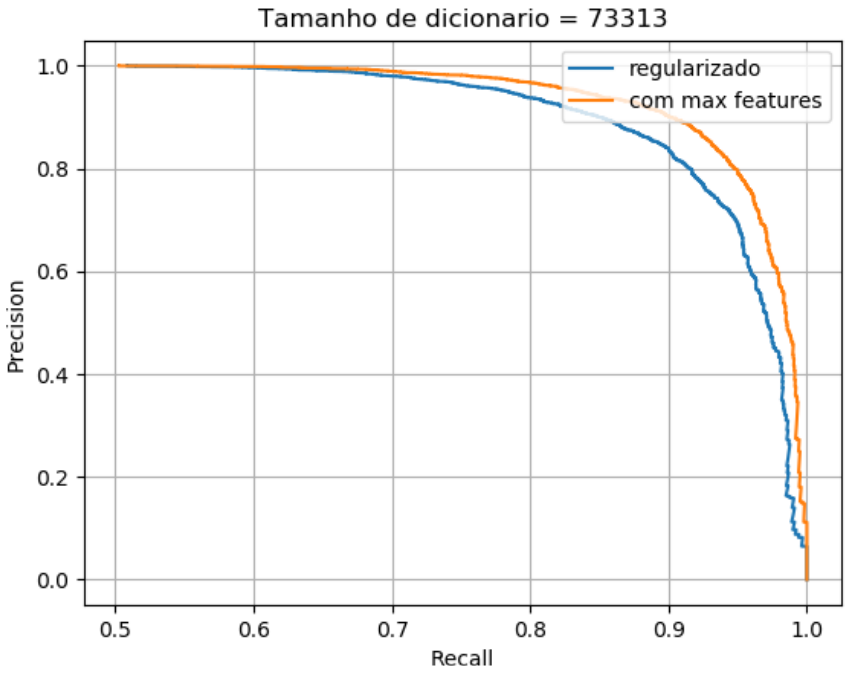


Dimensão do dicionário





Dimensão do dicionário



* É possível observar o momento em que o desempenho torna-se semelhante, concluímos que esta dimensão era ideal.

Desempenho Binário dos classificadores

C	Dimensão Dicionário	Regularizado	Com Max features
0.1	73	0.798	0.683
1	821	0.883	0.854
10	6539	0.894	0.894
100	9004	0.890	0.891
1000	14126	0.895	0.894
10000	35769	0.897	0.901
100000	73313	0.877	0.902

Dimensão do dicionário

- Após a redução do dicionário pelo parâmetro max_features testámos os 3 classificadores com os datasets novos originados

MC	DL	SVM L	DC
Dataset1	0. 429	0. 428925	0.339125
mf = 6539	0. 428725	0.4318	0.344125
mf = 9004	0.433725	0.4324	0.346675
mf = 14126	0.438925	0.437325	0.349725
mf = 35769	0.4395	0.43865	0.35255

mf – max features

Dimensão do dicionário

- Após a redução do dicionário pelo parâmetro max_features testámos os 3 classificadores com os datasets novos originados

Bin	DL	SVM L	DC
Dataset1	0. 8795	0.878325	0.806925
mf = 6539	0. 883075	0. 8795	0.809675
mf = 9004	0.885	0.881125	0.812925
mf = 14126	0.887425	0.88405	0.81515
mf = 35769	0.889525	0.8897	0.818875

mf – max features

Clustering

- Apresenta como objetivo o agrupamento de dados de modo a que dados pertencentes ao mesmo cluster possuam mais informação semelhante comparativamente a outros clusters

- Pode-se recorrer a métodos de agrupamento não supervisionado para identificar padrões no conjunto de críticas nomeadamente quanto aos tópicos abordados

- O valor do parâmetro k controla o número de clusters a estimar que no contexto das críticas pode ser interpretado como tópicos

- No contexto do projeto, utilizou-se o algoritmo k-médias e com a matriz de dados reduzida a 10.000 amostras de modo a reduzir o tempo de processamento

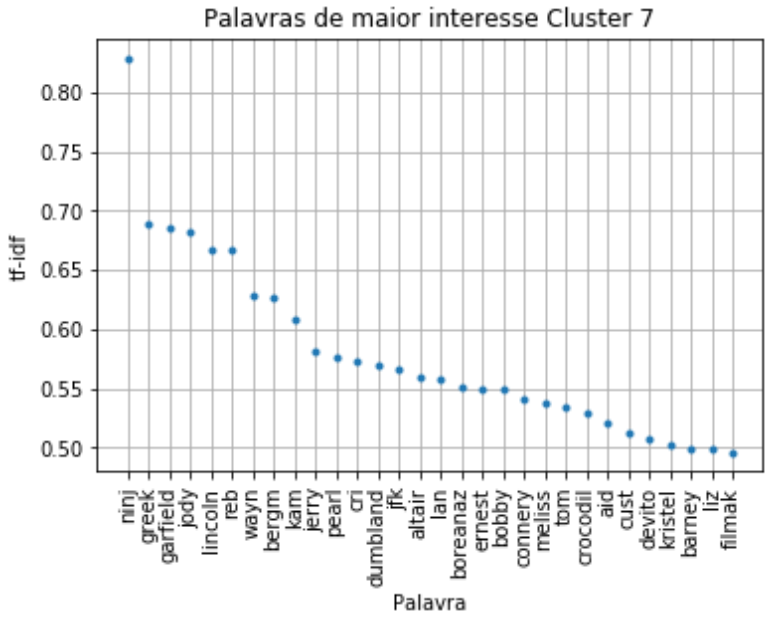
K-médias com k=5

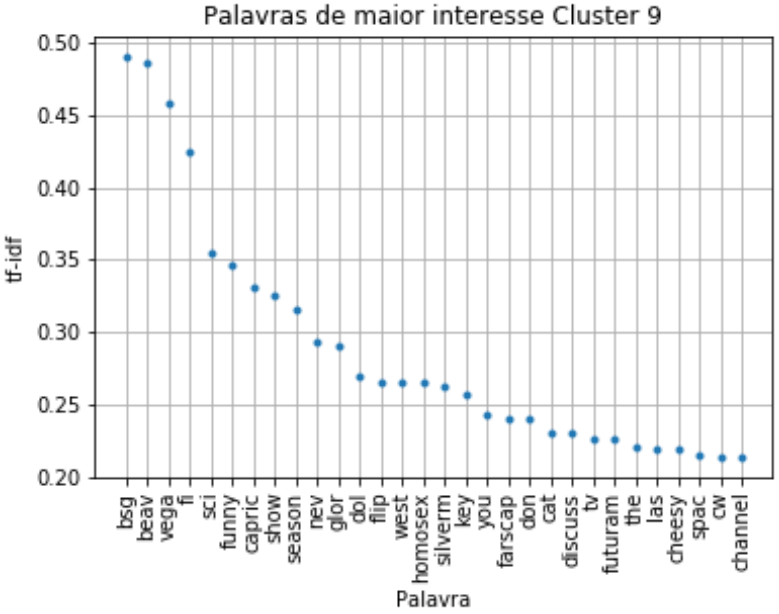
```
A processar cluster 0
['sasquatch' 'dahm' 'pau' 'asterix' 'cagney' 'grinch' 'crit' 'trem'
 'biograph' 'ufo' 'singap' 'dil' 'ha' 'puppet' 'caddyshack' 'aeon' 'miik'
 'pavarott' 'godard' 'mongol' 'colby' 'gang' 'lucy' 'beethov' 'rommel'
 'doct' 'fox' 'killjoy' 'episod' 'eagl' 'batm' 'dickson' 'detm' 'rugr'
 'jigsaw' 'helicopt' 'book' 'dant' 'shaq' 'cartoon' 'chupacabr' 'lincoln'
 'horrorfest' 'bourn' 'fido' 'jfk' 'shark' 'beowulf' 'schmid' 'robin'
 'hitch' 'boreanaz' 'kersey' 'heaton' 'lucil' 'gomez' 'soap' 'krist' 'pok'
 'host']
```

```
A processar cluster 1
['lv' 'nuk' 'darkm' 'botch' 'woodbury' 'lennon' 'ninj' 'jok' 'barney'
 'dahm' 'serb' 'winfield' 'killjoy' 'ranm' 'taffy' 'surf' 'edy' 'gandh'
 'puppet' 'muppet' 'modesty' 'sarno' 'rainbow' 'pokemon' 'snak' 'dolph'
 'batm' 'dent' 'gundam' 'shemp' 'gam' 'cappy' 'belush' 'nacho' 'magoo'
 'pie' 'vick' 'rosett' 'aquari' 'jeremy' 'what' 'amir' 'jigsaw' 'herc'
 'pau' 'twain' 'airwolf' 'rachael' 'gallo' 'carax' 'istanb' 'pack' 'plump'
 'traum' 'dj' 'whal' 'zomb' 'pasolin' 'shear' 'greek']
```

- Como se pode observar o k é bastante reduzido para serem verificados tópicos em cada cluster, notando-se que existe pouca relação entre as palavras em cada padrão

K-médias com k=12





- No cluster 7 podemos verificar grande quantidade de nomes: jordy, Lincoln, kam, jerry, pearl, cri, connery
- No cluster 9 observa-se palavras relativas a ficção científica: bsg, futuram, farscap, sci, fi, spac, capric, cheesy

K-médias com k=20

```
A processar cluster 12
['gam' 'boss' 'mario' 'rpg' 'chess' 'tr' 'soul' 'aot' 'multiplay' 'the'
 'bond' 'pandor' 'allegr' 'fps' 'sup' 'system' 'geek' 'card' 'mod' 'cup'
 'nintendo' 'foray' 'level' 'control' 'platform' 'easy' 'undy' 'ps'
 'tomorrow' 'dc' 'med' 'goldeney' 'atmosph' 'quak' 'virt' 'think' 'calib'
 'pc' 'you' 'unlock' 'tot' 'bang' 'firefight' 'bark' 'analog' 'hor'
 'monkey' 'hero' 'guard' 'existenz' 'pok' 'bows' 'play' 'gre' 'influ' 'vr'
 'greatest' 'sev' 'remot' 'diabol']
```

```
A processar cluster 0
['bond' 'connery' 'gam' 'goldeney' 'octopussy' 'brosn' 'jam' 'the' 'sean'
 'nev' 'nsna' 'brandau' 'best' 'rent' 'moor' 'tough' 'sydow' 'fatim'
 'icon' 'to' 'off' 'you' 'cool' 'is' 'unoff' 'est' 'von' 'largo' 'again'
 'ground' 'blofeld' 'and' 'but' 'should' 'good' 'real' 'system' 've'
 'miss' 'savala' 'fun' 'unnecess' 'of' 'rog' 'al' 'play' 'charact' 'say'
 'deficy' 'comp' 'seen' 'pretty' 'mad' 'spy' 'his' 'star' 'look' 'flem'
 'suav' 'him']
```

- Tópico observado cluster 12: videojogos
- Tópico observado cluster 0: filmes e atores relacionados com o 007

Conclusões – funções da aplicação

- text2vector – Que vocabulário utilizar para calcular os valores tf-idf das criticas recebidas?
- binClassify – Que classificador utilizar para o problema binário e com que dataset treinar o classificador?
- multiClassify - Que classificador utilizar para o problema multiclasse e com dataset treinar o classificador?

Conclusões – funções da aplicação

- O vocabulário escolhido teve origem na investigação da dimensão do dicionário – 'dataset_max_6539.p'. Concluímos que foi o vocabulário com menor dimensão que não prejudicasse consideravelmente o desempenho dos classificadores

- Com base no desempenho observado:

  - Os classificadores discriminante logístico e SVM linear apresentam ambos resultados bastante semelhantes quanto ao seu desempenho em ambos os problemas. A distância ao centroide apresenta resultados mais fracos quanto ao seu desempenho.
  - Como o tempo de processamento é superior no SVM Linear, escolheu-se o classificador discriminante logístico para ambos os problemas