Pedro Morais nº 41019 Francisco do Carmo nº 41027 Pedro Bento nº 41082
16 de janeiro de 2017

APRENDIZAGEM AUTOMÁTICA
CLASSIFICAÇÃO DE CRÍTICAS DE CINEMA NO IMDB

Introdução

Neste trabalho iremos analisar dados de texto, em particular críticas de cinema do IMDb. O trabalho está dividido em duas tarefas principais: 1) Classificação – neste ponto o objetivo é determinar se uma crítica é positiva ou negativa, baseado no texto da crítica. 2) Regressão – aqui o objetivo é prever a pontuação de uma crítica numa escala de 1 a 10 baseado no texto da mesma. Adaptar o modelo de regressão num classificador de críticas positivas e negativas e comparar os resultados obtidos com os resultados do ponto 1).

Desenvolvimento

Primeiramente é realizada uma limpeza às críticas de modo a filtrar palavras que possam ser menos úteis na classificação dos textos. A limpeza realizada é a mesma, quer para os documentos de treino, quer para os documentos de teste. No entanto, o modelo usado nos documentos de teste é o modelo formado pelos documentos de treino. A nossa limpeza começa por transformar todos os símbolos de mudança de linha HTML num espaço e através de uma expressão regular os textos ficam apenas com caracteres alfabéticos. A estes caracteres irá ser aplicado uma técnica de *stemming*, mais concretamente um *Porter Stemmer*, que transforma as palavras na sua raiz, o que permite mapear palavras semelhantes numa única palavra. O último passo de limpeza é feito aplicando uma técnica tf-idf. Esta técnica só irá extrair palavras compostas por quatro ou mais letras, que apareçam em cinco ou mais documentos e que não sejam "*stop words"* na língua inglesa. Para testar várias dimensões de vocabulário ao tf-idf pode também ser definido o valor máximo do tamanho do vocabulário que se pretende utilizar. Se o limite máximo da dimensão não for definido, no fim da limpeza iremos ter uma dimensão de 16989.

O primeiro classificador que iremos utilizar é um discriminante logístico. Este classificador irá criar um modelo usando os dados de treino e irá ser avaliado com a classificação dos dados de teste. Para verificar estas avaliações irão ser testadas várias dimensões de vocabulário para verificar qual a que apresenta melhores resultados.

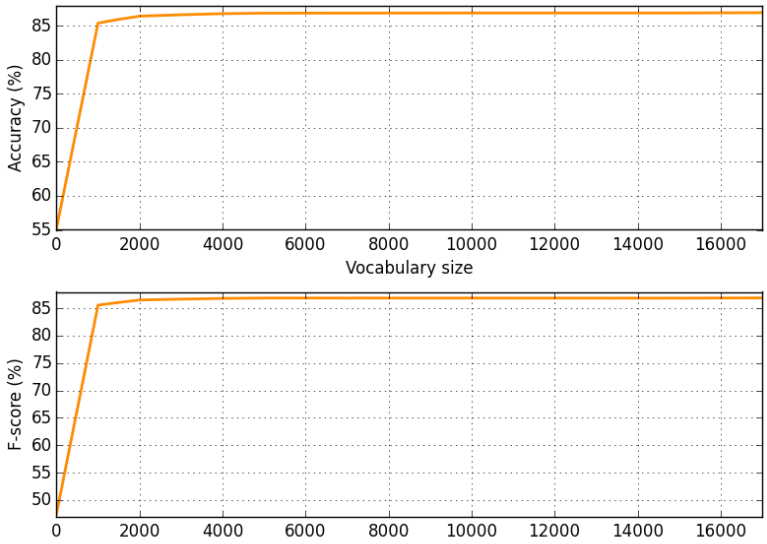

Figura 1 - Métricas de avaliação discriminante logístico.

Como é possível verificar no gráfico à esquerda, ambas as métricas de desempenho utilizadas para avaliar este classificador têm curvas muito parecidas. Os valores da curva são tendencialmente constantes até aos 3000, pelo que os próximos classificadores irão utilizar sempre um vocabulário com dimensão igual a 3000. Este facto também é verificável se visualizarmos as matrizes de confusão. Comparando as matrizes do vocabulário com dimensão de 3000 e dimensão de 16989 vemos que, apesar de com 3000 haver mais erros de

classificação, uma vez que esses erros em relação ao número total de documentos são tão pequenos, no final os resultados obtidos acabam por ser praticamente os mesmos e temos a vantagem de pouparmos tempo computacional.

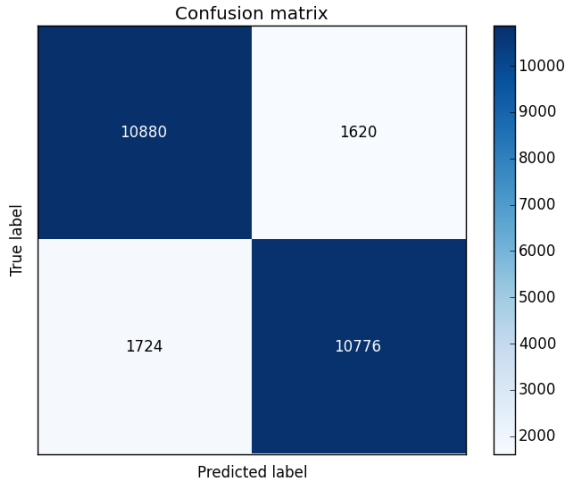

Figura 2 - Matriz de confusão para dimensão do vocabulário de 16989.

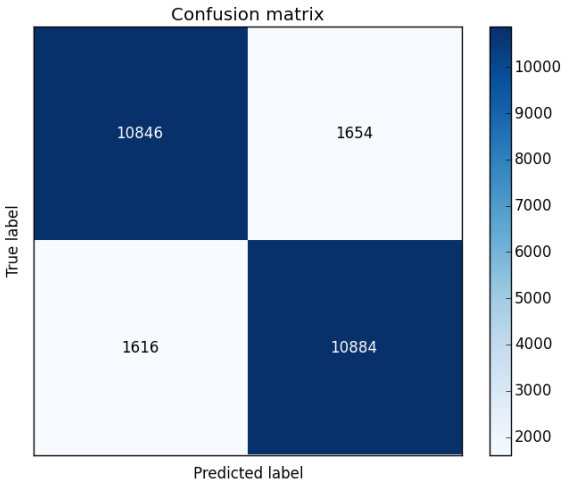

Figura 3 - Matriz de confusão para dimensão do vocabulário de 3000.

O outro classificador utilizado é um classificador k-vizinhos mais próximos com uma métrica de distância Euclidiana. O classificador irá ser avaliado com diferentes números de vizinhos de modo a ser escolhido o número ideal. Como os classificadores deste tipo calculam a distância entre todos os pontos não é possível criar um modelo com os 25.000 documentos de treino (são demasiadas contas para a RAM do computador), para resolver este problema os números de documentos de treino usados no modelo têm de ser reduzidos, sendo que esta redução pode fazer com que o classificador no final apresente resultados piores do que se o modelo tivesse sido construído com todos os documentos. O número final de documentos possíveis é 650 e o valor ideal de vizinhos encontrado foi o de 195 vizinhos como é possível verificar no gráfico.

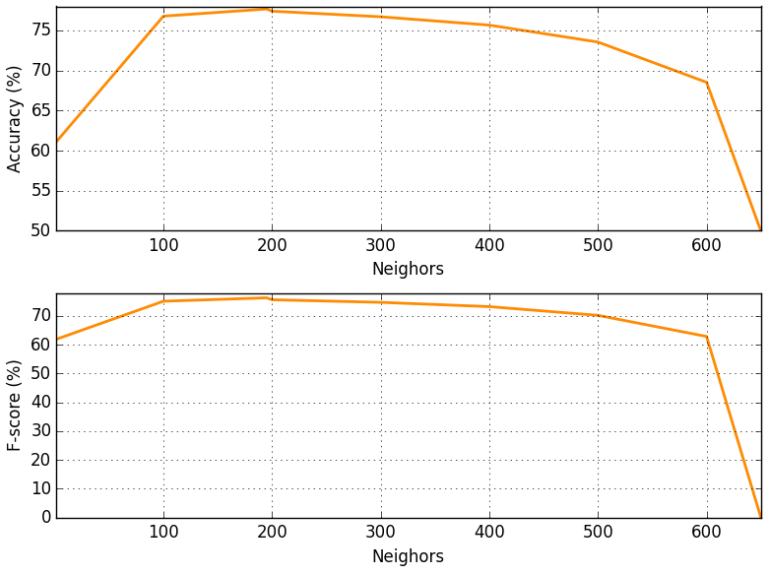

Figura 4 - Métricas de avaliação classificador k-vizinhos mais próximos.

Os dados apenas são obtidos com a representação tf-idf, porque quando estes foram pré-processados de modo a terem média nula o computador não tem poder para colocar a média nula nos 25.000 documentos de treino, o que impossibilita a realização deste ponto. Se compararmos a matriz de confusão para 195 vizinhos podemos ver que os resultados obtidos com o kNN são piores comparativamente com os obtidos com o discriminante logístico.

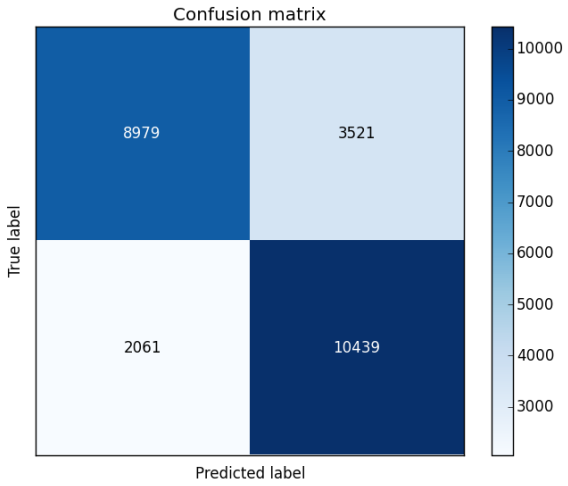

Figura 5 - Matriz de confusão para 195 vizinhos.

Para compararmos os dois classificadores e verificar qual tem melhor desempenho, para além de compararmos os resultados acima apresentados, podemos também comparar as curvas ROC e as áreas das cuvas

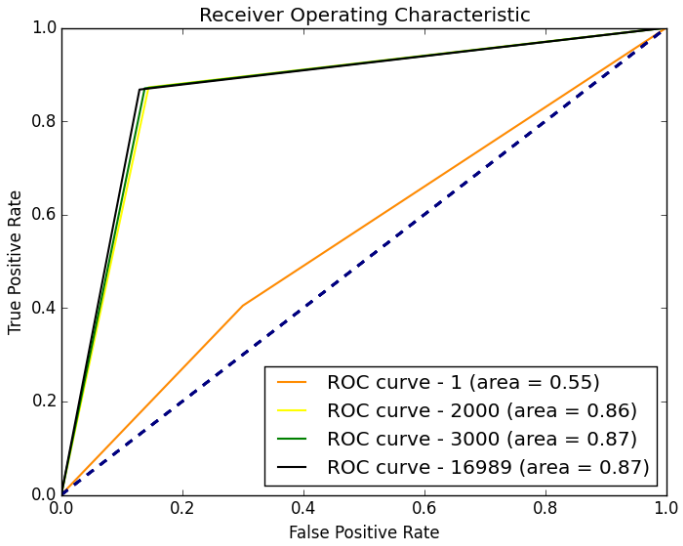

Figura 6 - Curva ROC para o discriminante logístico.

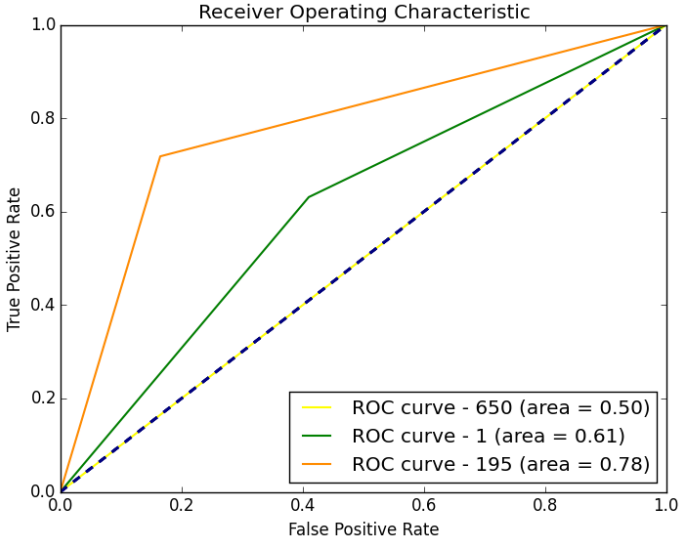

Figura 7 - Curva ROC para o classificador k-vizinhos mais próximos.

No gráfico da esquerda, referente ao discriminante logístico, podemos verificar que o pior resultado que este classificador pode obter é quando o vocabulário tem dimensão 1. Com dimensão 2000 já se nota alguma diferença comparativamente com vocabulários com dimensão 3000 e 16989. No gráfico da direita, referente ao classificador kNN, é possível verificar que se o número de vizinhos for igual ao número dos documentos usados na criação do modelo, o classificador torna-se num classificador aleatório. O classificador kNN nunca apresenta melhor desempenho que o discriminante logístico – este facto pode ser devido ao número reduzido de documentos utilizados na formação do modelo no classificador kNN –, mas em contrapartida o classificador kNN é muito mais rápido a classificar as críticas.

Entrando na segunda parte do trabalho, projetamos um modelo de regressão linear para prever a pontuação das críticas baseado nos seus textos. Numa primeira fase retiramos a informação sobre o valor da pontuação, que se encontra no nome de cada documento. No enunciado é-nos sugerido que possivelmente devido à elevada dimensão do vocabulário possa ser necessário pré-processar os dados com PCA, de modo a ficar com apenas as componentes principais. No entanto, este passo foi-nos impossível de realizar devido aos erros de memória que ocorrem na transformação das matrizes esparsas num *array*. Deste modo, utilizamos então a regressão linear presente na biblioteca *scikit-learn*, que faz com que não seja necessário pré-processar os dados com PCA, criando o modelo com os dados de treino e avaliando o regressor com os dados de teste.

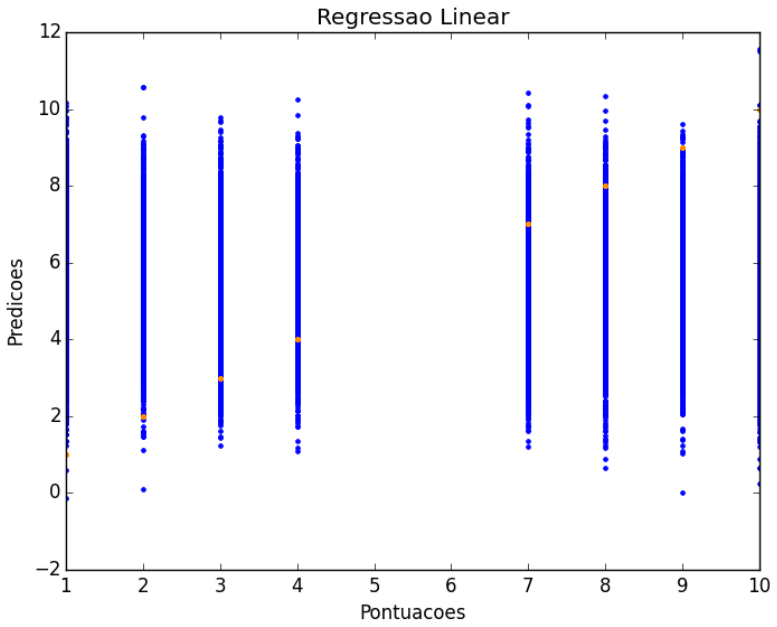

Figura 8 - Regressor Linear.

Como podemos observar os resultados são bastante maus. Os pontos laranjas indicam-nos o valor ideal para as determinadas pontuações e a azul encontram-se os coeficientes que obtivemos com o regressor linear. Existe uma grande discrepância entre a previsão e os resultados obtidos provando que os resultados não são mesmo os melhores.

Outra forma de avaliar este regressor é calcular a potência do erro. No nosso regressor a potência do erro é de 0,148 e esta métrica de avaliação fornece-nos uma 'medida' de como os resultados bem classificados são replicados no modelo, com base na proporção da variação total dos resultados apresentados pelo modelo. O facto da potência do erro nos dar um valor negativo e não um valor entre 0 e 1 como é expectável, deve-se ao facto da medição ser feita entre os dados calculados pelo regressor linear e os dados originais.

Para podermos comparar o regressor com os classificadores projetados anteriormente, convertemos o nosso regressor num classificador de críticas positivas e negativas.

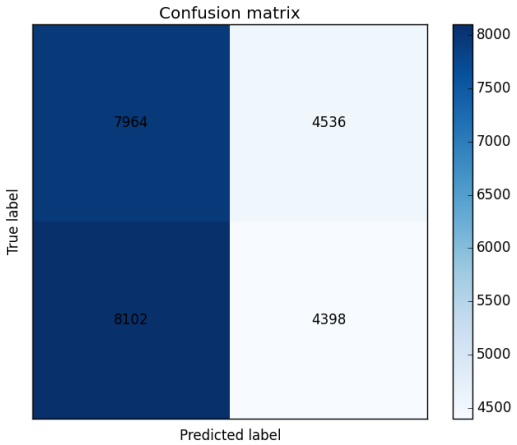

Figura 9 - Matriz de confusão para o regressor transformado num classificador.

A partir da matriz de confusão podemos observar de forma mais detalhada o resultado da nossa classificação. Num total de 25000 criticas, 8934 foram classificadas como negativas e 16066 classificadas como positivas. Das 8934 classificadas como negativas somente 4398 foram bem classificas e das 16066 positivas somente 7964. A percentagem de acertos é bastante baixa, fazendo apanágio aos gráficos anteriormente ilustrados, tendo um valor de 49,47%, o que é pior do que lançar uma moeda ao ar e ver qual a face que calha para cima.

As criticas positivas são as que obtém as melhores classificações, tendo uma percentagem de acertos de 63,712%, enquanto que as criticas negativas têm somente 35,184% de acertos.

Conclusão

Concluindo, trabalhar com dados de grandes dimensões como estes pode levar a vários erros devido à memória disponível e ao tempo útil de espera para o qual deixa de ser proveitoso esperar para obter a classificação.

De todos os classificadores projetados o que obteve melhores desempenhos, consoante as métricas de avaliação utilizadas, foi o discriminante logístico, mas melhores desempenhos fazem com que o discriminante seja mais lento. O classificador mais rápido é o classificador dos k-vizinhos mais próximos.

Bibliografia

http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html (acedido a 10 de janeiro de 2017)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics (acedido a 13 de Janeiro de 2017)

http://scikit-learn.org/stable/modules/feature_extraction.html (acedido a 5 de Janeiro de 2017)

http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (acedido a 5 de Janeiro de 2017)

http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html (acedido a 5 de Janeiro de 2017)

http://scikit-learn.org/stable/modules/linear_model.html (acedido a 12 de janeiro de 2017)

http://scikit-learn.org/stable/modules/model_evaluation.html (acedido a 14 de janeiro de 2017)

http://www.bogotobogo.com/Algorithms/Machine_Learning_NLP_Sentiment_Analysis_1.php (acedido a 5 de Janeiro de 2017)

https://datamize.wordpress.com/2015/01/24/how-to-plot-a-roc-curve-in-scikit-learn/ (acedido a 13 de Janeiro de 2017)

https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html (acedido a 5 de Janeiro de 2017)

https://en.wikipedia.org/wiki/Coefficient_of_determination (acedido a 15 de janeiro de 2017)

https://en.wikipedia.org/wiki/Information_retrieval#Precision (acedido a 13 de Janeiro de 2017)