FUNDAÇÃO GETULIO VARGAS

MESTRADO EM MATEMÁTICA APLICADA

ALYNE PEREIRA DE OLIVEIRA

TREINAMENTO DE UM CLASSIFICADOR PARA RECONHECIMENTO DE CLASSES DE NOTÍCIAS

RIO DE JANEIRO

2022

Sumário

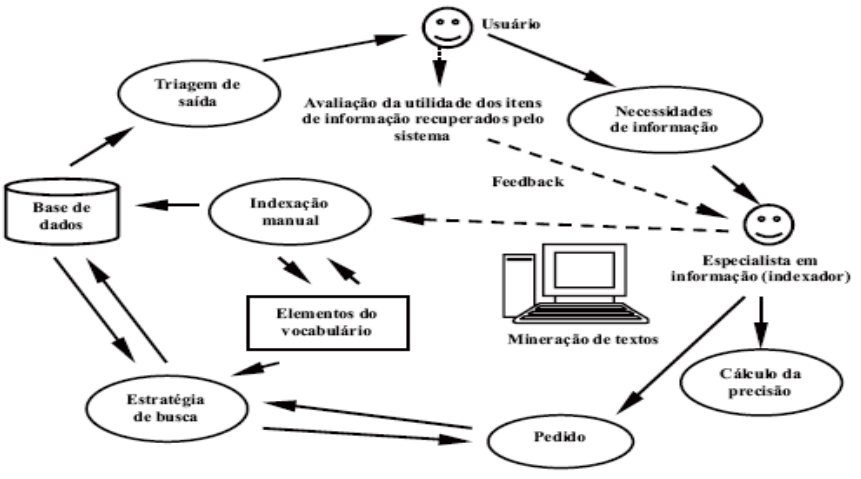
1. INTRODUÇÃO

A linguagem natural é a língua da maneira que os humanos comunicam e escrevem. O PLN, técnica responsável em processar esse linguajar, concilia a linguística, informática e a inteligência artificial - métodos de machine learning e deep learning. Com o uso desse recurso é possível recuperar informações escritas à natureza da língua.

Devido a interconexão entre sistemas informáticos e humanos, a quantidade de informações acessadas por computadores são cada vez maiores, assim, o processamento de linguagem natural, torna-se imprescindível para que as máquinas sejam capazes de ler e interpretá-las, a fim de entregar ao usuário uma informação enxuta e de qualidade. É possível através dessa técnica realizar sumarização, medir sentimentos e classificar as informações. Um dos exemplos mais famosos são os motores de busca que, através de pequenas frases, conseguem entregar documentos compatíveis com o que o utente deseja.

Desde os primórdios da organização social os humanos organizam as informações com intuito de facilitar a busca e a recuperação das mesmas. Um exemplo desse fato são as grandes bibliotecas, a mais antiga que se tem conhecimento foi criada em Elba, que com a criação de índices manuais que fornecem a categoria dos livros possibilitam uma melhor experiência a quem deseja encontrar algum texto.

FIGURA 1
Fatores de influência nos resultados de busca em uma base de dados



Fonte: adaptado de Lancaster (1998).

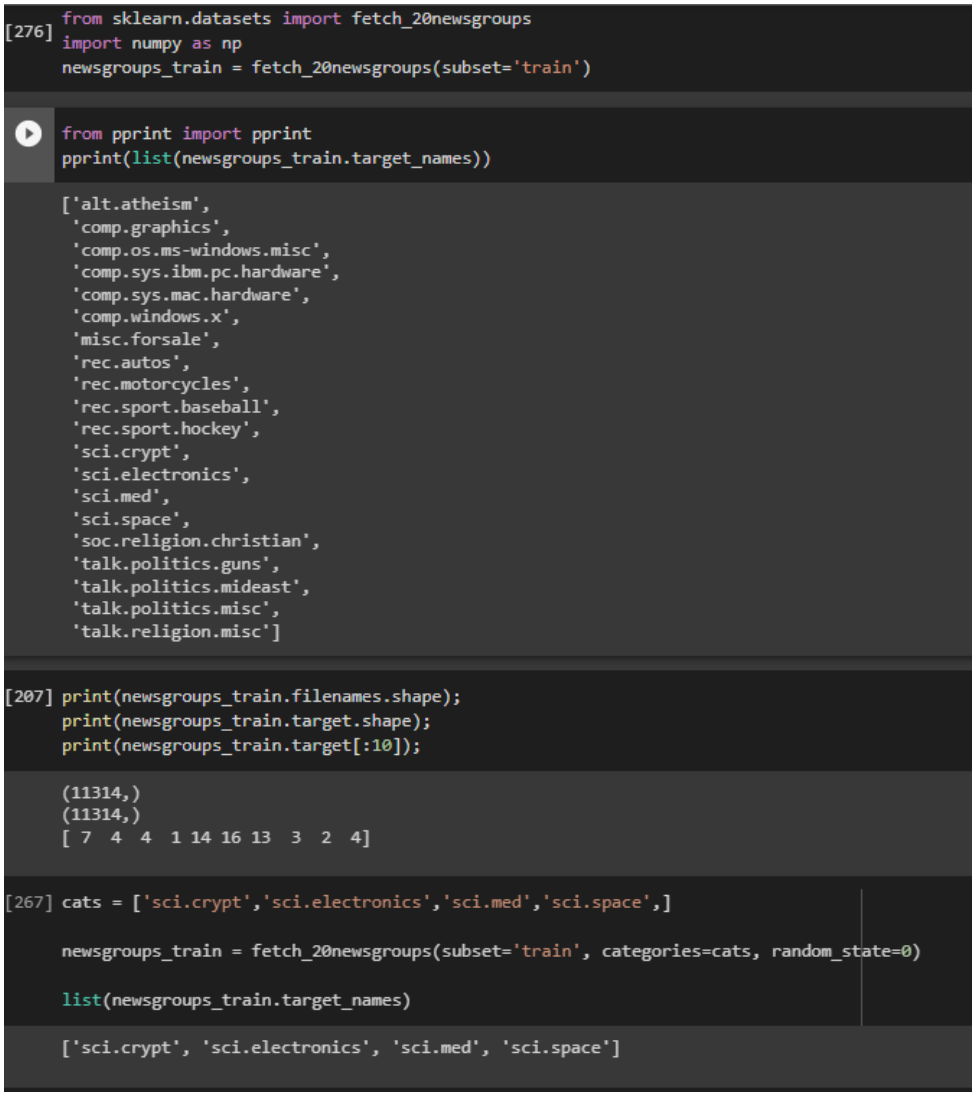
A partir do surgimento da internet, a recuperação de informações deixou de ser vista como algo relacionado apenas às bibliotecas, mas sim algo indispensável para lidar com o volume de conhecimento humano publicado na web. Os motores de busca tem que lidar de maneira assertiva com esse extenso repositório, ao passo que tem a função de traduzir a necessidade do usuário em algo que as máquinas possam entender e entregar algo que seja condizente com as palavras chaves informadas e não documentos ruidosos. Logo, essas tecnologias devem ser programadas, testadas e melhoradas continuamente.

As informações se diferem dos dados por agregarem valor, uma vez que elas dão sentido, enquanto os dados não são capazes, por si só, de descreverem algo, nesse viés, eles são a base para a geração da informação. Portanto, apesar de não representarem a mesma coisa, ambos caminham juntos.

No presente trabalho, será utilizado um corpus(conjunto de documentos) que compreende aproximadamente 18.000 postagens de grupos de notícias classificados em 20 tópicos e divididos subconjuntos, um para treinamento e outro para teste. O conjunto “The 20 newsgroups” faz parte da biblioteca sklearn.datasets. Os arquivos serão carregados, tokenizados, filtrados, refinados e ,posteriormente, serão base para treinar um classificador que terá, em sequência, seu desempenho avaliado. O objetivo é que a classificação do documento gerada pelo sistema deve ser condizente com a real.

2. METODOLOGIA

O treinamento do classificador será realizado na linguagem Python por intermédio da plataforma Google Colab. Em princípio, o corpus “The 20 newsgroups” será importado do conjunto de datasets sklearn.datasets. Em seguida serão definidas as categorias de documentos que iremos utilizar, pois não iremos trabalhar com as 20 categorias existentes.

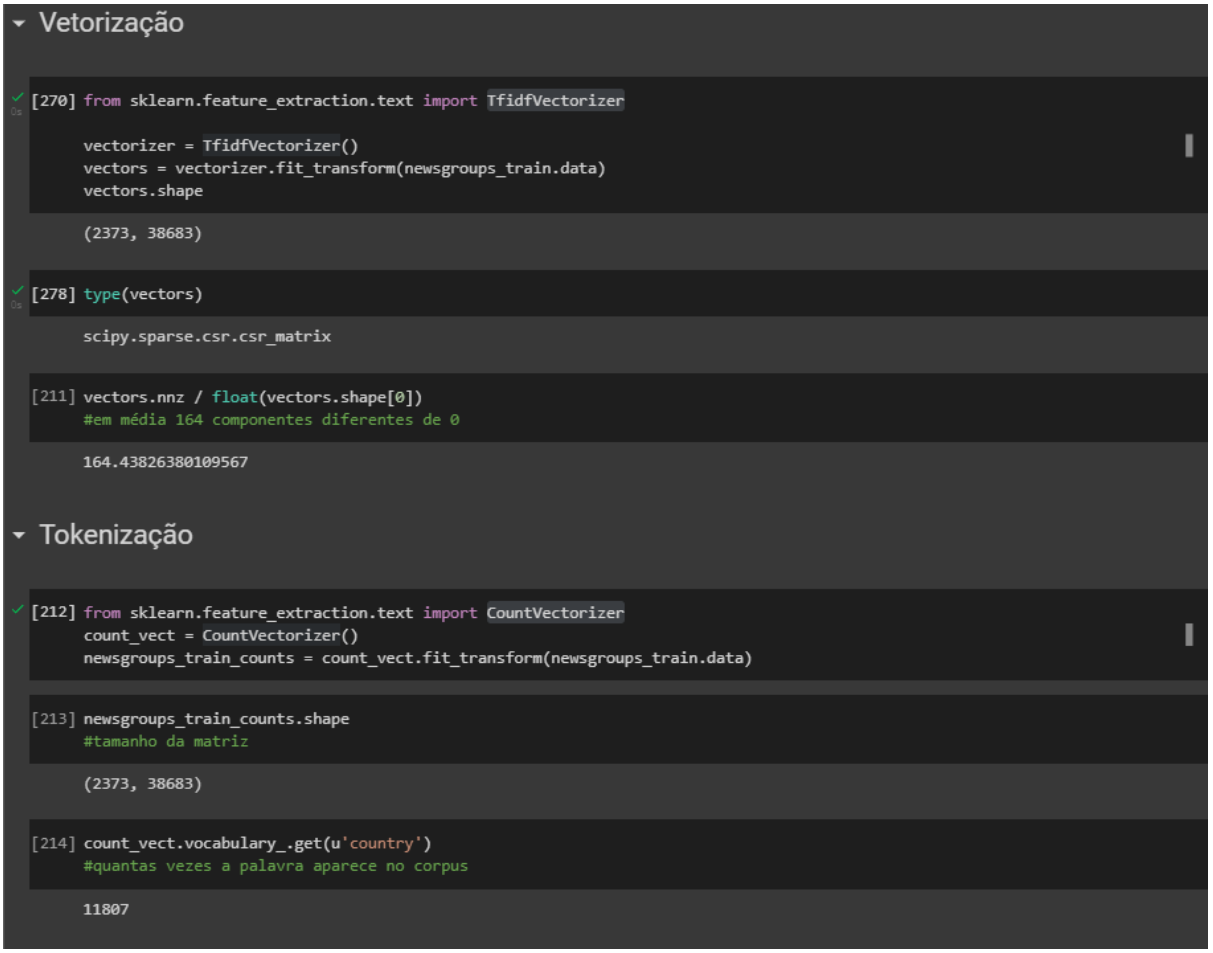


```
[276] from sklearn.datasets import fetch_20newsgroups
      import numpy as np
      newsgroups_train = fetch_20newsgroups(subset='train')

      from pprint import pprint
      pprint(list(newsgroups_train.target_names))

      ['alt.atheism',
       'comp.graphics',
       'comp.os.ms-windows.misc',
       'comp.sys.ibm.pc.hardware',
       'comp.sys.mac.hardware',
       'comp.windows.x',
       'misc.forsale',
       'rec.autos',
       'rec.motorcycles',
       'rec.sport.baseball',
       'rec.sport.hockey',
       'sci.crypt',
       'sci.electronics',
       'sci.med',
       'sci.space',
       'soc.religion.christian',
       'talk.politics.guns',
       'talk.politics.mideast',
       'talk.politics.misc',
       'talk.religion.misc']

[207] print(newsgroups_train.filenames.shape);
      print(newsgroups_train.target.shape);
      print(newsgroups_train.target[:10]);

      (11314,)
      (11314,)
      [ 7  4  4  1 14 16 13  3  2  4]

[267] cats = ['sci.crypt','sci.electronics','sci.med','sci.space',]

      newsgroups_train = fetch_20newsgroups(subset='train', categories=cats, random_state=0)

      list(newsgroups_train.target_names)

      ['sci.crypt', 'sci.electronics', 'sci.med', 'sci.space']
```

fonte: autoria própria

Para prosseguir com o trabalho os textos serão tokenizados, assim, eles serão convertidos em uma representação vetorial de recursos que faz referência a frequência dos termos, além e ser filtrado para retirar as stopwords, conjunto de palavras consideradas irrelevantes para a recuperação de informações como

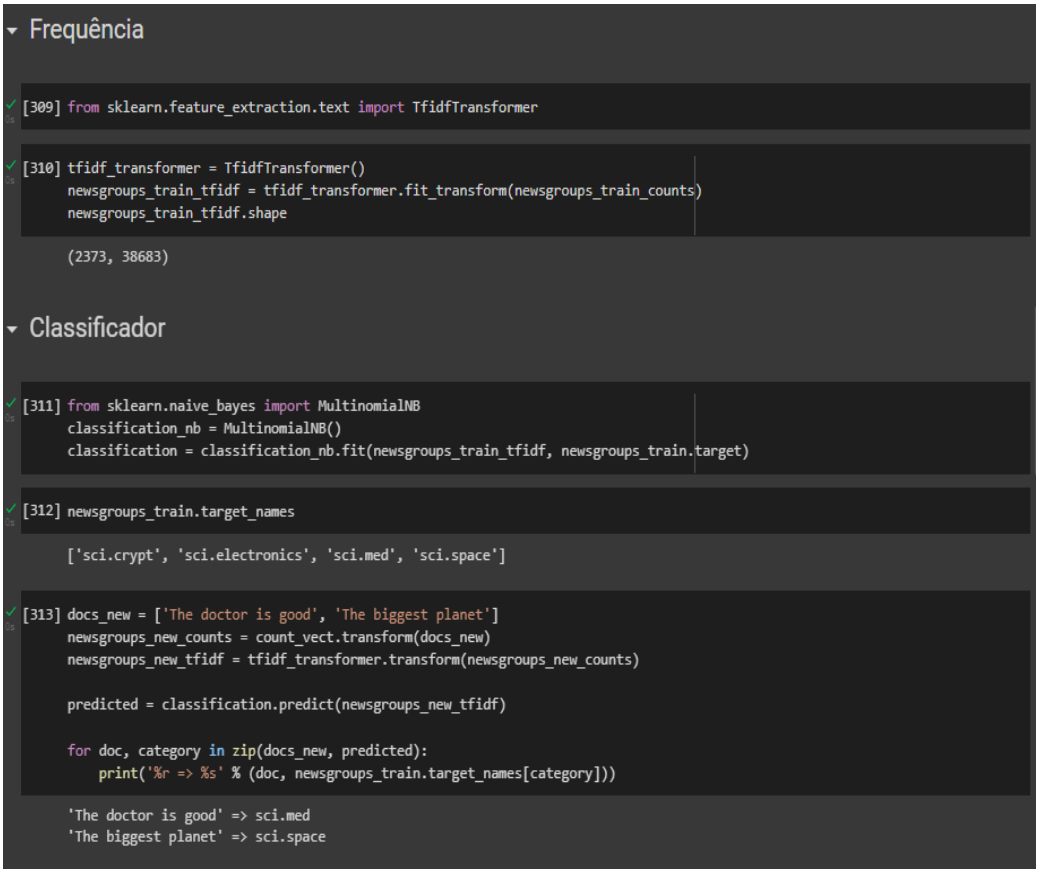
conjunções, e deixar o documento mais enxuto. Essas ações podem ser realizadas através tanto por meio do procedimento TfidfVectorizer, quanto do CountVectorizer. Para o próximo passo manteremos o foco no CountVectorizer.



fonte: autoria própria

Por meio do recurso TfidfTransformer em conjunto com o CountVectorizer a ocorrência dos termos serão transformadas em uma frequência de fato. Esse método leva em consideração não apenas quantas vezes a palavra aparece no documento, mas também a quantidade de palavras que ele contém e retorna a pontuação tf-idf.

Para o primeiro teste será utilizado o método multinomial Naive Bayes que considera características discretas, à exemplo, a pontuação tf-idf. Deste modo, serão escritos dois títulos fictícios de obras, por meio das palavras presentes o classificador irá categorizar os possíveis documentos em categorias de acordo com a tf-idf.



fonte: autoria própria

No exemplo acima os títulos ‘The doctor is goo’ e ‘The biggest planet’ foram categorizados em sci.med e sci.space, respectivamente.

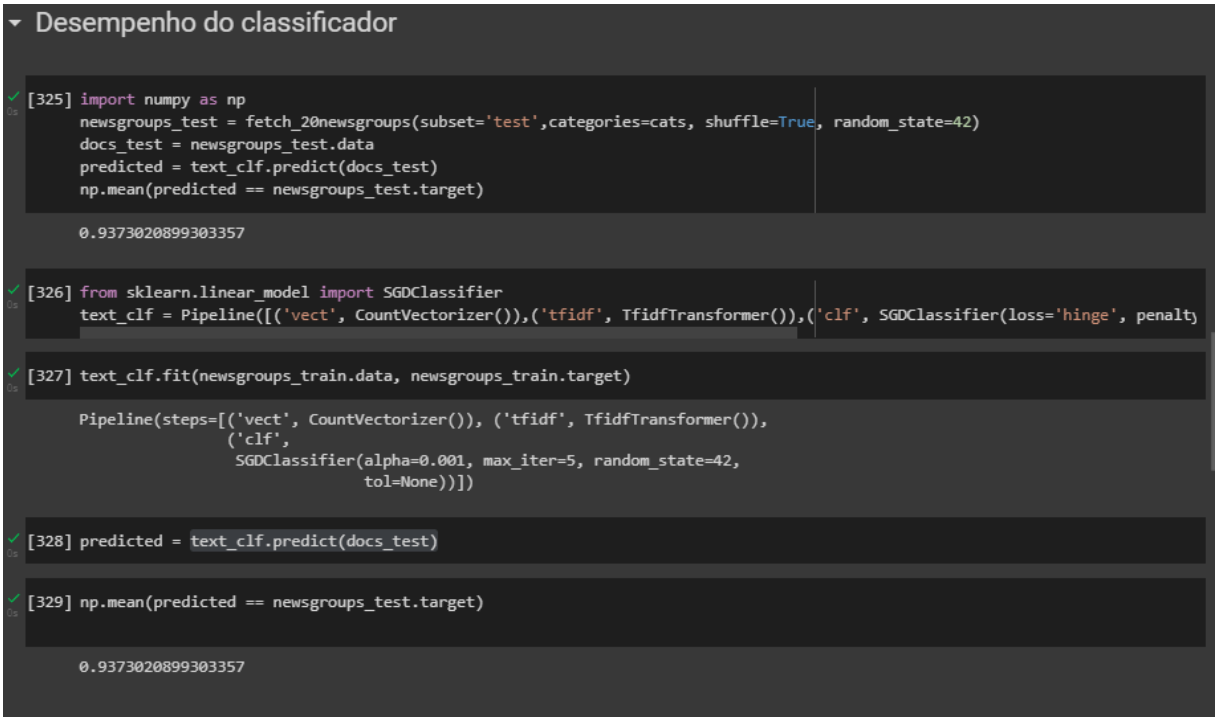
Por fim será construído um pipeline, uma espécie de classificador composto que une várias fases de machine learning em um fluxo. Logo, por meio dele é possível fazer todo trabalho realizado no classificador anterior em apenas uma linha.

fonte: autoria própria

3. RESULTADOS E DISCUSSÕES

Os classificadores serão avaliados de diferentes modos, a fim de analisar a precisão dos modelos. O desempenho de um classificador é de suma importância, pois atender uma requisição de maneira errônea pode ocasionar diversas frustrações.

Em princípio, será analisado o desempenho do SGDC, que funciona como um classificador linear com treinamento SGD, através do Pipeline. Em sequência trocaremos apenas o método multinomial por uma máquina de vetor de suporte linear para ver se a acurácia se difere. A qualidade será medida através de quantas vezes, em média, a categoria classificada se iguala à original.

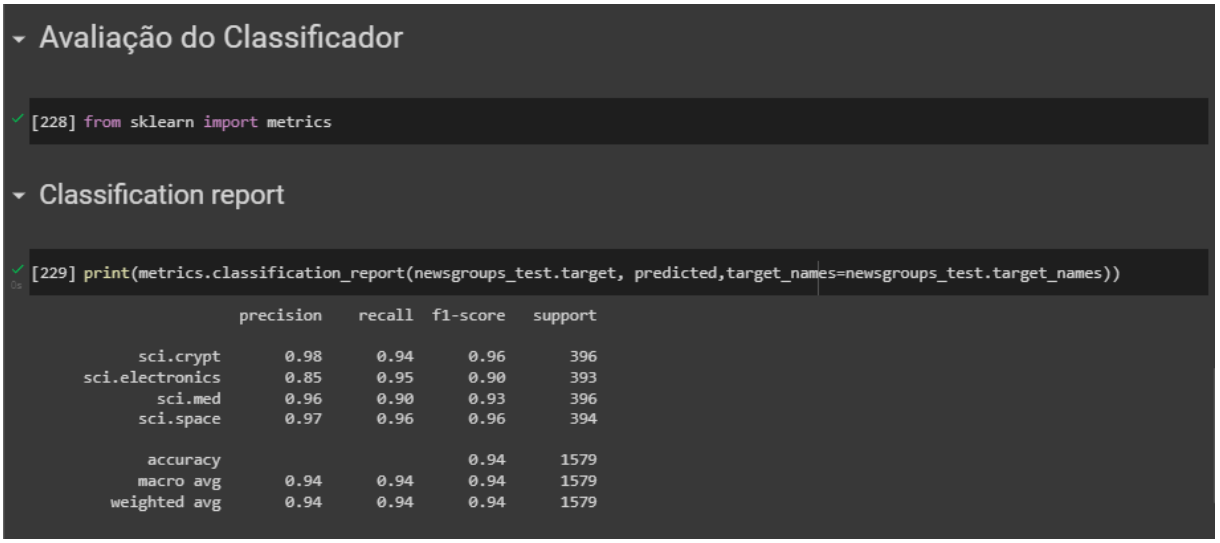


fonte: autoria própria

É possível verificar que ambos os classificadores alcançaram 93,73% de precisão, ou seja, responderam de maneira idêntica e, para essa classificação, não existe um classificador preferível.

Nesse momento, será elaborado um relatório com as principais métricas de classificação (precisão, revocação, F-measure) para cada uma das classes utilizadas para a análise.

- Precisão: preditivo positivo, retornará um valor relacionado a quantidade de elementos relevantes recuperados, ou seja, os verdadeiros positivos em relação ao total de elementos classificados;
- Revocação: sensibilidade, refere-se a quantidade de positivos recuperados, os positivos categorizados em relação ao total de positivos reais;
- F-measure: média harmônica ponderada de precisão e revocação, mede a eficácia.



fonte: autoria própria

Apenas a classe sci.med apresentou uma precisão menor que 90%, o que demonstra que o classificador apresentou um alto desempenho. Além disso, a revocação foi alta em todos os casos, portanto, o número de verdadeiros recuperados foi quase idêntico ao real.

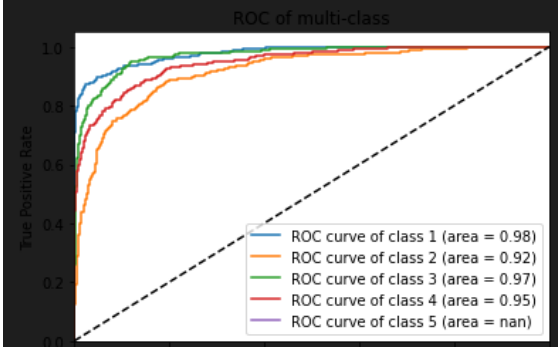
Para finalizar a avaliação da análise realizada serão plotadas curvas roque multiclasse. Os classificadores avaliados serão multinomial, linear e KNeigbords. O eixo Y dessa avaliação representa a taxa de verdadeiro positivo (TPR) resultante do classificador, enquanto o eixo X, a taxa de falso positivo(FPR). Nesse viés, o canto superior esquerdo do gráfico, com FPR zero e TPR um, é o ponto ideal.

As curvas serão apresentadas todas juntas em um mesmo gráfico. Para isso será necessário utilizar a estratégia multiclasse One-vs-the-Rest, assim, em cada etapa uma determinada classe será considerada positiva e as demais negativas.

```
ROC_multi_class(vectors, newsgroups_train.target, vectors_test, newsgroups_test.target, classification_nb)
```

Python

```
/usr/local/lib/python3.8/dist-packages/sklearn/multiclass.py:79: UserWarning: Label not 4 is present in all
training examples.
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_ranking.py:999: UndefinedMetricWarning: No positive
samples in y_true, true positive value should be meaningless
  warnings.warn(
```

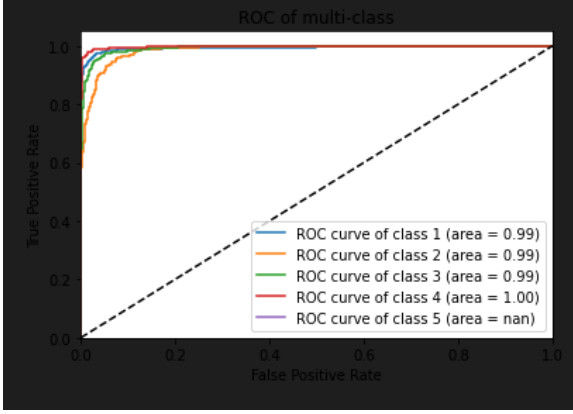


fonte: autoria própria

```
ROC_multi_class(vectors, newsgroups_train.target, vectors_test, newsgroups_test.target, clf_svc)
```

Python

```
/usr/local/lib/python3.8/dist-packages/sklearn/multiclass.py:79: UserWarning: Label not 4 is present in all
training examples.
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_ranking.py:999: UndefinedMetricWarning: No positive
samples in y_true, true positive value should be meaningless
  warnings.warn(
```

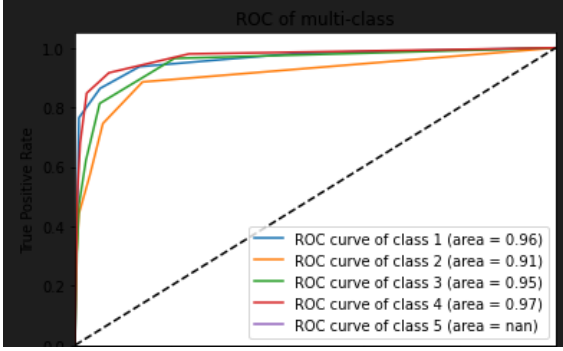


fonte: autoria própria

```
ROC_multi_class(vectors, newsgroups_train.target, vectors_test, newsgroups_test.target,clf_kn)
```

Python

```
/usr/local/lib/python3.8/dist-packages/sklearn/multiclass.py:79: UserWarning: Label not 4 is present in all
training examples.
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_ranking.py:999: UndefinedMetricWarning: No positive
samples in y_true, true positive value should be meaningless
  warnings.warn(
```



fonte: autoria própria

Os três classificadores apresentaram um bom desempenho, no entanto, a área sob a curva do modelo linear foi a maior e chegou a alcançar um valor 1. Isso pode ser explicado em razão de sua maior flexibilidade na escolha de penalidades e funções de perdas, além do fato que a classificação de vetores de suporte linear suporta um grande volume de dados e a multiplicidade de classes.

4. REFERÊNCIAS

JÚNIOR, Rogério Henrique de Araújo; TARAPANOFF, KIRA. Precisão no processo de busca e recuperação da informação: uso da mineração de textos.Scielo, 2007.Disponível em: <https://www.scielo.br/j/ci/a/wshBz99WbXhPWtD8MLQpDLd/?lang=pt#>. Acesso em: 14 Dez. 2022.

GUIMARÃES, Leandro. Qual a diferença entre dado e informação? Entenda agora!. Disponível em: <https://www.knowsolution.com.br/diferenca-dado-e-informacao/#:~:text=Enquanto%20os%20dados%20podem%20ser,trabalho%20com%20elementos%20mais%20brutos.>. Acesso em: 14 Dez. 2022.

WIKIPÉDIA. Precisão e revocação. Disponível em: <https://pt.wikipedia.org/wiki/Precis%C3%A3o_e_revoca%C3%A7%C3%A3o> Acesso em: 15 Dez. 2022.

MICROSOFT. Pipeline Classe. Disponível em: <https://learn.microsoft.com/pt-br/python/api/azureml-pipeline-core/azureml.pipeline.core.pipeline.pipeline?view=azure-ml-py>. Acesso em: 15 Dez. 2022.

BAEZA-YATES, R.; RIBEIRO-NETO, B. Recuperação de Informação. Bookman, 2013.. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582600498/. Acesso em: 16 Dec 2022.

SCIKIT-LEARN.Aprendizado de máquina em Python. Disponível em: <https://scikit-learn.org/stable/index.html> . Acesso em: 10 Dez. 2022.