Preprocessando e extraindo dados relevantes de textos com tf-idf

Igor Moura

- Sei brincar com python
- Mexo com eletrônicos
- Gosto de sofrer com computadores

BEYOND CLOUD

E.S.T.U.F.

@igormp

Mas que danado é tf-idf?

term frequency–inverse document frequency é função matemática que **rankeia** palavras dada a sua relevância

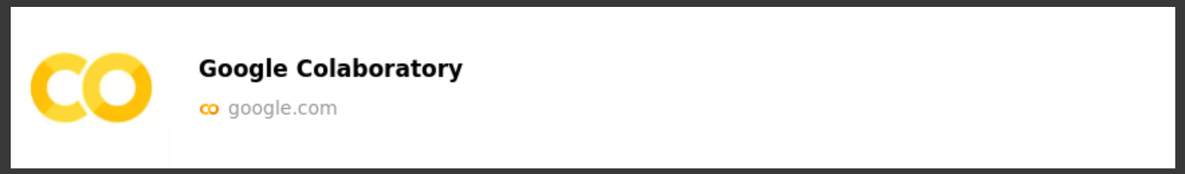
$$TF\text{-}IDF = TF(t, d) * IDF(t, D)$$

$$TF(t,d) = \frac{count(t,d)}{count(*,d)}$$

$$IDF(t, D) = log_e(\frac{D}{amount(t, D)})$$

E como implementar mesmo?

Experimente no Colab!



```
!wget http://mlg.ucd.ie/files/datasets/bbc-fulltext.zip
!unzip bbc-fulltext.zip
!rm bbc-fulltext.zip*
%cd bbc
!ls
!rm README.TXT
```

```
documents = []
for root, dirs, files in os.walk("."):
    for name in files:
        if name.endswith((".txt")):
            filepath = root + os.sep + name
            with open(filepath, 'r', encoding='latin-1') as file:
                data = file.read().replace('\n', ' ').split()
                documents.append(data)
```

```
documents = [[word for word in sublist if (len(word)>1)]
             for sublist in documents]
```

```
documents = [[word.lower() for word in sublist]
             for sublist in documents]
```

```
symbols = "!\"#$%&()*+-./:;<=>?@[\]^_`{|}~\'"
for i in symbols:
    documents = [[''.join([char for char in word if char != i])
                  for word in sublist]
                  for sublist in documents]
```

```
nltk.download('stopwords')
stopwords = nltk.corpus.stopwords.words("english")
documents = [[word for word in sublist if word not in stopwords]
             for sublist in documents]
```

```
lemmatizer = WordNetLemmatizer()
documents = [[lemmatizer.lemmatize(word) for word in sublist]
             for sublist in documents]

stemmer = SnowballStemmer('english')
documents = [[stemmer.stem(word) for word in sublist]
             for sublist in documents]
```

Stemming:

studies -> studi

Lemmatization:

studies -> study

- Remoção de palavras com tamanho unitário
- Deixar tudo em minúsculas
- Remover símbolos
- Remover "stopwords"
- Lemmatização seguida de stemmização

```
dfs = {}
for i in range(len(documents)):
    for word in documents[i]:
        try:
            dfs[word].add(i)
        except:
            dfs[word] = {i}
```

```
1 for i in dfs:
2     dfs[i] = len(dfs[i])
```

```
tf_idf = {}
for i in range(len(documents)):
    counter = Counter(documents[i])
    for token in np.unique(documents[i]):
        tf = counter[token]/len(documents[i])
        df = dfs[token]
        idf = np.log(len(documents)/(df+1))
        tf_idf[i, token] = tf*idf
```

```
(pd.DataFrame(list(tf_idf.items()))
    .sort_values(by=[1], ascending=False)
    .head(5))
```

	0	1
200533	(1181, rugby),)	0.716521
66307	(324, commodor)	0.492562
32093	(162, commodor)	0.492562
216404	(1297, pountney)	0.477752
91123	(451, axa)	0.458059

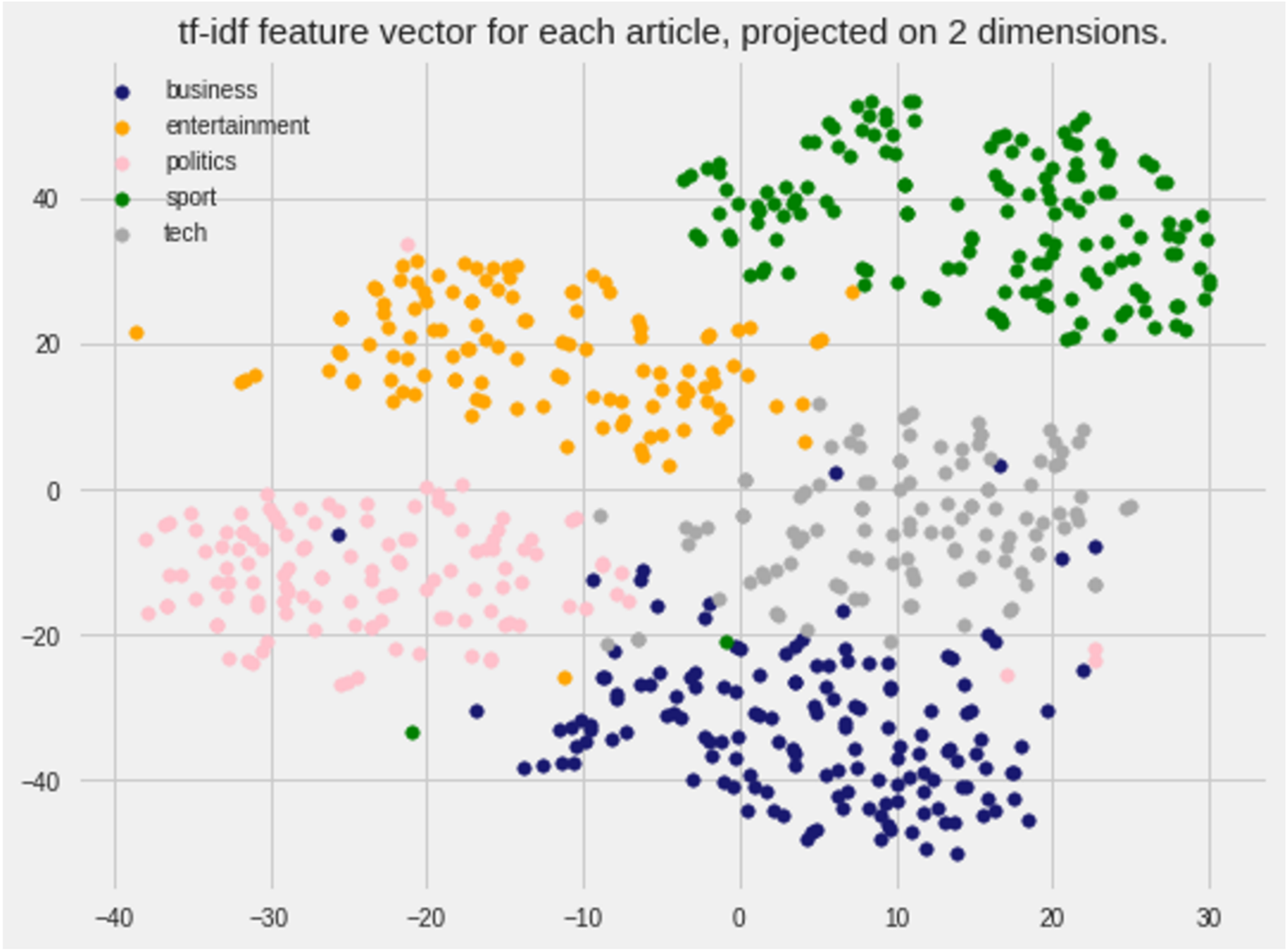
```
1 from sklearn.feature_extraction.text import TfidfVectorizer
2
3 tfidf = TfidfVectorizer(sublinear_tf=True,
4                         min_df=5, norm='l2',
5                         encoding='latin-1',
6                         ngram_range=(1, 2),
7                         stop_words='english')
```

Mas qual a utilidade disso?

Ferramentas de busca

- Recuperação de informação
- Rankamento de páginas
- Extração de keywords

Natural Language Processing

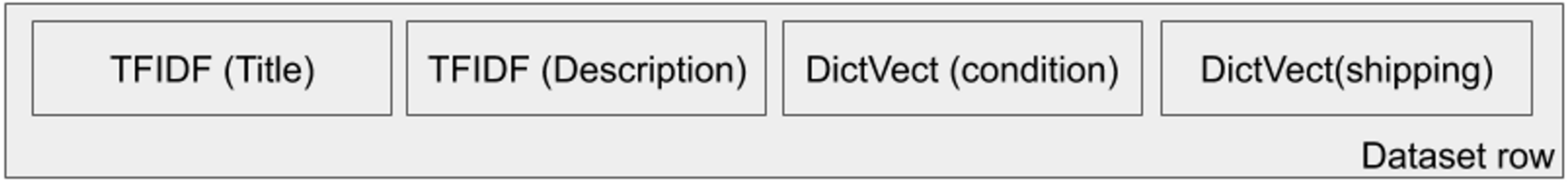

tf-idf feature vector for each article, projected on 2 dimensions.
business
entertainment
politics
sport
tech
40
20
0
−20
−40
−40
−30
−20
−10
0
10
20
30

Mercado Livre

Using TensorFlow to predict product weight and dimensions

A Guest Post by Rodolfo Bonnin from the Mercado Libre Applied Machine Learning team

Medium / TensorFlow / Sep 25



Como utilizamos na firma

MARTORELLI

ADVOGADOS

CERVEJARIA

ambev

- Petição inicial
- Ata
- Sentença
- Acordão
- Alvarás

Obrigado!

igormp2

igormp

igormp

imp2@cin.ufpe.br

BEYOND CLOUD

E.S.T.U.F.