REPORT 1

1　準備したデータ

以下の 10 個のデータを Google News から入手し、テキスト化した。
用意したデータは、１と２、３と４、５と６と７、８と９と１０で分類されている。

1.1　大阪府、宣言延長要請へ調整　知事「緩和、解除は困難」
https://news.yahoo.co.jp/articles/1e23e29fbcd51e4772e545ff893e0f0ef8438a82

1.2　大阪府で新たに８８４人感染、２日連続１千人下回る…吉村知事が明らかに
https://www.yomiuri.co.jp/national/20210504-OYT1T50096

1.3　北アルプス・槍ケ岳で遭難、3 人死亡　現場は当時吹雪
https://www.asahi.com/articles/ASP5445GWP54UOHB001.html

1.4　北アルプス・槍ヶ岳で遭難　男性 3 人死亡　1 人は滑落　周辺は吹雪で視界不良「ホワイトアウト」に近い状態
https://news.yahoo.co.jp/articles/3d68d08ddfef1001e752e272f57d88f52a8d2adf

1.5　多摩川でバーベキューの男性が死亡…服着たまま川を渡ろうとする
https://article.auone.jp/detail/1/2/2/162_2_r_20210504_1620123602307121

1.6　多摩川で流され？20 歳男性死亡　仲間とバーベキュー
https://news.yahoo.co.jp/articles/e52ab9d53124f5259873ab472779798d31a13175

1.7　多摩川河川敷でバーベキューの男性 川で溺れ死亡 川崎
https://www3.nhk.or.jp/shutoken-news/20210504/1000063967.html

1.8　ブラマヨ小杉　〝ドン・ファン〟元妻の単独犯行に疑問「別にもう一人おらんとなんか…」
https://article.auone.jp/detail/1/5/9/103_9_r_20210504_1620105002220643

1.9　“紀州のドン・ファン”元妻の飼い猫、逮捕後放置されていた　須藤容疑者は友人に「動物保護の仕事を」と将来の目標も
https://maidonanews.jp/article/14342315

1.10　「紀州のドン・ファン」元妻と共演したセクシー男優「凄く印象深い」と発言し、大反響〈dot.〉
https://news.yahoo.co.jp/articles/f7bd49cd73221ef84bff445966c4dc0594451e34

2　作成したプログラム

2.1　要求[1]

```
import MeCab
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

documents = []
output_file = open("./text/output_1.txt","w")

for i in range(1, 11):
    wikifile = open("./text/"+str(i)+".txt","r",encoding="utf_8")
    tagger = MeCab.Tagger()
    for file in wikifile:
        node = tagger.parseToNode(file)
        words = ""
        while node:
            node_features=node.feature.split(",")
            if node_features[0]=="名詞" and (node_features[1]=="一般" or node_features[1]=="固有名
詞"):
                words = words+" "+node.surface
            node = node.next
    documents.append(words)

npdocs=np.array(documents)
vectorizer = TfidfVectorizer(norm=None, smooth_idf=False,max_features=100)
vecs = vectorizer.fit_transform(npdocs)

terms = vectorizer.get_feature_names()
print("単語文書行列（TF-IDF)=",file=output_file)
print("単語¥t",end='',file=output_file)
for term in terms:
    print("%6s" % term, end='',file=output_file)
print("¥n",file=output_file)
```

```
tfidfs = vecs.toarray()
for n, tfidf in enumerate(tfidfs):
    print("文書", n+1, "¥t", end='',file=output_file)
    for t in tfidf:
        print("%8.4f" % t, end='',file=output_file)
    print("¥n",file=output_file)
```

2.2　要求[2]

```
import MeCab
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

documents = []
output_file = open("./text/output_2.txt","w")

for i in range(1, 11):
    wikifile = open("./text/"+str(i)+".txt","r",encoding="utf_8")
    tagger = MeCab.Tagger()
    for file in wikifile:
        node = tagger.parseToNode(file)
        words = ""
        while node:
            node_features=node.feature.split(",")
            if node_features[0]=="名詞" and (node_features[1]=="一般" or node_features[1]=="固有名
詞"):
                words = words+" "+node.surface
            node = node.next
    documents.append(words)

npdocs=np.array(documents)
vectorizer = TfidfVectorizer(norm=None, smooth_idf=False)
vecs = vectorizer.fit_transform(npdocs)

terms = vectorizer.get_feature_names()

tfidfs = vecs.toarray()

print("文書番号¥t",end='',file=output_file)
for n in range(1,11):
    print("文書", n, "¥t", end='',file=output_file)
print("¥n",file=output_file)
```

```
similarity = cosine_similarity(tfidfs)
for n, simi in enumerate(similarity):
    print("文書", n+1, "¥t", end='',file=output_file)
    for t in simi:
        print("%8.4f" % t, end='',file=output_file)
    print("¥n",file=output_file)
```

3　実行結果に対する考察

3.1　要求[1]

要求[1]に基づき作成したプログラムから出力された結果を以下に示す。

```
単語文書行列（TF-IDF）=
単語     キャリーバッグ    コロナ    ダリア     ドンバーベキュー    ファン    フード    ペット マンション     ワン
文書 1     0.0000  2.6094  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
文書 2     0.0000  2.6094  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
文書 3     0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
文書 4     0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
文書 5     0.0000  0.0000  0.0000  0.0000  4.4079  0.0000  0.0000  0.0000  0.0000  0.0000  4.4079
文書 6     0.0000  0.0000  0.0000  0.0000  4.4079  0.0000  0.0000  0.0000  0.0000  0.0000  4.4079
文書 7     0.0000  0.0000  0.0000  0.0000  6.6119  0.0000  0.0000  0.0000  0.0000  0.0000  2.2040
文書 8     0.0000  0.0000  0.0000  4.4079  0.0000  4.4079  0.0000  0.0000  0.0000  0.0000  0.0000
文書 9     6.6052  0.0000 13.2103  2.2040  0.0000  2.2040  6.6052 36.3284  6.6052  6.6052  0.0000
文書 10       0.0000  0.0000  0.0000  2.2040  0.0000  2.2040  0.0000  0.0000  0.0000  0.0000  0.0
```

この結果を見てみると、先頭に来ているキャリーバッグでは文書 9 での重要度が高く算出されている。文書 9 は、https://maidonanews.jp/article/14342315 のニュースで確かにキャリーバッグが入っていることがわかる。

元神奈川県警刑事で犯罪ジャーナリストの小川泰平氏は当サイトの取材に対し、友人の証言として同容疑者が「動物保護の仕事をしてペットの殺処分をなくしたい」という将来の目標を語っていたことを明かした。

小川氏は「須藤容疑者の友人から猫の画像を入手しました。その友人によると、この猫はメスの保護猫で、5歳ぐらいだと容疑者から聞いており、名前は『ダリア』ちゃんとのことです」と明かし、「警察は須藤容疑者を逮捕後、この猫をマンション内に放置したまま帰ってしまったようです。その後、警察から『ダリア』ちゃんに関してはどこにも連絡がないことから、友人、知人らが連絡を取り合い無事保護しています」と説明した。

「東京・品川区内のタワーマンションにある須藤容疑者の部屋にはペットキャリーバッグがあったが、ダリアちゃんは放置されたままだったようです」。そう説明した小川氏は「保護された猫は毛並みもよく、ペットサロンできれいにトリミングされていたようです。中学時代からの友人（同級生）を取材したのですが、その友人女性によると、須藤容疑者は『将来、動物保護活動の仕事をしたい。（ペットの）殺処分をなくしたい』と話していたそうです」と明かした。

さらに、猫と須藤容疑者の生活について、小川氏は「半年に１度ほどのペースで都内の自宅を捜索されており、その度に引っ越していたようですが、猫はペット用のキャリーバッグに入れて移動していたようです」と補足した。

次に、コロナという単語について見てみる。
この単語では文書１、２で重要度が高く算出されている。
文書１、２を確認すると、確かにコロナという単語が多く使われていることがわかる。

大阪府、宣言延長要請へ調整　知事「緩和、解除は困難」

5/4(火) 17:32 配信 2542

KYODO

1 / 2

取材に応じる大阪府の吉村洋文知事＝4日午後、大阪府庁

新型コロナの緊急事態宣言に関し、大阪府が11日に迫った期限の延長を政府に要請する方向で調整していることが4日、関係者への取材で分かった。吉村洋文知事は同日、記者団に「現状の認識としては、今の措置の内容を緩めたり解除したりするのは難しいと思っている」と述べた。6日か7日に対策本部会議を開き、考えを決定する。府は4日、884人の感染と20人の死亡を発表した。

大阪府内の感染状況と吉村知事の主な発言

大阪府の吉村洋文知事は４日に出演した民放番組で、新たに８８４人が新型コロナウイルスに感染したことを明らかにした。１０００人を下回ったのは２日連続。

ここでは、上記の 1.で示した 10 個のニュースの各単語の各記事における TF-IDF を MeCab および scikit-learn を用いて算出した。これらの結果を踏まえると、今回のソースコードで重要度を算出することができており、実際に文書内の単語の重要度を測れていることが確認できた。

3.2　要求[2]

要求[2]に基づいて作成したプログラムによって生成された結果を以下に示す。

文書番号	文書 1	文書 2	文書 3	文書 4	文書 5	文書 6	文書 7	文書 8	文書 9	文書 10
文書 1	1.0000	0.4157	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
文書 2	0.4157	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0354	0.0000	0.0000
文書 3	0.0000	0.0000	1.0000	0.1685	0.0482	0.1351	0.0389	0.0000	0.0105	0.0083
文書 4	0.0000	0.0000	0.1685	1.0000	0.1514	0.1811	0.1689	0.0000	0.0406	0.0172
文書 5	0.0000	0.0000	0.0482	0.1514	1.0000	0.6267	0.6128	0.0000	0.0847	0.0179
文書 6	0.0000	0.0000	0.1351	0.1811	0.6267	1.0000	0.5498	0.0000	0.0520	0.0320
文書 7	0.0000	0.0000	0.0389	0.1689	0.6128	0.5498	1.0000	0.0000	0.1833	0.0000
文書 8	0.0000	0.0354	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.2395	0.4007
文書 9	0.0000	0.0000	0.0105	0.0406	0.0847	0.0520	0.1833	0.2395	1.0000	0.3180
文書 10	0.0000	0.0000	0.0083	0.0172	0.0179	0.0320	0.0000	0.4007	0.3180	1.0000

上記の結果から、類似度が 0.2 以上の場合の分類をすると、

- 文書１、文書２
- 文書５、文書６、文書７
- 文書８、文書９、文書１０

となる。

今回の正常な分類としては、

- 文書１、文書２
- 文書３、文書４
- 文書５、文書６、文書７
- 文書８、文書９、文書１０

であるので、文書３、４の類似度以外は大方問題なく分類されていることがわかる。

多少の誤差はあったが、大方正しい類似度が算出できており、要求[1]で求めた単語文書行列を使用した文書間のコサイン類似度の計算がある程度の精度で上手く計算できていることがわかった。また、今回使用したテキストファイル、出力ファイルなどを下記のリポジトリでバージョン管理した。

https://github.com/10kaoru12/4y_university_information_recommender_system