Обнаружение событий, связанных с копроментацией корпоративной электронной почты, в процессе мониторинга исходящей корреспонденции на основе методов машинного обучения

Подготовка датасетов

Используем открьтый датасет компании Enron. Разархивируем его и превратим его в нужные нам датасеты.

In [1]:
```
!pip install -r requirements.txt
```

```
WARNING: The directory '/home/tech/.cache/pip' or its parent directory is no
t owned or is not writable by the current user. The cache has been disabled.
Check the permissions and owner of that directory. If executing pip with sud
o, you may want sudo's -H flag.
Requirement already satisfied: pandas==2.2.1 in /home/tech/.local/lib/python
3.9/site-packages (from -r requirements.txt (line 1)) (2.2.1)
Requirement already satisfied: mail-parser==3.15.0 in /home/tech/.local/lib/
python3.9/site-packages (from -r requirements.txt (line 2)) (3.15.0)
Requirement already satisfied: scikit-learn==1.4.1.post1 in /home/tech/.loca
l/lib/python3.9/site-packages (from -r requirements.txt (line 3)) (1.4.1.pos
t1)
Requirement already satisfied: numpy==1.26.4 in /home/tech/.local/lib/python
3.9/site-packages (from -r requirements.txt (line 4)) (1.26.4)
Requirement already satisfied: nltk==3.8.1 in /home/tech/.local/lib/python3.
9/site-packages (from -r requirements.txt (line 5)) (3.8.1)
Requirement already satisfied: tensorflow==2.16.1 in /home/tech/.local/lib/p
ython3.9/site-packages (from -r requirements.txt (line 6)) (2.16.1)
Requirement already satisfied: torch==2.2.1 in /home/tech/.local/lib/python
3.9/site-packages (from -r requirements.txt (line 7)) (2.2.1)
Requirement already satisfied: matplotlib==3.8.3 in /home/tech/.local/lib/py
thon3.9/site-packages (from -r requirements.txt (line 8)) (3.8.3)
Requirement already satisfied: seaborn==0.13.2 in /home/tech/.local/lib/pyth
on3.9/site-packages (from -r requirements.txt (line 9)) (0.13.2)
Requirement already satisfied: tqdm==4.66.2 in /home/tech/.local/lib/python
3.9/site-packages (from -r requirements.txt (line 10)) (4.66.2)
Collecting transformers==4.38.2
  Downloading transformers-4.38.2-py3-none-any.whl (8.5 MB)

     |                                  | 10 kB 16.2 MB/s eta 0:00:01
     |                                  | 20 kB 2.6 MB/s eta 0:00:04
     ||                                 | 30 kB 3.8 MB/s eta 0:00:03
```

```
|                                        | 40 kB 1.3 MB/s eta 0:00:07
|                                        | 51 kB 1.4 MB/s eta 0:00:07
|                                        | 61 kB 1.7 MB/s eta 0:00:06
|                                        | 71 kB 1.8 MB/s eta 0:00:05
|                                        | 81 kB 1.9 MB/s eta 0:00:05
|                                        | 92 kB 2.1 MB/s eta 0:00:05
|                                        | 102 kB 1.6 MB/s eta 0:00:06
|                                        | 112 kB 1.6 MB/s eta 0:00:06
|                                        | 122 kB 1.6 MB/s eta 0:00:06
|                                        | 133 kB 1.6 MB/s eta 0:00:06
|                                        | 143 kB 1.6 MB/s eta 0:00:06
|                                        | 153 kB 1.6 MB/s eta 0:00:06
|                                        | 163 kB 1.6 MB/s eta 0:00:06
|                                        | 174 kB 1.6 MB/s eta 0:00:06
|                                        | 184 kB 1.6 MB/s eta 0:00:06
|                                        | 194 kB 1.6 MB/s eta 0:00:06
|                                        | 204 kB 1.6 MB/s eta 0:00:06
|                                        | 215 kB 1.6 MB/s eta 0:00:06
|                                        | 225 kB 1.6 MB/s eta 0:00:06
|                                        | 235 kB 1.6 MB/s eta 0:00:06
|                                        | 245 kB 1.6 MB/s eta 0:00:06
|                                        | 256 kB 1.6 MB/s eta 0:00:06
|                                        | 266 kB 1.6 MB/s eta 0:00:06
|                                        | 276 kB 1.6 MB/s eta 0:00:06
|                                        | 286 kB 1.6 MB/s eta 0:00:06
|                                        | 296 kB 1.6 MB/s eta 0:00:06
|                                        | 307 kB 1.6 MB/s eta 0:00:06
|                                        | 317 kB 1.6 MB/s eta 0:00:06
|                                        | 327 kB 1.6 MB/s eta 0:00:06
|                                        | 337 kB 1.6 MB/s eta 0:00:06
|                                        | 348 kB 1.6 MB/s eta 0:00:06
|                                        | 358 kB 1.6 MB/s eta 0:00:06
|                                        | 368 kB 1.6 MB/s eta 0:00:06
|                                        | 378 kB 1.6 MB/s eta 0:00:06
|                                        | 389 kB 1.6 MB/s eta 0:00:06
|                                        | 399 kB 1.6 MB/s eta 0:00:06
|                                        | 409 kB 1.6 MB/s eta 0:00:06
|                                        | 419 kB 1.6 MB/s eta 0:00:06
|                                        | 430 kB 1.6 MB/s eta 0:00:06
|                                        | 440 kB 1.6 MB/s eta 0:00:06
|                                        | 450 kB 1.6 MB/s eta 0:00:06
|                                        | 460 kB 1.6 MB/s eta 0:00:06
|                                        | 471 kB 1.6 MB/s eta 0:00:06
|                                        | 481 kB 1.6 MB/s eta 0:00:06
|                                        | 491 kB 1.6 MB/s eta 0:00:06
|                                        | 501 kB 1.6 MB/s eta 0:00:06
|                                        | 512 kB 1.6 MB/s eta 0:00:06
|                                        | 522 kB 1.6 MB/s eta 0:00:06
|                                        | 532 kB 1.6 MB/s eta 0:00:06
```

```
|▉▉                                      | 542 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 552 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 563 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 573 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 583 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 593 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 604 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 614 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 624 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 634 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 645 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 655 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 665 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 675 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 686 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 696 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 706 kB 1.6 MB/s eta 0:00:06
|▉▉                                      | 716 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 727 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 737 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 747 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 757 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 768 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 778 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 788 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 798 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 808 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 819 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 829 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 839 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 849 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 860 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 870 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 880 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 890 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 901 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 911 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 921 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 931 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 942 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 952 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 962 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 972 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 983 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 993 kB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.0 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.0 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.0 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.0 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.0 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉                                      | 1.1 MB 1.6 MB/s eta 0:00:05
```

```
|▉▉▉▉                                     | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉                                     | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉                                     | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉                                     | 1.1 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.2 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉                                    | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.3 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.4 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.5 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.6 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.7 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.7 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉                                   | 1.7 MB 1.6 MB/s eta 0:00:05
```

```
|▉▉▉▉▉▉▉                                  | 1.7 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.7 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.7 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.7 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.7 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.7 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.8 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
|▉▉▉▉▉▉▉▉                                 | 1.9 MB 1.6 MB/s eta 0:00:05
```

```
|                                        | 1.9 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.0 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.1 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.2 MB 1.6 MB/s eta 0:00:05
|                                        | 2.3 MB 1.6 MB/s eta 0:00:05
|                                        | 2.3 MB 1.6 MB/s eta 0:00:05
|                                        | 2.3 MB 1.6 MB/s eta 0:00:05
|                                        | 2.3 MB 1.6 MB/s eta 0:00:04
|                                        | 2.3 MB 1.6 MB/s eta 0:00:04
|                                        | 2.3 MB 1.6 MB/s eta 0:00:04
|                                        | 2.3 MB 1.6 MB/s eta 0:00:04
|                                        | 2.3 MB 1.6 MB/s eta 0:00:04
|                                        | 2.3 MB 1.6 MB/s eta 0:00:04
|                                        | 2.3 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.4 MB 1.6 MB/s eta 0:00:04
|                                        | 2.5 MB 1.6 MB/s eta 0:00:04
|                                        | 2.5 MB 1.6 MB/s eta 0:00:04
|                                        | 2.5 MB 1.6 MB/s eta 0:00:04
|                                        | 2.5 MB 1.6 MB/s eta 0:00:04
|                                        | 2.5 MB 1.6 MB/s eta 0:00:04
|                                        | 2.5 MB 1.6 MB/s eta 0:00:04
```

```
|                                                  | 2.5 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.5 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.5 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.5 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.6 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 2.9 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.0 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.1 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.1 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.1 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.1 MB 1.6 MB/s eta 0:00:04
```

```
|                                        | 3.1 MB 1.6 MB/s eta 0:00:04
|                                        | 3.1 MB 1.6 MB/s eta 0:00:04
|                                        | 3.1 MB 1.6 MB/s eta 0:00:04
|                                        | 3.1 MB 1.6 MB/s eta 0:00:04
|                                        | 3.1 MB 1.6 MB/s eta 0:00:04
|                                        | 3.1 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.2 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.3 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.4 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.5 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.6 MB 1.6 MB/s eta 0:00:04
|                                        | 3.7 MB 1.6 MB/s eta 0:00:04
```

```
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.7 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.8 MB 1.6 MB/s eta 0:00:04
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 3.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
```

```
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.4 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.5 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.6 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
```

```
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.7 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.8 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 4.9 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.0 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.1 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.2 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.3 MB 1.6 MB/s eta 0:00:03
|                                                  | 5.3 MB 1.6 MB/s eta 0:00:03
```

```
|                                        | 5.3 MB 1.6 MB/s eta 0:00:03
|                                        | 5.3 MB 1.6 MB/s eta 0:00:03
|                                        | 5.3 MB 1.6 MB/s eta 0:00:03
|                                        | 5.3 MB 1.6 MB/s eta 0:00:03
|                                        | 5.3 MB 1.6 MB/s eta 0:00:03
|                                        | 5.3 MB 1.6 MB/s eta 0:00:03
|                                        | 5.3 MB 1.6 MB/s eta 0:00:03
|                                        | 5.4 MB 1.6 MB/s eta 0:00:03
|                                        | 5.4 MB 1.6 MB/s eta 0:00:03
|                                        | 5.4 MB 1.6 MB/s eta 0:00:03
|                                        | 5.4 MB 1.6 MB/s eta 0:00:03
|                                        | 5.4 MB 1.6 MB/s eta 0:00:03
|                                        | 5.4 MB 1.6 MB/s eta 0:00:03
|                                        | 5.4 MB 1.6 MB/s eta 0:00:02
|                                        | 5.4 MB 1.6 MB/s eta 0:00:02
|                                        | 5.4 MB 1.6 MB/s eta 0:00:02
|                                        | 5.4 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.5 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.6 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.7 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
|                                        | 5.8 MB 1.6 MB/s eta 0:00:02
```

```
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 5.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.0 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.1 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.2 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.3 MB 1.6 MB/s eta 0:00:02
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
```

```
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
|                                        | 6.4 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.5 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.6 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.7 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.8 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 6.9 MB 1.6 MB/s eta 0:00:02
|                                        | 7.0 MB 1.6 MB/s eta 0:00:02
|                                        | 7.0 MB 1.6 MB/s eta 0:00:02
|                                        | 7.0 MB 1.6 MB/s eta 0:00:02
|                                        | 7.0 MB 1.6 MB/s eta 0:00:01
|                                        | 7.0 MB 1.6 MB/s eta 0:00:01
```

```
|                                        | 7.0 MB 1.6 MB/s eta 0:00:01
|                                        | 7.0 MB 1.6 MB/s eta 0:00:01
|                                        | 7.0 MB 1.6 MB/s eta 0:00:01
|                                        | 7.0 MB 1.6 MB/s eta 0:00:01
|                                        | 7.0 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.1 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.2 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.3 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.4 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.5 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
```

```
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.6 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.7 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
|                                        | 7.8 MB 1.6 MB/s eta 0:00:01
```

```
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.8 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 7.9 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.0 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.1 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.2 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.3 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
|━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
```

```
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.4 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s eta 0:00:01
     |━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━| 8.5 MB 1.6 MB/s
ERROR: Could not find a version that satisfies the requirement ipython==8.2
2.2
ERROR: No matching distribution found for ipython==8.22.2
```

In [2]:
```python
import os
import mailparser
import re
import pandas as pd
```

Как выглядит типичное письмо из данных.

In [3]:
```python
with open("enron_mail_20150507/maildir/delainey-d/sent/1.") as f:
    print(f.read())
```

Message-ID: <4591008.1075854462117.JavaMail.evans@thyme>
Date: Tue, 12 Dec 2000 12:19:00 -0800 (PST)
From: david.delainey@enron.com
To: janet.dietrich@enron.com, christopher.calger@enron.com,
brian.redmond@enron.com, max.yzaguirre@enron.com,
rob.milnthorp@enron.com
Subject: Re: MEH Turbines
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: David W Delainey
X-To: Janet R Dietrich, Christopher F Calger, Brian Redmond, Max Yzaguirre,
Rob Milnthorp
X-cc:
X-bcc:
X-Folder: \David_Delainey_Dec2000\Notes Folders\Sent
X-Origin: Delainey-D
X-FileName: ddelain.nsf

Guys, here are the details on the ESA MEH turbines - please forward to me your project details, economics and strategy ASAP. We will allocate these turbines to the project with the best strategic/economic rationale once confirming that they are worth more in ENA versus ESA. If you need more detailed information on these turbines please let me know.

Rob, please give me call on this one.

Regards
Delainey
---------- Forwarded by David W Delainey/HOU/ECT on 12/12/2000
08:17 PM ----------

Brett R Wiggs@ENRON
12/11/2000 11:54 AM
To: David W Delainey/HOU/ECT@ECT
cc: Joe Kishkill/SA/Enron@Enron
Subject: Re: MEH Turbines

Dave:

The attached sheet provides a summary of information on the turbines.

Brett

David W Delainey@ECT
12/08/2000 06:38 PM
To: Brett R Wiggs/SA/Enron@Enron
cc: Joe Kishkill/SA/Enron@Enron
Subject: MEH Turbines

Brett, can you give me the particulars on the four MEH turbines (ie)

```
scheduled delivery, price, fuel options, heat rate, output,etc - including
transferablility to NA.

Regards
Delainey
```

Письма раскиданы по папкам. Берём все папки, в которых встречается слово sent - папки с исходящей корреспонденцией.

In [4]:
```
print(os.listdir("enron_mail_20150507/maildir/delainey-d"))
```

```
['all_documents', 'notes_inbox', 'inbox', 'discussion_threads', 'sent', 'sen
t_items', 'deleted_items', '_sent_mail']
```

In [5]:
```
def parse_emails_by_sender(path="enron_mail_20150507"):
    outgoing_emails_by_user = {}
    unsuccessful = 0

    if not os.path.exists(path):
        return
    total_emails = 0

    for root, dirs, files in os.walk(path):
        # Обрабатываем только отправленные сообщения
        if not ("sent" in root):
            continue

        for file_name in files:
            try:
                msg = mailparser.parse_from_file(os.path.join(root, file_nam
                sender = msg.headers.get("From")
                body = msg.body

                if sender:
                    sender = re.search(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\
                    if sender:
                        sender = sender.group().lower()
                        if sender not in outgoing_emails_by_user:
                            outgoing_emails_by_user[sender] = set()
                        outgoing_emails_by_user[sender].add(body)
                        total_emails += 1
            except Exception:
                unsuccessful += 1

    print(f"Total unsuccessful: {unsuccessful}/{total_emails}")
    return outgoing_emails_by_user


def extract_reply_from_email(text: str, sender_email: str):
    i = text.find("\n\n\n")
    if i != -1:
        text = text[:i]
```

```
    i = text.find("----- Forwarded")
    if i != -1:
        text = text[:i]
    i = text.find("Original Message")
    if i != -1:
        text = text[:i]
    text = re.sub("-{3,50}", "", text)
    name = sender_email.split("@")[0].split(".")
    if name:
        if len(name) > 1:
            r = re.compile(re.escape(name[1]), re.IGNORECASE)
            text = r.sub("", text)
        if len(name[0]) > 2:
            r = re.compile(re.escape(name[0]), re.IGNORECASE)
            text = r.sub("", text)
    text = text.rstrip(" \n")
    text = re.sub(r'(\n\n.{3,50})$', '', text)
    text = re.sub("thanks.?", "", text, flags=re.IGNORECASE)
    text = re.sub(r"(\w{3,10}[,:](\n|\n\n)|^\w{3,10}[,:]|\n\w{3,10}[,:])", "
    return text
```

In [6]:
```
emails_dict = parse_emails_by_sender(path="enron_mail_20150507")
skip_emails = {"no.address@enron.com", "40enron@enron.com"}
senders, emails = [], []
for sender, emails_ in zip(emails_dict.keys(), emails_dict.values()):
    if sender not in skip_emails:
        unique_emails = set()
        for email in emails_:
            email = extract_reply_from_email(email, sender)
            if len(email) > 10 and "Outlook Migration Team" not in email:
                unique_emails.add(email)
        for email in unique_emails:
            senders.append(sender)
            emails.append(email)
df = pd.DataFrame(list(zip(senders, emails)), columns=['sender', 'text'])
df.to_csv("raw_emails.csv")
```

```
Total unsuccessful: 5/126841
```

In [7]:
```
import random
```

In [8]:
```
df = pd.read_csv('raw_emails.csv')
# print(df["sender"].value_counts())  # Люди с максимальным и минимальным чи
d = df.to_dict()
senders = list(d["sender"].values())
texts = list(d["text"].values())
datasets = dict()
for i, sender in enumerate(senders):
    if sender in datasets:
        datasets[sender].append([texts[i], 1])
    else:
        datasets[sender] = [[texts[i], 1]]
numbers_of_owned_emails = {sender: len(datasets[sender]) for sender in datas
for sender in datasets:
    other_texts = []
    for i, text in enumerate(texts):
```

```
        if sender != senders[i]:
            other_texts.append(text)
    for _ in range(numbers_of_owned_emails[sender]):
        datasets[sender].append([random.choice(other_texts), 0])
if not os.path.exists("datasets"):
    os.mkdir("datasets")
for sender in datasets:
    df = pd.DataFrame(list(zip([row[0] for row in datasets[sender]],
                               [row[1] for row in datasets[sender]])),
                      columns=['text', 'label'])
    df.to_csv(os.path.join("datasets", f"{sender.split('@')[0].replace('.',
```

В итоге, получаем папку datasets с датасетом для каждого работника компании, в котором размечены письма (письма от этого человека - 1, другие письма - 0). Другие письма выбираются случайным образом.

In [9]:
```
print(os.listdir("datasets"))
```

['shelley_corman.csv', 'john_buchanan.csv', 'lynn_blair.csv', 'bradley_holme
s.csv', 'toby_kuehl.csv', 'maggie_matheson.csv', 'robert_badeer.csv', 'richa
rd_ring.csv', 'martin_cuilla.csv', 'brad_mckay.csv', 'diana_scholtes.csv',
'greg_whalley.csv', 'liz_taylor.csv', 'michelle_cash.csv', 'twanda_sweet.cs
v', 'carol_coats.csv', 'robert_benson.csv', 'mary_hain.csv', 'lysa_akin.cs
v', 'mark_guzman.csv', 'larry_may.csv', 'kam_keiser.csv', 'm__smith.csv', 'r
osalee_fleming.csv', 'tori_wells.csv', 'kenneth_lay.csv', 'scott_neal.csv',
'kimberly_brown.csv', 'pam_butler.csv', 'marie_heard.csv', 'tamara_black.cs
v', 'kevin_presto.csv', 'm__presto.csv', 'david_forster.csv', 'rogers_herndo
n.csv', 'louise_kitchen.csv', 'debra_bailey.csv', 'madhup_kumar.csv', 'dutch
_quigley.csv', 'monika_causholli.csv', 'laura_vuittonet.csv', 'a__martin.cs
v', 'thomas_martin.csv', 'bill_williams.csv', 'stacy_dickson.csv', 'matthew_
lenhart.csv', 'robin_rodrigue.csv', 'stephanie_panus.csv', 'mark_taylor.cs
v', 'taylor.csv', 'e_taylor.csv', 'albert_meyers.csv', 'geir_solberg.csv',
'david_delainey.csv', 'kay_chapman.csv', 'beverly_stephens.csv', 'w__delaine
y.csv', 'jeff_dasovich.csv', 'joseph_alamo.csv', 'susan_bailey.csv', 'holden
_salisbury.csv', 'darrell_schoolcraft.csv', 'jane_joyce.csv', 'michael_norde
n.csv', 'danny_mccarty.csv', 'drew_fossum.csv', 'martha_benner.csv', 'derek_
r_ahmann.csv', 'benjamin_rogers.csv', 'clint_dean.csv', 'craig_dean.csv', 'm
__forney.csv', 'john_forney.csv', 'tom_donohoe.csv', 'peter_keavey.csv', 'f_
_keavey.csv', 'eric_linder.csv', 'kevin_ruscitti.csv', 'john_griffith.csv',
'eric_saibi.csv', 'joe_stepenovitch.csv', 'sean_crandall.csv', 'jason_wolfe.
csv', 'hunter_shively.csv', 's__shively.csv', 'jim_schwieger.csv', 'doug_gil
bert-smith.csv', 'barry_tycholiz.csv', 'eric_moon.csv', 'chris_germany.csv',
'dkinney.csv', 'steven_merris.csv', 'gerald_nemec.csv', 'sherri_sera.csv',
'joannie_williamson.csv', 'sherri_reinartz.csv', 'katherine_brown.csv', 'jef
f_skilling.csv', 'amelia_alder.csv', 'enron_announcements.csv', 'd__steffes.
csv', 'elizabeth_sager.csv', 'mike_grigsby.csv', 'debra_perlingiere.csv', 'f
rank_ermis.csv', 'reminger.csv', 'ursula_brenner.csv', 'llaidlaw.csv', 'lora
_sullivan.csv', 'katrina_burton.csv', 'terry_west.csv', 'andrew_miles.csv',
'rita_hartfield.csv', 'pges.csv', 'margaret_carson.csv', 'lexi_elliott.csv',
'linda_robertson.csv', 'elizabeth_linnell.csv', 'michael_grimes.csv', 'rob_b
radley.csv', 'mark_schroeder.csv', 'richard_shapiro.csv', 'mark_palmer.csv',
'maureen_raymond.csv', 'maureen_mcvicker.csv', 'sgaines1.csv', 'simon_shih.c
sv', 'ginger_dernehl.csv', 'alhamd_alkhayat.csv', 'miyung_buster.csv', 'lysa
_tracy.csv', 'james_steffes.csv', 'gwyn_koepke.csv', 'janet_butler.csv', 'je
ff_gray.csv', 'sergai_daigre.csv', 'jeffrey_keeler.csv', 'steven_kean.csv',
'j__kean.csv', 'john_lavorato.csv', 'kimberly_hillis.csv', 'angela_mcculloc
h.csv', 'lavorato.csv', 'darin_talley.csv', 'jane_tholt.csv', 'm__tholt.cs
v', 'errol_mclaughlin.csv', 'steven_south.csv', 'barbo.csv', 'dan_hyvl.csv',
'andrew_lewis.csv', 'h__lewis.csv', 'mike_maggi.csv', 'andy_zipper.csv', 'jo
e_quenet.csv', 'vladi_pimenov.csv', 'eric_bass.csv', 'randall_gay.csv', 'rob
_gay.csv', 'l__gay.csv', 'phillip_love.csv', 'm__love.csv', 'tana_jones.cs
v', 'sandra_brawner.csv', 'f__brawner.csv', 'mark_fisher.csv', 'mary_fische
r.csv', 'mark_haedicke.csv', 'janette_elbertson.csv', 'sylvia_sauseda.csv',
'carol_kincannon.csv', 'e__haedicke.csv', 'mark_e_haedicke.csv', 'matt_smit
h.csv', 'bill_rapp.csv', 'teb_lokey.csv', 'sheila_nacey.csv', 'gaby_adamek.c
sv', 'd__martin.csv', 'judy_hernandez.csv', 'juan_hernandez.csv', 'phillip_p
latter.csv', 'sevil_yaman.csv', 'grant_masson.csv', 'vasant_shanbhogue.csv',
'stinson_gibner.csv', 'anita_dupont.csv', 'shirley_crenshaw.csv', 'vince_kam
inski.csv', 'pinnamaneni_krishnarao.csv', 'vkaminski.csv', 'alla_gil.csv',
'rob_wilson.csv', 'j_kaminski.csv', 'kaminski.csv', 'j__kaminski.csv', 'coop
er_richey.csv', 'fletcher_sturm.csv', 'j__sturm.csv', 'casey_evans.csv', 'je
ff_king.csv', 'jeffrey_hodge.csv', 'jenny_helton.csv', 't__hodge.csv', 'john
_hodge.csv', 'suzanne_adams.csv', 'cheryl_johnson.csv', 'jay_reitmeyer.csv',
'daren_farmer.csv', 'j__farmer.csv', 'd__thomas.csv', 'paul_thomas.csv', 'su

```
san_pereira.csv', 'w__pereira.csv', 'click_home.csv', 'rick_buy.csv', 'karen
_heathman.csv', 'kate_symes.csv', 'rod_hayslett.csv', 'a__howard.csv', 'trac
y_geaccone.csv', 'scott_hendrickson.csv', 'susan_scott.csv', 'm__scott.csv',
'darron_giron.csv', 'c__giron.csv', 'elena_chilkina.csv', 'phillip_allen.cs
v', 'ina_rangel.csv', 'k__allen.csv', 'kevin_hyatt.csv', 'ryan_slinger.csv',
'sara_shackleton.csv', 'kaye_ellis.csv', 'theresa_staab.csv', 'lindy_donoho.
csv', 'sally_beck.csv', 'patti_thompson.csv', 'gretel_smith.csv', 'cara_semp
erger.csv', 'skennedy.csv', 'suresh_raghavan.csv', 'harry_arora.csv', 'judy_
townsend.csv', 'richard_sanders.csv', 'b__sanders.csv', 'lisa_gang.csv', 'je
ffrey_shankman.csv', 'a__shankman.csv', 'john_zufferli.csv', 'mark_whitt.cs
v', 'james_derrick.csv', 'j_harris.csv', 'zionette_vincent.csv', 'stacey_whi
te.csv', 'w__white.csv', 'c__gossett.csv', 'scotty_gilbert.csv', 'greg_couc
h.csv', 'sonia_hennessy.csv', 'torrey_moorer.csv', 'terry_sanvido.csv', 'car
l_tricoli.csv', 'stanley_horton.csv', 'cindy_stark.csv', 'horton.csv', 'don_
baughman.csv', 'joe_parks.csv', 'keith_holst.csv', 'mike_carson.csv', 'kay_m
ann.csv', 'patrice_mims.csv', 'l__mims.csv', 'matt_motley.csv', 'dana_davis.
csv', 'carol_clair.csv', 't__lucci.csv', 'mike_swerzbin.csv', 'andrea_ring.c
sv', 'john_arnold.csv', 'melissa_dozier.csv', 'monique_sanchez.csv', 'geoff_
storey.csv', 'chris_dorland.csv', 'liz_prior.csv', 'kim_ward.csv', 'ward.cs
v', 's__ward.csv', 'brenda_whitehead.csv', 'michelle_lokay.csv', 'michele_lo
kay.csv', 'v_weldon.csv', 'charles_weldon.csv', 'chris_stokley.csv', 'daniel
_kang.csv', 'paige_grumulaitis.csv', 'kimberly_watson.csv', 'jonathan_mckay.
csv', 'peggy_hedstrom.csv', 'jason_williams.csv', 'williams.csv', 'mark_koen
ig.csv', 'julie_ferrara.csv', 'jason_rios.csv', 'cathy_phillips.csv', 'jonat
hon_pielop.csv', 'rick_bergsieker.csv', 'brenda_johnston.csv', 'mike_mcconne
ll.csv', 'mark_mcconnell.csv', 'tori_kuykendall.csv', 'larry_campbell.csv',
'f__campbell.csv', 'rjbaker.csv']
```

Самые распространенные слова в сообщениях.

In [10]:
```
from collections import Counter

words_stat = dict()
for dataset in os.listdir("datasets"):
    df = pd.read_csv(f"datasets/{dataset}")
    for text in df["text"].values:
        for pair in Counter(text.replace('\n', '').split()).most_common(100)
            if pair[0] in words_stat:
                words_stat[pair[0]] += pair[1]
            else:
                words_stat[pair[0]] = pair[1]
print(sorted(words_stat.items(), key=lambda x:x[1], reverse=True)[:100])
```

```
[('the', 269281), ('to', 217786), ('I', 133642), ('and', 124292), ('a', 1071
52), ('of', 97071), ('you', 94981), ('for', 81231), ('is', 78149), ('in', 76
594), ('that', 68622), ('on', 60117), ('have', 55830), ('be', 53226), ('wit
h', 50363), ('this', 46610), ('we', 45029), ('will', 42772), ('are', 37346),
('me', 36049), ('it', 35508), ('at', 31987), ('not', 26700), ('your', 2626
9), ('as', 25740), ('if', 25400), ('my', 23742), ('from', 23649), ('would',
23612), ('was', 22339), ('or', 21346), ('know', 21263), ('get', 20389), ('ca
n', 20177), ('but', 19325), ('-', 19316), ('The', 17538), ('about', 16808),
('our', 16387), ('need', 15880), ('any', 15765), ('an', 15721), ('am', 1531
9), ('do', 15077), ('so', 15046), ('by', 14782), ('has', 14777), ('out', 147
10), ('Please', 14612), ('all', 14267), ('up', 14098), ('like', 13633), ('W
e', 13493), ('what', 12546), ('should', 12458), ('they', 12402), ('think', 1
2336), ('just', 12076), ('Enron', 11652), ('call', 11560), ("I'm", 11431),
('i', 11287), ('going', 11234), ('some', 11139), ('he', 10759), ('been', 107
31), ('let', 10647), ('If', 9650), ('please', 9471), ('there', 9269), ("do
n't", 9223), ('had', 9097), ('want', 8963), ('one', 8832), ('when', 8823),
('see', 8803), ('time', 8784), ('make', 7980), ('which', 7879), ('back', 765
7), ('them', 7558), ('more', 7354), ('these', 7274), ('send', 7197), ('wer
e', 7165), ("I'll", 7147), ('could', 7136), ('good', 7052), ('go', 7020),
('This', 7020), ('may', 7000), ('new', 6953), ('other', 6702), ('last', 659
3), ('It', 6568), ('deal', 6366), ('next', 6351), ('their', 6269), ('work',
6258), ('how', 6137)]
```

Выбор датасетов для обучения

Выберем 5 человек, у которых больше всего исходящих писем и отсутствуют подписи к письмам.

In [11]:
```
chosen_datasets = ['kate_symes.csv', 'sally_beck.csv', 'carol_clair.csv', 'm
```

In [12]:
```
for dataset in chosen_datasets:
    print(pd.read_csv(f"datasets/{dataset}"))
```

```
      Unnamed: 0                                               text  label
0              0  I'm a little confused - 559066 is a Prebon dea...      1
1              1  Prebon is right on this. Both deals are 25 MW....      1
2              2  Hey there stranger!\n\nfor the pictures - they...      1
3              3  Two of these should have fees; two should not....      1
4              4  Mark's changing deal 581615 to APB - he had it...      1
...          ...                                                ...    ...
2529        2529   pending the sale of the Wilton Centre unit, I...      0
2530        2530  Same to you!  \nAnd I hope you and your family...      0
2531        2531  \n I would appreciate your help in locating fi...      0
2532        2532  \t2- SURVEY/INFORMATION EMAIL - 7/19/01\n\nCur...      0
2533        2533    see if this works!  If it does, see "13."  w...      0

[2534 rows x 3 columns]
      Unnamed: 0                                               text  label
0              0  I won't sent this to you daily, but this is wh...      1
1              1   \n\nWhen I tried to approve this request (She...      1
2              2  As discussed - happy reading on the plane...\n...      1
3              3  I will be in London the week of January 15th. ...      1
4              4  I have completed internal feedback forms throu...      1
...          ...                                                ...    ...
2815        2815  glad you were succesful, i am not, getting run...      0
2816        2816  I believe our new Tennessee rep is Sherry Glaz...      0
2817        2817   \n\nI found a copy of the confirmation letter...      0
2818        2818  Deal NN6788.1 with Citibank has been killed.  ...      0
2819        2819  It's really easy to assume that companies are ...      0

[2820 rows x 3 columns]
      Unnamed: 0                                               text  label
0              0  This is the e-mail where DF objected to my pro...      1
1              1  Attached is a draft of 2 minor amendments that...      1
2              2  Sara and I would like to meet with you sometim...      1
3              3  suzanne;\nchocolate works!\n\n St. \nEB 3892\n...      1
4              4  There are a number of factors that determine p...      1
...          ...                                                ...    ...
2229        2229  I still need the info I requested to comment i...      0
2230        2230                                       Absolutely.      0
2231        2231  for your help.  Due to my lack of a complete u...      0
2232        2232  Brent Price, Bob Hall, Leslie Reeves and I wou...      0
2233        2233  You should try looking at Duke,  they seem to ...      0

[2234 rows x 3 columns]
      Unnamed: 0                                               text  label
0              0  \nI will defer to your expertise on whether we...      1
1              1  \nso much for extending your business trip to ...      1
2              2   we could set up an LLC entity to employ those...      1
3              3   I just called our counsel on this matter this...      1
4              4  I am working with Mary Joyce on that right now...      1
...          ...                                                ...    ...
2021        2021  I have lunch plans tomorrow (and I'm not sorry...      0
2022        2022  micha will pick us up at my place saturday @ 1...      0
2023        2023  try having no work to do for two months and th...      0
2024        2024              i think i am going to go to the game.      0
2025        2025  I was not aware that the market closed at 1:00...      0

[2026 rows x 3 columns]
```

```
      Unnamed: 0                                                text  label
0              0   Kathryn (EES Logistics) is going to send me s...      1
1              1  This doesn't have everything on it AND i thoug...      1
2              2  I faxed a copy of the PSNC invoice for March t...      1
3              3   not a bad idea.\n\n_61@hotmail.com\t\n2701 Re...      1
4              4  Should I call you up and sing !?!?!!?!?!?!?!  ...      1
...          ...                                                ...    ...
5929        5929  For your information. please find attached the...      0
5930        5930  Sorry.  Don't work.  I can do 1:45 California ...      0
5931        5931  \nDo you know anything about the Congress Plaz...      0
5932        5932  Please email me whatever version of the doc yo...      0
5933        5933                            I have no such studies.      0

[5934 rows x 3 columns]
```

Обучение моделей

In [13]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pickle

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score, prec
precision_recall_curve, ConfusionMatrixDisplay
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer

from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
```

Будем использовать 2 способа векторизации текстов - TF-IDF и CountVectorizer

In [14]:
```
def tfidf(df: pd.DataFrame):
    tfidf = TfidfVectorizer()
    return np.array(tfidf.fit_transform(df['text']).todense())


def vectorize(df: pd.DataFrame):
    count_vec = CountVectorizer()
    return np.array(count_vec.fit_transform(df['text']).todense())
```

Определяем класс для обучения и тестирования моделей. Обучающая выборка - 70% датасета. to_vec_func - выбранный способ векторизации.

In [15]:
```
class ModelTrainer:

    def __init__(self, csv_path, train_dir_path):
```

```
        self.dataset = pd.read_csv(os.path.join("datasets", csv_path))
        self.dataset_name = csv_path.split(".")[0]
        self.train_info = {}
        self.train_dir_path = train_dir_path

    def update_train_info(self, info, model_name, vec_name):
        if model_name in self.train_info:
            self.train_info[model_name][vec_name] = info
        else:
            self.train_info[model_name] = {vec_name: info}

    def save_train_info_into_pickle(self):
        if not os.path.exists(self.train_dir_path):
            os.mkdir(self.train_dir_path)
        with open(os.path.join(self.train_dir_path, f"{self.dataset_name}.pi
            pickle.dump(self.train_info, f)

    def count_acc(self, pred_scores, scores):
        correct = 0
        res_acc = []
        pred_scores = list(pred_scores)
        scores = list(scores)
        for i in range(len(scores)):
            if pred_scores[i] == scores[i]:
                correct += 1
            res_acc.append(correct / (i + 1))
        return res_acc

    def plot_accuracy(self, train_pred_scores, train_scores, test_pred_score
        plt.figure(figsize=(15, 8))
        plt.title('Accuracy')
        plt.xticks([])
        plt.plot(self.count_acc(train_pred_scores, train_scores), color='r',
        plt.plot(self.count_acc(test_pred_scores, test_scores), color='b', l
        plt.legend()
        plt.show()

    def plot_precision_recall_curve(self, y_test, y_pred):
        precision, recall, thresholds = precision_recall_curve(y_test, y_pre
        plt.figure(figsize=(15, 8))
        plt.xlabel('Recall')
        plt.ylabel('Presicion')
        plt.title('Precision-recall curve')
        plt.xticks(np.arange(0, 1.1, 0.1))
        plt.yticks(np.arange(0, 1.1, 0.1))
        plt.grid()
        plt.plot(recall, precision)
        plt.show()

    def plot_confusion_matrix(self, conf_matrix):
        cm = ConfusionMatrixDisplay(conf_matrix)
        cm.plot()
        plt.show()

    def train(self, model, to_vec_func):
        model_name = model.__class__.__name__
```

```
        X = to_vec_func(self.dataset)
        y = self.dataset["label"]
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=
        model = model.fit(X_train, y_train)
        y_train_pred = model.predict(X_train)
        y_pred = model.predict(X_test)
        self.plot_accuracy(y_train_pred, y_train, y_pred, y_test)
        self.plot_precision_recall_curve(y_test, y_pred)
        conf_matrix = confusion_matrix(y_test, y_pred)
        self.plot_confusion_matrix(conf_matrix)
        self.update_train_info({
            'Accuracy': accuracy_score(y_test, y_pred),
            'Precision': precision_score(y_test, y_pred),
            'Recall': recall_score(y_test, y_pred),
            'F1 score': f1_score(y_test, y_pred, average="macro"),
            'Confusion matrix': conf_matrix
        }, model_name, to_vec_func.__name__)
```

In [16]:

```
models = [MultinomialNB, LogisticRegression, SVC, KNeighborsClassifier, Deci
for dataset in chosen_datasets:
    model_trainer = ModelTrainer(dataset, train_dir_path="train")
    print(f"Start processing {dataset}")
    for i, model in enumerate(models):
        model_trainer.train(model(), vectorize)
        model_trainer.train(model(), tfidf)
        print(f"Processed model {i + 1}/{len(models)}")
    model_trainer.save_train_info_into_pickle()
```

```
Start processing kate_symes.csv
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

242
78
8
306
True label
Predicted label
0
1
0
1
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 1/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

305
15
43
271
True label
Predicted label
0
1
0
1
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 2/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

311
9
75
239
True label
Predicted label
0
1
0
1
300
250
200
150
100
50

Accuracy
1.00
0.95
0.90
0.85
0.80
Train
Test

Precision-recall curve
1.0
0.9
0.8
0.7
0.6
0.5
Presicion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

```
Processed model 3/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

216
104
66
248
True label
Predicted label
0
1
0
1
225
200
175
150
125
100
75

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 4/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

270
50
60
254
True label
Predicted label
0
1
0
1
250
225
200
175
150
125
100
75
50

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 5/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

290
30
42
272
True label
Predicted label
0
1
0
1
250
200
150
100
50

Accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

```
Processed model 6/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

300
20
69
245
True label
Predicted label
0
1
0
1
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 7/7
Start processing sally_beck.csv
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

275
77
11
342
True label
Predicted label
0
1
0
1
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 1/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

True label
Predicted label
0
1
0
1
324
28
54
299
300
250
200
150
100
50

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 2/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

307
45
82
271
True label
Predicted label
0
1
0
1
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.98
0.96
0.94
0.92
0.90
0.88

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 3/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

247
105
96
257
True label
Predicted label
0
1
0
1
240
220
200
180
160
140
120
100

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 4/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

252
100
89
264
True label
Predicted label
0
1
0
1
260
240
220
200
180
160
140
120
100

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 5/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

280
72
57
296
True label
Predicted label
0
1
0
1
250
200
150
100

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 6/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

299
53
65
288
True label
Predicted label
0
1
0
1
250
200
150
100

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 7/7
Start processing carol_clair.csv
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

238
47
17
257
True label
Predicted label
0
1
0
1
250
200
150
100
50

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 1/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

252
33
44
230
True label
Predicted label
0
1
0
1
250
225
200
175
150
125
100
75
50

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 2/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

259
26
77
197
True label
Predicted label
0
1
0
1
250
200
150
100
50

Accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

```
Processed model 3/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

202
83
62
212
True label
Predicted label
0
1
0
1
200
180
160
140
120
100
80

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 4/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

220
65
62
212
True label
Predicted label
0
1
0
1
220
200
180
160
140
120
100
80

Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

```
Processed model 5/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

254
31
59
215
True label
Predicted label
0
1
0
1
250
225
200
175
150
125
100
75
50

Accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

```
Processed model 6/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

262
23
72
202
True label
Predicted label
0
1
0
1
250
200
150
100
50

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 7/7
Start processing michelle_cash.csv
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

179
66
29
233
True label
Predicted label
0
1
0
1
225
200
175
150
125
100
75
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 1/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

193
52
62
200
True label
Predicted label
0
1
0
1
200
180
160
140
120
100
80
60

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 2/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

213
32
104
158
True label
Predicted label
0
1
0
1
200
175
150
125
100
75
50

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 3/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

143
102
88
174
True label
Predicted label
0
1
0
1
170
160
150
140
130
120
110
100
90

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 4/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

182
63
98
164
True label
Predicted label
0
1
0
1
180
160
140
120
100
80

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 5/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

211
34
81
181
True label
Predicted label
0
1
0
1
200
180
160
140
120
100
80
60
40

Accuracy
1.0
0.9
0.8
0.7
0.6
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 6/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

214
31
98
164
True label
Predicted label
0
1
0
1
200
175
150
125
100
75
50

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 7/7
Start processing chris_germany.csv
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

648
96
134
606
True label
Predicted label
0
1
0
1
600
500
400
300
200
100

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 1/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

612
132
139
601
True label
Predicted label
0
1
0
1
600
500
400
300
200

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 2/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

653
91
226
514
True label
Predicted label
0
1
0
1
600
500
400
300
200
100

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 3/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

534
210
331
409
True label
Predicted label
0
1
0
1
500
450
400
350
300
250

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 4/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

545
199
272
468
True label
Predicted label
0
1
0
1
500
450
400
350
300
250
200

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
Train
Test

Precision-recall curve
1.0
0.9
0.8
0.7
0.6
0.5
Presicion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

```
Processed model 5/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

651
93
204
536
True label
Predicted label
0
1
0
1
600
500
400
300
200
100

Accuracy
1.00
0.95
0.90
0.85
0.80
Train
Test

Precision-recall curve
1.0
0.9
0.8
0.7
0.6
0.5
Presicion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

```
Processed model 6/7
```

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

658
86
300
440
True label
Predicted label
0
1
0
1
600
500
400
300
200
100

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

```
Processed model 7/7
```

In [17]:
```
import copy
```

In [18]:
```
def analyze_train_data(dir_path):
    res_data = {}
    for train_data in os.listdir(dir_path):
        with open(os.path.join(dir_path, train_data), 'rb') as f:
            d = pickle.load(f)
            for name in d:
                if name not in res_data:
                    res_data[name] = {'vectorize': {}, 'tfidf': {}}
                    for vec_name in d[name]:
                        for key in d[name][vec_name]:
                            res_data[name][vec_name][key] = [d[name][vec_nam
                else:
                    for vec_name in d[name]:
                        for key in d[name][vec_name]:
                            res_data[name][vec_name][key].append(d[name][vec
    tmp = copy.deepcopy(res_data)
    max_score = 0
    model_name = ""
    for m in tmp:
        for vec_name in tmp[m]:
            for key in tmp[m][vec_name]:
                if key != "Confusion matrix":
                    res_data[m][vec_name][key] = np.average(res_data[m][vec_
                    if key == "Accuracy" and res_data[m][vec_name][key] > ma
                        max_score = res_data[m][vec_name][key]
                        model_name = m
```

```
                else:
                    res_data[m][vec_name].pop("Confusion matrix")
    print(res_data)
    print(f"Best accuracy: {max_score}")
    print(f"Best model: {model_name}")


analyze_train_data("train")
```

```
{'MultinomialNB': {'vectorize': {'Accuracy': 0.8565354415032548, 'Precisio
n': 0.8202021862163796, 'Recall': 0.9179097840607058, 'F1 score': 0.85554727
84041329}, 'tfidf': {'Accuracy': 0.8088400276837258, 'Precision': 0.75714780
99465017, 'Recall': 0.9313438958634365, 'F1 score': 0.8038208351905329}}, 'L
ogisticRegression': {'vectorize': {'Accuracy': 0.8493985384405761, 'Precisio
n': 0.8700038378656683, 'Recall': 0.8250039639547498, 'F1 score': 0.84926316
01967003}, 'tfidf': {'Accuracy': 0.8632770075957268, 'Precision': 0.89977345
03253921, 'Recall': 0.8196084464144817, 'F1 score': 0.8629643720071687}}, 'S
VC': {'vectorize': {'Accuracy': 0.8042504007652574, 'Precision': 0.877175682
0210038, 'Recall': 0.7090956022300853, 'F1 score': 0.8023340190150989}, 'tfi
df': {'Accuracy': 0.8674466959416517, 'Precision': 0.9183608839699641, 'Reca
ll': 0.807952645593128, 'F1 score': 0.8669815743454972}}, 'KNeighborsClassif
ier': {'vectorize': {'Accuracy': 0.68961086739971, 'Precision': 0.6848624380
838427, 'Recall': 0.7016803421641996, 'F1 score': 0.6886435558704466}, 'tfid
f': {'Accuracy': 0.7821538174722555, 'Precision': 0.746527932174623, 'Recal
l': 0.8645367143864426, 'F1 score': 0.778747803085319}}, 'DecisionTreeClassi
fier': {'vectorize': {'Accuracy': 0.7392564435433862, 'Precision': 0.7500520
274235448, 'Recall': 0.7177803620405638, 'F1 score': 0.738957971700734}, 'tf
idf': {'Accuracy': 0.7515306058641958, 'Precision': 0.7570337439640518, 'Rec
all': 0.741911725167869, 'F1 score': 0.7514837976593478}}, 'RandomForestClas
sifier': {'vectorize': {'Accuracy': 0.8230991180868173, 'Precision': 0.85460
01093214466, 'Recall': 0.7809209004450783, 'F1 score': 0.8225032192740823},
'tfidf': {'Accuracy': 0.8316864440039333, 'Precision': 0.8923747673267952,
'Recall': 0.7552188673055581, 'F1 score': 0.8303715296456406}}, 'GradientBoo
stingClassifier': {'vectorize': {'Accuracy': 0.8015507090616344, 'Precisio
n': 0.8688816803779094, 'Recall': 0.7107787740346186, 'F1 score': 0.79928637
13248755}, 'tfidf': {'Accuracy': 0.8060895705083088, 'Precision': 0.86167151
31095543, 'Recall': 0.7305838077405757, 'F1 score': 0.8045632062296771}}}
Best accuracy: 0.8674466959416517
Best model: SVC
```

Препроцессинг текстов

In [19]:
```
import nltk
from nltk.corpus import stopwords
from string import punctuation

nltk.download("stopwords")

stop_words = stopwords.words("english")
```

```
[nltk_data] Downloading package stopwords to /home/tech/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [20]:
```
def remove_punctuation(text):
    translator = str.maketrans(punctuation, ' ' * len(punctuation))
```

```
    return text.translate(translator)


def text_preprocessing(text):
    text = re.sub(r'https?://\S+|www\.\S+', ' ', text)
    text = re.sub(r'<.*?>', ' ', text)
    text = re.sub(r'{[^>]+}', ' ', text)
    emoji_pattern = re.compile("["
                               u"\U0001F600-\U0001F64F"
                               u"\U0001F300-\U0001F5FF"
                               u"\U0001F680-\U0001F6FF"
                               u"\U0001F1E0-\U0001F1FF"
                               u"\U00002702-\U000027B0"
                               u"\U000024C2-\U0001F251"
                               "]+", flags=re.UNICODE)
    text = emoji_pattern.sub(r'', text)
    text = re.sub(r'\S*@\S*\s?', ' ', text)
    text = re.sub(r'[0-9]+', ' ', text)
    text = re.sub(r'-|«|»', ' ', text)
    text = re.sub(r'[^\w\s]', ' ', text)
    text = re.sub(r'\n', ' ', text)
    text = text.lower()
    text = remove_punctuation(text)
    text = text.strip()
    # text = ' '.join([word for i, word in enumerate(text.split()) if word r
    return text
```

In [21]:

```
def tfidf(df: pd.DataFrame):
    tfidf = TfidfVectorizer()
    return np.array(tfidf.fit_transform([text_preprocessing(t) for t in df['


def vectorize(df: pd.DataFrame):
    count_vec = CountVectorizer()
    return np.array(count_vec.fit_transform([text_preprocessing(t) for t in
```

In [22]:

```
models = [MultinomialNB, LogisticRegression, SVC, KNeighborsClassifier, Deci
for dataset in chosen_datasets:
    model_trainer = ModelTrainer(dataset, "train_preprocessing")
    print(f"Start processing {dataset}")
    for i, model in enumerate(models):
        model_trainer.train(model(), vectorize)
        model_trainer.train(model(), tfidf)
        print(f"Processed model {i + 1}/{len(models)}")
    model_trainer.save_train_info_into_pickle()
```

```
Start processing kate_symes.csv
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

248
72
10
304
0
1
True label
0
1
Predicted label
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70

```
Processed model 1/7
```

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

304
16
42
272
0
1
True label
0
1
Predicted label
300
250
200
150
100
50

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

```
Processed model 2/7
```

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

310
10
72
242
0
1
True label
0
1
Predicted label
300
250
200
150
100
50

Accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
Train
Test

```
Processed model 3/7
```

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

269
51
93
221
0
1
True label
0
1
Predicted label
250
225
200
175
150
125
100
75

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

```
Processed model 4/7
```

Accuracy
1.00
0.95
0.90
0.85
0.80
Train
Test

Precision-recall curve
1.0
0.9
0.8
0.7
0.6
0.5
Presicion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

272
48
71
243
0
1
True label
0
1
Predicted label
250
225
200
175
150
125
100
75
50

Accuracy
1.00
0.95
0.90
0.85
0.80
Train
Test

```
Processed model 5/7
```

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

287
33
50
264
0
1
True label
0
1
Predicted label
250
200
150
100
50

Accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
Train
Test

```
Processed model 6/7
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

300
20
69
245
0
1
True label
0
1
Predicted label
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80

```
Processed model 7/7
Start processing sally_beck.csv
```

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

282
70
11
342
0
1
True label
0
1
Predicted label
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75

```
Processed model 1/7
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

319
33
55
298
0
1
True label
0
1
Predicted label
300
250
200
150
100
50

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800

```
Processed model 2/7
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

306
46
82
271
0
1
True label
0
1
Predicted label
300
250
200
150
100
50

Accuracy
Train
Test
1.00
0.98
0.96
0.94
0.92
0.90
0.88

```
Processed model 3/7
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

259
93
108
245
0
1
True label
0
1
Predicted label
240
220
200
180
160
140
120
100

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

```
Processed model 4/7
```

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

259
93
86
267
0
1
True label
0
1
Predicted label
250
225
200
175
150
125
100

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
Train
Test

```
Processed model 5/7
```

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

280
72
59
294
0
1
True label
0
1
Predicted label
250
200
150
100

Accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
Train
Test

```
Processed model 6/7
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

301
51
66
287
0
1
True label
0
1
Predicted label
300
250
200
150
100

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70

```
Processed model 7/7
Start processing carol_clair.csv
```

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

218
67
12
262
0
1
True label
0
1
Predicted label
250
200
150
100
50

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

```
Processed model 1/7
```

Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

255
30
44
230
0
1
True label
0
1
Predicted label
250
225
200
175
150
125
100
75
50

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

```
Processed model 2/7
```

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

256
29
73
201
0
1
True label
0
1
Predicted label
250
200
150
100
50

Accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
Train
Test

```
Processed model 3/7
```

Accuracy
Train
Test
0.0
0.2
0.4
0.6
0.8

Precision-recall curve
Presicion
Recall
0.5
0.6
0.7
0.8
0.9
1.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

206
79
65
209
0
1
True label
0
1
Predicted label
200
180
160
140
120
100
80

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

```
Processed model 4/7
```

Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

232
53
69
205
0
1
True label
0
1
Predicted label
220
200
180
160
140
120
100
80
60

Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Train
Test

```
Processed model 5/7
```

Accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

245
40
47
227
0
1
True label
0
1
Predicted label
225
200
175
150
125
100
75
50

Accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
Train
Test

```
Processed model 6/7
```

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

259
26
69
205
0
1
True label
0
1
Predicted label
250
200
150
100
50

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

```
Processed model 7/7
Start processing michelle_cash.csv
```

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

182
63
27
235
0
1
True label
0
1
Predicted label
225
200
175
150
125
100
75
50

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75

```
Processed model 1/7
```

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Train
Test

Precision-recall curve
Presicion
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

196
49
63
199
0
1
True label
0
1
Predicted label
180
160
140
120
100
80
60

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6

```
Processed model 2/7
```

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

213
32
101
161
0
1
True label
0
1
Predicted label
200
175
150
125
100
75
50

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Train
Test

```
Processed model 3/7
```

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5
0.4

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

172
73
126
136
0
1
True label
0
1
Predicted label
160
140
120
100
80

Accuracy
Train
Test
1.0
0.8
0.6
0.4
0.2
0.0

```
Processed model 4/7
```

Accuracy
1.0
0.9
0.8
0.7
0.6
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

179
66
89
173
0
1
True label
0
1
Predicted label
160
140
120
100
80

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Train
Test

```
Processed model 5/7
```

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Train
Test

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

210
35
77
185
0
1
True label
0
1
Predicted label
200
180
160
140
120
100
80
60
40

Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Train
Test

```
Processed model 6/7
```

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5
0.4

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

213
32
99
163
0
1
True label
0
1
Predicted label
200
175
150
125
100
75
50

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

```
Processed model 7/7
Start processing chris_germany.csv
```

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

632
112
124
616
0
1
True label
0
1
Predicted label
600
500
400
300
200

Accuracy
Train
Test
1.000
0.975
0.950
0.925
0.900
0.875
0.850

```
Processed model 1/7
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

607
137
135
605
0
1
True label
0
1
Predicted label
600
500
400
300
200

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80

```
Processed model 2/7
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

645
99
224
516
0
1
True label
0
1
Predicted label
600
500
400
300
200
100

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80

```
Processed model 3/7
```

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

441
303
262
478
0
1
True label
0
1
Predicted label
475
450
425
400
375
350
325
300
275

Accuracy
Train
Test
1.0
0.9
0.8
0.7
0.6
0.5

```
Processed model 4/7
```

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
Train
Test

Precision-recall curve
1.0
0.9
0.8
0.7
0.6
0.5
Presicion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

533
211
236
504
0
1
True label
0
1
Predicted label
500
450
400
350
300
250

Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Train
Test

```
Processed model 5/7
```

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
Train
Test

Precision-recall curve
1.0
0.9
0.8
0.7
0.6
0.5
Presicion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall

618
126
172
568
True label
Predicted label
0
1
0
1
600
500
400
300
200

Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
Train
Test

```
Processed model 6/7
```

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70

Precision-recall curve
Presicion
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

654
90
299
441
0
1
True label
0
1
Predicted label
600
500
400
300
200
100

Accuracy
Train
Test
1.00
0.95
0.90
0.85
0.80
0.75
0.70

```
Processed model 7/7
```

In [23]:
```
analyze_train_data("train_preprocessing")
```

```
{'MultinomialNB': {'vectorize': {'Accuracy': 0.8555801217129735, 'Precisio
n': 0.8139409518887663, 'Recall': 0.9245149540068385, 'F1 score': 0.85463690
45723674}, 'tfidf': {'Accuracy': 0.805711771819829, 'Precision': 0.749765608
9951071, 'Recall': 0.9368422856009818, 'F1 score': 0.8004417045882993}}, 'Lo
gisticRegression': {'vectorize': {'Accuracy': 0.8494239397860139, 'Precisio
n': 0.8694290360207562, 'Recall': 0.825392056694404, 'F1 score': 0.849300364
7608818}, 'tfidf': {'Accuracy': 0.8639362147507985, 'Precision': 0.898378608
6857561, 'Recall': 0.8223142343497493, 'F1 score': 0.8636745221843342}}, 'SV
C': {'vectorize': {'Accuracy': 0.8053302160173595, 'Precision': 0.8724682750
273394, 'Recall': 0.7167567551611785, 'F1 score': 0.8036935151103324}, 'tfid
f': {'Accuracy': 0.8656147502400435, 'Precision': 0.9164025699060454, 'Recal
l': 0.8057804454415356, 'F1 score': 0.8651398228199018}}, 'KNeighborsClassif
ier': {'vectorize': {'Accuracy': 0.6913857446916231, 'Precision': 0.70516001
4054423, 'Recall': 0.6651352571183408, 'F1 score': 0.6906547256723178}, 'tfi
df': {'Accuracy': 0.5301185899508135, 'Precision': 0.9387420743277641, 'Reca
ll': 0.06396262731005695, 'F1 score': 0.39887571528408194}}, 'DecisionTreeCl
assifier': {'vectorize': {'Accuracy': 0.7466444802754547, 'Precision': 0.760
0072667498627, 'Recall': 0.7239641790139617, 'F1 score': 0.746425585058048
7}, 'tfidf': {'Accuracy': 0.7368277600718532, 'Precision': 0.733559565610533
8, 'Recall': 0.7463206720892861, 'F1 score': 0.7367496892587942}}, 'RandomFo
restClassifier': {'vectorize': {'Accuracy': 0.8211837169771178, 'Precision':
0.840341547654951, 'Recall': 0.7951534224186275, 'F1 score': 0.8209489143055
75}, 'tfidf': {'Accuracy': 0.8422595162914532, 'Precision': 0.88970723838672
54, 'Recall': 0.7817952534054547, 'F1 score': 0.8415195730159045}}, 'Gradien
tBoostingClassifier': {'vectorize': {'Accuracy': 0.8006411297013247, 'Precis
ion': 0.8654985051683781, 'Recall': 0.7119088943102003, 'F1 score': 0.798484
6950365808}, 'tfidf': {'Accuracy': 0.8042680342174521, 'Precision': 0.857290
3625638177, 'Recall': 0.7313207765968069, 'F1 score': 0.8028806431140139}}}
Best accuracy: 0.8656147502400435
Best model: SVC
```

Исследование препроцессинга на лучшей модели SVM

In [24]:
```
# Raw 'Accuracy': 0.864054582489543
# Preprocessing without lemmatization 'Accuracy': 0.8566528760292869
# Save stop words + without lemmatization 0.8605192309990312
# Lemmatization + save stop words 0.8605192309990312
# Only Lower text 0.8605459869746198
```

В итоге, лучше всего модель работает на обычном тексте.

Нейросети

In [25]:
```
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, SimpleRNN, Dense

tf.config.list_physical_devices('GPU')
```

```
2024-04-04 14:50:08.442048: I tensorflow/core/platform/cpu_feature_guard.cc:
210] This TensorFlow binary is optimized to use available CPU instructions i
n performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild
TensorFlow with the appropriate compiler flags.
2024-04-04 14:50:08.928271: W tensorflow/compiler/tf2tensorrt/utils/py_util
s.cc:38] TF-TRT Warning: Could not find TensorRT
2024-04-04 14:50:09.330644: I external/local_xla/xla/stream_executor/cuda/cu
da_executor.cc:998] successful NUMA node read from SysFS had negative value
(-1), but there must be at least one NUMA node, so returning NUMA node zero.
See more at https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/te
sting/sysfs-bus-pci#L344-L355
2024-04-04 14:50:09.349095: W tensorflow/core/common_runtime/gpu/gpu_device.
cc:2251] Cannot dlopen some GPU libraries. Please make sure the missing libr
aries mentioned above are installed properly if you would like to use GPU. F
ollow the guide at https://www.tensorflow.org/install/gpu for how to downloa
d and setup the required libraries for your platform.
Skipping registering GPU devices...
```

Out[25]: []

In [26]:

```python
def print_metrics(history, epochs=2):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']

    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs_range = range(epochs)

    plt.figure(figsize=(15, 8))
    plt.subplot(1, 3, 1)
    plt.plot(epochs_range, acc, label='Train')
    plt.plot(epochs_range, val_acc, label='Test')
    plt.legend()
    plt.grid()
    plt.title('Accuracy')

    plt.subplot(1, 3, 2)
    plt.plot(epochs_range, loss, label='Train')
    plt.plot(epochs_range, val_loss, label='Test')
    plt.legend()
    plt.grid()
    plt.title('Loss')

    plt.show()


def print_conf_matrix(labels, pred_labels):
    cm = ConfusionMatrixDisplay(confusion_matrix(labels, pred_labels))
    cm.plot()
    plt.show()
```

In [27]:

```python
accuracy = []
precision = []
```

```
recall = []
for dataset in chosen_datasets:
    df = pd.read_csv(os.path.join("datasets", dataset))
    X_train, X_test, y_train, y_test = train_test_split(df["text"].values, c
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(X_train)
    sequences = tokenizer.texts_to_sequences(X_train)
    max_length = max([len(seq) for seq in sequences])
    padded_sequences = pad_sequences(sequences, maxlen=max_length)
    new_sequences = tokenizer.texts_to_sequences(X_test)
    new_padded_sequences = pad_sequences(new_sequences, maxlen=max_length)

    model = Sequential()
    model.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=
    model.add(SimpleRNN(32))
    model.add(Dense(1, activation='sigmoid'))

    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['ac

    epochs = 3
    history = model.fit(padded_sequences, y_train, validation_data=(new_padc
    print_metrics(history, epochs=epochs)

    predictions = model.predict(new_padded_sequences)
    correct = 0
    predicted_labels = []
    for i in range(len(predictions)):
        prediction_label = 1 if predictions[i] > 0.5 else 0
        predicted_labels.append(prediction_label)
        if y_test[i] == prediction_label:
            correct += 1
    print_conf_matrix(y_test, predicted_labels)
    accuracy.append(correct / len(predictions))
    print(f"Accuracy: {correct / len(predictions)}")
    print(f"Precision: {precision_score(y_test, predicted_labels)}")
    precision.append(precision_score(y_test, predicted_labels))
    print(f"Recall: {recall_score(y_test, predicted_labels)}")
    recall.append(recall_score(y_test, predicted_labels))
print(f"Average accuracy: {np.average(accuracy)}")
print(f"Average precision: {np.average(precision)}")
print(f"Average recall: {np.average(recall)}")
```

```
Epoch 1/3
 1/60 ━━━━━━━━━━━━━━━━━━━━ 58s 994ms/step - accuracy: 0.6562 - loss: 0.6730
✓✓✓
```

✓✓✓

✓✓✓

✓✓✓

```
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
 1/20 ━━━━━━━━━━━━━━━━━━━━ 1s 100ms/step
✓✓✓
```

```
Accuracy: 0.8596214511041009
Precision: 0.8617363344051447
Recall: 0.8535031847133758
Epoch 1/3
 1/67 ━━━━━━━━━━━━━━━━━━━━ 1:01 936ms/step - accuracy: 0.4062 - loss: 0.7055
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

✓✓✓

✓✓✓

```
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

```
 1/23 ━━━━━━━━━━━━━━━━━━━━ 1s 88ms/step
✓✓✓
```

```
Accuracy: 0.8411347517730496
Precision: 0.8433048433048433
Recall: 0.8385269121813032
Epoch 1/3
 1/53 ━━━━━━━━━━━━━━━━━━━━ 46s 896ms/step - accuracy: 0.5625 - loss: 0.6767
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
 1/16 ━━━━━━━━━━━━━━━━━━━━ 1s 91ms/step
✓✓✓
```

```
Accuracy: 0.7080867850098619
Precision: 0.7375
Recall: 0.6755725190839694
Epoch 1/3
  1/140 ━━━━━━━━━━━━━━━━━━━━ 2:05 906ms/step - accuracy: 0.3750 - loss: 0.71
49
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

✓✓✓

✓✓✓

✓✓✓

✓✓✓

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
 1/47 ━━━━━━━━━━━━━━━━━━━━ 3s 85ms/step
✓✓✓
```

```
✓✓✓
```

```
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['ac

    epochs = 5
    history = model.fit(padded_sequences, y_train, validation_data=(new_padc
    print_metrics(history, epochs=epochs)

    predictions = model.predict(new_padded_sequences)
    correct = 0
    predicted_labels = []
    for i in range(len(predictions)):
        prediction_label = 1 if predictions[i] > 0.5 else 0
        predicted_labels.append(prediction_label)
        if y_test[i] == prediction_label:
            correct += 1
    print_conf_matrix(y_test, predicted_labels)
    accuracy.append(correct / len(predictions))
    print(f"Accuracy: {correct / len(predictions)}")
    print(f"Precision: {precision_score(y_test, predicted_labels)}")
    precision.append(precision_score(y_test, predicted_labels))
    print(f"Recall: {recall_score(y_test, predicted_labels)}")
    recall.append(recall_score(y_test, predicted_labels))
print(f"Average accuracy: {np.average(accuracy)}")
print(f"Average precision: {np.average(precision)}")
print(f"Average recall: {np.average(recall)}")
```

```
Epoch 1/5
 1/60 ━━━━━━━━━━━━━━━━━━━━ 1:20 1s/step - accuracy: 0.3438 - loss: 0.6944
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
 1/60 ━━━━━━━━━━━━━━━━━━━━ 12s 206ms/step - accuracy: 1.0000 - loss: 0.0631
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

✓✓✓

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
 1/67 ━━━━━━━━━━━━━━━━━━━━ 11s 180ms/step - accuracy: 0.9375 - loss: 0.1780
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

```
 1/23 ━━━━━━━━━━━━━━━━━━━━ 2s 127ms/step
✓✓✓
```

```
Accuracy: 0.8609929078014185
Precision: 0.9411764705882353
Recall: 0.7705382436260623
Epoch 1/5
 1/53 ━━━━━━━━━━━━━━━━━━━━ 53s 1s/step - accuracy: 0.3125 - loss: 0.6945
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
 1/18 ━━━━━━━━━━━━━━━━━━━━ 1s 99ms/step
✓✓✓
```

```
Accuracy: 0.8586762075134168
Precision: 0.8853754940711462
Recall: 0.8175182481751825
Epoch 1/5
 1/48 ━━━━━━━━━━━━━━━━━━━━ 53s 1s/step - accuracy: 0.4688 - loss: 0.6935
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

✓✓✓

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
  1/140 ━━━━━━━━━━━━━━━━━━━━ 19s 139ms/step - accuracy: 0.9375 - loss: 0.148
1
✓✓✓
```

✓✓✓

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

✓✓✓

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
 1/47 ━━━━━━━━━━━━━━━━━━━━ 5s 120ms/step
✓✓✓
```

```
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
27/47 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
29/47 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
31/47 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
33/47 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
35/47 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
37/47 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
39/47 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
41/47 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
43/47 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
45/47 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
47/47 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
47/47 ━━━━━━━━━━━━━━━━━━━━ 2s 35ms/step
```

```
Accuracy: 0.8463611859838275
Precision: 0.8585434173669467
Recall: 0.8283783783783784
Average accuracy: 0.8528584971215846
Average precision: 0.8642070457434622
Average recall: 0.8476473571739612
```

In [30]:
```
from tensorflow.keras.layers import Conv1D, GlobalMaxPooling1D
```

In [31]:
```
accuracy = []
precision = []
recall = []
for dataset in chosen_datasets:
    df = pd.read_csv(os.path.join("datasets", dataset))
    X_train, X_test, y_train, y_test = train_test_split(df["text"].values, c
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(X_train)
    sequences = tokenizer.texts_to_sequences(X_train)
    max_length = max([len(seq) for seq in sequences])
    padded_sequences = pad_sequences(sequences, maxlen=max_length)
    new_sequences = tokenizer.texts_to_sequences(X_test)
    new_padded_sequences = pad_sequences(new_sequences, maxlen=max_length)

    model = Sequential()
    model.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=
    model.add(Conv1D(128, 5, activation='relu'))
    model.add(GlobalMaxPooling1D())
    model.add(Dense(1, activation='sigmoid'))

    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['ac

    epochs = 5
    history = model.fit(padded_sequences, y_train, validation_data=(new_padc
    print_metrics(history, epochs=epochs)

    predictions = model.predict(new_padded_sequences)
    correct = 0
    predicted_labels = []
    for i in range(len(predictions)):
        prediction_label = 1 if predictions[i] > 0.5 else 0
        predicted_labels.append(prediction_label)
        if y_test[i] == prediction_label:
            correct += 1
    print_conf_matrix(y_test, predicted_labels)
    accuracy.append(correct / len(predictions))
    print(f"Accuracy: {correct / len(predictions)}")
    print(f"Precision: {precision_score(y_test, predicted_labels)}")
    precision.append(precision_score(y_test, predicted_labels))
    print(f"Recall: {recall_score(y_test, predicted_labels)}")
    recall.append(recall_score(y_test, predicted_labels))
print(f"Average accuracy: {np.average(accuracy)}")
print(f"Average precision: {np.average(precision)}")
print(f"Average recall: {np.average(recall)}")
```

```
Epoch 1/5
 1/60 ━━━━━━━━━━━━━━━━━━━━ 33s 573ms/step - accuracy: 0.3438 - loss: 0.6993
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
 1/60 ━━━━━━━━━━━━━━━━━━━━ 2s 37ms/step - accuracy: 0.9688 - loss: 0.1455
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
━━━━━━━━━━━━━━━━━━━━
18/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
━━━━━━━━━━━━━━━━━━━━
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
```

```
Accuracy: 0.9006309148264984
Precision: 0.9169435215946844
Recall: 0.8789808917197452
Epoch 1/5
 1/67 ━━━━━━━━━━━━━━━━━━━━ 37s 564ms/step - accuracy: 0.5000 - loss: 0.6938
━━━━━━━━━━━━━━━━━━━━
 3/67 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - accuracy: 0.5365 - loss: 0.6934
━━━━━━━━━━━━━━━━━━━━
 6/67 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - accuracy: 0.5484 - loss: 0.6925
━━━━━━━━━━━━━━━━━━━━
 9/67 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - accuracy: 0.5493 - loss: 0.6917
━━━━━━━━━━━━━━━━━━━━
12/67 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - accuracy: 0.5481 - loss: 0.6911
━━━━━━━━━━━━━━━━━━━━
15/67 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - accuracy: 0.5463 - loss: 0.6907
━━━━━━━━━━━━━━━━━━━━
18/67 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - accuracy: 0.5427 - loss: 0.6905
━━━━━━━━━━━━━━━━━━━━
21/67 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - accuracy: 0.5381 - loss: 0.6905
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
 1/23 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
✓✓✓
```

```
Accuracy: 0.8822695035460993
Precision: 0.8590425531914894
Recall: 0.9150141643059491
Epoch 1/5
 1/53 ━━━━━━━━━━━━━━━━━━━━ 29s 561ms/step - accuracy: 0.3750 - loss: 0.6958
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
 1/18 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
✓✓✓
```

```
Accuracy: 0.855098389982111
Precision: 0.8458781362007168
Recall: 0.8613138686131386
Epoch 1/5
 1/48 ━━━━━━━━━━━━━━━━━━━━ 26s 571ms/step - accuracy: 0.5938 - loss: 0.6790
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
Accuracy: 0.7928994082840237
Precision: 0.807843137254902
Recall: 0.7862595419847328
Epoch 1/5
  1/140 ━━━━━━━━━━━━━━━━━━━━ 1:18 565ms/step - accuracy: 0.3750 - loss: 0.69
68
✓✓✓
```

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

✓✓✓

✓✓✓

```
✓✓✓
```

✓✓✓

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
✓✓✓
```

```
 1/47 ━━━━━━━━━━━━━━━━━━━━ 1s 35ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
14/47 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
26/47 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
39/47 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
47/47 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
47/47 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
```

```
Accuracy: 0.8342318059299192
Precision: 0.8191214470284238
Recall: 0.8567567567567568
Average accuracy: 0.8530260045137303
Average precision: 0.8497657590540433
Average recall: 0.8596650446760645
```

In [32]:
```
from tensorflow.keras.layers import Dense, Embedding, Flatten
```

In [33]:
```
accuracy = []
precision = []
recall = []
for dataset in chosen_datasets:
    df = pd.read_csv(os.path.join("datasets", dataset))
    X_train, X_test, y_train, y_test = train_test_split(df["text"].values, c
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(X_train)
    sequences = tokenizer.texts_to_sequences(X_train)
    max_length = max([len(seq) for seq in sequences])
    padded_sequences = pad_sequences(sequences, maxlen=max_length)
    new_sequences = tokenizer.texts_to_sequences(X_test)
    new_padded_sequences = pad_sequences(new_sequences, maxlen=max_length)

    model = Sequential()
    model.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=
    model.add(Flatten())
    model.add(Dense(1, activation='sigmoid'))

    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['ac
```

```
    epochs = 3
    history = model.fit(padded_sequences, y_train, validation_data=(new_padc
    print_metrics(history, epochs=epochs)

    predictions = model.predict(new_padded_sequences)
    correct = 0
    predicted_labels = []
    for i in range(len(predictions)):
        prediction_label = 1 if predictions[i] > 0.5 else 0
        predicted_labels.append(prediction_label)
        if y_test[i] == prediction_label:
            correct += 1
    print_conf_matrix(y_test, predicted_labels)
    accuracy.append(correct / len(predictions))
    print(f"Accuracy: {correct / len(predictions)}")
    print(f"Precision: {precision_score(y_test, predicted_labels)}")
    precision.append(precision_score(y_test, predicted_labels))
    print(f"Recall: {recall_score(y_test, predicted_labels)}")
    recall.append(recall_score(y_test, predicted_labels))
print(f"Average accuracy: {np.average(accuracy)}")
print(f"Average precision: {np.average(precision)}")
print(f"Average recall: {np.average(recall)}")
```

```
Epoch 1/3
 1/60 ━━━━━━━━━━━━━━━━━━━━ 24s 409ms/step - accuracy: 0.5000 - loss: 0.6940
✓✓✓
```

```
Epoch 3/3
 1/60 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - accuracy: 0.9062 - loss: 0.4488
✓✓✓
```

```
Accuracy: 0.8848580441640379
Precision: 0.9140893470790378
Recall: 0.8471337579617835
Epoch 1/3
 1/67 ━━━━━━━━━━━━━━━━━━━━ 26s 397ms/step - accuracy: 0.6875 - loss: 0.6822
✓✓✓
```

```
✓✓✓
```

```
 1/23 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓
23/23 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
```

```
Accuracy: 0.8680851063829788
Precision: 0.8385416666666666
Recall: 0.9121813031161473
Epoch 1/3
 1/53 ━━━━━━━━━━━━━━━━━━━━ 20s 388ms/step - accuracy: 0.5000 - loss: 0.6934
✓✓✓
```

```
✓✓✓
```

```
Accuracy: 0.815742397137746
Precision: 0.9572192513368984
Recall: 0.6532846715328468
Epoch 1/3
 1/48 ━━━━━━━━━━━━━━━━━━━━ 18s 404ms/step - accuracy: 0.6250 - loss: 0.6904
✓✓✓
```

```
Epoch 3/3
 1/48 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - accuracy: 0.8750 - loss: 0.5963
✓✓✓
```

```
Accuracy: 0.5877712031558185
Precision: 0.9491525423728814
Recall: 0.21374045801526717
Epoch 1/3
  1/140 ━━━━━━━━━━━━━━━━━━━━ 1:24 605ms/step - accuracy: 0.3750 - loss: 0.70
27
━━━━━━━━━━━━━━━━━━━━
 18/140 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4551 - loss: 0.7263
━━━━━━━━━━━━━━━━━━━━
 34/140 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4728 - loss: 0.7192
━━━━━━━━━━━━━━━━━━━━
 51/140 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4815 - loss: 0.7147
━━━━━━━━━━━━━━━━━━━━
 68/140 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4866 - loss: 0.7115
━━━━━━━━━━━━━━━━━━━━
 85/140 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4910 - loss: 0.7090
━━━━━━━━━━━━━━━━━━━━
102/140 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4952 - loss: 0.7071
━━━━━━━━━━━━━━━━━━━━
119/140 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4991 - loss: 0.7054
━━━━━━━━━━━━━━━━━━━━
137/140 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5026 - loss: 0.7039
```

```
✓✓✓
```

```
✓✓✓
```

```
Accuracy: 0.8362533692722371
Precision: 0.8326639892904953
Recall: 0.8405405405405405
Average accuracy: 0.7985420240225636
Average precision: 0.898333359349196
Average recall: 0.693376146233317
```

In [34]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, \
    recall_score, confusion_matrix

import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import TensorDataset, DataLoader, RandomSampler, Weigh

from transformers import AutoTokenizer, AutoModelForSequenceClassification

import os
from tqdm.notebook import tqdm
from collections import defaultdict
from time import time
```

In [35]:

```python
device = 'cuda' if torch.cuda.is_available() else 'cpu'
```

In [36]:

```
class Bert(nn.Module):
    def __init__(self, bert, bert_output_dim, output_dim):
        super().__init__()
        self.bert = bert
        self.relu = nn.ReLU()
        self.fc = nn.Linear(bert_output_dim, output_dim)

    def forward(self, sent_id, mask):
        logits = self.bert(sent_id, mask, return_dict=False)[0]
        outputs = self.relu(logits)
        outputs = self.fc(outputs)
        return outputs


def plot_learning_curves(history):
    sns.set_style(style='whitegrid')
    fig = plt.figure(figsize=(20, 14))

    plt.subplot(2, 2, 1)
    plt.title('Accuracy', fontsize=15)
    plt.plot(history['accuracy']['train'], label='Train')
    plt.plot(history['accuracy']['val'], label='Test')
    plt.xlabel('epoch', fontsize=15)
    plt.legend()

    plt.subplot(2, 2, 2)
    plt.title('Loss', fontsize=15)
    plt.plot(history['loss']['train'], label='Train')
    plt.plot(history['loss']['val'], label='Test')
    plt.xlabel('epoch', fontsize=15)
    plt.legend()

    plt.show()


def train(
    model,
    criterion,
    optimizer,
    train_loader,
    test_loader,
    num_epochs,
    lr_scheduler,
    device='cpu'
):
    history = defaultdict(lambda: defaultdict(list))
    for epoch in tqdm(range(num_epochs), total=num_epochs):
        # Train
        start_time = time()
        model.train(True)
        train_loss = 0
        test_loss = 0
        print(f'epoch: {epoch + 1}/{num_epochs}')
        print('Train')
        preds, labels = np.array([], dtype=int), np.array([], dtype=int)
```

```
        for sent_id, mask, y_batch in train_loader:
            sent_id, mask, y_batch = sent_id.to(device), mask.to(device), y_
            outputs = model(sent_id, mask).logits.squeeze()
            if outputs.size() == torch.Size([]):
                outputs = outputs.unsqueeze(0)
            loss = criterion(outputs, y_batch.float())
            loss.backward()
            optimizer.step()
            optimizer.zero_grad()
            train_loss += np.sum(loss.detach().cpu().numpy())
            y_pred = nn.Sigmoid()(outputs).detach().cpu().numpy() > 0.5
            preds = np.concatenate((preds, y_pred))
            labels = np.concatenate((labels, y_batch.detach().cpu().numpy()))
        train_loss /= len(train_loader)
        train_accuracy = accuracy_score(labels, preds)
        train_precision = precision_score(labels, preds)
        train_recall = recall_score(labels, preds)
        history['loss']['train'].append(train_loss)
        history['accuracy']['train'].append(train_accuracy)
        history['precision']['train'].append(train_precision)
        history['recall']['train'].append(train_recall)
        lr_scheduler.step()
        print('Test')
        model.train(False)
        preds, labels = np.array([], dtype=int), np.array([], dtype=int)
        for sent_id, mask, y_batch in test_loader:
            sent_id, mask, y_batch = sent_id.to(device), mask.to(device), y_
            with torch.no_grad():
                outputs = model(sent_id, mask).logits.squeeze()
                if outputs.size() == torch.Size([]):
                    outputs = outputs.unsqueeze(0)
                loss = criterion(outputs, y_batch.float())
            test_loss += np.sum(loss.detach().cpu().numpy())
            y_pred = nn.Sigmoid()(outputs).detach().cpu().numpy() > 0.5
            preds = np.concatenate((preds, y_pred))
            labels = np.concatenate((labels, y_batch.detach().cpu().numpy()))
        test_loss /= len(test_loader)
        val_accuracy = accuracy_score(labels, preds)
        val_precision = precision_score(labels, preds)
        val_recall = recall_score(labels, preds)
        history['loss']['val'].append(test_loss)
        history['accuracy']['val'].append(val_accuracy)
        history['precision']['val'].append(val_precision)
        history['recall']['val'].append(val_recall)
        print(f'Epoch {epoch + 1} of {num_epochs} took {time() - start_time:
        print(f'  training loss (in-iteration): \t{train_loss:.6f}')
        print(f'  validation loss (in-iteration): \t{test_loss:.6f}')
        print(f'\n  training accuracy: \t\t\t{train_accuracy * 100:.2f} %')
        print(f'  validation accuracy: \t\t\t{val_accuracy * 100:.2f} %')
        print(f'\n  training precision: \t\t\t{train_precision * 100:.2f} %'
        print(f'  validation precision: \t\t{val_precision * 100:.2f} %')
        print(f'\n  training recall: \t\t\t{train_recall * 100:.2f} %')
        print(f'  validation recall: \t\t\t{val_recall * 100:.2f} %')
        if epoch > 0:
            plot_learning_curves(history)
    return model, history
```

```
def predict(model, test_batch_gen, device='cpu'):
    res_outputs = np.array([])
    res_targets = np.array([])
    for batch in test_batch_gen:
        input_ids, attention_mask, target = batch
        input_ids, attention_mask, target = input_ids.to(device), \
                                            attention_mask.to(device), \
                                            target.to(device)
        output = model(input_ids, attention_mask).logits.squeeze()
        output = nn.Sigmoid()(output) > 0.5
        res_outputs = np.append(res_outputs,
                                output.detach().cpu().numpy())
        res_targets = np.append(res_targets,
                                target.view(-1, 1).cpu().numpy())

    print(f'accuracy_score: {accuracy_score(res_targets, res_outputs)}')
    print(f'precision_score: {precision_score(res_targets, res_outputs)}')
    print(f'recall_score: {recall_score(res_targets, res_outputs)}')
    print('confusion matrix')
    print(print_conf_matrix(res_targets, res_outputs))
    return accuracy_score(res_targets, res_outputs), precision_score(res_tar
```

In [37]:

```
accuracy = []
precision = []
recall = []
for dataset in chosen_datasets:
    df = pd.read_csv(os.path.join("datasets", dataset))
    tokenizer = AutoTokenizer.from_pretrained("distilbert/distilbert-base-ur
    bert = AutoModelForSequenceClassification.from_pretrained("distilbert/di
    # model = Bert(bert, bert.classifier.out_features, 1)
    bert.classifier = nn.Linear(in_features=768, out_features=1, bias=True)
    model = bert
    X_train, X_test, y_train, y_test = train_test_split(df["text"].values, c

    max_seq_len = 256
    tokens_train = tokenizer.batch_encode_plus(
        X_train,
        max_length = max_seq_len,
        padding=True,
        truncation=True
    )

    tokens_test = tokenizer.batch_encode_plus(
        X_test,
        max_length = max_seq_len,
        padding=True,
        truncation=True
    )

    train_seq = torch.tensor(tokens_train['input_ids'])
    train_mask = torch.tensor(tokens_train['attention_mask'])
    train_y = torch.tensor(y_train)

    test_seq = torch.tensor(tokens_test['input_ids'])
```

```
    test_mask = torch.tensor(tokens_test['attention_mask'])
    test_y = torch.tensor(y_test)

    batch_size = 16
    train_data = TensorDataset(train_seq, train_mask, train_y)
    train_labels, train_counts = np.unique(y_train, return_counts=True)
    train_class_weights = [sum(train_counts) / c for c in train_counts]
    train_weigths = [train_class_weights[e] for e in y_train]
    train_sampler = WeightedRandomSampler(weights=train_weigths, num_samples
    train_dataloader = DataLoader(train_data, batch_size=batch_size, sampler

    test_data = TensorDataset(test_seq, test_mask, test_y)
    test_sampler = RandomSampler(test_data)
    test_dataloader = DataLoader(test_data, batch_size=batch_size, sampler=t


    output_dim = 1
    lr = 3e-5
    num_epochs = 3

    model = model.to(device)
    optimizer = optim.AdamW(model.parameters(), lr=lr, weight_decay=0.2)
    criterion = nn.BCEWithLogitsLoss()
    lr_scheduler = optim.lr_scheduler.ExponentialLR(optimizer=optimizer,
                                                    gamma=0.9, verbose=True)
    model, history = train(
    model,
    criterion,
    optimizer,
    train_dataloader,
    test_dataloader,
    num_epochs,
    lr_scheduler,
    device
    )
    acc, prec, rec = predict(model, test_dataloader, device)
    accuracy.append(acc)
    precision.append(prec)
    recall.append(rec)
```

```
Some weights of DistilBertForSequenceClassification were not initialized fro
m the model checkpoint at distilbert/distilbert-base-uncased and are newly i
nitialized: ['classifier.bias', 'classifier.weight', 'pre_classifier.bias',
'pre_classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use
it for predictions and inference.
/home/tech/.local/lib/python3.9/site-packages/torch/optim/lr_scheduler.py:2
8: UserWarning: The verbose parameter is deprecated. Please use get_last_lr
() to access the learning rate.
  warnings.warn("The verbose parameter is deprecated. Please use get_last_lr
() "
epoch: 1/3
Train
Test
```

```
Epoch 1 of 3 took 37.691s
  training loss (in-iteration):         0.378731
  validation loss (in-iteration):       0.255282

  training accuracy:                    80.95 %
  validation accuracy:                  89.91 %

  training precision:                   83.04 %
  validation precision:                 89.31 %

  training recall:                      79.86 %
  validation recall:                    90.45 %
epoch: 2/3
Train
Test
Epoch 2 of 3 took 37.818s
  training loss (in-iteration):         0.139737
  validation loss (in-iteration):       0.312449

  training accuracy:                    95.53 %
  validation accuracy:                  86.28 %

  training precision:                   97.05 %
  validation precision:                 80.93 %

  training recall:                      93.87 %
  validation recall:                    94.59 %
```

```
epoch: 3/3
Train
Test
Epoch 3 of 3 took 37.995s
  training loss (in-iteration):         0.066964
  validation loss (in-iteration):       0.346790

  training accuracy:                    97.95 %
  validation accuracy:                  88.96 %

  training precision:                   98.33 %
  validation precision:                 85.88 %

  training recall:                      97.36 %
  validation recall:                    92.99 %
```

```
accuracy_score: 0.889589905362776
precision_score: 0.8588235294117647
recall_score: 0.9299363057324841
confusion matrix
```

```
None
```

```
Some weights of DistilBertForSequenceClassification were not initialized fro
m the model checkpoint at distilbert/distilbert-base-uncased and are newly i
nitialized: ['classifier.bias', 'classifier.weight', 'pre_classifier.bias',
'pre_classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use
it for predictions and inference.
```

```
/home/tech/.local/lib/python3.9/site-packages/torch/optim/lr_scheduler.py:2
8: UserWarning: The verbose parameter is deprecated. Please use get_last_lr
() to access the learning rate.
  warnings.warn("The verbose parameter is deprecated. Please use get_last_lr
() "
```

```
epoch: 1/3
Train
Test
Epoch 1 of 3 took 42.674s
  training loss (in-iteration):         0.378373
  validation loss (in-iteration):       0.267634

  training accuracy:                    82.74 %
  validation accuracy:                  89.08 %

  training precision:                   86.11 %
  validation precision:                 87.70 %

  training recall:                      79.23 %
  validation recall:                    90.93 %
epoch: 2/3
Train
Test
Epoch 2 of 3 took 43.124s
  training loss (in-iteration):         0.150284
  validation loss (in-iteration):       0.223829

  training accuracy:                    94.23 %
  validation accuracy:                  91.49 %

  training precision:                   95.00 %
  validation precision:                 89.07 %

  training recall:                      93.17 %
  validation recall:                    94.62 %
```

```
epoch: 3/3
Train
Test
```

```
Epoch 3 of 3 took 43.237s
  training loss (in-iteration):         0.094092
  validation loss (in-iteration):       0.235162

  training accuracy:                    96.45 %
  validation accuracy:                  92.34 %

  training precision:                   97.08 %
  validation precision:                 89.03 %

  training recall:                      95.68 %
  validation recall:                    96.60 %
```

```
accuracy_score: 0.9234042553191489
precision_score: 0.8903394255874674
recall_score: 0.9660056657223796
confusion matrix
```

```
None
```

```
Some weights of DistilBertForSequenceClassification were not initialized fro
m the model checkpoint at distilbert/distilbert-base-uncased and are newly i
nitialized: ['classifier.bias', 'classifier.weight', 'pre_classifier.bias',
'pre_classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use
it for predictions and inference.
/home/tech/.local/lib/python3.9/site-packages/torch/optim/lr_scheduler.py:2
8: UserWarning: The verbose parameter is deprecated. Please use get_last_lr
() to access the learning rate.
  warnings.warn("The verbose parameter is deprecated. Please use get_last_lr
() "
epoch: 1/3
Train
Test
Epoch 1 of 3 took 34.168s
  training loss (in-iteration):         0.452715
  validation loss (in-iteration):       0.341314

  training accuracy:                    77.43 %
  validation accuracy:                  84.26 %

  training precision:                   75.58 %
  validation precision:                 87.80 %

  training recall:                      81.49 %
  validation recall:                    78.83 %
epoch: 2/3
Train
Test
Epoch 2 of 3 took 34.275s
  training loss (in-iteration):         0.195410
  validation loss (in-iteration):       0.270207

  training accuracy:                    92.42 %
  validation accuracy:                  89.62 %

  training precision:                   93.94 %
  validation precision:                 91.86 %

  training recall:                      90.96 %
  validation recall:                    86.50 %
```

```
epoch: 3/3
Train
Test
Epoch 3 of 3 took 34.272s
  training loss (in-iteration):         0.094524
  validation loss (in-iteration):       0.277183

  training accuracy:                    97.01 %
  validation accuracy:                  89.80 %

  training precision:                   98.12 %
  validation precision:                 93.93 %

  training recall:                      95.72 %
  validation recall:                    84.67 %
```

```
accuracy_score: 0.8980322003577818
precision_score: 0.9392712550607287
recall_score: 0.8467153284671532
confusion matrix
```

```
None
```

```
Some weights of DistilBertForSequenceClassification were not initialized fro
m the model checkpoint at distilbert/distilbert-base-uncased and are newly i
nitialized: ['classifier.bias', 'classifier.weight', 'pre_classifier.bias',
'pre_classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use
it for predictions and inference.
/home/tech/.local/lib/python3.9/site-packages/torch/optim/lr_scheduler.py:2
8: UserWarning: The verbose parameter is deprecated. Please use get_last_lr
() to access the learning rate.
  warnings.warn("The verbose parameter is deprecated. Please use get_last_lr
() "
```

```
epoch: 1/3
Train
Test
Epoch 1 of 3 took 31.017s
  training loss (in-iteration):         0.548258
  validation loss (in-iteration):       0.504199

  training accuracy:                    69.85 %
  validation accuracy:                  76.73 %

  training precision:                   77.69 %
  validation precision:                 92.35 %

  training recall:                      52.99 %
  validation recall:                    59.92 %
epoch: 2/3
Train
Test
```

```
Epoch 2 of 3 took 31.112s
  training loss (in-iteration):         0.238369
  validation loss (in-iteration):       0.344670

  training accuracy:                    91.05 %
  validation accuracy:                  85.60 %

  training precision:                   90.49 %
  validation precision:                 82.25 %

  training recall:                      91.69 %
  validation recall:                    91.98 %
```

```
epoch: 3/3
Train
Test
Epoch 3 of 3 took 31.097s
  training loss (in-iteration):         0.151144
  validation loss (in-iteration):       0.423298

  training accuracy:                    94.73 %
  validation accuracy:                  84.02 %

  training precision:                   94.14 %
  validation precision:                 82.67 %

  training recall:                      96.01 %
  validation recall:                    87.40 %
```

```
accuracy_score: 0.8402366863905325
precision_score: 0.8267148014440433
recall_score: 0.8740458015267175
confusion matrix
```

```
None
```

```
Some weights of DistilBertForSequenceClassification were not initialized fro
m the model checkpoint at distilbert/distilbert-base-uncased and are newly i
nitialized: ['classifier.bias', 'classifier.weight', 'pre_classifier.bias',
'pre_classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use
it for predictions and inference.
/home/tech/.local/lib/python3.9/site-packages/torch/optim/lr_scheduler.py:2
8: UserWarning: The verbose parameter is deprecated. Please use get_last_lr
() to access the learning rate.
  warnings.warn("The verbose parameter is deprecated. Please use get_last_lr
() "
```

```
epoch: 1/3
Train
Test
```

```
Epoch 1 of 3 took 90.763s
  training loss (in-iteration):         0.473939
  validation loss (in-iteration):       0.420007

  training accuracy:                    75.35 %
  validation accuracy:                  81.13 %

  training precision:                   76.22 %
  validation precision:                 83.72 %

  training recall:                      71.30 %
  validation recall:                    77.16 %
epoch: 2/3
Train
Test
Epoch 2 of 3 took 90.873s
  training loss (in-iteration):         0.225971
  validation loss (in-iteration):       0.399078

  training accuracy:                    91.17 %
  validation accuracy:                  83.29 %

  training precision:                   91.58 %
  validation precision:                 85.65 %

  training recall:                      90.89 %
  validation recall:                    79.86 %
```

```
epoch: 3/3
Train
Test
Epoch 3 of 3 took 90.666s
  training loss (in-iteration):         0.141609
  validation loss (in-iteration):       0.475095

  training accuracy:                    95.08 %
  validation accuracy:                  83.09 %

  training precision:                   95.11 %
  validation precision:                 79.00 %

  training recall:                      95.07 %
  validation recall:                    90.00 %
```

```
accuracy_score: 0.8308625336927223
precision_score: 0.7900355871886121
recall_score: 0.9
confusion matrix
```

```
None
```

In [38]:
```
print(f"Average accuracy: {np.average(accuracy)}")
print(f"Average precision: {np.average(precision)}")
print(f"Average recall: {np.average(recall)}")
```

```
Average accuracy: 0.8764251162245923
Average precision: 0.8610369197385233
Average recall: 0.9033406202897469
```