customer-sentiment-prediction

September 18, 2023

0.1 Airline Tweets: Sentiment Analysis & Simple Classification

0.1.1 ISM6564

**Week05

[ ]:

0.2 Introduction

In this notebook we will analysze the sentiment from a dataset on how travelers tweeted about their airline-related feelings, scraped from Twitter in February 2015;.

[ ]:
```
import pandas as pd
import re
from matplotlib import pyplot as plt
import numpy as np

# import tools to pre-process the text data
from sklearn import preprocessing

# Extracting features from text files
#   SciKit Learn includes a number of useful feature extraction classes
#   https://scikit-learn.org/stable/modules/classes.html#module-sklearn.
 ↪feature_extraction.text
#
#   We will use TfidfVectorizer (which includes pre-processing, tokenization,␣
 ↪and filtering out stop words)
from sklearn.feature_extraction.text import TfidfVectorizer

# import tools to reduce the dimensionality of the data
from sklearn.decomposition import TruncatedSVD

# import tools to split the data into training and test sets
from sklearn.model_selection import train_test_split
```

```
# import ML classifiers we will use to model the data
from sklearn.linear_model import SGDClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier

# import tools to evaluate the model performance
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import accuracy_score

from nltk.tokenize import TweetTokenizer

from sklearn.model_selection import RandomizedSearchCV
```

[ ]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

0.3 Dsata Load and Preparation

[ ]:
```
# read in the data
df = pd.read_csv('./data/TweetsWtSentiment.csv')
df.head()
```

[ ]:
```
             tweet_id airline_sentiment  airline_sentiment_confidence  \
0  570306133677760513           neutral                        1.0000
1  570301130888122368          positive                        0.3486
2  570301083672813571           neutral                        0.6837
3  570301031407624196          negative                        1.0000
4  570300817074462722          negative                        1.0000

  negativereason  negativereason_confidence         airline  \
0            NaN                        NaN  Virgin America
1            NaN                     0.0000  Virgin America
2            NaN                        NaN  Virgin America
3     Bad Flight                     0.7033  Virgin America
4     Can't Tell                     1.0000  Virgin America

  airline_sentiment_gold        name negativereason_gold  retweet_count  \
0                    NaN     cairdin                 NaN              0
1                    NaN    jnardino                 NaN              0
2                    NaN  yvonnalynn                 NaN              0
3                    NaN    jnardino                 NaN              0
4                    NaN    jnardino                 NaN              0

                                                text tweet_coord  \
0                @VirginAmerica What @dhepburn said.         NaN
1  @VirginAmerica plus you've added commercials t…         NaN
2  @VirginAmerica I didn't today… Must mean I n…         NaN
```

```
3  @VirginAmerica it's really aggressive to blast…          NaN
4  @VirginAmerica and it's a really big bad thing…          NaN

               tweet_created tweet_location               user_timezone
0  2015-02-24 11:35:52 -0800            NaN  Eastern Time (US & Canada)
1  2015-02-24 11:15:59 -0800            NaN  Pacific Time (US & Canada)
2  2015-02-24 11:15:48 -0800      Lets Play  Central Time (US & Canada)
3  2015-02-24 11:15:36 -0800            NaN  Pacific Time (US & Canada)
4  2015-02-24 11:14:45 -0800            NaN  Pacific Time (US & Canada)
```

[ ]:
```
tokenizer = TweetTokenizer()

def clear_text(text):
    # tokenize the text
    # nltk.download('punkt') # uncomment if you need to download the punkt␣
 ↪package
    tokens = tokenizer.tokenize(text)

    # remove all tokens that are not alphabetic
    tokens = [word for word in tokens if word.isalpha()]

    # make lowercase
    tokens = [word.lower() for word in tokens]

    # remove all tokens that are only one character
    tokens = [word for word in tokens if len(word) > 1]

    return ' '.join(tokens)

df['clean_text']=df.text.apply(lambda x: clear_text(x))
df['hashtags'] = df.text.apply(lambda text: re.findall(r"#(\w+)", text))
df['handles'] = df.text.apply(lambda text: re.findall(r"@(\w+)", text))
```

[ ]:
```
df.head(10)
```

[ ]:
```
             tweet_id airline_sentiment  airline_sentiment_confidence  \
0  570306133677760513           neutral                        1.0000
1  570301130888122368          positive                        0.3486
2  570301083672813571           neutral                        0.6837
3  570301031407624196          negative                        1.0000
4  570300817074462722          negative                        1.0000
5  570300767074181121          negative                        1.0000
6  570300616901320704          positive                        0.6745
7  570300248553349120           neutral                        0.6340
8  570299953286942721          positive                        0.6559
9  570295459631263746          positive                        1.0000
```

```
  negativereason  negativereason_confidence         airline  \
0            NaN                        NaN  Virgin America
1            NaN                     0.0000  Virgin America
2            NaN                        NaN  Virgin America
3     Bad Flight                     0.7033  Virgin America
4     Can't Tell                     1.0000  Virgin America
5     Can't Tell                     0.6842  Virgin America
6            NaN                     0.0000  Virgin America
7            NaN                        NaN  Virgin America
8            NaN                        NaN  Virgin America
9            NaN                        NaN  Virgin America

  airline_sentiment_gold        name negativereason_gold  retweet_count  \
0                    NaN     cairdin                 NaN              0
1                    NaN    jnardino                 NaN              0
2                    NaN  yvonnalynn                 NaN              0
3                    NaN    jnardino                 NaN              0
4                    NaN    jnardino                 NaN              0
5                    NaN    jnardino                 NaN              0
6                    NaN  cjmcginnis                 NaN              0
7                    NaN       pilot                 NaN              0
8                    NaN    dhepburn                 NaN              0
9                    NaN  YupitsTate                 NaN              0

                                                text tweet_coord  \
0                @VirginAmerica What @dhepburn said.         NaN
1  @VirginAmerica plus you've added commercials t…         NaN
2  @VirginAmerica I didn't today… Must mean I n…         NaN
3  @VirginAmerica it's really aggressive to blast…         NaN
4  @VirginAmerica and it's a really big bad thing…         NaN
5  @VirginAmerica seriously would pay $30 a fligh…         NaN
6  @VirginAmerica yes, nearly every time I fly VX…         NaN
7  @VirginAmerica Really missed a prime opportuni…         NaN
8    @virginamerica Well, I didn't…but NOW I DO! :-D         NaN
9  @VirginAmerica it was amazing, and arrived an …         NaN

              tweet_created    tweet_location               user_timezone  \
0  2015-02-24 11:35:52 -0800               NaN  Eastern Time (US & Canada)
1  2015-02-24 11:15:59 -0800               NaN  Pacific Time (US & Canada)
2  2015-02-24 11:15:48 -0800         Lets Play  Central Time (US & Canada)
3  2015-02-24 11:15:36 -0800               NaN  Pacific Time (US & Canada)
4  2015-02-24 11:14:45 -0800               NaN  Pacific Time (US & Canada)
5  2015-02-24 11:14:33 -0800               NaN  Pacific Time (US & Canada)
6  2015-02-24 11:13:57 -0800  San Francisco CA  Pacific Time (US & Canada)
7  2015-02-24 11:12:29 -0800       Los Angeles  Pacific Time (US & Canada)
8  2015-02-24 11:11:19 -0800         San Diego  Pacific Time (US & Canada)
9  2015-02-24 10:53:27 -0800       Los Angeles  Eastern Time (US & Canada)
```

```
                                          clean_text hashtags  \
0                                          what said       []
1     plus added commercials to the experience tacky       []
2          today must mean need to take another trip       []
3  really aggressive to blast obnoxious entertain…       []
4                  and really big bad thing about it       []
5  seriously would pay flight for seats that have…       []
6  yes nearly every time fly vx this ear worm won…       []
7  really missed prime opportunity for men withou…       []
8                                  well but now do       []
9  it was amazing and arrived an hour early too g…       []

                     handles
0  [VirginAmerica, dhepburn]
1            [VirginAmerica]
2            [VirginAmerica]
3            [VirginAmerica]
4            [VirginAmerica]
5            [VirginAmerica]
6            [VirginAmerica]
7            [VirginAmerica]
8            [virginamerica]
9            [VirginAmerica]
```

[ ]:
```
df = df[['clean_text', 'airline', 'airline_sentiment']]
```

[ ]:
```
df.isna().sum()
```

[ ]:
```
clean_text           0
airline              0
airline_sentiment    0
dtype: int64
```

[ ]:
```
df["airline"].unique()
```

[ ]:
```
array(['Virgin America', 'United', 'Southwest', 'Delta', 'US Airways',
       'American'], dtype=object)
```

[ ]:
```
df = pd.get_dummies(
    df,
    prefix_sep='_',
    dummy_na=False,
    drop_first=False,
    columns=['airline'],
    dtype='int32'
)
```

```
df.head(10)
```

[ ]:

```
                                        clean_text airline_sentiment  \
0                                        what said           neutral
1     plus added commercials to the experience tacky          positive
2          today must mean need to take another trip           neutral
3  really aggressive to blast obnoxious entertain…          negative
4                  and really big bad thing about it          negative
5  seriously would pay flight for seats that have…          negative
6  yes nearly every time fly vx this ear worm won…          positive
7  really missed prime opportunity for men withou…           neutral
8                                  well but now do           positive
9  it was amazing and arrived an hour early too g…          positive

   airline_American  airline_Delta  airline_Southwest  airline_US Airways  \
0                 0              0                  0                   0
1                 0              0                  0                   0
2                 0              0                  0                   0
3                 0              0                  0                   0
4                 0              0                  0                   0
5                 0              0                  0                   0
6                 0              0                  0                   0
7                 0              0                  0                   0
8                 0              0                  0                   0
9                 0              0                  0                   0

   airline_United  airline_Virgin America
0               0                       1
1               0                       1
2               0                       1
3               0                       1
4               0                       1
5               0                       1
6               0                       1
7               0                       1
8               0                       1
9               0                       1
```

[ ]:

```
df['airline_sentiment'] = df['airline_sentiment'].apply(lambda x: -1 if x.
↪lower() == 'negative' else (1 if x.lower() == 'positive' else 0))

df.head(10)
```

[ ]:

```
                                        clean_text  airline_sentiment  \
0                                        what said                  0
1     plus added commercials to the experience tacky                  1
```

```
2          today must mean need to take another trip                  0
3  really aggressive to blast obnoxious entertain…                   -1
4                  and really big bad thing about it                  -1
5  seriously would pay flight for seats that have…                   -1
6  yes nearly every time fly vx this ear worm won…                    1
7  really missed prime opportunity for men withou…                    0
8                                  well but now do                    1
9  it was amazing and arrived an hour early too g…                    1

   airline_American  airline_Delta  airline_Southwest  airline_US Airways  \
0                 0              0                  0                   0
1                 0              0                  0                   0
2                 0              0                  0                   0
3                 0              0                  0                   0
4                 0              0                  0                   0
5                 0              0                  0                   0
6                 0              0                  0                   0
7                 0              0                  0                   0
8                 0              0                  0                   0
9                 0              0                  0                   0

   airline_United  airline_Virgin America
0               0                       1
1               0                       1
2               0                       1
3               0                       1
4               0                       1
5               0                       1
6               0                       1
7               0                       1
8               0                       1
9               0                       1
```

[ ]:
```
X = df.drop(['airline_sentiment'], axis=1)
y = df['airline_sentiment']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

[ ]:
```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

[ ]:
```
((10248, 7), (4392, 7), (10248,), (4392,))
```

[ ]:
```
X_train.isna().sum(), X_test.isna().sum()
```

[ ]:
```
(clean_text               0
 airline_American         0
 airline_Delta            0
```

```
airline_Southwest         0
airline_US Airways        0
airline_United            0
airline_Virgin America    0
dtype: int64,
clean_text                0
airline_American          0
airline_Delta             0
airline_Southwest         0
airline_US Airways        0
airline_United            0
airline_Virgin America    0
dtype: int64)
```

[ ]:
```
X_train.head(10)
```

[ ]:
```
                                                clean_text  airline_American  \
10298  more than an hour holding to change flight on …                    0
1688   rather spend two days in the back seat of volk…                    0
6677   am but it says yall are sold out me my coworke…                    0
2934   that may be true however after weeks matter is…                    0
2307   cancelled flighted my flight from iad to jax w…                    0
9695   have the most rude unreliable horrible company…                    0
8347   up with lack coffee on am flight was told some…                    0
4214   flt no gate agent no announcements so unprofes…                    0
4365   appreciate getting the gate issue figured grr …                    0
13429  tisk tisk rude flight attendants yet again han…                    1

       airline_Delta  airline_Southwest  airline_US Airways  airline_United  \
10298              0                  0                   1               0
1688               0                  0                   0               1
6677               0                  1                   0               0
2934               0                  0                   0               1
2307               0                  0                   0               1
9695               0                  0                   1               0
8347               1                  0                   0               0
4214               0                  0                   0               1
4365               0                  1                   0               0
13429              0                  0                   0               0

       airline_Virgin America
10298                       0
1688                        0
6677                        0
2934                        0
2307                        0
9695                        0
```

```
8347                    0
4214                    0
4365                    0
13429                   0
```

[ ]:
```
vectorizer = TfidfVectorizer()
tfidf = vectorizer.fit_transform(X_train['clean_text'])

tfidf_df = pd.DataFrame(tfidf.toarray(), columns=vectorizer.
 ↪get_feature_names_out())

print(tfidf_df.isna().sum())

tfidf_df.head(10)
```

```
aa              0
aaaand          0
aadavantage     0
aadv            0
aadvantage      0
               ..
zone            0
zones           0
zoom            0
zukes           0
zurich          0
Length: 8234, dtype: int64
```

[ ]:

	aa	aaaand	aadavantage	aadv	aadvantage	aal	aaron	ab	aback	\
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

	abandoned	…	zabsonre	zambia	zcc	zero	zip	zone	zones	zoom	zukes	\
0	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

```
7        0.0  …       0.0     0.0  0.0   0.0  0.0   0.0    0.0   0.0    0.0
8        0.0  …       0.0     0.0  0.0   0.0  0.0   0.0    0.0   0.0    0.0
9        0.0  …       0.0     0.0  0.0   0.0  0.0   0.0    0.0   0.0    0.0

   zurich
0     0.0
1     0.0
2     0.0
3     0.0
4     0.0
5     0.0
6     0.0
7     0.0
8     0.0
9     0.0

[10 rows x 8234 columns]
```

[ ]:
```
X_train.shape, tfidf_df.shape
```

[ ]:
```
((10248, 7), (10248, 8234))
```

[ ]:
```
X_train = X_train.reset_index(drop=True) # need to do this, since X_train and␣
 ↪tfidf_df have different indices

X_train = pd.concat([X_train, tfidf_df], axis=1)

X_train.isna().sum()
```

[ ]:
```
clean_text            0
airline_American      0
airline_Delta         0
airline_Southwest     0
airline_US Airways    0
                     ..
zone                  0
zones                 0
zoom                  0
zukes                 0
zurich                0
Length: 8241, dtype: int64
```

[ ]:
```
X_train = X_train.drop('clean_text',axis=1)

X_train.head(10)
```

[ ]:

	airline_American	airline_Delta	airline_Southwest	airline_US Airways	\
0	0	0	0	1	
1	0	0	0	0	
2	0	0	1	0	
3	0	0	0	0	
4	0	0	0	0	
5	0	0	0	1	
6	0	1	0	0	
7	0	0	0	0	
8	0	0	1	0	
9	1	0	0	0	

	airline_United	airline_Virgin America	aa	aaaand	aadavantage	aadv	\
0	0	0	0.0	0.0	0.0	0.0	
1	1	0	0.0	0.0	0.0	0.0	
2	0	0	0.0	0.0	0.0	0.0	
3	1	0	0.0	0.0	0.0	0.0	
4	1	0	0.0	0.0	0.0	0.0	
5	0	0	0.0	0.0	0.0	0.0	
6	0	0	0.0	0.0	0.0	0.0	
7	1	0	0.0	0.0	0.0	0.0	
8	0	0	0.0	0.0	0.0	0.0	
9	0	0	0.0	0.0	0.0	0.0	

	…	zabsonre	zambia	zcc	zero	zip	zone	zones	zoom	zukes	zurich
0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[10 rows x 8240 columns]

[ ]:
```
X_train.isna().sum()
```

[ ]:
```
airline_American      0
airline_Delta         0
airline_Southwest     0
airline_US Airways    0
airline_United        0
                     ..
zone                  0
```

```
zones                 0
zoom                  0
zukes                 0
zurich                0
Length: 8240, dtype: int64
```

[ ]:
```
tfidf = vectorizer.transform(X_test['clean_text'])
tfidf_df = pd.DataFrame(tfidf.toarray(), columns=vectorizer.
 ↪get_feature_names_out())

X_test = X_test.reset_index(drop=True) # need to do this, since X_train and␣
 ↪tfidf_df have different indices
X_test = pd.concat([X_test, tfidf_df], axis=1)
X_test = X_test.drop('clean_text',axis=1)

X_test.head(10)
```

[ ]:

	airline_American	airline_Delta	airline_Southwest	airline_US Airways	\
0	0	1	0	0	
1	0	0	0	0	
2	1	0	0	0	
3	1	0	0	0	
4	0	0	0	0	
5	0	0	1	0	
6	0	1	0	0	
7	0	1	0	0	
8	0	1	0	0	
9	0	0	0	0	

	airline_United	airline_Virgin America	aa	aaaand	aadavantage	aadv	\
0	0	0	0.0	0.0	0.0	0.0	
1	1	0	0.0	0.0	0.0	0.0	
2	0	0	0.0	0.0	0.0	0.0	
3	0	0	0.0	0.0	0.0	0.0	
4	1	0	0.0	0.0	0.0	0.0	
5	0	0	0.0	0.0	0.0	0.0	
6	0	0	0.0	0.0	0.0	0.0	
7	0	0	0.0	0.0	0.0	0.0	
8	0	0	0.0	0.0	0.0	0.0	
9	1	0	0.0	0.0	0.0	0.0	

	…	zabsonre	zambia	zcc	zero	zip	zone	zones	zoom	zukes	zurich
0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
5  …       0.0     0.0  0.0   0.0  0.0   0.0    0.0   0.0    0.0     0.0
6  …       0.0     0.0  0.0   0.0  0.0   0.0    0.0   0.0    0.0     0.0
7  …       0.0     0.0  0.0   0.0  0.0   0.0    0.0   0.0    0.0     0.0
8  …       0.0     0.0  0.0   0.0  0.0   0.0    0.0   0.0    0.0     0.0
9  …       0.0     0.0  0.0   0.0  0.0   0.0    0.0   0.0    0.0     0.0

[10 rows x 8240 columns]
```

[ ]:
```
X_train.isna().sum()
```

[ ]:
```
airline_American      0
airline_Delta         0
airline_Southwest     0
airline_US Airways    0
airline_United        0
                     ..
zone                  0
zones                 0
zoom                  0
zukes                 0
zurich                0
Length: 8240, dtype: int64
```

[ ]:
```
X_test.shape, X_train.shape
```

[ ]:
```
((4392, 8240), (10248, 8240))
```

0.4 Create a Reduced Dimensions Dataset

User singular value decomposition (SVD) to create a reduced dimension dataset.

[ ]:
```
svd = TruncatedSVD(n_components=500, n_iter=10) #n_components is the number of␣
 ↪topics, which should be less than the number of features, and number of rows␣
 ↪in the matrix

X_train_dim_reduct = svd.fit_transform(X_train)
X_test_dim_reduct = svd.transform(X_test)
```

[ ]:
```
X_train_dim_reduct.shape, X_test_dim_reduct.shape
```

[ ]:
```
((10248, 500), (4392, 500))
```

[ ]:
```
## Fit RandomForrestClassifier using the dimension reduced training data
```

[ ]:
```
%%time

rf_clf = RandomForestClassifier(n_estimators=2000, max_leaf_nodes=50,␣
 ↪n_jobs=-1).fit(X_train_dim_reduct, y_train)
```

```
CPU times: user 3min 58s, sys: 862 ms, total: 3min 59s
Wall time: 26.2 s
```

[ ]:
```
%time

y_train_pred = rf_clf.predict(X_train_dim_reduct)
y_test_pred = rf_clf.predict(X_test_dim_reduct)
```

```
CPU times: user 1e+03 ns, sys: 1e+03 ns, total: 2 µs
Wall time: 3.1 µs
```

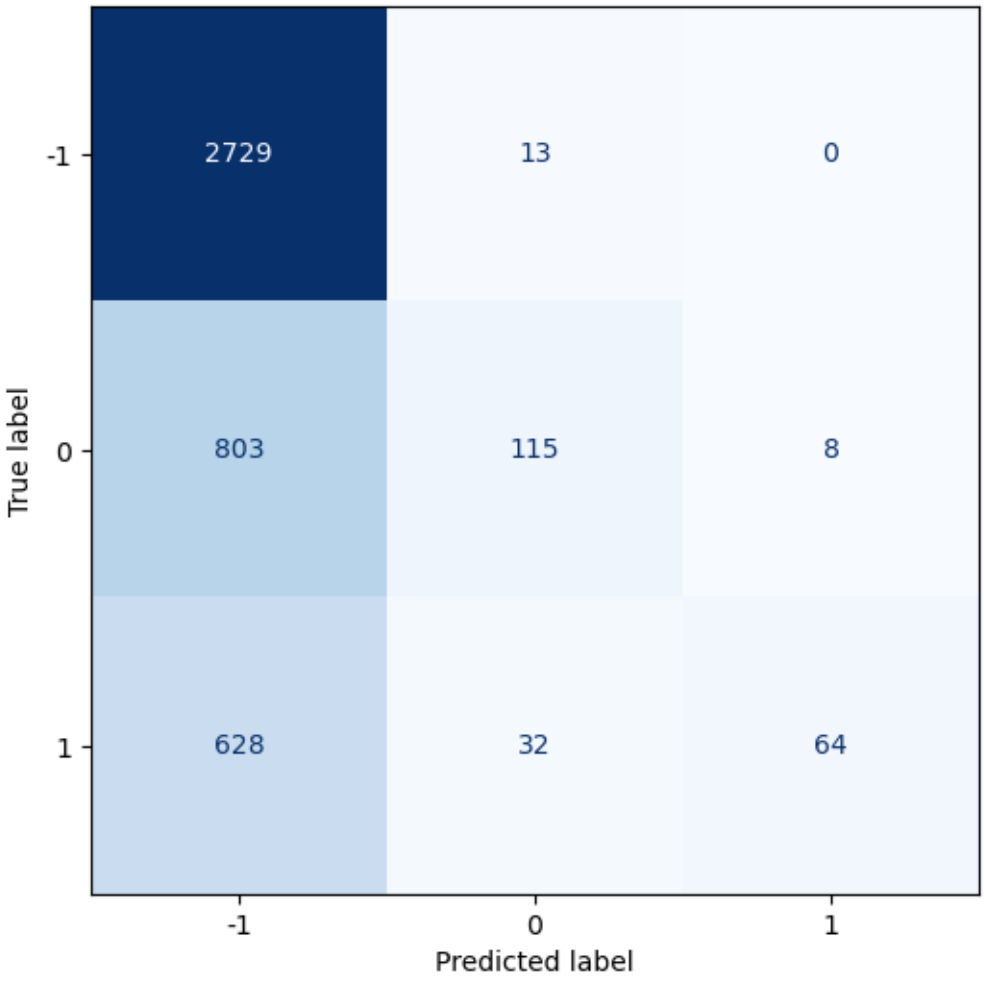
[ ]:
```
print(f"Train accuracy: {accuracy_score(y_train, y_train_pred):.4f}")
print(f"Test accuracy: {accuracy_score(y_test, y_test_pred):.4f}")
```

```
Train accuracy: 0.6782
Test accuracy: 0.6621
```

[ ]:
```
confusion_matrix(y_test, y_test_pred)
```

[ ]:
```
array([[2729,   13,    0],
       [ 803,  115,    8],
       [ 628,   32,   64]])
```

[ ]:
```
# display the confusion matrix
cm = confusion_matrix(y_test, y_test_pred)
#disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=le.classes_,␣
 ↪colorbar=False)
fig, ax = plt.subplots(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(
    y_test, y_test_pred, display_labels=[-1,0,1], ax=ax, colorbar=False,␣
 ↪cmap=plt.cm.Blues
)
plt.show()
```



0.5 Fit RandomForrestClassifier using the original training data

[ ]:
```
%%time

rf_clf = RandomForestClassifier(n_estimators=2000, max_leaf_nodes=50,␣
↪n_jobs=-1).fit(X_train, y_train)
```

```
CPU times: user 1min 8s, sys: 833 ms, total: 1min 9s
Wall time: 8.77 s
```

[ ]:
```
%time

y_train_pred = rf_clf.predict(X_train)
y_test_pred = rf_clf.predict(X_test)
```

```
CPU times: user 2 µs, sys: 0 ns, total: 2 µs
Wall time: 3.1 µs
```

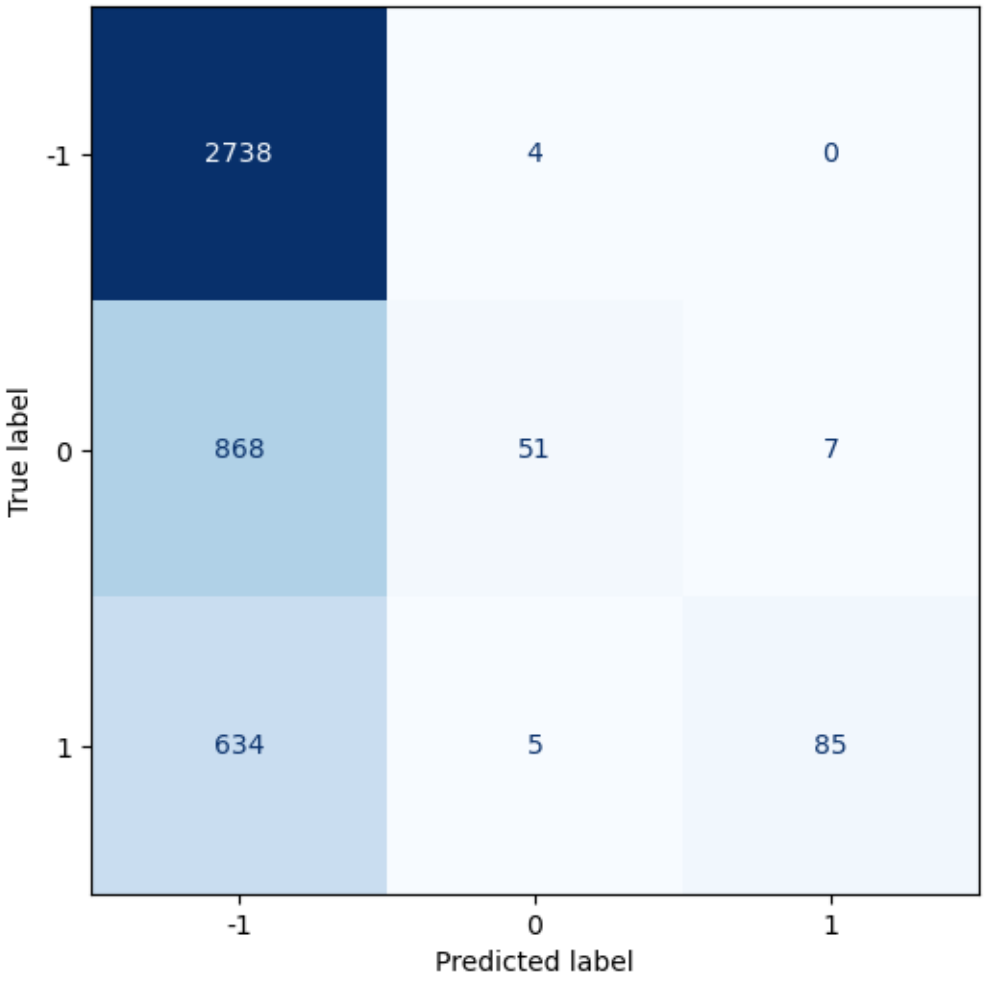
[ ]:
```
print(f"Train accuracy: {accuracy_score(y_train, y_train_pred):.4f}")
print(f"Test accuracy: {accuracy_score(y_test, y_test_pred):.4f}")
```

```
Train accuracy: 0.6594
Test accuracy: 0.6544
```

[ ]:
```
confusion_matrix(y_test, y_test_pred)
```

[ ]:
```
array([[2738,    4,    0],
       [ 868,   51,    7],
       [ 634,    5,   85]])
```

[ ]:
```
# display the confusion matrix
cm = confusion_matrix(y_test, y_test_pred)
#disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=le.classes_,␣
 ↪colorbar=False)
fig, ax = plt.subplots(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(
    y_test, y_test_pred, display_labels=[-1,0,1], ax=ax, colorbar=False,␣
 ↪cmap=plt.cm.Blues
)
plt.show()
```



0.6 Fit RandomForrestClassifier using the dimension reduced training data and Hyperparameter Tuning

[ ]:
```
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 30, stop =70, num = 10)]
# Number of features to consider at every split
max_features = [len(X_train.columns), len(X_train.columns)//2, len(X_train.
 ↪columns)//5]
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 50, num = 10)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
```

```
bootstrap = [True, False]

random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}

# Use the random grid to search for best hyperparameters

rf = RandomForestClassifier()
rf_random = RandomizedSearchCV(
    estimator = rf,
    param_distributions = random_grid,
    n_iter = 20,
    cv = 3,
    verbose=1,
    random_state=42,
    n_jobs = -1)

_ = rf_random.fit(X_train, y_train)
```

```
Fitting 3 folds for each of 20 candidates, totalling 60 fits
```

[ ]:
```
RandomizedSearchCV(cv=3, estimator=RandomForestClassifier(), n_iter=20,
                   n_jobs=-1,
                   param_distributions={'bootstrap': [True, False],
                                        'max_depth': [10, 14, 18, 23, 27, 32,
                                                      36, 41, 45, 50, None],
                                        'max_features': [8240, 4120, 1648],
                                        'min_samples_leaf': [1, 2, 4],
                                        'min_samples_split': [2, 5, 10],
                                        'n_estimators': [30, 34, 38, 43, 47, 52,
                                                         56, 61, 65, 70]},
                   random_state=42, verbose=1)
```

[ ]:
```
rf_random.best_params_
```

[ ]:
```
{'n_estimators': 52,
 'min_samples_split': 5,
 'min_samples_leaf': 2,
 'max_features': 'sqrt',
 'max_depth': 50,
 'bootstrap': False}
```

[ ]:
```
y_train_pred = rf_random.best_estimator_.predict(X_train)
y_test_pred = rf_random.best_estimator_.predict(X_test)
```

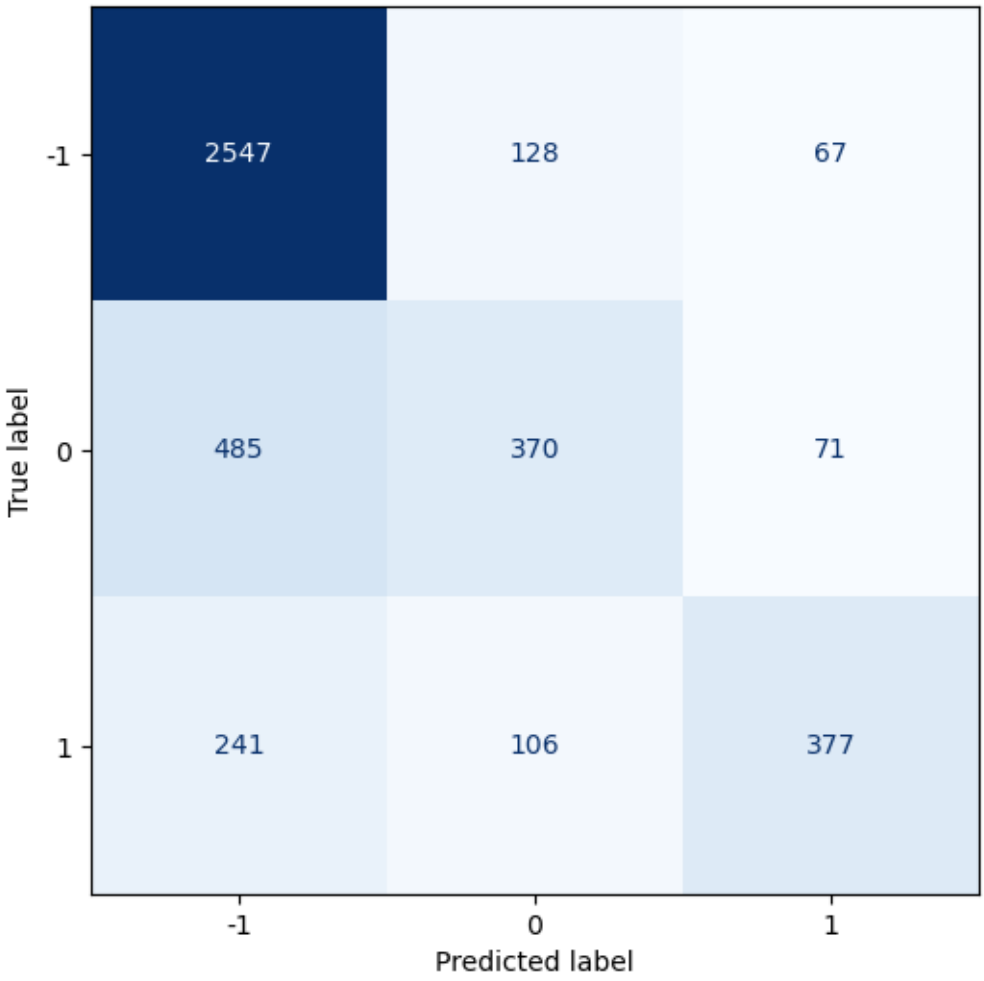
[ ]:
```
print(f"Train accuracy: {accuracy_score(y_train, y_train_pred):.4f}")
print(f"Test accuracy: {accuracy_score(y_test, y_test_pred):.4f}")
```

```
Train accuracy: 0.9453
Test accuracy: 0.7500
```

[ ]:
```
confusion_matrix(y_test, y_test_pred)
```

[ ]:
```
array([[2547,  128,   67],
       [ 485,  370,   71],
       [ 241,  106,  377]])
```

[ ]:
```
# display the confusion matrix
cm = confusion_matrix(y_test, y_test_pred)
#disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=le.classes_,␣
 ↪colorbar=False)
fig, ax = plt.subplots(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(
    y_test, y_test_pred, display_labels=[-1,0,1], ax=ax, colorbar=False,␣
 ↪cmap=plt.cm.Blues
)
plt.show()
```


2547
128
67
485
370
71
241
106
377
-1
0
1
True label
-1
0
1
Predicted label