```
import math
import os

import numpy as np
import pandas as pd
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline
from imblearn.under_sampling import RandomUnderSampler
from sklearn import preprocessing

from sklearn.compose import ColumnTransformer, make_column_selector
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MultiLabelBinarizer

os.environ['LOKY_MAX_CPU_COUNT'] = str(os.cpu_count()-1)  # To silence
warning : Could not find the number of physical cores

def allOutliersToBound(data):
    dfOutput = data.copy()

    for col in dfOutput.columns:
        if dfOutput[col].dtype == 'int64':  # The columns with 'int64'
come from encoded data and only having values of 0 and 1, function may
cause every data become same value
            continue
        if col == 'imdb_rating':            # Prevent adjusting the
target
            continue
        outliersToBound(dfOutput, col)

    return dfOutput

def outliersToBound(data, col):
    # Calculate 25% and 75% quarter, IQR and bound values
    Q1 = data[col].quantile(0.25)
    Q3 = data[col].quantile(0.75)
    IQR = Q3 - Q1
    lowerBound = Q1 - 1.5 * IQR
    upperBound = Q3 + 1.5 * IQR

    # Cap the outliers to lower bound or upper bound
    data[col] = data[col].clip(lower=lowerBound, upper=upperBound)
    return data
```

Load source file and show data for preview

```
filePath = '../Dataset/movie_metadata.csv'
df = pd.read_csv(filePath)
df
```

```
      color      director_name  num_critic_for_reviews  duration  \
0     Color      James Cameron                   723.0     178.0   
1     Color     Gore Verbinski                   302.0     169.0   
2     Color         Sam Mendes                   602.0     148.0   
3     Color  Christopher Nolan                   813.0     164.0   
4       NaN        Doug Walker                     NaN       NaN   
...     ...                ...                     ...       ...   
5038  Color        Scott Smith                     1.0      87.0   
5039  Color                NaN                    43.0      43.0   
5040  Color   Benjamin Roberds                    13.0      76.0   
5041  Color        Daniel Hsia                    14.0     100.0   
5042  Color           Jon Gunn                    43.0      90.0   

      director_facebook_likes  actor_3_facebook_likes      
actor_2_name  \
0                         0.0                   855.0  Joel David 
Moore   
1                       563.0                  1000.0     Orlando 
Bloom   
2                         0.0                   161.0      Rory 
Kinnear   
3                     22000.0                 23000.0    Christian 
Bale   
4                       131.0                     NaN        Rob 
Walker   
...                       ...                     ...               ..
.   
5038                      2.0                   318.0     Daphne 
Zuniga   
5039                      NaN                   319.0     Valorie 
Curry   
5040                      0.0                     0.0     Maxwell 
Moody   
5041                      0.0                   489.0     Daniel 
Henney   
5042                     16.0                    16.0  Brian 
Herzlinger   

      actor_1_facebook_likes        gross                    
genres  \
0                     1000.0  760505847.0  Action|Adventure|Fantasy|
Sci-Fi   
1                    40000.0  309404152.0         Action|Adventure|
Fantasy   
2                    11000.0  200074175.0        Action|Adventure|
Thriller   
3                    27000.0  448130642.0                   Action|
Thriller   
4                      131.0          NaN                          
Documentary   
```

```
...                          ...          ...                                 
...
5038                       637.0          NaN                      Comedy|
Drama
5039                       841.0          NaN     Crime|Drama|Mystery|
Thriller
5040                         0.0          NaN            Drama|Horror|
Thriller
5041                       946.0      10443.0             Comedy|Drama|
Romance
5042                        86.0      85222.0                           
Documentary

      ... num_user_for_reviews language  country  content_rating       
budget  \
0     ...               3054.0  English      USA           PG-13  
237000000.0   
1     ...               1238.0  English      USA           PG-13  
300000000.0   
2     ...                994.0  English       UK           PG-13  
245000000.0   
3     ...               2701.0  English      USA           PG-13  
250000000.0   
4     ...                  NaN      NaN      NaN             NaN          
NaN   
...   ...                  ...      ...      ...             ...          
...   
5038  ...                  6.0  English   Canada             NaN          
NaN   
5039  ...                359.0  English      USA           TV-14          
NaN   
5040  ...                  3.0  English      USA             NaN       
1400.0   
5041  ...                  9.0  English      USA           PG-13          
NaN   
5042  ...                 84.0  English      USA              PG       
1100.0   

      title_year actor_2_facebook_likes imdb_score  aspect_ratio  \
0         2009.0                  936.0        7.9          1.78   
1         2007.0                 5000.0        7.1          2.35   
2         2015.0                  393.0        6.8          2.35   
3         2012.0                23000.0        8.5          2.35   
4            NaN                   12.0        7.1           NaN   
...          ...                    ...        ...           ...   
5038      2013.0                  470.0        7.7           NaN   
5039         NaN                  593.0        7.5         16.00   
5040      2013.0                    0.0        6.3           NaN   
5041      2012.0                  719.0        6.3          2.35   
5042      2004.0                   23.0        6.6          1.85   
```

```
     movie_facebook_likes
0                   33000
1                       0
2                   85000
3                  164000
4                       0
...                   ...
5038                   84
5039                32000
5040                   16
5041                  660
5042                  456

[5043 rows x 28 columns]
```

Start data processing

Drop duplicate data

Treat movie_imdb_link as the unique identifier, drop the duplicate rows and check the data again

```
df.sort_values(by='num_voted_users',kind="mergesort",inplace=True)
# Sort by num_voted_users, default ascending
dfNoDuplicate = df.drop_duplicates(subset=['movie_imdb_link'],
keep="last")      # Keep the lastest data (assume num_voted_users only
increase along the time)
dfNoDuplicate
```

```
                 color       director_name  num_critic_for_reviews
duration  \
4702             Color       Bill Benenson                     1.0
71.0
4958   Black and White    Harry F. Millarde                     1.0
110.0
279                NaN  Christopher Barnard                     NaN
22.0
4244             Color           Dan Perri                     1.0
100.0
4716             Color      Lance McDaniel                     NaN
90.0
...                ...                 ...                     ...
...
3355             Color   Quentin Tarantino                   215.0
178.0
683              Color       David Fincher                   315.0
151.0
97               Color   Christopher Nolan                   642.0
148.0
```

```
66              Color    Christopher Nolan                    645.0
152.0
1937            Color       Frank Darabont                    199.0
142.0

      director_facebook_likes  actor_3_facebook_likes
actor_2_name  \
4702                      0.0                    21.0             Dave
Fennoy
4958                      0.0                     0.0          Johnnie
Walker
279                       0.0                     NaN
NaN
4244                      0.0                   338.0            David
Proval
4716                      0.0                   271.0  Steven Michael
Quezada
...                       ...                     ...
...
3355                  16000.0                   857.0             Eric
Stoltz
683                   21000.0                   637.0
Meat Loaf
97                    22000.0                 23000.0
Tom Hardy
66                    22000.0                 11000.0            Heath
Ledger
1937                      0.0                   461.0          Jeffrey
DeMunn

      actor_1_facebook_likes        gross  \
4702                  1000.0          NaN
4958                     2.0    3000000.0
279                      5.0          NaN
4244                   749.0          NaN
4716                   595.0          NaN
...                      ...          ...
3355                 13000.0  107930000.0
683                  11000.0   37023395.0
97                   29000.0  292568851.0
66                   23000.0  533316061.0
1937                 11000.0   28341469.0

                                  genres  ... num_user_for_reviews
\
4702                         Documentary  ...                  1.0

4958                         Crime|Drama  ...                  1.0

279                               Comedy  ...                  NaN
```

```
4244  Comedy|Drama|Mystery|Romance|Thriller  ...                  1.0

4716                  Action|Drama|Thriller  ...                  1.0

...                                     ...  ...                  ...

3355                            Crime|Drama  ...               2195.0

683                                   Drama  ...               2968.0

97          Action|Adventure|Sci-Fi|Thriller  ...               2803.0

66               Action|Crime|Drama|Thriller  ...               4667.0

1937                            Crime|Drama  ...               4144.0


     language  country  content_rating       budget  title_year  \
4702  English      USA             NaN     650000.0      2014.0
4958      NaN      USA             NaN     100000.0      1920.0
279       NaN      NaN             NaN          NaN         NaN
4244  English      USA             NaN    2100000.0      2015.0
4716  English      USA           PG-13     600000.0      2014.0
...       ...      ...             ...          ...         ...
3355  English      USA               R    8000000.0      1994.0
683   English      USA               R   63000000.0      1999.0
97    English      USA           PG-13  160000000.0      2010.0
66    English      USA           PG-13  185000000.0      2008.0
1937  English      USA               R   25000000.0      1994.0

     actor_2_facebook_likes imdb_score  aspect_ratio
movie_facebook_likes
4702                  338.0        7.4           NaN
5
4958                    2.0        4.8          1.33
0
279                     NaN        7.2           NaN
0
4244                  354.0        6.7          2.39
14
4716                  412.0        8.0           NaN
9
...                     ...        ...           ...
...
3355                  902.0        8.9          2.35
45000
683                   783.0        8.8          2.35
48000
97                  27000.0        8.8          2.35
```

```
175000
66                    13000.0        9.0           2.35
37000
1937                    745.0        9.3           1.85
108000

[4919 rows x 28 columns]
```

Remove irrelevant and unsuitable columns

Dropping the links (irrelevant) and names (unsuitable)

```
dfValueFeature = dfNoDuplicate.drop(columns=
    ['movie_imdb_link',     # Link is nothing related to the score
     'movie_title',
     'director_name',
     'actor_1_name',
     'actor_2_name',
     'actor_3_name'])       # Title, director and actors might affect
the score, but training a model with 'names' is highly lead to
overfitting
dfValueFeature

                  color  num_critic_for_reviews  duration  \
4702              Color                     1.0      71.0
4958    Black and White                     1.0     110.0
279                 NaN                     NaN      22.0
4244              Color                     1.0     100.0
4716              Color                     NaN      90.0
...                 ...                     ...       ...
3355              Color                   215.0     178.0
683               Color                   315.0     151.0
97                Color                   642.0     148.0
66                Color                   645.0     152.0
1937              Color                   199.0     142.0

      director_facebook_likes  actor_3_facebook_likes
actor_1_facebook_likes  \
4702                      0.0                    21.0
1000.0
4958                      0.0                     0.0
2.0
279                       0.0                     NaN
5.0
4244                      0.0                   338.0
749.0
4716                      0.0                   271.0
595.0
...                       ...                     ...
```

```
...
3355                  16000.0                    857.0
13000.0
683                   21000.0                    637.0
11000.0
97                    22000.0                  23000.0
29000.0
66                    22000.0                  11000.0
23000.0
1937                      0.0                    461.0
11000.0

            gross                                genres
num_voted_users  \
4702          NaN                           Documentary
5
4958    3000000.0                           Crime|Drama
5
279           NaN                                Comedy
6
4244          NaN  Comedy|Drama|Mystery|Romance|Thriller
6
4716          NaN                 Action|Drama|Thriller
6
...           ...                                   ...
...
3355  107930000.0                           Crime|Drama
1324680
683    37023395.0                                 Drama
1347461
97    292568851.0       Action|Adventure|Sci-Fi|Thriller
1468200
66    533316061.0           Action|Crime|Drama|Thriller
1676169
1937   28341469.0                           Crime|Drama
1689764

      cast_total_facebook_likes  ...  num_user_for_reviews language
country  \
4702                       1359  ...                   1.0  English
USA
4958                          4  ...                   1.0      NaN
USA
279                           5  ...                   NaN      NaN
NaN
4244                       1814  ...                   1.0  English
USA
4716                       1754  ...                   1.0  English
USA
```

```
...                           ...  ...                     ...      ...
...
3355                        16557  ...                  2195.0  English
USA
683                         13209  ...                  2968.0  English
USA
97                          81115  ...                  2803.0  English
USA
66                          57802  ...                  4667.0  English
USA
1937                        13495  ...                  4144.0  English
USA

     content_rating       budget title_year  actor_2_facebook_likes  \
4702            NaN     650000.0     2014.0                   338.0
4958            NaN     100000.0     1920.0                     2.0
279             NaN          NaN        NaN                     NaN
4244            NaN    2100000.0     2015.0                   354.0
4716          PG-13     600000.0     2014.0                   412.0
...             ...          ...        ...                     ...
3355              R    8000000.0     1994.0                   902.0
683               R   63000000.0     1999.0                   783.0
97            PG-13  160000000.0     2010.0                 27000.0
66            PG-13  185000000.0     2008.0                 13000.0
1937              R   25000000.0     1994.0                   745.0

      imdb_score  aspect_ratio  movie_facebook_likes
4702         7.4           NaN                     5
4958         4.8          1.33                     0
279          7.2           NaN                     0
4244         6.7          2.39                    14
4716         8.0           NaN                     9
...          ...           ...                   ...
3355         8.9          2.35                 45000
683          8.8          2.35                 48000
97           8.8          2.35                175000
66           9.0          2.35                 37000
1937         9.3          1.85                108000

[4919 rows x 22 columns]
```

Fill missing values

Fill 0 or median for numeric data and most frequent value for object data

```
# The minimum value is 1 instead of 0, while the website is showing 0,
mean most NaN refer to 0
dfValueFeature['num_critic_for_reviews'] =
df['num_critic_for_reviews'].fillna(0)
```

```
# Same as 'num_critic_for_reviews'
dfValueFeature['num_user_for_reviews'] =
df['num_user_for_reviews'].fillna(0)

fillNanTransformer = ColumnTransformer(
    transformers=[
        # Minimum value of 'likes' type data are 0 (NaN might not
refer to 0 in these case), gross and budget likely not possible to be
0, fills these columns with median
        ('num', SimpleImputer(strategy='median'),
make_column_selector(dtype_include=np.number)),
        # Fill by most frequent
        ('cat', SimpleImputer(strategy='most_frequent'),
make_column_selector(dtype_include=object)),
    ], verbose_feature_names_out=False).set_output(transform="pandas")

dfFilledData = fillNanTransformer.fit_transform(dfValueFeature)
# Check if any missing value haven't filled
dfFilledData
```

```
      num_critic_for_reviews  duration  director_facebook_likes  \
4702                     1.0      71.0                      0.0   
4958                     1.0     110.0                      0.0   
279                      0.0      22.0                      0.0   
4244                     1.0     100.0                      0.0   
4716                     0.0      90.0                      0.0   
...                      ...       ...                      ...   
3355                   215.0     178.0                  16000.0   
683                    315.0     151.0                  21000.0   
97                     642.0     148.0                  22000.0   
66                     645.0     152.0                  22000.0   
1937                   199.0     142.0                      0.0   

      actor_3_facebook_likes  actor_1_facebook_likes        gross  \
4702                    21.0                  1000.0   25035665.0   
4958                     0.0                     2.0    3000000.0   
279                    365.5                     5.0   25035665.0   
4244                   338.0                   749.0   25035665.0   
4716                   271.0                   595.0   25035665.0   
...                      ...                     ...          ...   
3355                   857.0                 13000.0  107930000.0   
683                    637.0                 11000.0   37023395.0   
97                   23000.0                 29000.0  292568851.0   
66                   11000.0                 23000.0  533316061.0   
1937                   461.0                 11000.0   28341469.0   

      num_voted_users  cast_total_facebook_likes  facenumber_in_poster
\
4702              5.0                     1359.0                   1.0
```

```
4958           5.0                         4.0                    1.0

279            6.0                         5.0                    0.0

4244           6.0                      1814.0                    0.0

4716           6.0                      1754.0                    0.0

...            ...                         ...                    ...

3355     1324680.0                     16557.0                    1.0

683      1347461.0                     13209.0                    2.0

97       1468200.0                     81115.0                    0.0

66       1676169.0                     57802.0                    0.0

1937     1689764.0                     13495.0                    0.0


      num_user_for_reviews  ...  actor_2_facebook_likes  imdb_score  \
4702                   1.0  ...                   338.0         7.4
4958                   1.0  ...                     2.0         4.8
279                    0.0  ...                   593.0         7.2
4244                   1.0  ...                   354.0         6.7
4716                   1.0  ...                   412.0         8.0
...                    ...  ...                     ...         ...
3355                2195.0  ...                   902.0         8.9
683                 2968.0  ...                   783.0         8.8
97                  2803.0  ...                 27000.0         8.8
66                  4667.0  ...                 13000.0         9.0
1937                4144.0  ...                   745.0         9.3

      aspect_ratio  movie_facebook_likes            color  \
4702          2.35                   5.0            Color
4958          1.33                   0.0  Black and White
279           2.35                   0.0            Color
4244          2.39                  14.0            Color
4716          2.35                   9.0            Color
...            ...                   ...              ...
3355          2.35               45000.0            Color
683           2.35               48000.0            Color
97            2.35              175000.0            Color
66            2.35               37000.0            Color
1937          1.85              108000.0            Color

                                  genres  \
4702                         Documentary
4958                         Crime|Drama
```

```
279                                   Comedy
4244  Comedy|Drama|Mystery|Romance|Thriller
4716                 Action|Drama|Thriller
...                                    ...
3355                           Crime|Drama
683                                  Drama
97         Action|Adventure|Sci-Fi|Thriller
66              Action|Crime|Drama|Thriller
1937                           Crime|Drama

                                          plot_keywords language
country  \
4702  east africa|hunter gatherer|indigenous|rift va...  English
USA
4958    family relationships|gang|idler|poorhouse|thief  English
USA
279                                      based on novel  English
USA
4244                                     based on novel  English
USA
4716            china|faith|panama|photography|suspense  English
USA
...                                                 ...      ...      .
..
3355  black comedy|cunnilingus|neo noir|nonlinear ti...  English
USA
683   anti establishment|dark humor|fighting|multipl...  English
USA
97    ambiguous ending|corporate espionage|dream|sub...  English
USA
66    based on comic book|dc comics|psychopath|star ...  English
USA
1937  escape from prison|first person narration|pris...  English
USA

     content_rating
4702              R
4958              R
279               R
4244              R
4716          PG-13
...             ...
3355              R
683               R
97            PG-13
66            PG-13
1937              R

[4919 rows x 22 columns]
```

Encode data

For class having dominant groups, replace value to belongs to the dominant group or not For multiple group distributed, do one hot encoding

```
# Replace color column to binary value, true(1) for color and false(0)
for black and white
dfFilledData['is_color'] = np.where(dfFilledData['color'] == 'Color',
1, 0)
dfIsColor = dfFilledData.drop('color', axis=1)

# Replace language column to binary value, true(1) for English and
false(0) for other
dfIsColor['is_english'] = np.where(dfIsColor['language'] == 'English',
1, 0)
dfIsEnglish = dfIsColor.drop('language', axis=1)

# Replace country column to two binary column, true(1) in is_USA for
USA or true(1) in is_UK for UK, other if false(0) for both
countryCat =
dfIsEnglish['country'].where(dfIsEnglish['country'].isin(['USA',
'UK']), 'Other')
countries_dummies = pd.get_dummies(countryCat, dtype=int, prefix='is')
dfSplitCountry : pd.DataFrame = pd.concat([dfIsEnglish,
countries_dummies], axis=1)
dfCountries = dfSplitCountry.drop(columns=['is_Other','country'])
```

Split the cell's value as there are some cells contain more than one value Do one-hot encoding and check total numbers of features Built the dataframe after it

```
dfCountries['contentRatingList'] =
dfCountries['content_rating'].str.split(r'\s*\|\s*')
mlbCont = MultiLabelBinarizer()
contEncoded = mlbCont.fit_transform(dfCountries['contentRatingList'])
contEncoded.shape[1]
```

```
18
```

```
dfCountries['genresList'] = dfCountries['genres'].str.split(r'\s*\|\
s*')
mlbGenre = MultiLabelBinarizer()
genreEncoded = mlbGenre.fit_transform(dfCountries['genresList'])
genreEncoded.shape[1]
```

```
26
```

```
dfCountries['keywordsList'] =
dfCountries['plot_keywords'].str.split(r'\s*\|\s*')
mlbKeyword = MultiLabelBinarizer()
keywordEncoded = mlbKeyword.fit_transform(dfCountries['keywordsList'])
keywordEncoded.shape[1]
```

```
8086

# Encoded content_rating and genres; drop keyword as there are more
than 8000 different value
dfCont = pd.DataFrame(contEncoded, columns=mlbCont.classes_,
index=dfCountries.index)
dfContInData = pd.concat([dfCountries, dfCont], axis=1)

dfGenres = pd.DataFrame(genreEncoded, columns=mlbGenre.classes_,
index=dfContInData.index)
dfGenreInData = pd.concat([dfContInData, dfGenres], axis=1)

dfGenreEncoded =
dfGenreInData.drop(columns=['content_rating','contentRatingList','genr
es','genresList','plot_keywords','keywordsList'])

# Process the target columns, convert from continuous data to class
data
dfGenreEncoded['imdb_rating'] = pd.cut(dfGenreEncoded['imdb_score'],
bins=[0, 2, 4, 6, 8, 10], labels=[1, 2, 3, 4, 5])
dfFeaValue = dfGenreEncoded.drop('imdb_score',axis=1)
```

Handling outliers

Replace outliers with the bounds of the data

```
dfRemoveOutliers = allOutliersToBound(dfFeaValue)
```

Resampling

Constraint data to suitable types

```
# Most of the data columns are counted by people and only one is year
columns = [
    'num_critic_for_reviews',
    'director_facebook_likes',
    'actor_3_facebook_likes',
    'actor_1_facebook_likes',
    'num_voted_users',
    'cast_total_facebook_likes',
    'facenumber_in_poster',
    'num_user_for_reviews',
    'title_year',
    'actor_2_facebook_likes',
    'movie_facebook_likes'
]
# Change the data type to int to prevent data generate by oversampling
contain float value
dfRemoveOutliersInt = dfRemoveOutliers.astype({col: int for col in
```

```
columns})
dfRemoveOutliersInt

      num_critic_for_reviews  duration  director_facebook_likes  \
4702                       1      71.0                        0   
4958                       1     110.0                        0   
279                        0      55.5                        0   
4244                       1     100.0                        0   
4716                       0      90.0                        0   
...                      ...       ...                      ...   
3355                     215     155.5                      459   
683                      315     151.0                      459   
97                       407     148.0                      459   
66                       407     152.0                      459   
1937                     199     142.0                        0   

      actor_3_facebook_likes  actor_1_facebook_likes         gross  \
4702                      21                    1000  2.503566e+07   
4958                       0                       2  3.000000e+06   
279                      365                       5  2.503566e+07   
4244                     338                     749  2.503566e+07   
4716                     271                     595  2.503566e+07   
...                      ...                     ...           ...   
3355                     857                   13000  1.079300e+08   
683                      637                   11000  3.702340e+07   
97                      1378                   26588  1.150246e+08   
66                      1378                   23000  1.150246e+08   
1937                     461                   11000  2.834147e+07   

      num_voted_users  cast_total_facebook_likes  facenumber_in_poster
\
4702                5                       1359                     1

4958                5                          4                     1

279                 6                          5                     0

4244                6                       1814                     0

4716                6                       1754                     0

...               ...                        ...                   ...

3355           221859                      16557                     1

683            221859                      13209                     2

97             221859                      31937                     0

66             221859                      31937                     0
```

```
1937          221859                      13495                    0


      num_user_for_reviews  ...  News  Reality-TV  Romance  Sci-Fi
Short  \
4702                     1  ...     0           0        0       0
0
4958                     1  ...     0           0        0       0
0
279                      0  ...     0           0        0       0
0
4244                     1  ...     0           0        1       0
0
4716                     1  ...     0           0        0       0
0
...                    ...  ...   ...         ...      ...     ...
...
3355                   704  ...     0           0        0       0
0
683                    704  ...     0           0        0       0
0
97                     704  ...     0           0        0       1
0
66                     704  ...     0           0        0       0
0
1937                   704  ...     0           0        0       0
0

      Sport  Thriller  War  Western  imdb_rating
4702      0         0    0        0            4
4958      0         0    0        0            3
279       0         0    0        0            4
4244      0         1    0        0            4
4716      0         1    0        0            4
...     ...       ...  ...      ...          ...
3355      0         0    0        0            5
683       0         0    0        0            5
97        0         1    0        0            5
66        0         1    0        0            5
1937      0         0    0        0            5

[4919 rows x 64 columns]
```

Define strategy for resampling

```
X = dfRemoveOutliersInt.drop('imdb_rating', axis=1)
y = dfRemoveOutliersInt['imdb_rating']
```

Scaling

Use minmax scaler to scale all data except target to range 0 to 1

```
ct = ColumnTransformer([('scale', preprocessing.MinMaxScaler(),
dfRemoveOutliersInt.columns.drop('imdb_rating'))],
                       remainder='passthrough',
verbose_feature_names_out=False).fit(dfRemoveOutliersInt)
ct.set_output(transform='pandas')
dfProcessedData = ct.transform(dfRemoveOutliersInt)
```

Write data into file

```
dfProcessedData.to_csv(f'../Dataset/dataFrameProcessed', index=False)
```

Resampling

```python
import os

import numpy as np
import pandas as pd
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline
from matplotlib import pyplot as plt

from sklearn.ensemble import RandomForestClassifier,
HistGradientBoostingClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.metrics import make_scorer, precision_score,
confusion_matrix, ConfusionMatrixDisplay, get_scorer
from sklearn.model_selection import train_test_split, cross_validate,
StratifiedKFold, GridSearchCV
from sklearn.preprocessing import MinMaxScaler

from sklearn.svm import SVC

os.environ['LOKY_MAX_CPU_COUNT'] = str(os.cpu_count()-1)  # To silence
warning : Could not find the number of physical cores

randomState = 42

# Function to calculate weighted specificity
def multiclassSpecificity(yTrue, yPredict):
    cm = confusion_matrix(yTrue, yPredict)

    specificities = []
    for i in range(len(cm)):
        trueNegative = np.sum(cm) - np.sum(cm[i, :]) - np.sum(cm[:,
i]) + cm[i, i]
        falsePositive = np.sum(cm[:, i]) - cm[i, i]
        specificity = trueNegative / (trueNegative + falsePositive) if
(trueNegative + falsePositive) > 0 else 0
        specificities.append(specificity)

    return np.mean(specificities)
```

Read data and define target

```python
df = pd.read_csv('../Dataset/dataFrameProcessed')
targetCol = 'imdb_rating'
```

Split features and target data

```python
X = df.drop(columns=[targetCol])
y = df[targetCol]
```

Built set of models selected

```
models = {
    'Support Vector Machine': SVC(kernel='rbf',
random_state=randomState),
    'Random Forest'         :
RandomForestClassifier(random_state=randomState),
    'Hist Gradient Boosting':
HistGradientBoostingClassifier(random_state=randomState),
}
```

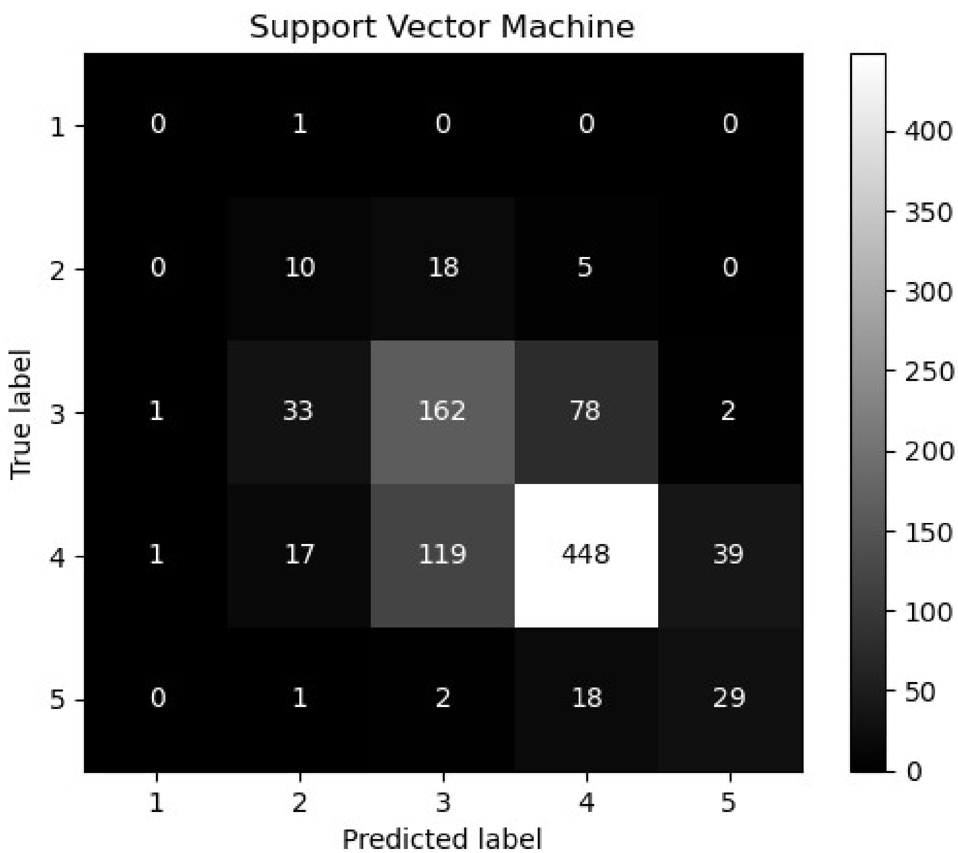
Built scoring metrix

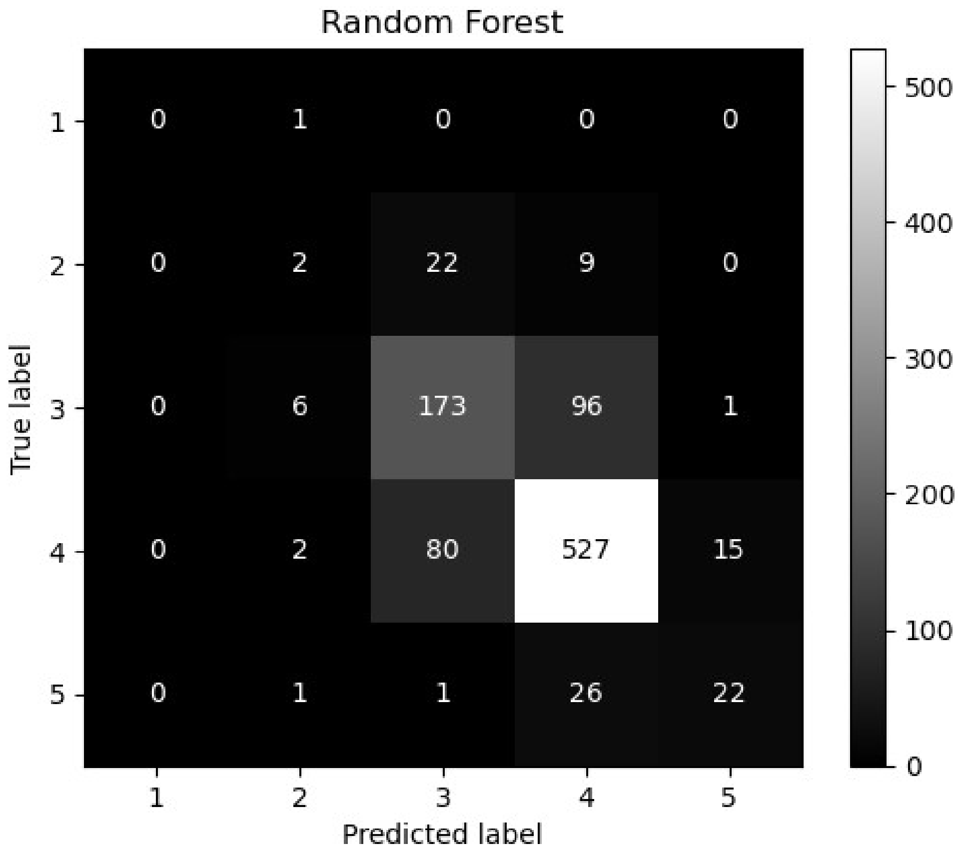
```
scoringMetrix = {
    'accuracy'    : 'accuracy',
    'precision'   : make_scorer(precision_score, average='weighted',
zero_division=1),
    'recall'      : 'recall_weighted',
    'f1'          : 'f1_weighted',
    'specificity' : make_scorer(multiclassSpecificity)
}
```

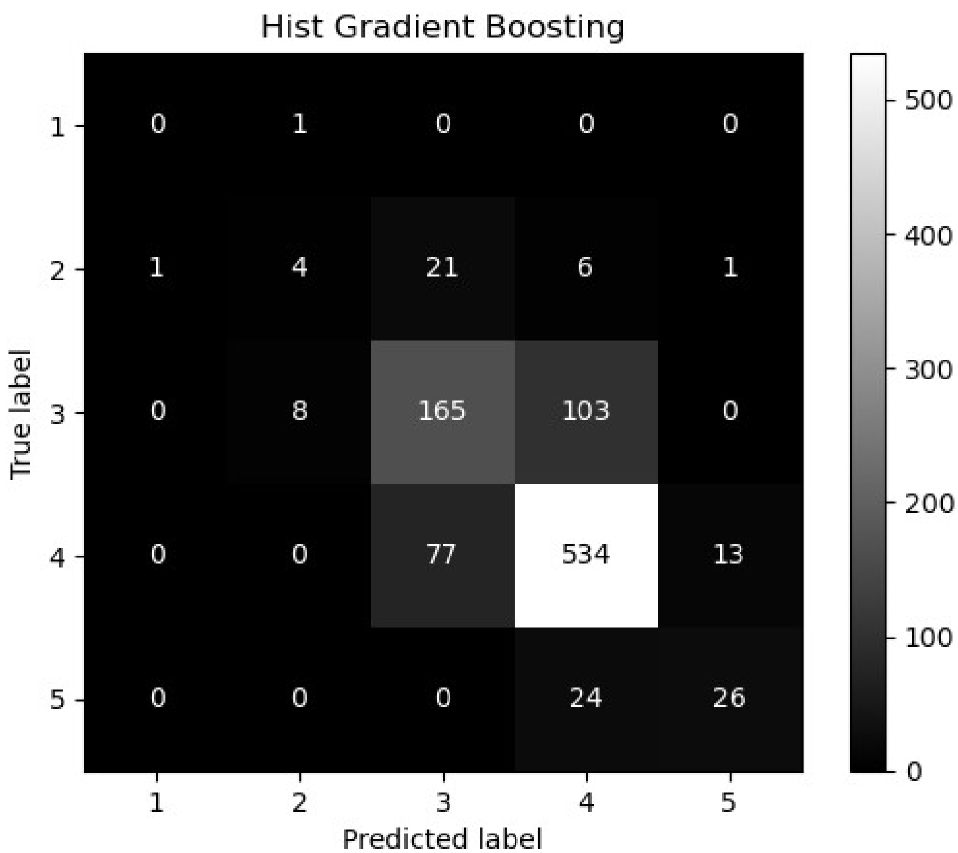
Split data (80:20)

```
XTrain, XTest, yTrain, yTest = train_test_split(X, y, test_size=0.2,
random_state=randomState , stratify=y)

# Function to make pipeline
def makePipeline(modelToUsed):
    steps = [
        ('over', SMOTE(sampling_strategy='auto', random_state=42,
k_neighbors=3)),
        ('scaling', MinMaxScaler().set_output(transform='pandas')),
        ('feature_selection', SelectFromModel(
            estimator=RandomForestClassifier(random_state=randomState,
n_jobs=-1),
            threshold='median')),
        ('classifier', modelToUsed),
    ]
    return Pipeline(steps=steps)

for name, model in models.items():
    # Fit pipeline
    pipeCM = makePipeline(model)
    pipeCM.fit(XTrain, yTrain)

    # Show confusion metrix
    dispCM = ConfusionMatrixDisplay.from_estimator(
        pipeCM,
        XTest,
        yTest,
```

```
        cmap='gist_gray',
    )
    dispCM.ax_.set_title(name)
    plt.show()
```




Random Forest
True label
Predicted label
1
2
3
4
5
0 1 0 0 0
0 2 22 9 0
0 6 173 96 1
0 2 80 527 15
0 1 1 26 22
500
400
300
200
100
0




```
for name, model in models.items():
    pipeSC = makePipeline(model)
    pipeSC.fit(XTrain, yTrain)

    # Print model name and its results
    print(f'\n{name:<12}')
    for metricName, scorer in scoringMetrix.items():
        scorer = get_scorer(scorer)
        score = scorer(pipeSC, XTest, yTest)
        print(f"{metricName:<12}: {score:.5f}")
```

```
Support Vector Machine
accuracy    : 0.65955
precision   : 0.69490
recall      : 0.65955
f1          : 0.67352
specificity : 0.88450

Random Forest
accuracy    : 0.73577
precision   : 0.71973
recall      : 0.73577
```

```
f1          : 0.72557
specificity : 0.89260

Hist Gradient Boosting
accuracy    : 0.74085
precision   : 0.72702
recall      : 0.74085
f1          : 0.73153
specificity : 0.89333
```

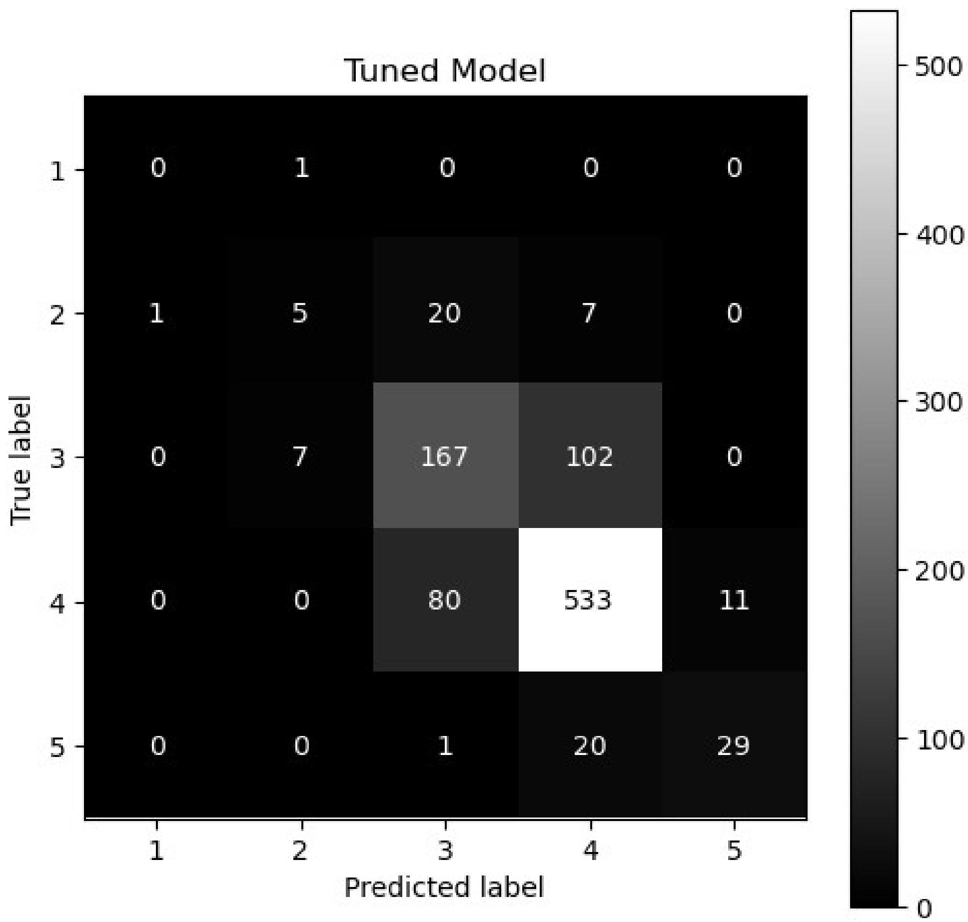
```
dataset_results = []

for name, model in models.items():
    pipeCV = makePipeline(model)

    cvResult = cross_validate(
        pipeCV,
        X,
        y,
        cv= StratifiedKFold(n_splits=5, shuffle=True,
random_state=randomState),
        scoring=scoringMetrix,
        n_jobs=-1,
        error_score='raise'
    )

    dataset_results.append({
        'Model'          : name,
        'Accuracy'       : np.mean(cvResult['test_accuracy']),
        'Precision'      : np.mean(cvResult['test_precision']),
        'Recall'         : np.mean(cvResult['test_recall']),
        'F1 Score'       : np.mean(cvResult['test_f1']),
        'Specificity'    : np.mean(cvResult['test_specificity'])
    })

print(pd.DataFrame(dataset_results).set_index('Model'))
```

```
                        Accuracy  Precision    Recall  F1 Score
Specificity
Model

Support Vector Machine  0.666801   0.709935  0.666801  0.682750
0.890008
Random Forest           0.730029   0.722563  0.730029  0.724174
0.893109
Hist Gradient Boosting  0.744256   0.734465  0.744256  0.736981
0.896896
```

```
pipeGrid =
makePipeline(HistGradientBoostingClassifier(random_state=randomState))
```

```python
param_grid = {
    'classifier__learning_rate': [0.1, 0.2],
    'classifier__max_iter': [100,  200],
    'classifier__max_depth': [None, 10],
    'classifier__max_leaf_nodes': [10, 20],
    'classifier__l2_regularization': [0.01, 0.02],
}

gridSearch = GridSearchCV(
    estimator=pipeGrid,
    param_grid=param_grid,
    scoring='f1_weighted',
    cv=5,
    n_jobs=-1,
    verbose=3
)

gridSearch.fit(XTrain, yTrain)

print(gridSearch.best_params_)
for metricName, scorer in scoringMetrix.items():
    scorer = get_scorer(scorer)
    score = scorer(gridSearch, XTest, yTest)
    print(f"{metricName:<12}: {score:.5f}")

yPred = gridSearch.predict(XTest)
confusionMatrix = confusion_matrix(yTest, yPred)
display = ConfusionMatrixDisplay(confusion_matrix=confusionMatrix,
display_labels=gridSearch.classes_)

fig, axes = plt.subplots(figsize=(6, 6))
display.plot(cmap='gist_gray', ax=axes, values_format='d')
plt.title(f'Tuned Model')
plt.show()
```

```
Fitting 5 folds for each of 32 candidates, totalling 160 fits
{'classifier__l2_regularization': 0.02, 'classifier__learning_rate':
0.1, 'classifier__max_depth': 10, 'classifier__max_iter': 200,
'classifier__max_leaf_nodes': 20}
accuracy    : 0.74593
precision   : 0.73509
recall      : 0.74593
f1          : 0.73791
specificity : 0.89556
```


Tuned Model
True label
Predicted label
1
2
3
4
5
0 1 0 0 0
1 5 20 7 0
0 7 167 102 0
0 0 80 533 11
0 0 1 20 29
500
400
300
200
100
0