Sampling&Modelling

April 2, 2021

[1]:
```
try:
    from collections import OrderedDict
except ImportError:
    from ordereddict import OrderedDict

import pandas as pd

# Array
import numpy as np

# Decompress the file
import gzip

# Visualizations
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import seaborn as sns
import matplotlib.colors as colors
%matplotlib inline

# Datetime
from datetime import datetime

## Warnings
import warnings
from scipy import stats
warnings.filterwarnings('ignore')

# Large dataset
import dask.bag as db
Cleanreview_df = pd.read_csv('Cleanreview_VideoGames.csv' )
```

[2]:
```
Cleanreview_df.head()
```

[2]:

	Rating	ReviewerID	ProductID	ReviewerName	\
0	5	A1HP7NVNPFMA4N	0700026657	Ambrosia075	
1	4	A1JGAP0185YJI6	0700026657	travis	

```
2       3  A1YJWEXHQBWK2B  0700026657  Vincent G. Mezera
3       2  A2204E1TH211HT  0700026657         Grandma KR
4       5  A2RF5B5H74JLPE  0700026657                jon

                                ProductDescription  Price  \
0  Anno 2070, the newest version of the award-win…  39.99
1  Anno 2070, the newest version of the award-win…  39.99
2  Anno 2070, the newest version of the award-win…  39.99
3  Anno 2070, the newest version of the award-win…  39.99
4  Anno 2070, the newest version of the award-win…  39.99

                        Categories  \
0  [['Video Games', 'PC', 'Games']]
1  [['Video Games', 'PC', 'Games']]
2  [['Video Games', 'PC', 'Games']]
3  [['Video Games', 'PC', 'Games']]
4  [['Video Games', 'PC', 'Games']]

                                          ReviewText RatingClass  ReviewDate  \
0  but when you do it's great. This game is a bit…    positive  2015-10-17
1  But in spite of that it was fun, I liked it I …    positive  2015-07-27
2                              Three Stars ok game.    positive  2015-02-23
3  Two Stars found the game a bit too complicated…    negative  2015-02-20
4  love this game great game, I love it and have …    positive  2014-12-25

                                          CleanText
0               great game bite hard get hang great
1  spite fun like play alright steam bite trouble…
2                                three star ok game
3  two star find game bite complicate not expect …
4       love game great game love play since arrive
```

[4]:
```
Cleanreview_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 497240 entries, 0 to 497239
Data columns (total 11 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Rating              497240 non-null  int64
 1   ReviewerID          497240 non-null  object
 2   ProductID           497240 non-null  object
 3   ReviewerName        497131 non-null  object
 4   ProductDescription  287372 non-null  object
 5   Price               356582 non-null  float64
 6   Categories          359654 non-null  object
 7   ReviewText          497240 non-null  object
```

```
 8   RatingClass         497240 non-null  object
 9   ReviewDate          497240 non-null  object
 10  CleanText           497187 non-null  object
dtypes: float64(1), int64(1), object(9)
memory usage: 41.7+ MB
```

[3]:
```
#df['date'].dt.year
Cleanreview_df['ReviewDate'] = pd.to_datetime(Cleanreview_df['ReviewDate'])

Cleanreview_df['ReviewYear'] = Cleanreview_df['ReviewDate'].dt.year
Cleanreview_df.head()
```

[3]:

	Rating	ReviewerID	ProductID	ReviewerName	ProductDescription	Price	Categories	ReviewText	RatingClass	ReviewDate	CleanText	ReviewYear
0	5	A1HP7NVNPFMA4N	0700026657	Ambrosia075	Anno 2070, the newest version of the award-win…	39.99	[['Video Games', 'PC', 'Games']]	but when you do it's great. This game is a bit…	positive	2015-10-17	great game bite hard get hang great	2015
1	4	A1JGAP0185YJI6	0700026657	travis	Anno 2070, the newest version of the award-win…	39.99	[['Video Games', 'PC', 'Games']]	But in spite of that it was fun, I liked it I …	positive	2015-07-27	spite fun like play alright steam bite trouble…	2015
2	3	A1YJWEXHQBWK2B	0700026657	Vincent G. Mezera	Anno 2070, the newest version of the award-win…	39.99	[['Video Games', 'PC', 'Games']]	Three Stars ok game.	positive	2015-02-23	three star ok game	2015
3	2	A2204E1TH211HT	0700026657	Grandma KR	Anno 2070, the newest version of the award-win…	39.99	[['Video Games', 'PC', 'Games']]	Two Stars found the game a bit too complicated…	negative	2015-02-20	two star find game bite complicate not expect …	2015
4	5	A2RF5B5H74JLPE	0700026657	jon	Anno 2070, the newest version of the award-win…	39.99	[['Video Games', 'PC', 'Games']]	love this game great game, I love it and have …	positive	2014-12-25	love game great game love play since arrive	2014

[24]:
```
 classCounts=Cleanreview_df.groupby(['Rating','ReviewYear']).size()
print(classCounts)


#class_counts
class_counts.to_csv('ReviewRatingClass.csv')
```

```
Rating  ReviewYear
1       1999              3
        2000             99
        2001            260
        2002            394
        2003            326
                        …
5       2014          47867
        2015          62012
        2016          51143
        2017          26678
        2018           7942
Length: 100, dtype: int64
```

[15]:
```
rows=[3,4]
column=['ReviewText','CleanText','RatingClass']
Cleanreview_df.loc[rows,column]
```

[15]:

	ReviewText	CleanText	RatingClass
3	Two Stars found the game a bit too complicated…	two star find game bite complicate not expect …	negative
4	love this game great game, I love it and have …	love game great game love play since arrive	positive

[7]:
```
pip install imblearn
```

```
Requirement already satisfied: imblearn in /opt/conda/lib/python3.7/site-
packages (0.0)
Requirement already satisfied: imbalanced-learn in
/opt/conda/lib/python3.7/site-packages (from imblearn) (0.8.0)
Requirement already satisfied: joblib>=0.11 in /opt/conda/lib/python3.7/site-
packages (from imbalanced-learn->imblearn) (0.15.1)
Requirement already satisfied: numpy>=1.13.3 in /opt/conda/lib/python3.7/site-
packages (from imbalanced-learn->imblearn) (1.19.5)
Requirement already satisfied: scipy>=0.19.1 in /opt/conda/lib/python3.7/site-
packages (from imbalanced-learn->imblearn) (1.4.1)
Requirement already satisfied: scikit-learn>=0.24 in
/opt/conda/lib/python3.7/site-packages (from imbalanced-learn->imblearn)
(0.24.1)
```

```
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from scikit-learn>=0.24->imbalanced-
learn->imblearn) (2.1.0)
Note: you may need to restart the kernel to use updated packages.
```

[8]:
```
pip install catboost
```

```
Requirement already satisfied: catboost in /opt/conda/lib/python3.7/site-
packages (0.25)
Requirement already satisfied: numpy>=1.16.0 in /opt/conda/lib/python3.7/site-
packages (from catboost) (1.19.5)
Requirement already satisfied: graphviz in /opt/conda/lib/python3.7/site-
packages (from catboost) (0.16)
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.7/site-
packages (from catboost) (3.2.1)
Requirement already satisfied: plotly in /opt/conda/lib/python3.7/site-packages
(from catboost) (4.14.3)
Requirement already satisfied: pandas>=0.24.0 in /opt/conda/lib/python3.7/site-
packages (from catboost) (1.0.3)
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages
(from catboost) (1.4.1)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from catboost) (1.15.0)
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->catboost) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->catboost) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib->catboost) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->catboost) (2.4.7)
Requirement already satisfied: retrying>=1.3.3 in /opt/conda/lib/python3.7/site-
packages (from plotly->catboost) (1.3.3)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.24.0->catboost) (2020.1)
Note: you may need to restart the kernel to use updated packages.
```

[10]:
```
Cleanreview_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 497240 entries, 0 to 497239
Data columns (total 12 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Rating              497240 non-null  int64
 1   ReviewerID          497240 non-null  object
 2   ProductID           497240 non-null  object
 3   ReviewerName        497131 non-null  object
```

```
 4   ProductDescription  287372 non-null  object
 5   Price               356582 non-null  float64
 6   Categories          359654 non-null  object
 7   ReviewText          497240 non-null  object
 8   RatingClass         497240 non-null  object
 9   ReviewDate          497240 non-null  datetime64[ns]
 10  CleanText           497187 non-null  object
 11  ReviewYear          497240 non-null  int64
dtypes: datetime64[ns](1), float64(1), int64(2), object(8)
memory usage: 45.5+ MB
```

[4]:
```
Cleanreview_df.isnull().sum()

Cleanreview_df=Cleanreview_df.dropna(subset=['CleanText'])
```

[6]:
```
Cleanreview_df.isnull().sum()
```

[6]:
```
Rating                     0
ReviewerID                 0
ProductID                  0
ReviewerName             109
ProductDescription    209843
Price                 140639
Categories            137567
ReviewText                 0
RatingClass                0
ReviewDate                 0
CleanText                  0
ReviewYear                 0
dtype: int64
```

[32]:
```
contains_digit = any(map(str.isdigit, Cleanreview_df['RatingClass']))
print(contains_digit)
```

```
False
```

[13]:
```
pip install xgboost
```

```
Requirement already satisfied: xgboost in /opt/conda/lib/python3.7/site-packages
(1.3.3)
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages
(from xgboost) (1.4.1)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages
(from xgboost) (1.19.5)
Note: you may need to restart the kernel to use updated packages.
```

[14]:
```
pip install gensim
```

```
Requirement already satisfied: gensim in /opt/conda/lib/python3.7/site-packages
(4.0.0)
Requirement already satisfied: scipy>=0.18.1 in /opt/conda/lib/python3.7/site-
packages (from gensim) (1.4.1)
Requirement already satisfied: smart-open>=1.8.1 in
/opt/conda/lib/python3.7/site-packages (from gensim) (5.0.0)
Requirement already satisfied: numpy>=1.11.3 in /opt/conda/lib/python3.7/site-
packages (from gensim) (1.19.5)
Note: you may need to restart the kernel to use updated packages.
```

[7]:
```
#sampling with SMOTE
from imblearn.over_sampling import SMOTE
from collections import Counter
from matplotlib import pyplot
from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import cross_validate
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_predict
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import learning_curve
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.decomposition import TruncatedSVD
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer,␣
 ↪HashingVectorizer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from catboost import CatBoostClassifier, Pool
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from  sklearn.metrics import precision_recall_fscore_support
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import recall_score
from gensim.models import Word2Vec
from tqdm import tqdm
import xgboost as xgb
from xgboost import XGBClassifier
from sklearn.dummy import DummyClassifier
# define the dataset location
```

```
X = Cleanreview_df['CleanText']
y = Cleanreview_df['Rating']


from sklearn.feature_extraction.text import TfidfVectorizer
vec = TfidfVectorizer()
X = vec.fit_transform(X)


'''
from imblearn.over_sampling import SMOTE
sm = SMOTE(k_neighbors=1, random_state = 2)
X_train_res, y_train_res = sm.fit_sample(X, df.labels)
'''
```

[7]:
```
'\nfrom imblearn.over_sampling import SMOTE \nsm = SMOTE(k_neighbors=1,
random_state = 2) \nX_train_res, y_train_res = sm.fit_sample(X, df.labels) \n'
```

[12]:
```
#X.shape
y.shape
```

[12]:
```
(497187,)
```

[8]:
```
#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
↪random_state=42)
x_train, x_val, y_train, y_val = train_test_split(X, y,train_size=60000,
                                                  test_size = 20000,
                                                  random_state=12)
```

[17]:
```
#x_train.shape
y_train.shape
```

[17]:
```
(90000,)
```

[9]:
```
sm = SMOTE(random_state=12)
x_train_res, y_train_res = sm.fit_resample(x_train, y_train)
```

[12]:
```
x_test_res, y_test_res = sm.fit_resample(x_val, y_val)
```

[10]:
```
#Verifying oversampling
#y_test_res.value_counts()
y_train_res.value_counts()
```

[10]:
```
5    36281
4    36281
3    36281
```

```
2    36281
1    36281
Name: Rating, dtype: int64
```

[11]:
```
pip install itertools
```

```
ERROR: Could not find a version that satisfies the requirement itertools
(from versions: none)
ERROR: No matching distribution found for itertools
Note: you may need to restart the kernel to use updated packages.
```

[12]:
```
#Modelling
import itertools
# Confusion Matrix function
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title = 'Confusion matrix',
                          cmap = plt.cm.ocean):
    """
    Create a confusion matrix plot for 'good' and 'bad' rating values
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis = 1)[:, np.newaxis]
    plt.imshow(cm, interpolation = 'nearest', cmap = cmap)
    plt.title(title, fontsize = 20)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, fontsize = 20)
    plt.yticks(tick_marks, classes, fontsize = 20)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.

    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt), horizontalalignment = "center",
                 color = "white" if cm[i, j] < thresh else "black", fontsize =␣
↪40)

    plt.tight_layout()
    plt.ylabel('True Label', fontsize = 30)
    plt.xlabel('Predicted Label', fontsize = 30)

    return plt
```

[22]:
```
def disp_confusion_matrix(y_pred, model_name, vector = 'CounterVectorizing'):
    """
```

```
    Display confusion matrix for selected model with countVectorizer
    """
    cm = confusion_matrix(y_val, predictions)
    fig = plt.figure(figsize=(10, 10))
    plot = plot_confusion_matrix(cm, classes=[1,2,3,4,5], normalize=False,
                                 title = model_name + " " + 'with' + " " +␣
↪vector + " "+ '\nConfusion Matrix')
    plt.show()
```

[16]:
```
def modeling(Model, Xtrain = x_train_res, Xtest = x_val):
    """
    This function apply countVectorizer with machine learning algorithms.
    """

    # Instantiate the classifier: model
    model = Model

    # Fitting classifier to the Training set (all features)
    model.fit(x_train_res, y_train_res)

    global y_pred
    # Predicting the Test set results
    y_pred = model.predict(x_val)

    # Assign f1 score to a variable
    score = f1_score(y_val, y_pred, average = 'weighted')

    # Printing evaluation metric (f1-score)
    print("f1 score: {}".format(score))
```

[17]:
```
#Function call for Logistic Regression

logisticRegr = LogisticRegression()

logisticRegr.fit(x_train_res, y_train_res)

predictions = logisticRegr.predict(x_val)
```

[18]:
```
score = logisticRegr.score(x_val, y_val)
#score = f1_score(y_test_res, predictions, average = 'weighted')
print(score)
```

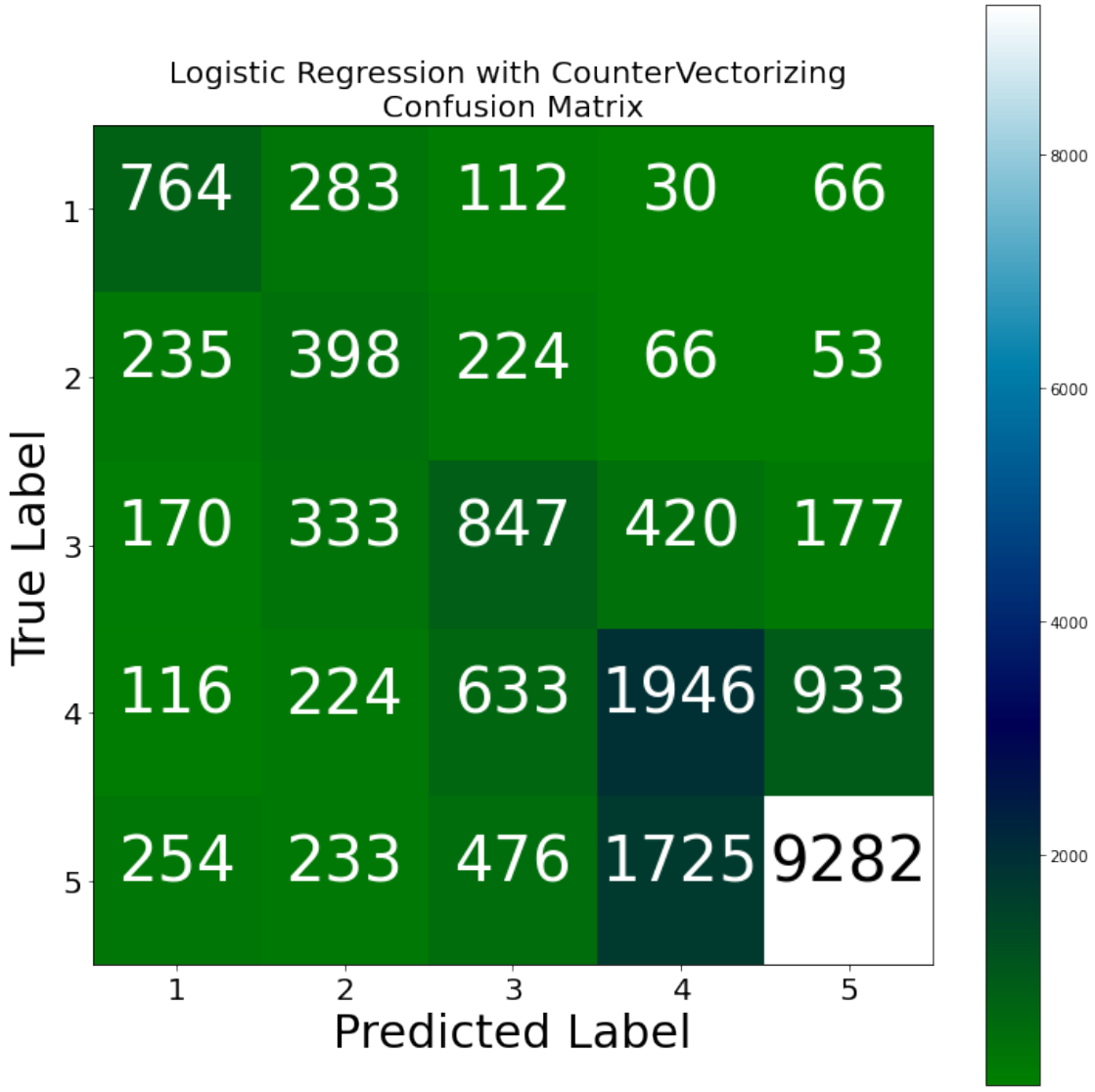
```
0.66185
```

[19]:
```
print(classification_report(y_val, predictions))
```

```
              precision    recall  f1-score   support
```

1	0.50	0.61	0.55	1255
2	0.27	0.41	0.33	976
3	0.37	0.44	0.40	1947
4	0.46	0.51	0.48	3852
5	0.88	0.78	0.83	11970
accuracy			0.66	20000
macro avg	0.50	0.55	0.52	20000
weighted avg	0.70	0.66	0.68	20000

[23]:
```
disp_confusion_matrix(predictions, "Logistic Regression")
```



[21]:
```
# Fuction to compute alpha value
def naive_bayes(X_train, y_train):

    alpha_values = np.arange(1, 500, 0.5)


    # empty list that will hold cv scores
    cv_scores = []

    # perform 10-fold cross validation
    for alpha in alpha_values:
        mnb = MultinomialNB(alpha = alpha)
        scores = cross_val_score(mnb, X_train, y_train, cv = 10, scoring =␣
↪'accuracy')
        cv_scores.append(scores.mean())

    # changing to misclassification error
    MSE = [1 - x for x in cv_scores]

    # determining best alpha
    optimal_alpha = alpha_values[MSE.index(min(MSE))]
    print('\nThe optimal number of alpha is %d.' % optimal_alpha)

    # plot misclassification error vs alpha
    plt.plot(alpha_values, MSE, marker = '*')

    #for xy in zip(alpha_values, np.round(MSE,3)):
        #plt.annotate('(%s, %s)' % xy, xy=xy, textcoords='data')
    plt.title("Misclassification Error vs alpha")
    plt.xlabel('value of alpha')
    plt.ylabel('Misclassification Error')
    plt.show()

    #print("the misclassification error for each value of alpha is : ", np.
↪round(MSE,3))
    return optimal_alpha
```

[23]:
```
#optimal_alpha_bow = naive_bayes(x_train_res, y_train_res)
```

[24]:
```
#Modelling using Naive Bayes
# instantiate learning model alpha = optimal_alpha

from sklearn.naive_bayes import MultinomialNB
nb_optimal =  MultinomialNB(alpha = 1.0)

# fitting the model
nb_optimal.fit(x_train_res, y_train_res)
```

```
# predict the response
```

[24]:
```
MultinomialNB()
```

[25]:
```
predictions = nb_optimal.predict(x_val)
```

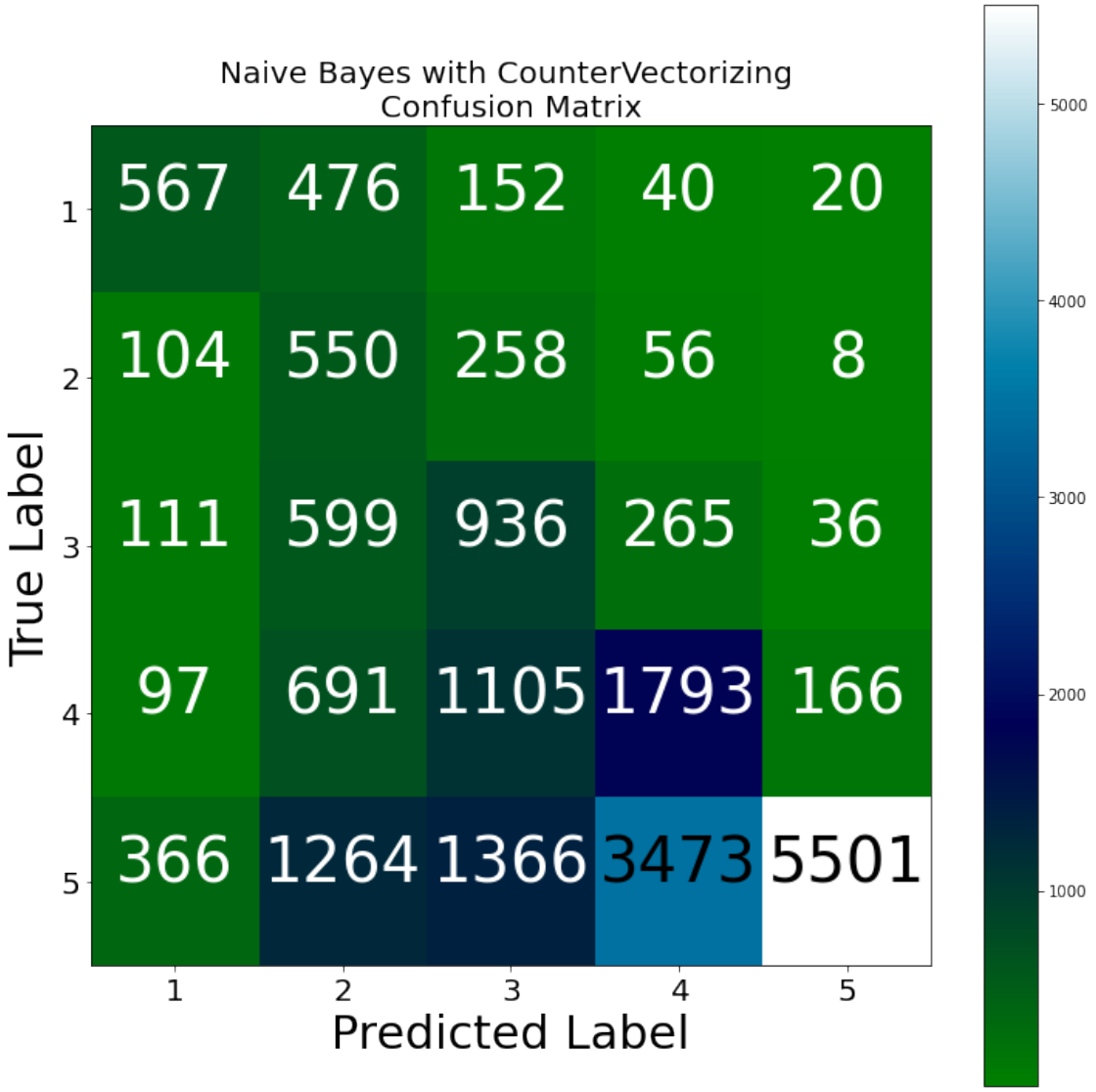
[26]:
```
print(classification_report(y_val, predictions))
```

```
              precision    recall  f1-score   support

           1       0.46      0.45      0.45      1255
           2       0.15      0.56      0.24       976
           3       0.25      0.48      0.32      1947
           4       0.32      0.47      0.38      3852
           5       0.96      0.46      0.62     11970

    accuracy                           0.47     20000
   macro avg       0.43      0.48      0.40     20000
weighted avg       0.70      0.47      0.52     20000
```

[28]:
```
disp_confusion_matrix(predictions, "Naive Bayes")
```



[30]:
```
pip install keras
```

```
Requirement already satisfied: keras in /opt/conda/lib/python3.7/site-packages
(2.4.3)
Requirement already satisfied: h5py in /opt/conda/lib/python3.7/site-packages
(from keras) (2.10.0)
Requirement already satisfied: pyyaml in /opt/conda/lib/python3.7/site-packages
(from keras) (5.3.1)
Requirement already satisfied: scipy>=0.14 in /opt/conda/lib/python3.7/site-
packages (from keras) (1.4.1)
Requirement already satisfied: numpy>=1.9.1 in /opt/conda/lib/python3.7/site-
packages (from keras) (1.19.5)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
```

```
(from h5py->keras) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

[32]:
```
pip install tensorflow
```

```
Requirement already satisfied: tensorflow in /opt/conda/lib/python3.7/site-
packages (2.4.1)
Requirement already satisfied: wheel~=0.35 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (0.36.2)
Requirement already satisfied: wrapt~=1.12.1 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (1.12.1)
Requirement already satisfied: tensorboard~=2.4 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (2.4.1)
Requirement already satisfied: google-pasta~=0.2 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: astunparse~=1.6.3 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: numpy~=1.19.2 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (1.19.5)
Requirement already satisfied: six~=1.15.0 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (1.15.0)
Requirement already satisfied: flatbuffers~=1.12.0 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (1.12)
Requirement already satisfied: h5py~=2.10.0 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (2.10.0)
Requirement already satisfied: tensorflow-estimator<2.5.0,>=2.4.0 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (2.4.0)
Requirement already satisfied: protobuf>=3.9.2 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (3.11.4)
Requirement already satisfied: opt-einsum~=3.3.0 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: gast==0.3.3 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (0.3.3)
Requirement already satisfied: typing-extensions~=3.7.4 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (3.7.4.2)
Requirement already satisfied: keras-preprocessing~=1.1.2 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: termcolor~=1.1.0 in
/opt/conda/lib/python3.7/site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: absl-py~=0.10 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (0.12.0)
Requirement already satisfied: grpcio~=1.32.0 in /opt/conda/lib/python3.7/site-
packages (from tensorflow) (1.32.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/lib/python3.7/site-packages (from tensorboard~=2.4->tensorflow)
(1.8.0)
Requirement already satisfied: werkzeug>=0.11.15 in
/opt/conda/lib/python3.7/site-packages (from tensorboard~=2.4->tensorflow)
```

```
(1.0.1)
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/lib/python3.7/site-
packages (from tensorboard~=2.4->tensorflow) (3.3.4)
Requirement already satisfied: google-auth<2,>=1.6.3 in
/opt/conda/lib/python3.7/site-packages (from tensorboard~=2.4->tensorflow)
(1.16.1)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/lib/python3.7/site-packages (from tensorboard~=2.4->tensorflow)
(2.23.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/lib/python3.7/site-packages (from tensorboard~=2.4->tensorflow)
(0.4.4)
Requirement already satisfied: setuptools>=41.0.0 in
/opt/conda/lib/python3.7/site-packages (from tensorboard~=2.4->tensorflow)
(46.1.3.post20200325)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
/opt/conda/lib/python3.7/site-packages (from
markdown>=2.6.8->tensorboard~=2.4->tensorflow) (1.6.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/lib/python3.7/site-packages (from google-
auth<2,>=1.6.3->tensorboard~=2.4->tensorflow) (0.2.8)
Requirement already satisfied: rsa<4.1,>=3.1.4 in /opt/conda/lib/python3.7/site-
packages (from google-auth<2,>=1.6.3->tensorboard~=2.4->tensorflow) (4.0)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from google-
auth<2,>=1.6.3->tensorboard~=2.4->tensorflow) (4.1.0)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-
packages (from requests<3,>=2.21.0->tensorboard~=2.4->tensorflow) (2.9)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/opt/conda/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard~=2.4->tensorflow) (1.25.9)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard~=2.4->tensorflow) (2020.4.5.2)
Requirement already satisfied: chardet<4,>=3.0.2 in
/opt/conda/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard~=2.4->tensorflow) (3.0.4)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard~=2.4->tensorflow) (1.3.0)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.7/site-
packages (from importlib-metadata; python_version <
"3.8"->markdown>=2.6.8->tensorboard~=2.4->tensorflow) (3.1.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/opt/conda/lib/python3.7/site-packages (from pyasn1-modules>=0.2.1->google-
auth<2,>=1.6.3->tensorboard~=2.4->tensorflow) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/lib/python3.7/site-
packages (from requests-oauthlib>=0.7.0->google-auth-
```

```
oauthlib<0.5,>=0.4.1->tensorboard~=2.4->tensorflow) (3.0.1)
Note: you may need to restart the kernel to use updated packages.
```

[33]:
```
pip install nltk
```

```
Requirement already satisfied: nltk in /opt/conda/lib/python3.7/site-packages
(3.5)
Requirement already satisfied: joblib in /opt/conda/lib/python3.7/site-packages
(from nltk) (0.15.1)
Requirement already satisfied: click in /opt/conda/lib/python3.7/site-packages
(from nltk) (7.1.2)
Requirement already satisfied: regex in /opt/conda/lib/python3.7/site-packages
(from nltk) (2021.3.17)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.7/site-packages
(from nltk) (4.45.0)
Note: you may need to restart the kernel to use updated packages.
```

[35]:
```
pip install plotly==4.14.3
```

```
Requirement already satisfied: plotly==4.14.3 in /opt/conda/lib/python3.7/site-
packages (4.14.3)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from plotly==4.14.3) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in /opt/conda/lib/python3.7/site-
packages (from plotly==4.14.3) (1.3.3)
Note: you may need to restart the kernel to use updated packages.
```

[36]:
```
pip install chart_studio
```

```
Requirement already satisfied: chart_studio in /opt/conda/lib/python3.7/site-
packages (1.1.0)
Requirement already satisfied: requests in /opt/conda/lib/python3.7/site-
packages (from chart_studio) (2.23.0)
Requirement already satisfied: plotly in /opt/conda/lib/python3.7/site-packages
(from chart_studio) (4.14.3)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from chart_studio) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in /opt/conda/lib/python3.7/site-
packages (from chart_studio) (1.3.3)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.7/site-packages (from requests->chart_studio)
(2020.4.5.2)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-
packages (from requests->chart_studio) (2.9)
Requirement already satisfied: chardet<4,>=3.0.2 in
/opt/conda/lib/python3.7/site-packages (from requests->chart_studio) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/opt/conda/lib/python3.7/site-packages (from requests->chart_studio) (1.25.9)
```

```
Note: you may need to restart the kernel to use updated packages.
```

[37]:
```
pip install cufflinks
```

```
Requirement already satisfied: cufflinks in /opt/conda/lib/python3.7/site-
packages (0.17.3)
Requirement already satisfied: pandas>=0.19.2 in /opt/conda/lib/python3.7/site-
packages (from cufflinks) (1.0.3)
Requirement already satisfied: six>=1.9.0 in /opt/conda/lib/python3.7/site-
packages (from cufflinks) (1.15.0)
Requirement already satisfied: colorlover>=0.2.1 in
/opt/conda/lib/python3.7/site-packages (from cufflinks) (0.3.0)
Requirement already satisfied: setuptools>=34.4.1 in
/opt/conda/lib/python3.7/site-packages (from cufflinks) (46.1.3.post20200325)
Requirement already satisfied: numpy>=1.9.2 in /opt/conda/lib/python3.7/site-
packages (from cufflinks) (1.19.5)
Requirement already satisfied: plotly>=4.1.1 in /opt/conda/lib/python3.7/site-
packages (from cufflinks) (4.14.3)
Requirement already satisfied: ipython>=5.3.0 in /opt/conda/lib/python3.7/site-
packages (from cufflinks) (7.14.0)
Requirement already satisfied: ipywidgets>=7.0.0 in
/opt/conda/lib/python3.7/site-packages (from cufflinks) (7.5.1)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.19.2->cufflinks) (2020.1)
Requirement already satisfied: python-dateutil>=2.6.1 in
/opt/conda/lib/python3.7/site-packages (from pandas>=0.19.2->cufflinks) (2.8.1)
Requirement already satisfied: retrying>=1.3.3 in /opt/conda/lib/python3.7/site-
packages (from plotly>=4.1.1->cufflinks) (1.3.3)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from ipython>=5.3.0->cufflinks) (3.0.5)
Requirement already satisfied: decorator in /opt/conda/lib/python3.7/site-
packages (from ipython>=5.3.0->cufflinks) (4.4.2)
Requirement already satisfied: pygments in /opt/conda/lib/python3.7/site-
packages (from ipython>=5.3.0->cufflinks) (2.6.1)
Requirement already satisfied: pexpect; sys_platform != "win32" in
/opt/conda/lib/python3.7/site-packages (from ipython>=5.3.0->cufflinks) (4.8.0)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.7/site-
packages (from ipython>=5.3.0->cufflinks) (0.7.5)
Requirement already satisfied: backcall in /opt/conda/lib/python3.7/site-
packages (from ipython>=5.3.0->cufflinks) (0.1.0)
Requirement already satisfied: jedi>=0.10 in /opt/conda/lib/python3.7/site-
packages (from ipython>=5.3.0->cufflinks) (0.17.0)
Requirement already satisfied: traitlets>=4.2 in /opt/conda/lib/python3.7/site-
packages (from ipython>=5.3.0->cufflinks) (4.3.3)
Requirement already satisfied: nbformat>=4.2.0 in /opt/conda/lib/python3.7/site-
packages (from ipywidgets>=7.0.0->cufflinks) (5.0.6)
Requirement already satisfied: ipykernel>=4.5.1 in
/opt/conda/lib/python3.7/site-packages (from ipywidgets>=7.0.0->cufflinks)
```

```
(5.3.0)
Requirement already satisfied: widgetsnbextension~=3.5.0 in
/opt/conda/lib/python3.7/site-packages (from ipywidgets>=7.0.0->cufflinks)
(3.5.1)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.7/site-packages
(from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=5.3.0->cufflinks)
(0.1.9)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.7/site-
packages (from pexpect; sys_platform != "win32"->ipython>=5.3.0->cufflinks)
(0.6.0)
Requirement already satisfied: parso>=0.7.0 in /opt/conda/lib/python3.7/site-
packages (from jedi>=0.10->ipython>=5.3.0->cufflinks) (0.7.0)
Requirement already satisfied: ipython-genutils in
/opt/conda/lib/python3.7/site-packages (from
traitlets>=4.2->ipython>=5.3.0->cufflinks) (0.2.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in
/opt/conda/lib/python3.7/site-packages (from
nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (3.2.0)
Requirement already satisfied: jupyter-core in /opt/conda/lib/python3.7/site-
packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (4.6.3)
Requirement already satisfied: jupyter-client in /opt/conda/lib/python3.7/site-
packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (6.1.3)
Requirement already satisfied: tornado>=4.2 in /opt/conda/lib/python3.7/site-
packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (6.0.4)
Requirement already satisfied: notebook>=4.4.1 in /opt/conda/lib/python3.7/site-
packages (from widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (6.0.3)
Requirement already satisfied: pyrsistent>=0.14.0 in
/opt/conda/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (0.16.0)
Requirement already satisfied: attrs>=17.4.0 in /opt/conda/lib/python3.7/site-
packages (from
jsonschema!=2.5.0,>=2.4->nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (19.3.0)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
/opt/conda/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks) (1.6.0)
Requirement already satisfied: pyzmq>=13 in /opt/conda/lib/python3.7/site-
packages (from jupyter-client->ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks)
(19.0.1)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.7/site-packages
(from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks)
(2.11.2)
Requirement already satisfied: Send2Trash in /opt/conda/lib/python3.7/site-
packages (from
notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks)
(1.5.0)
Requirement already satisfied: nbconvert in /opt/conda/lib/python3.7/site-
packages (from
notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks)
```

```
(5.6.1)
Requirement already satisfied: terminado>=0.8.1 in
/opt/conda/lib/python3.7/site-packages (from
notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks)
(0.8.3)
Requirement already satisfied: prometheus-client in
/opt/conda/lib/python3.7/site-packages (from
notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks)
(0.8.0)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.7/site-
packages (from importlib-metadata; python_version <
"3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.2.0->ipywidgets>=7.0.0->cufflinks)
(3.1.0)
Requirement already satisfied: MarkupSafe>=0.23 in
/opt/conda/lib/python3.7/site-packages (from jinja2->notebook>=4.4.1->widgetsnbe
xtension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (1.1.1)
Requirement already satisfied: pandocfilters>=1.4.1 in
/opt/conda/lib/python3.7/site-packages (from nbconvert->notebook>=4.4.1->widgets
nbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (1.4.2)
Requirement already satisfied: mistune<2,>=0.8.1 in
/opt/conda/lib/python3.7/site-packages (from nbconvert->notebook>=4.4.1->widgets
nbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (0.8.4)
Requirement already satisfied: entrypoints>=0.2.2 in
/opt/conda/lib/python3.7/site-packages (from nbconvert->notebook>=4.4.1->widgets
nbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (0.3)
Requirement already satisfied: testpath in /opt/conda/lib/python3.7/site-
packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets
>=7.0.0->cufflinks) (0.4.4)
Requirement already satisfied: defusedxml in /opt/conda/lib/python3.7/site-
packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets
>=7.0.0->cufflinks) (0.6.0)
Requirement already satisfied: bleach in /opt/conda/lib/python3.7/site-packages
(from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->
cufflinks) (3.1.5)
Requirement already satisfied: webencodings in /opt/conda/lib/python3.7/site-
packages (from bleach->nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ip
ywidgets>=7.0.0->cufflinks) (0.5.1)
Requirement already satisfied: packaging in /opt/conda/lib/python3.7/site-
packages (from bleach->nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ip
ywidgets>=7.0.0->cufflinks) (20.4)
Requirement already satisfied: pyparsing>=2.0.2 in
/opt/conda/lib/python3.7/site-packages (from packaging->bleach->nbconvert->noteb
ook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->cufflinks) (2.4.7)
Note: you may need to restart the kernel to use updated packages.
```

[38]:
```
#LSTM modelling
import numpy as np
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from keras.callbacks import EarlyStopping
from keras.layers import Dropout
import re
from nltk.corpus import stopwords
from nltk import word_tokenize
STOPWORDS = set(stopwords.words('english'))
from bs4 import BeautifulSoup
import plotly.graph_objs as go
#import plotly.plotly as py
import chart_studio.plotly as py
import cufflinks
from IPython.core.interactiveshell import InteractiveShell
import plotly.figure_factory as ff
InteractiveShell.ast_node_interactivity = 'all'
from plotly.offline import iplot
cufflinks.go_offline()
cufflinks.set_config_file(world_readable=True, theme='pearl')
```

[39]:
```
# The maximum number of words to be used. (most frequent)

'''
X = Cleanreview_df['CleanText']
y = Cleanreview_df['Rating']
'''
MAX_NB_WORDS = 50000
# Max number of words in each Review.
MAX_SEQUENCE_LENGTH = 250
# This is fixed.
EMBEDDING_DIM = 100
#, lower=True
tokenizer = Tokenizer(num_words=MAX_NB_WORDS, filters='!"#$%&()*+,-./:;<=>?
 ↪@[\]^_`{|}~', lower=True
                      )
tokenizer.fit_on_texts(Cleanreview_df['CleanText'].values)
word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))
```

[39]:
```
"\nX = Cleanreview_df['CleanText']\ny = Cleanreview_df['Rating']\n"
```

```
Found 308309 unique tokens.
```

[40]:
```python
X = tokenizer.texts_to_sequences(Cleanreview_df['CleanText'].values)
X = pad_sequences(X, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data tensor:', X.shape)
```

```
Shape of data tensor: (497187, 250)
```

[41]:
```python
Y = pd.get_dummies(Cleanreview_df['Rating']).values
print('Shape of label tensor:', Y.shape)
```

```
Shape of label tensor: (497187, 5)
```

[13]:
```python
X[0]
```

[13]:
```
array([   0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    7,    1,   85,   91,    4, 1033,    7], dtype=int32)
```

[42]:
```python
#X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.10,␣
↪random_state = 42)
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 10000,␣
↪train_size=30000, random_state = 4)
print(X_train.shape,Y_train.shape)
print(X_test.shape,Y_test.shape)
```

```
(30000, 250) (30000, 5)
(10000, 250) (10000, 5)
```

[43]:
```
sm = SMOTE(random_state=12)
x_train_res, y_train_res = sm.fit_resample(X_train, Y_train)
#x_test_res, y_test_res = sm.fit_resample(X_test, Y_test)
```

[44]:
```
model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM, input_length=X.shape[1]))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(5, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',␣
↪metrics=['accuracy'])
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 250, 100)          5000000
_________________________________________________________________
spatial_dropout1d (SpatialDr (None, 250, 100)          0
_________________________________________________________________
lstm (LSTM)                  (None, 100)               80400
_________________________________________________________________
dense (Dense)                (None, 5)                 505
=================================================================
Total params: 5,080,905
Trainable params: 5,080,905
Non-trainable params: 0
_________________________________________________________________
None
```

[45]:
```
epochs = 5
batch_size = 64

history = model.fit(X_train, Y_train, epochs=epochs,␣
↪batch_size=batch_size,validation_split=0.
↪1,callbacks=[EarlyStopping(monitor='val_loss', patience=3, min_delta=0.
↪0001)])
```

```
Epoch 1/5
422/422 [==============================] - 233s 543ms/step - loss: 1.0588 -
accuracy: 0.6145 - val_loss: 0.7661 - val_accuracy: 0.6927
Epoch 2/5
422/422 [==============================] - 230s 546ms/step - loss: 0.6868 -
accuracy: 0.7310 - val_loss: 0.7404 - val_accuracy: 0.7083
Epoch 3/5
422/422 [==============================] - 228s 540ms/step - loss: 0.5779 -
```

```
accuracy: 0.7710 - val_loss: 0.7893 - val_accuracy: 0.6953
Epoch 4/5
422/422 [==============================] - 228s 541ms/step - loss: 0.4815 -
accuracy: 0.8149 - val_loss: 0.8208 - val_accuracy: 0.6927
Epoch 5/5
422/422 [==============================] - 229s 544ms/step - loss: 0.4085 -
accuracy: 0.8482 - val_loss: 0.9103 - val_accuracy: 0.6850
```

[22]:
```
accr = model.evaluate(X_test,Y_test)
print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))
```

```
313/313 [==============================] - 19s 60ms/step - loss: 1.0325 -
accuracy: 0.6928
Test set
  Loss: 1.032
  Accuracy: 0.693
```

[46]:
```
from sklearn.metrics import classification_report

# predict
pred = model.predict(X_test, batch_size = 32)
#pred = np.argmax(predictions, axis=1)
# label
y_train = np.argmax(Y_test, axis=1)

#print(y_train.shape, pred.shape)
#print(y_train[:5], pred[:5])
```

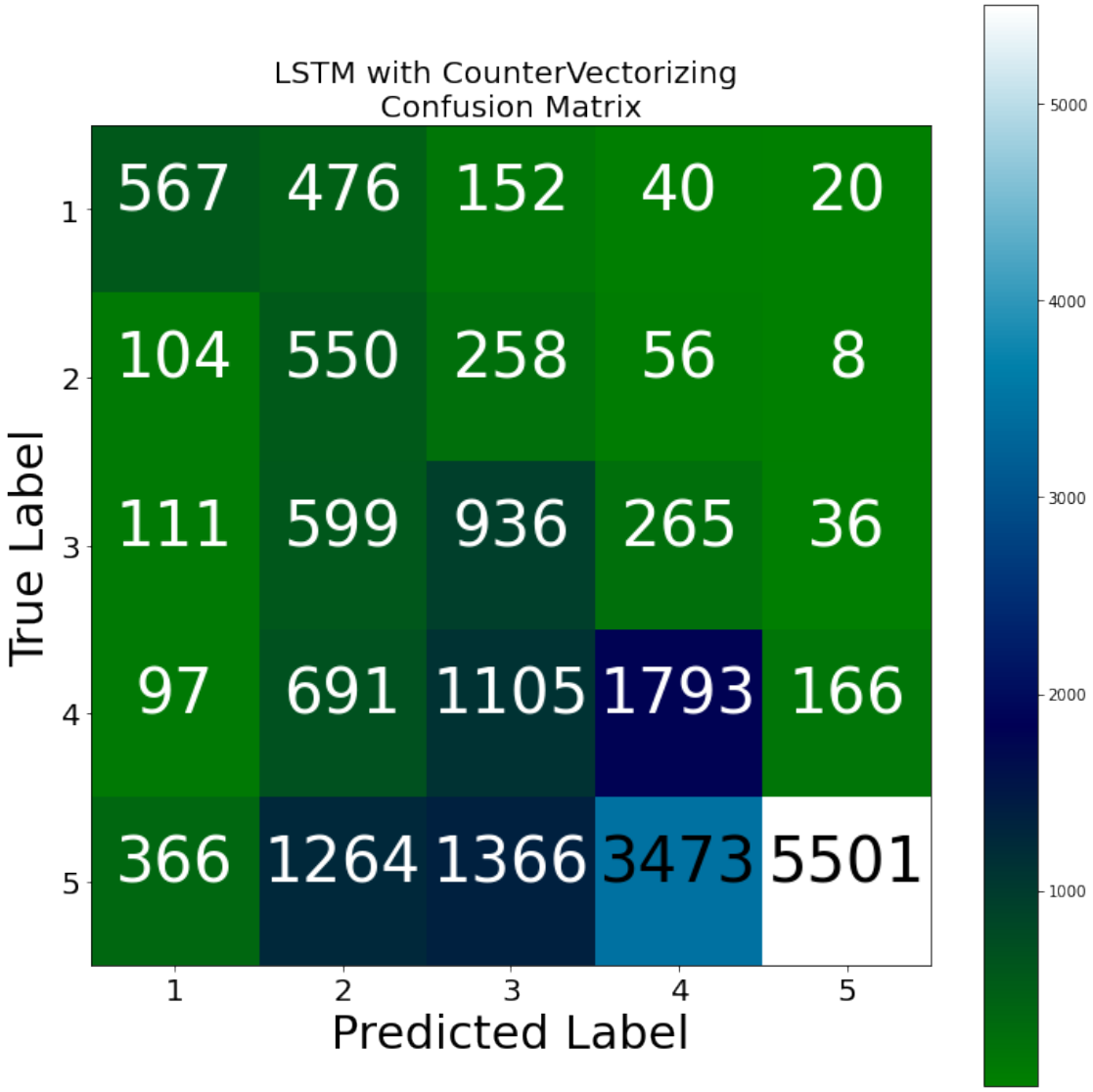
[47]:
```
print(classification_report(y_train, np.argmax(pred, axis = 1)))
```

```
              precision    recall  f1-score   support

           0       0.53      0.55      0.54       653
           1       0.32      0.28      0.30       501
           2       0.38      0.43      0.41       945
           3       0.51      0.35      0.41      1866
           4       0.82      0.89      0.85      6035

    accuracy                           0.69     10000
   macro avg       0.51      0.50      0.50     10000
weighted avg       0.67      0.69      0.68     10000
```

[48]:
```
disp_confusion_matrix(pred, "LSTM")
```



[49]:
```
#binomial predictions flow
X_bin = Cleanreview_df['CleanText']
y_bin = Cleanreview_df['RatingClass']


from sklearn.feature_extraction.text import TfidfVectorizer
vec = TfidfVectorizer()
X_bin = vec.fit_transform(X_bin)

#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
 ↪random_state=42)
x_train_bin, x_val_bin, y_train_bin, y_val_bin = train_test_split(X_bin,␣
 ↪y_bin,train_size=30000,
```

```
                                                    test_size = 10000,
                                                    random_state=12)

'''
from imblearn.over_sampling import SMOTE
sm = SMOTE(k_neighbors=1, random_state = 2)
X_train_res, y_train_res = sm.fit_sample(X, df.labels)
'''

sm = SMOTE(random_state=12)
x_train_res, y_train_res = sm.fit_resample(x_train_bin, y_train_bin)
```

[49]:
```
'\nfrom imblearn.over_sampling import SMOTE \nsm = SMOTE(k_neighbors=1,
random_state = 2) \nX_train_res, y_train_res = sm.fit_sample(X, df.labels) \n'
```

[34]:
```
x_test_res, y_test_res = sm.fit_resample(x_val_bin, y_val_bin)
```

[50]:
```
#Modelling using Naive Bayes
# instantiate learning model alpha = optimal_alpha

from sklearn.naive_bayes import MultinomialNB
nb_optimal =  MultinomialNB(alpha = 1.0)

# fitting the model
nb_optimal.fit(x_train_res, y_train_res)

# predict the response
```

[50]:
```
MultinomialNB()
```

[52]:
```
predictions = nb_optimal.predict(x_val_bin)
```

[53]:
```
score = f1_score(y_val_bin, predictions, average = 'weighted')
print(score)
```

```
0.7946178660151261
```

[54]:
```
print(classification_report(y_val_bin, predictions))
```

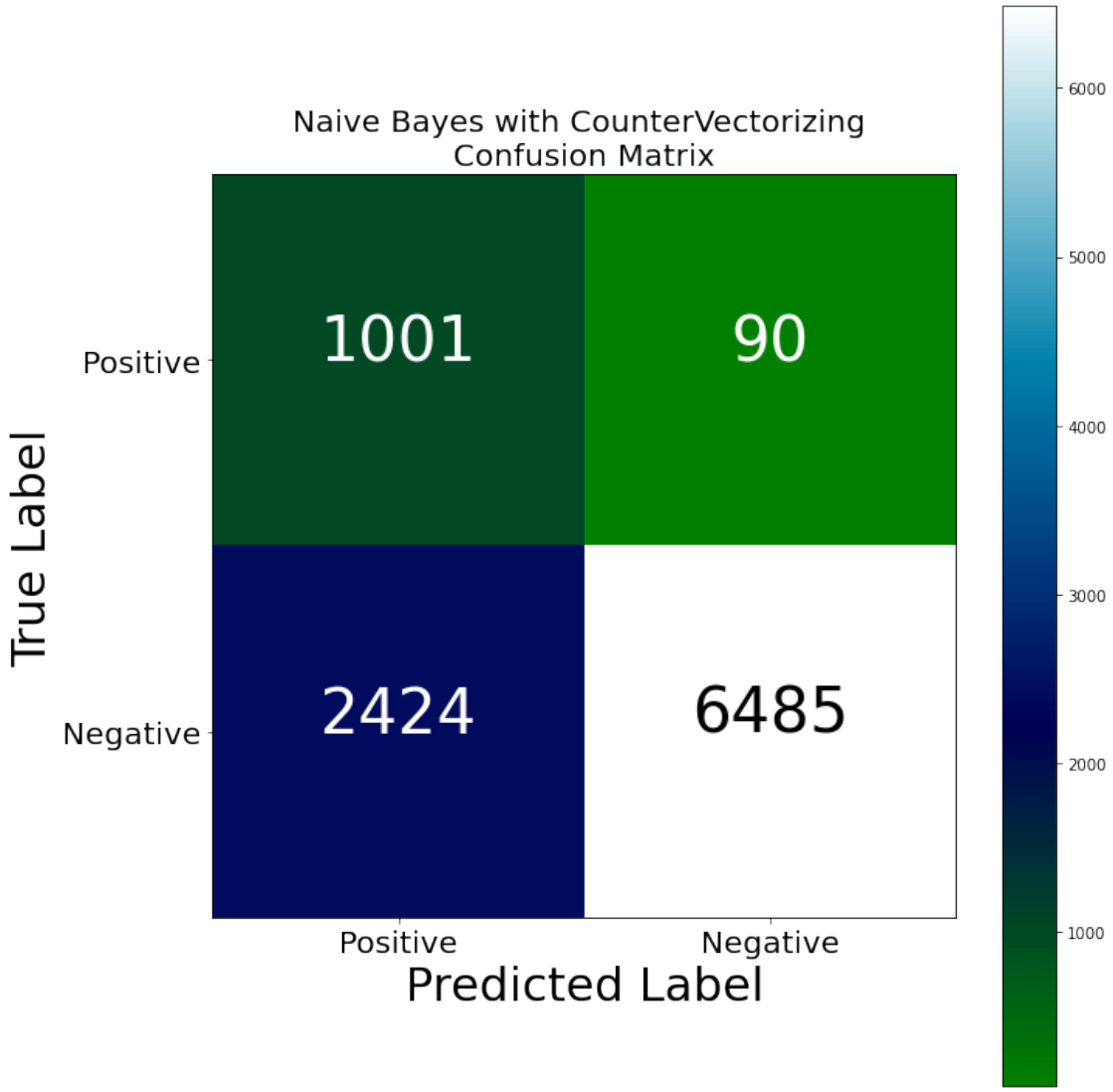
```
              precision    recall  f1-score   support

    negative       0.29      0.92      0.44      1091
    positive       0.99      0.73      0.84      8909

    accuracy                           0.75     10000
   macro avg       0.64      0.82      0.64     10000
weighted avg       0.91      0.75      0.79     10000
```

[56]:
```
def disp_confusion_matrix_bin(y_pred, model_name, vector =␣
↪'CounterVectorizing'):
    """
    Display confusion matrix for selected model with countVectorizer
    """
    cm = confusion_matrix(y_val_bin, predictions)
    fig = plt.figure(figsize=(10, 10))
    plot = plot_confusion_matrix(cm, classes=['Positive','Negative'],␣
↪normalize=False,
                                 title = model_name + " " + 'with' + " " +␣
↪vector + " "+ '\nConfusion Matrix')
    plt.show()
```

[57]:
```
disp_confusion_matrix_bin(predictions, "Naive Bayes")
```



[58]:
```
#Function call for Logistic Regression using binomial class

logisticRegr = LogisticRegression()

logisticRegr.fit(x_train_res, y_train_res)

predictions = logisticRegr.predict(x_val_bin)
```

[58]:
```
LogisticRegression()
```

[59]:
```
score = logisticRegr.score(x_val_bin, y_val_bin)
#score = f1_score(y_test_res, predictions, average = 'weighted')
print(score)
```

```
0.91
```

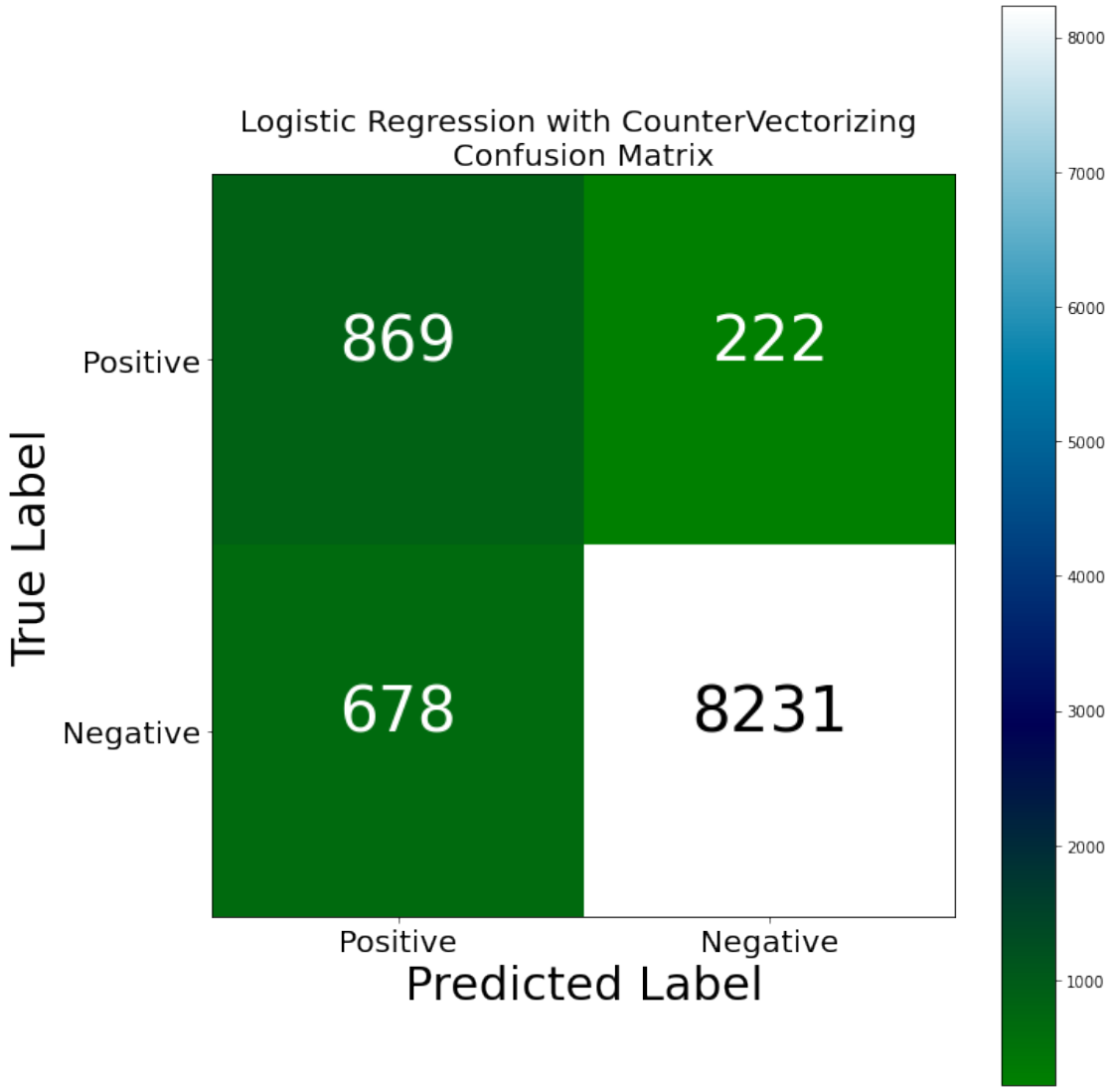
[61]:
```
print(classification_report(y_val_bin, predictions))
```

```
              precision    recall  f1-score   support

    negative       0.56      0.80      0.66      1091
    positive       0.97      0.92      0.95      8909

    accuracy                           0.91     10000
   macro avg       0.77      0.86      0.80     10000
weighted avg       0.93      0.91      0.92     10000
```

[62]:
```
disp_confusion_matrix_bin(predictions, "Logistic Regression")
```



[63]:
```
#LSTM using binomial class
```

[64]:
```
#sampling with SMOTE
from imblearn.over_sampling import SMOTE
from collections import Counter
from matplotlib import pyplot
from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import cross_validate
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_predict
```

```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import learning_curve
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.decomposition import TruncatedSVD
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer,␣
 ↪HashingVectorizer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from catboost import CatBoostClassifier, Pool
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from  sklearn.metrics import precision_recall_fscore_support
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import recall_score
from gensim.models import Word2Vec
from tqdm import tqdm
import xgboost as xgb
from xgboost import XGBClassifier
from sklearn.dummy import DummyClassifier
# define the dataset location
```

[65]:
```
# The maximum number of words to be used. (most frequent)

'''
X = Cleanreview_df['CleanText']
y = Cleanreview_df['Rating']
'''
MAX_NB_WORDS = 50000
# Max number of words in each Review.
MAX_SEQUENCE_LENGTH = 250
# This is fixed.
EMBEDDING_DIM = 100
#, lower=True
tokenizer = Tokenizer(num_words=MAX_NB_WORDS, filters='!"#$%&()*+,-./:;<=>?
 ↪@[\]^_`{|}~', lower=True
                      )
tokenizer.fit_on_texts(Cleanreview_df['CleanText'].values)
word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))
```

[65]:
```
"\nX = Cleanreview_df['CleanText']\ny = Cleanreview_df['Rating']\n"
```

```
Found 308309 unique tokens.
```

[66]:
```
X = tokenizer.texts_to_sequences(Cleanreview_df['CleanText'].values)
X = pad_sequences(X, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data tensor:', X.shape)
```

```
Shape of data tensor: (497187, 250)
```

[67]:
```
Y = pd.get_dummies(Cleanreview_df['RatingClass']).values
print('Shape of label tensor:', Y.shape)
```

```
Shape of label tensor: (497187, 2)
```

[69]:
```
#X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.10,␣
 ↪random_state = 42)
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 10000,␣
 ↪train_size=30000, random_state = 4)
print(X_train.shape,Y_train.shape)
print(X_test.shape,Y_test.shape)
```

```
(30000, 250) (30000, 2)
(10000, 250) (10000, 2)
```

[70]:
```
sm = SMOTE(random_state=12)
x_train_res, y_train_res = sm.fit_resample(X_train, Y_train)
```

[ ]:
```

```

[19]:
```
x_test_res, y_test_res = sm.fit_resample(X_test, Y_test)
```

[24]:
```
print(x_train_res.shape[1])
```

```
250
```

[ ]:
```
#y_train_res = np.asarray(train_labels).astype('float32').reshape((-1,1))
#y_test = np.asarray(test_labels).astype('float32').reshape((-1,1))
```

[71]:
```
model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM, input_length=X_train.shape[1]))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(2, activation='sigmoid'))
#model.compile(loss='sparse_categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
model.compile(loss='binary_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
print(model.summary())
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 250, 100)          5000000
_________________________________________________________________
spatial_dropout1d_1 (Spatial (None, 250, 100)          0
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               80400
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 202
=================================================================
Total params: 5,080,602
Trainable params: 5,080,602
Non-trainable params: 0
_________________________________________________________________
None
```

[72]:
```
epochs = 5
batch_size = 64

history = model.fit(X_train, Y_train, epochs=epochs,␣
↪batch_size=batch_size,validation_split=0.
↪1,callbacks=[EarlyStopping(monitor='val_loss', patience=3, min_delta=0.
↪0001)])
```

```
Epoch 1/5
422/422 [==============================] - 230s 538ms/step - loss: 0.3356 -
accuracy: 0.8889 - val_loss: 0.2228 - val_accuracy: 0.9093
Epoch 2/5
422/422 [==============================] - 228s 542ms/step - loss: 0.1718 -
accuracy: 0.9319 - val_loss: 0.1924 - val_accuracy: 0.9213
Epoch 3/5
422/422 [==============================] - 230s 544ms/step - loss: 0.1155 -
accuracy: 0.9571 - val_loss: 0.1831 - val_accuracy: 0.9217
Epoch 4/5
422/422 [==============================] - 228s 541ms/step - loss: 0.0871 -
accuracy: 0.9705 - val_loss: 0.2540 - val_accuracy: 0.9283
Epoch 5/5
422/422 [==============================] - 229s 542ms/step - loss: 0.0590 -
accuracy: 0.9793 - val_loss: 0.2566 - val_accuracy: 0.9210
```

[73]:
```
from sklearn.metrics import classification_report

# predict
predictions = model.predict(X_test, batch_size = 32)
#pred = np.argmax(predictions, axis=1)
```

```
# label
y_train = np.argmax(Y_test, axis=1)
```

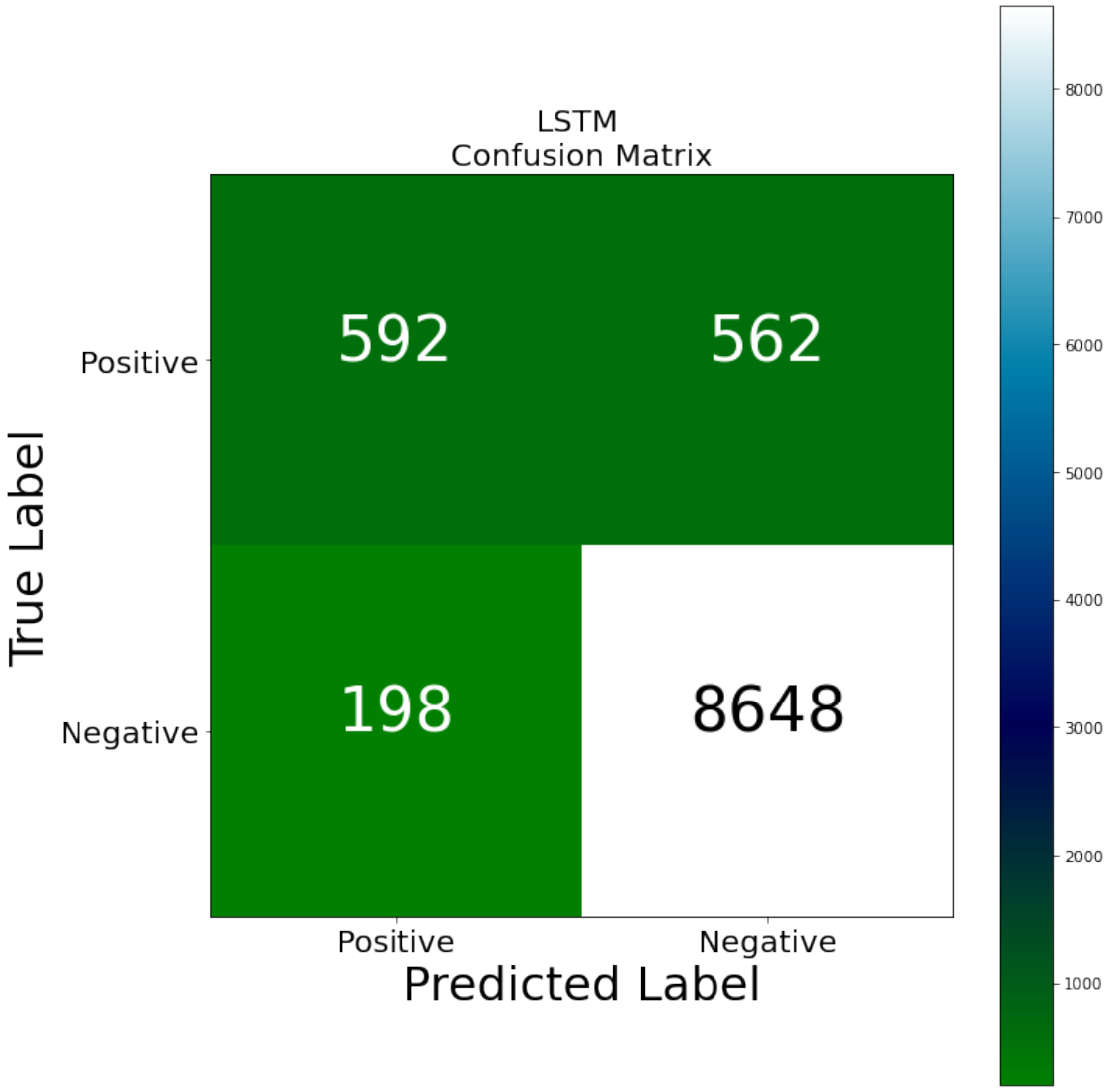
[74]:
```
print(classification_report(y_train, np.argmax(predictions, axis = 1)))
```

```
              precision    recall  f1-score   support

           0       0.75      0.51      0.61      1154
           1       0.94      0.98      0.96      8846

    accuracy                           0.92     10000
   macro avg       0.84      0.75      0.78     10000
weighted avg       0.92      0.92      0.92     10000
```

[75]:
```
    cm = confusion_matrix(y_train, np.argmax(predictions, axis = 1))
    fig = plt.figure(figsize=(10, 10))
    plot = plot_confusion_matrix(cm, classes=['Positive','Negative'],␣
 ↪normalize=False,
                                 title = "LSTM" + " "+ '\nConfusion Matrix')
    plt.show()
```



[58]:
```
#y_test_res.value_counts()
#y_train_res.value_counts()

unique_train, counts_train = np.unique(y_train_res, return_counts=True)
dict(zip(unique_train, counts_train))

unique_test, counts_test = np.unique(y_test_res, return_counts=True)
dict(zip(unique_test, counts_test))
```

[58]: {0: 26709, 1: 26709}

[58]: {0: 8846, 1: 8846}

[76]:
```
model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM, input_length=x_train_res.
 ↪shape[1]))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(2, activation='softmax'))
#model.compile(loss='sparse_categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
print(model.summary())
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 250, 100)          5000000
_________________________________________________________________
spatial_dropout1d_2 (Spatial (None, 250, 100)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 100)               80400
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 202
=================================================================
Total params: 5,080,602
Trainable params: 5,080,602
Non-trainable params: 0
_________________________________________________________________
None
```

[77]:
```
epochs = 5
batch_size = 64

history = model.fit(x_train_res, y_train_res, epochs=epochs,␣
 ↪batch_size=batch_size,validation_split=0.
 ↪1,callbacks=[EarlyStopping(monitor='val_loss', patience=3, min_delta=0.
 ↪0001)])
```

```
Epoch 1/5
752/752 [==============================] - 397s 525ms/step - loss: 0.5089 -
accuracy: 0.7373 - val_loss: 0.2077 - val_accuracy: 0.9347
Epoch 2/5
752/752 [==============================] - 395s 525ms/step - loss: 0.2361 -
accuracy: 0.9049 - val_loss: 0.1601 - val_accuracy: 0.9465
Epoch 3/5
752/752 [==============================] - 392s 521ms/step - loss: 0.1484 -
accuracy: 0.9446 - val_loss: 0.1854 - val_accuracy: 0.9410
```

```
Epoch 4/5
752/752 [==============================] - 392s 521ms/step - loss: 0.1079 -
accuracy: 0.9601 - val_loss: 0.1342 - val_accuracy: 0.9483
Epoch 5/5
752/752 [==============================] - 399s 531ms/step - loss: 0.0758 -
accuracy: 0.9722 - val_loss: 0.1260 - val_accuracy: 0.9549
```

[79]:
```
from sklearn.metrics import classification_report

# predict
predictions = model.predict(X_test, batch_size = 32)
#pred = np.argmax(predictions, axis=1)
# label
y_train = np.argmax(Y_test, axis=1)
```

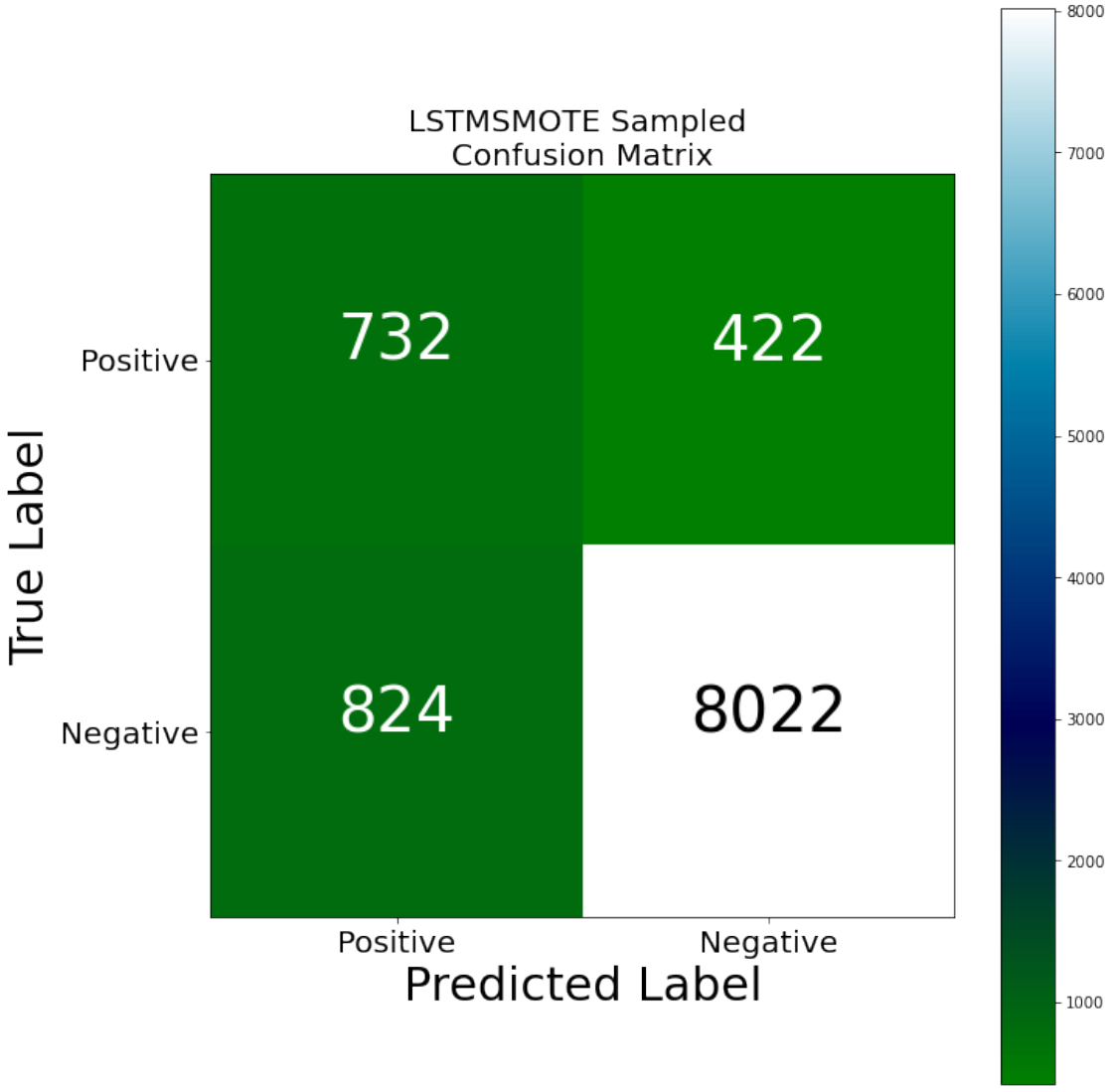
[80]:
```
print(classification_report(y_train, np.argmax(predictions, axis = 1)))
```

```
              precision    recall  f1-score   support

           0       0.47      0.63      0.54      1154
           1       0.95      0.91      0.93      8846

    accuracy                           0.88     10000
   macro avg       0.71      0.77      0.73     10000
weighted avg       0.89      0.88      0.88     10000
```

[81]:
```
cm = confusion_matrix(y_train, np.argmax(predictions, axis = 1))
fig = plt.figure(figsize=(10, 10))
plot = plot_confusion_matrix(cm, classes=['Positive','Negative'],␣
↪normalize=False,
                             title = "LSTM" +"SMOTE Sampled"+ " "+␣
↪'\nConfusion Matrix')
plt.show()
```


LSTMSMOTE Sampled
Confusion Matrix
732
422
824
8022
Positive
Negative
True Label
Positive
Negative
Predicted Label
8000
7000
6000
5000
4000
3000
2000
1000

[ ]: