

```
import numpy as np
from sklearn import preprocessing
import csv
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, f1_score
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.multiclass import OneVsRestClassifier
```

```
# Here is the file path to give
```

```
# This is cleaning data converting into matrices form
```

```
# Here stripped of whitespaces and converted into lower case
```

```
with open("/content/drive/MyDrive/Colab Notebooks/DF_project/Full_set.csv") as file:
    reader = csv.reader(file, delimiter=',')
    next(reader)
    count = 0
    set1 = set()
    X_tmp = []
    y_tmp = []
    for line in reader:
        y1 = []
        X_tmp.append(line[0].lower().strip())
        count = count + 1
        #new = line[1].split("|")
        #print(new)
        y1.append(line[1].lower().strip())
        set1.add(line[1].lower().strip())
        if line[2] != '':
            y1.append(line[2].lower().strip())

        y_tmp.append(y1)

    #if count == 10:
        #break
```

```
print("X_Samples = ", len(X_tmp))
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```
#print(len(y_tmp))
```

```
print(f"Labels = {set1}")
```

```
print("count_labels=", len(set1))
```

```
print(y_tmp)
```

```
X_Samples = 1000
y_Samples = 1000
Labels = {'ag', 'iot', 'public', 'cyber', 'util', 'dei', 'mobility', 'edu', 'govt data', 'connect', 'enviro', 'other'}
count_labels= 12
[['public', 'govt data'], ['public', 'mobility'], ['public', 'mobility'], ['mobility', 'enviro'], ['public', 'util'], ['mobility', 'con
```

```
# Now splitting into train and test data
```

```
X_txt_train, X_txt_test, y_train_text, y_test_text = train_test_split(X_tmp, y_tmp, test_size=0.2, random_state=42)
```

```
#print(y_test_text)
```

```
#print(X_txt_test)
```

```
# converting into numpy arrays
```

```
X_train = np.array(X_txt_train)
```

```

X_test = np.array(X_txt_test)

from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import precision_score, recall_score, f1_score
import sklearn.metrics
from sklearn.multioutput import MultiOutputClassifier

lb = preprocessing.MultiLabelBinarizer(classes=('connect', 'dei', 'iot', 'cyber', 'enviro', 'util', 'edu', 'other', 'govt data', 'ag', 'publi
y_train = lb.fit_transform(y_train_text)
y_test = lb.transform(y_test_text)
#print(y_test)
print(y_train.shape)
print(y_test.shape)
LogReg_pipeline = Pipeline([('vectorizer', CountVectorizer()),
('tfidf', TfidfTransformer()),
('clf', OneVsRestClassifier(LogisticRegression()))])
LogReg_pipeline.fit(X_train, y_train)
predicted = LogReg_pipeline.predict(X_test)

print(predicted)

print("Accuracy Score: ",accuracy_score(y_test, predicted))

precision = precision_score(predicted,y_test,average="macro")
recall = recall_score(predicted,y_test,average="macro")
f1 = f1_score(y_test,predicted, average="macro")

f1 = f1_score(y_test,predicted, average="macro")
print("Precision: {:.4f}".format(precision))
print("Recall: {:.4f}".format(recall))
print("F1: {:.4f}".format(f1))

"""
a =metrics.f1_score(y_test, y_pred, average='weighted', labels=np.unique(y_pred)

```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```

(800, 12)
(200, 12)
[[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]]
Accuracy Score: 0.025
Precision: 0.0195
Recall: 0.1278
F1: 0.0324
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: Undefined
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: Undefined
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1580: Undefined
_warn_prf(average, "true nor predicted", "F-score is", len(true_sum))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1580: Undefined
_warn_prf(average, "true nor predicted", "F-score is", len(true_sum))
'\na =metrics.f1_score(y_test, y_pred, average='weighted', labels=np.unique(y_pred)\np
rint(a)\n'

```

✓ 0s completed at 3:23 PM

● ✕

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)