# test documentation

## test Documentation

- unpickle_and_test

built with pdoc

## test

View Source

```
__authors__ = 'Abdullah + Vinayak'

import pickle
import pandas as pd
import time
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
def unpickle_and_test(name,X_test,y_test):
    """
    This method uses the pickled classifiers wich is RandomForest,MNB,SVM,LR  in dir '../mod
    an generate the predict result with X_test and  y_test.classification_report also will b

    :param name:string use for model(classifier) name
    :param X_test:the data set with Test_Features the 25 %
    :param y_test:the data set with Test_labels the 25%
    :return:
    """
    #open the pickled models and use it to predict to labels
    loaded_obj = None
    with open('../models/'+name+'.pickle','rb') as f:
        loaded_obj = pickle.load(f)
    assert loaded_obj is not None
    y_pred = loaded_obj.predict(X_test)
    print(classification_report(y_pred,y_test))
    print(accuracy_score(y_pred,y_test))
```

```
if __name__ == '__main__':
    """
      main method to use the split test data set and send it to def unpickle_and_test metho
      the pickled trained models, to unpickle the models and use the trained models to try a
      data and lastly generate the classification_report for the performance of the models
      """

    # Load data
    X_test = pd.read_csv(r'../Data/Test_Features.csv', index_col=0)
    y_test = pd.read_csv(r'../Data/Test_Labels.csv', index_col=0)
    classifiers = ['MNB','LR','RF' ,'SVM','MLP']
    for x in classifiers:
        start = time.time()
        print(x)
        unpickle_and_test(x,X_test,y_test)
        print((time.time()-start) , 'sec')
```

# def unpickle_and_test(name, X_test, y_test):

View Source

```
def unpickle_and_test(name,X_test,y_test):
    """
    This method uses the pickled classifiers wich is RandomForest,MNB,SVM,LR  in dir '../mod
    an generate the predict result with X_test and  y_test.classification_report also will b

    :param name:string use for model(classifier) name
    :param X_test:the data set with Test_Features the 25 %
    :param y_test:the data set with Test_labels the 25%
    :return:
    """
    #open the pickled models and use it to predict to labels
    loaded_obj = None
    with open('../models/'+name+'.pickle','rb') as f:
        loaded_obj = pickle.load(f)
    assert loaded_obj is not None
    y_pred = loaded_obj.predict(X_test)
    print(classification_report(y_pred,y_test))
    print(accuracy_score(y_pred,y_test))
```

This method uses the pickled classifiers wich is RandomForest,MNB,SVM,LR in dir '../models..' an generate the predict result with X_test and y_test.classification_report also will be generated

:param name:string use for model(classifier) name :param X_test:the data set with Test_Features the 25 % :param y_test:the data set with Test_labels the 25% :return: