

Kaggle submission script

December 17, 2018

```
In [5]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-py
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the fi

# Any results you write to the current directory are saved as output.
from sklearn import svm #support vector machines
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier

def clean(s) :
    clean_s = s.replace('-', ' ') # treat low-fat and low fat as the same thing
    clean_s = ''.join([c for c in clean_s if (c.isalpha() or c == ' ')]) # drop numbers
    return clean_s

#### First we read in the training data
data = pd.read_json('../input/train.json')
recipie_list_list = data.ingredients.values.tolist()
recipie_string_list = [clean(" ".join(ing)) for ing in recipie_list_list]
vectorizer = CountVectorizer(min_df = 3)
X_train = vectorizer.fit_transform(recipie_string_list)
y_train = data.cuisine.values

del data, recipie_list_list, recipie_string_list

#### Then we read in the test data
data = pd.read_json('../input/test.json')
recipie_list_list = data.ingredients.values.tolist()
recipie_string_list = [clean(" ".join(ing)) for ing in recipie_list_list]
```

```

X_test = vectorizer.transform(recipie_string_list)
test_ids = data.id.values.tolist()
del data, recipie_list_list, recipie_string_list

#####
#####
## The classifier goes here
#####
#####
logistic_clf = LogisticRegression(solver='lbfgs', multi_class='multinomial', C = 1)
forrest_clf = RandomForestClassifier(n_estimators = 100, max_depth = None)
mlp_clf = MLPClassifier(hidden_layer_sizes = (1000), alpha = 0.01, max_iter = 10)
clf = VotingClassifier(estimators=[('forrest', forrest_clf), ('mlp', mlp_clf), ('logistic', logistic_clf)],
                        voting='soft')

#####
#####
## train classifier and predict
clf.fit(X_train,y_train)
predictions = clf.predict(X_test)

#####
#####
## Write prediction as output file
#####
#####
idpreds = zip(test_ids, predictions)
file = open('sub.csv', 'w')
file.write('id,cuisine\n')
for t in idpreds :
    file.write(str(t[0])+',' +t[1]+'\\n')

```

<zip object at 0x000001C571393588>