In [15]:
```
import json
import numpy as np
from pprint import pprint

with open('atistest.json') as f:
    testa = json.load(f)

with open('atistrain.json') as f:
    traina = json.load(f)

with open('atisdev.json') as f:
    deva = json.load(f)
```

In [18]:
```
pprint(testa['body'][0])
```

```
{'intent': 'flight',
 'text': 'i would like to find a flight from charlotte to las vegas tha
t makes '
         'a stop in st. louis'}
```

In [20]:
```
testd = testa['body']
traind = traina['body']
devd = deva['body']

#given a dictionary of text and intent, return the text in the list for
 tfidf
def getCorp(dataset):
    result = []
    for sample in dataset:
        result.append(sample['text'])
    return result

traint = getCorp(traind)
testt = getCorp(testd)
devt = getCorp(devd)
```

In [49]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()
trainX = vectorizer.fit_transform(traint)
testX = vectorizer.transform(testt)
devX = vectorizer.transform(devt)
#devX = getX(devt)
```

In [50]:
```
def getLabel(dataset):
    result = []
    for exp in dataset:
        result.append(exp['intent'])

    return result

testy = getLabel(testd)
devy = getLabel(devd)
trainy = getLabel(traind)
```

In [54]:
```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()
clf.fit(trainX, trainy)

y_predtest = clf.predict(testX)
y_preddev = clf.predict(devX)
```

In [72]:
```
#get accuracy on the test set
(testy == y_predtest).sum()/len(testd)
#(devy == y_preddev).sum()/len(devd)
```

Out[72]:
```
0.793952967525196
```

In [73]:
```
from sklearn.metrics import f1_score
m = f1_score(testy, y_predtest,  average='macro')
print(m)
```

```
0.19801365329004605
```

In [66]:
```
classlabs = list(set(testy.copy()))
print(len(classlabs))
```

```
20
```

In [74]:
```
for label in classlabs:
    print(label, f1_score(testy, y_predtest, label, average='macro'))
```

```
ground_fare 0.14642878612947982
quantity 0.2727681523566062
flight+airline 0.1783537573200327
aircraft 0.20133958092803475
distance 0.20133958092803475
airline 0.2301023782034683
ground_service 0.2420603210265964
flight 0.4161587670800763
meal 0.3550601142370219
flight_no 0.17896851638047534
flight_time 0.14642878612947982
airfare+flight 0.1783537573200327
airport 0.2301023782034683
capacity 0.20133958092803475
airfare 0.23010237820346827
flight_no+airline 0.14687956485179163
city 0.3550601142370219
day_name 0.20133958092803475
abbreviation 0.20807938354003816
flight+airfare 0.1783537573200327
```

In [76]:
```
from sklearn.metrics import precision_score
for label in classlabs:
    print(label, precision_score(testy, y_predtest, label, average='macr
o'))
```

```
ground_fare 0.2526315789473684
quantity 0.40986842105263155
flight+airline 0.25540461000987313
aircraft 0.34736842105263155
distance 0.34736842105263155
airline 0.39699248120300756
ground_service 0.31743468519784307
flight 0.5959440900230374
meal 0.44473684210526315
flight_no 0.30877192982456136
flight_time 0.2526315789473684
airfare+flight 0.25540461000987313
airport 0.3969924812030075
capacity 0.34736842105263155
airfare 0.3969924812030075
flight_no+airline 0.21033320824342497
city 0.44473684210526315
day_name 0.34736842105263155
abbreviation 0.2979720450115187
flight+airfare 0.25540461000987313
```

In [77]:
```
from sklearn.metrics import recall_score
for label in classlabs:
    print(label, recall_score(testy, y_predtest, label, average='macro'
))
```

```
ground_fare 0.12223007587840123
quantity 0.251399687666135
flight+airline 0.1673534682986715
aircraft 0.1680663543328017
distance 0.1680663543328017
airline 0.19207583352320196
ground_service 0.23282965877486195
flight 0.39049142603023346
meal 0.30981691919191917
flight_no 0.1493923149624904
flight_time 0.12223007587840123
airfare+flight 0.1673534682986715
airport 0.19207583352320196
capacity 0.1680663543328017
airfare 0.19207583352320196
flight_no+airline 0.1378205033047883
city 0.30981691919191917
day_name 0.1680663543328017
abbreviation 0.19524571301511676
flight+airfare 0.1673534682986715
```

In [ ]: