Задание 3:

np.random.seed(76543)

lda2 = models.ldamodel.LdaModel(corpus2, num\_topics=40, iterations=5)

kohs = lda.top\_topics(corpus)

kohs2 = lda2.top\_topics(corpus2)

def avr\_koh(kohs):

s = 0

for i in xrange(len(kohs)):

s += kohs[i][1]

return s / len(kohs)

print avr\_koh(kohs), avr\_koh(kohs2)

>>> -655.217606861 -753.838235527

np.random.seed(76543)

lda = models.LdaModel(corpus=corpus, id2word = dictionary, num\_topics=40, passes = 5)

tupleTopics = lda.show\_topics(num\_topics=40, num\_words=10, log=False, formatted=False)

import copy

from sklearn.ensemble import RandomForestClassifier

from sklearn.cross\_validation import cross\_val\_score

import numpy as np

import pandas as pd

np.random.seed(76543)

dictionary2 = copy.deepcopy(dictionary)

corpus2 = [dictionary2.doc2bow(text) for text in texts]

ldamodel2 = models.ldamodel.LdaModel(corpus2, id2word=dictionary2, num\_topics=40, passes=5)

number\_of\_topics = 40

documents\_to\_topics = np.ndarray(shape=(len(texts), number\_of\_topics))

for text\_index, current\_text in enumerate(texts):

current\_text\_bag\_of\_words = dictionary2.doc2bow(current\_text)

current\_text\_topics\_probabilities = ldamodel2.get\_document\_topics(current\_text\_bag\_of\_words)

for current\_text\_topic\_probability\_tuple in current\_text\_topics\_probabilities:

documents\_to\_topics[text\_index, current\_text\_topic\_probability\_tuple[0]] = current\_text\_topic\_probability\_tuple[1]

cuisine\_names = []

for current\_recipe\_dict in recipes:

cuisine\_names.append(current\_recipe\_dict['cuisine'])

cuisine\_ids = pd.get\_dummies(cuisine\_names)

random\_forest\_classifier = RandomForestClassifier(n\_estimators = 100, random\_state=0)

cross\_validation\_scores = cross\_val\_score(estimator=random\_forest\_classifier, X=documents\_to\_topics, y=cuisine\_ids)

accuracy\_mean\_score = np.mean(cross\_validation\_scores)



Полученный результат: accuracy\_mean\_score = 0,405893297129.

lda = models.ldamodel.LdaModel(corpus, num\_topics=40, passes=5)

np.random.seed(76543)

lda.show\_topics(num\_topics=40, num\_words=10, formatted=True)

3.

np.random.seed(76543)

lda2 = gensim.models.ldamodel.LdaModel(corpus2, num\_topics=40, eval\_every=5)

np.random.seed(76543)

cogerlist1 = lda.top\_topics(corpus)

cogerlist2 = lda2.top\_topics(corpus2)

np.random.seed(76543)

coherence = 0

for i in cogerlist1:

coherence = coherence + i[1]

coherence = coherence/40

coherence2 = 0

for i in cogerlist2:

coherence2 = coherence2 + i[1]

coherence2 = coherence2/40

ответы получаются

-606.402294 -654.827038

где строка с seed это новый селл

4.

np.random.seed(76543)

lda3 = gensim.models.ldamodel.LdaModel(corpus2, num\_topics=40, passes=5, alpha=1, eval\_every=5)

np.random.seed(76543)

count\_model2 = 0

for j in range(len(corpus2)):

x = lda2.get\_document\_topics(corpus2[j], minimum\_probability=0.01)

count\_model2 = count\_model2 + len(x)

np.random.seed(76543)

count\_model3 = 0

for j in range(len(corpus2)):

x = lda3.get\_document\_topics(corpus2[j], minimum\_probability=0.01)

count\_model3 = count\_model3 + len(x)

ответы получаются

180817 1590960

dictionary2\_new = dictionary2.filter\_tokens(bad\_ids=spisok)

Сам метод filter\_tokens() только убирает токены, после него нужно использовать compactify()

Task 5:

# Импорт библиотек

from sklearn.ensemble import RandomForestClassifier

from sklearn.cross\_validation import cross\_val\_score

#Представляю массив в удобном для себя виде

X=np.array(ldamodel2.get\_document\_topics(corpus2))

Добавляю нулями пропущенные значения

for i in range(len(X)):

s = []

for z in range(len(X[i])):

s.append(X[i][z][0])

for j in range(40):

if j not in s:

X[i].append((j,0))

X[i].sort()

estimator = RandomForestClassifier(n\_estimators = 100)

score = cross\_val\_score(estimator,X,recipes, cv = 3)

1. Что бы ничего нулями заполнять не пришлось используйте аргумент minimum\_probability=0 для метода get\_document\_topics.
2. Что бы ваш код можно было прочитать используйте форматирование.
3. Дело в том, что на выходе вашего алгоритма заполнения получается одномерный массив из списков картежей. На вход RandomForestClassifier надо подавать двумерный массив из чисел (а не из списка картежей). Таким образом, либо переделывайте полученный массив в двумерный, либо перепишите получение исходного массива с учетом рекомендации в п.1.

lda = models.LdaModel(corpus,num\_topics=40, id2word=dictionary,iterations=5)

topics\_words = lda.show\_topics(num\_topics=40, num\_words=10,formatted=True)

wanted\_words = {"salt":0, "sugar":0, "water":0, "mushrooms":0, "chicken":0, "eggs":0 }

for words in topics\_words:

words\_list = str(words[1]).split('"')

for word in words\_list:

if word in wanted\_words.keys():

wanted\_words[word] += 1

print wanted\_words["salt"],wanted\_words["sugar"],wanted\_words["water"],wanted\_words["mushrooms"],wanted\_words["chicken"],wanted\_words["eggs"]