

docs = X\_toy

indptr = [0]

indices = []

data = []

vocabulary = {}

for d in docs:

for t in d:

index = vocabulary.setdefault(t, t)

indices.append(index)

data.append(1)

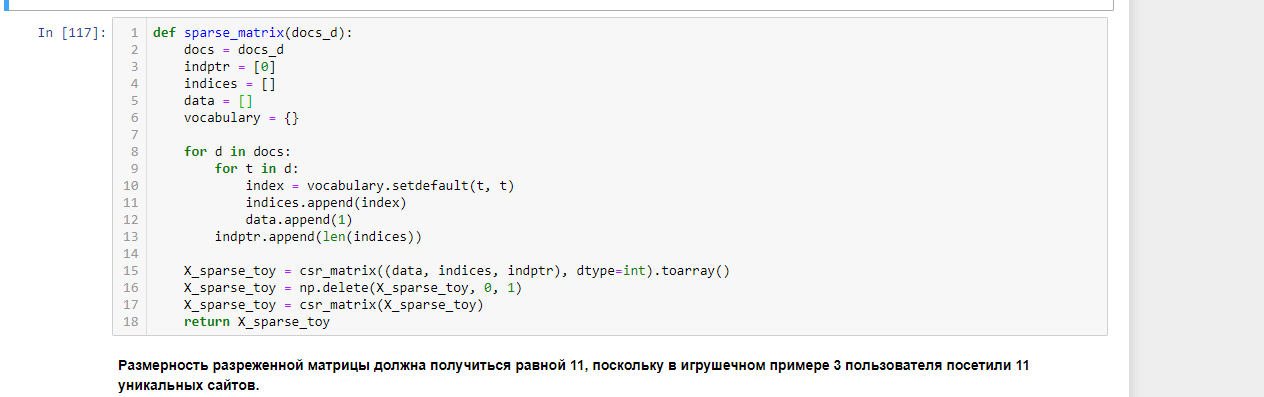
indptr.append(len(indices))

X\_sparse\_toy = csr\_matrix((data, indices, indptr), dtype=int).toarray()

X\_sparse\_toy = np.delete(X\_sparse\_toy, 0, 1)

X\_sparse\_toy = csr\_matrix(X\_sparse\_toy)

X\_sparse\_toy



def sparse\_matrix(docs\_d):

docs = docs\_d

indptr = [0]

indices = []

data = []

vocabulary = {}

for d in docs:

for t in d:

index = vocabulary.setdefault(t, t)

indices.append(index)

data.append(1)

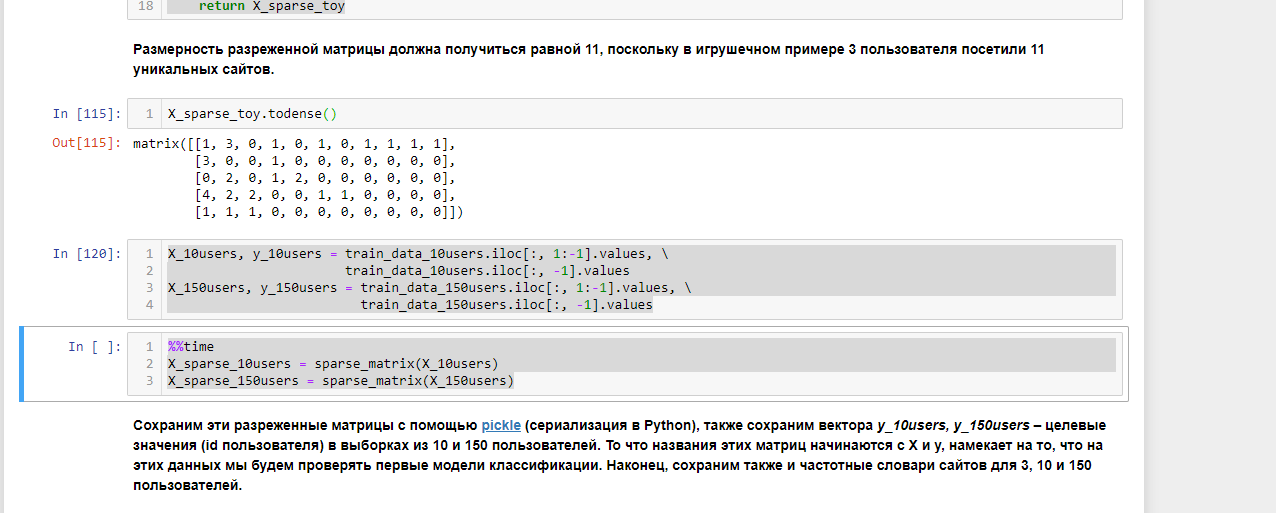
indptr.append(len(indices))

X\_sparse\_toy = csr\_matrix((data, indices, indptr), dtype=int).toarray()

X\_sparse\_toy = np.delete(X\_sparse\_toy, 0, 1)

X\_sparse\_toy = csr\_matrix(X\_sparse\_toy)

return X\_sparse\_toy



%%time

X\_sparse\_10users = sparse\_matrix(X\_10users)

X\_sparse\_150users = sparse\_matrix(X\_150users)