from twitter.deepbird.io.util import \_get\_feature\_id

import numpy as np

def numpy\_hashing\_uniform(the\_id, bin\_idx, output\_bits):

"""

integer\_multiplicative\_hashing

This is a reimplementation, for testing purposes, of the

c++ version found in hashing\_discretizer\_impl.cpp

"""

hashing\_constant = 2654435761

N = 32

with np.errstate(over='ignore'):

the\_id \*= hashing\_constant

the\_id += bin\_idx

the\_id \*= hashing\_constant

the\_id >>= N - output\_bits

the\_id &= (1 << output\_bits) - 1

return the\_id

def make\_feature\_id(name, num\_bits):

feature\_id = \_get\_feature\_id(name)

return np.int64(limit\_bits(feature\_id, num\_bits))

def limit\_bits(value, num\_bits):

return value & ((2 \*\* num\_bits) - 1)