In [1]:
```
pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.1.2.tar.gz (212.4 MB)
     |████████████████████████████████| 212.4 MB 63 kB/s
Collecting py4j==0.10.9
  Downloading py4j-0.10.9-py2.py3-none-any.whl (198 kB)
     |████████████████████████████████| 198 kB 45.5 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.1.2-py2.py3-none-any.whl size=212880768 sha256=1cd545a48bda9246ae2e2c57fcf5048b6e39f1cd540ccbeba1b9f79e7d8dde4
  Stored in directory: /root/.cache/pip/wheels/a5/0a/c1/9561f6fecb759579a7d863dcd846daaa95f5987... 
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9 pyspark-3.1.2
WARNING: Running pip as root will break packages and permissions. You should install packages reliably by using venv: https://pip.pypa.io/warnings/venv
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```python
from pyspark.sql import functions as F
from pyspark.sql import DataFrame as DFrame
from pyspark.sql.functions import udf, col, when
import matplotlib.pyplot as plt
import pyspark as ps
import os, sys, requests, json
from pyspark.sql.functions import col,size,regexp_replace,lit
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS

from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
from pyspark.ml import Pipeline
from pyspark.sql import Row
import numpy as np
import math
from pyspark.sql.functions import regexp_replace
from pyspark.sql import *


spark = (SparkSession.builder.appName("ModelTraining").config("spark.executor.memory","16g").getOrCreate())

sc = spark.sparkContext
```

In [3]:
```python
import numpy as np
import pandas as pd

from operator import add
from pyspark.ml.feature import RegexTokenizer, CountVectorizer, Tokenizer
from pyspark.ml.feature import StopWordsRemover, VectorAssembler
from pyspark.ml.feature import Word2Vec, Word2VecModel
from pyspark.ml.feature import IDF

from pyspark.ml import Pipeline, PipelineModel


from pyspark.sql.functions import *

from pyspark.sql.types import *

import folium
import html
```

In [4]:
```python
pro=spark.read.csv('../input/data-science-for-good-careervillage/professionals.csv', header=True,quote='"',sep=",",multiLine=True)
email=spark.read.csv('../input/data-science-for-good-careervillage/emails.csv', header=True,quote='"',sep=",",multiLine=True)
ques=spark.read.csv('../input/data-science-for-good-careervillage/questions.csv', header=True,quote='"',sep=",",multiLine=True)
match=spark.read.csv('../input/data-science-for-good-careervillage/matches.csv', header=True,quote='"',sep=",",multiLine=True)
ans=spark.read.csv('../input/data-science-for-good-careervillage/answers.csv', header=True,quote='"',sep=",",multiLine=True)
ans_score=spark.read.csv('../input/data-science-for-good-careervillage/answer_scores.csv', header=True,quote='"',sep=",",multiLine=True)
```

In [5]:
```python
#match_ques=match.join(ques,match.matches_question_id==ques.questions_id)
#match_ques=match_ques.join(ans,match_ques.answers_question_id)
#match_ques_email=match_ques.join(email,match_ques.matches_email_id==email.emails_id)
#match_ques_email_pro=match_ques_email.join(pro,match_ques.professionals_id)
```

In [6]:
```python
#ques_ans=match_ques.select("questions_id",'questions_body','answers_id','answers_body','professionals_id')
```

In [7]:
```python
#ques_ans=ques_ans.join(ans_score,ques_ans.answers_id==ans_score.id).select("questions_id","questions_body","professionals_id","score")
```

In [8]:
```python
#ques_ans=ques_ans.select("questions_id","questions_body","professionals_id","score")
```

In [9]:
```python
#Data cleaning question
def datacleaning_que(text):
    user_regex=r"(@\w{1,15})"
    clean_text=text.withColumn("user_mentioned",f.array_remove(
        f.array(
            f.regexp_extract(f.col("Full_questions"),user_regex,1),
            f.regexp_extract(
                f.col("Full_questions"), "".join([f"(?:{user_regex}).*?" for i in range(0,2)]),1),
                f.regexp_extract(
                f.col("Full_questions"), "".join([f"(?:{user_regex}).*?" for i in range(0,3)]),1),
                f.regexp_extract(
                f.col("Full_questions"), "".join([f"(?:{user_regex}).*?" for i in range(0,4)]),1),
                f.regexp_extract(
                f.col("Full_questions"), "".join([f"(?:{user_regex}).*?" for i in range(0,5)]),1),
                f.regexp_extract(
                f.col("Full_questions"), "".join([f"(?:{user_regex}).*?" for i in range(0,6)]),1)), "")).alias("user_mentioned"))
    clean_text=clean_text.withColumn("original_text",f.col("Full_questions")).withColumn("Full_questions",f.regexp_replace(f.col("Full_questions"),user_regex,"")).alias("Full_questions"))

    url_regex=r"((https?|ftp|file):\/{2,3})+([-\w+&@#/%=~|$?!:,.]*)|(www.)+([-\w+&@#/%=~|$?!:,.]*)"
    clean_text=clean_text.withColumn("Full_questions", f.regexp_replace(f.col("Full_questions"), url_regex, ""))
    # clean html
    clean_text = clean_text.withColumn('Full_questions', f.regexp_replace(f.col('Full_questions'), r'<.*?>', ''))

    email_regex=r"[\w.-]+@[\w.-]+\.[a-zA-Z]{1,}"
    clean_text=clean_text.withColumn("Full_questions",f.regexp_replace(f.col("Full_questions"),email_regex,""))

    text_cleaned=clean_text.withColumn('Full_questions', f.regexp_replace(f.col('Full_questions'),"[\n\r\t]"," ")).withColumn('Full_questions', f.regexp_replace(f.col('Full_questions'),"[^a-zA-Z]+", " ")).withColumn('Full_questions',f.trim(f.col("Full_questions"))).filter("Full_questions != ''")
    text_data=text_cleaned.select("questions_id",'Full_questions',"questions_author_id").coalesce(1).cache()
    return text_data
```

In [10]:
```python
#a=f.concat(f.col("questions_title"),f.lit(' '),f.col("questions_body"))
#ques=ques.withColumn("Full_questions",a).alias("full")
```

In [11]:
```python
ques=ques.withColumnRenamed("questions_body","Full_questions")
ques.show()
```

```
+--------------------+--------------------+--------------------+--------------------+--------------------+
|        questions_id| questions_author_id|questions_date_added|     questions_title|      Full_questions|
+--------------------+--------------------+--------------------+--------------------+--------------------+
|332a511f1569444...|8f6f374ffd834d258...|2016-04-26 11:14:...|Teacher   career ...|What  is  a  math...|
|eb80205482e4424c...|acccbda28edd4362a...|2016-05-20 16:48:...|I want to become ...|I am trying to fi...|
|4ec31632938a40b98...|f2c179a563024ccc9...|2017-02-08 19:13:...|Will going abroad...|I'm planning on g...|
|2f6a9a99d9b24e5ba...|2c30ffba444e40eab...|2017-09-01 14:05:...|To become a speci...|I hear business m...|
|5af8880460c141dbb...|aa9eb1a2ab184ebbb...|2017-09-01 02:36:...|Are there any sch...|I'm trying to fin...|
|7c336403258f4da3a...|ede458fc0e784c62b...|2017-03-01 06:27:...|How many years of...|To be an enginee...|
|be3c5edfdb07423e9...|71b4556d4a824251a...|2017-09-01 04:59:...|I want to become ...|I want to becom...|
|0f1d6a4f276c4a058...|585ac2330154f7cc9...|2016-05-19 22:16:...|What kind of  col...|I like soccer be...|
|d6999cdc47004fa1a...|654e1d6f1b947249...|2017-08-31 19:20:...|What are the col...|I'm asking becau...|
|e214acfbe6444b5b...|1690819691a48ed9...|2012-09-09 05:33:...|what does it take...|I am a sophomore ...|
|635cc2f2e146b155...|4627ecfaf1814d2e8...|2016-08-16 20:10:...|What major is bes...|Major #occupational...|
|a978f83b826c4fba8...|7fbe51b6214dadf84...|2016-04-12 15:23:...|How does the gend...|As a woman of co...|
|0317c8fd34f4a8c9...|26fa5e5e4ad668b0...|2016-10-25 21:09:...|As a civil engine...|I would love to b...|
|fec4bcf4814434a...|3acd97cf60da4b29...|2016-01-25 22:03:...|what is the harde...|Throughout the p...|
|bea937b4cd184a2a...|43411aa04f6041b0a...|2019-03-12 19:38:...|How to have a boa...|I've been accept...|
|eb0027b3dcd4488b...|aa425c619c7b49af9...|2017-03-10 01:03:...|To be a dental hy...|I'm a sophomore ...|
|0e0bac33e0594b21a...|8801c375ce6a439a9...|2016-07-09 00:01:...|Is it better to s...|I have a lot of ...|
|5cf8f9929e1a4357a...|7b7ad31c5947eb35b...|2016-05-13 01:49:...|Who/what do you t...|Most people cla...|
|f156171f9c4944de9...|454f346c03a14cb3a...|2018-01-17 17:26:...|How can I get my ...|I have developed ...|
|fb615c6770d4b7b9...|70a650136864d90a6...|2016-03-03 18:52:...|Which topics in c...|Hi, I am a high s...|
+--------------------+--------------------+--------------------+--------------------+--------------------+
only showing top 20 rows
```

In [12]:
```python
ques = datacleaning_que(ques)
ques.show()
```

```
+--------------------+--------------------+--------------------+
|        questions_id|      Full_questions| questions_author_id|
+--------------------+--------------------+--------------------+
|332a511f1569444...|What is a math t...|8f6f374ffd834d258...|
|eb80205482e4424c...|I am trying to fi...|acccbda28edd4362a...|
|4ec31632938a40b98...|I m planning on g...|f2c179a563024ccc9...|
|2f6a9a99d9b24e5ba...|I hear business m...|2c30ffba444e40eab...|
|5af8880460c141dbb...|I m trying to fin...|aa9eb1a2ab184ebbb...|
|7c336403258f4da3a...|To be an engineer...|ede458fc0e784c62b...|
|be3c5edfdb07423e9...|I want to become ...|71b4556d4a824251a...|
|0f1d6a4f276c4a058...|I like soccer bec...|585ac2330154f7cc9...|
|d6999cdc47004fa1a...|I m asking becaus...|654e1d6f1b947249...|
|e214acfbe6444b5b...|I am a sophomore ...|1690819691a48ed9...|
|635cc2f2e146b155...|Major occupational...|4627ecfaf1814d2e8...|
|a978f83b826c4fba8...|As a woman of col...|7fbe51b6214dadf84...|
|0317c8fd34f4a8c9...|I would love to b...|26fa5e5e4ad668b0...|
|fec4bcf4814434a...|Throughout the pa...|3acd97cf60da4b29...|
|bea937b4cd184a2a...|I ve been accepte...|43411aa04f6041b0a...|
|eb0027b3dcd4488b...|I m a sophomore ...|aa425c619c7b49af9...|
|0e0bac33e0594b21a...|I have a lot of i...|8801c375ce6a439a9...|
|5cf8f9929e1a4357a...|Most people clai...|7b7ad31c5947eb35b...|
|f156171f9c4944de9...|I have developed ...|454f346c03a14cb3a...|
|fb615c6770d4b7b9...|Hi I am a high sc...|70a650136864d90a6...|
+--------------------+--------------------+--------------------+
only showing top 20 rows
```

In [13]:
```python
ques_ans_final = ques.dropDuplicates(['questions_id','Full_questions'])
ques_ans_final=ques_ans_final.na.drop(subset=["Full_questions"])
```

In [14]:
```python
from pyspark.sql.functions import lit,row_number,col
from pyspark.sql.window import Window

w = Window().partitionBy(lit('a')).orderBy(lit('a'))

ques_ans_final = ques_ans_final.withColumn("row_num", row_number().over(w))
```

In [15]:
```python
# Build the pipeline
tokenizer = RegexTokenizer(gaps = False, pattern = '\w+',inputCol = "Full_questions", outputCol = "token")
stopWordsRemover = StopWordsRemover(inputCol = 'token', outputCol = 'nostopwrd')
countVectorizer = CountVectorizer(inputCol="nostopwrd", outputCol="rawFeature")
iDF = IDF(inputCol="rawFeature", outputCol="idf_vec")
word2Vec = Word2Vec(vectorSize = 100, minCount = 5, inputCol = 'nostopwrd', outputCol = 'word_vec', seed=123)
vectorAssembler = VectorAssembler(inputCols=['idf_vec', 'word_vec'], outputCol='comb_vec')
pipeline = Pipeline(stages=[tokenizer, stopWordsRemover, countVectorizer, iDF, word2Vec, vectorAssembler])

# fit the model
pipeline_mdl = pipeline.fit(ques_ans_final)
```

In [16]:
```python
ques_ans_final.show(5)
```

```
+--------------------+--------------------+-------+
|        questions_id|      Full_questions|row_num|
+--------------------+--------------------+-------+
|16bae94c385044b9...|Mainly asking thi...|      1|
|c850f2f3eea9849a9...|I ts really rare ...|      2|
|e16b6cde5e3f4e5d8...|How do you become...|      3|
|a50fd05e8d7481d6b...|I want to be abou...|      4|
|4f44f9e3646c8441c8...|I want to find a ...|      5|
+--------------------+--------------------+-------+
only showing top 5 rows
```

In [17]:
```python
# transform the question data

ques_pipeline_df = pipeline_mdl.transform(ques_ans_final)
```

In [18]:
```python
def CosineSim(vec1, vec2):
    return np.dot(vec1, vec2) / np.sqrt(np.dot(vec1, vec1)) / np.sqrt(np.dot(vec2, vec2))
```

In [19]:
```python
all_ques_vecs = ques_pipeline_df.select('questions_id', 'word_vec').rdd.map(lambda x: (x[0], x[1])).collect()
```

In [20]:
```python
def getQuestionDetails(in_ques):

    a = in_ques.alias("a")
    b = ques.alias("b")

    return a.join(b, col("a.questions_id") == col("b.questions_id"), 'inner') \
            .select([col('a.'+xx) for xx in a.columns] + [col("b.Full_questions")])
```

In [21]:
```python
def getKeyWordsRecomm(key_words, sim_bus_limit):

    #print('\nQuestion Relevent ' + key_words + '')

    input_words_df = sc.parallelize([(0, key_words)]).toDF(['questions_id', 'Full_questions'])

    # transform the the key words to vectors
    input_words_df = pipeline_mdl.transform(input_words_df)

    # choose word2vec vectors
    input_key_words_vec = input_words_df.select('word_vec').collect()[0][0]

    # get similarity
    sim_bus_byword_rdd = sc.parallelize((i[0], float(CosineSim(input_key_words_vec, i[1]))) for i in all_ques_vecs)

    sim_bus_byword_df = spark.createDataFrame(sim_bus_byword_rdd) \
        .withColumnRenamed('_1', 'questions_id') \
        .withColumnRenamed('_2', 'score') \
        .orderBy("score", ascending = False)

    sim_bus_byword_df=sim_bus_byword_df.na.drop(subset=["score"])
    # return top 10 similar
    a = sim_bus_byword_df.limit(sim_bus_limit)
    return getQuestionDetails(a).sort("score", ascending=False)
```

In [22]:
```python
ans_score=ans_score.withColumnRenamed("score","vote")
```

In [23]:
```python
def ecc_lipe(id_ques):

    key_words = str(np.array(ques.filter(ques.questions_id==id_ques).select('Full_questions').collect())[0][0])
    keywords_recom_df = getKeyWordsRecomm(key_words, 20)

    list_truth=np.array(ans.filter(ans.answers_question_id==id_ques).select('answers_author_id').collect())

    rs1 = keywords_recom_df.join(ans,keywords_recom_df.questions_id==ans.answers_question_id).select("questions_id","answers_author_id","score")
    rs2 = rs1.join(ans_score,rs1.answers_id==ans_score.id).select("answers_author_id","question_id","vote","score")
    rs2 = rs2.withColumn("ts",rs2.vote*rs2.score)
    rs2 = rs2.sort("ts", ascending=False).limit(20)
        # new top 20 | 0.9807
    #rs4 = rs3.sort("score", ascending=False).limit(20)
        # new top 20 vote >1 0.4219
    #rs2 = rs1.join(ans_score,rs1.answers_id==ans_score.id).select("answers_author_id","question_id","vote","score")
    #rs2=rs2.filter(rs2.vote>0)
    #rs2 = rs2.sort("score", ascending=False).limit(20)
        # Remove 1.0 score
    #rs2 = rs2.filter(rs2.score < 0.999)
    #rs2 = rs2.sort("score", ascending=False).limit(20)
    rs3 = rs2.join(pro,rs2.answers_author_id==pro.professionals_id).select("professionals_id","professionals_headline")
    list_pro=np.array(rs3.select("professionals_id").dropDuplicates().collect())
    count=0
    num_count=ans.filter(ans.answers_question_id==id_ques).count()
    for i in list_recom:
        for j in list_truth:
            if i==j: count=count+1
    count=float(count/num_count)
    print(count)
    return count
```

In [24]:
```python
def score(test):
    sum=0
    for i in test:
        sum=sum+ecc_lipe(i)
    score = float(sum/len(test))
    return score
```

In [25]:
```python
test=["01352c4d676e435ca9e74f3f3f2ad62a",
"03eee1ca0717487fb1b077027ab4fd4c",
"04a99f6e7fd40b9a07b61ae7a527ee",
"062f49f153de4b8793e4e669ec5b5331",
"093963c69d894a9f9e4e71e521641338",
"09e3bdc69a6143aa84585bbc1819fac37",
"0d7fab591dc1454c494ba4a5a078b77f",
"0db4ed6d2d4f43f188d39f8fcdc65ddbe",
"1a03fb2b8f1004478b3869ff015e9ebc43",
"1a444e5e5824446ea4370314a3fd0ce0"]
```

In [26]:
```python
a = score(test)
a
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:21: RuntimeWarning: invalid value encountered in double_scalars
```

```
1.0
1.0
0.1111111111111111
0.1818181818181818
0.16666666666666666
0.23529411764705882
0.13333333333333333
0.0
0.7692307692307693
0.15384615384615385
```

Out[26]: 0.44513003393632747