EXPERIMENT 4

NAME	**Shreya Shetty**
UID	**2019140059**
CLASS	**TE IT**
BATCH	**A**
SUBJECT	**Big Data Analytics Lab**

AIM: Implement algorithms on data streams in big data

ALGORITHM: LSH (Locality Sensitive Hashing)

DATASET: Netflix Movies and TV Shows

KAGGLE LINK: https://www.kaggle.com/shivamb/netflix-shows/

There are 8807 unique rows in the dataset chosen. Following are the 12 attributes in this dataset-

Sr. No.	Attribute Name	Attribute Description	Datatype
1	Show Id	Unique ID for every Movie / Tv Show	String
2	Type	Identifier - A Movie or TV Show	String
3	Title	Title of the Movie / Tv Show	String
4	Director	Director of the Movie	String
5	Cast	Actors involved in the movie / show	String
6	Country	Country where the movie / show was produced	String
7	Date Added	Date it was added on Netflix	Date
8	Release Year	Actual Release year of the move / show	Integer
9	Rating	TV Rating of the movie / show	String
10	Duration	Total Duration - in minutes or number of seasons	String
11	Listed In	Genre of Movie / TV Show	String
12	Description	The summary description	String

IMPLEMENTING LSH ALGORITHM ON 'NETFILX MOVIES AND TV SHOWS' DATASET IN PYTHON

1. Importing the required libraries:

```
# Importing required libraries
import numpy as np
import pandas as pd
import re
import time
from datasketch import MinHash, MinHashLSHForest
```

2. Reading the Netflix Movies and TV Show Dataset Downloaded from Kaggle:

```
# Importing and reading the dataset csv file
db = pd.read_csv('netflix_titles.csv')
```

3. Getting the number of rows and columns in dataset using .shape():

```
# Shape of dataset
print('Dataset Shape : ',db.shape)
```

```
Dataset Shape :  (8807, 12)
```

4. .head() returns top rows of the database:

```
db.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

5. Describing the Dataset to get the mean, count, nim, max, std deviation of all atributes:

```
# Decribing the dataset
db.describe()
```

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

6. Getting the information of Dataset:

```
# Information of dataset
print('Dataset Information : ',db.info)
```

```
Dataset Information :  <bound method DataFrame.info of      show_id     type                title         director  \
0         s1    Movie   Dick Johnson Is Dead  Kirsten Johnson
1         s2  TV Show          Blood & Water             NaN
2         s3  TV Show              Ganglands  Julien Leclercq
3         s4  TV Show  Jailbirds New Orleans             NaN
4         s5  TV Show           Kota Factory             NaN
...      ...      ...                   ...             ...
8802   s8803    Movie                Zodiac    David Fincher
8803   s8804  TV Show            Zombie Dumb             NaN
8804   s8805    Movie             Zombieland  Ruben Fleischer
8805   s8806    Movie                  Zoom     Peter Hewitt
8806   s8807    Movie                Zubaan      Mozez Singh

                                                  cast        country  \
0                                                  NaN  United States
1     Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...   South Africa
2     Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...            NaN
3                                                  NaN            NaN
4     Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...          India
...                                                 ...            ...
8802  Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...  United States
8803                                               NaN            NaN
8804  Jesse Eisenberg, Woody Harrelson, Emma Stone, ...  United States
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...  United States
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...          India

              date_added  release_year rating   duration  \
0     September 25, 2021          2020  PG-13     90 min
1     September 24, 2021          2021  TV-MA  2 Seasons
2     September 24, 2021          2021  TV-MA   1 Season
3     September 24, 2021          2021  TV-MA   1 Season
4     September 24, 2021          2021  TV-MA  2 Seasons
...                  ...           ...    ...        ...
8802   November 20, 2019          2007      R    158 min
8803        July 1, 2019          2018  TV-Y7  2 Seasons
8804    November 1, 2019          2009      R     88 min
8805    January 11, 2020          2006     PG     88 min
8806       March 2, 2019          2015  TV-14    111 min

                                        listed_in  \
0                                   Documentaries
1     International TV Shows, TV Dramas, TV Mysteries
2     Crime TV Shows, International TV Shows, TV Act...
3                             Docuseries, Reality TV
4     International TV Shows, Romantic TV Shows, TV ...
...                                             ...
```

```
8803            Kids' TV, Korean TV Shows, TV Comedies
8804                           Comedies, Horror Movies
8805                 Children & Family Movies, Comedies
8806      Dramas, International Movies, Music & Musicals

                                           description
0     As her father nears the end of his life, filmm...
1     After crossing paths at a party, a Cape Town t...
2     To protect his family from a powerful drug lor...
3     Feuds, flirtations and toilet talk go down amo...
4     In a city of coaching centers known to train I...
...                                                 ...
8802  A political cartoonist, a crime reporter and a...
8803  While living alone in a spooky town, a young g...
8804  Looking to survive in a world taken over by zo...
8805  Dragged from civilian life, a former superhero...
8806  A scrappy but poor boy worms his way into a ty...

[8807 rows x 12 columns]>
```

7. Finding Missing Values in Dataset:

```
# Finding missing values
db.isnull().sum()
```

```
show_id          0
type             0
title            0
director      2634
cast           825
country        831
date_added      10
release_year     0
rating           4
duration         3
listed_in        0
description      0
dtype: int64
```

8. Cleaning the Cast and Director columns from the dataset since these attributes are used in LSH algorithm for recommendation:

```
# Director and cast information is required, thus cleaning it
db["director"] = db["director"].fillna("Missing")
db["cast"] = db["cast"].fillna("Missing")
```

9. After correcting the Cast and Director columns, percentage of missing values in those attributes will be 0:

```
# percentage of missing values after correcting director and cast
(db.isnull().sum()/len(db.index))*100
```

```
show_id         0.000000
type            0.000000
title           0.000000
director        0.000000
cast            0.000000
country         9.435676
date_added      0.113546
release_year    0.000000
rating          0.045418
duration        0.034064
listed_in       0.000000
description     0.000000
dtype: float64
```

10. Implementing LSH algorithm:

i. Firstly, after importing the dataset, preprocess the database by removing all punctuation, then lowercase all text and then create unigram shingles (tokens) by separating any white space.

```
# Implementing LSH
# Preprocess will split a string of text into individual tokens/shingles based on whitespace.
def preprocess(text):
    text = re.sub(r'[^\w\s]','',text)
    tokens = text.lower()
    tokens = tokens.split()
    return tokens
```

ii. In order to create the Minhash Forest:

- Pass in a dataframe with every string we want to query.
- Preprocess a string of text using our preprocessing step above.
- Set the number of permutations in MinHash.
- MinHash the string on all of the shingles in the string.
- Store the MinHash of the string.
- Repeat 2-5 for all strings on the dataframe.
- Build a forest of all the MinHashed strings.
- Index your forest to make it searchable.

```
# Creating the Minhash Forest
def get_forest(data, perms):
    start_time = time.time()
    minhash = []
    for text in data['text']:
        # Preprocessing the text
        tokens = preprocess(text)
        m = MinHash(num_perm=perms)
        for s in tokens:
            m.update(s.encode('utf8'))
        minhash.append(m)
    forest = MinHashLSHForest(num_perm=perms)
    for i,m in enumerate(minhash):
        forest.add(i,m)
    forest.index()
    print('It took %s seconds to build the forest.' %(time.time()-start_time))
    return forest
```

iii. Set the parameters (permutations and number of recommendations)

```
# Choosing parameters
#Number of Permutations, standard number of permutations of 128
permutations = 128
#Number of Recommendations to return
num_recommendations = 5
```

iv. Then, following steps are used to evaluate queries :

- In order to query the forest that was built, we will follow the steps below:
- Preprocess given text into shingles.
- Set the same number of permutations for your MinHash as used to build the forest.
- Create MinHash on the text using all the shingles.
- Query the forest with the created MinHash and return the number of requested recommendations.
- Provide the titles and description of each Movie/ TV show recommended.

```
# Evaluate Queries
def predict(text, database, perms, num_results, forest):
    start_time = time.time()
    # Preprocessing the input text
    tokens = preprocess(text)
    m = MinHash(num_perm=perms)
    for s in tokens:
        m.update(s.encode('utf8'))
    idx_array = np.array(forest.query(m, num_results))
    if len(idx_array) == 0:
        return None # if query is empty, return none
    # Returning the movie/show name along with director, cast and decription
    # result = database.iloc[idx_array]['title'] + ' by ' + database.iloc[idx_array]['director'] + '
    result = database.iloc[idx_array]['title'] + ' : ' + database.iloc[idx_array]['description']
    print('It took %s seconds to query forest.' %(time.time()-start_time))
    return result
```

v. Test the Recommendation Engine on Netflix Movies and TV Shows Dataset:

- Create a new field ‘text’ that combines the title, cast, director and description into one field, in order to build are shingles using all these fields.

```
# Testing the Recommendation Engine

db['text'] = db['title'] + ' ' + db['cast'] + ' ' + db['director'] + ' ' + db['description']
```

- Building the Min Hash forest using the given dataset

```
# Building Min Hash forest
forest = get_forest(db, permutations)
```

Output:

```
It took 48.987688064575195 seconds to build the forest.

Input :  Crime Stories: India Detectives
It took 0.023995161056518555 seconds to query forest.
```

- Finally, query any string of text such as a title or description of Movie/ TV Show or name of the director, or cast names to return a list of recommendations. Below, I have given 4 different inputs & the corresponding recommendations are printed in the output below:

```
# Giving input of a movie name
test_input = 'Crime Stories: India Detectives'
print('\nInput : ',test_input)
result = predict(test_input, db, permutations, num_recommendations, forest)
print('\nTop Recommendation(s) is(are) \n')
print(result)

# Giving input of director name
test_input = 'Farah Khan'
print('\n\nInput : ',test_input)
result = predict(test_input, db, permutations, num_recommendations, forest)
print('\nTop Recommendation(s) is(are) \n')
print(result)

# Giving input of cast name
test_input = 'Shah Rukh Khan, Deepika Padukone'
print('\n\nInput : ',test_input)
result = predict(test_input, db, permutations, num_recommendations, forest)
print('\nTop Recommendation(s) is(are) \n')
print(result)

# Giving input of description
test_input = "three miserable engineering students and best friends struggle to beat the school's dr
print('\n\nInput : ',test_input)
result = predict(test_input, db, permutations, num_recommendations, forest)
print('\nTop Recommendation(s) is(are) \n')
print(result)
```

Output:

```
Top Recommendation(s) is(are)

8235    The Calling by Bumpy starring Missing : Food-c...
14      Crime Stories: India Detectives by Missing sta...
7129    Jhansi Ki Rani by Missing starring Ulka Gupta,...
7035    I Am by Onir starring Juhi Chawla, Rahul Bose,...
1885    Bad Boy Billionaires: India by Missing starrin...
dtype: object


Input :  Farah Khan
It took 0.015983104705810547 seconds to query forest.

Top Recommendation(s) is(are)

8161    Tees Maar Khan by Farah Khan starring Katrina ...
1192    The Present by Farah Nabulsi starring Saleh Ba...
4012    My Pride by Missing starring Khaled Amin, Elha...
7245    Kurt Seyit & Sura by Missing starring Kıvanç T...
4952    Main Hoon Na by Farah Khan starring Shah Rukh ...
dtype: object


Input :  Shah Rukh Khan, Deepika Padukone
It took 0.007987499237060547 seconds to query forest.

Top Recommendation(s) is(are)

3141    Karthik Calling Karthik by Vijay Lalwani starr...
301     Chennai Express by Rohit Shetty starring Shah ...
5617    Happy New Year by Farah Khan starring Shah Ruk...
4954    Om Shanti Om by Farah Khan starring Shah Rukh ...
4734    Tamasha by Imtiaz Ali starring Ranbir Kapoor, ...
dtype: object
```

```
Input :  three miserable engineering students and best friends struggle to beat the school's draconian system
It took 0.01598978042602539 seconds to query forest.

Top Recommendation(s) is(are)

516     Girl from Nowhere by Missing starring Chicha A...
3466    Girls Hostel by Missing starring Srishti Shriv...
6578    Deadly Scholars by Danny J. Boyle starring Ken...
7763    Power Rangers Dino Thunder by Missing starring...
1114    3 Idiots by Rajkumar Hirani starring Aamir Kha...
dtype: object
```

CONCLUSION:

In this experiment, I applied LSH on the Netflix Movies and TV Shows Dataset available on Kaggle and used it for recommendation of Movies/TV Shows based on given input which could be description of Movie/ TV Show or Name of the Director, or Cast names or any similar Title.