1) I) **Million Song Dataset:**
a) URL to access the data set:
http://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset

b) The Million Song Dataset has huge collection of audio features and metadata for a million popular soundtracks which we can treat as objects. And each of which has several attributes. Below are a list of all attributes available in each of the files of the dataset. If any of the particular field is exist then it will be displayed. If any of the field includes a large amount of numerical data then we provide only the shape of the data array. Most of these fields are processed using Echo Nest Analyze API's. The Echo Nest API allows us to call methods that respond in JSON or XML. We can call a method by using an HTTP GET to retrieve along with method specific arguments.

Field name	Type	Description
analysis sample rate	Float	sample rate of the audio used
artist digital id	Int	ID from digital.com or -1
artist familiarity	Float	algorithmic estimation
artist hotness	Float	algorithmic estimation
artist id	String	Echo Nest ID
artist latitude	Float	Latitude
artist location	String	location name
artist longitude	Float	Longitude
artist mbid	String	ID from musicbrainz.org
artist mbtags	array string	tags from musicbrainz.org
artist mbtags count	array int	tag counts for musicbrainz tags
artist name	String	artist name
artist playme id	Int	ID from playme.com, or -1
artist terms	array string	Echo Nest tags
artist terms freq	array float	Echo Nest tags freqs
artist terms weight	array float	Echo Nest tags weight

audio md5	String	audio hash code
bars confidence	array float	confidence measure
bars start	array float	beginning of bars, usually on a beat
beats confidence	array float	confidence measure
beats start	array float	result of beat tracking
Dance ability	Float	algorithmic estimation
Duration	Float	in seconds
end of fade in	Float	seconds at the beginning of the song
Energy	Float	energy from listener point of view
Key	Int	key the song is in
key confidence	Float	confidence measure
Loudness	Float	overall loudness in dB
Mode	Int	major or minor
mode confidence	Float	confidence measure
Release	String	album name
release digital id	Int	ID from digital.com or -1
sections confidence	array float	confidence measure
sections start	array float	largest grouping in a song, e.g. verse
segments confidence	array float	confidence measure
segments loudness max	array float	max dB value
segments loudness max time	array float	time of max dB value, i.e. end of attack
segments loudness max start	array float	dB value at onset
segments pitches	2D array float	chroma feature, one value per note

segments start	array float	musical events, ~ note onsets
segments timbre	2D array float	texture features (MFCC+PCA-like)
similar artists	array string	EchoNest artist IDs
song hotness	Float	algorithmic estimation
song id	String	Echo Nest song ID
start of fade out	Float	time in sec
Tatum's confidence	array float	confidence measure
Tatum's start	array float	smallest rythmic element
Tempo	Float	estimated tempo in BPM
time signature	Int	estimate of number of beats per bar, e.g. 4
time signature confidence	Float	confidence measure
Title	String	song title
track id	String	Echo Nest track ID
track digital id	Int	ID from digital.com or -1
Year	Int	song release year from MusicBrainz or 0

c) This data set has huge collection of popular contemporary soundtracks and audio features. Mining of this multimedia data provides detailed information about particular song like title, artist name, duration, mode, time, song hotness etc. We can even deduce patterns from the data set using similar mode or similar tempo or similar genre of songs which improves chances of songs appearing together.

d) When user searches for a song which is present in this data set, we can show or suggest to the user about the similar kind of songs which he may be interested in. It will be useful to extract popular songs faster based on user requirements. There are plenty of applications out there for listening to songs which manage audio features and data like this data set, for example Soundcloud, Spotify etc. All these applications extract data based on user requirements to display the song user desired. Besides, we can find the pattern to show similar kind of songs.

1) II) **Stanford Large Network Dataset Collection (Social Networks)**

a) URL to access the data set: http://snap.stanford.edu/data/index.html

Please note that I have concentrated more on Social Networks data set.

b) SNAP library actively focused on analysis of large social and information networks. All networks maintains nodes and edges which represent relation between the nodes. We have different objects for different type of network. And, each of these objects have few attributes. For example, Social Networks has few objects like Facebook, Twitter, Google Plus, etc. Each of which have few attributes like Name, Type, Nodes, Edges and Description. Here nodes represents people and edges represents interaction between people. If any of the particular attribute is available then it will be displayed. Most of these attributes data are collected after researching or conducting survey in social and communication networks. Each person's id is replaced with internal id to protect every one's privacy while storing profile in data set.

c) This data set has huge collection of social and communication networks data. We can extract the data about people's interaction, how well they are connected with others, friends list, social profile, interests, achievements etc in social networks like Facebook, Twitter etc. From this, we can dig about human interactions and can research further to know the impact of social networks on humans. How people's interaction is changed based on situation over the time, advantages and disadvantages of social networks from the people's perspective.

d) Social Networks like Facebook,Twitter etc are widely popular and connects millions people all over the world. We can find similar profiles who shares common interests or who are from same place or studies or studied in same school. Also, we can even find and connect old school friends who studied in same school but not in touch afterwards. Social networks really helps to connect people all over the world to maintain healthy relationships. Apart from this, they also provide platform to spread the message or awareness and can be used to help the people who are in need of something in desperate times.

2)

a) Title: Efficient Algorithms for Public-Private Social Networks.

Authors: Flavio Chierichetti, Alessandro Epasto, Ravi Kumar, Silvio Lattanzi, Vahab Mirrokni

Conference venue: KDD'15.

Authors	Affiliations
Flavio Chierichetti	Sapienza University,Rome, Italy
Alessandro Epasto	Brown University Providence, RI

Ravi Kumar	Google Mountain View, CA
Silvio Lattanzi	Google New York, NY
Vahab Mirrokni	Google New York, NY

b) Online social networks introduced a new feature where users can define public and private friends, or public and private lists or circles. This new feature inspired them and in this paper, they addressed about the public-private model of computation for online social networks. They proposed efficient algorithms for many important social network problems and described the computational benefits of the their framework by experimental analysis. Nowadays privacy plays key role in social networks. To be precise, privacy controls the way information is shared among the members of the social network, and also influences the way in which the network itself can be viewed and processed by algorithms. Privacy issues are a major factor in the algorithmic analysis of social networks. Recent studies in New York, reported that more than 50% of facebook users hid their friends list. So we can say that public-private feature is well received by most of the users and it opened doors for further research. Social networks are massively growing and produces huge collection of the data daily. With growing demand and massive data, it will be a challenge to develop efficient algorithms in the public-private network model.

c) In this paper, they proposed an efficient algorithm for public-private network model. To develop that, they considered two powerful computational paradigms for massive graphs, sketching and sampling. Sketching algorithms have been developed for some basic graph problems including connectivity and cut sizes and more social-network specific problems such as neighborhood estimation and reachability. Sampling algorithms are used to describe the model with nontrivial algorithms for three key social network problems, estimating all-pair distances, estimating node similarities, and correlation clustering. Details about the algorithms are mentioned in the paper.

d) They illustrated the effectiveness of the model and the computational efficiency of the algorithms by performing experiments on real-world social networks. For experimental purposes to show the public-private graph model, they choose the following.

Algorithms Used: Reachability tree size estimation by sketching, shortest path by sampling, and correlation clustering by sampling.
Data Set: Stanford's SNAP dataset (http://snap.stanford.edu/data).
Social Networks Used: Slashdot, Epinions, Wiki-Vote, YouTube, Email-EUAll, Gnutella, DBLP, LiveJournal, Orkut.

3)