1. Data Import—code sample
2. Data Layer with permanent struct—struct sample
3. Data Deduplication
4. Fact movies integration for star schema
5. Match Results Sample explained

1.Data Import—code sample

The data may be collected or produced from a number of external data sources by using various method. We add here a basic example of web scraping (see code sample in file imdb.py). The data content is load to primary data in temporary table with a 1:1 content (see struct example in script1 ) .Main ideea to keep a ‘look-up’ copy of data that are used later in the etl workflow

2. Data Layer with permanent struct

See script sample enclosed .(script 2) , this permanent table will contain the original score values from primary sources and the natural business key id.Using this key we are able to join the mai fact table fact\_movies with the primary sources.

When these tables will be populate first time (bul load) they will have a truncate content form and after that they will work with the delete under data load condition (see script enclosed—truncate.sql and delete\_under\_conition .sql – delta load.

3. Deduplication Layer

In this project a kind off data merging process will be indicated in order to ensure data consistency , data deduplication and the unicity of the records.We need to take into account that we have multiple data sources and a movie may exists in a single system but also we may have the same movie in multiple data systems. No matter what data merge, methods will be selected the deduplication will be a challenge that need to be fixed. Duplication of the record’s may exists in multiple systems and may have different shapes:

-multiple records in many systems data sources that in fact represent the same movie

-multiple attributes combination not always consistent (see genre of the movie)

-duplicate movies in the same system

The solution to this issue is more complex and the proposal is to fix this aspects in three stages .

A.first for each primary data set we detect if we have duplicate rows inside the system

B.based on the primary natural common key (name of the movie and the year of occurrence we detect the presence of the movie in multiple systems.We collect this id into a look-up table

After that per each system we will insert the rows into main table avoiding the values from look-up table , Finally at the end of the cycle we insert the rows with common values .

C.Stage 3 is about to apply fuzzy algorithms to detect if some duplicate values still remain to be detected

The primary data matching algorithm is based on movie name and the year of apparition ,so, we build a primary key based on the name of the movie and the year . The sensitive part of this primary data matching algorithm is the fact that the movie name is a descriptive value that may include abbreviation or other kind of issues like : acronyms, name reversal, phonetic spelling , deliberate misspelling, insertion of special char , different spelling of the name etc

To conclude into a more refined stage we will need to resolve all this issues and for this stage 3 we recommend few fuzzy algorithm’s like :

Levenshtein Distance,Jaro Winkler,Jaccard Index,Acronym,Name Variant

That will be used to check if we have two records with the same content .

Stage two is designed to use this fuzzy algoriythms and at the end a kind of ensemble learning method we may decide if from the rest of data we have some similar movies etc.

At the end of this process we will know if a pair of records are duplicated or by case rejected as non-duplicates.For the records that cannot be included in this two categories we will let a human operator to decide .Probably we can automate about 95-98 % from the total number of movies using this validations combined.

The fully process is explosed in machine\_learning\_engine .jpg picture (enclosed)

This process in full size will take time for the first iteration but if we start a delta process , means the new data that will occur is daily consumed only as a difference .

Particular day (delta process) can solve the data processing task in a shorter amount of time.

Related to the genre we must mention that it may be possible to have various situation when a movie have different classification in different data sources systems but also it may be possible to have common classification . In order to avoid duplication and also to maintain consistency we use the function as the one described in the script singular\_genre\_function.sql

Basically if we have source 1 genre : Drama, Action, Scifi and into the second data source a kind of : Drama ,Thriller ,Fiction when the function is called we will obtain a deduplicated string like : Drama,Action,Scifi,Thriller,Fiction each genre type will be present only once.

In my example we assume that we have 4 distinct source .

%4Fact movies integration for star schema

See main\_script\_fact\_movies.sql

5%Match Results Sample explained

See xls fact\_movie.xls