Task 1.1

Because I think 'year' is an useful field for blocking and 'title' for a movie is very easy to distinguish. Also, match 'directors' field can increase the matching accuracy. In 'directors', I split sirectors' names into tokens and match between imdb and tmd to make sure two movies are the same, instead of just matching with 'title'. The reason I don't use more attributes is because I don't false neative become too low.

Task 1.2

I choose to use 'year' as blocking because two movies from some year are more likely to be the same movie. Although I can use 'title' as well, I think block with 'year' field is easier to implement and it is efficient.

```
Reduction ratio is:
0.9642440595155491
pairs completeness is:
0.9928
```

Task 1.3

I use Levenshtein distance and a threshold to determine if two title are 'close', which mean how many CRUD operations one string needs to take until match another string. If Levenshtein distance between two titles are too far then return 0, else return 1. Also, I compare two set of tokens to determind if 'directors' fields are similar enough, two set of tokens are same, return 1, else return 0.

I then design a scoring function that return true only if both 'title' and 'director' similarity checks are passed. So I give 'title' similarity 0.3 and 'director' similarity 0.7. only if they add together equal or exceed 1, then return true. I think this socring function is enough for current design and as the result, the **precison is 1.0, recall is 0.92578125, f-measure is 0.9614604462474645**. So I think my scoring function is good enough.

Task 2.1

I design my model use a class, which is a sub class of Movie class to represent movies, inside this class, I wrote 6 properties called *writers, actors, dataPublised, productionCompany and title*. Since writers properties doesn't exist in http://schema.org I design this writers as a sub property from name property. I also create two class call imdb and tmd, which are the sub class of Organization, to represent the source of each entity. Therefore, productionCompany will be either imdb or tmd.

Task 2.3