```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```

Dataset

Complet dataset

```
In [2]: dataset_path = "dataset/"
        film_data_path = dataset_path + "films.csv"
        films = pd.read_csv(film_data_path)
        films
```

Out[2]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija
0	The Imitation Game	8.0	1	1	1	0	0
1	Ex Machina	7.7	0	1	0	0	1
2	A Beautiful Mind	8.2	1	1	0	0	0
3	Good Will Hunting	8.3	0	1	0	0	0
4	Forrest Gump	8.8	0	1	0	0	0
5	21	6.8	0	1	0	0	0
6	Gifted	7.6	0	1	0	0	0
7	Travelling Salesman	5.9	0	1	0	0	1
8	Avatar	7.9	0	0	0	0	0
9	The Wolf of Wall Street	8.2	1	0	0	1	0
10	A Time To Kill	7.4	0	1	1	0	0
11	Interstellar	8.6	0	1	0	0	0
12	The Wind Rises	7.8	1	1	0	0	0

Target film: The Imitation Game

```
In [3]: target_film = films.loc[films["Film"] == "The Imitation Game"]
        target_film
```

Out[3]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija
0	The Imitation Game	8.0	1	1	1	0	0

Other films

In [4]:
```python
new_films = films.loc[films["Film"] != "The Imitation Game"]
new_films
```

Out[4]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija
1	Ex Machina	7.7	0	1	0	0	1
2	A Beautiful Mind	8.2	1	1	0	0	0
3	Good Will Hunting	8.3	0	1	0	0	0
4	Forrest Gump	8.8	0	1	0	0	0
5	21	6.8	0	1	0	0	0
6	Gifted	7.6	0	1	0	0	0
7	Travelling Salesman	5.9	0	1	0	0	1
8	Avatar	7.9	0	0	0	0	0
9	The Wolf of Wall Street	8.2	1	0	0	1	0
10	A Time To Kill	7.4	0	1	1	0	0
11	Interstellar	8.6	0	1	0	0	0
12	The Wind Rises	7.8	1	1	0	0	0

In [5]:
```python
new_films_values = new_films.drop("Film", axis=1)
new_films_values = new_films_values.values
new_films_values
```

Out[5]:
```
array([[7.7, 0. , 1. , 0. , 0. , 1. ],
       [8.2, 1. , 1. , 0. , 0. , 0. ],
       [8.3, 0. , 1. , 0. , 0. , 0. ],
       [8.8, 0. , 1. , 0. , 0. , 0. ],
       [6.8, 0. , 1. , 0. , 0. , 0. ],
       [7.6, 0. , 1. , 0. , 0. , 0. ],
       [5.9, 0. , 1. , 0. , 0. , 1. ],
       [7.9, 0. , 0. , 0. , 0. , 0. ],
       [8.2, 1. , 0. , 0. , 1. , 0. ],
       [7.4, 0. , 1. , 1. , 0. , 0. ],
       [8.6, 0. , 1. , 0. , 0. , 0. ],
       [7.8, 1. , 1. , 0. , 0. , 0. ]])
```

In [6]:
```python
target_film_values = target_film.drop("Film", axis=1)
target_film_values = target_film_values.values
target_film_values
```

Out[6]:
```
array([[8., 1., 1., 1., 0., 0.]])
```

Euclidean distance

In [7]:
```python
euclidean_distances = np.linalg.norm(new_films_values - target_film_values, axis=1)
euclidean_distances
```

Out[7]:
```
array([1.75783958, 1.0198039 , 1.44568323, 1.62480768, 1.8547237 ,
       1.46969385, 2.72213152, 1.73493516, 1.74355958, 1.16619038,
       1.53622915, 1.0198039 ])
```

Manhattan distance

In [8]:
```python
def manhattan_distance(a, b):
    return np.array(list(map(np.sum,np.array(list(map(np.abs, a - b))))))

manhattan_distances = manhattan_distance(new_films_values, target_film_values[0])
manhattan_distances
```

Out[8]:
```
array([3.3, 1.2, 2.3, 2.8, 3.2, 2.4, 5.1, 3.1, 3.2, 1.6, 2.6, 1.2])
```

Sort by distance

In [9]:
```python
new_film_euclidean = new_films.copy()
new_film_euclidean.insert(len(new_film_euclidean.columns), "Euklidsko rastojanje", euclidean_distances)
new_film_euclidean.sort_values(by=["Euklidsko rastojanje"])
```

Out[9]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija	Euklidsko rastojanje
2	A Beautiful Mind	8.2	1	1	0	0	0	1.019804
12	The Wind Rises	7.8	1	1	0	0	0	1.019804
10	A Time To Kill	7.4	0	1	1	0	0	1.166190
3	Good Will Hunting	8.3	0	1	0	0	0	1.445683
6	Gifted	7.6	0	1	0	0	0	1.469694
11	Interstellar	8.6	0	1	0	0	0	1.536229
4	Forrest Gump	8.8	0	1	0	0	0	1.624808
8	Avatar	7.9	0	0	0	0	0	1.734935
9	The Wolf of Wall Street	8.2	1	0	0	1	0	1.743560
1	Ex Machina	7.7	0	1	0	0	1	1.757840
5	21	6.8	0	1	0	0	0	1.854724
7	Travelling Salesman	5.9	0	1	0	0	1	2.722132

In [10]:
```python
k = 6
euclidean_nearest_neighbor_ids = euclidean_distances.argsort()[:k] # python indexes start = 0
new_films["Film"][euclidean_nearest_neighbor_ids + 1]  # pandas indexes start = 1
```

Out[10]:
```
2        A Beautiful Mind
12         The Wind Rises
10          A Time To Kill
3       Good Will Hunting
6                  Gifted
11           Interstellar
Name: Film, dtype: object
```

In [11]:
```python
new_film_manhattan = new_films.copy()
new_film_manhattan.insert(len(new_film_manhattan.columns), "Menhetn rastojanje", manhattan_distances)
new_film_manhattan.sort_values(by=["Menhetn rastojanje"])
```

Out[11]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija	Menhetn rastojanje
2	A Beautiful Mind	8.2	1	1	0	0	0	1.2
12	The Wind Rises	7.8	1	1	0	0	0	1.2
10	A Time To Kill	7.4	0	1	1	0	0	1.6
3	Good Will Hunting	8.3	0	1	0	0	0	2.3
6	Gifted	7.6	0	1	0	0	0	2.4
11	Interstellar	8.6	0	1	0	0	0	2.6
4	Forrest Gump	8.8	0	1	0	0	0	2.8
8	Avatar	7.9	0	0	0	0	0	3.1
9	The Wolf of Wall Street	8.2	1	0	0	1	0	3.2
5	21	6.8	0	1	0	0	0	3.2
1	Ex Machina	7.7	0	1	0	0	1	3.3
7	Travelling Salesman	5.9	0	1	0	0	1	5.1

In [12]:
```python
k = 6
manhattan_nearest_neighbor_ids = manhattan_distances.argsort()[:k]
new_films["Film"][manhattan_nearest_neighbor_ids + 1]
```

Out[12]:
```
2        A Beautiful Mind
12         The Wind Rises
10          A Time To Kill
3       Good Will Hunting
6                  Gifted
11           Interstellar
Name: Film, dtype: object
```

Normalization

```
In [13]:  imdb_min = np.min(films["IMDB"].values)
          imdb_max = np.max(films["IMDB"].values)
          print(imdb_min, imdb_max)
```

```
5.9 8.8
```

Target film values normalization

```
In [14]:  target_film_imdb_values_normalized = target_film["IMDB"].values
          target_film_imdb_values_normalized = (target_film_imdb_values_normalized - imd
          b_min) / (imdb_max - imdb_min)
          target_film_imdb_values_normalized
```

```
Out[14]:  array([0.72413793])
```

Other films values normalization

```
In [15]:  new_films_imdb_values_normalized = new_films["IMDB"].values
          new_films_imdb_values_normalized = (new_films_imdb_values_normalized - imdb_mi
          n) / (imdb_max - imdb_min)
          new_films_imdb_values_normalized
```

```
Out[15]:  array([0.62068966, 0.79310345, 0.82758621, 1.        , 0.31034483,
                 0.5862069 , 0.        , 0.68965517, 0.79310345, 0.51724138,
                 0.93103448, 0.65517241])
```

Normalized data

In [16]:
```python
new_films_normalized = new_films.drop("IMDB", axis=1)
new_films_normalized.insert(1, "IMDB", new_films_imdb_values_normalized)
new_films_normalized
```

Out[16]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija
1	Ex Machina	0.620690	0	1	0	0	1
2	A Beautiful Mind	0.793103	1	1	0	0	0
3	Good Will Hunting	0.827586	0	1	0	0	0
4	Forrest Gump	1.000000	0	1	0	0	0
5	21	0.310345	0	1	0	0	0
6	Gifted	0.586207	0	1	0	0	0
7	Travelling Salesman	0.000000	0	1	0	0	1
8	Avatar	0.689655	0	0	0	0	0
9	The Wolf of Wall Street	0.793103	1	0	0	1	0
10	A Time To Kill	0.517241	0	1	1	0	0
11	Interstellar	0.931034	0	1	0	0	0
12	The Wind Rises	0.655172	1	1	0	0	0

In [17]:
```python
target_film_normalized = target_film.drop("IMDB", axis=1)
target_film_normalized.insert(1, "IMDB", target_film_imdb_values_normalized)
target_film_normalized
```

Out[17]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija
0	The Imitation Game	0.724138	1	1	1	0	0

In [18]:
```python
new_films_values_normalized = new_films_normalized.drop("Film", axis=1)
new_films_values_normalized = new_films_values_normalized.values

target_film_values_normalized = target_film_normalized.drop("Film", axis=1)
target_film_values_normalized = target_film_values_normalized.values

euclidean_distances_normalized = np.linalg.norm(new_films_values_normalized - target_film_values_normalized, axis=1)
euclidean_distances_normalized

manhattan_distances_normalized = manhattan_distance(new_films_values_normalized, target_film_values_normalized[0])
manhattan_distances_normalized
```

Out[18]:
```
array([3.10344828, 1.06896552, 2.10344828, 2.27586207, 2.4137931 ,
       2.13793103, 3.72413793, 3.03448276, 3.06896552, 1.20689655,
       2.20689655, 1.06896552])
```

In [19]:
```python
new_film_euclidean_normalized = new_films_normalized.copy()
new_film_euclidean_normalized.insert(len(new_film_euclidean_normalized.columns
), "Euklidsko rastojanje", euclidean_distances_normalized)
new_film_euclidean_normalized.sort_values(by=["Euklidsko rastojanje"])
```

Out[19]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija	Euklidsko rastojanje
2	A Beautiful Mind	0.793103	1	1	0	0	0	1.002375
12	The Wind Rises	0.655172	1	1	0	0	0	1.002375
10	A Time To Kill	0.517241	0	1	1	0	0	1.021179
3	Good Will Hunting	0.827586	0	1	0	0	0	1.417992
6	Gifted	0.586207	0	1	0	0	0	1.420924
11	Interstellar	0.931034	0	1	0	0	0	1.429268
4	Forrest Gump	1.000000	0	1	0	0	0	1.440868
5	21	0.310345	0	1	0	0	0	1.473508
8	Avatar	0.689655	0	0	0	0	0	1.732394
9	The Wolf of Wall Street	0.793103	1	0	0	1	0	1.733423
1	Ex Machina	0.620690	0	1	0	0	1	1.735137
7	Travelling Salesman	0.000000	0	1	0	0	1	1.877332

In [20]:
```python
k = 6
euclidean_nearest_neighbor_normalized_ids = euclidean_distances_normalized.arg
sort()[:k]
new_films_normalized["Film"][euclidean_nearest_neighbor_normalized_ids + 1]
```

Out[20]:
```
2       A Beautiful Mind
12        The Wind Rises
10        A Time To Kill
3      Good Will Hunting
6                 Gifted
11          Interstellar
Name: Film, dtype: object
```

In [21]:
```python
new_film_manhattan_normalized = new_films_normalized.copy()
new_film_manhattan_normalized.insert(len(new_film_manhattan_normalized.columns
), "Menhetn rastojanje", manhattan_distances_normalized)
new_film_manhattan_normalized.sort_values(by=["Menhetn rastojanje"])
```

Out[21]:

	Film	IMDB	Biografski	Drama	Triler	Komedija	Misterija	Menhetn rastojanje
2	A Beautiful Mind	0.793103	1	1	0	0	0	1.068966
12	The Wind Rises	0.655172	1	1	0	0	0	1.068966
10	A Time To Kill	0.517241	0	1	1	0	0	1.206897
3	Good Will Hunting	0.827586	0	1	0	0	0	2.103448
6	Gifted	0.586207	0	1	0	0	0	2.137931
11	Interstellar	0.931034	0	1	0	0	0	2.206897
4	Forrest Gump	1.000000	0	1	0	0	0	2.275862
5	21	0.310345	0	1	0	0	0	2.413793
8	Avatar	0.689655	0	0	0	0	0	3.034483
9	The Wolf of Wall Street	0.793103	1	0	0	1	0	3.068966
1	Ex Machina	0.620690	0	1	0	0	1	3.103448
7	Travelling Salesman	0.000000	0	1	0	0	1	3.724138

In [22]:
```python
k = 6
manhattan_nearest_neighbor_normalized_ids = manhattan_distances_normalized.arg
sort()[:k]
new_films_normalized["Film"][manhattan_nearest_neighbor_normalized_ids + 1]
```

Out[22]:
```
2        A Beautiful Mind
12         The Wind Rises
10         A Time To Kill
3        Good Will Hunting
6                  Gifted
11           Interstellar
Name: Film, dtype: object
```

In [ ]: