



```
install.packages("FNN")
install.packages("data.table")
install.packages("cluster")
```

 Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

```
library(FNN)
library(data.table)
library(cluster)
library(dplyr)
```

 Attaching package: ‘dplyr’

The following objects are masked from ‘package:data.table’:

between, first, last

The following objects are masked from ‘package:stats’:


filter, lag

The following objects are masked from ‘package:base’:

intersect, setdiff, setequal, union

```
data <- read.csv("C:/Users/DELL/Downloads/merged_data_movielens.csv")
```

```
head(data)
```

 A data.frame: 6 × 24

	X	movieId	title	X.no.genres.listed.	Action	Adventure	Animation	Children	Comedy	Crime	...	Horror	IMAX	Musical	Myste
	<int>	<int>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	...	<int>	<int>	<int>	<in
1	0	1	Toy Story (1995)	0	0	1	1	1	1	0	...	0	0	0	
2	1	2	Jumanji (1995)	0	0	1	0	1	0	0	...	0	0	0	
3	2	3	Grumpier Old Men (1995)	0	0	0	0	0	1	0	...	0	0	0	
4	3	4	Waiting to Exhale (1995)	0	0	0	0	0	1	0	...	0	0	0	
5	4	5	Father of the Bride Part II (1995)	0	0	0	0	0	1	0	...	0	0	0	
6	5	6	Heat (1995)	0	1	0	0	0	0	1	...	0	0	0	

```
dim(data)
```

 20700 24

```
id_title <- data[, c("movieId", "title")]
data$title <- NULL
```

```
head(data)
```

A data.frame: 6 × 23

	X	movieId	X.no.genres.listed.	Action	Adventure	Animation	Children	Comedy	Crime	Documentary	...	Horror	IMAX	Musical	My
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	0	1		0	0	1	1	1	1	0	0 ...	0	0	0	
2	1	2		0	0	1	0	1	0	0	0 ...	0	0	0	
3	2	3		0	0	0	0	0	1	0	0 ...	0	0	0	
4	3	4		0	0	0	0	0	1	0	0 ...	0	0	0	
5	4	5		0	0	0	0	0	1	0	0 ...	0	0	0	
6	5	6		0	1	0	0	0	0	1	0 ...	0	0	0	

```
x <- data[, -c(1, 2)]
```

```
head(x)
```

A data.frame: 6 × 21

	X.no.genres.listed.	Action	Adventure	Animation	Children	Comedy	Crime	Documentary	Drama	Fantasy	...	Horror	IMAX	Musical	My
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1		0	0	1	1	1	1	0	0	0	1 ...	0	0	0	
2		0	0	1	0	1	0	0	0	0	1 ...	0	0	0	
3		0	0	0	0	0	1	0	0	0	0 ...	0	0	0	
4		0	0	0	0	0	1	0	0	1	0 ...	0	0	0	
5		0	0	0	0	0	1	0	0	0	0 ...	0	0	0	
6		0	1	0	0	0	0	1	0	0	0 ...	0	0	0	

Nearest Neighbors approach.

```
train_matrix <- as.matrix(data[, -c(1, 2)])
train_structure <- knn.index(train_matrix)
save(train_structure, file = "knn_structure.RData")
```

```
get_second_closest_neighbors <- function(data, input_movie, train_matrix, k = 2) {
  filtered_movies <- data[data$movieId %in% input_movie, ]
  test_matrix <- as.matrix(filtered_movies[, -c(1, 2)])
  knn_result <- get.knnx(data = train_matrix, query = test_matrix, k = k)
  return(knn_result$nn.index[, 2])
}
```

```
load("knn_structure.RData")
input_movie <- c(1, 22, 34, 40)
res <- get_second_closest_neighbors(data, input_movie, train_matrix, k = 2)
```

```
cat("The movie recommendation ids are:", res, "\n")
```

```
↗ The movie recommendation ids are: 4791 20826 8077 3095
```

Kmeans

```
# Use only numeric columns from the movie_features dataset
X <- data %>%
  select(where(is.numeric)) %>%
  as.matrix()
```

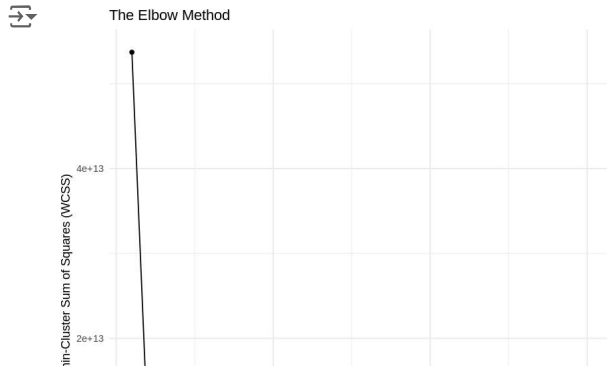
[illegible]

```

elbow_plot <- data.frame(k_values = k_values, wcss = wcss)

ggplot(elbow_plot, aes(x = k_values, y = wcss)) +
  geom_line() +
  geom_point() +
  labs(title = "The Elbow Method",
       x = "Number of Clusters (k)",
       y = "Within-Cluster Sum of Squares (WCSS)") +
  theme_minimal()

```



Here, we can select number of clusters as 6, 7 or 8.

```

train_matrix <- as.matrix(data[, -c(1, 2)])
train_structure <- knn.index(train_matrix)
save(train_structure, file = "knn_structure_mean.RData")

```

```

get_recommendations <- function(data, input_movie, train_matrix, k = 2) {
  filtered_movies <- data[data$movieId %in% input_movie, ]
  test_matrix <- colMeans(filtered_movies, na.rm = TRUE)
  knn_result <- get.knnx(data = train_matrix, query = test_matrix, k = k)
  return(knn_result$nn.index[, 2])
}

```

```

load("knn_structure_mean.RData")
input_movie <- c(1000)
res <- get_second_closest_neighbors(data, input_movie, train_matrix, k = 2)

```

```
cat("The movie recommendation ids are:", res, "\n")
```

```
➡ The movie recommendation ids are: 15596
```

Start coding or [generate](#) with AI.