**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BYAICTE & GOVT.OF KARNATAKA



**MACHINE LEARNING LA REPORT**

on

**Movie recommendation using KNN algorithm**

*Submitted in partial fulfilment of the requirement for the award of Degree of*

*Bachelor of Engineering*

*in*

*Computer Science and Engineering*

*Submitted by:*

*Anish Pokhrel - 1NT18CS191*

*Tej Narayan Chauhan- 1NT18CS209*

*Shreeyut Shretsha- 1NT18CS206*

**

## Department of Computer Science and Engineering

### (Accredited by NBA Tier-1)

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

, APPROVED BY AICTE & GOVT.OF KARNATAKA)

## Department of Computer Science and Engineering

### (Accredited by NBA Tier-1)

****

**CERTIFICATE**

This is to certify that the Report on Movie recommendation using KNN algorithmis an authentic work carried out by **Anish Pokhrel(1NT18CS191) , Shreeeyut Shrestha(1NT18CS206) and TejNarayan Chauhan (1NT18CS209)** bona fide students of **Nitte Meenakshi Institute of Technology**, Bangalore in partial fulfillment for the award of the degree of ***Bachelor of Engineering*** in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belagavi during the academic year ***2021-2022.*** It is certified that all corrections and suggestions indicated during the internal assessment has been incorporated in the report.

**Signature Of Guide Signature of the HOD**

**Dr. Vani V Dr. Sarojadevi H Professor,**

**Course coordinator Head, Dept. CSE**

**NMIT Bangalore. NMIT Bangalore.**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. We express my sincere gratitude to our Principal **Dr.**

**H. C. Nagaraj**, Nitte Meenakshi Institute of Technology for providing facilities.

We wish to thank our HoD**, Dr. SarojaDevi H** for the excellent environment created to further educational growth in our college. We also thank him for the invaluable guidance provided which has helped in the creation of a better project.

Thanks to our Departmental Project coordinators. We also thank all our friends, teaching and non- teaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the project.

|  |  |  |
| --- | --- | --- |
| **Name** | **USN** | **Signature** |
| Shreeyut Shrestha | 1NT18CS206 |  |
| Tej Narayan Chauhan | 1NT18CS209 |  |
| Anish Pokhrel | 1NT18CS191 |  |

# Table Of Content

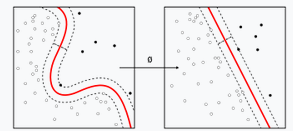
**1. Abstract  
2. Introduction  
2.1. Motivation  
2.2. Problem Domain  
2.3. Aim and Objectives  
3. Data Source and Data Quality  
4. Data Pre-processing  
5. Machine Learning Methods  
6. Results and Discussions  
7. Conclusion & Future Directions  
8. Lessons learnt  
9. References  
10. Appendix – Link to the dataset chosen  
10.1.1. Python Codes Implemented**

**1. Abstract**

This learning activity will entail designing a machine learning algorithm that uses the basic K Nearest Neighbors algorithm to find the most suitable movie recommendation for a particular individual based on their prior preferences. We will be looking at the various identifiers in our library to identify key traits in those movies and also look at the features of the movies that the user has previously watched and liked to give them new movie choices that they have not viewed before.

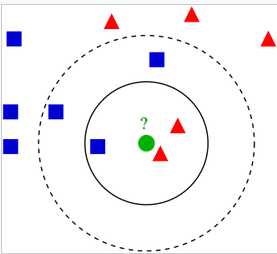
**2. Introduction**

Machine learning is the study of specific computer algorithm that can improve through either the efforts of random chance or with specific inputs of the programmer with the data that was provided to it. It can be taken as a sub-genre of artificial intelligence as they are not essentially hard coded code but instead as this fluid entity that seeks to complete an objective with the closest output vector.



Machine learning is closely related to computational statistics as we are trying to produce the model with the best fit to the data that we possess with trying to avoid issues such as over-fitting and under-fitting so that our algorithm can be used in almost all general use case topics that might come into the algorithms field. Predictions are made using computers with mathematical optimization delivery methods. There are many algorithms that fit the bill to achieve this task. For our project we will be looking at K Nearest Neighbors algorithm to achieve our goals.

KNN algorithm is a non-parametric classification method that was first developed by Evelyn Fix and Joseph Hodges in 1951. It is an algorithm based around classification and regression. It takes the input that is the k closest to the training example from the data set.



**2.1. Motivation**

Our generation is the century of innovation and with that innovation comes a huge influx into each market that can use these new technologies. The new camera improvements, developments in microphone and sound techs in general has brought a huge boom to the movie industry in the past industry. Even with the current pandemic people have been producing high volumes of video format entertainment services that include movies, videos, streams etc. With this newfound volume in the content that we can consume also comes the problem of endless choices to make in what we choose to entertain ourselves with. To produce a simple solution to the problem we wish to classify each movie to their raw traits on their genre, topic, actors etc. so that we can correlate that data to what any particular user finds entertaining to produce a short list of incredibly likable movies for the user. It is a project that hopes to form a means of convenience for any individual in this next generational age with its next generation problems

**2.2. Problem Domain**

The project has a few key identifying traits to it. We have the name and various features of a high selection of movies. The task requires the data to be sifted through to find similar movies based on their genre, cast and other factors. This information then must be cross referenced with the data on the movie that the user has chosen as movies they liked using the K nearest neighbors’ algorithm to find the closest movies in our raw data set. A model must be generated with the data set to classify the movies into what accurate classifications the algorithm can use to find the best recommendation.

**2.3. Aim and Objectives**

The aims for the project can be specified as the following:

* Classify the movies in the data set according to their key attributes.
* Make a model to find the best fit neighbor to the movies that the user has watched.
* Figure out the key attributes in movies that the user finds entertaining i.e., what they prefer to watch
* Process the raw data into movie, cast, ratings, tags, genre and others.

**3. Data source and Data quality**

We will use the dataset from the Donate-a-cry campaign. The dataset is obtained from the following link –

<https://grouplens.org/datasets/movielens/latest/>

This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from [MovieLens](http://movielens.org/), a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018.

The content in the file are as follows

1. User ID: MovieLens users were selected at random for inclusion. Their ids have been anonymized. User ids are consistent between ratings.csv and tags.csv.
2. Movie ID: Only movies with at least one rating or tag are included in the dataset. These movie ids are consistent with those used on the MovieLens web site.
3. Ratings Data File Structure: All ratings are contained in the file ratings.csv. Each line of this file after the header row represents one rating of one movie by one user, and has the following format:

userId, movieId, rating, timestamp

1. Movie Data File Structure: Movie information is contained in the file movies.csv. Each line of this file after the header row represents one movie, and has the following format:

movieId, title, genres

Genres are a pipe-separated list, and are selected from the following:

* Action
* Adventure
* Animation
* Children's
* Comedy
* Crime
* Documentary
* Drama
* Fantasy
* Film-Noir
* Horror
* Musical
* Mystery
* Romance
* Sci-Fi
* Thriller
* War
* Western
* (no genres listed)

1. Links Data File Structure : Identifiers that can be used to link to other sources of movie data are contained in the file links.csv. Each line of this file after the header row represents one movie, and has the following format:

movieId,imdbId,tmdbId

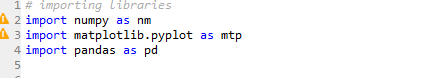
As for the quality of the data, Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

Thus, we can come to the conclusion that the data is fairly unbiased and the information can be regarded as a decent representation of the real-world values for the subjective data such as ratings and tags. As for the other attributes such as genre and name of the movie they have been verified thoroughly and so are completely accurate.

**4. Data pre-processing**

The following steps will be performed in data pre-processing in our project:

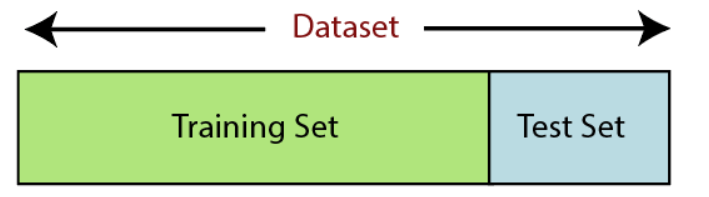
* Getting the data: We will be getting our data from the aforementioned site group lens and we will be using the movie lens data that they have collected in the form of csv files.
* Import libraries: Then we will proceed by importing the necessary libraries to process our information which will be
  + Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. It will be imported as below:



* Import the dataset: Now we need to import the datasets which we have collected for our machine learning project. Now to import the dataset, we will use read\_csv() function of pandas library, which is used to read a csv file and performs various operations on it. Using this function, we can read a csv file locally as well as through an URL.

We can use read\_csv function as below:



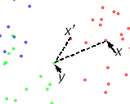
* Handle missing data: Then we can handle missing data so when we are creating our model it doesn’t run into any problems. The main ways to handle missing data are as follows:
  + Deleting the particular row: If a row contains some missing information, then we can just delete the entire row and carry on with our modelling.
  + By calculating the mean: If the missing value of the row is a non-discrete, continuous value then we can assign the missing value of the row as the mean of the entire column as it wont interfere with our calculations.
* Encode the categorical data: The KNN machine learning algorithm works on the basis of mathematical approximation it would be suitable to encode the categorical data such as genre and tags as numerical representation for easy and quick computation.
* Split the data into training and test set: After all that we will have to decide what data from the data set will be used for the purpose of training the model and what portion will be used for testing the data set. For our purpose as we want to be as unbiased as possible, we will be using an arbitrary random selection of the data.
* Feature scaling: For the purpose in the mathematical computation in the KNN algorithm we will be using the non-Euclidean distance between the values to generate the model to find the closest neighbor. Here we will standardize the independent variables in the data table to a specific range. This is performed so that no variable will dominate the other variable.

**5. Machine learning methods:**

In our project we will only be using one machine learning method that is K Nearest Neighbors algorithm. KNN algorithm is a non-parametric classification method that was first developed by Evelyn Fix and Joseph Hodges in 1951. It is an algorithm based around classification and regression. It takes the input that is the k closest to the training example from the data set.

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.

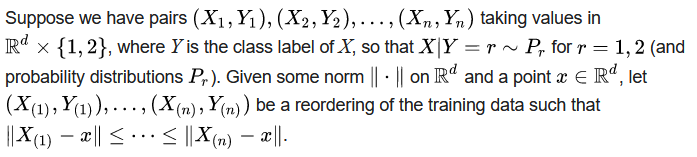
A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or Hamming distance). In the context of gene expression microarray data, for example, k-NN has been employed with correlation coefficients, such as Pearson and Spearman, as a metric.[6] Often, the classification accuracy of k-NN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbor or Neighborhood components analysis.

A drawback of the basic "majority voting" classification occurs when the class distribution is skewed. That is, examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the k nearest neighbors due to their large number.[7] One way to overcome this problem is to weight the classification, taking into account the distance from the test point to each of its k nearest neighbors. The class (or value, in regression problems) of each of the k nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the test point. Another way to overcome skew is by abstraction in data representation. For example, in a self-organizing map (SOM), each node is a representative (a center) of a cluster of similar points, regardless of their density in the original training data. K-NN can then be applied to the SOM.

The best choice of *k* depends upon the data; generally, larger values of *k* reduce effect of the noise on the classification,[8] but make boundaries between classes less distinct. A good *k* can be selected by various heuristic techniques. The special case where the class is predicted to be the class of the closest training sample (i.e. when *k* = 1) is called the nearest neighbor algorithm.

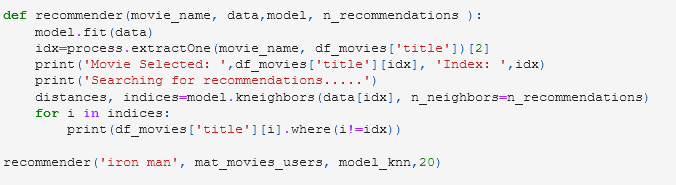
The accuracy of the *k*-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification. A particularly popularapproach is the use of evolutionary algorithms to optimize feature scaling.[9] Another popular approach is to scale features by the mutual information of the training data with the training classes.

In binary (two class) classification problems, it is helpful to choose *k* to be an odd number as this avoids tied votes. One popular way of choosing the empirically optimal *k* in this setting is via bootstrap method.[10]

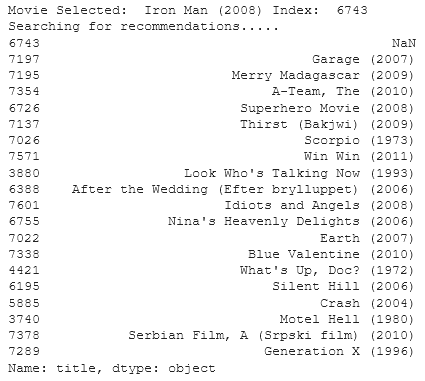


**6. Results and discussions:**

Here the model was generated and we got an accurate movie recommendation related to the movie liked by the exemplary user.



Iron man was taken as the movie of choice and the algorithm correctly identified its genre as action, adventure with some other tags such as superhero and other movies of similar time period as shown below.



1. **Conclusion and future directions:**

Thus, the model that was generated was fairly accurate through our tests. It successfully identified key attributes in each of the movie in both the training sets and the movies that we chose to test it with to find exciting new movie recommendations for movies that weren’t just a carbon copy of the first movie. Instead, it recommended movies that had unique similarities to the given movie such as time period and others.

In the future the program can be improved in many ways. More variables can be brought into the fray to bring out newer connections between movies. The model can also benefit from being able to take a wider amount of input from the user such as movie recommendations based on actors or directors that they like. The dataset that was used was also quite limiting for other purposes so it could be combined with other data sets in future iterations to bring out more selection options. Another improvement that could be made is an integration with some online movie rating site so we can have access to almost all the movies that have ever come out and a more accurate reading on the correct real-world rating of the movie based on the experiences of a vastly larger sample size that the rating sites host. Rating from the user and critics can be separated into their own categories to create a new variable as critic and viewer score has been known to vary among the movies.

1. **Lessons learnt:**

Our group learnt valuable new skills on the proper procedure for creating a machine learning algorithm. We especially found the proper way to handle the pre processing of data very important as it is the building block of the entire project and so has to be done absolutely by the book to make sure the algorithm is made without any underlying error from the data being bad.

We learnt how to properly implement the KNN algorithm that we had researched about during our lectures. It was a good chance to apply the knowledge that we had accumulated over the course.

Finally, we also learnt how to use the inbuilt python function to easily produce our algorithm. For this we learnt the various functions in numpy, matplot and pandas library.

**9. References**

1. Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.A
2. The definition "without being explicitly programmed" is often attributed to Arthur Samuel, who coined the term "machine learning" in 1959, but the phrase is not found verbatim in this publication, and may be a paraphrase that appeared later. Confer "Paraphrasing Arthur Samuel (1959), the question is: How can computers learn to solve problems without being explicitly programmed?" in Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. doi:10.1007/978-94-009-0279-4\_9.www.javatpoint.com. 2022. *Data Preprocessing in Machine learning - Javatpoint*. [online] Available at: <https://www.javatpoint.com/data-preprocessing-machine-learning> [Accessed 17 January 2022].F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. <https://doi.org/10.1145/2827872>
3. Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F., "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning" IEEE Transactions on Vehicular Technology, 2020.Jaskowiak, Pablo A.; Campello, Ricardo J. G. B. "Comparing Correlation Coefficients as Dissimilarity Measures for Cancer Classification in Gene Expression Data". Brazilian Symposium on Bioinformatics (BSB 2011): 1–8. CiteSeerX 10.1.1.208.993.A
4. Coomans, Danny; Massart, Desire L. (1982). "Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules". Analytica Chimica Acta. 136: 15–27.
5. Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks". Medium. Retrieved 2021-08-15. Domingos 2015, Chapter 6, Chapter 7.
6. Jaskowiak, Pablo A.; Campello, Ricardo J. G. B. "Comparing Correlation Coefficients as Dissimilarity Measures for Cancer Classification in Gene Expression Data". Brazilian Symposium on Bioinformatics (BSB 2011): 1–8. CiteSeerX 10.1.1.208.993.
7. Coomans, Danny; Massart, Desire L. (1982). "Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules". Analytica Chimica Acta. 136: 15–27.
8. Everitt, Brian S.; Landau, Sabine; Leese, Morven; and Stahl, Daniel (2011) "Miscellaneous Clustering Methods", in *Cluster Analysis*, 5th Edition, John Wiley & Sons, Ltd., Chichester, UK
9. Nigsch, Florian; Bender, Andreas; van Buuren, Bernd; Tissen, Jos; Nigsch, Eduard; Mitchell, John B. O. (2006). "Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization". Journal of Chemical Information and Modeling. **46** (6): 2412–2422. [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1021/ci060149f](https://doi.org/10.1021%2Fci060149f).
10. *Hall, Peter; Park, Byeong U.; Samworth, Richard J. (2008). "Choice of neighbor order in nearest-neighbor classification".* [*Annals of Statistics*](https://en.wikipedia.org/wiki/Annals_of_Statistics)*.* ***36*** *(5): 2135–2152.* [*arXiv*](https://en.wikipedia.org/wiki/ArXiv_(identifier))*:*[*0810.5276*](https://arxiv.org/abs/0810.5276)*.* [*Bibcode*](https://en.wikipedia.org/wiki/Bibcode_(identifier))*:*[*2008arXiv0810.5276H*](https://ui.adsabs.harvard.edu/abs/2008arXiv0810.5276H)*.* [*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1214/07-AOS537*](https://doi.org/10.1214%2F07-AOS537)*.* [*S2CID*](https://en.wikipedia.org/wiki/S2CID_(identifier))[*14059866*](https://api.semanticscholar.org/CorpusID:14059866)*.*

**10) Appendix**

**Link to the dataset chosen-**

[**https://grouplens.org/datasets/movielens/latest/**](https://grouplens.org/datasets/movielens/latest/)

**Python codes implemented**

import pandas as pd

from scipy.sparse import csr\_matrix

from sklearn.neighbors import NearestNeighbors

from fuzzywuzzy import process

movies='movies.csv'

ratings='ratings.csv'

df\_movies=pd.read\_csv(movies, usecols=['movieId','title'], dtype={'movieId':'int32','title':'str'})

df\_ratings=pd.read\_csv(ratings, usecols=['userId','movieId','rating'],dtype={'userId':'int32',

movies\_users=df\_ratings.pivot(index='movieId', columns='userId',values='rating').fillna(0)

mat\_movies\_users=csr\_matrix(movies\_users.values)

model\_knn= NearestNeighbors(metric='cosine', algorithm='brute', n\_neighbors=20)

model\_knn.fit(mat\_movies\_users)

def recommender(movie\_name, data,model, n\_recommendations ):

model.fit(data)

idx=process.extractOne(movie\_name, df\_movies['title'])[2]

print('Movie Selected: ',df\_movies['title'][idx], 'Index: ',idx)

print('Searching for recommendations.....')

distances, indices=model.kneighbors(data[idx], n\_neighbors=n\_recommendations)

for i in indices:

print(df\_movies['title'][i].where(i!=idx))

recommender('iron man', mat\_movies\_users, model\_knn,20)