**ARTIFICIAL INTELLIGENCE**

**PROJECT-01 REPORT**

A close up of a sign

Description automatically generated

**Submitted By:**

Faiqa Irfan

2017-CE-62

Areej Mirza

2017-CE-64

Rukhsar Ijaz

2017-CE-68

**Submitted to:**

Dr. Awais Hassan

**Date:** December 09,2019

**UNIVERSITY OF ENGINEERING AND TECHNOLOGY, LAHORE**

**Movie Recommendation System Using K-**

**Nearest Neighbors:**

**Abstract:**

We were given to make a recommendation system so we selected to make a movie recommendation system. We basically used KNN method for this which is simplest classification algorithm and it is one of the most used learning algorithms. First of all we created a virtual environment for downloading flask. Then we downloaded pandas. **Pandas** is mainly used for machine learning in form of data frames.  **Pandas** allow importing data of various file formats such as csv, excel etc. We also worked on HTML and CSS for making interface of our project. We were given raw data we also preprocessed it by normalization and standardization and missing values.

**Introduction:**

The project we are going to do is a Recommendation System. **Recommender systems** help the users to get personalized **recommendations**, helps users to take correct decisions in their online transactions, increase sales and redefine the users web browsing experience, retain the customers, enhance their shopping experience.  **Recommendation** engines provide personalization. There are many recommendation systems like books, jobs, songs, online tutorials related to our search etc.

Two critically important methods are widely used for recommender systems. One is content based filtering, where we attempt to shape the users preferences using data retrieved, and  
suggest items based on that profile. The other is collaborative filtering, where in we try to cluster alike users together and use data about the group to make recommendations to the customer.

Here we are going to make a recommendation system for movies. The system requires a search input as the name and year of the movie which in return gives a list of other recommended movies based on the same year or same genera.

**Data Set Details:**

Our project instructor provides us with a data set for the recommendation system of movies that includes their id, movie titles, year and generas. There is approximately 9709 movies in that data set and we get 17 columns for each movie after normalization. The link for raw data is given below:

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

[ml-latest-small.zip](http://files.grouplens.org/datasets/movielens/ml-latest-small.zip)

**Data Preprocessing Details:**

**Data preprocessing** is a **data** mining technique that involves transforming raw **data** into an understandable format. Real-world **data** is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. We use normalization method for the preprocessing of our data. We want to reduce the redundancy of our data, speed will be efficient and less memory is used.

**Methods:**

**k-Nearest Neighbor Method:**

1. Determine parameter K = number of nearest neighbors.
2. Calculate the distance between the query-instance and all the training samples.
3. Sort the distance and determine nearest neighbors based on the K-th minimum distance.
4. Gather the category of the nearest neighbors.
5. Use simple majority of the category of nearest neighbors as the prediction value of the query instance.

**Architecture of program:**

**Implantation Details:**

1. Download the data and normalize it.
2. Create virtual environment and install flask in order to connect with server.
3. Install libraries to read data such as pandas, NumPy etc.
4. Apply knn to the normalized data.
5. Design an interface using HTML & CSS and linked it through flask
6. Run project on IDLE.

**Screens:**

**Execution of knn code with flask and html results in:**

A screenshot of a computer screen

Description automatically generated

**Web Page against the above URL in cmd is given below with two different instances:**

* Search results for Wild Wild West (1999)

**A screenshot of a computer screen

Description automatically generated**

* When we search for bedrooms & hallways

A screenshot of a computer screen

Description automatically generated

**Experiments and Results:**

Enter the movie title in the search box. If the generas of recommended movies matched with the search movies genera in the given data. It guaranteed the correctness and validation of the program and project.

**Instructions to execute the Program:**

The downloadable link of GitHub from the program is:

<https://github.com/rukhsar68/Movies-Recommender/blob/master/recommended%20system.zip>

1. Run the knn code on cmd.
2. A URL is received.
3. Copy the URL and paste on any browser(we use chrome).
4. A web page is opened displaying a search box and search button.
5. Enter any movie title from data in search box .
6. Click search button.
7. Recommended list of movies will appear on the left of the page.

**References:**

<https://github.com/Nazanin1369/DataMining-KNN>

<https://medium.com/@dhiraj.sanju/build-product-recommendation-system-using-turicreate-and-deploy-to-production-using-flask-and-32d9689bd8e4>