

**Final Project Report**

**Course Recommendation System**

By

Yashwanth Reddy Boddireddy (yb244)

Bindu Aradhy Janagani( )

Sovit Nayak( )  
  
**Professor:** Khalid  
DS 675 – Machine Learning

Contents

[Abstract: 3](#_Toc165409223)

[Introduction 3](#_Toc165409224)

[Project Workflow 4](#_Toc165409225)

[Dataset: 4](#_Toc165409226)

[Data Pre-processing and EDA: 4](#_Toc165409227)

[Model Training and Evaluation: 10](#_Toc165409228)

[TF-IDF: 10](#_Toc165409229)

[Results: 10](#_Toc165409230)

[We have included other features to the cosine similarity model with TF-IDF: 11](#_Toc165409231)

[Word2Vec: 12](#_Toc165409232)

[Results: 13](#_Toc165409233)

[Random Forest: 13](#_Toc165409234)

[Hyperparameter Tuning: 13](#_Toc165409235)

[Results: 14](#_Toc165409236)

[Autoencoder: 15](#_Toc165409237)

[Results: 16](#_Toc165409238)

[KNN: 17](#_Toc165409239)

[Results: 18](#_Toc165409240)

[Classification of Course Level: 18](#_Toc165409241)

[Results: 19](#_Toc165409242)

[Conclusion 20](#_Toc165409243)

[Recommendations and Future Work: 20](#_Toc165409244)

[Steps to run the Program: 21](#_Toc165409245)

[STEP 1: Cloning the Repository 21](#_Toc165409246)

[STEP 2: Create Virtual Environment. 21](#_Toc165409247)

[STEP 3: Running the Notebook Locally 21](#_Toc165409248)

[Alternative for STEP 3: Running the Notebook on Google Colab 22](#_Toc165409249)

[References and Links: 22](#_Toc165409250)

# Abstract:

This project develops an advanced recommendation system for online courses by utilizing a combination of content-based filtering techniques and machine learning models. The system integrates textual and categorical data from course descriptions to recommend courses that closely match users' interests and educational needs. Key techniques include TF-IDF vectorization, Word2Vec embeddings, and machine learning models such as Random Forest, KNN and Autoencoders. The models are fine-tuned using hyperparameter optimization to ensure optimal performance. This recommendation system aims to enhance the learning experience by providing personalized course suggestions, thereby improving user engagement and satisfaction in e-learning platforms.

# Introduction

**Background**

In the rapidly expanding domain of e-learning, personalized course recommendations have become essential for enhancing user engagement and educational outcomes. Traditional recommendation systems often fall short in addressing the diverse needs and preferences of learners. This project seeks to bridge this gap by implementing a sophisticated machine learning-based recommendation system that leverages both the content of the courses and user interaction data.

**Objectives**

The primary objective of this project is to develop a robust recommendation system for online courses that can accurately predict and suggest courses based on a user’s past interactions and preferences. To achieve this, the project utilizes a blend of content-based filtering mechanisms and various machine learning techniques to analyze course features such as titles, descriptions, and categorical metadata.

**Methodology**

The project employs several advanced techniques and algorithms, including:

* **TF-IDF Vectorization**: To convert textual data into a numerical format that captures the importance of words in course descriptions.
* **Word2Vec Embeddings**: To generate dense word vectors that capture semantic meanings of the words in course titles and descriptions, enhancing the system's ability to recommend courses with similar content.
* **Machine Learning Models**: Including Random Forest for clustering similar courses based on feature proximity, KNN, and Autoencoders for dimensionality reduction and feature extraction from complex datasets.
* **Hyperparameter Tuning**: Using Grid Search and Keras Tuner to find the optimal settings for each model, ensuring the highest possible accuracy and efficiency of the recommendation system.

**Significance**

By integrating these methods, the recommendation system not only suggests courses that are topically similar but also aligns with the individual learning paths and goals of users. This personalized approach is anticipated to significantly improve user retention and satisfaction, making learning more enjoyable and effective.

# Project Workflow

Dataset:

Coursera Dataset

**Dataset Source:** [Coursera Dataset (kaggle.com)](https://www.kaggle.com/datasets/elvinrustam/coursera-dataset)

**Dataset type:** Tabular Data with 8370 rows and 13 columns

**Features:** (Source Kaggle)

* **Course Title:** This column contains the title of the course offered on Coursera.
* **Rating:** The rating column likely contains the average rating of the course, as provided by users who have completed the course. Ratings are often given on a scale, such as 1 to 5 stars.
* **Level:** This column indicates the difficulty or complexity level of the course. It might categorize courses as beginner, intermediate, or advanced, for example.
* **Schedule:** This column may specify the schedule or timing of the course, such as whether it is a flexible schedule or hands-on learning.
* **What you will learn:** This column likely outlines the learning objectives or topics covered in the course. It provides a summary of the knowledge or skills that participants can expect to gain.
* **Skill gain:** This column may detail the specific skills that participants will acquire upon completion of the course.
* **Modules:** The modules column likely lists the different sections or units that make up the course. It could provide an overview of the course's structure and organization.
* **Instructor:** This column contains information about the instructor(s) or lecturer(s) who teach the course.
* **Offered By:** This column likely specifies the institution or organization offering the course on the Coursera platform.
* **Keyword:** This column may contain keywords or tags associated with the course, which can help users search for relevant courses based on specific topics or themes.
* **Course Url**: This column likely contains the URL or web link to the course page on the Coursera platform.
* **Duration to complete (Approx.):** This column specifies the approximate time required to complete the course. It is given in terms of hours
* **Number of Reviews:** This column contains the count of reviews or ratings submitted by users who have completed the course. It indicates the course's popularity and user satisfaction level.

## Data Pre-processing and EDA:

**Checking for missing values**

A screenshot of a computer

Description automatically generated

**Filled missing values with “Not specified”.**

A close up of text

Description automatically generated

**Dropped rows where “What you will Learn” is equal to “Not specified”.**



**Dropped duplicates checked on “course url”.**

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

**Data Distributions:**

A green and white graph

Description automatically generated

1. **Rating Distribution**: This histogram shows a common pattern in product or service ratings where the majority of ratings are high (4 or 5 stars). Very few ratings are at the lower end (0, 1, or 2 stars). This often indicates general customer satisfaction or a tendency for users to rate only if they had a positive experience.
2. **Duration Distribution**: The duration histogram illustrates a rapid drop-off as duration increases, which is typical for many activities, such as watching movies or engaging in leisure activities. Most values are clustered at the lower end (close to 0-100 hours), suggesting that the duration of the activity is generally short.
3. **Review Count Distribution**: This histogram shows that most items have very few reviews (close to 0), with an extremely sharp decline in frequency as the number of reviews increases. This could imply that only a few items get a large number of reviews, possibly due to popularity or a high sales volume, while the majority receive few to no reviews.

A close-up of a graph

Description automatically generated

1. **Course Level Distribution** (Pie Chart):
   * **Beginner Level**: The majority of courses, about 64.2%, are designed for beginners. This suggests that the platform is heavily focused on introductory material, possibly to attract a wide and varied audience that is new to the subject matter.
   * **Intermediate Level**: Approximately 28.3% of the courses are for intermediate learners. This indicates a decent availability of courses for those who have some foundational knowledge and are looking to deepen their understanding.
   * **Advanced Level**: Only 4.4% of the courses are categorized as advanced. This might reflect a lower demand for higher-level courses or a strategic decision to focus more on accessible content.
   * **Not Specified**: A small fraction of courses (3.1%) do not specify their difficulty level. This could be due to a variety of reasons, such as being applicable to all levels or a lack of classification by the course providers.
2. **Top 10 Keyword Categories** (Bar Chart):
   * The most popular categories for courses are **Computer Science**, **Business**, and **Information Technology**, suggesting a strong focus on technical and professional fields likely driven by high demand in the job market.
   * Other significant areas include **Health**, **Personal Development**, and **Physical Science and Engineering**.
   * **Social Sciences**, **Data Science**, and **Arts and Humanities** show moderate numbers of courses, offering a broad range of subjects.
   * **Math and Logic** courses are less numerous, which might reflect specific audience targeting or the integration of these subjects into other course categories rather than as standalone offerings.

A screenshot of a graph

Description automatically generated

1. **Top 10 Institutions by Number of Courses**:
   * **Coursera Project Network** and **Google Cloud** lead, suggesting a significant presence in online education, likely offering a wide range of technical and career-oriented courses.
   * Traditional universities like **Arizona State University**, **Johns Hopkins University**, and **University of Michigan** show strong online course offerings, indicating their expansion into online education.
   * Tech companies like **Microsoft**, **Meta**, **Google**, and **IBM** are also prominent, which could reflect their focus on providing training in areas relevant to their business, such as AI, cloud computing, and other tech-specific skills.
2. **Top 10 Institutions by Average Rating**:
   * This chart is dominated by less traditional entities like **Deep Teaching Solutions**, **Keller Williams**, and specialized programs like **Anaplan** and **Measure What Matters**, which suggests that niche or specialized courses might receive higher ratings due to targeted content that meets specific learner needs.
   * Prestigious universities like **Stanford University**, **The University of Notre Dame**, and **Nanyang Technological University, Singapore** also feature here, indicating high satisfaction among participants, potentially due to the quality of instruction and course material.
   * The presence of **Radio y Televisión Española** highlights diverse content delivery, possibly through media-rich courses that engage a wide audience.

A red and blue squares

Description automatically generated

1. **Rating and Duration to Complete (Approx.):** The correlation coefficient is 0.17, which indicates a very weak positive relationship. This suggests that longer courses don't necessarily receive better ratings, but there may be a slight tendency for longer courses to have marginally higher ratings.
2. **Rating and Number of Reviews:** The correlation coefficient here is 0.10, also indicating a very weak positive relationship. This means there is little to no strong link between the number of reviews a course receives and its overall rating.
3. **Duration to Complete (Approx.) and Number of Reviews:** The correlation is 0.19, which is still quite weak but slightly stronger than the other two correlations. This suggests that longer courses might receive slightly more reviews, possibly due to a higher level of engagement or investment required from students.

# Model Training and Evaluation:

## TF-IDF:

A close-up of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

### Results:

A screenshot of a computer program

Description automatically generated

## We have included other features to the cosine similarity model with TF-IDF:

A screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

## Word2Vec:

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

### Results:

A white text on a white background

Description automatically generated

## Random Forest:

### Hyperparameter Tuning:

A screenshot of a computer program

Description automatically generated



A screen shot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

### Results:

A white text on a white background

Description automatically generated

## Autoencoder:

A computer screen shot of a white screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer code with many colored text

Description automatically generated with medium confidence

A computer screen shot of a code

Description automatically generated

### Results:

A screenshot of a computer

Description automatically generated

## KNN:

A screen shot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

### Results:

A screenshot of a computer code

Description automatically generated

## Classification of Course Level:

A screenshot of a computer code

Description automatically generated

A computer screen shot of a program

Description automatically generated

### Results:

A screenshot of a computer

Description automatically generated

# Conclusion

This project aimed to develop an effective course recommendation system for Coursera users by leveraging various data science and machine learning techniques. Throughout the project, significant emphasis was placed on understanding the nature of the data, cleaning and preparing it for analysis, and employing sophisticated analytical techniques to derive meaningful insights.

The recommendation system was built using Autoencoders and K Nearest Neighbours (KNN), which facilitated the generation of personalized course suggestions based on user preferences and course content. The system's backbone, feature engineering, involved transforming course descriptions using TF-IDF vectorization, which proved crucial in capturing the textual nuances necessary for accurate recommendations.

The models were rigorously tested and tuned, with the KNN model showing particularly promising results in terms of recommending courses that closely match user interests. The evaluation metrics indicated that the models not only performed well in terms of accuracy but also in the precision and relevance of the recommendations provided.

Despite the successes, there were challenges, particularly in handling sparse data and in the computational intensity required for model training and hyperparameter tuning. However, these challenges were addressed through strategic data sampling and by leveraging efficient algorithms.

## Recommendations and Future Work:

To enhance the recommendation system further, the following steps could be considered:

* **Integration of User Feedback:** Incorporating real-time user feedback to continually refine and adjust the recommendation algorithms.
* **Expanding Data Sources:** Utilizing additional data sources such as user demographic information and historical interaction data could potentially improve the system's accuracy and personalization.
* **Hybrid Recommendation Techniques:** Combining content-based and collaborative filtering methods could provide a more holistic approach to course recommendation.
* **Advanced NLP Techniques:** Employing more advanced natural language processing techniques such as word embeddings or BERT for deeper analysis of course content.

# Steps to run the Program:

Minimum requirements to run the program:

* **CPU**: Intel Core i3 or equivalent AMD
* **RAM**: 4GB (8GB recommended for smoother performance)
* **Storage**: 256GB SSD (for faster read/write speeds)
* **OS**: Windows 10, macOS, or a modern Linux distribution
* **Software**: Latest version of Python, Jupyter Notebook or Visual Studio Code

## STEP 1: Cloning the Repository

Clone the repository, follow these steps:

1. Open the terminal on your machine.
2. Change the current working directory to the location where you want the cloned directory.
3. Type the following command and press Enter:
4. git clone https://github.com/yashwanthreddy7178/ DS675\_Final\_Project\_Course\_Recommendation\_System.git
5. Navigate to the cloned repository:
6. cd stroke-prediction

## STEP 2: Create Virtual Environment.

Create a conda environment after opening the repository. It is recommended to do this project in a virtual environment to avoid conflicts with other Python packages you may have installed.

conda **create** -n myenv python -y  
conda **activate** myenv

or

**python** -m venv **env**  
**source** **env**/bin/activate

Note: On Windows, use `env\Scripts\activate`

## STEP 3: Running the Notebook Locally

Run the notebook locally, follow these steps:

1. Ensure you have Jupyter Notebook installed, if not, install it using pip:
2. pip install notebook.
3. Install the required libraries:
4. pip **install** -r requirements.txt
5. Start the Jupyter Notebook server:

“Jupyter notebook”

1. The command will open a new tab in your default web browser. Navigate to the .ipynb file and open it.
2. Run the cells in the Jupyter Notebook to execute the code.

## Alternative for STEP 3: Running the Notebook on Google Colab

You can also run the notebook on Google Colab without needing to install anything on your local machine:

1. Open the Google Colab website: [Google Colab](https://colab.research.google.com/)
2. Sign in with your Google account if you are not already logged in.
3. Go to File > Open notebook.
4. Select the GitHub tab and enter the URL of the repository.
5. Open the .ipynb file from the list.
6. You can now run the notebook cells one by one.

Remember to save a copy of the notebook to your Google Drive if you make changes and wish to keep them.

Note: Please make sure you have installed the required packages before running the code. If any package is missing, please install it from the requirements.txt file.Conclusion

# References and Links:

**Note to the Grader:** I have created a repository using my personal account. Because my college email is linked with it. If I remove that mail id, I will lose all the benefits of GitHub pro and student benefits. So, please consider this. Thank You.

**GitHub Repository link:** https://github.com/yashwanthreddy7178/

**References:**

1. https://stackoverflow.com/questions/
2. Lectures Notes