**Movie Recommendation System Using (Content-Based Recommendation System) – A Business Perspective**

**Introduction: The Importance of Movie Recommendation Systems in Business**

In today’s entertainment industry, businesses like **Netflix, Amazon Prime, and Disney+** rely on **recommendation systems** to enhance user engagement, increase watch time, and boost revenue.

A **movie recommendation system** helps platforms **retain users** by providing personalized suggestions based on their preferences. This leads to:

* **Better customer satisfaction** 🎯
* **Increased subscriptions** 📈
* **Higher engagement and watch time** ⏳

**Example: How Netflix Uses Movie Recommendations**

Netflix **invests millions** in improving its recommendation algorithms. Their AI-driven system suggests movies based on:

1. **User’s watch history**
2. **Movie genre preferences**
3. **Trending content among similar users**

By doing this, Netflix **reduces user churn**, meaning **more people stay subscribed** instead of leaving due to lack of interesting content.

**Project Overview**

The goal of this project was to **build a simple yet effective movie recommendation system** using data from **TMDB (The Movie Database) API**. The system suggests movies based on **similar descriptions and genres**, allowing users to discover content they might enjoy.

**Steps in the Project**

1. **Data Collection** – Fetching movie data using TMDB API
2. **Data Cleaning & Preprocessing** – Handling missing values and formatting genres
3. **Model Selection** – Using **TF-IDF Vectorization & Cosine Similarity**
4. **Building a Web App** – Implementing a **Gradio-based interface**

**Data Collection & Cleaning**

**1️⃣ Fetching Data from TMDB API**

I started by setting up a **TMDB developer account** and registering for an API key. Using Python’s requests library, I collected **popular movies** (up to 1000 movies) with attributes like:

* **Title**
* **Genres**
* **Overview (Movie Description)**
* **Popularity & Ratings**

**2️⃣ Data Cleaning & Preparation**

To ensure data quality, I performed **various cleaning steps**:  
✅ **Checked for missing values** and replaced them:

* Missing **Overviews** → "No overview available"
* Missing **Release Dates** → "Unknown"

✅ **Mapped Genre IDs** to actual genre names using the TMDB API

✅ **Transformed Genre Data** from a list (e.g., ['Action', 'Comedy']) to a single string ("Action Comedy") for text processing

✅ **Created a Combined Text Column**:

* "Overview + Genres" → Used for similarity calculations

**Model Selection & Performance**

**Why I Chose TF-IDF & Cosine Similarity**

Instead of using deep learning (which requires large datasets), I used **TF-IDF (Term Frequency-Inverse Document Frequency) and Cosine Similarity**, which are great for **text-based similarity tasks**.

**How It Works**

1. **TF-IDF Vectorization** converts the movie descriptions and genres into **numerical representations**.
2. **Cosine Similarity** finds the **top 5 most similar movies** based on this vector representation.

**Advantages of This Model**

✅ **Fast & Efficient** – No need for massive datasets or GPU processing  
✅ **Accurate** – Works well for text-based similarity tasks  
✅ **Lightweight** – Can be deployed as a simple web app

**Deploying the Model with Gradio**

To make the recommendation system **user-friendly**, I built a **Gradio-based web interface**, allowing users to **enter a movie title and get 5 similar recommendations instantly**.

**Example Recommendations**

* **User searches for:** *Maze Runner*
* **System suggests:** *Tomorrow Before After, Maze Runner: The death Cure, Comis Chaos, Deva, Maze Runner: The Scorch Trials*

**Conclusion: Business Value of This System**

This project demonstrates how **businesses in entertainment** can leverage AI-powered recommendations to:  
🚀 **Enhance user engagement**  
🎯 **Increase revenue through subscriptions**  
📊 **Reduce customer churn**

**Next Steps for Improvement**

✅ **Use a Hybrid Model** – Combine content-based filtering with collaborative filtering  
✅ **Improve Search Accuracy** – Implement deep learning-based embeddings

With these improvements, the system can become **more personalized and scalable**, making it a valuable asset for streaming platforms.