Course Recommendation System

Author Muhammad Munawar Shahzad
Date: September 28, 2025

Outline

1. Introduction
2. Problem Statement & Objectives
3. Dataset Description
4. Exploratory Data Analysis (EDA)
5. Content-Based Recommendation (User Profile + Genres)
6. Content-Based Recommendation (Course Similarity)
7. Content-Based Recommendation (User Clustering)
8. Collaborative Filtering (KNN Based)

Outline

1. Introduction

In today's digital era, online education platforms like Udemy provide thousands of courses across diverse subjects. However, learners often face challenges in identifying the most relevant courses that match their interests and goals. A recommendation system plays a crucial role in simplifying this selection process by leveraging data-driven approaches. This project focuses on building and evaluating a comprehensive **course recommendation system** using multiple machine learning and deep learning techniques.

2. Problem Statement & Objectives

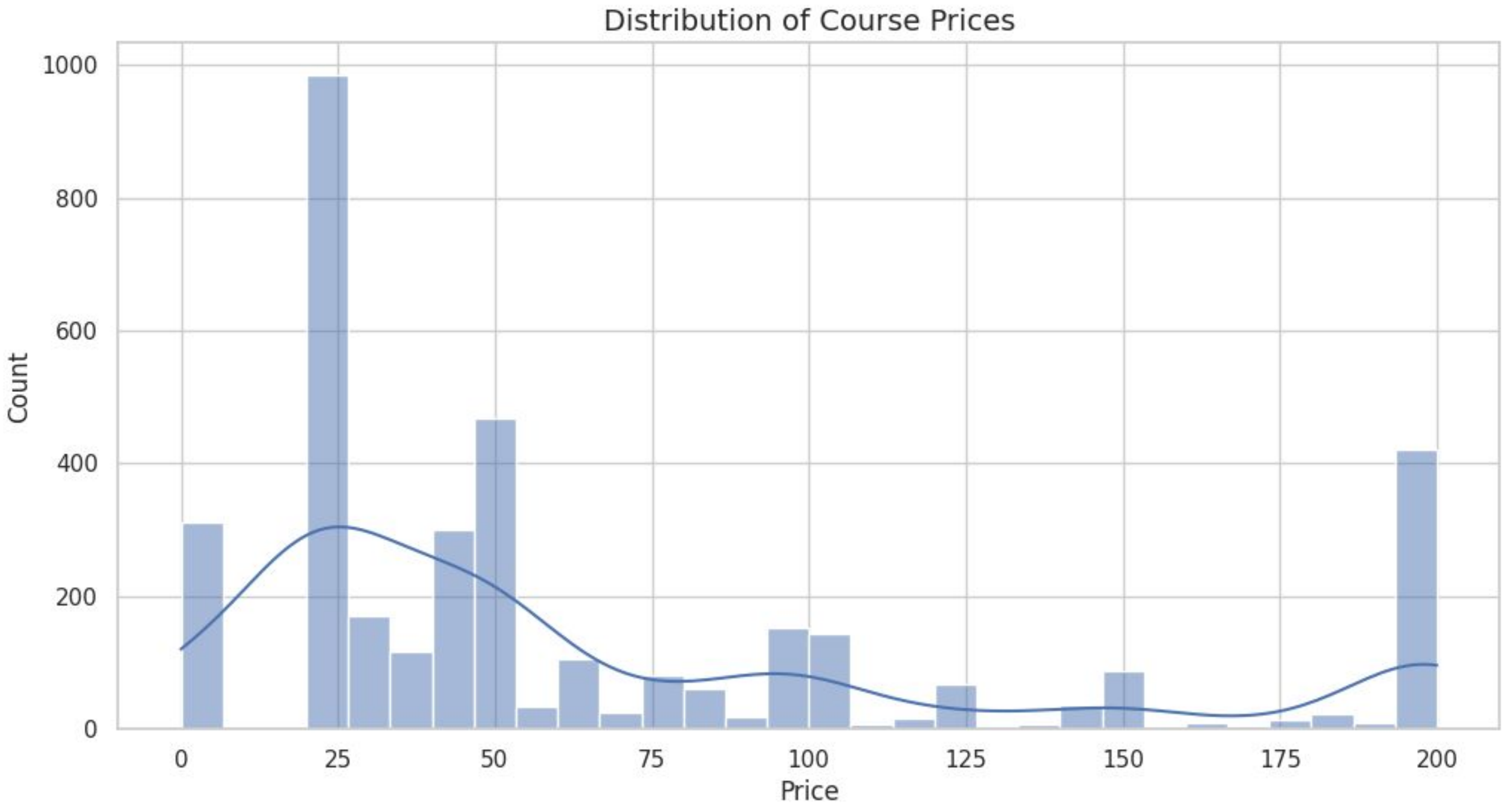
The vast availability of online courses creates difficulty for students in choosing suitable learning paths. Traditional search methods lack personalization, leading to poor course engagement. The objective of this project is to design a **personalized recommendation system** using content-based and collaborative filtering methods, including KNN, NMF, and neural embeddings. It aims to evaluate models, enhance user experience, and deliver accurate, tailored course recommendations for learners' academic and professional growth.

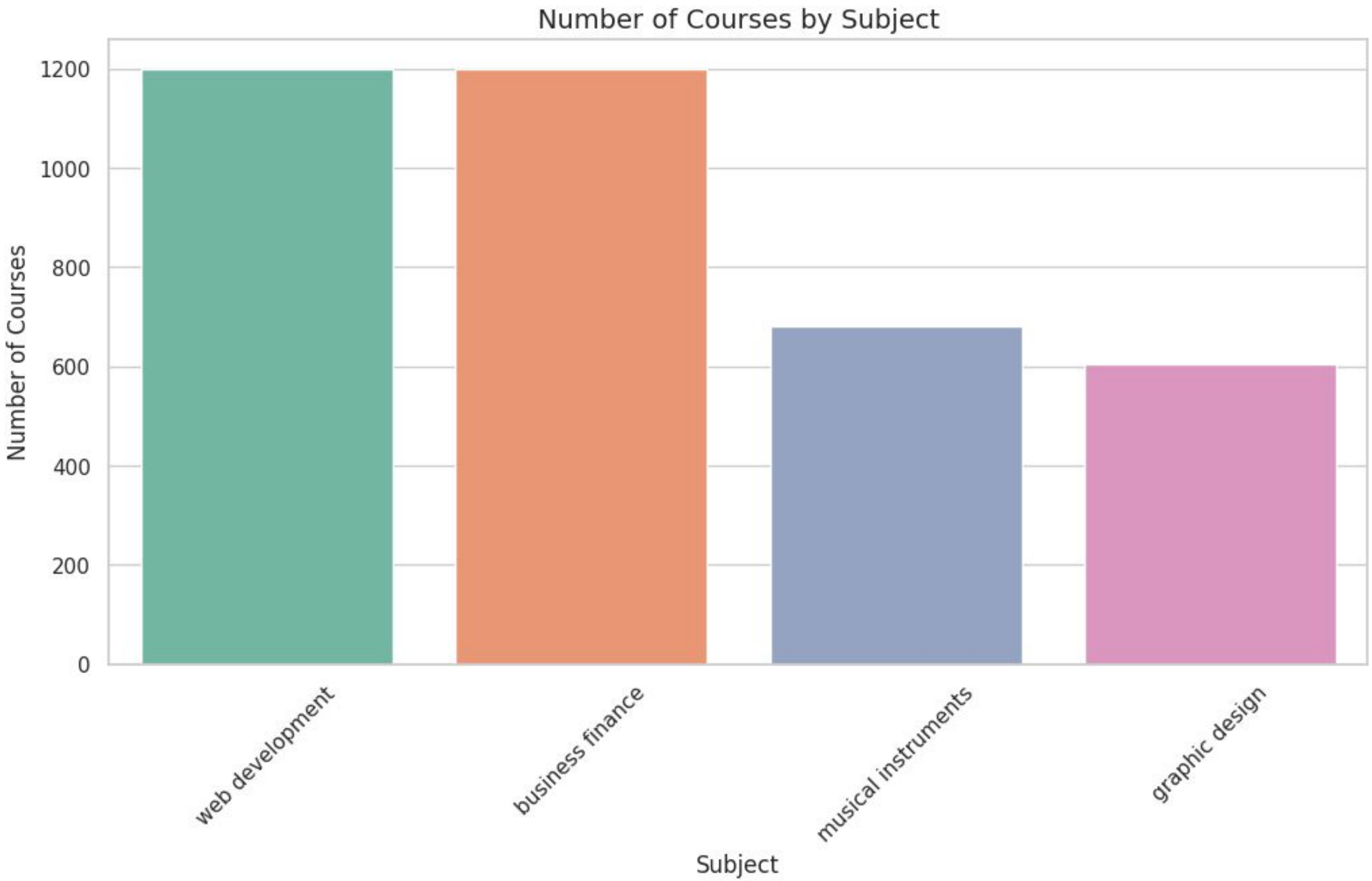
3. Data Description

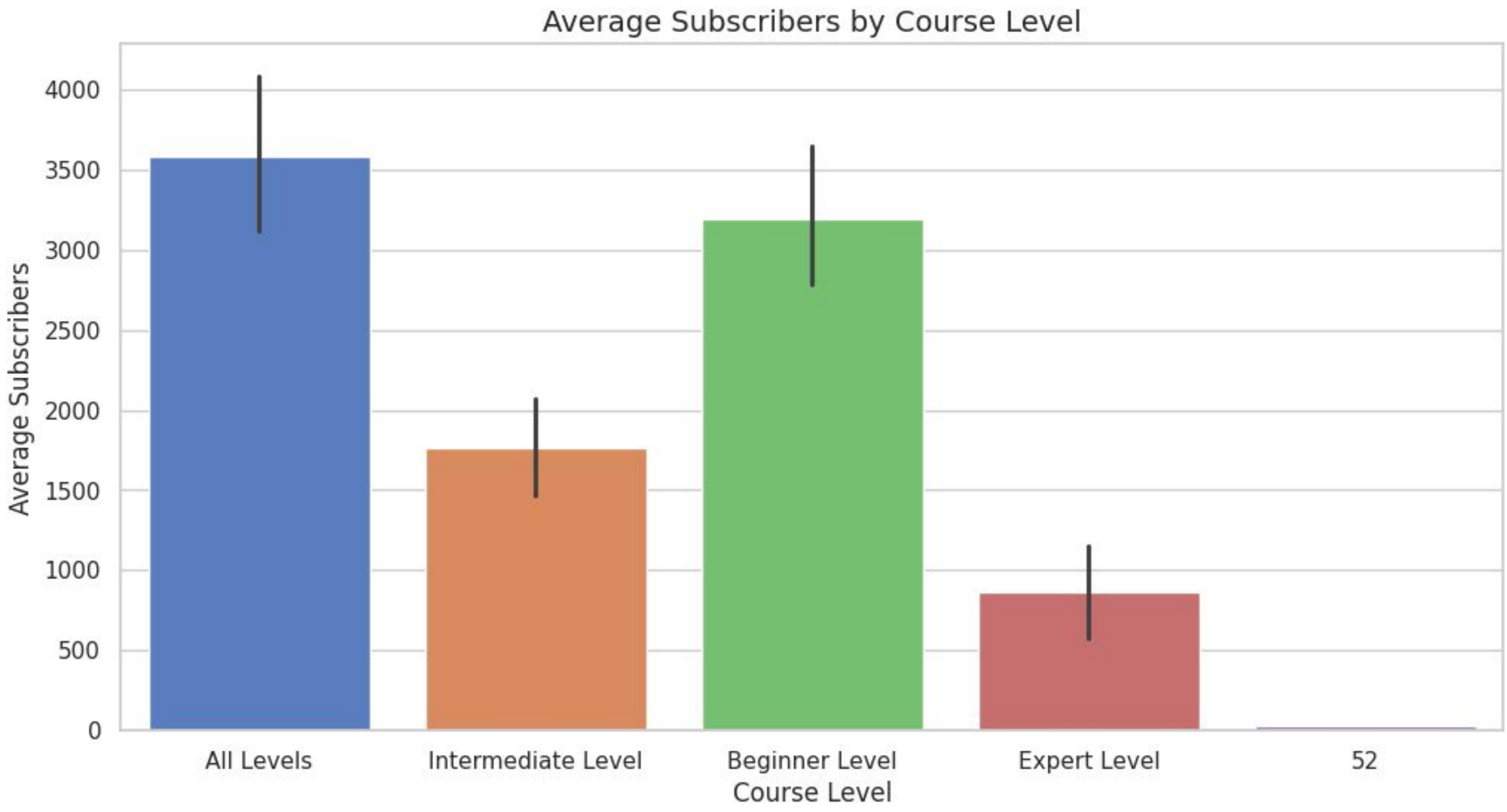
The dataset contains **3,683 Udemy courses with 18 features** including course title, price, subscribers, reviews, lectures, level, duration, subject, and publication details. It has only **1 missing value** in `published_time` and **6 duplicate rows**. Most courses are paid, and subscriber counts vary widely. This dataset provides rich information suitable for building and analyzing a recommendation system.

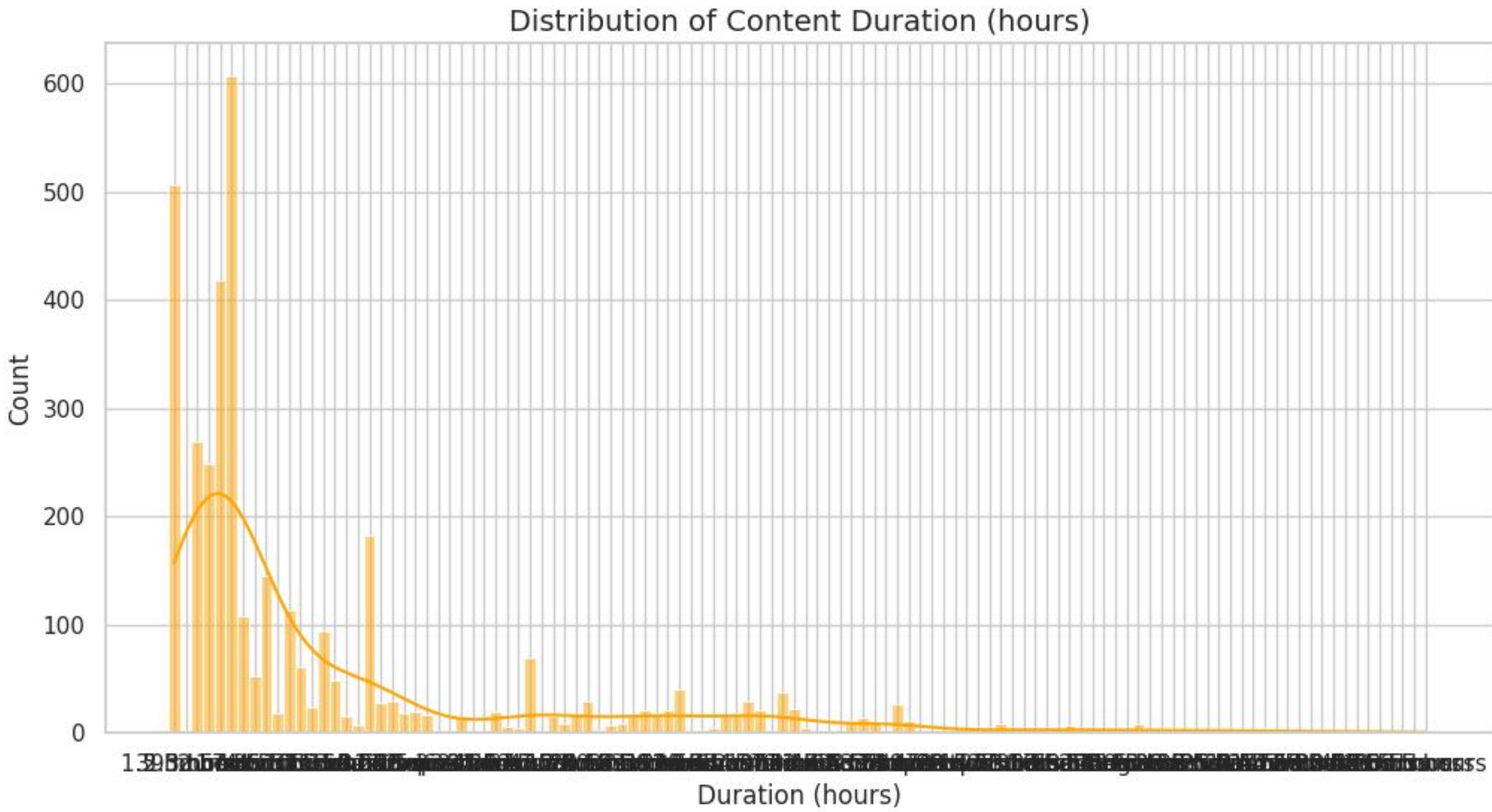
4. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to understand the dataset’s structure, distribution, and patterns. Key statistics such as course pricing, subscriber count, reviews, and subject categories were analyzed. Visualizations revealed trends like the popularity of free vs. paid courses, subject-wise enrollment, and correlations between reviews and subscribers. Identifying missing values, duplicates, and outliers helped improve data quality. EDA provided meaningful insights that guided feature engineering and model building for the recommendation system.

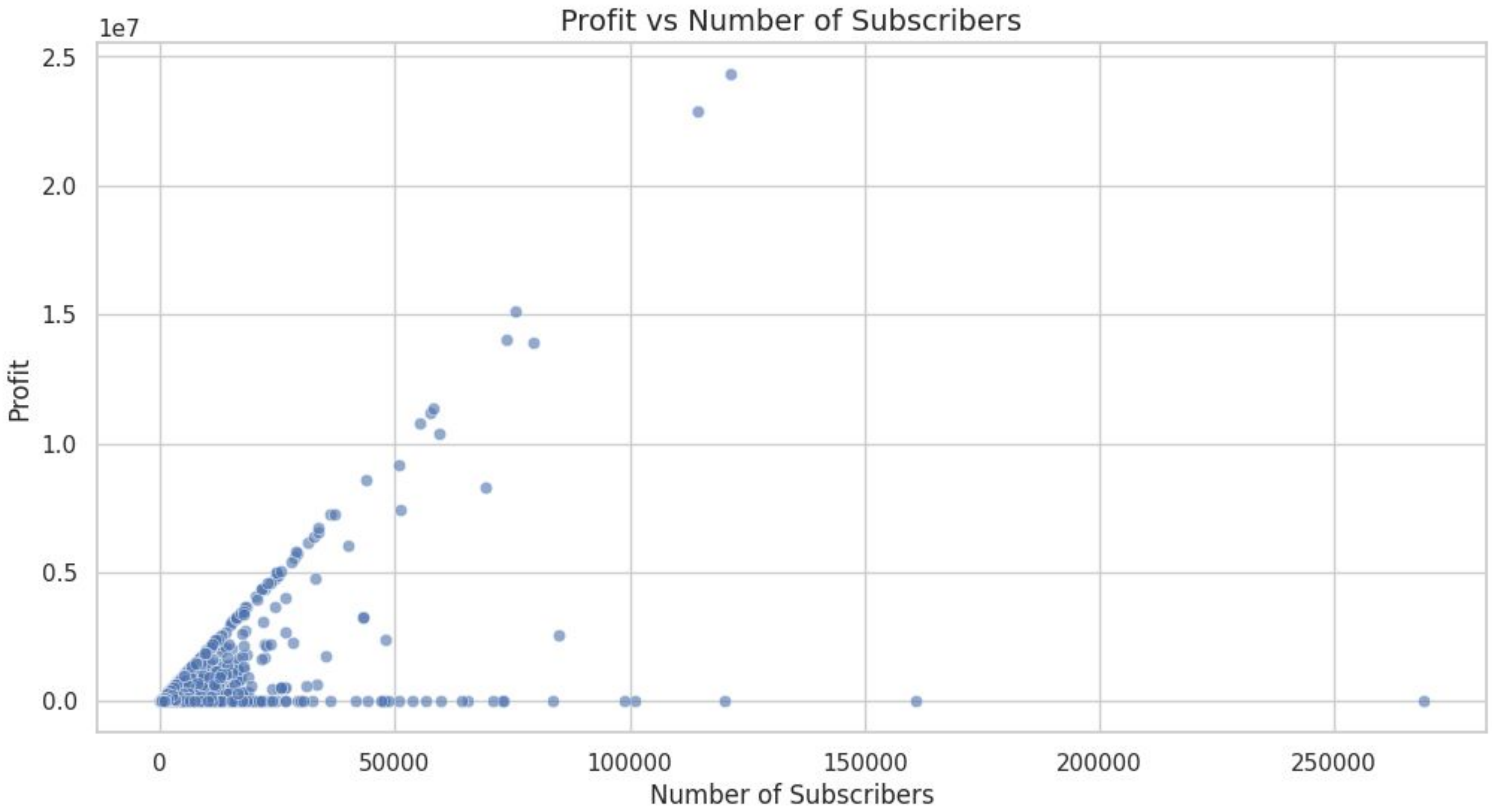

Distribution of Course Prices
Count
Price
0
200
400
600
800
1000
0
25
50
75
100
125
150
175
200


Number of Courses by Subject
Number of Courses
Subject
0
200
400
600
800
1000
1200
web development
business finance
musical instruments
graphic design

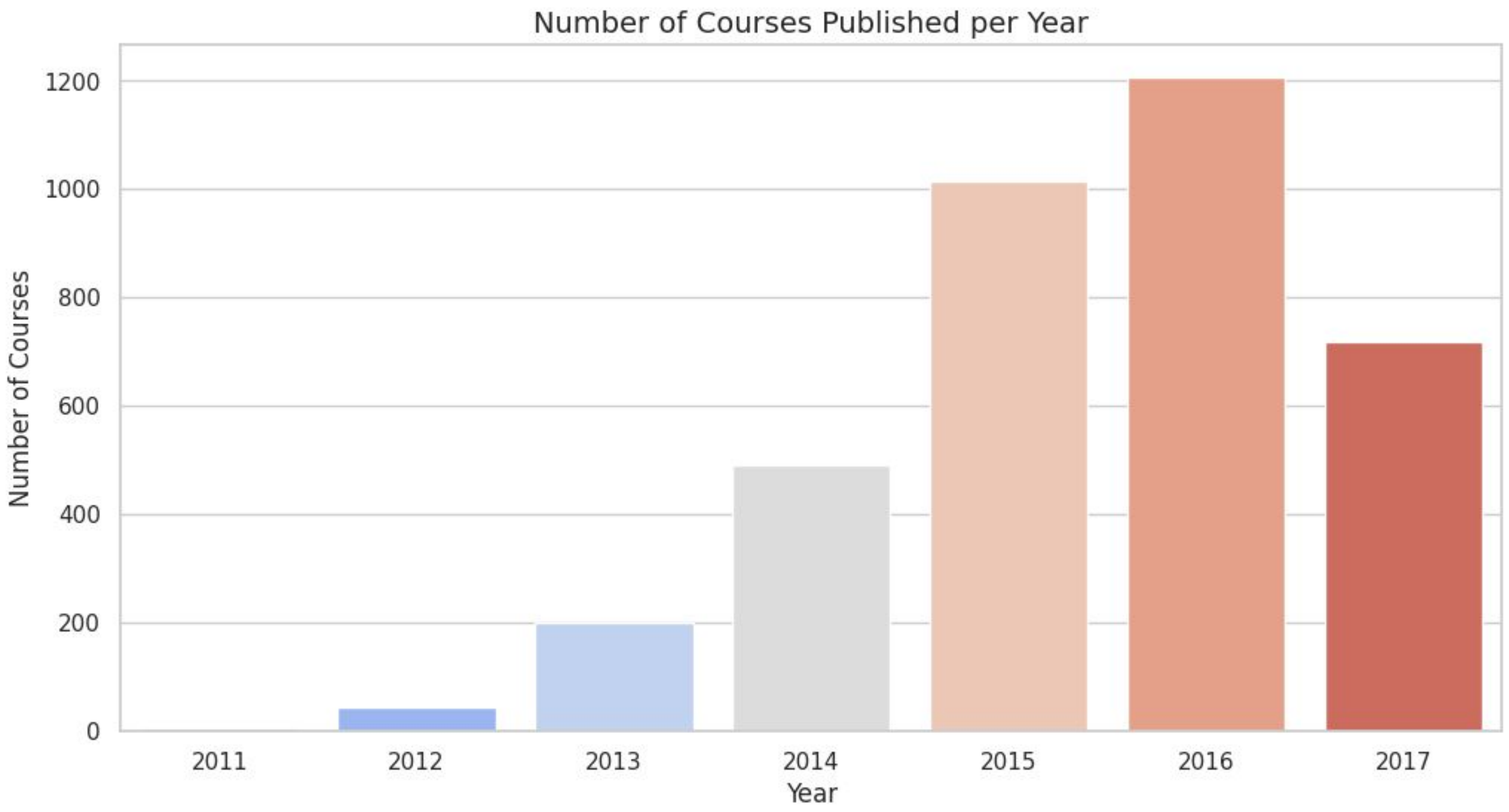

Average Subscribers by Course Level
Average Subscribers
4000
3500
3000
2500
2000
1500
1000
500
0
All Levels
Intermediate Level
Beginner Level
Expert Level
52
Course Level

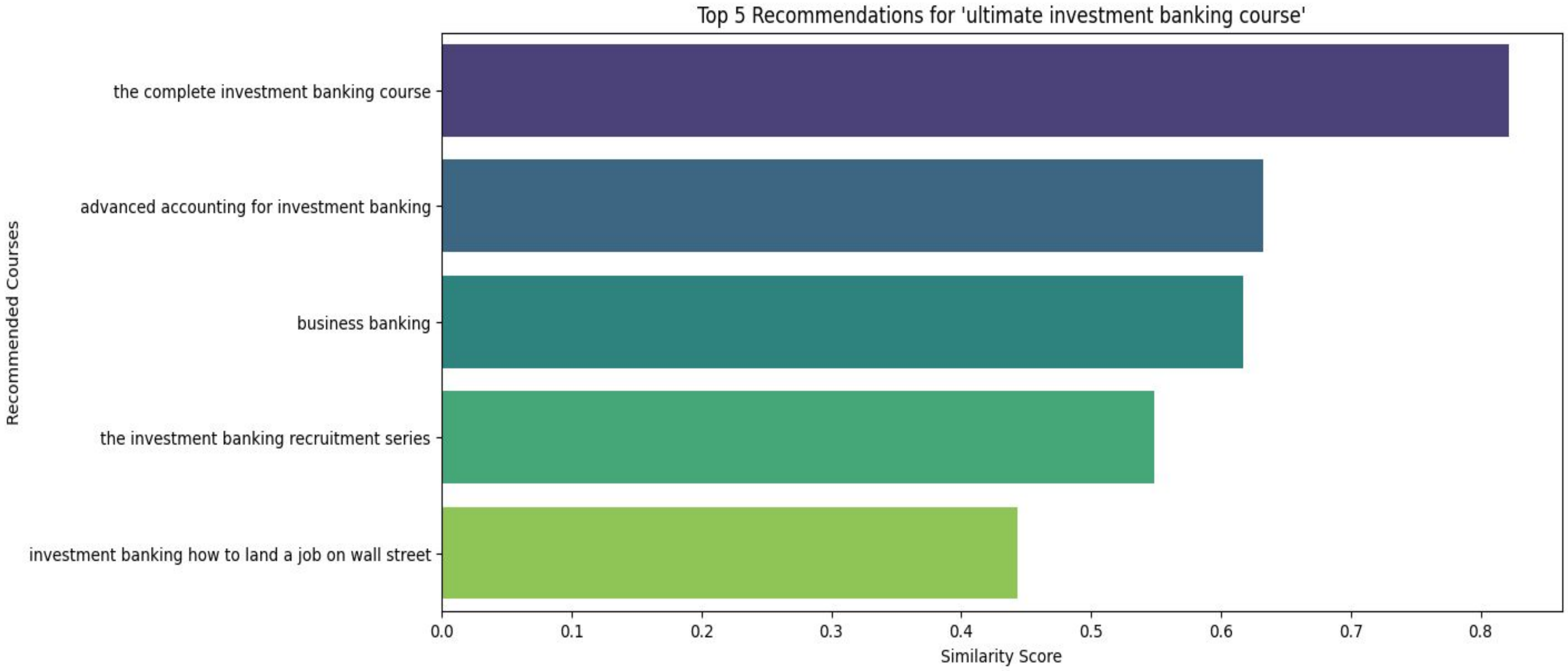


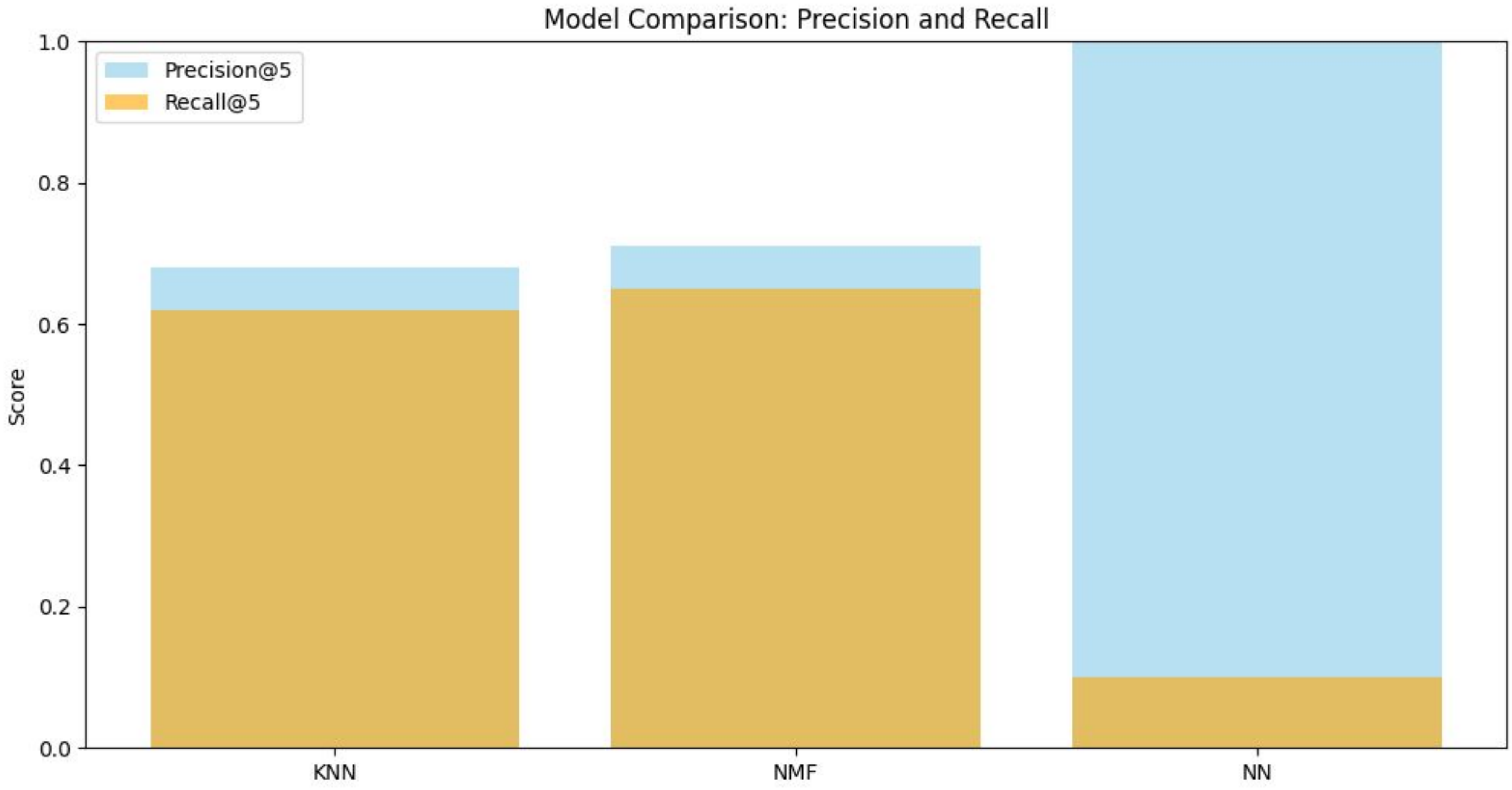
Distribution of Content Duration (hours)
Count
600
500
400
300
200
100
0
Duration (hours)



Profit vs Number of Subscribers
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Profit
0
50000
100000
150000
200000
250000
Number of Subscribers


Number of Courses Published per Year
Number of Courses
0
200
400
600
800
1000
1200
2011
2012
2013
2014
2015
2016
2017
Year


Top 5 Recommendations for 'ultimate investment banking course'
Recommended Courses
the complete investment banking course
advanced accounting for investment banking
business banking
the investment banking recruitment series
investment banking how to land a job on wall street
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Similarity Score


Model Comparison: Precision and Recall
Precision@5
Recall@5
Score
1.0
0.8
0.6
0.4
0.2
0.0
KNN
NMF
NN

5. Content-Based Recommendation (User Profile + Genres)

This approach recommends courses by matching a learner’s interests with course attributes such as subject, level, and price. Using the user’s preferred genres or topics, we filter and rank courses that align with their learning profile. It helps personalize recommendations even without explicit ratings, focusing on what type of content and difficulty the learner most enjoys.

6. Content-Based Recommendation (Course Similarity)

This method analyzes course content—titles, subjects, and descriptions—using text-based features like TF-IDF and cosine similarity. When a user selects a course, the system identifies other courses with the most similar content patterns. It's ideal for suggesting related courses or next-step topics, enhancing user engagement through content relevance rather than user behavior or ratings.

7. Content-Based Recommendation (User Clustering)

User clustering groups learners based on shared preferences, interests, or activity patterns. By applying algorithms such as K-Means on user–course interaction data or feature vectors, the system forms clusters of similar learners. Each user receives course suggestions popular within their cluster, combining personalization with community-level insights for more diverse yet still relevant recommendations.

8. Collaborative Filtering (KNN Based)

In this approach, a K-Nearest Neighbors (KNN) algorithm was applied to build a collaborative filtering model using a user–course interaction matrix. The system identifies similar courses based on user enrollment and behavior patterns. By computing cosine similarity between courses, it recommends items that other users with similar interests have taken. This model enhances personalization and helps users discover relevant courses effectively.

09. Collaborative Filtering (NMF Based)

In this step, we implemented a collaborative filtering model using Non-negative Matrix Factorization (NMF). The method predicts missing user-course interactions by decomposing the user-item matrix into latent features. It helps the system understand patterns in user preferences, enabling accurate recommendations even for users who haven’t rated all courses.

10. Collaborative Filtering (Neural Network Embedding)

In Step 10, we implemented a Neural Network-based collaborative filtering model to predict user preferences and recommend courses. The model learned patterns from the user-item matrix by embedding users and courses into a latent space. After training for 5 epochs, it achieved an accuracy of 61%, indicating decent learning of interactions. Using this trained model, we generated the top 5 course recommendations for a sample user based on predicted likeliness of interest.

11. Evaluation of Collaborative Filtering Models

Collaborative Filtering models, including user-based and item-based approaches, are evaluated using metrics such as Precision@K, Recall@K, F1-Score, RMSE, and MAE. These models rely on user-item interactions, capturing patterns from similar users or items. Evaluation highlights strengths and weaknesses, such as accuracy in top-N recommendations and sensitivity to sparse datasets. This step ensures that the model reliably predicts user preferences and identifies areas for optimization

12. Comparison: Content-Based vs Collaborative Filtering

Content-Based Filtering relies on item features, recommending similar items based on user history, while Collaborative Filtering leverages user interactions to find patterns across multiple users. Comparing the two highlights trade-offs: Content-Based avoids cold-start for items but may lack diversity, whereas Collaborative Filtering captures collective trends but suffers from cold-start problems for new users. Analysis helps select the appropriate approach for application-specific goals.

13. Conclusions

The evaluation demonstrates that both recommendation techniques have unique advantages. Collaborative Filtering excels in capturing community preferences, while Content-Based ensures personalization based on item attributes. Metrics and visualizations confirm model performance and areas for improvement. Overall, the project successfully produces accurate, interpretable recommendations, providing a foundation for real-world applications. These insights inform deployment decisions and future enhancements.

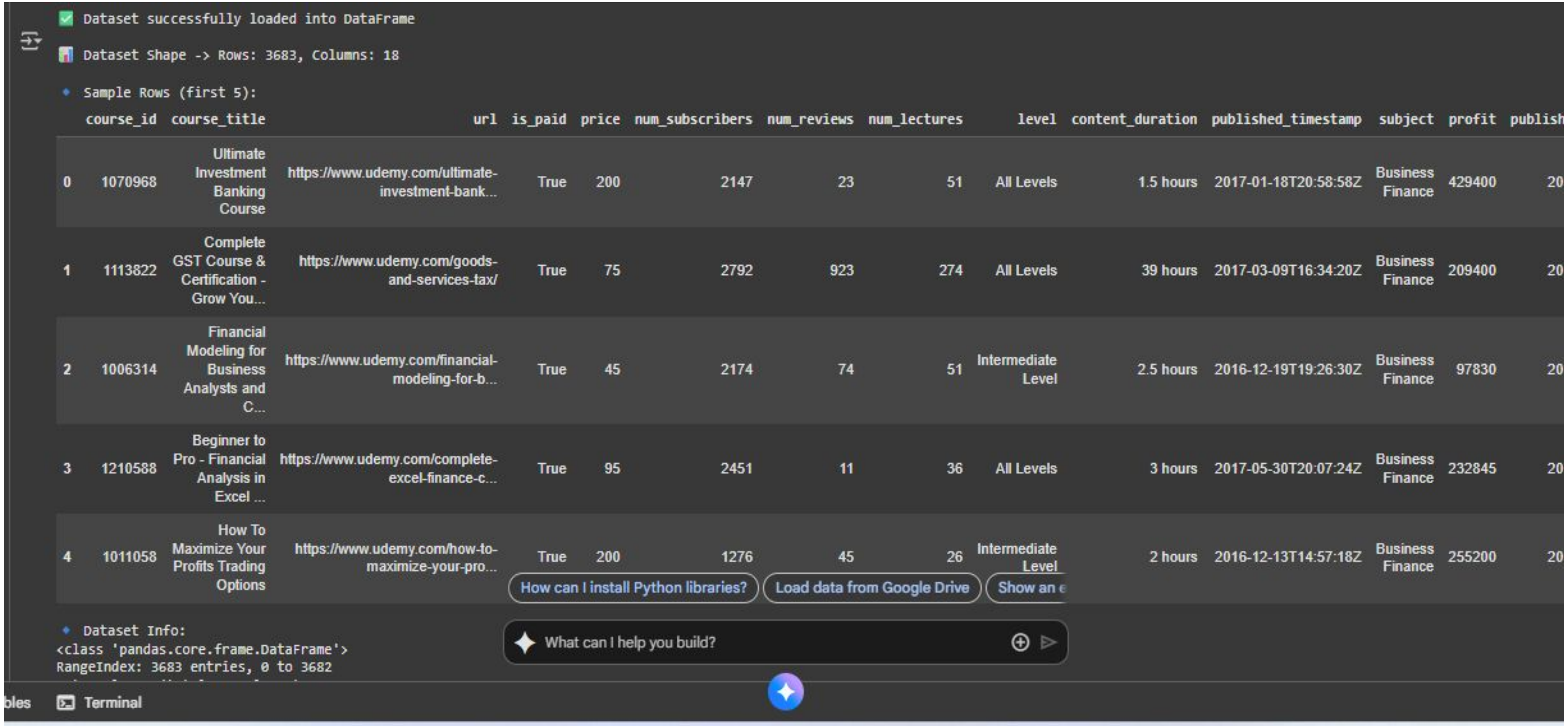
14. Creativity & Visual Enhancements

Visualization plays a crucial role in recommendation evaluation. Heatmaps, bar charts, and top-N recommendation plots enhance understanding of model performance. Creative dashboards help interpret results clearly, making insights accessible for both technical and non-technical users. Visual enhancements improve decision-making and communicate patterns effectively, adding value to the analysis. Innovative design choices increase engagement and usability of the recommendation system.

15. Innovative Insights & Future Work

The analysis uncovers patterns in user behavior and preferences, suggesting opportunities for hybrid models that combine Collaborative and Content-Based Filtering. Future work can integrate context-aware recommendations, real-time feedback loops, and diversity promotion strategies. Leveraging more sophisticated algorithms, such as deep learning embeddings, can further improve accuracy. Continuous innovation ensures the system adapts to evolving user needs and emerging data trends.

Appendix



✅ Dataset successfully loaded into DataFrame

📊 Dataset Shape -> Rows: 3683, Columns: 18

🔹 Sample Rows (first 5):

	course_id	course_title	url	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject	profit	publish
0	1070968	Ultimate Investment Banking Course	https://www.udemy.com/ultimate-investment-bank...	True	200	2147	23	51	All Levels	1.5 hours	2017-01-18T20:58:58Z	Business Finance	429400	20
1	1113822	Complete GST Course & Certification - Grow You...	https://www.udemy.com/goods-and-services-tax/	True	75	2792	923	274	All Levels	39 hours	2017-03-09T16:34:20Z	Business Finance	209400	20
2	1006314	Financial Modeling for Business Analysts and C...	https://www.udemy.com/financial-modeling-for-b...	True	45	2174	74	51	Intermediate Level	2.5 hours	2016-12-19T19:26:30Z	Business Finance	97830	20
3	1210588	Beginner to Pro - Financial Analysis in Excel ...	https://www.udemy.com/complete-excel-finance-c...	True	95	2451	11	36	All Levels	3 hours	2017-05-30T20:07:24Z	Business Finance	232845	20
4	1011058	How To Maximize Your Profits Trading Options	https://www.udemy.com/how-to-maximize-your-pro...	True	200	1276	45	26	Intermediate Level	2 hours	2016-12-13T14:57:18Z	Business Finance	255200	20

🔹 Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3683 entries, 0 to 3682

01_EDA.ipynb

File Edit View Insert Runtime Tools Help

mands + Code + Text Run all Reconnect T4

Share

```
 13  published_date      3683 non-null   object
 14  published_time      3682 non-null   object
 15  year                3683 non-null   int64
 16  month               3683 non-null   int64
 17  day                 3683 non-null   int64
dtypes: bool(1), int64(9), object(8)
memory usage: 492.9+ KB
```

◆ Dataset Description:

	course_id	course_title	url	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject	profit	pul
count	3.683000e+03	3683	3683	3683	3683.000000	3683.000000	3683.000000	3683.000000	3683	3683	3683	3683	3.683000e+03	
unique	NaN	3668	3677	2	NaN	NaN	NaN	NaN	5	110	3677	4	NaN	
top	NaN	Creating an animated greeting card via Google ...	https://www.udemy.com/cfa-level-2-quantitative...	True	NaN	NaN	NaN	NaN	All Levels	1 hour	2017-07-02T14:29:35Z	Web Development	NaN	
freq	NaN	3	2	3373	NaN	NaN	NaN	NaN	1932	607	2	1200	NaN	
mean	6.764546e+05	NaN	NaN	NaN	65.992398	3193.371165	156.448004	40.062178	NaN	NaN	NaN	NaN	2.402885e+05	
std	3.437217e+05	NaN	NaN	NaN	60.985586	9498.231406	935.078241	50.366788	NaN	NaN	NaN	NaN	1.000760e+06	
min	8.324000e+03	NaN	NaN	NaN	0.000000	0.000000	0.000000	0.000000	NaN	NaN	NaN	NaN	0.000000e+00	
25%	4.077270e+05	NaN	NaN	NaN	20.000000	110.000000	4.000000	15.000000	NaN	NaN	NaN	NaN	1.567500e+03	
50%	6.882440e+05	NaN	NaN	NaN	45.000000	911.000000	18.000000	25.000000	NaN	NaN	NaN	NaN	2.305000e+04	
75%	9.617290e+05	NaN	NaN	NaN	95.000000	2537.500000	67.000000	45.000000	NaN	NaN	NaN	NaN	1.182600e+05	
max	1.282064e+06	NaN	NaN	NaN	200.000000	268923.000000	27445.000000	779.000000	NaN	NaN	NaN	NaN	2.431680e+07	

How can I install Python libraries? Load data from Google Drive Show an

What can I help you build?

◆ Missing Values in each column:

```
course_id            0
```

02_Content_Based.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all

[3]

```python
# Step 3: Combine selected text features into one column
df['combined_features'] = df['title'] + " " + df['subject'] + " " + df['level']

# Step 4: Preview the results
print("✅ Combined features created successfully!")
print(df[['title', 'subject', 'level', 'combined_features']].head())
```

```
✅ Combined features created successfully!
                                               title           subject  \
0                 ultimate investment banking course  business finance
1  complete gst course certification grow your ca...  business finance
2  financial modeling for business analysts consu...  business finance
3        beginner to pro financial analysis in excel  business finance
4       how to maximize your profits trading options  business finance

                level                                  combined_features
0          All Levels  ultimate investment banking course business fi...
1          All Levels  complete gst course certification grow your ca...
2  Intermediate Level  financial modeling for business analysts consu...
3          All Levels  beginner to pro financial analysis in excel bu...
4  Intermediate Level  how to maximize your profits trading options b...
```

[4]

02_Content_Based.ipynb

File Edit View Insert Runtime Tools Help

Commands | + Code + Text | ▷ Run all

[8]
```
        print(f"{i}. {course}")

else:
    print(f"❌ '{user_input}' not found in dataset. Try another title.")
```

```
Enter a course title: business banking

✅ Recommended Courses similar to 'business banking':

1. the complete investment banking course
2. ultimate investment banking course
3. accounting finance banking a comprehensive study
4.
5.
```

[9]
```
# 🔹 Step 1: Create a sample user profile (simulated)
user_profile = {
    'preferred_subjects': ['business finance', 'web development'],
    'preferred_levels': ['All Levels', 'Beginner Level'],
    'price_range': (0, 100)  # user prefers free or cheap courses
}

print("✅ Sample user profile created!")
```

```
✅ Sample user profile created!
```

[10]
```
# 🔹 Step 2: Filter courses according to the user profile
filtered_courses = df[
```

02_Content_Based.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all

[10]

```
print(f"✅ Found {len(filtered_courses)} matching courses for user profile.")
filtered_courses[['title', 'subject', 'level', 'price']].head(10)
```

✅ Found 1584 matching courses for user profile.

	title	subject	level	price
1	complete gst course certification grow your ca...	business finance	All Levels	75
3	beginner to pro financial analysis in excel	business finance	All Levels	95
6	investing trading for beginners mastering pric...	business finance	Beginner Level	65
7	trading stock chart patterns for immediate exp...	business finance	All Levels	95
12	financial management risk return for securities	business finance	All Levels	30
16	basic technical analysis learn the structure o...	business finance	Beginner Level	20
18	deadly mistakes of investing that will slash y...	business finance	All Levels	50
19	financial statements made easy	business finance	Beginner Level	95
22	create a business from home trading stocks tod...	business finance	All Levels	75
23	introduction to accounting mastering financial...	business finance	Beginner Level	50

```python
# Step 4: Recommend top courses from a cluster
def recommend_from_cluster(cluster_id, top_n=5):
    """
    Recommend top N courses from a given cluster based on popularity.
    """
    cluster_courses = df[df['cluster'] == cluster_id]
    top_courses = cluster_courses.sort_values(
        by=['num_subscribers', 'num_reviews'],
        ascending=False
    ).head(top_n)
    return top_courses[['title', 'subject', 'level', 'price']]

# Example test for cluster 2
print("🎯 Top recommendations for users in Cluster 2:\n")
print(recommend_from_cluster(2))
```

```
🎯 Top recommendations for users in Cluster 2:

                                                title           subject  \
494   bitcoin how i learned to stop worrying love cr...  business finance
105             stock market investing for beginners  business finance
1259       logo designing for your business in an hour    graphic design
1371     learn to design a letterhead a beginners course    graphic design
1413           graphic design an overview of the field    graphic design

               level  price
494       All Levels      0
105   Beginner Level      0
1259      All Levels     20
1371      All Levels      0
1413  Beginner Level      0
```

How can I install Python libraries? Load data from Google Drive Show an e

CO 04_Collaborative_NMF.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[11] 0s

```
    # Predict scores for unrated courses
    scores = []
    for course_id in unrated_courses:
        course_index = list(user_item_matrix.columns).index(course_id)
        distances, indices = knn_model.kneighbors(
            user_item_matrix.T.iloc[course_index, :].values.reshape(1, -1),
            n_neighbors=6
        )
        similarity = 1 - distances.flatten()[1:]  # skip self
        scores.append((course_id, np.mean(similarity)))  # average similarity score

    # Sort by similarity score
    recommended = sorted(scores, key=lambda x: x[1], reverse=True)[:n_recommendations]

    # Display top recommendations
    print(f"\n Recommended Courses for {user_id}:")
    for i, (course_id, score) in enumerate(recommended, 1):
        print(f"{i}. Course ID: {course_id} | Predicted Similarity: {score:.2f}")

# Step 2: Test recommendation for any user
recommend_courses("user_5")
```

```
 Recommended Courses for user_5:
1. Course ID: 140168 | Predicted Similarity: 0.93
2. Course ID: 709160 | Predicted Similarity: 0.93
3. Course ID: 792703 | Predicted Similarity: 0.93
4. Course ID: 1193536 | Predicted Similarity: 0.93
5. Course ID: 294292 | Predicted Similarity: 0.91
```

04_Collaborative_NMF.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all

[12]

```
    result = df[df['course_title'].str.contains(course_name, case=False, na=False)]
    return result[['course_id', 'course_title']]

# 🔍 Example test
get_course_id_by_name("Python", df)
```

	course_id	course_title
14	1196544	Python Algo Trading: Sentiment Trading with News
30	1170894	Python Algo Stock Trading: Automate Your Trading!
41	1035472	Python for Finance: Investment Fundamentals & ...
149	1070886	Python Algo Trading: FX Trading with Oanda
336	815482	Stock Technical Analysis with Python
538	529828	Python for Trading & Investing
764	1088656	Quantitative Trading Analysis with Python
866	902888	Investment Portfolio Analysis with Python
1686	546848	Learn to code in Python and learn Adobe Photos...
2502	16646	Web Programming with Python
2533	391546	Learn Python and Django: Payment Processing
2558	938560	The Complete Ethical Hacking Course 2.0: Pytho...
2575	47963	Coding for Entrepreneurs: Learn Python, Djan
2686	477702	Python for Beginners: Python Programming Langu...
2965	270808	Projects in Django and F
3138	574082	Web Scraping with Python, Ruby & import. io

How can I install Python libraries? Load data from Google Drive

What can I help you build?

Variables Terminal

04_Collaborative_NMF.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all

[17]

```python
        distances, indices = knn_model.kneighbors(
            user_item_matrix[selected_course_id].values.reshape(1, -1),
            n_neighbors=6
        )

        similar_course_ids = user_item_matrix.columns[indices.flatten()[1:]]
        similar_courses = df[df['course_id'].isin(similar_course_ids)][['course_id', 'course_title']]

        print("\n📘 Top Recommended Courses:\n")
        for i, row in enumerate(similar_courses.itertuples(), 1):
            print(f"{i}. {row.course_title} (Course ID: {row.course_id})")

        print("\n✅ Recommendation complete!\n")

# 🔹 Auto interactive part — user just types the course name
course_query = input("🔍 Enter Course Name: ")
get_recommendations(course_query)
```

```
🔍 Enter Course Name: python

🎯 Selected Course: Python Algo Trading: Sentiment Trading with News (ID: 1196544)

📘 Top Recommended Courses:

1. Option Trading for Rookies: Make & Manage Profitable Trades (Course ID: 941120)
2. Accounting for Depreciation (Collage Level) (Course ID: 258174)
3. 5 Exotic Guitar Scales and How to Use Them Effectively (Course ID: 830568)
4. Learn to play and improve 12 bar blues harmonica solos
5. Build CRUD Application - PHP & Mysql (Course ID: 120105

✅ Recommendation complete!
```

How can I install Python libraries? Load data from Google Drive Show an e

What can I help you build?

05_Collaborative_NN.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[9]

```python
# Select first 100 courses from dataset
courses_subset = df['course_id'].iloc[:num_courses]

# Random interactions (0 = not enrolled, 1 = enrolled)
user_item_matrix = pd.DataFrame(
    np.random.randint(0, 2, size=(num_users, num_courses)),
    columns=courses_subset
)

print("✅ Dummy user-item matrix created")
user_item_matrix.head()
```

✅ Dummy user-item matrix created

course_id	1070968	1113822	1006314	1210588	1011058	192870	739964	403100	476268	1167710	...	891484	1217064	382204	1259560	308696	1270254	474928	1148774	959144	1233350
0	1	0	0	1	1	0	0	0	1	1	...	1	0	0	1	0	1	0	1	0	0
1	0	1	1	1	0	1	0	0	1	0	...	0	1	1	0	1	1	0	0	1	1
2	1	1	1	0	1	1	0	1	1	1	...	0	0	0	1	0	1	0	1	0	0
3	1	1	0	1	0	1	0	1	1	1	...	0	0	1	0	0	1	1	1	1	0
4	1	1	0	1	1	0	1	1	1	0	...	0	0	0	1	0	1	1	0	0	0

5 rows × 100 columns

[14]

```python
# ========================
# 10. Collaborative Filtering (Neural Network Embedding)
# ========================
```

How can I install Python libraries? Load data from Google Drive Show an

05_Collaborative_NN.ipynb ☆ Saving...

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all

[15] 4s

```python
    predictions = model.predict([np.full_like(course_indices, user_idx), course_indices], verbose=0).flatten(

    top_indices = predictions.argsort()[::-1][:top_n]
    top_courses = user_item_matrix.columns[top_indices]

    print(f"\n Top {top_n} recommended courses for User {user_id}:")
    for i, course_id in enumerate(top_courses):
        print(f"{i+1}. {course_id}")

# 🔹 Step 6: Example: Recommend for user_id = 0
recommend_courses_nn(user_id=0, user_item_matrix=user_item_matrix, model=nn_model_trained)
```

```
Epoch 1/5
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.5015 - loss: 0.4982
Epoch 2/5
157/157 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4995 - loss: 0.4524
Epoch 3/5
157/157 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.5130 - loss: 0.2690
Epoch 4/5
157/157 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.6050 - loss: 0.2358
Epoch 5/5
157/157 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.6312 - loss: 0.2297
✅ Neural Network Embedding model training done!

 Top 5 recommended courses for User 0:
1. 1210588
2. 285638
3. 43319
4. 606928
5. 302562
```

How can I install Python libraries? Load data from Goog

[ ] Start coding or generate with AI.

What can I help you build?

Thank You

Regards: Muhammad Munawar Shahzad