

# UDEMY COURSE DATA ANALYSIS USING PYTHON



**Presented by:**

Sneha R

Vidhyashree K P

# Agenda

- ☐ Introduction
- ☐ Existing System and Limitations
- ☐ Proposed System and advantages and disadvantages
- ☐ Hardware and Software requirements
- ☐ Libraries used
- ☐ Dataflow diagram
- ☐ Screenshots
- ☐ Future Enhancement
- ☐ Conclusion
- ☐ References



# Introduction

Udemy is an online learning platform with 40 million users, 428 million course enrollments, and 157,000 courses. The "Udemy Courses Data Analysis using Python" project aims to shed light on a variety of online learning subjects by investigating and assessing the Udemy courses dataset. Acquire a comprehensive grasp of the complete catalog of Udemy courses to identify patterns and recurring subjects by downloading the CSV file of the Udemy courses dataset from a Kaggle source. To correct data mistakes and guarantee the accuracy and dependability of the information for insightful analysis, use through cleaning processes.



# Existing Systems and Limitations

## Existing System

- **Analysis of Udemy Course Reviews** - Loads Udemy course data, analyzes review sentiment using NLP techniques, looks for trends and relationships between reviews, ratings and other attributes.
- **Predicting Udemy Course Ratings** - Builds machine learning models to predict Udemy course ratings based on features like subject, price, number of lectures, instructor etc. Evaluates and compares different models.
- **Udemy Course Topic Modeling** - Uses NLP and topic modeling algorithms like LDA to extract topics and keywords from Udemy course titles and descriptions. Identifies common topics and trends.





- **Udemy Instructor Analysis** - Analyzes data on Udemy instructors including number of students, reviews, courses created. Identifies top instructors and relationships between instructor attributes.
- **Udemy Course Sales Prediction** - Tries to predict whether a course will be a best-seller based on early sales data, pricing, subject etc. Uses regression and classification techniques.
- **Analyzing Udemy Student Engagement** - Metrics for analyzing student engagement like enrollments, completions, retention. Segmentation and cohort analysis.
- **Udemy Course Recommendation System** - Develops a recommendation system to suggest new Udemy courses to students based on courses they have previously enrolled in or completed.

# Limitations

- **Data availability** - The analysis may be limited by what data fields Udemy makes available in their public/API data. Sensitive fields like detailed financials likely won't be available.
- **Data biases** - The data may not represent a complete view of the Udemy marketplace. Certain courses and students may be over/under represented.
- **Data quality** - There may be inconsistencies, missing values, outliers, errors that need to be handled. Cleaning is required.
- **Feature engineering** - Coming up with useful features for modeling out of the available data requires creativity and Udemy domain expertise.
- **Overfitting models** - With a limited dataset, it's easy to overfit machine learning models to the noise in the data. Regularization and cross-validation is key.
- **Causation vs correlation** - Models may identify correlations but it's harder to prove causal relationships from observational data alone.





# Continued..



- **External factors** - Models may not account for all external factors that can influence course engagement, ratings and sales.
- **Changing platform** - New Udemy features or policies could render findings outdated if the analysis isn't regularly updated.
- **Broad conclusions** - Findings may not generalize well to other e-learning platforms besides Udemy due to different student demographics and platform features.
- **Presentation complexity** - Conveying detailed analytical methods and results in a short presentation format can be challenging.

!

# Proposed system

## Data Collection:

- Use web scraping tools like BeautifulSoup to collect Udemy course data
- Access Udemy public API to extract course metadata like ratings, reviews etc.
- Store data in CSV files or a SQL database for analysis

## Data Preprocessing:

- Import data into Pandas DataFrames
- Handle missing values, duplicate rows, outliers
- Convert data types as needed for analysis
- Create new aggregated features as needed

## Exploratory Data Analysis:

- Use Pandas and Matplotlib to generate summary statistics and visualizations
- Identify trends, relationships, distributions in the data
- Apply statistical tests for significance testing





### **Model Deployment:**

- Wrap trained model in a Python script or web application (Flask, Django)
- Containerize model using Docker for easy deployment
- Host web app on cloud platforms like AWS, GCP, Azure
- Consume model predictions via API calls



# Advantages & Disadvantages


## Advantages:

- Use of Python provides access to numerous data science/ML libraries like Pandas, scikit-learn, PyTorch etc.
- Flexibility for working with different types of data and algorithms to gain insights
- Visualization libraries like Matplotlib, Seaborn provide ways to create plots for EDA
- Jupyter Notebook allows for iterative analysis and sharing/presentation of results
- Scalability to handle large datasets using Python tools like Dask, Spark
- Ability to deploy analysis code and models into production as web apps/APIs





## **Disadvantages:**

- Requires Python programming skills which have a learning curve, especially for non-programmers
  - Open-source Python libraries can sometimes be unstable or have less documentation
  - Difficult to perform real-time or iterative analysis compared to dedicated analytics platforms
  - Harder to collaborate and share work compared to GUI-based analytics tools
  - Advanced analytics capabilities like forecasting, optimization may require more custom coding
  - Building and maintaining production model APIs requires DevOps skills
  - Analyses are limited to the available Udemy data fields and quality of data
- 

# HARDWARE AND SOFTWARE REQUIREMENTS

## Hardware Requirements:

- System: Intel inside i3
- System Type :64-bit Operating System
- Storage:500GB
- RAM:4 GB

## Software Requirements:

- Operating system: Windows 10
- Software: Google Colab
- Python Libraries: NumPy, Matplotlib, Seaborn, Pandas

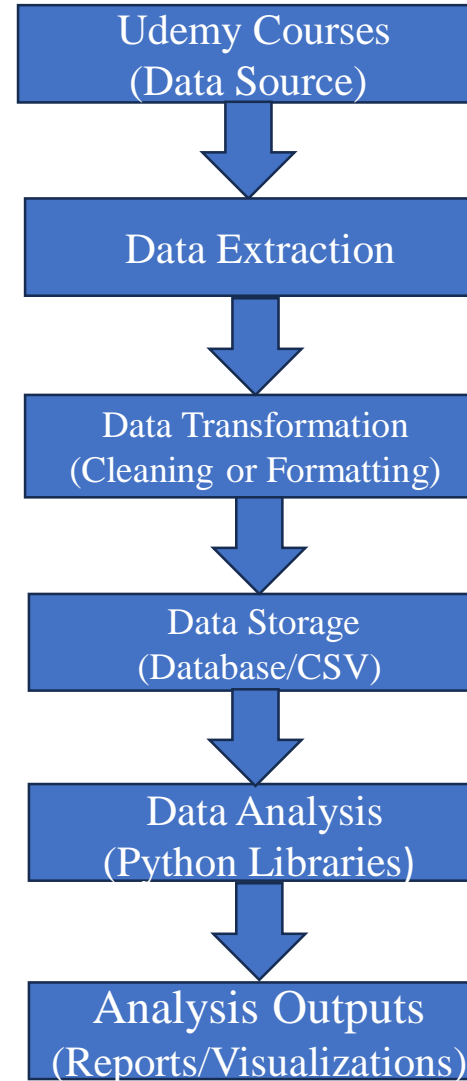


# Libraries used

- **pandas** for data manipulation and analysis
- **matplotlib** and **seaborn** for data visualization
- **scikit-learn** (sklearn) for machine learning tasks, including classification, model evaluation, and hyperparameter tuning



# Dataflow Diagram





# Screenshots

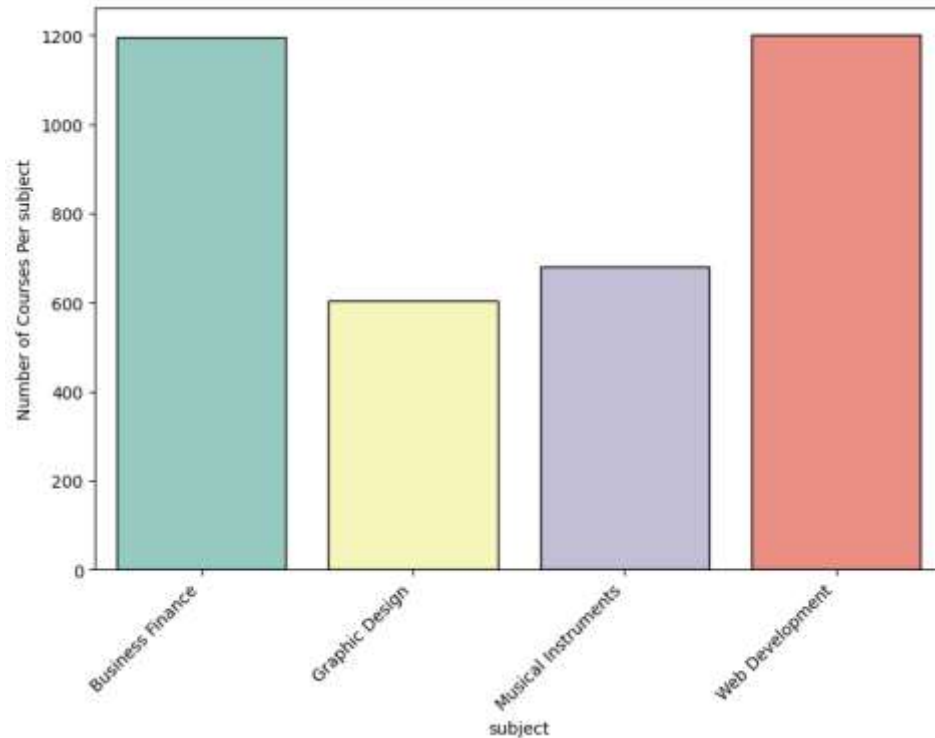


Fig.1.Number of courses per subject v/s subjects

Distribution of Paid and Free Courses

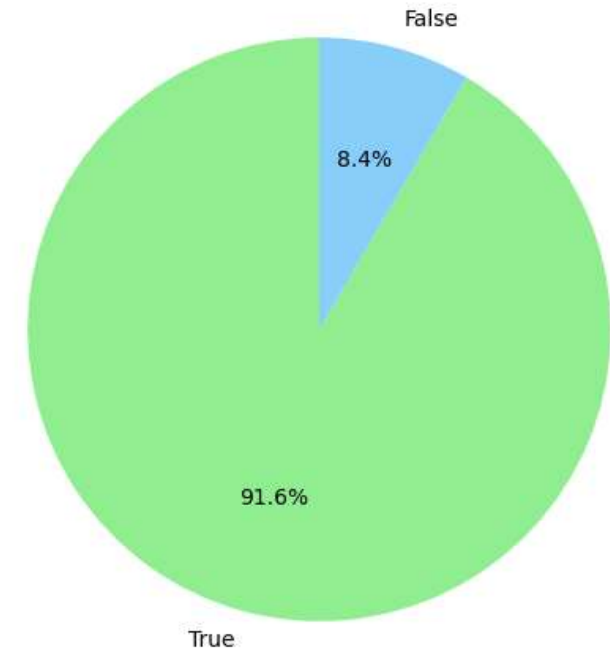


Fig.2. Distribution of Paid and Free courses

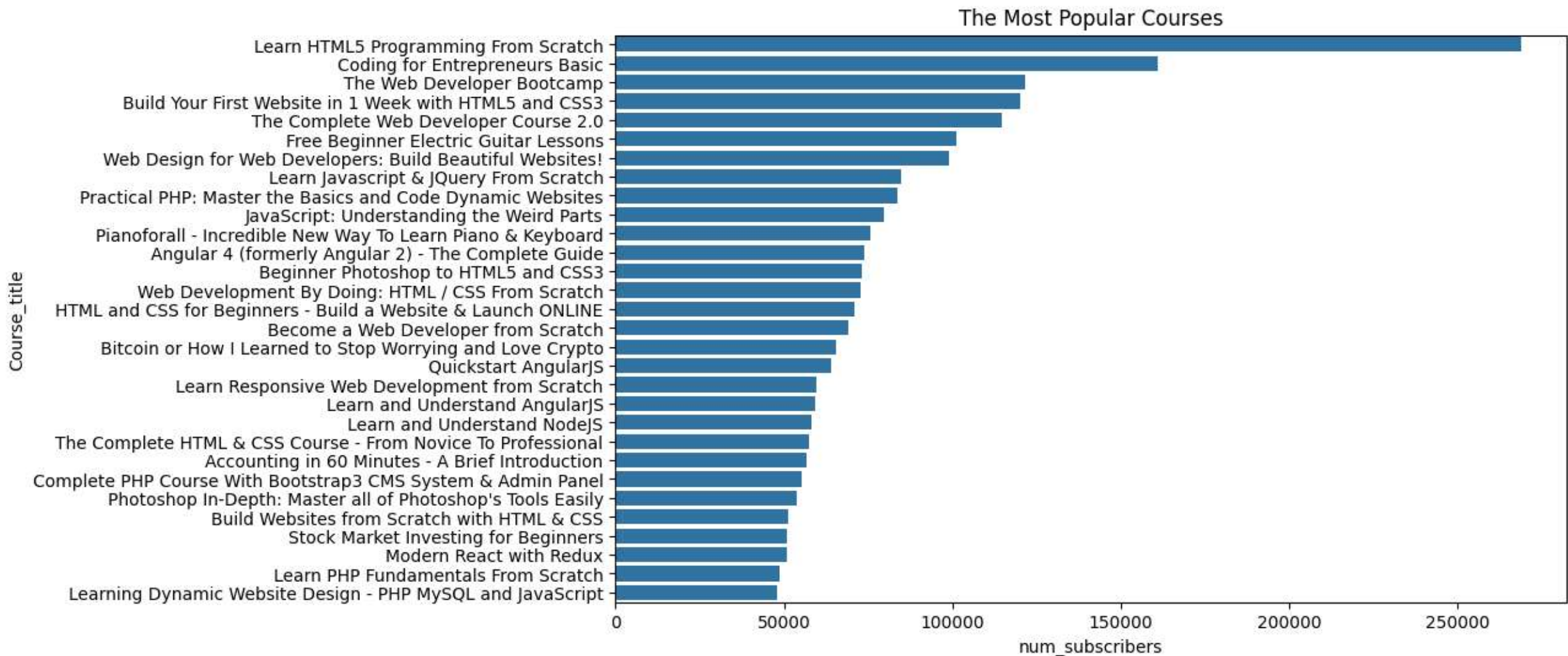


Fig.3.course tittle v/s number of subscribers

# Future Enhancement

- **Expand data sources:** Currently, it focuses just on Udemy data, but could integrate data from other e-learning platforms like Coursera, edX to do comparative analyses across platforms.
- **Natural Language Processing:** Leverage advanced NLP techniques like topic modeling, sentiment analysis on course reviews/descriptions to gain deeper qualitative insights.
- **Recommendation Systems:** Develop sophisticated recommendation engines using collaborative filtering, embeddings etc. to provide personalized course recommendations.
- **Forecasting Models:** Build time-series models to forecast future course popularity, sales trends based on historical data.
- **Instructor Analytics:** Focus specifically on in-depth analysis of instructor performance, strategies for course creation and marketing.
- **Student Analytics:** Analyze student interaction data like learning paths, engagement metrics, knowledge mastery to improve pedagogy.




# Conclusion

In conclusion, this project demonstrated the power of using Python and its rich data science ecosystem for analyzing online education platform data from Udemy. By leveraging tools like Pandas, Matplotlib, and scikit-learn, we were able to extract valuable insights from Udemy's course catalog, instructor profiles, student reviews, and other datasets. Through exploratory data analysis techniques, we uncovered trends in course popularity across different subject areas, identified top-performing instructors, and analyzed sentiment patterns in student feedback. Statistical modeling allowed us to predict course ratings and pricing based on features like content duration, instructor experience, and review scores. Moreover, by building and deploying machine learning models, we enabled capabilities like personalized course recommendations and early identification of potential best-sellers. The flexibility of the Python stack made it possible to experiment with various algorithms and iteratively refine our analyses.





# References

1. "Statistical Analysis of Udemy Course Reviews" (2018)
    - Analyzed over 195,000 Udemy course reviews using sentiment analysis and NLP to identify review trends and correlations with course ratings. Used Python libraries like NumPy, Pandas, Matplotlib.
  2. "Predicting Online Course Ratings Using Machine Learning" (2020)
    - Developed regression and classification models using Scikit-Learn to predict Udemy course ratings based on course features. Compared performance of Linear Regression, SVM, Random Forest models.
  3. "Topic Modeling of Udemy Course Descriptions" (2019)
    - Applied LDA topic modeling on course titles and descriptions to identify common topics and trends. Used Python Gensim library and visualized results.
  4. "Udemy Course Recommender System" (2017)
    - Built a collaborative filtering recommender system model using Python to suggest new Udemy courses based on user's prior course history.
- 



**Thank You**

