Hello, and welcome! In this video, we’ll be going through a quick introduction to recommendation systems. So, let’s get started. Even though peoples’ tastes may vary, they generally follow patterns. By that, I mean that there are similarities in the things that people tend to like … or another way to look at it, is that people tend to like things in the same category or things that share the same characteristics. For example, if you’ve recently purchased a book on Machine Learning in Python and you’ve enjoyed reading it, it’s very likely that you’ll also enjoy reading a book on Data Visualization. People also tend to have similar tastes to those of the people they’re close to in their lives. Recommender systems try to capture these patterns and similar behaviors, to help predict what else you might like. Recommender systems have many applications that I’m sure you’re already familiar with. Indeed, Recommender systems are usually at play on many websites. For example, suggesting books on Amazon and movies on Netflix. In fact, everything on Netflix’s website is driven by customer selection. If a certain movie gets viewed frequently enough, Netflix’s recommender system ensures that that movie gets an increasing number of recommendations. Another example can be found in a daily-use mobile app, where a recommender engine is used to recommend anything from where to eat or what job to apply to. On social media, sites like Facebook or LinkedIn, regularly recommend friendships. Recommender systems are even used to personalize your experience on the web. For example, when you go to a news platform website, a recommender system will make note of the types of stories that you clicked on and make recommendations on which types of stories you might be interested in reading in future. There are many of these types of examples and they are growing in number every day. So, let’s take a closer look at the main benefits of using a recommendation system. One of the main advantages of using recommendation systems is that users get a broader exposure to many different products they might be interested in. This exposure encourages users towards continual usage or purchase of their product. Not only does this provide a better experience for the user but it benefits the service provider, as well, with increased potential revenue and better security for its customers. There are generally 2 main types of recommendation systems: Content-based and collaborative filtering. The main difference between each, can be summed up by the type of statement that a consumer might make. For instance, the main paradigm of a Content-based recommendation system is driven by the statement: “Show me more of the same of what I've liked before." Content-based systems try to figure out what a user's favorite aspects of an item are, and then make recommendations on items that share those aspects. Collaborative filtering is based on a user saying, “Tell me what's popular among my neighbors because I might like it too.” Collaborative filtering techniques find similar groups of users, and provide recommendations based on similar tastes within that group. In short, it assumes that a user might be interested in what similar users are interested in. Also, there are Hybrid recommender systems, which combine various mechanisms. In terms of implementing recommender systems, there are 2 types: Memory-based and Model-based. In memory-based approaches, we use the entire user-item dataset to generate a recommendation system. It uses statistical techniques to approximate users or items. Examples of these techniques include: Pearson Correlation, Cosine Similarity and Euclidean Distance, among others. In model-based approaches, a model of users is developed in an attempt to learn their preferences. Models can be created using Machine Learning techniques like regression, clustering, classification, and so on. This is the end of our video. Thanks for watching! (Music)

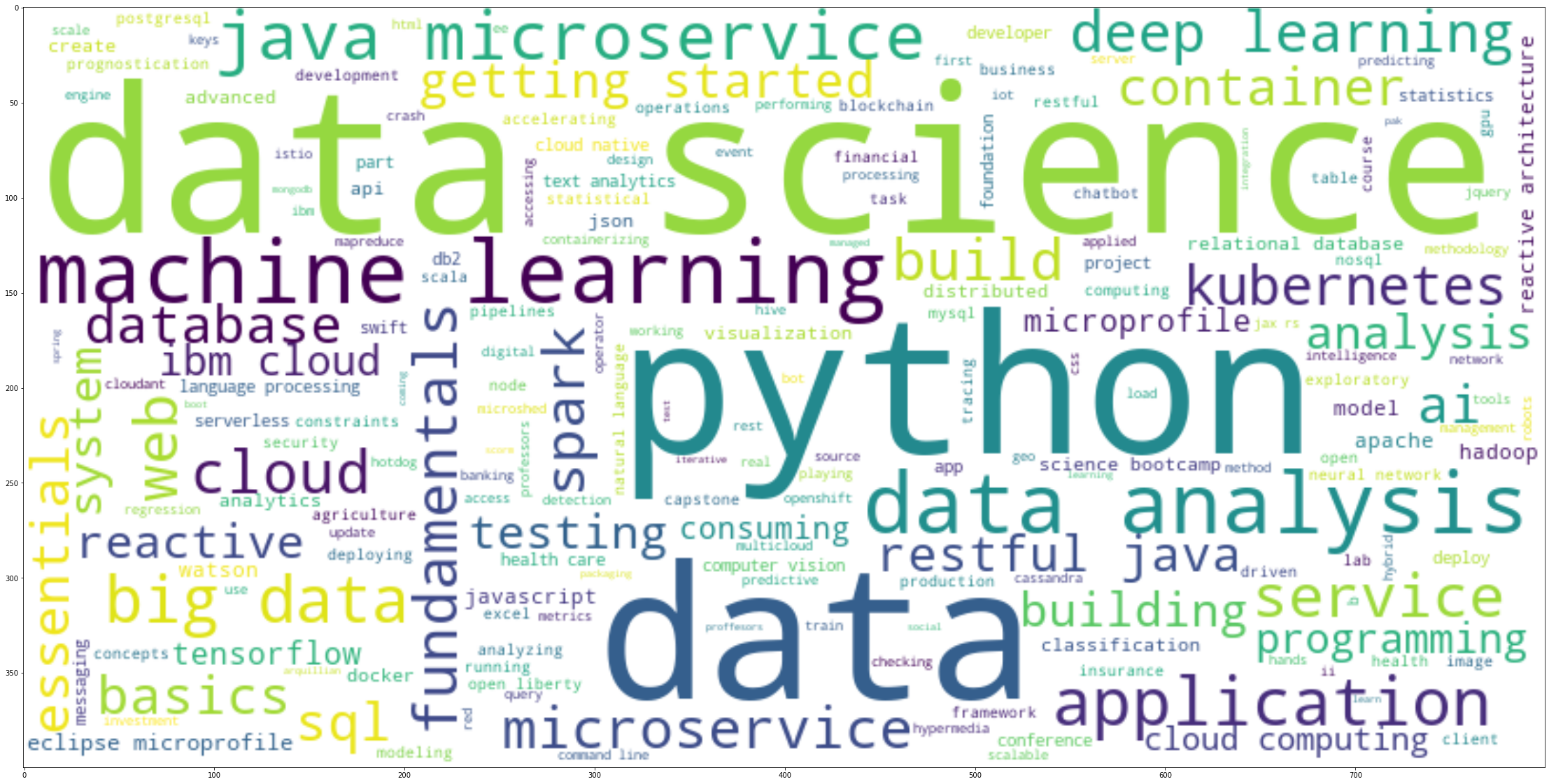
# Capstone Overview

In this capstone course, you will apply various machine learning knowledge and skills that you have learned as part of the previous courses to solve some real-world industrial challenges.

# Project Scenario

Assume you are a new machine learning engineer in a Massive Open Online Courses (MOOCs) startup called AI Training Room. In AI Training Room, learners across the world can learn leading technologies such as Machine Learning, AI, Data Science, Cloud, App development, etc. Your company grows rapidly and reaches millions of learners in a very short period.

The learning topics of AI Training Room can be summarized in the following word cloud:

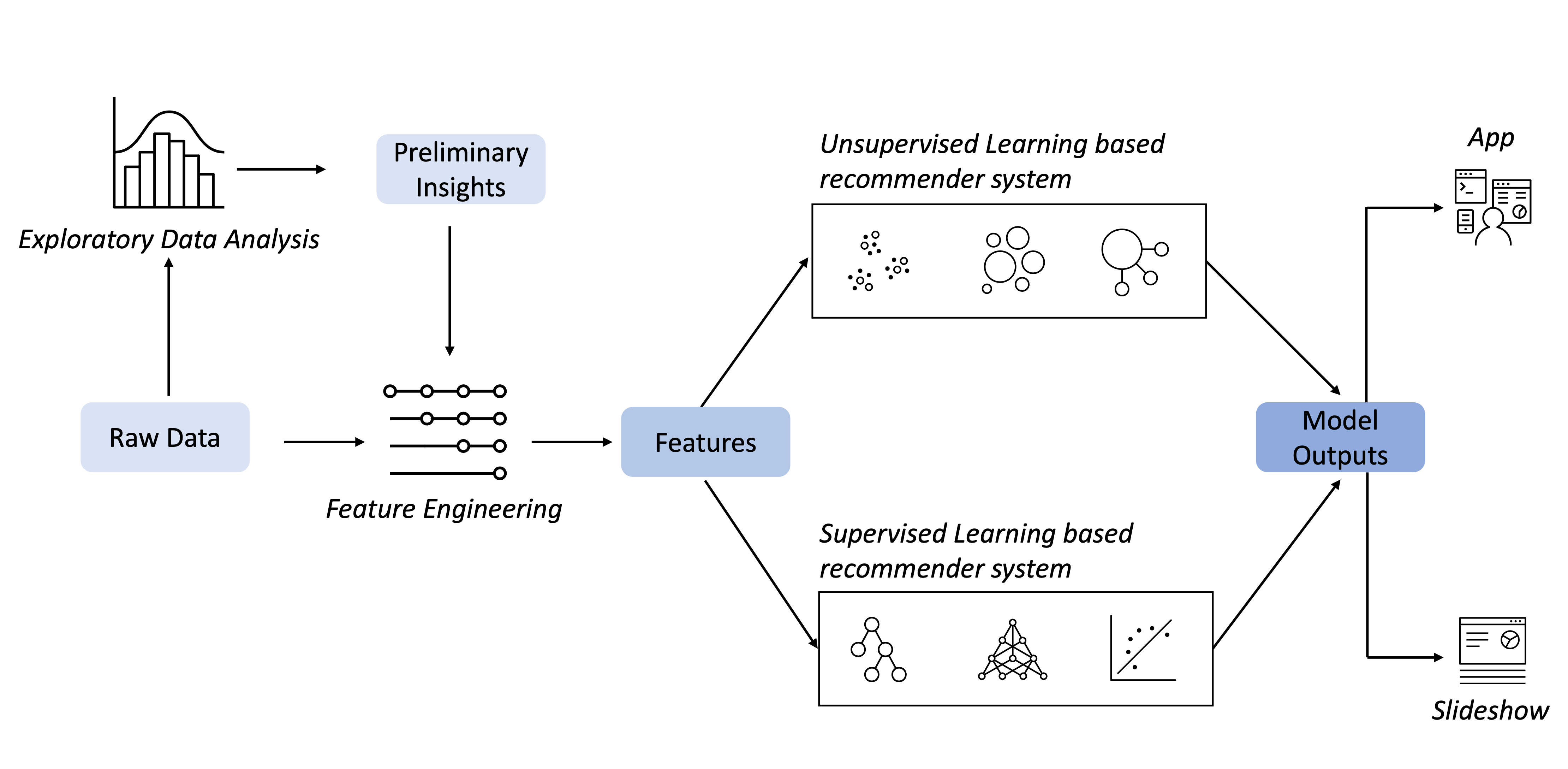


Starting this year, your machine learning engineer team is working very hard on a recommender system project. The main goal of this project is to improve learners' learning experience via helping them quickly find new interested courses and better paving their learning paths. Meanwhile, with more learners interacting with more courses via your recommender systems, your company's revenue may also be increased.

This project is currently at the Proof of Concept (PoC) phase so your main focus at this moment is to explore and compare various machine learning models and find one with the best performance in off-line evaluations.

## Your Tasks

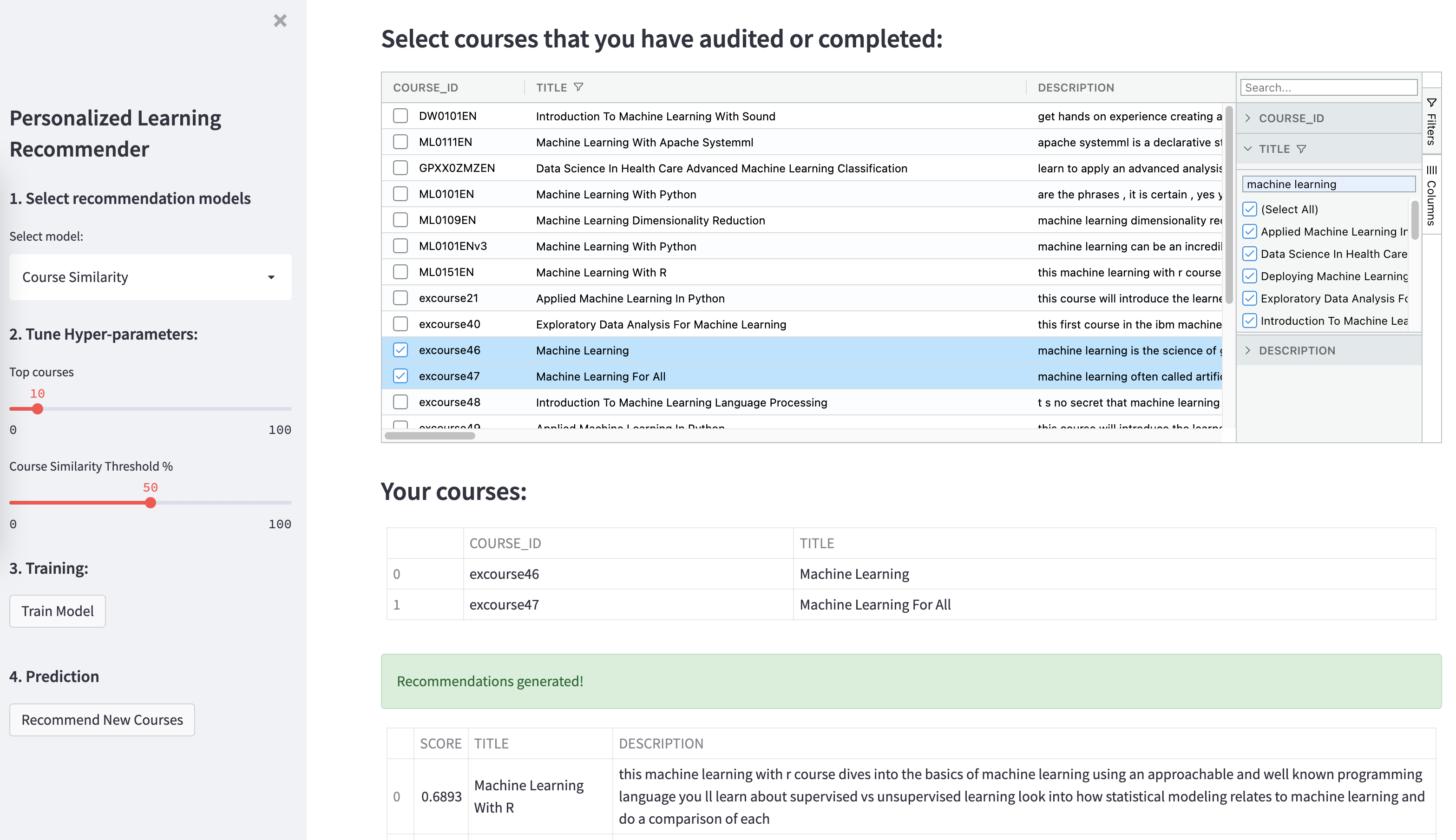
Your tasks in this project are summarized in the following workflow, and you will be guided through them in hands-on labs.



More specifically, you will undertake the tasks of:

* Collecting and understanding data
* Performing exploratory data analysis on online course enrollments datasets
* Extracting Bag of Words (BoW) features from course textual content
* Calculating course similarity using BoW features
* Building content-based recommender systems using various unsupervised learning algorithms, such as:
  + Distance/Similarity measurements, K-means, Principal Component Analysis (PCA), etc.
* Building collaborative-filtering recommender systems using various supervised learning algorithms
  + K Nearest Neighbors, Non-negative Matrix Factorization (NMF), Neural Networks, Linear Regression, Logistic Regression, RandomForest, etc.
* Creating an insightful and informative slideshow and presenting it to your peers

If you have extra bandwidth, you can also deploy and demonstrate your models via a web app built with streamlit. Streamlit is an open-source app framework for Machine Learning and Data Science to quickly demonstrate their works.

Your course recommender app where you select different recommendation models and generate recommendations, may look like the following screenshot: 

This project is a great opportunity to showcase your machine learning skills, and demonstrate your proficiency to potential employers.

## Grading Schema

* Graded Quizzes: **30 pts**
* Final presentation, peer-review: **70 pts**

# Development Environments

In this project, you have at least three development environments you may choose from:

## Skills Network Labs

Skills Network Labs is a virtual lab environment reserved for the exclusive use by the learners on IBM Developer Skills Network portals and its partners.

## Use your local Python, Jupyter Notebook, and IDE

If you experience any issues with the above two cloud environments, you may install Python and JupyterNotebook / JupyterLab on your own environments like a desktop or laptop computer. All the notebooks and data used in the capstone can be downloaded and executed locally.

## Watson Studio

For this project, you will use Watson Studio as your main development environment. Watson Studio is a component of IBM Cloud Pak for Data, is a suite of tools and a collaborative environment for data scientists, data analysts, AI and machine learning engineers, and domain experts to develop and deploy your projects.

# Next Steps

Now you should have a basic understanding of this capstone project.

In the next step of your project, you will start with collecting and exploring the datasets.