Problem:
The global e-learning market was worth USD 215 billion in 2021. It is estimated to reach an expected value of USD 645 billion by 2030 at a CAGR of 13% during the forecast period (2022–2030). With this demand forecast, we can expect a huge amount of discussion,in the form of question-answering. There are even sites like Quora, Stack Overflow and Reddit dedicated just to forum discussions. One of the main problems these sites face is dealing with duplicate questions, for example, in our Big-Data Coursys forum one student might have asked a question and it has been answered, but at a later point in time, another student might ask a similar question, and if there are many such duplicate questions, it maybe a waste of resources to answer the same questions again.

So, we chose to solve this problem by building a text similarity system that can identify if two questions are similar.

Solution:

- Data extraction is a key part of our project, as our approach will include extracting the required dataset from a large dump of raw StackOverflow data using Big Data tools.
- Once the data is extracted, we will do some feature engineering on text to extract useful features from raw text.
- Further, we aim to solve this problem by training a few ml and NLP models that can identify semantic similarities between pairs of questions.
- Visualization is one of the main parts of the project which helps us to understand the dataset better.
- Subsequently, we will use the transformed data to train ML/NLP models which can recognize identical questions.
- In addition to that, we train an NLP model which can convert a piece of text into powerful embedding. We can store these embeddings for future use.

Approach:
We have divided our approach into the following categories:

Data extraction:
We intend to use archived StackOverflow data (can be found here). Following are the main tables that we would use in solving the problem.-

1. Badges
2. Comments
3. PostHistory
4. Postlinks
5. Posts
6. Tags
7. Users
8. Votes

Duplicate questions in Stack Overflow are marked with a special marker. To collect these questions we need to parse the posts database file. We will extract those questions that were closed as duplicate and appended ‘[Duplicate]’ to their titles. We shall parse the postlinks and posts database files in order to collect the master questions of the duplicate questions that were identified earlier. We plan to perform the above two steps using Spark. Once we collect the pairs, consisting of master question and duplicate question we shall store this in the S3 bucket that shall be fed into the ML model for training.

Feature engineering:
Creating features like word count in a question, common words between question pairs, etc.

Building ML model:
We aim to Train two ML/NLP models:

1. First model is used to identify similarities between the input questions based on the features extracted from the text.
2. The second model is for semantic search. This model can find similar questions from a set of existing questions based on embeddings generated from the model. We will store generated embedding in a FAISS-like database for a fast query operation.

Interactive UI and data visualization:
We plan to get some insights into the data by performing Exploratory data analysis and answering some questions like:

1. What percent of questions were marked duplicate in a category?
2. What is the monthly average of duplicate questions generated?
3. With help of statistical plots, we can understand the usefulness of extracted features(how well a feature can differentiate duplicate questions from different questions).

For inference, we will also integrate our trained models into UI.

Technologies that we may use:

1. Amazon EMR(spark), S3, and Databricks - to store and extract the required dataset
2. Tabulea - to understand and visualize the custom-generated features
3. Text vectorizers like word2vec, sentence BERT
4. Amazon Sagemaker - to train model
5. ML models, NLP models
6. Faiss search - to store text embedding
7. Docker and Amazon EC2 to deploy a trained model
8. Streamlit - To build a UI