Project Report

Major Project-

Quora Question Pair Similarity

Submitted by- **TEAM**

MEMBERS: **NIK AVENGERS**

- Himanshu Vaish
- Promise C. Uzoagulu
- Alonge Daniel
- Sachin Alfred
- Peris Wangari

Submitted to-

Mr. Kanav Bansal sir

Table Of Content

Introduction:

Quora is a question-and-answer website where people go to find information. Every piece of content on the site is generated by users, meaning it is created, edited, and organized by the same people that use the website.

The **Quora dataset** is composed of pairs of questions, and the task is to determine if the two questions are duplicates of each other, that is, that they have the same meaning. The dataset contains **404k pairs of Quora questions.**

Project Objective:

- The objective of this project is to develop a machine learning model which will be capable of predicting whether a pair of questions on Quora are duplicates or not. By using an automated identification process, to enhance the overall user experience.
- This could be useful to instantly provide answers to questions that have already been answered.

Motivation:

- If any question is misclassified as not a duplicate, it can lead to several problems for both the consumer and the provider.
- For consumers, it becomes challenging to find the correct answer, resulting in increased struggles. Additionally, providers will experience higher computational costs due to duplicated content, leading to increased resource consumption.
- Moreover, as the number of duplicated questions increases, it also contributes to longer waiting times for clients to obtain the desired answer. This, in turn, negatively impacts customer retention.
- Customers do not want to spend excessive time searching for a single answer, and this may cause them to leave the page. Ultimately, such an outcome can have a detrimental effect on the business.

Technologies Used:

- **NLP**
  - Text preprocessing
  - Feature extraction
  - Vector conversion
- **ML**
  - Data preprocessing
  - Model building
  - Model evaluation
- **MLOPs**
  - Experiment tracking - Mlflow
  - Workflow Orchestration – Prefect

Implementation Steps:

- **Data collection** for model training and building.
- **Data preprocessing** using machine learning library like pandas, numpy and seaborn.
- **Text preprocessing** using NLP techniques like tokenization, stemming, lemmatization, encoding, vector conversion for valuable feature extraction from the raw data for model training and building.

- **Model building** for prediction and identification that a given question is duplicate or not. We have used/tried GradientBoost. XGBoost, RandomForest, Naïve Bayes and LGBM Classifier for this task. Among these the XGBoost has performed well and given the best log loss score along with better accuracy and it is not overfitting also.
- **Evaluation of model** was taken using classification reports like accuracy and F1 scores, and log loss is that we are looking after.
- **Experiments** was done and their **Tracking** was done using MLFlow UI. Experiment tracking like hyperparameter tunning and weights adjustment was done and tracked for various models at same time that has given better understanding of model and better control over solving the problem.
- **Productionized the model** that has best log loss score and, in our case, XGBoost was best among all the models.
- **Workflow Orchestration & Deployment** of model taken place using Prefect API that allows Server-side configuration remotely. It is scheduled for 1hr daily.

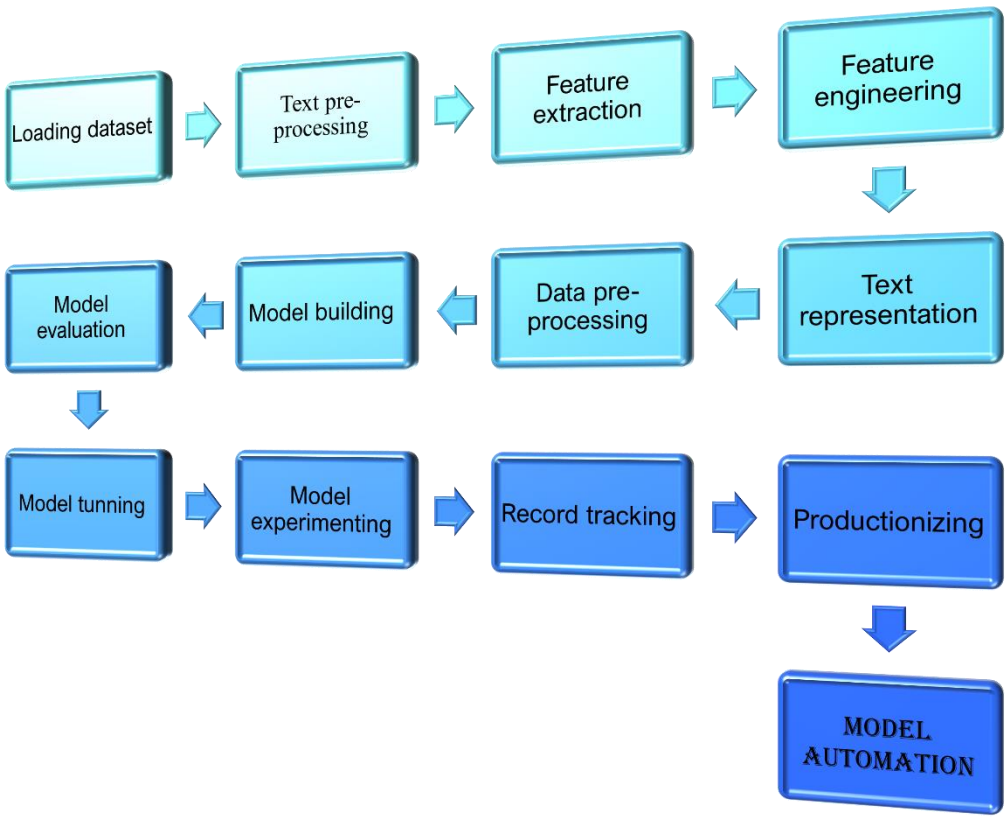
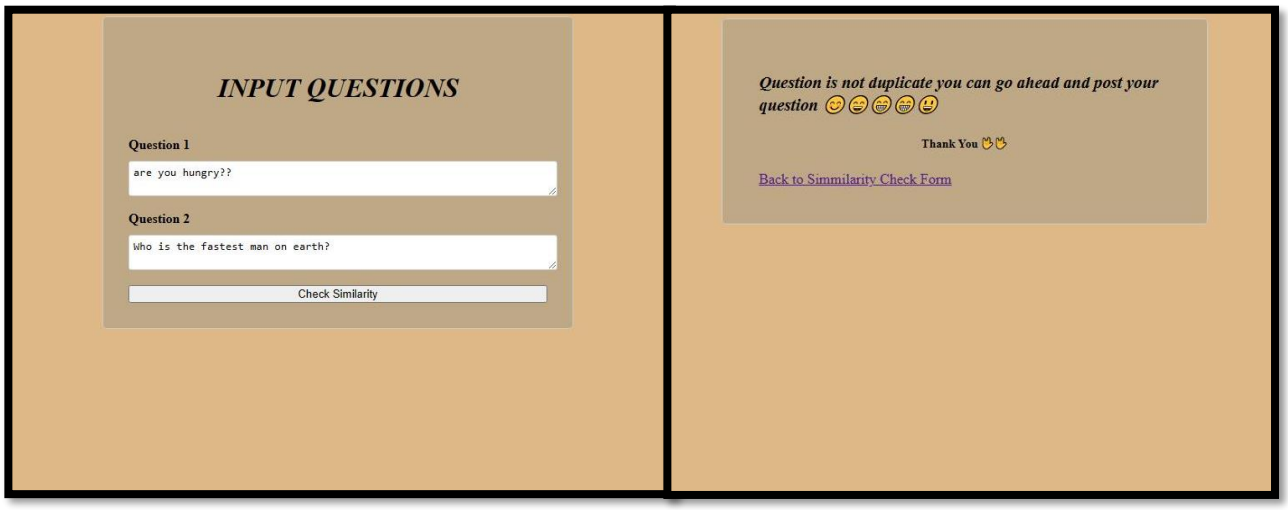


Fig: Detailed roadmap of implementation steps.

Prediction/Results:

We got the satisfactory results and our predictions are almost accurate. See below for the proofs from the deployment of our model.



Conclusion:

Duplication of questions in this large scale can lead to serious issues like higher computational costs and poor consumer satisfaction which will adversely affect the business and this should be treated carefully.

References:

You can find some piece of code below in the given repository for more code related references.

GitHub - GitHub repo link here