Azure Medical word embedding and search engine development Project Overview

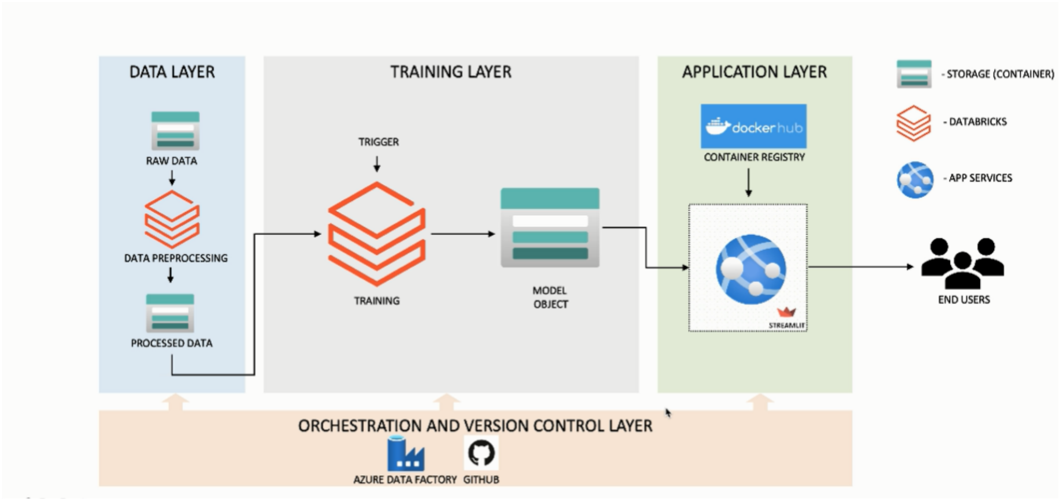
Business Context:
We all must have wondered that if we search for a particular word in google, it does not show just the results that contain the very same word but also shows results that are very closely related to it. For example, if we search for the term ‘medicine’ in google, you can see results that not just include the word ‘medicine’ but also terms such as ‘health,’ ‘pharmacy’, ‘WHO’ and so on. So, google somehow understands that these terms are closely related to each other. This is where word embeddings come into the picture. Word embeddings are nothing but numerical representations of words in a sentence depending on the context. In this project, we will be learning how word embeddings work and how can we build a smart search engine, particularly for medical science using word embeddings.

Aim :
To develop a machine learning application that can understand the relationship and pattern between various words used together in the field of medical science, create a smart search engine for records containing those terms, and finally build a machine learning pipeline in azure to deploy and scale the application.

Tech Stack:

- ➔ Language: Python
- ➔ Packages: NLTK, Scikit-Learn, Pandas, Numpy, Streamlit, etc.
- ➔ Cloud: Azure, Azure Data Factory, Azure Blob Storage, Azure Databricks
- ➔ Code Management: Git, Github, Docker, Dockerhub

Approach:
The flowchart below represents the overall approach that is used in the project in a superficial manner.



<u>Data Layer</u>

The data layer is the primary layer where we perform all the basic data preprocessing steps on the data that we obtain from the Azure Blob storage and again store the cleaned data to Azure Blob storage as a different file

<u>Training Layer</u>

The training layer is the intermediary layer which involves the crucial step of model building. The model is trained using Azure Databricks and the generated model object is stored back to Azure Blob Storage. A trigger has also been set to the model training process in order to train the model periodically with new data obtained.

<u>Application Layer</u>

This layer involves building a user interface using Streamlit to the model built so that the users can interact with our model object. The user application is then deployed to the cloud using Azure App Services

Prerequisites:

- Word2Vec and FastText Word Embedding with Gensim in Python

Note: Kindly Download the training pipeline for Demo Application from here.

Key Takeaways:

1. Introduction to NLP using Python
2. How to clean textual data using Python?
3. What are the steps involved in text preprocessing?
4. What are word embeddings?

5. Why are word embeddings important to build search engines?
6. How to build your own search engine?
7. How to develop a user interface for a model using python and streamlit?
8. What is MLOps?
9. What are the advantages of MLOps?
10. How to deploy a machine learning model into production using Azure?
11. What are the various services provided by Azure?
12. What is Azure Data Storage?
13. What are the types of storage options available in Azure Data Storage?
14. What are Azure Containers?
15. What is Azure Data Factory?
16. How to build machine learning pipelines using Azure Data Factory?
17. What is Azure Databricks?
18. How to deploy your machine learning code into Azure Databricks?
19. What is Git?
20. What is the difference between Git and Github?
21. What is Docker?
22. How to create a docker image for your machine learning application?
23. How to containerize your machine learning application using Docker?
24. What is Azure App Services
25. How to deploy a Streamlit user interface using Azure App Services?