CHATBOT FOR ML APPS

CAPSTONE PROJECT REPORT SUBMITTED

TO THE

INDIAN SCHOOL OF BUSINESS

FOR THE

ADVANCED MANAGEMENT PROGRAMME IN BUSINESS ANALYTICS

SUBMITTED BY

Group 22

Sidhartha Reddy-12110071

Mahesh G-12110019

Varshini Modulla-12110052

Niharika Miriyala-12110013

UNDER THE GUIDANCE OF

Prof. Sumanta Singh

INSTITUTE OF DATA SCIENCE

INDIAN SCHOOL OF BUSINESS

December 2022

CAPSTONE PROJECT REPORT

SUBMITTED TO THE INDIAN SCHOOL OF BUSINESS FOR ADVANCED MANAGEMENT PROGRAMME IN BUSINESS ANALYTICS

HEADING	DETAILS
Title of the Capstone project	Chatbot For ML Apps
Name of the Candidate	Sidhartha Reddy Mahesh G Varshini Modulla Niharika Miriyala
Name and Affiliation of the Faculty Mentor	Sumanta Singh, Assistant Professor of Information Systems, ISB
Name and Affiliation of the Industry Mentor	Vineet Srivastava
Batch	16
Date/Month of submission of Capstone Project Report	December 03, 2022

Acknowledgement

We would like to take this opportunity to convey our special gratitude to the Project sponsor Mr **Sriram M** of Zero Code Learning and Client Mentor Mr **Vineet Srivastava** for providing us with a great opportunity to work on **'Chatbot for ML Apps'** project and providing the necessary guidance and support in every phase of the project. We will always be thankful to you in this regard.

Our sincere thanks to Prof. **Sumanta Singh** for mentoring us to overcome technical challenges and making us think outside of the box by providing great guidance and support.

We are also thankful to the Indian School of Business for providing all the required facilities for working on the project.

Sidhartha Reddy

Mahesh G

Varshini Modulla

Niharika Miriyala

December 2022

Contents

List of Tables

List of Figures

Abstract

Customer satisfaction is a key driver to business growth and ensuring effective customer support availability 24*7 enhances customer experience. However, following a traditional customer support approach i.e., email & call support adds to the cost of operations. Efficient chatbots come in handy in providing 24*7 support with lower operations costs. With the rise of natural language processing techniques, chatbots are becoming more advance and efficient to address the majority of user queries.

Zero coding learning is an online platform which provides a wide variety of machine learning models to explore without the need for coding by users. Presently the company supports users on the platform with email support and is not scalable. The proposal is to develop chatbots to provide a quick resolution to the customer and direct to customer service for unanswered questions. This project's scope is to develop an AI+NLP-based chatbot for proof of concept (POC). The delivered POC is based on BERT based Question Answering Model which uses dynamic model selection approach to choose best output from traditional BERT QA Model + pattern matching module and hugging face BERT QA Model pipeline. The performance of the model is measured using ROUGE approach and the model delivered accuracy of 91% and average response time of 33 seconds in personal laptop configuration.

Executive Summary

Zero-code learning provides an online platform referred to as ML apps which enables users to explore a wide variety of ML models without the need for coding. It is an Edutech platform primarily used by academicians and students. Users reach out to the customer support team via email in case of any issues faced while accessing the platform. The traditional method of email support is not sustainable due to the increase in the user base on the platform. There is a delay in addressing customer queries hence impacting overall customer experience. The objective of this project is to provide a chatbot which can address queries and reduce the burden on the customer support team. The key deliverable of the project is to provide POC (Proof of concept) on a Chatbot using NLP and Machine Learning techniques. The project is divided into various stages i.e., requirement gathering, literature review, data collection, model approach finalisation, model building, training and validation.

In the requirement gathering phase, it is understood that there is limited availability of user query data which adds to the complexity of the project in delivering accuracy. However, the initial expectation was to build to base accuracy model which can be trained further with more data.

As per the literature review conducted, it is identified that BERT based Intent classification and Question Answering (QA) models are state of art architecture for chatbot solutions. However, during exploring stage, Intent classification model was returning mostly incorrect responses due to small dataset. Since large dataset is not available, it was not feasible to improve the performance of intent classification model. In the next stage of exploration, QA BERT is modelled. QA is an NLP task which allows users to pose a question in Natural Language and the answer is extracted from a reference text provided.

The solution is built in multiple stages, in first stage the solution architecture takes the input query of the user, perform the basic NLP pre-processing tasks and fed into BERT based QA Model. The model fetches the relevant content from the user manual input which is fed in particular format. As the QA Model returns only phrase as an output, an additional pattern matching module is added to extract the complete sentence from the context. The above model is referred as base-line model in the solution document. As the base-line model has lower accuracy, hugging face BERT QA model pipeline is explored and this model provided better accuracy. However, during the model evaluation phase, it is identified for few of questions, base model gave better response then pipeline model. In order to get the best of both the models, dynamic model section is added to architecture. The user query is passed to updated architecture in which the responses of both the base model and pipeline model is captured and is compared with question. The response which has highest cosine sentence similarity with the question is chosen dynamically and best responses is passed as output to the user.

In order to enable continuous learning, user feedback in terms of ratings is collected after chatbot response. These rating are stored against user ID and query asked by user in the tables. On regular interval, low rating questions are analysed and source input is updated to add the missing content. This process also enables to understand the frequent questions asked in chatbot by the users and also to asses model performance from user perspective.

Overall Model performance is measured using Accuracy and Latency. Model accuracy is measured using ROUGE metric which provides details of precision, recall and F1 score and response time is captured using ‘Python Time’ package.

Chapter 1: Introduction

1.1 Introduction

A chatbot is a software application that simulates human conversation through a set of pre-designed rules to mimic real-life interactions and answer customer queries. Chatbots that use Artificial Intelligence and Natural Language Processing can analyze these interactions at an almost human level. They can instantly engage with the users with a specific message tailored to each user.

There are two types of Chatbots that an enterprise can use based on their necessity – Rule-Based Chatbots & AI chatbots. Rule-based Chatbots communicate through pre-defined rules and a set of questions. They are not capable of generating answers on their own but with extensive training & smartly designed rules, rule-based chatbots are quite helpful. Also, they do not need a lot of example questions to feed the model for it to give an accurate response.

AI-based Chatbots use a complex model of Machine Learning & NLP to understand sentence structure and self-learn with the help of data that is fed and generate the answers accordingly. These chatbots can understand the intent behind the user's complex questions, improves response accuracy through continuous learning

The popularity of chatbots is increasing due to the increasing benefits it is offering to organizations. The tool helps the companies provide immediate resolutions to the unpredictable amount of customer queries/complaints which were not possible for the customer service team to handle alone.

This report covers a detailed approach to AI+NLP Chatbot to address user queries of the ML apps platform provided by Zero Code Learning. AI-based chatbot requires fewer human interventions compared to rule-based and understanding human intents & self-learn, this would be of great help to the organization in providing immediate assistance and accommodating multiple users at the same time. Also, an AI-based chatbot can take the data of the existing user manuals & FAQ documents and can answer the queries of users by recognizing multiple forms of the same query.

1.2 Glossary/Terminologies

Intent classification: An Intent classifier analyzes texts and categorizes them automatically into intents. Every customer interaction has a purpose or an intention. An intent classifier is useful in understanding the intentions behind customer queries and gaining valuable insights from them. Intents map the user queries with the 'action' that the Bot must perform

BERT Language model: Bidirectional Encoder Representations from Transformers is a transformer-based machine learning technique used for natural language processing (NLP) pre-training. BERT learns contextual relations between words in a text through an attention mechanism. The transformer includes two separate mechanisms of which an encoder reads the text input, and a decoder produces a prediction for the task

LSTM: LSTM is an advanced form of Recurring neural network (RNN) that can handle long-term dependencies. The internal functioning of LSTM has 3 parts: First part chooses if the information that is coming from the previous timestamp is to be remembered or is irrelevant so can be forgotten. In the second part, the cell tries to learn new information from the input received by this cell and in the third part, the cell passes on the updated information from the current timestamp to the next one. The cell state is referred to as long-term memory whereas the hidden state is short-term memory.

Regular Expression (RegEx): A regular expression is a way to do pattern-matching patterns with some sequence of characters.

Hugging face: Hugging Face is an open-source data science platform that provides tools that enable users to build, train and deploy ML models.

QuestionAnswering : Question Answering models retrieve the answers to a question asked from the given input text document which is used to search answer. The QA systems have different variants such as 'Extractive QA' and 'Generative QA'.

In Extractive based QA, the model extracts the answers from the context/input file and provides it directly to the user. It usually leverages 'BERT-like' models. In Generative QA, the model gives free text directly based on the context and this method uses 'Text Generation' models

ROUGE: Recall-Oriented Understudy for Gisting Evaluation(ROUGE) is a set of metrics for performance evaluation of NLP tasks. ROUGE-N measures the number of matching ‘n-grams’ between model generated text and a ‘reference’ which is input document. The ‘N’ represents the n-gram that we are using for the model. Based on the ‘N’ decided recall, precision, or F1 score are calculated.

Chapter 2: Review of Related Literature

A literature review is conducted focusing on the following challenge in the solution approach

- Limited data and variety in user queries
- A significant increase in user queries, both by volume and by variety, is expected
- Chatbot to resolve both tool-based and concept-based queries

Besides looking for applied solutions for the above-mentioned challenges, we also reviewed publications to understand the latest techniques in chatbots. The end goal is to provide a flexible, learning chatbot that is easy to maintain for the business

Publications in journals of international repute, like IEEE, Journal of Applied ML, Springer etc., were explored. While many papers were reviewed, following publications are most relevant to the problem statement and challenges.

Below is the synopsis of the literature review:

Title	Design Approach
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	**BERT** is finetuned for **NLP tasks like paraphrasing, question answering, text classification, and sequence tagging**. Also explains the feature extraction-based approach with Bert Use Case: Multiple NLU and NLG tasks Dataset: SQuAD, SWAG Metrics: GLUE score, MultiNLI accuracy and F1 score
Question and Answering Using BERT	**Bert model** with the embedding layer at the top, encoder layer and end layer with Linear layer. In output layer, SoftMax generates the predicted probability of start & end position of the answer
End-to-End Open Domain Question Answering with BERTserini	BERTserini is comprised of two modules: Retriever is responsible for selecting segments of text that contain the answer, which is then passed to the reader to identify an answer span
Chatbot: A Conversational Agent employed with Named Entity Recognition Model using Artificial Neural Network	Intent Classification using BOW (bag of words) + multi-class NN NER using Spacy + multi-class NN Use Case: Chatbot for Cab Booking Dataset: Manually prepared test set. CoNLL-2003 dataset for validation

	Metrics: Intent Classification - Precision, Recall, F1-score NER - Confusion matrix and classification report
A New Chatbot for Customer Service on social media	LSTM networks are applied to generate responses for customer-service requests on social media. The system takes a request as the input, computes its vector representations, feeds it to LSTM, and then outputs the response. Use case: Social media customer service Dataset: Twitter Conversations between users and agents over 60+ brands Metrics: The similarity measure was based on a TF-IDF weighted vector space model. The quality of responses was measured by human judgments and an automatic evaluation metric.
An Automated Conversation System Using Natural Language Processing (NLP) Chatbot in Python	This paper provides NN based NLP chatbot covers all the packages required to explore NN based chatbot Use case: Building GUI and NLP+NN based chatbot Dataset: Kaggle dataset

Table-2.1

Note: The details of other relevant research papers have been included in the reference section

In addition to publications, sources like Kaggle, and GitHub are also used for developing solution approach

Chapter 3: Project Description

3.1 Business/Research Problem

Zero Code Learning offers ready-to-use Machine Learning models through ML Apps for B -schools, academicians, and students. ML Apps provide simple and easy-to-navigate buttons and tabs to explore the models, parameters, and data insight.

Users can train the model using their data set and make predictions. Zero Code Learning aims to help students and academicians to focus on learning business challenges and outcomes instead of dealing with the nuances of coding the Machine Learning models. However, users often need help with tool usage. They reach out to the support team through emails/tickets and often, there is a delay in assistance which is negatively impacting customer learning experience and satisfaction.

3.2 Data Available

- The sponsor shared the information of login-related queries that are presently received on the platform which is an excel with 230 records
- User manuals & FAQs were shared for 13 models that are available on the platform

Below snapshots of the data provided:

S. No	Nature of Call	Call Status	Remarks
1	Unable to Login	Closed	Account reset done
2	Segmentation App not Available	Closed	Instructor to Enable
3	Unable to Login	Closed	Account reset done
4	Login Credentials Not Received	Closed	Email ID Mismatch
5	Unable to Login	Closed	Account reset done

Fig.3.2.1

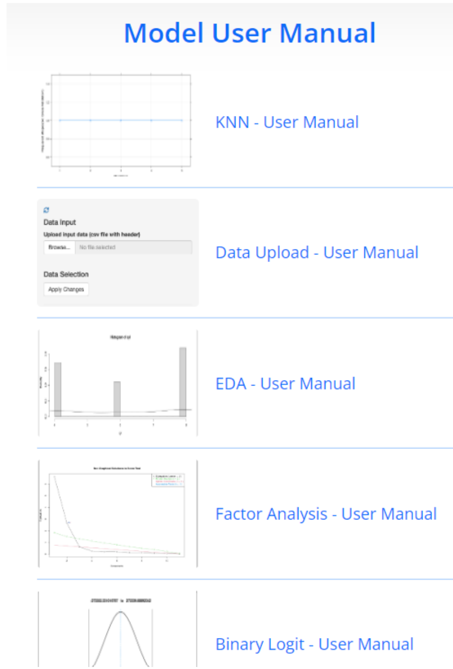

Model User Manual
KNN - User Manual
Data Input
Upload input data (csv file with header)
Browse...
No file selected
Data Selection
Apply Changes
Data Upload - User Manual
EDA - User Manual
Factor Analysis - User Manual
Binary Logit - User Manual

Fig.3.2.2

Chapter 4: Assumptions, Approach & Process

4.1 Assumptions

The following assumptions are made **regarding the model**:

- Users will use Chatbot only to pose questions on tool usage and basic theoretical questions on the models
- Though we initially have minimal data, the database will be periodically updated by the support team to create a huge corpus
- The user queries will be in English language only
- The user will ask only one question at a time

The following assumptions are made **regarding Deployment & Maintenance**:

- The support team will maintain the list of questions that are routed to them and will periodically update the context file
- Users rating captured after the response will be periodically analyzed and low rating questions will be addressed by updating the context file

4.2 Approach

Chatbots are conversational tools that help in customer service. There are traditional rule-based chatbots which are pre-programmed in the backend based on a few rules and conditions. It is trained to identify specific phrases or a combination of phrases and is triggered to act upon encountering them. But the disadvantage with this Keyword based approach is that they must be explicitly trained and the chatbot can understand the user only up to what it has been trained to.

4.2.1 Why Conversational AI Chatbots

On the other hand, Conversational AI Chatbots use Natural Language Understanding (NLU) and Natural Language Processing (NLP), and they can learn from each interaction with the user. They can understand almost everything, especially when Deep Learning Techniques like Transformers are used. The advantages of AI chatbots include increased customer engagement, high availability, cost saving, and proactive assistance, leading to improved customer satisfaction and overall business.

4.3 Proposed Architecture

The following image describes our proposed architecture:

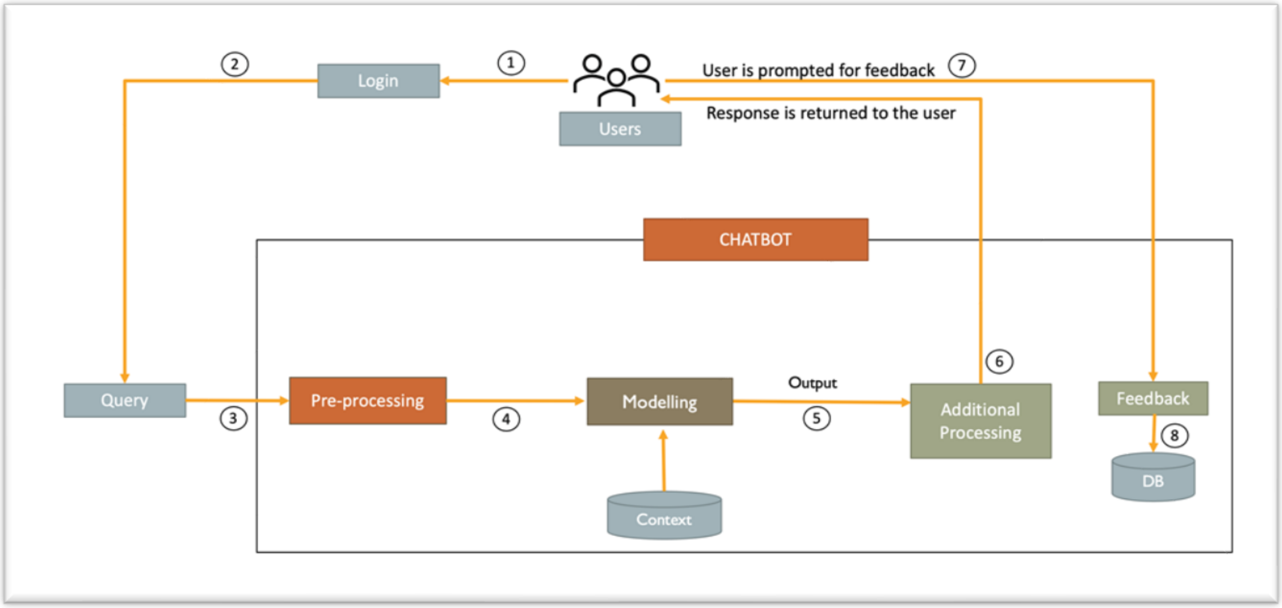


Fig.4.3

Step 1: User logins and tries out multiple models

Step 2: When the user gets a query, the user will shoot it to the Chatbot

Step 3: The Chatbot pre-processes the query using NLP techniques

Step 4: the query will be passed to the model trained

Step 5: The model returns the output to the query by referring to the context text

Step 6: Additional processing like Pattern Matching is performed to retrieve the complete sentence is done using the output

Step 7: The response is then returned to the user. If the model is unable to answer the question, it will return a standard response asking the user to reach out to info@zerocodelearning.com

Step 8: The user is then asked to rate their conversation experience on a scale of 1-10

Step 9: The feedback from the user is stored in the database along with the username, user ID, query posed by user and response returned by the chatbot

4.4 Technical Architecture

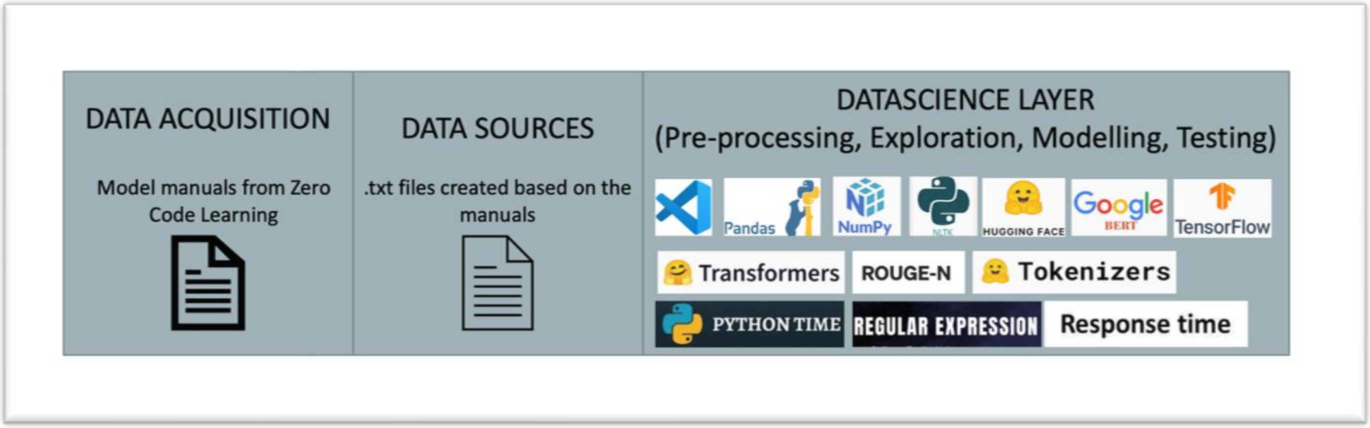


Fig.4.4

4.5 Technology Stack

Name	Version
Visual Studio Code	1.58
Python	3.8.8
Tensor Flow	2.8.3
Hugging Face Transformers	4.15.0
ROUGE	1.0.1
Hugging Face Sentence Transformers	2.1.0
Text Wrap	0.9.2
GitHub	3.5.8
Time	3.8.8

Table-4.5

4.6 Solution Design Diagram

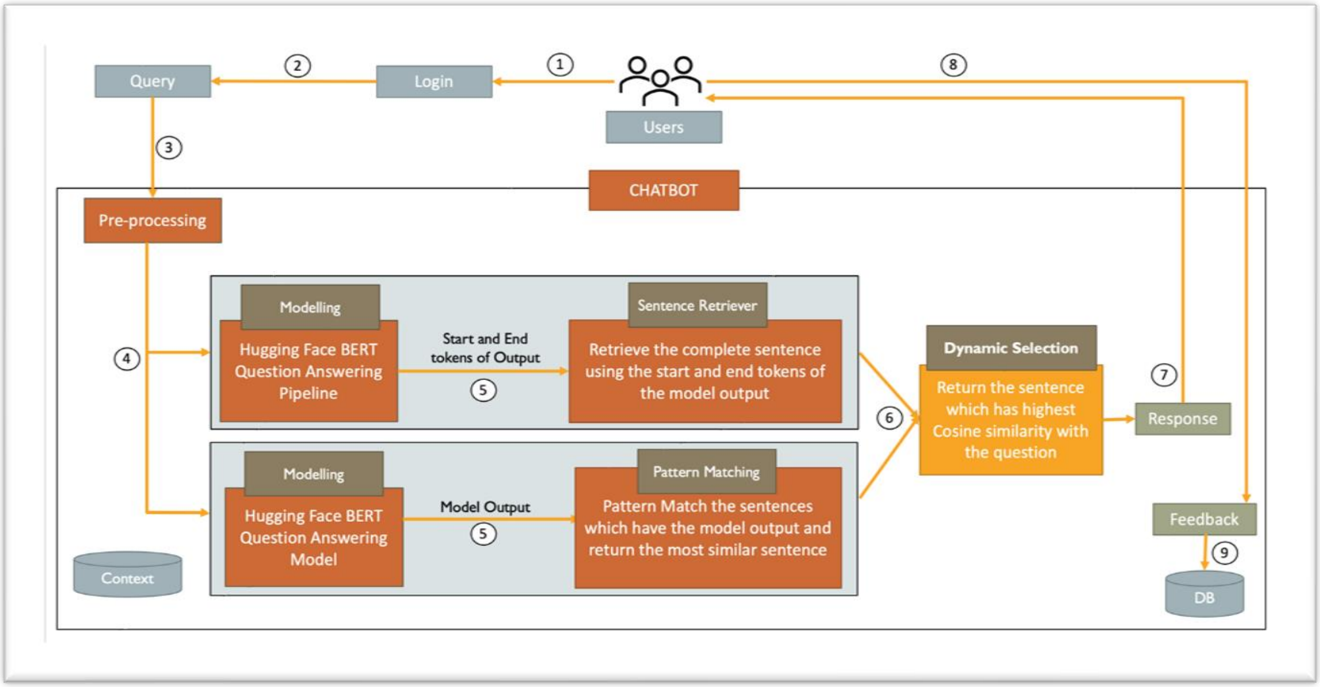


Fig.4.6

4.7 Solution Description

The following steps describe the propose solution

Step 1: User logins and tries out multiple models

Step 2: When the user gets a query, the user will shoot it to the Chatbot

Step 3: The Chatbot pre-processes the query using NLP techniques

Step 4: The query will be passed to 2 models – Baseline and Pipeline BERT Question Answering models

Step 5: The Baseline model returns the output phrase and then Phrase Matching is performed on the context file to retrieve the whole sentence. The Pipeline model returns the start and end tokens of the output using which model can retrieve the whole sentence from the corpus

Step 6: The best response is selected from both the model responses by calculating cosine similarity against each response and the question.

Step 7: The best response is then returned to the user. If the model is unable to answer the question, it will return a standard response asking the user to reach out to info@zerocodelearning.com

Step 8: The user is then asked to rate their conversation experience on a scale of 1-10

Step 9: The feedback from the user is stored in the database along with the username, user ID, query posed by user, and the response returned by the chatbot

Chapter 5: Feature Selection and Engineering

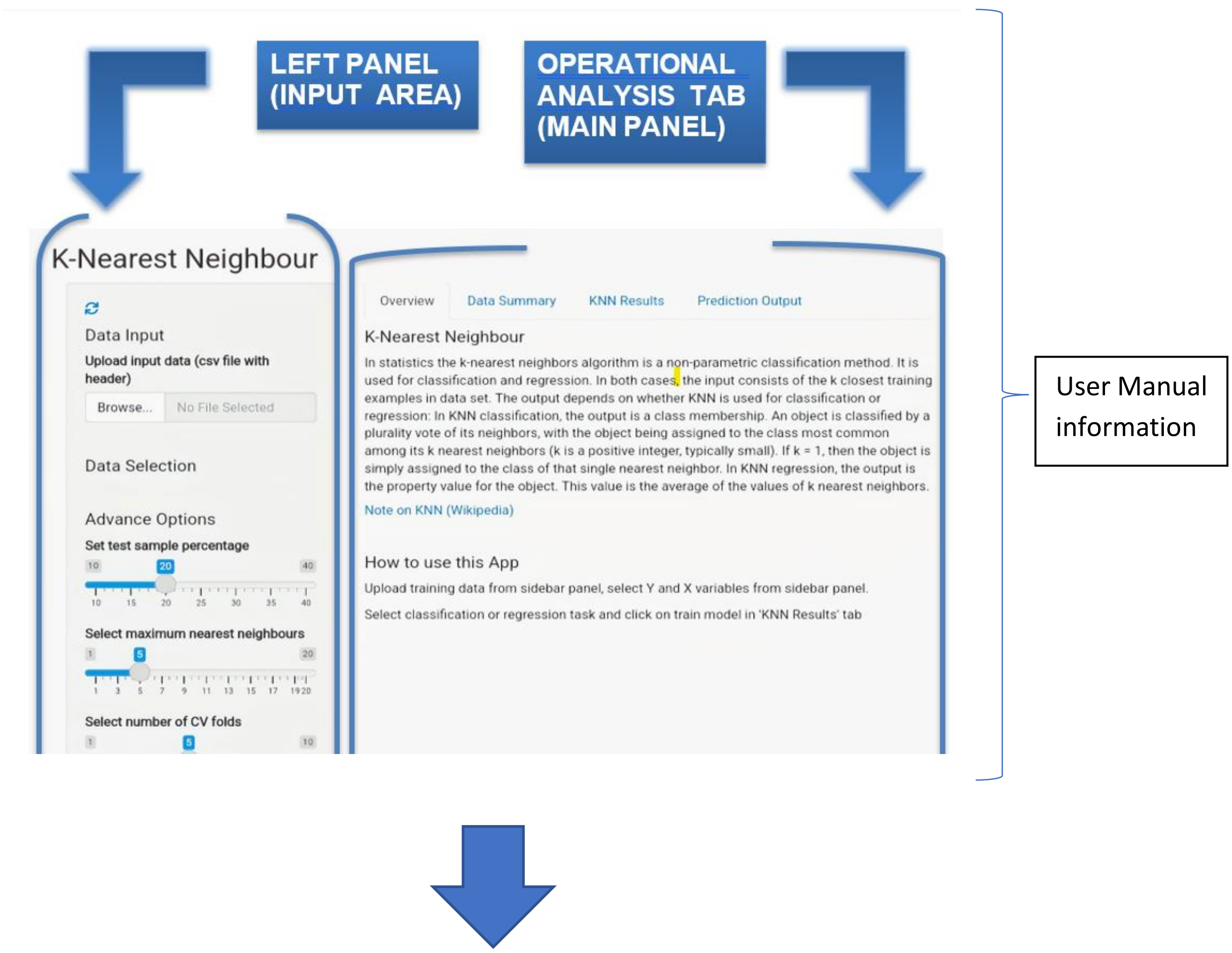
Built a database of two textual documents from the User manuals

- A textual document containing the common functionality of all models
- A textual document containing model-specific functionalities

Input file preparation:

The manual received from zero code team contains images and textual information about the models. In order for the model to process the information, the manuals are converted into text files post processing. Further, the information from the manuals is paraphrased to the extent necessary for the model to retrieve accurate responses to the user queries.

Sample User Manual:



Processed Text (referred as Context file):

```
The left panel has Advanced Options section where there is option to Set test sample percentage, Use the toggle bar to set required percentage of test sampl
The left panel has Advanced Options section where there is option to Select maximum nearest neighbours, Use the toggle bar to set required number of nearest
The left panel has Advanced Options section where there is option to Select number of CV folds, Use the toggle bar to set required number of Cross-Validatio
KNN results tab consists of model accuracy & performance for different k-values along with a graphical representation for ther same.
```

Paraphrased Textual document

Fig.5.1

Model standard pre-processing:

Few preprocessing tasks on the context file is performed using the inbuilt libraries of hugging face & Bert such as tokenization, conversation of texts to lower case, punctuation removal etc.,

Chapter 6: Models Used

6.1 Transformers

The Transformers were introduced in 2017 through a paper titled ‘need. Before transformers, the state-of-the-art techniques were Recurrent Neural Networks (RNN) and Long Short-Term Models (LSTM). Transformers were able to beat LSTMs as they capture the context of a word from both left to right and right to left. Their architecture uses Attention heads to significantly improve the performance of deep learning models, especially for Natural Language Understanding (NLU) and Natural Language Processing (NLP) applications.

6.2 Bi-Directional Encoding Representation Transformers (BERT)

BERT is a Transformer based Deep Learning technique developed by Google in 2018. It is the state-of-the-art technique for NLP tasks for three reasons:

- It’s trained on a huge corpus of Wikipedia and Book Corpus dataset
- It captures context from both left and right sides (Bidirectional Encoding)
- It uses attention mechanisms with Next Sentence Prediction (NSP) and Masked Language Modelling (MLM) techniques to understand the context

Hugging Face provides many pre-trained versions of BERT. Bert Large has 24 layers of encoders stacked and while Bert Base has 12 layers. Due to its large size, BERT Large has 340 million parameters making its production incompatible. So, we used the Bert Base model.

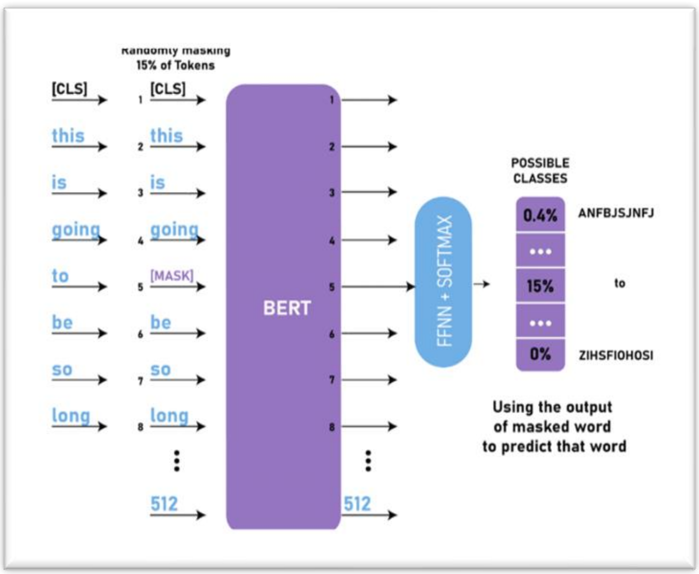


Fig.6.2

6.3 Intent Classification

As the user base increases for a platform, we receive many customer interactions a day regarding different models. These interactions are unstructured and for a chatbot to respond to a query, it gets difficult to understand the context of the question. Intent Classification helps chatbots by classifying the query into pre-defined categories called intents.

6.3.1 Architecture

To model the Intent classification, we used the following approach.

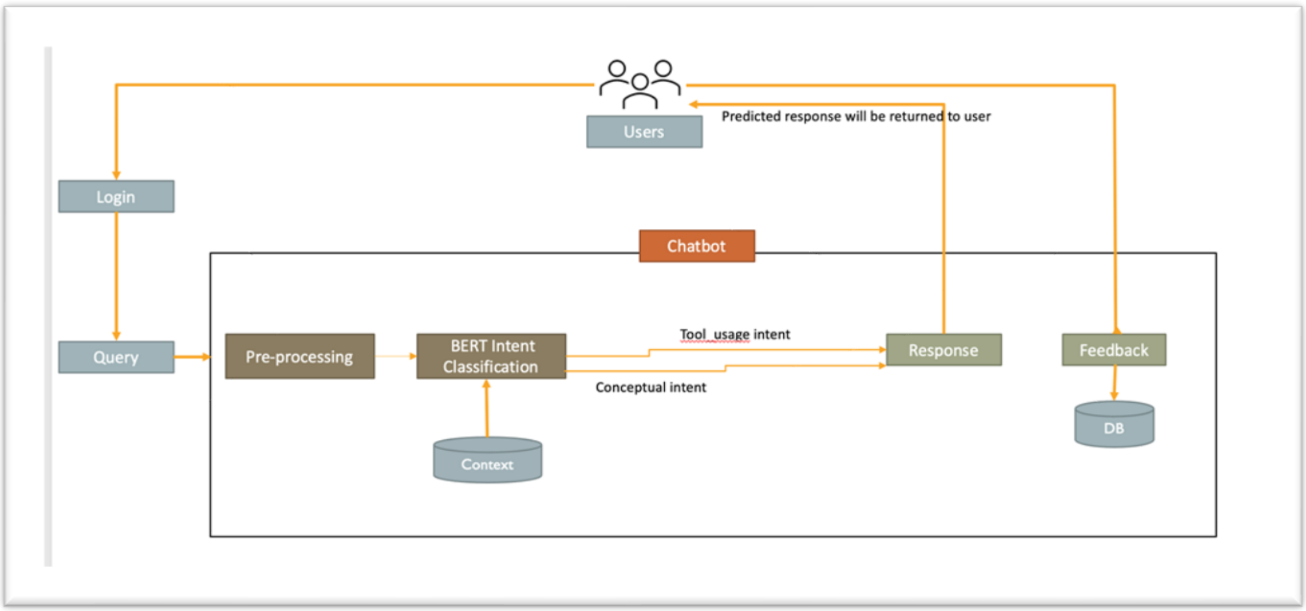


Fig.6.3.1

6.3.2 Solution Description

Step 1: User logins and tries out multiple models.

Step 2: When the user gets a query, the user will shoot it to the Chatbot

Step 3: The Chatbot pre-processes the query using NLP techniques

Step 4: The query will be passed to the BERT Intent Classification model and it identifies whether it is a tool_uage intent or conceptual intent and return the response based on the intent

Step 5: The response is then returned to the user. If the model is unable to answer the question, it will return a standard response asking the user to reach out to info@zerocodelearning.com

Step 6: The user is then asked to rate his conversation experience on a scale of 1-10.

Step 7: The feedback from the user is stored in the database along with the username, user ID, query posed by user, and response returned by the chatbot.

6.3.3 Disadvantages of Intent Classification Model

- Due to the very small dataset, the number of incorrect answers was very high and there was no way to improve without more data.

6.4 Question Answering

Question Answering is an NLP task which allows users to pose a question in Natural Language and the answer is extracted from a reference text provided. Question Answering is widely used in Search Engines, Chatbots and other conversational interfaces.

6.4.1 BERT for Question Answering

Hugging Face provides 'TFBertForQuestionAswering', a BERT model which is pre-trained on Stanford Question Answering Dataset (SQuAD) and Conversational Question Answering dataset (CoQA) with Next Sentence Prediction (NSP) and Masked Language Model (MLM) objectives to learn to extract information from the reference text. The SQuAD dataset is a crowd-sourced reading comprehension dataset prepared by posing questions on Wikipedia articles and the CoQA dataset is a large dataset for building conversational interfaces. This model is so good that it works very well on new datasets with no need for fine-tuning. As a result of this, the computing power, and efforts of finetuning and retraining the models are no longer needed.

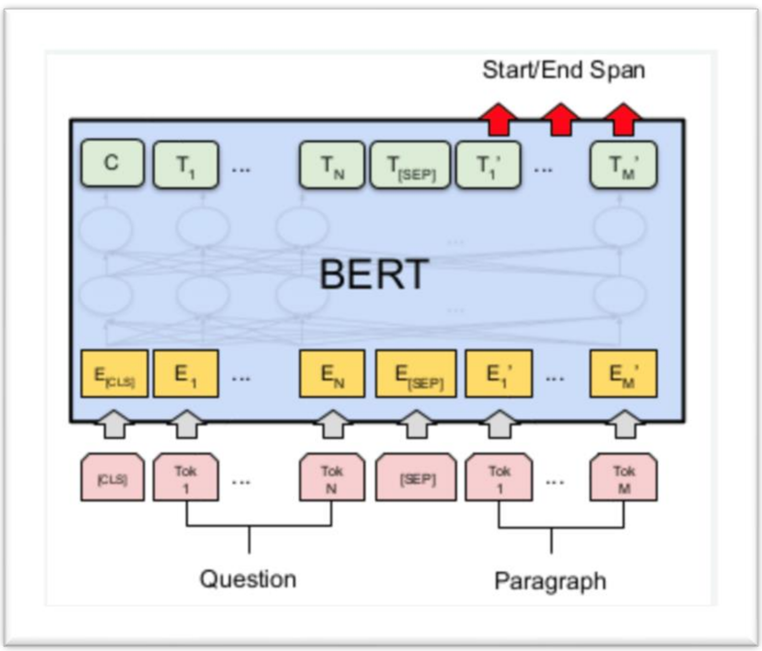


Fig.6.4.1

For the Question Answering System, BERT takes two parameters, the input question, and passage as a single packed sequence. They have separated with [SEP] tokens. Along with Token Embeddings, BERT uses Segment Embeddings and Position Embeddings. The Segment Embeddings help differentiate the question and reference text by using a vector of 0s for the question and 1s for the reference text. The Position Embeddings help specify the position of words in the provided sequence, the input embeddings are the sum of the token embeddings and the segment embeddings

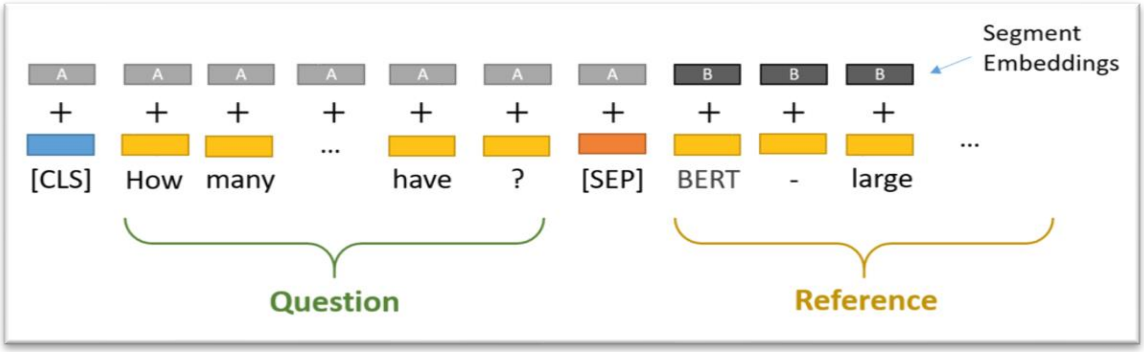


Fig.6.4.2

6.5 QA BERT Baseline Model

BertForQuestionAnswering is a BERT Question Answering model provided by Hugging Face Transformers. This architecture is modelled in this baseline architecture to return a response to the query.

6.5.1 Architecture

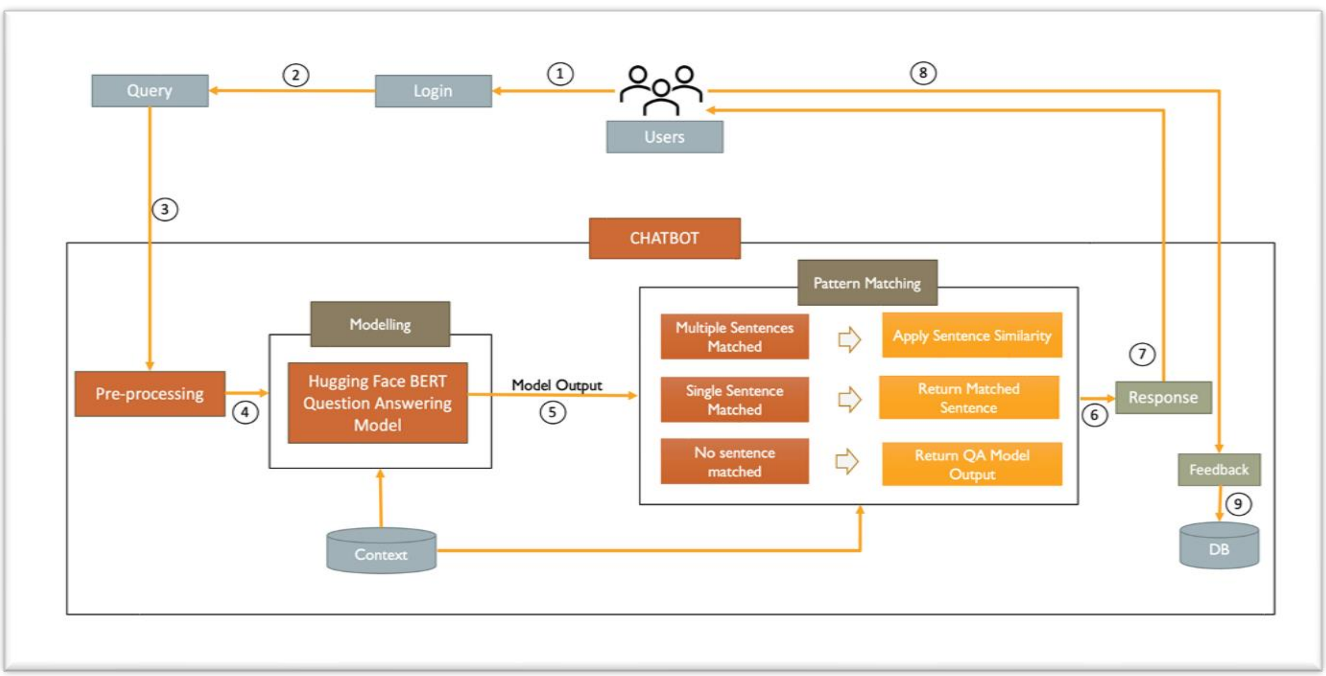


Fig.6.5.1

6.5.2 Solution Description

Step 1: User logins and tries out multiple models.

Step 2: When the user gets a query, the user will shoot it to the Chatbot

Step 3: The Chatbot pre-processes the query using NLP techniques

Step 4: The query will be passed to the BertForQuestionAnswering model and it returns the output phrase by referring to the context files

Step 5: Using the output phrase, Phrase Matching is performed on the whole context to retrieve the whole sentence which has this phrase

Step 6: If there are more than one sentences are matching with the output phrase, then sentence similarity is calculated between each of the sentences and the question and the sentence with the highest similarity is returned. If only one sentence is matching, then it is returned directly. If no sentence is matching but the model is able to derive the answer from two or three sentences, then the output phrase is returned directly

Step 7: The best response is then returned to the user. If the model is unable to answer the question, it will return a standard response asking the user to reach out to info@zerocodelearning.com

Step 8: The user is then asked to rate their conversation experience on a scale of 1-10

Step 9: The feedback from the user is stored in the database along with the username, user ID, query posed by user, and response returned by the chatbot.

6.5.3 Sample Workflow

Reference text:
The Main panel has PCA visualization tab which shows 2D visualization of clusters.
The Main panel consists of a Tab called Cluster Analysis which has a graph showing optimal number of clusters by dotted line.

Question: Where can I see PCA visualization?

Answer returned by model : "main panel"

List of sentences returned after pattern matching:
[' the main panel has pca visualization tab which shows 2d visualization of clusters.',
'the main panel consists of a tab called cluster analysis which has a graph showing optimal number of clusters by dotted line.']

Sentence Similarity Scores:

	question	answer	score
0	Where can I see PCA visualization?	main panel has pca visualization tab which shows 2d visualization of clusters.	0.7697
1	Where can I see PCA visualization?	the main panel consists of a tab called cluster analysis which has a graph showing optimal number of clusters by dotted line.	0.3941

Response to the User: 'the main panel has pca visualization tab which shows 2d visualization of clusters.'

Fig.6.5.3

As we can see in the above picture, the output of the model is the 'main panel'. To retrieve the whole sentence having the main panel phrase, an additional pattern-matching module is used. In the above example, there are two sentences having the output phrase. So, the sentence similarity between each sentence and the query is calculated and the one with the highest score is returned as the response to the model.

6.5.4 Disadvantages of Baseline Model

- The sanctity of the reference text is lost when it is tokenized. So, it is not possible to return the complete sentence to the model

6.6 QA BERT Pipeline Model

The pipelines offer an excellent and simple method for using models for inference. These pipelines are objects that provide a straightforward API dedicated to various tasks while abstracting most of the complex code from the library.

6.6.1 Architecture

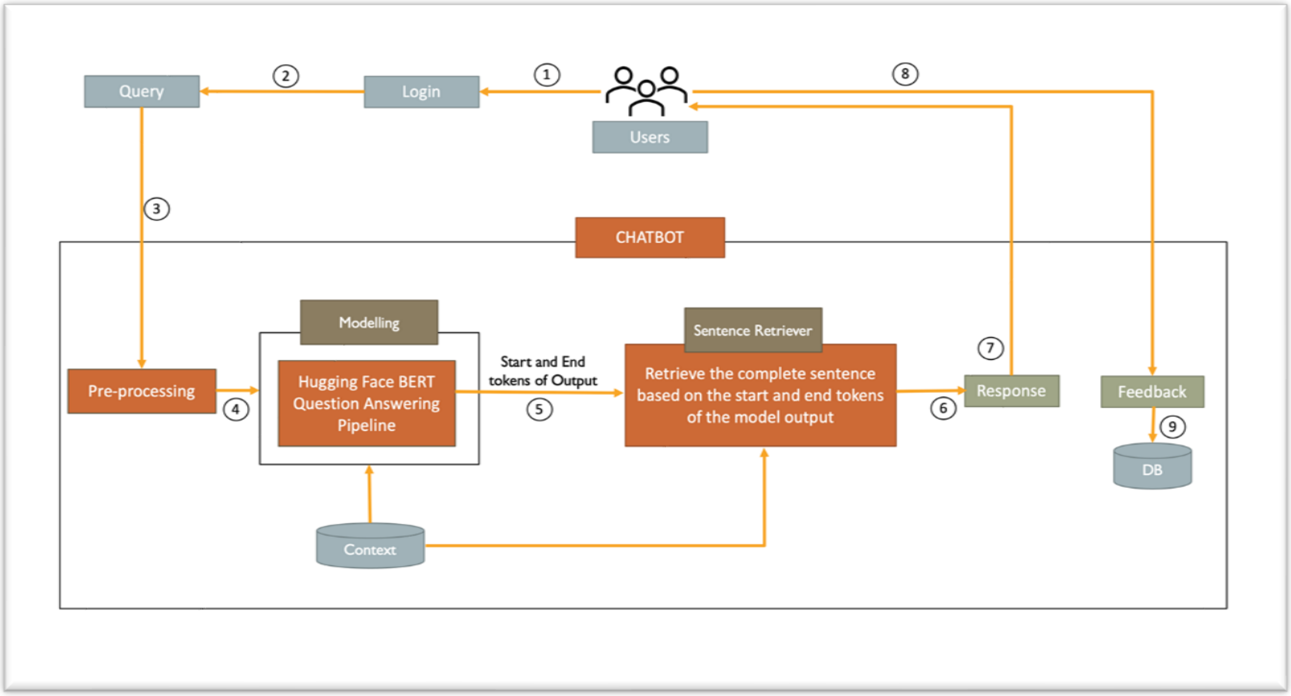


Fig.6.6.1

6.6.2 Solution Description

Step 1: User logins and tries out multiple models

Step 2: When the user gets a query, the user will shoot it to the Chatbot

Step 3: The Chatbot pre-processes the query using NLP techniques

Step 4: The query will be passed to the BertForQuestionAnswering pipeline

Step 5: The model outputs the answer along with the start and end tokens of the output

Step 6: With the start and end tokens, the complete sentence in which the answer is present is retrieved

Step 7: The best response is then returned to the user. If the model is unable to answer the question, it will return a standard response asking the user to reach out to info@zerocodelearning.com

Step 8: The user is then asked to rate their conversation experience on a scale of 1-10

Step 9: The feedback from the user is stored in the database along with the username, user ID, query posed by user, and response returned by the chatbot.

6.6.3 Sample Workflow

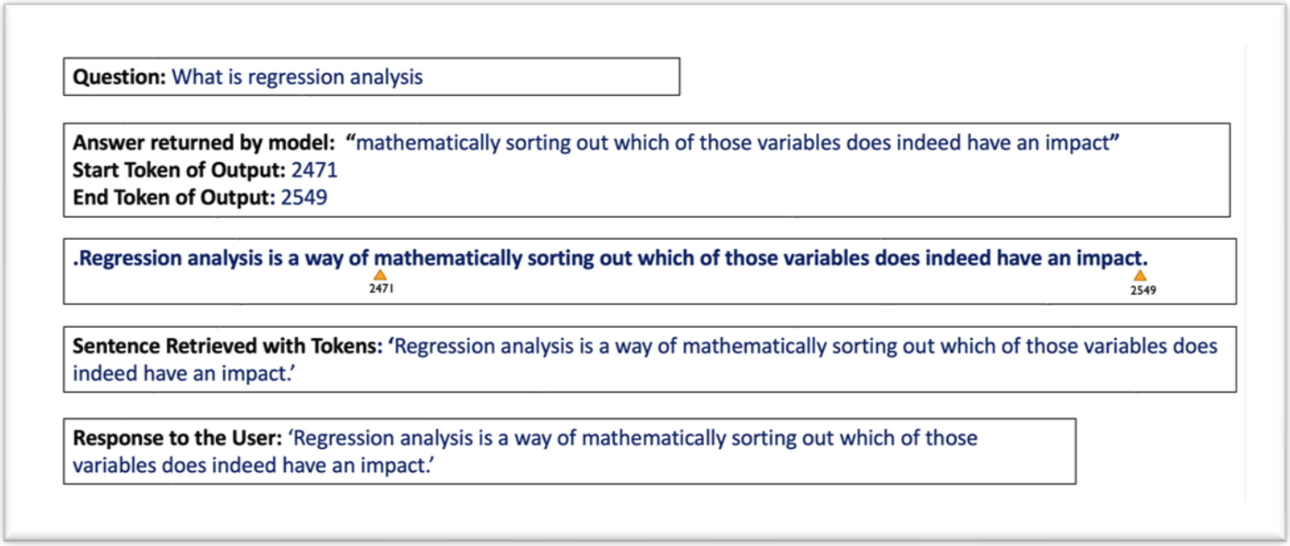


Fig.6.6.3

In the above picture, we can see that the model is returning the start and end tokens of the model output. Based on these tokens, the whole sentence between two full stops is retrieved and returned to user.

6.6.4 Advantages of Pipeline Model

- The Hugging Face pipelines have additional post-processing to enhance the model's output
- The pipeline returns the start and end tokens of the answer, preserving the sanctity of the text, and allowing us to extract the complete sentence
- Accuracy increases as there is no information loss caused by pattern matching
- Probability score returned by the pipeline can be used to analyze the confidence of the answer given by chatbot

6.7 QA BERT Dynamic Model Selection

The motivation for building this model is to answer the few questions which pipeline model is incorrectly answering but the baseline model is correctly answering. This architecture gets the best out of the baseline and pipeline models.

6.7.1 Architecture

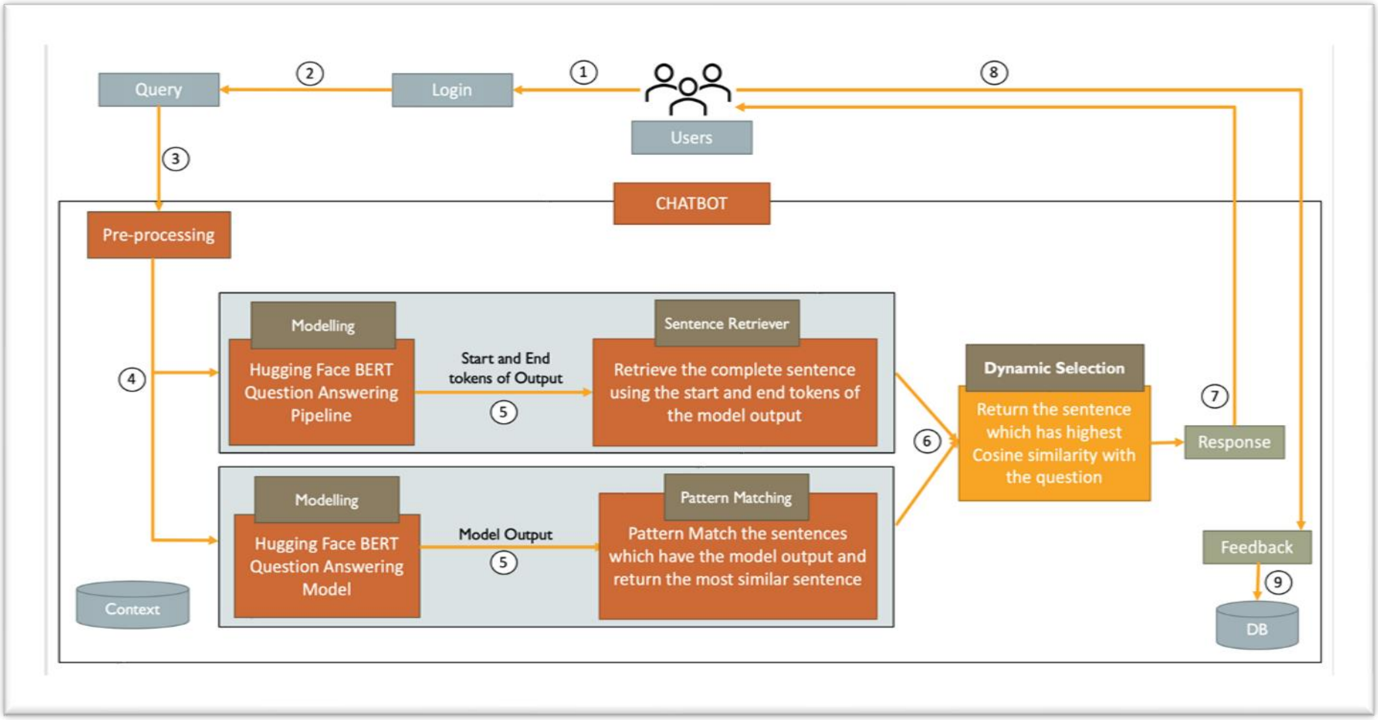


Fig.6.7.1

6.7.2 Solution Description

Step 1: User logins and tries out multiple models.

Step 2: When the user gets a query, the user will shoot it to the Chatbot

Step 3: The Chatbot pre-processes the query using NLP techniques

Step 4: The query will be passed to 2 models – Baseline and Pipeline BERT Question Answering models

Step 5: The Baseline model returns the output phrase and then Phrase Matching is performed on the whole context to retrieve the whole sentence. The Pipeline model returns the start and end tokens of the output using which we can retrieve the whole sentence from the corpus

Step 6: The best response is selected from both the model responses by calculating cosine similarity against each response and the question.

Step 7: The best response is then returned to the user. If the model is unable to answer the question, it will return a standard response asking the user to reach out to info@zerocodelearning.com

Step 8: The user is then asked to rate their conversation experience on a scale of 1-10

Step 9: The feedback from the user is stored in the database along with the username, user ID, query posed by user, and response returned by the chatbot

6.7.3 Sample Workflow

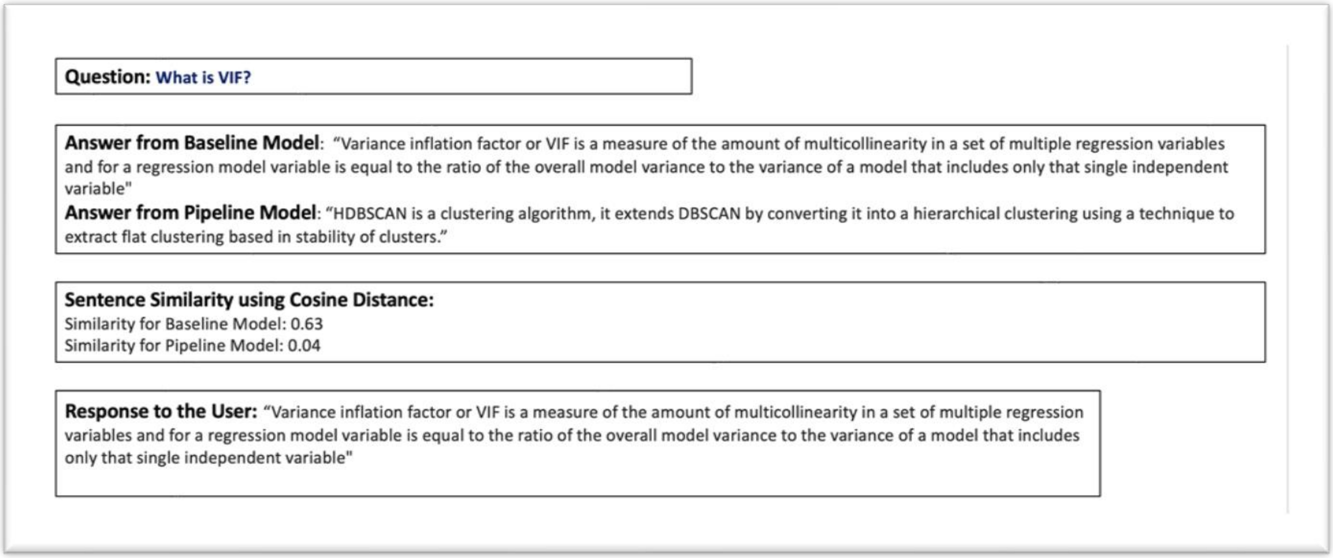


Fig.6.7.3

As shown in the above image, the query is passed to both Baseline and Pipeline models. Then Cosine Similarity is calculated between the question-answer pairs and the answer with the highest similarity is returned to the user.

6.7.4 Advantages of Dynamic Model Selection

- The few questions which are incorrectly answered by Pipeline model are correctly answered by Baseline model
- Combining them both with a layer of Dynamic Selection increases the accuracy of the chatbot
- The Dynamic Selection module calculates sentence cosine similarity between the answers given by each model and the question and returns the answer which matches the most

6.7.5 Disadvantages of Dynamic Model Selection

TRADEOFF: As the accuracy increases, the response time also increases as two models are deployed

6.8 Models and Metrics Considered

Model	Brief approach	Challenges
Baseline BERT model	This model uses the pretrained generic model (AutoModelForQuestionAnswering) as provided by Hugging Face. Based on the start and end tokens generated, complete sentence is extracted as potential response to user query. To deal with multiple responses, SentenceTransformer is used to calculate cosine similarity between each question-answer pair. Based on the similarity score, best response is outputted.	When the model learns the answer from two sentences, the Pattern Matching module will generate relevant phrases from both sentences. However, when combining them as an output, sentence quality will suffer.
BERT model with Pipeline	'question-answering' pipeline from Hugging Face is used. This pipeline is made up of a tokenizer, a model to make predictions from the inputs and post processing for enhancing models output.	Response time of the model is higher
Dynamic selection / Hybrid model	This approach combines the best of both the above models. SentenceTransformer is used to calculate cosine similarity between each question-answer pair from both models. The response with highest score is returned to the user.	Since both Baseline and pipeline models need to be executed, Response time to use further increase.

Table.6.8

Reference links for Baseline and Pipeline models included in Reference Section.

6.9 Testing and Model Evaluation

Our desired outcomes

In an ideal scenario, we wanted to deliver a model with:

- High Accuracy
- Low Response time
- High Scalability

Since BERT transformers were state-of-the-art, our solutions were built using BERT. Different BERT models were explored to deliver on the above objectives.

Scalability of the model

Scalability was important for the Sponsor since they foresee significant increase in volume and variety of queries with more ML Apps being uploaded on their portal. We therefore developed models which took simple ".txt" files as reference / context for the model. We demonstrated how their existing user manuals would suffice if converted to ".txt". This allayed any concerns on scalability.

Trade-off between Accuracy and Response time

While scalability was addressed, there was a trade-off between Accuracy and Response time. With increasing accuracy, response time also increased.

To check for accuracy, a test set of Questions & Answers was manually created. Model responses were compared against this to determine accuracy. ROUGE scores were used to check for the quality of the response sentence.

ROUGE is the abbreviation for Recall-Oriented Understudy for Gisting Evaluation. Because it works by comparing the automatically generated text with the human-generated reference text, it is now widely used by textual QA benchmarks especially by those with free-form reference answers. ROUGE is not a single score but a set of metrics, namely Recall, Precision and F1 score. We have evaluated ROUGE 1 (unigram) and ROUGE 2 (Bi-gram) scores across models.

Table 6.2

Assumption considered	Model or Metric Chosen
Accuracy of response takes precedence over response time	% of queries correctly responded to
Context file will be maintained in the desired quality	ROUGE scores

Table.6.9.1

ROUGE 1 (uni-gram) scores comparison across models

	Recall	Precision	F1 Score
Baseline	0.37	0.47	0.38
Pipeline	0.61	0.72	0.64
Hybrid	0.64	0.77	0.68

Table.6.9.2

ROUGE 2 (bi-gram) scores comparison across models

	Recall	Precision	F1 Score
Baseline	0.27	0.31	0.27
Pipeline	0.50	0.59	0.52
Hybrid	0.50	0.58	0.52

Table.6.9.3

Response time summary statistics across models:

	Baseline	Pipeline	Hybrid
mean	8.707267	23.155675	32.774993
std	2.481541	4.582789	6.496115
min	0.000842	0.000115	0.069677
25%	8.047797	22.971122	32.669934
50%	8.123450	23.707143	33.253658
75%	8.847082	24.792775	34.821488
max	16.845257	26.823984	37.969647

Table.6.9.4

Sponsors preferences

The models shown in table 6.1 were demonstrated to the Sponsor. Trade-off between model accuracy and response time were explained. Since the Sponsors preference was for a model with high accuracy, Hybrid model is recommended as the final model.

Hybrid / Dynamic model provide accurate response to 91% of user queries and further model responded within 33 seconds latency. In addition, ROUGE-1 and ROUGE-2 scores were better for Hybrid model.

6.10 Results of the models

Testing approach

To test the models, 55 questions were randomly generated based on the model manuals and general FAQs. From the context file, correct responses to these 55 questions were also generated. This is demonstarted in below table with Q1 being the question and A1 as the correct answer from the context file.

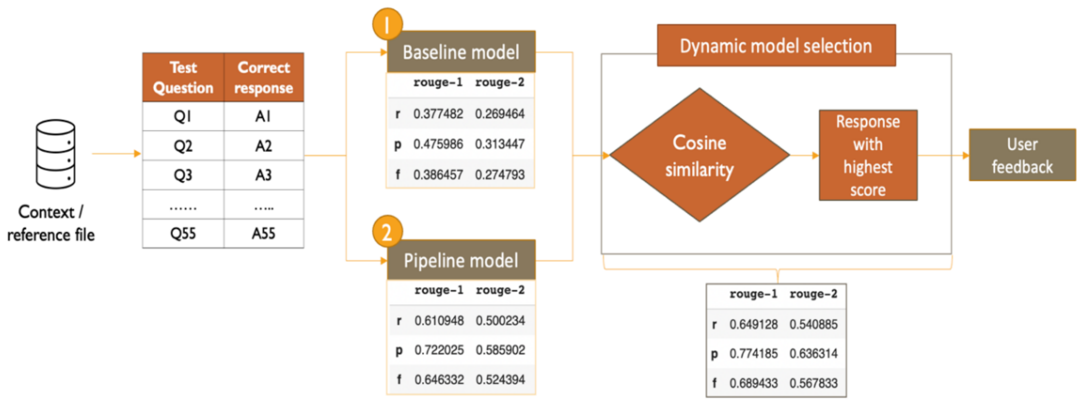


Fig.6.10

Snapshot of Questions & Correct answers (from context):

Question	Correct_Answer (from Context file)
where to upload data	The left panel has an option of Input data - Click on the Browse option and upload dataset in CSV format here.
How to remove missinig values	The left panel has Advanced Options section where there is option of Impute missing values or drop missing value rows - Click on the drop down to select the option of dropping or imputing the missing values.
what is a correlation matrix	A Correlation matrix is a table showing correlation coefficients between variables where each cell in the table shows the correlation between two variables.
what is k-nearest neighbors	k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.
how to set test sample percentage	The left panel has Advanced Options section where there is option to Set test sample percentage, Use the toggle bar to set required percentage of test sample.
What does regression summary tab contain	The regression summary tab provides the overall analysis result under a few methods like OLS (Ordinary Least Square), RMSE (Root Mean Square Error), Variance Inflation Factor (VIF) etc.
What is Ordinary Least Squares	Ordinary Least Squares regression or OLS is a common technique for estimating coefficients of linear regression equations which describe the relationship between one or more independent quantitative variables and a dependent variable.
what is RMSE	Root Mean Square Error or RMSE is the standard deviation of the residuals or it tells you how concentrated the data is around the line of best fit, Residuals or errors are a measure of how far from the regression line data points are.
How is variable importance determined	Variable importance is determined by calculating the relative influence of each variable: whether that variable was selected to split on during the tree building process, and how much the squared error (over all trees) improved (decreased) as a result.
What is Spectral clustering	Spectral clustering is a technique with roots in Graph theory, where the approach is used to identify communities of nodes in a graph based on the edges connecting them and the method is flexible and allows us to cluster non graph data as well.

Table.6.10.1

Sample list of Test Questions, their correct answers from context file, responses from both the models and manual evaluation of the model with best response is included as an annexure.

Best response evaluation

In our manual observation of the correct answer Vs model generated responses, we observed that pipeline model was more accurate. However, we observed that for a decent number of questions, Baseline model generated accurate responses compared to the Pipeline model.

		Pipeline model	
		Correct answer	Incorrect answer
Baseline model	Correct answer	17	9
	Incorrect answer	26	5

Table.6.10.2

- In isolation, the Baseline model is accurately responding to 44% of questions (24 / 55)
- The pipeline model is accurately responding to 75% of questions (41 / 55).
- By dynamic selection, we are getting the best of both models. 91% (50 / 55) of questions were accurately responded to.

6.11 Model Deployment and Evaluation

The model is developed in python. Simple ".txt" files were used as input files. This can be deployed in any environment the Sponsor chooses to integrate into. To the Sponsor, we demonstrated the model working through command prompt.

Sponsors technical team are currently evaluating their overall front-end needs. They will develop a front-end best suited to their overall needs and integrate the model into the same.

Additional requirements for deployment and production at client's side

Storage:

Store data in a location where model training and feedback from the user will be delivered. Cloud storage is typically utilised for cloud ML training and serving, although data can also be stored on-premises, in the cloud, or in a hybrid environment.

Deploying the ML Model:

- Create a python training environment in development
- Install all the dependencies/libraries mentioned in the requirements.txt

- Build the ML model and save it
- The Chatbot model is an online inference model or real-time inference model. So, it will have latency constraints that require best deployment environment
- Host the code in GitHub
- Prepare a container for deployment to deploy the code with a CI/CD pipeline using Jenkins
- Or connect GitHub repository to the deployment container either through StreamLit or Heroku
- Effective monitoring will be needed to understand which questions are not correctly answered by the Chatbot, which answers are the users not satisfied with and which answers need to be more elaborated etc.,

Chapter 7: Conclusion/Recommendations

7.1. Recommendations & Business Impact

The model was developed and tested on personal laptop with 16GB RAM, 1 TB hard-disk and i7 Processor. On these system specifications, average response time is ~33 seconds. On Sponsors servers, the response time would be much quicker.

File maintenance:

The model is designed to scale up for multiple models.

The context/reference source for the model is the text file (.txt) in which manual contents are maintained in the form of simple sentences. For any new model addition or standard user query, the simple text needs to be appended. If the input text file is drafted cleanly and passed, the model will provide responses.

While BERT extracts only relevant phrases from the context, to make it more readable for the user, the model is built to extract the complete sentence containing the relevant phrase. The model identifies sentence start and end by locating full stops.

It is therefore very important to keep the sentences crisp, and relevant when appending new text. Sentences must end with a full stop.

User Feedback loop:

The model has a built-in feedback loop. After giving response to a question, model takes user feedback on the response quality on a scale of 1 to 10. It is recommended to have periodic reviews on user feedback scores.

For queries with low feedback scores, the context file contents could be improved. FAQs could be identified and responses could be included in the context files.

References

JOURNAL/ NEWSPAPER/ MAGAZINE ARTICLES:

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova.(2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.arXiv:1810.04805 [cs.CL], 3-9.

Nazakat Ali. (2020). Chatbot: A Conversational Agent employed with Named Entity Recognition Model using Artificial Neural Network. arXiv:2007.04248 [cs.CL], 2-9.

Yu Wu, Wei Wu, Zhoujun Li, Ming Zhou.(2016). Response Selection with Topic Clues for Retrieval-based Chatbots. arXiv:1605.00090v3, 2-5.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazaré, Jason Weston.(2019). Learning from Dialogue after Deployment: Feed Yourself, Chatbot! arXiv:1901.05415v4 [cs.CL], 1-8.

Pierre Lison. (2012). Probabilistic Dialogue Models with Prior Domain Knowledge. Retrieved from https://aclanthology.org/W12-1625.pdf.

Jason D. Williams, Geoffrey Zweig.(2016). End-to-end LSTM-based dialogue control optimized with supervised and reinforcement learning. arXiv:1606.01269v1 [cs.CL], 2-8.

Lei Cui, Furu Wei, Shaohan Huang, Chuanqi Tan, Chaoqun Duan, Ming Zhou. (2017). SuperAgent: A Customer Service Chatbot for E-commerce Websites. Retrieved from https://www.microsoft.com/en-us/research/publication/superagent-customer-service-chatbot-e-commerce-websites/.

Aiswarya Surendran, Reena Murali, Remesh K. R. Babu. (2020). Conversational AI - A Retrieval-Based Chatbot. Retrieved from https://easychair.org/publications/preprint/LnTd.

Haritha Akkineni, Papineni v s Lakshmi. (2022). Design and Development of Retrieval-Based Chatbot Using Sentence Similarity. CC BY 4.0,4-15.

Regin R, Suman S, Shynu T, Jerusha A, Steffi R. (2022). Automated conversation system using natural language processing (NLP) chatbot in Python. Central asian journal of medical and natural sciences, 03(04), 1-23.

Satheesh k, Tamilselvan S, Ibrahim B, Harish S (2018). Artificial intelligence powered banking chatbot. IJESC, 08(03), 1-4

Eleni A, Lefteris M (2020). Chatbots: history, technology, and applications. Machine learning with applications, 02(100006), 1-18

Patel, N., Trivedi, S.(2020). Leveraging Predictive Modeling, Machine Learning Personalization And NLP Customer Support and Chatbots to Increase Customer Loyalty. EQMResearchBerg, 3(3), 1–24

Vijay k, Chinmay G, Yashvardhan S, Lavika Goel (2022). Domain-specific chatbot development using the deep learning-based rasa framework. Springer, 461, 1-14

HUGGING FACE REFERENCE LINKS:

https://huggingface.co/transformers/v3.0.2/model_doc/auto.html#automodelforpretraining

https://huggingface.co/docs/transformers/v4.24.0/en/main_classes/pipelines#transformers.QuestionAnsweringPipeline

BOOKS

Xiaoquan K, Guan W, Alan N. (2021). Conversational AI with Rasa: Build, test, and deploy AI-powered, enterprise-grade virtual assistants and chatbots. Uk: Packt Publishing

Srini Janarthanam. (2017). Hands-On Chatbots and Conversational UI Development. Mumbai. Packt.

Deepika M, Vijay C, Amitendra S, Srinivas M. (2019). AI and ML Powering the Agents of Automation. Mumbai. BPB Publications.

WEBPAGE OR PIECE OF ONLINE CONTENT

Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, Rama Akkiraju.(2017). A New Chatbot for Customer Service on Social Media. Retrieved from https://sci-hub.se/10.1145/3025453.3025496

Ghose, Supratip; Barua, Jagat Joyti.(2013). Toward the implementation of a topic specific dialogue based natural language chatbot as an undergraduate advisor. Retrieved from https://sci-hub.se/10.1109/iciev.2013.6572650

Aniruddha Karajgi. (2021, Sep 19). Building a Chatbot with Rasa. Towards Data Science. https://towardsdatascience.com/building-a-chatbot-with-rasa-3f03ecc5b324

Euchigere (2019, Oct 21). slackbot, Github. https://github.com/Euchigere/slackbot/blob/master/chatbot.ipynb

Karanjit, S (2021, May 3). Question and Answering Using BERT. Github. https://github.com/skaranjit/BertQA

Briggs, J (2021, Feb 20). Question And Answering With Bert. towardsdatascience.com .https://towardsdatascience.com/question-and-answering-with-bert-6ef89a78dac

Simple Transformers (May, 2021 2). Multi-Class Classification.

simpletransformers.com. https://simpletransformers.ai/docs/multi-class-classification/

Singh, V. (2021, Jun 24). Named Entity Recognition using Transformers. ner_transformers. colab.research.google.com. https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/nlp/ipynb/ner_transformers.ipynb#scrollTo=U931XIQptMfu

Calam, J (2021, May 28). jamescalam / transformers, Github. https://github.com/jamescalam/transformers/blob/main/course/project_meditations_odqa/00_setup_elasticsearch.ipynb

ANNEXURE 1

Sample Testing Questions, Correct responses (from Context file), model responses and manual validation of the model with best response

Question	Model_Answer_Baseline	Model_Answer_Pipeline	Best respon	Correct_Answer (from Context file)
what is a correlation matrix	a pair plot gives pairwise relationships in a dataset to understand the best set of features to explain a relationship between two variables or to form the most separated clusters the pair plot function creates a grid of axes such that each variable in data will be shared in the y axis across a single row and on the x axis across a single column.	The Summary OLS tab evaluates the correlation coefficients between variables and represents them through a correlation map where each cell depicts a correlation between 2 variables and the size and color of the circles in each cell depict the degree of correlation the larger the size and darker the color shade the higher is the correlation.	Pipeline	A Correlation matrix is a table showing correlation coefficients between variables where each cell in the table shows the correlation between two variables.
what is KNN	k nearest neighbors algorithm also known as knn or k nn is a non parametric supervised learning classifier which uses proximity to make classifications or predictions about the grouping of an individual data point.	k nearest neighbors algorithm also known as KNN or k NN is a non parametric supervised learning classifier which uses proximity to make classifications or predictions about the grouping of an individual data point.	same response	k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.
what is k-nearest neighbors	Unable to find the answer to your question.	k nearest neighbors algorithm also known as KNN or k NN is a non parametric supervised learning classifier which uses proximity to make classifications or predictions about the grouping of an individual data point.	Pipeline	k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.
how to set test sample percentage	the left panel has advanced options section where there is option of select sub sample click on the drop down option to select random number of observations for testing or the whole data itself.	The left panel has Advanced Options section where there is option to Set test sample percentage Use the toggle bar to set required percentage of test sample.	Pipeline	The left panel has Advanced Options section where there is option to Set test sample percentage, Use the toggle bar to set required percentage of test sample.
how to set train test sample percentage	Unable to find the answer to your question.	A Correlation Table is a two way tabulation of the relations between correlates row headings are the scores on one variable and column headings are the scores on the second variable and a cell shows how many times the score on that row was associated with the score in that column.	check	The left panel has Advanced Options section where there is option to Set test sample percentage, Use the toggle bar to set required percentage of test sample.
How to select maximum nearest neighbours	the left panel has advanced options section where there is option to select maximum nearest neighbours use the toggle bar to set required number of nearest neighbours.	The left panel has Advanced Options section where there is option to Select maximum nearest neighbours Use the toggle bar to set required number of nearest neighbours.	same response	The left panel has Advanced Options section where there is option to Select maximum nearest neighbours, Use the toggle bar to set required number of nearest neighbours.
How to set maximum nearest neighbours	the left panel has advanced options section where there is option to select maximum nearest neighbours use the toggle bar to set required number of nearest neighbours.	The left panel has Advanced Options section where there is option to Select maximum nearest neighbours Use the toggle bar to set required number of nearest neighbours.	same response	The left panel has Advanced Options section where there is option to Select maximum nearest neighbours, Use the toggle bar to set required number of nearest neighbours.
How to set CV folds	the left side of the screen left panel has an option of input data click on the browse option and upload dataset in csv format here.	The left panel has Advanced Options section where there is option to Select number of CV folds Use the toggle bar to set required number of Cross Validation CV folds.	Pipeline	The left panel has Advanced Options section where there is option to Select number of CV folds, Use the toggle bar to set required number of Cross-Validation CV folds.