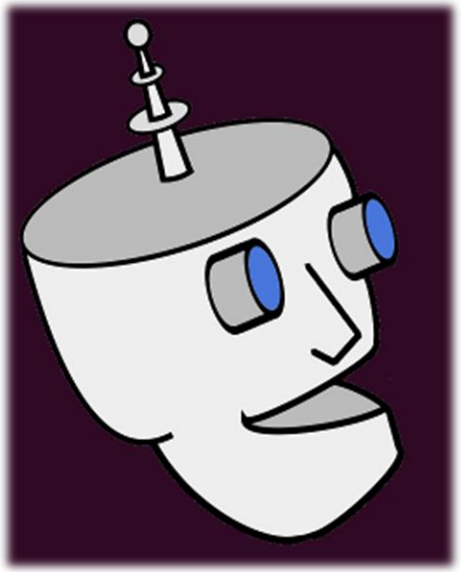
Project Title Slide

Intent Classification Chatbot



Introduction to Project

Chatbot is widely adapted by various companies to enhance their service for customers, internal and external stakeholder and becomes trending these days.

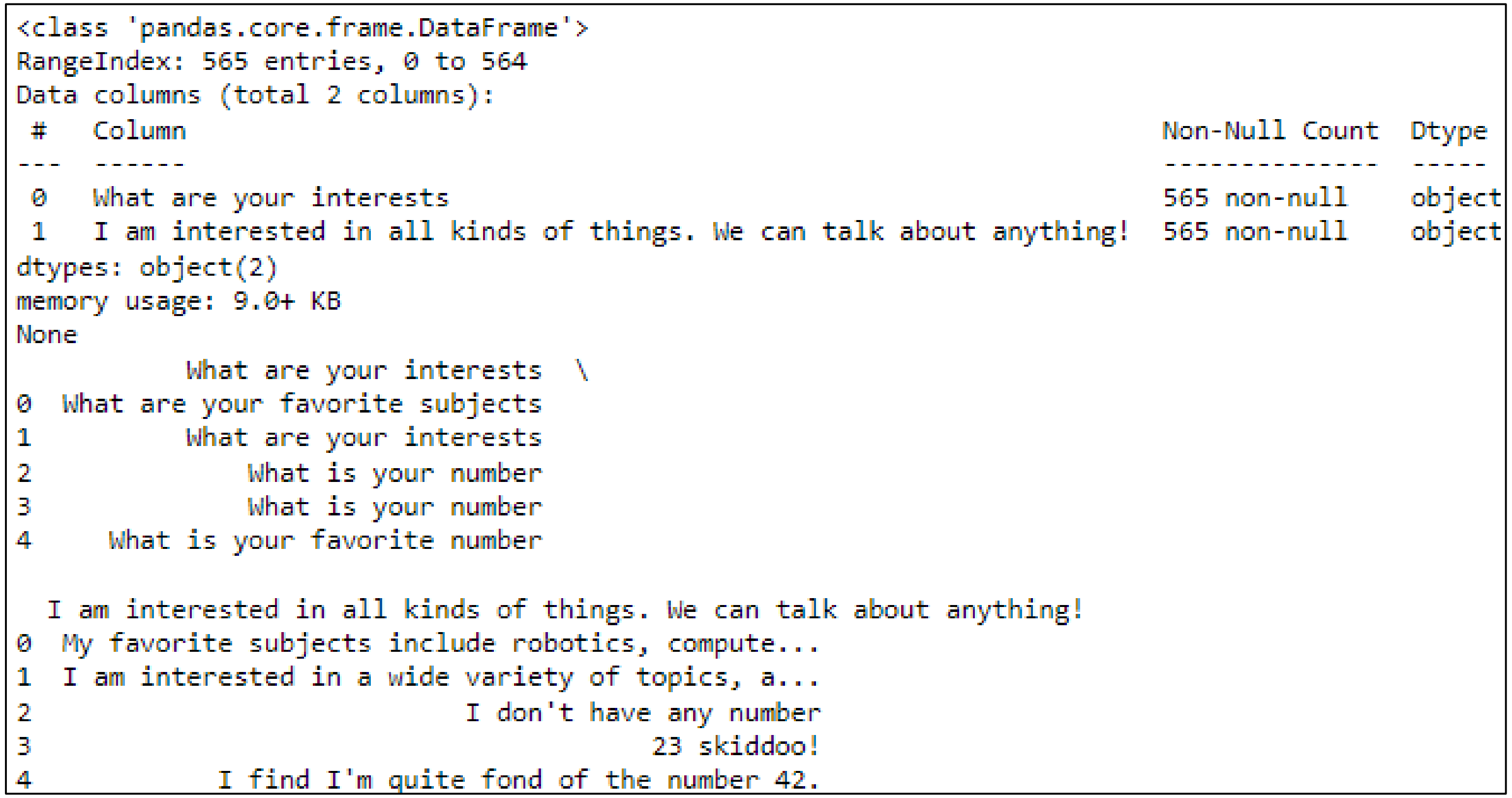
This project aim to explore the performance of apply context-dependent deep learning pre-trained language model, which is **ELMo (Embedding from Language Models)**, as its primarily functions is as an encoder in the context of language models. ELMo’s strengths is **generating different embedding representations(output numeric word vectors for different sentences)** to take into account context, leading to improved performance on a range of natural language processing tasks.

We apply ELMo as our embedding method on a casual dialog dataset from Kaggle and train **four machine learning models (SVM, Logistic Regression, Random Forest, and Gradient Boosting)** to leverage the performance.

At last, we try to create an interaction conversation model by Chatterbot and output intent prediction in a Jupyter Notebook file.

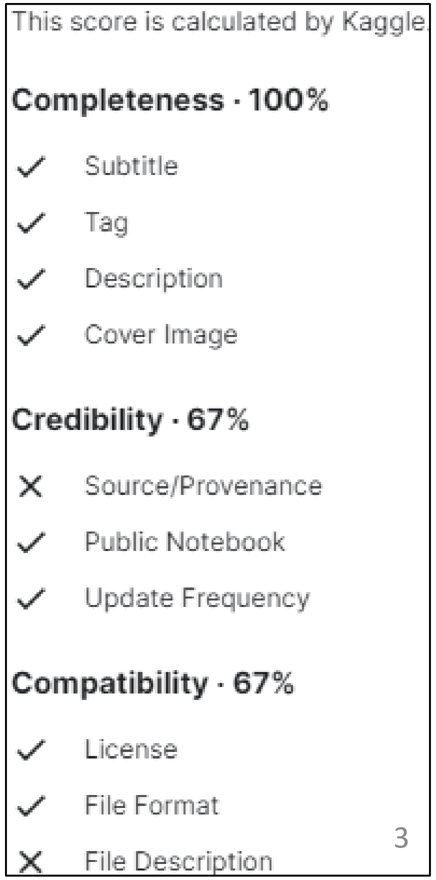
Dataset Description

Source: https://www.kaggle.com/datasets/saurabhprajapat/chatbot-training-dataset/data
Owner: Saurabh Prajapat
Subtile: Chatbot Training Dataset
Dialogs: 565 lines
Dataset Type and Size: .txt file, 49.46 kB
License: CC0: Public Domain (No Copyright, can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission)
Tags: No tags



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 565 entries, 0 to 564
Data columns (total 2 columns):
 #   Column                                                     Non-Null Count  Dtype
---  ------                                                     --------------  -----
 0   What are your interests                                    565 non-null    object
 1   I am interested in all kinds of things. We can talk about anything!  565 non-null    object
dtypes: object(2)
memory usage: 9.0+ KB
None
           What are your interests  \
0  What are your favorite subjects
1          What are your interests
2              What is your number
3              What is your number
4     What is your favorite number

  I am interested in all kinds of things. We can talk about anything!
0  My favorite subjects include robotics, compute...
1  I am interested in a wide variety of topics, a...
2                          I don't have any number
3                                      23 skiddoo!
4            I find I'm quite fond of the number 42.
```

Usability detail on Kaggle page:



This score is calculated by Kaggle.

Completeness · 100%

✓ Subtitle
✓ Tag
✓ Description
✓ Cover Image

Credibility · 67%

✕ Source/Provenance
✓ Public Notebook
✓ Update Frequency

Compatibility · 67%

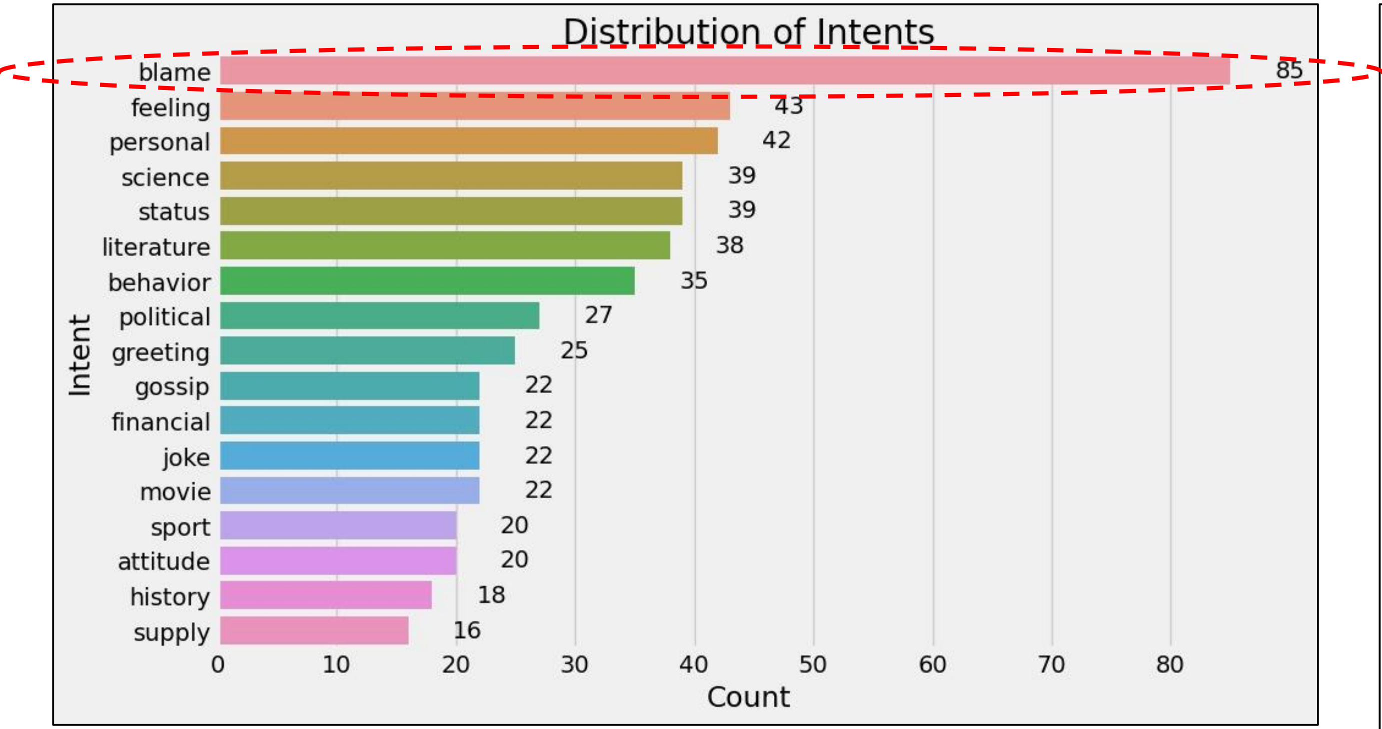
✓ License
✓ File Format
✕ File Description

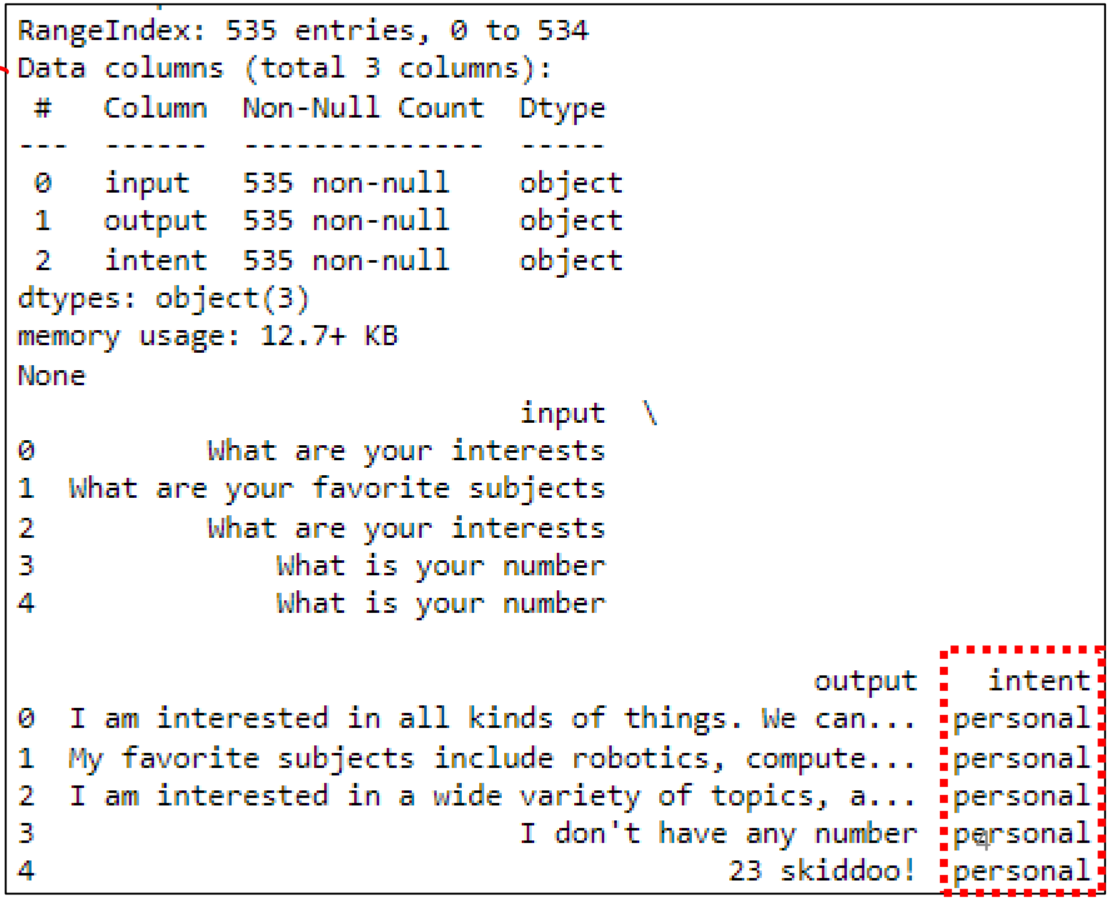
Dataset Preprocessing-1

1) **Manually annotated 535**/565 **lines** and removed 30/565 incompletion/unclear lines.

2) Applied regularization:

   - **Lower case**: no significant patterns for caps or lower case in original dataset
   - **Punctuation, special characters and underline**: no significant patterns in original dataset

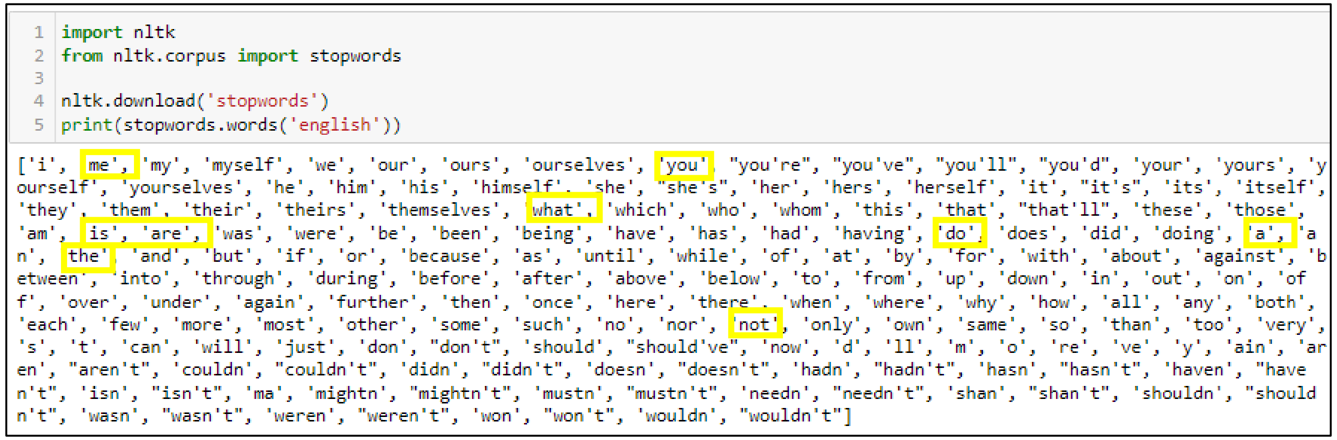




```
RangeIndex: 535 entries, 0 to 534
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   input   535 non-null    object
 1   output  535 non-null    object
 2   intent  535 non-null    object
dtypes: object(3)
memory usage: 12.7+ KB
None
                          input  \
0           What are your interests
1    What are your favorite subjects
2           What are your interests
3              What is your number
4              What is your number

                                         output    intent
0  I am interested in all kinds of things. We can...  personal
1  My favorite subjects include robotics, compute...  personal
2  I am interested in a wide variety of topics, a...  personal
3                            I don't have any number  personal
4                                      23 skiddoo!  personal
```

Dataset Preprocessing-2

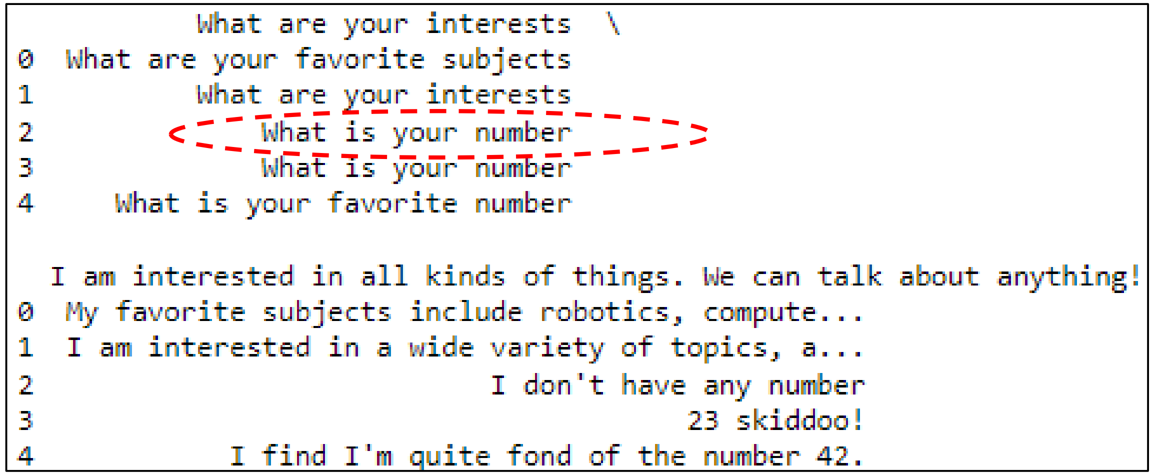
3. Didn't remove **stopwords** because:
   - The dataset is small.
   - Remain the original lines as possible for better sentence meaning in NLU because we are going to use sentence embedding technique.
   - Both input and output are not long and complex sentence.



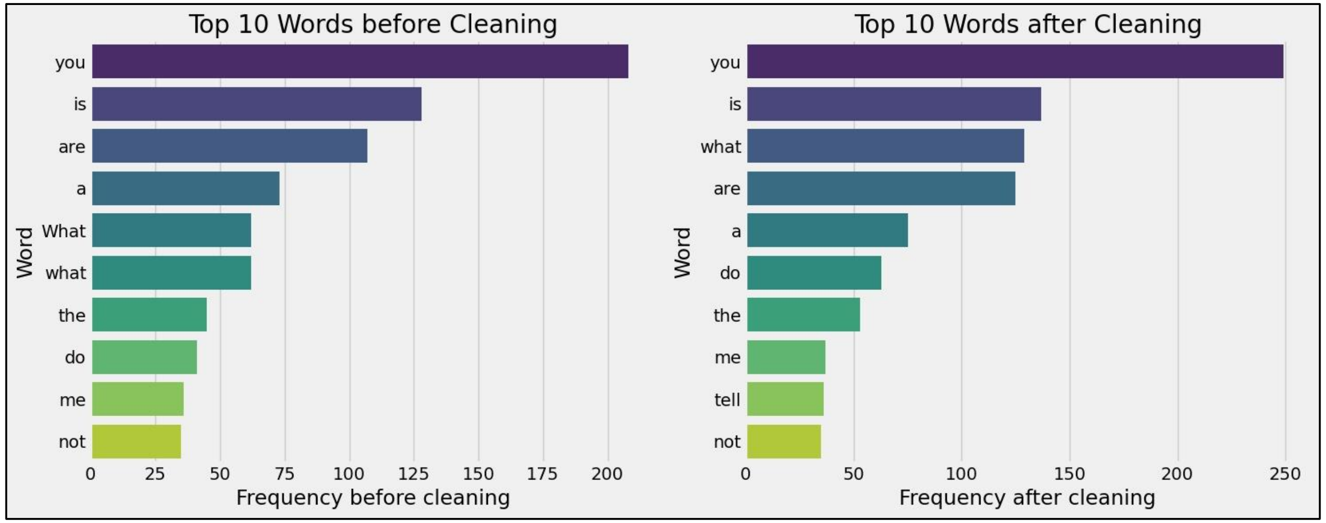
```
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'y
ourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself',
'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those',
'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'a
n', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'b
etween', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'of
f', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very',
's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'ar
en', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "have
n't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "should
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

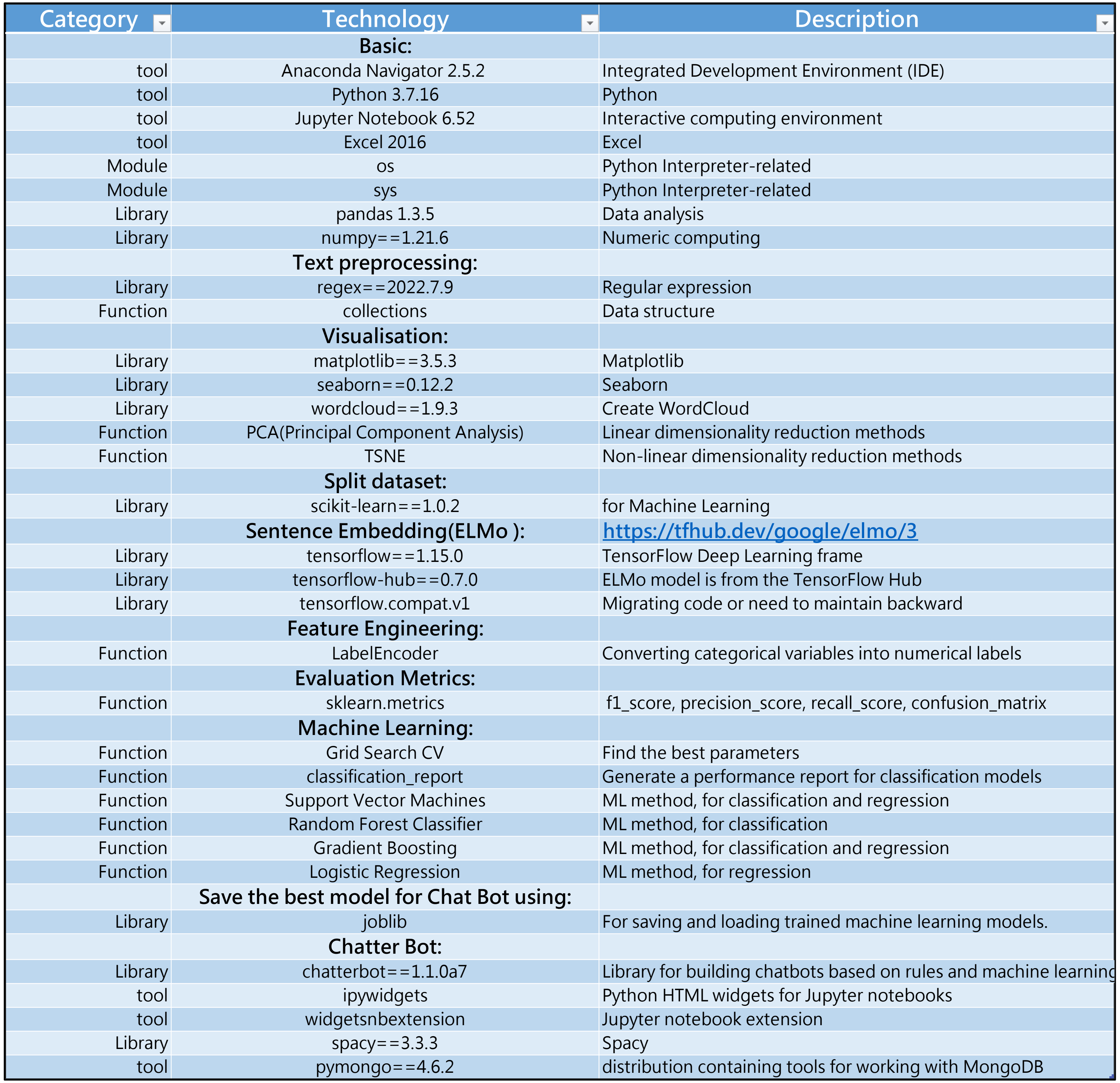
you, is, are, a,
what, the, do,
me, not



```
              What are your interests  \
0  What are your favorite subjects
1              What are your interests
2                  What is your number
3                  What is your number
4         What is your favorite number

  I am interested in all kinds of things. We can talk about anything!
0  My favorite subjects include robotics, compute...
1  I am interested in a wide variety of topics, a...
2                           I don't have any number
3                                        23 skiddoo!
4           I find I'm quite fond of the number 42.
```

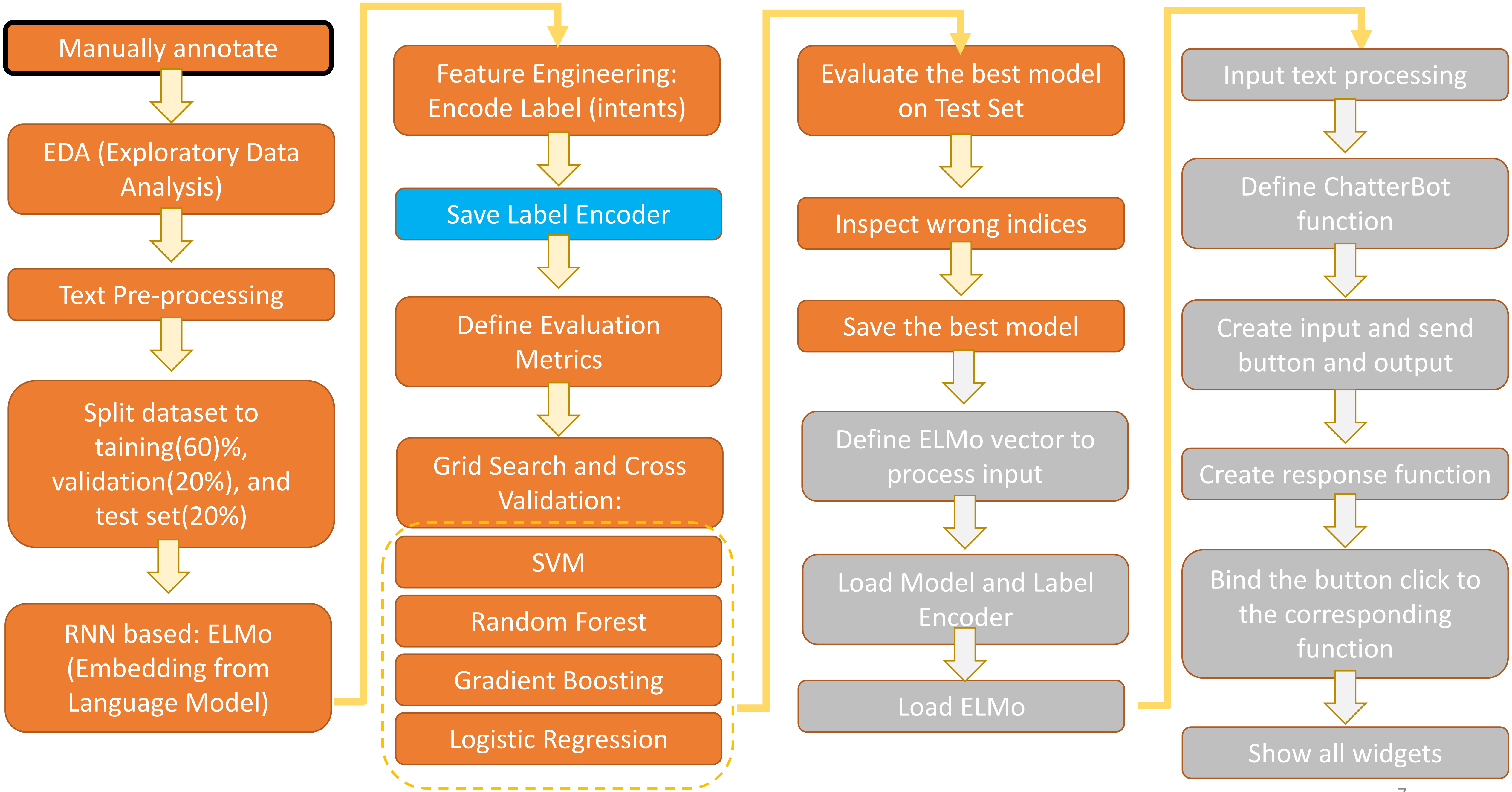


Technology Stack Used



Category	Technology	Description
	Basic:	
tool	Anaconda Navigator 2.5.2	Integrated Development Environment (IDE)
tool	Python 3.7.16	Python
tool	Jupyter Notebook 6.52	Interactive computing environment
tool	Excel 2016	Excel
Module	os	Python Interpreter-related
Module	sys	Python Interpreter-related
Library	pandas 1.3.5	Data analysis
Library	numpy==1.21.6	Numeric computing
	Text preprocessing:	
Library	regex==2022.7.9	Regular expression
Function	collections	Data structure
	Visualisation:	
Library	matplotlib==3.5.3	Matplotlib
Library	seaborn==0.12.2	Seaborn
Library	wordcloud==1.9.3	Create WordCloud
Function	PCA(Principal Component Analysis)	Linear dimensionality reduction methods
Function	TSNE	Non-linear dimensionality reduction methods
	Split dataset:	
Library	scikit-learn==1.0.2	for Machine Learning
	Sentence Embedding(ELMo):	https://tfhub.dev/google/elmo/3
Library	tensorflow==1.15.0	TensorFlow Deep Learning frame
Library	tensorflow-hub==0.7.0	ELMo model is from the TensorFlow Hub
Library	tensorflow.compat.v1	Migrating code or need to maintain backward
	Feature Engineering:	
Function	LabelEncoder	Converting categorical variables into numerical labels
	Evaluation Metrics:	
Function	sklearn.metrics	f1_score, precision_score, recall_score, confusion_matrix
	Machine Learning:	
Function	Grid Search CV	Find the best parameters
Function	classification_report	Generate a performance report for classification models
Function	Support Vector Machines	ML method, for classification and regression
Function	Random Forest Classifier	ML method, for classification
Function	Gradient Boosting	ML method, for classification and regression
Function	Logistic Regression	ML method, for regression
	Save the best model for Chat Bot using:	
Library	joblib	For saving and loading trained machine learning models.
	Chatter Bot:	
Library	chatterbot==1.1.0a7	Library for building chatbots based on rules and machine learning
tool	ipywidgets	Python HTML widgets for Jupyter notebooks
tool	widgetsnbextension	Jupyter notebook extension
Library	spacy==3.3.3	Spacy
tool	pymongo==4.6.2	distribution containing tools for working with MongoDB

Architectural flow of the NLP Tool Developed



Manually annotate

EDA (Exploratory Data Analysis)

Text Pre-processing

Split dataset to taining(60)%, validation(20%), and test set(20%)

RNN based: ELMo (Embedding from Language Model)

Feature Engineering: Encode Label (intents)

Save Label Encoder

Define Evaluation Metrics

Grid Search and Cross Validation:

SVM

Random Forest

Gradient Boosting

Logistic Regression

Evaluate the best model on Test Set

Inspect wrong indices

Save the best model

Define ELMo vector to process input

Load Model and Label Encoder

Load ELMo

Input text processing

Define ChatterBot function

Create input and send button and output

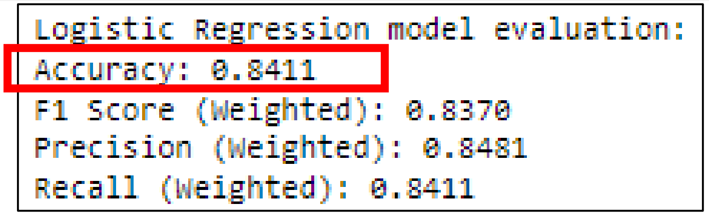
Create response function

Bind the button click to the corresponding function

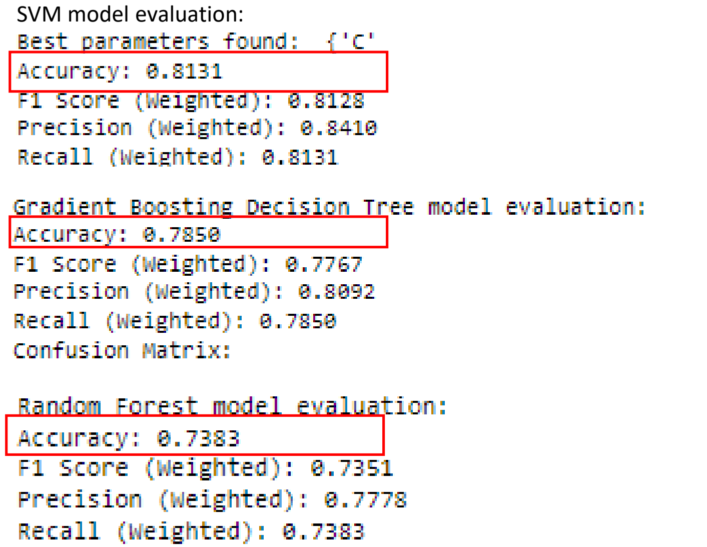
Show all widgets

Results and Output

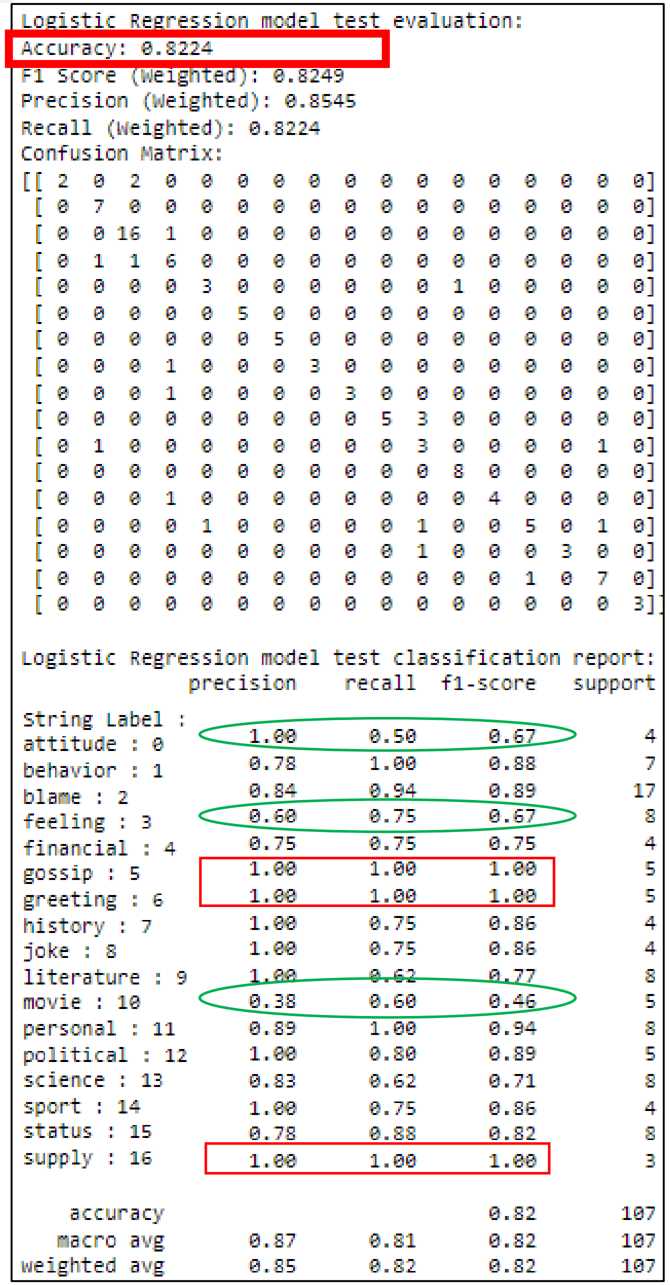
Logistic Regression performed best:



```
Logistic Regression model evaluation:
Accuracy: 0.8411
F1 Score (Weighted): 0.8370
Precision (Weighted): 0.8481
Recall (Weighted): 0.8411
```

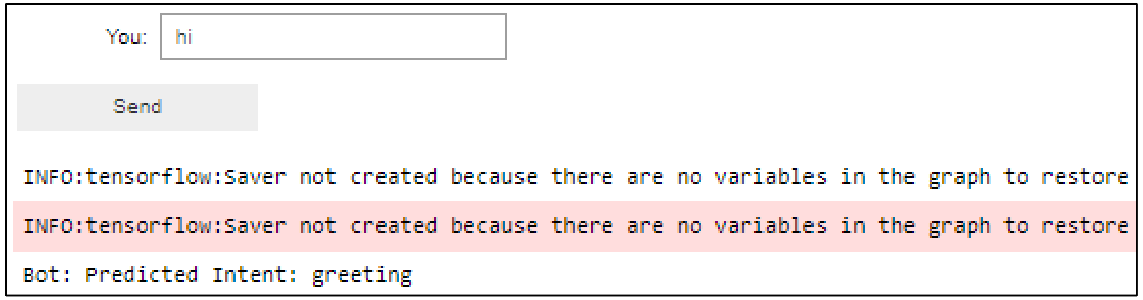


```
SVM model evaluation:
Best parameters found:  {'C'
Accuracy: 0.8131
F1 Score (Weighted): 0.8128
Precision (Weighted): 0.8410
Recall (Weighted): 0.8131

Gradient Boosting Decision Tree model evaluation:
Accuracy: 0.7850
F1 Score (Weighted): 0.7767
Precision (Weighted): 0.8092
Recall (Weighted): 0.7850
Confusion Matrix:

Random Forest model evaluation:
Accuracy: 0.7383
F1 Score (Weighted): 0.7351
Precision (Weighted): 0.7778
Recall (Weighted): 0.7383
```



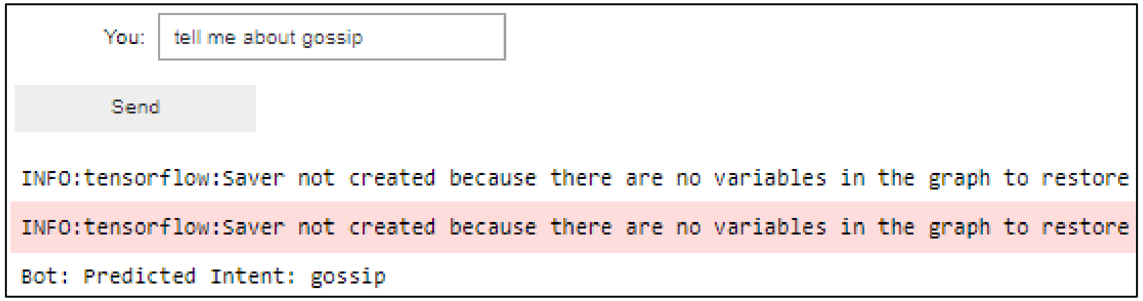
```
Logistic Regression model test evaluation:
Accuracy: 0.8224
F1 Score (Weighted): 0.8249
Precision (Weighted): 0.8545
Recall (Weighted): 0.8224
Confusion Matrix:
[[ 2  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 16  1  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  1  1  6  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  1  0  0  0  0  0]
 [ 0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  3  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  3  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  5  3  0  0  0  0  0  0]
 [ 0  1  0  0  0  0  0  0  0  0  3  0  0  0  0  1  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  8  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  4  0  0  0  0]
 [ 0  0  0  0  1  0  0  0  0  0  1  0  0  5  0  1  0]
 [ 0  0  0  0  0  0  0  0  0  0  1  0  0  0  3  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  7  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3]]

Logistic Regression model test classification report:
                   precision    recall  f1-score   support

String Label :
attitude : 0          1.00      0.50      0.67         4
behavior : 1          0.78      1.00      0.88         7
blame : 2             0.84      0.94      0.89        17
feeling : 3           0.60      0.75      0.67         8
financial : 4         0.75      0.75      0.75         4
gossip : 5            1.00      1.00      1.00         5
greeting : 6          1.00      1.00      1.00         5
history : 7           1.00      0.75      0.86         4
joke : 8              1.00      0.75      0.86         4
literature : 9        1.00      0.62      0.77         8
movie : 10            0.38      0.60      0.46         5
personal : 11         0.89      1.00      0.94         8
political : 12        1.00      0.80      0.89         5
science : 13          0.83      0.62      0.71         8
sport : 14            1.00      0.75      0.86         4
status : 15           0.78      0.88      0.82         8
supply : 16           1.00      1.00      1.00         3

      accuracy                          0.82       107
     macro avg        0.87      0.81    0.82       107
  weighted avg        0.85      0.82    0.82       107
```

Chatbot prediction and response:



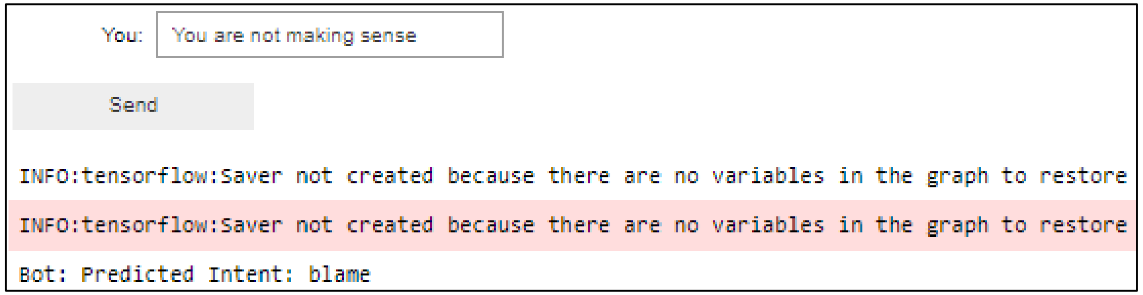
You: hi

```
INFO:tensorflow:Saver not created because there are no variables in the graph to restore
INFO:tensorflow:Saver not created because there are no variables in the graph to restore
Bot: Predicted Intent: greeting
```



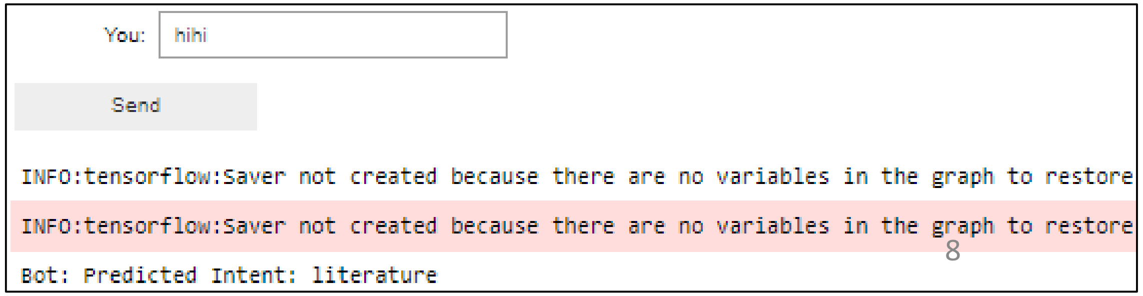
You: tell me about gossip

```
INFO:tensorflow:Saver not created because there are no variables in the graph to restore
INFO:tensorflow:Saver not created because there are no variables in the graph to restore
Bot: Predicted Intent: gossip
```



You: You are not making sense

```
INFO:tensorflow:Saver not created because there are no variables in the graph to restore
INFO:tensorflow:Saver not created because there are no variables in the graph to restore
Bot: Predicted Intent: blame
```



You: hihi

```
INFO:tensorflow:Saver not created because there are no variables in the graph to restore
INFO:tensorflow:Saver not created because there are no variables in the graph to restore
Bot: Predicted Intent: literature
```

Key Learnings

1) Check the support version for our target library. **Chatterbot** only **supports Python version** between **3.4** to **3.7.9**.
2) For a dialog task, sentence embedding could have a better representation because it consider the meaning of whole sentences.
3) When we use **ELMo (sentence embedding, contextual word embedding)** to our task,

   we usually **don't** have to consider to:
   - Convert text labels to numeric types in advanced (such as **TF-IDF, Co-occurrence matrix**).
   - Word Sense Disambiguation (**WSD**).
   - Manual **POS tagging**, **NER**, or **lemmatization**(**could try**).
   - Perform Top-Down **Parsing** and Bottom-Up Parsing.

   Should consider:
   - **Pragmatics**. Because ELMo doesn’t directly relate to some specific needs that do not areas such as conversation analytics or customer service bots .
   - **Anaphora and Coreference**. Could be benefit under some complex situation.
   - **Semantic Analysis**. For highly specialized semantic analysis tasks (finance, medical, etc.).
4) **Category imbalance handling.** Could consider to use **SMOTE**(Synthetic Minority Over-sampling Technique) to increasing the number of cases in the dataset in a balanced way.
5) Could **separately process “inputs” and “outputs” text**. Reduce confusion between what constitutes an input and an output during training
6) Could compare the performance with **GloVe**-Non-Contextual (Static) Word Embeddings.
7) Could try to use Flask and deploy the Chatbot to Heroku.

References

SMOTE for Imbalanced Classification with Python: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
Chatterbot help: https://chatterbot.readthedocs.io/en/stable/tutorial.html#getting-help
Chatterbot: https://www.kaggle.com/code/aishasana/chatterbot
My Chatbot with chatterbot: https://www.kaggle.com/code/aaroha33/my-chatbot-with-chatterbot
Chatbot_Starter_with_NLTK: https://www.kaggle.com/code/santoshroy1/chatbot-starter-with-nltk
GrapeNLP grammar engine in a Kaggle notebook: https://www.kaggle.com/code/javiersastre/grapenlp-grammar-engine-in-a-kaggle-notebook
Chatbot With python: https://www.kaggle.com/code/noorsaeed/chatbot-with-python
ChatGPT 4.0