**AASKKK ---- Text-based chatbot**

**(Ashwani Ayush Saswat Karnati Karthikeyan Kshitish)**

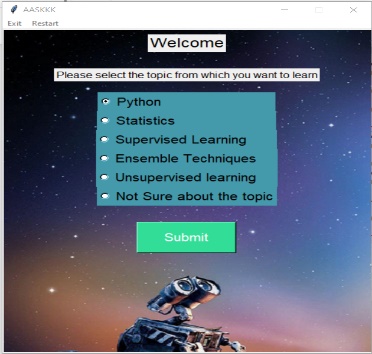
**How to Run AASKKK:** Please launch "**AASKKK\_GUI\_Integrated.ipynb**" file to run the GUI integrated text based chatbot.

\*\*NLP was not taught as part of curriculum we did our research to understand the concept related to NLP and incorporated in this Chatbot

**AASKKK Overview:**

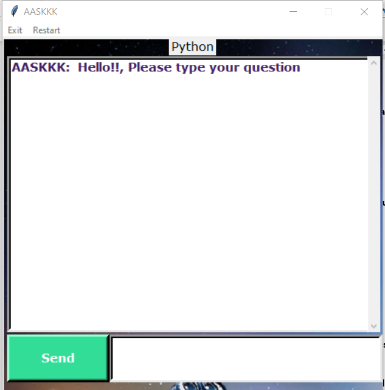
* **CORPUS** – Develop a corpus on machine learning related content – which is used to train the model

There is a lot of effort we put in to come up with this innovative corpus which represents our entire curriculum in around 600 questions.

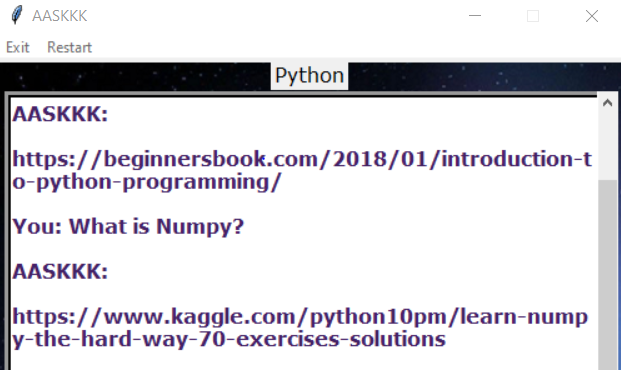
****

**GUI**

* **Landing page - Topic selection page 🡪**

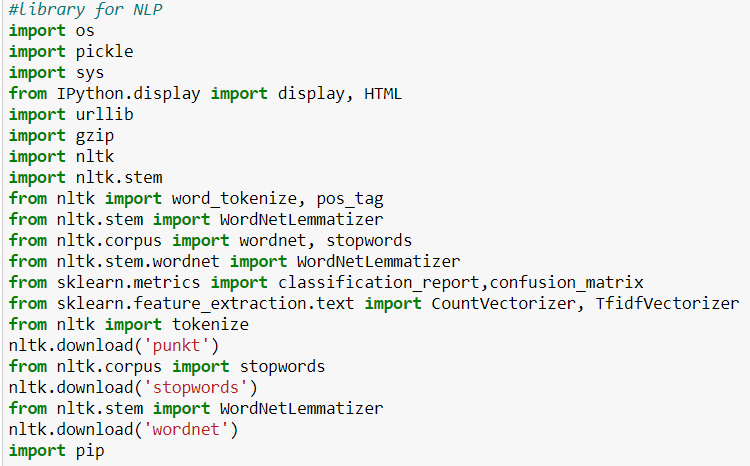
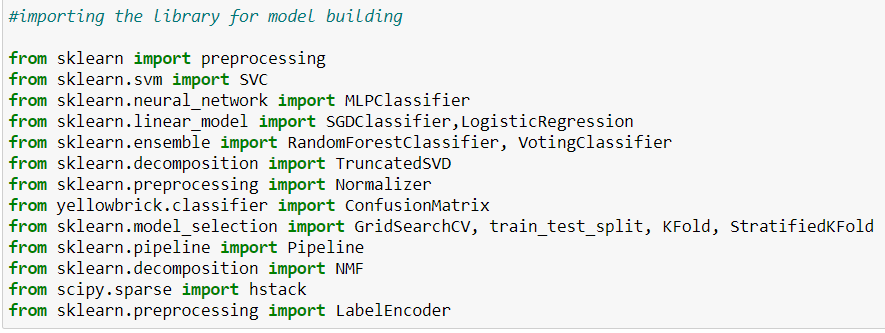


* **AASKKK question page🡪**



**Modeling:**

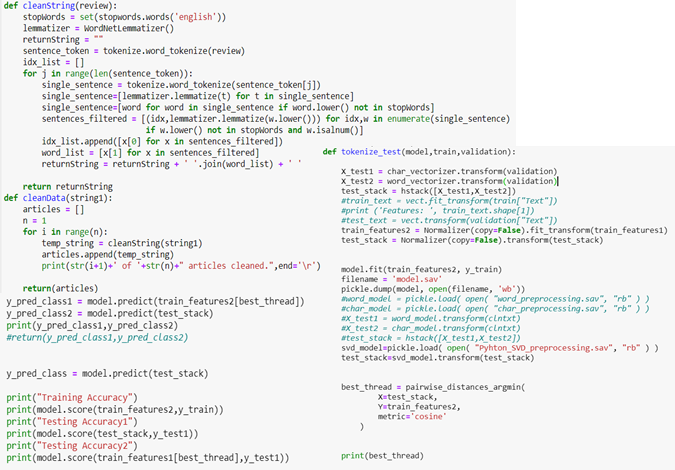
* **Importing the necessary libraries**



--- Basic libraries

--- Importing the library for model building

--- Library for nlp



* **Preprocessing of data**

Cleanstring() -- function clean the data both incoming and our train data.

Stopwords() -- used here to remove the redundant word from our sentences.

Lematizer() -- used here to remove the verb, adjective form in order to provide the required word from the sentence. All the sentences are converted to one similar format(here converted to lower).

* **Feature engineering**

****Truncatedsvd used here to reduce the dimension in order to get best performance of the model.(n\_components =200,200 dimensions)

Character vectorizer and word vectorizer both are used to get the vector form the tokenized word. We can use any vectorizer but here we used both for better performance of our model.

'Pickle' is used to load our model.

Pairwise distance is used here based on cosine similarity in order to find out the match between word. It also helps model to find the context which will help model to find the proper response for a required question.

Normalizer is also used here to keep our data under one scale.

* **Dimensionality reduction**

1)Truncatedsvd used here to reduce the dimension in order to get best performance of the model.(n\_components=200,200 dimensions)

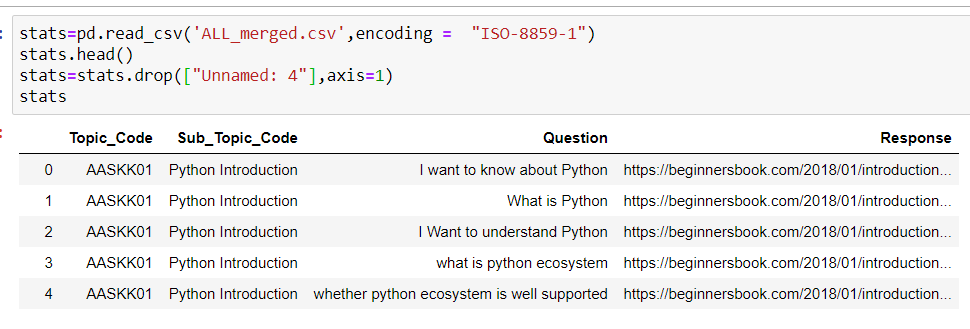
2) dimensions of the character vectorizer is set at 20k so that all possible combinations of characters are captured by the character model.

3) dimension of the word vectorizer is set at 3k so that all possible combinations of words are captured by the word model.

4) we then create a context definition by applying tfidf on the stemmed words.

5) we apply stemming twice to reduce the dimensions of the models and our model fits in 25mb if we apply stemming twice once through the tfidf and another through a script.

6) we took enough care to have minimal dimensions by reprojection the higher dimension to 500x500 dimension using principal component analysis.(svd)

* **Importing our main corpus**

**Data preparation for model building**

1.Feature encoding is carried out to change the sub\_topic\_code into respective label.

2. Labelencoder.pickle is used to load it into model.

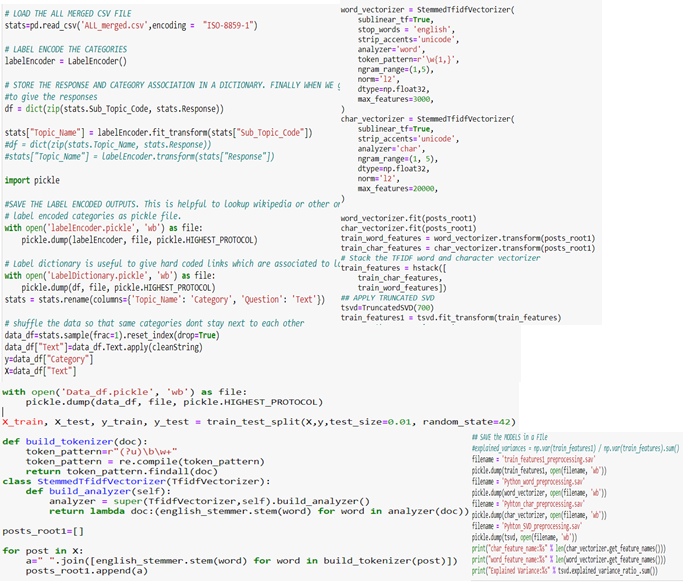
3. Sampling of data also done in order to get all the sub\_topic in equal proportion.

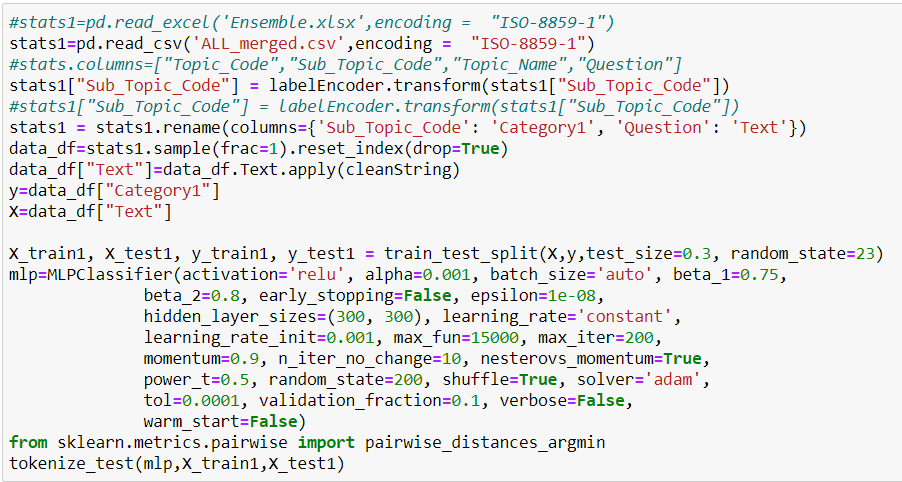
4. Dataset is split into train set and test set.

5. Stemmedtfidfvectorizer is used here to convert word or character to its respective vector.it is also an advanced version of tfidfvectorizer.

6. We are used both word\_vectorizer and character\_vectorizer for better performance.

**Load the all merged csv file**

* Label encode the categories
* Store the response and category association in a dictionary. Finally, when we give the results, we lookup the dictionary to give the responses
* Save the label encoded outputs. This is helpful to lookup Wikipedia or other online resources for links once we have
* Label encoded categories as pickle file.
* Label dictionary is useful to give hard coded links which are associated to label encoded outputs
* Shuffle the data so that same categories don’t stay next to each other
* Test train split the data with 1% validation data and rest is used to build the model.
* Special tokenizer function integrated with tfidfvectorizer. This along with stemmer is incorporated as a function
* Special stemmer based tfidf which is unique to our code. We have incorporated stemming along with tfidf to reduce the dimensions.
* So each incoming word is stemmed twice to reduce the dimension.
* This is an overlay method which works along with the built in tfidf vectorizer method
* Apply word vectorizer.
* Stop words are used as arguments
* Word analyzer with token pattern is used.
* we have used ngram of ( 1,5 ) up to 5 Word groupings can be found in our database.
* We return l2 norm outputs so that output stays sparse after tfidf.
* Maximum features considered are 3k only
* This returns 3k dimensions after tfidf transformation
* Character stemmer
* Apply character vecotirzer.
* Character analyzer with token pattern is used.
* We have used ngram of ( 1,5 ) up to 5-character groupings can be found in our database.
* We return l2 norm outputs so that output stays sparse after tfidf.
* Maximum features considered are 20k only.
* This returns 20k dimensions after tfidf transformation
* Fit and transform the tfidf word and character vectorizer.
* Stack the tfidf word and character vectorizer
* Apply truncated svd
* Save the models in a file



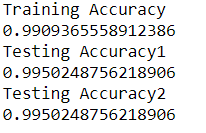
* **Model Performance :**

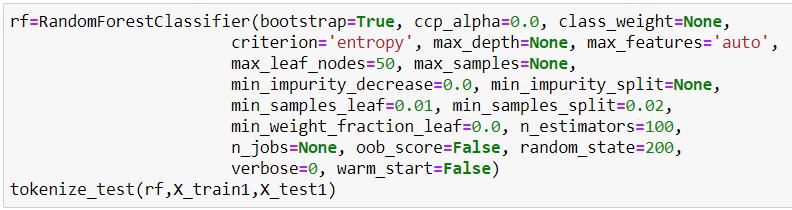
1. **With Neural Network Classifier**

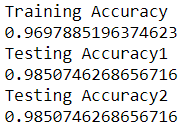
printing the predicted class

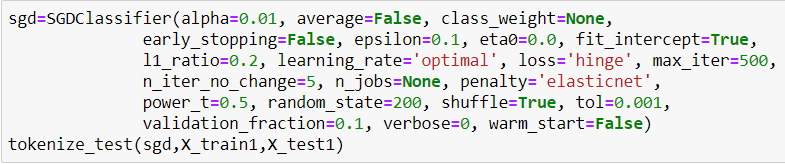
y\_test1.head()

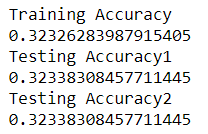
y

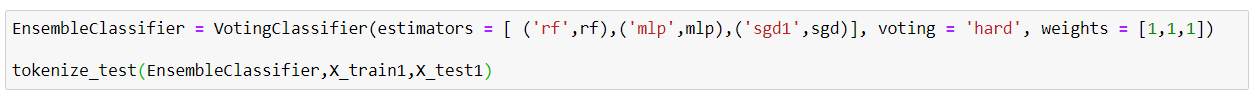


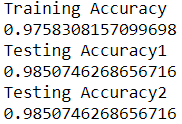
1. **With Regularized Random Forest**

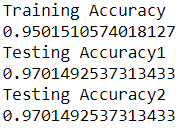


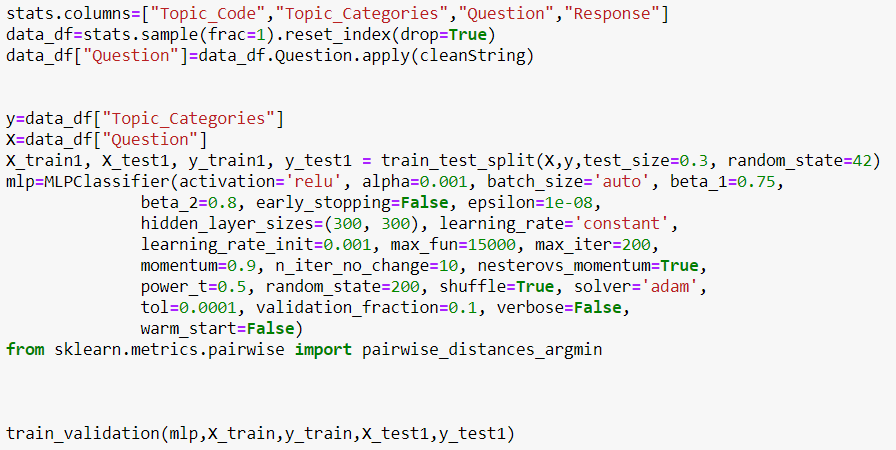
1. **Stochastic gradient descent (SGD) Classifier**



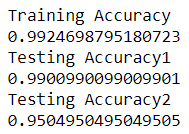


1. **With Voting Classifier**
2. **With Multinomial Naive Bayes**





1. **With MLP Classifier**



**CONCLUSION:**

* We would like to thank Krishna Dave for the idea and teaching us the direction
* We would like to thank Arti Nair for this Capstone assignment
* We would like to thank Prof. Rao in explaining concepts related to SVD, Text context analysis
* Thank you all for the support and learning