CLASSIFICATION OF QUERY COMING FROM MULTILINGUAL DATA SET

October 19, 2020

Name: Tasnim Ferdous Dima

0.1 GOAL

Given Data set which contains queries in three different languages English, Bangla and Romanized Bangla the job is to make sure to read a new query and correctly map to their their label/ Intent.

0.2 DATA PRE PROCESSING

0.2.1 Read data and assigned Labels

The data set is contains in a zip file named dataset.zip. there are six(6) CSV files. Each CSV files contains 1 column and each row contain a query. No label is given in the csv files. I assumed each CSV files names is the corresponding label/Intent. and there are 6 CSV files. I give a numerical label to each row in the CSV files. The files names and my given corresponding label names are:

- annual_fee.csv to label no = 1
- eligibility.csv to label no = 2
- facilities.csv to label no = 3
- interest-rate.csv to label no = 4
- mobile-recharge.csv to label no = 5
- required-documents.csv to label no = 6

I also add header and indexing to the files.the header 'sentence' represent a query and 'label' represents what the query is about.After loading all the data I make one large big dataset that contains all the data loaded from these six files. Now the combined data of all the files is relatively small . So it effect on the memory complexity. If the datas were large in shape then there's no need of creating an entire dataframe. I will just loop over all the dataset and did the work that I did. But as the data is small and I prefer to keep code as readable as possible I make an entire dataset combining all the sentences with their corresponding label and header and worked on that dataset. I also shuffled the dataset to the datas of different labels distributes evenly. This large dataset is called "df" and the number of row is 5842 and 3

```
df.head(12)
```

	index	sentence	label
4881	165	What do you have to bring to a credit card?	6.0
1641	500	মাসিক কত টাকা আয় করলে ক্রেডিট কার্ড নেওয়া যাবে?	2.0
5438	722	আপনাদের ক্রেদিত কার্ড নিতে কি ডকুমেন্টস লাগবে?	6.0
1576	435	ক্রেডিট কার্ড পাওার জন্য মাসিক কত টাকা বেতন পাওয়া লাগবে?	2.0
715	715	সিগনেচার কার্ডে বিলের হিসাব দেন।	1.0
476	476	কোন কার্ডের ফিস কত?	1.0
1908	767	ক্রেডিট কার্ড নিতে হইলে কত বেতন পাওয়া লাগব?	2.0
1600	459	ক্রেডিট কার্ড পাওার জন্য কত টাকা মাসিক বেতন পাওয়া লাগবে?	2.0
4748	32	Am I need to submt any papers for crdt card?	6.0
5737	1021	ক্রেডিট কার্ড নিতে আপনার কি কি দরকারি কাগজপত্র নেয়া হবে?	6.0
5530	814	কোন জিনিসচান আপনারা ক্রেডিট কার্ড খোলার ক্ষেত্রে?	6.0
5302	586	ক্রেডিট কার্ড নিব, কোন কাগজ লাগবে?	6.0

Figure 1: (Dataset after combining and shuffeling)

columns namely ["index","sentence","label"]. Here's a picture of the 1st 12 rows of df dataset.

0.2.2 Functions to check Query Language

For working further, we need to clean the data as much as possible, now the cleaning library for bangla text and english text are different. So I use a function that takes a string and detect if the given string is writing in english alphabet or not.If the string is in english language then it'll return true otherwise false.

```
check('ভাই, প্লাতিনুম আর সিগ্নাতুর ক্রেদিত কার্ডে এক বছরে বিল কত কাটবে?')
```

```
False
```

```
check('What do you have to bring to a credit card?')
```

```
True
```

Figure 2: Checking if the letters are in english or not

0.2.3 Splitting rows into different arrays based on language

Applying the checking functions I devide the dataset into 2 different arrays. Queries that are english and romanized bangla are in one array and the bangla Language query are in other array.The 1st five row of each bangla and english array are shown below. Next I can start applying all the cleaning process into this two arrays.

```
|: ban_sentences_final[0:5],ban_labels_final[0:5]

|: (array(['মাসিক কত টাকা আয় করলে ক্রেডিট কার্ড নেওয়া যাবে?',
         'আপনাদের ক্রেদিত কার্ড নিতে কি ডকুমেন্টস লাগবে?',
         'ক্রেডিট কার্ড পাওার জন্য মাসিক কত টাকা বেতন পাওয়া লাগবে?',
         'সিগনেচার  কার্ডে বিলের হিসাব দেন।', 'কোন কার্ডের ফিস কত?'],
        dtype='<U75'), array([2., 6., 2., 1., 1.]))

|: eng_sentences_final[0:5],eng_labels_final[0:5]

|: (array(['What do you have to bring to a credit card?',
         'Am I need to submt any papers for crdt card?',
         'Credit Card interest charge kto mnthly',
         'Quickly recharge my mobile from bank balance',
         'What kind of documents will you take for credit card?'],
        dtype='<U105'), array([6., 6., 4., 5., 6.]))
```

Figure 3: 2 arrays based on different language

0.2.4 Make relatively cleaner dataset for further use

As i said before I can start applying all the cleaning process into this two arrays.But I want to create a new notebook for further processing and also the shape of the english and bangla text containing array are quite small. So I create another 2 dataset that contains the query and the label of that query and each query of one specific dataframe have same language. after make 2 dataframe I converted them to .csv files named 'englisgTexts.csv' and 'banglaTexts.csv' and downloaded them. From now on I worked on these 2 dataframes.I also strart working on new notebook to keep a better track of my codes.

	sentence	label
0	What do you have to bring to a credit card?	6.0
1	Am I need to submt any papers for crdt card?	6.0
2	Credit Card interest charge kto mnthly	4.0
3	Quickly recharge my mobile from bank balance	5.0
4	What kind of documents will you take for credit card?	6.0
5	annual costing for credit card?	1.0
6	please tell how mch do brac charge fr credit card mnthly	4.0
7	Gold card, what kind of facilitates?	3.0
8	amar monthly income 30000, ami credit card nite parbo?	2.0
9	ami amar phone ta recharge korte chaitesi	5.0

```
df2.head(10)
```

	sentence	label
0	মাসিক কত টাকা আয় করলে ক্রেডিট কার্ড নেওয়া যাবে?	2.0
1	আপনাদের ক্রেদিত কার্ড নিতে কি ডকুমেন্টস লাগবে?	6.0
2	ক্রেডিট কার্ড পাওার জন্য মাসিক কত টাকা বেতন পাওয়া লাগবে?	2.0
3	সিগনেচার কার্ডে বিলের হিসাব দেন।	1.0
4	কোন কার্ডের ফিস কত?	1.0
5	ক্রেডিট কার্ড নিতে হইলে কত বেতন পাওয়া লাগব?	2.0

Figure 4: 2 dataframe

0.2.5 Loading data in new notebook

Iload the data that I saved as the csv files to started working on them. /i first try to see the number of query under each label, so i use sns counterplot to get a better understang of the data. From the box plot we can see that the data are distributed quite normally. And there is no extreme spike on the data.

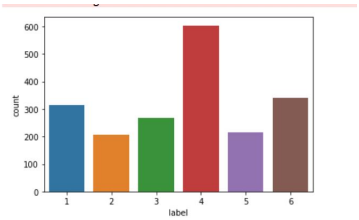

Figure 5: plot on english texts

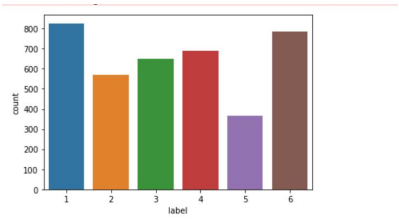

Figure 6: plot on bangla texts

0.2.6 Started cleaning text

0.2.6.1 Making Corpus

For NLP it's important to clean the text as much as possible and try to remove unnecessary words, punctuation's, digit or anything that will not be important to predict the result.He words lke "hm,please, sir" won't make any difference in predicting result.Tjese are called stopwords.There are many stopwords pkg in python to hep us for that. For english /i used the most well known NLTK stopwords And for bangla I found the most better is stopwordsiso. Also it's important to keep the root of any words such as "walk, walked,walking" all of them should be represented with "walk" as it's the root of these 3 words.To make the words in their root from is a process called Stamming. There are many different stamming library to stem the words for us. NLTK is the most used and most common one and I use nltk for stamming the english font text. and for bangla there are some of stamming library are available, but none of them are as enrich as the english words, However these bangla stopwrds and stammers helped me a lot to clean the bangla fonted text.here are the 1st few row of cleaned querys of both datas

```
(['স্য ক্রডিট কার্ড সুদ হ আপনা জানান প্লিজ',
  'ব্র্যাক ক্রেদিত কার্ড চাইল কাগজপত্র জমা দিব',
  'ব্র্যাক ক্রেডিট কার্ড বিল ব্যাপার',
  'ক্রেডিট কার্ড চা মাসিক সু হ',
  'হা মাসিক সুদ ক্রেডিট কার্ড',
  'আপনা ক্রেডিট কার্ড কাগজপত্র জমা ন',
  'প্লিজ',
  'ইনকাম চা কম ক্রেডিট কার্ড দিব',
  'ফ্যাসিলি পাবো ব্র্যাক ব্যাংক ক্রেডিট কার্ড নিল',
  'দাদা মাসিক ইন্টারেস্ট রেট ক আপনা জানান ক্রেডিট কার্ড'],
 ['credit card nite chail koto incom kora lagb shei shompork jant chacchi',
  'way rechag phone bank account',
  'ami hajar taka proti mash incom korl ki amak credit card dibe',
  'top',
  'amar credit card er pichon kto khoroch hobe bochor',
  'dear sir credit card interest charg',
  'much yearli bill credit card',
  'credit card nite monthli incom koto lage ei info lagb',
  'proti mash koto taka pele credit card nite parbo shei info lagb',
  'amar credit card ase ekta kto katb year e'])
```

Figure 7: corpus of bangla and english texts

0.2.6.2 Creating Bag of words and exporting them for further use

Bag of words model can be done by assigning each word a unique number. Then any document we see can be encoded as a fixed-length vector with the length of the vocabulary of known words. The value in each position in the vector could be filled with a count or frequency of each word in the encoded document.For word counting and mapping data I used CountVectorizer.The CountVectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.the fit_transform model learn from given documents, in this case corpus and encode each as a vector. As I have 2 entirely different corpus I use 2 count vectorizer one for bangla and english each.I also exported them as I needed them later when to predict a new text.For predicting a new text, the new cleaned text is need to be transformed accordint to the appropriate vectorizer. I use cv1 as my englishVectorizer and cv22 for bangla, detecting wheather new text is english or bangla i send them to transformed by appropriate vectorizer.

Make Bag of Words and export corresponding countingVectorizer

In [ ]:
```
def bag_of_words(textType):
  if(textType == 'en'):
    cv1 = CountVectorizer(max_features = 5100)
    X = cv1.fit_transform(corpusEnglish).toarray()
    y = df1.iloc[:,1].values
    cv = cv1
    pickle.dump(cv1, open("vectorEng.pickel", "wb"))
  else:
    cv2 = CountVectorizer(max_features = 5100)
    X = cv2.fit_transform(corpusBangla).toarray()
    y = df2.iloc[:,1].values
    cv = cv2
    pickle.dump(cv2, open("vectorBan.pickel", "wb"))
  return X,y,cv
```

In [ ]:
```
X_eng,y_eng,cvEng = bag_of_words('en')
X_ban,y_ban,cvBan = bag_of_words('bn')
```

In [ ]:
```
X_eng.shape,X_ban.shape
```

Out[ ]:
```
((1950, 618), (3892, 190))
```

Figure 8: count vectorizer

0.3 Train Test Split

Here we are using the 80/20 ratio for training and testing dataset.Now for ideal scenario the dataset should be splitted into 3 part. Train, validation and Test.But the data I have are already quite small so I'm using my test set as both validation and test set.

0.4 Testing Different ML Models

Now I apply some ML models on test part of both datasets.and also plotting the results on boxplot.These 2 figures are the box plot of english and bangla respectively.

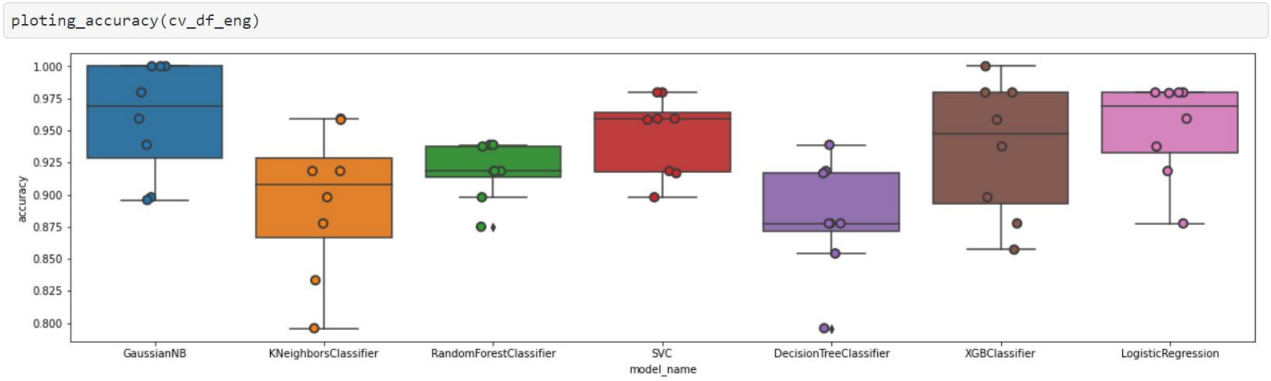

Figure 9: model vs accuracy on english test set

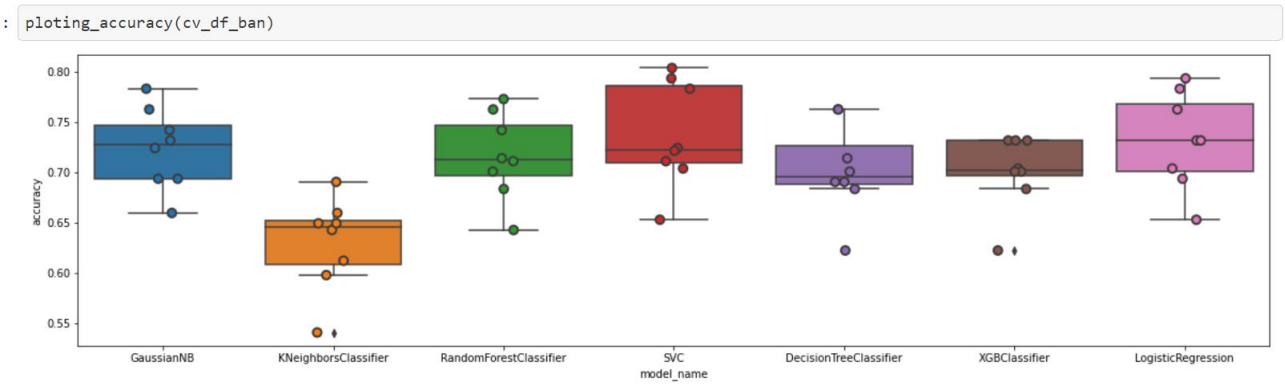

Figure 10: model vs accuracy on bangla test set

0.5 Model Evaluation

0.5.1 Mode Summary

From the boxplot we can see that gaussianNB works best for english model and SVC works best for bangla model. A numerical summary of all these models and their predicting score will give us a better understanding.The figures shows summary for english and then bangla respectively.

```
cv_df_eng.groupby('model_name').accuracy.mean()
```

```
model_name
DecisionTreeClassifier    0.882068
GaussianNB                0.958918
KNeighborsClassifier      0.894877
LogisticRegression        0.951318
RandomForestClassifier    0.917889
SVC                       0.946110
XGBClassifier             0.935959
Name: accuracy, dtype: float64
```

Figure 11: model accuracy on english test set

```
: cv_df_ban.groupby('model_name').accuracy.mean()
```

```
: model_name
  DecisionTreeClassifier    0.703582
  GaussianNB                0.724082
  KNeighborsClassifier      0.630418
  LogisticRegression        0.731893
  RandomForestClassifier    0.716442
  SVC                       0.737008
  XGBClassifier             0.701018
  Name: accuracy, dtype: float64
```

Figure 12: model accuracy on bangla test set

0.5.2 Checking ovrefitting or not

From the figure we can see that in both dataset the 3 best models are SVC, GaussianNB, LogisticRegression. Now I'm applying all these 3 model to each of eng and bangla text types

both training and testing model. If the accuracy difference from training and testing dataset are less then 3 percent then i'm considering as a good model. The model name,accuracy on test set, accuracy on train set are saved to an array of objects. The first 3 rows are for english data and the last 3rd are for bangla data.

```
GaussianNB,0.9935897435897436,0.9794871794871794
SVC,0.9955128205128205,0.9897435897435898
LogisticRegression,0.9974358974358974,0.9923076923076923
GaussianNB,0.7709604882749759,0.7406931964056482
SVC,0.7995502730485062,0.7650834403080873
LogisticRegression,0.7947317699967876,0.7560975609756098
```

Figure 13: model and the accuracy on train and test set

From these figure it's clear that LogisticRegression working best for the english dataet and SVC for bangla Dataset.Despite of having many data in the bangla dataset the accuracy of english dataset are much higher then bangla dataset.The key reason for this is that the stopwords andstammer for english are much more enrich then bengali.

0.5.3 Model Create

After evaluation I decided to make the model for englishText in LogisticRegression and for banglaText in SVC

0.6 Model Export

After creating the model I export the model in .pkl format for later use in deployment.

0.7 Accuracy difference between notebook and exported model

As we can see that both models give the same accuracy so the models are exported correctly.

```
loaded_model = pickle.load(open('modelEnglish.pkl', 'rb'))
y_predTestEng = classifierEnglish.predict(X_test_eng)
resultEng = loaded_model.predict(X_test_eng)
accuracyModelEng = accuracy_score(y_test_eng, y_predTestEng)
accuracyImportedModelEng = accuracy_score(y_test_eng, y_predTestEng)
```

```
accuracyModelEng,accuracyImportedModelEng
```

```
(0.9923076923076923, 0.9923076923076923)
```

Figure 14: accuracy on notebook and exported model

```
loaded_model = pickle.load(open('modelBangla.pkl', 'rb'))
y_predTestBan = classifierBangla.predict(X_test_ban)
resultBan = loaded_model.predict(X_test_ban)
accuracyModelBan = accuracy_score(y_test_ban, y_predTestBan)
accuracyImportedModelBan = accuracy_score(y_test_ban, y_predTestBan)
```

```
accuracyModelBan,accuracyImportedModelBan
```

```
(0.7650834403080873, 0.7650834403080873)
```

Figure 15: accuracy on notebook and exported model

0.8 Some Discussions

0.8.1 Devide Dataset based upon different Language

we can just put all dataset and can apply a model but that way there were more chances of unnecessary tokenizing. So it's ideal to sub-devide data and apply the languare based tokenizer

0.8.2 Model Selection

Both ML Classification (as we classify the query) and deep learnig model can be applied here. But deep learning model work best when there are a huge data. The data we are working on here are relatively small and I try to structure it as much as I can, and ML models tends to work well on structured data. So I choose an ML classification model here.

0.8.3 Bias

Bias is when the ml model generalize results. IT results to underfitting , meaning the model doesn't work well on train data. But here after model evaluation, the models are giving much

better results.

0.8.4 Overfitting

The test accuracy and train accuracy difference are less then3 percent. So it can savely say that it's free of overfitting.

0.8.5 Spelling Error Resilient

0.8.5.1 For English Model

For english model I used logigisticRegression. While prediction, this test the input using all the count vectorizer value and which ever value gives the highest value between zero and one considering you are using sigmoid transfer function, the input means that particular word.

0.8.5.2 For Bangla Model

SVC kernel is used for bangla text. whenever a new unknown near to train matrix included word appears it detects the best fit thisw incorrect word should go to and thus it's spelling error resillieng

0.9 FURTHER IMPROVEMENT

- detect Roman bangla from englishText and convert it into english/bangla
- use better stopwords and stammer for banglaText
- use word2vec instead of bag of words model
- try to aapply Deep Learning Model