TELUS CASE STUDY

By Shafin Rezwan

Preprocess

Before beginning any work on the data set it is important to understand what we are looking at.

- The structure of the data set consists of
  - Conversation ID: a unique identifier for each conversation
  - TurnNumber: sequence number within the conversation
  - utteranceID: unique identifier for each individual message
  - Utterance: actual text of message
  - authorRole: specifies if message was sent by agent or customer

- The main piece of data we would need to help us categorize the data set will be utterance from the customers.
- Before we begin we will need to clean the data to ensure there are no mistakes in how it was collected
  - This is done in code by ensuring there is no empty data blocks as well as ensuring the data is sorted in the correct order

APPROACH #1 - Latent Dirichlet Allocation (LDA) model

- Using this model I found the most frequently used words for 5 topics
- The way this model works is it converts that text into a vector format for machine learning and NLP models
- Then the frequency of these words are calculated in a matrix

Results from Conducting the LDA

LDA Model with 5 Topics:

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
0	casey	account	time	hii	money
1	riley	byee	05	account	want
2	sir	close	18	address	need
3	did	information	nice	change	check
4	thank	sir	june	okay	balance
5	hello	change	hai	lost	transfer
6	bye	new	06	number	sure
7	thanks	want	day	credit	help
8	yes	address	2018	ssn	account
9	ok	thank	great	card	hi

After receiving these results I was not happy with this approach . The model does not purpose a topic name just a number which is not too helpful.

It does categorize important words in the convo together which is great

However there are inconsistencies here. As you can see 'hii' is in topic 4 and 'hello' is in topic 1. It should be paired together.

APPROACH #2 - Random Forest

- Based on what was done in the previous model I took that and conducted the random forest approach
- It works by ensembling multiple decision trees that make a prediction on which category the text should be apart of
- For this I manually had to create categories that the model uses

```
categories = {
    "account": 0,
    "technical support": 1,
    "loan inquiries": 2,
    "credit card": 3,
    "transaction history": 4,
   # "general_inquiry": 5
}
```

These categories were chosen based on the results of frequent words I found from the LDA as well as skimming through the data manually and self determining how to label these customer interactions.

To implement this:

- The data set is split up in 2.
- - a training set takes up 80% and a testing set that uses 20% of the data
- After which this trained model can be used for future new messages and handling how to categorize those

RESULTS

In conclusion the Random Forest approach was much more effective.

- It was more accurate and useful for this case
- It also runs efficiently and does not take too long to run

```
Model Accuracy: 0.9941046425939573

Classification Report:
                     precision    recall  f1-score   support

            account       1.00      1.00      1.00      3263
  technical support       0.99      0.99      0.99       801
     loan inquiries       0.96      0.98      0.97       299
        credit card       1.00      1.00      1.00      2200
transaction history       0.98      0.95      0.97       222

           accuracy                           0.99      6785
          macro avg       0.99      0.98      0.99      6785
       weighted avg       0.99      0.99      0.99      6785
```

My only recommendation is that since the categories need to be declared manually. It can be prone to errors and inaccuracies. Another method called the BERT model is found to be very accurate as it uses natural language processing understanding the context of the speech better. However it is much more taxing on the computer system causing it to be expensive.

This is the results from the testing.

- The model is 99% accurate
- Precision and recall scores are high among all categories
  - This means the model classifies the customer utterances correctly

Based on the testing I will say the Random Approach model is quite accurate and would be my chosen method for this data. To further test individual cases I created the predict.py file where I inputted individual utterances, some I made myself, some I took from the data and printed the individual categories they were assigned to. It worked very well.

```
print(classify_text("I WANT TO CHANGE MY ACCOUNT ADDRESS "))
print(classify_text("I was wrongly charged for late fees even if I paid back everything on time"))
print(classify_text("I lost my credit card, what should I do?"))
print(classify_text("Order 47 checks for me and tell me bank's routing number"))
```

```
account
transaction history
credit card
account
```