Ticket Classification

You have been given a dataset about customer complaints, pulled from the customer support desk of a leading multinational bank.

Business problem:

The bank has observed that it sometimes takes a long time to respond to customer queries. This happens partly because the stakeholder who can resolve the customer complaint is usually not the first person who gets to look into the matter. This on an average delays the first effective response by 18 hours.

The bank wants to solve this problem by automating the process of assigning customer tickets to relevant stakeholders.

You need to help the bank to come up with a solution to be able to classify the customers' tickets automatically with a reasonable degree of accuracy.

Dataset:

The data set is given to you as two `json` files, `complaints.json` and `mappings.json`. There is also a csv file that has no tags here. This can be used as a test file.
The `complaints.json` has the following format:


```
{
    "data": [
        {
            "text": "I recently opened a Citibank CitiGold Checking Account that was
advertised to have a signup bonus of   AAdvantage points upon the completion of two
consecutive bill payments and {$1000.00} in debit card purchases. After signing up
with the intent to complete these requirements a Citibank representative confirmed
that the offer applied to my new account and that I should complete the offer
requirements. This confirmation was communicated via their online secure message
feature, and I have attached a copy for reference. Once I completed the requirements
as directed, I inquired about the expected delivery date of the earned bonus via
secure message and I was told that my account was not targeted and that I would not
receive the bonus as promised. This conflicts with what I was previously told and I
have already spent a considerable amount of time to meet the requirements as
directed. ",
            "complaint_id": "bc_5"
        },
        {
            "text": "I went into Capital One bank to open a checking account. I
signed the signature card, provided my ID, made the opening deposit, and received a
temporary check/debit card. Today I still cannot get access to my account online. I
have spoken in person at the branch and over the phone to different representatives.
None of them have been able to help so far and checking back tomorrow seems to be the
best option after spending between 30 minutes and an hour with each representative
trying to resolve the issue of online access. It never occurred to me that  online
access would be difficult at any bank. If I had not already paid to order checks and
```

```
deposit slips, I would simply close this account and find a more competent bank. Can
you help??",
            "complaint_id": "bc_3"
        }
    ]
}
```

The `complaint_id` key is the alphanumeric id of the type of complaint that a customer has made. The description of what each `complaint_id` stands for is in the file named `mappings.json`. Its structure is as follows:

```
"all": {
        "Problems caused by my funds being low": "bc_0",
        "Problem caused by your funds being low": "bc_1",
        "Using a debit or ATM card": "bc_2"
}
```

There is also a file called `respondent.csv`, which has the mapping of issue_id with respondent id. This can be used to ascertain who should respond given the issue_id

Tasks:

1. **Data Understanding(Task 1)**:
    - Find out how many labeled customer complaints are in the data
    - Find out the relative frequency of tags in the dataset
    - Check if there are any encoding issues (try to read the data using open() and see if you need to use any special text encoding options, other than utf-8)
    - Create a report that identifies how much of each complaint is composed of commonly used English stopwords.

Complaint	Percentage of stopwords
1	20%
2	10%
3	50%
4	60%
..	30%

2. **Feature Engineering(Task 2)**:
    - Create a count matrix or a tfidf matrix

- Can you think of adding any other features apart from count matrix or tfidf matrix?
- If you are comfortable working with word-embeddings then create features based on pre-trained word vectors or features out of a BERT model (optional)

3. **Modeling(Task 3)**:
   - What models can you use to create a ticket classification system?
   - Try atleast one supervised learning approach (Naive Bayes, Linear Classifier, SVM, Tree based ensembles)
   - Can you implement a knn here?
   - Try deep learning based techniques such as rnn, lstm and bert (optional)
   - Will an unsupervised technique such as clustering work here?
   - Report the recall and precision for each class of complaint in the data that your finalized model is able to achieve
4. **Model Deployment(Task 4)**:
   - Once you finalize the model, write a python script that can be used to provide predictions on new complaints. Your prediction module should follow the following structure:

```
├── predict.py
│   ├── models
│   │   ├── model files (.h5 or .pickle files)
│   ├── test_data
├── README.MD
```

`predict.py` should contain the logic to load the saved model in models directory and provide predictions on tickets stored in the `test_data` folder. Your predictions should also include guidance on who should respond to the customer complaint.

Deliverables

- A well designed deck outlining the conclusions and the analysis (.ppt)
- A well structured code pushed on github (Write an informative README, well structured code/notebooks)
- *Optional*: A blog post on medium/personal blog/blogger/linkedin