Probem statement:

After the success of Tap Portable Bluetooth Speaker, Amazon is about to launch a new version in the market. To understand what features customers liked, Amazon did a focus group discussion with some selected customers - the audio in the discussions has been recorded. The reviews that the panelists gave to Tap is what interests Amazon.

As the data scientist, your task is convert the given audio file to text, assess which features of the bluetooth speaker are being talked about in the audio reviews. In Module 3, we extracted the top 15 features from the reviews. We can use this as our feature list, and assess which of these are present in the audio reviews.

Also, for future utility and for immediate analysis, you need to make a process/function that captures audio from the microphone, converts to text, analyses, and returns which of the features are being discussed in the audio.

Steps -

1. From speech_recognition module, import the sr utility. Instantiate the Recognizer class from the utility.
2. Load the given audio file "Recording2.wav" - this contains a sample audio for you to get comfortable with the module, use the 'AudioFile' method
3. With this as source, 'record' the audio from the file
4. Using 'recognize_google' method, convert the audio to text
5. Create a function to return the text for any given audio file (.wav format) as input.
6. Apply this function to the file 'Tap Review.wav' to extract the text from the audio review from the Amazon Tap focus group discsussions
7. Pre-process the text - tokenize into individual terms using NLTKs word_tokenize
8. Define feature_list as list of features of the product, containing the following terms -

- "echo", "alexa", "music", "sound", "button","bluetooth", "voice", "battery", "dot", "phone"

9. Identify which of the features are being talked of in the audio review
10. Recording from mic using the Microphone method

- instantiate
- with the mic as source, 'listen' to the mic - you can say anything you like using your own microphone
- speak a dummy review into the microphone, use the function from earlier to identify the product features being discussed in the audio

1. From speech_recognition module, import the sr utility. Instantiate the Recognizer class from the utility.

In [1]:
```
import speech_recognition as sr
```

In [3]:
```
recog = sr.Recognizer()
```

In [4]:
```
?recog.record
```

In [3]:
```
dir(recog)
```

Out[3]:
```
['__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__enter__',
 '__eq__',
 '__exit__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 'adjust_for_ambient_noise',
 'dynamic_energy_adjustment_damping',
 'dynamic_energy_ratio',
 'dynamic_energy_threshold',
 'energy_threshold',
 'listen',
 'listen_in_background',
 'non_speaking_duration',
 'operation_timeout',
 'pause_threshold',
 'phrase_threshold',
 'recognize_api',
 'recognize_bing',
 'recognize_google',
 'recognize_google_cloud',
 'recognize_houndify',
 'recognize_ibm',
 'recognize_sphinx',
 'recognize_wit',
 'record',
 'snowboy_wait_for_hot_word']
```

2. Load the given audio file "Recording2.wav" - this contains a sample audio for you to get comfortable with the module, use the 'AudioFile' method

In [18]:
```
samp = sr.AudioFile("Recording2.wav")
```

3. With this as source, 'record' the audio from the file

In [19]:
```
with samp as source:
    audio = recog.record(samp)
```

4. Using 'recognize_google' method, convert the audio to text

In [28]:
```
res = recog.recognize_google(audio)
```

In [30]:
```
res
```

Out[30]:
```
'I am so happy to make my first speech to text converter'
```

5. Create a function to return the text for any given audio file (.wav format) as input

In [41]:
```
def speech_to_text(file):
    samp = sr.AudioFile(file)

    with samp as source:
        audio = recog.record(samp)

    return recog.recognize_google(audio)
```

6. Apply this function to the file 'Tap Review.wav' to extract the text from the audio review from the Amazon Tap focus group discsussions

In [42]:
```
op_text =  speech_to_text("Tap Review.wav")
```

In [43]:
```
op_text
```

Out[43]:
```
'I love this little bluetooth speaker the Bluetooth connectivity is good sound qua
lity is amazing I listen to music all the time and I use Alexa all the time'
```

7. Pre-process the text - tokenize into individual terms using NLTKs word_tokenize

In [36]:
```
from nltk.tokenize import word_tokenize
```

In [48]:
```
tokens = word_tokenize(op_text.lower())
```

8. Define feature_list as list of features of the product, containing the following terms -

- "echo", "alexa", "music", "sound", "button","bluetooth", "voice", "battery", "dot", "phone"

From an earlier exercise, we had identified the top feautures for Amazon tap. This will be our list of features. We need to find which of these features are being talked of in this audio review of the product.

In [45]:
```
feature_list = ["echo", "alexa", "music", "sound", "button","bluetooth", "voice", '
```

9. Identify which of the features are being talked of in the audio review

In [49]:
```
review_features = [term for term in tokens if term in feature_list]
```

In [50]:
```
review_features
```

Out[50]:
```
['bluetooth', 'bluetooth', 'sound', 'music', 'alexa']
```

In [51]:
```
review_features = list(set(review_features))
```

In [52]:
```
review_features
```

Out[52]:
```
['music', 'sound', 'alexa', 'bluetooth']
```

10. Recording from mic using the Microphone method

- instantiate
- with the mic as source, 'listen' to the mic - you can say anything you like using your own microphone
- speak a dummy review into the microphone, use the function from earlier to identify the product features being discussed in the audio

In [53]:
```
mic = sr.Microphone()
```

In [54]:
```
mic.list_microphone_names()
```

Out[54]:
```
['Microsoft Sound Mapper - Input',
 'Microphone Array (Realtek High ',
 'Microsoft Sound Mapper - Output',
 'Speaker/HP (Realtek High Defini']
```

In [56]:
```
with mic as source:
    audio = recog.listen(source)
```

In [57]:
```
recog.recognize_google(audio)
```

Out[57]:
```
'I just love this phone'
```

In [59]:
```
with mic as source:
    audio = recog.listen(source)
```

In [60]:
```
recog.recognize_google(audio)
```

Out[60]:
```
'the speaker is amazing with good backup and excellent Bluetooth connectivity I ju
st love this bluetooth speaker'
```

In [61]:
```
def get_review_features(review_text):
    feature_list = ["echo", "alexa", "music", "sound", "button","bluetooth", "voice
    tokens = word_tokenize(review_text.lower())
    review_features = [term for term in tokens if term in feature_list]
    review_features = list(set(review_features))
    return review_features
```

In [ ]:
```
res = recog.recognize_google(audio)
```

In [ ]:
```
get_review_features(res)
```