L2: Evaluate Inputs: Classification

Setup

Load the API key and relevant Python libaries.

In this course, we've provided some code that loads the OpenAI API key for you.

```python
In [8]: import os
        import openai
        from dotenv import load_dotenv, find_dotenv
        _ = load_dotenv(find_dotenv()) # read local .env file

        openai.api_key  = os.environ['OPENAI_API_KEY']
```

```python
In [9]: def get_completion_from_messages(messages,
                                         model="gpt-3.5-turbo",
                                         temperature=0,
                                         max_tokens=500):
            response = openai.ChatCompletion.create(
                model=model,
                messages=messages,
                temperature=temperature,
                max_tokens=max_tokens,
            )
            return response.choices[0].message["content"]
```

Classify customer queries to handle different cases

In [10]:
```python
delimiter = "####"
system_message = f"""
You will be provided with customer service queries. \
The customer service query will be delimited with \
{delimiter} characters.
Classify each query into a primary category \
and a secondary category.
Provide your output in json format with the \
keys: primary and secondary.

Primary categories: Billing, Technical Support, \
Account Management, or General Inquiry.

Billing secondary categories:
Unsubscribe or upgrade
Add a payment method
Explanation for charge
Dispute a charge

Technical Support secondary categories:
General troubleshooting
Device compatibility
Software updates

Account Management secondary categories:
Password reset
Update personal information
Close account
Account security

General Inquiry secondary categories:
Product information
Pricing
Feedback
Speak to a human

"""
user_message = f"""\
I want you to delete my profile and all of my user data"""
messages =  [
{'role':'system',
 'content': system_message},
{'role':'user',
 'content': f"{delimiter}{user_message}{delimiter}"},
]
response = get_completion_from_messages(messages)
print(response)
```

```
{
   "primary": "Account Management",
   "secondary": "Close account"
}
```

In [11]:
```python
user_message = f"""\
Tell me more about your flat screen tvs"""
messages =  [
{'role':'system',
 'content': system_message},
{'role':'user',
 'content': f"{delimiter}{user_message}{delimiter}"},
]
response = get_completion_from_messages(messages)
print(response)
```

```
{
  "primary": "General Inquiry",
  "secondary": "Product information"
}
```

In [ ]: