In [2]:
```
from dotenv import load_dotenv
import os
```

In [3]:
```
import openai
# Fine tune example
# https://github.com/norahsakal/fine-tune-gpt3-model/blob/main/fine_tune_ste
```

In [4]:
```
API_KEY = "sk-........................."

load_dotenv('demo.env')
API_KEY = os.environ.get("api_key")
```

In [5]:
```
openai.api_key = API_KEY
os.environ["OPENAI_API_KEY"] = API_KEY
```

Validate the data file is formatted correctly

In [6]:
```
#!openai tools fine_tunes.prepare_data -f data.jsonl
```

Method1: Fine tune using OpenAI api CLI

In [7]:
```
# openai api fine_tunes.create -t <TRAIN_FILE_ID_OR_PATH> -m <BASE_MODEL>
#!openai api fine_tunes.create -t data3.jsonl -m davinci --suffix "fish_n_ar
```

Method2: Fine tune using OpenAI api

Upload the FILE to use it on fine tuning refer : https://platform.openai.com/docs/api-reference/files

In [8]:
```
# define data file name
file_name =  "data5.jsonl"
```

In [9]:
```
upload_response = openai.File.create(
  file=open(file_name, "rb"),
  purpose='fine-tune'
)
upload_response
```

Out[9]:
```
<File file id=file-MemCXJ5jUwmmQKJAPaHqhhn2 at 0x7f3a862ce770> JSON: {
  "bytes": 271,
  "created_at": 1678208575,
  "filename": "file",
  "id": "file-MemCXJ5jUwmmQKJAPaHqhhn2",
  "object": "file",
  "purpose": "fine-tune",
  "status": "uploaded",
  "status_details": null
}
```

In [10]:
```
file_id = upload_response.id
file_id
```

Out[10]:
```
'file-MemCXJ5jUwmmQKJAPaHqhhn2'
```

Fine-tune a model

In [11]:
```
## If you'd like to use DaVinci instead, then add it as a base model to fine
## openai.FineTune.create(training_file=file_id, model="davinci")
```

In [12]:
```
fine_tune_response = openai.FineTune.create(training_file=file_id)
fine_tune_response
```

Out[12]:

```
<FineTune fine-tune id=ft-UqmDkpRzJgscPg5hhyla79gM at 0x7f3a54812720> JSON:
{
  "created_at": 1678208620,
  "events": [
    {
      "created_at": 1678208620,
      "level": "info",
      "message": "Created fine-tune: ft-UqmDkpRzJgscPg5hhyla79gM",
      "object": "fine-tune-event"
    }
  ],
  "fine_tuned_model": null,
  "hyperparams": {
    "batch_size": null,
    "learning_rate_multiplier": null,
    "n_epochs": 4,
    "prompt_loss_weight": 0.01
  },
  "id": "ft-UqmDkpRzJgscPg5hhyla79gM",
  "model": "curie",
  "object": "fine-tune",
  "organization_id": "org-0tuNTBQvZ1vslwbsfURfnRNi",
  "result_files": [],
  "status": "pending",
  "training_files": [
    {
      "bytes": 271,
      "created_at": 1678208575,
      "filename": "file",
      "id": "file-MemCXJ5jUwmmQKJAPaHqhhn2",
      "object": "file",
      "purpose": "fine-tune",
      "status": "processed",
      "status_details": null
    }
  ],
  "updated_at": 1678208620,
  "validation_files": []
}
```


In [13]:
```
# Fine tune progress
fine_tune_events = openai.FineTune.list_events(id=fine_tune_response.id)
fine_tune_events
```

Out[13]:

```
<OpenAIObject list at 0x7f3a862ced10> JSON: {
  "data": [
    {
      "created_at": 1678208620,
      "level": "info",
      "message": "Created fine-tune: ft-UqmDkpRzJgscPg5hhyla79gM",
      "object": "fine-tune-event"
    }
  ],
  "object": "list"
}
```

In [14]:
```
# Fine tune object
retrieve_response = openai.FineTune.retrieve(id=fine_tune_response.id)
retrieve_response
```

Out[14]:
```
<FineTune fine-tune id=ft-UqmDkpRzJgscPg5hhyla79gM at 0x7f3a5486c950> JSON:
{
  "created_at": 1678208620,
  "events": [
    {
      "created_at": 1678208620,
      "level": "info",
      "message": "Created fine-tune: ft-UqmDkpRzJgscPg5hhyla79gM",
      "object": "fine-tune-event"
    }
  ],
  "fine_tuned_model": null,
  "hyperparams": {
    "batch_size": null,
    "learning_rate_multiplier": null,
    "n_epochs": 4,
    "prompt_loss_weight": 0.01
  },
  "id": "ft-UqmDkpRzJgscPg5hhyla79gM",
  "model": "curie",
  "object": "fine-tune",
  "organization_id": "org-0tuNTBQvZ1vslwbsfURfnRNi",
  "result_files": [],
  "status": "pending",
  "training_files": [
    {
      "bytes": 271,
      "created_at": 1678208575,
      "filename": "file",
      "id": "file-MemCXJ5jUwmmQKJAPaHqhhn2",
      "object": "file",
      "purpose": "fine-tune",
      "status": "processed",
      "status_details": null
    }
  ],
  "updated_at": 1678208620,
  "validation_files": []
}
```

In [15]:
```
## Save the fine tune model
if fine_tune_response.fine_tuned_model == None:
    fine_tune_list = openai.FineTune.list()
    fine_tuned_model = fine_tune_list['data'][0].fine_tuned_model
```

In [17]:
```
## openai.FineTune.list()
```

In [18]:
```
new_prompt = "What type is a dolphin.->"
```

In [19]:
```
answer = openai.Completion.create(
  model=fine_tuned_model,
```

```
  prompt=new_prompt,
  max_tokens=10, # Change amount of tokens for longer completion
  temperature=0
)
```

In [21]:
```
#answer['choices'][0]['text']
```

In [ ]:
```

```