

GenAI For Business Analysis:

Fine-Tuning LLMs

Key Takeaways

Task 1

Set Up the Project Environment

- Storing secret keys in an .env file is a vital security practice, bolstering data protection and mitigating risks.

Task 2

Prepare the training data

- To be able to use the data for the fine-tuning purpose, we first need to convert each row of the dataframe into the following format:

```
{  
    "system_message": {  
        "role": "system",  
        "content": "example of system prompt"  
    },  
    "user_message": {  
        "role": "user",  
        "Content": "example of user prompt"  
    },  
    "assistant_message": {  
        "role": "assistant",  
        "content": "desired output based on the user and system prompts."  
    }  
}
```

Task 3

Fine-tune GPT-3.5 based on our training data

- At a high level, fine-tuning consists of the following steps:
 1. Prepare and upload training data.
 2. Train a new fine-tuned model.
 3. Evaluate results and return to step 1 if necessary.
 4. Utilize your fine-tuned model.
- Fine-tuning beats few-shot learning with more training data, improving task performance while reducing costs and enabling faster, more accurate responses.
- You can adjust hyperparameters while fine-tuning your model, such as batch size; learning rate multiplier; and number of epochs (n_epochs).

Task 4

Evaluate model

- There are different parameters to consider while evaluating the performance of your fine-tuned model:
 1. Training Loss
 2. Training mean token accuracy
- An effective fine-tuning is indicated by a decrease in training loss over the learning steps and an increase in training mean token accuracy over the steps.

Task 5

Deploy our model

- It can be helpful to use fine-tuning when the context of your data is important.