ASSIGNMENT : 4.5

NAME : G DEEKSHITH RAJ
ROLL NO : 2403A51248

Prepare sample data

Subtask : Create a list of 10 short email samples with corresponding categories.

Reasoning: Define the zero-shot prompt, select an email sample, and combine them.

Code:

```
email_samples = [
    ("Regarding your recent invoice", "Billing"),
    ("My internet connection is down", "Technical Support"),
    ("Suggestion for improving your service", "Feedback"),
    ("Meeting request", "Others"),
    ("Payment confirmation", "Billing"),
    ("Trouble logging in", "Technical Support"),
    ("Complaint about a feature", "Feedback"),
    ("Holiday greetings", "Others"),
    ("Question about my bill", "Billing"),
    ("Software update failed", "Technical Support"),
]

print(email_samples)
```

Out put:

[('Regarding your recent invoice', 'Billing'), ('My internet connection is down', 'Technical Support'), ('Suggestion for improving your service', 'Feedback'), ('Meeting request', 'Others'), ('Payment confirmation', 'Billing'), ('Trouble logging in', 'Technical Support'), ('Complaint about a feature', 'Feedback'), ('Holiday greetings', 'Others'), ('Question about my bill', 'Billing'), ('Software update failed', 'Technical Support')]

Zero-shot prompting

Subtask:

Design a zero-shot prompt to classify a single email and apply it to one of the sample email

Reasoning: Define the zero-shot prompt, select an email sample, and combine them.

OUT PUT:

Classify the following email into one of these categories: "Billing", "Technical Support", "Feedback", and "Others".

Email:

My internet connection is down

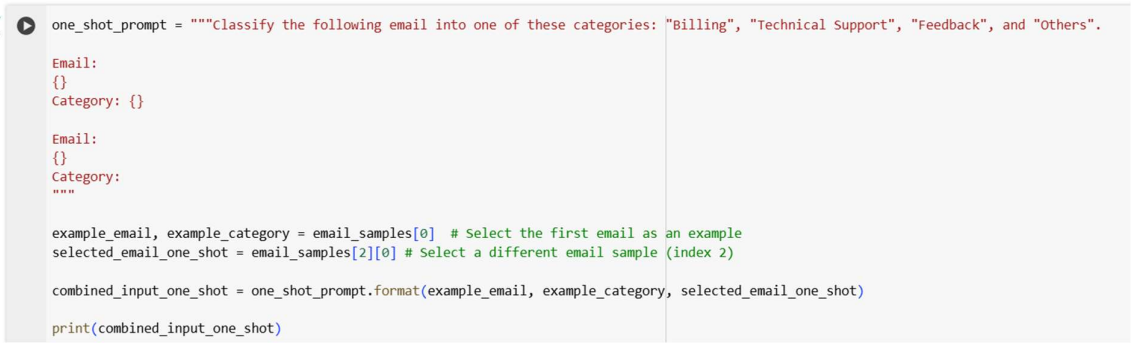
Category:

One-shot prompting

Subtask:

Design a one-shot prompt by adding one labeled example and apply it to a new email.

Reasoning: Define the one-shot prompt structure, select an example email and a new email to classify, and then format the prompt.



```
one_shot_prompt = """Classify the following email into one of these categories: "Billing", "Technical Support", "Feedback", and "Others".

Email:
{}
Category: {}

Email:
{}
Category:
"""

example_email, example_category = email_samples[0]  # Select the first email as an example
selected_email_one_shot = email_samples[2][0] # Select a different email sample (index 2)

combined_input_one_shot = one_shot_prompt.format(example_email, example_category, selected_email_one_shot)

print(combined_input_one_shot)
```

OUT PUT:

Classify the following email into one of these categories: "Billing", "Technical Support", "Feedback", and "Others".

Email:

Regarding your recent invoice

Category: Billing

Email:

Suggestion for improving your service

Category:

Few-shot prompting

Subtask:

Design a few-shot prompt by adding 3-5 labeled examples and apply it to a new email.

Reasoning: Define the few-shot prompt template, select example emails and a new email to classify, and format the prompt.

```
few_shot_prompt = """Classify the following email into one of these categories: "Billing", "Technical Support", "Feedback", and "Others".

Email:
{}
Category: {}

Email:
{}
Category: {}

Email:
{}
Category: {}

Email:
{}
Category:
"""

# Select 3 example emails
example_emails_few_shot = [email_samples[0], email_samples[3], email_samples[4]]

# Select a new email to classify, different from the examples
selected_email_few_shot = email_samples[6][0]

combined_input_few_shot = few_shot_prompt.format(
    example_emails_few_shot[0][0], example_emails_few_shot[0][1],
    example_emails_few_shot[1][0], example_emails_few_shot[1][1],
    example_emails_few_shot[2][0], example_emails_few_shot[2][1],
    selected_email_few_shot
)

print(combined_input_few_shot)
```

OUT PUT: Classify the following email into one of these categories: "Billing", "Technical Support", "Feedback", and "Others".

Email:

Regarding your recent invoice

Category: Billing

Email:

Meeting request

Category: Others

Email:

Payment confirmation

Category: Billing

Email:

Complaint about a feature

Category:

Evaluation

Subtask:

Select 5 test emails and run zero-shot, one-shot, and few-shot prompting on them. Compare the accuracy and clarity of the responses.

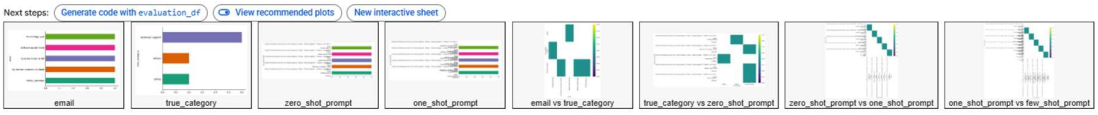
Reasoning: Select 5 test emails that were not used as examples in the one-shot or few-shot prompts, and then generate the zero-shot, one-shot, and few-shot prompts for each of these test emails. Store these prompts along with the original email and its true category in a structured format.

```python
# Select test emails that were not used as examples
used_in_one_shot = [email_samples[0], email_samples[2]]
used_in_few_shot = [email_samples[0], email_samples[3], email_samples[4], email_samples[6]]
used_emails = used_in_one_shot + used_in_few_shot
unique_used_emails = list(set(used_emails))

test_emails = [email for email in email_samples if email not in unique_used_emails]

# Ensure we have at least 5 test emails
if len(test_emails) < 5:
    print("Warning: Not enough unique emails left for 5 test samples. Using all remaining emails.")
    test_emails = test_emails[:5]
else:
    test_emails = test_emails[:5]


evaluation_data = []

for email, category in test_emails:
    zero_shot_input = zero_shot_prompt.format(email)
    one_shot_input = one_shot_prompt.format(example_email, example_category, email)
    few_shot_input = few_shot_prompt.format(
        example_emails_few_shot[0][0], example_emails_few_shot[0][1],
        example_emails_few_shot[1][0], example_emails_few_shot[1][1],
        example_emails_few_shot[2][0], example_emails_few_shot[2][1],
        email
    )

    evaluation_data.append({
        "email": email,
        "true_category": category,
        "zero_shot_prompt": zero_shot_input,
        "one_shot_prompt": one_shot_input,
        "few_shot_prompt": few_shot_input
    })

import pandas as pd

evaluation_df = pd.DataFrame(evaluation_data)
display(evaluation_df)
```

OUT PUT:

	email	true_category	zero_shot_prompt	one_shot_prompt	few_shot_prompt
0	My internet connection is down	Technical Support	Classify the following email into one of these...	Classify the following email into one of these...	Classify the following email into one of these...
1	Trouble logging in	Technical Support	Classify the following email into one of these...	Classify the following email into one of these...	Classify the following email into one of these...
2	Holiday greetings	Others	Classify the following email into one of these...	Classify the following email into one of these...	Classify the following email into one of these...
3	Question about my bill	Billing	Classify the following email into one of these...	Classify the following email into one of these...	Classify the following email into one of these...
4	Software update failed	Technical Support	Classify the following email into one of these...	Classify the following email into one of these...	Classify the following email into one of these...

Next steps: Generate code with evaluation_df | View recommended plots | New interactive sheet



Summary:

Summary:

Data Analysis Key Findings

- Zero-shot, one-shot, and few-shot prompts were successfully generated for email classification based on provided templates and examples.

- Five test emails, distinct from the examples used in prompt generation, were selected for evaluation.
- For each test email, the corresponding zero-shot, one-shot, and few-shot prompt strings were created and stored in a pandas DataFrame along with the email content and its true category.

Insights or Next Steps

- The next crucial step is to use a language model to process the generated prompts for the 5 test emails and obtain the predicted categories for each prompting technique.
- After obtaining the predictions, the accuracy of each prompting technique can be calculated by comparing the predicted categories to the true categories, allowing for a comparison and reflection on their effectiveness.

This directory includes a few sample datasets to get you started.

*   `california_housing_data*.csv` is California housing data from the 1990 US
    Census; more information is available at:
    https://docs.google.com/document/d/e/2PACX-1vRhYtsvc5eOR2FWNCwaBiKL6suIOrxJig8LcSBbmCbyYsayia_DvPOOBlXZ4CAlQ5nlDD8kTaIDRwrN/pub

*   `mnist_*.csv` is a small sample of the
    [MNIST database](https://en.wikipedia.org/wiki/MNIST_database), which is
    described at: http://yann.lecun.com/exdb/mnist/

*   `anscombe.json` contains a copy of
    [Anscombe's quartet](https://en.wikipedia.org/wiki/Anscombe%27s_quartet); it
    was originally described in

    Anscombe, F. J. (1973). 'Graphs in Statistical Analysis'. American
    Statistician. 27 (1): 17-21. JSTOR 2682899.

    and our copy was prepared by the
    [vega_datasets library](https://github.com/altair-viz/vega_datasets/blob/4f67bdaad10f45e3549984e17e1b3088c731503d/vega_datasets/_data/anscombe.json).