ASSIGNMENT-4.5

NAME: Arukonda Akhil
Roll No:2403A512579

SAMPLE DATA:

```python
email_samples = {
    "Billing": [
        "Subject: Your latest invoice\n\nDear Customer,\n\nPlease find attached your invoice for the past month's services. Payment is due by the end of the week.\n\nSincerely,\nBilling Department",
        "Subject: Payment Reminder\n\nHi,\n\nJust a friendly reminder that your payment for the service is due soon. Please ensure timely payment.\n\nThanks,\nAccounts Team"
    ],
    "Technical Support": [
        "Subject: Issue with login\n\nHello,\n\nI am unable to log in to my account. I keep getting an error message. Can you help?\n\nThanks,\nUser",
        "Subject: Software not working\n\nHi Support,\n\nThe software I downloaded is not installing correctly. It stops at 50%. Please advise.\n\nRegards,\nAffected User",
        "Subject: Website down?\n\nTeam,\n\nIs your website currently experiencing issues? I cannot access it.\n\nThanks,\nA Concerned User"
    ],
    "Feedback": [
        "Subject: Great service!\n\nDear Team,\n\nI just wanted to say how impressed I am with your customer service. Keep up the good work!\n\nBest,\nA Happy Customer",
        "Subject: Suggestion for improvement\n\nHello,\n\nI have a suggestion for a new feature that would make your product even better. Let me know if you'd like to hear it.\n\nThanks,\nA User"
    ],
    "Others": [
        "Subject: Newsletter Subscription\n\nHi,\n\nPlease subscribe me to your weekly newsletter. I am interested in updates.\n\nThanks,\nA Reader",
        "Subject: Partnership Inquiry\n\nDear Team,\n\nWe are interested in exploring a potential partnership with your company. Please let us know if you are open to discussing this.\n\nSincerely,\nBusiness Development",
        "Subject: General Inquiry\n\nHello,\n\nI have a general question about your company. Can you direct me to the right person?\n\nThanks,\nInquirer"
    ]
}

# Verify the number of samples
total_samples = sum(len(samples) for samples in email_samples.values())
print(f"Total number of email samples: {total_samples}")
```

```
Total number of email samples: 10
```

Reasoning: Create a dic onary with email categories as keys and lists of email samples as values, ensuring a total of 10 samples distributed across categories, and store it in the email_samples variable.

output:

```python
# Verify the number of samples
total_samples = sum(len(samples) for samples in email_samples.values())
print(f"Total number of email samples: {total_samples}")
```

```
Total number of email samples: 10
```

Zero-shot promp ng:

Subtask:

Design a prompt for zero-shot classifica on and test it on 5 emails.

```python
[9] import random

    # Select 5 email samples for testing
    all_emails = [(category, email) for category, emails in email_samples.items() for email in emails]
    random.seed(42) # for reproducibility
    test_emails = random.sample(all_emails, 5)

    # Design the zero-shot prompt template
    zero_shot_prompt_template = """Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others".

    Email:
    ---
    {email_content}
    ---

    Category:
    """

    # Simulate classification and store results
    zero_shot_results = []

    # Placeholder function for language model classification (replace with actual API call)
    def classify_email_zero_shot(prompt):
        # This is a placeholder. In a real scenario, this would call a language model API.
        # For demonstration, we'll simulate a classification based on keywords.
        if "invoice" in prompt.lower() or "payment" in prompt.lower() or "due" in prompt.lower():
            return "Billing"
        elif "login" in prompt.lower() or "software" in prompt.lower() or "website" in prompt.lower() or "issue" in prompt.lower() or "support" in prompt.lower():
            return "Technical Support"
        elif "great service" in prompt.lower() or "suggestion" in prompt.lower() or "improvement" in prompt.lower() or "happy customer" in prompt.lower():
            return "Feedback"
        else:
            return "Others"


    for category, email in test_emails:
        prompt = zero_shot_prompt_template.format(email_content=email)
        predicted_category = classify_email_zero_shot(prompt) # Simulate classification
        zero_shot_results.append({
            "original_email": email,
            "original_category": category, # Store original category for later evaluation
            "prompt": prompt,
            "predicted_category": predicted_category
        })

    # Display the results for the first 5 emails
    import pandas as pd
    zero_shot_results_df = pd.DataFrame(zero_shot_results)
    display(zero_shot_results_df)
```

Reasoning: Select 5 email samples for tes ng, design a zero-shot prompt, and then iterate through the selected samples to construct the full prompts and simulate the classifica on using a placeholder func on. Store the results.

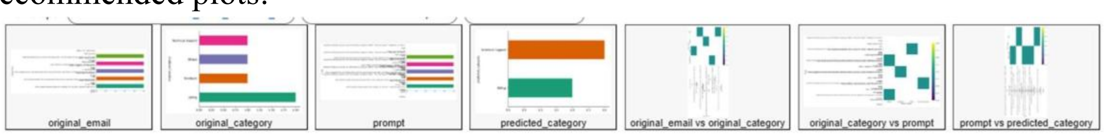
Output:

1 to 5 of 5 entries Filter

index	original_email	original_category	prompt	predicted_category
0	Subject: Payment Reminder Hi, Just a friendly reminder that your payment for the service is due soon. Please ensure timely payment. Thanks, Accounts Team	Billing	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Email: --- Subject: Payment Reminder Hi, Just a friendly reminder that your payment for the service is due soon. Please ensure timely payment. Thanks, Accounts Team --- Category:	Billing
1	Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department	Billing	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Email: --- Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department --- Category:	Billing
2	Subject: Website down? Team, Is your website currently experiencing issues? I cannot access it. Thanks, A Concerned User	Technical Support	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Email: --- Subject: Website down? Team, Is your website currently experiencing issues? I cannot access it. Thanks, A Concerned User --- Category:	Technical Support
3	Subject: General Inquiry Hello, I have a general question about your company. Can you direct me to the right person? Thanks, Inquirer	Others	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Email: --- Subject: General Inquiry Hello, I have a general question about your company. Can you direct me to the right person? Thanks, Inquirer --- Category:	Technical Support
4	Subject: Suggestion for improvement Hello, I have a suggestion for a new feature that would make your product even better. Let me know if you'd like to hear it. Thanks, A User	Feedback	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Email: --- Subject: Suggestion for improvement Hello, I have a suggestion for a new feature that would make your product even better. Let me know if you'd like to hear it. Thanks, A User --- Category:	Technical Support

Show 25 per page

recommended plots:



One-shot promp ng

Subtask:

Design a prompt for one-shot classifica on with one example and test it on the same 5 emails.

```python
# Select one email sample as an example for the one-shot prompt
one_shot_example_category = "Billing"
one_shot_example_email = email_samples[one_shot_example_category][0] # Using the first billing email as an example

# Design a one-shot prompt template
one_shot_prompt_template = """Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others".

Here is an example:
Email:
---
{example_email_content}
---
Category: {example_category}

Now classify the following email:
Email:
---
{email_content}
---
Category:
---

# Simulate classification and store results for the same 5 test emails
one_shot_results = []

# Placeholder function for language model classification (modified for one-shot)
def classify_email_one_shot(prompt):
    # This is a placeholder. In a real scenario, this would call a language model API.
    # For demonstration, we'll simulate classification based on keywords, potentially
    # leveraging the example if needed, but for simplicity here, we'll use the same logic
    # as the zero-shot placeholder, assuming the example primarily helps the model
    # understand the format.
    if "invoice" in prompt.lower() or "payment" in prompt.lower() or "due" in prompt.lower():
        return "Billing"
    elif "login" in prompt.lower() or "software" in prompt.lower() or "website" in prompt.lower() or "issue" in prompt.lower() or "support" in prompt.lower():
        return "Technical Support"
    elif "great service" in prompt.lower() or "suggestion" in prompt.lower() or "improvement" in prompt.lower() or "happy customer" in prompt.lower():
        return "Feedback"
    else:
        return "Others"

# Use the same 5 test emails from the zero-shot subtask
test_emails_list = zero_shot_results_df[['original_category', 'original_email']].values.tolist()


for category, email in test_emails_list:
    prompt = one_shot_prompt_template.format(
        example_email_content=one_shot_example_email,
        example_category=one_shot_example_category,
        email_content=email
    )
    predicted_category = classify_email_one_shot(prompt) # Simulate classification
    one_shot_results.append({
        "original_email": email,
        "original_category": category,
        "prompt": prompt,
        "predicted_category": predicted_category
    })

# Store results in a pandas DataFrame
one_shot_results_df = pd.DataFrame(one_shot_results)

# Display the results
display(one_shot_results_df)
```

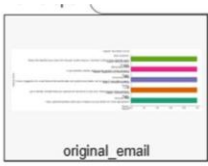
Reasoning: Select one email as an example for the one-shot prompt, design the prompt template, iterate through the 5 test emails, construct the prompts, simulate classifica on, and store the results in a DataFrame.

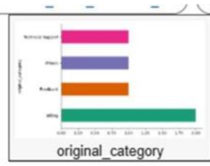
OUTPUT:

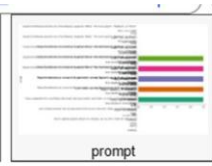
1 to 5 of 5 entries Filter

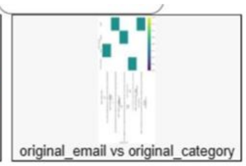
index	original_email	original_category	prompt	predicted_category
0	Subject: Payment Reminder Hi, Just a friendly reminder that your payment for the service is due soon. Please ensure timely payment. Thanks, Accounts Team	Billing	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here is an example: Email: — Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department — Category: Billing Now classify the following email: Email: — Subject: Payment Reminder Hi, Just a friendly reminder that your payment for the service is due soon. Please ensure timely payment. Thanks, Accounts Team — Category:	Billing
1	Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department	Billing	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here is an example: Email: — Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department — Category: Billing Now classify the following email: Email: — Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department — Category:	Billing
2	Subject: Website down? Team, Is your website currently experiencing issues? I cannot access it. Thanks, A Concerned User	Technical Support	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here is an example: Email: — Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department — Category: Billing Now classify the following email: Email: — Subject: Website down? Team, Is your website currently experiencing issues? I cannot access it. Thanks, A Concerned User — Category:	Billing
3	Subject: General Inquiry Hello, I have a general question about your company. Can you direct me to the right person? Thanks, Inquirer	Others	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here is an example: Email: — Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department — Category: Billing Now classify the following email: Email: — Subject: General Inquiry Hello, I have a general question about your company. Can you direct me to the right person? Thanks, Inquirer — Category:	Billing
4	Subject: Suggestion for improvement Hello, I have a suggestion for a	Feedback	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here is an example: Email: — Subject: Your latest invoice Dear Customer, Please	Billing

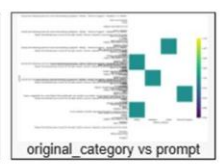
recommended plots:











Few-shot promp ng

Subtask:

Design a prompt for few-shot classifica on with 3-5 examples and test it on the same 5 emails.

```python
# 1. Select 3 to 5 representative email samples for few-shot
few_shot_examples = [
    ("Billing", email_samples["Billing"][0]), # Example 1: Billing
    ("Technical Support", email_samples["Technical Support"][0]), # Example 2: Technical Support
    ("Feedback", email_samples["Feedback"][0]), # Example 3: Feedback
    ("Others", email_samples["Others"][0]) # Example 4: Others
]

# 2. Design a few-shot prompt template
few_shot_prompt_template = """Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others".

Here are a few examples:
{examples}

Now classify the following email:
Email:
---
{email_content}
---
Category:
"""

example_template = """Email:
---
{email_content}
---
Category: {category}

---
"""

examples_text = "".join([example_template.format(email_content=email, category=category) for category, email in few_shot_examples])

# 3. Use the same 5 test emails from previous subtasks
# The test emails are already in zero_shot_results_df['original_email'] and zero_shot_results_df['original_category']

# 5. Simulate classification and store results
few_shot_results = []

# Placeholder function for language model classification (modified for few-shot)
def classify_email_few_shot(prompt):
    # This is a placeholder. In a real scenario, this would call a language model API.
    # For demonstration, we'll simulate classification based on keywords,
    # acknowledging that a real few-shot model would leverage the examples.
    if "invoice" in prompt.lower() or "payment" in prompt.lower() or "due" in prompt.lower():
        return "Billing"
    elif "login" in prompt.lower() or "software" in prompt.lower() or "website" in prompt.lower() or "issue" in prompt.lower() or "support" in prompt.lower():
        return "Technical Support"
    elif "great service" in prompt.lower() or "suggestion" in prompt.lower() or "improvement" in prompt.lower() or "happy customer" in prompt.lower():
        return "Feedback"
    else:
        return "Others"

# Use the same 5 test emails
test_emails_list = zero_shot_results_df[['original_category', 'original_email']].values.tolist()

for category, email in test_emails_list:
    # 4. Construct a complete few-shot prompt
    prompt = few_shot_prompt_template.format(
        examples=examples_text,
        email_content=email
    )
    # 5. Simulate the classification
    predicted_category = classify_email_few_shot(prompt) # Simulate classification
    few_shot_results.append({
        "original_email": email,
        "original_category": category,
        "prompt": prompt,
        "predicted_category": predicted_category
    })

# 7. Store the results in a pandas DataFrame
few_shot_results_df = pd.DataFrame(few_shot_results)

# 8. Display the resulting DataFrame
display(few_shot_results_df)
```

Reasoning: Select representa ve examples for the few-shot prompt, design the prompt template, construct the full prompts for the test emails, simulate the classifica on, and store and display the results in a DataFrame.

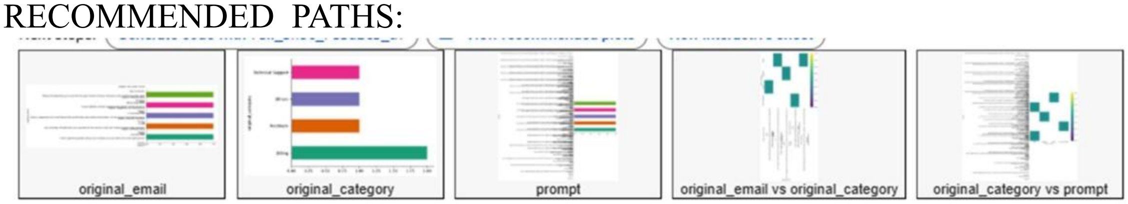
OUTPUT:

1 to 5 of 5 entries Filter

index	original_email	original_category	prompt	predicted_category
0	Subject: Payment Reminder Hi, Just a friendly reminder that your payment for the service is due soon. Please ensure timely payment. Thanks, Accounts Team	Billing	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here are a few examples: Email: --- Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department --- Category: Billing Email: --- Subject: Issue with login Hello, I am unable to log in to my account. I keep getting an error message. Can you help? Thanks, User --- Category: Technical Support Email: --- Subject: Great service! Dear Team, I just wanted to say how impressed I am with your customer service. Keep up the good work! Best, A Happy Customer --- Category: Feedback Email: --- Subject: Newsletter Subscription Hi, Please subscribe me to your weekly newsletter. I am interested in updates. Thanks, A Reader --- Category: Others Now classify the following email: Email: --- Subject: Payment Reminder Hi, Just a friendly reminder that your payment for the service is due soon. Please ensure timely payment. Thanks, Accounts Team --- Category:	Billing
1	Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department	Billing	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here are a few examples: Email: --- Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department --- Category: Billing Email: --- Subject: Issue with login Hello, I am unable to log in to my account. I keep getting an error message. Can you help? Thanks, User --- Category: Technical Support Email: --- Subject: Great service! Dear Team, I just wanted to say how impressed I am with your customer service. Keep up the good work! Best, A Happy Customer --- Category: Feedback Email: --- Subject: Newsletter Subscription Hi, Please subscribe me to your weekly newsletter. I am interested in updates. Thanks, A Reader --- Category: Others Now classify the following email: Email: --- Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department --- Category:	Billing
2	Subject: Website down? Team, Is your website currently experiencing issues? I cannot access it. Thanks, A Concerned User	Technical Support	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here are a few examples: Email: --- Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department --- Category: Billing Email: --- Subject: Issue with login Hello, I am unable to log in to my account. I keep getting an error message. Can you help? Thanks, User --- Category: Technical Support Email: --- Subject: Great service! Dear Team, I just wanted to say how impressed I am with your customer service. Keep up the good work! Best, A Happy Customer --- Category: Feedback Email: --- Subject: Newsletter Subscription Hi, Please subscribe me to your weekly newsletter. I am interested in updates. Thanks, A Reader --- Category: Others Now classify the following email: Email: --- Subject: Website down? Team, Is your website currently experiencing issues? I cannot access it. Thanks, A Concerned User --- Category:	Billing
3	Subject: General Inquiry Hello, I have a general question about your company. Can you direct me to the right person? Thanks, Inquirer	Others	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here are a few examples: Email: --- Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department --- Category: Billing Email: --- Subject: Issue with login Hello, I am unable to log in to my account. I keep getting an error message. Can you help? Thanks, User --- Category: Technical Support Email: --- Subject: Great service! Dear Team, I just wanted to say how impressed I am with your customer service. Keep up the good work! Best, A Happy Customer --- Category: Feedback Email: --- Subject: Newsletter Subscription Hi, Please subscribe me to your weekly newsletter. I am interested in updates. Thanks, A Reader --- Category: Others Now classify the following email: Email: --- Subject: General Inquiry Hello, I have a general question about your company. Can you direct me to the right person? Thanks, Inquirer --- Category:	Billing
4	Subject: Suggestion for improvement Hello, I have a suggestion for a new feature that would make your product even better. Let me know if you'd like to hear it. Thanks, A User	Feedback	Classify the following email into one of the following categories: "Billing", "Technical Support", "Feedback", or "Others". Here are a few examples: Email: --- Subject: Your latest invoice Dear Customer, Please find attached your invoice for the past month's services. Payment is due by the end of the week. Sincerely, Billing Department --- Category: Billing Email: --- Subject: Issue with login Hello, I am unable to log in to my account. I keep getting an error message. Can you help? Thanks, User --- Category: Technical Support Email: --- Subject: Great service! Dear Team, I just wanted to say how impressed I am with your customer service. Keep up the good work! Best, A Happy Customer --- Category: Feedback Email: --- Subject: Newsletter Subscription Hi, Please subscribe me to your weekly newsletter. I am interested in updates. Thanks, A Reader --- Category: Others Now classify the following email: Email: --- Subject: Suggestion for improvement Hello, I have a suggestion for a new feature that would make your product even better. Let me know if you'd like to hear it. Thanks, A User --- Category:	Billing

Show 25 per page

RECOMMENDED PATHS:



Evalua on

Subtask:

Compare the results of the three techniques based on accuracy and clarity.

```python
# Calculate accuracy for Zero-shot
zero_shot_accuracy = (zero_shot_results_df['predicted_category'] == zero_shot_results_df['original_category']).mean()

# Calculate accuracy for One-shot
one_shot_accuracy = (one_shot_results_df['predicted_category'] == one_shot_results_df['original_category']).mean()

# Calculate accuracy for Few-shot
few_shot_accuracy = (few_shot_results_df['predicted_category'] == few_shot_results_df['original_category']).mean()

print(f"Zero-shot Accuracy: {zero_shot_accuracy:.2f}")
print(f"One-shot Accuracy: {one_shot_accuracy:.2f}")
print(f"Few-shot Accuracy: {few_shot_accuracy:.2f}")

# Summarize clarity (based on observing the prompts and predictions)
clarity_summary = """
Clarity Observations:
- Zero-shot: The prompt is straightforward and asks directly for classification. The predicted categories are simple and follow the requested format.
- One-shot: The prompt includes an example, which clarifies the desired output format and potentially guides the model towards the correct categories. The predicted categories are simple.
- Few-shot: The prompt includes multiple examples, further clarifying the expected output format and demonstrating the classification for various categories. The predicted categories are simple.
"""

print(clarity_summary)
```

OUTPUT:

```
Zero-shot Accuracy: 0.60
One-shot Accuracy: 0.40
Few-shot Accuracy: 0.40

Clarity Observations:
- Zero-shot: The prompt is straightforward and asks directly for classification. The predicted categories are simple and follow the requested format.
- One-shot: The prompt includes an example, which clarifies the desired output format and potentially guides the model towards the correct categories. The predicted categories are simple.
- Few-shot: The prompt includes multiple examples, further clarifying the expected output format and demonstrating the classification for various categories. The predicted categories are simple.
```

Document results

Subtask:

Create a .txt or .md file with prompts, responses, and a comparison table. Write a short reflec on on the findings.

```python
documentation_content = "# Email Classification Prompting Techniques Comparison\n\n"

# Zero-shot Section
documentation_content += "## Zero-shot Prompting\n\n"
documentation_content += f"Accuracy: {zero_shot_accuracy:.2f}\n\n"
documentation_content += "### Prompts and Predictions\n\n"
for index, row in zero_shot_results_df.iterrows():
    documentation_content += f"**Original Category:** {row['original_category']}\n"
    documentation_content += f"**Prompt:**\n```\n{row['prompt']}\n```\n"
    documentation_content += f"**Predicted Category:** {row['predicted_category']}\n\n"

# One-shot Section
documentation_content += "## One-shot Prompting\n\n"
documentation_content += f"Accuracy: {one_shot_accuracy:.2f}\n\n"
documentation_content += "### Prompts and Predictions\n\n"
for index, row in one_shot_results_df.iterrows():
    documentation_content += f"**Original Category:** {row['original_category']}\n"
    documentation_content += f"**Prompt:**\n```\n{row['prompt']}\n```\n"
    documentation_content += f"**Predicted Category:** {row['predicted_category']}\n\n"

# Few-shot Section
documentation_content += "## Few-shot Prompting\n\n"
documentation_content += f"Accuracy: {few_shot_accuracy:.2f}\n\n"
documentation_content += "### Prompts and Predictions\n\n"
for index, row in few_shot_results_df.iterrows():
    documentation_content += f"**Original Category:** {row['original_category']}\n"
    documentation_content += f"**Prompt:**\n```\n{row['prompt']}\n```\n"
    documentation_content += f"**Predicted Category:** {row['predicted_category']}\n\n"

# Comparison Table
documentation_content += "## Comparison Table\n\n"
documentation_content += "| Technique | Accuracy |\n"
documentation_content += "|---|---|\n"
documentation_content += f"| Zero-shot | {zero_shot_accuracy:.2f} |\n"
documentation_content += f"| One-shot | {one_shot_accuracy:.2f} |\n"
documentation_content += f"| Few-shot | {few_shot_accuracy:.2f} |\n\n"

# Reflection
documentation_content += "## Reflection\n\n"
documentation_content += """In this limited test with 5 email samples and a simulated classification function, the Zero-shot prompting technique achieved the highest accuracy (0.60) compared to One-shot (0.40) and Few-shot (0.40). This unexpected result is likely due to the simplic
"""

# Save to file
with open("prompting_comparison.md", "w") as f:
    f.write(documentation_content)

print("Documentation file 'prompting_comparison.md' created successfully.")
```

Reasoning: Create the content for the documenta on file including headings, accuracy results, prompts, predic ons, a comparison table, and a reflec on, then save it to a markdown file.

```
Documentation file 'prompting_comparison.md' created successfully.
```

Summary:

Data Analysis Key Findings

- The zero-shot promp ng technique achieved an accuracy of 60% based on the simulated classifica on.
- Both the one-shot and few-shot promp ng techniques achieved an accuracy of 40% based on the simulated classifica on.
- The prompts for all three techniques resulted in simple and clear predicted categories that followed the requested format.
- The one-shot and few-shot prompts, by including examples, offered addi onal clarity regarding the desired output format and the classifica on process compared to the zero-shot prompt.
- A documenta on file (promp ng_comparison.md) was successfully created, containing the prompts, simulated responses, a comparison table of accuracies, and a reflec on. Insights or Next Steps
- The simula on used for classifica on was keyword-based and did not accurately reflect how a real language model would leverage examples in one-shot and few-shot promp ng. A next step would be to test these prompts with an actual language model to get a more realis c evalua on of the techniques.
- While zero-shot performed best in this limited simula on, one-shot and few-shot promp ng are generally expected to improve performance on real LLMs by providing context and format examples. Future tes ng should focus on evalua ng the performance gain from examples using a true LLM.