```
Start coding or generate with AI.
```

```
import pandas as pd
import random
from sklearn.model_selection import train_test_split

# Load the datasets
fake_df = pd.read_csv("Fake.csv")
true_df = pd.read_csv("True.csv")

# Add a label column
fake_df['label'] = 'Fake'
true_df['label'] = 'Real'

# Combine the datasets
data = pd.concat([fake_df[['title', 'text', 'label']], true_df[['title', 'text', 'label']]], ignore_index=True)

# Shuffle the data
data = data.sample(frac=1).reset_index(drop=True)

# Split into training, validation, and test sets
train, temp = train_test_split(data, test_size=0.4, random_state=42)
val, test = train_test_split(temp, test_size=0.5, random_state=42)

print(f"Training size: {len(train)}, Validation size: {len(val)}, Test size: {len(test)}")
```

```
Training size: 26938, Validation size: 8980, Test size: 8980
```

```
import random

def prepare(n):

    global train

    # Randomly sample n rows
    examples = []
    sampled_rows = train.sample(n)  # Randomly sample n rows
    for _, row in sampled_rows.iterrows():
        examples.append(f"{row['title']} => {row['label']}")
    return "\n".join(examples)

# Example:
prepared_examples = prepare(3)

# Display the prepared examples
print("Prepared Examples:\n")
print(prepared_examples)
```

```
Prepared Examples:

 Trump to host Thai prime minister on October 2: White House => Real
 The U.S. Establishment vs The Rest of World => Fake
 End 'containment' of asylum-seekers on islands, aid groups tell Greek PM => Real
```

```
def generate_prompt(n, example_row):

    # Generate a prompt with n examples from the training set and one example from the validation/test
    # Prepare n examples from the training set
    few_shot_examples = prepare(n)

    # Append the example from the validation/test set
    validation_example = f"{example_row['title']} =>"

    # Combine into a single prompt
    prompt = f"{few_shot_examples}\n{validation_example}"
    return prompt

# Example
example_row = val.iloc[0]  # Take the first row from the validation set
n = 3  # Number of training examples to include
prompt = generate_prompt(n, example_row)
```

```
# Display the generated prompt
print("Generated Prompt:\n")
print(prompt)
```

```
Generated Prompt:

     ‘SLEEPY’ JUSTICE GINSBURG: Excites Crowd By Saying She’d Back Abolition Of The Electoral College [Video] => Fake
      It’s Not Over Yet: Jill Stein Files Federal Lawsuit Over Pennsylvania Recount => Fake
     Trump meets insurers, promises catastrophic year for Obamacare => Real
     Lockheed Martin wins $582 million U.S. defense contract: Pentagon =>
```

```
from transformers import GPT2Tokenizer, GPT2LMHeadModel
import torch

# Load GPT-2 model and tokenizer
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
model = GPT2LMHeadModel.from_pretrained("gpt2")

def gpt2_generate(prompt, max_tokens=1):
    #Generate a prediction using GPT-2 based on the given prompt.

    # Encode the prompt
    inputs = tokenizer.encode(prompt, return_tensors="pt")

    # Generate response
    outputs = model.generate(
        inputs,
        max_new_tokens=max_tokens,  # Number of tokens to generate
        do_sample=True,  # sampling for randomness
        temperature=0.7,  # control randomness
        top_k=50,  # top-k sampling for diversity
        pad_token_id=tokenizer.eos_token_id  # set pad_token_id to eos_token_id
    )

    # Decode and return the output
    return tokenizer.decode(outputs[0][inputs.size(-1):], skip_special_tokens=True).strip()

# Example
example_prompt = generate_prompt(3, val.iloc[0])  # Generate a prompt with 3 examples from training set
prediction = gpt2_generate(example_prompt)

# Display the prediction
print("Generated Prediction:\n")
print(prediction)
```

```
Generated Prediction:

     Fake
```

```
from sklearn.metrics import classification_report

def score(n, dataset):

    #Scores GPT-2's performance on a given dataset using n examples for few-shot training.

    # Store predictions and true labels
    predictions = []
    ground_truths = []

    # Iterate through each row in the dataset
    for index, row in dataset.iterrows():
        # Generate the prompt
        prompt = generate_prompt(n, row)

        # Generate the prediction
        prediction = gpt2_generate(prompt)

        # Print the prompt and prediction
        print(f"Prompt:\n{prompt}\nPrediction: {prediction}\n")

        # Append the prediction and ground truth
        predictions.append(prediction)
        ground_truths.append(row['label'])
```

```
    # return classification metrics
    metrics = classification_report(ground_truths, predictions, output_dict=True, zero_division=0)
    return metrics

# Reduce the dataset for speed
small_val = val.sample(n=10, random_state=42)

# Example usage with reduced dataset
n = 3
metrics = score(n, small_val)
print(f"Validation Metrics for n = {n}:\n", metrics)
```

```
EP #17: Patrick Henningsen LIVE – ‘Parallax Politics in DC’ with guest Daniel Faraci => Fake
 Trump Lawyer LITERALLY Argued That Muslim Ban Shouldn’t Be Based In Reality, And Twitter Is LOSING IT (VIDEO/TWEETS) => Fake
Former House Speaker Hastert seeks sentencing delay due to poor health =>
Prediction: Fake

Prompt:
 Trump’s Deplorables Freak Out, Demand We Repeal Women’s Right To Vote (TWEETS) => Fake
LEBRON JAMES Brags About Being “Well-Spoken”…Proceeds To Use Vocabulary Of A 4-Yr-Old To Trash President Trump => Fake
Tennessee lawmaker ousted after sex harassment allegations: media => Real
EU drone rules in balance as member states hold off endorsing deal =>
Prediction: Fake

Prompt:
Spain arrests four former Venezuelan officials for U.S. probe => Real
Crowd controversy: The making of an Inauguration Day photo => Real
Sessions confirmed as U.S. attorney general after battle with Democrats => Real
THE JACK BLOOD SHOW: ‘From May Day Riots to Globalism’ with 21WIRE guest Shawn Helton =>
Prediction: @

Prompt:
CNBC DEBATE HACK Proves Allegiance To Democrat Party With This Tweet Celebrating Mass Murder => Fake
OUCH! BERNIE SANDERS Responds To Hillary’s Criticism…Slams Her With Category-5 Zinger [VIDEO] => Fake
 Former KKK Leader Thanks Trump’s VP For Being So Nice (AUDIO) => Fake
Democrats link U.S. debt limit vote to Republican tax cut moves =>
Prediction: Obama

Prompt:
Wheelbarrow bomb kills man pushing it in Somalia’s Puntland, police say => Real
Trump says Brexit to be 'a great thing', wants quick trade deal with UK => Real
Elite Nazi-allied Order From Hungary Claims Trump Adviser Sebastian Gorka Is Sworn Member => Fake
SYRIAN MUSLIM MAN WHOSE Family Perished On Trip So He Could Get Free Dental Care Has New Spokesperson Role =>
Prediction: Real

Prompt:
SCOUNDREL HILLARY SUPPORTER STARTS “TrumpLeaks” Campaign…Desperate Move! => Fake
U.S. tax debate employing more than half of Washington's lobbyists: report => Real
[VIDEO] OUR RACIST PRESIDENT INVITES MUSLIMS To Join Blacks In Victim Pool While Celebrating Ramadan at White House => Fake
 Elizabeth Warren Calls Trump A ‘Two-Bit Dictator,’ Slams His RNC Speech =>
Prediction: Real

Prompt:
Philippine president's Senate foes, allies vow to block budget cut for rights body => Real
GERMANY’S DEFENSE MINISTER Refuses To Wear Hijab During Saudi Arabia Visit…Says Trump’s Election Proves Political Correctness Has B
WOW! BRITISH ACTRESS HAMMERS EU Leaders: “Every one of you who said refugees are welcome, You Are Responsible For Brussels…Europe i
 Bernie Sanders Just Received Some Excellent News About His Campaign =>
Prediction: Fake

Prompt:
LT GEN JOHN KELLY’S INCREDIBLE SPEECH Just Days After His Own Son Was Killed In Action: “Brave Men On Watch All Over the World Toni
Greg Gutfeld Scores a Priceless TV Moment…Wears CNN Box During ‘The Five’ [Video] => Fake
 Those Intel Reports That Nunes Ran To Trump With Reveal A BOMBSHELL That Trump Will HATE => Fake
Tight race as South Africa's ANC prepares to elect Zuma successor =>
Prediction: Fake

Validation Metrics for n = 3:
 {'@': {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 0.0}, 'Fake': {'precision': 0.3333333333333333, 'recall': 0.4,
```

```
# Evaluate performance for n in a range
results = {}
range_n = range(1, 6)  # n (1 to 5)

# compute validation metrics
for n in range_n:
    print(f"Evaluating for n = {n}")
    metrics = score(n, small_val)
    results[n] = metrics
```

```
        # accuracy, precision, recall, and F1-score
        fake_metrics = metrics.get("Fake", {"precision": 0, "recall": 0, "f1-score": 0})
        real_metrics = metrics.get("Real", {"precision": 0, "recall": 0, "f1-score": 0})
        accuracy = metrics.get("accuracy", 0)

        # display summary for this value of n
        print(f"Summary for n = {n}:")
        print(f"  Accuracy: {accuracy:.2f}")
        print(f"  Fake News - Precision: {fake_metrics['precision']:.2f}, Recall: {fake_metrics['recall']:.2f}, F1-Score: {fake_metrics['f1-score
        print(f"  Real News - Precision: {real_metrics['precision']:.2f}, Recall: {real_metrics['recall']:.2f}, F1-Score: {real_metrics['f1-score
        print("-" * 40)

    # Identify the best n based on F1-score (weighted avg)
    optimal_n = max(results, key=lambda n: results[n]["weighted avg"]["f1-score"])
    print(f"\nOptimal n based on validation performance: {optimal_n}")

    # Evaluate on the test set with the optimal n
    print(f"\nEvaluating on test set with n = {optimal_n}")
    test_metrics = score(optimal_n, small_test)  # Use a reduced test set for speed

    # test results
    test_fake_metrics = test_metrics.get("Fake", {"precision": 0, "recall": 0, "f1-score": 0})
    test_real_metrics = test_metrics.get("Real", {"precision": 0, "recall": 0, "f1-score": 0})
    test_accuracy = test_metrics.get("accuracy", 0)

    # display summary
    print("\nTest Set Summary:")
    print(f"  Accuracy: {test_accuracy:.2f}")
    print(f"  Fake News - Precision: {test_fake_metrics['precision']:.2f}, Recall: {test_fake_metrics['recall']:.2f}, F1-Score: {test_fake_metric
    print(f"  Real News - Precision: {test_real_metrics['precision']:.2f}, Recall: {test_real_metrics['recall']:.2f}, F1-Score: {test_real_metric
    print("-" * 40)
```

```
 This Kansas Republican Thought Touting 'Profound' Hitler Quote Was A Good Idea - It Wasn't =>
Prediction: Fake


Test Set Summary:
  Accuracy: 0.50
  Fake News - Precision: 0.63, Recall: 0.63, F1-Score: 0.63
  Real News - Precision: 0.38, Recall: 0.30, F1-Score: 0.33
---------------------------------------
```

Summary for n = 1: Accuracy: 0.30 Fake News - Precision: 0.67, Recall: 0.40, F1-Score: 0.50 Real News - Precision: 1.00, Recall: 0.20, F1-Score: 0.33

Summary for n = 2: Accuracy: 0.40 Fake News - Precision: 0.50, Recall: 0.20, F1-Score: 0.29 Real News - Precision: 0.75, Recall: 0.60, F1-Score: 0.67

Summary for n = 3: Accuracy: 0.40 Fake News - Precision: 0.75, Recall: 0.60, F1-Score: 0.67 Real News - Precision: 0.33, Recall: 0.20, F1-Score: 0.25

Summary for n = 4: Accuracy: 0.40 Fake News - Precision: 0.50, Recall: 0.40, F1-Score: 0.44 Real News - Precision: 0.50, Recall: 0.40, F1-Score: 0.44

Summary for n = 5: Accuracy: 0.60 Fake News - Precision: 0.67, Recall: 0.40, F1-Score: 0.50 Real News - Precision: 0.57, Recall: 0.80, F1-Score: 0.67

Optimal n based on validation performance: 5

Evaluating on test set with n = 5

Test Set Summary: Accuracy: 0.50 Fake News - Precision: 0.63, Recall: 0.63, F1-Score: 0.63 Real News - Precision: 0.38, Recall: 0.30, F1-Score: 0.33

Part 2:

1. What I usually find works well is trimming the description of the problem to contain only essential requirements and break them into numbered steps. Specify the input and output and provide a sample input/output, also specify an approach, for example:"solve using Greedy/DP if ideal"
2. GPT often generate good stories and high quality cover letters if the prompt includes details and personal backgrounds, because many existing dataset/patterns already exists. It's not very accurate at mathematical proof because unless we are proving something that have already been proved many times before in textbooks, it's hard to follow a strict,precise logic without generating extra/creative texts that may not be suited for a proof.