classify emails using LLM

```
# Import Libraries
import pandas as pd
from transformers import pipeline
from tqdm import tqdm

#  Load and Prepare Dataset
df = pd.read_csv("spam.csv", encoding="latin-1")[["text", "target"]]
df = df.sample(n=100, random_state=42).reset_index(drop=True)  # sample 100 for performance
df["text"] = df["text"].astype(str).str.replace(r'\s+', ' ', regex=True).str.strip()

# Load Zero-Shot Classification Pipeline
classifier = pipeline("zero-shot-classification", model="facebook/bart-large-mnli")

# Define Candidate Labels
labels = ["spam", "ham"]

# Apply Model to Classify Emails
predictions = []
for text in tqdm(df["text"]):
    result = classifier(text[:512], candidate_labels=labels)
    predictions.append(result["labels"][0])  # top label

# Store Predictions
df["predicted_label"] = predictions

# Evaluate
from sklearn.metrics import accuracy_score, classification_report

df["target"] = df["target"].str.lower()
print("Accuracy:", accuracy_score(df["target"], df["predicted_label"]))
print(classification_report(df["target"], df["predicted_label"]))
df.head()
```

```
config.json:       1.15k/?  [00:00<00:00,  20.6kB/s]
model.safetensors:  100%       1.63G/1.63G  [00:53<00:00,  46.3MB/s]
tokenizer_config.json:  100%   26.0/26.0  [00:00<00:00,  2.08kB/s]
vocab.json:        899k/?  [00:00<00:00,  5.68MB/s]
merges.txt:        456k/?  [00:00<00:00,  8.50MB/s]
tokenizer.json:    1.36M/?  [00:00<00:00,  17.5MB/s]
Device set to use cpu
100%|██████████| 100/100 [02:39<00:00,  1.59s/it]Accuracy: 0.73
              precision    recall  f1-score   support

         ham       0.90      0.78      0.84        88
        spam       0.17      0.33      0.23        12

    accuracy                           0.73       100
   macro avg       0.54      0.56      0.53       100
weighted avg       0.81      0.73      0.76       100
```

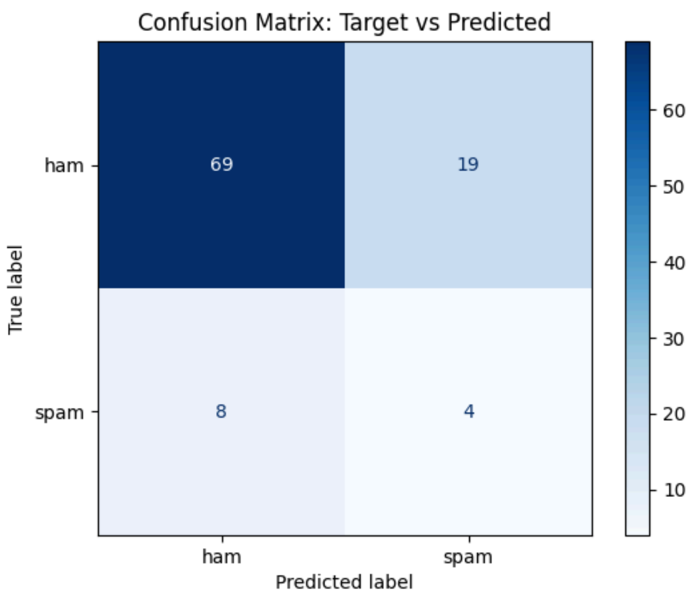
	text	target	predicted_label
0	Funny fact Nobody teaches volcanoes 2 erupt, t...	ham	ham
1	I sent my scores to sophas and i had to do sec...	ham	ham
2	We know someone who you know that fancies you....	spam	spam
3	Only if you promise your getting out as SOON a...	ham	ham
4	Congratulations ur awarded either Ã¥Â£500 of C...	spam	ham

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Compute confusion matrix
cm = confusion_matrix(df["target"], df["predicted_label"], labels=["ham", "spam"])

# Display
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["ham", "spam"])
disp.plot(cmap="Blues")
```

```
plt.title("Confusion Matrix: Target vs Predicted")
plt.show()
```


Confusion Matrix: Target vs Predicted
True label
ham
spam
Predicted label
ham
spam