

```
!pip install transformers datasets
```



```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-package
Collecting datasets
```

```
  Downloading datasets-3.5.0-py3-none-any.whl.metadata (19 kB)
```

```
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (f
```

```
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3
```

```
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages
```

```
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-pack
```

```
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages
```

```
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-pa
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (f
```

```
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/di
```

```
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-p
```

```
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages
```

```
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.11/dist-pack
```

```
Collecting dill<0.3.9,>=0.3.0 (from datasets)
```

```
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (frc
```

```
Collecting xxhash (from datasets)
```

```
  Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

```
Collecting multiprocessing<0.70.17 (from datasets)
```

```
  Downloading multiprocessing-0.70.16-py311-none-any.whl.metadata (7.2 kB)
```

```
Collecting fsspec<=2024.12.0,>=2023.1.0 (from fsspec[http]<=2024.12.0,>=2023.1.0->dat
```

```
  Downloading fsspec-2024.12.0-py3-none-any.whl.metadata (11 kB)
```

```
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (fr
```

```
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/c
```

```
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-pac
```

```
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packag
```

```
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-pa
```

```
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-
```

```
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-pac
```

```
Requirement already satisfied: yarll<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-pa
```

```
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.1
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/
```

```
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-package
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-p
```

```
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-p
```

```
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/di
```

```
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-package
```

```
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packa
```

```
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (f
```

```
Downloading datasets-3.5.0-py3-none-any.whl (491 kB)
```

```
_____ 491.2/491.2 kB 21.4 MB/s eta 0:00:00
```

```
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
```

```
_____ 116.3/116.3 kB 12.5 MB/s eta 0:00:00
```

```
Downloading fsspec-2024.12.0-py3-none-any.whl (183 kB)
```

```
_____ 183.9/183.9 kB 20.1 MB/s eta 0:00:00
```

```
Downloading multiprocessing-0.70.16-py311-none-any.whl (143 kB)
```

```
_____ 143.5/143.5 kB 15.4 MB/s eta 0:00:00
```

```
Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (
```

```
_____ 194.8/194.8 kB 18.1 MB/s eta 0:00:00
```

```
Installing collected packages: xxhash, fsspec, dill, multiprocessing, datasets
```

```
  Attempting uninstall: fsspec
```

```
Attempting uninstall: fsspec
```

```
Found existing installation: fsspec 2025.3.2
```

```
Uninstalling fsspec-2025.3.2:
```

```
Successfully uninstalled fsspec-2025.3.2
```

```
ERROR: pip's dependency resolver does not currently take into account all the package
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2024.12.0 which is incorrec
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8; platform_system == "Linux" a
```

```
import pandas as pd
from datasets import Dataset
from transformers import DistilBertTokenizerFast, DistilBertForSequenceClassification, Tra
from sklearn.model_selection import train_test_split
import torch
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

```
# 1. Load Dataset
```

```
df = pd.read_csv("/content/drive/MyDrive/phishing_email.csv")
print(f"Dataset shape: {df.shape}")
print("\nFirst 5 rows:")
print(df.head())
```

```
df = df[['text_combined', 'label']].dropna()
df['label'] = df['label'].astype(int)
```

```
train_texts, remaining_texts, train_labels, remaining_labels = train_test_split(
    df['text_combined'], df['label'], test_size=0.2, stratify=df['label'], random_state=42)
val_texts, test_texts, val_labels, test_labels = train_test_split(
    remaining_texts, remaining_labels, test_size=0.5, stratify=remaining_labels, random_st
```

```
↗ Dataset shape: (82486, 2)
```

```
First 5 rows:
```

	text_combined	label
0	hpl nom may 25 2001 see attached file hplno 52...	0
1	nom actual vols 24 th forwarded sabrae zajac h...	0
2	enron actuals march 30 april 1 201 estimated a...	0
3	hpl nom may 30 2001 see attached file hplno 53...	0
4	hpl nom june 1 2001 see attached file hplno 60...	0

```
# Step 4: Tokenization
```

```
tokenizer = DistilBertTokenizerFast.from_pretrained('distilbert-base-uncased')
```

```
def tokenize(batch):
```

```
    tokens = tokenizer(batch['text_combined'], padding='max_length', truncation=True, max
    return {'input_ids': tokens['input_ids'], 'attention_mask': tokens['attention_mask']}
```

```
train_ds = Dataset.from_dict({'text_combined': train_texts.tolist(), 'label': train_label
val_ds = Dataset.from_dict({'text_combined': val_texts.tolist(), 'label': val_labels.toli
test_ds = Dataset.from_dict({'text_combined': test_texts.tolist(), 'label': test_labels.t
```

```

train_ds = train_ds.map(tokenize, batched=True).remove_columns(['text_combined'])
val_ds = val_ds.map(tokenize, batched=True).remove_columns(['text_combined'])
test_ds = test_ds.map(tokenize, batched=True).remove_columns(['text_combined'])

```

/usr/local/lib/python3.11/dist-packages/huggingface\_hub/utils/\_auth.py:94: UserWarning  
The secret `HF\_TOKEN` does not exist in your Colab secrets.  
To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>)  
You will be able to reuse this secret in all of your notebooks.  
Please note that authentication is recommended but still optional to access public models.  
warnings.warn(

```

tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 4.73kB/s]

vocab.txt: 100% 232k/232k [00:00<00:00, 11.5MB/s]

tokenizer.json: 100% 466k/466k [00:00<00:00, 2.71MB/s]

config.json: 100% 483/483 [00:00<00:00, 33.0kB/s]

Map: 100% 65988/65988 [01:40<00:00, 996.66 examples/s]

```

# Step 5: Load Model

```
model = DistilBertForSequenceClassification.from_pretrained("distilbert-base-uncased", num_labels=2)
```

```
print(model.config)
```

# Step 6: Training Configuration

```

training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    eval_strategy="epoch",
    save_strategy="epoch",
    logging_dir='./logs',
    logging_steps=100,
    load_best_model_at_end=True,
    metric_for_best_model="eval_loss",
    save_total_limit=1,
    report_to="none"
)

```

)

Xet Storage is enabled for this repo, but the 'hf\_xet' package is not installed. Fall  
 WARNING:huggingface\_hub.file\_download:Xet Storage is enabled for this repo, but the '  
 model.safetensors: 100% 268M/268M [00:00<00:00, 321MB/

s]

Some weights of DistilBertForSequenceClassification were not initialized from the moc  
 You should probably TRAIN this model on a down-stream task to be able to use it for p  
 DistilBertConfig {

```

    "_attn_implementation_autoset": true,
    "activation": "gelu",
    "architectures": [
        "DistilBertForMaskedLM"
    ],
    "attention_dropout": 0.1,
    "dim": 768,
    "dropout": 0.1,
    "hidden_dim": 3072,
    "initializer_range": 0.02,
    "max_position_embeddings": 512,
    "model_type": "distilbert",
    "n_heads": 12,
    "n_layers": 6,
    "pad_token_id": 0,
    "qa_dropout": 0.1,
    "seq_classif_dropout": 0.2,
    "sinusoidal_pos_embs": false,
    "tie_weights_": true,
    "torch_dtype": "float32",
    "transformers_version": "4.51.3",
    "vocab_size": 30522
}

```

# Step 7: Define Metrics

```
import numpy as np
```

```
from sklearn.metrics import accuracy_score, precision_recall_fscore_support
```

```
def compute_metrics(pred):
```

```
    labels = pred.label_ids
```

```
    preds = np.argmax(pred.predictions, axis=1)
```

```
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average='bi
```

```
    acc = accuracy_score(labels, preds)
```

```
    return {"accuracy": acc, "f1": f1, "precision": precision, "recall": recall}
```

```
trainer = Trainer(
```

```
    model=model,
```

```
    args=training_args,
```

```
    train_dataset=train_ds
```

```

train_dataset=train_ds,
eval_dataset=val_ds,
compute_metrics=compute_metrics
)


```

# Step 9: Train the Model

```
trainer.train()
```

# Step 10: Evaluate

```
trainer.evaluate()
```

 [24747/24747 1:20:07, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.057300	0.038988	0.991272	0.991587	0.994142	0.989044
2	0.008700	0.035668	0.992969	0.993213	0.997180	0.989277
3	0.014400	0.026782	0.995393	0.995572	0.995340	0.995804

 [1032/1032 01:02]

```

{'eval_loss': 0.026782264932990074,
 'eval_accuracy': 0.9953933810158807,
 'eval_f1': 0.9955721277091587,
 'eval_precision': 0.9953401677539608,
 'eval_recall': 0.9958041958041958,
 'eval_runtime': 62.2699,
 'eval_samples_per_second': 132.472,
 'eval_steps_per_second': 16.573,
 'epoch': 3.0}

```

```
history = trainer.state.log_history
```

```

trainer.save_model("./Phishing_model_BERT")
tokenizer.save_pretrained("./Phishing_model_BERT")
!cp -r ./Phishing_model_BERT /content/drive/MyDrive/

```

```

# Assuming 'loss', 'eval_loss', 'accuracy', 'eval_accuracy' are the keys in the history
train_loss = [d['loss'] for d in history if 'loss' in d]
val_loss = [d['eval_loss'] for d in history if 'eval_loss' in d]
train_acc = [d['accuracy'] for d in history if 'accuracy' in d]
val_acc = [d['eval_accuracy'] for d in history if 'eval_accuracy' in d]

```

```

# Extract 'eval_loss' and 'accuracy' from the log_history
# Extract eval_loss for validation loss
val_loss = [log['eval_loss'] for log in trainer.state.log_history if 'eval_loss' in log]

```

```

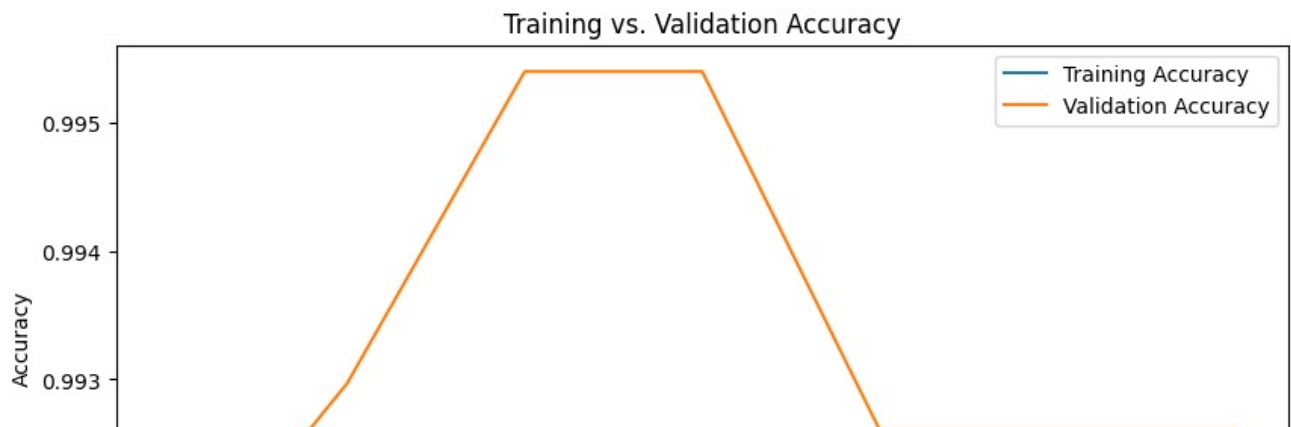
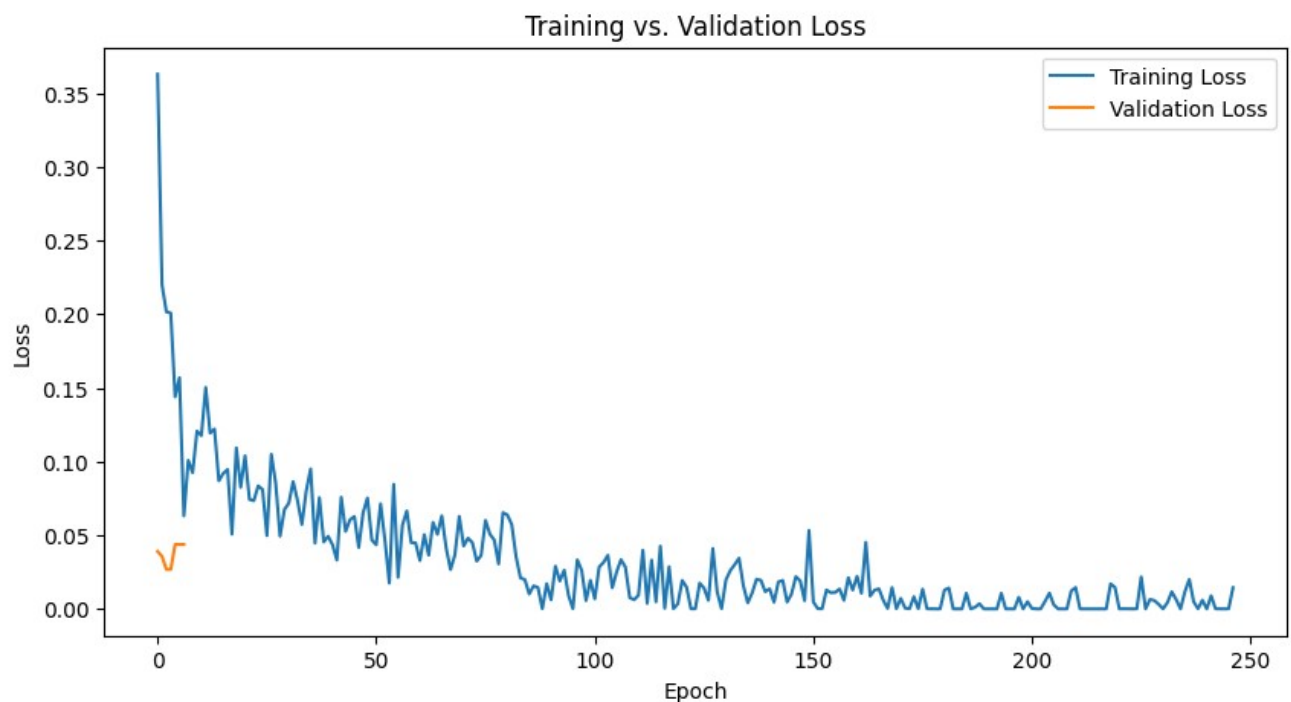
# Extract accuracy for training accuracy
train_acc = [log['accuracy'] for log in trainer.state.log_history if 'accuracy' in log]

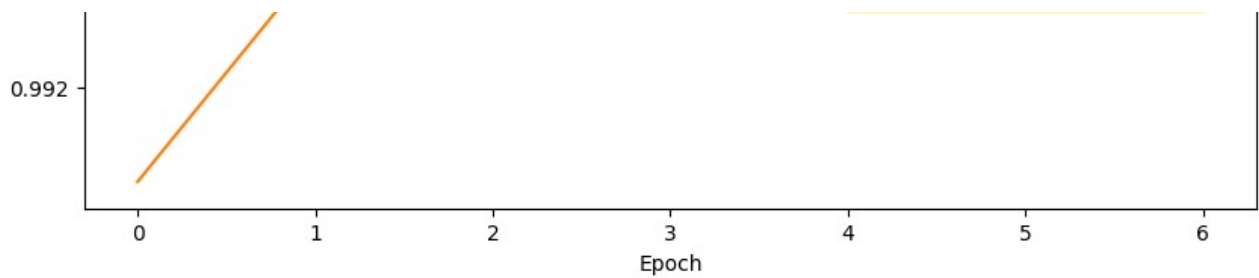
```

```
import matplotlib.pyplot as plt
```

```
# Plotting Loss
plt.figure(figsize=(10, 5))
plt.plot(train_loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.title('Training vs. Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()

# Plotting Accuracy
plt.figure(figsize=(10, 5))
plt.plot(train_acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.title('Training vs. Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```





```
# Evaluate on Test Set and Generate Full Report
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

train_accuracy = [] # Initialize an empty list to store training accuracy

def compute_metrics(pred):
    global train_accuracy # Access the global train_accuracy list
    labels = pred.label_ids
    preds = np.argmax(pred.predictions, axis=1)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average=
    acc = accuracy_score(labels, preds)
    train_accuracy.append(acc) # Append the accuracy to the list
    return {"accuracy": acc, "f1": f1, "precision": precision, "recall": recall}

results = trainer.evaluate(eval_dataset=test_ds)
#test_accuracy = results['accuracy']
val_accuracy = [d['eval_accuracy'] for d in history if 'eval_accuracy' in d][-1]
print("Test Results:", results)

print(f"Validation Accuracy: {val_accuracy}")
#print(f"Test Accuracy: {test_accuracy}")

preds = trainer.predict(test_ds)
predicted_labels = preds.predictions.argmax(-1)
target_names = ['Legitimate', 'Phishing']

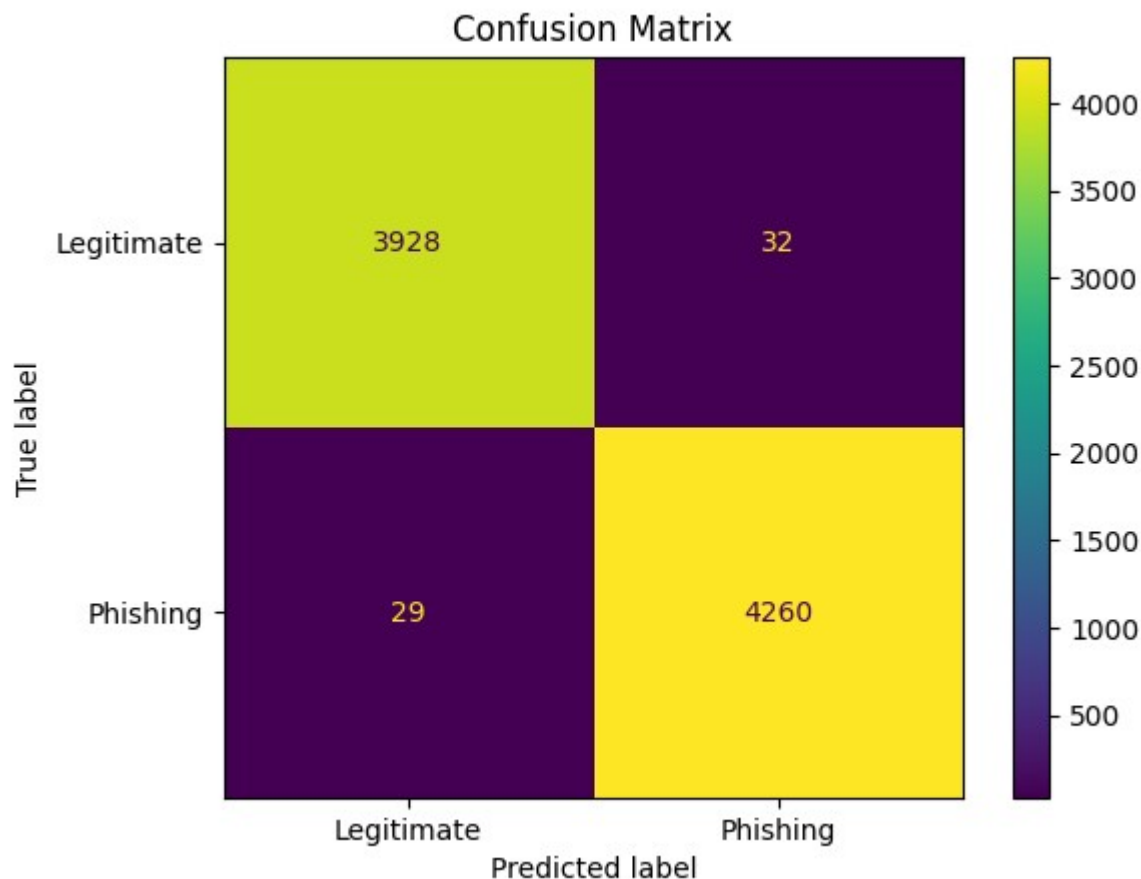
report = classification_report(preds.label_ids, predicted_labels, target_names=target_names)
print(report)

cm = confusion_matrix(preds.label_ids, predicted_labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=target_names)
disp.plot()
plt.title("Confusion Matrix")
plt.show()
```

```
Test Results: {'eval_loss': 0.04372600093483925, 'eval_accuracy': 0.9926051642623348,
Validation Accuracy: 0.9926051642623348
              precision    recall  f1-score   support
```

```
Legitimate      0.9927      0.9919      0.9923      3960
Phishing        0.9925      0.9922      0.9923      4280
```

precision	0.9925	0.9932	0.9929	4289
accuracy			0.9926	8249
macro avg	0.9926	0.9926	0.9926	8249
weighted avg	0.9926	0.9926	0.9926	8249



```
from transformers import DistilBertForSequenceClassification, DistilBertTokenizer, pipeline
import pandas as pd
```

```
# Define the model path
```

```
model_path = "/content/drive/MyDrive/Phishing_model_BERT"
```

```
# Load model and tokenizer separately
```

```
model = DistilBertForSequenceClassification.from_pretrained(model_path)
```

```
tokenizer = DistilBertTokenizer.from_pretrained(model_path)
```

```
# Now create the pipeline manually
```

```
classifier = pipeline("text-classification", model=model, tokenizer=tokenizer)
```

```
# Load your CSV file containing emails to be classified
```

```
new_emails = pd.read_csv("/content/Book1.csv") # Replace with your file path
```

```
# Assuming the CSV file has a column named 'text_combined' for the email content
```

```
predictions = []
```

```
for email in new_emails['text_combined']:
```



```

# Pass the raw email text directly to the classifier
prediction = classifier(email, truncation=True, max_length=512)
predictions.append(prediction[0]['label']) # Extract the predicted label

# Add the predictions to the dataframe
new_emails["prediction"] = predictions

# Print the updated dataframe
print(new_emails)

# Save the results back to a CSV
new_emails.to_csv("/content/drive/MyDrive/predicted_emails.csv", index=False) # Replace

Device set to use cuda:0
You seem to be using the pipelines sequentially on GPU. In order to maximize efficiency
      text_combined prediction
0    mr tim chapman freelance gentleman of letters... LABEL_0
1    smith graham computing technician wrote URL th... LABEL_0
2    geege wrote a strange story i know a guy who i... LABEL_1
3    paul moore but let s walk before we run after... LABEL_0
4    on mon nov NUMBER NUMBER at NUMBER NUMBER NUMB... LABEL_0
..                                     ...
104  drywall your home refinance loan is approved ... LABEL_1
105  dear sir madam if you are fed up of being ripp... LABEL_1
106  diversified underwriters services inc impaire... LABEL_1
107  guaranteed to increase lift and firm your bre... LABEL_1
108  attn president from mrs helina karimu i am an ... LABEL_1

[109 rows x 2 columns]

```