

Task: Building a Real Spam Classifier

Step 1: Explore the Dataset

- Load the ``sms_spam`` dataset from Hugging Face.
- Check column names and label names.
- Expected:
 - Text column: ``text``
 - Label column: ``label``
 - Labels: ``['ham', 'spam']``

Step 2: Create Label Dictionary

- Map ID → Label name and Label name → ID.
`label_map = {0: 'ham', 1: 'spam'}`
`id_map = {'ham': 0, 'spam': 1}`

Step 3: Tokenize and Preprocess

- Use AutoTokenizer from distilbert-base-uncased.
- Apply padding, truncation, and max_length=128.

Step 4: Split Train & Evaluation Data

- Shuffle dataset.
- Use 5000 samples for training and 1000 for evaluation.

Step 5: Fine-Tune DistilBERT

- Load AutoModelForSequenceClassification with num_labels=2.
- Use Trainer API with metrics: Accuracy, Precision, Recall, F1.
- Suggested hyperparameters: learning_rate=2e-5, batch_size=16, epochs=2.

Step 6: Save Model

```
trainer.save_model("./spam_model")  
tokenizer.save_pretrained("./spam_model")
```

Step 7: Load Model & Make Predictions

- Load model & tokenizer.
- Create a prediction function returning human-readable labels.

Test examples:

```
texts = [  
    "Congratulations! You've won a free ticket.",  
    "Hey, are we meeting tomorrow?",  
]  
  
for t in texts:  
    print(predict_with_label(t))
```