PacktPublishing/Hands-On-Artificial-Intelligence-for-Cybersecurity: Hands-On Artificial Intelligence for Cybersecurity, publised by Packt

https://github.com/PacktPublishing/Hands-On-Artificial-Intelligence-for-Cybersecurity/tree/2a49d5f22ada8244861140359dabe9be60670a18/Chapter03/sources

After deploying on Streamlit Cloud,
replace REPLACE_WITH_YOUR_STREAMLIT_APP_URL above with your app URL (e.g., https://your-app-name.streamlit.app). A simple, reproducible pipeline to classify messages/emails as spam or ham using scikit? earn and OpenSpec.

- Preprocessing report: docs/PREPROCESSING.md
- OpenSpec change proposal: openspec/changes/add-spam-email-classifier/

This project builds upon patterns and datasets related to the Spam Email problem from Chapter 3 of the Packt repository below. We used it to expand the preprocessing steps and add richer visualization work (step outputs, metrics, and CLI/Streamlit views).

In a fresh virtual environment (recommended)

pip install -r requirements.txt

- Raw dataset (headerless 2-column CSV): datasets/sms_spam_no_header.csv
- Cleaned dataset (generated): datasets/processed/sms_spam_clean.csv

python scripts/preprocess_emails.py \

--input datasets/sms_spam_no_header.csv \

--output datasets/processed/sms_spam_clean.csv \

--no-header --label-col-index 0 --text-col-index 1 \

--output-text-col text_clean \

--save-step-columns \

--steps-out-dir datasets/processed/steps

```
python scripts/train_spam_classifier.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 --text-col text_clean
python scripts/predict_spam.py --text "Free entry in 2 a wkly comp to win cash"
python scripts/predict_spam.py \
    --input datasets/processed/sms_spam_clean.csv \
    --text-col text_clean \
    --output predictions.csv
```

- Artifacts are saved to models/ for reuse (vectorizer, model, label mapping).
- See docs/PREPROCESSING.md for detailed step-by-step preprocessing with examples.
- OpenSpec usage: openspec validate add-spam-email-classifier --strict

```
python scripts/train_spam_classifier.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 --text-col text_clean \
    --class-weight balanced \
    --ngram-range 1,2 \
    --min-df 2 \
    --sublinear-tf \
    --C 2.0 \
    --eval-threshold 0.50
```

Observed (held-out): Precision ? 0.923, Recall ? 0.966, F1 ? 0.944.

```
# Class distribution
python scripts/visualize_spam.py \
```

```
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 \
    --class-dist

# Token frequency (top 20 per class)
python scripts/visualize_spam.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 --text-col text_clean \
    --token-freq --topn 20

# Confusion matrix, ROC, PR (requires trained artifacts in models/)
python scripts/visualize_spam.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 --text-col text_clean \
    --models-dir models \
    --confusion-matrix --roc --pr

# Threshold sweep (CSV + plot)
python scripts/visualize_spam.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 --text-col text_clean \
    --models-dir models \
    --threshold-sweep
```

Observed (held-out): Precision ? 0.923, Recall ? 0.966, F1 ? 0.944.

```
# Class distribution
python scripts/visualize_spam.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 \
    --class-dist

# Token frequency (top 20 per class)
python scripts/visualize_spam.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 --text-col text_clean \
    --token-freq --topn 20

# Confusion matrix, ROC, PR (requires trained artifacts in models/)
python scripts/visualize_spam.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 --text-col text_clean \
    --models-dir models \
    --confusion-matrix --roc --pr

# Threshold sweep (CSV + plot)
python scripts/visualize_spam.py \
    --input datasets/processed/sms_spam_clean.csv \
    --label-col col_0 --text-col text_clean \
    --models-dir models \
```

--threshold-sweep

streamlit run app/streamlit_app.py

- Dataset and column pickers
- Class distribution and top tokens by class
- Confusion matrix, ROC/PR curves (requires trained artifacts in models/)
- Threshold slider with live precision/recall/f1
- Live Inference: type a message to see predicted label and spam probability with a probability bar and threshold marker
- Quick test: use the built-in "Use spam example" / "Use ham example" buttons to auto-fill the input and try predictions immediately

cipengxu/openspec: Spec-driven development for AI coding assistants.