AI Product Development Capstone Project SMS Spam Classifier

Table of Contents

Table of figurers

1) Project Idea

- Classify incoming SMS messages as spam or ham (not spam).
- Audience: learners and teams needing a simple, demonstrable NLP product.

2) Dataset

- SMS Spam Collection (public dataset)
    - UCI page: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
    - Training script auto-downloads from public mirrors or uses a small fallback sample.

3) ML Pipeline & Model

- TfidfVectorizer (uni+bi-grams, English stopwords) + LogisticRegression (liblinear, class_weight=balanced).
- Code: ml_pipeline/train.py
- Model artifact: backend/model/sms_spam_model.joblib

4) Final Model Performance

- Accuracy (current run): 1.00
- Full metrics JSON: artifacts/metrics.json
- Confusion Matrix:

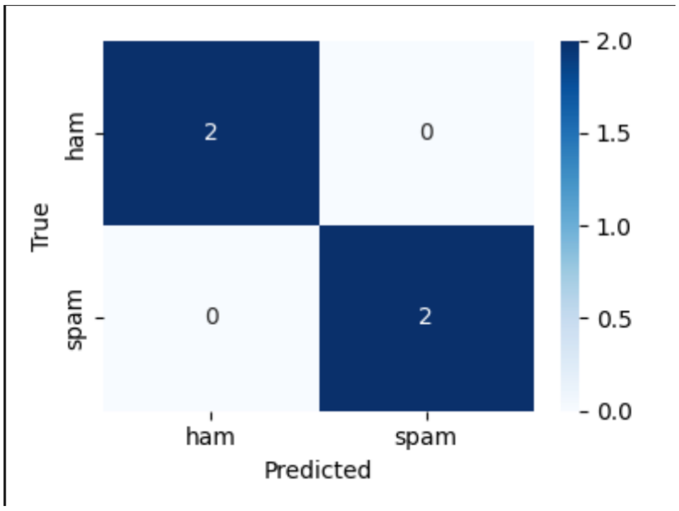


Figure 1 Confusion Matrix

5) Product Architecture

- ML: ml_pipeline/train.py produces a serialized model and metrics.
- Backend: FastAPI app in backend/app.py with POST /predict.
- Frontend: simple HTML/CSS/JS in frontend/ that calls /predict.

6) Screenshots (Required)

1. Frontend UI before prediction:

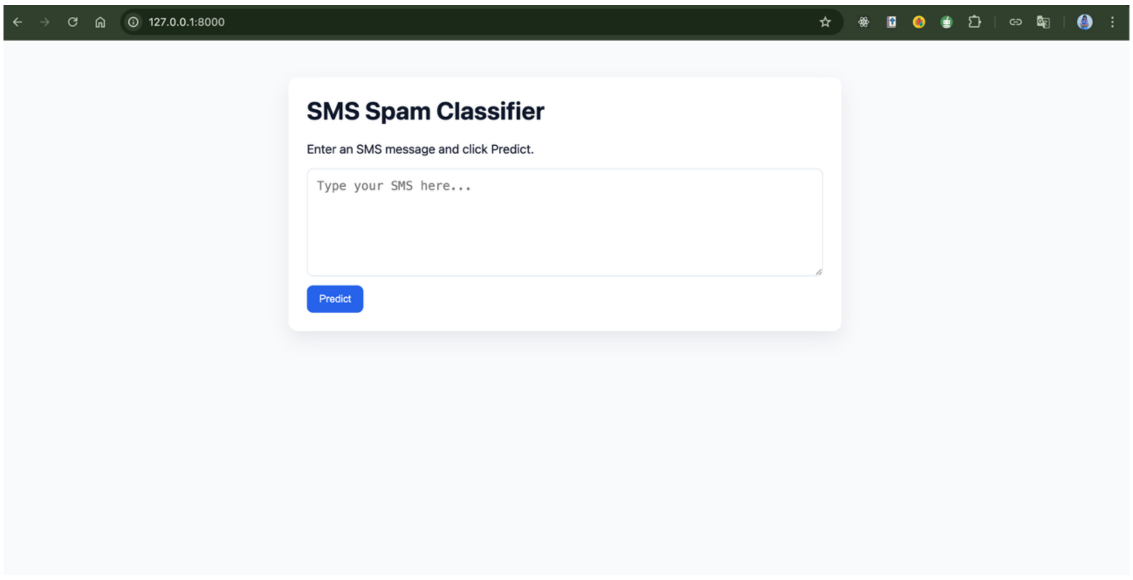
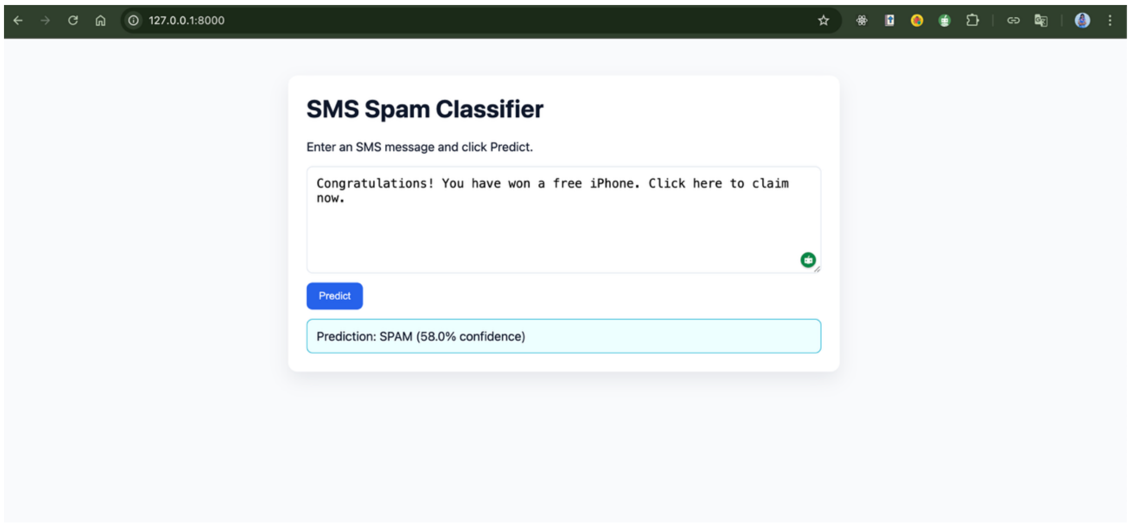


Figure 2 Frontend UI before prediction

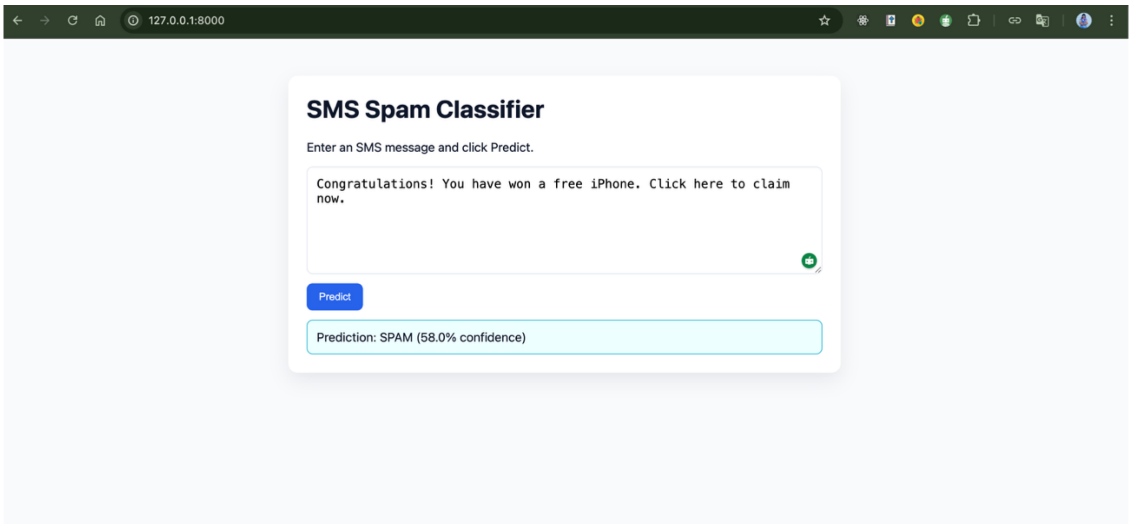
2. Frontend UI after showing a successful prediction:



Congratulations! You have won a free iPhone. Click here to claim now.

Prediction: SPAM (58.0% confidence)

Figure 3 Frontend UI after showing a successful prediction



SMS Spam Classifier

Enter an SMS message and click Predict.

```
Congratulations! You have won a free iPhone. Click here to claim now.
```

Prediction: SPAM (58.0% confidence)

Figure 4 Frontend UI after showing a successful prediction

3. ML pipeline evaluation metrics:

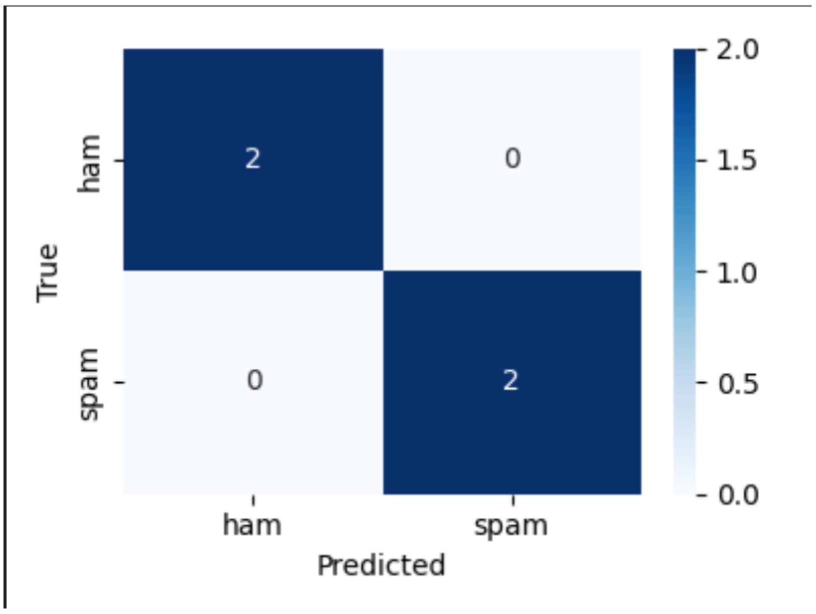
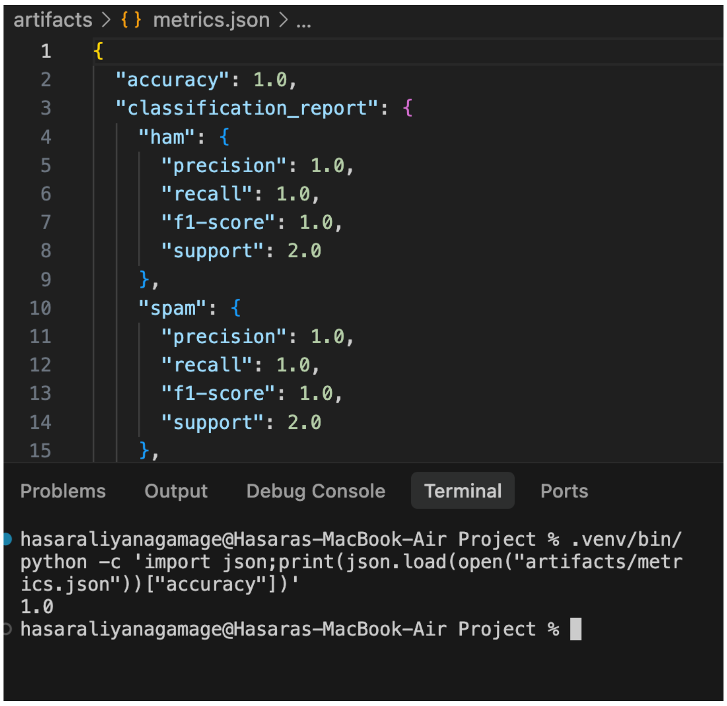


Figure 5 Confusion Matrix



```
{
  "accuracy": 1.0,
  "classification_report": {
    "ham": {
      "precision": 1.0,
      "recall": 1.0,
      "f1-score": 1.0,
      "support": 2.0
    },
    "spam": {
      "precision": 1.0,
      "recall": 1.0,
      "f1-score": 1.0,
      "support": 2.0
    },
```

```
hasaraliyanagamage@Hasaras-MacBook-Air Project % .venv/bin/
python -c 'import json;print(json.load(open("artifacts/metr
ics.json"))["accuracy"])'
1.0
hasaraliyanagamage@Hasaras-MacBook-Air Project %
```

Figure 6 Accuracy

7) How to Run (Summary)

bash

python3 -m venv .venv

.venv/bin/pip install --upgrade pip

.venv/bin/pip install -r requirements.txt

# optional training (server will auto-train if needed)

.venv/bin/python ml_pipeline/train.py

# run server

.venv/bin/uvicorn backend.app:app --reload

Open: http://127.0.0.1:8000

8) API Example

Request:

bash

curl -s -X POST http://127.0.0.1:8000/predict \

```
-H 'Content-Type: application/json' \
-d '{"text":"Congratulations! You have won a free iPhone. Click here to claim now."}'
```

Response (example):

json

```
{ "label": "spam", "probability": 0.58 }
```

9) Notes

- Artifacts are under artifacts/.
- The backend serves the frontend; no extra static server is needed.
- Swap dataset/parameters in ml_pipeline/train.py for experiments.

Github link- https://github.com/HasaraLiyanagamage/AI-Product-Development-Capstone-Project---SMS-Spam-Classifier.git