~~~
# Sentiment Analysis with Flask API

This project contains a Sentiment Analysis model trained using Word2Vec embeddings
and served through a Flask API.

---

## 🚀 Getting Started

### 1. Run the Notebook
1. Open the Jupyter Notebook (`sentiment_analysis.ipynb`).
2. Train the model by running all cells.
3. The trained model will be saved for later use in the Flask API.

---

### 2. Run the Flask API
You can run the API in two ways:

#### Option A: Using `flask run`
```bash
export FLASK_APP=app.py
flask run
~~~

Option B: Using Python directly

```
python app.py
```

By default, the API will start at:

```
http://127.0.0.1:5000
```

---

3. Test the API

✅ Health Check

Check if the API is running:

```
curl http://127.0.0.1:5000/health
```

Expected response:

```
{
  "message": "Sentiment Analysis API is running",
  "model_loaded": true,
  "status": "healthy"
}
```

🎯 Predict Sentiment for a Single Text

```
curl -X POST http://127.0.0.1:5000/predict \
    -H "Content-Type: application/json" \
    -d '{"text": "I really loved this movie!"}'
```

Example response:

```
{
  "confidence": 0.93,
  "predicted_label": "positive"
}
```

📦 Predict Sentiment for Multiple Texts (Batch)

```
curl -X POST http://127.0.0.1:5000/predict/batch \
    -H "Content-Type: application/json" \
    -d '{"texts": ["This was amazing!", "I hated this so much."]}'
```

Example response:

```
[
  {
    "text": "This was amazing!",
    "confidence": 0.95,
    "predicted_label": "positive"
  },
  {
    "text": "I hated this so much.",
    "confidence": 0.91,
    "predicted_label": "negative"
  }
]
```

---

4. Test with Postman

1. Open **Postman**.

2. Create a new request:

   - Method: `POST`
   - URL: `http://127.0.0.1:5000/predict`

3. In the **Body** tab → Select **raw** → Choose **JSON**.

4. Enter:

```json
{
  "text": "I really loved this movie!"
}
```

5. Send the request and check the response.

📊 Model Evaluation

- Accuracy, Loss, F1-score, Classification Report, and Confusion Matrix are included in the notebook.
- A confusion matrix plot is also generated (`confusion_matrix.png`).

```
Do you want me to also **add installation steps** (like `pip install -r
requirements.txt`) in the README? That will make it more complete.
```