

llm-project

September 17, 2025

1 *MINI LARGE LANGUAGE MODEL (LLM)PROJECT*

2 LLM (Large Language Model) :-

3 *A smart AI model trained on large amounts of text that can read, understand, and generate human-like text, helping in tasks like answering questions, analyzing text, or automating communication*

```
[ ]: # -----  
# Step 1: Problem Definition  
# Task: Sentiment classification (Positive / Negative) using LLM  
# -----  
  
# Step 2: Import Libraries  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.metrics import accuracy_score, classification_report,  
    ↪confusion_matrix, ConfusionMatrixDisplay  
from transformers import pipeline
```

```
[ ]: # -----  
# Step 3: Data Collection  
# Load dataset (Excel file with Text + Label)  
# -----  
data = pd.read_excel("/content/sample_texts.xlsx") # replace with your file if  
    ↪needed
```

```
[ ]: print("Data")  
print(data.head())
```

Data

	ID	Text	Label
0	1	I love learning about AI and Machine Learning.	Positive
1	2	This is the worst movie I have ever watched.	Negative
2	3	Python is a very useful programming language.	Positive

```

3      4      I am not happy with the customer service.  Negative
4      5      The weather is nice and sunny today.  Positive

```

```

[ ]: # -----
# Step 4: Data Preprocessing
# (For demo: dataset is already clean, so no extra steps)
# -----
print("\n No cleaning needed, dataset already structured.")

```

No cleaning needed, dataset already structured.

```

[ ]: # -----
# Step 5: Choose & Load Pre-trained LLM
# We'll use Hugging Face sentiment-analysis pipeline
# -----
classifier = pipeline("sentiment-analysis")

```

No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f (<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>).

Using a pipeline without specifying a model name and revision in production is not recommended.

/usr/local/lib/python3.12/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>), set it as secret in your Google Colab and restart your session.

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 or datasets.

warnings.warn(

Device set to use cpu

```

[ ]: # Step 6: Model Inference (Prediction)
# Run LLM on each text in the dataset
# -----
results = []
for text in data["Text"]:
    result = classifier(text)[0]    # Predict sentiment
    results.append(result['label'])

data["Predicted"] = results

print("\n Predictions Added:")
print(data)# -----

```

```
#
```

Predictions Added:

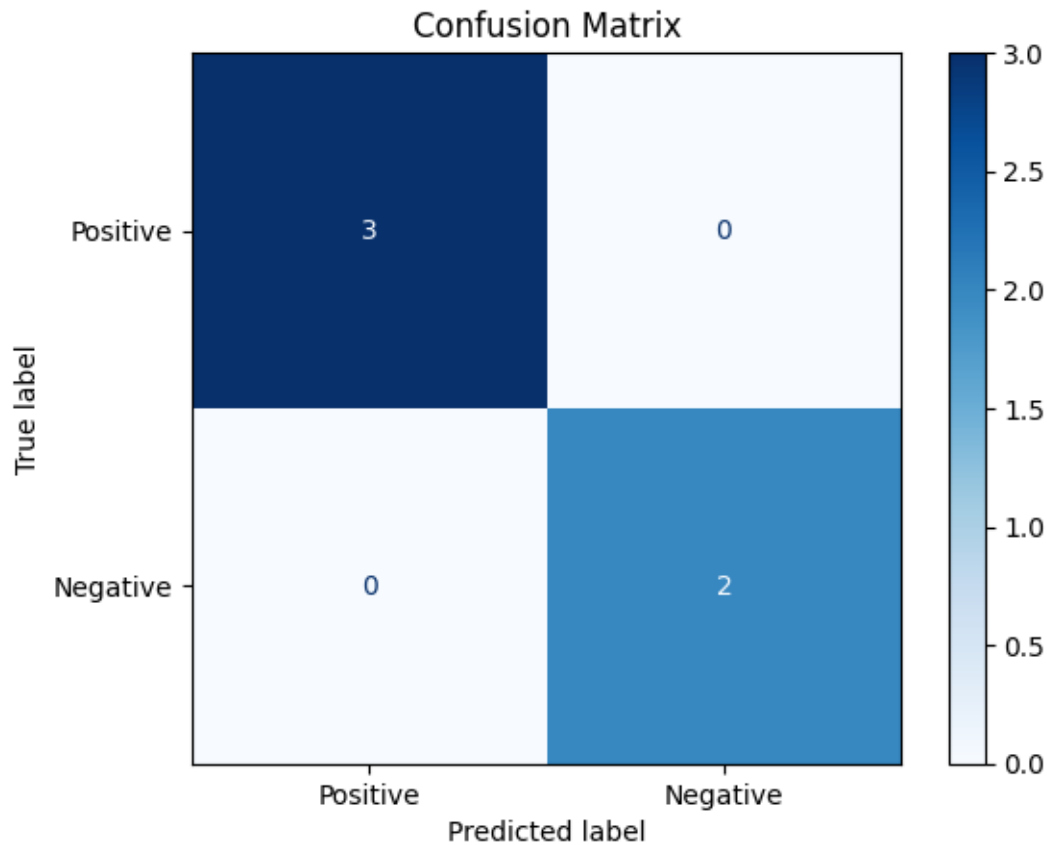
	ID	Text	Label	Predicted
0	1	I love learning about AI and Machine Learning.	Positive	POSITIVE
1	2	This is the worst movie I have ever watched.	Negative	NEGATIVE
2	3	Python is a very useful programming language.	Positive	POSITIVE
3	4	I am not happy with the customer service.	Negative	NEGATIVE
4	5	The weather is nice and sunny today.	Positive	POSITIVE

```
[ ]: # -----  
# Step 7: Evaluation  
# Compare Predicted vs Actual labels  
# -----  
# Convert Hugging Face outputs ("POSITIVE"/"NEGATIVE") to match dataset labels  
data["Predicted"] = data["Predicted"].replace({"POSITIVE": "Positive",  
↪ "NEGATIVE": "Negative"})  
  
accuracy = accuracy_score(data["Label"], data["Predicted"])  
print(f"\n Accuracy: {accuracy:.2f}")  
  
print("\n Classification Report:")  
print(classification_report(data["Label"], data["Predicted"]))  
  
# Confusion Matrix  
cm = confusion_matrix(data["Label"], data["Predicted"], labels=["Positive",  
↪ "Negative"])  
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Positive",  
↪ "Negative"])  
disp.plot(cmap="Blues")  
plt.title("Confusion Matrix")  
plt.show()
```

Accuracy: 1.00

Classification Report:

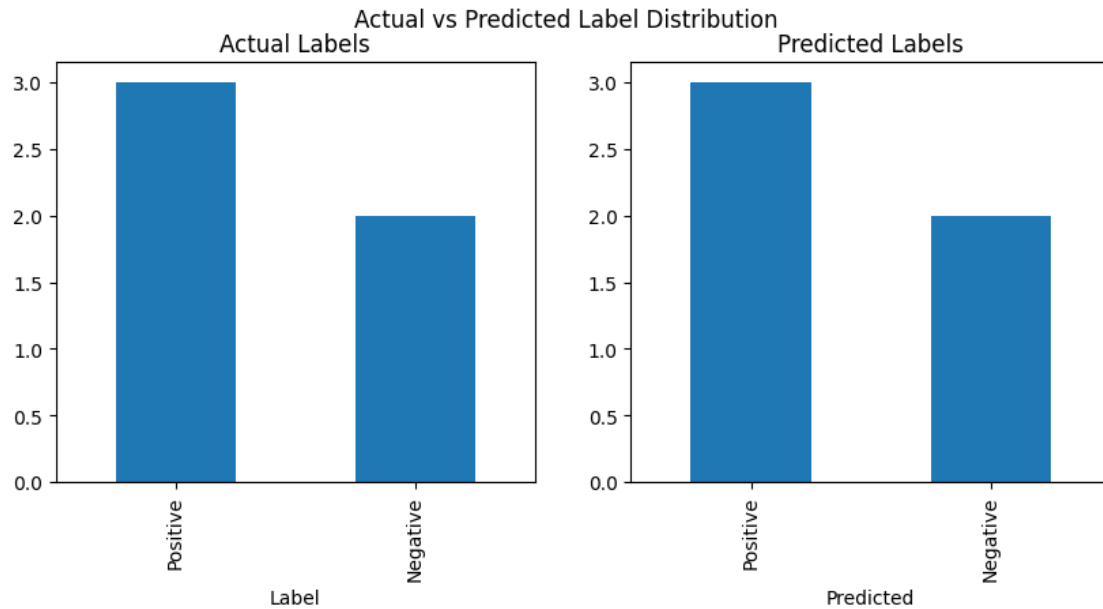
	precision	recall	f1-score	support
Negative	1.00	1.00	1.00	2
Positive	1.00	1.00	1.00	3
accuracy			1.00	5
macro avg	1.00	1.00	1.00	5
weighted avg	1.00	1.00	1.00	5



```
[ ]: # -----
# Step 8: Results & Visualization
# Show distribution of actual vs predicted labels
# -----
fig, axes = plt.subplots(1, 2, figsize=(10, 4))

data["Label"].value_counts().plot(kind="bar", ax=axes[0], title="Actual Labels")
data["Predicted"].value_counts().plot(kind="bar", ax=axes[1], title="Predicted_
↪Labels")

plt.suptitle("Actual vs Predicted Label Distribution")
plt.show()
```



```
[ ]: # -----
# Step 9: Save Results (Deployment Preparation)
# Save predictions to Excel
# -----
data.to_excel("llm_results.xlsx", index=False)
print("\n Results saved as 'llm_results.xlsx'")

# (Optional) Step 9b: Deployment with Streamlit or Flask can be added later
```

Results saved as 'llm_results.xlsx'

```
[ ]: !pip install streamlit
```

```
Requirement already satisfied: streamlit in /usr/local/lib/python3.12/dist-
packages (1.49.1)
Requirement already satisfied: altair!=5.4.0,!5.4.1,<6,>=4.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (5.5.0)
Requirement already satisfied: blinker<2,>=1.5.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (1.9.0)
Requirement already satisfied: cachetools<7,>=4.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (5.5.2)
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.12/dist-
packages (from streamlit) (8.2.1)
Requirement already satisfied: numpy<3,>=1.23 in /usr/local/lib/python3.12/dist-
packages (from streamlit) (2.0.2)
Requirement already satisfied: packaging<26,>=20 in
```

/usr/local/lib/python3.12/dist-packages (from streamlit) (25.0)
 Requirement already satisfied: pandas<3,>=1.4.0 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (2.2.2)
 Requirement already satisfied: pillow<12,>=7.1.0 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (11.3.0)
 Requirement already satisfied: protobuf<7,>=3.20 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (5.29.5)
 Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib/python3.12/dist-
 packages (from streamlit) (18.1.0)
 Requirement already satisfied: requests<3,>=2.27 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (2.32.4)
 Requirement already satisfied: tenacity<10,>=8.1.0 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (8.5.0)
 Requirement already satisfied: toml<2,>=0.10.1 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (0.10.2)
 Requirement already satisfied: typing-extensions<5,>=4.4.0 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (4.15.0)
 Requirement already satisfied: watchdog<7,>=2.1.5 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (6.0.0)
 Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (3.1.45)
 Requirement already satisfied: pydeck<1,>=0.8.0b4 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (0.9.1)
 Requirement already satisfied: tornado!=6.5.0,<7,>=6.0.3 in
 /usr/local/lib/python3.12/dist-packages (from streamlit) (6.4.2)
 Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages
 (from altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (3.1.6)
 Requirement already satisfied: jsonschema>=3.0 in
 /usr/local/lib/python3.12/dist-packages (from
 altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (4.25.1)
 Requirement already satisfied: narwhals>=1.14.2 in
 /usr/local/lib/python3.12/dist-packages (from
 altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (2.3.0)
 Requirement already satisfied: gitdb<5,>=4.0.1 in
 /usr/local/lib/python3.12/dist-packages (from
 gitpython!=3.1.19,<4,>=3.0.7->streamlit) (4.0.12)
 Requirement already satisfied: python-dateutil>=2.8.2 in
 /usr/local/lib/python3.12/dist-packages (from pandas<3,>=1.4.0->streamlit)
 (2.9.0.post0)
 Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-
 packages (from pandas<3,>=1.4.0->streamlit) (2025.2)
 Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-
 packages (from pandas<3,>=1.4.0->streamlit) (2025.2)
 Requirement already satisfied: charset_normalizer<4,>=2 in
 /usr/local/lib/python3.12/dist-packages (from requests<3,>=2.27->streamlit)
 (3.4.3)
 Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-
 packages (from requests<3,>=2.27->streamlit) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3,>=2.27->streamlit) (2.5.0)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3,>=2.27->streamlit) (2025.8.3)

Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.12/dist-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->streamlit) (5.0.2)

Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (3.0.2)

Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=3.0->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (25.3.0)

Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=3.0->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (2025.4.1)

Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=3.0->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (0.36.2)

Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=3.0->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (0.27.1)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas<3,>=1.4.0->streamlit) (1.17.0)

```
[ ]: from transformers import pipeline
      from textblob import TextBlob

      # Initialize Hugging Face sentiment classifier
      classifier = pipeline("sentiment-analysis")

      print(" Enter your text to check sentiment (type 'quit' to exit)\n")

      while True:
          text = input("Enter text: ")

          if text.lower() == "quit":
              print("Exiting... ")
              break

          # Hugging Face analysis
          hf_result = classifier(text)[0]
          hf_label = hf_result['label']          # 'POSITIVE' or 'NEGATIVE'
          hf_score = hf_result['score']          # Confidence score
```

```

# TextBlob analysis
blob = TextBlob(text)
polarity = blob.sentiment.polarity
if polarity > 0:
    tb_sentiment = "Positive"
elif polarity < 0:
    tb_sentiment = "Negative"
else:
    tb_sentiment = "Neutral"

# Combine both for a final sentiment
if hf_label == "POSITIVE" and polarity >= 0:
    final_sentiment = "Positive "
elif hf_label == "NEGATIVE" and polarity <= 0:
    final_sentiment = "Negative "
else:
    # If models disagree or TextBlob is neutral, trust Hugging Face
    ↪confidence
    if hf_score >= 0.7:
        final_sentiment = "Positive " if hf_label == "POSITIVE" else ↪
        ↪"Negative "
    else:
        final_sentiment = "Neutral "

print(f" Hugging Face: {hf_label} | Confidence: {hf_score:.2f}")
print(f" TextBlob: {tb_sentiment}")
print(f" Final Sentiment: {final_sentiment}\n")

```

No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f (<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>).

Using a pipeline without specifying a model name and revision in production is not recommended.

Device set to use cpu

Enter your text to check sentiment (type 'quit' to exit)

Hugging Face: NEGATIVE | Confidence: 1.00
TextBlob: Neutral
Final Sentiment: Negative

Hugging Face: POSITIVE | Confidence: 1.00
TextBlob: Positive
Final Sentiment: Positive

Hugging Face: NEGATIVE | Confidence: 1.00
TextBlob: Negative

Final Sentiment: Negative

Hugging Face: NEGATIVE | Confidence: 0.99

TextBlob: Neutral

Final Sentiment: Negative

```
[ ]: print("\n LLM Mini Project Completed Successfully ")
```

LLM Mini Project Completed Successfully

```
[ ]: data_ = pd.read_excel("/content/llm_results.xlsx")
```

```
[ ]: data_
```

```
[ ]:
ID          Text      Label Predicted
0    1  I love learning about AI and Machine Learning.  Positive   Positive
1    2   This is the worst movie I have ever watched.  Negative   Negative
2    3  Python is a very useful programming language.  Positive   Positive
3    4    I am not happy with the customer service.  Negative   Negative
4    5    The weather is nice and sunny today.      Positive   Positive
```

#Aim-

#The objective of this project is to classify text sentiment (positive or negative) by applying pre-trained Large Language Models (LLMs) without training or fine-tuning the model yourself.

4 Approach and Tools Used-

- 5 * Pre-trained LLM (DistilBERT) from Hugging Face Transformers was used to make predictions on the dataset. This model has been trained on large-scale public datasets for sentiment tasks and is used directly for inference.
- 6 * TextBlob, a lexical-based sentiment analyzer, was optionally used to offer an alternative sentiment score.
- 7 * scikit-learn metrics were used only to evaluate the predictions (accuracy, classification report, confusion matrix).
- 8 * pandas for data loading and manipulation.
- 9 * matplotlib for data visualization.

#Key Details-

- There is no custom training or fine-tuning performed in this project, making it a straightforward use case of LLMs for sentiment classification.
- This is a common and effective approach, since pre-trained LLMs bring deep language understanding without needing large labeled datasets or expensive training.
- The project covers all important steps: data loading, inference using pre-trained LLMs, result evaluation, and visualization.

10 Conclusion-

- 11 This project demonstrates sentiment classification using pre-trained Large Language Models without custom training. The Hugging Face DistilBERT sentiment-analysis pipeline was leveraged directly to infer sentiment labels on the dataset, supported by evaluation metrics. This approach is a practical, real-world application of LLMs for text analysis tasks.

[]: