Baseline Model - 1

MODEL: Logistic Regression with TF- IDF

In [1]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score
from sklearn.model_selection import train_test_split
import pandas as pd

# === CONFIG ===
DATA_PATH = "multilang_sarcasm_dataset.csv"
N_SHOT = 64
MAX_TEST_SAMPLES = 1000

# === LOAD & PREPROCESS ===
df = pd.read_csv(DATA_PATH)
df = df[df["lang"] == "en"]
df = df[["article_title", "is_sarcastic"]].rename(columns={"article_ti
df = df.dropna(subset=["text", "label"])

# === TRAIN/TEST SPLIT ===
train_df, test_df = train_test_split(df, test_size=0.2, stratify=df["l

# === FEW-SHOT SAMPLING ===
def sample_few_shot(df, n=64):
    return df.groupby("label").apply(lambda x: x.sample(n=min(n, len(x

fewshot_train_df = sample_few_shot(train_df, N_SHOT)
test_subset_df = test_df.sample(n=min(len(test_df), MAX_TEST_SAMPLES),

# === TF-IDF + LOGISTIC REGRESSION ===
X_train = fewshot_train_df["text"]
y_train = fewshot_train_df["label"]
X_test = test_subset_df["text"]
y_test = test_subset_df["label"]

# TF-IDF vectorization
vectorizer = TfidfVectorizer(ngram_range=(1, 2), max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Logistic Regression classifier
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train_tfidf, y_train)
y_pred_lr = lr.predict(X_test_tfidf)

# Evaluation
acc = accuracy_score(y_test, y_pred_lr)
f1 = f1_score(y_test, y_pred_lr)
```

```
print(f"Logistic Regression (TF-IDF) | Accuracy: {acc:.4f} | F1 Score:
```

```
Logistic Regression (TF-IDF) | Accuracy: 0.7030 | F1 Score: 0.7021
```

```
/var/folders/lv/xd91rcv91cq23cjjl0_c93nh0000gn/T/ipykernel_84805/422360
6387.py:23: DeprecationWarning: DataFrameGroupBy.apply operated on the
grouping columns. This behavior is deprecated, and in a future version
of pandas the grouping columns will be excluded from the operation. Eit
her pass `include_groups=False` to exclude the groupings or explicitly
select the grouping columns after groupby to silence this warning.
  return df.groupby("label").apply(lambda x: x.sample(n=min(n, len(x)),
random_state=42)).reset_index(drop=True)
```

Baseline Model - 2

MODEL: Zero- Shot Classification

In [2]:

```python
#Baseline 2: Zero-Shot Classification with BART-MNLI

from transformers import pipeline
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score
import pandas as pd

# === CONFIG ===
DATA_PATH = "multilang_sarcasm_dataset.csv"
MAX_TEST_SAMPLES = 1000
EVAL_SIZE = 200  # limit for speed

# === LOAD & PREPROCESS ===
df = pd.read_csv(DATA_PATH)
df = df[df["lang"] == "en"]
df = df[["article_title", "is_sarcastic"]].rename(columns={"article_ti
df = df.dropna(subset=["text", "label"])

# === TRAIN/TEST SPLIT === (note: train not needed for zero-shot)
_, test_df = train_test_split(df, test_size=0.2, stratify=df["label"],
test_subset_df = test_df.sample(n=min(len(test_df), MAX_TEST_SAMPLES),

# === ZERO-SHOT CLASSIFICATION ===
from transformers import pipeline

# Load model
classifier = pipeline("zero-shot-classification", model="facebook/bart

# Labels for binary classification
candidate_labels = ["sarcastic", "not sarcastic"]

# Prepare subset for faster evaluation
texts = test_subset_df["text"].tolist()[:EVAL_SIZE]
true_labels = test_subset_df["label"].tolist()[:EVAL_SIZE]
```

```python
# Predict
preds = []
for text in texts:
    result = classifier(text, candidate_labels)
    pred_label = result["labels"][0]
    pred = 1 if pred_label == "sarcastic" else 0
    preds.append(pred)

# Evaluation
acc = accuracy_score(true_labels, preds)
f1 = f1_score(true_labels, preds)

print(f" Zero-Shot (BART-MNLI) | Accuracy: {acc:.4f} | F1 Score: {f1:.
```

```
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
Device set to use mps:0
 Zero-Shot (BART-MNLI) | Accuracy: 0.5350 | F1 Score: 0.1622
```