Double-click (or enter) to edit

Upload the Dataset

```
from google.colab import files
uploaded = files.upload()

import pandas as pd

# Replace the filename with the actual name after upload
df = pd.read_csv('/content/Bitext_Sample_Customer_Support_Training_Dataset_27K_responses-v11.csv')
```

```
Choose Files Bitext_Sam…-v11 (1).csv
• Bitext_Sample_Customer_Support_Training_Dataset_27K_responses-v11 (1).csv(text/csv) - 19202474 bytes, last modified: 5/15/2025 - 100% done
Saving Bitext_Sample_Customer_Support_Training_Dataset_27K_responses-v11 (1).csv to Bitext
```

Data Exploration

```
# View the first few records
df.head()

# Dataset shape
print("Shape:", df.shape)

# Column names
print("Columns:", df.columns.tolist())

# Info about dataset
df.info()

# Summary stats
df.describe(include='all')
```

```
Shape: (26872, 5)
Columns: ['flags', 'instruction', 'category', 'intent', 'response']
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26872 entries, 0 to 26871
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   flags        26872 non-null  object
 1   instruction  26872 non-null  object
 2   category     26872 non-null  object
 3   intent       26872 non-null  object
 4   response     26872 non-null  object
dtypes: object(5)
memory usage: 1.0+ MB
```

	flags	instruction	category	intent	response
count	26872	26872	26872	26872	26872
unique	394	24635	11	27	26870
top	BL	shiping to {{Delivery City}}	ACCOUNT	contact_customer_service	Firstly, I truly understand how pivotal the {{...
freq	5212	8	5986	1000	2

Check for Missing Values and Duplicates

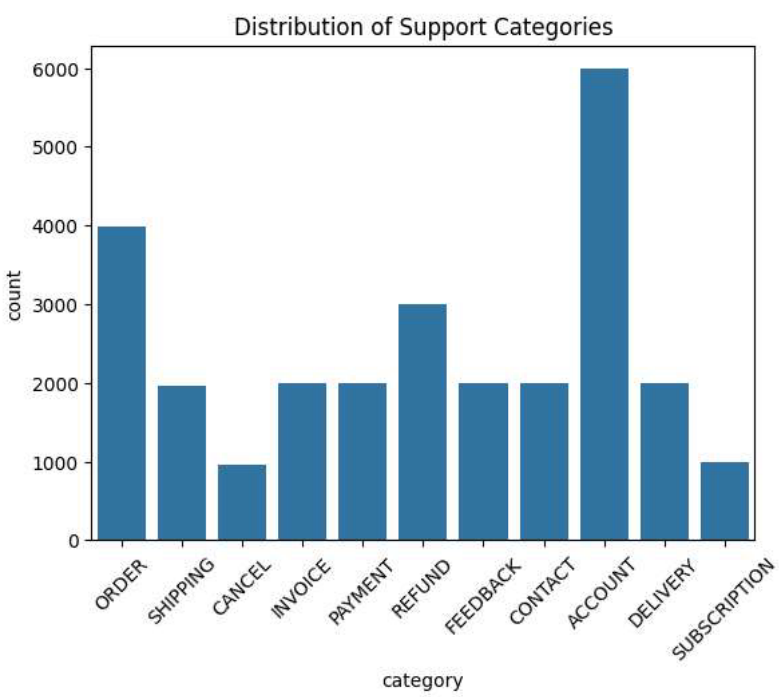
Generate print hello world using rot13 Close

```
# Check for missing values
print("Missing values:\n", df.isnull().sum())

# Check for duplicates
print("Duplicate rows:", df.duplicated().sum())
```

```
Missing values:
 flags          0
instruction    0
category       0
intent         0
response       0
dtype: int64
Duplicate rows: 0
```

Visualize Key Features (adjust column names based on your dataset)

```
import seaborn as sns
import matplotlib.pyplot as plt

# Example if you have a 'category' column or sentiment label
if 'category' in df.columns:
    sns.countplot(x='category', data=df)
    plt.title('Distribution of Support Categories')
    plt.xticks(rotation=45)
    plt.show()
```



Preprocessing (Text Cleaning Example)

```
import re

def clean_text(text):
    text = str(text).lower()
    text = re.sub(r"http\S+|www\S+|https\S+", '', text)
    text = re.sub(r'\@w+|\#','', text)
    text = re.sub(r'[^A-Za-z0-9\s]', '', text)
    return text

# Assuming 'response' is your main text column
df['cleaned_text'] = df['response'].apply(clean_text)
```

Text Vectorization (TF-IDF)

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(stop_words='english', max_features=5000
X = tfidf.fit_transform(df['cleaned_text'])
```

```
# Set your target column (e.g., category or sentiment)
y = df['category']  # Replace with actual column name
```

Train-Test Split & Modeling python Copy Edit

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.9988837209302326
Classification Report:
                precision    recall  f1-score   support

       ACCOUNT       1.00      1.00      1.00      1160
        CANCEL       1.00      1.00      1.00       199
       CONTACT       1.00      1.00      1.00       409
      DELIVERY       1.00      1.00      1.00       389
      FEEDBACK       1.00      1.00      1.00       427
       INVOICE       1.00      1.00      1.00       407
         ORDER       1.00      1.00      1.00       763
       PAYMENT       1.00      1.00      1.00       410
        REFUND       1.00      1.00      1.00       601
      SHIPPING       1.00      1.00      1.00       444
  SUBSCRIPTION       1.00      1.00      1.00       166

      accuracy                           1.00      5375
     macro avg       1.00      1.00      1.00      5375
  weighted avg       1.00      1.00      1.00      5375
```

Predict New Input (Text)

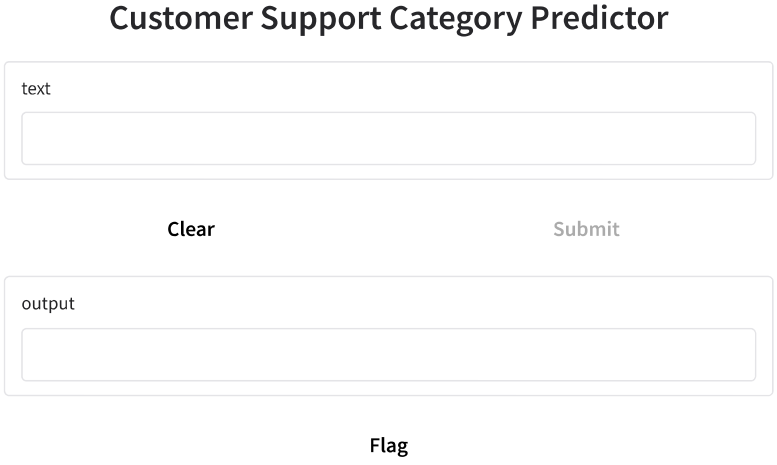
```
sample_text = ["My internet is not working, please help!"]
sample_cleaned = [clean_text(text) for text in sample_text]
sample_vec = tfidf.transform(sample_cleaned)
print("Predicted category:", model.predict(sample_vec))
```

```
Predicted category: ['ACCOUNT']
```

(Optional) Gradio Deployment

```
!pip install gradio

import gradio as gr

def predict_category(text):
    text_clean = clean_text(text)
    vec = tfidf.transform([text_clean])
    pred = model.predict(vec)[0]
    return pred

gr.Interface(fn=predict_category, inputs="text", outputs="text", title="Customer Support Category Predictor").launch()
```

```
Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-packages (5.29.1)
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-packages (from grad
Requirement already satisfied: gradio-client==1.10.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: huggingface-hub>=0.28.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (from grad
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from gra
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from ht
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from h
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from h
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: hf-xet<2.0.0,>=1.1.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from p
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from
It looks like you are running Gradio on a hosted a Jupyter notebook. For the Gradio app to

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://b9a3ba16d2d6b3742f.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradi
```



Use via API · Built with Gradio · Settings