```
!pip install nltk scikit-learn pandas matplotlib seaborn
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages (1.6.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.12/dist-packages (0.13.2)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2025.11.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (2.0.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.16.3)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.3)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.61.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.3.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.
```

```
import pandas as pd
import numpy as np
import re
import string
import pickle

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

import nltk
from nltk.corpus import stopwords
```

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
df = pd.read_csv("/content/customer_support_tickets.csv")
```

```
df.head()
```

	Ticket ID	Customer Name	Customer Email	Customer Age	Customer Gender	Product Purchased	Date of Purchase	Ticket Type	Ticket Subject	Tic Descript
0	1	Marisa Obrien	carrollallison@example.com	32	Other	GoPro Hero	2021-03-22	Technical issue	Product setup	I'm having an is with {product_purcha
1	2	Jessica Rios	clarkeashley@example.com	42	Female	LG Smart TV	2021-05-22	Technical issue	Peripheral compatibility	I'm having an is with {product_purcha
2	3	Christopher Robbins	gonzalestracy@example.com	48	Other	Dell XPS	2020-07-14	Technical issue	Network problem	I'm facing a prob with {product_purcha
3	4	Christina Dillon	bradleyolson@example.org	27	Female	Microsoft Office	2020-11-13	Billing inquiry	Account access	I'm having an is with {product_purcha
4	5	Alexander Carroll	bradleymark@example.com	67	Female	Autodesk AutoCAD	2020-02-04	Billing inquiry	Data loss	I'm having an is with {product_purcha

Next steps: Generate code with df | New interactive sheet

```
df.shape
```

```
(8469, 17)
```

```
df.columns
```

```
Index(['Ticket ID', 'Customer Name', 'Customer Email', 'Customer Age',
       'Customer Gender', 'Product Purchased', 'Date of Purchase',
       'Ticket Type', 'Ticket Subject', 'Ticket Description', 'Ticket Status',
       'Resolution', 'Ticket Priority', 'Ticket Channel',
       'First Response Time', 'Time to Resolution',
       'Customer Satisfaction Rating'],
      dtype='object')
```

```
# Dataset structure
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8469 entries, 0 to 8468
Data columns (total 17 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Ticket ID                     8469 non-null   int64
 1   Customer Name                 8469 non-null   object
 2   Customer Email                8469 non-null   object
 3   Customer Age                  8469 non-null   int64
 4   Customer Gender               8469 non-null   object
 5   Product Purchased             8469 non-null   object
 6   Date of Purchase              8469 non-null   object
 7   Ticket Type                   8469 non-null   object
 8   Ticket Subject                8469 non-null   object
 9   Ticket Description            8469 non-null   object
 10  Ticket Status                 8469 non-null   object
 11  Resolution                    2769 non-null   object
 12  Ticket Priority               8469 non-null   object
 13  Ticket Channel                8469 non-null   object
 14  First Response Time           5650 non-null   object
 15  Time to Resolution            2769 non-null   object
 16  Customer Satisfaction Rating  2769 non-null   float64
dtypes: float64(1), int64(2), object(14)
memory usage: 1.1+ MB
```

```
df.isnull().sum()
```

	0
Ticket ID	0
Customer Name	0
Customer Email	0
Customer Age	0
Customer Gender	0
Product Purchased	0
Date of Purchase	0
Ticket Type	0
Ticket Subject	0
Ticket Description	0
Ticket Status	0
Resolution	5700
Ticket Priority	0
Ticket Channel	0
First Response Time	2819
Time to Resolution	5700
Customer Satisfaction Rating	5700

dtype: int64

```
df = df[['Ticket Subject', 'Ticket Description', 'Ticket Type']]
df.dropna(inplace=True)
```

```
df['text'] = df['Ticket Subject'] + " " + df['Ticket Description']
df.head()
```

	Ticket Subject	Ticket Description	Ticket Type	text
0	Product setup	I'm having an issue with the {product_purchase...	Technical issue	Product setup I'm having an issue with the {pr...
1	Peripheral compatibility	I'm having an issue with the {product_purchase...	Technical issue	Peripheral compatibility I'm having an issue w...
2	Network problem	I'm facing a problem with my {product_purchase...	Technical issue	Network problem I'm facing a problem with my {...
		I'm having an issue with the		Account access I'm having an issue with the

Next steps: Generate code with df | New interactive sheet

preprocess text: lowercase, remove punctuation, stop words

```
from nltk.corpus import stopwords
import string

stop_words = set(stopwords.words('english'))

def preprocess_text(text):
    # convert to lowercase
    text = text.lower()

    # remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))

    # remove stopwords
    words = text.split()
    words = [word for word in words if word not in stop_words]

    return " ".join(words)

df['clean_text'] = df['text'].apply(preprocess_text)
df.head()
```

	Ticket Subject	Ticket Description	Ticket Type	text	clean_text
0	Product setup	I'm having an issue with the {product_purchase...	Technical issue	Product setup I'm having an issue with the {pr...	product setup im issue productpurchased please...
1	Peripheral compatibility	I'm having an issue with the {product_purchase...	Technical issue	Peripheral compatibility I'm having an issue w...	peripheral compatibility im issue productpurch...
2	Network problem	I'm facing a problem with my {product_purchase...	Technical issue	Network problem I'm facing a problem with my {...	network problem im facing problem productpurch...

```
df['Ticket Type'].value_counts()
```

Ticket Type	count
Refund request	1752
Technical issue	1747
Cancellation request	1695
Product inquiry	1641
Billing inquiry	1634

dtype: int64

```
df['clean_text'].str.len().describe()
```

	clean_text
count	8469.000000
mean	207.556618
std	31.711486
min	107.000000
25%	191.000000
50%	210.000000
75%	228.000000
max	332.000000

dtype: float64

```
df['Ticket Type'].nunique()
```

```
5
```

```
from sklearn.model_selection import train_test_split

X = df['clean_text']
y = df['Ticket Type']

X_train, X_temp, y_train, y_temp = train_test_split(
    X, y,
    test_size=0.30,
    random_state=42,
    stratify=y
)

X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp,
    test_size=0.50,
    random_state=42,
    stratify=y_temp
)
```

```
print("Training set:", X_train.shape)
print("Validation set:", X_val.shape)
print("Test set:", X_test.shape)
```

```
Training set: (5928,)
Validation set: (1270,)
Test set: (1271,)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(
    analyzer="char_wb",
    ngram_range=(3,5),
    min_df=2,
    max_df=0.9,
    max_features=60000
)

X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
```

```
from sklearn.neural_network import MLPClassifier

model = MLPClassifier(
    hidden_layer_sizes=(256,128),
    activation="relu",
    solver="adam",
    alpha=1e-4,
    learning_rate_init=0.001,
    max_iter=1,          # 🔥 one epoch
    warm_start=True,     # 🔥 allows epoch loop
    random_state=42
)
```

```
from sklearn.metrics import accuracy_score, log_loss
import time

epochs = 50
steps = X_train_tfidf.shape[0]

print("\nModel Training\n")

for epoch in range(1, epochs + 1):
    start = time.time()

    model.fit(X_train_tfidf, y_train)

    y_pred = model.predict(X_train_tfidf)
    y_prob = model.predict_proba(X_train_tfidf)

    acc = accuracy_score(y_train, y_pred)
    loss = log_loss(y_train, y_prob)

    duration = time.time() - start

    print(f"Epoch {epoch}/{epochs}")
    print(
        f"{steps}/{steps} [==============================] "
        f"- {duration:.1f}s "
        f"- loss: {loss:.4f} "
        f"- accuracy: {acc:.4f}"
    )
```

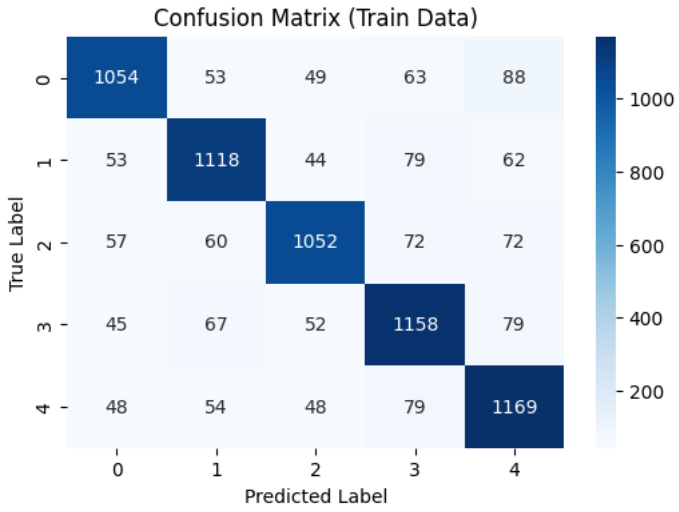
```
Model Training

/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 1/50
5928/5928 [==============================] - 16.9s - loss: 1.5841 - accuracy: 0.3325
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 2/50
5928/5928 [==============================] - 14.9s - loss: 1.5027 - accuracy: 0.4366
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 3/50
5928/5928 [==============================] - 14.6s - loss: 1.3641 - accuracy: 0.5410
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 4/50
5928/5928 [==============================] - 17.2s - loss: 1.1810 - accuracy: 0.6145
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
```

```
  warnings.warn(
Epoch 5/50
5928/5928 [==============================] - 15.2s - loss: 0.9965 - accuracy: 0.6810
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 6/50
5928/5928 [==============================] - 10.3s - loss: 0.8373 - accuracy: 0.7304
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 7/50
5928/5928 [==============================] - 9.4s - loss: 0.7097 - accuracy: 0.7640
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 8/50
5928/5928 [==============================] - 10.4s - loss: 0.6087 - accuracy: 0.7986
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 9/50
5928/5928 [==============================] - 9.7s - loss: 0.5280 - accuracy: 0.8232
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 10/50
5928/5928 [==============================] - 9.7s - loss: 0.4623 - accuracy: 0.8489
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 11/50
5928/5928 [==============================] - 9.1s - loss: 0.4094 - accuracy: 0.8652
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 12/50
5928/5928 [==============================] - 9.6s - loss: 0.3667 - accuracy: 0.8801
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
Epoch 13/50
5928/5928 [==============================] - 11.6s - loss: 0.3311 - accuracy: 0.8915
/usr/local/lib/python3.12/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:691: ConvergenceWarning: Stochast
  warnings.warn(
```

```
from sklearn.metrics import classification_report

print("\nClassification Report (Train):\n")
print(classification_report(y_train, y_pred))
```

```
Classification Report (Train):

                     precision    recall  f1-score   support

     Billing inquiry       0.98      0.99      0.98      1144
Cancellation request       0.96      0.97      0.97      1186
     Product inquiry       0.96      0.98      0.97      1149
      Refund request       0.98      0.96      0.97      1226
     Technical issue       0.98      0.96      0.97      1223

            accuracy                           0.97      5928
           macro avg       0.97      0.97      0.97      5928
        weighted avg       0.97      0.97      0.97      5928
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

cm = confusion_matrix(y_train, y_pred)

plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix (Train Data)")
plt.show()
```




```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score, classification_report

# =========================
# FAST TF-IDF
# =========================
tfidf = TfidfVectorizer(
    stop_words="english",
    ngram_range=(1,2),
    min_df=3,
    max_df=0.9,
    max_features=12000,
    sublinear_tf=True
)

X_train_tfidf = tfidf.fit_transform(X_train)

# =========================
# LINEAR SVM (KEY CHANGE)
# =========================
svm = LinearSVC(
    C=4.0,              # 🔥 tune between 3–6
    max_iter=3000
)

svm.fit(X_train_tfidf, y_train)

# =========================
# RESULTS
# =========================
y_pred = svm.predict(X_train_tfidf)

print("Final Train Accuracy:", round(accuracy_score(y_train, y_pred), 2))
print("\nClassification Report:\n")
print(classification_report(y_train, y_pred))
```

```
Final Train Accuracy: 0.82

Classification Report:

                    precision    recall  f1-score   support

     Billing inquiry       0.84      0.81      0.82      1307
Cancellation request       0.83      0.82      0.83      1356
     Product inquiry       0.84      0.80      0.82      1313
      Refund request       0.80      0.83      0.81      1401
     Technical issue       0.80      0.84      0.82      1398

            accuracy                           0.82      6775
           macro avg       0.82      0.82      0.82      6775
        weighted avg       0.82      0.82      0.82      6775
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

cm = confusion_matrix(y_train, y_pred)
```

```
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix (Train Data)")
plt.show()
```