# Binary Classifier with Revised EDA

```
!pip install kagglehub[pandas-datasets] --quiet

!pip install transformers evaluate accelerate --quiet
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/84.0 kB ? eta -:--:--
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 84.0/84.0 kB 3.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/491.2 kB ? eta -:--:--
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 491.2/491.2 kB 16.1 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/116.3 kB ? eta -:--:--
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 9.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 183.9/183.9 kB 14.6 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 12.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 4.6 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 64.3 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 34.0 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 23.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.0 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 5.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 12.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 6.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 4.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 41.1 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 16.6 MB/s eta
0:00:00
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec
2024.12.0 which is incompatible.
```

```
!pip install tensorflow[and-cuda] --quiet
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.3/363.3 MB 3.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 45.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.9/24.9 MB 31.0 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 895.7/895.7 kB 26.6 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 577.2/577.2 MB 2.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 192.5/192.5 MB 6.8 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 13.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 130.3/130.3 MB 6.6 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 217.6/217.6 MB 5.6 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.3/21.3 MB 69.7 MB/s eta
0:00:00
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cublas-cu12 12.5.3.2 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-cupti-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-nvrtc-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-runtime-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-runtime-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cudnn-cu12==9.1.0.70;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cudnn-cu12 9.3.0.75 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cufft-cu12 11.2.3.61 which is incompatible.
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-curand-cu12 10.3.6.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cusolver-cu12 11.6.3.83 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cusparse-cu12 12.5.1.3 which is incompatible.
```

```
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-nvjitlink-cu12 12.5.82 which is incompatible.

!pip install keras-tuner --quiet

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/129.1 kB ? eta -:--:--
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.1/129.1 kB 10.1 MB/s eta
0:00:00

!pip install shap --quiet

# Data Related
import kagglehub
from kagglehub import KaggleDatasetAdapter
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from imblearn.under_sampling import RandomUnderSampler



# NLP and DL
import tensorflow as tf
import tensorflow_hub as hub
import tensorflow_text as text
from tensorflow.keras import mixed_precision
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.regularizers import l2
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import (Embedding,
                                     Concatenate,
                                     Input,
                                     LSTM,
                                     Bidirectional,
                                     Dense,
                                     GlobalAveragePooling1D,
                                     GlobalMaxPooling1D,
                                     BatchNormalization,
                                     Dropout)
from tensorflow.keras.metrics import Precision, Recall, F1Score
from kerastuner import HyperModel
from kerastuner.tuners import RandomSearch
import shap
from sklearn.metrics import precision_score, recall_score, f1_score
```

```
# Misc
import sys
from wordcloud import WordCloud
from collections import Counter
import re
import nltk

nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk import ngrams
import pickle
```

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

```
print("Num GPUs Available:",
len(tf.config.experimental.list_physical_devices('GPU')))
# Enable Tensor Core acceleration
mixed_precision.set_global_policy("mixed_float16")
```

```
Num GPUs Available: 1
```

## Version Check

```
print('Python: {}'.format(sys.version))
print('Pandas: {}'.format(pd.__version__))
print('NumPy: {}'.format(np.__version__))
print('Tensorflow: {}'.format(tf.__version__))
```

```
Python: 3.11.11 (main, Dec  4 2024, 08:55:07) [GCC 11.4.0]
Pandas: 2.2.2
NumPy: 2.0.2
Tensorflow: 2.18.0
```

## Data Cleaning

```
file_path = "dataset-tickets-multi-lang-4-20k.csv"

# Load the latest version
df = kagglehub.load_dataset(
  KaggleDatasetAdapter.PANDAS,
  "tobiasbueck/multilingual-customer-support-tickets",
  file_path,
)
```

```
<ipython-input-9-3398c303bdef>:4: DeprecationWarning: load_dataset is
deprecated and will be removed in future version.
  df = kagglehub.load_dataset(
```

```
Downloading from
https://www.kaggle.com/api/v1/datasets/download/tobiasbueck/multilingual-customer-support-tickets?dataset_version_number=9&file_name=dataset-tickets-multi-lang-4-20k.csv...

100%|██████████| 17.9M/17.9M [00:02<00:00, 6.90MB/s]
```

```
# Checking data types of the variables
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   subject   18539 non-null  object
 1   body      19998 non-null  object
 2   answer    19996 non-null  object
 3   type      20000 non-null  object
 4   queue     20000 non-null  object
 5   priority  20000 non-null  object
 6   language  20000 non-null  object
 7   tag_1     20000 non-null  object
 8   tag_2     19954 non-null  object
 9   tag_3     19905 non-null  object
 10  tag_4     18461 non-null  object
 11  tag_5     13091 non-null  object
 12  tag_6     7351 non-null   object
 13  tag_7     3928 non-null   object
 14  tag_8     1907 non-null   object
dtypes: object(15)
memory usage: 2.3+ MB
```

## Inital Data Loading

```
print("Shape:", df.shape)
df.head()
```

```
Shape: (20000, 15)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 20000,\n  \"fields\":
[\n    {\n      \"column\": \"subject\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 18539,\n
\"samples\": [\n          \"Software crashes during data visualization
due to potential memory compatibility issues with NAS-System, even
after restarts and updates.\",\n          \"Personalizing Billing
Settings in Dynamics\",\n          \"Improving Brand Expansion Digital
Tactics Online\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"body\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 19998,\n        \"samples\": [\n
\"What are the system requirements for using your project management
SaaS in conjunction with QuickBooks Online integration?\",\n
\"I am writing to request enhanced user authentication security,
specifically multi-factor authentication, to improve the safety of my
account and help protect it from unauthorized access.\",\n
\"Sehr geehrte Kundenservice, ich w\\u00fcrde gerne mehr \\u00fcber
die Dienstleistungen zur Sicherung medizinischer Daten in Krankenhaus-
IT-Systemen erfahren. K\\u00f6nnten Sie bitte detaillierte
Informationen \\u00fcber die Sicherheitsma\\u00dfnahmen geben,
einschlie\\u00dflich Verschl\\u00fcsselung, Firewalls und
Zugriffskontrolle? Bitte stellen Sie auch relevante Zertifizierungen
und die Einhaltung der Branche-Standard bereit. Zudem w\\u00e4re ich
dankbar f\\u00fcr Beispiele erfolgreicher Umsetzungen und Fallstudien.
Ich suche einen zuverl\\u00e4ssigen und vertrauensw\\u00fcrdigen
Partner, der die Vertraulichkeit, Integrit\\u00e4t und Verf\\
u00fcgbarkeit sensibler medizinischer Informationen gew\\
u00e4hrleistet. Danke.\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"answer\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 19996,\n
\"samples\": [\n          \"Please investigate the issue and contact
+1-800-123-4567 for assistance with potential solutions.\",\n
\"Sehr geehrte [name], ich bedaure die Probleme mit Ihren mehreren
Integrationsproblemen zu h\\u00f6ren. Ich m\\u00f6chte dies genauer
untersuchen. K\\u00f6nnten Sie bitte Details \\u00fcber die Docker-
Aktualisierungen, die Sie k\\u00fcrzlich vorgenommen haben, und die
genauen Fehlernachrichten, die Sie sehen, bereitstellen? Ich k\\
u00f6nnte m\\u00f6glicherweise einen Anruf mit Ihnen vereinbaren, um
dies weiter zu bereden, w\\u00e4ren Sie am [tel_num] zu sprechen?
Bitte geben Sie mir einen geeigneten Zeitpunkt f\\u00fcr einen Anruf,
damit wir das Problem so schnell wie m\\u00f6glich f\\u00fcr Ihr
[acc_num] l\\u00f6sen k\\u00f6nnen.\",\n          \"Hello [Name], we
will provide you with resources and guidelines for using MATLAB in
data analytics. You can access tutorials and documentation via email.
If you need further assistance, please contact us at [Tel Num] to
discuss the next steps.\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"type\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Request\",\n          \"Change\",\n
\"Incident\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"queue\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 10,\n        \"samples\": [\n
\"Returns and Exchanges\",\n          \"Customer Service\",\n
\"Billing and Payments\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
```

```
\"column\": \"priority\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"low\",\n          \"medium\",\n          \"high\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"language\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"en\",\n
\"de\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_1\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 148,\n        \"samples\": [\n
\"Technical,Customer,Integration,Documentation,Guidance\",\n
\"Privacy\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_2\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 204,\n        \"samples\": [\n
\"Compliance\",\n          \"Outage\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"tag_3\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\":
344,\n        \"samples\": [\n          \"Digital Marketing\",\n
\"Social Media Marketing\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"tag_4\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 481,\n
\"samples\": [\n          \"AccessControl\",\n          \"Data
Analysis\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_5\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 578,\n        \"samples\": [\n
\"Datenauswertung\",\n          \"GCP\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"tag_6\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\":
566,\n        \"samples\": [\n          \"Workload\",\n
\"Brand\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_7\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 492,\n        \"samples\": [\n
\"Git\",\n          \"Notification\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"tag_8\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\":
386,\n        \"samples\": [\n          \"Code\",\n          \"Demo\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

## Filtering by English Tickets

```
df = df[df['language'] == 'en']
print("Shape of filtered dataframe:", df.shape)

df.head()
```

```
Shape of filtered dataframe: (11923, 15)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 11923,\n  \"fields\":
[\n    {\n      \"column\": \"subject\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 10891,\n
\"samples\": [\n          \"Medical Data Encryption Encounter
Issues\",\n          \"Issues with Billing\",\n          \"Urgent
Assistance Needed for Unauthorized Access to Patient Records\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"body\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\":
11922,\n        \"samples\": [\n          \"The encryption process for
medical data has failed, leading to the exposure of patient
information because of an outdated version of Avast antivirus software
and incompatible Airtable plugins.\",\n          \"I am encountering
problems with incorrect invoice totals due to a billing system glitch.
Despite restarting, relogging, and resynchronizing data, the issue
still exists. I would greatly appreciate your timely assistance in
addressing this concern.\",\n          \"Could you provide detailed
information on the Xero integration for project management SaaS? Your
support is greatly appreciated. We look forward to hearing back
soon.\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"answer\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 11920,\n        \"samples\": [\n          \"We
will provide API guides, setup tutorials, and relevant documentation
to assist with integrating SaaS with Microsoft SQL Server 2019. Please
allow us to send this information. You can also contact us at
<tel_num> for a detailed discussion on system requirements and
limitations.\",\n          \"Noted the email about the team meeting
schedule arrangement issue. Please share more details about the
conflict. I am ready to assist in resolving it as soon as
possible.\",\n          \"We are here to help with the integration
issue between QuickBooks Online and Smartsheet. To better assist you,
could you please provide the version numbers of the applications and
the exact error message you are receiving while attempting to sync?
This information will help us determine the cause of the issue and
provide a suitable solution. If needed, we can schedule a call at your
convenience to discuss and work on resolving the problem. Please let
me know a suitable time for contact at <tel_num>.\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"type\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 4,\n
```

```
\"samples\": [\n          \"Problem\",\n          \"Change\",\n
\"Request\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"queue\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 10,\n        \"samples\": [\n
\"Sales and Pre-Sales\",\n          \"Technical Support\",\n
\"Service Outages and Maintenance\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"priority\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"medium\",\n
\"high\",\n          \"low\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"language\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          \"en\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_1\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 87,\n        \"samples\": [\n
\"Pricing\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_2\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 176,\n        \"samples\": [\n
\"Campaign\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_3\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 297,\n        \"samples\": [\n
\"Online\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_4\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 400,\n        \"samples\": [\n
\"Ad\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_5\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 477,\n        \"samples\": [\n
\"Tutorial\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_6\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 468,\n        \"samples\": [\n
\"Real-time\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_7\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 419,\n        \"samples\": [\n
\"SaaS\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"tag_8\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 332,\n        \"samples\": [\n
\"Keras\"\n        ],\n        \"semantic_type\": \"\",\n
```

\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}

## Dropping Any Unnecessary Columns

```
# Extra Columns that we don't intend to use
misc_metadata = ['tag_1','tag_2','tag_3','tag_4',
                 'tag_5', 'tag_6', 'tag_7', 'tag_8']

df = df.drop(misc_metadata, axis=1)

df.head()
```

{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 11923,\n  \"fields\":
[\n    {\n      \"column\": \"subject\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 10891,\n
\"samples\": [\n          \"Medical Data Encryption Encounter
Issues\",\n          \"Issues with Billing\",\n          \"Urgent
Assistance Needed for Unauthorized Access to Patient Records\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"body\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\":
11922,\n        \"samples\": [\n          \"The encryption process for
medical data has failed, leading to the exposure of patient
information because of an outdated version of Avast antivirus software
and incompatible Airtable plugins.\",\n          \"I am encountering
problems with incorrect invoice totals due to a billing system glitch.
Despite restarting, relogging, and resynchronizing data, the issue
still exists. I would greatly appreciate your timely assistance in
addressing this concern.\",\n          \"Could you provide detailed
information on the Xero integration for project management SaaS? Your
support is greatly appreciated. We look forward to hearing back
soon.\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"answer\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 11920,\n        \"samples\": [\n          \"We
will provide API guides, setup tutorials, and relevant documentation
to assist with integrating SaaS with Microsoft SQL Server 2019. Please
allow us to send this information. You can also contact us at
<tel_num> for a detailed discussion on system requirements and
limitations.\",\n          \"Noted the email about the team meeting
schedule arrangement issue. Please share more details about the
conflict. I am ready to assist in resolving it as soon as
possible.\",\n          \"We are here to help with the integration
issue between QuickBooks Online and Smartsheet. To better assist you,
could you please provide the version numbers of the applications and
the exact error message you are receiving while attempting to sync?
This information will help us determine the cause of the issue and
provide a suitable solution. If needed, we can schedule a call at your
convenience to discuss and work on resolving the problem. Please let

```
me know a suitable time for contact at <tel_num>.\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"type\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 4,\n
\"samples\": [\n          \"Problem\",\n          \"Change\",\n
\"Request\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"queue\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 10,\n        \"samples\": [\n
\"Sales and Pre-Sales\",\n          \"Technical Support\",\n
\"Service Outages and Maintenance\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"priority\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"medium\",\n
\"high\",\n          \"low\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"language\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          \"en\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


## Combining "medium" and "high" priority into one category

```
# Replace 'medium' and 'high' with 'med/high'
df['priority'] = df['priority'].replace({'medium': 'med/high', 'high':
'med/high'})

df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 11923,\n  \"fields\":
[\n    {\n      \"column\": \"subject\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 10891,\n
\"samples\": [\n          \"Medical Data Encryption Encounter
Issues\",\n          \"Issues with Billing\",\n          \"Urgent
Assistance Needed for Unauthorized Access to Patient Records\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"body\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\":
11922,\n        \"samples\": [\n          \"The encryption process for
medical data has failed, leading to the exposure of patient
information because of an outdated version of Avast antivirus software
and incompatible Airtable plugins.\",\n          \"I am encountering
problems with incorrect invoice totals due to a billing system glitch.
Despite restarting, relogging, and resynchronizing data, the issue
still exists. I would greatly appreciate your timely assistance in
addressing this concern.\",\n          \"Could you provide detailed
information on the Xero integration for project management SaaS? Your
support is greatly appreciated. We look forward to hearing back
```

```
soon.\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"answer\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 11920,\n        \"samples\": [\n          \"We
will provide API guides, setup tutorials, and relevant documentation
to assist with integrating SaaS with Microsoft SQL Server 2019. Please
allow us to send this information. You can also contact us at
<tel_num> for a detailed discussion on system requirements and
limitations.\",\n          \"Noted the email about the team meeting
schedule arrangement issue. Please share more details about the
conflict. I am ready to assist in resolving it as soon as
possible.\",\n          \"We are here to help with the integration
issue between QuickBooks Online and Smartsheet. To better assist you,
could you please provide the version numbers of the applications and
the exact error message you are receiving while attempting to sync?
This information will help us determine the cause of the issue and
provide a suitable solution. If needed, we can schedule a call at your
convenience to discuss and work on resolving the problem. Please let
me know a suitable time for contact at <tel_num>.\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"type\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 4,\n
\"samples\": [\n          \"Problem\",\n          \"Change\",\n
\"Request\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"queue\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 10,\n        \"samples\": [\n
\"Sales and Pre-Sales\",\n          \"Technical Support\",\n
\"Service Outages and Maintenance\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"priority\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"low\",\n
\"med/high\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"language\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          \"en\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


## Missing Values & Duplicates Analysis

```
# Check for missing values in each column
print("\nMissing Values by Column:")
print(df.isnull().sum())

# Check for duplicates based on a subset of columns
```

```
duplicate_rows = df.duplicated(subset=["subject", "body", "priority"])
print(f"\nNumber of duplicate records: {duplicate_rows.sum()}")
```

```
Missing Values by Column:
subject     1032
body           1
answer         3
type           0
queue          0
priority       0
language       0
dtype: int64

Number of duplicate records: 0
```

## Imputing Null Subject Lines and Combining Request Data

```
# Imputing Null Subject Lines
df['subject'].fillna('[No Subject]', inplace=True)

print("Null Subject Values:", df['subject'].isnull().sum())
```

```
Null Subject Values: 0

<ipython-input-16-9828db89b150>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df['subject'].fillna('[No Subject]', inplace=True)
```

```
# Combining subject and body data into one column
df["combined_request"] = df.apply(
    lambda row: f"Subject: {row['subject']} Body: {row['body']}",
axis=1
).fillna('')

df["combined_request"].head()
```

```
1    Subject: Customer Support Inquiry Body: Seekin...
2    Subject: Data Analytics for Investment Body: I...
4    Subject: Security Body: Dear Customer Support,...
```

```
5    Subject: Concerns About Securing Medical Data ...
7    Subject: Problem with Integration Body: The in...
Name: combined_request, dtype: object
```

# Exploratory Data Analysis

## Descriptive Statistics

```
# Omitting text columns
df_categories = df[['type','queue','priority']]

df_categories.describe()
```


```
{"summary":"{\n  \"name\": \"df_categories\",\n  \"rows\": 4,\n
\"fields\": [\n    {\n      \"column\": \"type\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n          4,\n
\"4642\",\n          \"11923\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"queue\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          10,\n          \"3412\",\n          \"11923\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"priority\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          2,\n          \"9523\",\n          \"11923\"\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```


```
all_words = [word for text in df['combined_request'] for word in
text.split()]
word_counts = Counter(all_words)

# Set threshold to remove rare words (e.g., words appearing <5 times)
vocab_size = sum(1 for count in word_counts.values() if count >= 5)
max_words = min(vocab_size, 20000)
print('Max Words:',max_words)

text_lengths = np.array([len(text.split()) for text in
df['combined_request']])

# Set max_length to the 95th percentile (avoiding extreme outliers)
max_length = int(np.percentile(text_lengths, 95))
print(f"Optimal max_length: {max_length}")
```

```
Max Words: 5296
Optimal max_length: 129
```

## Checking for Imbalance

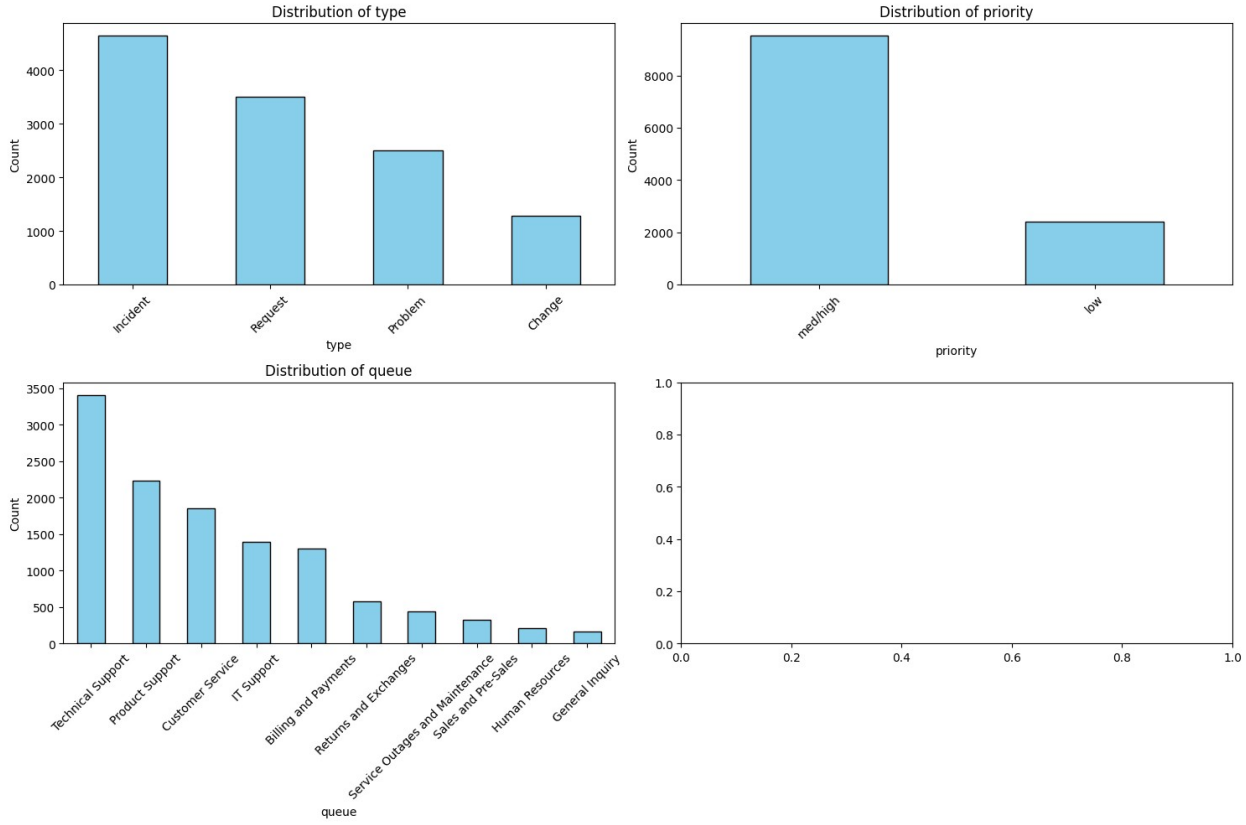
```
for col in df_categories:
  print(df_categories[col].value_counts())

print('------------------------------------------------------------')
```

```
type
Incident    4642
Request     3498
Problem     2498
Change      1285
Name: count, dtype: int64
------------------------------------------------------------
queue
Technical Support                   3412
Product Support                     2232
Customer Service                    1859
IT Support                          1391
Billing and Payments                1302
Returns and Exchanges                582
Service Outages and Maintenance      442
Sales and Pre-Sales                  330
Human Resources                      205
General Inquiry                      168
Name: count, dtype: int64
------------------------------------------------------------
priority
med/high    9523
low         2400
Name: count, dtype: int64
------------------------------------------------------------
```

## Visualizations

```
plot_categories = df[['type','priority','queue']]

# Set up subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))

# Flatten axes array for easy iteration
axes = axes.flatten()

# Iterate through each categorical column and plot value counts
for i, col in enumerate(plot_categories.columns):
    df_categories[col].value_counts().plot(kind='bar', ax=axes[i],
color='skyblue', edgecolor='black')
    axes[i].set_title(f'Distribution of {col}')
    axes[i].set_ylabel('Count')
    axes[i].set_xlabel(col)
```

```
    axes[i].tick_params(axis='x', rotation=45)  # Rotate x-axis labels for better readability

# Adjust layout to prevent overlapping
plt.tight_layout()
plt.show()
```

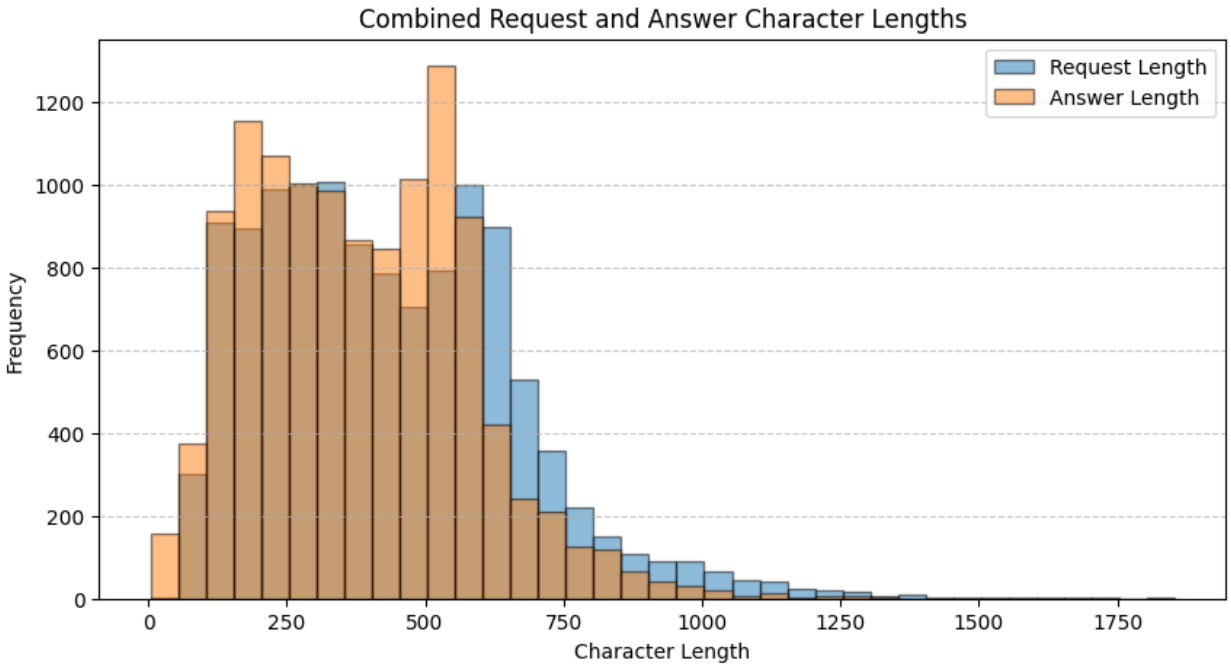



## Body and Answer Character Lengths

```
# Calculate character lengths
df["body_length"] = df["combined_request"].str.len()
df["answer_length"] = df["answer"].str.len()
df["combined_length"] = df["body_length"] + df["answer_length"]

# Define bin width
bin_width = 50
min_length = min(df["body_length"].min(), df["answer_length"].min())
max_length = max(df["body_length"].max(), df["answer_length"].max())

print(f"Minimum Length: {min_length}")
print(f"Maximum Length: {max_length}")

# Create bins with fixed width
bins = np.arange(min_length, max_length + bin_width, bin_width)
```

```
# Plot histogram
plt.figure(figsize=(10,5))
plt.hist(df["body_length"], bins=bins, alpha=0.5, label="Request
Length", edgecolor='black')
plt.hist(df["answer_length"], bins=bins, alpha=0.5, label="Answer
Length", edgecolor='black')
plt.xlabel("Character Length")
plt.ylabel("Frequency")
plt.title("Combined Request and Answer Character Lengths")
plt.legend()
plt.grid(axis="y", linestyle="--", alpha=0.7)

plt.show()
```

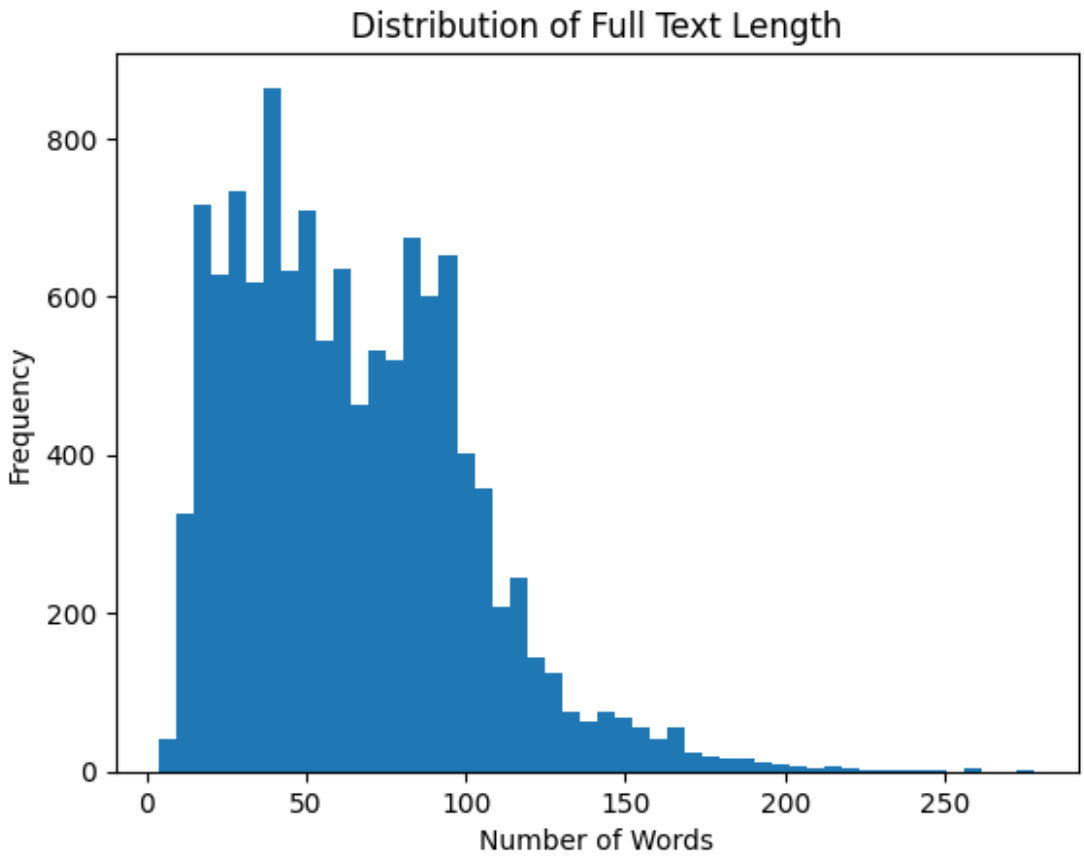
```
Minimum Length: 4.0
Maximum Length: 1840
```



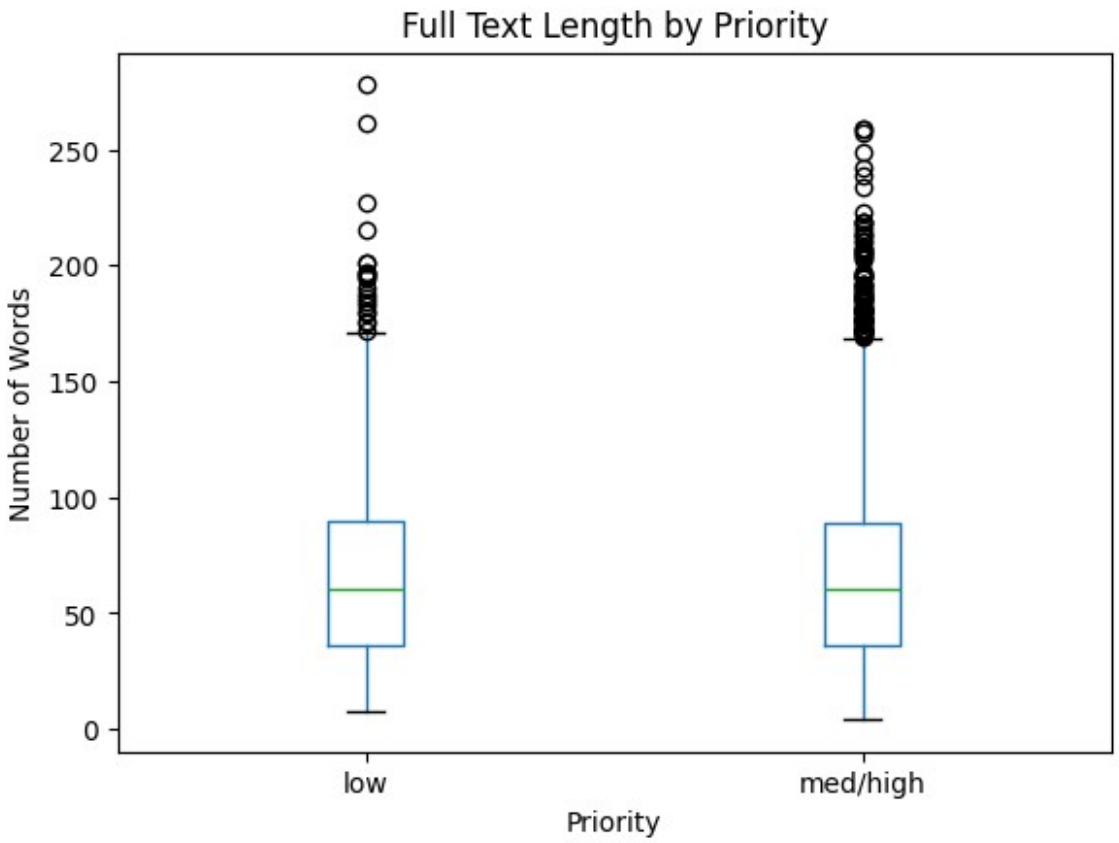

```
df['text_length'] = df['combined_request'].apply(lambda x:
len(str(x).split()))
print("\nFull Text(Subject + Body) Length Statistics:")
print(df['text_length'].describe())

plt.figure()
plt.hist(df['text_length'], bins=50)
plt.title("Distribution of Full Text Length")
plt.xlabel("Number of Words")
plt.ylabel("Frequency")
plt.show()
```

```
plt.figure()
df.boxplot(column='text_length', by='priority', grid=False)
plt.title("Full Text Length by Priority")
plt.suptitle("")
plt.xlabel("Priority")
plt.ylabel("Number of Words")
plt.show()

plt.figure()
df.boxplot(column='answer_length', by='priority', grid=False)
plt.title("Answer Length by Priority")
plt.suptitle("")
plt.xlabel("Priority")
plt.ylabel("Number of Words")
plt.show()
```

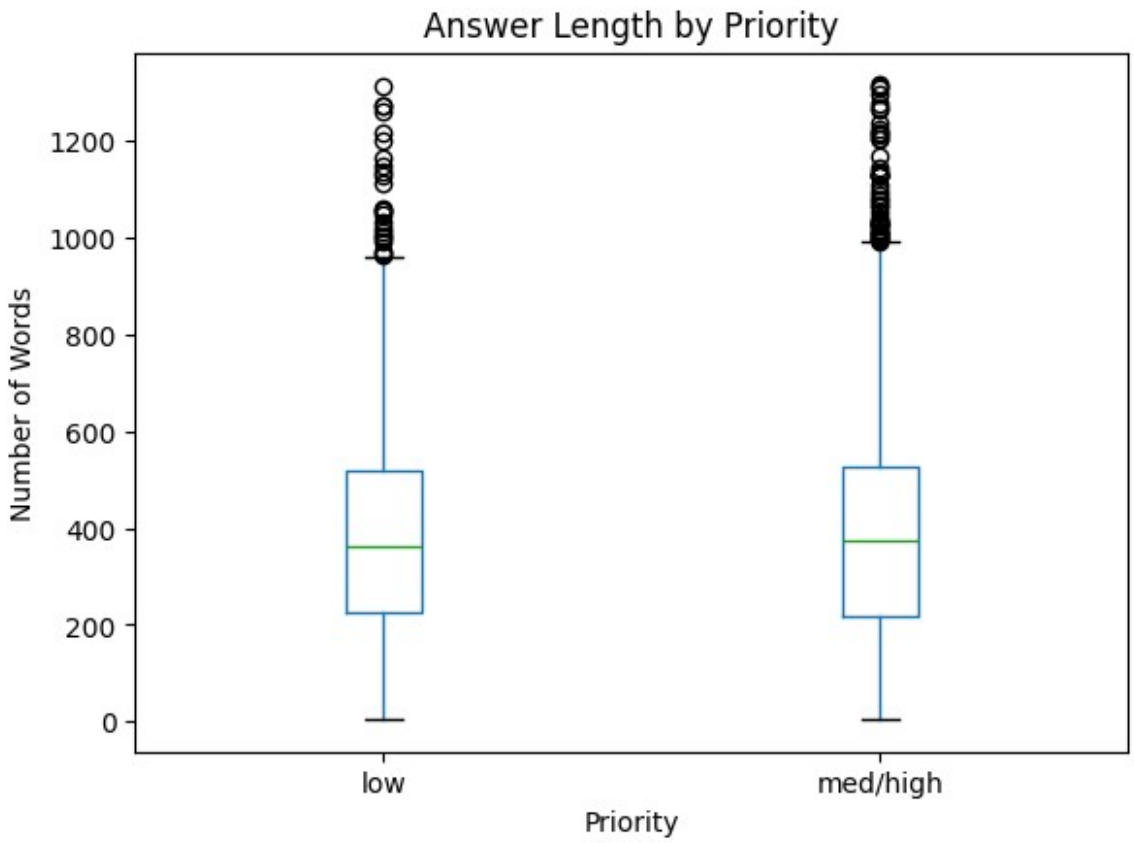
```
Full Text(Subject + Body) Length Statistics:
count    11923.000000
mean        64.896167
std         36.444482
min          4.000000
25%         36.000000
50%         60.000000
75%         89.000000
max        278.000000
Name: text_length, dtype: float64
```




```
<Figure size 640x480 with 0 Axes>
```



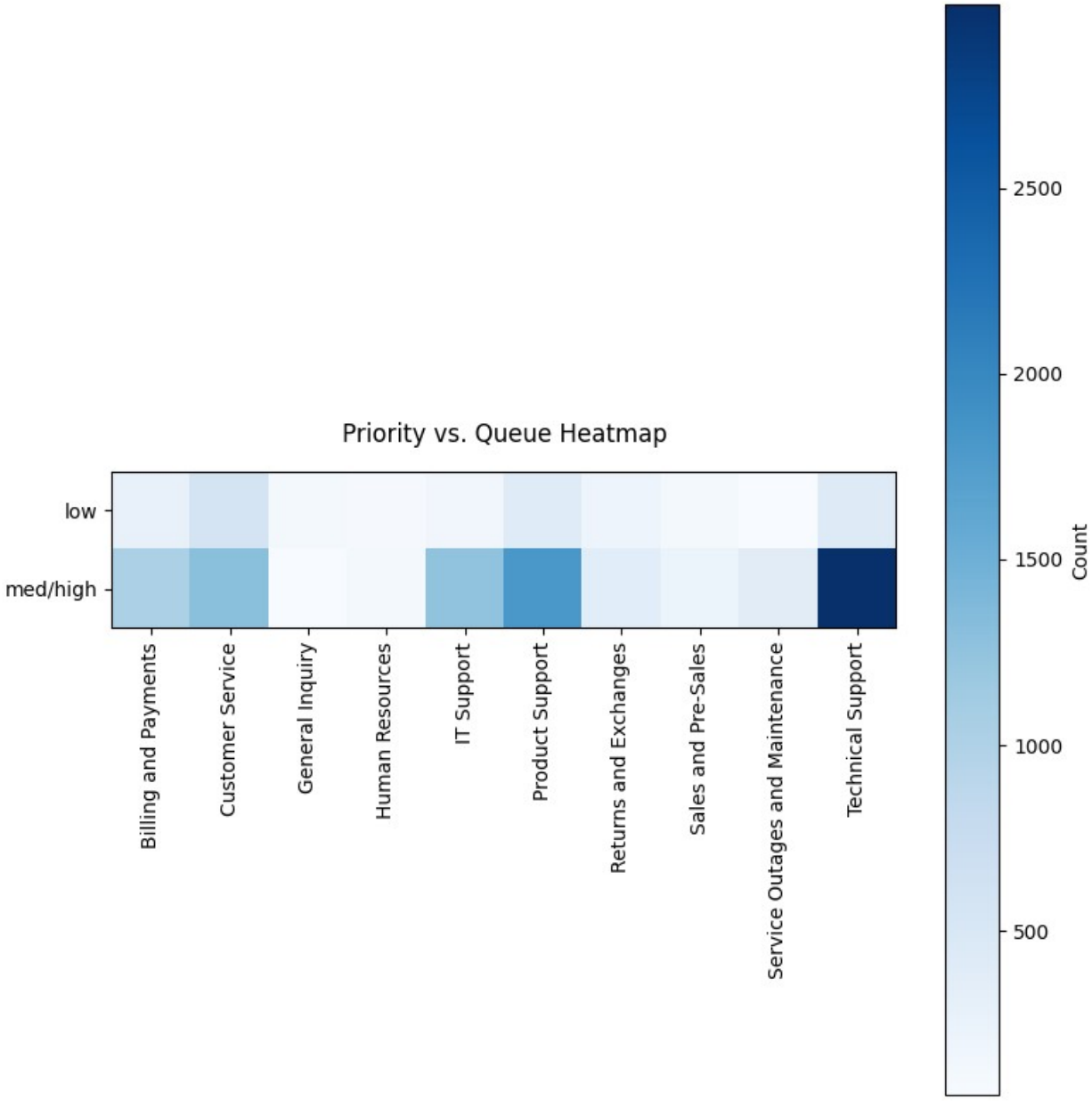

```
<Figure size 640x480 with 0 Axes>
```




Cross-tabulation (Priority vs. Queue)

```
if 'queue' in df.columns:
    priority_vs_queue = pd.crosstab(df['priority'], df['queue'])
    print("\nPriority vs. Queue Cross-Tab:")
    print(priority_vs_queue)

    # Create a square figure (e.g., 8x8)
    fig, ax = plt.subplots(figsize=(8, 8))

    # Display as an image with equal aspect ratio
    im = ax.imshow(priority_vs_queue, cmap='Blues', aspect='equal')

    # Add colorbar
    cbar = fig.colorbar(im, ax=ax)
    cbar.set_label("Count", rotation=90)

    # Set title
    ax.set_title("Priority vs. Queue Heatmap", pad=15)

    # Set tick labels
    ax.set_xticks(range(len(priority_vs_queue.columns)))
    ax.set_xticklabels(priority_vs_queue.columns, rotation=90)
    ax.set_yticks(range(len(priority_vs_queue.index)))
    ax.set_yticklabels(priority_vs_queue.index)
```

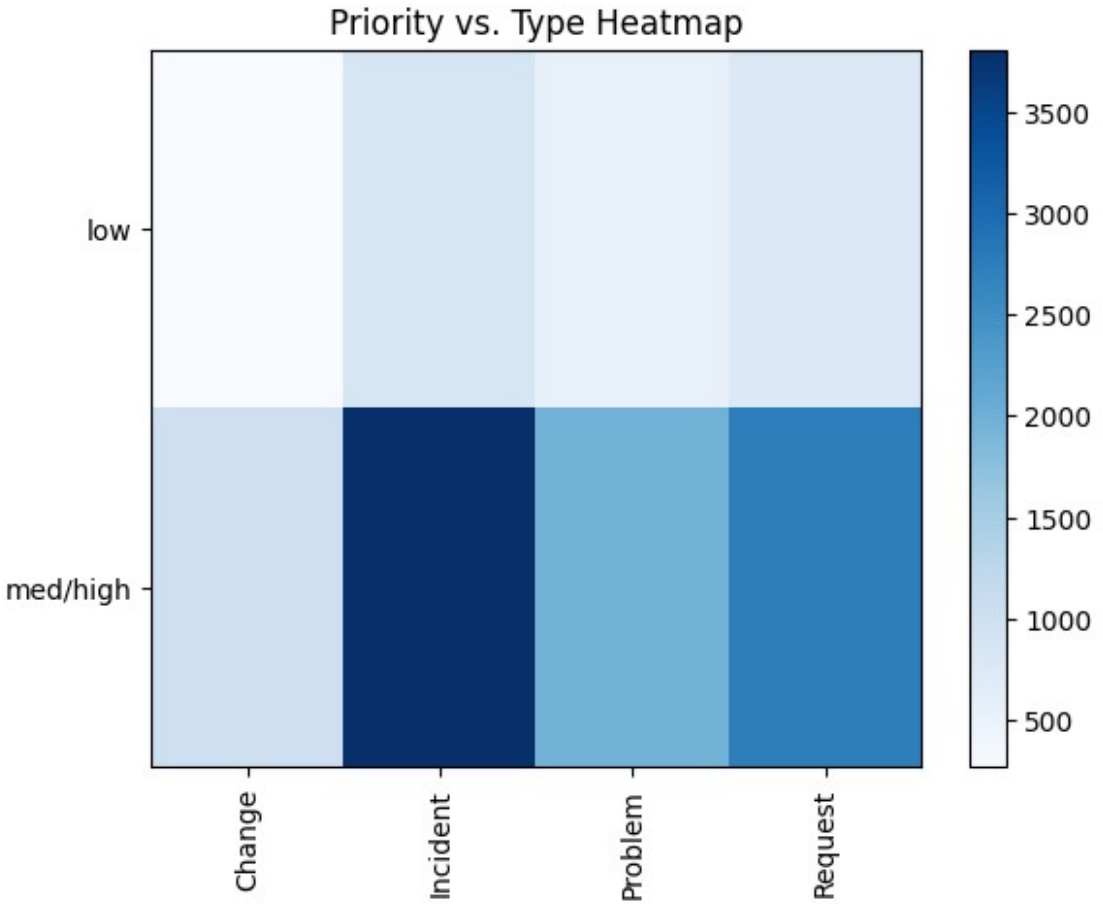
```
    # Adjust layout to avoid label cutoff
    plt.tight_layout()
    plt.show()


Priority vs. Queue Cross-Tab:
queue     Billing and Payments  Customer Service  General Inquiry  \
priority
low                        273               566              110
med/high                  1029              1293               58

queue     Human Resources  IT Support  Product Support  Returns and
Exchanges  \
priority

low                    92         151               413
204
med/high              113        1240              1819
378

queue     Sales and Pre-Sales  Service Outages and Maintenance  \
priority
low                       116                                58
med/high                  214                               384

queue     Technical Support
priority
low                     417
med/high               2995
```




Cross-tabulation (Priority vs. Type)

```
if 'type' in df.columns:
    priority_vs_queue = pd.crosstab(df['priority'], df['type'])
    print("\nPriority vs. Type Cross-Tab:")
    print(priority_vs_queue)

    plt.figure()
    plt.imshow(priority_vs_queue, cmap='Blues', aspect='auto')
    plt.colorbar()
    plt.title("Priority vs. Type Heatmap")
    plt.xticks(ticks=range(len(priority_vs_queue.columns)),
labels=priority_vs_queue.columns, rotation=90)
```

```
    plt.yticks(ticks=range(len(priority_vs_queue.index)),
labels=priority_vs_queue.index)
    plt.show()


Priority vs. Type Cross-Tab:
type      Change  Incident  Problem  Request
priority
low          268       836      538      758
med/high    1017      3806     1960     2740
```




# Feature Eningeering

## Gathering Categorical Features

```
# Limiting data for our text classifier - priority = target
model_data = df[["combined_request","type","priority"]]

model_data.head()

{"summary":"{\n  \"name\": \"model_data\",\n  \"rows\": 11923,\n
\"fields\": [\n    {\n      \"column\": \"combined_request\",\n
```

```
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 11923,\n        \"samples\": [\n
\"Subject: Request for Information on Smart-Thermometer Reporting
Capabilities Body: I am reaching out to inquire about the reporting
features available in Smart-Thermometer, particularly regarding data
analytics integration. Could you please provide me with details on the
types of reports that can be generated and the data that can be
exported? Additionally, I would appreciate any guidance on how to
integrate this data into our analytics platform. Thank you for your
time and assistance. I look forward to hearing from you soon.\",\n
\"Subject: Subject: Assistance Needed with Incorrect Invoice Totals
Body: I am encountering problems with incorrect invoice totals due to
a billing system glitch. Despite restarting, relogging, and
resynchronizing data, the issue still exists. I would greatly
appreciate your timely assistance in addressing this concern.\",\n
\"Subject: Security Incident Notification Body: Dear Customer Support,
I have encountered an incident that involves potential unauthorized
access to medical data, which may have occurred due to a security
vulnerability in our systems. So far, we have conducted preliminary
assessments and implemented immediate security measures to safeguard
access. Our team is working diligently to investigate the matter and
ensure the integrity of our systems. We appreciate your attention to
this matter and look forward to your assistance in resolving the
issue. Please let us know the next steps to take. Thank you.\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"type\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
4,\n        \"samples\": [\n          \"Problem\",\n
\"Change\",\n          \"Request\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"priority\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"low\",\n
\"med/high\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"model_data"}
```


## Undersampling

```
## Reshape priority column for sampling
X = df[['combined_request','type']]
y = df['priority']  # Labels

# Apply undersampling
undersampler = RandomUnderSampler(sampling_strategy='not minority',
random_state=42)
X_resampled, y_resampled = undersampler.fit_resample(X, y)

# Create a new balanced DataFrame
```

```
df_balanced = pd.DataFrame({'priority': y_resampled})

# Create a new balanced DataFrame
df_balanced = pd.DataFrame(X_resampled, columns=X.columns)
df_balanced['priority'] = y_resampled

# Print results
print("Balanced Samples:")
print(df_balanced['priority'].value_counts())
print("New Shape:", df_balanced.shape)
df_balanced.head()
```

```
Balanced Samples:
priority
low         2400
med/high    2400
Name: count, dtype: int64
New Shape: (4800, 3)
```


```
{"summary":"{\n  \"name\": \"df_balanced\",\n  \"rows\": 4800,\n
\"fields\": [\n    {\n      \"column\": \"combined_request\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 4800,\n        \"samples\": [\n
\"Subject: Request for Information on Digital Strategies and Services
Body: Hello Customer Support, I am writing to inquire about the
digital strategies and services your company offers to aid in brand
growth. Could you provide more information on how your company
approaches digital marketing services for your clients? I would
greatly appreciate any detailed information you can provide about the
strategies and services you offer. Thank you, I look forward to
hearing from you at your earliest convenience.\",\n
\"Subject: Guidance for Integrating Project Management Software Body:
Hello Customer Support, I am writing to inquire about detailed setup
procedures for linking project management software with our current
systems. Our aim is to optimize our workflow and enhance team
collaboration. Would you be able to supply us with a comprehensive,
step-by-step guide on how to perform this integration? Additionally,
we would greatly appreciate any supplementary information or
references that could assist us in initiating the process.
Specifically, we are keen to understand how to synchronize tasks,
allocate roles, and monitor progress. We are excited about the
prospect of hearing from you and proceeding with the integration.
Thank you for your valuable time and support. Best regards, [Your
Name]\",\n          \"Subject: Assistance with Integrating Mailchimp
and Microsoft Office 365 for Enhanced Analytics Body: Seeking guidance
on integrating Mailchimp with Microsoft Office 365 for investment
analytics to better track and analyze customer engagement. Would
appreciate a detailed, step-by-step guide along with relevant
resources to facilitate the integration process. Thank you for your
support.\"\n        ],\n        \"semantic_type\": \"\",\n
```

```
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"type\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n
\"Problem\",\n          \"Request\",\n          \"Incident\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"priority\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"med/high\",\n          \"low\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df_balanced"}
```


## Encoding

```
# Label encoding for 'priority'
priority_le = LabelEncoder()
labels = priority_le.fit_transform(df_balanced['priority'])
print("Priority Classes:", priority_le.classes_)

# Label encoding for 'type'
type_le = LabelEncoder()
feature_inputs = type_le.fit_transform(df_balanced['type'])
print("Type Classes:", type_le.classes_)
```

```
Priority Classes: ['low' 'med/high']
Type Classes: ['Change' 'Incident' 'Problem' 'Request']
```

```
df_balanced.head()
```


```
{"summary":"{\n  \"name\": \"df_balanced\",\n  \"rows\": 4800,\n
\"fields\": [\n    {\n      \"column\": \"combined_request\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 4800,\n        \"samples\": [\n
\"Subject: Request for Information on Digital Strategies and Services
Body: Hello Customer Support, I am writing to inquire about the
digital strategies and services your company offers to aid in brand
growth. Could you provide more information on how your company
approaches digital marketing services for your clients? I would
greatly appreciate any detailed information you can provide about the
strategies and services you offer. Thank you, I look forward to
hearing from you at your earliest convenience.\",\n
\"Subject: Guidance for Integrating Project Management Software Body:
Hello Customer Support, I am writing to inquire about detailed setup
procedures for linking project management software with our current
systems. Our aim is to optimize our workflow and enhance team
collaboration. Would you be able to supply us with a comprehensive,
step-by-step guide on how to perform this integration? Additionally,
we would greatly appreciate any supplementary information or
references that could assist us in initiating the process.
Specifically, we are keen to understand how to synchronize tasks,
```

```
allocate roles, and monitor progress. We are excited about the
prospect of hearing from you and proceeding with the integration.
Thank you for your valuable time and support. Best regards, [Your
Name]\",\n          \"Subject: Assistance with Integrating Mailchimp
and Microsoft Office 365 for Enhanced Analytics Body: Seeking guidance
on integrating Mailchimp with Microsoft Office 365 for investment
analytics to better track and analyze customer engagement. Would
appreciate a detailed, step-by-step guide along with relevant
resources to facilitate the integration process. Thank you for your
support.\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"type\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n
\"Problem\",\n          \"Request\",\n          \"Incident\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"priority\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"med/high\",\n          \"low\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df_balanced"}
```


## Text preprocessing

```
# Tokenize the text first
all_words = [word for text in model_data['combined_request'] for word
in text.split()]
word_counts = Counter(all_words)

# Set threshold to remove rare words (e.g., words appearing <5 times)
vocab_size = sum(1 for count in word_counts.values() if count >= 5)
max_words = min(vocab_size, 20000)
print('Max Words:',max_words)

text_lengths = np.array([len(text.split()) for text in
model_data['combined_request']])

# Set max_length to the 95th percentile (avoiding extreme outliers)
max_length = int(np.percentile(text_lengths, 95))
print(f"Optimal max_length: {max_length}")

Max Words: 5296
Optimal max_length: 129

# Tokenize text data
tokenizer = Tokenizer(num_words=max_words, oov_token="<OOV>",
                      filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n')
tokenizer.fit_on_texts(df_balanced['combined_request'])

# Convert text into sequences
```

```
sequences =
tokenizer.texts_to_sequences(df_balanced['combined_request'])

# Pad sequences
padded_sequences = pad_sequences(sequences, maxlen=max_length,
padding='post', truncating='post')

# Reshaping and combining input features
feature_inputs = feature_inputs.reshape(-1,1)
features_combined = np.concatenate([padded_sequences,feature_inputs],
axis=1)

# Convert labels to a NumPy array
labels = np.array(labels)

padded_sequences[0]
```

```
array([  7,  17,  13, 338,  38, 158,   9, 125,  37,  20,   5,  23,
184,
        15,  28,  13,  51, 338,  38, 158,  32,  33,  93, 165,   5,
173,
        73,  56,  19,  58,  17,  13,   2, 242, 354,   5,  26, 121,
183,
         2, 158,   4, 362,  51, 737, 401,   2,  28,  83, 265,  18,
6,
        41, 300,   4, 103,  15, 109, 147,  11, 278, 279,  46,   6,
130,
        27, 104,  30,  62,  44,  41, 144,  44,  98,  60,   6,   8,
11,
        29,   4,   5,  64,  65,   3,  11,  76,   0,   0,   0,   0,
0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0],
      dtype=int32)
```

## Save tokenizer

```
with open("tokenizer.pkl", "wb") as handle:
    pickle.dump(tokenizer, handle)
```

## Train Test Split

```
x_train,x_test,y_train,y_test = train_test_split(

features_combined,labels,test_size=0.2,random_state=42,shuffle=True,
    stratify=labels)
```

# Modeling

## Model Architecture

```
# Binary classification
num_classes=2

# Setting this pretty low since we're dealing with a relatively small
vocab
embedding_dim = 50

# Setting an initial amount of 32, which is cut in half at the second
LSTM layer
lstm_units = 32

# Define the model
model = Sequential([
    Embedding(input_dim=max_words, output_dim=embedding_dim,
input_length=max_length),
    Bidirectional(LSTM(lstm_units, return_sequences=True)),
    Dropout(0.5),
    Bidirectional(LSTM(lstm_units // 2, return_sequences=True)),
    GlobalMaxPooling1D(),  # Pooling over the entire sequence
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')  # Change to single output node
with sigmoid
])

# Compile the model for multi-class classification
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy', Precision(), Recall()]
                )

# Model Summary
model.summary()
```

```
Model: "sequential_1"
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━
┓
┃ Layer (type)                         ┃ Output Shape                ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━
┩
│ embedding_1 (Embedding)              │ ?                           │
0 (unbuilt) │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────
┤
```

```
| bidirectional_2 (Bidirectional)      | ?                           |
0 (unbuilt) |
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
| dropout_2 (Dropout)                  | ?                           |
0 |
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
| bidirectional_3 (Bidirectional)      | ?                           |
0 (unbuilt) |
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
| global_max_pooling1d_1               | ?                           |
0 |
| (GlobalMaxPooling1D)                 |                             |
|
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
| dense_2 (Dense)                      | ?                           |
0 (unbuilt) |
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
| dropout_3 (Dropout)                  | ?                           |
0 |
├──────────────────────────────────────┼─────────────────────────────┼
──────────────────┤
| dense_3 (Dense)                      | ?                           |
0 (unbuilt) |
└──────────────────────────────────────┴─────────────────────────────┴
──────────────────┘

 Total params: 0 (0.00 B)

 Trainable params: 0 (0.00 B)

 Non-trainable params: 0 (0.00 B)
```

## Model Training

```
'''
Monitors the training loss of the model and terminates the training
process
if the loss doesn't improve for 10 consecutive epochs.
'''
early_stopping_callback =
tf.keras.callbacks.EarlyStopping(monitor='val_loss',

patience=10,

restore_best_weights=True)
```

```
"""
Saves the best model (best_model.h5) in workign directory.
Ensures the model is only saved if the monitored metric (`val_loss`)
improves.
Since we want to minimize the validation loss, we set this to `min`.
"""
checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(
    'best_model.h5',
    monitor='val_loss',
    save_best_only=True,
    mode='min',
    verbose=1
)

# Small Training session, we'll adjust epochs later
history = model.fit(
    x_train, y_train,
    epochs=30,
    batch_size=32,
    validation_data=(x_test, y_test),
    callbacks=[early_stopping_callback, checkpoint_callback]
)
```

```
Epoch 1/30
119/120 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.4942 - loss:
0.6938 - precision_1: 0.5019 - recall_1: 0.7206
Epoch 1: val_loss improved from inf to 0.69293, saving model to
best_model.h5

WARNING:absl:You are saving your model as an HDF5 file via
`model.save()` or `keras.saving.save_model(model)`. This file format
is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')` or
`keras.saving.save_model(model, 'my_model.keras')`.

 120/120 ━━━━━━━━━━━━━━━━━━━━ 18s 38ms/step - accuracy: 0.4943 - loss:
0.6938 - precision_1: 0.5019 - recall_1: 0.7178 - val_accuracy: 0.5115
- val_loss: 0.6929 - val_precision_1: 0.5062 - val_recall_1: 0.9312
Epoch 2/30
119/120 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - accuracy: 0.5300 - loss:
0.6916 - precision_1: 0.5248 - recall_1: 0.7379
Epoch 2: val_loss improved from 0.69293 to 0.69027, saving model to
best_model.h5

WARNING:absl:You are saving your model as an HDF5 file via
`model.save()` or `keras.saving.save_model(model)`. This file format
is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')` or
`keras.saving.save_model(model, 'my_model.keras')`.
```

```
 120/120 ━━━━━━━━━━━━━━━━━━━━ 3s 27ms/step - accuracy: 0.5299 - loss:
0.6916 - precision_1: 0.5248 - recall_1: 0.7347 - val_accuracy: 0.5323
- val_loss: 0.6903 - val_precision_1: 0.7925 - val_recall_1: 0.0875
Epoch 3/30
118/120 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.5969 - loss:
0.6735 - precision_1: 0.6360 - recall_1: 0.4365
Epoch 3: val_loss improved from 0.69027 to 0.66505, saving model to
best_model.h5

WARNING:absl:You are saving your model as an HDF5 file via
`model.save()` or `keras.saving.save_model(model)`. This file format
is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')` or
`keras.saving.save_model(model, 'my_model.keras')`.

 120/120 ━━━━━━━━━━━━━━━━━━━━ 3s 27ms/step - accuracy: 0.5975 - loss:
0.6732 - precision_1: 0.6361 - recall_1: 0.4391 - val_accuracy: 0.6125
- val_loss: 0.6651 - val_precision_1: 0.5831 - val_recall_1: 0.7896
Epoch 4/30
120/120 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - accuracy: 0.7253 - loss:
0.5537 - precision_1: 0.7277 - recall_1: 0.7306
Epoch 4: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 4s 29ms/step - accuracy: 0.7253 - loss:
0.5537 - precision_1: 0.7278 - recall_1: 0.7305 - val_accuracy: 0.6198
- val_loss: 0.6706 - val_precision_1: 0.6340 - val_recall_1: 0.5667
Epoch 5/30
118/120 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.8173 - loss:
0.4025 - precision_1: 0.8384 - recall_1: 0.7920
Epoch 5: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 5s 27ms/step - accuracy: 0.8170 - loss:
0.4030 - precision_1: 0.8379 - recall_1: 0.7918 - val_accuracy: 0.6375
- val_loss: 0.6992 - val_precision_1: 0.6422 - val_recall_1: 0.6208
Epoch 6/30
120/120 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.8827 - loss:
0.2949 - precision_1: 0.8900 - recall_1: 0.8754
Epoch 6: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 3s 27ms/step - accuracy: 0.8826 - loss:
0.2951 - precision_1: 0.8899 - recall_1: 0.8753 - val_accuracy: 0.6375
- val_loss: 0.8546 - val_precision_1: 0.6347 - val_recall_1: 0.6479
Epoch 7/30
119/120 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.9014 - loss:
0.2250 - precision_1: 0.9021 - recall_1: 0.9008
Epoch 7: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 6s 32ms/step - accuracy: 0.9013 - loss:
0.2253 - precision_1: 0.9020 - recall_1: 0.9008 - val_accuracy: 0.6313
- val_loss: 1.0120 - val_precision_1: 0.6388 - val_recall_1: 0.6042
Epoch 8/30
119/120 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - accuracy: 0.9300 - loss:
0.1766 - precision_1: 0.9273 - recall_1: 0.9357
Epoch 8: val_loss did not improve from 0.66505
```

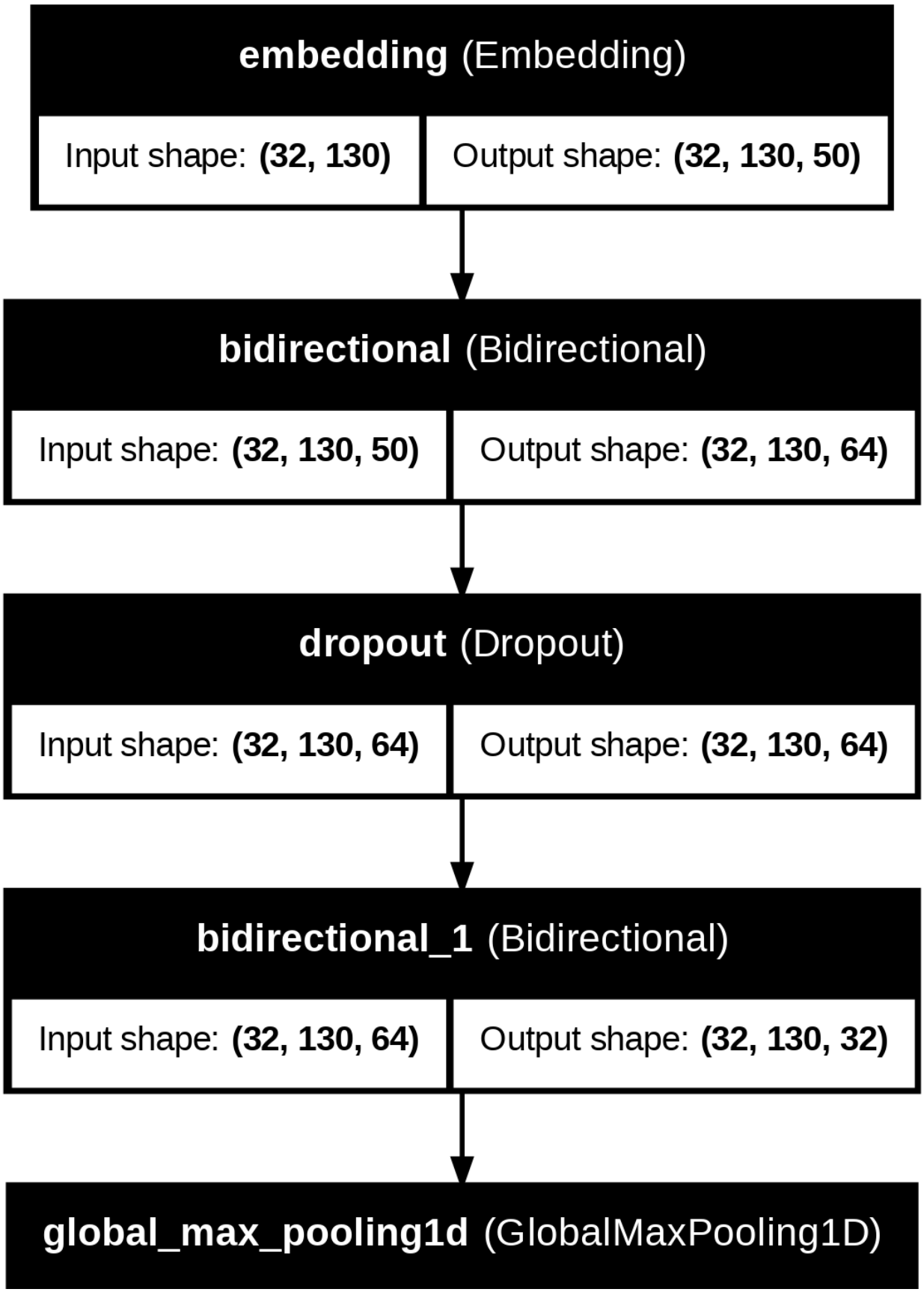
```
120/120 ━━━━━━━━━━━━━━━━━━━━ 3s 26ms/step - accuracy: 0.9299 - loss:
0.1767 - precision_1: 0.9271 - recall_1: 0.9357 - val_accuracy: 0.6375
- val_loss: 1.2462 - val_precision_1: 0.6701 - val_recall_1: 0.5417
Epoch 9/30
120/120 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - accuracy: 0.9476 - loss:
0.1331 - precision_1: 0.9388 - recall_1: 0.9564
Epoch 9: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 3s 26ms/step - accuracy: 0.9476 - loss:
0.1331 - precision_1: 0.9388 - recall_1: 0.9564 - val_accuracy: 0.6365
- val_loss: 1.4284 - val_precision_1: 0.6520 - val_recall_1: 0.5854
Epoch 10/30
118/120 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 0.9532 - loss:
0.1136 - precision_1: 0.9467 - recall_1: 0.9631
Epoch 10: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 6s 37ms/step - accuracy: 0.9530 - loss:
0.1136 - precision_1: 0.9464 - recall_1: 0.9630 - val_accuracy: 0.6667
- val_loss: 1.4918 - val_precision_1: 0.6493 - val_recall_1: 0.7250
Epoch 11/30
120/120 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.9589 - loss:
0.0956 - precision_1: 0.9456 - recall_1: 0.9739
Epoch 11: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 4s 27ms/step - accuracy: 0.9588 - loss:
0.0957 - precision_1: 0.9456 - recall_1: 0.9738 - val_accuracy: 0.6635
- val_loss: 1.6084 - val_precision_1: 0.6639 - val_recall_1: 0.6625
Epoch 12/30
120/120 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - accuracy: 0.9638 - loss:
0.0885 - precision_1: 0.9589 - recall_1: 0.9664
Epoch 12: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 6s 31ms/step - accuracy: 0.9639 - loss:
0.0885 - precision_1: 0.9589 - recall_1: 0.9665 - val_accuracy: 0.6583
- val_loss: 1.7104 - val_precision_1: 0.6603 - val_recall_1: 0.6521
Epoch 13/30
119/120 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.9707 - loss:
0.0791 - precision_1: 0.9620 - recall_1: 0.9795
Epoch 13: val_loss did not improve from 0.66505
120/120 ━━━━━━━━━━━━━━━━━━━━ 3s 28ms/step - accuracy: 0.9707 - loss:
0.0791 - precision_1: 0.9620 - recall_1: 0.9795 - val_accuracy: 0.6490
- val_loss: 2.1140 - val_precision_1: 0.6357 - val_recall_1: 0.6979
```

```python
# Predict on validation/test data
y_pred_probs = model.predict(x_test)

# Converts probabilities to 0 or 1
y_pred = (y_pred_probs > 0.5).astype("int32")

print("F1 Score:", f1_score(y_test, y_pred))
```

```
30/30 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
F1 Score: 0.6707964601769911
```

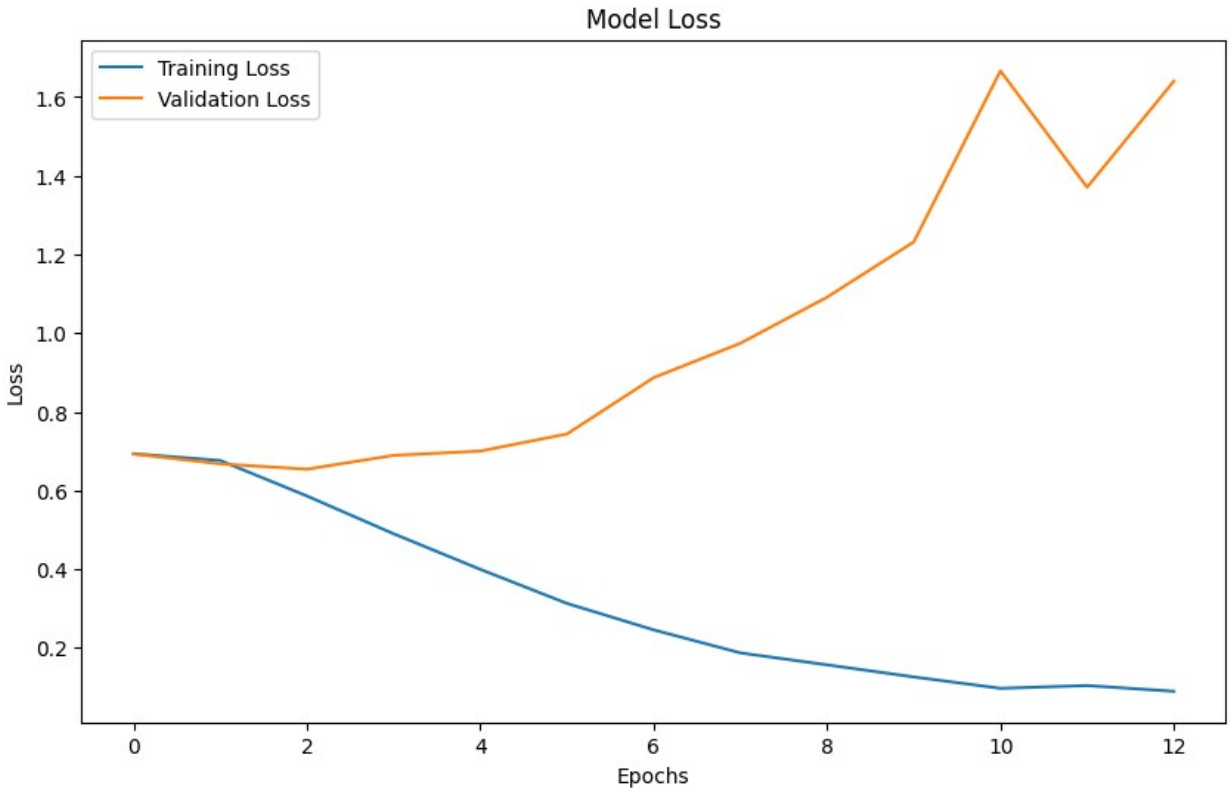
## Architecture Visualization

```
tf.keras.utils.plot_model(model,show_shapes=True,
show_layer_names=True)
```

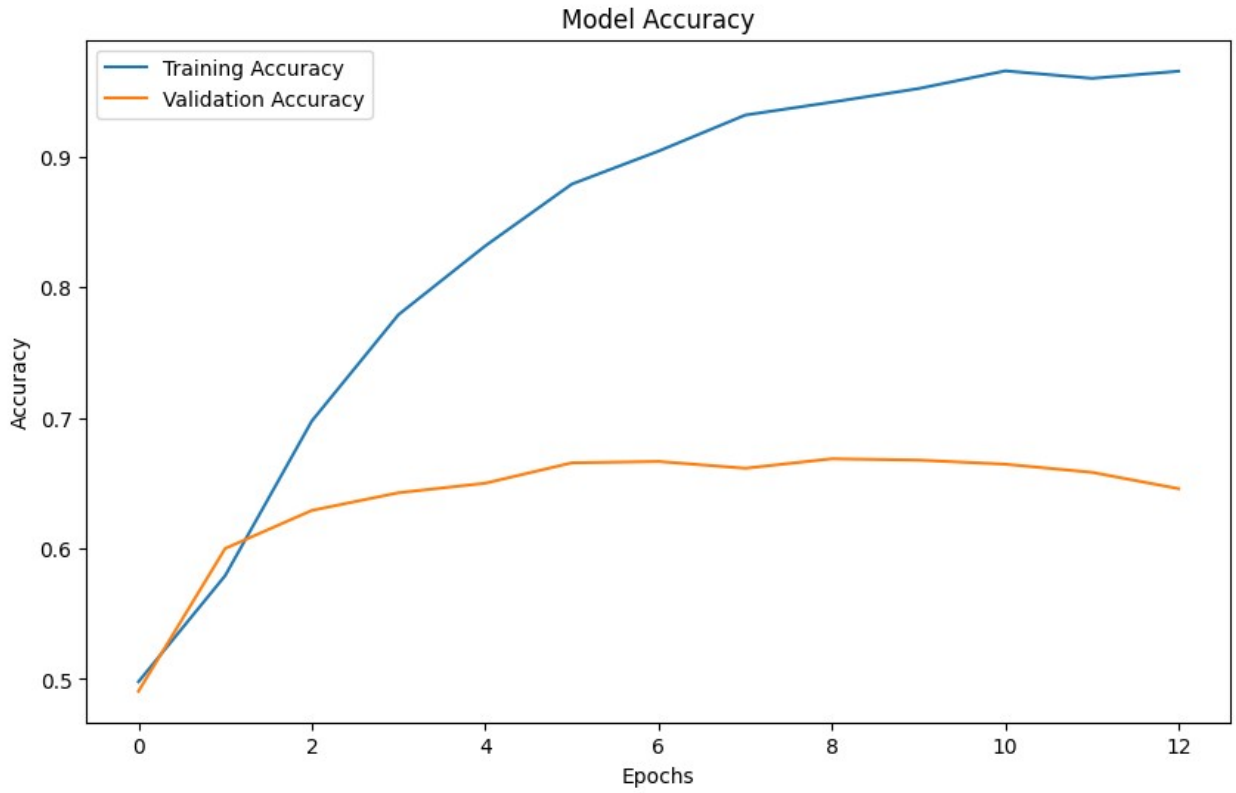

embedding (Embedding)
Input shape: (32, 130)
Output shape: (32, 130, 50)
bidirectional (Bidirectional)
Input shape: (32, 130, 50)
Output shape: (32, 130, 64)
dropout (Dropout)
Input shape: (32, 130, 64)
Output shape: (32, 130, 64)
bidirectional_1 (Bidirectional)
Input shape: (32, 130, 64)
Output shape: (32, 130, 32)
global_max_pooling1d (GlobalMaxPooling1D)

## Initial Evaluation

```
# Loss Plots
plt.figure(figsize=(10, 6))
plt.plot(history.history['loss'], label = 'Training Loss')
plt.plot(history.history['val_loss'], label = 'Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```




```
# Accuracy Plots
plt.figure(figsize=(10, 6))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```




## Hyperparameter Tuning

```
# HyperModel class for defining the model
class TextClassificationHyperModel(HyperModel):
    def build(self, hp):
        # Hyperparameters to tune
        embedding_dim = hp.Int('embedding_dim', min_value=32,
max_value=128, step=16)
        lstm_units = hp.Int('lstm_units', min_value=16, max_value=128,
step=16)
        dropout_rate = hp.Float('dropout_rate', min_value=0.2,
max_value=0.6, step=0.1)
        dense_units = hp.Int('dense_units', min_value=64,
max_value=256, step=64)

        # Define the model
        model = Sequential([
            Embedding(input_dim=max_words, output_dim=embedding_dim,
input_length=max_length),
            Bidirectional(LSTM(lstm_units, return_sequences=True)),
            Dropout(dropout_rate),
            Bidirectional(LSTM(lstm_units // 2,
return_sequences=True)),
            GlobalMaxPooling1D(),
            Dense(dense_units, activation='relu'),
```

```
            Dropout(dropout_rate),
            Dense(1, activation='sigmoid')
        ])

        # Compile the model
        model.compile(
            loss='binary_crossentropy',
            optimizer=Adam(),
            metrics=['accuracy',Precision(), Recall()]
        )

        return model

# Define hyperparameter tuner
tuner = RandomSearch(
    TextClassificationHyperModel(),
    objective='val_accuracy',
    max_trials=10,  # Number of models to try
    executions_per_trial=3,  # How many times to train each model
    directory='tuning_results',
    project_name='binary_classification_tuning'
)


# Fit the tuner to the data
tuner.search(x_train, y_train, epochs=10, validation_data=(x_test,
y_test))

# Get the best model
best_model = tuner.get_best_models(num_models=1)[0]

# Summarize the best model
best_model.summary()

# Save the best model
best_model.save('best_binary_model_after_tuning.h5')
```

```
Trial 10 Complete [00h 02m 17s]
val_accuracy: 0.6461805502573649

Best val_accuracy So Far: 0.6500000158945719
Total elapsed time: 00h 23m 05s

/usr/local/lib/python3.11/dist-packages/keras/src/saving/
saving_lib.py:757: UserWarning: Skipping variable loading for
optimizer 'adam', because it has 2 variables whereas the saved
optimizer has 36 variables.
  saveable.load_own_variables(weights_store.get(inner_path))

Model: "sequential"
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape                ┃ Param #         ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ embedding (Embedding)                │ (32, 130, 112)              │ 593,152         │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ bidirectional (Bidirectional)        │ (32, 130, 96)               │ 61,824          │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dropout (Dropout)                    │ (32, 130, 96)               │ 0               │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ bidirectional_1 (Bidirectional)      │ (32, 130, 48)               │ 23,232          │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ global_max_pooling1d                 │ (32, 48)                    │ 0               │
│ (GlobalMaxPooling1D)                 │                             │                 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense (Dense)                        │ (32, 256)                   │ 12,544          │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dropout_1 (Dropout)                  │ (32, 256)                   │ 0               │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_1 (Dense)                      │ (32, 1)                     │ 257             │
└──────────────────────────────────────┴─────────────────────────────┴─────────────────┘

 Total params: 691,009 (2.64 MB)

 Trainable params: 691,009 (2.64 MB)

 Non-trainable params: 0 (0.00 B)

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras
```

```
format, e.g. `model.save('my_model.keras')` or
`keras.saving.save_model(model, 'my_model.keras')`.
```

# XAi (SHAP)

## Splitting Inputs

```
# Extract the tokenized text sequences
x_train_text = x_train[:, :max_length]  # First `max_length` columns
are text

# Extract the numerical feature(s)
x_train_num = x_train[:, max_length:]  # Remaining columns are
numerical features

# Do the same for the test set
x_test_text = x_test[:, :max_length]
x_test_num = x_test[:, max_length:]

x_test_text
```

```
array([[   7,   58, 1325, ...,    0,    0,    0],
       [   7,  289,   13, ...,    0,    0,    0],
       [   7,  117,    7, ...,    0,    0,    0],
       ...,
       [   7,   74,    8, ...,    0,    0,    0],
       [   7,  117,    7, ...,    0,    0,    0],
       [   7,  308,   17, ...,    0,    0,    0]])
```

```
# Choose a small subset as background data
background_data = x_train[:10]  # Keep as a single concatenated input
sample_data = x_test[:10]

explainer = shap.KernelExplainer(model.predict, background_data)
shap_values = explainer.shap_values(sample_data)

# Plot results
shap.summary_plot(shap_values, sample_data)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
```


```
{"model_id":"8c8548c208414a19b48059500dee4d5d","version_major":2,"vers
ion_minor":0}
```


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 6s 9ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 7s 9ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 7s 9ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
```

```
721/721 ━━━━━━━━━━━━━━━━━━━━ 6s 8ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 7s 9ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 6s 9ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 6s 8ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 7s 9ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 7s 9ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
721/721 ━━━━━━━━━━━━━━━━━━━━ 6s 8ms/step

<Figure size 640x480 with 0 Axes>
```