Applied Machine Learning Coursework (001293509)

Import libraries

```
!pip install -U pyforest hvplot
!pip install --upgrade spacy
!python -m spacy download en_core_web_sm
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyforest
  Downloading pyforest-1.1.0.tar.gz (15 kB)
  Preparing metadata (setup.py) ... done
Collecting hvplot
  Downloading hvplot-0.8.3-py2.py3-none-any.whl (3.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.2/3.2 MB 54.1 MB/s eta 0:00:00
Requirement already satisfied: bokeh>=1.0.0 in /usr/local/lib/python3.9/dist-packages (from hvplot) (2.4.3)
Requirement already satisfied: holoviews>=1.11.0 in /usr/local/lib/python3.9/dist-packages (from hvplot) (1.15.4)
Requirement already satisfied: colorcet>=2 in /usr/local/lib/python3.9/dist-packages (from hvplot) (3.0.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (from hvplot) (23.0)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.9/dist-packages (from hvplot) (1.22.4)
Requirement already satisfied: panel>=0.11.0 in /usr/local/lib/python3.9/dist-packages (from hvplot) (0.14.4)
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-packages (from hvplot) (1.5.3)
Requirement already satisfied: param>=1.9.0 in /usr/local/lib/python3.9/dist-packages (from hvplot) (1.13.0)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.9/dist-packages (from bokeh>=1.0.0->hvplot) (6.2)
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.9/dist-packages (from bokeh>=1.0.0->hvplot) (8.4.(
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.9/dist-packages (from bokeh>=1.0.0->hvplot) (6.0)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.9/dist-packages (from bokeh>=1.0.0->hvplot) (3.1.2)
Requirement already satisfied: typing-extensions>=3.10.0 in /usr/local/lib/python3.9/dist-packages (from bokeh>=1.0.0->hv
Requirement already satisfied: pyct>=0.4.4 in /usr/local/lib/python3.9/dist-packages (from colorcet>=2->hvplot) (0.5.0)
Requirement already satisfied: pyviz-comms>=0.7.4 in /usr/local/lib/python3.9/dist-packages (from holoviews>=1.11.0->hvpl
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas->hvplot) (2022.7.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.9/dist-packages (from pandas->hvplot) (2.
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from panel>=0.11.0->hvplot) (2.27.1)
Requirement already satisfied: markdown in /usr/local/lib/python3.9/dist-packages (from panel>=0.11.0->hvplot) (3.4.3)
Requirement already satisfied: tqdm>=4.48.0 in /usr/local/lib/python3.9/dist-packages (from panel>=0.11.0->hvplot) (4.65.
Requirement already satisfied: setuptools>=42 in /usr/local/lib/python3.9/dist-packages (from panel>=0.11.0->hvplot) (67.
Requirement already satisfied: bleach in /usr/local/lib/python3.9/dist-packages (from panel>=0.11.0->hvplot) (6.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from Jinja2>=2.9->bokeh>=1.0.0-
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.8.1->pandas->h
Requirement already satisfied: webencodings in /usr/local/lib/python3.9/dist-packages (from bleach->panel>=0.11.0->hvplot
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.9/dist-packages (from markdown->panel>=(
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests->panel>=0.11.(
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests->panel>=0.1
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests->panel>
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests->panel>=0.11.0->hvpl
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from importlib-metadata>=4.4->markdow
Building wheels for collected packages: pyforest
  Building wheel for pyforest (setup.py) ... done
  Created wheel for pyforest: filename=pyforest-1.1.0-py2.py3-none-any.whl size=14606 sha256=c1e894106d0f6425194c2df6156c
  Stored in directory: /root/.cache/pip/wheels/d5/1a/3e/6193fe1c56168f5df4aef57d8411033ba4611881135d495727
Successfully built pyforest
Installing collected packages: pyforest, hvplot
Successfully installed hvplot-0.8.3 pyforest-1.1.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: spacy in /usr/local/lib/python3.9/dist-packages (3.5.1)
Collecting spacy
  Downloading spacy-3.5.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.6/6.6 MB 36.2 MB/s eta 0:00:00
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.9/dist-packages (from spacy) (2.0.7)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.9/dist-packages (from spacy) (8.1.9)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.9/dist-packages (from spacy) (2.4.6)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (0.7.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (23.0)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (3.3.0)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.9/dist-packages (from spacy) (3.0.8)
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-packages (from spacy) (67.6.1)
```

```
import holoviews as hv
import hvplot.pandas
import itertools
import matplotlib.pyplot as plt
import numpy as np
import nltk
import pandas as pd
import re
import sklearn.model_selection
import spacy
import string
import seaborn as sns
import tensorflow as tf
import io
```

```
nltk.download('omw-1.4')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
from google.colab import files
from imblearn.pipeline import Pipeline as ImbPipeline
from imblearn.over_sampling import RandomOverSampler, SMOTE
from keras import datasets, layers, models
from nltk import tokenize, MWETokenizer, TreebankWordTokenizer
from scipy import stats
from sklearn.base import TransformerMixin
from sklearn.ensemble import RandomForestClassifier
from sklearn.experimental import enable_halving_search_cv
from sklearn.feature_extraction.text import TfidfTransformer, TfidfVectorizer
from sklearn.impute import SimpleImputer
from sklearn.linear_model import SGDClassifier, LogisticRegression
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, f1_score, accuracy_score, roc_curve, auc, roc_auc_score,
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV, HalvingGridSearchCV, train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import NearestNeighbors
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier
from pyforest import *
from tensorflow import keras
from time import time as tt
```

Load Dataset

```
url = 'https://drive.google.com/file/d/10zV9L_LogAHwlaX0okgInnqAmW6QRQNn/view?usp=sharing'
data_file = 'https://drive.google.com/uc?export=download&id='+url.split('/')[-2]
data = pd.read_csv(data_file,encoding='latin-1')
data.head()
```

	text	label	query_index
0	Can I automatically top-up when traveling?	top_up_queries_or_issues	526cd7f17526
1	What kind of fiat currency can I used for hold...	other	f3cf7343067e
2	I did not get the item I ordered. How should ...	other	9a19501c3a3c
3	Freeze my account it's been hacked.	needs_troubleshooting	d76b07db8cf8
4	is there a reason that my payment didnt go thr...	other	bd95ba09a18d

```
data.describe()
```

	text	label	query_index
count	14195	13674	14195
unique	13084	8	13672
top	#	other	fc9b781a6b97
freq	68	5036	2

Data pre processing

```
data.isna().sum()
```

```
text            0
label         521
query_index     0
dtype: int64
```

```
data.count()
```

```
text           14195
label          13674
query_index    14195
dtype: int64
```

```
for i in list(data.columns):
  print(data[i].value_counts())
```

```
#
I topped up but the app did not accept it.
How do I know if my top up was unsuccessful?
Oh my goodness, my card has been declined twice at ATM! I tried two different ATM, but each one declined my card! Can you
I don't understand where this debit came from and want it removed.

My statement shows different transaction times.
What are the steps I need to take to cancel a transaction?
after i got married i need to change my name
I still have not received an answer as to why I was charged $1.00 in a transaction?
Am I able to track the card that was just sent to me?
Name: text, Length: 13084, dtype: int64
other                        5036
needs_troubleshooting        4305
card_queries_or_issues       2598
top_up_queries_or_issues     1684
Other                          21
Card_queries_or_issues         12
Top_up_queries_or_issues       11
Needs_troubleshooting           7
Name: label, dtype: int64
fc9b781a6b97    2
ea614b5c8b9a    2
0c50afa79ab7    2
a6e840dd13ce    2
5fc6b0b669aa    2
               ..
a0b6c96420d2    1
1bb97b06d70c    1
1ec16e8d1edd    1
42d144e1974c    1
f7e5a9b88449    1
Name: query_index, Length: 13672, dtype: int64
```

Data quality assessment and Exploratory Data Analysis for cleaning data

Drop rows that have # symbol in text column

```
print(data[data["text"] == "#"])
```

```
      text                     label   query_index
106      #     needs_troubleshooting  226t0c5be7cf
139      #                     other  10et93272e2e
266      #                     other  849t6f85a049
346      #                     other  73ft45a21d0f
530      #    card_queries_or_issues  ef6teff01216
...    ...                       ...           ...
13505    #  top_up_queries_or_issues  33dte465441a
13531    #     needs_troubleshooting  451t8fa10c0d
13799    #                     other  b25teb82e4a2
13825    #  top_up_queries_or_issues  763tcd9275b1
13884    #                     other  9eetccf1515a

[68 rows x 3 columns]
```

```
data = data.drop(data[data.text == '#'].index)
```

Start all sentences with capital letter in text column

```
data['text'] = data['text'].apply(lambda x: x.capitalize())
```

```
print(data)
```

```
                                                   text  \
0              Can i automatically top-up when traveling?
1      What kind of fiat currency can i used for hold...
2      I did not get the item i ordered.  how should ...
3                     Freeze my account it's been hacked.
4      Is there a reason that my payment didnt go thr...
...                                                  ...
14190  Can you tell me what the disposable cards are ...
14191  The atm won't give me my card back. i need it ...
```

```
14192  Can you please tell me why my card payments ar...
14193  The rate for a currency exchange was wrong whe...
14194  Am i able to track the card that was just sent...

                          label   query_index
0      top_up_queries_or_issues  526cd7f17526
1                         other  f3cf7343067e
2                         other  9a19501c3a3c
3          needs_troubleshooting  d76b07db8cf8
4                         other  bd95ba09a18d
...                         ...           ...
14190    card_queries_or_issues  bd6df98cc746
14191    card_queries_or_issues  e6197a1334b3
14192    card_queries_or_issues  b922a2a5f687
14193     needs_troubleshooting  cb1ed2c3ca95
14194    card_queries_or_issues  f7e5a9b88449

[14127 rows x 3 columns]
```

Fill empty values of label column with the most frequent value, in our case is other

```
print(data["label"].isnull().sum())
```

```
521
```

```
'''# handling missing data
from sklearn.impute import SimpleImputer
# 1. Imputer
imptr_empl = SimpleImputer(missing_values = np.nan, strategy = 'most_frequent')

# 2. Fit the imputer object to the feature matrix
imptr_empl = imptr_empl.fit(data[['label']])

# 3. Call Transform to replace missing data in train_dataset (on specific columns) by the most frequent of the column to which
data[['label']] = imptr_empl.transform(data[['label']]) '''
```

```
'# handling missing data\nfrom sklearn.impute import SimpleImputer \n# 1. Imputer\nimptr_empl = SimpleImputer(missing_va
t_frequent')  \n\n# 2. Fit the imputer object to the feature matrix\nimptr_empl = imptr_empl.fit(data[['label']])\n\n# 3.
ssing data in train_dataset (on specific columns) by the most frequent of the column to which that missing data belongs t
pl.transform(data[['label']]) '
```

Removing duplicates

```
l =list(data.columns)
for i in l:
  print(data[i].value_counts())
```

```
I would like to exchange currencies
How do i know if my top up was unsuccessful?
Oh my goodness, my card has been declined twice at atm! i tried two different atm, but each one declined my card! can you
I don't understand where this debit came from and want it removed.
What do i do if the atm took my card?

My statement shows different transaction times.
What are the steps i need to take to cancel a transaction?
After i got married i need to change my name
I still have not received an answer as to why i was charged $1.00 in a transaction?
Am i able to track the card that was just sent to me?
Name: text, Length: 13083, dtype: int64
other                        5004
needs_troubleshooting        4293
card_queries_or_issues       2586
top_up_queries_or_issues     1672
Other                          21
Card_queries_or_issues         12
Top_up_queries_or_issues       11
Needs_troubleshooting           7
Name: label, dtype: int64
bda7ab74290d    2
dcd5dac08c5f    2
d921fa76a483    2
a862cbe02605    2
529f3f67f869    2
               ..
19fdb4300302    1
40b009669cb4    1
ba221ecd6e40    1
ec1b8fbbcc65    1
f7e5a9b88449    1
Name: query_index, Length: 13604, dtype: int64
```

Check if we have removed duplicates by renaming it

```
data.loc[data.label == 'Other','label'] = 'other'
data.loc[data.label == 'Needs_troubleshooting','label'] = 'needs_troubleshooting'
data.loc[data.label == 'Card_queries_or_issues','label'] = 'card_queries_or_issues'
data.loc[data.label == 'Top_up_queries_or_issues','label'] = 'top_up_queries_or_issues'
```

```
data=data.dropna()
data = data.drop_duplicates()
```

```
data['label'].value_counts()
```

```
other                       4803
needs_troubleshooting       4153
card_queries_or_issues      2495
top_up_queries_or_issues    1632
Name: label, dtype: int64
```

```
data.count()
```

```
text           13083
label          13083
query_index    13083
dtype: int64
```

ENCODING CATEGORICAL DATA

```
# First check: what are the target categories?
data.value_counts()
```

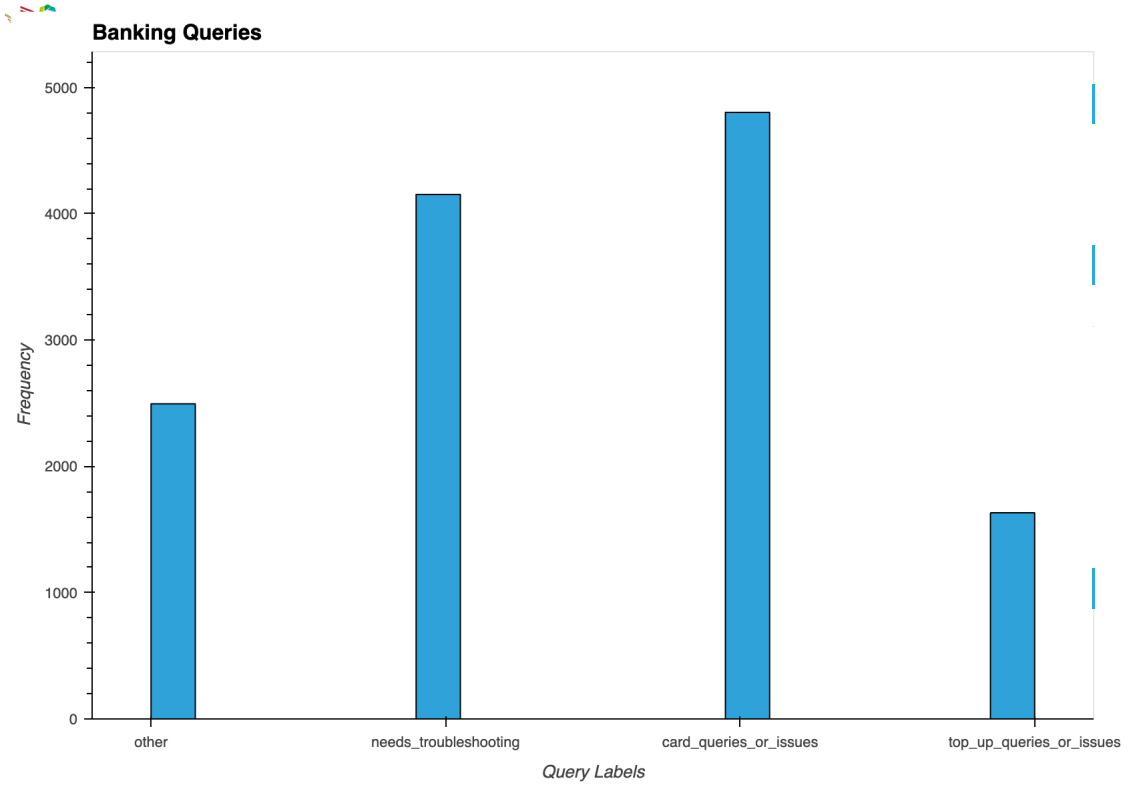
```
text
label                     query_index
\n\nwhat businesses accept this card?
card_queries_or_issues    532693cb5d73    1
My withdrawal is pending, why?
other                     8b98db331107    1
My wallet doesn't show my recent top up.
top_up_queries_or_issues  e78731b51985    1
My wallet got stolen a couple hours ago and now i've seen there already is a withdrawal. help this is absolutely urgent
i don't want to loose more money  needs_troubleshooting     9c262a198983    1
My wallet is empty even though i topped it up an hour ago.
top_up_queries_or_issues  aa2c910b42e1    1

..
I don't think the charges made when i had currency exchanged are right.
needs_troubleshooting     11ce0fdc214a    1
I don't think the exchange rate was right.
needs_troubleshooting     73b70bf28dfa    1
I don't think the transaction has gone through, so can i cancel a transfer?
other                     60a77bb2ef8d    1
I don't understand how to top up my account, can you please explain the process?
other                     6a0ad9821af4    1
Ýýý1 was in my statement as an extra fee.
needs_troubleshooting     b03e2605e919    1
Length: 13083, dtype: int64
```

```
temp = data.copy() #we make a copy and use a temporary variable name, since this is not the final transformation

# encode categorical data for the 'Label' column

# create an object of the LabelEncoder class
lblEncoder_X = LabelEncoder()
# apply LblEncoder object to our categorical variables (columns - 'Label') using the fit_transform method. This returns the co

temp['label'] = lblEncoder_X.fit_transform(temp['label']) # we can fit and transform all at once

temp.head()
```

	text	label	query_index
0	Can i automatically top-up when traveling?	3	526cd7f17526
1	What kind of fiat currency can i used for hold...	2	f3cf7343067e
2	I did not get the item i ordered. how should ...	2	9a19501c3a3c
3	Freeze my account it's been hacked.	1	d76b07db8cf8
4	Is there a reason that my payment didnt go thr...	2	bd95ba09a18d

```
labl = ['other', 'needs_troubleshooting', 'card_queries_or_issues','top_up_queries_or_issues']
plot = temp['label'].hvplot.hist(
```

```
plot = temp['label'].hvplot.hist(
    frame_height=500, frame_width=750,
    xlabel='Query Labels', ylabel='Frequency',
    title='Banking Queries',legend='bottom', xticks = [(0, labl[0]), (1, labl[1]), (2, labl[2]), (3, labl[3])],
    alpha=1, muted_alpha=0, muted_fill_alpha=0, muted_line_alpha=0.5,
    tools=['pan', 'box_zoom', 'wheel_zoom', 'undo', 'redo', 'hover', 'save', 'reset']
)
hv.extension('bokeh')
plot
```



TEXT PRE-PROCESSING

```
!pip install ml-datasets
!pip install --upgrade spacy
!python -m spacy download en_core_web_sm
!python -m spacy download en_core_web_md
import sklearn.model_selection
import spacy
import seaborn as sns
nlp = spacy.load('en_core_web_sm')
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting ml-datasets
  Downloading ml_datasets-0.2.0-py3-none-any.whl (15 kB)
Requirement already satisfied: catalogue<3.0.0,>=0.2.0 in /usr/local/lib/python3.9/dist-packages (from ml-datasets) (2.0.
Requirement already satisfied: numpy>=1.7.0 in /usr/local/lib/python3.9/dist-packages (from ml-datasets) (1.22.4)
Requirement already satisfied: tqdm<5.0.0,>=4.10.0 in /usr/local/lib/python3.9/dist-packages (from ml-datasets) (4.65.0)
Requirement already satisfied: srsly<3.0.0,>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from ml-datasets) (2.4.6)
Installing collected packages: ml-datasets
Successfully installed ml-datasets-0.2.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: spacy in /usr/local/lib/python3.9/dist-packages (3.5.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (23.0)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.9/dist-packages (from spacy) (3.0.8)
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-packages (from spacy) (67.6.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (2.27.1)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (3.3.0)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.9/dist-packages (from spacy) (2.4.6)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 in /usr/local/lib/python3.9/dist-packages (from spac
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (0.10.1)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (1.0.4)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.9/dist-packages (from spacy) (1.1.1)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.9/dist-packages (from spacy) (8.1.9)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.9/dist-packages (from spacy) (3.1.2)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (4.65.0)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (1.0.9)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.9/dist-packages (from spacy) (2.0.7)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.9/dist-packages (from spacy) (2.0.8)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (0.7.0)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.9/dist-packages (from spacy) (6.3.0)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.9/dist-packages (from spacy) (3.0.12
```

```
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (1.22.4)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.9/dist-packages (from pydantic!=1.8,!=1
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.13.0->spac
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.13.0
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.1
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.9/dist-packages (from thinc<8.2.0,>=8.1.8->sp
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.9/dist-packages (from thinc<8.2.0,>=8.1
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.9/dist-packages (from typer<0.8.0,>=0.3.0->s
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from jinja2->spacy) (2.1.2)
2023-04-12 10:24:20.890012: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find Tensor
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting en-core-web-sm==3.5.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.5.0/en_core_web_sm-3.5.0-py3-n
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 64.9 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.6.0,>=3.5.0 in /usr/local/lib/python3.9/dist-packages (from en-core-web-sm==3.5.0)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->e
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->en-core
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 in /usr/local/lib/python3.9/dist-packages (from spac
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->e
Requirement already satisfied: jinja2 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->en-core-web-sm
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.
Requirement already satisfied: typer<0.8.0,>=0.3.0 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->e
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->en-co
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->en-core-we
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.
```

REMOVING STOP WORDS, PUNCTUATIONS and Lemmatisation

```
import string


#removing punctuations
def rem_pun(txt):
  txt_nopunt = "".join([c for c in txt if c not in string.punctuation])
  return txt_nopunt
data['no_punc'] = data['text'].apply(lambda x: rem_pun(x))


#tokenisation
def tokenize(column):
    tokenizer = TreebankWordTokenizer()
    result = tokenizer.tokenize(column)
    return result
data['tokens'] = data.apply(lambda x: tokenize(x['no_punc'].lower()), axis=1)



#removing stopwords
stopwords = nltk.corpus.stopwords.words('english')
def remove_stop_words(tokens):
  txt_clean = [word for word in tokens if word not in stopwords]
  return txt_clean
data['clean_text']=data['tokens'].apply(lambda x: remove_stop_words(x))
data['clean_text']=data['clean_text'].apply(lambda x: ' '.join([str(i) for i in x]))
data['clean_text'].head()
```

```
0                         automatically topup traveling
1              kind fiat currency used holding exchange
2    get item ordered go cancel order payment pleas...
3                                 freeze account hacked
4                               reason payment didnt go
Name: clean_text, dtype: object
```

```
nltk.download('wordnet')
```

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

```
#Lemmatization
wn = nltk.WordNetLemmatizer()
ps = nltk.PorterStemmer()
def lemmatization(tokens):
  text = [wn.lemmatize(word) for word in tokens]
  return text
data['lemmatised'] = data['clean_text'].apply(lambda x : lemmatization(x))
data['lemmatised'] = data['lemmatised'].apply(lambda x: ''.join(map(str, x)))
data['lemmatised'].head()
```

```
0                       automatically topup traveling
1             kind fiat currency used holding exchange
2    get item ordered go cancel order payment pleas...
3                              freeze account hacked
4                                  reason payment didnt go
```

```
#data after preprocessing
data.head()
```

	text	label	query_index	no_punc	tokens	clea
0	Can i automatically top-up when traveling?	top_up_queries_or_issues	526cd7f17526	Can i automatically topup when traveling	[can, i, automatically, topup, when, traveling]	automatically topup t
1	What kind of fiat currency can i used for hold...	other	f3cf7343067e	What kind of fiat currency can i used for hold...	[what, kind, of, fiat, currency, can, i, used,...	kind fiat curren holding ex
2	I did not get the item i ordered. how should ...	other	9a19501c3a3c	I did not get the item i ordered how should i...	[i, did, not, get, the, item, i, ordered, how,...	get item ordered go order payment
3	Freeze my account it's been hacked.	needs_troubleshooting	d76b07db8cf8	Freeze my account its been hacked	[freeze, my, account, its, been, hacked]	freeze account
4	Is there a reason that my payment didnt go thr...	other	bd95ba09a18d	Is there a reason that my payment didnt go thr...	[is, there, a, reason, that, my, payment, didn...	reason payment

Final data with clean text and labels

```
data_final = pd.DataFrame(columns=['text','clean','label'])
data_final['text'] = data['text']
data_final['clean'] = data['clean_text']
data_final['label'] = data['label']
data_final.head()
```

	text	clean	label
0	Can i automatically top-up when traveling?	automatically topup traveling	top_up_queries_or_issues
1	What kind of fiat currency can i used for hold...	kind fiat currency used holding exchange	other
2	I did not get the item i ordered. how should ...	get item ordered go cancel order payment pleas...	other
3	Freeze my account it's been hacked.	freeze account hacked	needs_troubleshooting
4	Is there a reason that my payment didnt go thr...	reason payment didnt go	other

```
lblEncoder_X = LabelEncoder()
# apply LblEncoder object to our categorical variables (columns - 'Label') using the fit_transform method. This returns the co

data_final['label'] = lblEncoder_X.fit_transform(data_final['label'])
data_final.head()
```

	text	clean	label
0	Can i automatically top-up when traveling?	automatically topup traveling	3
1	What kind of fiat currency can i used for hold...	kind fiat currency used holding exchange	2
2	I did not get the item i ordered. how should ...	get item ordered go cancel order payment pleas...	2
3	Freeze my account it's been hacked.	freeze account hacked	1
4	Is there a reason that my payment didnt go thr...	reason payment didnt go	2

Dataset split and Feature Scaling

```
X_train, X_test, y_train, y_test = train_test_split(data_final["clean"],data_final["label"],test_size=0.35,shuffle=True)
tfidf_vectorizer = TfidfVectorizer(use_idf=True)

X_train_vectors_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_vectors_tfidf = tfidf_vectorizer.transform(X_test)
```

Classification Task

```
#FITTING THE CLASSIFICATION MODEL using Naive Bayes(tf-idf)
nb_tfidf = MultinomialNB()
```

```
18
#nb_tfidf.fit(X_train_vectors_tfidf, y_train)
nb_tfidf.fit(X_train_vectors_tfidf, y_train)
#Predict y value for test dataset
y_predict = nb_tfidf.predict(X_test_vectors_tfidf)
y_prob = nb_tfidf.predict_proba(X_test_vectors_tfidf)[:,1]
print('Accuracy: {:.2f}'.format(accuracy_score(y_test, y_predict)))
print(classification_report(y_test,y_predict))
print('Confusion Matrix:\n',confusion_matrix(y_test, y_predict))
```

```
Accuracy: 0.82
              precision    recall  f1-score   support

           0       0.90      0.75      0.82       887
           1       0.75      0.89      0.82      1412
           2       0.81      0.82      0.82      1712
           3       0.98      0.74      0.84       569

    accuracy                           0.82      4580
   macro avg       0.86      0.80      0.82      4580
weighted avg       0.83      0.82      0.82      4580

Confusion Matrix:
 [[ 668  115   97    7]
 [  30 1261  120    1]
 [  44  255 1411    2]
 [   4   40  105  420]]
```

```
#Classifying the data Using Random Forest Classifier model
classifier = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=100)
classifier.fit(X_train_vectors_tfidf, y_train)
# Predicting the Test set results
y_pred = classifier.predict(X_test_vectors_tfidf)
y_prob = classifier.predict_proba(X_test_vectors_tfidf)[:,1]
print('Accuracy: {:.2f}'.format(accuracy_score(y_test, y_pred)))
print(classification_report(y_test,y_pred))
print('Confusion Matrix: \n',confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.53
              precision    recall  f1-score   support

           0       0.93      0.09      0.17       887
           1       0.85      0.51      0.64      1412
           2       0.45      0.96      0.61      1712
           3       0.00      0.00      0.00       569

    accuracy                           0.53      4580
   macro avg       0.56      0.39      0.35      4580
weighted avg       0.61      0.53      0.46      4580

Confusion Matrix:
 [[  82   17  788    0]
 [   6  720  686    0]
 [   0   73 1639    0]
 [   0   36  533    0]]
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-s
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-s
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-s
  _warn_prf(average, modifier, msg_start, len(result))
```

```
#Classifying the data Using SGD Classifier model (Optional)
clf1 = SGDClassifier(loss="log_loss", penalty="l2")
clf1.fit(X_train_vectors_tfidf, y_train)

y_pre = clf1.predict(X_test_vectors_tfidf)

print('Accuracy: {:.2f}'.format(accuracy_score(y_test, y_pre)))
print(classification_report(y_test,y_pre))
print('Confusion Matrix: \n',confusion_matrix(y_test, y_pre))
```

```
Accuracy: 0.87
              precision    recall  f1-score   support

           0       0.88      0.88      0.88       887
           1       0.84      0.88      0.85      1412
           2       0.87      0.85      0.86      1712
           3       0.95      0.92      0.93       569

    accuracy                           0.87      4580
   macro avg       0.88      0.88      0.88      4580
weighted avg       0.87      0.87      0.87      4580
```

```
Confusion Matrix:
 [[ 781   42   48   16]
 [  37 1236  139    0]
 [  65  186 1449   12]
 [   9   16   23  521]]
```

Logistic Regression

```
# create custom class to add to the pipeline
from sklearn.base import TransformerMixin
class DenseTransformer(TransformerMixin):
    ''' Pipeline step to transform a sparse matrix into a dense one '''
    def fit(self, X, y=None, **fit_params):
        return self

    def transform(self, X, y=None, **fit_params):
        return X.toarray()


#FITTING THE CLASSIFICATION MODEL using Logistic Regression(tf-idf)
lr_tfidf=LogisticRegression(solver = 'liblinear', C=10, penalty = 'l2')
lr_tfidf.fit(X_train_vectors_tfidf, y_train)
#Predict y value for test dataset
y_predict = lr_tfidf.predict(X_test_vectors_tfidf)
y_prob = lr_tfidf.predict_proba(X_test_vectors_tfidf)[:,1]
print(classification_report(y_test,y_predict))
print('Confusion Matrix:',confusion_matrix(y_test, y_predict))

# fpr, tpr, thresholds = roc_curve(y_test, y_prob)
# roc_auc = auc(fpr, tpr)
# print('AUC:', roc_auc)
```

```
                   precision    recall  f1-score   support

                0       0.89      0.89      0.89       887
                1       0.85      0.89      0.87      1412
                2       0.88      0.86      0.87      1712
                3       0.94      0.93      0.94       569

         accuracy                           0.88      4580
        macro avg       0.89      0.89      0.89      4580
     weighted avg       0.88      0.88      0.88      4580

     Confusion Matrix: [[ 788   37   47   15]
      [  27 1252  130    3]
      [  58  168 1471   15]
      [   8   15   16  530]]
```

```
#Logistic Regression
lr_tfidf=LogisticRegression(solver = 'liblinear', C=10, penalty = 'l2')
lr_tfidf.fit(X_train_vectors_tfidf, y_train)
#Predict y value for test dataset
y_predict = lr_tfidf.predict(X_test_vectors_tfidf)
y_prob = lr_tfidf.predict_proba(X_test_vectors_tfidf)[:,1]
print(classification_report(y_test,y_predict))
print('Confusion Matrix:',confusion_matrix(y_test, y_predict))
```

```
                   precision    recall  f1-score   support

                0       0.89      0.89      0.89       887
                1       0.85      0.89      0.87      1412
                2       0.88      0.86      0.87      1712
                3       0.94      0.93      0.94       569

         accuracy                           0.88      4580
        macro avg       0.89      0.89      0.89      4580
     weighted avg       0.88      0.88      0.88      4580

     Confusion Matrix: [[ 788   37   47   15]
      [  27 1252  130    3]
      [  58  168 1471   15]
      [   8   15   16  530]]
```

Neural Network Model

Pre-process data

```
# create custom class to add to the pipeline
from sklearn.base import TransformerMixin
```

```
class DenseTransformer(TransformerMixin):
    ''' Pipeline step to transform a sparse matrix into a dense one '''
    def fit(self, X, y=None, **fit_params):
        return self

    def transform(self, X, y=None, **fit_params):
        return X.toarray()



#create pipeline
preprocessor = Pipeline(
        steps =[('tfidf', TfidfVectorizer(stop_words='english', lowercase= True,
                                          max_features = 3000,
                                          ngram_range=(1,2))),
                ('to_dense', DenseTransformer()),
      ]
    )

preprocessor.fit(data_final['clean'])
X_train_preprocessed = preprocessor.transform(X_train)
X_test_preprocessed = preprocessor.transform(X_test)




y_train
```

```
994     1
2243    1
9053    3
3711    1
7571    0
       ..
1613    0
6067    0
7613    0
5223    2
1067    1
Name: label, Length: 8503, dtype: int64
```

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models

# create a simple model with ONE hidden layer only
model = models.Sequential()
# we create a hidden layer with 20 nodes.
hidden_layer_nodes = 50
num_of_input_features = X_train_preprocessed.shape[1] #number of features = number of columns in the input matrix


model.add(layers.Input(shape=(num_of_input_features,)))
model.add(layers.Dense(hidden_layer_nodes, activation='relu'))
# let's add a dropout layer
dropout_rate = 0.2
model.add(layers.Dropout(rate= dropout_rate))
num_categories = len(y_train.value_counts())
model.add(layers.Dense(num_categories, activation='softmax')) #is it clear why here we use "sigmoid" and use "softmax" for mul


learning_rate = 0.01
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rate),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(), # we use this function for MULTI-CLASS PROBLEMS. It expect
              metrics=['accuracy'])

# let's print a summary of the model to see what it's like
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 50)                150050

 dropout (Dropout)           (None, 50)                0

 dense_1 (Dense)             (None, 4)                 204

=================================================================
Total params: 150,254
Trainable params: 150,254
Non-trainable params: 0
_________________________________________________________________
None
```

```
y_train.dtype
```

```
dtype('int64')
```

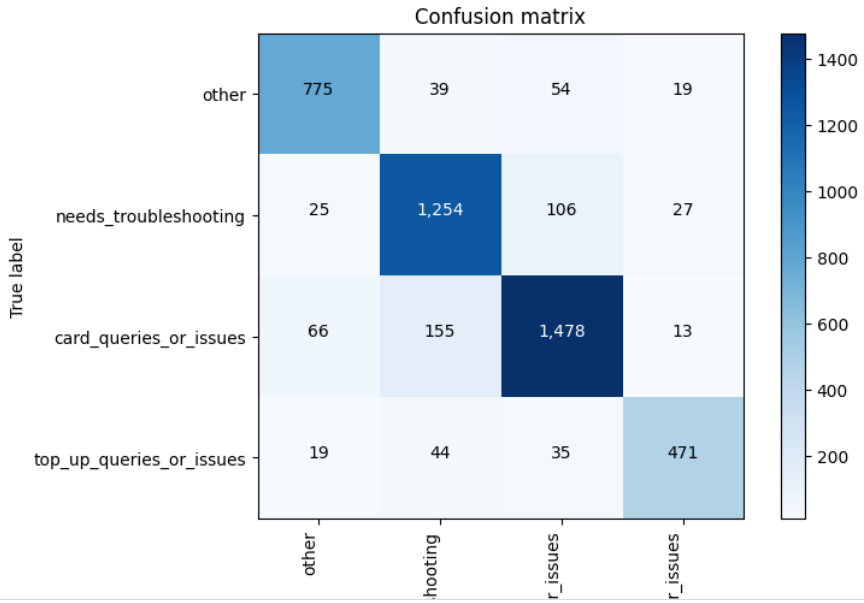
```
batch_size = 130 # The batch size indicates how many data points we use to compute each update to the parameters of the models
epochs = 50 #how long we train the model for
validation_split = 0.2  #Fraction of the training data to be used as validation data.

history = model.fit(X_train_preprocessed,
                    y_train.to_numpy(), #we add .to_numpy() because Keras doesn't like dataframes so we need to transform the
                    epochs=epochs,
                    batch_size=batch_size,
                    validation_split= validation_split)
```

```
Epoch 1/50
53/53 [==============================] - 1s 11ms/step - loss: 0.7526 - accuracy: 0.7146 - val_loss: 0.3634 - val_accuracy
Epoch 2/50
53/53 [==============================] - 0s 6ms/step - loss: 0.2619 - accuracy: 0.9056 - val_loss: 0.3285 - val_accuracy:
Epoch 3/50
53/53 [==============================] - 0s 5ms/step - loss: 0.1704 - accuracy: 0.9394 - val_loss: 0.3320 - val_accuracy:
Epoch 4/50
53/53 [==============================] - 0s 6ms/step - loss: 0.1302 - accuracy: 0.9541 - val_loss: 0.3502 - val_accuracy:
Epoch 5/50
53/53 [==============================] - 0s 5ms/step - loss: 0.1118 - accuracy: 0.9569 - val_loss: 0.3722 - val_accuracy:
Epoch 6/50
53/53 [==============================] - 1s 11ms/step - loss: 0.1004 - accuracy: 0.9643 - val_loss: 0.3861 - val_accuracy
Epoch 7/50
53/53 [==============================] - 1s 11ms/step - loss: 0.0867 - accuracy: 0.9685 - val_loss: 0.4050 - val_accuracy
Epoch 8/50
53/53 [==============================] - 1s 12ms/step - loss: 0.0807 - accuracy: 0.9702 - val_loss: 0.4244 - val_accuracy
Epoch 9/50
53/53 [==============================] - 1s 16ms/step - loss: 0.0722 - accuracy: 0.9728 - val_loss: 0.4346 - val_accuracy
Epoch 10/50
53/53 [==============================] - 1s 12ms/step - loss: 0.0669 - accuracy: 0.9737 - val_loss: 0.4599 - val_accuracy
Epoch 11/50
53/53 [==============================] - 0s 9ms/step - loss: 0.0632 - accuracy: 0.9774 - val_loss: 0.4729 - val_accuracy:
Epoch 12/50
53/53 [==============================] - 1s 12ms/step - loss: 0.0583 - accuracy: 0.9815 - val_loss: 0.4819 - val_accuracy
Epoch 13/50
53/53 [==============================] - 1s 11ms/step - loss: 0.0555 - accuracy: 0.9800 - val_loss: 0.5011 - val_accuracy
Epoch 14/50
53/53 [==============================] - 1s 12ms/step - loss: 0.0552 - accuracy: 0.9790 - val_loss: 0.5133 - val_accuracy
Epoch 15/50
53/53 [==============================] - 1s 15ms/step - loss: 0.0526 - accuracy: 0.9810 - val_loss: 0.5150 - val_accuracy
Epoch 16/50
53/53 [==============================] - 1s 14ms/step - loss: 0.0522 - accuracy: 0.9791 - val_loss: 0.5289 - val_accuracy
Epoch 17/50
53/53 [==============================] - 1s 12ms/step - loss: 0.0483 - accuracy: 0.9812 - val_loss: 0.5440 - val_accuracy
Epoch 18/50
53/53 [==============================] - 1s 13ms/step - loss: 0.0493 - accuracy: 0.9819 - val_loss: 0.5496 - val_accuracy
Epoch 19/50
53/53 [==============================] - 1s 12ms/step - loss: 0.0479 - accuracy: 0.9816 - val_loss: 0.5484 - val_accuracy
Epoch 20/50
53/53 [==============================] - 1s 13ms/step - loss: 0.0476 - accuracy: 0.9824 - val_loss: 0.5841 - val_accuracy
Epoch 21/50
53/53 [==============================] - 1s 15ms/step - loss: 0.0457 - accuracy: 0.9825 - val_loss: 0.5789 - val_accuracy
Epoch 22/50
53/53 [==============================] - 1s 11ms/step - loss: 0.0451 - accuracy: 0.9834 - val_loss: 0.5763 - val_accuracy
Epoch 23/50
53/53 [==============================] - 1s 10ms/step - loss: 0.0437 - accuracy: 0.9840 - val_loss: 0.5897 - val_accuracy
Epoch 24/50
53/53 [==============================] - 1s 14ms/step - loss: 0.0432 - accuracy: 0.9837 - val_loss: 0.5945 - val_accuracy
Epoch 25/50
53/53 [==============================] - 1s 11ms/step - loss: 0.0439 - accuracy: 0.9832 - val_loss: 0.6032 - val_accuracy
Epoch 26/50
53/53 [==============================] - 1s 16ms/step - loss: 0.0417 - accuracy: 0.9827 - val_loss: 0.6088 - val_accuracy
Epoch 27/50
53/53 [==============================] - 1s 17ms/step - loss: 0.0408 - accuracy: 0.9835 - val_loss: 0.6291 - val_accuracy
Epoch 28/50
53/53 [==============================] - 1s 16ms/step - loss: 0.0416 - accuracy: 0.9838 - val_loss: 0.6040 - val_accuracy
Epoch 29/50
53/53 [==============================] - 1s 18ms/step - loss: 0.0396 - accuracy: 0.9849 - val_loss: 0.6162 - val_accuracy
```

```
#Computing the Metrics of model by plotting confusion Matrix
def plot_confusion_matrix(cm,target_names,title='Confusion matrix',cmap=None,normalize=True):

    accuracy = np.trace(cm) / np.sum(cm).astype('float')
    misclass = 1 - accuracy

    if cmap is None:
        cmap = plt.get_cmap('Blues')

    plt.figure(figsize=(8, 6))
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
```

```
    if target_names is not None:
        tick_marks = np.arange(len(target_names))
        plt.xticks(tick_marks, target_names, rotation=45)
        plt.yticks(tick_marks, target_names)

    if normalize:
        cm = cm.astype('int')


    thresh = cm.max() / 1.5 if normalize else cm.max() / 2
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        if normalize:
            plt.text(j, i, "{:,}".format(cm[i, j]),
                     horizontalalignment="center",
                     color="white" if cm[i, j] > thresh else "black")
        else:
            plt.text(j, i, "{:,}".format(cm[i, j]),
                     horizontalalignment="center",
                     color="white" if cm[i, j] > thresh else "black")


    plt.tight_layout()
    plt.xticks(rotation=90, ha='right')
    plt.ylabel('True label')
    plt.xlabel('Predicted label\n\naccuracy={:0.4f}; misclass={:0.4f}'.format(accuracy, misclass))
    plt.show()


# Check performance on test data
test_probabilities = model.predict(X_test_preprocessed)
# Since it's a multiclass problem, the output probabilities are given as one probability PER CLASS
# To get the final predicted label, we need to find the category with the HIGHEST probability
# We can get this by using the function np.argmax()
test_predictions = np.argmax(test_probabilities, axis=1)
# the result should be one integer number per data point that we can compare with the target labels

# let's show the classification report with all the metrics
# think about which metrics you think are the most important ones for this problem!
print(classification_report(y_test.to_numpy(),test_predictions,
                            target_names= labl)) # this is to give the real categories, not their encoded numbers

# let's also print the balanced accuracy score, since we know the dataset is not balanced
print(f'The balanced accuracy score is {balanced_accuracy_score(y_test.to_numpy(),test_predictions):.3f}\n')

# let's get all the numbers for the confusion matrix
cm = confusion_matrix(y_test.to_numpy(),test_predictions)

plot_confusion_matrix(cm=cm,target_names= labl)
```

```
144/144 [==============================] - 1s 3ms/step
                         precision    recall  f1-score   support

                  other       0.88      0.87      0.87       887
   needs_troubleshooting       0.84      0.89      0.86      1412
   card_queries_or_issues       0.88      0.86      0.87      1712
top_up_queries_or_issues       0.89      0.83      0.86       569

               accuracy                           0.87      4580
              macro avg       0.87      0.86      0.87      4580
           weighted avg       0.87      0.87      0.87      4580

The balanced accuracy score is 0.863
```



✓ 2s completed at 11:25