Consumer Complaint Routing

Ritesh KC

February 28, 2026

[1]:

```
# Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid")
sns.set_palette("husl")
from wordcloud import WordCloud
```

[2]:

```
df = pd.read_csv("input_data/complaints.csv.zip", low_memory=False)
```

[3]:

```
print(df.shape)
df.head(3)
```

```
(13843560, 18)
```

[3]:

```
  Date received                                            Product  \
0    2020-07-13                                           Mortgage
1    2023-12-16  Credit reporting or other personal consumer re…
2    2024-01-27  Credit reporting or other personal consumer re…

        Sub-product                                 Issue  \
0      FHA mortgage        Trouble during payment process
1  Credit reporting  Incorrect information on your report
2  Credit reporting  Incorrect information on your report

                             Sub-issue  \
0                                  NaN
1  Information belongs to someone else
2        Account information incorrect

                        Consumer complaint narrative  \
0  I currently have my mortgage with GMFS, servic…
1                                                NaN
2                                                NaN

                             Company public response  \
0                                                NaN
```

```
1                                                NaN
2  Company has responded to the consumer and the …

                                  Company State ZIP code Tags  \
0                   SAMC Honebee TRS, LLC    AL    36695  NaN
1                       FactorTrust, Inc.    NY    13224  NaN
2  TRANSUNION INTERMEDIATE HOLDINGS, INC.    NY    11714  NaN

  Consumer consent provided? Submitted via Date sent to company  \
0           Consent provided           Web           2020-07-13
1       Consent not provided           Web           2023-12-16
2       Consent not provided           Web           2024-01-27

      Company response to consumer Timely response? Consumer disputed?  \
0          Closed with explanation              Yes                NaN
1          Closed with explanation              Yes                NaN
2  Closed with non-monetary relief              Yes                NaN

   Complaint ID
0       3743391
1       8011171
2       8236394
```

[4]:
```
# For this I only care about consumer complaint narrative and product
cols = ['Consumer complaint narrative', 'Product']
df = df[cols].dropna()
print(df.shape)
(df.value_counts())
```

```
(3729318, 2)
```

[4]:
```
Consumer complaint narrative
Product
In accordance with the Fair Credit Reporting act. The List of accounts below has
violated my federally protected consumer rights to privacy and confidentiality
under 15 USC 1681.\n\n15 U.S.C 1681 section 602 A. States I have the right to
privacy.\n\n15 U.S.C 1681 Section 604 A Section 2 : It also states a consumer
reporting agency can not furnish a account without my written instructions 15
U.S.C 1681c. ( a ) ( 5 ) Section States : no consumer reporting agency may make
any consumer report containing any of the following items of information Any
other adverse item of information, other than records of convictions of crimes
which antedates the report by more than seven years.\n\n15 U.S.C. 1681s-2 ( A )
( 1 ) A person shall not furnish any information relating to a consumer to any
consumer reporting agency if the person knows or has reasonable cause to believe
that the information is inaccurate.
Credit reporting or other personal consumer reports
26439
My credit reports are inaccurate. These inaccuracies are causing creditors to
```

deny me credit. You have the duty to report accurate information about consumers. Please investigate these accounts and inquires and update these accounts accordingly to avoid future litigation.
Credit reporting or other personal consumer reports
22298
You have reported inaccurate and unauthorized accounts on my credit report, which is a violation of the Fair Credit Reporting Act (15 U.S. Code 1681i) requiring a proper reinvestigation of disputed items, and 1681e (b), which mandates maximum possible accuracy. These false entries are damaging and unjust, especially since Ive never opened or authorized these accounts. If you fail to investigate and correct this, I may pursue legal action under the FCRA and FDCPA (15 U.S. Code 1692e) for deceptive and misleading reporting.
Credit reporting or other personal consumer reports
18190
Upon reviewing my credit report, I have identified inaccurate accounts that need reporting and correction through your company.
Credit reporting or other personal consumer reports
14308
I am writing to have the following information removed from my credit file, the items that I need deleted are going to be attached in a word document. I am a victim of identity theft. I have multiple accounts and inquiries that I DID NOT apply for listed on my credit report. I ask that these items be deleted so my credit report will only show accurate information. I reported the theft of my identity to the Federal Trade Commission and have enclosed copies of the report. I would like these items deleted from my credit report as soon as possible. Please block this information from my credit report, pursuant to section 605B of the Fair Credit Reporting Act.
Credit reporting or other personal consumer reports
12197

…

I have had numerous creditors send me results XXXX XXXX explaining that my credit score is XXXX the dates are XX/XX/2022 from XXXX XXXX, XX/XX/2022 from XXXX XXXX, XX/XX/2022 from XXXX XXXX and XXXX XXXX, on XX/XX/2022 from XXXX XXXX ; I use 3 credit monitoring services, all 3 report the XXXX score at XXXX for the last 4 months (see attachment)
Credit reporting, credit repair services, or other personal consumer reports
1
I have had numerous conversations with numerous employees about a transaction that was incorrect. I have been trying to get this corrected since the end of XXXX to see how to correct this error. I was finally told that documentation needed to be sent by the bank that needed to say the funds were for me and should not be sent to the requesting bank. I had XXXX XXXX fax a statement stating these funds were mine, I had a XXXX of a US Bank XXXX follow up to make sure they had what was needed. She was told they did have everything they needed to address this situation. In following up again today I was told they did not receive the information even after the XXXX I have been working with was told everything was fine. I need help getting a truthful answer and this situation

corrected. I am extremely frustrated as I don't believe the fraud investigation team is honest or interested in putting this to rest.

Checking or savings account

1

I have had numerous conversations with XXXX, The first time i would try to move my phone number to XXXX XXXX was the beginning of our problem this was in XX/XX/year> I was unable to succeed due my phone was too old, So I contacted XXXX and ask them to reopen my service and after XXXX days i was back in business with XXXX and then I received a bill for {$210.00} so i called XXXX to ask why my phone bill was so high and learned that my plan had Grandfathered out and the new plan was the best plan XXXX had to offer me at that time. So I decided on that day to go to Spectrum and buy a new phone and have my Phone number follow me and in doing so i cancelled my service with XXXX XXXX on the next i mailed XXXX XXXX {$120.00} because after all I had purchased a new phone with XXXX and then the problem begin when I heard from XXXX again They sent me Two Letters one letter said they Owed me {$140.00} and the other letter said I Owed XXXX {$210.00}. When I call XXXX my code was not working and so they made me an appointment for XX/XX/year>XXXX XXXX XXXX XXXX at the XXXX XXXX and I was there and spoke with XXXX of there employees and thought the problem was resolved but it was not. I have had several conversation with XXXX and I have even sent them a response via XXXX XXXX XXXX. I do not owe XXXX {$140.00}.

Debt collection

1

I have had numerous conversations with XXXX regarding my account # XXXX XXXX XXXX XXXX - Which was an old XXXX account that was converted into XXXX through XXXX. <P/>They charged me an unexpected fee, failed to disclose it, and then marked me 60 days delinquent for XXXX 2017. I am in the middle of a mortgage transaction and this is now causing serious problems and I am in jeopardy of losing thousands of dollars. I have talked to XXXX and they have agreed to remove the derogatory reporting, yet as of XX/XX/2017 they updated the credit bureaus and re-confirmed the 60 day late. Now my reports are updated with a late payment still on their. It is a nightmare to navigate the XXXX phone system, and every customer service rep I speak to is unable to understand my problem, or offer a solution. I urge, and plead for your assistance in resolving this matter ASAP. I am on a deadline of XXXX and need written confirmation before then that this late payment has been deleted from my credit history once and for all.

Credit reporting, credit repair services, or other personal consumer reports

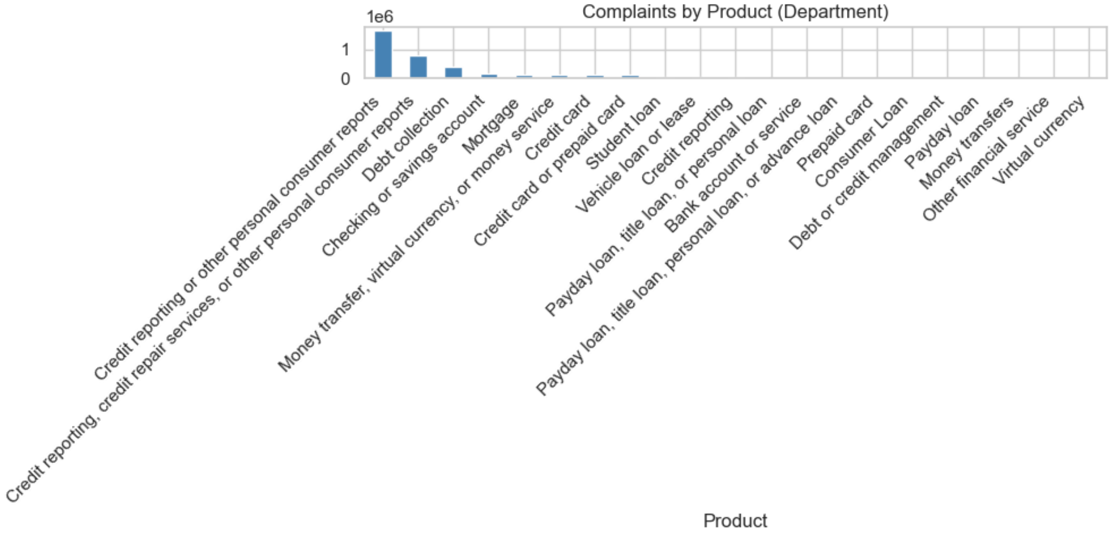
1

•\tSchool Loan servicer XXXX defaulted my loans while I was paying on them•\tXXXX never informed even though they continued to take my payments•\tXXXX sent (so they said) a default notice to an address I had not been at in over 5 years•\tXXXX sent correspondences within those 5 years to my current address but sends this very important notice to an old address.•\tI had already had a claim in with Attn Gen on them because they pulled a bait and switch on me with a forbearance request that automatically excluded me from loan payment reduction programs and did not disclose.•\tI believe they sold my loans because of the claim. The consumer affairs office said the claim needed to be resent with the new information to the original issue. I did that.•\tECMC was the new loan company and before I even knew they had my loans or what happened they tacked on XXXX in collection fees. I have been paying on these loans for almost 10 years, the interest was taken first and my original amount of XXXX was only down to XXXX principle. Now, after collection costs it is XXXX.•\tECMC reached out to me at work and said I needed to immediately pay them etc. and that is when I found out what happened. From XX/XX/XXXX when XXXX sold the loans to XX/XX/XXXX had no idea, was still just paying my loans.•\tI reached out to a lawyer - he asked me to have ECMC send all documentation via certified mail of everything they had done without my knowledge since the bought the loans.•\tXX/XX/XXXX I get the packet, except that they already sent a notice of wage garnishment out, to the same address XXXX used, however they never included a copy in that packet. My company never received notice either.•\tWhen my company found out by notice of penalty for non-compliance, they contacted me. I had no clue and neither did they. I called ECMC immediately and they had me send in a request for a hearing but said it was probably too late.•\tI sent the letter immediately but they still garnished my wages taking the max amount which over the next two months fighting them almost made me homeless.•\tI finally got the Ombudsman involved but they just gave me the run around. I have all documentation. They side-stepped the fact that they never included the garnishment notice and in the end stood behind the Dept of Ed code as a response.•\tI called the lawyer that was working with me but he dropped off the earth.•\tThe Consumer Affairs woman handling the XXXX case said that my case would be part of the class action. So I went back to the board restating my loan should have never been in default so they let me appeal the decision, but that was denied.•\tIn XX/XX/XXXX, after speaking to another Ombudsman, she advised taking the default and going through the rehab program as it would be the only way to stop the garnishment which had left me broke and unable to pay rent or care for my sick mother.•\tI set up the rehab as I was left no choice. I tried a few lawyers in the plan and none of them handle these types of cases.•\tI completed the rehab program XX/XX/XXXX. This was by contract supposed to stop future garnishment, resolve the default and restore my credit. As of today it sits unresolved and they added more collection fees (not in the contract) as well as three more loans that are not mine.•\tI immediately called dept of ed, they confirm I only had the 5 loans, but ECMC has not removed them nor the default status.•\tI asked about what is next so I do not miss a payment and get garnished again, no response.•\tI went back to the rehab specialist with the signed contract about the 5 loans included and the payment amount that was a direct withdrawal from my bank account, but they were still taking money from my paycheck also not in the contract.•\tFinally the financial statement with the new fees added has put the account, which I have paid over 3k on back up to XXXX. I wrote to them citing the documentation from the rehab and the Dept of Ed. But have not heard anything. They are financially robbing me and there seems to be no recourse for their actions. I cannot go through another wage garnishment. They lied and never informed me, I had no opportunity to have a hearing and they knew it. Even my payroll department hear backs that up and I have seen complaints online that others had the same experience. I was paying on

```
my loans at the time they garnished me and they are not supposed to do that
unless they counseled me on the amount and tried to handle it that way per one
of their own Reps. I also know the people at my old address and they confirmed
they never received that notice nor a certified letter and they would have
brought it to me as they often bring me my mail. This company is a complete
Fraud.Today, I just found out by logging in the ECMC website they have sold my
loans BACK to XXXX and added on XXXX in fees.. How can they do that? I have a
claim against XXXX and it is because of them this all started. I also do not
understand how this is legal and I have received no information from either
company.  I almost feel this is retaliatory for putting a claim against them. I
do not even know what recourse I have and I have no paperwork or anything other
than the loans listed in the website to understand.  It just say's contact the
new provider (XXXX) for information. I know with ECMC they started a wage
garnishment right away so I have to contact XXXX.These two companies are
committing Fraud and need to be investigated. They are robbing people blind with
bad business practices and countless additional fees and costs.  Student loan
1
Name: count, Length: 2492324, dtype: int64
```

[5]:

```python
plt.figure(figsize=(10,5))
df['Product'].value_counts().plot(kind='bar', color='steelblue')
plt.title("Complaints by Product (Department)")
plt.xticks(rotation=45, ha ='right')
plt.tight_layout()
plt.show()
```



[6]:

```python
# I will just keep top 6
top_products = df['Product'].value_counts().nlargest(6).index
```

```
df = df[df['Product'].isin(top_products)]

print(df['Product'].value_counts())
```

```
Product
Credit reporting or other personal consumer reports
1666408
Credit reporting, credit repair services, or other personal consumer reports
807271
Debt collection
405390
Checking or savings account
168449
Mortgage
139602
Money transfer, virtual currency, or money service
111892
Name: count, dtype: int64
```

[7]:

```
merge_map = {
    'Credit reporting, credit repair services, or other personal consumer␣
 ↪reports': 'Credit reporting',
    'Credit reporting or other personal consumer reports': 'Credit reporting',
    'Debt collection': 'Debt collection',
    'Mortgage': 'Mortgage',
    'Checking or savings account': 'Checking or savings account',
    'Money transfer, virtual currency, or money service': 'Money transfer'
}

df['Product'] = df['Product'].map(merge_map)

df = df.dropna(subset=['Product'])
print(df['Product'].value_counts())
```

```
Product
Credit reporting               2473679
Debt collection                 405390
Checking or savings account     168449
Mortgage                        139602
Money transfer                  111892
Name: count, dtype: int64
```

[8]:

```
# I will downsample each class to 10,000 samples.
df_dsp = (
    df.groupby('Product', group_keys=False)
    .apply(lambda x: x.sample(10000, random_state=56))
    .reset_index(drop=True)
)
```

```
print(df_dsp['Product'].value_counts())
print("Total:", len(df_dsp))
```

```
Product
Checking or savings account    10000
Credit reporting               10000
Debt collection                10000
Money transfer                 10000
Mortgage                       10000
Name: count, dtype: int64
Total: 50000
```

```
/var/folders/p_/_ly9dw594410_6bdw0vy3kh80000gn/T/ipykernel_62281/4086923423.py:4
: FutureWarning: DataFrameGroupBy.apply operated on the grouping columns. This
behavior is deprecated, and in a future version of pandas the grouping columns
will be excluded from the operation. Either pass `include_groups=False` to
exclude the groupings or explicitly select the grouping columns after groupby to
silence this warning.
  .apply(lambda x: x.sample(10000, random_state=56))
```

[9]:

```
df_dsp.rename(columns={'Consumer complaint narrative':'complaint'},␣
 ↪inplace=True)
```

[10]:

```
# Clean the text
import re

def clean_text(text):
    text = text.lower()
    text = re.sub(r'x{2,}', ' ', text) #remove XXXX redacted info
    text = re.sub(r'[^a-z\s]', ' ', text) #remove special chars/numbers
    text = re.sub(r'\s+', ' ', text).strip()
    return text

df_dsp['complaint_clean'] = df_dsp['complaint'].apply(clean_text)

df_dsp[['complaint', 'complaint_clean']].head(3)
```

[10]:

```
                                           complaint  \
0  I am writing to file a complaint against Navy …
1  I am writing to file a complaint against Citib…
2  Capital One changed the joint checking account…

                                     complaint_clean
0  i am writing to file a complaint against navy …
1  i am writing to file a complaint against citib…
2  capital one changed the joint checking account…
```

[11]:

```
# Check the average complaint length

df_dsp['word_count'] = df_dsp['complaint_clean'].apply(lambda x: len(x.split()))

print(df_dsp['word_count'].describe())
```

```
count    50000.000000
mean       186.069500
std        217.313564
min          0.000000
25%         71.000000
50%        118.000000
75%        227.000000
max       5038.000000
Name: word_count, dtype: float64
```

0.1 Classification

I am using TF-IDF and Logistic Regression as my baseline. It's fast, interpretable, and works well on text.

[12]:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df_dsp['label'] = le.fit_transform(df_dsp['Product'])

print(dict(zip(le.classes_, le.transform(le.classes_))))
```

```
{'Checking or savings account': 0, 'Credit reporting': 1, 'Debt collection': 2,
'Money transfer': 3, 'Mortgage': 4}
```

[13]:

```
df_dsp.sample(5)
```

[13]:

```
                                               complaint  \
16701  In accordance with the Fair Credit Reporting a…
1293   Capital One holding back my direct deposit I h…
12222  There is an account from XXXX XXXX XXXX  that …
9818   The Huntington Bank had a national news story …
4787   I received recently a bank statement from Bank…

                           Product  \
16701             Credit reporting
1293   Checking or savings account
12222             Credit reporting
9818   Checking or savings account
4787   Checking or savings account

                                         complaint_clean  word_count  label
16701  in accordance with the fair credit reporting a…         142      1
```

```
1293   capital one holding back my direct deposit i h…          37      0
12222  there is an account from that is aged past yea…          51      1
9818   the huntington bank had a national news story …         158      0
4787   i received recently a bank statement from bank…         212      0
```

[14]:

```python
print(len(df_dsp['complaint_clean']), len(df_dsp['label']))
```

```
50000 50000
```

[15]:

```python
# Train-test split
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    df_dsp['complaint_clean'], df_dsp['label'],
    test_size = 0.2, random_state = 56, stratify = df_dsp['label']
)

print(f"Train:{len(X_train)} | Test:{len(X_test)}")
```

```
Train:40000 | Test:10000
```

[16]:

```python
# TF-IDF Vectorization
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(max_features=30000, ngram_range=(1,2), max_df=0.95,␣
  ↪min_df=5)

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

print(X_train_tfidf.shape)
```

```
(40000, 30000)
```

[17]:

```python
# Train Logistic regression
from sklearn.linear_model import LogisticRegression

mod = LogisticRegression(max_iter=1000, random_state=42)
mod.fit(X_train_tfidf, y_train)
```

[17]:

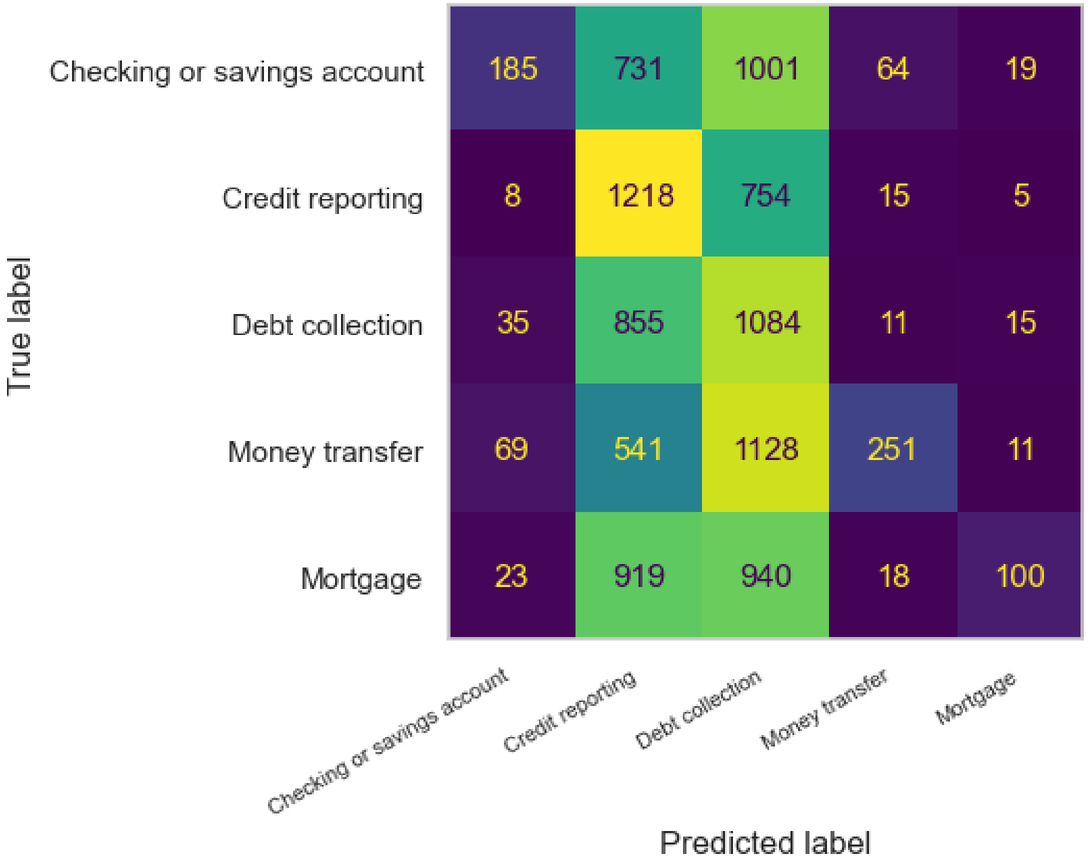
```
LogisticRegression(max_iter=1000, random_state=42)
```

[18]:

```python
# Evaluate
from sklearn.metrics import classification_report, confusion_matrix,␣
  ↪ConfusionMatrixDisplay

y_pred = mod.predict(X_test_tfidf)

print(classification_report(y_pred, y_test, target_names=le.classes_))
```

```
                             precision    recall  f1-score   support

Checking or savings account       0.09      0.58      0.16       320
           Credit reporting       0.61      0.29      0.39      4264
            Debt collection       0.54      0.22      0.31      4907
             Money transfer       0.13      0.70      0.21       359
                   Mortgage       0.05      0.67      0.09       150

                   accuracy                           0.28     10000
                  macro avg       0.28      0.49      0.23     10000
               weighted avg       0.53      0.28      0.33     10000
```

```
[19]: disp = ConfusionMatrixDisplay.from_predictions(y_test, y_pred,␣
       ↪display_labels=le.classes_,
                                                     xticks_rotation=45,␣
       ↪colorbar=False)
      disp.ax_.grid(False)
      disp.ax_.set_xticklabels(le.classes_, rotation=30, ha='right', fontsize=8)
      plt.tight_layout()
      plt.show()
```



Despite balanced training data, logistic regression with TF-IDF collapsed toward a semantically central class due to vocabulary overlap. Check column 3, the model is heavily predicting debt collection, which means when uncertain, it predicts class3. Also notice that how macro avg F1 < than accuracy? This suggest that model performs worse on at least one class and that class is likely a minority class. Despite balancing the class, I encountered class imbalance issue. This demonstrates limitations of bag-of-words models for nuanced complaint categorization. I need to switch to a transformer-based model.

[20]:

```
from transformers import pipeline
embedder = pipeline('feature-extraction', model='distilbert-base-uncased',
                    truncation=True, max_length=128)

def get_embedding(text):
    output = embedder(text[:512])  # truncate input
    return np.mean(output[0], axis=0)  # mean pool

print("Test embedding shape:", get_embedding("test complaint").shape)
```

```
/Users/riteshkc/Desktop/projects/medical_report_summarizer/medical-report-
summarizer/venv/lib/python3.12/site-packages/requests/__init__.py:113:
RequestsDependencyWarning: urllib3 (2.5.0) or chardet
(6.0.0.post1)/charset_normalizer (3.4.3) doesn't match a supported version!
  warnings.warn(
```

```
Test embedding shape: (768,)
```

[21]:

```
# Generate embedding
from tqdm import tqdm
tqdm.pandas()

print("Embedding train set...")
X_train_emb = np.array([get_embedding(t) for t in tqdm(X_train)])

print("Embedding test set...")
X_test_emb = np.array([get_embedding(t) for t in tqdm(X_test)])

print("Done! Shape:", X_train_emb.shape)
```

```
huggingface/tokenizers: The current process just got forked, after parallelism
has already been used. Disabling parallelism to avoid deadlocks…
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true |
false)
```

```
Embedding train set…
```

```
100%|                | 40000/40000 [21:03<00:00, 31.66it/s]
```

```
Embedding test set…

100%|                    | 10000/10000 [05:12<00:00, 32.01it/s]

Done! Shape: (40000, 768)
```

[22]:

```
lr_model = LogisticRegression(max_iter=1000, random_state=56) # samples are␣
  ↪manually balanced to 10K, 'class_weight' removed.
lr_model.fit(X_train_emb, y_train)
print("Done!")
```

```
Done!
```

[23]:

```
y_pred = lr_model.predict(X_test_emb)
print(classification_report(y_test, y_pred, target_names=le.classes_))
```

```
                             precision    recall  f1-score   support

Checking or savings account       0.78      0.81      0.79      2000
           Credit reporting       0.85      0.86      0.85      2000
            Debt collection       0.81      0.81      0.81      2000
             Money transfer       0.86      0.80      0.83      2000
                   Mortgage       0.90      0.92      0.91      2000

                   accuracy                           0.84     10000
                  macro avg       0.84      0.84      0.84     10000
               weighted avg       0.84      0.84      0.84     10000
```

[24]:

```
# Download nltk data
import nltk
nltk.download('punkt')
nltk.download('punkt_tab')
import sumy
```

```
[nltk_data] Downloading package punkt to /Users/riteshkc/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]     /Users/riteshkc/nltk_data…
[nltk_data]   Package punkt_tab is already up-to-date!
```

[25]:

```
# Summarizer function
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer
from sumy.summarizers.lsa import LsaSummarizer

def summarize (text, num_sentences=2):
    parser = PlaintextParser.from_string(text, Tokenizer("english"))
    summarizer = LsaSummarizer()
    summary = summarizer(parser.document, num_sentences)
    return " ".join(str(s) for s in summary)
```

[26]:
```
# use original complaint, not complaint_clean. Because cleaning affected the␣
 ↪sentence organization. Can go back and modify cleaning function.
sample_idx = X_test.index[2]
original = df_dsp.loc[sample_idx, 'complaint']
print("ORIGINAL:\n", original)
print("\nSUMMARY:\n", summarize(original))
```

```
ORIGINAL:
 I filed identity theft report reference # XXXX and sent it with disbute letter
to all bureaus of fraud that I seen, XXXX XXXX XXXX, XXXX. XXXX, child support
accounts XXXX, XXXX, XXXX, XXXX. I sent to bureaus XXXX XXXX Ive been trying to
contact them to no one telling me much of anything please help me dispute this
fraud of my credit report

SUMMARY:
 I filed identity theft report reference # XXXX and sent it with disbute letter
to all bureaus of fraud that I seen, XXXX XXXX XXXX, XXXX. I sent to bureaus
XXXX XXXX Ive been trying to contact them to no one telling me much of anything
please help me dispute this fraud of my credit report
```

[27]:
```
def process_complaint(complaint_text):
    # 1. Summarize
    summary = summarize(complaint_text)

    # 2. Clean for embedding
    cleaned = clean_text(complaint_text)

    # 3. Embed & predict
    embedding = get_embedding(cleaned).reshape(1, -1)
    label_idx = lr_model.predict(embedding)[0]
    department = le.inverse_transform([label_idx])[0]
    confidence = lr_model.predict_proba(embedding).max() * 100

    return {
        "department": department,
        "confidence": f"{confidence:.1f}%",
        "summary": summary
    }
```

[28]:
```
test_complaint = df_dsp['complaint'].iloc[5]
result = process_complaint(test_complaint)

print(f"  COMPLAINT:\n{test_complaint}\n")
print(f"  ROUTE TO: {result['department']}")
print(f"  CONFIDENCE: {result['confidence']}")
print(f"  SUMMARY: {result['summary']}")
```

```
  COMPLAINT:
Withdrawal was made from the account I had by others.

  ROUTE TO: Debt collection
  CONFIDENCE: 65.1%
  SUMMARY: Withdrawal was made from the account I had by others.
```

The pipeline works. But the routing seems slightly off — “Withdrawal from account” should go to Checking or savings account, not Debt collection. This may be expected since it’s a very short complaint with little context for the model to work with.

[29]:

```
# Let's test with a more detailed complaint

complaints = [
    "I have been trying to get a fraudulent account removed from my credit␣
 ↪report for months. The credit bureau keeps verifying the account despite me␣
 ↪sending them proof that I never opened it.",
    "I took out a 30 year fixed mortgage in 2018 and the bank has been charging␣
 ↪me incorrect escrow amounts every month. They refuse to provide a proper␣
 ↪escrow analysis.",
    "A debt collector keeps calling me 5 times a day even after I sent them a␣
 ↪cease and desist letter. They are violating the FDCPA by continuing to␣
 ↪harass me.",
]

for c in complaints:
    result = process_complaint(c)
    print(f"  COMPLAINT: {c[:80]}...")
    print(f"  ROUTE TO: {result['department']}")
    print(f"  CONFIDENCE: {result['confidence']}")
    print(f"  SUMMARY: {result['summary']}")
    print("-" * 60)
```

```
  COMPLAINT: I have been trying to get a fraudulent account removed from my
credit report for…
  ROUTE TO: Credit reporting
  CONFIDENCE: 82.9%
  SUMMARY: I have been trying to get a fraudulent account removed from my credit
report for months. The credit bureau keeps verifying the account despite me
sending them proof that I never opened it.
------------------------------------------------------------
  COMPLAINT: I took out a 30 year fixed mortgage in 2018 and the bank has been
charging me in…
  ROUTE TO: Mortgage
```

```
 CONFIDENCE: 99.2%
 SUMMARY: I took out a 30 year fixed mortgage in 2018 and the bank has been
charging me incorrect escrow amounts every month. They refuse to provide a
proper escrow analysis.
------------------------------------------------------------
 COMPLAINT: A debt collector keeps calling me 5 times a day even after I sent
them a cease a…
 ROUTE TO: Debt collection
 CONFIDENCE: 99.9%
 SUMMARY: A debt collector keeps calling me 5 times a day even after I sent
them a cease and desist letter. They are violating the FDCPA by continuing to
harass me.
------------------------------------------------------------
```

[30]:

```
# Save model aftifacts
import pickle

with open('model.pkl', 'wb') as f:
    pickle.dump(lr_model, f)

with open('label_encoder.pkl', 'wb') as f:
    pickle.dump(le, f)

print("Saved!")
```

```
Saved!
```

[31]:

```
# Lets try SBERT
!{sys.executable} -m pip install sentence-transformers
```

```
zsh:1: parse error near `-m'
```

```
huggingface/tokenizers: The current process just got forked, after parallelism
has already been used. Disabling parallelism to avoid deadlocks…
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true |
false)
```

[32]:

```
# Lets try SBERT
from sentence_transformers import SentenceTransformer

sbert = SentenceTransformer('all-mpnet-base-v2')

print("Embedding train set...")
X_train_sbert = sbert.encode(X_train.tolist(), batch_size=64,␣
 ↪show_progress_bar=True)

print("Embedding test set...")
```

```python
X_test_sbert = sbert.encode(X_test.tolist(), batch_size=64,␣
 ↪show_progress_bar=True)

print("Shape:", X_train_sbert.shape)
```

```
/Users/riteshkc/Desktop/projects/medical_report_summarizer/medical-report-
summarizer/venv/lib/python3.12/site-
packages/huggingface_hub/file_download.py:945: FutureWarning: `resume_download`
is deprecated and will be removed in version 1.0.0. Downloads always resume when
possible. If you want to force a new download, use `force_download=True`.
  warnings.warn(
```

```
Embedding train set…
```

```
Batches:   0%|          | 0/625 [00:00<?, ?it/s]
```

```
Embedding test set…
```

```
Batches:   0%|          | 0/157 [00:00<?, ?it/s]
```

```
Shape: (40000, 768)
```

[33]:

```python
from sklearn.svm import LinearSVC
from sklearn.calibration import CalibratedClassifierCV

svm = LinearSVC(max_iter=2000, class_weight='balanced')
svm_calibrated = CalibratedClassifierCV(svm)  # needed for predict_proba
svm_calibrated.fit(X_train_sbert, y_train)
print("Done!")
```

```
Done!
```

[34]:

```python
y_pred_sbert = svm_calibrated.predict(X_test_sbert)
print(classification_report(y_test, y_pred_sbert, target_names=le.classes_))
```

```
                             precision    recall  f1-score   support

Checking or savings account       0.84      0.88      0.86      2000
           Credit reporting       0.87      0.88      0.87      2000
            Debt collection       0.87      0.86      0.86      2000
             Money transfer       0.90      0.86      0.88      2000
                   Mortgage       0.95      0.96      0.95      2000

                   accuracy                           0.89     10000
                  macro avg       0.89      0.89      0.89     10000
               weighted avg       0.89      0.89      0.89     10000
```

[35]:

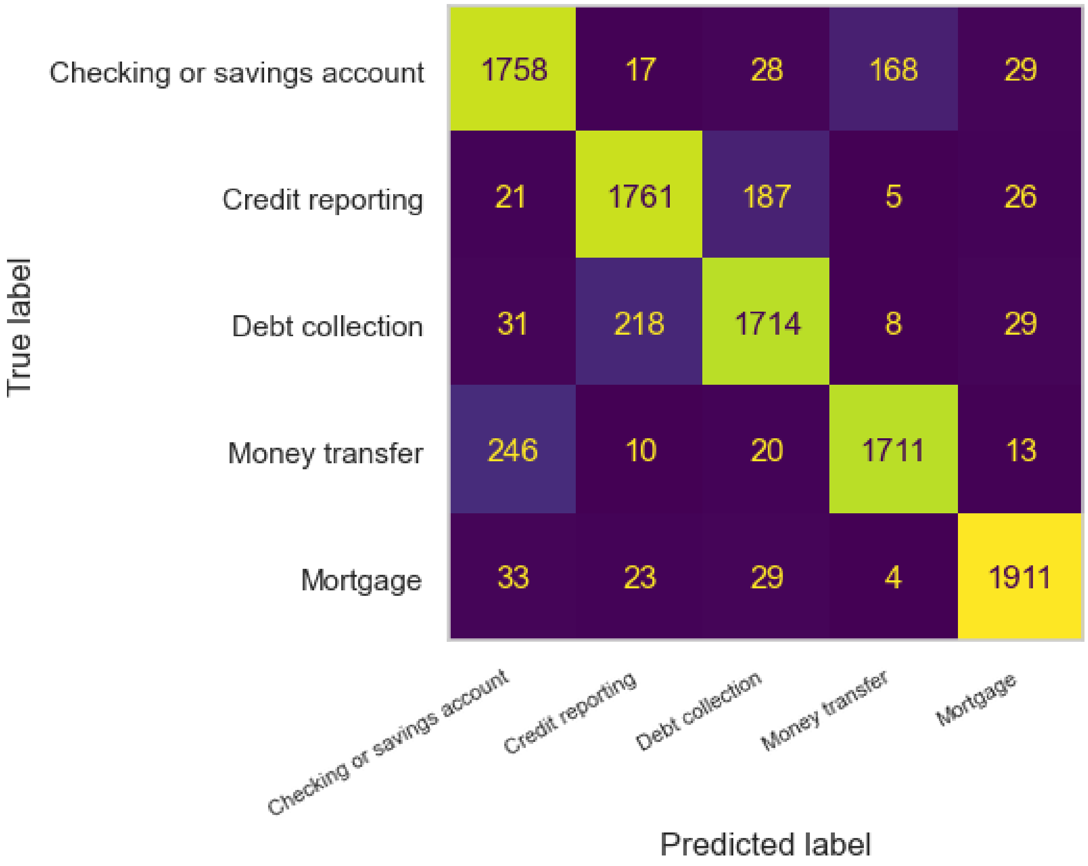
```python
import pickle

with open('svm_model.pkl', 'wb') as f:
    pickle.dump(svm_calibrated, f)

with open('label_encoder.pkl', 'wb') as f:
    pickle.dump(le, f)

print("Saved!")
```

```
Saved!
```

[36]:

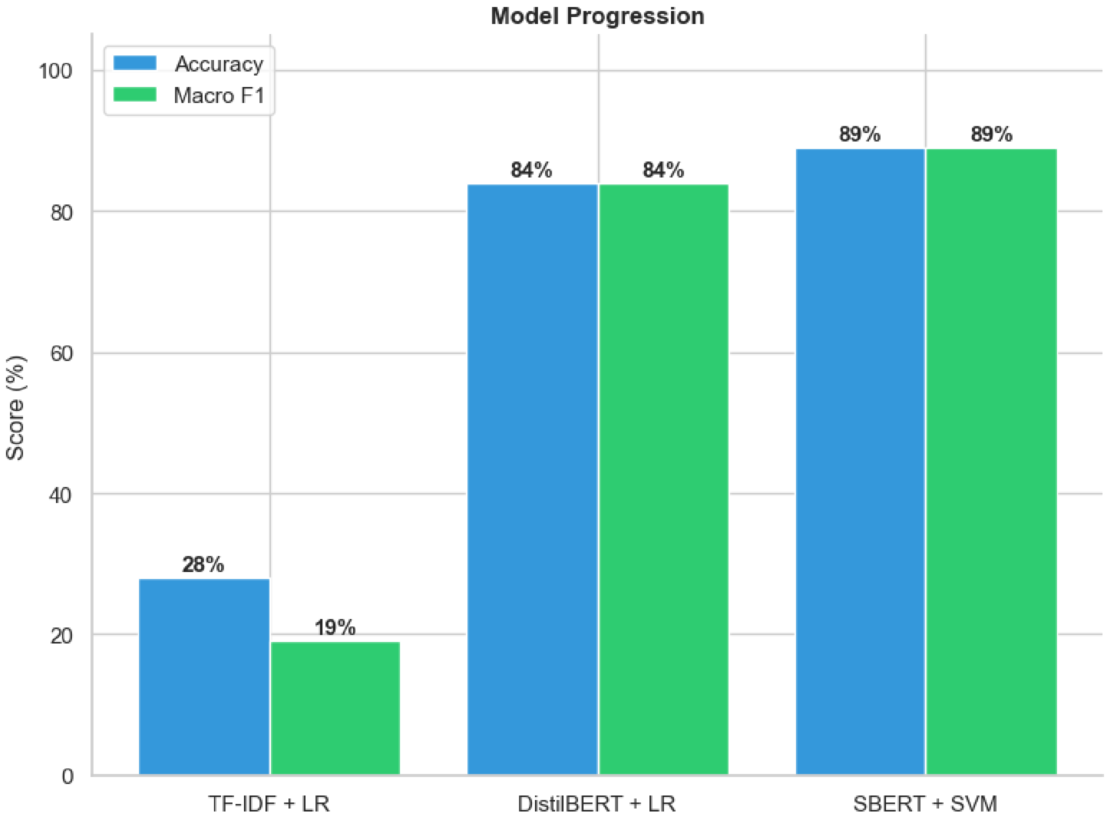
```
# Confusion matrix for final model.
plt.figure(figsize=(8,6))
cm_final = ConfusionMatrixDisplay.from_predictions(y_test, y_pred_sbert,␣
  ↪display_labels=le.classes_,
                                                  xticks_rotation=45,␣
  ↪colorbar=False)
cm_final.ax_.grid(False)
cm_final.ax_.set_xticklabels(le.classes_, rotation=30, ha='right', fontsize=8)
plt.tight_layout()

plt.savefig("figures/sbert_cm.png", dpi=300, bbox_inches="tight")

plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



[37]:

```python
# Create a combined bar chart
models = ['TF-IDF + LR', 'DistilBERT + LR', 'SBERT + SVM']
accuracy = [28, 84, 89]
macro_f1 = [19, 84, 89]

x = np.arange(len(models))
fig, ax = plt.subplots(figsize=(8, 6))

b1 = ax.bar(x - 0.2, accuracy, 0.4, label='Accuracy', color='#3498db')
b2 = ax.bar(x + 0.2, macro_f1, 0.4, label='Macro F1', color='#2ecc71')

for bar in b1 + b2:
    ax.text(bar.get_x() + 0.2, bar.get_height() + 1,
            f'{bar.get_height()}%', ha='center', fontsize=11, fontweight='bold')

ax.set_xticks(x)
ax.set_xticklabels(models)
ax.set_ylim(0, 105)
ax.set_ylabel('Score (%)')
ax.set_title('Model Progression', fontweight='bold')
ax.legend(loc='upper left')
ax.spines[['top', 'right']].set_visible(False)
plt.tight_layout()

plt.savefig("figures/model_comparison.png", dpi=300, bbox_inches="tight")

plt.show()
```


Model Progression
Accuracy
Macro F1
Score (%)
TF-IDF + LR
DistilBERT + LR
SBERT + SVM