Financial Consumer Complaint Classification

By Qidu Fu, Yuqi Liu, Shixin Pan, Haotian Shen

CIS 600

Syracuse University

Qidu Fu: Overview & preprocessing part2 (later)

Yuqi Liu: Data-loading & preprocessing part1

Qidu Fu: Preprocessing part2 & vectorization

Shixin Pan: Modeling part 1

Haotian Shen: Modeling part 2 & discussion

Syracuse University

Overview: problems and objectives

Problems:

Financial consumer complaint texts are 1) of large volume, 2) unstructured, 3) inefficient for manual processing.

Objectives: to better understand consumer financial issues and improve complaint handling

Understand consumer financial issues

Analyze the complaint data to identify key pain points and issues consumers face with financial services

Improve complaint handling

Leverage insights from complaint analysis to improve complaint resolution processes

Syracuse University

Qidu Fu: Overview & preprocessing part2 (later)

Yuqi Liu: Data-loading & preprocessing part1

Qidu Fu: Preprocessing part2 & vectorization

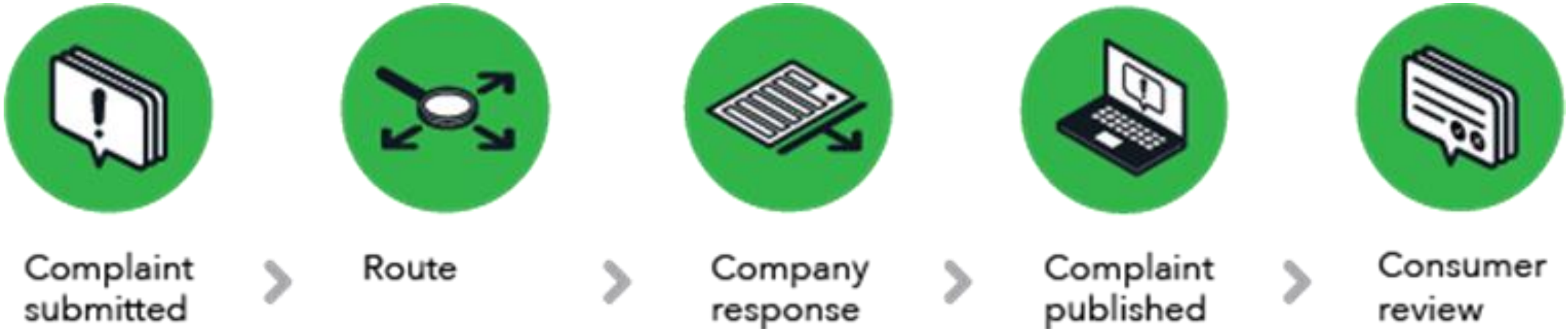
Shixin Pan: Modeling part 1

Haotian Shen: Modeling part 2 & discussion

Syracuse University

Introduce Dataset

The Consumer Complaint Database is a collection of complaints about consumer financial products and services that Consumer Financial Protection Bureau (CFPD) sent to companies for response. Complaints are published after the company responds



Syracuse University

Field reference

- **Use API to connect to data server**
- **Focus on consumer complaint narrative**
- **Extract data from 2023–03-01 to 2024-03-30**
- **Volume: 469,799 complaints**

Field name	Description	Data type	Notes
Date received	The date the CFPB received the complaint	date & time	
Product	The type of product the consumer identified in the complaint	plain text	This field is a categorical variable.
Consumer complaint narrative	Consumer complaint narrative is the consumer-submitted description of "what happened" from the complaint. Consumers must opt-in to share their narrative. We will not publish the narrative unless the consumer consents, and consumers can opt-out at any time. The CFPB takes reasonable steps to scrub personal information from each complaint that could be used to identify the consumer.	plain text	Consumers' descriptions of what happened are included if consumers consent to publishing the description and after we take steps to remove personal information.
Complaint ID	The unique identification number for a complaint	number	

Syracuse University

Preprocessing part 1

Import all necessary libraries

- ‘nltk’, ‘re’, ‘string’
- ‘WordNetLemmatizer’, ‘PorterStemmer’ from ‘nltk.stem’
- ‘stopwords’ from ‘nltk.corpus’

Drop nan and duplicated rows

Remove duplicate and empty rows using ‘dropna()’ and ‘drop_duplicates()’

Select relevant columns

Select 'complaint_what_happened' and 'issue' columns for following steps

Remove punctuation

Use string.punctuation to remove punctuation

Normalize data

- Remove punctuation using ‘string.punctuation’
- Convert to lowercase
- Eliminate dates, monetary values, non-English characters, extra spaces, and etc. using ‘re’ and Python string methods

Syracuse University

Preprocessing part 1

Import all necessary libraries	**Drop nan and duplicated rows**	Select relevant columns	Remove punctuation	Normalize data
• 'nltk', 're', 'string' • 'WordNetLemmatizer', 'PorterStemmer' from 'nltk.stem' • 'stopwords' from 'nltk.corpus'	Remove duplicate and empty rows using 'dropna()' and 'drop_duplicates()'	Select 'complaint_what_happened' and 'issue' columns for following steps	Use string.punctuation to remove punctuation	• Remove punctuation using 'string.punctuation' • Convert to lowercase • Eliminate dates, monetary values, non-English characters, extra spaces, and etc. using 're' and Python string methods

Syracuse University

Preprocessing part 1

Import all necessary libraries

- ‘nltk’, ‘re’, ‘string’
- ‘WordNetLemmatizer’, ‘PorterStemmer’ from ‘nltk.stem’
- ‘stopwords’ from ‘nltk.corpus’

Drop nan and duplicated rows

Remove duplicate and empty rows using ‘dropna()’ and ‘drop_duplicates()’

Select relevant columns

Select 'complaint_what_happened' and 'issue' columns for following steps

Remove punctuation

Use string.punctuation to remove punctuation

Normalize data

- Remove punctuation using ‘string.punctuation’
- Convert to lowercase
- Eliminate dates, monetary values, non-English characters, extra spaces, and etc. using ‘re’ and Python string methods

Syracuse University

Preprocessing

- ‘nltk’, ‘re’, ‘string’
- ‘WordNetLemmatizer’, ‘PorterStemmer’ from ‘nltk.stem’
- ‘stopwords’ from ‘nltk.corpus’

Remove duplicate and empty rows using ‘dropna()’ and ‘drop_duplicates()’

Select 'complaint_what_happened' and 'issue' columns for following steps

Use string.punctuation to remove punctuation

- Remove punctuation using ‘string.punctuation’
- Convert to lowercase
- Eliminate dates, monetary values, non-English characters, extra spaces, and etc. using ‘re’ and Python string methods

Syracuse University

Preprocessing part 1

Import all necessary libraries

- ‘nltk’, ‘re’, ‘string’
- ‘WordNetLemmatizer’, ‘PorterStemmer’ from ‘nltk.stem’
- ‘stopwords’ from ‘nltk.corpus’

Drop nan and duplicated rows

Remove duplicate and empty rows using ‘dropna()’ and ‘drop_duplicates()’

Select relevant columns

Select 'complaint_what_happened' and 'issue' columns for following steps

Remove punctuation

Use string.punctuation to remove punctuation

Normalize data

- Remove punctuation using ‘string.punctuation’
- Convert to lowercase
- Eliminate dates, monetary values, non-English characters, extra spaces, and etc. using ‘re’ and Python string methods

Syracuse University

Qidu Fu: Overview & preprocessing part2 (later)

Yuqi Liu: Data-loading & preprocessing part1

Qidu Fu: Preprocessing part2 & vectorization

Shixin Pan: Modeling part 1

Haotian Shen: Modeling part 2 & discussion

Syracuse University

Preprocessing part 2 removes missing values and short-long words, stems and lemmatizes words, removes emoticons and emojis, and corrects spelling

Remove rows if the narrative and issues columns NAN/NULL/blanks

Use one-hot encoding or label encoding to convert categorical features into numeric values.

Standardize features by removing the mean and scaling to unit variance to handle varying magnitudes.

Derive new features from existing variables through mathematical transformations to capture new information.

Use techniques like PCA to reduce number of features and avoid overfitting.

Syracuse University

Preprocessing part 2 removes missing values and short-long words, stems and lemmatizes words, removes emoticons and emojis, and corrects spelling

Remove rows if the narrative and issues columns NAN/NULL/blanks

Remove words with lengths < 3 or > 15.

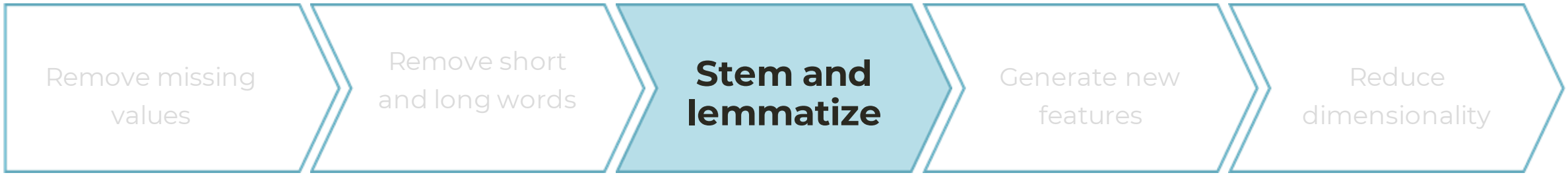
Standardize features by removing the mean and scaling to unit variance to handle varying magnitudes.

Derive new features from existing variables through mathematical transformations to capture new information.

Use techniques like PCA to reduce number of features and avoid overfitting.

Syracuse University

Preprocessing part 2 removes missing values and short-long words, stems and lemmatizes words, removes emoticons and emojis, and corrects spelling



Remove rows if the narrative and issues columns NAN/NULL/blanks

Remove words with lengths < 3 or > 15

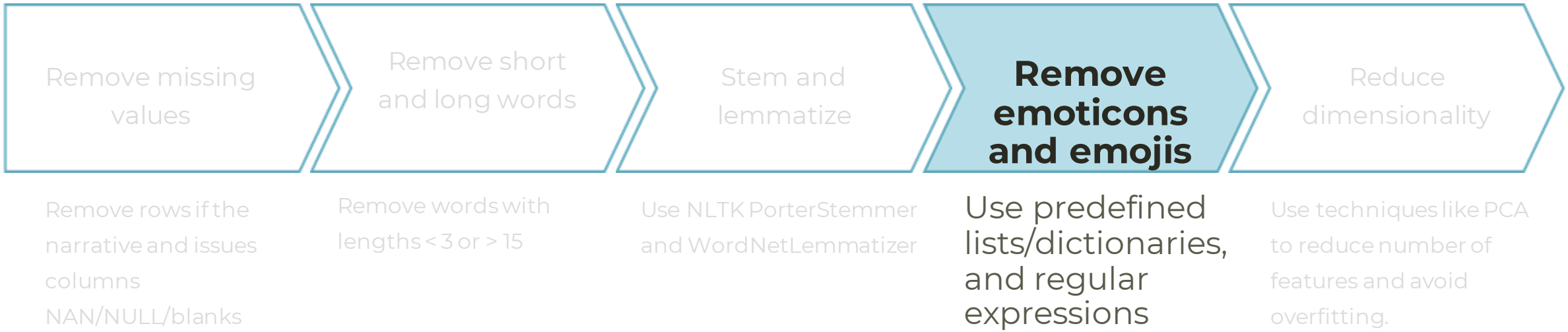
Use NLTK PorterStemmer and WordNetLemmatizer

Derive new features from existing variables through mathematical transformations to capture new information.

Use techniques like PCA to reduce number of features and avoid overfitting.

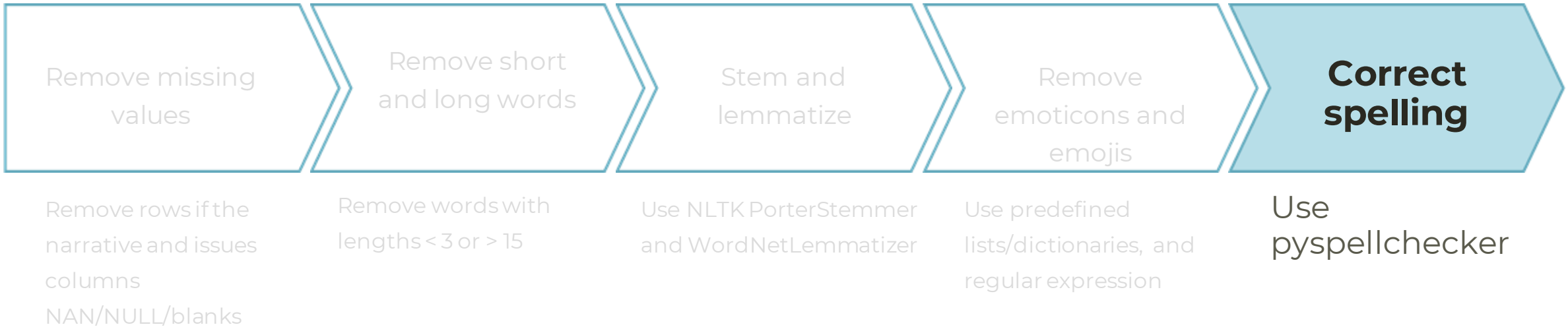
Syracuse University

Preprocessing part 2 removes missing values and short-long words, stems and lemmatizes words, removes emoticons and emojis, and corrects spelling



Syracuse University

Preprocessing part 2 removes missing values and short-long words, stems and lemmatizes words, removes emoticons and emojis, and corrects spelling



Syracuse University

Vectorizing issue and narrative columns with TFIDF from SK-Learn, Mini-BERT embedding from Hugging Face, and/or self-embedding from a transformer encoder

TFIDF	Naïve Bayes Classifier
Mini-Bert embedding	K-Means, GXBoosting, Random-Forest
Self-Trained embedding	BertClassifier: transformer encoder

Syracuse University

Qidu Fu: Overview & preprocessing part2 (later)

Yuqi Liu: Data-loading & preprocessing part1

Qidu Fu: Preprocessing part2 & vectorization

Shixin Pan: Modeling part 1

Haotian Shen: Modeling part 2 & discussion

Syracuse University

Modelling Part

Unsupervised learning

K-means

supervised learning

Support Vector Machines

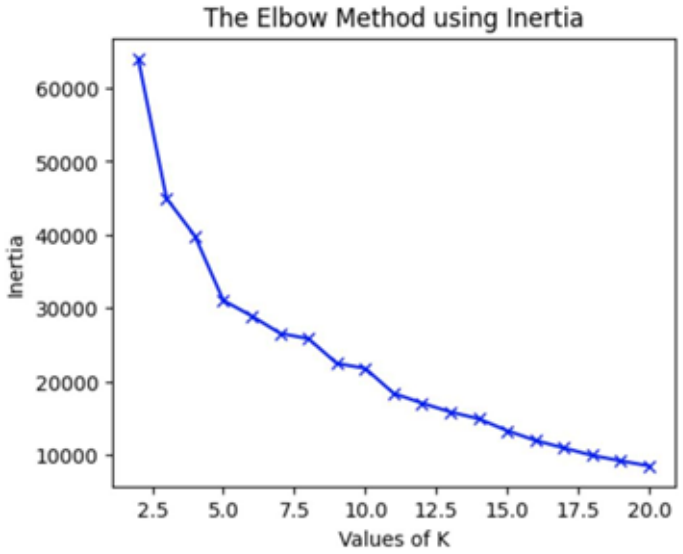
Naive Bayes

Random Forests

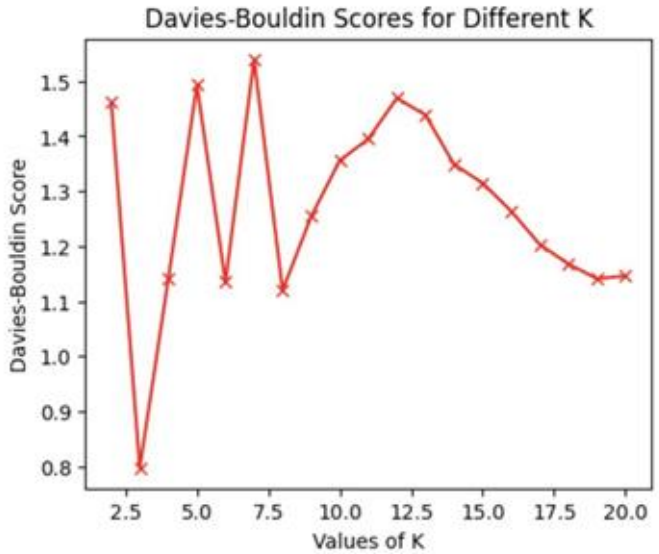
...

Syracuse University

K-means clustering



The Elbow Method



Davies-Bouldin Scores

K=8

Syracuse University

K-means clustering

```
K = 8
kmeans_model = KMeans(n_clusters=K, n_init='auto', random_state=600)
kmeans_model.fit(df_selected['issue_vec'].tolist())
df_selected['label'] = kmeans_model.labels_
pd.set_option('display.max_colwidth', None)
for label in range(K):
    label_rows = df_selected[df_selected['label'] == label].sample(10, random_state=600)
```

```
Label 0:
280012    incorrect inform report
418985    incorrect inform report
42031     incorrect inform report
12706     incorrect inform report
455767    incorrect inform report
313076    incorrect inform report
128465    incorrect inform report
373739    incorrect inform report
42139     incorrect inform report
323618    incorrect inform report
```

```
Label 1:
296419                          problem fraud alert secur freez
468971    problem credit report compani investig exist problem
230017                          problem purchas transfer
181808                               featur term problem
422062    problem credit report compani investig exist problem
231688    problem credit report compani investig exist problem
403592    problem credit report compani investig exist problem
217703    problem credit report compani investig exist problem
106819    problem credit report compani investig exist problem
178221    problem credit report compani investig exist problem
49165     problem credit report compani investig exist problem
```

```
Label 2:
97143     improp use report
12336     improp use report
67157     improp use report
450904    improp use report
19893     improp use report
274736    improp use report
423075    improp use report
201833    improp use report
151034    improp use report
277582    improp use report
```

```
Label 3:
241184    attempt collect debt owe
132787    attempt collect debt owe
166931    attempt collect debt owe
412645    attempt collect debt owe
311651    attempt collect debt owe
261074    attempt collect debt owe
202315    attempt collect debt owe
8994      attempt collect debt owe
224659    attempt collect debt owe
59648     attempt collect debt owe
```

```
Label 4:
45441                troubl payment process
103404               troubl payment process
19023     advertis market includ promot offer
284703    took threaten take neg legal action
20532     appli mortgag refinanc exist mortgag
460706                         get loan lea
407889                         get loan lea
20789                troubl payment process
294089                   deal lender servic
457219                         fraud scam
```

```
Label 5:
301279    manag account
261290    close account
407216    manag account
167147    manag account
199860    manag account
149385     open account
215722    manag account
289818    manag account
75181     manag account
339361    close account
```

```
Label 6:
51430     problem compani investig exist problem
140138    problem compani investig exist problem
107958    problem compani investig exist problem
145862    problem compani investig exist problem
65475     problem compani investig exist problem
370452    problem compani investig exist problem
318280    problem compani investig exist problem
333603    problem compani investig exist problem
195638    problem compani investig exist problem
381012    problem compani investig exist problem
```

```
Label 7:
272126    written notif debt
431494    written notif debt
372856    written notif debt
273369    written notif debt
147563    written notif debt
80248     written notif debt
434198    written notif debt
164538    written notif debt
260764    written notif debt
20879     written notif debt
```

Syracuse University

Naive Bayes

```
X_train = unigram_tfidf_vectorizer.fit_transform(df_train['complaint_what_happened'].tolist())
X_test = unigram_tfidf_vectorizer.transform(df_test['complaint_what_happened'].tolist())
y_train = df_train['label'].tolist()
y_test = df_test['label'].tolist()
nb = MultinomialNB()
nb.fit(X_train, y_train)
y_pred = nb.predict(X_test)
```

	precision	recall	f1-score	support
0	0.71	0.52	0.60	28021
1	0.41	0.13	0.19	17502
2	0.74	0.64	0.69	20294
3	0.28	0.59	0.38	3325
4	0.59	0.63	0.61	9617
5	0.39	0.91	0.55	5009
6	0.34	0.65	0.45	8711
7	0.15	0.51	0.23	1481
accuracy			0.52	93960
macro avg	0.45	0.57	0.46	93960
weighted avg	0.57	0.52	0.51	93960

Syracuse University

Support Vector Machine

```
svm = SVC(class_weight='balanced')
svm.fit(df_train['complaint_what_happened_vec'].tolist(),
df_train['label'].tolist())
y_pred = svm.predict(df_test['complaint_what_happened_vec'].tolist())
```

	precision	recall	f1-score	support
0	0.73	0.55	0.62	28021
1	0.50	0.20	0.29	17502
2	0.70	0.73	0.71	20294
3	0.24	0.48	0.32	3325
4	0.56	0.60	0.58	9617
5	0.43	0.78	0.55	5009
6	0.39	0.60	0.48	8711
7	0.14	0.51	0.22	1481
accuracy			0.54	93960
macro avg	0.46	0.56	0.47	93960
weighted avg	0.59	0.54	0.54	93960

Syracuse University

Random Forest

```
rfc = RandomForestClassifier()
rfc.fit(df_train['complaint_what_happened_vec'].tolist(),  df_train['label'].tolist())
y_pred = rfc.predict(df_test['complaint_what_happened_vec'].tolist())
```

	precision	recall	f1-score	support
0	0.80	0.59	0.68	28021
1	0.65	0.28	0.39	17502
2	0.77	0.75	0.76	20294
3	0.20	0.51	0.28	3325
4	0.50	0.59	0.54	9617
5	0.38	0.73	0.50	5009
6	0.54	0.64	0.59	8711
7	0.15	0.61	0.23	1481
accuracy			0.58	93960
macro avg	0.50	0.59	0.50	93960
weighted avg	0.66	0.58	0.59	93960

Syracuse University

Qidu Fu: Overview & preprocessing part2 (later)

Yuqi Liu: Data-loading & preprocessing part1

Qidu Fu: Preprocessing part2 & vectorization

Shixin Pan: Modeling part 1

Haotian Shen: Modeling part 2 & discussion

Syracuse University

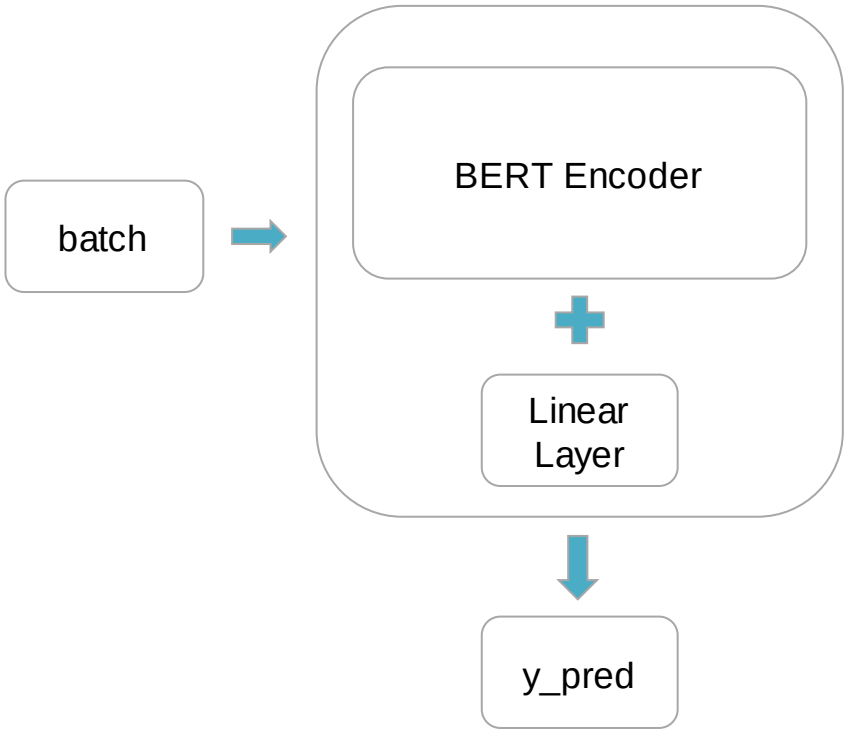
Gradient Boosting

```
hgbc = HistGradientBoostingClassifier()
hgbc.fit(df_train['complaint_what_happened_vec'].tolist(),  df_train['label'].tolist())
y_pred = hgbc.predict(df_test['complaint_what_happened_vec'].tolist())
```

	precision	recall	f1-score	support
0	0.79	0.58	0.67	28021
1	0.55	0.31	0.40	17502
2	0.75	0.75	0.75	20294
3	0.23	0.50	0.32	3325
4	0.54	0.58	0.56	9617
5	0.43	0.75	0.54	5009
6	0.47	0.64	0.54	8711
7	0.17	0.60	0.27	1481
accuracy			0.58	93960
macro avg	0.49	0.59	0.51	93960
weighted avg	0.63	0.58	0.59	93960

Syracuse University

BERT




```
Epoch 1/10, Loss: 1.4160
Epoch 2/10, Loss: 1.1393
Epoch 3/10, Loss: 1.0504
Epoch 4/10, Loss: 0.9948
Epoch 5/10, Loss: 0.9463
Epoch 6/10, Loss: 0.9053
Epoch 7/10, Loss: 0.8694
Epoch 8/10, Loss: 0.8378
Epoch 9/10, Loss: 0.8041
Epoch 10/10, Loss: 0.7753
              precision    recall  f1-score   support

           0       0.81      0.62      0.70     28021
           1       0.61      0.40      0.48     17502
           2       0.76      0.77      0.77     20294
           3       0.27      0.67      0.38      3325
           4       0.71      0.60      0.65      9617
           5       0.53      0.86      0.65      5009
           6       0.48      0.64      0.54      8711
           7       0.24      0.71      0.36      1481

    accuracy                           0.63     93960
   macro avg       0.55      0.66      0.57     93960
weighted avg       0.68      0.63      0.64     93960
```

Syracuse University