

Machine Learning, Sentiment Analysis, and the United Healthcare CEO Shooting

Amrita Pathak, Hailey Hansen, Yolanda Pan & Zixuan Zhou

Our Research Question and Motivation

Question: How has public opinion related to health insurance changed in the wake of the UnitedHealthcare CEO shooting?

Why this question?

- Radical events generate strong emotional reactions
 - Public opinion related to health insurance is not always evident
 - Internet discourse related to this topic is generally limited to enrollment season or anecdotal experiences involving bad health insurance going viral
- The UnitedHealthcare CEO shooting generated significant increase in internet discourse regarding health insurance in a short amount of time
- Personal significance: USHIP (UChicago health insurance) is a UnitedHealthcare plan

Literature Review

Title	Authors	Data	Methodology
Analyzing Twitter Data to Evaluate People's Attitudes towards Public Health Policies and Events in the Era of COVID-19	Tsai, M. H., & Wang, Y. (2021).	<ul style="list-style-type: none">• Real-time tweets collected every 3 days from 1 March 2020 to 14 June 2020 (Total: 37 days)• Tweepy for accessing twitter API• Keywords: "coronavirus", "COVID-19", "COVID19", "COVID_19", "SARSCOV2", "SARS-COV-2", and "SARS_COV_2"	<ul style="list-style-type: none">• Sentiment levels: very negative, negative, neutral, positive, and very positive (on a scale of -2 to 2)• Sentiment Analysis: recursive neural tensor network (RNTN) using Stanford CoreNLP• Sentiment score calculation: late-trained model to compute the overall sentiment scores of each collection day (surveillance system of people's sentiment)
Customer Sentiment Analysis and Prediction of Insurance Products' Reviews Using Machine Learning Approaches	Hossain, M. S., & Rahman, M. F. (2023)	<ul style="list-style-type: none">• Downloaded directly from Yelp (8,635,403 ratings for 160,585 business organizations in the initial Yelp list, including 507 insurance providers)• Filtered for insurance-related reviews using machine learning	<ul style="list-style-type: none">• Sentiment levels: negative, neutral, positive• *Sentiment analysis: two unsupervised learning built-in packages - AFINN sentiment and VADER• Sentiment prediction: decision tree, K Neighbours, SVM, logistic regression, and random forest

Literature Review (cont.)

Title	Authors	Data	Methodology
Using Social Media to Identify Consumers' Sentiments towards Attributes of Health Insurance during Enrollment Season	Van Den Broek-Altenburg, E. M., & Atherly, A. J. (2019)	<ul style="list-style-type: none">• Twitter API; used keywords of "health insurance" and "health plan"; collected approximately 2.7 million tweets from November 1, 2016 to January 31, 2017• Word vectorization and dimension reduction	<ul style="list-style-type: none">• Used Word Association to find words closely associated with keywords such as "premium," "access," and "network", as well as how words denoting trust/fear are related to medical terms/insurance practices• Sentiment levels: binary positive/negative as well as classes of emotion: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust• Sentiment analysis: NRC Emolex


RQ Formulated as an ML task:

In order to see how public opinion related to health insurance has changed in the wake of the UnitedHealthcare CEO shooting, we will conduct a machine learning **classification task** to predict whether Reddit posts have positive, negative, or neutral sentiments.

Data Collection

- arctic_shift
- Raw data downloads in .jsonl format
- Document contains dictionaries of data related to each comment, including metadata, date posted (in unicode format), username of poster, and text from the post
- Our dataset: a dictionary mapping each unicode date (key) to its corresponding text content (value)

Download tool



Download posts and comments from a subreddit or user. Very large subreddits can take a long time to download. In that case, you can maybe narrow down the time range. Alternatively, you can download [subreddit dumps](#) through [Academic Torrents](#) or [monthly dumps](#).

r/

u/

healthinsurance

Approximately 61k posts and 471k comments

Start date

End date

2024-11-20

2024-12-18

☒ Download posts

☒ Download comments

New download

Downloaded 2169 Posts in 0:00:07

Download complete 🏁

2024-12-17 23:57:35

2024-12-18

Downloaded 23845 Comments in 0:00:34

Download complete 🏁

2024-12-17 23:59:05

2024-12-18

Data Labeling

	Date	Text	Rating 1	Rating 2	agreement	final
174	2024-11-25_143	ABSOLUTELY a SCAM I am at the doctor s office right now it doesn t even show up as credible after I gave them my ins Card	-99	-1	1	-1
503	2024-11-23_108	It s not special I promise you	-99	-1	1	-99
901	2024-11-22_129	I noticed Forward is currently hiring on Linkedin and that makes zero sense given the circumstances I am concerned about how we were just sent our medical files unencrypted If anyone has heard about a class action regarding HIPAA violations please post so we can all join Finally is there any way to reach out to any of the doctors we saw under Forward I had one particular doctor who really understood all my medical nuances and was so helpful I d love to find her and become an actual patient So disappointed at this whole ordeal	-99	-1	1	-99
2	2024-12-08_321	Yes gross taxable pretax investments lower gross taxable income	-99	0	1	-99
23	2024-12-17_211	Regardless this isn t a sign of a bad company They could be in a state that requires it	-99	0	1	-99
55	2024-12-02_166	I was thinking about using this benefit but wanted to see if people had better experiences recently I thought this might be a nice way to earn towards the Apple Watch Ultra but sort of nervous after reading some of these reviews It seems like if I refresh the app weekly I should be fine but wanted to confirm from those who went through it fully	-99	0	1	-99
78	2024-12-11_304	hours is for hospital stay ONLY This legislation was to prevent mother and baby being discharged too early when certain illnesses for baby may not have presented themselves	-99	0	1	-99
219	2024-11-20_146	Thank you honestly I don t understand why the point you are restricted to Kaiser only doctors is mentioned so often in this context Isn t that the case for every HMO that I am restricted to in network providers	-99	0	1	0
349	2024-12-03_4	Apply for UPS	-99	0	1	-99
412	2024-12-04_17	Right here is your answer https://www.valuepenguin.com/health-insurance-claim-denials-and-appeals	-99	0	1	0
504	2024-12-10_260	yes we had a baby this year	-99	0	1	-99
510	2024-12-07_189	Aetna got a CEO	-99	0	1	0

Data Labeling

This is how we labeled data:

	Rating 1	Rating 2		Positive	1
Hailey	Green	Purple		Neutral	0
Amrita	Orange	Green		Negative	-1
Yolanda	Purple	Blue		Irrelevant to Health Insurance	-99
Charlotte	Blue	Orange			

A good example of the text where we had initial disagreement was a comment something like this:

1	Text	Rating 1	Rating 2	agreement	Final		
293	I d be pissed if my employer switched Right now I have zero deductable and co pays are only Employer pays of the insurance also	1	-1		1	1	

These were the comments that we had to come to a conclusion for - depending on the sentiment towards the health insurance in the comment.

Data Preprocessing

Preprocessing: regex and nltk toolkit

- Using regex to remove punctuation, strip whitespace, remove urls/user handles, replacing unique fonts
- Removing stop words
- Lemmatizing
- Vectorizing with TfidfVectorizer

	ability	able	absolutely	aca	aca marketplace	aca plan	aca pliant	aca subsidy	accept	accept medicaid
0	0.0	0.0	0.406282	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

EDA

- Class distribution for target data:

```
final
0      444
-99    325
-1     196
1       35
Name: count, dtype: int64
```

- Feature space: (7018, 1577) 7018
comments, 1577 features
 - dtypes: all floats

- Top 10 most frequent features:

	feature	frequency
0	insurance	286.341797
1	plan	225.600192
2	pay	161.089628
3	state	158.268268
4	post	157.723278
5	question	151.143759
6	reddit	142.961821
7	healthinsurance	138.359082
8	thank	138.214272
9	solicitation	136.587910

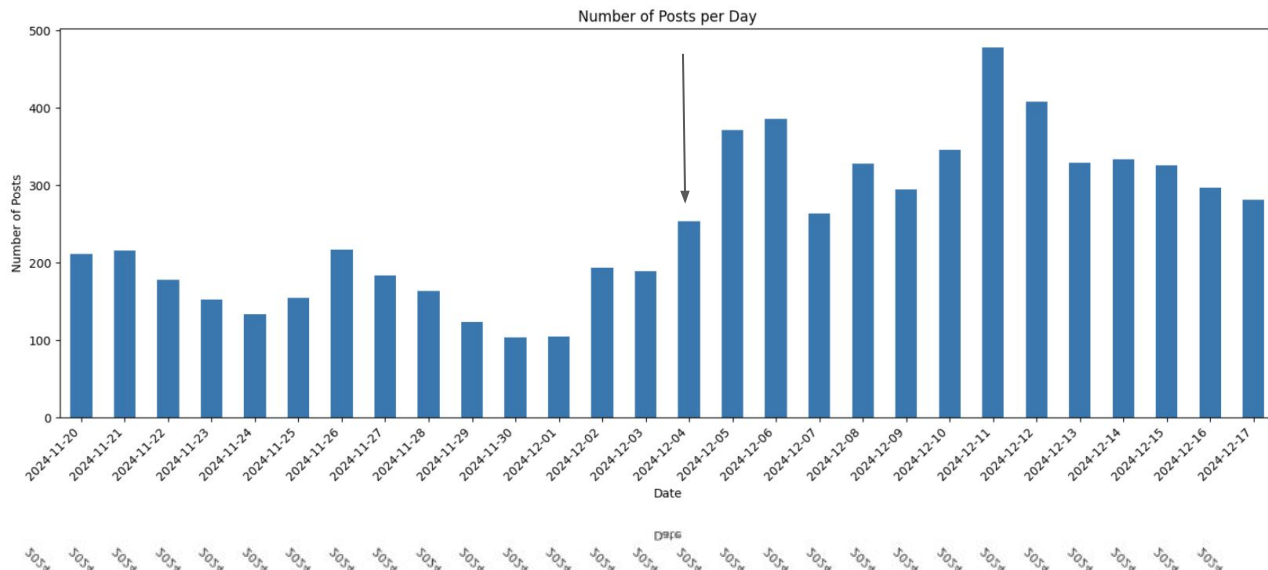
EDA (cont.)

- Distribution of comments in our dataset over time
 - Demonstrates a notable increase in commenting frequency following the date of the shooting (12/4/2024)

Overview

```
plt.figure(figsize=(15,6))
date_counts.plot(kind='bar')

plt.title('Number of Posts per Day')
plt.xlabel('Date')
plt.ylabel('Number of Posts')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Modeling - Decision Tree

- **Training set:** 1, 000 labelled data (70% for training, 30% for testing)

```
X_train, X_test, y_train = train_test_split(X, y, test_size=0.3, random_state=42)
X_train.shape, X_test.shape
✓ 0.0s
((700, 1574), (300, 1574),
```

- **Baseline model:** 1
 - Parameters:
 - Yields an ac
 - Fine tuning

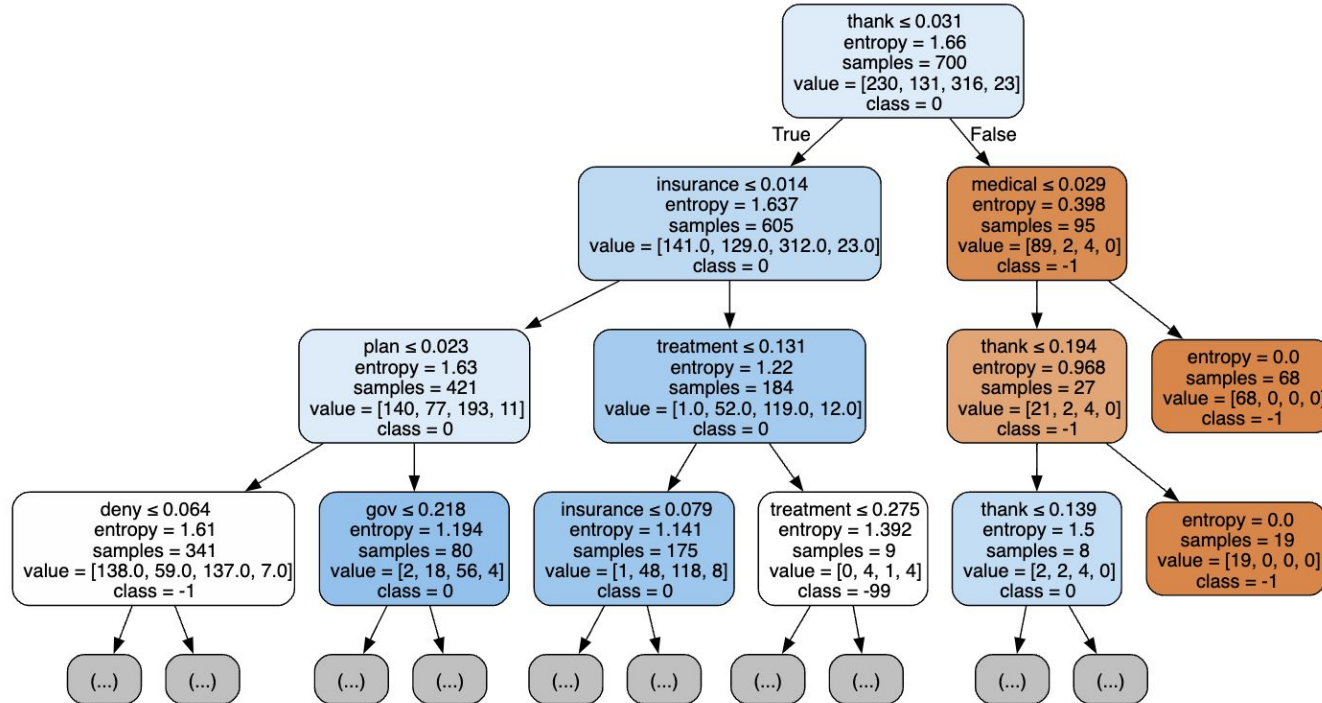
```
param_grid = {
    'max_depth': [n for n in range(10, 1000, 10)],
    'min_samples_split': [2, 5, 10, 20, 30, 40],
    'min_samples_leaf': [1, 2, 4]
}

dt_clf = DecisionTreeClassifier(random_state=42, criterion='entropy')
grid_search = GridSearchCV(dt_clf, param_grid, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)

print("Best parameters:", grid_search.best_params_)
print("Best cross-validation score:", grid_search.best_score_)
```

```
Best parameters: {'max_depth': 40, 'min_samples_leaf': 4, 'min_samples_split': 40}
Best cross-validation score: 0.6228571428571429
```

Modeling - Decision Tree



Modeling - Random Forest

- We want to look for a model with better predictive powers
- **Baseline model:**
 - `n_estimators=1000`, `criterion='entropy'`, `random_state=42`
 - Yields an accuracy score of **0.640**
- **Tuning:**
 - For random forest model fine tuning, dimension reduction/feature selection is crucial
 - Feature selection Method 1 - Using **SelectKBest** for the best 750 features (~50% of features)
 - **SelectKBest** evaluates each feature individually by applying a scoring function to measure the relationship between the feature and the target variable
 - Fine-tuning using Grid Search - best parameters: `max_depth=40`, `max_features="log2"`, `criterion="entropy"`, `n_estimators=2000`, `min_samples_leaf=1`, `min_samples_split=2`, `random_state=42`
 - Accuracy score: 0.640

Selected features: ['absolutely' 'aca' 'aca plan' 'aca pliant' 'accept' 'accident' 'act'
'action' 'action perform' 'actual' 'actually' 'additional' 'advantage'
'advocate' 'aetna' 'affect' 'afford' 'affordability' 'affordable' 'age'
'age state' 'agent' 'ago' 'agree' 'ah' 'allow' 'amaze' 'ambetter'
'ambulance' 'america' 'annual' 'answer' 'answer depend' 'answer question'
'answer reddit' 'anthem' 'appeal' 'application' 'appointment'
'appreciate' 'approval' 'ask' 'automatically' 'automatically contact'
'available' 'available insurance' 'avoid' 'avoid post' 'away' 'bad'
'balance' 'ban' 'ban report' 'bankrupt' 'barely' 'base ine' 'basically'
'bc' 'bcbs' 'bee' 'benefit' 'benefit available' 'best' 'bet' 'big'
'billing' 'billion' 'birth' 'bit' 'blame' 'blood' 'body' 'bone' 'born'
'bot' 'bot action' 'brand' 'breast' 'broken' 'broker' 'buck' 'cal'
'california' 'cancel' 'cancer' 'cap' 'care' 'care act' 'care need'
'carefully' 'carefully avoid' 'case' 'cash' 'cash price' 'catastrophic'
'cause' 'certain' 'chance' 'change' 'charge' 'cheap' 'cheaper' 'choice']

Modeling - Random Forest

- Tuning:

- Feature selection Method 2 - Using only the top 100 features provided by the Decision Tree
- Fine tune the model using Grid Search
- The parameter that gives the best performance:

```
n_estimators=1000, random_state=42
```

- Accuracy score: 0.657
- Recall that baseline is accuracy is 0.64 -> feature selection gives us an **additional 1.7%** accuracy

```
top_100_features = dt.feature_importance.T.sort_values(by=0, ascending=False).head(100)
selected_feature_names = top_100_features.index.tolist()
selected_feature_names
```

✓ 0.0s

```
['thank',
 'insurance',
 'plan',
 'medicaid',
 'pay',
 'money',
 'cover',
 'deny',
 'treatment',
 'medical',
 'network',
 'uhc',
 ...]
```

Top 20 one-gram/two-gram features

Top 20 features, DT model

0	0
thank 0.130	healthcare gov 0.008
insurance 0.092	healthcare cost 0.006
plan 0.048	insurance cover 0.005
medicaid 0.027	marketplace plan 0.004
pay 0.023	health insurance 0.002
money 0.017	plan work 0.000
cover 0.017	plan pay 0.000
deny 0.016	perform automatically 0.000
treatment 0.015	plan need 0.000
medical 0.015	permanent ban 0.000
network 0.012	plan offer 0.000
uhc 0.011	plan plan 0.000
coverage 0.011	plan month 0.000
healthcare 0.010	plan option 0.000
time 0.010	plan employer 0.000
doctor 0.010	plaint state 0.000
look 0.009	plan aca 0.000
try 0.009	pjpg auto 0.000
personally 0.009	plan available 0.000
patient 0.009	plan choose 0.000

Top 20 features, RF model (with selected 100 features)

0	0
insurance 0.081	healthcare gov 0.004
plan 0.053	marketplace plan 0.001
thank 0.048	insurance cover 0.001
pay 0.035	healthcare cost 0.000
medicaid 0.023	
reminder 0.023	
coverage 0.022	
cover 0.020	
year 0.020	
deny 0.019	
medical 0.019	
care 0.018	
network 0.018	
website 0.018	
deductible 0.018	
let 0.017	
benefit 0.017	
time 0.017	
doctor 0.016	
tax 0.016	

Modeling - Logistic Regression

- Baseline model
 - `random_state=42, max_iter=1000`
 - Yields an accuracy score of 0.637
- Fine-tuning with Randomized Search
 - All features are preserved
 - Randomized search picks random combinations of hyperparameter values from the search space, instead of testing for all parameter combinations
 - The randomized search gives us best parameters of 'C': 9.07996948975164, 'max_iter': 5000
 - This yields an accuracy score of **0.687** (higher than both the DT and RF model)
- This is the final model we used to label the rest of our data.

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import uniform

parameter_distributions = {
    'C': uniform(0.001, 10),
    'max_iter': [1000, 5000]
}

random_search = RandomizedSearchCV(
    LogisticRegression(random_state=42),
    param_distributions=parameter_distributions,
    n_iter=10,
    cv=5,
    scoring='accuracy'
)

random_search.fit(X_train, y_train)

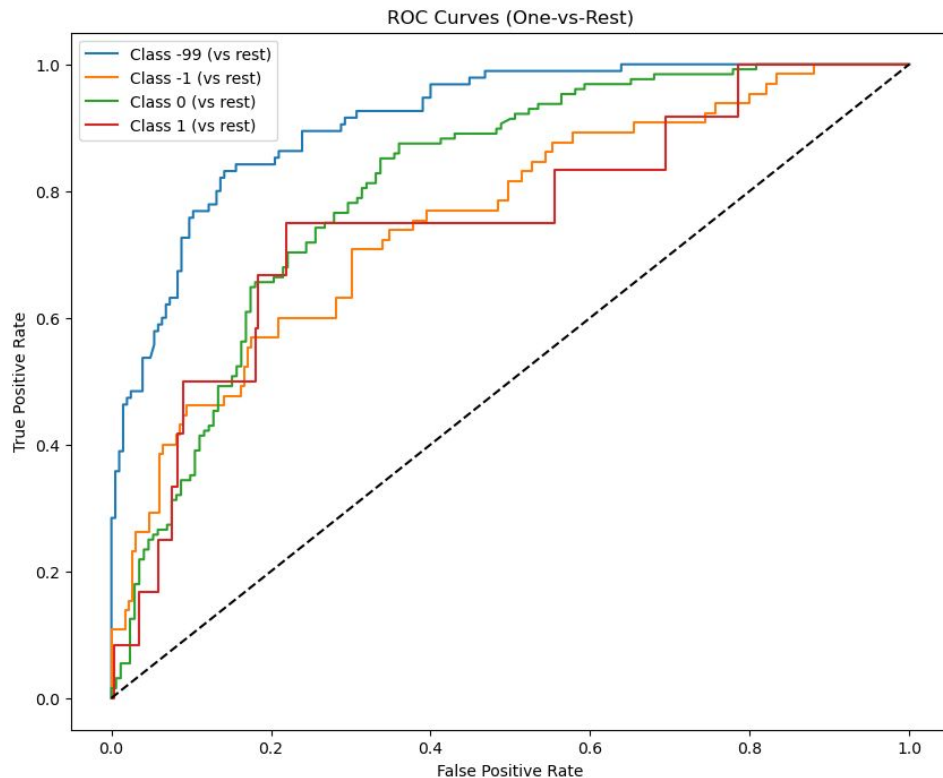
print("Best parameters:", random_search.best_params_)
print("Best cross-validation score:", random_search.best_score_)

✓ 5.8s
```

Best parameters: {'C': 9.07996948975164, 'max_iter': 5000}
Best cross-validation score: 0.6657142857142857

Evaluation on Modeling - Logistic Regression

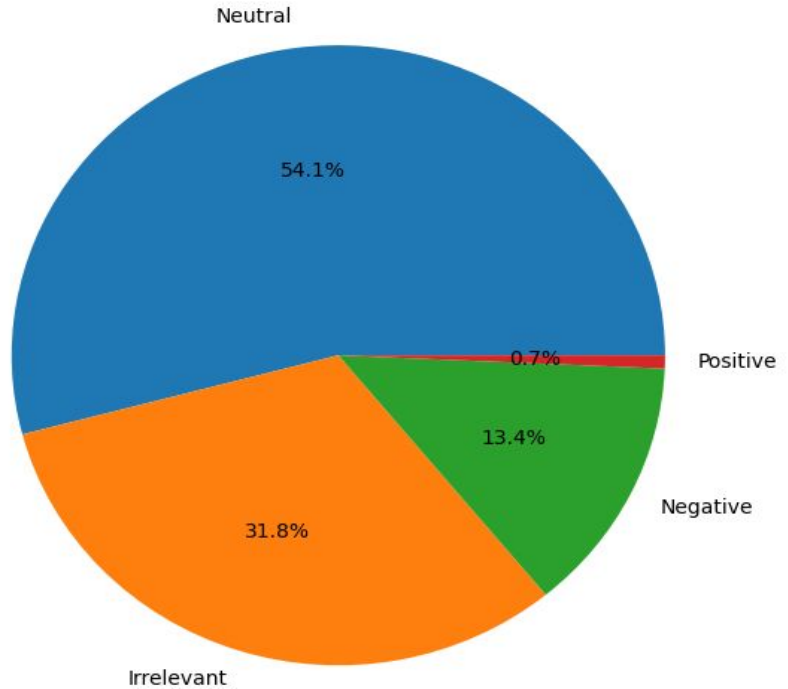
	precision	recall	f1-score	support
-99	0.76	0.77	0.76	95
-1	0.65	0.37	0.47	65
0	0.65	0.85	0.74	128
1	nan	0.00	0.00	12
accuracy			0.69	300
macro avg	0.69	0.50	0.49	300
weighted avg	0.69	0.69	0.66	300



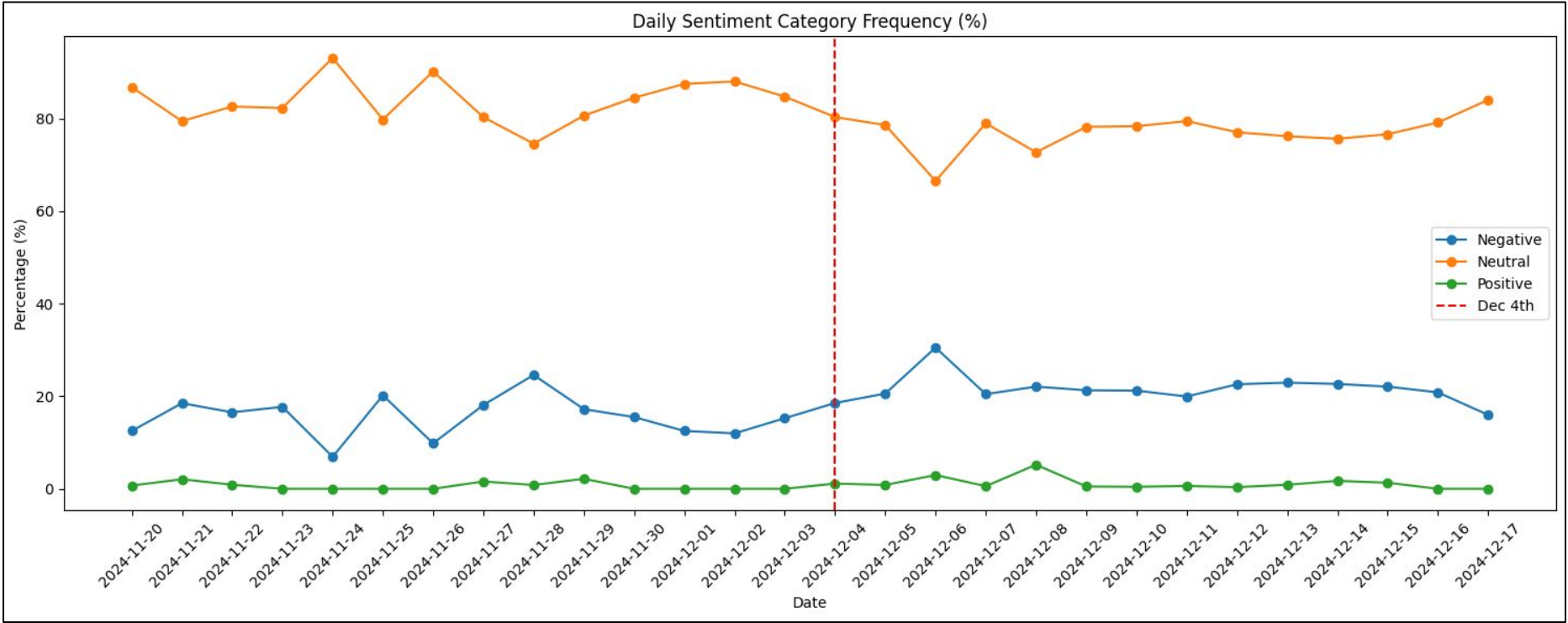
Data Analysis

Overall Sentiment Category Distribution

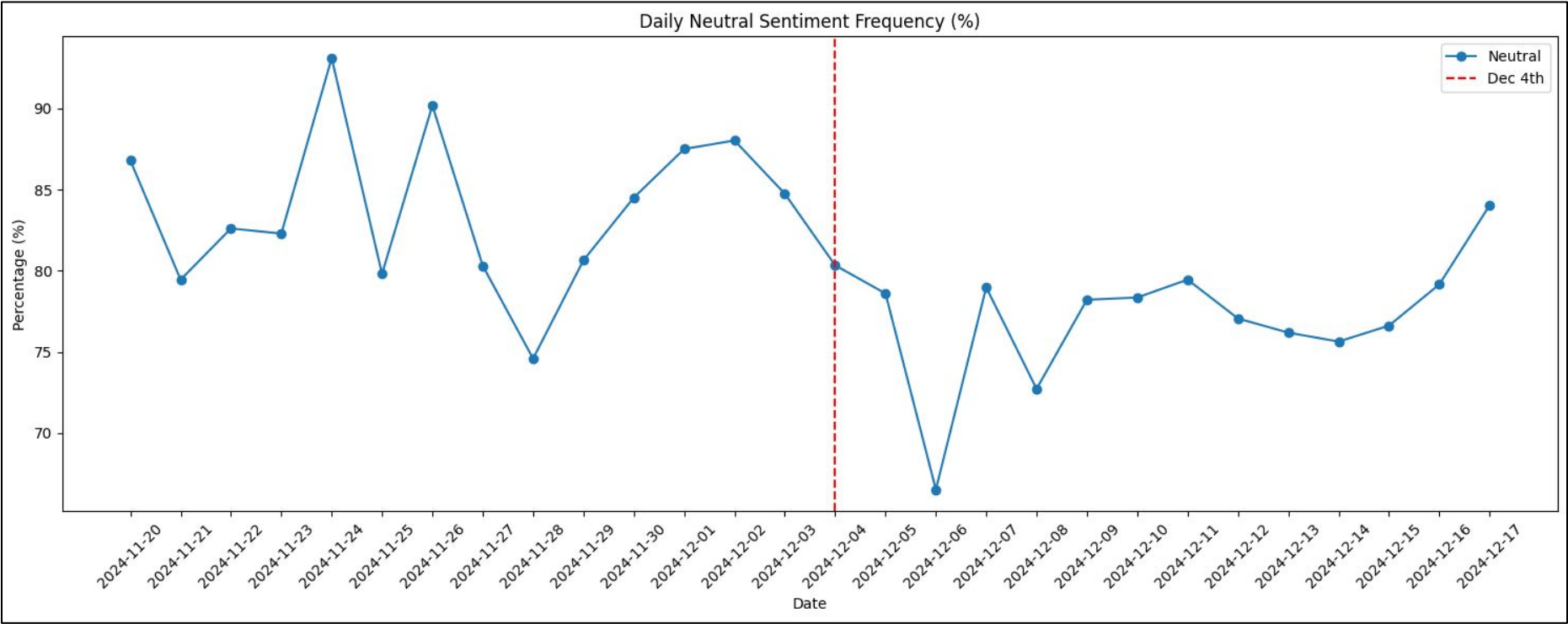
Overall Sentiment Category Distribution



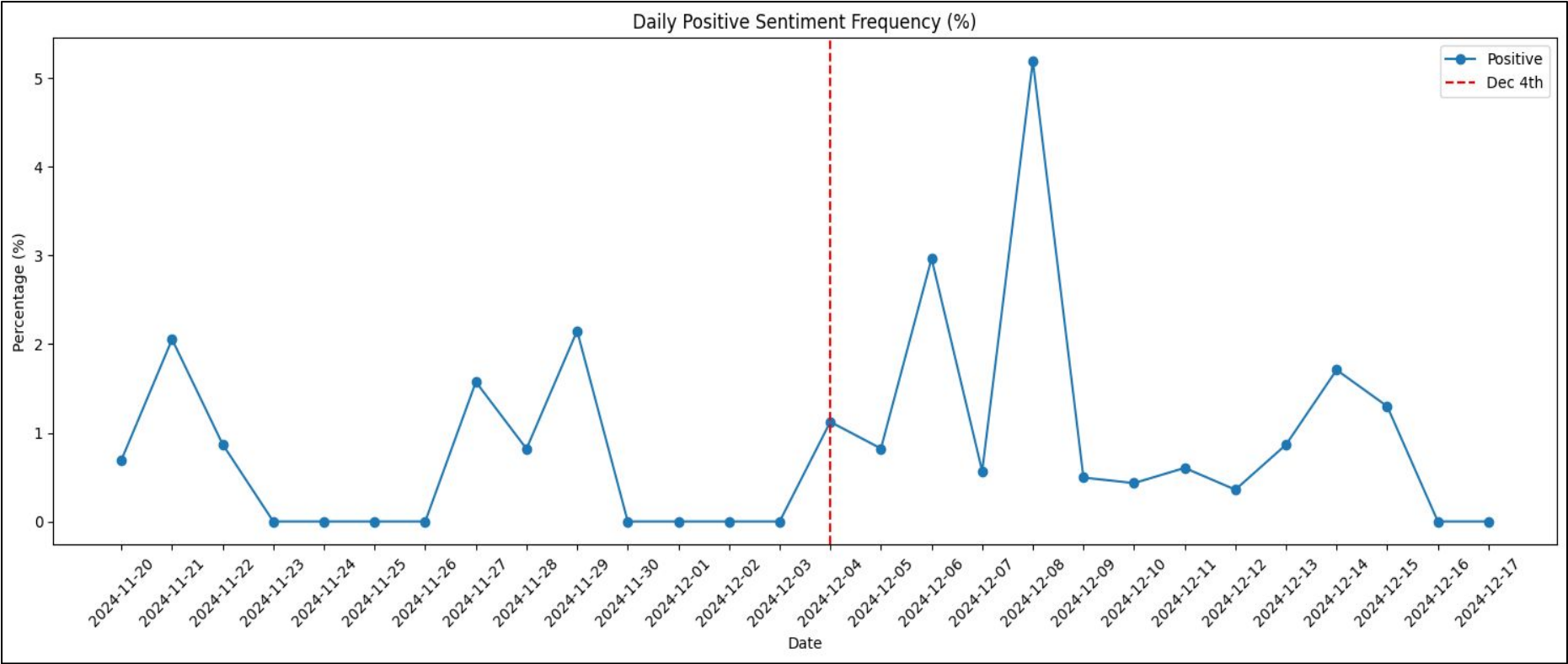
Daily Sentiment Category Frequency (%)



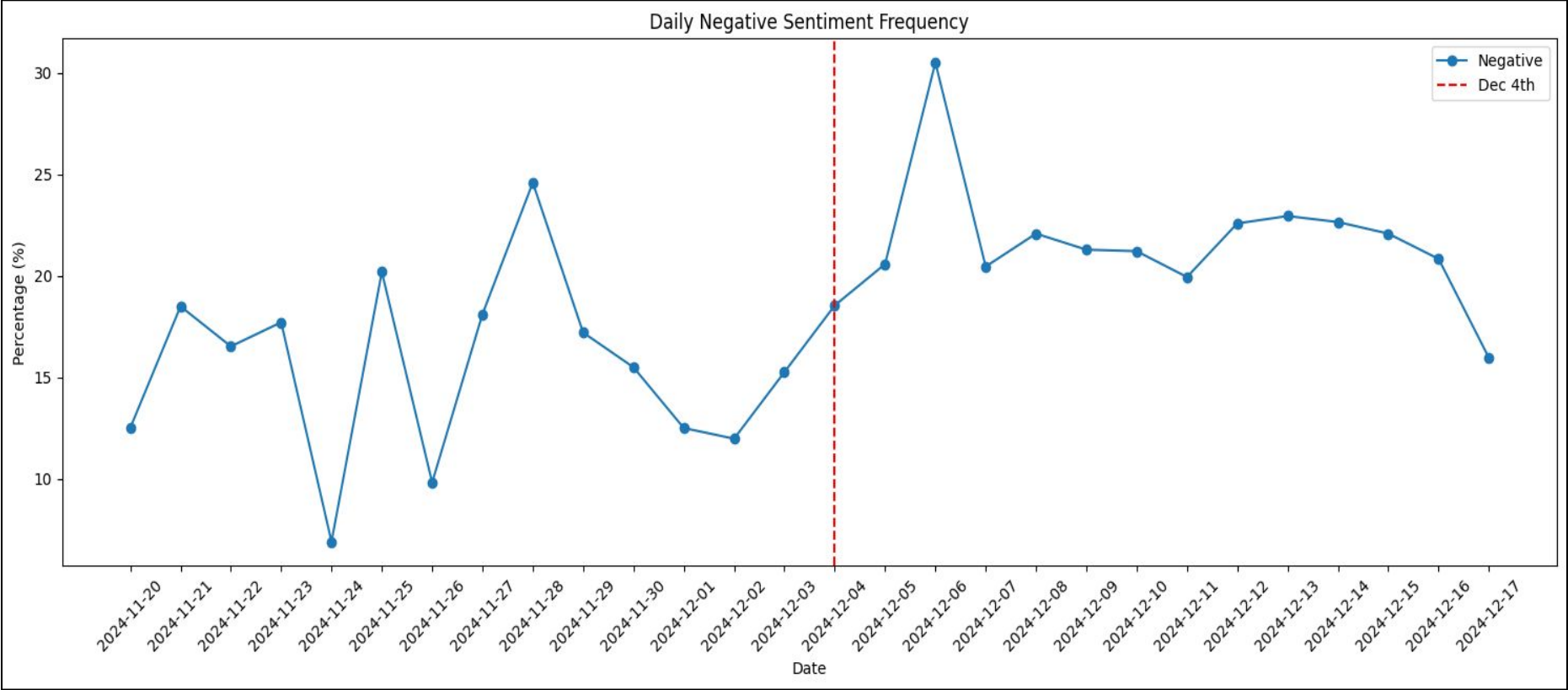
Daily Neutral Sentiment Category Frequency (%)



Daily Positive Sentiment Category Frequency (%)



Daily Negative Sentiment Category Frequency (%)



Linear Regression as a Quasi-T-Test Approach

OLS Regression Results

```
=====
Dep. Variable:          frequency    R-squared:                0.391
Model:                  OLS          Adj. R-squared:           0.367
Method:                 Least Squares  F-statistic:              16.66
Date:                  Mon, 10 Feb 2025  Prob (F-statistic):       0.000378
Time:                  20:01:37       Log-Likelihood:           -76.850
No. Observations:      28            AIC:                     157.7
Df Residuals:          26            BIC:                     160.4
Df Model:               1
Covariance Type:       nonrobust
=====
```

```
=====
              coef      std err          t      P>|t|      [0.025      0.975]
-----
const         15.5180         1.044      14.861      0.000      13.372      17.664
dummy          6.0276         1.477       4.082      0.000         2.992         9.063
=====
```

```
=====
Omnibus:                 2.824    Durbin-Watson:           2.217
Prob(Omnibus):            0.244    Jarque-Bera (JB):         1.431
Skew:                     0.335    Prob(JB):                 0.489
Kurtosis:                 3.882    Cond. No.                  2.62
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Feedbacks from Jonathan and Benny:

- It would be better to retain punctuation before labeling, as stripping it too early might impact the accuracy of the labels. All preprocessing steps should be handled separately during the preprocessing stage.
- You are making excellent progress and demonstrating a clear understanding of the project objectives. Your clarity and foresight in approaching the analysis are commendable. (“You guys are so nice.” - Us)