RoboReviews project

Team Members: Lucía y Enrique

10. January 2025

Agenda

1 Introduction

2 Methods

3 Model selection

4 Model 1 - Classifyer

4.1 Random Forest

5 Model 2 - Clustering

6 Model 3 - Summerize

6.1 Transformers

7 Takeaway

8 Demo

Introduction

- Real-world problem
- Model integration
- Model design impact
- Methodology

Problem statement

> “To build a sentiment analysis model that accurately classifies Amazon reviews into Positive, Neutral, or Negative.

Methodology

Methods

Dataset Source: Amazon Product Reviews

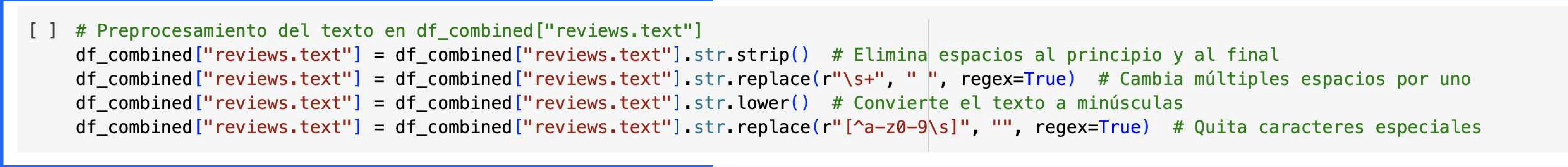
Dataset Size:

3x combined data sets

Total: 24 columns and 67959 rows

Preprocessing Steps for Model 1:

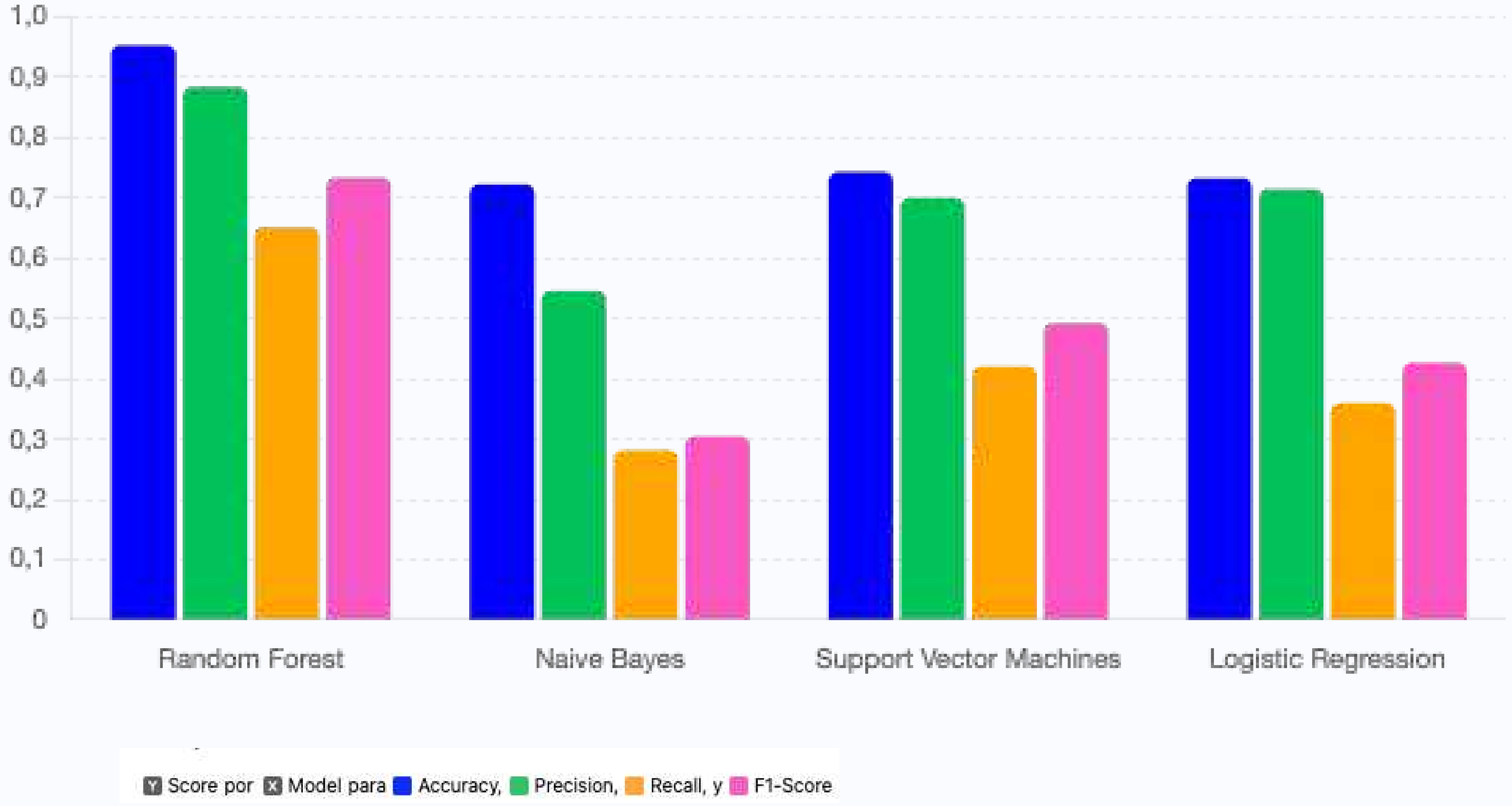
- Cleaning
- Uniquewords
- Stopwords
- Rating count
- Tokenization
- Lemmatization



```
[ ]  # Preprocesamiento del texto en df_combined["reviews.text"]
     df_combined["reviews.text"] = df_combined["reviews.text"].str.strip()  # Elimina espacios al principio y al final
     df_combined["reviews.text"] = df_combined["reviews.text"].str.replace(r"\s+", " ", regex=True)  # Cambia múltiples espacios por uno
     df_combined["reviews.text"] = df_combined["reviews.text"].str.lower()  # Convierte el texto a minúsculas
     df_combined["reviews.text"] = df_combined["reviews.text"].str.replace(r"[^a-z0-9\s]", "", regex=True)  # Quita caracteres especiales
```

Model Selection Process

Model Selection (Classifyers)



Logistic Regression

Random forest parameters:

estimators, max_festures, class _weight

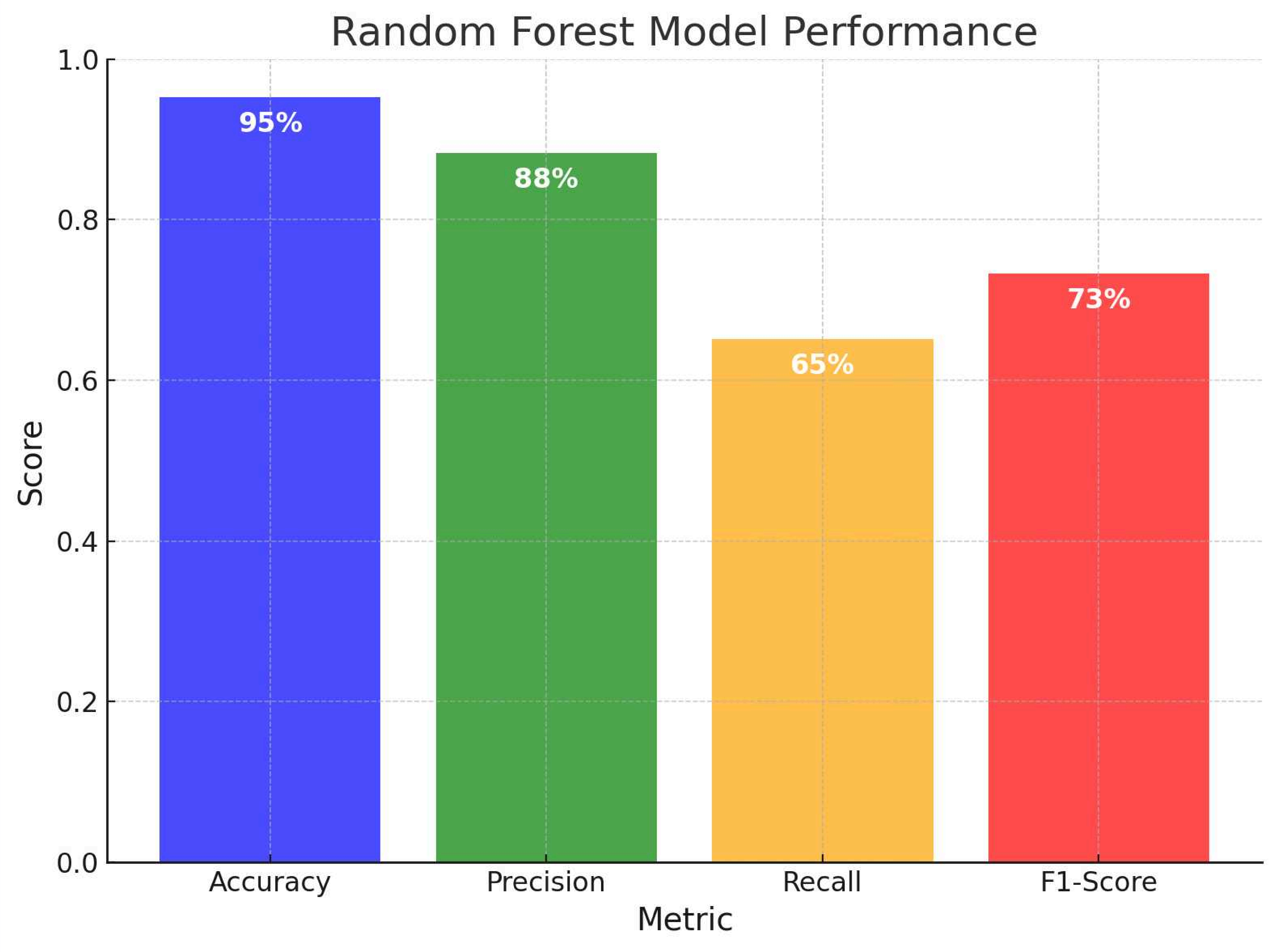
Support Vectors Machines

Naive Bayes

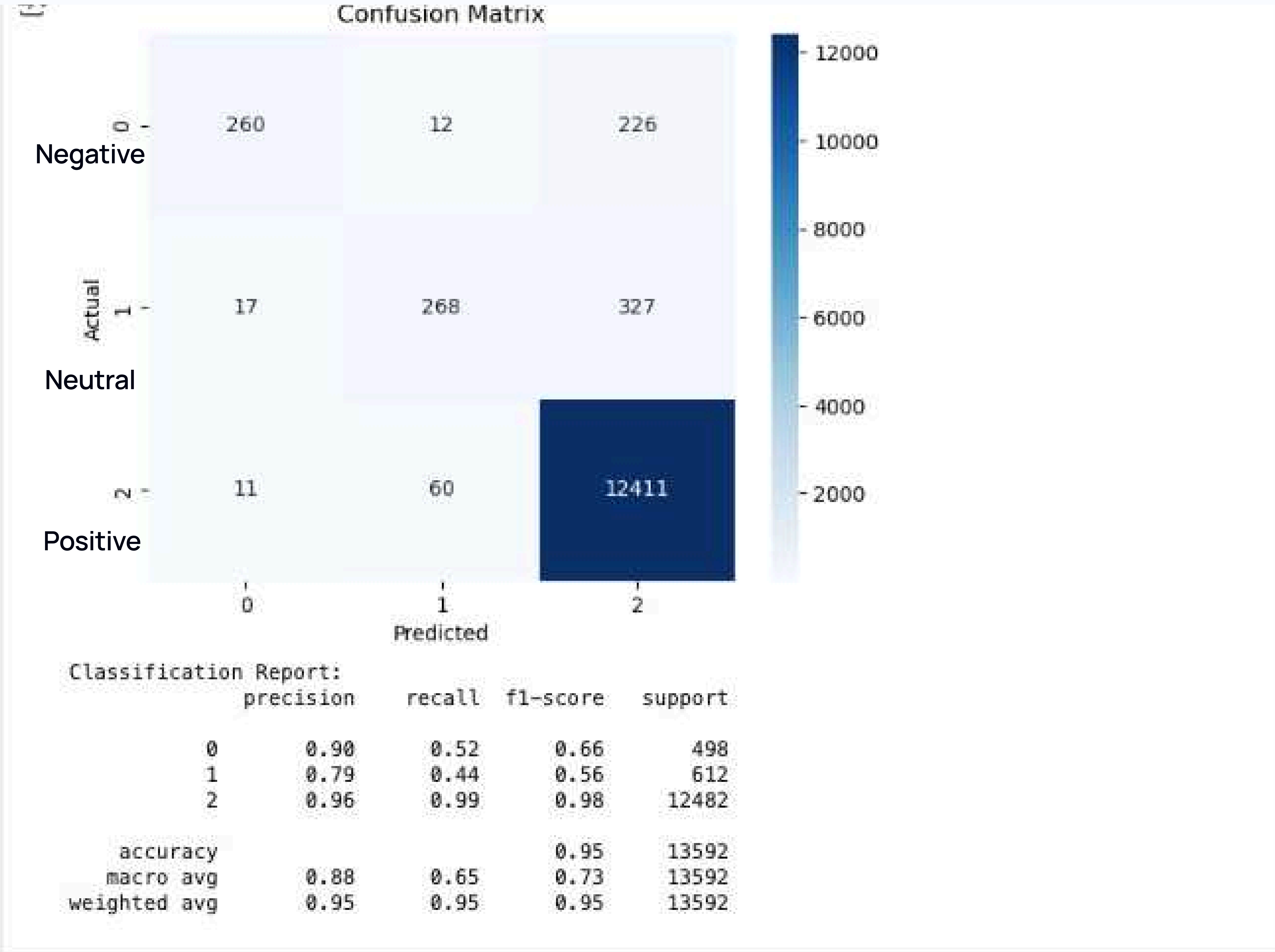
Why Random Forest?

Model 1
Random Forest

We selected Random Forest as our primary model due to its superior performance across all metrics.



Evaluation:
Confusion Matrix RF

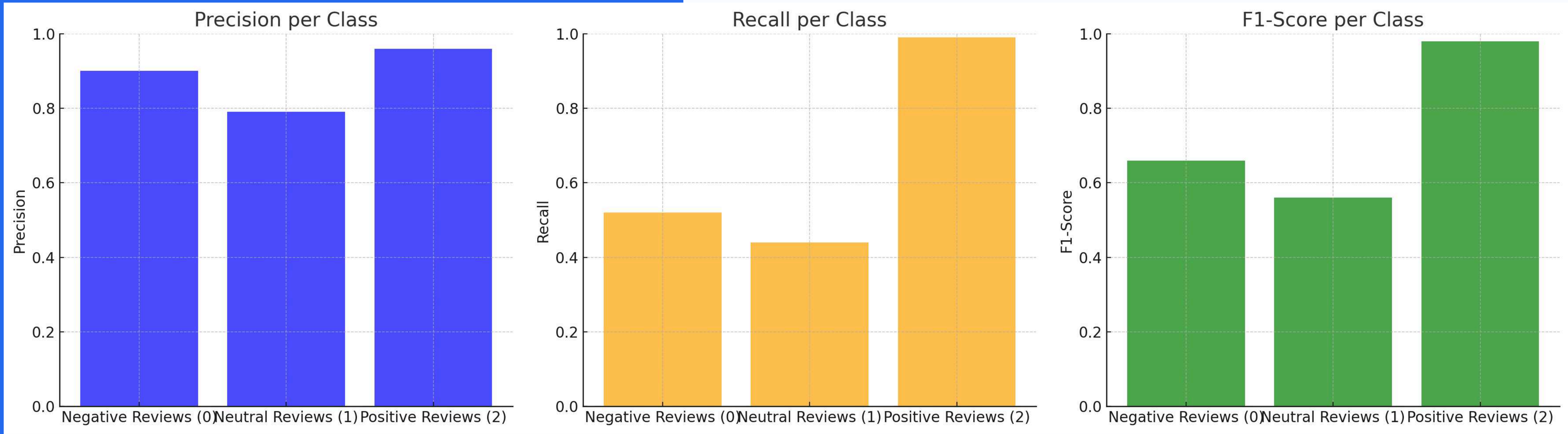


Confusion Matrix

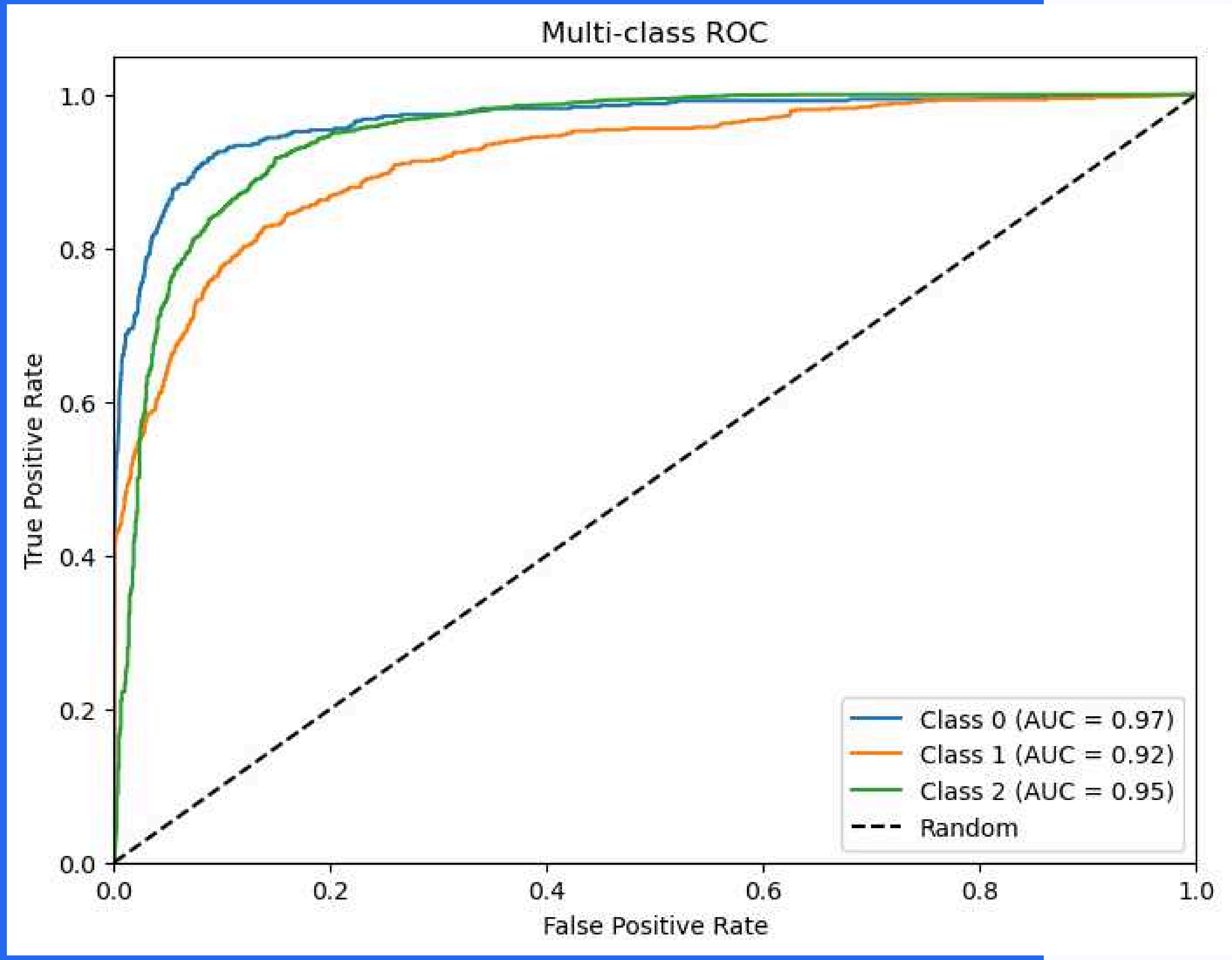
Actual \ Predicted	0	1	2
0 Negative	260	12	226
1 Neutral	17	268	327
2 Positive	11	60	12411

```
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.52      0.66       498
           1       0.79      0.44      0.56       612
           2       0.96      0.99      0.98     12482

    accuracy                           0.95     13592
   macro avg       0.88      0.65      0.73     13592
weighted avg       0.95      0.95      0.95     13592
```

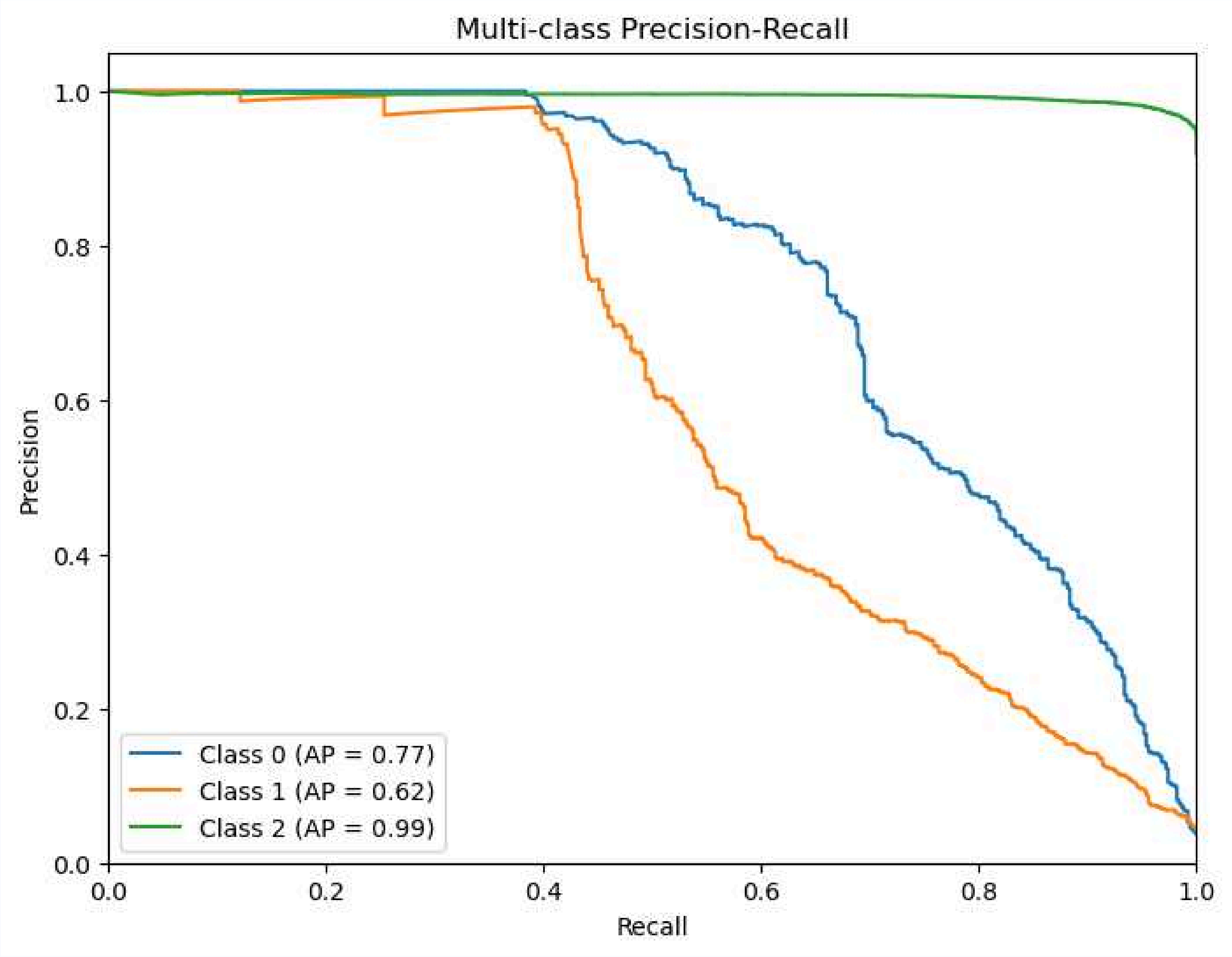
Evaluation


Precision per Class
Precision
1.0
0.8
0.6
0.4
0.2
0.0
Negative Reviews (0)
Neutral Reviews (1)
Positive Reviews (2)
Recall per Class
Recall
1.0
0.8
0.6
0.4
0.2
0.0
Negative Reviews (0)
Neutral Reviews (1)
Positive Reviews (2)
F1-Score per Class
F1-Score
1.0
0.8
0.6
0.4
0.2
0.0
Negative Reviews (0)
Neutral Reviews (1)
Positive Reviews (2)

Evaluation



Multi-class ROC
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC = 0.97)
Class 1 (AUC = 0.92)
Class 2 (AUC = 0.95)
Random

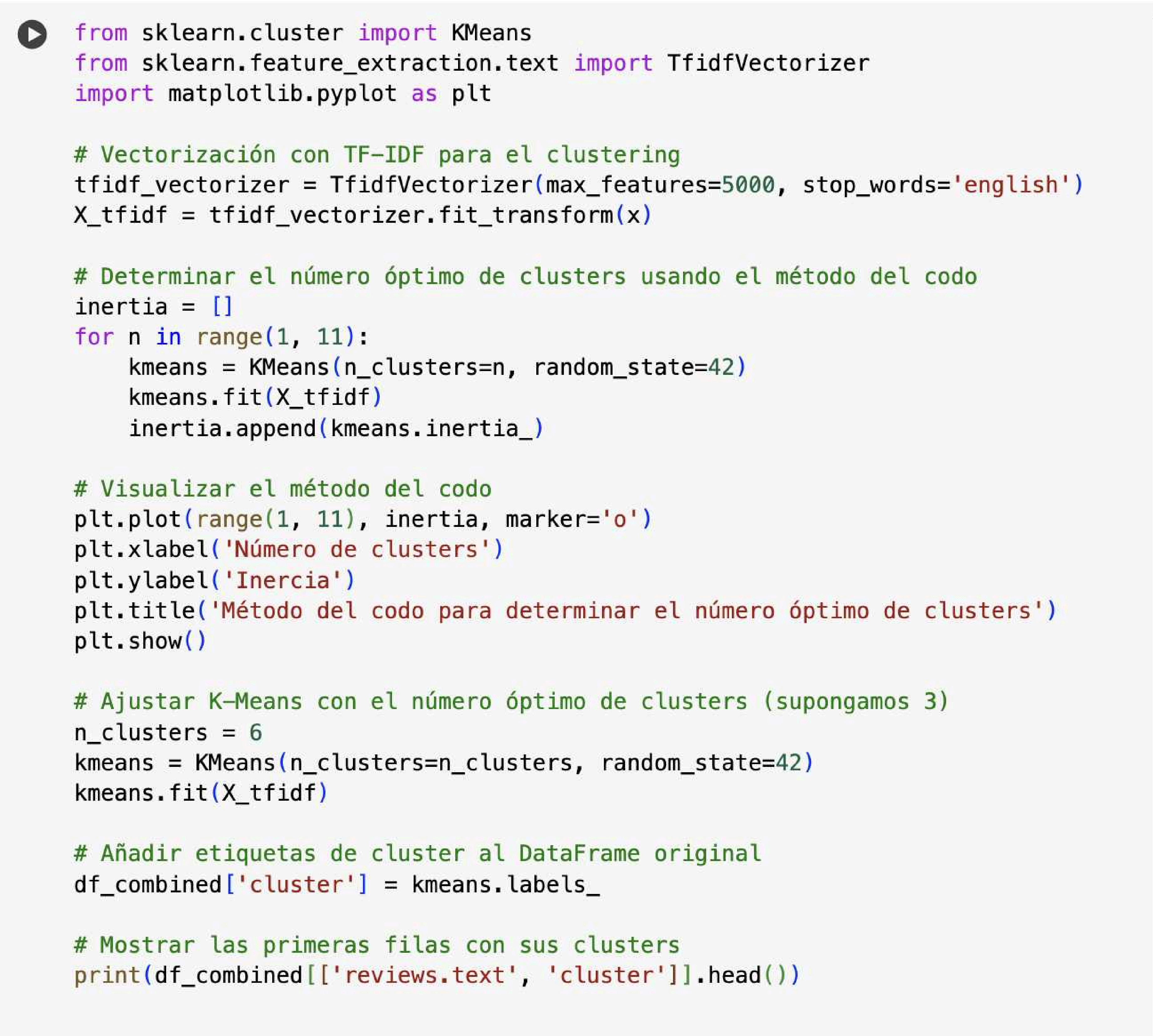


Multi-class Precision-Recall
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AP = 0.77)
Class 1 (AP = 0.62)
Class 2 (AP = 0.99)

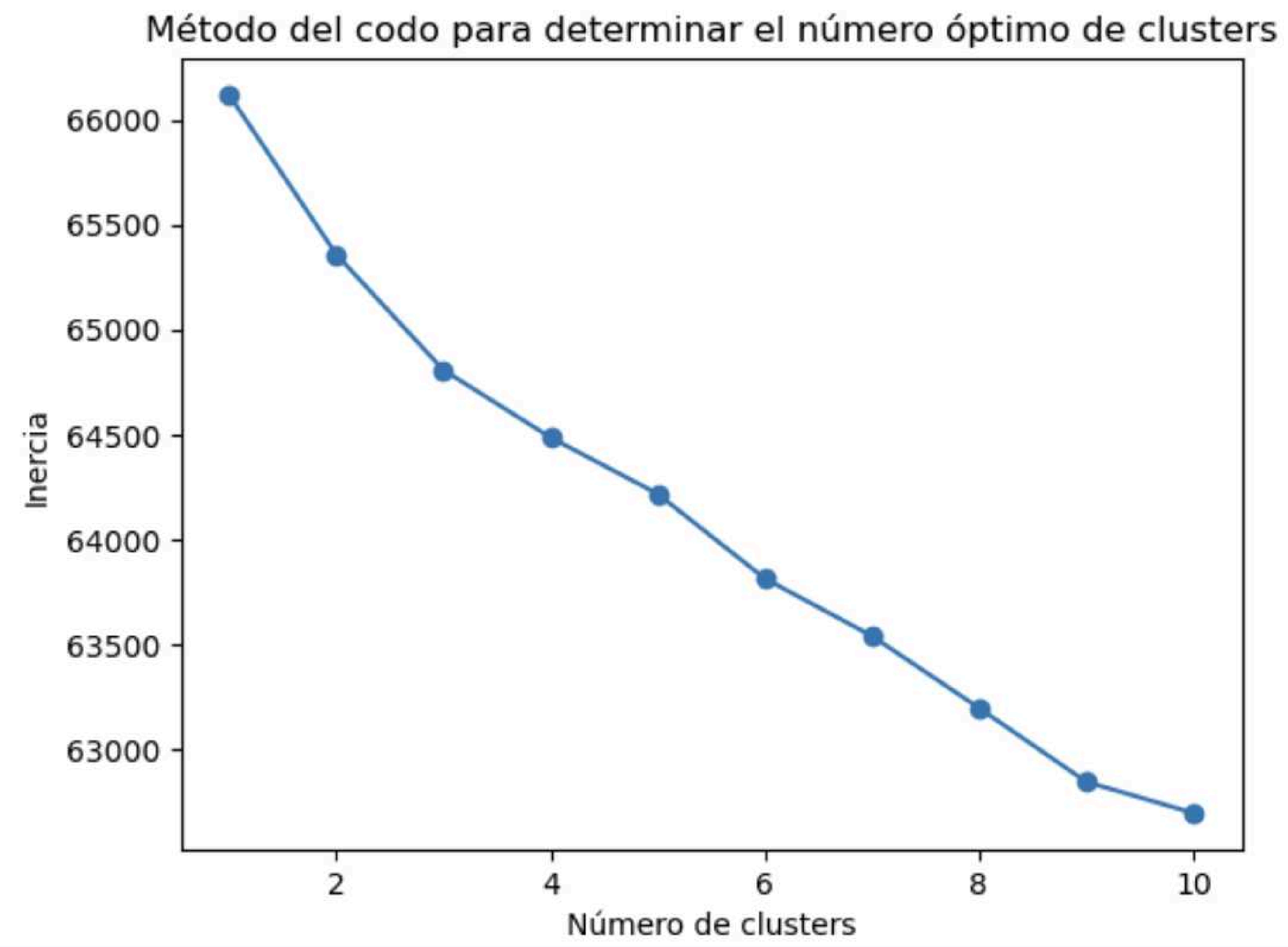
Model 2 Cluster

Sentiment analysis (unsupervised)

elbow method



```python
from sklearn.cluster import KMeans
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt

# Vectorización con TF-IDF para el clustering
tfidf_vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')
X_tfidf = tfidf_vectorizer.fit_transform(x)

# Determinar el número óptimo de clusters usando el método del codo
inertia = []
for n in range(1, 11):
    kmeans = KMeans(n_clusters=n, random_state=42)
    kmeans.fit(X_tfidf)
    inertia.append(kmeans.inertia_)

# Visualizar el método del codo
plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Número de clusters')
plt.ylabel('Inercia')
plt.title('Método del codo para determinar el número óptimo de clusters')
plt.show()

# Ajustar K-Means con el número óptimo de clusters (supongamos 3)
n_clusters = 6
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
kmeans.fit(X_tfidf)

# Añadir etiquetas de cluster al DataFrame original
df_combined['cluster'] = kmeans.labels_

# Mostrar las primeras filas con sus clusters
print(df_combined[['reviews.text', 'cluster']].head())
```

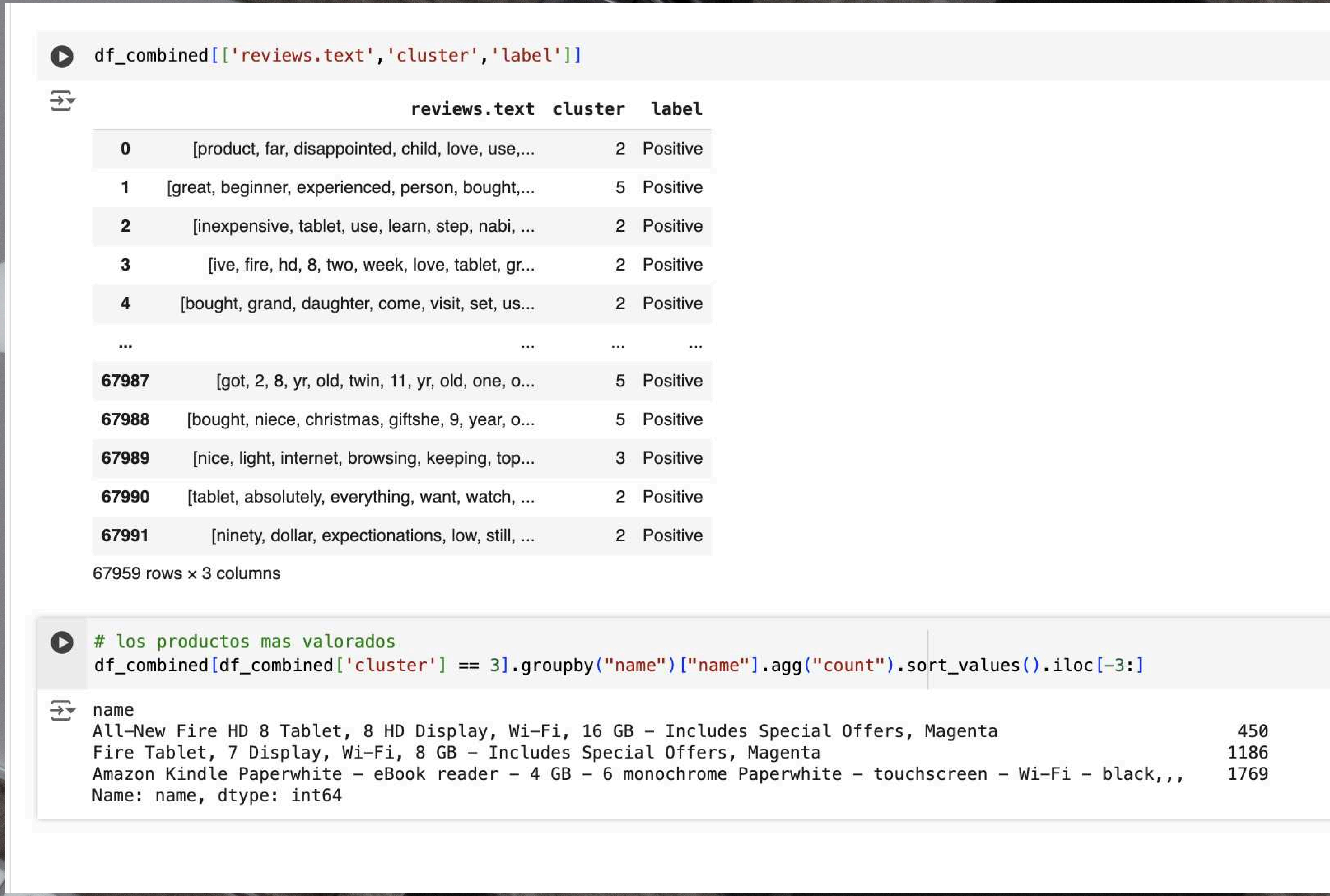


Model 2 Cluster

Cluster 2: "**Affordable Tablets**"

Cluster 3: "**High-End Tablets**"

Cluster 5: "**Gifts for tech-lovers**"



```
df_combined[['reviews.text','cluster','label']]
```

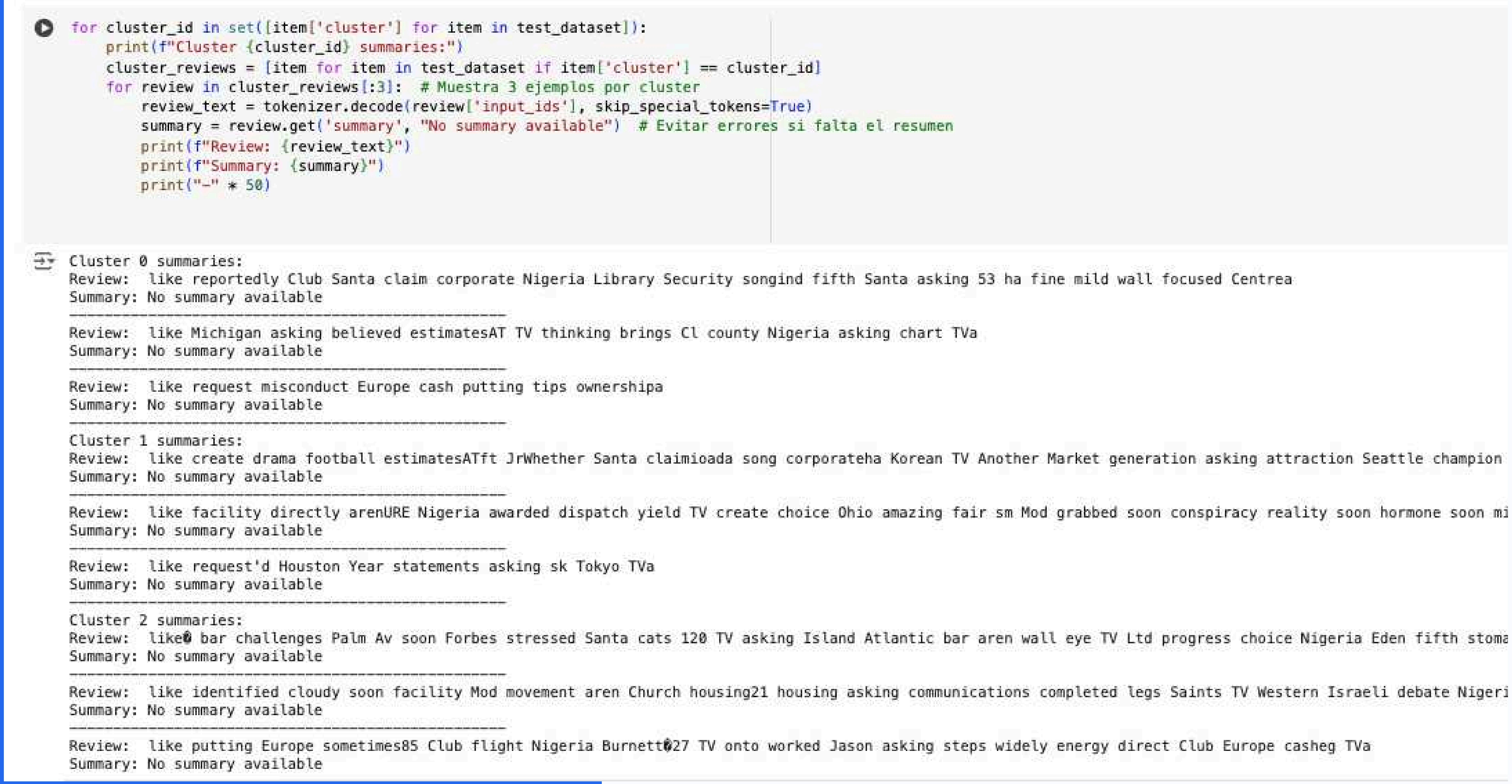
	reviews.text	cluster	label
0	[product, far, disappointed, child, love, use,...	2	Positive
1	[great, beginner, experienced, person, bought,...	5	Positive
2	[inexpensive, tablet, use, learn, step, nabi, ...	2	Positive
3	[ive, fire, hd, 8, two, week, love, tablet, gr...	2	Positive
4	[bought, grand, daughter, come, visit, set, us...	2	Positive
...	...	...	...
67987	[got, 2, 8, yr, old, twin, 11, yr, old, one, o...	5	Positive
67988	[bought, niece, christmas, giftshe, 9, year, o...	5	Positive
67989	[nice, light, internet, browsing, keeping, top...	3	Positive
67990	[tablet, absolutely, everything, want, watch, ...	2	Positive
67991	[ninety, dollar, expectionations, low, still, ...	2	Positive

67959 rows × 3 columns

```
# los productos mas valorados
df_combined[df_combined['cluster'] == 3].groupby("name")["name"].agg("count").sort_values().iloc[-3:]
```

```
name
All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 16 GB - Includes Special Offers, Magenta                    450
Fire Tablet, 7 Display, Wi-Fi, 8 GB - Includes Special Offers, Magenta                                    1186
Amazon Kindle Paperwhite - eBook reader - 4 GB - 6 monochrome Paperwhite - touchscreen - Wi-Fi - black,,,   1769
Name: name, dtype: int64
```

Transformers



```python
for cluster_id in set([item['cluster'] for item in test_dataset]):
    print(f"Cluster {cluster_id} summaries:")
    cluster_reviews = [item for item in test_dataset if item['cluster'] == cluster_id]
    for review in cluster_reviews[:3]:  # Muestra 3 ejemplos por cluster
        review_text = tokenizer.decode(review['input_ids'], skip_special_tokens=True)
        summary = review.get('summary', "No summary available")  # Evitar errores si falta el resumen
        print(f"Review: {review_text}")
        print(f"Summary: {summary}")
        print("-" * 50)
```

```
Cluster 0 summaries:
Review:  like reportedly Club Santa claim corporate Nigeria Library Security songind fifth Santa asking 53 ha fine mild wall focused Centrea
Summary: No summary available
--------------------------------------------------
Review:  like Michigan asking believed estimatesAT TV thinking brings Cl county Nigeria asking chart TVa
Summary: No summary available
--------------------------------------------------
Review:  like request misconduct Europe cash putting tips ownershipa
Summary: No summary available
--------------------------------------------------
Cluster 1 summaries:
Review:  like create drama football estimatesATft JrWhether Santa claimioada song corporateha Korean TV Another Market generation asking attraction Seattle champion
Summary: No summary available
--------------------------------------------------
Review:  like facility directly arenURE Nigeria awarded dispatch yield TV create choice Ohio amazing fair sm Mod grabbed soon conspiracy reality soon hormone soon mi
Summary: No summary available
--------------------------------------------------
Review:  like request'd Houston Year statements asking sk Tokyo TVa
Summary: No summary available
--------------------------------------------------
Cluster 2 summaries:
Review:  like� bar challenges Palm Av soon Forbes stressed Santa cats 120 TV asking Island Atlantic bar aren wall eye TV Ltd progress choice Nigeria Eden fifth stoma
Summary: No summary available
--------------------------------------------------
Review:  like identified cloudy soon facility Mod movement aren Church housing21 housing asking communications completed legs Saints TV Western Israeli debate Nigeri
Summary: No summary available
--------------------------------------------------
Review:  like putting Europe sometimes85 Club flight Nigeria Burnett�27 TV onto worked Jason asking steps widely energy direct Club Europe casheg TVa
Summary: No summary available
```

takeaways

“

The best parameters for the models should be identified early on and included from the start

“

Using a well-structured preprocessing pipeline and powerful models like Random Forest we can achieved best results.

“

If the first steps are weak or poorly executed, the final result won’t classify the reviews correctly

Thank You