

ĐỒ ÁN MÔN MÁY HỌC

Nhận diện mức độ cảm xúc của khách hàng

Giảng viên hướng dẫn: TS.Nguyễn Văn Dũ

Thành viên nhóm 13



Thành viên: Hồ Thanh Hoài An - 20130193

Nguyễn Thị Kim Anh - 20130197

Phân công công việc

Hồ Thanh Hoài An	Nguyễn Thị Kim Anh
<ul style="list-style-type: none">- Crawl dữ liệu- Mô tả dữ liệu- Vector hóa dữ liệu- Xử lý imbalanced data- Xây dựng model KNN, Neural Network, SVM Linear, SVM Sigmoid	<ul style="list-style-type: none">- Gán nhãn dữ liệu- Chuẩn hóa dữ liệu số- Xây dựng model SVM RBF, Randomforest, Naïve Bayes, chạy GridSearchCV cho Randomforest

I. Mô tả dữ liệu

II. Tiền xử lý dữ liệu

III. Xây dựng model và
đánh giá

IV. Kết luận

I. Mô tả dữ liệu

1. Giới thiệu

- Tiki là một trang thương mại điện tử lớn, đối với người mua hàng thì mục đích chính là nơi để người dùng xác định chất lượng sản phẩm để đưa ra quyết định mua hàng. Việc nhận diện cảm xúc của khách hàng có thể giúp cho nhà sản xuất hoặc người bán hiểu được sự hài lòng hoặc không hài lòng của khách hàng và đưa ra các biện pháp giải quyết.

- Bộ dữ liệu dùng để xây dựng model là bộ dữ liệu về review sản phẩm trên tiki được thu thập thông qua api của tiki
- Bộ dữ liệu bao gồm 6612 mẫu, 4 đặc trưng lần lượt là:
 - + content: đây là đặc trưng chứa thông tin chi tiết và đánh giá của khách hàng về sản phẩm
 - + thank_count: số đánh giá hữu ích của một comment trong đánh giá sản phẩm. Có thể hiểu đơn giản đây là số lượng like của một review
 - + comment_count: số lượng phản hồi đánh giá của một review
 - + rating: số sao đánh giá sản phẩm khi review của khách hàng

2. Thăm dò dữ liệu

Khi nhìn sơ qua bộ dữ liệu thì ta thấy ở content có giá trị NaN và bộ dữ liệu này chưa được gán nhãn vậy ta phải thực hiện gán nhãn để có thể xây dựng model

```
data = pd.read_excel('review.xlsx')  
data.head()
```




	content	thank_count	comment_count	rating
0	Samsung chất lượng ổn định. nhờ đơn vị bán hàn...	0	0	5
1	NaN	0	0	5
2	Sản phẩm nguyên seal , đóng gói cẩn thận	1	0	5
3	Sp tốt	0	0	5
4	Quá nhanh luôn	0	0	5




Từ hàm info ta có thể có những đánh giá tổng quan về dữ liệu như sau:

- Trong 4 đặc trưng thì có 3 đặc trưng là dữ liệu số và content là dữ liệu object (ở đây sẽ là text), vậy ta sẽ phải xử lý vector hóa chữ thành số để có thể xây dựng model.
- Nhìn qua thì ta thấy dữ liệu của content chỉ có 2988 so với các đặc trưng khác là 6612, vậy ta phải xử lý vấn đề thiếu dữ liệu cho content.

Từ bộ dữ liệu ban đầu có thể thấy có 1 trường trong content có giá trị là NaN. Ta sẽ xem thử số giá trị còn thiếu của content


 data.info()


 <class 'pandas.core.frame.DataFrame'>
RangeIndex: 6612 entries, 0 to 6611
Data columns (total 4 columns):

#	Column	Non-Null Count	Dtype
0	content	2988 non-null	object
1	thank_count	6612 non-null	int64
2	comment_count	6612 non-null	int64
3	rating	6612 non-null	int64


dtypes: int64(3), object(1)
memory usage: 206.8+ KB


✓
0s

 data.isna().sum()

 content 3624
thank_count 0
comment_count 0
rating 0
dtype: int64

- Ta có thể đánh giá tổng quan dữ liệu số thông qua describe() như sau:
- **thank_count**
 - + Dữ liệu biến động mạnh thông qua std cao 3.43 và max là 130
 - + Phần lớn các review (75%) không có lượt hữu ích giá trị bằng 0.
 - **comment_count**
 - + Dữ liệu cũng có sự biến động, nhưng ít thông qua std 0.7 và max là 21
 - + Phần lớn các review (75%) không có bất kỳ bình luận nào giá trị bằng 0
 - **rating**
 - + Dữ liệu phân phối gần như tương đối tập trung và gần với giá trị mean là 4.77 và std 0.69 cho biết phân tán không quá lớn
 - + Đánh giá trung bình của các review là khá cao (gần 5), cho thấy hầu hết người dùng đều đánh giá sản phẩm cao.

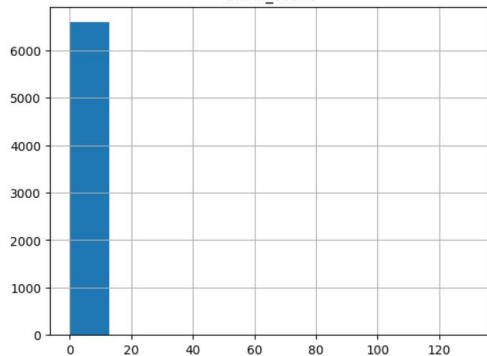
 data.describe()



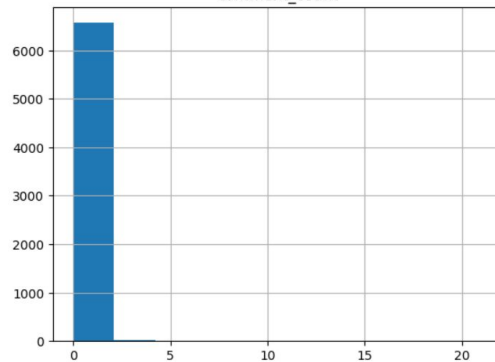
	thank_count	comment_count	rating
count	6612.000000	6612.000000	6612.000000
mean	0.340139	0.080157	4.771022
std	3.434545	0.700446	0.690420
min	0.000000	0.000000	1.000000
25%	0.000000	0.000000	5.000000
50%	0.000000	0.000000	5.000000
75%	0.000000	0.000000	5.000000
max	130.000000	21.000000	5.000000



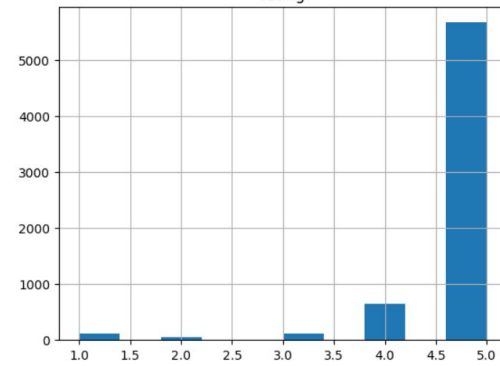
thank_count



comment_count



rating

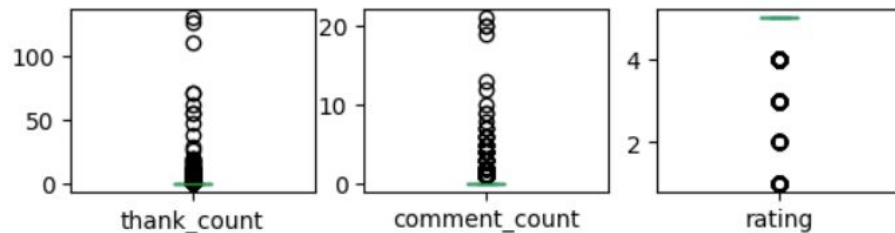


Qua biểu đồ histogram của các đặc trưng có thể thấy việc đánh giá tổng quát phía trên tương đối đúng tất cả đặc trưng định lượng đều có phân bố bất thường, không đồng dạng và không cân đối. Vậy với dữ liệu như thế này chúng ta cần chuẩn hóa dữ liệu để đạt được phân phối gần phân phối chuẩn.

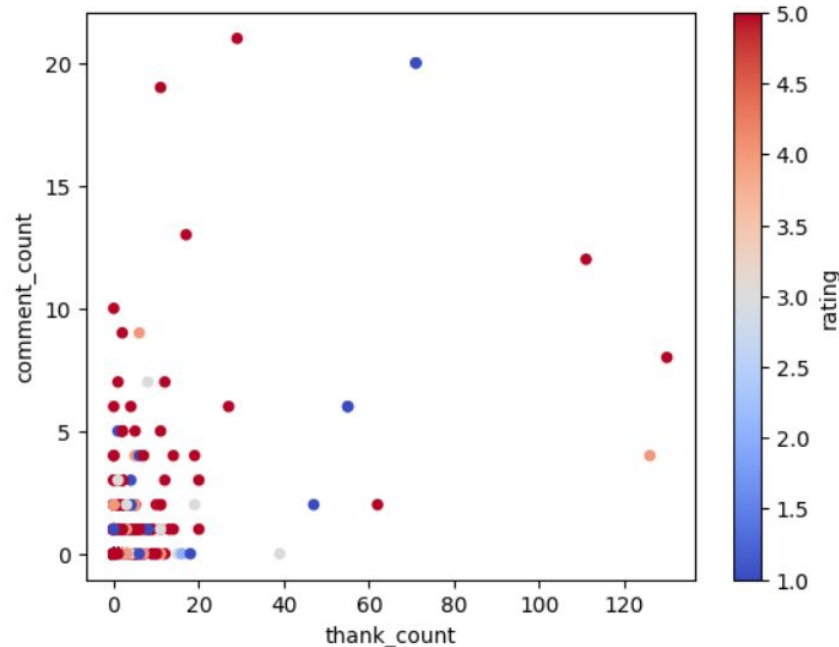
Sự phân phối bất thường này chủ yếu là phân phối không cân bằng và phân phối bị lệch, có thể xem xét đến việc sự xuất hiện của outlier để xử lý.

Qua biểu đồ ta có thể thấy hình dạng hộp hộp kì dị so với bình thường, chứng tỏ có các điểm ngoại lai khiến cho hộp nằm lệch về một phía

dtype: object



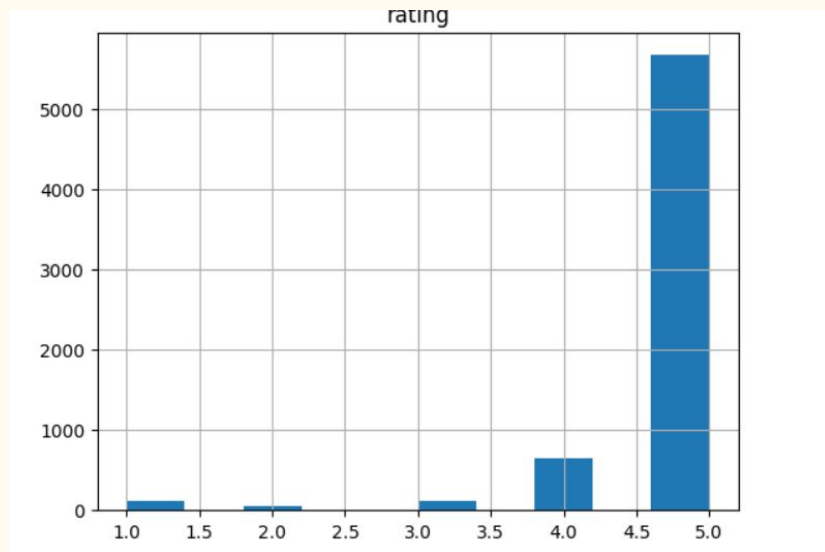
Từ biểu đồ phân tán bên ta có thể thấy là có sự xuất hiện của các outlier với giá trị đáng ngờ so với phân bố dữ liệu.



Từ ma trận tương quan bên ta có thể thấy 2 đặc trưng `thank_count` và `comment_count` có mức độ tương quan cũng khá cao là 0.55 nhưng có vẻ sẽ không ảnh hưởng nhiều. Tuy nhiên vẫn cần xem xét khi xây dựng model có score thấp.

	<code>thank_count</code>	<code>comment_count</code>	<code>rating</code>
<code>thank_count</code>	1.00	0.55	-0.12
<code>comment_count</code>	0.55	1.00	-0.11
<code>rating</code>	-0.12	-0.11	1.00

Ta thấy việc cảm xúc khách hàng liên quan mức thiết đến rating nên ta có thể dùng rating để gán nhãn dữ liệu và khi làm điều đó việc phân bố không đồng đều của các giá trị trong dữ liệu có thể gây nên việc imbalanced data. Ta cần chú ý và xử lý sau khi gán nhãn.



II. Tiền xử lý dữ liệu

—

Những vấn đề cần xử lý

Sau khi quan sát và đánh giá dữ liệu ở trên thì những vấn đề chúng ta cần xử lý là:

- + Gán nhãn dữ liệu
- + Vector hóa văn bản cho đặc trưng content
- + Chuẩn hóa dữ liệu số
- + Xử lý outlier
- + Xử lý imbalanced data

Gán nhãn dữ liệu

Ta có thể thấy việc rating có liên quan mật thiết với phân loại cảm xúc khách hàng
rating cao thì cảm xúc của khách hàng là positive, rating thấp thì cảm xúc của khách hàng là negative

Từ đó ta có thể phân ra để gán nhãn như sau

1*, 2*: Negative

3*: Neutral

4*, 5*: Positive

```
# Gán nhãn dữ liệu
def label_data(rating):
    if rating == 1 or rating == 2:
        return 'negative'
    elif rating == 4 or rating == 5:
        return 'positive'
    else:
        return 'neutral'

data['label'] = data['rating'].apply(label_data)

data.head()
```

	content	thank_count	comment_count	rating	label
0	Samsung chất lượng ổn định. nhờ đơn vị bán hàn...	0	0	5	positive
1	NaN	0	0	5	positive
2	Sản phẩm nguyên seal , đóng gói cẩn thận	1	0	5	positive
3	Sp tốt	0	0	5	positive
4	Quá nhanh luôn	0	0	5	positive

Vector hóa dữ liệu cho content

Ta sử dụng doc2vec từ thư viện gensim để thực hiện vector hóa:

- Ta sử dụng TaggedDocument để gắn nhãn ở đây ta tách các từ và mỗi văn bản là tập hợp các từ và được gắn nhãn
- Sau đó xây dựng mô hình với vector_size là 100
- Sau đó xây dựng và huấn luyện để chuyển đổi.

```
from gensim.models import Doc2Vec
from gensim.models.doc2vec import TaggedDocument

text_data = data['content'].fillna('')

# Chuẩn bị dữ liệu đã được gắn nhãn
tagged_data = [TaggedDocument(words=str(text).split(), tags=[str(i)]) for i, text in enumerate(text_data)]

# Khởi tạo mô hình Doc2Vec
model = Doc2Vec(vector_size=100, min_count=2, epochs=40)

# Xây dựng từ vựng
model.build_vocab(tagged_data)

# Huấn luyện mô hình
model.train(tagged_data, total_examples=model.corpus_count, epochs=model.epochs)

# Chuyển đổi văn bản thành vector
document_vectors = [model.infer_vector(str(text).split()) for text in text_data]

[14] len(document_vectors)

6612

[15] df_vectors = pd.DataFrame(document_vectors, columns=[f'vector_{i}' for i in range(100)])
data = pd.concat([ df_vectors, data], axis=1)
```



© 2013 Pearson Education, Inc. or its affiliate(s). All rights reserved. This material is intended solely for the personal use of the individual user and is not to be disseminated broadly.

Xử lý outlier

Từ phân tích trên ta có thể thấy outlier đến từ 2 đặc trưng `thank_count`, `comment_count` có một vài giá trị cao bất thường. Trong việc nhận diện cảm xúc khách hàng có vẻ như outlier của 2 trường này sẽ không ảnh hưởng quá nhiều đến việc nhận diện ta có thể giữ lại các outlier này không cần xử lý.

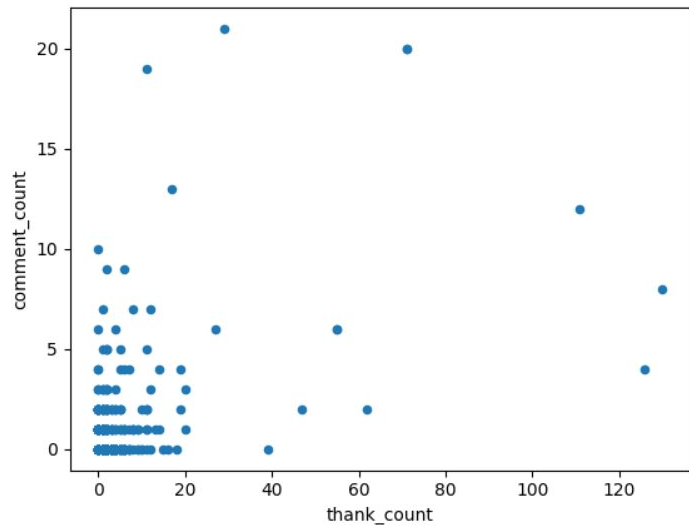
Chuẩn hóa dữ liệu số

Ta sử dụng QuantileTransformer vì có thể chuyển đổi phân phối giúp cho các giá trị sẽ được trải đều hơn trên phạm vi.

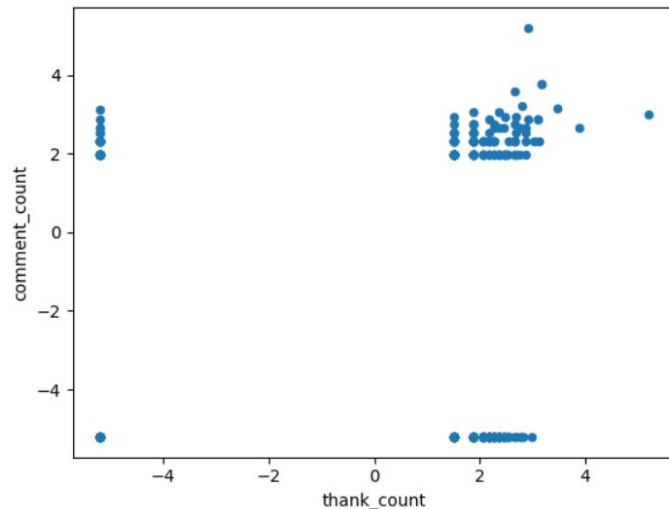


```
from sklearn.preprocessing import QuantileTransformer
transformer = QuantileTransformer(output_distribution='normal')
data['comment_count'] = transformer.fit_transform(data[['comment_count']])
data['thank_count'] = transformer.fit_transform(data[['thank_count']])
```

<Axes: xlabel='thank_count', ylabel='comment_count'>

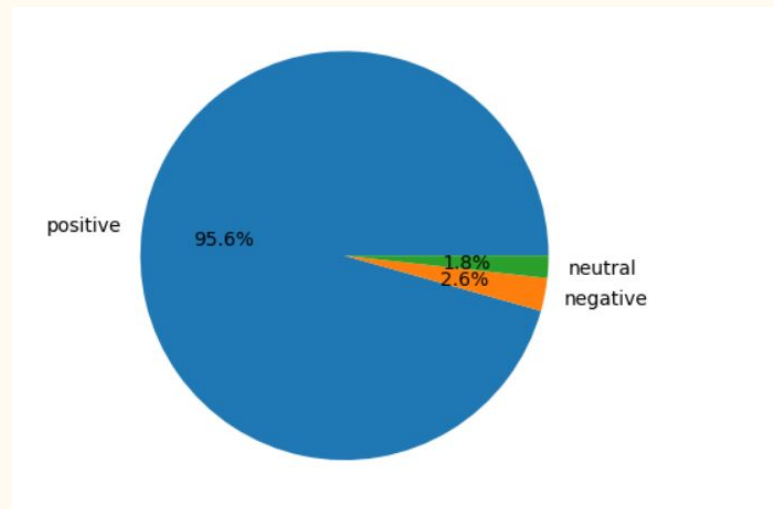
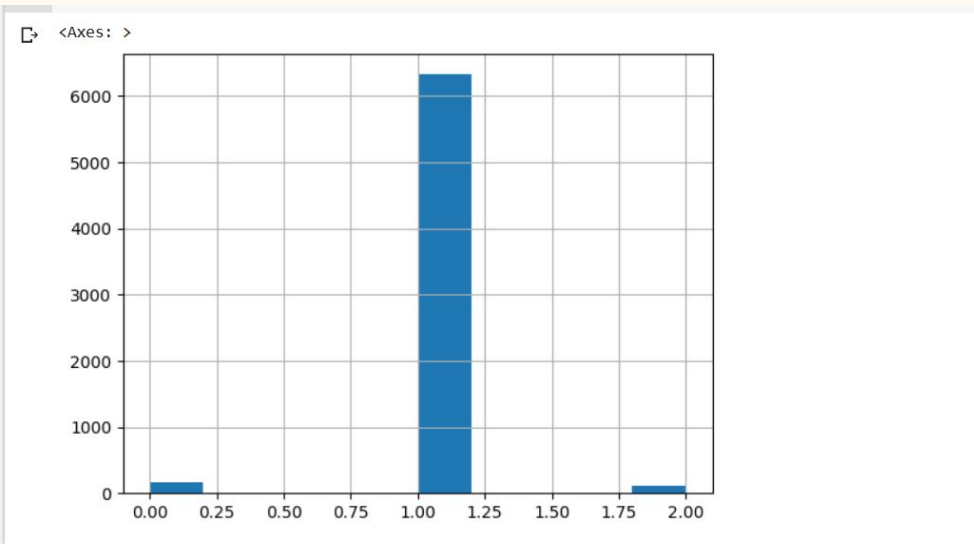


<Axes: xlabel='thank_count', ylabel='comment_count'>



Xử lý imbalanced data

Từ phân tích trước đó thì khi gán nhãn dữ liệu bằng rating sẽ dẫn đến việc imbalanced data



Ta sử dụng ADASYS bởi vì nó là một phương pháp cải tiến của SMOT sẽ giúp tăng các dữ liệu sẽ chú ý tới sự phân bố



```
from imblearn.over_sampling import ADASYN
adasyn = ADASYN(sampling_strategy='not majority', n_neighbors=5)
x_train_new, y_train_new = adasyn.fit_resample(x_train, y_train)
print(y_train_new.value_counts())
print(y_train.value_counts())
```



```
label
negative    5061
neutral     5059
positive    5059
dtype: int64
label
positive    5059
negative     145
neutral       85
dtype: int64
```

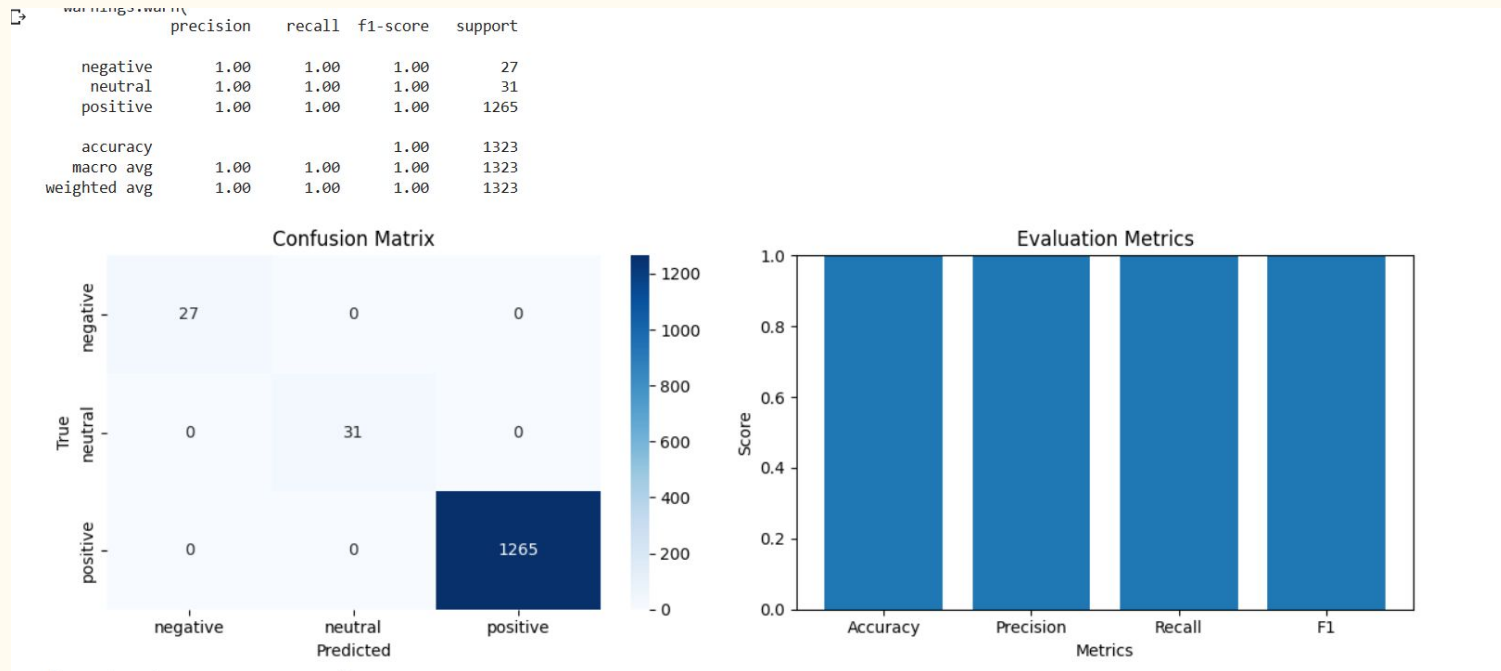
III. Xây dựng model và đánh giá

—

1. SVM

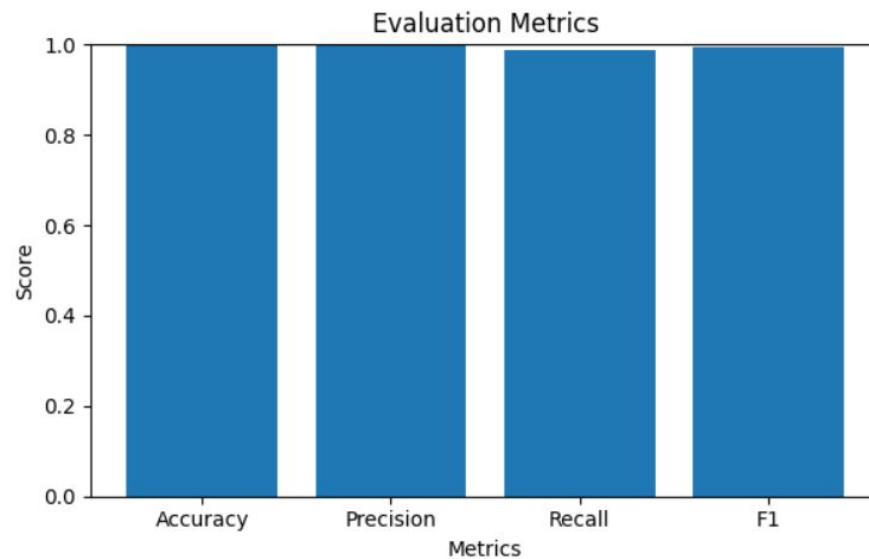
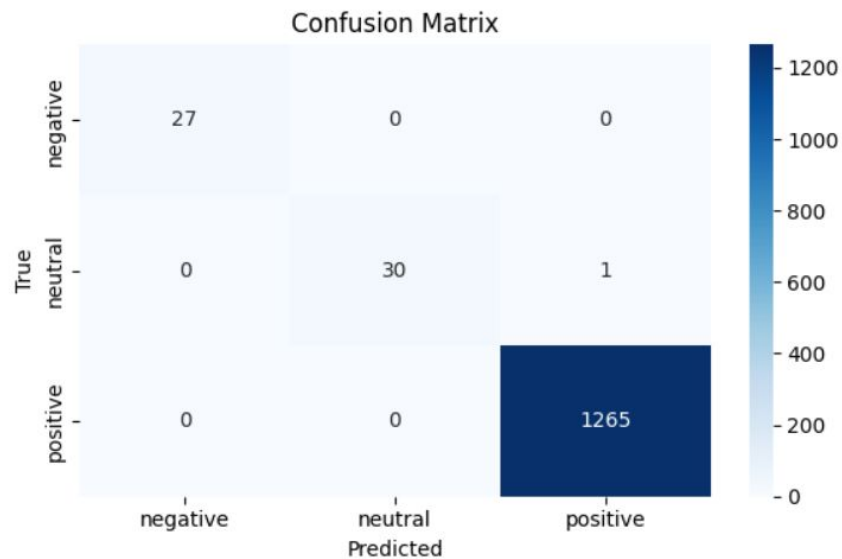
- Với SVM ta thử với ba kernel khác nhau

```
clf = svm.SVC(kernel='linear')
```



```
clf = svm.SVC(kernel='rbf')
```

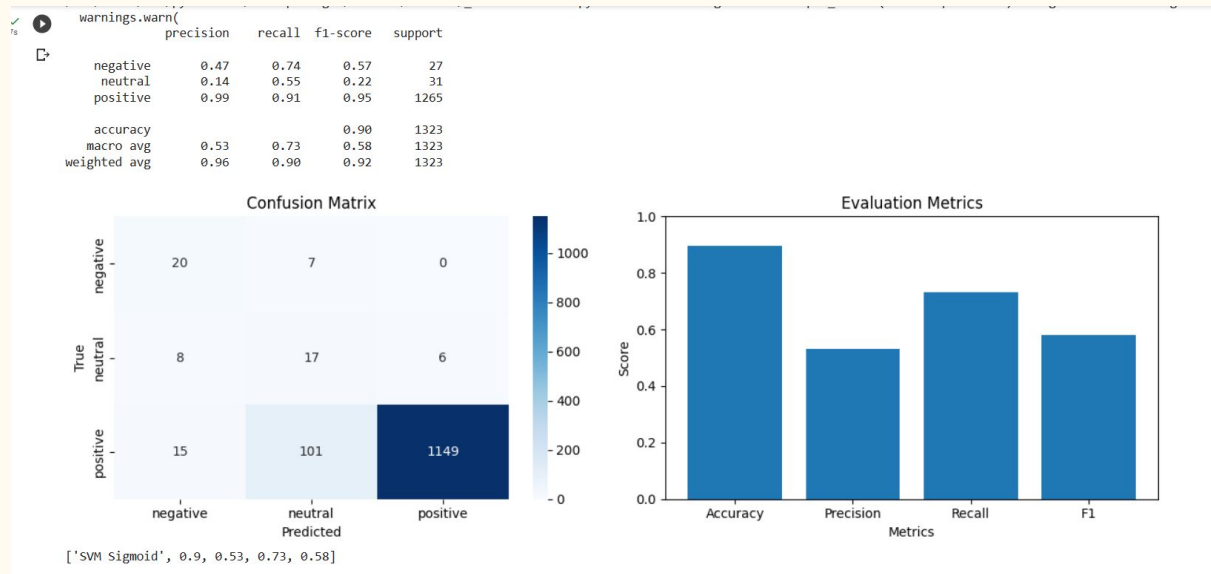
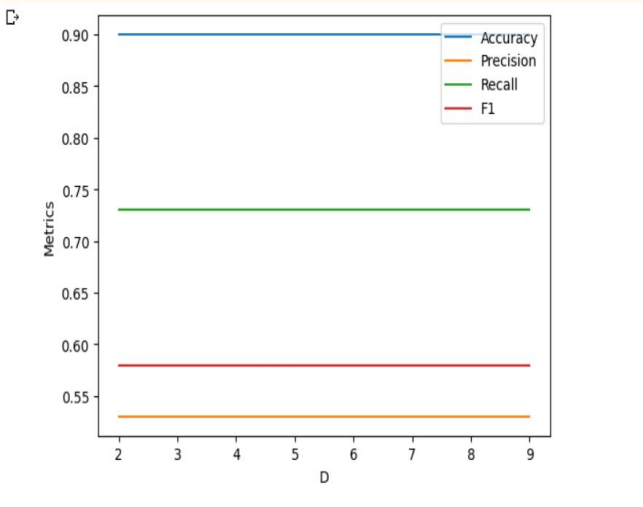
	precision	recall	f1-score	support
negative	1.00	1.00	1.00	27
neutral	1.00	0.97	0.98	31
positive	1.00	1.00	1.00	1265
accuracy			1.00	1323
macro avg	1.00	0.99	0.99	1323
weighted avg	1.00	1.00	1.00	1323



```
['SVM RBF', 1.0, 1.0, 0.99, 0.99]
```

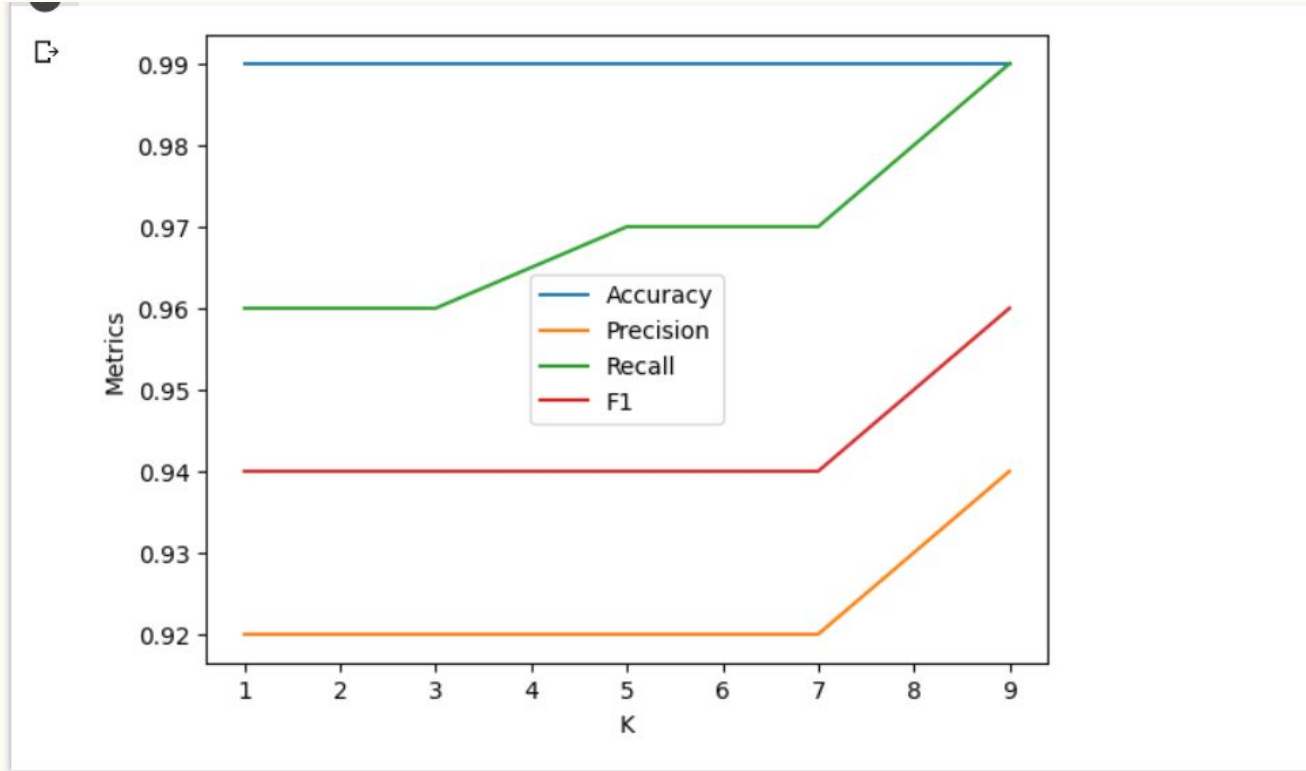
```
clf = svm.SVC(kernel='sigmoid')
```

Với sigmoid, ta sẽ với với nhiều degree khác nhau (2,10)

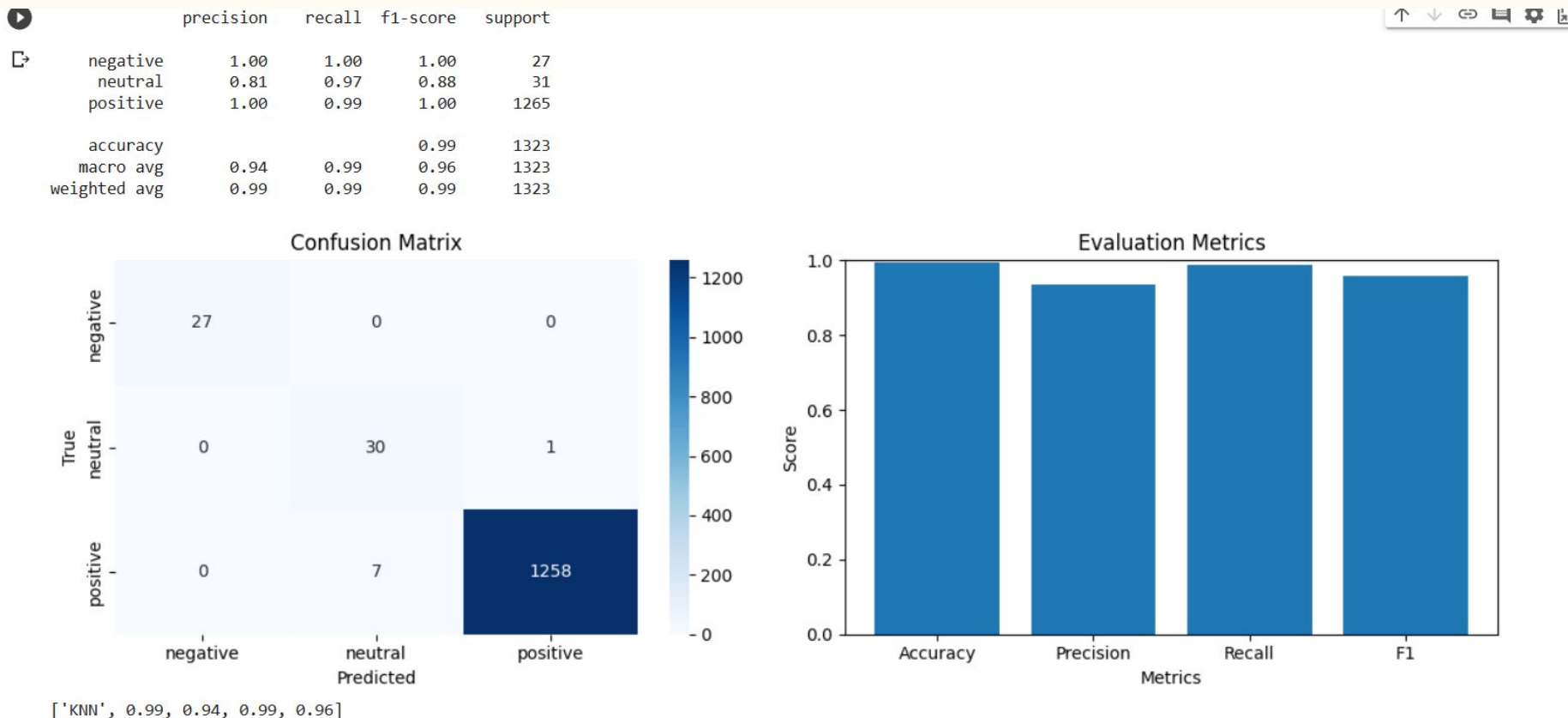


2. KNN

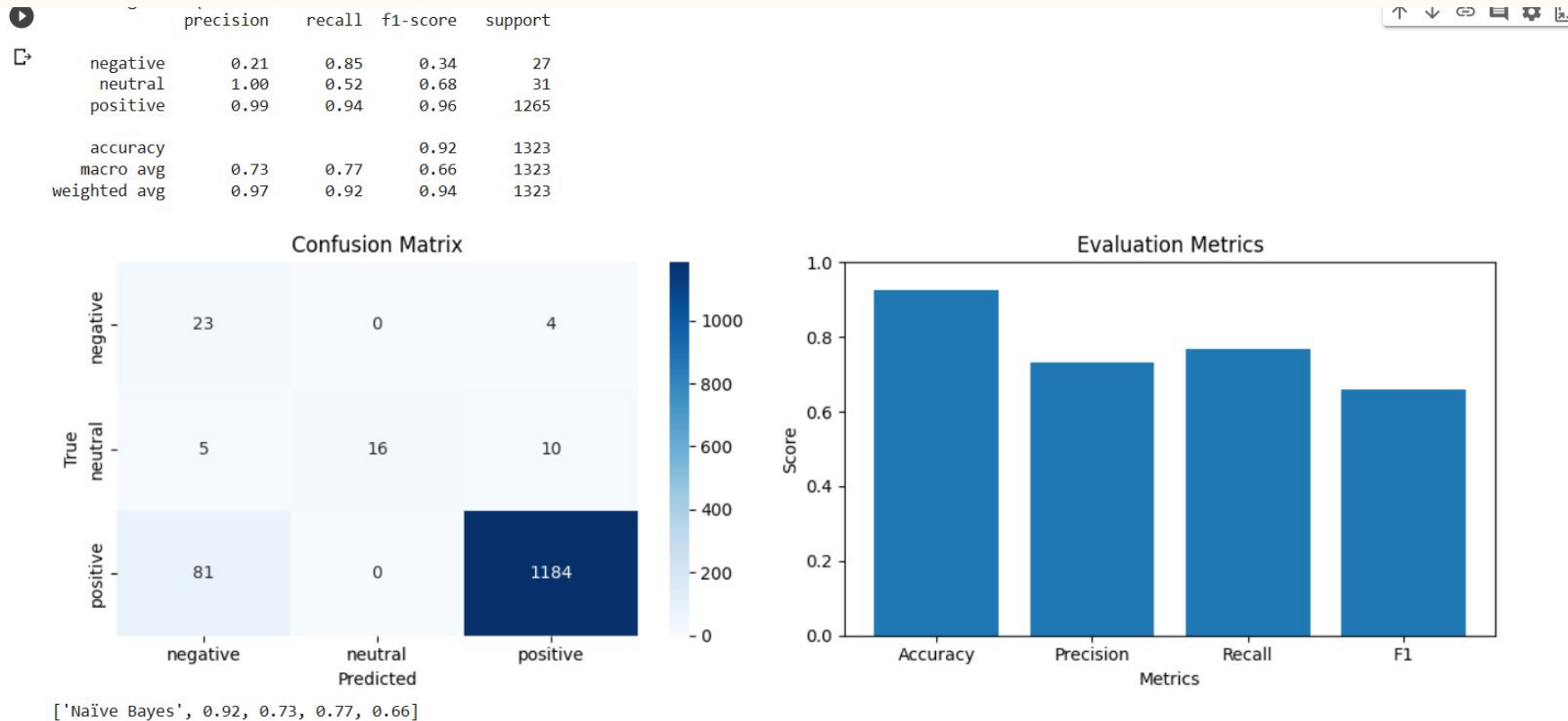
- Ta thử xây dựng với nhiều k khác nhau (1,10,2)



```
clf = KNeighborsClassifier(n_neighbors=9)
```



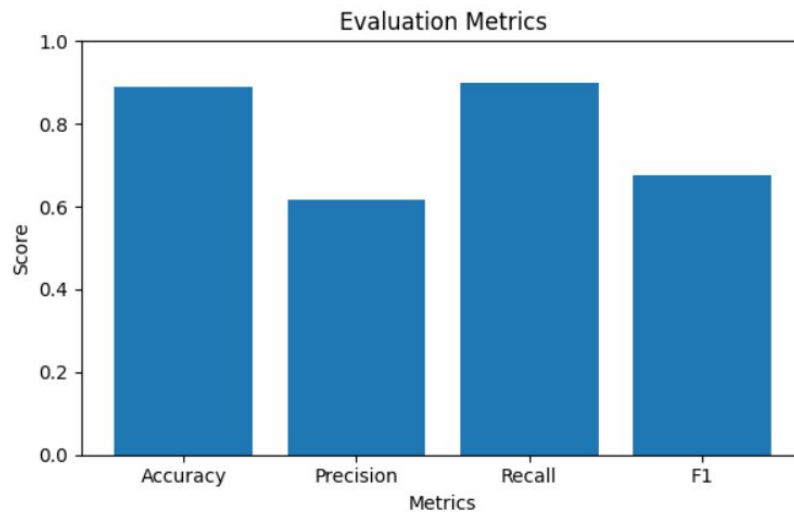
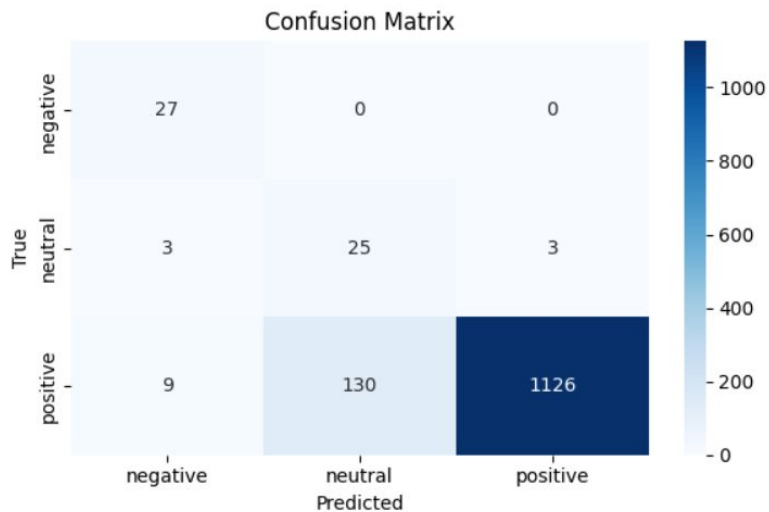
3. Naïve Bayes



4. Randomforest

warnings.warn(
precision recall f1-score support

negative	0.69	1.00	0.82	27
neutral	0.16	0.81	0.27	31
positive	1.00	0.89	0.94	1265
accuracy			0.89	1323
macro avg	0.62	0.90	0.68	1323
weighted avg	0.97	0.89	0.92	1323



['Random Forest', 0.89, 0.62, 0.9, 0.68]

5. Neural network

Model: "sequential_8"

Layer (type)	Output Shape	Param #
dense_24 (Dense)	(None, 500)	52000
dense_25 (Dense)	(None, 256)	128256
dense_26 (Dense)	(None, 180)	46260
dense_27 (Dense)	(None, 100)	18100
dense_28 (Dense)	(None, 3)	303

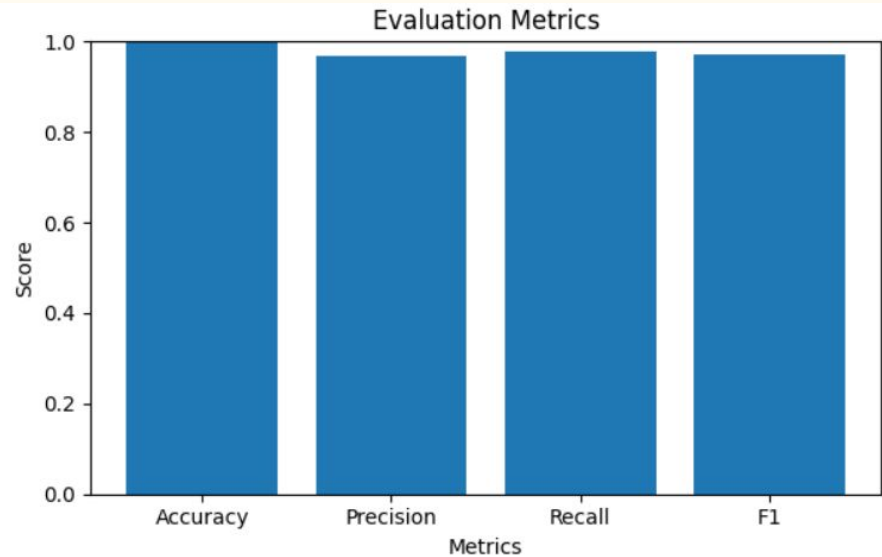
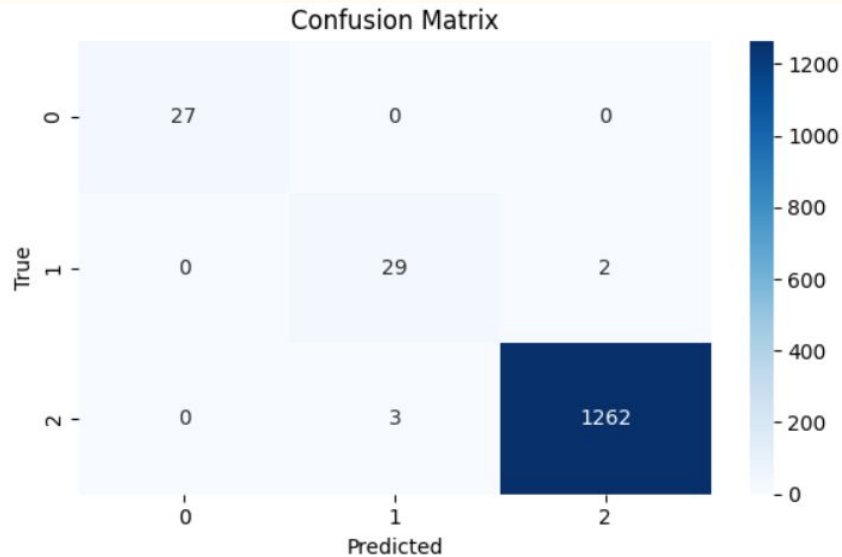
Total params: 244,919

Trainable params: 244,919

Non-trainable params: 0

Xây dựng với cấu trúc mạng như sau

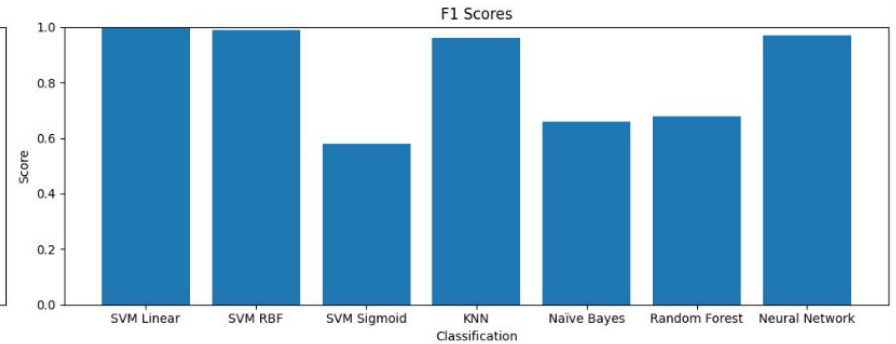
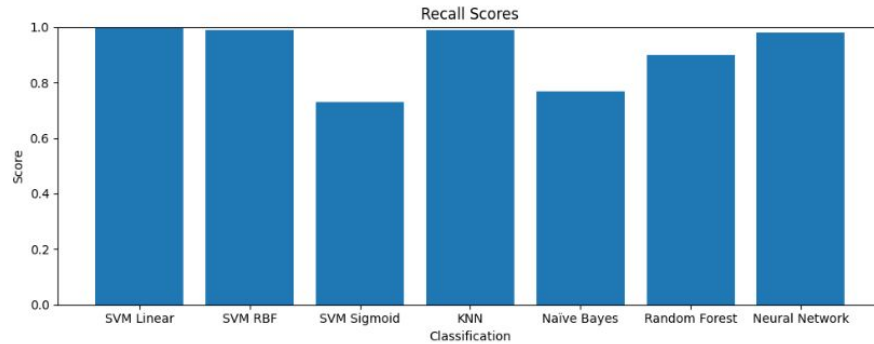
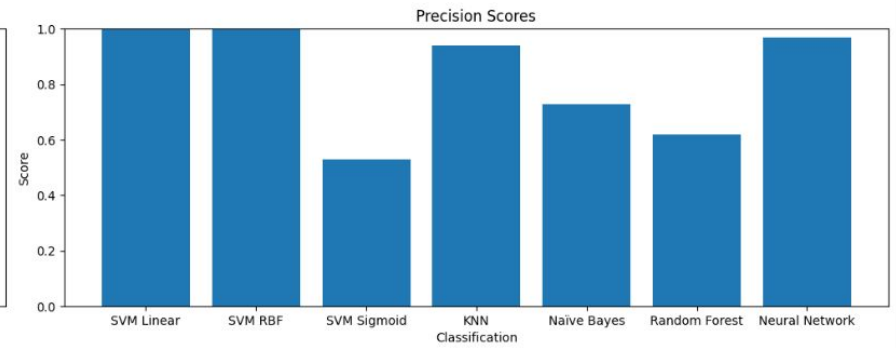
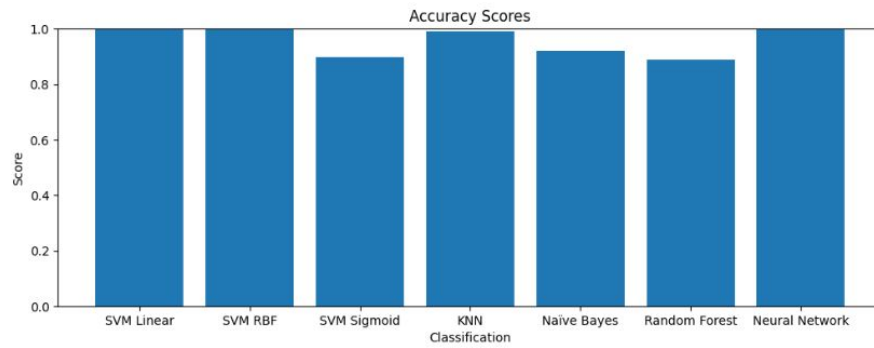
		precision	recall	f1-score	support
	0	1.00	1.00	1.00	27
	1	0.91	0.94	0.92	31
	2	1.00	1.00	1.00	1265
	accuracy			1.00	1323
	macro avg	0.97	0.98	0.97	1323
	weighted avg	1.00	1.00	1.00	1323



['Neural Network', 1.0, 0.97, 0.98, 0.97]

IV. Kết luận





CẢM ƠN THẦY VÀ CÁC
BẠN ĐÃ LẮNG NGHE