MÔ HÌNH KHAI PHÁ Ý KIẾN VÀ PHÂN TÍCH CẢM XÚC TỪ BÌNH LUẬN CỦA KHÁCH HÀNG THƯƠNG MẠI ĐIỆN TỬ BẰNG PHƯƠNG PHÁP HỌC MÁY

Thái Tường An, Nguyễn Thúy Thùy Dương

Khoa Công nghệ thông tin, Đại học Nông Lâm TP.HCM

Tóm tắt: Hiện nay, khi thương mại điện tử ngày càng tăng lên và trở thành xu hướng, nơi mà sự đánh giá và phản hồi từ khách hàng đóng vai trò quan trọng trong việc xây dựng và duy trì mối quan hệ với khách hàng. Việc phân tích ý kiến và cảm xúc của khách hàng từ dữ liệu văn bản trong thương mại điện tử giúp cho doanh nghiệp hiểu rõ hơn về nhu cầu và mong muốn của khách hàng, từ đó giúp doanh nghiệp cải thiện sản phẩm của họ để đáp ứng đúng nhu cầu của thị trường. Phương pháp áp dụng trong dự án này bao gồm việc sử dụng kỹ thuật xử lý ngôn ngữ tự nhiên để tiền xử lý dữ liệu văn bản và áp dụng các thuật toán học máy như SVM, KNN và BERT để phân loại cảm xúc của khách hàng.

1. Giới thiệu

Trong mô hình khai phá ý kiến và phân tích cảm xúc khách hàng thương mại điện tử, một số vấn đề chính mà mô hình phải đối mặt bao gồm xử lý ngôn ngữ tự nhiên không chính xác và sự phức tạp trong biểu hiện cảm xúc của khách hàng. Khả năng thích nghi của mô hình với mong muốn của khách hàng theo thời gian cũng rất quan trọng để phản ánh đúng xu hướng và ngữ cảnh hiện tại.

Mô hình giúp doanh nghiệp hiểu rõ hơn về nhu cầu và đánh giá của khách hàng, từ đó cải thiện trải nghiệm của khách hàng. Doanh nghiệp có thể tối ưu hóa chiến lược kinh doanh, cải thiện chất lượng sản phẩm, tăng cường dịch vụ và xây dựng mối quan hệ vững chắc với khách hàng.

Các phương pháp áp dụng trong mô hình bao gồm TF-IDF được sử dụng để trích xuất đặc trưng từ văn bản; GridSearchCV để tối ưu hóa tham số; các thuật toán SVM, BERT, KNN dùng để phân loại và phân tích cảm xúc từ các bình luận.

2. Các công trình liên quan:

2.1 Mô hình khai phá ý kiến và phân tích cảm xúc khách hàng trực tuyến trong ngành thực phẩm:

Bài nghiên cứu sử dụng thuật toán hồi quy **L**ogistic, Naïve Bayes và Decision Tree để phân tích cảm xúc của khách hàng thành tích cực và tiêu cực từ dữ liệu thu thập được từ Foody. Nghiên cứu cho thấy sau khi thực nghiệm **chỉ ra** mô hình tốt nhất được chọn dựa trên F1-score, với độ chính xác cao nhất đạt 90% thuộc về thuật toán Hồi quy Logistic **và** Naïve Bayes có độ chính xác thấp nhất là 78%.

2.2 Phân tích cảm xúc cho các bài đánh giá sản phẩm thương mại điện tử bằng tiếng Trung dựa trên từ điển cảm xúc và học sâu

Bài nghiên cứu tập trung vào phân tích cảm xúc người dùng trên các nền tảng thương mại điện tử của Trung Quốc và đề xuất một mô hình mới là SLCABG, là mô hình kết hợp từ các phương pháp sentiment lexicon (từ điển cảm xúc), CNN, Bi-directional Gated Recurrent Unit (BiGRU), và cơ chế attention để cải thiện hiệu suất phân tích cảm xúc.Mô hình bắt đầu từ việc xây dựng từ điển cảm xúc dựa trên các từ điển có ở các nguồn mở. Văn bản đầu vào sẽ được so khớp và trọng số hóa từng từ để tạo ra một giá trị tổng hợp cho cảm xúc của văn bản. Sau đó dùng CNN và BiGRU để trích xuất ra các đặc trưng và ngữ cảnh quan trọng. Cơ chế attention được áp dụng để xác định tiếp các phần quan trọng nhất và tính trọng số cho các văn bản.

2.3 Mô hình học máy trong phân tích ý kiến khách hàng dựa trên văn bản tiếng Việt

Nghiên cứu này đề xuất một mô hình học máy trong phân tích ý kiến khách hàng dựa trên văn bản tiếng Việt: trường hợp bài toán dịch vụ khách sạn, phân lớp sắc thái một review là tích cực hay tiêu cực. Trong đó, giải pháp tập trung cải tiến: tiền xử lý, chuẩn hóa, gán lại nhãn cho dữ liệu huấn luyện bằng kỹ thuật Error Analysis; Tăng cường dữ liệu huấn luyện bằng từ điển cảm xúc; Sử dụng 5-FoldCV, Confusion Matrix để kiểm soát overfitting và underfitting và kiểm thử mô hình; Tunning siêu tham số để tối ưu hóa tham số mô hình; Ensemble Methods kết hợp sub models để đưa ra mô hình học máy kết hợp có hiệu suất cao nhất. Trong đó dữ liệu được thu thập từ website booking.com. Kết quả thực nghiệm mô hình với tập dữ liệu thực cho thấy phương pháp đề xuất là khá hiệu quả so với các nghiên cứu trước đây, độ chính xác F1 đạt đến 96,03%.

3. Phát biểu bài toán:

3.1. Bài toán

Xây dựng một mô hình có khả năng đọc và hiểu nghĩa các đánh giá của khách hàng. Mục tiêu của mô hình là phân loại các đánh giá của khách hàng thành các loại cảm xúc tích cực và tiêu cực.

Input: Các đoạn văn bản đánh giá sản phẩm từ khách hàng. Đây là dữ liệu thô được lấy trực tiếp từ các bình luận, nhận xét và phản hồi của người dùng trên các sản phẩm.

Output: Phân loại cảm xúc của mỗi đánh giá thành một trong hai nhãn: tích cực hoặc tiêu cực.

3.2. Thuật toán

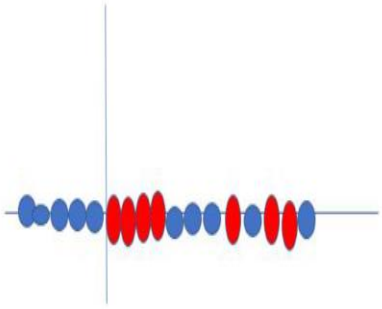
Support Vector Machine:

Support Vector Machine (SVM) là một thuật toán học máy có giám sát được sử dụng cho các bài toán phân loại và hồi quy, dựa trên việc tìm ra siêu phẳng tối ưu để phân tách các lớp dữ liệu.

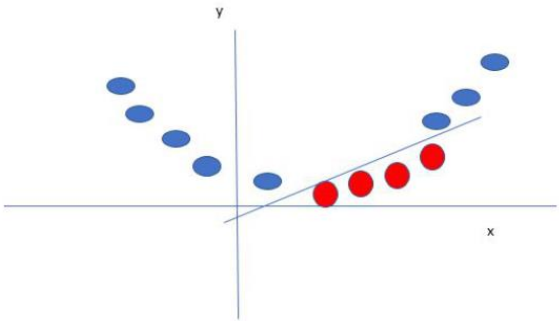
SVM hoạt động bằng cách tìm ra một siêu phẳng tốt nhất để phân tách các điểm dữ liệu của các lớp khác nhau trong không gian nhiều chiều. Mục tiêu là tối đa hóa khoảng cách giữa siêu phẳng

và các điểm dữ liệu gần nhất của mỗi lớp, gọi là margin (Margin = $\frac{2}{||w||}$). Điều này đồng nghĩa với việc tối thiểu hóa $\frac{1}{2}$ **||w||²** với ràng buộc $y_i(w^T x_i + b) >= 1$.

Trong trường hợp dữ liệu không phân tách tuyến tính, SVM sử dụng kernel trick để ánh xạ dữ liệu vào không gian đặc trưng cao hơn. Các vector hỗ trợ là các điểm dữ liệu gần nhất với siêu phẳng, quyết định vị trí của siêu phẳng và margin, từ đó xác định kết quả phân loại.



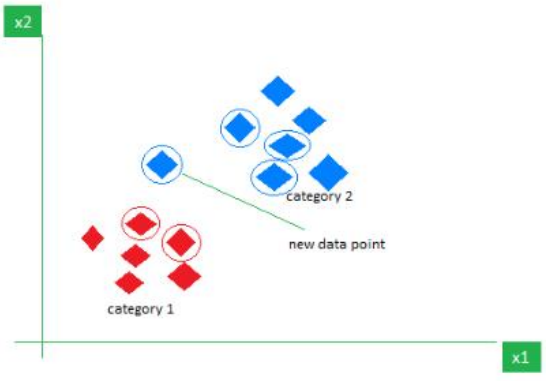
Hình 1.1. Tập dữ liệu gốc



Hình 1.2. Hàm kernel ánh xạ tập dữ liệu sang chiều cao hơn

K-Nearest Neighbor (KNN):

K-NN là một thuật toán học máy có giám sát được sử dụng để giải quyết các vấn đề phân loại và hồi quy. Đây là một phương pháp phi tham số đưa ra dự đoán dựa trên sự giống nhau của các điểm dữ liệu trong một tập dữ liệu nhất định. Thuật toán K-NN hoạt động bằng cách tìm K lân cận gần nhất với một điểm dữ liệu nhất định dựa trên thước đo khoảng cách (Euclide - độ dài của đường thẳng nối hai điểm đang xét, Manhattan - khoảng cách giữa hai điểm trong không gian nhiều chiều dựa trên tổng của các khoảng cách theo các trục tọa độ, Minkowski - một phép đo tổng quát hóa khoảng cách Euclidean và khoảng cách Manhattan).



Hình 2. Xác định lớp của điểm dữ liệu mới dựa trên số lượng k lân cận

Trong thuật toán K-NN, bước đầu tiên là chọn giá trị tối ưu của K số lượng các điểm lân cận gần nhất để sử dụng trong dự đoán. Tiếp theo, tính toán khoảng cách giữa điểm dữ liệu mục tiêu và tất cả các điểm trong tập huấn luyện theo loại khoảng cách đã chọn. Sau đó, xác định k điểm dữ liệu gần nhất dựa trên các khoảng cách này. Cuối cùng, đối với bài toán phân loại, dự đoán lớp của điểm dữ liệu mục tiêu được thực hiện bằng cách lấy lớp phổ biến nhất trong k điểm lân cận gần nhất thông qua biểu quyết đa số.

Bidirectional Encoder Representations from Transformers (BERT)

BERT là một mô hình học sâu được thiết kế để xử lý ngôn ngữ tự nhiên (NLP). BERT được thiết lập nhiều chuẩn mực mới bao gồm các tác vụ như phân loại văn bản, gán nhãn thực thể, và trả lời câu hỏi. BERT sử dụng cấu trúc Transformer, điều này có nghĩa là trong quá trình huấn luyện, BERT xem xét cả ngữ cảnh trái và phải của một từ trong câu, nắm bắt được ngữ nghĩa ngữ cảnh chính xác hơn. BERT được huấn luyện trước trên một lượng lớn dữ liệu văn bản để tạo ra các biểu diễn ngữ nghĩa ngữ cảnh.

Các bước thực hiện

- **Chuẩn bị dữ liệu**:

Dữ liệu được token hóa và mã hóa bằng cách sử dụng BertTokenizer từ thư viện transformers.

Chuyển đổi các token đầu vào thành vector nhúng thông qua:

$$\boldsymbol{E = Etoken + Eposition + Esegment}$$

Trong đó:

Etoken: embedding của từ token.

Eposition: positional encoding - vector mã hóa vị trí của từng từ trong câu

Esegment: segment embedding - vector nhúng phân biệt các đoạn văn bản khác nhau trong một câu.

- **Điều chỉnh siêu tham số**:

Learning Rate: Tốc độ học của mô hình, thường từ 1e-5 đến 3e-5. Learning rate ảnh hưởng đến độ hội tụ của mô hình.

Batch Size: Kích thước của batch, thường từ 16 đến 32. Batch Size ảnh hưởng đến hiệu suất và yêu cầu bộ nhớ.

- **Huấn luyện mô hình**:

Sử dụng AdamW optimizer để huấn luyện mô hình. Quá trình huấn luyện bao gồm việc tính toán loss và cập nhật trọng số của mô hình thông qua backpropagation. Cơ chế attention được tích

hợp trong lớp BertModel và được sử dụng khi mô hình BERT tính toán sự chú ý giữa các từ trong câu cho các token đầu vào.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

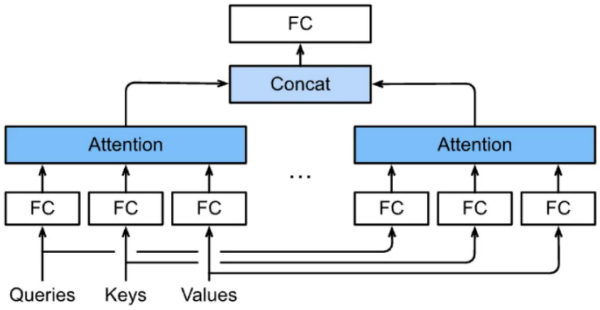
Trong đó:

Q (Query) là ma trận truy vấn

K (Key) là ma trận khóa

V (Value) là ma trận giá trị

Dk là kích thước của vector khóa



Sau mỗi lớp self-attention, đầu ra được đưa qua một mạng feed-forward.

$$\boldsymbol{FFN(x) = max(0, xW1 + b1)\, W2 + b2}$$

Trong đó:

- W1, W2 là các ma trận trọng số
- b1, b2 là các hệ số bias
- $max\ (0, x)$ là hàm kích hoạt ReLU

BERT sử dụng Cross-Entropy Loss cho các nhiệm vụ phân loại.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i log(p_i)$$

Trong đó:

- N là số lượng mẫu
- Yi là nhãn thực tế
- Pi là xác suất dự đoán của mô hình

4. Thực nghiệm

4.1 Dữ liệu

- **Mô tả dữ liệu**

Dataset Amazon Musical Instruments Reviews là tập hợp đánh giá sản phẩm cho các nhạc cụ được bán trên Amazon.com. dataset này bao gồm hơn 100.000 đánh giá với thông tin về sản phẩm, đánh giá của khách hàng và xếp hạng. Tập dữ liệu này có thể hữu ích cho các dự án học máy liên quan đến phân tích cảm xúc và khai thác ý kiến khách hàng.

Cấu trúc Dataset: Dataset được cung cấp trong tệp CSV với các thông tin sau:

```
The shape of the  data is (row, column):(10261, 9)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10261 entries, 0 to 10260
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   reviewerID      10261 non-null  object
 1   asin            10261 non-null  object
 2   reviewerName    10234 non-null  object
 3   helpful         10261 non-null  object
 4   reviewText      10254 non-null  object
 5   overall         10261 non-null  float64
 6   summary         10261 non-null  object
 7   unixReviewTime  10261 non-null  int64
 8   reviewTime      10261 non-null  object
dtypes: float64(1), int64(1), object(7)
memory usage: 721.6+ KB
None
```

Tập dữ liệu có dạng bảng với 10261 dòng và 9 cột.

reviewerID: Mã nhận dạng duy nhất của người đánh giá (Kiểu dữ liệu: object, không có giá trị thiếu).

asin: Mã nhận dạng duy nhất của sản phẩm được đánh giá (Kiểu dữ liệu: object, không có giá trị thiếu).

reviewerName: Tên người đánh giá (Kiểu dữ liệu: object, có 27 giá trị thiếu).

helpful: Cột helpful trong tập dữ liệu có định dạng dưới dạng danh sách [a, b], trong đó:

- a là số lượng người đánh giá thấy đánh giá này hữu ích.
- b là tổng số lượng người đã đánh giá xem đánh giá này có hữu ích hay không

(Kiểu dữ liệu: object, không có giá trị thiếu).

reviewText: Nội dung đánh giá (Kiểu dữ liệu: object, có 7 giá trị thiếu).

overall: Xếp hạng tổng thể (từ 1 đến 5 sao) cho sản phẩm (Kiểu dữ liệu: float64, không có giá trị thiếu).

summary: Tóm tắt đánh giá của khách hàng (Kiểu dữ liệu: object, không có giá trị thiếu).

unixReviewTime: Thời gian đánh giá theo định dạng Unix (Kiểu dữ liệu: int64, không có giá trị thiếu).

reviewTime: Ngày giờ đánh giá (Kiểu dữ liệu: object, không có giá trị thiếu).

TIỀN XỬ LÝ DỮ LIỆU

1. Loại bỏ các cột không cần thiết.

Các cột như *reviewerName, unixReviewTime, asin, reviewTime, helpful* không cung cấp thông tin hữu ích cho việc phân tích sentiment nên được loại bỏ khỏi dataset để giảm dung lượng bộ nhớ và tăng hiệu suất xử lý dữ liệu.

2. Tạo cột *reviews* mới bằng cách gộp *reviewText* và *summary*.

Cả *reviewText* và *summary* đều chứa thông tin phản hồi từ khách hàng. Kết hợp chúng sẽ tạo ra một dữ liệu văn bản đầy đủ hơn để phân tích sentiment, đồng thời cũng giải quyết được vấn đề dữ liệu bị thiếu trong cột *reviewText*.

3. Gán nhãn *sentiment* (Positive hoặc Negative) dựa trên giá trị *overall*.

Giá trị *overall* (số sao) trực tiếp biểu thị mức độ hài lòng của khách hàng. Sử dụng giá trị này để gán nhãn *sentiment* sẽ giúp phân loại đánh giá thành tích cực hoặc tiêu cực. Đánh giá từ 3 sao trở lên được gán nhãn là 'Positive', dưới 3 sao được gán nhãn là 'Negative'.

4. Làm sạch dữ liệu văn bản trong cột *reviews*.

Dữ liệu văn bản thường chứa nhiều ký tự không cần thiết như dấu câu, ký tự đặc biệt và số. Làm sạch dữ liệu giúp loại bỏ nhiễu và cải thiện chất lượng dữ liệu đầu vào cho mô hình học máy. Tiến hành chuyển đổi văn bản thành chữ thường, loại bỏ dấu câu, ký tự đặc biệt, và các ký tự số.

5. Loại bỏ các từ không mang ý nghĩa cảm xúc (stop words).

Các từ dừng (stop words) như "and", "the", "is",... không mang nhiều thông tin hữu ích cho việc phân tích sentiment. Loại bỏ chúng sẽ giúp tập trung vào các từ quan trọng hơn.

6. Tạo các biểu đồ tần suất xuất hiện của từ trong các đánh giá tích cực và tiêu cực.

Việc hiểu tần suất xuất hiện của các từ trong các đánh giá tích cực và tiêu cực giúp phát hiện các mẫu và xu hướng trong dữ liệu. Tạo các biểu đồ tần suất cho các từ xuất hiện trong đánh giá tích cực và tiêu cực, sử dụng biểu đồ thanh ngang để dễ dàng so sánh.

7. Mã hóa nhãn sentiment (Positive = 1, Negative = 0).

Sử dụng mã hóa nhãn (Label Encoding) để chuyển nhãn Positive thành 1 và Negative thành 0.

8. Tách từ và stemming.

Tách từ (tokenization) giúp phân chia văn bản thành các từ đơn lẻ. Stemming giúp giảm các từ về gốc từ (root word), giảm số lượng từ khác nhau cần xử lý.

9. Sử dụng TF-IDF để biến đổi dữ liệu văn bản.

TF-IDF (Term Frequency-Inverse Document Frequency) giúp biến đổi dữ liệu văn bản thành dạng vector, đánh trọng số cho các từ dựa trên tần suất xuất hiện của chúng trong tài liệu và trong toàn

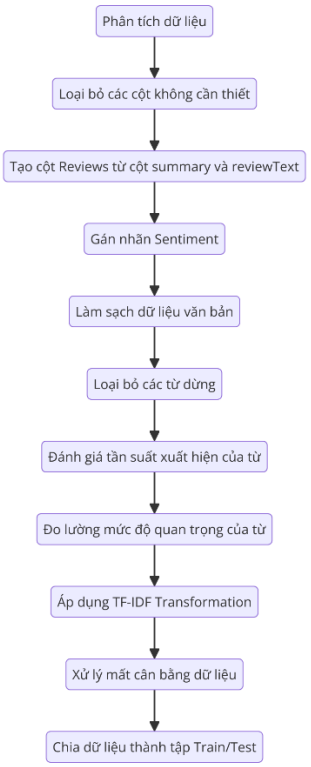
bộ tập dữ liệu. Áp dụng TF-IDF để biến đổi cột reviews thành ma trận đặc trưng (feature matrix) cho các bước phân tích tiếp theo

10. Xử lý mất cân bằng dữ liệu bằng SMOTE.

Dữ liệu ban đầu thường có sự chênh lệch giữa số lượng đánh giá tích cực và tiêu cực. Mất cân bằng này có thể làm giảm hiệu suất của mô hình. SMOTE (Synthetic Minority Over-sampling Technique) giúp tăng số lượng mẫu trong lớp ít xuất hiện để cân bằng dữ liệu.

11. Chia tập train và test

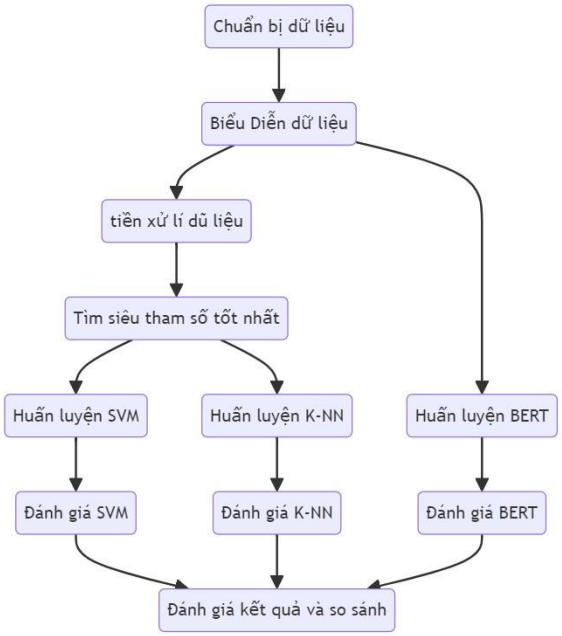
Chia tập dữ liệu thành 70% tập huấn luyện và 30% tập kiểm tra.



4.2. Phương pháp

Để phân loại bình luận của khách hàng thành các lớp ‘tích cực’ và ‘tiêu cực’, ba thuật toán học máy SVM, K-NN và BERT được áp dụng nhằm đánh giá đâu là thuật toán tốt nhất dành cho mô hình.

Dưới đây là sơ đồ luồng cho các bước thực hiện của các thuật toán:



Siêu tham số của từng thuật toán

Support Vector Machine

- C: ảnh hưởng đến độ chính xác của mô hình trong việc phân loại dữ liệu huấn luyện. Việc tối ưu hóa biên độ và xử lý các điểm dữ liệu bị phân loại sai được cân bằng bởi tham số điều chỉnh C trong SVM. Hình phạt cho việc vượt quá biên độ hoặc phân loại sai các điểm dữ liệu được quyết định bởi tham số này.
- γ (gamma): tham số này điều chỉnh phạm vi ảnh hưởng của một điểm dữ liệu đơn lẻ trên không gian đặc trưng. Nó chủ yếu áp dụng khi sử dụng các hàm kernel như Radial Basis Function (RBF) kernel.
- Kernel: xác định loại hàm kernel được sử dụng để ánh xạ dữ liệu vào không gian đặc trưng cao hơn.

K Nearest Neighbors

- n_neighbors: xác định số lượng điểm lân cận gần nhất mà mô hình sẽ xem xét khi thực hiện dự đoán cho một điểm dữ liệu mới. Thường dùng số lẻ để chọn n_neighbors để tránh bị ràng buộc trong phân loại.
- weights: xác định cách tính trọng số cho các hàng xóm gần nhất trong quá trình dự đoán. Có các tùy chọn cho tham số này là uniform (mỗi điểm có trọng số bằng nhau) và distance (trọng số được gáng dựa trên khoảng cách, điểm nào càng gần sẽ càng có trọng số lớn hơn).
- metric: xác định loại khoảng cách được sử dụng để tính toán khoảng cách giữa các điểm dữ liệu. Có thể chọn dựa trên các tùy chọn như euclidean, manhattan hoặc minkowski thông qua điều chỉnh tham số.

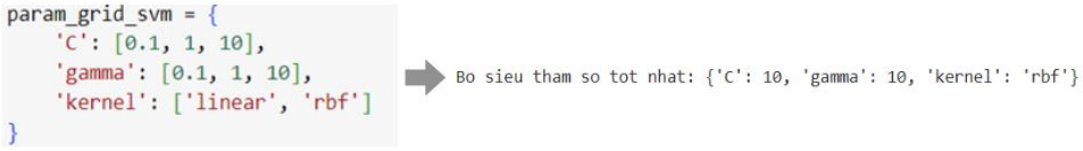
BER

- Learning Rate (lr): Ảnh hưởng đến tốc độ và chất lượng học của mô hình. Thường chọn giá trị nhỏ (ví dụ: 1e-5) để tránh làm mất ổn định mô hình.
- Batch Size: Số lượng mẫu dữ liệu được xử lý cùng một lúc. Batch size lớn giúp học nhanh hơn nhưng cần nhiều bộ nhớ hơn.
- Number of Epochs: Số lần toàn bộ tập dữ liệu được sử dụng để huấn luyện mô hình. Số lượng epoch lớn hơn có thể giúp mô hình học tốt hơn nhưng dễ gây overfitting.
- Max Sequence Length: Độ dài tối đa của chuỗi đầu vào. Độ dài lớn hơn giữ được nhiều thông tin hơn nhưng yêu cầu bộ nhớ và thời gian tính toán lớn hơn.
- AdamW Optimizer: Biến thể của Adam, tối ưu hóa học suất và kiểm soát trọng số mô hình, cải thiện hiệu suất và ổn định.
- Weight Decay: Tránh overfitting bằng cách thêm hình phạt vào hàm mất mát dựa trên trọng số của mô hình, giúp tổng quát hóa tốt hơn trên dữ liệu mới.

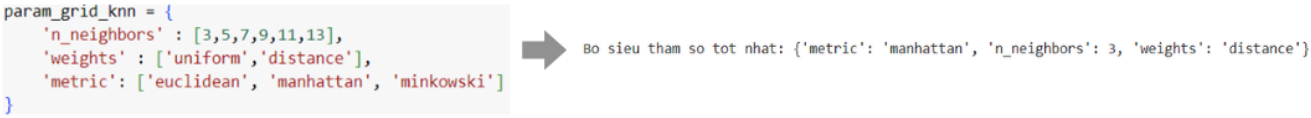
Fine-tune cho các thuật toán

Đối với mô hình khai phá ý kiến này, phương pháp GridSearchCV được áp dụng để điều chỉnh các siêu tham số cho từng thuật toán. GridSearchCV là một phương pháp trong học máy dùng để tối ưu hóa các siêu tham số của mô hình. Nó hoạt động bằng cách kiểm tra tất cả các kết hợp có thể của các giá trị siêu tham số trong một lưới (grid) đã định sẵn để tìm ra tổ hợp tốt nhất cho mô hình.

Ban đầu, thuật toán SVM và K-NN đều được đưa vào các bộ siêu tham số khác nhau và được GridSearchCV tìm ra được bộ siêu tham số tốt nhất.



Hình 3.1. Bộ siêu tham số tốt nhất cho SVM

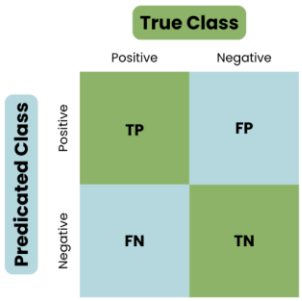


Hình 3.2. Bộ siêu tham số tốt nhất cho K-NN

Trong mô hình này, thuật toán BERT không cần điều chỉnh siêu tham số vì thuật toán này đã được huấn luyện trước trên một lượng lớn dữ liệu văn bản. Nó đã học được rất nhiều về cấu trúc ngữ pháp, ngữ nghĩa và ngữ cảnh trong ngôn ngữ tự nhiên.

Độ đo đánh giá

Accuracy và F1 score được dùng để đánh giá kết quả huấn luyện của mô hình.



Bảng ma trận nhầm lẫn

- Accuracy: đo lường tỷ lệ dự đoán đúng trên tổng số dự đoán. Công thức tính accuracy là:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

- F1 score:độ đo kết hợp giữa Precision (độ chính xác) và Recall (tỷ lệ phát hiện). Nó là trung bình điều hòa của Precision và Recall, và cung cấp một cái nhìn cân bằng giữa hai chỉ số này.

$$F1 = \frac{2 \times P \times R}{P+R}$$

Trong đó,

- P (Precision) = $\frac{TP}{TP+FP}$
- R(Recall) = $\frac{TP}{TP+FN}$

4.3. Kết quả

	SVM	KNN	BERT
ACCURACY	0.9715	0.8728	0.9493
F1 SCORE	0.9715	0.8728	0.9247

Phân tích và đánh giá kết quả

SVM (Support Vector Machine): SVM đạt được độ chính xác và F1 Score cao nhất trong các mô hình được so sánh. Điều này cho thấy SVM là một mô hình mạnh mẽ cho bài toán phân loại này, có khả năng tìm kiếm và tối ưu hóa siêu phẳng tốt để phân tách các lớp dữ liệu.

KNN (K-Nearest Neighbor): KNN có hiệu suất thấp hơn so với SVM, với cả độ chính xác và F1 Score thấp hơn. Điều này có thể do KNN là một mô hình dựa trên khoảng cách, không phù hợp tốt với dữ liệu có đặc tính phức tạp.

BERT: Mặc dù không qua quá trình tiền xử lý văn bản, vẫn đạt được hiệu suất cao với độ chính xác và F1 Score gần với SVM. Điều này cho thấy sức mạnh của mô hình BERT trong việc hiểu và xử lý ngôn ngữ tự nhiên mà không cần nhiều bước tiền xử lý.

So sánh giữa các phương pháp

- SVM là mô hình tốt nhất trong các phương pháp đã thử nghiệm, với hiệu suất vượt trội cả về độ chính xác và F1 Score.
- BERT cũng đạt được kết quả ấn tượng mà không cần qua quá trình tiền xử lý, cho thấy sự tiên tiến của các mô hình học sâu hiện đại trong việc xử lý ngôn ngữ tự nhiên.
- KNN có hiệu suất thấp hơn so với SVM và BERT, cho thấy chúng có thể không phải là lựa chọn tốt nhất cho bài toán này, đặc biệt khi dữ liệu có tính phức tạp cao.

5. Kết luận

SVM và BERT là hai mô hình mạnh mẽ nhất cho bài toán phân loại cảm xúc từ bình luận của khách hàng. SVM nổi bật với khả năng tối ưu hóa mạnh mẽ, đạt hiệu suất cao nhất với Accuracy và F1 Score. BERT với khả năng xử lý ngôn ngữ tự nhiên tiên tiến, cũng cho thấy kết quả cao khi huấn luyện. Mô hình này có thể đạt được hiệu suất cao ngay cả khi không cần nhiều bước tiền xử lý, cho thấy tiềm năng của các mô hình học sâu trong việc xử lý ngôn ngữ tự nhiên.

Ưu nhược điểm và đề xuất phát triển:

SVM có ưu điểm về hiệu suất cao nhưng cần điều chỉnh nhiều tham số, trong khi BERT mạnh mẽ trong xử lý ngôn ngữ tự nhiên nhưng yêu cầu tài nguyên tính toán lớn. KNN đơn giản, dễ triển khai nhưng hiệu suất thấp. Trong tương lai, có thể thử nghiệm thêm các thuật toán khác như Random Forest, Gradient Boosting, hoặc các mô hình học sâu khác, cải thiện phương pháp tiền xử lý dữ liệu, áp dụng kỹ thuật tăng cường dữ liệu và tối ưu hóa siêu tham số để nâng cao hiệu suất mô hình.

Tài liệu tham khảo

Nguyễn Thanh Thủy, Trần Thị Châu Giang. “Một mô hình học máy trong phân tích ý kiến khách hàng dựa trên văn bản tiếng việt: bài toán dịch vụ khách sạn”. Kỷ yếu hội thảo khoa học quốc gia 2019 “Công nghệ thông tin và ứng dụng trong các lĩnh vực”.

Nguyễn D L Đằng và cộng sự.” A text-based model for opinion mining and sentiment analysis from online customer reviews in food industry”. Tạp chí khoa học Đại học mở thành phố HCM 16(1) 67-78.

Li Yang, Ying Li, Jin Wang, R.Simon Sherratt. “Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning”, IEEE Access, 2020, 5-6.

K-Nearest Neighbor(KNN) Algorithm, 2024, GeeksforGeeks.

Support Vector Machine (SVM) Algorithm, 2024, GeeksforGeeks.

Pham Dinh Khanh, Bài 36 - BERT model, 2020.

Link source code: https://github.com/thaituongan/Project_ML

PHÂN CÔNG NHÓM

Tên thành viên	Nhiệm vụ
Nguyễn Thúy Thùy Dương	Tìm hiểu công trình ở mục 2.1, 2.2 Tìm hiểu các bước tiền xử lý Thực hiện tiền xử lý dữ liệu ở bước 1, 2, 3, 9, 10, 11 Tìm tham số cho mô hình SVM, K-NN Huấn luyện mô hình bằng thuật toán SVM, K-NN Phát biểu bài toán Thực hiện các bước điều chỉnh siêu tham số Hoàn thiện tài liệu
Thái Tường An	Tìm kiếm dữ liệu Mô tả dữ liệu Tìm hiểu công trình ở mục 2.3 Tìm hiểu các bước tiền xử lý Thực hiện tiền xử lý dữ liệu ở bước 4, 5, 6, 7, 8 Vẽ các flowchart Tìm tham số cho BERT Huấn luyện mô hình bằng thuật toán BERT Phân tích, đánh giá và so sánh kết quả