SAMSUNG

Big Data Course

Capstone Project Final Report

For students (instructor review required)

Phân tích dữ liệu tài chính và thương mại điện tử

14/08/24

OU - HCM - BigData02 – Nhóm 2

Lê Minh Kiệt
Võ Trần Yến Như
Dương Ngọc Minh Huy
Trần Hữu Hậu
Trần Trọng Nhân
Nguyễn Thanh Nam
Nguyễn Thanh Nở

Content

1. Introduction

1.1. Background Information

Bình luận và đánh giá của khách hàng là một phần quan trọng trong thương mại điện tử, giúp những khách hàng cũng như người bán có thể đánh giá chung về chất lượng sản phẩm và mức độ hài lòng của khách hàng sau khi mua hàng.
Các bình luận có thể được chia làm 3 loại:

- **Hài lòng**:
    - **Chất lượng sản phẩm:** Khách hàng bày tỏ sự hài lòng về chất lượng của sản phẩm, bao gồm cả hiệu suất, thiết kế, và độ bền. Những bình luận như "Sản phẩm tuyệt vời", "Hoạt động rất tốt", hay "Hoàn toàn hài lòng với chất lượng" là điển hình của nhóm này.
    - **Trải nghiệm mua hàng:** Quy trình mua hàng diễn ra suôn sẻ, từ việc đặt hàng, thanh toán đến giao hàng. Khách hàng thường khen ngợi về thời gian giao hàng nhanh, đóng gói cẩn thận, và sản phẩm đến tay trong tình trạng hoàn hảo. Các nhận xét như "Giao hàng rất nhanh", "Mua hàng dễ dàng" là đặc trưng cho nhãn này.
    - **Dịch vụ khách hàng:** Khách hàng nhận được sự hỗ trợ tận tình từ dịch vụ khách hàng, với phản hồi nhanh chóng và giải quyết các vấn đề một cách thỏa đáng. Các bình luận như "Dịch vụ khách hàng tuyệt vời", "Rất hài lòng với sự hỗ trợ" thể hiện sự hài lòng về dịch vụ.
    - **Cảm nhận chung:** Khách hàng có xu hướng thể hiện sự hài lòng chung với sản phẩm và dịch vụ, thể hiện bằng những bình luận tích cực như "Rất hài lòng", "Sẽ tiếp tục mua hàng", "Sản phẩm đáng giá".
- **Bình thường:**

    - **Chất lượng sản phẩm:** Sản phẩm có thể đạt được mong đợi ở mức cơ bản nhưng không nổi bật. Các nhận xét có thể bao gồm những cụm từ như "Chất lượng ổn", "Không có gì đặc biệt nhưng chấp nhận được", "Sản phẩm đúng như mô tả nhưng không ấn tượng".
    - **Trải nghiệm mua hàng:** Quy trình mua hàng diễn ra bình thường, không gặp sự cố lớn nhưng cũng không có điểm nổi bật nào để khen ngợi. Khách hàng có thể cảm thấy rằng các yếu tố như giao hàng, đóng gói và hỗ trợ khách hàng đều ở mức chấp nhận được nhưng không có gì xuất sắc.
    - **Dịch vụ khách hàng:** Dịch vụ khách hàng được đánh giá là trung bình, với các phản hồi như "Hỗ trợ bình thường", "Dịch vụ ở mức chấp nhận được", hoặc "Không có vấn đề gì lớn nhưng cũng không ấn tượng".
    - **Cảm nhận chung:** Khách hàng thể hiện sự trung lập, không có cảm xúc mạnh mẽ về sản phẩm hoặc dịch vụ. Những bình luận như "Cũng được", "Không tệ nhưng cũng không quá tốt", hoặc "Tạm ổn" thường xuất hiện trong nhãn này.
- **Không hài lòng:**
    - **Chất lượng sản phẩm:** khách hàng không hài lòng với sản phẩm, có thể do các vấn đề như sản phẩm không hoạt động đúng cách, chất lượng kém, hoặc không giống như mô tả. Các bình luận tiêu cực như "Sản phẩm tệ", "Không như mong đợi", "Chất lượng rất kém" là phổ biến trong nhóm này.
    - **Trải nghiệm mua hàng:** Trải nghiệm mua hàng không đạt yêu cầu, có thể do giao hàng chậm trễ, sản phẩm bị hỏng khi đến tay, hoặc các vấn đề khác trong quá trình mua sắm. Những phản hồi như "Giao hàng rất chậm", "Đóng gói kém" là dấu hiệu của sự không hài lòng.
    - **Dịch vụ khách hàng:** Khách hàng có thể không nhận được sự hỗ trợ thỏa đáng, dẫn đến sự thất vọng. Những bình luận như "Dịch vụ khách hàng rất kém", "Không nhận được sự hỗ trợ như mong đợi" thể hiện sự không hài lòng với dịch vụ.

- **Cảm nhận chung:** Khách hàng có cảm xúc tiêu cực mạnh mẽ về trải nghiệm tổng thể của họ, có thể dẫn đến những bình luận như "Không bao giờ mua lại", "Rất thất vọng", hoặc "Trải nghiệm tệ hại".

Sự đánh giá của khách hàng có vai trò quyết định sự thành công của sản phẩm trong thương mại điện tử, nếu một sản phẩm có đánh giá tích cực, nhận được sự hài lòng của khách hàng thì sản phẩm đó sẽ có khả năng thành công cao hơn rõ rệt so với những sản phẩm có đánh giá tiêu cực và sự hài lòng của khách hàng ở mức thấp. Báo cáo này nhằm đánh giá các bình luận của khách hàng cho từng sản phẩm nhằm giúp người dùng cũng như khách hàng có cái nhìn chính xác nhất về chất lượng cũng như mức độ hài lòng của khách hàng đối với sản phẩm.

1.2. Motivation and Objective

- Trong thị trường thương mại điện tử ngày càng mở rộng, việc đánh giá đúng chất lượng sản phẩm và mức độ hài lòng của người dùng là cực kỳ quan trọng. Bình luận của khách hàng đóng vai trò như một nguồn thông tin đáng tin cậy, giúp người tiêu dùng đưa ra quyết định mua sắm, đồng thời cung cấp cho nhà bán hàng thông tin cần thiết để cải thiện sản phẩm và dịch vụ của họ. Tuy nhiên, việc đánh giá sai chất lượng sản phẩm hoặc mức độ hài lòng của người dùng có thể gây ra nhiều tác hại nghiêm trọng cho cả người mua và người bán.
    - **Tác hại đối với người tiêu dùng:**
        - **Lựa chọn sản phẩm không phù hợp:** Khi chất lượng sản phẩm được đánh giá sai, người tiêu dùng có thể bị dẫn đến việc mua những sản phẩm không đáp ứng được nhu cầu hoặc mong đợi của họ. Điều này không chỉ gây mất thời gian và tiền bạc mà còn có thể làm giảm niềm tin của họ vào các nền tảng thương mại điện tử.

        - **Mất niềm tin vào đánh giá trực tuyến:** Nếu các đánh giá và bình luận bị phân loại sai, người tiêu dùng có thể cảm thấy khó tin tưởng vào các đánh giá trực tuyến, khiến họ gặp khó khăn trong việc đưa ra quyết định mua hàng trong tương lai. Điều này có thể dẫn đến việc họ chuyển sang mua sắm tại các cửa hàng truyền thống hoặc trên các nền tảng khác.
    - **Tác hại đối với nhà bán hàng**
        - **Mất khách hàng tiềm năng:** Khi một sản phẩm tốt bị đánh giá sai là kém chất lượng, nhà bán hàng có thể mất đi nhiều khách hàng tiềm năng. Ngược lại, nếu một sản phẩm kém được đánh giá cao một cách không chính xác, sẽ có nhiều khách hàng thất vọng sau khi mua, dẫn đến tăng tỷ lệ trả hàng và khiếu nại.
        - **Giảm uy tín và doanh thu:** Việc có quá nhiều đánh giá không chính xác có thể làm giảm uy tín của nhà bán hàng trên nền tảng thương mại điện tử. Uy tín giảm dẫn đến doanh thu giảm sút do người tiêu dùng có xu hướng tránh xa những nhà bán hàng hoặc sản phẩm có đánh giá không rõ ràng hoặc mâu thuẫn.
        - **Quản lý sản phẩm kém hiệu quả:** Nếu nhà bán hàng không nhận được phản hồi chính xác về sản phẩm, họ sẽ gặp khó khăn trong việc cải thiện và quản lý sản phẩm một cách hiệu quả. Điều này có thể dẫn đến việc không giải quyết kịp thời các vấn đề của sản phẩm, khiến chất lượng dịch vụ tổng thể bị ảnh hưởng.

Vì những lý do trên, việc phát triển một hệ thống tự động và chính xác để phân tích và đánh giá bình luận của khách hàng là cần thiết. Hệ thống này sẽ giúp giảm thiểu các sai sót trong việc đánh giá chất lượng sản phẩm và mức độ hài lòng của người dùng, từ đó tạo ra môi trường mua sắm trực tuyến tin cậy và hiệu quả hơn. Hơn nữa, hệ thống còn hỗ trợ nhà bán hàng trong việc duy trì và nâng cao chất lượng sản phẩm, từ đó tăng cường khả năng cạnh tranh trên thị trường.

- Mục tiêu chính của dự án là xây dựng một hệ thống dự đoán có khả năng phân loại sản phẩm thương mại điện tử thành ba nhóm "tốt", "trung bình" và "chưa tốt" dựa trên bình luận của người mua. Để đạt được điều này, dự án sẽ áp dụng các thuật toán học máy như Random Forest, Logistic Regression, Support Vector Machine và K-Nearest Neighbors kết hợp với kỹ thuật xử lý ngôn ngữ tự nhiên để phân tích nội dung bình luận. Ngôn ngữ lập trình Python sẽ được sử dụng để triển khai các mô hình và thực hiện quá trình xử lý dữ liệu. Hệ thống này nhằm cung cấp một công cụ đáng tin cậy cho cả người tiêu dùng lẫn người bán trong việc đánh giá chất lượng sản phẩm dựa trên phản hồi thực tế từ người dùng.

1.3. Members and Role Assignments

- Lê Minh Kiệt: Thu thập dữ liệu – Thuật toán Logistic – Xử lý ngôn ngữ tự nhiên – Thiết kế web – Làm sạch dữ liệu
- Võ Trần Yến Như: Thu thập dữ liệu – Làm sạch dữ liệu – Xử lý ngôn ngữ tự nhiên
- Dương Ngọc Minh Huy: Thu thập dữ liệu – Thuật toán SVM – Xử lý ngôn ngữ tự nhiên
- Trần Hữu Hậu: Thu thập dữ liệu – Thuật toán Random Forest – Xử lý ngôn ngữ tự nhiên
- Nguyễn Thanh Nam: Thu thập dữ liệu – Làm sạch dữ liệu – Thiết kế web – Xử lý ngôn ngữ tự nhiên
- Trần Trọng Nhân: Thu thập dữ liệu – Vector hóa dữ liệu – Thuật toán KNN – Xử lý ngôn ngữ tự nhiên
- Nguyễn Thanh Nở: Thu thập dữ liệu – Thiết kế web – Xử lý ngôn ngữ tự nhiên

1.4. Schedule and Milestones

Thời gian	Hoạt động	Mô tả
1/7/2024 - 2/7/2024	Tìm nguồn có dữ liệu phù hợp	
2/7/2024 - 4/7/2024	Viết chương trình lấy dữ liệu	
4/7/2024 - 6/7/2024	Lấy dữ liệu	
7/7/2024 - 8/7/2024	Loại bỏ giá trị null	
8/7/2024 - 10/7/2024	Kết hợp tất cả dữ liệu thành một file thống nhất	
11/7/2024 - 14/7/2024	Loại bỏ các dấu chấm phẩy (dấu câu)	
15/7/2024 - 18/7/2024	Xử lý ngôn ngữ tự nhiên	
19/7/2024 - 22/7/2024	Vector hóa	
23/7/2024 - 30/7/2024	Thuật toán Support Vector Machine	
23/7/2024 - 30/7/2024	Thuật toán Random Forest	
23/7/2024 - 30/7/2024	Thuật toán Logistic Regression	
23/7/2024 - 30/7/2024	Thuật toán K-Nearest Neighbors	
31/7/2024 - 5/8/2024	Thiết kế website	
6/8/2024 - 10/8/2024	Đánh giá mô hình - Lựa chọn thuật toán tối ưu nhất	
11/8/2024 - 14/8/2024	Viết báo cáo sản phẩm	

2. Project Execution

2.1. Simulated Scenario Description

Trong bối cảnh phát triển mạnh mẽ của thương mại điện tử, người tiêu dùng phải đối mặt với tình trạng quá tải thông tin khi mua sắm trực tuyến. Họ thường xuyên phải đọc qua hàng loạt nhận xét để đánh giá chất lượng sản phẩm trước khi đưa ra quyết định mua hàng. Điều này không chỉ tốn thời gian mà còn gây khó khăn khi phải lọc qua các nhận xét có giá trị thấp hoặc không liên quan.

Dự án của chúng em mô phỏng một tình huống thực tế, nơi người tiêu dùng có thể sử dụng một hệ thống thông minh để tự động phân tích nhận xét từ các trang web bán hàng. Hệ thống này sẽ xác định chất lượng của sản phẩm dựa trên nhận xét của người dùng, giúp người tiêu dùng đưa ra quyết định nhanh.

Tập dữ liệu sử dụng trong dự án này bao gồm các đánh giá sản phẩm từ một trang thương mại điện tử là Lazada. Mỗi dòng dữ liệu chứa thông tin về đánh giá của khách hàng, bao gồm:

Comment: Bình luận cảm xúc của khách hàng.

StarCount: Điểm số đánh giá sản phẩm (trong thang điểm từ 1 đến 5 sao)

Label: Được đánh giá dựa trên số sao (trong thang điểm từ 1 đến 3 tương ứng Chưa Tốt – Trung Bình – Tốt)

2.2. Datasets Selection and Description

Tập dữ liệu hơn 55.000 đánh giá, được phân chia cho các nhãn sản phẩm. Các đánh giá có chiều dài từ 10 đến 100 từ, bao gồm cả các từ ngữ tích cực và tiêu cực, phản ánh cảm nhận của khách hàng về sản phẩm. Dữ liệu sử dụng trong dự án được thu thập từ các trang thương mại điện tử lớn tại Việt Nam, nơi có nhiều nhận xét từ người tiêu dùng về các sản phẩm khác nhau. Chúng em tập trung vào các sản phẩm phổ biến như điện thoại di động, thiết bị gia dụng và mỹ phẩm.

2.3. Data Ingestion Pipeline

- Sử dụng thư viện pandas, chúng em sẽ đọc dữ liệu từ file XLSX chứa các đánh giá sản phẩm.
- Dữ liệu được thu thập thông qua quá trình scraping sử dụng công cụ Selenium.
- Scraping: Sử dụng Selenium để tự động thu thập dữ liệu từ các trang web thương mại điện tử.
- Lưu trữ tạm thời: Dữ liệu sau khi scraping được lưu trữ tạm thời trong định dạng XLSX để dễ dàng xử lý tiếp.
- Nhập dữ liệu vào cơ sở dữ liệu: Dữ liệu sau đó được nhập vào một cơ sở dữ liệu lớn, nơi chúng em có thể thực hiện các bước xử lý tiếp theo.

2.4. Data Transformation Processing

Sau khi dữ liệu được thu thập và nhập vào hệ thống, quá trình xử lý dữ liệu được thực hiện để chuẩn hóa và biến đổi dữ liệu thành dạng phù hợp cho phân tích:

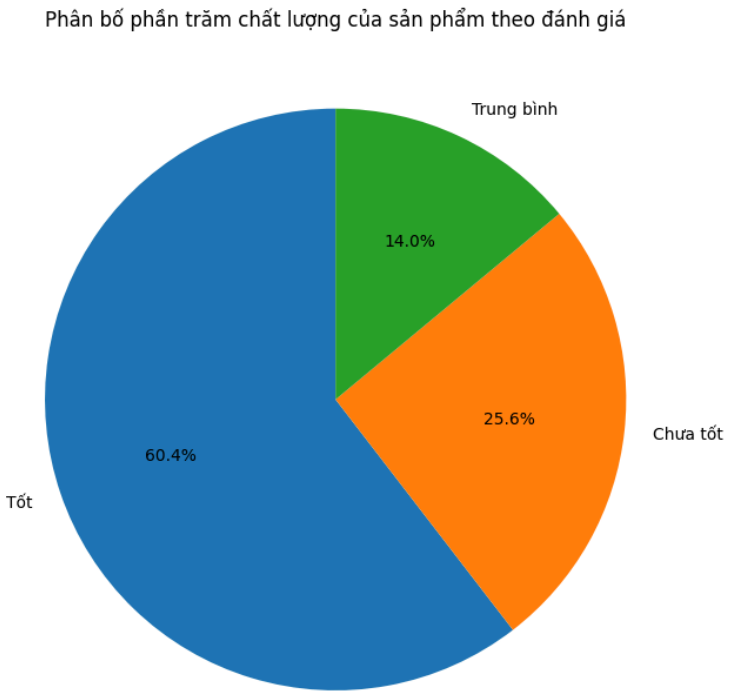
- Xử lý lỗi chính tả và viết tắt: Do dữ liệu nhận xét chủ yếu bằng tiếng Việt, chúng em phải xử lý các lỗi chính tả và các từ viết tắt phổ biến để đảm bảo tính nhất quán.
- Loại bỏ ký tự đặc biệt.
- Loại bỏ từ dừng (stop words).
- Loại bỏ các nhận xét không liên quan: Những nhận xét ngắn hoặc không cung cấp thông tin hữu ích sẽ bị loại bỏ.
- Vector hóa dữ liệu: Sử dụng TfidfVectorizer (TF-IDF) để biến đổi các nhận xét thành dạng vector số, phục vụ cho việc huấn luyện mô hình học máy.

2.5. Data Query and Insight

Sau khi dữ liệu đã được xử lý và biến đổi, chúng em thực hiện các truy vấn để rút ra những insight từ dữ liệu:

- Phân tích chất lượng sản phẩm: Dựa trên mô hình RandomForestClassifier, chúng em xác định tỷ lệ các sản phẩm có nhận xét Tốt - Trung bình - Chưa tốt.
- Xu hướng mua sắm của người tiêu dùng: Phân tích những sản phẩm nào nhận được nhiều nhận xét tích cực nhất, đánh giá cao từ trải nghiệm, giúp xác định xu hướng mua sắm.
- Nhận diện vấn đề phổ biến: Thông qua phân tích nội dung bình luận, chúng em có thể xác định các vấn đề thường gặp mà người tiêu dùng phản ánh, như chất lượng sản phẩm kém, thời gian giao hàng lâu, v.v.

Mô hình được đánh giá trên tập kiểm tra bằng các chỉ số phổ biến như độ chính xác, F1-score, độ nhạy (recall) và độ đặc hiệu (precision). Kết quả cho thấy mô hình đạt độ chính xác **92%** với F1-score trung bình là 0.901 (sử dụng average='weighted' do dữ liệu không cân bằng), cho thấy mô hình hoạt động tốt trong việc phân loại các sản phẩm dựa trên đánh giá của khách hàng.



Mô hình này có thể được tích hợp vào hệ thống quản lý sản phẩm của các doanh nghiệp thương mại điện tử để tự động phân loại các sản phẩm theo chất lượng dựa trên đánh giá khách hàng. Điều này không chỉ giúp tiết kiệm thời gian mà còn cải thiện chất lượng dịch vụ thông qua phản hồi nhanh chóng và chính xác từ khách hàng.

2.6 Building Models to Predict Product Quality Based on Customer Comments

1. Logistic Regression:

1.1. Introduction to Logistic Regression:

Hồi quy logistic là một kỹ thuật phân tích dữ liệu sử dụng toán học để tìm ra mối quan hệ giữa hai yếu tố dữ liệu. Sau đó, kỹ thuật này sử dụng mối quan hệ đã tìm được để dự đoán giá trị của những yếu tố đó dựa trên yếu tố còn lại. Dự đoán thường cho ra một số kết quả hữu hạn, như có hoặc không.

1.2. Building a Logistic Regression model.

- **Import the necessary libraries to build the model**

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.utils import resample
```

- **Read Datasets**

```
file_path = "D:/SIC/dataset.xlsx"
data = pd.read_excel(file_path, engine='openpyxl')
```

- **Build model**

```
# Giảm kích thước tập dữ liệu (nếu cần)
data_sample = resample(data, n_samples=10000, random_state=42)

# Chuẩn bị văn bản và nhãn
texts = data_sample['Comment'].astype(str).fillna('')
labels = data_sample['StarCount'].fillna(0).astype(int)

# Sử dụng TF-IDF để chuyển đổi văn bản thành vector
vectorizer = TfidfVectorizer(max_features=5000)  # Giới hạn số lượng đặc trưng
X = vectorizer.fit_transform(texts)

# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.2, random_state=42)

# Chuẩn hóa dữ liệu (không áp dụng mean vì dữ liệu dạng thưa)
scaler = StandardScaler(with_mean=False)
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Huấn luyện mô hình Logistic Regression
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

- **Model evaluation**

```
# Đánh giá mô hình
accuracy = model.score(X_test, y_test)
print(f'Accuracy: {accuracy}')
```

2. Random Forest:

2.1. Introduction to Random Forest:

Random là ngẫu nhiên, Forest là rừng, nên ở thuật toán Random Forest mình sẽ xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên mỗi cây quyết định sẽ khác nhau (có yếu tố random). Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định.

2.2. Building a Random Forest model.

- **Import the necessary libraries to build the model**

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import numpy as np
```

- **Read Datasets**

```
# Đọc dữ liệu từ file Excel
file_path = "D:/SIC/dataset.xlsx"
data = pd.read_excel(file_path, engine='openpyxl')
```

- **Build model**

```
# Chuẩn bị văn bản và nhãn
texts = data['Comment'].astype(str)
texts = [text if pd.notna(text) else '' for text in texts]
labels = data['Label_new']
labels = [star if pd.notna(star) else '' for star in labels]

# Sử dụng TF-IDF để chuyển đổi văn bản thành vector
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(texts)

# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.3, random_state=42)

# Huấn luyện mô hình RandomForest
clf = RandomForestClassifier(n_estimators=300, random_state=42, max_depth=100)
clf.fit(X_train, y_train)
```

- **Model evaluation**

```
from sklearn.metrics import recall_score, f1_score, accuracy_score
y_pred = clf.predict(X_test)
accuracy = clf.score(X_test, y_test)
print(f'Độ chính xác: {accuracy:.3f}')
correct_predictions = accuracy_score(y_test, y_pred, normalize=False)
recall = recall_score(y_test, y_pred, average='weighted')
f_score = f1_score(y_test, y_pred, average='weighted')
incorrect_predictions = len(y_test) - correct_predictions
print("Số lượng dự đoán chính xác (Prediction):", correct_predictions)
print("Số lượng dự đoán sai:", incorrect_predictions)
print("Recall:", recall)
print("F1-Score:", f_score)
```

3. Support Vector Machine (SVM):

3.1. Introduction to SVM:

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồi thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

3.2. Building a SVM model.

- **Import the necessary libraries to build the model**

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

- **Read Datasets**

```
# Đọc dữ liệu từ file Excel
file_path = "D:/SIC/dataset.xlsx"
data = pd.read_excel(file_path, engine='openpyxl')
```

- **Build model**

```
# Chuẩn bị văn bản và nhãn
texts = data['Comment'].astype(str).fillna('')
labels = data['Label_new'].fillna(0).astype(int)  # Chuyển nhãn thành số nguyên

# Sử dụng TF-IDF để chuyển đổi văn bản thành vector
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(texts)

# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.2, random_state=42)

# Chuẩn hóa dữ liệu (không áp dụng mean vì dữ liệu dạng thưa)
scaler = StandardScaler(with_mean=False)
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Huấn luyện mô hình
svc = SVC()
svc.fit(X_train, y_train)
```

- **Model evaluation**

```
# Đánh giá mô hình
accuracy = svc.score(X_test, y_test)
print(f'Accuracy: {accuracy}')
```

4. K-nearest neighbors (KNN):

4.1. Introduction to KNN:

K-nearest neighbor là một trong những thuật toán supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng

được vào cả hai loại của bài toán Supervised learning là Classification và Regression. KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.

4.2. Building a KNN model.

- **Import the necessary libraries to build the model**

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
import numpy as np
```

- **Read Datasets**

```
# Đọc dữ liệu từ file Excel
file_path = "D:/SIC/dataset.xlsx"
data = pd.read_excel(file_path, engine='openpyxl')
```

- **Build model**

```
# Sử dụng TF-IDF để chuyển đổi văn bản thành vector
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(texts)

# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.2, random_state=42)

# Huấn luyện mô hình K-Nearest Neighbors (KNN)
clf = KNeighborsClassifier(n_neighbors=5)  # Bạn có thể điều chỉnh số lượng k theo nhu cầu
clf.fit(X_train, y_train)
```

- **Model evaluation**

```
from sklearn.metrics import recall_score, f1_score, accuracy_score
y_pred = clf.predict(X_test)
accuracy = clf.score(X_test, y_test)
print(f'Độ chính xác: {accuracy:.3f}')
correct_predictions = accuracy_score(y_test, y_pred, normalize=False)
recall = recall_score(y_test, y_pred, average='weighted')
f_score = f1_score(y_test, y_pred, average='weighted')
incorrect_predictions = len(y_test) - correct_predictions
print("Số lượng dự đoán chính xác (Prediction):", correct_predictions)
print("Số lượng dự đoán sai:", incorrect_predictions)
print("Recall:", recall)
print("F1-Score:", f_score)
```

5. Comparison between algorithms

Kết quả đánh giá mô hình:

	Random Forest	Logistic Regression	KNN	SVM
Recall	0.923	0.814	0.917	0.918
F_score	0.901	0.817	0.906	0.893
Accuracy	92%	81,5%	91,7%	91,8%
Thời gian huấn luyện	175.6s	20.1s	6.5s	495.5s

3. Results

3.1. Data Ingestion Scripts and Code

Trong phần này, chúng em mô tả quy trình nạp dữ liệu, bao gồm việc thu thập và nhập dữ liệu từ sàn thương mại điện tử Lazada. Đối với dự án này, dữ liệu chủ yếu là các bình luận của khách hàng bằng tiếng Việt, được thu thập từ nền tảng thương mại điện tử Lazada. Vì dữ liệu này không có cấu trúc rõ ràng nên cần được tiền xử lý đáng kể khi sử dụng cho phân tích.

*Lấy dữ liệu từ Lazada

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.options import Options
from selenium.common.exceptions import NoSuchElementException,ElementNotInteractableException
from selenium.webdriver.common.by import By
import sys
from time import sleep
import numpy as np
import pandas as pd
import random
import time

# Cấu hình lại mã hóa đầu ra để hỗ trợ Unicode
sys.stdout.reconfigure(encoding='utf-8')
from selenium import webdriver


# # Open URL
service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service)
driver.get('https://www.lazada.vn/catalog/?spm=a2o4n.homepage.search.d_go&q=kh%C4%83n%20gi%E1%BA%A5y')
```

```
# ================================ GET link/title
elems = driver.find_elements(By.CSS_SELECTOR , ".RfADt [href]")
title = [elem.text for elem in elems]
links = [elem.get_attribute('href') for elem in elems]

# ================================ GET price
elems_price = driver.find_elements(By.CSS_SELECTOR , ".aBrP0")
len(elems_price)
price = [elem_price.text for elem_price in elems_price]

df1 = pd.DataFrame(list(zip(title, price, links)), columns = ['title', 'price','link_item'])
df1['index_']= np.arange(1, len(df1) + 1)

# ================================GET discount


elems_discountPercent = driver.find_elements(By.CSS_SELECTOR , ".WNoq3 .IcOsH")
discountPercent = [elem.text for elem in elems_discountPercent]

discount_idx, discount_percent_list = [], []
for i in range(1, len(title)+1):
    try:
        discount_percent = driver.find_element(By.XPATH, "/html/body/div[3]/div/div[3]/div[1]/div/div[1]/div[2]/div[{}]/div/div/div[2]
        discount_percent_list.append(discount_percent.text)
        print(i)
        discount_idx.append(i)
    except NoSuchElementException:
        print("No Such Element Exception " + str(i))

df2 = pd.DataFrame(list(zip(discount_idx , discount_percent_list)), columns = ['discount_idx','discount_percent_list'])

df3 = df1.merge(df2, how='left', left_on='index_', right_on='discount_idx')
#print(df3)

# ================================ GET location/countReviews
elems_countReviews = driver.find_elements(By.CSS_SELECTOR , "._6uN7R")
countReviews = [elem.text for elem in elems_countReviews]

df3['countReviews'] = countReviews
file_name = f'Giay.csv'
df3.to_csv(file_name, index=False, encoding='utf-8-sig')
# ================================ GET more infor of each item


def getDetailItems(link):
    driver.get(link)
    count = 1
    driver.execute_script("window.scrollTo(0, 500)")
    sleep(random.randint(2,4))
    driver.execute_script("window.scrollTo(500, 1200)")
    sleep(random.randint(2,4))

    name_comment, content_comment, skuInfo_comment, like_count, star_count = [], [], [], [], []
    while True:
        try:

            print("Crawl Page " + str(count))
            try:
                ban = driver.find_element(By.XPATH, '/html/body/div/div[2]/div/div[1]')
                if not ban.is_enabled():
                    print("Bị ban rồi nè")
                    sleep(60)
            except  NoSuchElementException:
                print("")
            driver.execute_script("window.scrollTo(1200, 1800)")
            sleep(random.randint(2,4))
            driver.execute_script("window.scrollTo(1800, 2500)")
```

```
        sleep(random.randint(2,4))
        elems_star = driver.find_elements(By.CSS_SELECTOR, ".mod-reviews > .item > .top > .left")
        for elem in elems_star:
            stars = elem.find_elements(By.TAG_NAME, "img")
            stars_text = ""
            for star in stars:
                img_src = star.get_attribute("src")
                if 'https://laz-img-cdn.alicdn.com/tfs/TB19ZvEgfDH8KJjy1XcXXcpdXXa-64-64.png' in img_src:
                    stars_text += '★'
                elif 'https://laz-img-cdn.alicdn.com/tfs/TB18ZvEgfDH8KJjy1XcXXcpdXXa-64-64.png' in img_src:
                    stars_text += '☆'
            star_count.append(stars_text)

        elems_name = driver.find_elements(By.CSS_SELECTOR , ".middle")
        name_comment = [elem.text for elem in elems_name] + name_comment

        elems_content = driver.find_elements(By.CSS_SELECTOR , ".item > .item-content > .content")
        content_comment = [elem.text for elem in elems_content] + content_comment

        elems_skuInfo= driver.find_elements(By.CSS_SELECTOR , ".item-content .skuInfo")
        skuInfo_comment = [elem.text for elem in elems_skuInfo] + skuInfo_comment
```

```
        elems_likeCount = driver.find_elements(By.CSS_SELECTOR , ".item > .item-content > .bottom > .left > .left-content")
        like_count = [elem.text for elem in elems_likeCount] + like_count

        try:

            next_pagination_cmt = driver.find_element(By.XPATH, "/html/body/div[4]/div/div[10]/div[1]/div[2]/div/div/div/div[3]
            if not next_pagination_cmt.is_enabled():
                print("Nút 'next' không khả dụng. Dừng lại.")
                break
            next_pagination_cmt.click()
            print("Đã click vào nút trang tiếp theo!")
            time.sleep(random.randint(1, 3))
            count += 1
        except NoSuchElementException:
            print("Không tìm thấy nút 'next'. Dừng lại.")
            break
    except ElementNotInteractableException:
        print("Element Not Interactable Exception!")
        break

df4 = pd.DataFrame(list(zip(name_comment , content_comment, skuInfo_comment, like_count, star_count)),
                   columns = ['name_comment', 'content_comment','skuInfo_comment', 'like_count', 'star'])
```

```
    df4.insert(0, "link_item", link)
    del name_comment, content_comment, skuInfo_comment, like_count, star_count
    return df4

df_list = []
for i, link in enumerate(links):
    try:
        df = getDetailItems(link)
        df_list.append(df)

        # Lưu DataFrame vào file CSV với tên dựa trên thứ tự
        file_name = f'comments_Giay_{i + 1}.csv'
        df.to_csv(file_name, index=False, encoding='utf-8-sig')
        print(f"Dữ liệu từ {link} đã được lưu vào file {file_name}")
    except IndexError as e:
        print(f"Lỗi: {e}. Vị trí: {i}.")
        break

driver.quit()
```

3.2. Data Transformation Scripts and Code

Dữ liệu sau khi được thu thập cần được biến đổi để phù hợp với mô hình học máy. Quy trình biến đổi dữ liệu bao gồm xử lý lỗi chính tả, chuẩn hóa từ ngữ, loại bỏ các từ không có ý nghĩa (stop words), và vector hóa dữ liệu. Phần xử lý lỗi chính tả, chuẩn hóa từ hay loại bỏ các từ không ý nghĩa được làm một cách thủ công. Do các công cụ đã không còn hỗ trợ hoặc đã ngừng cập nhật dữ liệu rất nhiều năm về trước.

*** Làm sạch dữ liệu**

```
import pandas as pd
!pip install xlrd
!pip install openpyxl
```

Import data

```
df_=pd.read_csv(r"F:\Nam\CODE\SIC 2024\Project\Cleaning\predata.csv",encoding='utf-8')
df=df_.copy()
df.info()
```

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 360212 entries, 0 to 360211
Data columns (total 6 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   Title          360211 non-null  object
 1   Price          336684 non-null  object
 2   CountPreview   354959 non-null  object
 3   NameComment    360103 non-null  object
 4   Comment        360211 non-null  object
 5   Star           336845 non-null  object
dtypes: object(6)
memory usage: 16.5+ MB
```

```
df.head()
```

Python

	Title	Price	CountPreview	NameComment	Comment	Star
0	"Test Kỹ" Tai Nghe Samsung Galasy J5 J7 Dùng c...	16000	2.4K Đã bán\n(406)\nHồ Chí Minh	0***2Chứng nhận đã mua hàng	nó bị giật điên khi đeo vào tai	NaN
1	"Test Kỹ" Tai Nghe Samsung Galasy J5 J7 Dùng c...	16000	2.4K Đã bán\n(406)\nHồ Chí Minh	*******668Chứng nhận đã mua hàng		★★★★☆
2	"Test Kỹ" Tai Nghe Samsung Galasy J5 J7 Dùng c...	16000	2.4K Đã bán\n(406)\nHồ Chí Minh	*******028Chứng nhận đã mua hàng		★★★★★

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 319673 entries, 0 to 319672
Data columns (total 6 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   Title          319673 non-null  object
 1   Price          317328 non-null  object
 2   CountPreview   316621 non-null  object
 3   NameComment    319609 non-null  object
 4   Comment        228473 non-null  object
 5   Star           298453 non-null  object
dtypes: object(6)
memory usage: 14.6+ MB
```

```
df1=df.drop_duplicates(subset=["NameComment","Comment","Star"])
```

```
df2 = df1.dropna(subset=['Star'])
```

```
df2.isna().sum()
```

```
Title             0
Price          7700
CountPreview   1717
NameComment       0
Comment           0
Star              0
dtype: int64
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 76410 entries, 1 to 348137
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Title         76410 non-null  object
 1   Price         68710 non-null  object
 2   CountPreview  74693 non-null  object
 3   NameComment   76410 non-null  object
 4   Comment       76410 non-null  object
 5   Star          76410 non-null  object
dtypes: object(6)
memory usage: 4.1+ MB
```

Handle star numbers

```
# Filter out rows with invalid characters in 'Star' column
def filter_invalid_stars(df):
    return df[df['Star'].str.match(r'^[\s★☆]*$')]

# Apply function to a copy of df2
df2_filtered = filter_invalid_stars(df2.copy())
```

```
df2_filtered['StarCount'] = df2_filtered['Star'].str.count('★')
```

```
df2_filtered.drop('Star', axis=1, inplace=True)
```

```
df2_filtered.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 75293 entries, 1 to 348137
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Title         75293 non-null  object
 1   Price         67593 non-null  object
 2   CountPreview  74445 non-null  object
 3   NameComment   75293 non-null  object
 4   Comment       75293 non-null  object
 5   StarCount     75293 non-null  int64 
dtypes: int64(1), object(5)
memory usage: 4.0+ MB
```

+ Code + Markdown

```
df2_filtered.head()
```

	Title	Price	CountPreview	NameComment	Comment	StarCount
1	"Test Kỹ" Tai Nghe Samsung Galasy J5 J7 Dùng c...	16000	2.4K Đã bán\n(406)\nHồ Chí Minh	*******668Chứng nhận đã mua hàng		4
2	"Test Kỹ" Tai Nghe Samsung Galasy J5 J7 Dùng c...	16000	2.4K Đã bán\n(406)\nHồ Chí Minh	*******028Chứng nhận đã mua hàng		5
3	"Test Kỹ" Tai Nghe Samsung Galasy J5 J7 Dùng c...	16000	2.4K Đã bán\n(406)\nHồ Chí Minh	bé V.Chứng nhận đã mua hàng		4
5	"Test Kỹ" Tai Nghe Samsung Galasy J5 J7 Dùng c...	16000	2.4K Đã bán\n(406)\nHồ Chí Minh	Truong L.Chứng nhận đã mua hàng		4
6	"Test Kỹ" Tai Nghe Samsung Galasy J5 J7 Dùng c...	16000	2.4K Đã bán\n(406)\nHồ Chí Minh	H***.Chứng nhận đã mua hàng		5

Filter special characters

```
import pandas as pd
import re

# Function to filter special characters for Vietnamese text
def filter_vietnamese_specials(text):
    """
    Filters special characters while preserving Vietnamese characters.

    Args:
        text: The text to be filtered.

    Returns:
        The filtered text with special characters removed.
    """
    # Regular expression to match special characters except Vietnamese characters
    special_chars_pattern = r"[^\w\sÀÁÃẢẠẰẮẴẲẶẦẤẪẨẬÈÉẼẺẸỀẾỄỂỆÌÍĨỈỊÒÓÕỎỌỒỐỖỔỘỜỚỠỞỢÙÚŨỦỤỪỨỮỬỰỲỴỶỸĐđ]"
    return re.sub(special_chars_pattern, " ", text)

# Apply the filtering function to the "Comment" column only
df3=df2_filtered.copy()
df3['Comment'] = df2_filtered['Comment'].apply(filter_vietnamese_specials)
```

*Loại bỏ Stop-Word và gán nhãn Comment

```python
import pandas as pd
import numpy as np
import nltk
from nltk.corpus import stopwords
import requests
import openpyxl
```

Loại bỏ StopWords

```python
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

```python
vietnamese_stopwords = ["à", "ạ", "á", "ấy", "cũng", "đấy", "đó", "và", "của", "là", "lắm", "rất", "rồi", "với", "bị", "nhé", "nha", "nhá", "nếu", "
```

Python

```python
stop_words.update(vietnamese_stopwords)
```

Python

```python
def remove_stop_words(comment):
    comment = str(comment).lower()

    words = comment.split()

    filtered_words = [word for word in words if word not in stop_words]

    filtered_comment = " ".join(filtered_words)

    return filtered_comment
```

[5]

```python
file_path = "D:/Visual Studio Code/CrawlData(Selenium-Tab1)/data_cleaning.xlsx"
data = pd.read_excel(file_path,engine='openpyxl')
data=data.replace(np.nan, '', regex=True)
```

```python
def get_label(star_count):
    if star_count in [4, 5]:
        return "Tốt"
    elif star_count == 3:
        return "Trung bình"
    else:
        return "Không tốt"
```

2]

```python
data['Label'] = data['StarCount'].apply(get_label)
```

3]

```python
data['Filtered_Comment'] = data['Comment'].apply(remove_stop_words)
```

7]

```python
data.to_excel("D:/Visual Studio Code/CrawlData(Selenium-Tab1)/dataset_filtered.xlsx", index=False)
```

***Vector hóa dữ liệu**

Chúng em sử dụng TF-IDF để vector hóa dữ liệu vì nó giúp xác định các từ quan trọng trong bình luận bằng cách giảm thiểu ảnh hưởng của các từ thông dụng, đồng thời tăng trọng số cho những từ mang ý nghĩa

đặc biệt trong ngữ cảnh cụ thể. Giúp cải thiện độ chính xác của mô hình khi phân tích và phân loại bình luận.

```
# Chuẩn bị văn bản và nhãn
texts = data['Comment'].astype(str)
texts = [text if pd.notna(text) else '' for text in texts]
labels = data['Label_new']
labels = [star if pd.notna(star) else '' for star in labels]

# Sử dụng TF-IDF để chuyển đổi văn bản thành vector
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(texts)
```

3.3. Description and Sample of Transformed Datasets

Cột Comment bên trái là cột do người dùng nhập, dữ liệu này được lấy từ những sản phẩm trên trang thương mại điện tử Lazada. Cột Filtered_Comment bên phải là cột đã được lọc bỏ đi những ký tự đặc biệt, những từ không mang ý nghĩa và không ảnh hưởng đến ý nghĩa của câu. Ngoài ra, những hàng có bình luận trùng lắp cũng sẽ bị loại khỏi dataset. Những chữ cái viết hoa hoặc những khoảng trắng lớn cũng sẽ được xử lý về dạng chữ cái thường, mỗi từ chỉ cách nhau một khoảng trắng. Ban đầu, thu thập dữ liệu từ Lazada khoảng 320.000 dòng, sau khi thực hiện bước loại bỏ giá trị trùng lắp và bỏ đi những dòng bình luận trống thì dataset còn 65.000 dòng.

	Comment	Filtered_Comment
2	hàng tốt cửa hàng phục vụ uy tín chất lượng chạy từ Bắc xuống Nam xước như thế này là ổn	hàng tốt cửa hàng phục vụ uy tín chất lượng chạy từ bắc xuống nam xước như thế ổn
3	Giống mô tả lần sau mua tiếp	giống mô tả lần sau mua tiếp
4	hàng đẹp chất lượng tốt	hàng đẹp chất lượng tốt
5	Giao hơi lâu nhưng chất lượng	giao hơi lâu nhưng chất lượng
6	kiếm tốt vững nhưng phải lắp ráp tuy vậy nhưng vẫn thích	kiếm tốt vững nhưng phải lắp ráp tuy vậy nhưng vẫn thích
7	đẹp đóng gói cẩn thận giao hàng nhanh rất đáng tiền	đẹp đóng gói cẩn thận giao hàng nhanh đáng tiền
8	đúng hình rất đẹp rất chắc nhưng rất khó đút vào bao mong các sản phẩm sau có thể khắc phục được	đúng hình đẹp chắc nhưng khó đút vào bao mong các sản phẩm sau có thể khắc phục được
9	kiếm xịn đó	kiếm xịn
10	Kiếm rất đẹp không có gì để chê cho cửa hàng 5 sao	kiếm đẹp không có gì để chê cho cửa hàng 5 sao
11	hàng rất tốt chỉ tiếc là không phải màu đen	hàng tốt chỉ tiếc không phải màu đen
12	hàng tốt chất lượng cửa hàng tư vấn nhiệt tình giao hàng nhanh	hàng tốt chất lượng cửa hàng tư vấn nhiệt tình giao hàng nhanh
13	Đẹp Trưng bày là hợp lý	đẹp trưng bày hợp lý
14	kiếm đẹp và nhẹ nữa chất lượng tốt	kiếm đẹp nhẹ chất lượng tốt
15	Hàng đẹp không lỗi lầm gì Nhưng mà giao hàng hơi lâu mong cửa hàng lần sau giao nhanh hơn chút	hàng đẹp không lỗi lầm gì nhưng giao hàng hơi lâu mong cửa hàng lần sau giao nhanh hơn chút
16	rất đẹp nha gửi nhầm màu rồi lúc đầu đặt màu đen mà	đẹp gửi nhầm màu lúc đầu đặt màu đen
17	Chất liệu gỗ cao cấp Kiểm đồ chơi tuyệt vời cho trẻ em	chất liệu gỗ cao cấp kiểm đồ chơi tuyệt vời cho trẻ em
18	kiểm ổn cửa hàng tư vấn nhiệt tình nói chung khá là tốt	kiểm ổn cửa hàng tư vấn nhiệt tình nói chung khá tốt
19	hàng bị bong tróc nước sơn lỗi không nên mua	hàng bong tróc nước sơn lỗi không nên mua
20	trên cả tuyệt vời cửa hàng giao đạt tiêu chuẩn đẹp nhiệt tình nữa	trên cả tuyệt vời cửa hàng giao đạt tiêu chuẩn đẹp nhiệt tình
21	kiếm bị bong tróc	kiếm bong tróc
22	kiếm đẹp và dễ lắp ráp nha anh em nên mua về trưng với mô hình	kiếm đẹp dễ lắp ráp anh em nên mua về trưng mô hình
23	một cây kiếm gỗ đúng nghĩa đừng nghe thằng ở trên dụ là gửi kiếm thật không có đâu	một cây kiếm gỗ đúng nghĩa đừng nghe thằng ở trên dụ gửi kiếm thật không có đâu
24	kiếm không đúng màu lưỡi kiếm mỏng	kiếm không đúng màu lưỡi kiếm mỏng
25	quá tuyệt vời cửa hàng rất chiều khách nha	tuyệt vời cửa hàng chiều khách
26	hàng tốt giống hình	hàng tốt giống hình
27	kiếm đẹp giao hàng nhanh tốt phù hợp giá tiền dùng để trang trí đẹp lắm	kiếm đẹp giao hàng nhanh tốt phù hợp giá tiền dùng để trang trí đẹp
28	Rất đáng tiền nên mua độ bền rất chắc chắn	đáng tiền nên mua độ bền chắc chắn

4. Projected Impact

4.1. Accomplishments and Benefits

Accomplishments

- Chúng em đã thu thập và xử lý thành công một lượng lớn dữ liệu từ các bình luận sản phẩm bằng tiếng Việt, một nhiệm vụ không hề dễ dàng do đặc thù ngôn ngữ và sự đa dạng trong cách diễn đạt của người dùng. Việc xử lý đã giúp nhóm nâng cao kỹ năng xử lý dữ liệu thực tế, đồng thời làm phong phú thêm kho dữ liệu để phân tích.
- Dự án đã giúp chúng em có thêm những kiến thức mới, bước ra khỏi vùng an toàn của bản thân. Dự án là nền tảng để chúng em đạt thêm nhiều thành tựu từ các khóa học khác hoặc trong cuộc sống

Benefits

- Một trong những lợi ích lớn nhất của dự án là khả năng giúp người tiêu dùng đưa ra quyết định mua sắm thông minh hơn. Thay vì phải đọc qua hàng ngàn bình luận, hệ thống của chúng em sử dụng các thuật toán machine learning tiên tiến như SVM, Random Forest, và Logistic Regression để phân loại và đánh giá chất lượng sản phẩm dựa trên những bình luận thực tế. Điều này không chỉ tiết kiệm thời gian cho người tiêu dùng mà còn tăng cường độ tin cậy trong việc lựa chọn sản phẩm phù hợp với nhu cầu.
- Ngoài ra, dự án còn đem lại nhiều lợi ích khác cho các nhà bán lẻ và nhà sản xuất. Thông qua việc phân tích và hiểu rõ hơn về phản hồi của khách hàng, họ có thể điều chỉnh chiến lược kinh doanh, cải thiện chất lượng sản phẩm, và tối ưu hóa dịch vụ chăm sóc khách hàng. Khả năng xử lý Big Data của dự án cũng mở ra cơ hội để các doanh nghiệp áp dụng những phân tích sâu hơn, phát hiện xu hướng tiêu dùng và dự đoán nhu cầu thị trường một cách chính xác hơn.

4.2. Future Improvements

- Chúng em mong muốn cải tiến chương trình hơn là có thể tự động thu thập được tất cả bình luận của sản phẩm khi người dùng nhập liên kết của sản phẩm, được hợp tác với các sàn thương mại điện tử

sẽ giúp chúng em thực hiện điều này một cách dễ dàng hơn.

- Bên cạnh đó, việc xử lý dữ liệu tiếng Việt, đặc biệt là các bình luận có chứa lỗi chính tả, viết tắt, hoặc tiếng lóng, vẫn còn là một thách thức lớn. Hiện tại, phần lớn quá trình này vẫn yêu cầu xử lý thủ công trong dự án lần này vì hầu như các công cụ đã ngừng hỗ trợ hoặc cập nhật ở nhiều năm trước. Do đó, một trong những cải tiến chính trong tương lai là phát triển và tích hợp các công cụ xử lý ngôn ngữ tự nhiên (NLP) mạnh mẽ hơn, giúp tự động hóa quá trình này một cách hiệu quả hơn. Điều này không chỉ giúp nâng cao độ chính xác của mô hình mà còn giảm bớt thời gian và công sức xử lý dữ liệu.
- Trong tương lai, chúng em sẽ thử nghiệm và triển khai thêm các mô hình machine learning tiên tiến khác như deep learning, đặc biệt là mạng neuron sâu (Deep Neural Networks), để cải thiện khả năng phân tích và dự đoán. Việc sử dụng các mô hình phức tạp hơn có thể giúp chúng em khai thác sâu hơn những thông tin ẩn chứa trong dữ liệu, từ đó đưa ra những dự đoán chính xác hơn.
- Hiện tại, hệ thống của chúng em chủ yếu phân tích các bình luận về sản phẩm bằng tiếng Việt. Trong tương lai, chúng em sẽ hướng đến việc mở rộng khả năng xử lý và phân tích bình luận bằng nhiều ngôn ngữ khác nhau, giúp dự án có tính ứng dụng toàn cầu

5. Team Member Review and Comment

<ATTACH A TEAM PICTURE HERE>

NAME	REVIEW and COMMENT

6. Instructor Review and Comment

CATEGORY	SCORE	REVIEW and COMMENT
IDEA	__/10	
APPLICATION	__/30	
RESULT	__/30	
PROJECT MANAGEMENT	__/10	
PRESENTATION & REPORT	__/20	

TOTAL	__/100	