معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Customer Review Analysis using AI

Presented By:
Nasser Alsharif
Hala Alqurashi
Rawabi Alharthi
Mohammed Alzanbahi

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Problem Statement

Problem 1

Thousands of product reviews are generated daily

Problem 2

Manual analysis is inefficient

Problem 3

Businesses need automated insight extraction.

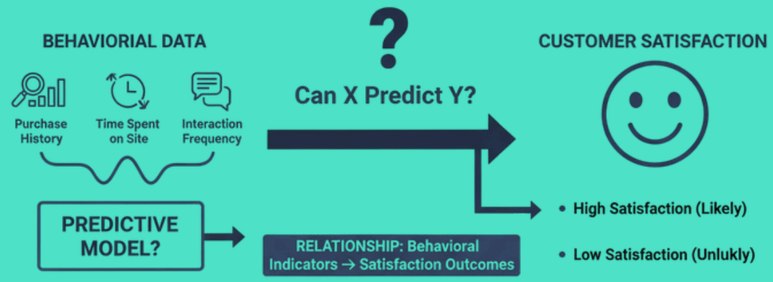
معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Research Question

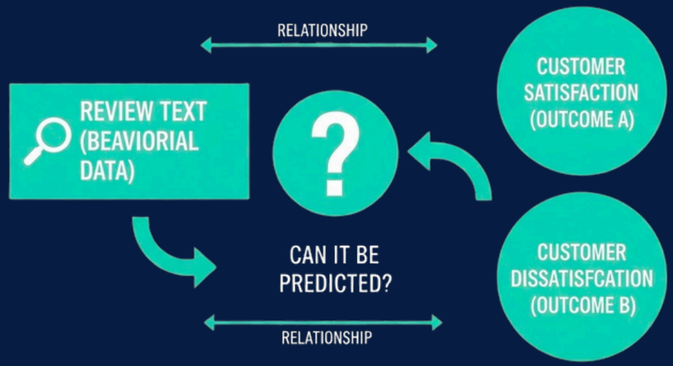
Can customer satisfaction be predicted using:

Sentiment signals

Behavioral indicators



Review text



معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Research Hypotheses

H0 (Null Hypothesis)

AI sentiment scores and behavioral metrics (text_length, helpful_ratio) have no significant relationship with customer satisfaction.

H1 (Alternative Hypothesis)

AI sentiment scores and behavioral metrics significantly predict and classify customer satisfaction levels.

H0 (Null Hypothesis)

Specific keywords do not differentiate between satisfied and dissatisfied customers.

H1 (Alternative Hypothesis)

Specific contextual keywords strongly differentiate between satisfied and dissatisfied customer

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Research Objectives

Examine the Relationship Between AI Signals and Customer Satisfaction

Identify Key Differentiating Features

Develop a Hybrid Predictive Model

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Dataset overview

- Amazon Fine Food Reviews
- Sample size: 2000
- Target variable: Customer Satisfaction (CR)

Loading Dataset from Kaggle code:

```
# Set the path to the file you'd like to load
# Replace 'Reviews.csv' with the actual file name from the dataset if it's different
file_path = 'Reviews.csv'

# Load the latest version
df = kagglehub.load_dataset(
  KaggleDatasetAdapter.PANDAS,
  "snap/amazon-fine-food-reviews",
  file_path,
  # Provide any additional arguments like
  # sql_query or pandas_kwargs. See the
  # documenation for more information:
  # https://github.com/Kaggle/kagglehub/blob/main/README.md#kaggledatasetadapterpandas
)
```

Extracting the sample code:

```
#We Take 2000 random samples
df_sample = df.sample(n=2000, random_state=42).copy()
```

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Data Cleaning & Feature Selection

Removed Features:

- ID
- USER ID
- PRODUCT ID
- PROFILE NAME
- TIME
- SUMMARY

Why?

- IRRELEVANT FOR PREDICTION
- HIGH CARDINALITY
- NO PREDICTIVE SIGNAL
- RISK OF NOISE OR OVERFITTING

Data Cleaning & Feature Selection code :

```
#حذف الاعمدة الغير مهمه
df_sample.drop(['Id', 'UserId',
                'Summary','ProductId','ProfileName',
                'Time'], axis=1, inplace=True, errors='ignore')
```

Data frame sample after :

	HelpfulnessNumerator	HelpfulnessDenominator	Score	Text
0	0	0	5	Having tried a couple of other brands of glute...
1	0	0	5	My cat loves these treats. If ever I can't fin...
2	0	2	3	A little less than I expected. It tends to ha...
3	0	1	2	First there was Frosted Mini-Wheats, in origin...
4	0	2	5	and I want to congratulate the graphic artist ...
...	...	...	...	...
1995	0	0	5	Received very timely, and packed safely; no bu...
1996	1	1	5	Absolutely love this candy. It's the best and...
1997	0	0	5	I have been having difficulty finding this ble...
1998	0	0	4	Arctic Zero is a tasty frozen treat with very ...
1999	4	4	5	Of course, my cats deserve only the best so I ...

2000 rows × 4 columns

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Feature Engineering

We transformed raw data into structured predictive features:

- **Helpful Ratio**

MEASURES HOW USEFUL THE REVIEW WAS TO OTHER USERS.

- **Text Length**

CAPTURES STRUCTURAL DIFFERENCES IN REVIEW WRITING.

- **Binary Target (CR)**

CONVERTED RATING SCORE INTO:

- 1 → SATISFIED
- 0 → NOT SATISFIED

Feature Engineering code :

```
# Helpful Ratio
df_sample["helpful_ratio"] = 0 #قيمة افتراضية

mask = df_sample["HelpfulnessDenominator"] > 0 #0 اخذ فقط الاصوات الي اكبر من

df_sample.loc[mask, "helpful_ratio"] = (
    df_sample.loc[mask, "HelpfulnessNumerator"] /
    df_sample.loc[mask, "HelpfulnessDenominator"]
)

# Text Lengh
df_sample["text_length"] = df_sample["Text"].str.len()

# CR
def customer_review(score):
    if score <= 3:
        return 0   # غير راضي
    else:
        return 1   # راضي

df_sample['CR'] = df_sample['Score'].apply(customer_review)
```

Data frame sample after :

	HelpfulnessNumerator	HelpfulnessDenominator	Score	Text	helpful_ratio	text_length	CR
0	0	0	5	Having tried a couple of other brands of glute...	0.0	485	1
1	0	0	5	My cat loves these treats. If ever I can't fin...	0.0	490	1
2	0	2	3	A little less than I expected. It tends to ha...	0.0	136	0
3	0	1	2	First there was Frosted Mini-Wheats, in origin...	0.0	1631	0
4	0	2	5	and I want to congratulate the graphic artist ...	0.0	649	1

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Sentiment Extraction using Transformer Model

We extracted sentiment signals from review text using a pre-trained transformer model.

- Used Hugging Face sentiment-analysis pipeline
- Classified each review as Positive or Negative
- Converted sentiment into binary feature (1 = Positive, 0 = Negative)
- Integrated sentiment output into the feature set

Hagging Face :

```
classifier = pipeline(
    "sentiment-analysis",
    device=0   # GPU
)
# التنبؤ بالمشاعر باستخدام المعالجة على دفعات لتسريع التنفيذ
results = []
BATCH_SIZE = 32
for i in tqdm(range(0, len(texts), 500)):
    batch = texts[i:i+500]
    results.extend(
        classifier(
            batch,
            truncation=True,
            batch_size=BATCH_SIZE
        )
    )
#  إضافة النتائج للـ DataFrame
df_sample["sentiment_label_HF"] = [r["label"] for r in results]
df_sample["sentiment_score_HF"] = [r["score"] for r in results]
# (للتحليل) تحويلها لقيم رقمية
df_sample["sentiment_HF"] = df_sample["sentiment_label_HF"].apply(
    lambda x: 1 if x == "POSITIVE" else 0
)
```

Data frame sample after :

	Text	sentiment_label_HF	sentiment_score_HF
0	Having tried a couple of other brands of glute...	POSITIVE	0.999240
1	My cat loves these treats. If ever I can't fin...	POSITIVE	0.997516
2	A little less than I expected. It tends to ha...	NEGATIVE	0.999433
3	First there was Frosted Mini-Wheats, in origin...	POSITIVE	0.937716
4	and I want to congratulate the graphic artist ...	POSITIVE	0.997636

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Custom Sentiment Feature Engineering

In addition to transformer-based sentiment analysis, we implemented a rule-based sentiment scoring approach:

- CLEANED REVIEW TEXT
- DEFINED CUSTOM POSITIVE AND NEGATIVE WORD LISTS
- COUNTED SENTIMENT-RELATED WORDS PER REVIEW
- ASSIGNED SENTIMENT LABEL BASED ON WORD DOMINANCE

This hybrid architecture improves robustness and generalization by combining rule-based and deep learning sentiment signals.

Sentiment Feature Engineering Code :

```
# تنظيف النصوص
stop_words = set(stopwords.words('english')).union({'br','one','would','product' , 'amazon'})
df_sample['clean_words'] = df_sample['Text'].apply(clean_text)

# إعداد محلل المشاعر
sia = SentimentIntensityAnalyzer()

# حساب عدد الكلمات الإيجابية والسلبية لكل مراجعة
positive_list = ['like', 'good', 'great', 'love', 'best']
negative_list = ['hard', 'bad', 'low', 'problem', 'bitter']

def count_sentiment_words(words):
    pos_count = sum(1 for w in words if w in positive_list)
    neg_count = sum(1 for w in words if w in negative_list)
    return pd.Series([pos_count, neg_count])

df_sample[['pos_count', 'neg_count']] = df_sample['clean_words'].apply(count_sentiment_words)

# تحديد النتيجة إيجابي/سلبي فقط (لا يوجد محايد)
def sentiment_label(row):
    if row['pos_count'] > row['neg_count']:
        return 1  # إيجابي
    elif row['neg_count'] > row['pos_count']:
        return 0  # سلبي
    else:
        return None  # متساوي أو صفر، يمكن حذفها لاحقًا

df_sample['sentiment_CR'] = df_sample.apply(sentiment_label, axis=1)
df_sample = df_sample.dropna(subset=['sentiment_CR'])  # حذف المحايد
```

	Text	pos_count	neg_count	sentiment_CR
0	Having tried a couple of other brands of glute...	2	0	1.0
1	My cat loves these treats. If ever I can't fin...	3	0	1.0
3	First there was Frosted Mini-Wheats, in origin...	2	0	1.0
6	I absolutely love Yorkshire tea and am so glad...	2	0	1.0
7	I have such a hard time finding loose tea loca...	0	1	0.0

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Text Vectorization (TF-IDF)

To convert unstructured text into numerical features, we applied TF-IDF (Term Frequency – Inverse Document Frequency)

- Extracted top 500 most informative words
- Removed common stopwords
- Converted text into structured numerical vectors
- Applied fit on training data and transform on test data

Sentiment Extraction code :

```
 #معالجة Text
stop_words = set(stopwords.words('english'))

def clean_text(text):
    text = str(text).lower()
    text = re.sub(r"http\S+", "", text)
    text = re.sub(r"[^a-zA-Z]", " ", text)
    words = text.split()
    words = [word for word in words if word not in stop_words]
    return words  # قائمة كلمات بدل نص

df_sample['clean_words'] = df_sample['Text'].apply(clean_text)

#  جمع كل الكلمات حسب CR
words_satisfied = [word for sublist in df_sample[df_sample['CR']==1]['clean_words'] for word in sublist]
words_unsatisfied = [word for sublist in df_sample[df_sample['CR']==0]['clean_words'] for word in sublist]
```

(TF-IDF)

```
from sklearn.feature_extraction.text import TfidfVectorizer

#نخلي المودل يركز على اهم 500 كلمة ويحطها كفيتشرز
tfidf = TfidfVectorizer(max_features=500, stop_words='english') # حذفنا الكلمات التي تعتبر بدون فايدة " (مثل a, an, in, of)


# المودل هنا "يتعلم" الكلمات المهمة من بيانات التدريب ويحولها لأرقام.
X_train_tfidf = tfidf.fit_transform(df_sample.loc[X_train.index, 'Text']).toarray()
# المودل هنا "يطبق" ما تعلمه فقط على بيانات الاختبار.
X_test_tfidf = tfidf.transform(df_sample.loc[X_test.index, 'Text']).toarray()


tfidf_train_df = pd.DataFrame(X_train_tfidf, index=X_train.index, columns=tfidf.get_feature_names_out())
tfidf_test_df = pd.DataFrame(X_test_tfidf, index=X_test.index, columns=tfidf.get_feature_names_out())

X_train_final = pd.concat([X_train, tfidf_train_df], axis=1)
X_test_final = pd.concat([X_test, tfidf_test_df], axis=1)

print(f"Number of features after TF-IDF: {X_train_final.shape[1]}")
```

```
Number of features after TF-IDF: 506
```

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Data Splitting & Final Feature Selection

Final Selected Features

We selected six structured features for training:

- text_length
- sentiment_HF
- helpful_ratio
- sentiment_score_HF
- pos_count
- neg_count

Train–Test Split Strategy

- 80% Training
- 20% Testing
- Fixed random state for reproducibility

Feature selection and Train/Test code :

```
#تحديد التارقت والفيتشرز
y = df_sample["CR"]
X = df_sample[[ "text_length", "sentiment_HF", "helpful_ratio", "sentiment_score_HF", "pos_count", "neg_count"]]
```

```
#تقسيم البيانات test and train
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
print("Training set shape:", X_train.shape)
print("Testing set shape:", X_test.shape)
```

```
Training set shape: (1088, 6)
Testing set shape: (273, 6)
```

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Data Scaling & Class Imbalance Handling

Feature Scaling

To normalize numerical features and reduce the impact of outliers, we applied:

- RobustScaler
- Fitted only on training data
- Transformed both training and test sets

Class Imbalance Handling

To prevent bias toward the majority class, we:

- Computed balanced class weights
- Adjusted model sensitivity toward minority class

Feature scaling and class imbalance code :

```
# ضبط المقاييس (Scaling) لمنع تسريب البيانات
rob_scaler = RobustScaler()
X_train[['text_length']] = rob_scaler.fit_transform(X_train[['text_length']])
X_test[['text_length']] = rob_scaler.transform(X_test[['text_length']])

#ال استخدمنا class weight عشان نعالج عدم التوازن
from sklearn.utils.class_weight import compute_class_weight
import numpy as np
weights = compute_class_weight('balanced', classes=np.unique(y_train), y=y_train)
print(f"Weights for (Not Satisfied 0): {weights[0]:.2f}")
print(f"Weights for (Satisfied 1): {weights[1]:.2f}")
```

```
Weights for (Not Satisfied 0): 2.76
Weights for (Satisfied 1): 0.61
```

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Model Training

To ensure robust performance, we trained multiple classification models and compared their results.

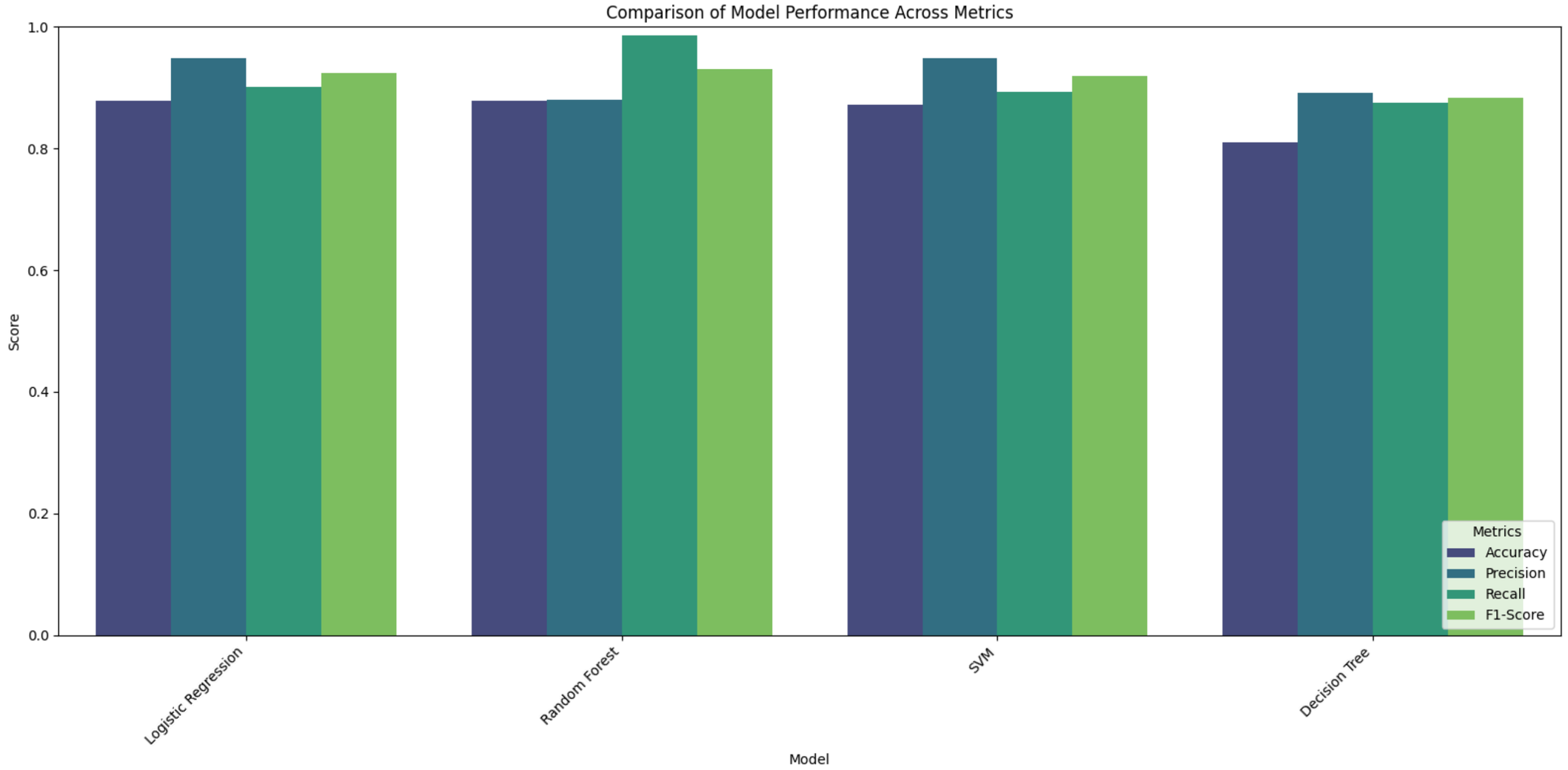
- Applied class balancing to handle imbalanced data
- Trained several algorithms
- Selected the best-performing model based on evaluation metrics

Model Training code :

```
sample_weights = compute_sample_weight(class_weight='balanced', y=y_train)

# 1. Support Vector Classifier
svm_model = SVC(class_weight='balanced', probability=True)
svm_model.fit(X_train_final, y_train)

# 2. K-Neighbors Classifier
knn_model = KNeighborsClassifier()
knn_model.fit(X_train_final, y_train)

# 3. Decision Tree
dt_model = DecisionTreeClassifier(class_weight='balanced')
dt_model.fit(X_train_final, y_train)

# 4. Logistic Regression
lr_model = LogisticRegression(class_weight='balanced', random_state=42)
lr_model.fit(X_train_final, y_train)

# 5. Passive Aggressive Classifier
pa_model = PassiveAggressiveClassifier(class_weight='balanced')
pa_model.fit(X_train_final, y_train)

# 6. Perceptron
per_model = Perceptron(class_weight='balanced')
per_model.fit(X_train_final, y_train)

# 7. Gaussian Naive Bayes
nb_model = GaussianNB()
nb_model.fit(X_train_final, y_train, sample_weight=sample_weights)

# 8. Multi-layer Perceptron (Neural Network)
ann_model = MLPClassifier(random_state=42)
ann_model.fit(X_train_final, y_train)

# 9. Random Forest
rf_model = RandomForestClassifier(class_weight='balanced', random_state=42)
rf_model.fit(X_train_final, y_train)
```

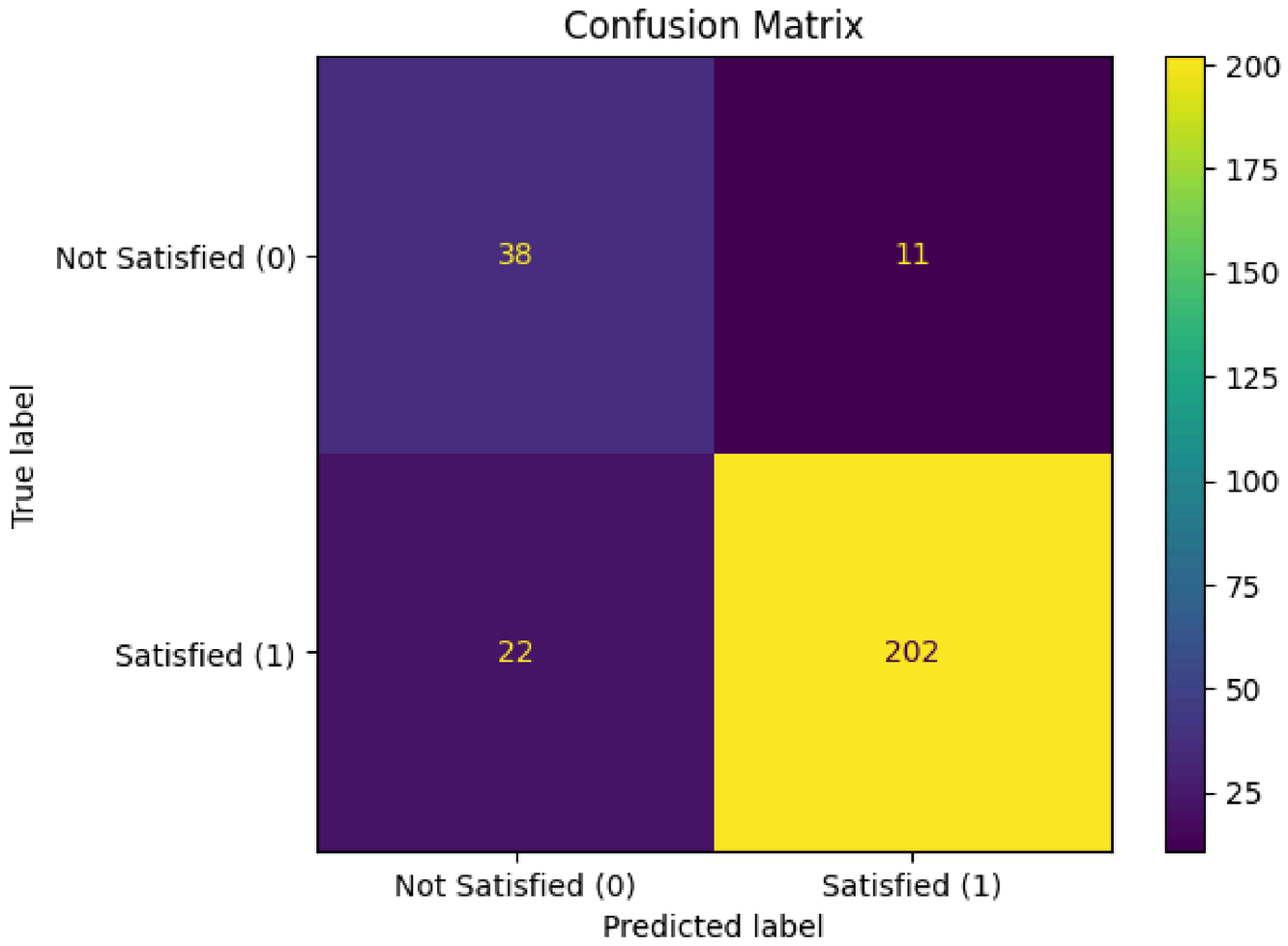
معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Model Evaluation



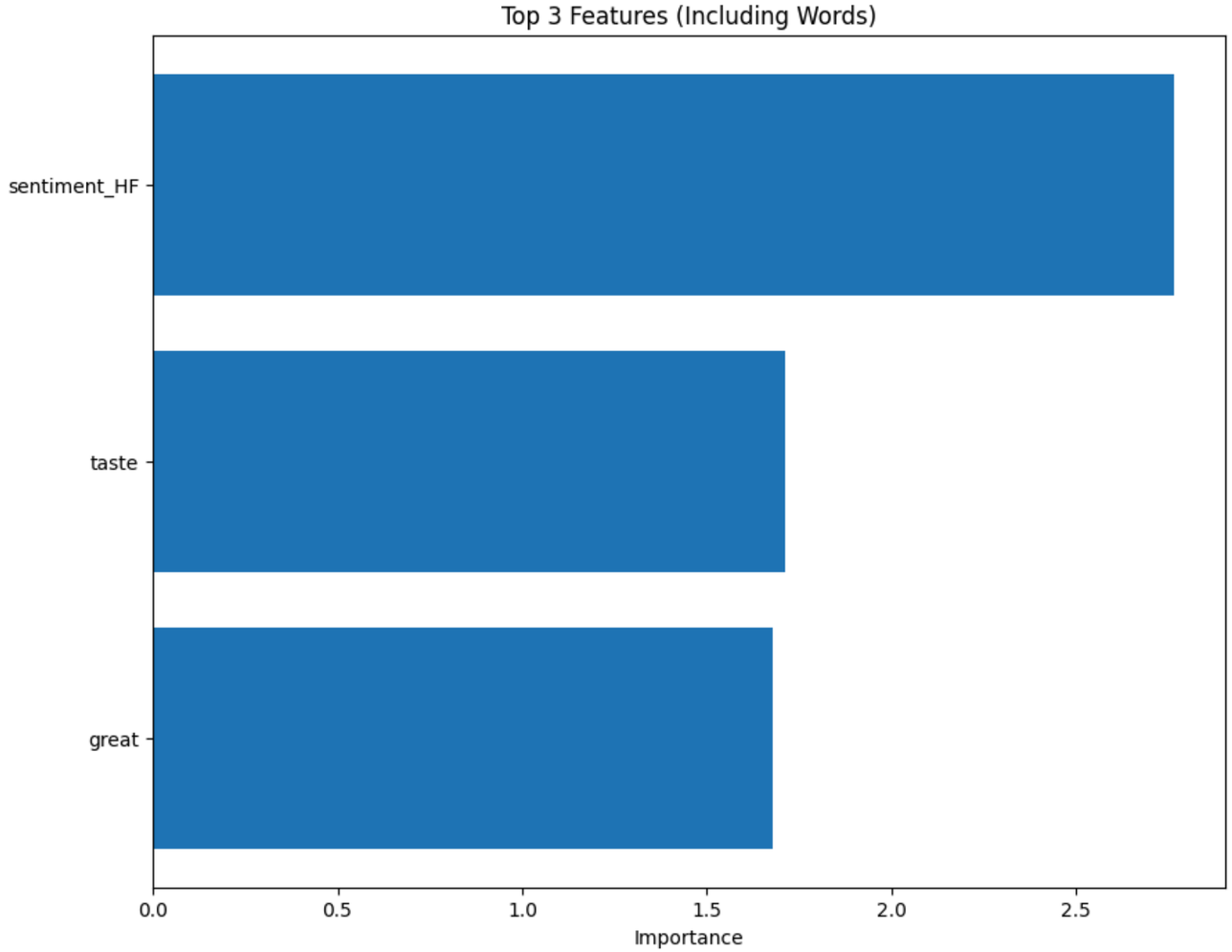
معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Logistic Regression Confusion Matrix



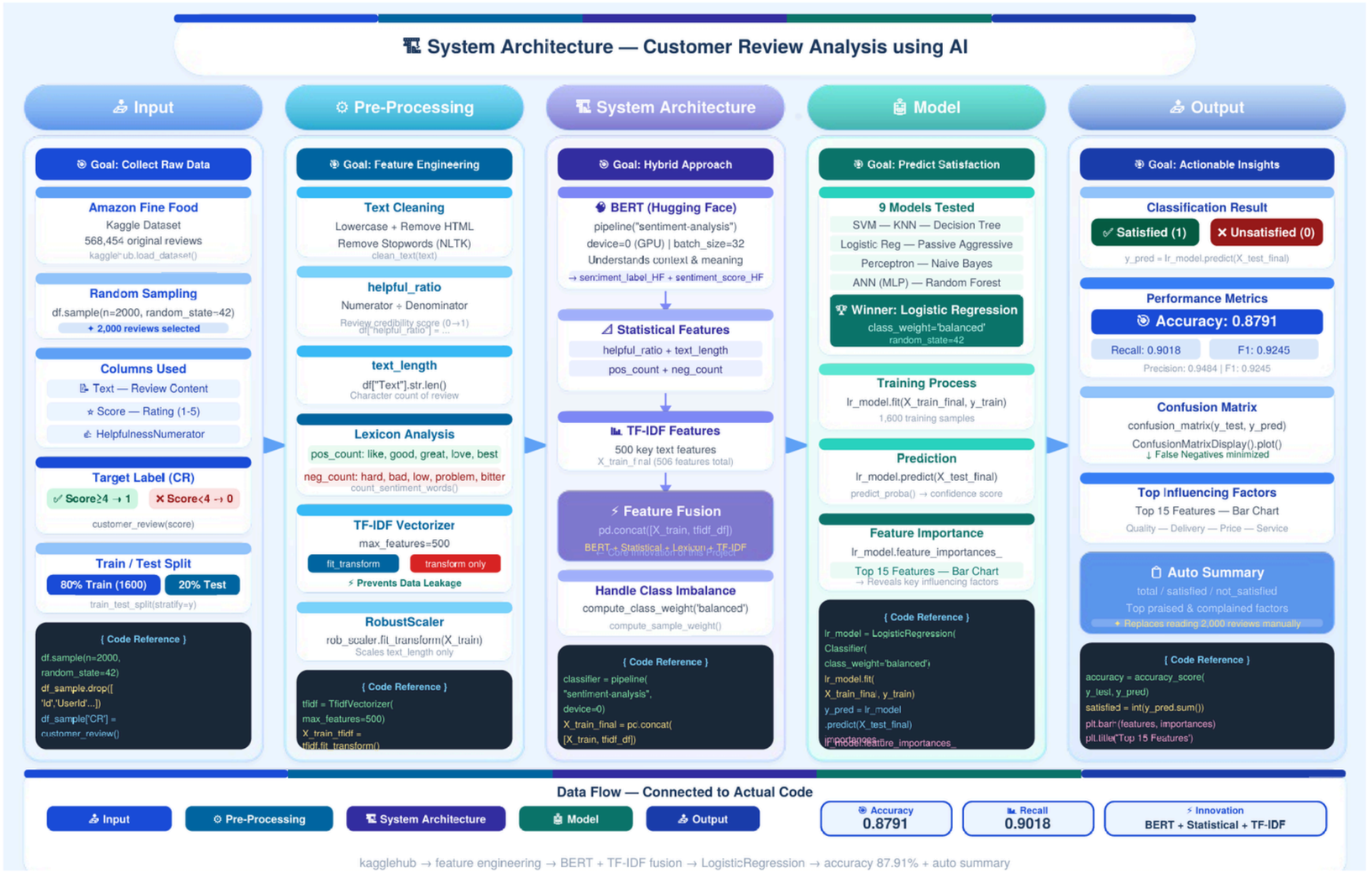
معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Top 3 Features



معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Final System Architecture Overview



معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Conclusion

Our project demonstrates that customer satisfaction can be effectively predicted using a hybrid approach combining behavioral indicators and advanced text analysis. By leveraging sentiment extraction, TF-IDF vectorization, and machine learning models, we were able to transform unstructured reviews into structured, actionable insights.
This system enables businesses to:

- Detect dissatisfied customers automatically
- Identify patterns in customer feedback
- Prioritize product improvements
- Support data-driven decision-making

Ultimately, this approach reduces manual effort and enhances strategic responsiveness in e-commerce environments.

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

Our Team

A collaborative effort combining data science and machine learning to build an intelligent review analysis system.

Mohammed Alzanbahi
Mohammed A.ALzanbahi

Nasser Alsharif
Nasser Alsharif

Rawabi Alharthi
Rawabi A.Alharthi

Hala Alqurashi
Hala Alqurashi

معهد طاقات المعتمد للتدريب
Taqat Certified Training Institute

THANK YOU
FOR LISTENING