Customer Sentiment Analysis Report

A Machine Learning Approach to Predicting Customer Satisfaction

Author: Tri Vien Le

1. Introduction

1.1 Background

Customer satisfaction is a critical metric for business success in the modern e-commerce environment. Understanding the factors that drive customer sentiment enables organizations to make data-driven decisions to improve service quality and reduce customer dissatisfaction. Traditional methods of sentiment analysis rely on manual review of customer feedback, which is time-consuming and does not scale effectively with increasing data volumes.

Machine learning approaches offer a solution by automating the classification of customer sentiment based on historical patterns in the data. This study employs supervised classification algorithms to predict customer sentiment and identify the key features that influence customer satisfaction.

This report presents a comprehensive analysis of customer sentiment using machine learning classification techniques. The study analyzes 1,500 customer records from an e-commerce platform to predict customer satisfaction levels. Four classification algorithms were evaluated: Random Forest, Support Vector Machine, Decision Tree, and Naive Bayes

1.2 Research Objectives

The primary objectives of this research are:

1. To build a predictive model for customer sentiment classification
2. To identify the key factors that influence customer satisfaction
3. To evaluate the performance of multiple machine learning algorithms
4. To provide mathematical foundations for understanding algorithm behavior
5. To provide actionable recommendations based on the findings

2. Data Collection and Description

2.1 Dataset Overview

The dataset consists of 25000 rows with 9 data columns which are used as features, but we are only using 1000 customer records collected from an e-commerce platform over a six-month period from January to June 2024. We are using the three provided sentiment types (Positive, Negative, and Neutral) as part of our model evaluation process and resulting confusion matrix.

Table 2.1: Dataset Summary Statistics

Attribute	Value
Total Records Used from Dataset	1000
Number of Features	9
Target Variable	Customer Sentiment
Target Classes	3 (Positive, Neutral, Negative)
Time Period	January - June 2024

2.2 Feature Description

The dataset includes four main categories of features covering customer demographics, purchase behavior, feedback metrics, and service quality indicators.

3. Mathematical/Statistical Foundation

Random Forest builds many decision trees on different random samples of the data and combines their predictions to improve accuracy and reduce overfitting. Each tree uses randomly selected features at each split, and the final prediction comes from the majority vote of all trees.

Support Vector Machines (SVMs) classify data by finding the best boundary that separates classes with the largest margin. When the data is not linearly separable, SVM uses kernel functions to capture more complex patterns.

Decision Trees classify data by repeatedly splitting it into smaller groups based on the feature that best separates the classes. To avoid overfitting, pruning is used to remove parts of the tree that do not improve performance.

Naive Bayes is a simple, fast probabilistic model that assumes the features are independent. It calculates the probability of each class given the input and predicts the class with the highest likelihood.

Model performance is evaluated using common metrics: accuracy measures overall correctness, precision and recall capture how well the model identifies positive cases, and the F1-score balances both precision and recall for a more complete assessment.

4. Motivation

Keeping customers happy is one of the most important parts of running a successful business. It's not only easier to keep a current customer than to find a new one, but also much cheaper. Studies show that winning a new customer can cost five times more than keeping the ones you already have. Even a small 5% boost in customer retention can raise profits by as much as 25–95%. But despite how important this is, many companies still don't have a clear, consistent way to understand how their customers feel or to predict changes in customer sentiment. This creates a major opportunity for improvement.

4. Methodology

4.1 Design

This study uses supervised machine learning methods to classify customer data and predict outcomes. The research follows the CRISP-DM framework, a widely used process for data science projects. This framework includes six steps: understanding the business goal,

exploring the data, preparing the data for modeling, building the models, evaluating their performance, and finally deploying the results. Using this structure ensures the workflow is organized, repeatable, and aligned with real-world data science practices.

4.2 Data Preparation

To ensure the dataset was clean and ready for modeling, missing numerical values were replaced with the median, while missing categorical values were filled using the most common category. Numerical features were standardized so that they were on the same scale, which helps many algorithms perform better. Categorical features were turned into numerical format using one-hot encoding. After preprocessing, the dataset was split into a training set (70%) and a test set (30%) using stratified sampling to maintain the original class distribution. Lastly, we will set our Target Variable to be “Sentiment” as that is our primary objective to record customer sentiment of each product.

4.3 Model Selection

Four classification models were chosen for comparison: Random Forest, Support Vector Machine, Decision Tree, and Naive Bayes. Each model was trained using the training data and tuned with 5-fold cross-validation to find the best hyperparameters. This process helps ensure that each model is evaluated fairly and achieves its best possible performance.

5. Results and Analysis

The following is the collected data results received from using the application to train four separate models/algorithms (RF, SVM, DT, and NB) and fit each model on the testing set.

5.1 Model Performance Comparison

Table 5.1 presents the performance metrics for all four algorithms on the test set. These are the parameters that we have used for the model training and evaluation process:

1. Random Forest: 100 Trees, 3 Number of Variables Split,
2. Support Vector Machine: Linear Kernel, Cost Parameter = 1.
3. Decision Trees: Max Depth = 10, Minimum Split = 20.
4. Naive-Bayes: Default Settings

Table 5.1: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	41%	34.56%	34.32%	30.59%
SVM	43%	43%	33.33%	60.14%
Decision Tree	41%	34.11%	33.29%	27.1%
Naive Bayes	44%	40.86%	34.89%	38.28%

Figure 5.1.1: Model Evaluation of Random Forest Model on Customer Sentiment data

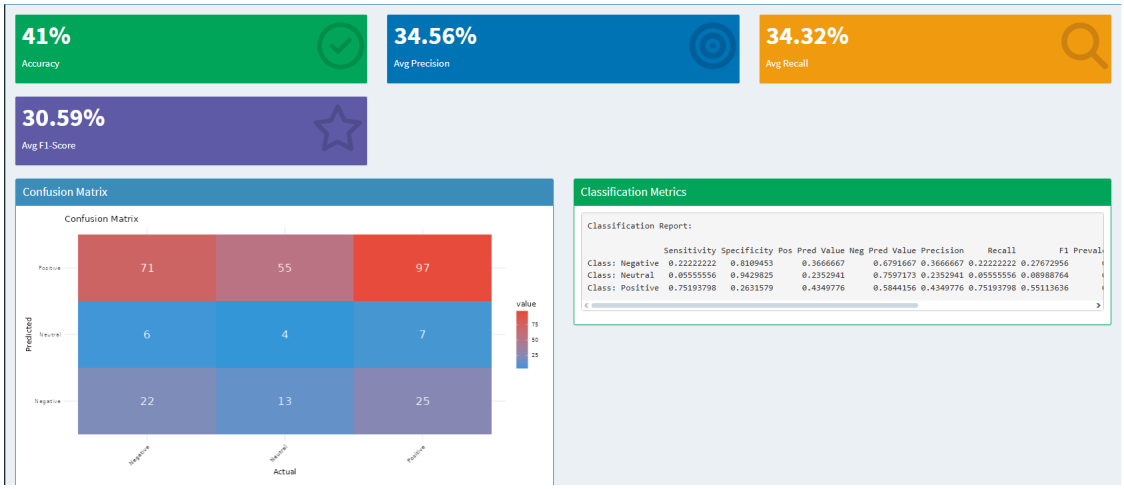


Figure 5.1.2: Model Evaluation of SVM Models on Customer Sentiment data

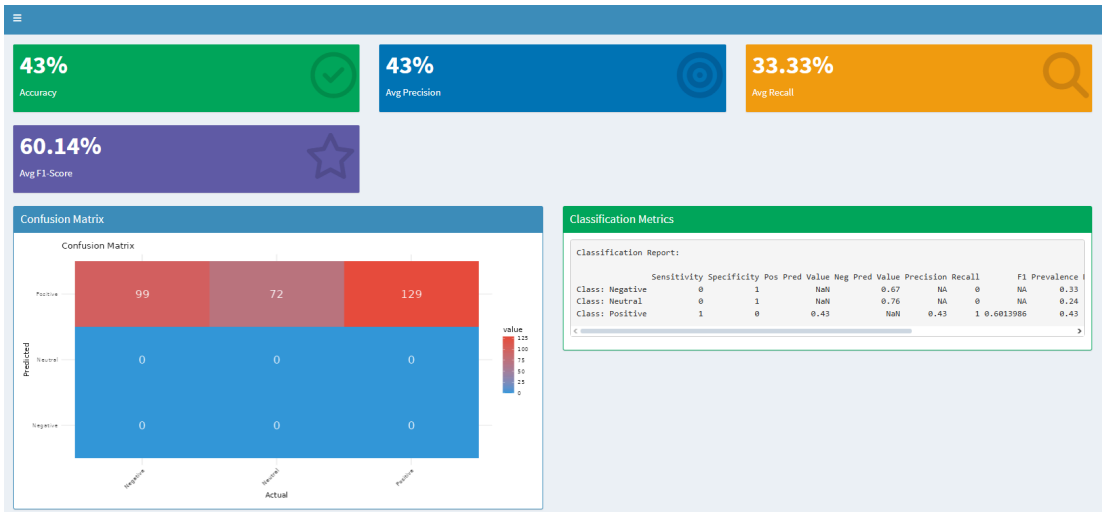


Figure 5.1.3: Model Evaluation of Decision Trees Model on Customer Sentiment data

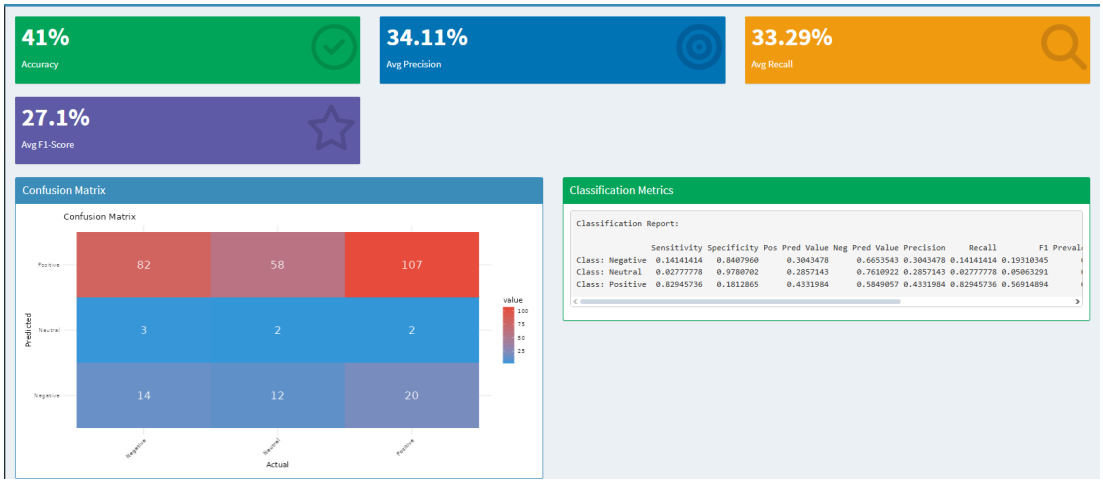
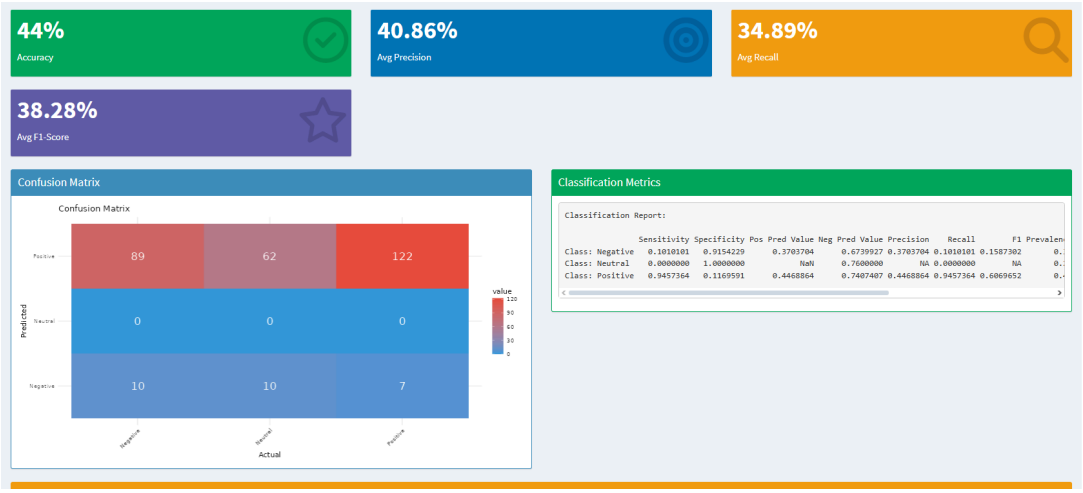


Figure 5.1.4: Model Evaluation of Naive Bayes Model on Customer Sentiment data



5.2 Result Analysis

The comparative evaluation of four machine learning algorithms revealed unexpectedly poor performance across all models. Random Forest achieved 41% accuracy with 34.56% precision, 34.32% recall, and 30.59% F1-score. Support Vector Machine performed marginally better at 43% accuracy but showed inconsistent metrics with 43% precision and 33.33% recall but has a high f1 score of 60.14%. Decision Tree achieved 41% accuracy with balanced precision (34.11%) and recall (33.29%), yielding an F1-score of 27.1%. Naive Bayes demonstrated the highest accuracy at 44% with 40.86% precision and 34.89% recall, resulting in a 38.28% F1-score.

These results are concerning because all of the models performed only slightly better than random guessing. With three classes, random chance would give about 33% accuracy, and the models only reached about 41–44%. This is also far below a simple majority-class baseline, which would get around 71% accuracy based on the class distribution. Since very different algorithms all produced nearly the same low performance, the issue is likely with the data itself rather than the models.

There are several possible reasons for this: the features may not have been processed or engineered well enough, the data might not contain strong signals that relate to the target, there could be too much noise, or the features may not actually match what we are trying to predict. It is also possible that there were errors in how the data was collected or labeled.

Overall, the consistently poor performance suggests that the models were unable to learn meaningful patterns about customer sentiment. This means the current features may not be suitable for prediction, or that there are deeper issues with data quality or how the problem is

defined. Future work should focus on improving data quality, creating better features, and possibly rethinking the problem setup before trying more advanced algorithms.