Analyzing Customer Experience Data from Utility Company

Table of Contents

1. Introduction
2. Methodology
3. Experience and results
4. Discussion
5. Conclusion
6. Reference:
7. Introduction

Background

The company is trying to get valuable insight to improve customer experience by analyzing customer feedback from email surveys. Classifying this feedback helps the business narrow their focus on where they should focus their resources and process improvement on

Problem Statement

Can Machine learning accurately help classify customer experience feedback.

Project Objectives

* Preprocess, tokenize and vectorize data
* Train and test data and evaluate different classification models
* Compare model

Scope and Limitations

* Focused on two models: Naïve Bayes and Random Forest
* Used Text-Only classification based on one column in the dataset

1. Methodology

Data and Tools Used

* Dataset: Customer Experience text data from work
* Tools : Python, scikit-learn, pandas, NLTK, matplotlib, seaborn

Design Decision

* Removal of punctuation, text cleaned to lowercase
* Introduction of stop word filtering using NLTK module
* Tokenizing and Vectorizing using TF-IDF
* K-Fold 5 Cross -Validation for performance evaluation

Models Used

* Multinomial Naive Bayes
* Random Forest Classifier (100 estimators, fixed random state)

Implementation Steps

* Load and clean data
* TF-IDF vectorization
* Apply K-Fold splitting
* Train each model and record scores
* Evaluate using precision, recall, and F1

1. Experiments and Results

Metrics used

* Accuracy
* Precision
* Recall
* F1 Score

## Results (Sample from code)

* Naive Bayes Recall: 0.64
* Naive Bayes Precision: 0.832
* Naive Bayes F1 Score: 0.682
* Random Forest Recall: 0.64
* Random Forest Precision: 0.832
* Random Forest F1 score :0.682
* Visualizations
* Confusion matrix
* Frequency Chart(Bar Chart)

1. Discussion

## Meaning of Results

The result showed that Random Forest after tunning had zero to no impact on Recall compared to Navies Bayes which had no hyperparameters. Additionally, I compared the data by removing stop word which resulted to getting the same result. In other to get of maximize Recall we may have to run other models like GridSearchCV or RandomizedSearchCV

## Objectives Met:

I got a Precision of 0.832 which mean the model predicts a positive 83.2% of the time. And you have a few false positive

I got a Recall of 0.64 which is moderate this tells you the model is only catching 64% of the actual positive cases

Finally, I got a F1 score of 0.682 which tells use the model is reasonably balanced

|  |  |  |
| --- | --- | --- |
| **Metric** | **Naïve Bayes** | **Random Forest** |
| *Recall* | 0.64 | 0.64 |
| *Precision* | 0.832 | 0.832 |
| *F1 score* | 0.682 | 0.682 |

## Challenges:

* Limited labeled data
* More data is needed
* The features don’t’ have enough complexity.
* Data is small or imbalanced,
* Text preprocessing or feature extraction may limit model differentiation.

1. Conclusion

Summary

This project focuses on using Machine learning techniques such as Naïve Bayes and Random Forest classifiers to classify customer data using sentimental analysis to gain overall customer experience. To achieve this, I had to preprocess, clean and tokenize the data. To see how well the techniques worked I used a combination of TF-IDF vectorization and cross validation to evaluate metrics such as accuracy, precision, recall and F1-score the performance. The result showed that Random Forest after tunning had zero to no impact on Recall compared to Navies Bayes which had no hyperparameters. Additionally, I compared the data by removing stop word which resulted to getting the same result. In other to get of maximize Recall we may have to run other models like GridSearchCV or RandomizedSearchCV.

Key Takeaways

* Preprocessing and vectorization are critical to achieve a better precision score
* Model selection impacts metric trade-offs but through tunning and experimenting with different model like GridSearchCV
* Cross-validation helps avoid overfitting

Suggestions for Future Work

* Include additional models like GridSearchCV or RandomizedSearchCV.
* Experiment with word embeddings (e.g., Word2Vec, BERT)
* Collect more diverse training data