Sentiment Analysis following KDD and AutoML

Neel Desai

September 2023

Abstract

Sentiment analysis, a significant subfield within Natural Language Processing (NLP), is vital for discerning sentiment or emotion in textual data, offering invaluable insights into public opinions about products, services, or topics. This research employs the Knowledge Discovery in Databases (KDD) methodology, a structured approach to data mining, to construct a sentiment analysis model using the Threads app reviews dataset. The main stages of the KDD process are discussed, emphasizing domain understanding and data transformation. The research's findings, derived from the Threads app dataset via the KDD methodology, shed light on sentiment classification intricacies. The Ridge Classifier, selected via the automated PyCaret library, is notably mentioned. Further details and implications are discussed in the paper.

1 Introduction

Sentiment analysis, a significant subfield within Natural Language Processing (NLP), focuses on discerning the sentiment or emotion present in textual data. This form of analysis can provide businesses and researchers alike with valuable insights into public opinion about products, services, or various topics. In the realm of data mining, the Knowledge Discovery in Databases (KDD) methodology offers a systematic and comprehensive framework, guiding researchers through the entire process from raw data to actionable knowledge.

In this research, we harness the KDD methodology to construct a sentiment analysis model using the Threads app reviews dataset. Our aim is to classify reviews into one of three categories: positive, negative, or neutral.

2 Methodologies

The Knowledge Discovery in Databases (KDD) process is a structured approach to data mining, encompassing various stages that transform raw data into actionable knowledge. The main stages of the KDD process are:

1. **Understanding the Domain:** This preliminary phase involves gathering domain-specific knowledge. It aids in setting clear objectives and provides context for the subsequent stages.

2. **Data Selection:** In this stage, relevant data for the analysis task is identified and fetched from the source. By selecting pertinent data, the process reduces noise and computational overhead.

3. **Data Preprocessing:** Raw data often contains inconsistencies, missing values, or noise. This stage addresses such issues through cleaning, transformation, and normalization. It ensures the data is in a format suitable for mining.

4. **Data Transformation:** This phase involves further refining the preprocessed data. It includes operations like data smoothing, aggregation, or generalization, transforming the data into forms apt for mining.

5. **Data Mining:** The core phase where specific algorithms are applied to extract patterns from the transformed data. Depending on the objective, techniques such as clustering, classification, regression, or association rule mining might be employed.

6. **Pattern Evaluation and Knowledge Representation:** Once patterns are identified, they are evaluated in this stage. Redundant or irrelevant patterns are filtered out, and the significant patterns are visually represented or translated into other intuitive formats.

7. **Use of Discovered Knowledge:** The culminating stage where the derived knowledge is applied to the domain, fulfilling the objectives set out in the initial phase.

3 Implementation

In line with the KDD methodology, our approach to building the sentiment analysis model on the Threads app reviews dataset was systematic and comprehensive. Below is a breakdown of our approach, mapped to the stages of the KDD process:

1. **Understanding the Domain:** We set a clear objective: to classify English statements from the Threads app reviews as positive, negative, or neutral. From the dataset, we inferred that ratings can serve as proxies for sentiments: a rating of 1 denotes negative, 2-4 signifies neutral, and 5 indicates positive.

2. **Data Selection:** Our dataset was sourced from Kaggle, specifically tailored for our objective.

3. **Data Preprocessing:** Initial exploration of the data helped in understanding its structure and contents, enabling the identification of anomalies, outliers, or missing values. Ratings were encoded based on our inference: 1 for negative, 2-4 for neutral, and 5 for positive. Furthermore, the dataset contained reviews in multiple languages. To ensure consistency and accuracy, we retained only reviews written in English.

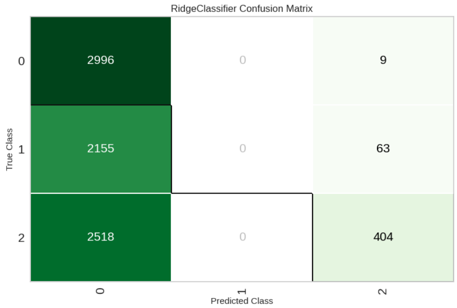

Figure 1: Confusion Matrix

4. **Data Transformation and Modeling:** We employed the automated machine learning library, PyCaret. This library facilitated seamless data transformation and enabled experimentation with multiple algorithms. Through PyCaret, the Ridge Classifier emerged as the top-performing model, suitable for our text classification problem.

5. **Pattern Evaluation and Knowledge Representation:** Post-training, we assessed the model's performance using key metrics. Based on our evaluations, the model achieved an accuracy of approximately 41.74% and a precision of approximately 44.86% for the Positive class. Given these metrics, it's evident that the model faces challenges in distinguishing between Positive and Neutral sentiments. The distinctions between these sentiments in the data might be subtle, or the challenge could be attributed to the chosen features and model.

6. **Use of Discovered Knowledge:** With the trained model in place, we tested its capability on new reviews to gain insights into user sentiments towards the Threads app. Specifically, for the statements “it is alright”, “better than twitter”, and “worse one so far”, the model's predictions were “Negative”, “Positive”, and “Negative”, respectively. Notably, the model misidentified the sentiment of the statement “it is alright” as “Negative” when, in reality, it should be “Neutral”. This underlines potential areas of improvement, particularly in distinguishing neutral sentiments from positive or negative ones.

4 Conclusion

The endeavor to create a sentiment analysis model using the Threads app reviews dataset through the KDD methodology provided valuable insights into the intricacies of sentiment classification. While the Ridge Classifier, chosen through the automated PyCaret library, emerged as our top-performing model, the results underscored certain challenges, particularly in discerning neutral sentiments from positive or negative ones. This suggests potential areas for further refinement, whether by incorporating more sophisticated features, leveraging larger or more diverse training datasets, or experimenting with advanced models.

As sentiment analysis continues to evolve, ensuring that models accurately capture the nuances of human sentiment remains paramount. Future research and iterations of our model can delve deeper into addressing the identified challenges, aiming for more robust and nuanced sentiment classification.

5 References

1. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), 37.

2. Chollet, F. (2015). Keras. GitHub. https://github.com/fchollet/keras

3. Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

4. PyCaret. (2020). An open-source, low-code machine learning library in Python. https://www.pycaret.org

5. Threads App Reviews Dataset. Kaggle. https://www.kaggle.com/datasets/jayagopal20/threads-app-reviews-dataset

6. Code https://github.com/neel26desai/cmpe255$_{a}ssignment3$