Projektbericht

Sentimentanalyse von Produktrezensionen

Studiengang
”Angewandte Künstliche Intelligenz”

Natural Language Processing
DLBAIPNLP01_D

Dawid Jedlinski

Matrikelnummer: IU14113900

dawid.jedlinski@iu-study.org

Tutor: Maja Popovic

Abgabedatum: dd.mm.yyyy

Inhaltsverzeichnis

I. Abbildungsverzeichnis

II. Tabellenverzeichnis

III. Abbreviations

EDA Explorative Datenanalyse
GMM Gaussian Mixture Model
HR Human Resources
LLE Locally Linear Embedding
MDS Multidimensional Scaling
MH Mental Health (Mentale Gesundheit)
ML Machine Learning
NLP Natural Language Processing
OSMI Open Sourcing Mental Illness
PCA Principal Component Analysis

1. Einleitung

- irgendwas... ————————————————————

1.1. Problemstellung

- Bedeutung von Online-Produktrezensionen - Herausforderung der manuellen Auswertung - Relevanz automatischer Sentimentanalyse

1.2. Zielsetzung

- Entwicklung eines Textklassifikationssystems - Vorhersage numerischer Bewertungen (1–5 Sterne) - Fokus auf Mehrklassen-Klassifikation

1.3. Vorgehensweise

- Überblick über Daten, Methodik und Evaluation - Kurze Beschreibung der einzelnen Kapitel

2. Explorative Datenanalyse

- irgendwas ... ————————————————————-

2.1. Quelle/Herkunft

- Amazon Reviews'23 Datensatz (McAuley Lab) - Allgemeine Beschreibung des Datensatzes

2.2. Datenumfang und Aufteilung

- Kategorien: allbeauty, handmadeProducts und healthAndPersonalCare, da sie von der Größe ähnlich sind (knapp 300MB) und unterschiedlich AllBeauty - 701.528 HandmadeProducts - 664.162 HealthAndPersonalHealth - 494.121 - es gibt 10 verschiedene Spalten, hauptsächlich type object, aber auch DateTime und 2 int64 für Rating und HelpfulVote, und bool bei verified_purchase - zum Glück gibts bei keiner Spalte fehlende Werte

2.3. Verteilung der Bewertungen

- AllBeauty mean bei 3.96 - HandmadeProducts mean bei 4.49 - HealthAndPersonalHealth mean bei 3.99 also sehr positiver Datensatz - std ist bei allen auch nicht groß liegt bei 1.4 ungefähr - die meisten Ratings sind 5, also später die Klassen ausgleichen, und Datenmenge sowieso reduzieren - wichtig zu erwähnen ist, dass bei handmade bei 5% reviews der Einkauf nicht verifiziert wurde, bei AllBeauty und Healthcare sind es sogar 10% - die Textlänge ist bis auf wenige Ausreißer bei AllBeauty und HealthCare gleich, Handmade hat etwas kürze - Bei Länge von Bewertungen, sieht man das min = 0 ist. D.h. obwohl es keine missing-values gab, gibt es trotzdem Bewertungen ohne Text, die man bei der Vorverarbeitung entfernen muss

3. Datenvorverarbeitung

- irgendwas über Preprocessing ————————————————————-

3.1. Data Cleaning

- Datenmenge reduzieren und die Verteilung von Rating ausgleichen - alle reviews löschen die nicht verifizierten Kauf haben - alle mit Textlänge ¡ 10 löschen - alle spalten löschen die unnötig sind (es bleiben nur: rating, title, text)

3.2. Textaufbereitung

- Zusammenführung von Titel und Text - lowercasing - Tokenisierung - Entfernen von Stoppwörtern - Umgang mit Sonderzeichen (Punctuation) - Entfernung von Emojis - Lemmatization (langsamer, aber genauer als Stemming und das ist hier wichtiger) - lemmatisierung ist bei default nur auf Nomen eingestellt, deswegen müssen wir zuerst ein POS-Tagging machen um jedes Wort mit der Wortart zu taggen. Als erstens wird pos_tag einen Tag zuweisen, danach muss man den - abhängig von der Ausgabe - die richtige Kategorie an lemmatizer übergeben

3.3. Numerische Repräsentation

- es wird für jedes df eine spalte category erstellt mit value (allbeauty, handmade oder healthcare) um es dann nachdem zusammenführen besser zu unterscheiden. Dies hilft bei einer gleichmäßigen Aufteilung in Train und Test. - danach alle 3 DF zusammenführen - hier wird es gesplittet. Train 0.8, Test 0.2, gleichmäßig nach category mithilfe von stratify-argument - Verwendung von TF-IDF da viele Wörter redundant (product, use, one, etc.), hebt wichtige wörter hervor (excellent, terrible, perfect) BoW zählt nur Wörter und beachtet keine Relevanz - n_gram auf 1,5 damit negationen erkannt werden, minDF auf 5 (Wort muss mind. in 5 Rezensionen vorkommen sonst unwichtig), maxDF auf 0.95 (wenn ein Wort mehr als in 95% der Rezensionen vorkommt dann unnötig) - traindaten fitten und transformieren, testdaten nur transformieren - alle daten und tfidf als datei speichern

3.4. Trennung von Eingabe- und Ausgabedaten

- Definition der Zielvariable (numerische Bewertung) EINGABE (Features, X): Text = Titel + Beschreibung AUSGABE (Target, y): Rating = 1, 2, 3, 4, 5 Sterne das vielleicht erst beim Modell und als Funktion! (text als eingabe und dann die gleiche Reihenfolge wie bei Preprocessing)

4. Modell und Methodik

- Problemformulierung als Klassifikationsaufgabe - Supervised Learning - Mehrklassen-Klassifikation (5 Klassen)

4.1. Logistic Regression

- [1] discriminative - [1] kategorien selbst bestimmen - Beschreibung des gewählten Klassifikators - Begründung der Eignung für Textklassifikation

4.2. Naive Bayes

- [1] generative - [1] rechnet mit wahrscheinlichkeiten - Beschreibung des gewählten Klassifikators - Begründung der Eignung für Textklassifikation

VERGLEICH - Both methods are simple; Naive Bayes is the simplest one. - Both methods are interpretable: you can look at the features which influenced the predictions most (in Naive Bayes - usually words, in logistic regression - whatever you defined). - Naive Bayes is very fast to train - it requires only one pass through the training data to evaluate the counts. For logistic regression, this is not the case: you have to go over the data many times until the gradient ascent converges. - Naive Bayes is too ”naive it assumed that features (words) are conditionally independent given class. Logistic regression does not make this assumption - we can hope it is better. - Both methods use manually defined feature representation (in Naive Bayes, BOW is the standard choice, but you still choose this yourself). While manually defined features are good for interpretability, they may be no so good for performance - you are likely to miss something which can be useful for the task.

4.3. Trainingsprozess

- Training auf dem Trainingsdatensatz - Begründung des Vorgehens - beim Training der beiden Modelle wurde zusätzlich eine Pipeline erstellt um das beste NGram für Vektorisierung zu finden. Dabei wurde der F1-Score analysiert - es wurden (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6) untersucht - dabei wurde auch die Trainingszeit gemessen - Ergebnisse - größter Sprung bei beiden von (1,2) auf (1,3). F1Score zwischen (1,5) und (1,6) ist nicht groß, aber die Trainingsdauer ist fast 1.5x länger. Daher wird (1,5) gewählt

5. Evaluation und Ergebnisse

- irgendwas... - accuracy bei beiden ungefähr 55%, scheint wenig zu sein, aber es ist normal bei multiclass (60%), bei binär wäre 80-90% -————————————————————

5.1. Evaluationsmetriken

- Accuracy, precision, ... - Confusion Matrix - Begründung der Metrikauswahl (Sentimentanalyse ist ein typisches Klassifizierungsproblem)

5.2. Gesamtergebnisse

- Accuracy auf dem Testsatz - Darstellung und Interpretation der Confusion Matrix

5.3. Ergebnisse von zwei Algorithmen (optional)

- Vergleich der Modellleistung zwischen Algorithmen (z.B. Naive Bayes und NN) - Kurze Analyse der Unterschiede

6. Schluss