1.

Record Nr.	UNINA9910735393903321
Autore	Kulkarni Akshay
Titolo	Natural language processing recipes : unlocking text data with machine learning and deep learning using Python / / Akshay Kulkarni, Adarsha Shivananda
Pubbl/distr/stampa	[Place of publication not identified] : , : Apress, , [2021] ©2021
ISBN	1-5231-5091-2 1-4842-7351-6
Edizione	[Second edition.]
Descrizione fisica	1 online resource (302 pages)
Disciplina	006.35
Soggetti	Natural language processing (Computer science)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Intro -- Table of Contents -- About the Authors -- About the Technical Reviewer -- Acknowledgments -- Introduction -- Chapter 1: Extracting the Data -- Introduction -- Client Data -- Free Sources -- Web Scraping -- Recipe 1-1. Collecting Data -- Problem -- Solution -- How It Works -- Step 1-1. Log in to the Twitter developer portal -- Step 1-2. Execute query in Python -- Recipe 1-2. Collecting Data from PDFs -- Problem -- Solution -- How It Works -- Step 2-1. Install and import all the necessary libraries -- Step 2-2. Extract text from a PDF file -- Recipe 1-3. Collecting Data from Word Files -- Problem -- Solution -- How It Works -- Step 3-1. Install and import all the necessary libraries -- Step 3-2. Extract text from a Word file -- Recipe 1-4. Collecting Data from JSON -- Problem -- Solution -- How It Works -- Step 4-1. Install and import all the necessary libraries -- Step 4-2. Extract text from a JSON file -- Recipe 1-5. Collecting Data from HTML -- Problem -- Solution -- How It Works -- Step 5-1. Install and import all the necessary libraries -- Step 5-2. Fetch the HTML file -- Step 5-3. Parse the HTML file -- Step 5-4. Extract a tag value -- Step 5-5. Extract all instances of a particular tag -- Step 5-6. Extract all text from a particular tag -- Recipe 1-6. Parsing Text Using Regular Expressions -- Problem -- Solution -- How It Works -- Tokenizing -- Extracting Email IDs -- Replacing Email IDs -- Extracting Data

from an eBook and Performing regex -- Recipe 1-7. Handling Strings -- Problem -- Solution -- How It Works -- Replacing Content -- Concatenating Two Strings -- Searching for a Substring in a String -- Recipe 1-8. Scraping Text from the Web -- Problem -- Solution -- How It Works -- Step 8-1. Install all the necessary libraries -- Step 8-2. Import the libraries -- Step 8-3. Identify the URL to extract the data. Step 8-4. Request the URL and download the content using Beautiful Soup -- Step 8-5. Understand the website's structure to extract the required information -- Step 8-6. Use Beautiful Soup to extract and parse the data from HTML tags -- Step 8-7. Convert lists to a data frame and perform an analysis that meets business requirements -- Step 8-8. Download the data frame -- Chapter 2: Exploring and Processing Text Data -- Recipe 2-1. Converting Text Data to Lowercase -- Problem -- Solution -- How It Works -- Step 1-1. Read/create the text data -- Step 1-2. Execute the lower() function on the text data -- Recipe 2-2. Removing Punctuation -- Problem -- Solution -- How It Works -- Step 2-1. Read/create the text data -- Step 2-2. Execute the replace() function on the text data -- Recipe 2-3. Removing Stop Words -- Problem -- Solution -- How It Works -- Step 3-1. Read/create the text data -- Step 3-2. Remove punctuation from the text data -- Recipe 2-4. Standardizing Text -- Problem -- Solution -- How It Works -- Step 4-1. Create a custom lookup dictionary -- Step 4-2. Create a custom function for text standardization -- Step 4-3. Run the text_std function -- Recipe 2-5. Correcting Spelling -- Problem -- Solution -- How It Works -- Step 5-1. Read/create the text data -- Step 5-2. Execute spelling correction on the text data -- Recipe 2-6. Tokenizing Text -- Problem -- Solution -- How It Works -- Step 6-1. Read/create the text data -- Step 6-2. Tokenize the text data -- Recipe 2-7. Stemming -- Problem -- Solution -- How It Works -- Step 7-1. Read the text data -- Step 7-2. Stem the text -- Recipe 2-8. Lemmatizing -- Problem -- Solution -- How It Works -- Step 8-1. Read the text data -- Step 8-2. Lemmatize the data -- Recipe 2-9. Exploring Text Data -- Problem -- Solution -- How It Works -- Step 9-1. Read the text data -- Step 9-2. Import necessary libraries.

Step 9-3 Check the number of words in the data -- Step 9-4. Compute the frequency of all words in the reviews -- Step 9-5. Consider words with length greater than 3 and plot -- Step 9-6. Build a word cloud -- Recipe 2-10. Dealing with Emojis and Emoticons -- Problem -- Solution -- How It Works -- Step 10-A1. Read the text data -- Step 10-A2. Install and import necessary libraries -- Step 10-A3. Write a function that coverts emojis into words -- Step 10-A4. Pass text with an emoji to the function -- Problem -- Solution -- How It Works -- Step 10-B1. Read the text data -- Step 10-B2. Install and import necessary libraries -- Step 10-B3. Write a function to remove emojis -- Step 10-B4. Pass text with an emoji to the function -- Problem -- Solution -- How It Works -- Step 10-C1. Read the text data -- Step 10-C2. Install and import necessary libraries -- Step 10-C3. Write function to convert emoticons into word -- Step 10-C4. Pass text with emoticons to the function -- Problem -- Solution -- How It Works -- Step 10-D1 Read the text data -- Step 10-D2. Install and import necessary libraries -- Step 10-D3. Write function to remove emoticons -- Step 10-D4. Pass text with emoticons to the function -- Problem -- Solution -- How It Works -- Step 10-E1. Read the text data -- Step 10-E2. Install and import necessary libraries -- Step 10-E3. Find all emojis and determine their meaning -- Recipe 2-11. Building a Text Preprocessing Pipeline -- Problem -- Solution -- How It Works -- Step 11-1. Read/create the text data -- Step 11-2. Process the text --

Chapter 3: Converting Text to Features -- Recipe 3-1. Converting Text to Features Using One-Hot Encoding -- Problem -- Solution -- How It Works -- Step 1-1. Store the text in a variable -- Step 1-2. Execute a function on the text data -- Recipe 3-2. Converting Text to Features Using a Count Vectorizer -- Problem.

Solution -- How It Works -- Recipe 3-3. Generating n-grams -- Problem -- Solution -- How It Works -- Step 3-1. Generate n-grams using TextBlob -- Step 3-2. Generate bigram-based features for a document -- Recipe 3-4. Generating a Co-occurrence Matrix -- Problem -- Solution -- How It Works -- Step 4-1. Import the necessary libraries -- Step 4-2. Create function for a co-occurrence matrix -- Step 4-3. Generate a co-occurrence matrix -- Recipe 3-5. Hash Vectorizing -- Problem -- Solution -- How It Works -- Step 5-1. Import the necessary libraries and create a document -- Step 5-2. Generate a hash vectorizer matrix -- Recipe 3-6. Converting Text to Features Using TF-IDF -- Problem -- Solution -- How It Works -- Step 6-1. Read the text data -- Step 6-2. Create the features -- Recipe 3-7. Implementing Word Embeddings -- Problem -- Solution -- How It Works -- skip-gram -- Continuous Bag of Words (CBOW) -- Recipe 3-8. Implementing fastText -- Problem -- Solution -- How It Works -- Recipe 3-9. Converting Text to Features Using State-of-the-Art Embeddings -- Problem -- Solution -- ELMo -- Sentence Encoders -- doc2vec -- Sentence-BERT -- Universal Encoder -- InferSent -- Open-AI GPT -- How It Works -- Step 9-1. Import a notebook and data to Google Colab -- Step 9-2. Install and import libraries -- Step 9-3. Read text data -- Step 9-4. Process text data -- Step 9-5. Generate a feature vector -- Sentence-BERT -- Universal Encoder -- Infersent -- Open-AI GPT -- Step 9-6. Generate a feature vector function automatically using a selected embedding method -- Chapter 4: Advanced Natural Language Processing -- Recipe 4-1. Extracting Noun Phrases -- Problem -- Solution -- How It Works -- Recipe 4-2. Finding Similarity Between Texts -- Solution -- How It Works -- Step 2-1. Create/read the text data -- Step 2-2. Find similarities -- Phonetic Matching.

Recipe 4-3. Tagging Part of Speech -- Problem -- Solution -- How It Works -- Step 3-1. Store the text in a variable -- Step 3-2. Import NLTK for POS -- Recipe 4-4. Extracting Entities from Text -- Problem -- Solution -- How It Works -- Step 4-1. Read/create the text data -- Step 4-2. Extract the entities -- Using NLTK -- Using spaCy -- Recipe 4-5. Extracting Topics from Text -- Problem -- Solution -- How It Works -- Step 5-1. Create the text data -- Step 5-2. Clean and preprocess the data -- Step 5-3. Prepare the document term matrix -- Step 5-4. Create the LDA model -- Recipe 4-6. Classifying Text -- Problem -- Solution -- How It Works -- Step 6-1. Collect and understand the data -- Step 6-2. Text processing and feature engineering -- Step 6-3. Model training -- Recipe 4-7. Carrying Out Sentiment Analysis -- Problem -- Solution -- How It Works -- Step 7-1. Create the sample data -- Step 7-2. Clean and preprocess the data -- Step 7-3. Get the sentiment scores -- Recipe 4-8. Disambiguating Text -- Problem -- Solution -- How It Works -- Step 8-1. Import libraries -- Step 8-2. Disambiguate word sense -- Recipe 4-9. Converting Speech to Text -- Problem -- Solution -- How It Works -- Step 9-1. Define the business problem -- Step 9-2. Install and import necessary libraries -- Step 9-3. Run the code -- Recipe 4-10. Converting Text to Speech -- Problem -- Solution -- How It Works -- Step 10-1. Install and import necessary libraries -- Step 10-2. Run the code with the gTTs function -- Recipe 4-11. Translating Speech -- Problem -- Solution -- How It Works -- Step 11-1. Install and import necessary libraries -- Step 11-2. Input text -- Step 11-3. Run the goslate function -- Chapter 5: Implementing Industry Applications -- Recipe 5-1. Implementing Multiclass Classification -- Problem -- Solution -- How It Works -- Step 1-1. Get the data from Kaggle. Step 1-2. Import the libraries.