BAB 4

Kekelompokan

Tujuan bab ini adalah untuk memecahkan masalah dunia nyata dengan bantuan algoritma pembelajaran terawasi dan tak terawasi. Pertama, kita akan mulai dengan konsep pengelompokan, menentukan cara mengorganisasikan data, menentukan jumlah komponen, dan kemudian melihat apakah keluaran klaster masuk akal secara intuitif.

ÿ **Catatan:** Buku ini menggunakan **Python 2.7.11** sebagai standar de facto untuk contoh pengkodean. Selain itu, Anda diharuskan menginstalnya untuk Latihan.

Dalam bab ini, kami akan menggunakan **dataset Accepted Papers** untuk mengelompokkan contoh kode. Data dump ini dapat diunduh dari

https://archive.ics.uci.edu/ml/datasets/AAAI+2014+Makalah+Yang+Diterima

Studi Kasus: Penentuan Kata Kunci Ekor Pendek untuk Pemasaran

Ross, direktur pemasaran sebuah konferensi kecerdasan buatan (AI), harus melapor kepada dewan direksi dengan rekomendasi kata kunci berekor pendek untuk pemasaran konferensi tahun 2015. Ia juga harus membuat visualisasi makalah penelitian yang telah diserahkan sejauh ini, dikelompokkan berdasarkan jenis makalahnya.

Konferensi ini diselenggarakan oleh sebuah perkumpulan ilmiah nirlaba yang memastikan kemajuan di bidang kecerdasan buatan. Hal ini dilakukan dengan meningkatkan pemahaman publik tentang bidang tersebut, serta menyediakan wadah bagi para peneliti untuk mempresentasikan temuan mereka dalam konferensi tahunan.

Untuk menemukan solusi atas masalah yang dihadapi, Ross memulai dengan pergi melalui situs web. Namun, besarnya isi makalah penelitian tersebut terlalu banyak untuk dianalisis.

Haroon, *Studi Kasus Pembelajaran Mesin Python,* DOI 10.1007/978-1-4842-2823-4_4

Ia pun mengetuk pintu Ted untuk mencari tahu apakah ada data yang tersedia untuk memulai analisis. Pertemuan dengan Ted, yang merupakan direktur departemen pergudangan data, memberi Ross secercah harapan. Sebagaimana yang diingat Ted:

> *Ross datang kepada saya dengan permintaan ini, tetapi masalahnya adalah kami sedang melakukan perombakan besar-besaran dalam mengintegrasikan proses SAP masuk/keluar. Untungnya, kami memiliki beberapa data dump yang tersedia di penyimpanan. Saya bertanya kepada Ross tentang tahun konferensi yang dia inginkan datanya, dan bum, kami mendapatkan data dump untuk tahun itu.*

Ross memuat data dump di Microsoft Excel dan mulai memeriksa fitur-fiturnya. Ia berpikir bahwa salah satu strateginya adalah mengelompokkan makalah berdasarkan kata kunci atau grup. Ia kemudian dapat menjalankan iklan terpisah di media sosial dan menggunakan SEO (optimasi mesin pencari) untuk menargetkan kata kunci masing-masing di dalamnya. Namun, terdapat terlalu banyak nilai yang berbeda dalam kata kunci dan fitur grup, sehingga akan menghasilkan banyak klaster kata kunci SEO. Strateginya adalah menghasilkan maksimal sepuluh grup kata kunci SEO yang berbeda dan membiarkannya berjalan selama beberapa hari untuk mendapatkan manfaatnya. Ia pernah mendengar tentang pengelompokan, di mana data dikelompokkan ke dalam sejumlah klaster yang tetap, sehingga ia memutuskan bahwa pengelompokan akan menjadi titik awalnya.

Setelah memiliki jalur yang harus diikuti, ia memiliki beberapa pertanyaan dalam benaknya. Berapa banyak klaster yang optimal dari data yang diberikan? Mengingat datanya kecil, dapatkah algoritma pembelajaran mesin apa pun menjadi yang paling cocok untuknya? Bagaimana ia harus menyajikan data secara visual dan, pada akhirnya, temuannya?

Ross percaya bahwa memahami fitur data yang diberikan oleh Ted akan membuktikan bermanfaat di kemudian hari. Ia merasa bahwa memahami hal tersebut akan membantunya menentukan pendekatan terbaik untuk menyelesaikan masalah yang dihadapi. Oleh karena itu, ia menyusun kamus data pada Tabel 4-1.

Tabel 4-1. *Kamus Data untuk Dataset Makalah yang Diterima*

Nama fitur	Keterangan
Judul	Judul makalah
Penulis	Penulis makalah
Kelompok	Kata kunci tingkat tinggi yang dipilih penulis
Kata Kunci	Kata kunci yang dibuat oleh penulis
Topik	Kata kunci tingkat rendah yang dipilih penulis
Abstrak	Abstrak makalah

Saat mengamati Tabel 4-1, Ross menyimpulkan bahwa semua fitur berada dalam format string. Lebih lanjut, ia memperhatikan bahwa naskah tidak termasuk dalam kumpulan data dan hanya abstrak yang disediakan.

Ross baru saja menyelesaikan spesialisasi Python yang ditawarkan oleh University of Michigan di Coursera. Oleh karena itu, ia yakin dengan kemampuan Python-nya untuk menyelesaikan pekerjaan tersebut. Sebelum memulai analisis, ia memutuskan untuk memuat semua paket yang relevan ke dalam memori seperti yang ditunjukkan pada Daftar 4-1.

Daftar 4-1. Mengimpor Paket yang Diperlukan untuk Bab Ini

```
%matplotlib sebaris

operator impor
impor itertools
impor numpy sebagai np
impor panda sebagai pd
dari ggplot impor impor *
seaborn sebagai sns
impor matplotlib sebagai mpl
dari campuran impor sklearn
impor matplotlib.pyplot sebagai plt
dari matplotlib.pylab impor rcParams
dari sklearn.dekomposisi impor PCA
dari wordcloud impor WordCloud, STOPWORDS
dari scipy.spatial.distance impor cdist, pdist
dari sklearn.cluster impor KMeans, SpectralClustering
dari sklearn.metrics impor jarak_euclidean, skor_siluet

rcParams['gambar.ukuranfig'] = 15, 5
```

Meskipun Ross mengetahui fitur dan deskripsinya, dia penasaran untuk mengetahui isinya.

Eksplorasi Fitur

Sebelum melanjutkan, tibalah waktunya baginya untuk menyusun strateginya. Setelah memikirkannya dengan saksama, ia akhirnya menemukan rencana yang berpotensi berhasil. Rencana awalnya adalah membagi makalah penelitian menjadi beberapa segmen sehingga ia dapat berfokus pada masing-masing segmen dengan menggunakan kata kunci ekor pendek dan ekor panjang yang ditargetkan. Ia kemudian melihat fitur-fitur pada Tabel 4-1 untuk menyaring fitur-fitur yang sesuai dengan pendekatannya. Ia berpendapat bahwa kolom Penulis tidak relevan dan kolom Abstrak terlalu detail.

Oleh karena itu ia mempersempit pencariannya menjadi hanya empat fitur (Judul, Grup, Kata Kunci, dan Topik). Dia menyaring fitur-fitur dengan menuliskan potongan kode pada Daftar 4-2.

Daftar 4-2. Membaca Data dalam Memori, dan Fitur Subset

```
data_train = pd.read_csv('contoh/[UCI] Makalah AAAI-14 yang Diterima - Makalah.csv')
data_train = data_train[['judul', 'grup', 'kata kunci', 'topik']]
```

Ross kini tertarik untuk mengetahui seberapa besar kumpulan data tersebut (yaitu, berapa banyak makalah penelitian yang tersedia). Ia juga tertarik untuk mengintip kumpulan data tersebut. Dia menulis naskah pada Daftar 4-3 untuk tujuan itu.

Daftar 4-3. Mencetak Ukuran Dataset dan Mencetak Beberapa Baris Pertama Dataset

```
cetak len(data_train)
data_train.head()
```

Keluaran

```
398
```

Tabel 4-2. *Cetakan Observasi Dataset*

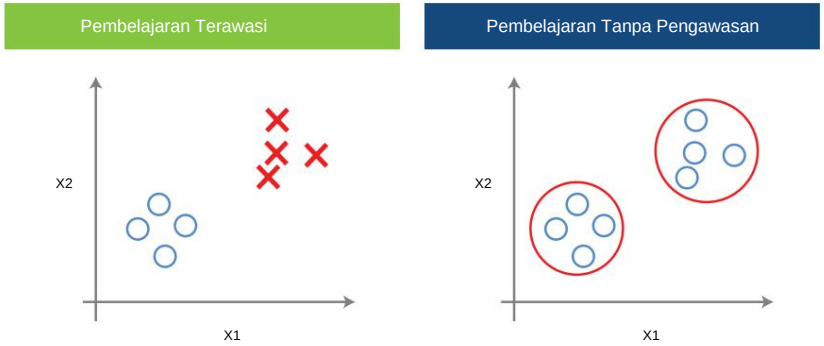
	judul	kelompok	kata kunci	topik
0	Pembelajaran Transfer Bayesian yang Terkemelisasi	Algoritma Pembelajaran Mesin Baru (NMLA)	pembelajaran lintas domain\ndomain adaptasi\nkern...	APP: Biomedis / Bioinformatika\nNMLA: Bayesi...
1	Pembelajaran Transfer "Bebas Sumber" untuk Teks Kelas...	AI dan Web (AIW)\nPembelajaran Mesin Baru A...	Pembelajaran Transfer\nPengambilan Data Tambahan\nT...	AIW: Akuisisi pengetahuan dari web\nAIW: ...
2	Generalisasi Serial Probabilistik ke Ra...	Teori Permainan dan Paradigma Ekonomi (GTEM)	teori pilihan sosial\npemungutan suara\npembagian yang adil\n...	GTEP: Teori Permainan\nGTEP: Pilihan Sosial / Kesejahteraan...
3	Variasi Leksikal Seumur Hidup di Media Sosial	NLP dan Penambangan Teks (NLPTM)	Model Generatif\nJaringan Sosial\nUsia Ramalan	AIW: Personalisasi web dan pengguna pemodelan\nNL...
4	Ambang Nilai Singular Hibrida untuk Tensor...	Representasi Pengetahuan dan Penalaran (KRR) \N...	penyelesaian tensor\npemulihan peringkat rendah\nhibrida s...	KRR: Representasi Pengetahuan (Umum/Lainnya) ...\...

Dari keluaran Daftar 4-3, Ross memperhatikan bahwa kumpulan data tersebut berisi data sekitar 400 makalah penelitian. Lebih lanjut, ia menemukan validasi untuk deduksinya bahwa semua fitur dalam kumpulan data tersebut berformat data string. Ia juga memperhatikan bahwa sebuah makalah penelitian dapat tergolong dalam lebih dari satu kelompok dan kata kunci.

Ross sedang dalam perjalanan pulang, di kereta bawah tanah, menunggu kereta berikutnya tiba, ketika ia bertemu dengan teman kuliahnya, Matt, yang pernah menjadi rekan kerjanya di bidang kerja sosial. Setelah kuliah, Matt berkarier di bidang analisis data. Ross pun mengemukakan masalah yang ingin dipecahkannya. Setelah mendengarkan dengan saksama, Matt menyarankan Ross untuk memutuskan apakah ia akan menggunakan pembelajaran terawasi atau pembelajaran tanpa pengawasan sebelum beralih ke pemilihan model. Percakapan itu singkat karena Matt harus pergi, dan meninggalkan banyak pertanyaan tentang konsep ini di benak Ross. Keesokan paginya, Ross baru kembali ke kantor dan melakukan riset tentang metode pembelajaran terawasi dan tanpa pengawasan.

Pembelajaran Terbimbing vs. Pembelajaran Tanpa Pengawasan

Algoritma pembelajaran mesin biasanya dibagi menjadi pembelajaran terawasi dan tanpa pengawasan.



Gambar 4-1. *Pembelajaran terawasi vs. pembelajaran tanpa pengawasan*

Ross memberikan penjelasan untuk kedua metode ini.

Pembelajaran Terawasi

Sesuai namanya, algoritma pembelajaran terawasi memerlukan supervisi agar dapat melatih model. Supervisi ini biasanya diperlukan dalam kasus klasifikasi di mana kita memiliki data berlabel yang digunakan untuk melatih model agar dapat memprediksi label data yang tidak terlihat. Di sini, supervisi dilakukan melalui label yang diberikan pada setiap observasi (yaitu, mengawasi proses pembelajaran). Contohnya termasuk klasifikasi untuk prediktor diskrit dan regresi untuk prediktor kontinu.

Pembelajaran Tanpa Pengawasan

Algoritma pembelajaran tanpa pengawasan tidak memerlukan pengawasan dari data saat melatih model. Contoh utama dari hal ini adalah pengelompokan, yang menemukan label tanpa pengawasan apa pun. Label yang ditemukan ini kemudian menjadi dasar untuk mengklasifikasikan data baru yang belum terlihat. Contoh lain dari pembelajaran tanpa pengawasan adalah aturan asosiasi, yang mencakup konsep komplemen dan substitusi. Komplemen mengacu pada fenomena di mana jika seorang pembeli membeli X, maka dengan tingkat kepastian yang tinggi, ia juga akan membeli Y. Substitusi mengacu pada perilaku di mana seorang pembeli akan membeli X atau Y.

Contoh lainnya termasuk deteksi anomali, metode momen (misalnya, rata-rata dan kovariansi), analisis komponen independen, dan analisis komponen utama (PCA).

Ross harus memutuskan mana di antara keduanya yang paling cocok untuk permasalahan yang dihadapinya dan data yang tersedia. Dari data pada Tabel 4-2, ia memperhatikan bahwa kumpulan data tersebut tidak memberikan data berlabel. Ini berarti ia harus memprediksi label dari awal menggunakan pembelajaran tanpa pengawasan.

Ross berasumsi bahwa pengelompokan, sebagai metode pembelajaran tanpa pengawasan, dapat membantunya menemukan segmen yang tepat untuk mencapai tujuan pemasaran. Namun, ia tidak terlalu yakin apakah pengelompokan merupakan pendekatan yang tepat. Oleh karena itu, ia menyusun materi berikut tentang topik tersebut.

Kekelompokan

Analisis klaster mengacu pada pengelompokan observasi sehingga objek-objek dalam setiap klaster memiliki properti yang serupa, dan properti semua klaster bersifat independen satu sama lain. Algoritma klaster biasanya mengoptimalkan dengan memaksimalkan jarak antar klaster dan meminimalkan jarak antar objek dalam klaster. Analisis klaster tidak selesai dalam satu iterasi, tetapi melalui beberapa iterasi hingga model konvergen. Konvergensi model berarti keanggotaan klaster semua objek konvergen dan tidak berubah pada setiap iterasi baru. Beberapa algoritma pengelompokan tidak menanyakan jumlah klaster/komponen, dan menghasilkan jumlah klaster yang secara statistik lebih masuk akal. Namun, sebagian besar algoritma pengelompokan meminta pengguna di awal untuk jumlah klaster/komponen yang diinginkannya dalam output.

Penting untuk dipahami bahwa pengelompokan, seperti halnya klasifikasi, digunakan untuk mengelompokkan data; namun, kelompok-kelompok ini sebelumnya tidak didefinisikan dalam kumpulan data pelatihan (yakni, data tersebut tidak berlabel).

Algoritma yang berbeda menerapkan teknik yang berbeda untuk komputasi klaster. Beberapa teknik tersebut adalah sebagai berikut:

- Pemartisian: Mengelompokkan data ke dalam sejumlah klaster tertentu sambil mengoptimalkan sasaran (misalnya, jarak).
- Hierarkis: Mengelompokkan data ke dalam hierarki klaster. Hierarki ini dibentuk dari atas ke bawah atau dari bawah ke atas.
- Berbasis kisi: Membagi data ke dalam sel hiper-persegi panjang, membuang sel berdensitas rendah, dan menggabungkan sel berdensitas tinggi untuk membentuk kluster.

Algoritma pengelompokan yang baik memenuhi persyaratan berikut:

- Kesamaan dalam klaster dan perbedaan antar klaster
- Dapat menangani dataset pelatihan yang:
  - Berdimensi tinggi
  - Terpengaruh oleh noise dan outlier

Ross kini memiliki pengetahuan tentang model terawasi dan tak terawasi; namun, ia tidak yakin apakah pengelompokan dapat diterapkan pada data yang bersifat tekstual. Lebih lanjut, ia tahu bahwa semakin banyak fitur yang dimasukkan, semakin tinggi peluang untuk menemukan fitur yang berpengaruh selama proses penemuan klaster. Oleh karena itu, ia harus menemukan cara untuk mengonversi data tekstual ke dalam bentuk numerik dan menggunakan keempat fitur yang ada untuk menghasilkan lebih banyak fitur.

Transformasi Data untuk Pemodelan

Ross menghabiskan waktu berjam-jam mencari solusi untuk misteri ini. Ia merasa kecewa karena ia berharap pendekatan ini berhasil dan membuahkan hasil. Untuk mempermudah, ia mempersempit pencariannya ke satu fitur (yaitu grup) dan mulai memikirkan bagaimana ia dapat mengodekannya kembali ke dalam bentuk numerik. Hari sudah larut malam, tetapi Ross tidak bisa tidur, dan kemudian sebuah ide muncul di benaknya. Mengapa tidak mengonversi fitur "grup" ke Boolean, sehingga makalah penelitian yang termasuk dalam suatu grup akan ditandai 1 di sebelahnya, dan 0 di sebelahnya. Representasi data yang dapat ia pikirkan adalah matriks di mana kolom akan mewakili berbagai grup dan baris akan mewakili makalah penelitian.

Dia tahu bahwa sebuah makalah penelitian bisa masuk ke dalam satu kelompok. Kelompok untuk penelitian tertentu Makalah-makalah tersebut dipisahkan oleh pembatas (yaitu,)ol dalam kumulan data. Oleh karena itu, iamemutuskan untuk memisahkan kelompok-kelompok dengan membagi pembatas tersebut sehingga jika sebuah makalah penelitian berada dalam tiga kelompok, makalah tersebut akan diwakili oleh tiga observasi berbeda (yaitu, baris) dalam kumpulan data.

nilai='bendera') matriks = matriks.nlma(0).reset_indeks() x_cols = matriks.kolom[1:] kembali

Daftar 4-4. Peregangan Bingkai Data Berdasarkan Baris sebagai Fungsi Grup

```
s = data_train['grup'].str.split('\n').apply(pd.Series, 1).stack() s.indeks =
s.indeks.droplevel(-1) s.nama =
'grup' del
data_train['grup']
data_train = data_train.gabung(s).atur ulang_indeks()
```

Selain memisahkan kelompok, Ross juga memutuskan untuk menambahkan fitur baru "flags" dan menetapkan nilai 1 untuk semua baris. Sesuai konsepnya, nilai 1 akan menunjukkan bahwa makalah penelitian di baris tersebut termasuk dalam kelompok di kolom "groups".

Daftar 4-5. Menambahkan Variabel Baru untuk Keanggotaan Grup

```
data_train['bendera'] = pd.Seri(np.satu(len(data_train)),
indeks=data_train.indeks)
data_train.kepala()
```

Tabel 4-3. *Hasil Observasi Dataset Setelah Transformasi Kerangka Data*

judul	kata kunci	topik	kelompok	bendera
Pembelajaran Transfer Bayesian yang Terkernelisasi	pembelajaran lintas domain\adaptasi domain\nkern...	APP: Biomedis / Bioinformatika\nNLA: Bayesi...	Algoritma Pembelajaran Mesin Baru (NMLA)	1.0
Pembelajaran Transfer "Bebas Sumber" untuk Kelas Teks...	Pembelajaran Transfer\nPengambilan Data Tambahan\nT...	AIW: Akuisisi pengetahuan dari web\nAIW: ...	AI dan Web (AIW)	1.0
Pembelajaran Transfer "Bebas Sumber" untuk Kelas Teks...	Pembelajaran Transfer\nPengambilan Data Tambahan\nT...	AIW: Akuisisi pengetahuan dari web\nAIW: ...	Algoritma Pembelajaran Mesin Baru (NMLA)	1.0
Generalisasi Serial Probabilistik ke Ra...	pilihan sosial\npemungutan suara\npembagian yang adil\n...	GTEP: Teori Permainan\nGTEP: Pilihan Sosial / teori ...	Teori Permainan dan Paradigma Ekonomi 1.0 (GTEM)	1.0
Variasi Leksikal Seumur Hidup di Media Sosial	Model Generatif\nJaringan Sosial\nUsia Ramalan	AIW: Personalisasi web dan pemodelan pengguna\nNL...	NLP dan Penambangan Teks (NLPTM)	1.0

Setelah mengembangkan sebuah makalah penelitian sebagai fungsi dari grup tempat makalah tersebut berada, dan mengaitkan sebuah bendera anggota dengannya, Ross menyadari bahwa sudah waktunya untuk mengonversi struktur data ini menjadi sebuah matriks. Untuk itu, ia mendefinisikan fungsi tersebut pada Daftar 4-6.

Daftar 4-6. Menambahkan Fungsi untuk Pembuatan Matriks

```
def matriks_dari_df(kereta_data):

    matriks = data_train.pivot_table(indeks = ['judul'], kolom=['grup'],


    matriks, x_cols
```

Selama bertahun-tahun, Ross telah menjadi ahli dalam tabel pivot di Microsoft Excel. Oleh karena itu, ia memutuskan untuk menggunakan metodologi yang sama pada Daftar 4-6 untuk transformasi kerangka data ke matriks. Oleh karena itu, ia menginisialisasi sebuah metode pada Daftar 4-6 dengan mengonversi data pelatihan menjadi pivot dengan judul sebagai indeks, setiap kelompok diwakili oleh kolom terpisah, yang berisi nilai-nilai dari fitur "bendera". Kemudian, ia memberi nilai 0 jika makalah penelitian dalam indeks tidak termasuk dalam kelompok tertentu. Ia memutuskan untuk menjalankan matriks tersebut dengan menggunakan metode yang didefinisikan pada Daftar 4-6.

Daftar 4-7. Ambil Matriks dan kolom x dari Metode matric_from_df

```
matriks, x_cols = matriks_dari_df(data_train)
matriks.head()
```

Tabel 4-4. *Pengamatan Awal Matriks*

kelompok	judul	AI dan Jaringan (AIW)	Aplikasi dan (APP)	Kognitif Pemodelan (CM)	Kognitif Sistem (CS) dan AI	Komputasi Keberlanjutan (CSAI)	Bermain Game dan Interaktif dan Hiburan (GPIE)	Permainan Teori Ekonomis Paradigma (GTEP)	Heuristis Pencarian dan Optimasi (H2SO)	Manusia-Perhitungan Sumber (HCC)	...
0	"Sumber Gratis" Transfer Pembelajaran untuk Kelas Teks...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1	Karakterisasi 0,0 Puncak Tunggal Tunggal...	dari	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...
2	A Metode Komputasi untuk (MSS, CoMSS) Partisi...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...

Ross sangat senang melihat hasilnya sesuai harapan. Untuk menguji pemahamannya, Ia memilih sebuah makalah penelitian dari Tabel 4-4 secara acak. Ia memilih "Metode Komputasi untuk (MSS.CoMSS)." Dengan melihat matriks pada Tabel 4-4, Ross menyimpulkan bahwa makalah tersebut termasuk dalam kelompok HSO dan tidak memiliki keanggotaan di kelompok lain.

Sekarang dia tahu apa itu pengelompokan, dan telah mengubah data ke dalam bentuk Karena mudah diterapkan, ia pun menerapkan pemodelan pengelompokan. Namun, pertama-tama ia harus menentukan metrik evaluasi yang dapat digunakan untuk mengevaluasi kebaikan model pengelompokan. Berbeda dengan teknik yang tersedia untuk mengevaluasi model regresi dan klasifikasi, tidak banyak teknik yang tersedia untuk mengevaluasi kebaikan model pengelompokan secara statistik. Alasannya adalah karena dalam regresi dan klasifikasi, data berlabel sudah tersedia dan menjadi dasar pelatihan model. Namun, hal ini tidak berlaku untuk pengelompokan.

Metrik Evaluasi Model Pengelompokan

Setelah melakukan riset, Ross menyadari bahwa tidak ada satu cara tunggal untuk mengevaluasi keluaran model pengelompokan. Ia menemukan dua pendekatan agar model tersebut berfungsi. Pendekatan pertama lebih bersifat teknis. Pendekatan ini menganggap suatu model pengelompokan cukup baik jika varians maksimum dijelaskan dalam setiap klaster, objek-objek di dalam klaster memiliki properti yang serupa, dan klaster-klaster berjauhan satu sama lain. Pendekatan kedua menunjukkan bahwa model tersebut kuat jika definisi klaster masuk akal secara intuitif.

Ross kini memiliki pemahaman yang jelas tentang pengelompokan dan metodologi untuk mengevaluasi keunggulannya. Oleh karena itu, ia mulai mencari model pengelompokan yang paling efektif dalam aplikasi terkait. Setelah riset yang mendalam, ia menghasilkan beberapa model pengelompokan.

Model Pengelompokan

Ross memutuskan untuk memulai dengan algoritma pengelompokan de facto (yaitu, pengelompokan k-means). Bagi Ross, memahami metodologi konvergensi klaster sangat penting sebelum menerapkannya pada dataset yang ada. Oleh karena itu, ia menyusun penjelasan tentang algoritma pengelompokan k-means.

Pengelompokan k-Means

Pengelompokan k-means membagi ruang data menjadi representasi sel Voronoi. Transformasi ini membagi observasi data menjadi k-kluster di mana setiap

pengamatan termasuk dalam klaster dengan nilai mean terdekat. Pengelompokan k-means dilakukan sebagai berikut:

1. k-centroid dipilih secara acak.
2. Setiap pengamatan terikat dengan centroid terdekat.
3. Centroid baru untuk setiap cluster dihitung ulang dengan
   mengambil nilai rata-rata dari pengamatan yang ada dalam setiap klaster.
4. Langkah 2 diulangi lagi.

Anda mungkin memperhatikan bahwa langkah 1 dan 3 mirip satu sama lain, perbedaannya adalah bahwa pada langkah 1, sentroid dipilih secara acak, sedangkan pada langkah 3, sentroid dihitung dengan mengambil rata-rata pengamatan dalam setiap klaster tersebut.

Kemudian, langkah 2 dan 4 merupakan pengulangan dari langkah yang sama. Langkah 3 dan 4 diulang hingga klaster konvergen—artinya, keanggotaan klaster tetap konstan untuk semua observasi dalam dataset. Hal ini memaksimalkan jarak antar klaster dan meminimalkan jarak antar klaster.

di antara observasi dalam setiap klaster. Pengelompokan k-means tidak selalu menemukan konfigurasi paling optimal sambil meminimalkan fungsi objektif global. Algoritme pengelompokan sangat sensitif terhadap bagaimana pusat klaster awalnya dipilih.

Ross kini sepenuhnya memahami bagaimana konvergensi dilakukan dalam algoritma pengelompokan k-means. Namun, ia tidak tahu berapa nilai k (yaitu, jumlah klaster yang harus ia tentukan di awal) yang optimal untuk data yang ada. Ia ingat pernah melihat artikel tentang pengelompokan k-means di LinkedIn. Oleh karena itu, ia mencari orang yang mengunggah artikel tersebut dan kemudian mengirim pesan kepadanya untuk meminta bantuan. Rekannya, Pollack, menyarankan agar Ross menggunakan metode berikut:

- Metode siku
- Varians dijelaskan
- Skor BIC

Dia memulai dengan melihat bagaimana metode Elbow bekerja dan kemudian menerapkannya untuk menemukan jumlah klaster yang optimal untuk model.

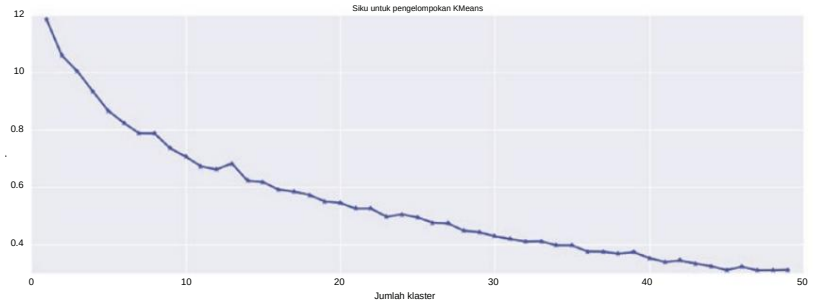
Metode Siku

Metode Elbow, menurut Ross, adalah persentase varians yang dijelaskan sebagai fungsi jumlah klaster. Metode ini menentukan seberapa besar varians marginal yang disumbangkan oleh klaster yang baru ditambahkan. Poin menariknya adalah klaster pertama akan menjelaskan varians maksimum dengan penurunan keuntungan marginal pada setiap penambahan klaster baru. Elbow akan menjadi titik di mana klaster baru akan menghasilkan penurunan marginal yang cukup besar, yang merupakan jumlah klaster optimal. Ross memutuskan untuk menerapkan metode Elbow pada dataset yang telah ditransformasikan matriks.

Daftar 4-8. Penerapan Metode Siku dan Penjelasan Varians pada Matriks Data

```
matriks, x_cols = matriks_dari_df(data_train)
X = matriks[x_kolom]

K = rentang(1,50)
KM = [KMeans(n_cluster=k).fit(X) untuk k dalam K]
centroid = [k.cluster_centers_ untuk k dalam KM]

D_k = [cdist(X, cent, 'euclidean') untuk persen dalam centroid]
dist = [np.min(D,sumbu=1) untuk D di D_k]
avgWithinSS = [jumlah(d)/X.shape[0] untuk d dalam dist]
```

```
gambar = plt.figure()
ax = fig.tambahkan_subplot(111)
kapak.plot(K, rata-rataDalamSS, 'b*-')
plt.grid(Benar)
plt.xlabel('Jumlah klaster')
plt.ylabel('Rata-rata jumlah kuadrat dalam klaster')
plt.title('Siku untuk pengelompokan KMeans')
```



Gambar 4-2. *Kurva metode siku sebagai fungsi jumlah klaster*

Dalam Listing 4-8, Ross mengingat pelatihan 49 model pengelompokan k-means, masing-masing memiliki nilai k yang berbeda, berkisar antara 1 hingga 50. Ia kemudian mengambil pusat klaster untuk model-model pengelompokan tersebut dan menentukan keanggotaan klaster untuk setiap model makalah penelitian. Terakhir, untuk setiap model, ia menghitung jumlah semua makalah penelitian dan membaginya dengan jumlah total makalah penelitian untuk mendapatkan rata-rata jumlah kuadrat dalam klaster. Ia kemudian memplot jumlah kuadrat klaster untuk semua 49 model klaster pada Gambar 4-2 untuk merumuskan metode Elbow.

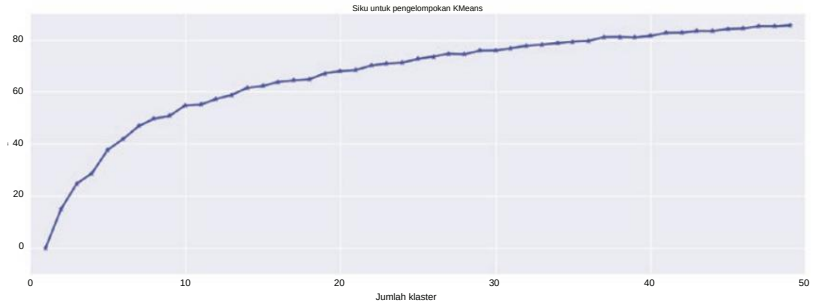
Sambil mengamati Gambar 4-2, Ross menyimpulkan bahwa siku terjadi pada k = 9. Namun, ia tidak yakin sampai lebih dari satu metode mencapai jumlah klaster yang sama. Dengan demikian, ia mulai memahami varians yang dijelaskan.

Varians Dijelaskan

Persentase varians yang dijelaskan, atau dengan kata lain uji-F, adalah rasio varians grup terhadap varians total. Di sini, sekali lagi, siku akan menentukan jumlah klaster optimal untuk model pengelompokan k-means. Ross mempraktikkan varians yang dijelaskan dengan menulis cuplikan kode pada Daftar 4-9.

Daftar 4-9. Penerapan Metode Siku dan Penjelasan Varians pada Matriks Data

```
matriks, x_cols = matriks_dari_df(data_train)
X = matriks[x_kolom]

K = rentang(1,50)
KM = [KMeans(n_cluster=k).fit(X) untuk k dalam K]
centroid = [k.cluster_centers_ untuk k dalam KM]

D_k = [cdist(X, cent, 'euclidean') untuk persen dalam centroid]
dist = [np.min(D,sumbu=1) untuk D di D_k]

wcss = [jumlah(d**2) untuk d dalam dist]
tss = jumlah(pdist(X)**2)/X.bentuk[0]
bss = tss - wcss

kIdx = 10-1

gambar = plt.figure()
ax = fig.tambahkan_subplot(111)
ax.plot(K, bss/tss*100, 'b*-')
plt.grid(Benar)
plt.xlabel('Jumlah klaster')
plt.ylabel('Persentase varians dijelaskan')
plt.title('Siku untuk pengelompokan KMeans')
```



Gambar 4-3. *Kurva varians yang dijelaskan sebagai fungsi jumlah cluster*

Ross ingat pernah menjelaskan Daftar 4-9 dengan kata-kata berikut:

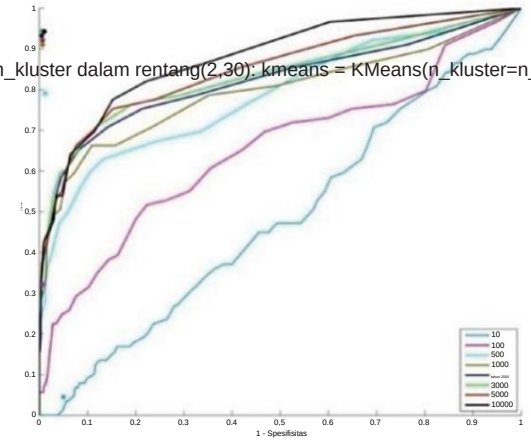
> *Setelah mendapatkan keanggotaan klaster dari setiap klaster makalah penelitian, jarak tersebut kemudian digunakan untuk menghitung jumlah kuadrat dalam klaster. Selanjutnya, jarak antar kolom dalam setiap makalah penelitian dihitung untuk menghitung jumlah kuadrat total. Terakhir, jumlah kuadrat dalam klaster dikurangi dari jumlah kuadrat total untuk mendapatkan jumlah kuadrat antar. Rasio antara jumlah kuadrat dan jumlah kuadrat total diplot pada Gambar 4-3.*

Bagi Ross Gambar 4-3 tampak seperti fungsi distribusi kumulatif eksponensial. Varians yang dijelaskan tampak halus dan terus meningkat seiring bertambahnya jumlah klaster. Hal ini menyulitkan Ross untuk menentukan siku. Setelah pertimbangan yang matang, Ross menyadari gradien mulai menghaluskan dari k = 9, serupa dengan yang ia amati dengan metode Siku.

Namun, dia tidak berhenti di situ dan terus mencari metode untuk membantu menentukan k optimal, dan dalam waktu singkat ia terpapar pada skor Bayesian Information Criterion (BIC).

Skor Kriteria Informasi Bayesian

BIC adalah teknik lain untuk memilih model terbaik dari sekumpulan model yang terbatas. Model dengan skor BIC terendah dianggap sebagai pemenang. Saat melatih model, terdapat kemungkinan tinggi untuk mendapatkan klaster yang akurat dengan menambahkan parameter tambahan; namun, ia tahu bahwa hal itu akan membuat model menjadi overfitting. BIC mengatasi masalah ini dengan menambahkan istilah penalti pada jumlah parameter dalam model. BIC bekerja dengan asumsi bahwa ukuran sampel harus jauh lebih besar daripada jumlah parameter dalam model. Oleh karena itu, BIC tidak cocok untuk model kompleks yang bekerja pada data berdimensi tinggi.



Gambar 4-4. *Contoh plot skor BIC*

Basis pengetahuan pada skor BIC membuat Ross mengurungkannya sebagai pilihan, karena skor BIC tersebut cocok untuk data berdimensi rendah, dan oleh karena itu transformasi matriks berarti bahwa sekarang data tersebut memiliki (jumlah kelompok + 1) dimensi, menjadikannya data berdimensi tinggi.

Dia penasaran untuk mengetahui apakah ada metode untuk menentukan jumlah optimal klaster, sebuah metode yang sesuai dengan cara kerja klaster k-means. Dengan kata lain, Ross ingin menemukan metode yang menentukan jumlah klaster optimal dengan jarak antar klaster maksimum dan jarak intra klaster minimum.

pencariannya membuatnya memeriksa makalah penelitian, di mana ia menemukan metode Silhouette.

Skor Siluet

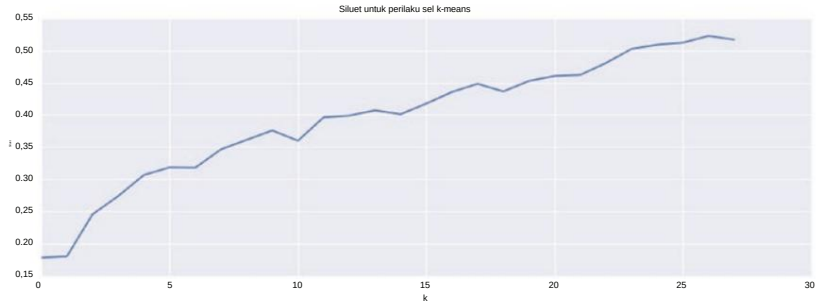
Skor Silhouette digunakan untuk mengukur seberapa dekat observasi dalam suatu klaster (yaitu, jarak intra-klaster (a)), dan seberapa berbeda klaster satu sama lain (yaitu, rata-rata jarak klaster terdekat (b)). Koefisien Silhouette dihitung sebagai berikut:

(ba) / maks (a, b)

Nilai terbaik adalah 1 dan yang terburuk adalah -1. Nilai koefisien 0 menunjukkan tumpang tindih klaster. Tantangan bagi Ross sekarang adalah memilih jumlah klaster yang menghasilkan skor Silhouette maksimum. Tanpa membuang waktu, ia menerapkan metode Silhouette pada model k-means, dalam rentang ukuran klaster 2 hingga 30.

Daftar 4-10. Membuat Plot Skor Siluet dari Matriks Data

```
s = []
untuk n_kluster dalam rentang(2,30): kmeans = KMeans(n_kluster=n_kluster) kmeans.fit(X) labels = kmeans.labels_ centroids = kmeans.cluster_centers_ s.append(silhouette_score(X, labels, metric='euclidean'))

plt.plot(s)
plt.ylabel("Siluet")
plt.xlabel("k")
plt.title("Siluet untuk perilaku sel K-means")
sns.despine()
```



Gambar 4-5. *Plot koefisien siluet sebagai fungsi jumlah klaster*

Ross bingung melihat plot pada Gambar 4-5 karena koefisien Silhouette tampaknya terus meningkat seiring bertambahnya jumlah klaster. Karena teori tersebut menyatakan jumlah klaster optimal adalah pada titik di mana koefisien Silhouette tertinggi, ia tidak dapat secara realistis memilih 27 sebagai jumlah klaster yang tepat karena alasan berikut:

- 27 klaster akan terlalu banyak untuk matriks data yang memiliki total 398 observasi.
- Dia secara realistis tidak dapat memasarkan 27 segmen berbeda dalam waktu yang dimilikinya terbatas.

Oleh karena itu, ia memutuskan untuk melanjutkan dengan sembilan klaster, yang ia tentukan dari Metode siku (yaitu, Gambar 4-3) dan varians dijelaskan (yaitu, Gambar 4-4).

Penerapan K-Means Clustering untuk Jumlah Optimal Gugusan

Ross menerapkan pengelompokan k-means dengan ukuran klaster 9 pada Daftar 4-11.

Daftar 4-11. Model Pelatihan k-means untuk Ukuran Klaster 9

```
matriks, x_cols = matriks_dari_df(data_train)
X = matriks[x_kolom]

klaster = KMeans(n_klaster = 9, keadaan_acak = 2)
matriks['cluster'] = cluster.fit_predict(X)
matriks.cluster.jumlah_nilai()
```

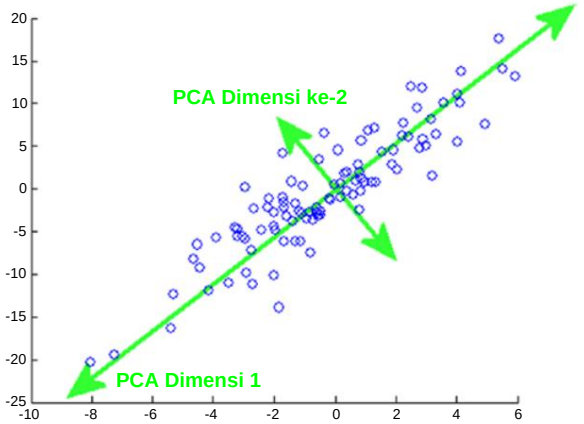
```
8 30

Nama: cluster, tipe data: int64
```

Pada Daftar 4-11, Ross memutuskan untuk mencetak jumlah makalah penelitian yang termasuk dalam masing-masing dari sembilan klaster. Ia memperhatikan bahwa kecuali dua klaster, yang masing-masing memiliki 80 observasi, tujuh klaster lainnya rata-rata memiliki 30 observasi.

Ross tertarik untuk melihat seperti apa klaster-klaster ini secara visual. Ia menyadari bahwa dimensi dalam kumpulan data merupakan fungsi dari jumlah kelompok yang berbeda. Ross tahu bahwa tidak ada paket dalam Python yang memadai untuk memplot gambar berdimensi setinggi itu. Sekalipun ia berhasil memplotnya, ia menyadari bahwa gambar tersebut akan terlalu rumit untuk dipahami secara intuitif. Oleh karena itu, ia mulai mencari metode yang dapat mengurangi jumlah dimensi. Ia tidak perlu menunggu lama untuk mempelajari analisis komponen utama (PCA).

Analisis Komponen Utama

PCA mengubah serangkaian pengamatan yang sangat berkorelasi menjadi serangkaian variabel yang tidak berkorelasi linear yang disebut komponen prinsip melalui transformasi ortogonal. Transformasi ini dilakukan sedemikian rupa sehingga komponen pertama memiliki varians tertinggi.



Gambar 4-6. *Ilustrasi PCA*

Komponen-komponen berikutnya mempunyai varians yang lebih rendah, dengan tetap mempertimbangkan kendala bahwa komponen tersebut ortogonal terhadap komponen-komponen sebelumnya seperti yang digambarkan pada Gambar 4-6. PCA sangat sensitif terhadap penskalaan relatif data. PCA dapat dianggap sebagai teknik reduksi dimensi yang mereduksi data berdimensi tinggi menjadi sejumlah komponen tetap.

Setelah membaca deskripsinya, Ross yakin bahwa dengan pengurangan dimensi dia melakukan hal yang benar. Oleh karena itu, tanpa membuang waktu lagi, ia menulis kode pada Listing 4-11 untuk mereduksi dimensi grup menjadi dua dimensi (yaitu, x, dan y). Ia memutuskan untuk mengubahnya menjadi dua dimensi agar mudah baginya untuk memplot klaster pada sumbu dua dimensi.

Daftar 4-12. Menggunakan PCA untuk Mengubah Fitur Terkait Grup menjadi Dua Komponen

```
pca = PCA(n_komponen=2)
matriks['x'] = pca.fit_transform(matriks[kolom_x])[:,0]
matriks['y'] = pca.fit_transform(matriks[x_cols])[:,1]
matriks = matriks.reset_indeks()

kluster_pelanggan = matriks[['judul', 'kluster', 'x', 'y']]
customer_clusters.head()
```

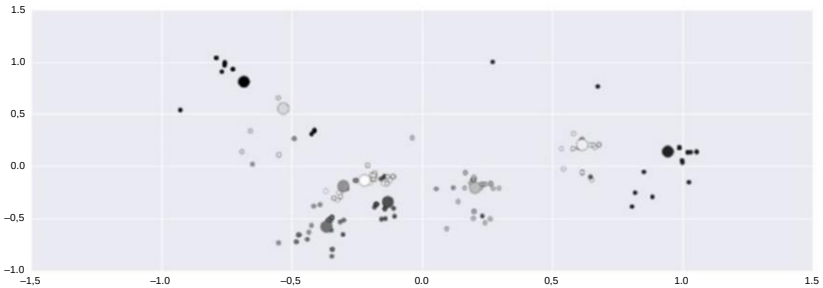
Tabel 4-5. *Cetak Cluster beserta Dua Komponen yang Baru Dibuat Menggunakan PCA*

kelompok	judul	gugus x		
0	Pembelajaran Transfer “Bebas Sumber” untuk Kelas Teks...	1	0,615810	-0,060295
1	Karakterisasi Puncak Tunggal Tunggal...	8	-0,756838	0.971322
2	Metode Komputasi untuk Partisi (MSS, CoMSS) ...	4	-0.287956	-0.216148
3	Dikotomi Kontrol untuk Aturan Penilaian Murni	2	-0.521295	0.570206
4	Formulasi Cembung untuk Multi-supervised Semi...	3	0.198578	-0.129668

Setelah melakukan transformasi pada Tabel 4-5, Ross sekarang harus memetakannya pada diagram sebar yang kodenya ia tulis pada Daftar 4-13.

Daftar 4-13. Merencanakan Gugus dalam Ruang Dua Dimensi

```
pusat_kluster = pca.transform(pusat_kluster_)
cluster_centers = pd.DataFrame(cluster_centers, kolom=['x', 'y'])
cluster_centers['cluster'] = rentang(0, len(pusat_cluster))

plt.scatter(kluster_pelanggan['x'], kluster_pelanggan['y'], s = 20, c=kluster_pelanggan['kluster'])

plt.scatter(pusat_kluster['x'], pusat_kluster['y'], s = 150, c=kluster_
pusat['gugus'])
```



Gambar 4-7. *Cluster dalam ruang dua dimensi*

Saat menuliskan kode pada Listing 4-13, Ross memastikan untuk menetapkan kode yang berbeda Label warna untuk setiap observasi klaster agar ia dapat melihat keanggotaan klasternya secara visual pada Gambar 4-7. Selain itu, ia juga memastikan untuk merepresentasikan pusat-pusat klaster. Keanggotaan klaster pada Gambar 4-7 tampak berbeda dan tidak tumpang tindih. Namun, mengingat tujuan awalnya untuk menemukan kata kunci yang terkait dengan setiap segmen, ia harus menggabungkan data yang telah direduksi dimensinya ke dalam data asli (yaitu, data sebelum transformasi matriks). Untuk tujuan tersebut, ia menuliskan skrip pada Daftar 4-14.

Daftar 4-14. Menggabungkan Matriks ke dalam Kerangka Data Asli

```
customer_clusters.columns.name = Tidak Ada
df = data_train.merge(customer_clusters, on='judul')
df.kepala()
```

Tabel 4-6. *Cetakan Observasi dari Kerangka Data Gabungan*

judul	kata kunci	topik	kelompok	cluster bendera x			_
Bayesian yang dikernelisasi Transfer Pembelajaran	lintas domain domain pembelajaran adaptasi\nkern...	APP: Biomedis / Bioinformatika\nNMLA: Bayesi...	Mesin Baru Algoritma Pembelajaran (NMLA)	1.0 1		0,613870	0,245408
Transfer “Bebas Sumber” Pembelajaran untuk Teks Kelas...	Transfer Pembelajaran\nData Tambahan Pengambilan_nT...	AIW: Akuisisi pengetahuan dari web\nAIW:...	AI dan Web (AIW)	1.0	1	0.615810	-0,060295
Transfer “Bebas Sumber” Pembelajaran untuk Teks Kelas...	Transfer Pembelajaran\nData Tambahan Pengambilan_nT...	AIW: Akuisisi pengetahuan dari web\nAIW:...	Mesin Baru Algoritma Pembelajaran (NMLA)	1.0	1	0.615810	-0,060295
Sebuah Generalisasi dari Serial Probailistik ke Ra...	pilihan sosial teori\npemungutan suara\nadil divisi\ns...	GTEP: Permainan Teori GTEP: Sosial Pilihan / Pemungutan Suara	Teori Permainan dan Paradigma Ekonomi (GTEP)	1.0	2	-0.521295	0.570206
Leksikal Seumur Hidup Variasi dalam Sosial Media	Mode generatif\nSosial Jaringan\nPrediksi Usia	AIW: Personalisasi web dan pemodelan pengguna\nNL...	NLP dan Penambangan Teks (NLPTM)	1.0	0	-0.183192	-0,090091

Ross memastikan untuk mencetak beberapa pengamatan pertama dari objek data gabungan ini. Dia Berhasil menggabungkan kedua representasi data tersebut karena kini data tersebut memiliki judul, kata kunci, topik, dan fitur gender dari dataset asli, serta fitur bendera, klaster, x, dan y dari dataset hasil reduksi dimensi. Ia kini harus menggunakan dataset gabungan ini untuk menghasilkan kata kunci bagi setiap segmen. Ia memutuskan untuk melakukannya dengan menggunakan wordcloud.
Awan kata akan menampilkan semua kata yang terkait dengan klaster tertentu, dengan ukuran fon kata yang mewakili frekuensi kemunculannya dalam fitur pivotal. Ia berencana untuk menjaga fitur pivotal tetap acak, dan fitur pivotal ini akan digunakan untuk memasok kata ke awan kata. Ia memulai dengan menulis sebuah metode pada Daftar 4-15 untuk menghasilkan awan kata.

Daftar 4-15. Membuat Fungsi untuk Menghasilkan Wordcloud

```
def wordcloud_object(string_kata):

FONT_ROOT = './fonts/' kembalikan
        wordcloud = WordCloud(font_path=FONT_ROOT +
        'arial.ttf',stopwords=STOPWORDS, background_color='hitam',
        lebar=1200, tinggi=1000).generate(' '.join(string_kata))
wordcloud
```

Ross menunjukkan bahwa prasyarat untuk kode pada Listing 4-15 adalah folder bernama Fonts harus dibuat di repositori yang sama dengan skrip itu sendiri, dan berkas font Arial harus dimasukkan ke dalamnya. Untuk pengujian, ia merekomendasikan untuk memasukkan input string apa pun sebagai parameter ke fungsi ini agar dapat menghasilkan wordcloud.

Ross melanjutkan lebih jauh dan menulis kode pada Listing 4-16 untuk menghasilkan wordcloud untuk masing-masing dari sembilan kluster.

Daftar 4-16. Membuat Fungsi untuk Memplot Wordcloud untuk Setiap Cluster

```
def plot_wordcloud(df, cluster, pivot):

fig = plt.figure(figsize=(15,29.5)) untuk kluster dalam rentang(kluster):

untuk x di df[df['cluster']==cluster][pivot]: coba: List_.extend(x.split('\n')) kecuali: lulus

jika Daftar_: ax

= fig.add_subplot(5,2,cluster+1) wordcloud = wordcloud_object(Daftar_)

plt.title('Cluster: %d'%(cluster+1)) ax.imshow(wordcloud) ax.axis('off')
```

Sebelum melanjutkan, Ross menunjukkan bahwa salah satu parameter dalam plot_wordcloud Metode adalah parameter pivot. Pivot adalah fitur yang akan digunakan untuk mengambil kumpulan kata untuk membentuk awan kata.

Ross sangat antusias melihat apa yang akan digambarkan oleh awan kata tersebut. Oleh karena itu, tanpa menunggu lebih lama lagi, ia menulis kode pada Daftar 4-17 untuk memanggil metode yang diinisialisasi pada Daftar 4-16, dan memasukkan "kata kunci" sebagai fitur penting.

Daftar 4-17. Menghasilkan Wordclouds dari Fitur Bernama 'Kata Kunci'

```
plot_wordcloud(df, cluster.n_cluster, 'kata kunci')
```


Gambar 4-8. *Wordcloud untuk setiap cluster yang dihasilkan dari kata kunci*

Setelah melihat Gambar 4-8, Ross memutuskan untuk mencoba kemampuan deduktifnya dan mendefinisikan sembilan klaster sebagai berikut:

- Klaster 1: Makalah yang membahas tentang pencarian dan robotika
- Klaster 2: Makalah yang membahas secara mendalam tentang pembelajaran dan pengoptimalan model
- Klaster 3: Topik penerapan analisis data dalam permainan dan analisis media sosial
- Klaster 4: Topik pengenalan gambar, robotika, dan media sosial analitik
- Klaster 5: Topik pemrograman linier dan pencarian
- Klaster 6: Makalah tentang model berbasis penalaran
- Klaster 7: Makalah tentang penerapan ilmu data dalam grafik sosial dan media daring lainnya
- Klaster 8: Topik yang berkisar pada grafik pengetahuan
- Klaster 9: Makalah yang berfokus pada teori permainan dan keamanan data

Ross menunjukkan fakta bahwa kata-kata dalam wordcloud bersifat unigram (yaitu, kata tunggal). Ia puas dengan hasil sejauh ini, tetapi menurutnya beberapa klaster memiliki istilah yang tumpang tindih; misalnya, klaster 1 dan 5 memiliki "pencarian" sebagai istilah yang tumpang tindih. Ross merasa penting untuk menghilangkan tumpang tindih tersebut agar kata kunci dan karakteristik klaster seunik mungkin. Oleh karena itu, ia berencana untuk mendalami detailnya secara mendalam untuk melihat apakah kedua klaster tersebut menyentuh topik SEO yang sama atau berbeda. Ia memulai dengan mendefinisikan sebuah metode pada Listing 4-18 yang akan mengambil istilah (misalnya, "pencarian") sebagai parameter dan mengembalikan kata kunci untuk setiap klaster yang memiliki istilah tersebut.

Daftar 4-18. Tentukan Metode untuk Menemukan Kata Kunci Lengkap untuk Klaster dan Unigram yang Diberikan

```
def perform_cluster_group_audit(cluster, istilah):
untuk berkelompok dalam kelompok:

df_cluster = df[df['cluster'] == cluster]
cetak 'Nomor klaster: %d'%(klaster + 1)

kata kunci = daftar(df_cluster['kata kunci']) kata kunci = [kata kunci.pisah('\n') untuk kata kunci dalam kata kunci]
kata kunci = [item untuk subdaftar dalam kata kunci untuk item dalam subdaftar]
kata kunci = [kata kunci.lower() untuk kata kunci dalam kata kunci jika istilah dalam
    kata kunci.lebih rendah()]
frekuensi_kata kunci = {x:jumlah_kata_kunci(x) untuk x dalam kata kunci}
cetak diurutkan(frekuensi_kata_kunci.item(), kunci=operator.pengambil_item(1),
    (terbalik=Benar)
cetak '\n'
```


Ross menunjukkan baris kedua terakhir dari Daftar 4-18 yang digunakan untuk mengurutkan kata kunci dalam urutan menurun berdasarkan frekuensinya. Sekarang saatnya menggunakan metode yang dijelaskan dalam Daftar 4-18. Ia memulai dengan melihat topik-topik yang membedakan klaster 1 dan 5 dalam aspek "pencarian".

Daftar 4-19. Menggunakan Fungsi untuk Menemukan Kata Kunci untuk Pencarian di Klaster 0 dan 4

```
melakukan audit_grup_kluster([0,4], 'cari')
```

Keluaran

```
Nomor klaster: 1
[('pencarian heuristik', 7), ('pencarian terbaik pertama yang rakus', 4), ('pencarian pohon
Monticello', 2), ('pencarian suboptimal terbatas', 1), ('pencarian waktu nyata', 1), ('pencarian
terbaik pertama', 1), ('pencarian inkremental', 1), ('pencarian paralel', 1), ('pencarian
kesamaan', 1), ('pencarian heuristik suboptimal', 1), ('pencarian yang berpusat pada agen',
1), ('pencarian tetangga terdekat perkiraan', 1), ('pencarian hierarkis', 1)]


Jumlah klaster: 5
[('pencarian', 3), ('pencarian heuristik', 3), ('pencarian lokal', 2), ('pencarian lokal stokastik',
2), ('dan/atau pencarian', 2)]
```

Ross menjelaskan bahwa klaster 1 dan 5 disebut sebagai 0 dan 4 karena indeks daftar mulai dari 0. Keluaran dari Daftar 4-19 menggambarkan topik-topik di sepanjang pencarian heuristik yang ditangkap oleh kedua segmen. Ia memperhatikan bahwa klaster 5 memiliki topik pada pencarian lokal dan klaster 1 memiliki topik pada algoritma pencarian lainnya.

Setelah mengamati Gambar 4-8, Ross memperhatikan bahwa klaster 3, 4, dan 7 berbagi topik di sepanjang ruang media sosial. Oleh karena itu, ia memutuskan untuk menyelidiki titik-titik perbedaan dengan menggunakan fungsi yang telah dijelaskan sebelumnya pada Daftar 4-20.

Daftar 4-20. Menggunakan Fungsi untuk Menemukan Kata Kunci Sosial di Klaster 2, 3, dan 6

```
melakukan audit_kelompok_klaster([2,3,6], 'sosial')
```

Keluaran

```
Jumlah klaster: 3
[('pilihan sosial komputasional', 11), ('teori pilihan sosial', 2), ('skema keputusan sosial', 2),
('pilihan sosial acak', 1)]

Jumlah klaster: 4
[('media sosial', 5), ('spammer sosial', 2), ('klasifikasi citra sosial', 2)]

Nomor klaster: 7
[('jejaring sosial', 6), ('jejaring sosial', 3), ('infeksi sosial', 3), ('analisis jaringan sosial',
2), ('jaringan sosial berbasis lokasi', 2), ('pengaruh sosial', 2), ('dinamika sosial', 1),
('penjelasan sosial', 1)]
```