Analiza i klasterovanje YouTube Trending video snimaka

Analiza i klasterovanje YouTube Trending video snimaka

1. Uvod

Tema ovog rada je **klasterizacija trending YouTube videa**. Cilj istraživanja je otkrivanje obrazaca, kao i izvođenje zaključaka i statističkih uvida na osnovu skupa podataka o YouTube trending listi za Nemačku (više detalja o samom skupu ispod).

Projekat koristi pet različitih algoritama za klasterovanje: `K-Means`, `Agglomerative Clustering`, `DBSCAN`, `GMM (Gaussian Mixture Models)` i `Spectral Clustering`. Važno je napomenuti da su svi eksperimenti reproduktivni zahvaljujući fiksiranom generatoru slučajnih brojeva (`RANDOM_STATE = 42`).

Projekat je organizovan modularno, pri čemu je svaki modul zadužen za određeni deo procesa (učitavanje podataka, pretprocesiranje, analiza rezultata i slično), dok glavni fajl služi samo za pozivanje odgovarajućih funkcija i prikaz rezultata.

2. Opis, prikupljanje i pretprocesiranje podataka

2.1. Opis i prikupljanje podataka

Za potrebe istraživanja korišćen je skup podataka **„Trending YouTube Video Statistics"**, specifično fajl `DEvideos.csv`, koji sadrži informacije o video snimcima koji su dospeli na *trending* listu u Nemačkoj. Podaci se dinamički preuzimaju prilikom pokretanja programa, a za njihovo preuzimanje i učitavanje zadužen je modul `download_dataset.py`.

Originalni skup podataka u neobrađenom obliku sadrži 16 atributa (kolona) i 40 840 redova. Nakon procesa pretprocesiranja i obogaćivanja dodatnim atributima (detaljnije opisano u nastavku), broj atributa raste na 102, dok se broj redova smanjuje na 29 627.

2.2. Pretprocesiranje podataka

Logika vezana za pretprocesiranje odvojena je u dva fajla - `data_preprocessing.py` i `feature_engineering.py`. Prvi na red stupa `data_preprocessing.py`, koji obavlja sledeće korake:

- **Uklanjanje duplikata** - Funkcija `remove_duplicates` uklanja duplikate na osnovu atributa `video_id`, koji predstavlja jedinstveni identifikator svakog videa. Ovaj korak je posebno važan jer isti video može više dana zaredom da bude u trendingu. Uklonjeno je 11 213 redova (oko 27% skupa podataka), što ukazuje na dominaciju manjeg broja videa u trendingu.

- **Popunjavanje nedostajućih vrednosti** - Funkcija `handle_missing_values` popunjava numeričke atribute medijanom, tekstualne atribute praznim stringom, a logičke atribute vrednostima iz susednih redova kako bi se očuvala struktura podataka.
- **Obrada datuma** - Kolone `trending_date` i `publish_time` konvertuju se iz tekstualnog u pravi datumski format (`datetime`) kako bi bilo moguće računati vremenske intervale i trajanja.
- **Brisanje nebitnih atributa** - Uklanjaju se atributi kao što su `video_id`, `link` i slični koji nisu bitni jer korisnik svakako ne može da ih vidi.

Sirovi podaci sami po sebi nisu toliko korisni, ali mi možemo da obogatimo skup izvođenjem novih atributa iz postojećih. Npr. likes i views sami po sebi jesu dobri indikatori, ali mozda je bolji indikator likes/views koji će nam dati bolji indikator engagmenta. Ovo je bio samo jedan primer moguće transformacije, proces obogaćivanja skupa podataka sastoji se iz više funkcija koje se pozivaju sledećim redosledom:

- **`create_temporal_features`** - Iz kolona `publish_time` i `trending_date` izdvajaju se numerički atributi kao što su sat objavljivanja, dan u nedelji, indikator vikenda (`is_weekend`) i `trending_duration`, odnosno vreme proteklo od objave do ulaska u trending.
- **`create_engagement_features`** - Računaju se odnosi (`likes_per_view`, `comments_per_view`, `dislike_ratio`) i primenjuje logaritamska transformacija (`np.log1p`) nad brojem pregleda i lajkova (`views_log`, `likes_log`) radi smanjenja uticaja ekstremnih vrednosti i bolje reprezentacije engagement-a korisnika.
- **`create_text_features`** - Analizira se struktura naslova kroz atribute kao što su dužina naslova, broj reči, odnos velikih slova (`caps_ratio`, ovo posmatramo kako bi uočili *clickbait*), broj uzvičnika i upitnika, kao i osnovni sentiment.
- **`create_statistical_features`** - Za svaki video računaju se statističke mere (prosek, standardna devijacija, minimum i maksimum) nad glavnim metrikama (`views`, `likes`, `comments`) kako bi se dobio sažetak performansi.
- **`create_category_features`** - Primena One-Hot Encoding-a nad kolonom `category_id` pretvara kategorijske vrednosti u skup binarnih kolona koje predstavljaju pripadnost određenoj kategoriji.
- **`create_tfidf_features`** - Nad kolonom `tags` primenjuje se TF-IDF vektorizacija pri čemu se bira 50 najvažnijih ključnih reči iz skupa podataka i za njih se kreiraju dodatne kolone. Neki tagovi kao što su HD, 2014 (ili bilo koja godina), viral i sl. su jako česti i ne želimo da im dajemo mnogo na značaju.
- **`get_numeric_features`** - Uklanjaju se sve originalne tekstualne i datumske kolone (`title`, `tags`, `publish_time` i sl.), čime u skupu ostaju isključivo numerički atributi generisani u prethodnim koracima.
- **`scale_features`** - Primenjuje se `StandardScaler` transformacija nad svim kolonama jer neki algoritmi koji su bazirani na distanci (kao što je K-Means na primer) zahtevaju da atributi budu na istoj skali.

3. Modeli, treniranje modela i evaluacija

U cilju detaljne analize podataka i ispunjenja zahteva projekta, modeli za klasterovanje trenirani su nad tri različite verzije skupa atributa kako bi se ispitao uticaj dimenzionalnosti i selekcije atributa na kvalitet dobijenih klastera:

1. **Full Features** - kompletan skup sa 102 atributa dobijen nakon procesa pretprocesiranja i obogaćivanja podataka.
2. **PCA (Principal Component Analysis)** - redukovani skup atributa koji zadržava 95% ukupne varijanse originalnih podataka.
3. **SelectKBest** - skup od 50 statistički najznačajnijih atributa odabranih metodom `f_classif`.

Na svakom od navedenih skupova trenirano je pet različitih algoritama za klasterovanje: `K-Means`, `Agglomerative Clustering`, `DBSCAN`, `Gaussian Mixture Models (GMM)` i `Spectral Clustering`. Kriterijum za biranje baš ovih algoritama je bio da se razlikuju po principu rada (particioni, hijerarhijski, probabilistički i zasnovani na gustini).

3.1 Metodologija treniranja

Podaci su prvo standardizovani kako bi rezultati bili međusobno uporedivi i kako različiti numerički opsezi atributa ne bi uticali na proces klasterovanja. Parametri algoritama podešavani su eksperimentalno, uz fokus na stabilnost rešenja i to da dobijeni klasteri imaju smisla za interpretaciju.

Da bi eksperimenti mogli lako da se ponove, u svim algoritmima sa slučajnom inicijalizacijom korišćen je fiksni generator slučajnih brojeva (RANDOM_STATE = 42).

3.2 Metodologija evaluacije

Pošto je klasterovanje oblik nenadgledanog učenja, kvalitet modela procenjivan je pomoću internih metrika koje mere koliko su klasteri kompaktni i koliko su međusobno jasno razdvojeni:

- **Silhouette Score** pokazuje koliko je svaki objekat sličan elementima u svom klasteru u poređenju sa drugim klasterima - veće vrednosti znače bolje razdvajanje.
- **Davies-Bouldin** indeks posmatra odnos između rasipanja unutar klastera i udaljenosti između klastera - manje vrednosti ukazuju na kvalitetnije klastere.
- **Calinski-Harabasz** indeks meri odnos između varijacije između klastera i unutar njih - veće vrednosti znače bolje definisane klastere.

Korišćenjem sve tri metrike dobija se potpunija slika o performansama modela iz različitih uglova.

3.3 Poređenje algoritama na kompletnom skupu atributa

Analiza rezultata na skupu **Full Features** pokazala je da algoritmi prilično variraju po performansama. Hijerarhijski pristup se pokazao kao najstabilniji, dok su oni algoritmi koji su osetljivi na oblik raspodele i outliere imali slabije rezultate.

Aglomerativno klasterovanje sa Ward metodom povezivanja ostvarilo je najbolji balans između kompaktnosti i razdvojenosti klastera, što sugeriše da u podacima postoji hijerarhijska struktura.

DBSCAN je imao dobre metrike, ali je značajan deo podataka označio kao šum, što mu smanjuje praktičnu vrednost, zbog toga je diskvalifikovan.

K-Means i GMM su pokazali slabije rezultate, verovatno zbog pretpostavke o sferičnom obliku klastera i osetljivosti na ekstremne vrednosti, dok je spektralno klasterovanje dalo solidne rezultate, ali uz veću računarsku složenost.

3.4 Uticaj redukcije dimenzionalnosti

Poređenje performansi najboljeg algoritma nad tri varijante skupa atributa pokazuje da korišćenje kompletnog skupa daje najkvalitetnije razdvajanje klastera.

Primena PCA metode dovodi do blagog pada performansi, ali značajno smanjuje dimenzionalnost i ubrzava treniranje, što predstavlja dobar kompromis između brzine i tačnosti.

Sa druge strane, pristup SelectKBest pokazao je dosta slabije rezultate, što ukazuje da sama selekcija atributa bez transformacije prostora nije dovoljna da očuva kompleksnu strukturu podataka potrebnu za kvalitetno klasterovanje.

3.5 Izbor finalnog modela (pobednika)

Na osnovu dobijenih rezultata, kao finalni model izabran je **Aglomerativni algoritam treniran nad kompletnim skupom atributa**, jer pruža najbolji odnos između kvaliteta klastera i interpretabilnosti rezultata. Ovaj pristup omogućava najpreciznije otkrivanje prirodnih grupa u podacima i predstavlja najpogodniju osnovu za dalju analizu i profilisanje klastera.

3.6 Vizuelna analiza

Radi dodatne interpretacije rezultata, generisane su vizuelizacije u prostoru nižih dimenzija korišćenjem PCA projekcija u dve i tri dimenzije, koje omogućavaju intuitivan uvid u raspodelu podataka i međusobni položaj klastera.

Takođe je analizirana matrica korelacije atributa, čime su identifikovane jake veze između pojedinih metrika, posebno između pregleda i interakcija, što dodatno potvrđuje konzistentnost podataka i opravdanost primenjenih transformacija.

4. Rezultati i analiza

4.1 Kvantitativno poređenje algoritama

Performanse algoritama upoređene su korišćenjem standardnih internih metrika validacije.

Algoritam	Silhouette Score	Davies-Bouldin	Napomena
Aglomerativno	**0.2583**	**1.8254**	**Najbolji balans.** Jasno definisani klasteri.

Algoritam	Silhouette Score	Davies-Bouldin	Napomena
DBSCAN	0.3617	0.7561	Matematički najbolji skor, ali je **odbacio >50% podataka kao šum**, što smanjuje praktičnu primenljivost.
Spektralno	0.1658	2.5199	Stabilni rezultati uz veću računarsku složenost.
KMeans	0.0665	2.6497	Slabiji rezultat zbog nesferičnog oblika klastera.
GMM	0.0101	5.8799	Najlošiji rezultat, visoka osetljivost na outliere.

Na prvi pogled može delovati da je DBSCAN najbolji zbog visokog Silhouette skora, ali detaljnijom analizom vidi se da model označava više od polovine podataka kao šum. To znači da zapravo prepoznaje samo najekstremnije primere i ignoriše veliki deo sadržaja, zbog čega nije praktično rešenje za ovaj problem.

Aglomerativno klasterovanje se pokazalo kao najstabilnije jer daje dobar balans između razdvojenosti i pokrivenosti podataka, bez velikog odbacivanja uzoraka. Upravo zbog toga je izabrano kao finalni model.

4.2 Kako izgledaju klasteri u praksi

Najbolji model je identifikovao četiri prirodne grupe videa koje imaju smisla i iz realne perspektive gledanja YouTube sadržaja.

- **Mainstream klaster (oko 93% podataka)** - Ovo je daleko najveća grupa i predstavlja "tipične" trending videe - vesti, vlogove, intervjue, recenzije, gaming i sličan sadržaj. Njihove metrike su uglavnom oko proseka i oni čine osnovu trending liste. Ovo je bilo potpuno očekivano jer većina sadržaja (tj. većina koja ljudima dospeva na *for you* page) na platformi spada u ovu kategoriju.
- **Film & entertainment klaster (oko 6.4%)** - Jasno izdvojena grupa videa vezanih za filmove, trejlere i zabavni sadržaj. Karakterišu ih specifični tagovi i relativno visoko angažovanje, ali ne ekstremno kao kod muzičkih hitova. Ovaj klaster pokazuje koliko tekstualne karakteristike doprinose razdvajanju tematskih celina.
- **Niche / how-to klaster (oko 0.4%)** - Mala ali jasno definisana grupa videa sa vrlo specifičnim sadržajem, često tutorijali ili vlogovi sa naslovima tipa „Top 10", „5 tips" i slično. Imaju specifične tagove i drugačiji obrazac angažovanja publike. Ovaj klaster je zanimljiv jer pokazuje da i mali segmenti mogu imati prepoznatljiv obrazac.
- **Mega-viral music klaster (oko 0.3%)** - Vrlo mala grupa ekstremnih outliera sa ogromnim brojem pregleda i interakcija. Uglavnom su u pitanju muzički spotovi velikih izvođača koji značajno odskaču od ostatka skupa podataka. Ovaj nalaz je očekivan i direktno potvrđuje "power law" prirodu popularnosti na YouTube-u.

4.3 Šta je bilo očekivano, a šta nije

Očekivano je bilo da postoji jedan dominantan klaster sa prosečnim vrednostima i nekoliko manjih grupa koje predstavljaju specifične tipove sadržaja. Takođe je bilo očekivano da algoritmi poput K-Means imaju problema zbog ekstremnih vrednosti i nesferičnog oblika klastera.

Pomalo iznenađujuće je koliko su granice između klastera „meke", što se vidi po relativno niskim Silhouette vrednostima kod svih algoritama. Ovo zapravo ima smisla jer se tipovi sadržaja na YouTube-u često preklapaju i ne postoje jasno odvojene kategorije.

Još jedan zanimljiv nalaz je koliko su tekstualni atributi (posebno TF-IDF tagovi) doprineli razdvajanju klastera. Bez njih bi se većina videa grupisala samo na osnovu popularnosti, što ne bi dalo ovako jasnu tematsku podelu.

4.4 Opšti utisci iz analize

Rezultati jasno pokazuju da trending sadržaj nema uniformnu strukturu već se sastoji od jedne velike „osnovne mase" videa i nekoliko manjih grupa koje značajno odstupaju po karakteristikama.

Takođe se vidi da algoritmi koji ne prave pretpostavke o obliku klastera bolje odgovaraju ovakvom tipu podataka, dok jednostavniji modeli ne uspevaju da uhvate kompleksnu strukturu.

Sve u svemu, dobijeni klasteri imaju smisla i iz statističke i iz intuitivne perspektive, što potvrđuje da je pipeline pretprocesiranja i izbora atributa uspešno pripremio podatke za analizu.

5. Zaključak

Cilj ovog rada bio je da se kroz proces klasterovanja otkriju obrasci u trending YouTube videima i bolje razume kako se različiti tipovi sadržaja međusobno razlikuju. Korišćenjem kombinacije pažljivog pretprocesiranja, obogaćivanja atributa i primene više algoritama za klasterovanje, dobijen je jasan uvid u strukturu podataka.

Rezultati su pokazali da trending sadržaj nije homogen, već se sastoji od jedne velike grupe „tipičnih" videa i nekoliko manjih, ali jasno prepoznatljivih grupa koje se izdvajaju po tematici ili nivou popularnosti. Posebno se izdvajaju ekstremno viralni muzički videi, kao i specifične niše poput tutorijala i filmskog sadržaja.

Od testiranih algoritama, aglomerativno klasterovanje se pokazalo kao najstabilnije i najinterpretabilnije rešenje, jer najbolje hvata hijerarhijsku strukturu podataka i ne nameće stroge pretpostavke o obliku klastera. Takođe se pokazalo da tekstualne karakteristike (posebno tagovi) imaju ključnu ulogu u razdvajanju tematskih grupa.

Iako su dobijeni rezultati smisleni i konzistentni, treba imati u vidu da granice između klastera nisu strogo definisane, što je očekivano za realne podatke sa društvenih mreža gde se različiti tipovi sadržaja često preklapaju.

6. Tehnički detalji i struktura projekta

Projekat je organizovan modularno kako bi pipeline bio jasan, pregledan i lako reproduktivan. Svaki deo procesa (učitavanje podataka, pretprocesiranje, treniranje modela i vizualizacija) nalazi se u zasebnom modulu sa jasno definisanom odgovornošću.

Struktura peojekta

```
project/
│
├── data/
│   ├── raw/                    # Originalni dataset
│   └── processed/              # Dataset nakon pretprocesiranja
│
├── src/
│   ├── download_dataset.py     # Preuzimanje i učitavanje podataka
│   ├── data_preprocessing.py   # Pretprocesiranje podataka
│   ├── feature_engineering.py  # Obogacivanje podataka
│   ├── clustering.py           # Treniranje modela
│   ├── evaluation.py           # Računanje metrika
│   ├── visualization.py        # Generisanje grafika
|       ├── main.py                 # Pokretanje celog pipeline-a
│
├── visualizations/             # Sačuvani grafici
├── results/                    # Generisani summary izveštaji
├── models/                     # Sačuvani modeli
├── requirements.txt
└── README.md
```

Generisanje chartova i summary

Svi chartovi generišu se u modulu `visualization.py`. Prilikom pokretanja pipeline-a, grafici se automatski čuvaju u direktorijumu `visualizations/`.

Summary rezultata (metrike modela, osnovne statistike i poređenja) generiše se pomoću `evaulation.py` modula i čuva u direktorijumu `results/` kao tekstualni ili CSV fajl.

Setupovanje i pokretanje projekta

Napomena. Koraci ispod su primarno namenjeni za macOS i Linux. U slučaju Windowsa verovatno nije potrebno praviti virtuelno okruženje i dovoljno je pokrenuti iz nekog IDE, bez obzira na to kreiranje virtuelnog okruženja je svakako preporuceno da ne bi dolazilo do konflikta između različitih verzija biblioteka za različite projekte.

Kloniranje projekta:

```
git clone https://github.com/AleksaVukadinovic/Trending-YouTube-Video-Statistics.git
cd Trending-YouTube-Video-Statistics
```

Kreiranje virtuelnog okruženja:

```
python -m venv venv
```

Aktivacija okruženja:

Linux / macOS:

```
source venv/bin/activate
```

Windows:

```
venv\Scripts\activate
```

Sve potrebne biblioteke navedene su u fajlu `requirements.txt`. Instalacija se vrši komandom:

```
python3 -m pip install --upgrade pip # nije neophodno, ali pozeljno
pip install -r requirements.txt
```

Glavna ulazna tačka je `main.py`. Pokretanjem ove skripte izvršava se ceo pipeline od početka do kraja. Može se pokrenuti kroz neki IDE kao što je PyCharm (preporučeno pošto je i sam projekat rađen u njemu), Spyder itd. Takođe se može pokrenuti iz terminala preko komande:

```
python3 main.py
```