

Tema 2 Inteligenta Artificiala
~ *Introducere în ML* ~

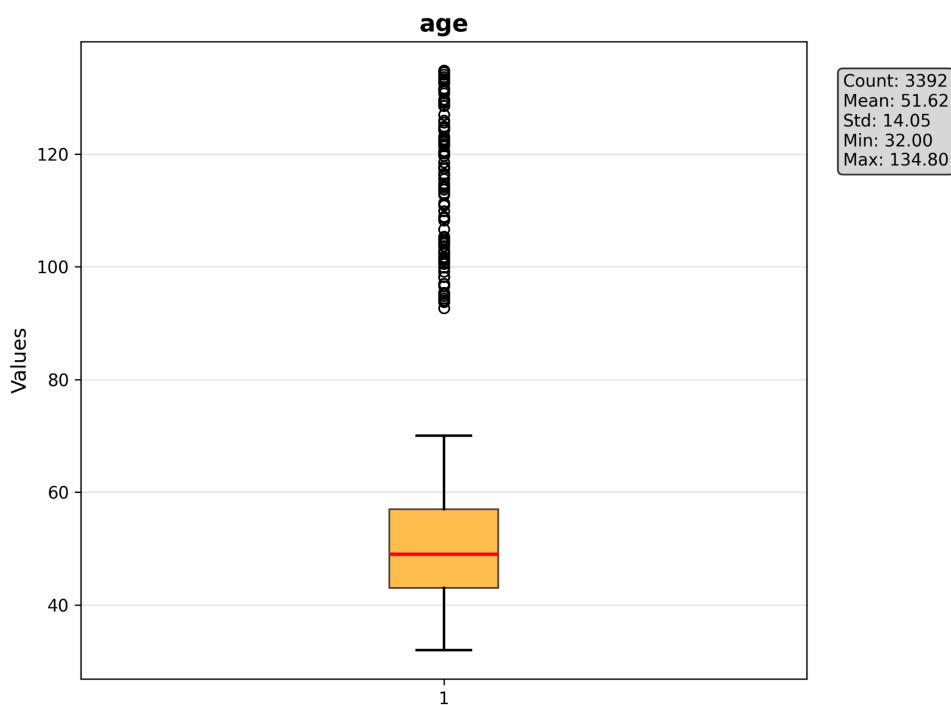
3.1. Analiza Datelor (*'analyze_data.py'*)

Primul pas al acestei parti a constat in extragerea si retinerea datelor de train si de test pentru cele doua seturi de date referitoare la popularitatea stirilor in mediul online si la riscul de dezvoltare al bolilor coronariene. Pentru aceasta am folosit functiile `load_heart_set()` care retine intr-un dictionar cele doua tipuri de date: cel pentru antrenare si cel folosit pentru aplicarea algoritmilor si `load_news_set()` cu aceeasi functionalitate. Cele doua dictionare rezultate sunt `'heart_set` (heart_disease_dataset) si `news_set` (news_popularity_dataset).

Urmatorul pas a fost reprezentat de distingerea intre atributele numerice continue si cele ordonale / discrete. Pentru aceasta am consultat anexa documentatiei temei cu tipurile de date specificie atributelor fiecarui set. Astfel, am delimitat `news_categorical_cols` / `heart_categorical_cols` si `news_numerical_cols` / `heart_numerical_cols`. Am salvat aceste coloane pentru `heart_set['heart_train']` si `news_set['news_train']` pentru a putea lucra cu ele in viitor.

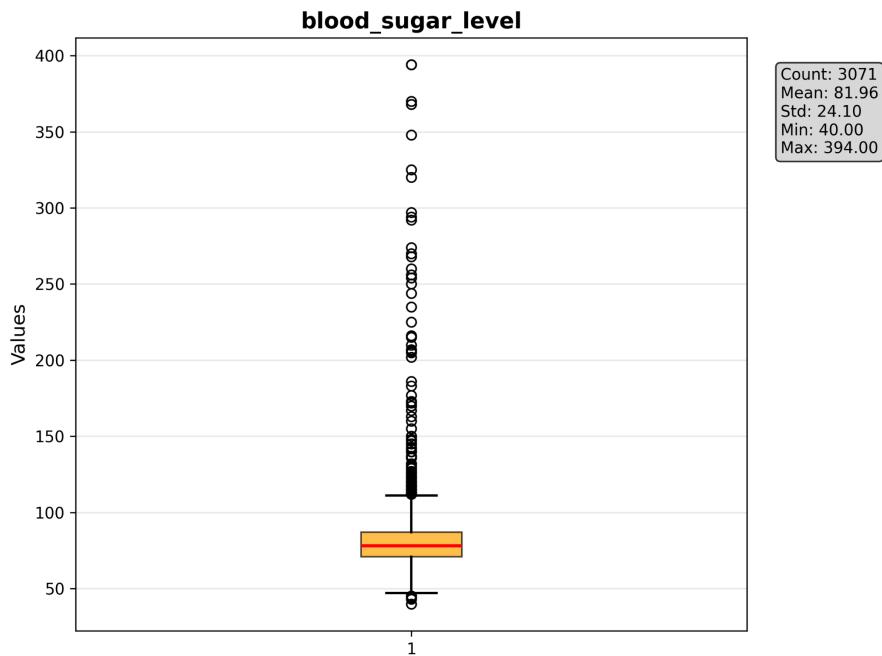
Datele fiind delimitate, am putut incepe analiza propriu-zisa a celor numerice si a celor discrete. Functia utilizata pentru analiza datelor numerice este `analyze_num_data` si calculeaza statistici descriptive (medie, std, min, max, quartile) pentru fiecare coloana numerica, genereaza boxplot-uri individuale pentru fiecare atribut numeric si salveaza graficele ca fisiere png.

Grafice pentru analiza datelor numerice referitoare la boala coronariana:



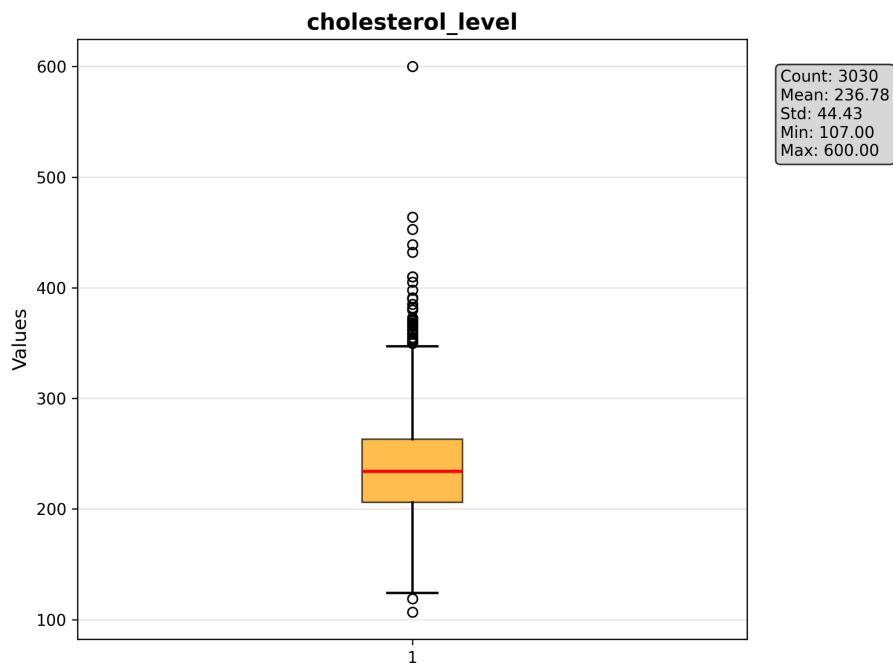
Varsta

- Distribuție: normală cu outlieri la varste înaintate (>70 ani)
- Semnificație: varsta medie de 51 ani = grupa de risc crescut pentru boli cardiovasculare



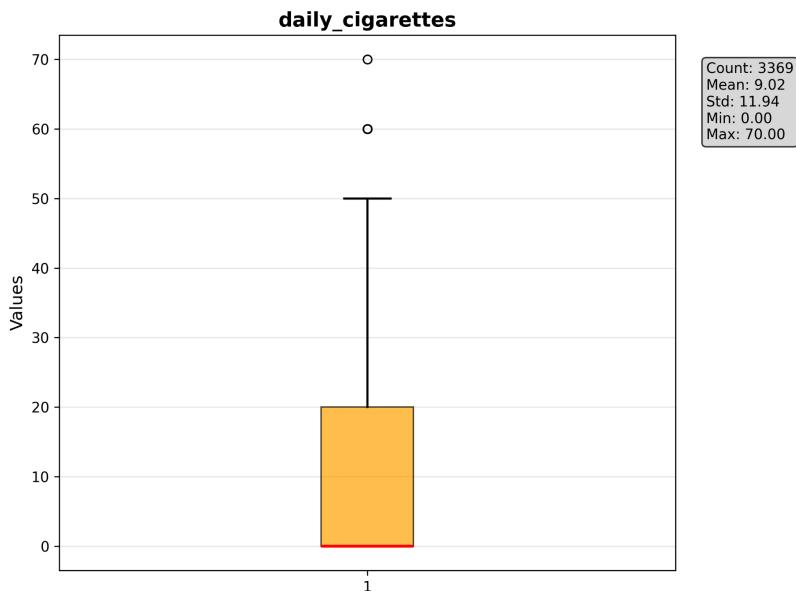
Glicemia

- Distribuție: normală cu outlieri la valori mari (>200)
- Semnificație clinică: media în limite normale



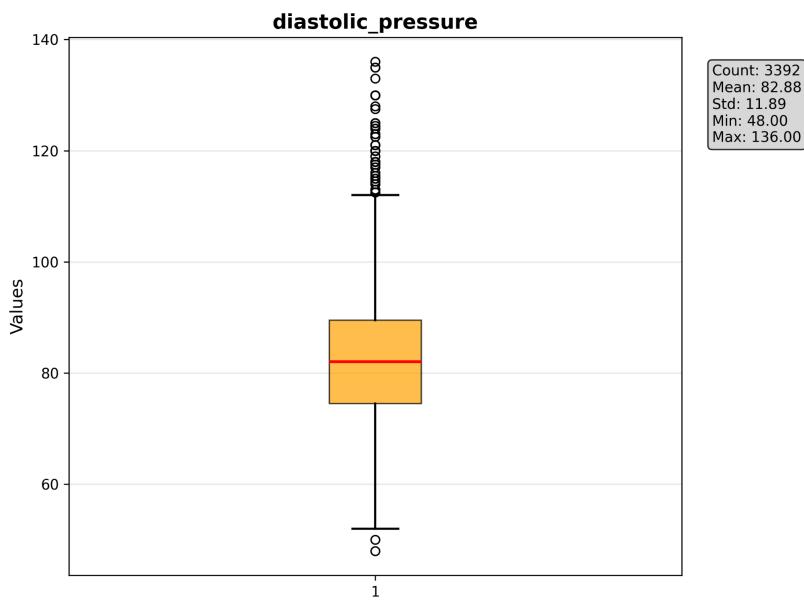
Colesterolul

- Distribuție: ușor deformata cu mulți outlieri



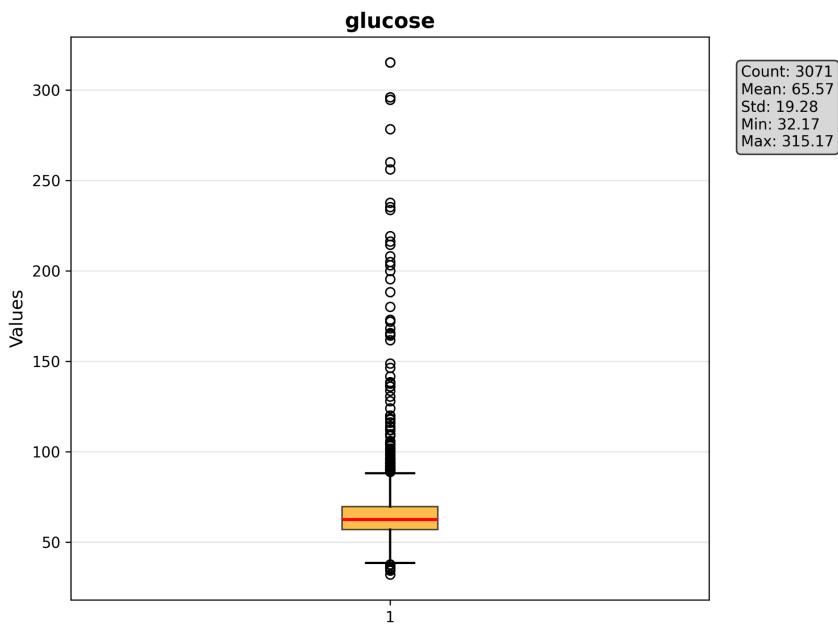
Țigări zilnice

- Distribuție: extrem de deformata
- Observație: mulți nefumători, dar fumătorii consumă până la 70 țigări/zi



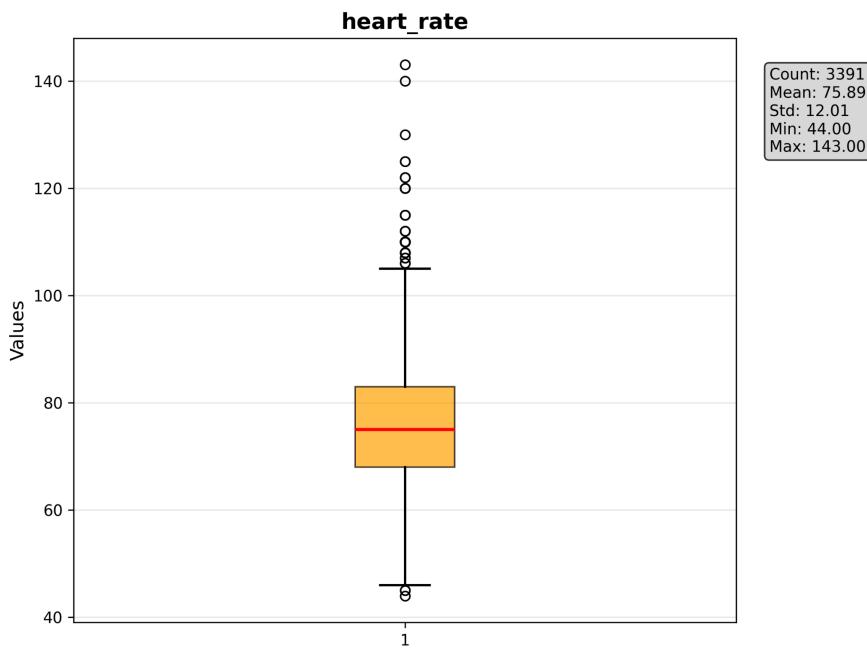
Tensiunea diastolică

- Distribuție: aproape normală cu câțiva outlieri
- Semnificație: media ușor ridicată



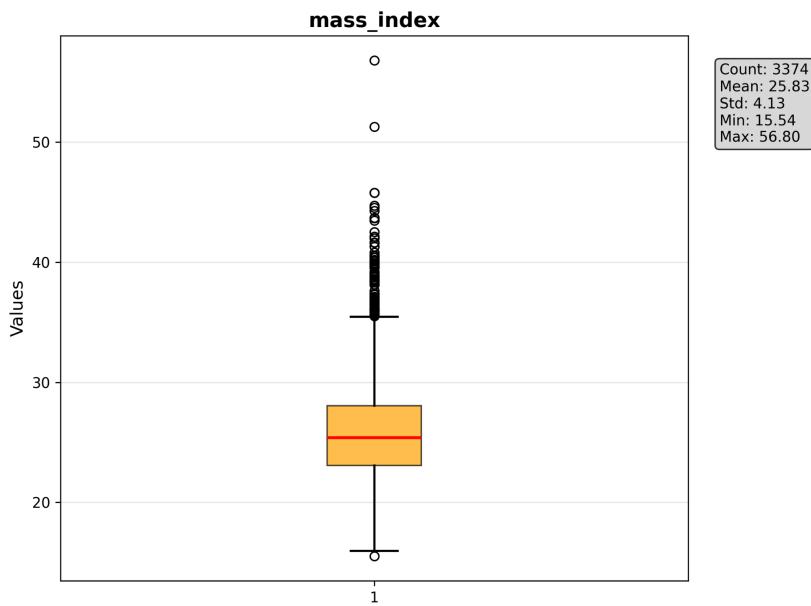
Glucoza

- Distribuție: mulți outlieri la valori mari
- Interpretare: valorile mari indică tulburări metabolice



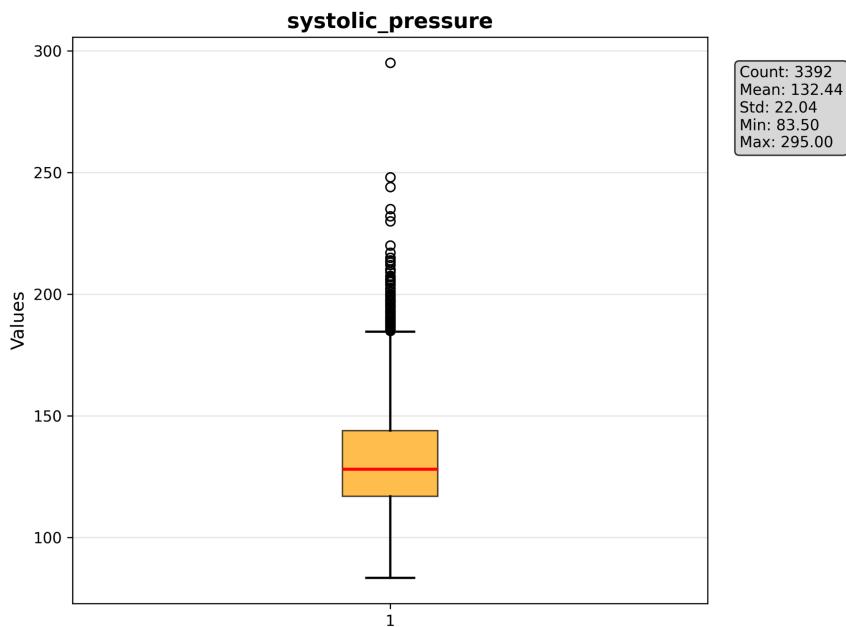
Ritmul cardiac

- Distribuție: normală cu outlieri la extreame
- Semnificație: media în limite normale



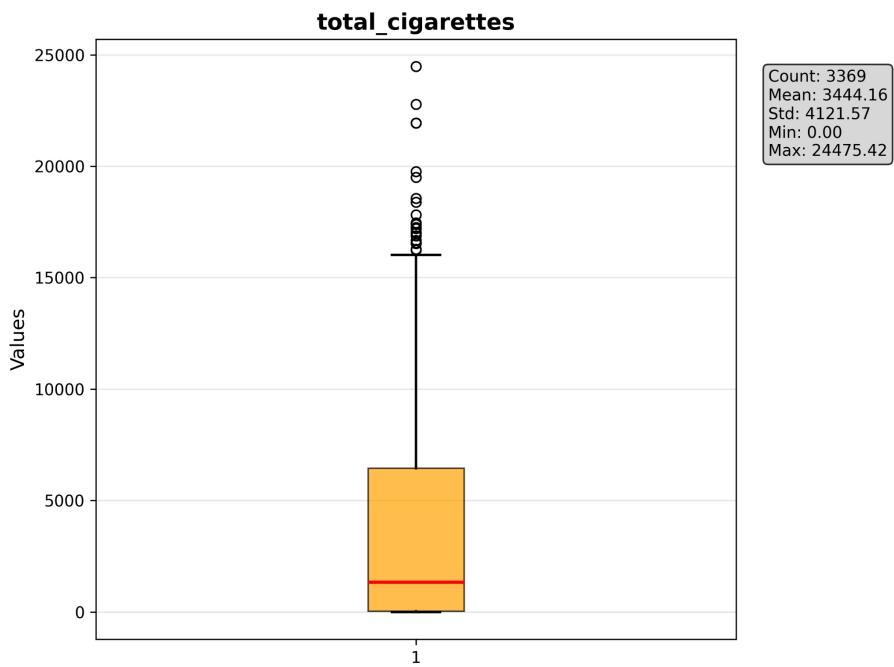
Mass index (BMI)

- Distribuție: normală cu outlieri la obezitate



Tensiunea sistolică

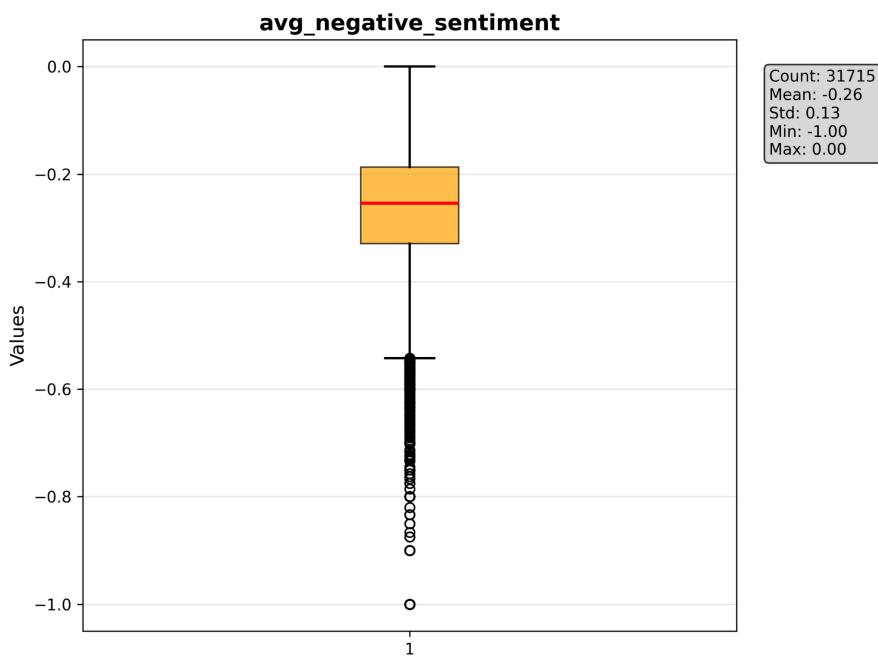
- Distribuție: mulți outlieri la valori foarte mari (>200)



Total țigări

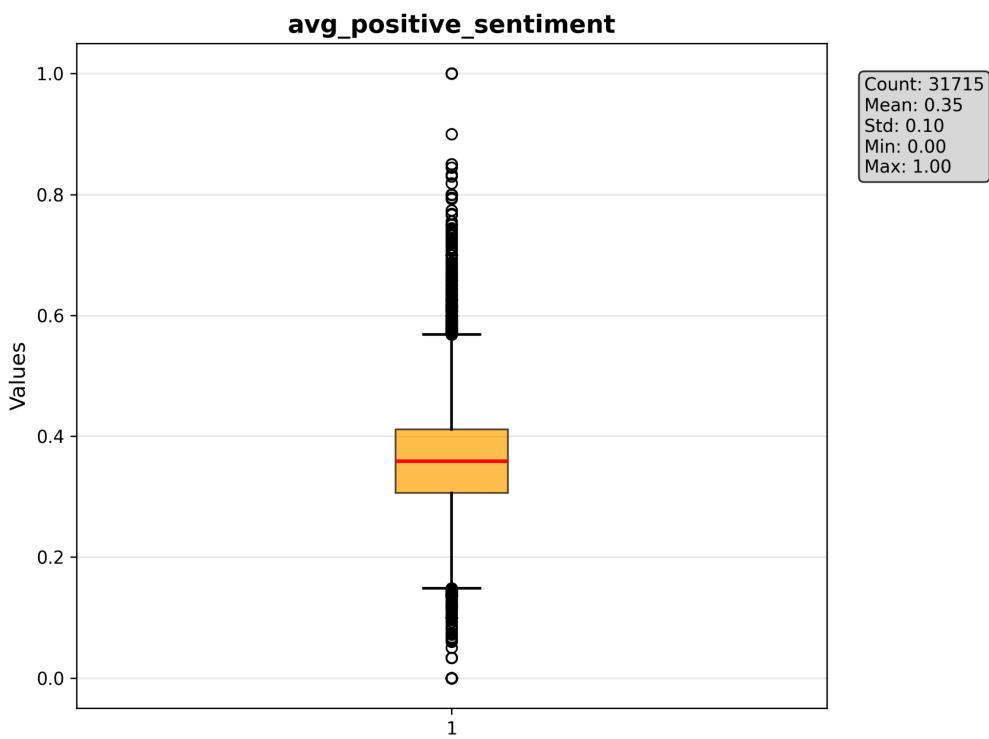
- Distribuție: extrem de deformată

Grafice pentru analiza datelor numerice referitoare la popularitatea știrilor în mediul online:



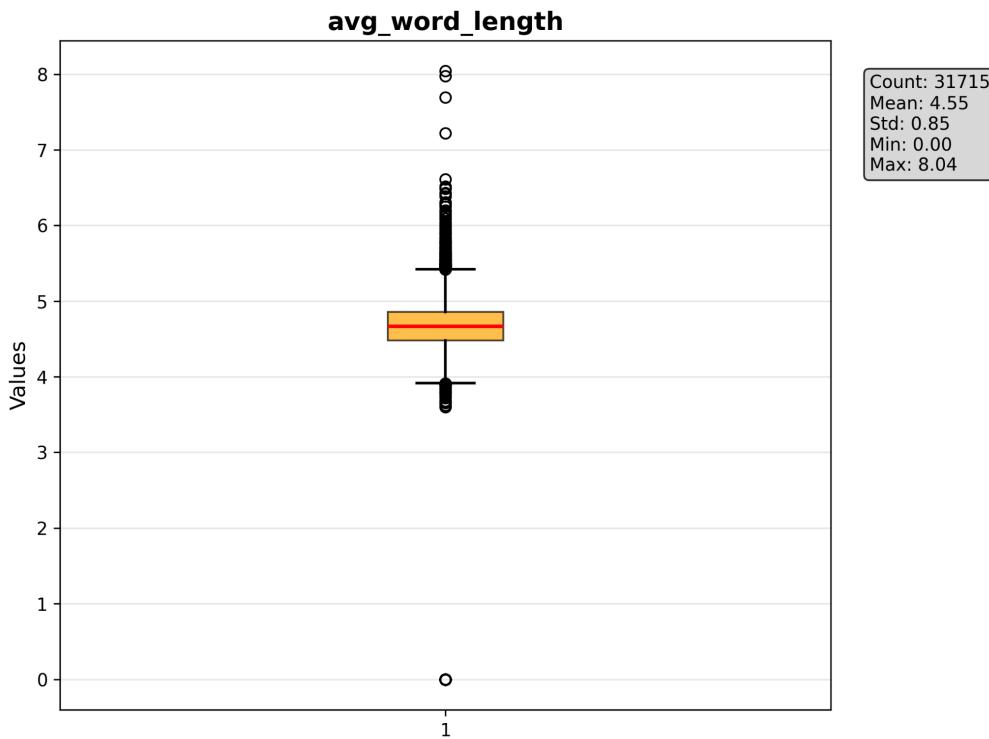
Sentimentul negativ mediu

- Distribuție: concentrată între -0.4 și 0, cu outlieri la -1.0
- Semnificație: Media de -0.26 = sentiment moderat negativ



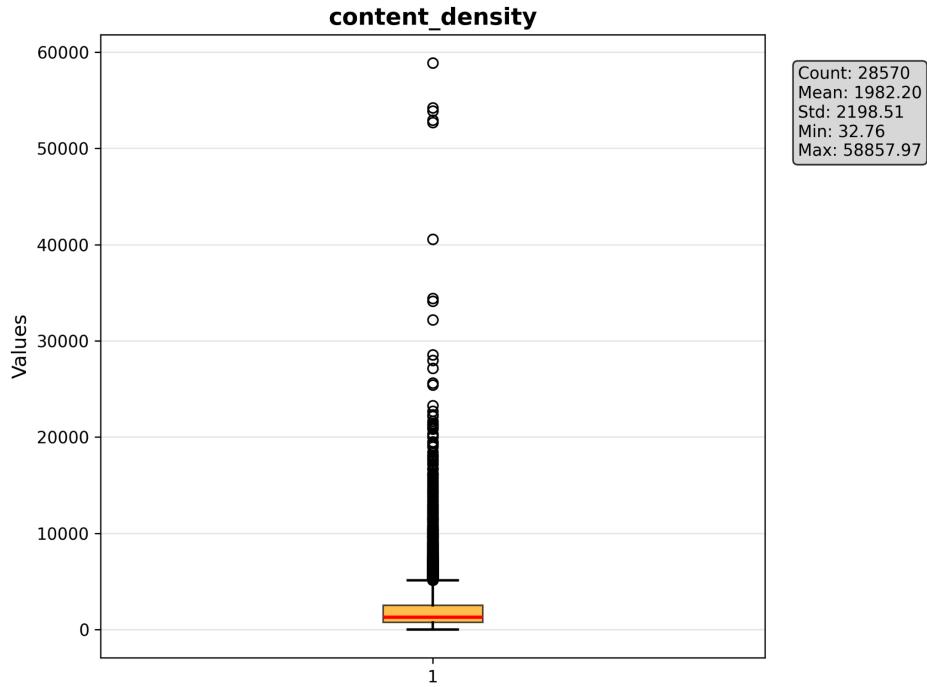
Sentimentul pozitiv mediu

- Distribuție: concentrată între 0.3-0.4, cu outlieri până la 1.0
- Semnificație: media de 0.35 = sentiment pozitiv moderat



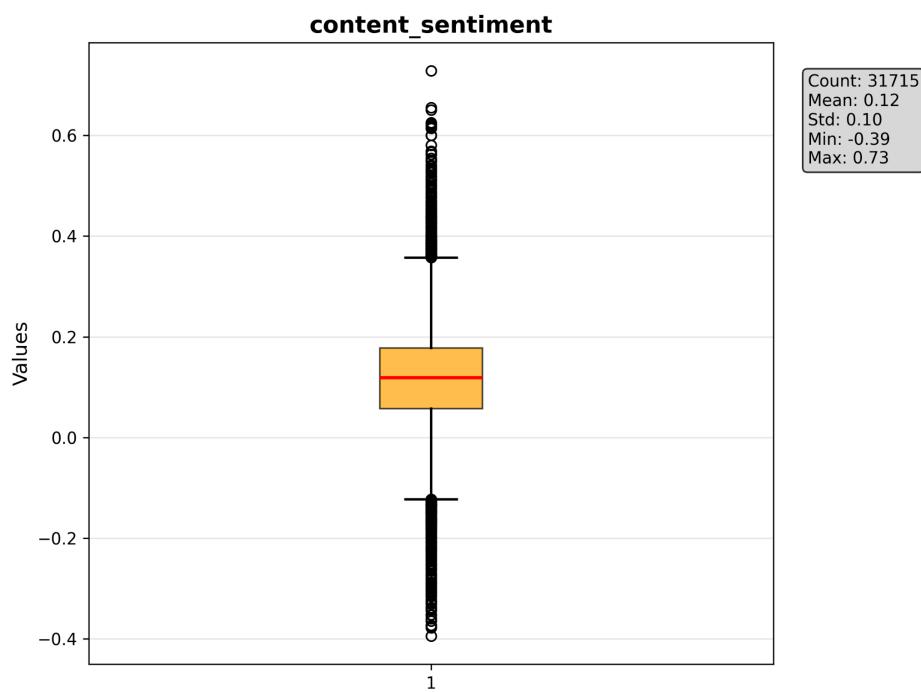
Lungimea medie a cuvintelor

- Distribuție: normală cu câțiva outlieri
- Semnificație: ~4.5 caractere/cuvânt = limbaj accesibil



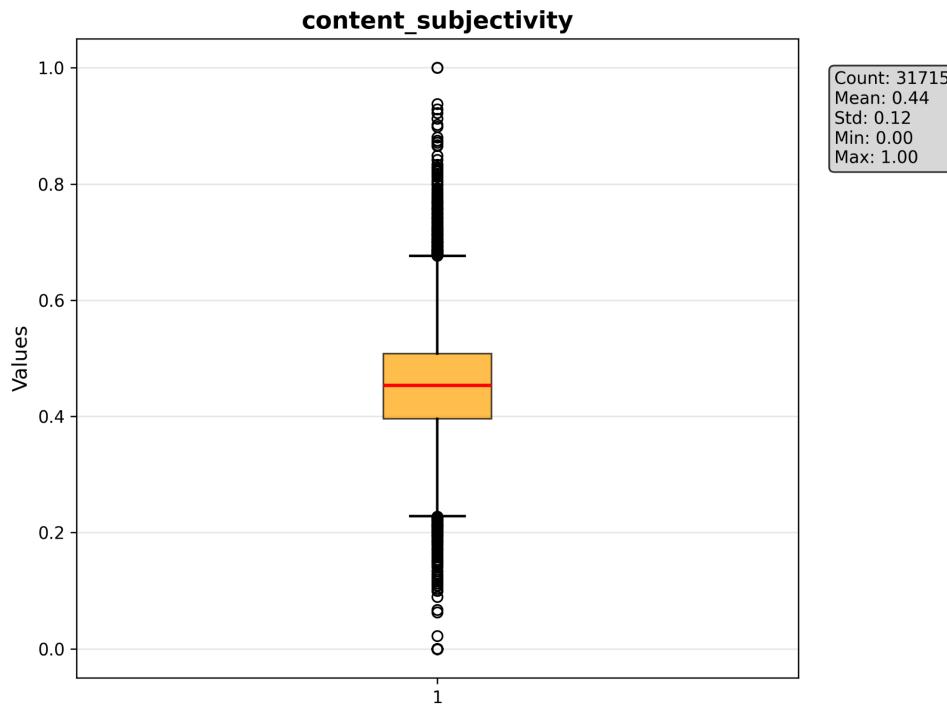
Densitatea conținutului

- Distribuție extrem de deformată cu outlieri uriași (58,857)
- Interpretare: majoritatea postărilor = conținut light, dar există mega-postări



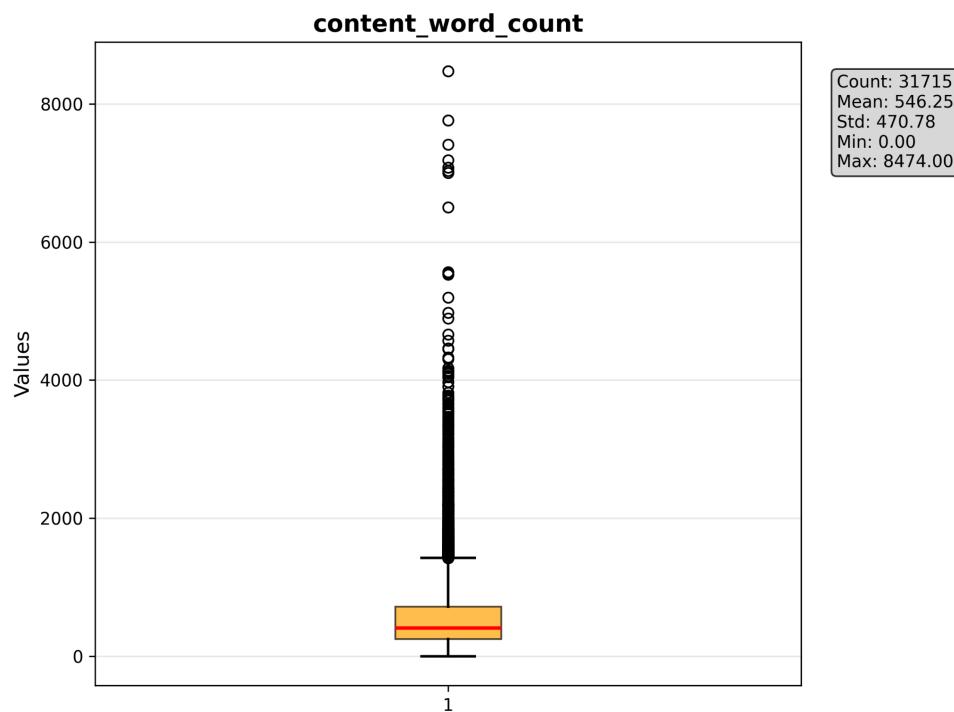
Sentimentul conținutului

- Distribuție: centrat pe 0, aproape neutru
- Semnificație: sentiment general neutru-pozitiv



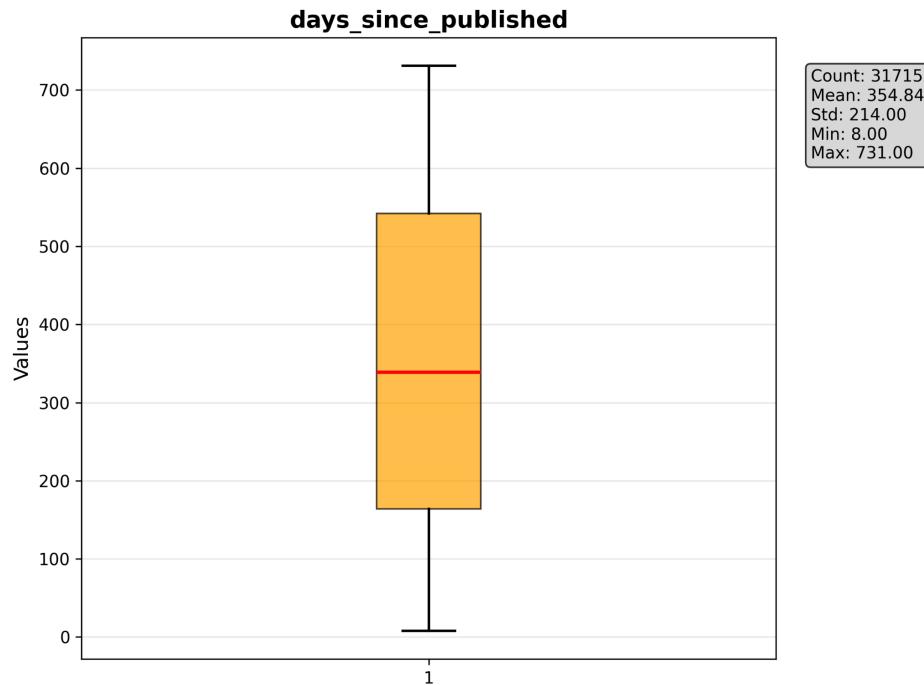
Subiectivitatea

- Distribuție: centrat pe 0.4-0.5
- Interpretare: echilibru optim între obiectiv și subiectiv



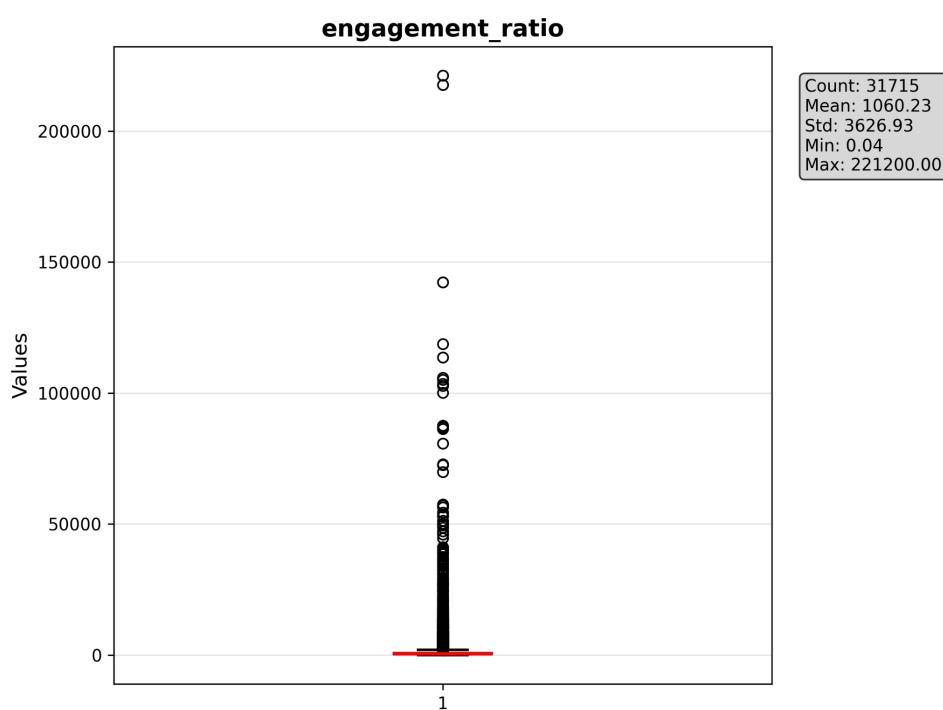
Numărul de cuvinte

- Distribuție: foarte deformata cu outlieri la 8,474 cuvinte
- Postări medii



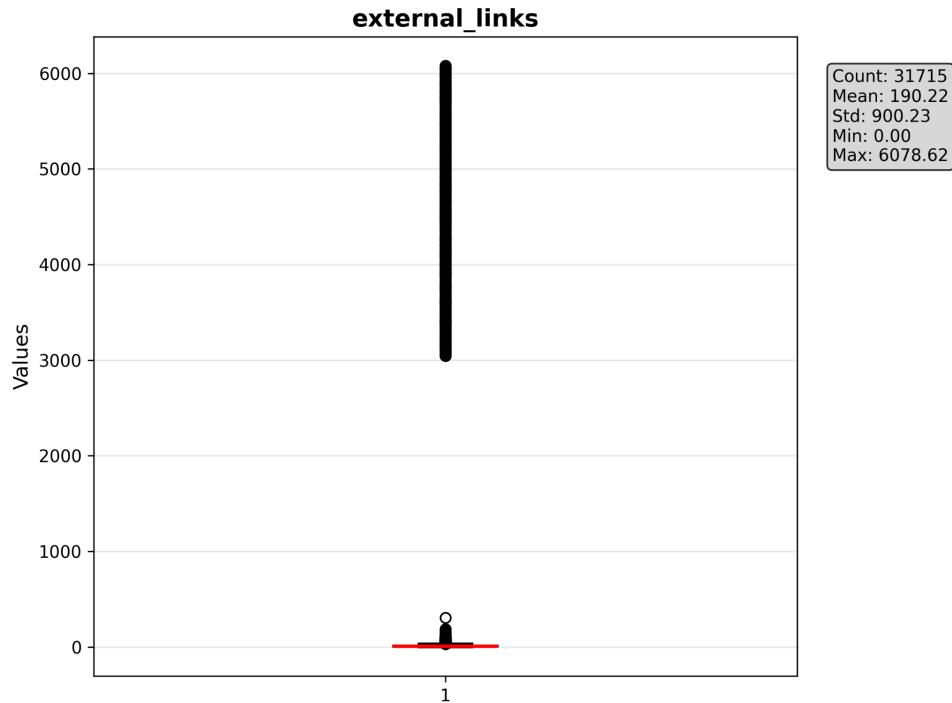
Zile de la publicare

- Distribuție: aproape normală (8-731 zile)



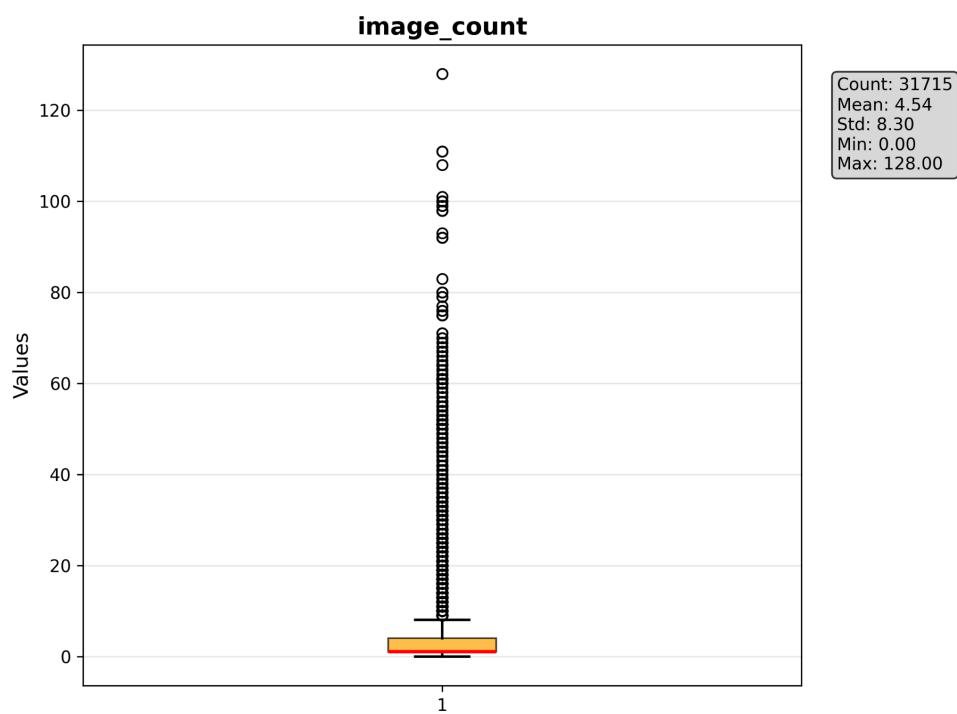
Rata de angajament

- Distribuție extrem de deformată - outlier la 221,200
- Realitate: majoritatea postărilor = engagement scăzut



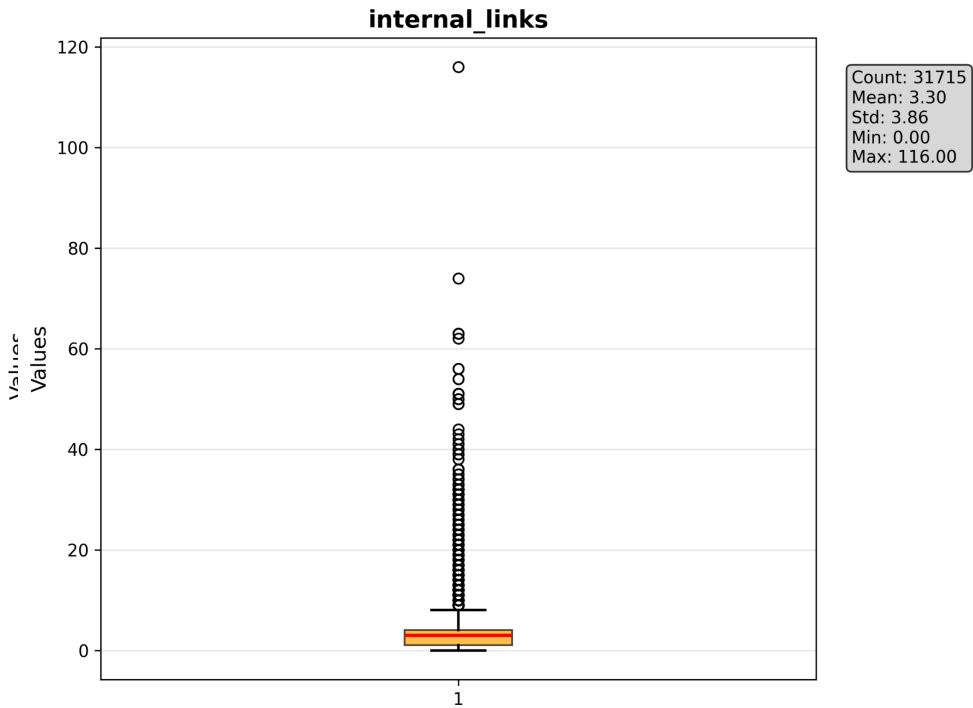
Link-uri externe:

- Interpretare: majoritatea postărilor fără link-uri externe



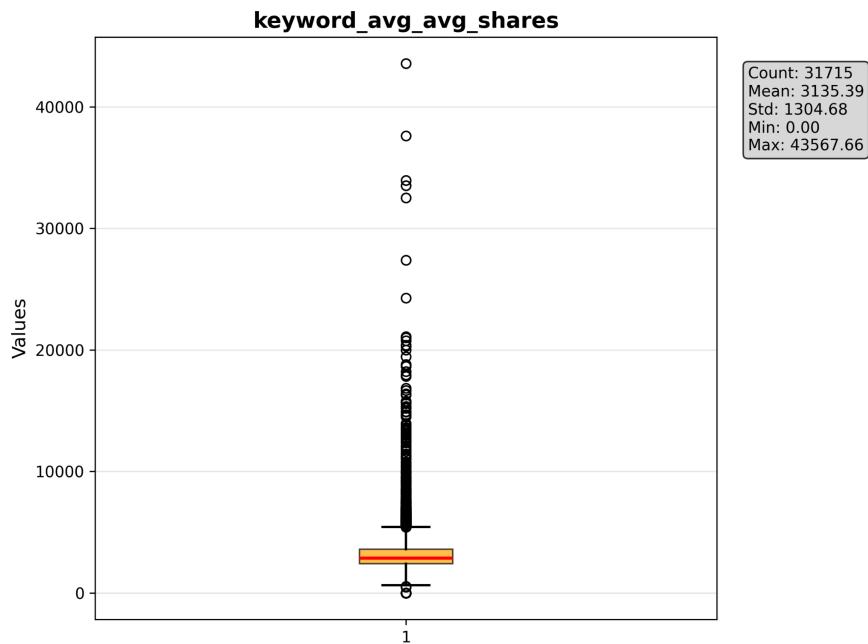
Numărul de imagini

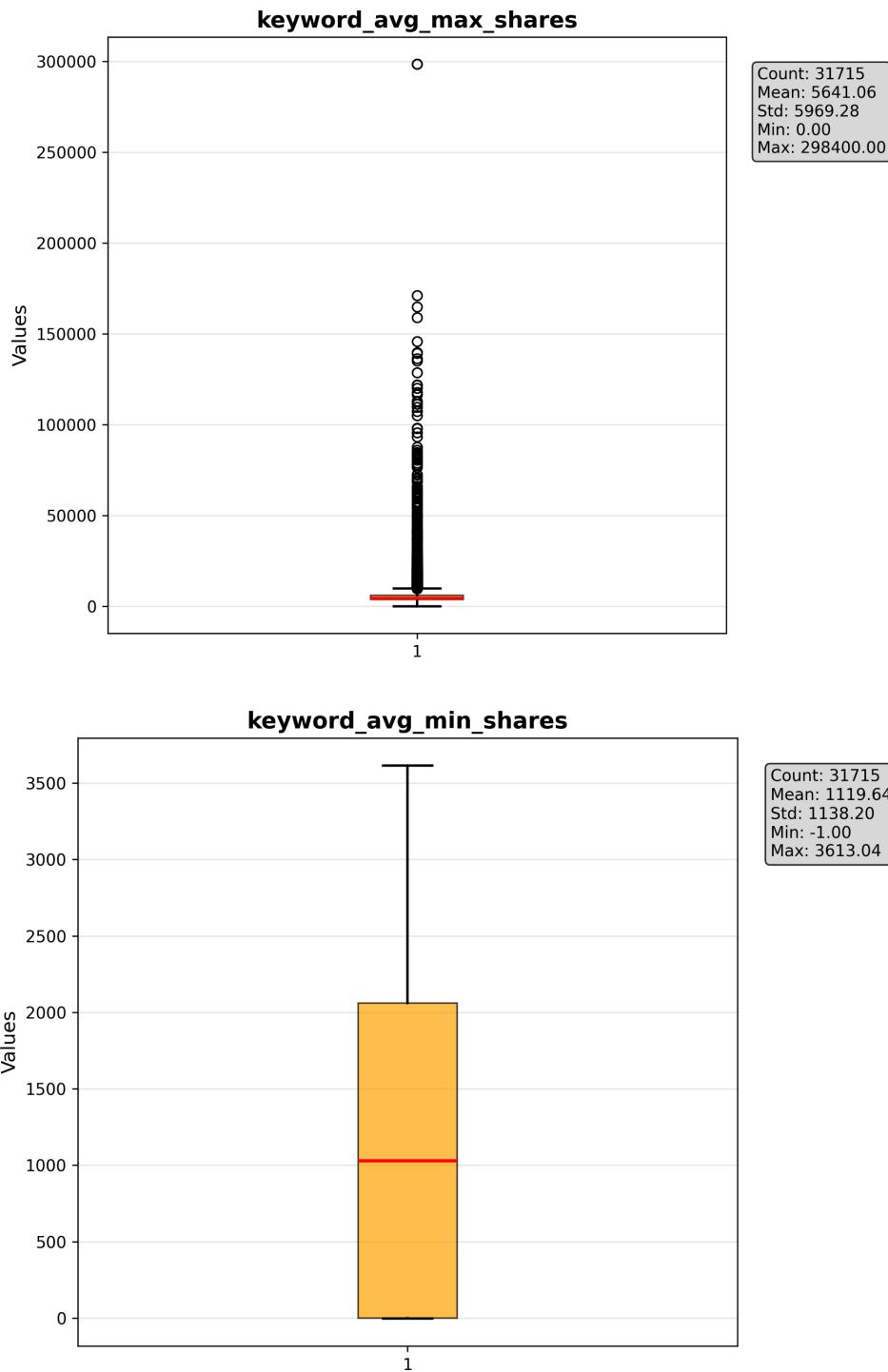
- Distribuție: foarte deformata \approx 2-3 imagini
- Insight: majoritatea postărilor = 2-3 imagini, dar există outlieri cu numar mare de imagini



Link-uri interne

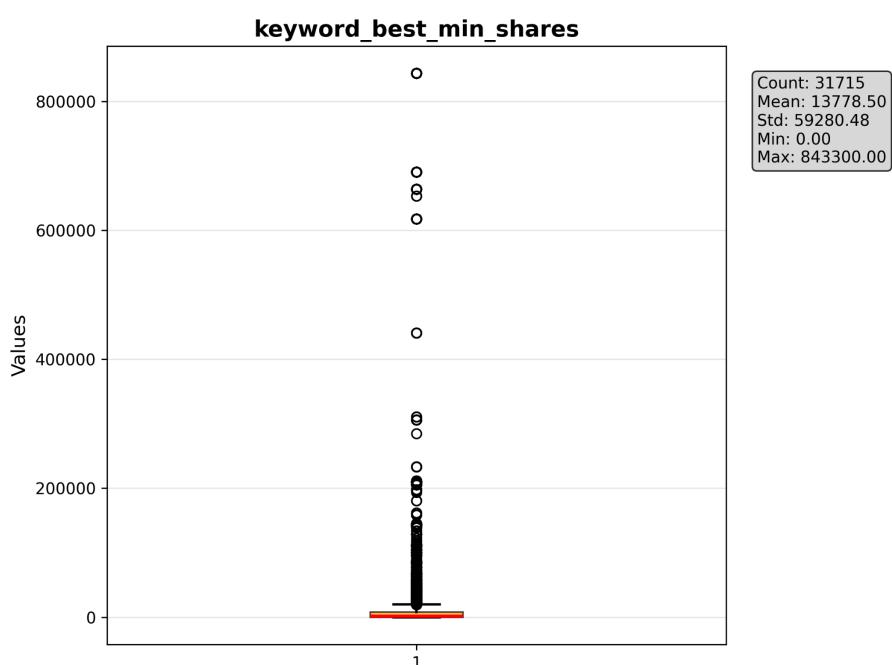
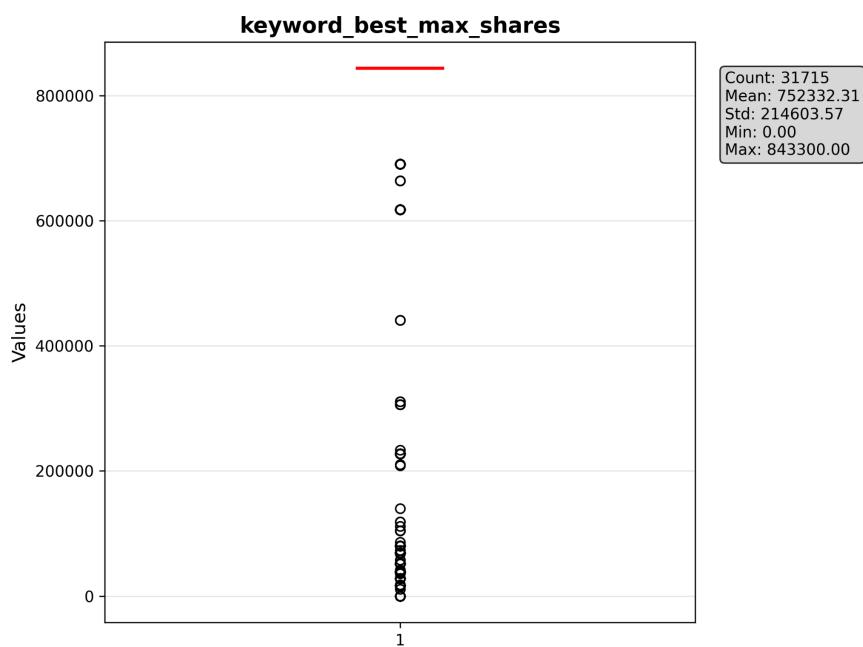
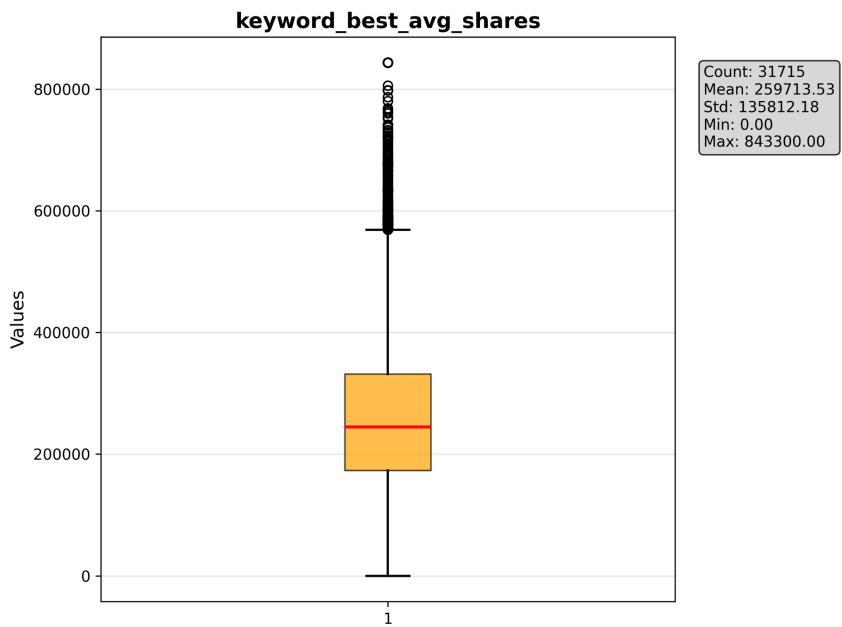
- Distribuție: similară cu image_count - deformata către low
- Comparație: diferența mare intre internal si external





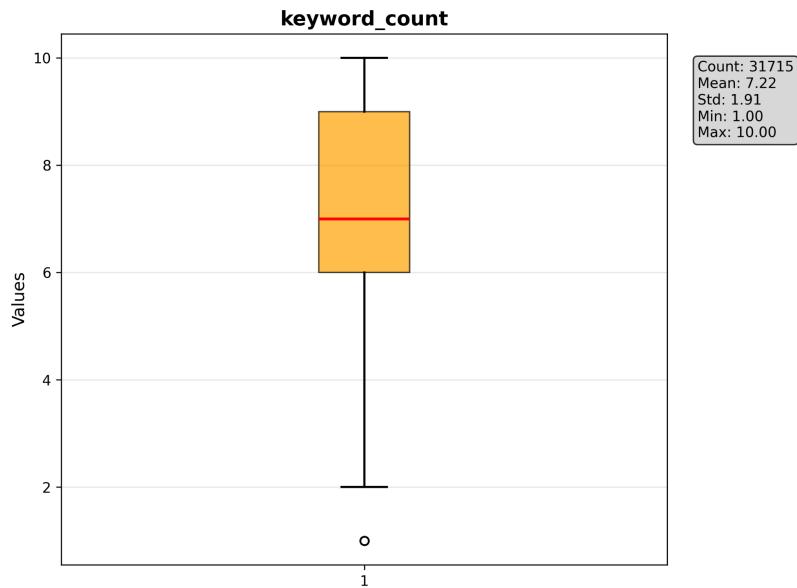
- **avg_shares:** $3,135 \pm 1,305$ (moderat)
- **max_shares:** $5,641 \pm 5,969$ (deformată)
- **min_shares:** $1,120 \pm 1,138$ (distribuție normală)

Keyword-urile medii au performanță predictibilă și consistentă



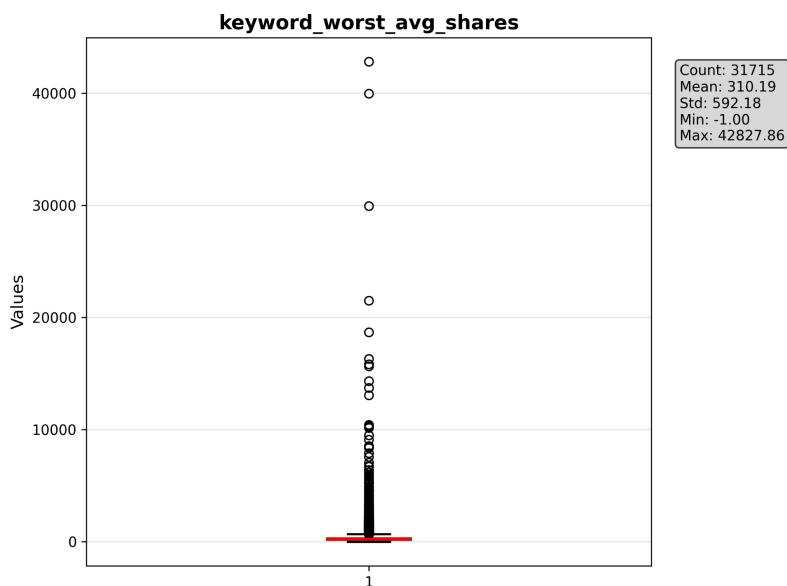
- **avg_shares:** $259,713 \pm 135,812$ (foarte mare)
- **max_shares:** $752,332 \pm 214,603$ (extrem de mare)
- **min_shares:** $13,779 \pm 59,280$

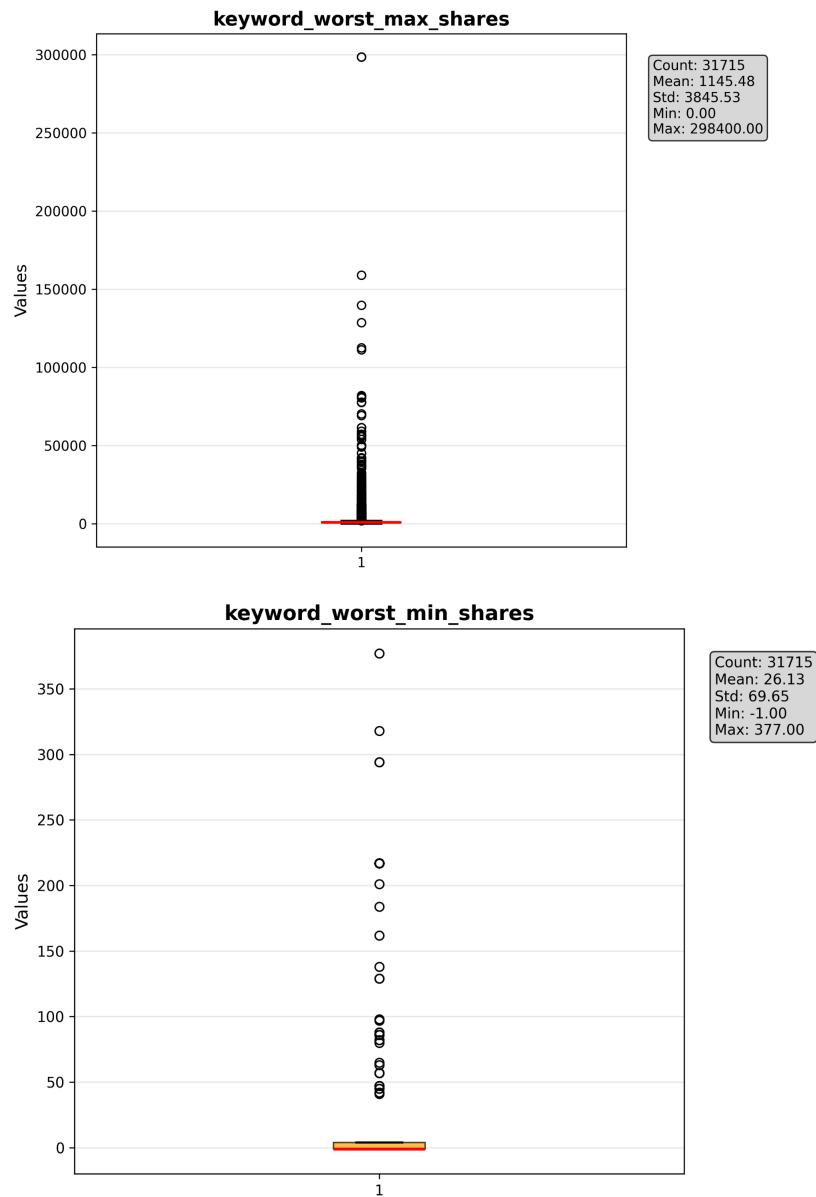
Diferență între "best" vs "avg" keywords = 83x mai multe share-uri



Numarul de keyword-uri:

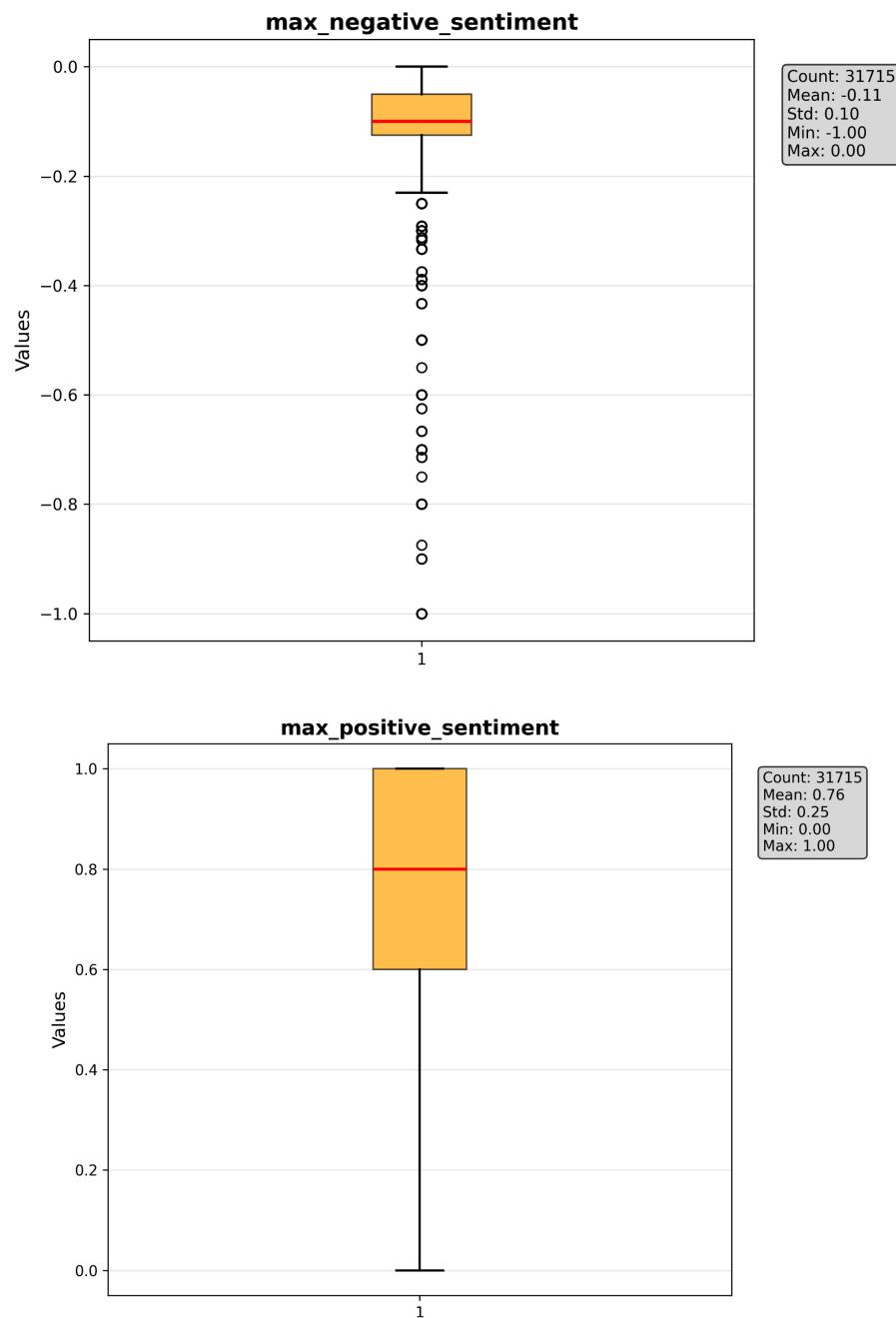
- Distribuție: normală (1-10 keywords)

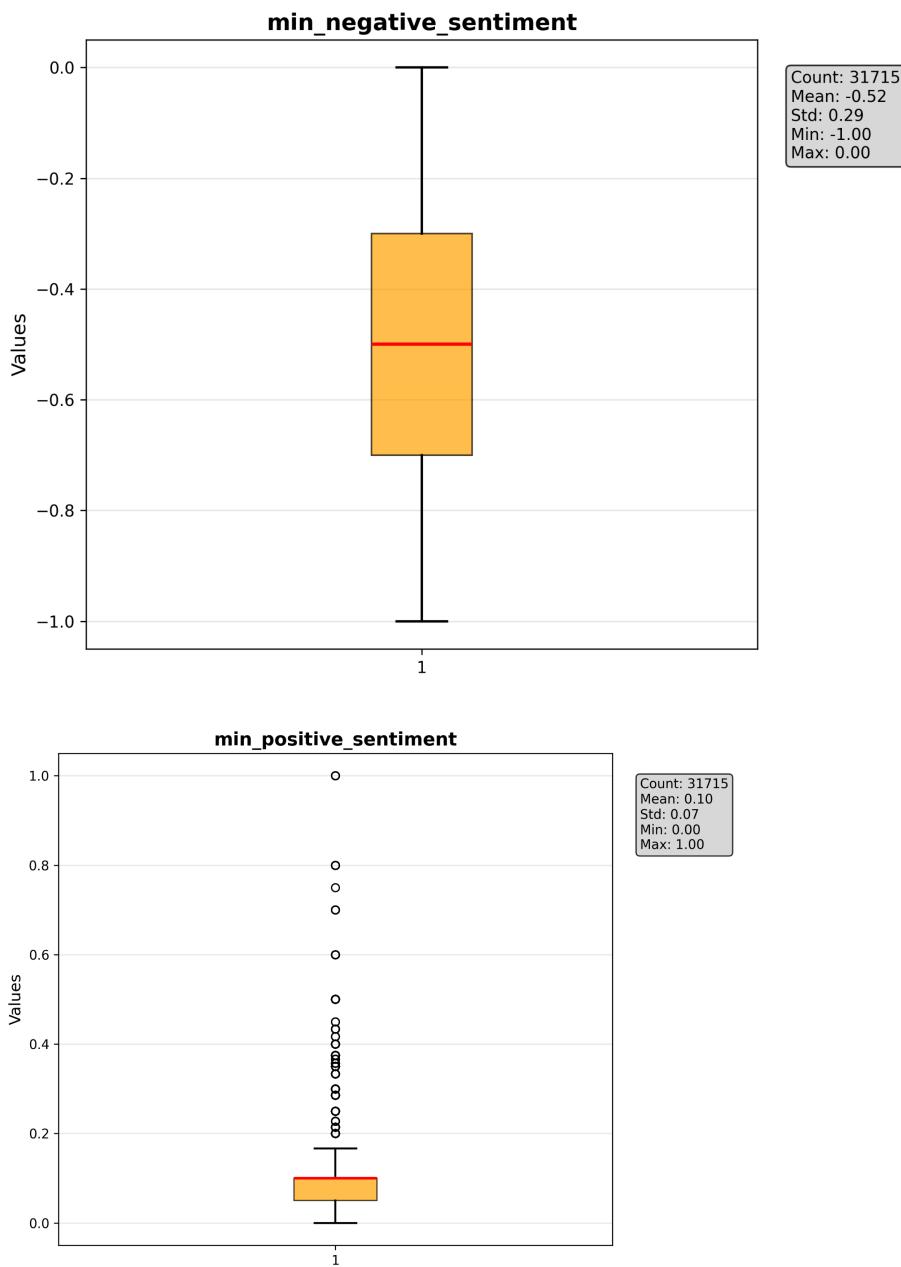




- **avg_shares:** 310.19 ± 592 (vs best: 259,713)
- **max_shares:** $1,145 \pm 3,845$ (vs best: 752,332)
- **min_shares:** 26.13 ± 69 (aproape zero engagement)

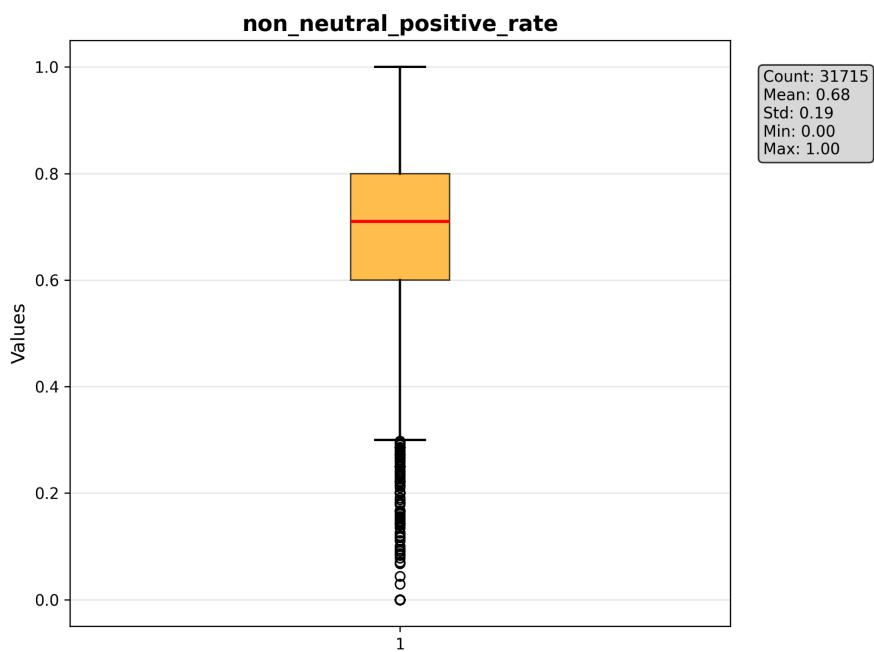
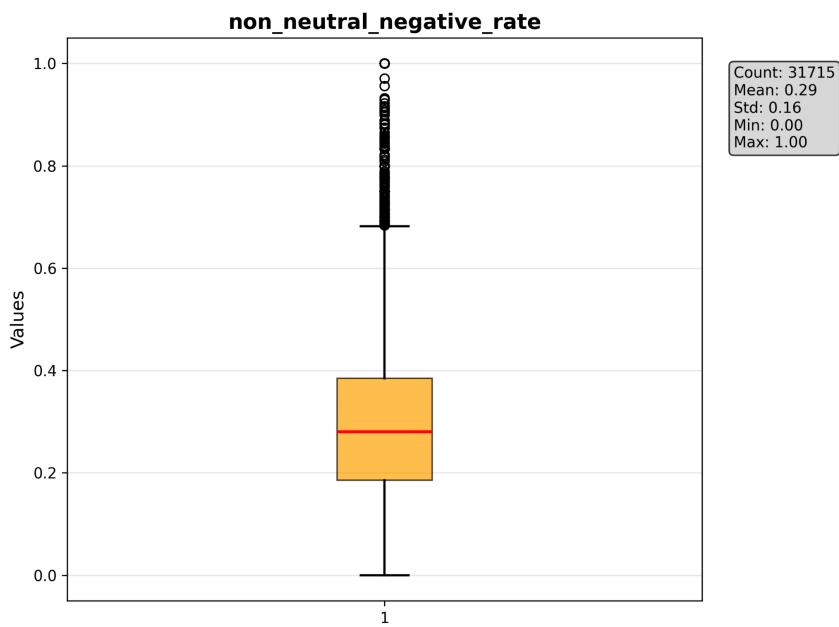
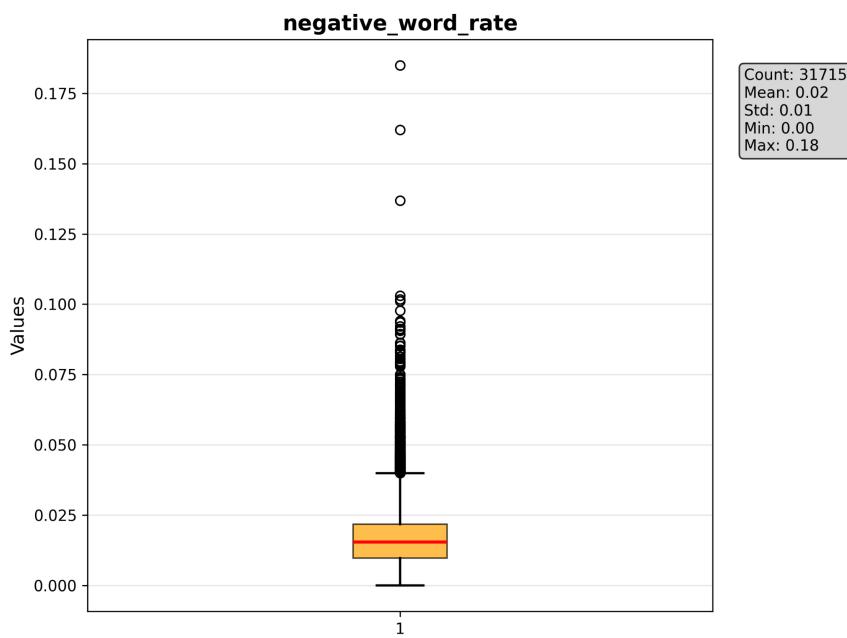
Diferență între keyword-uri best vs worst = 2,369x mai multe share-uri

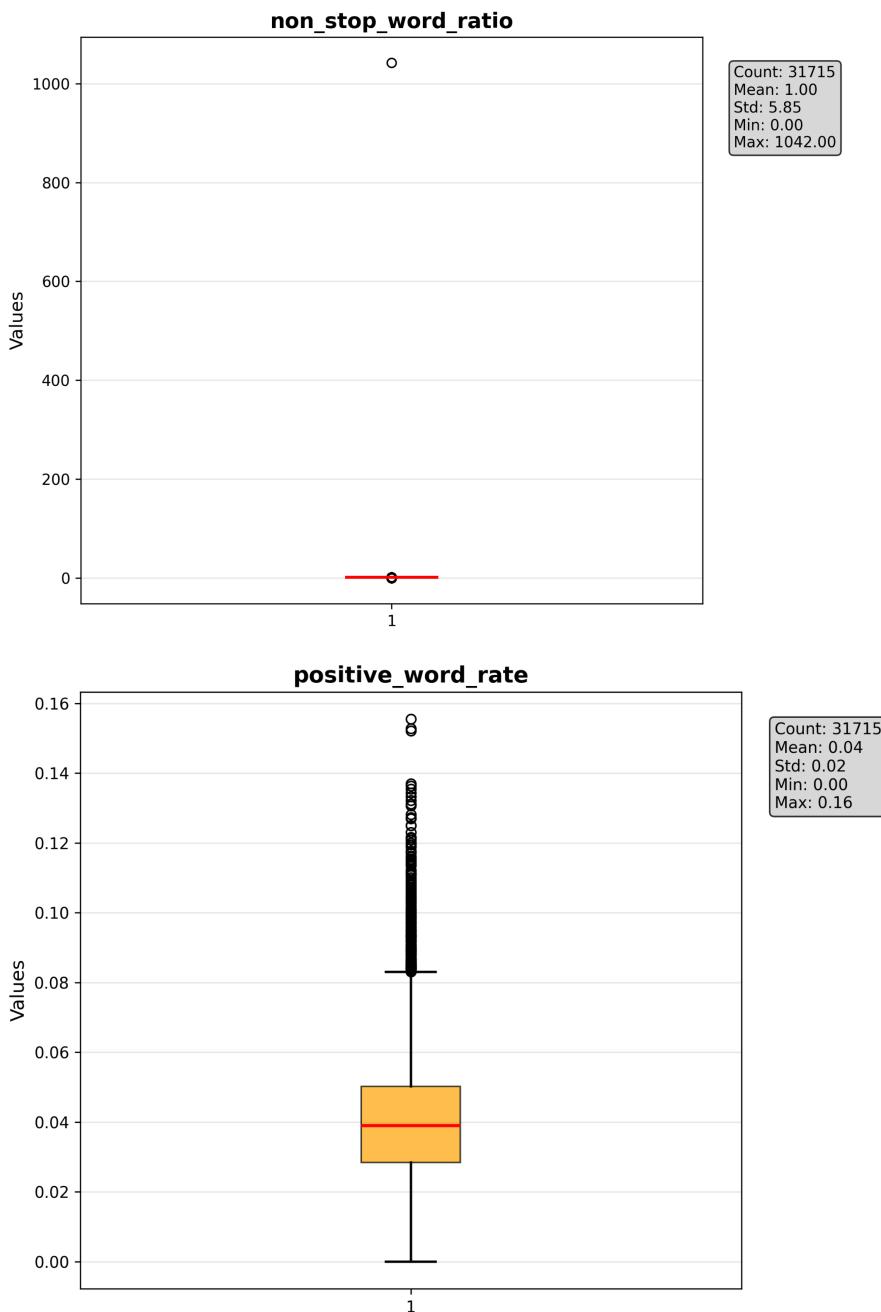




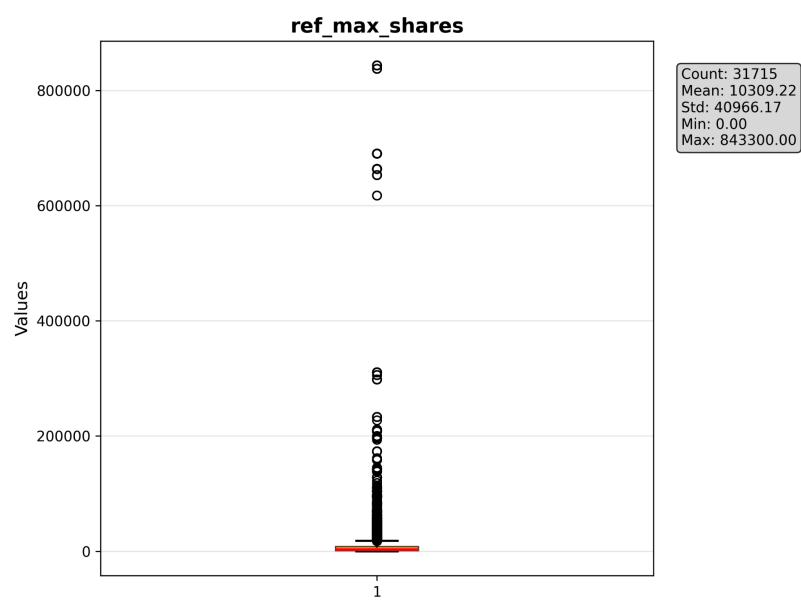
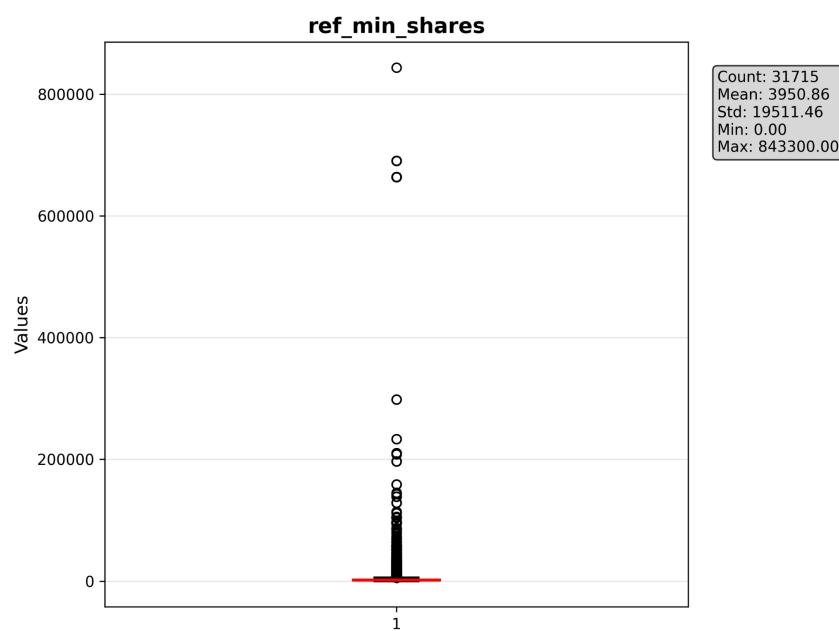
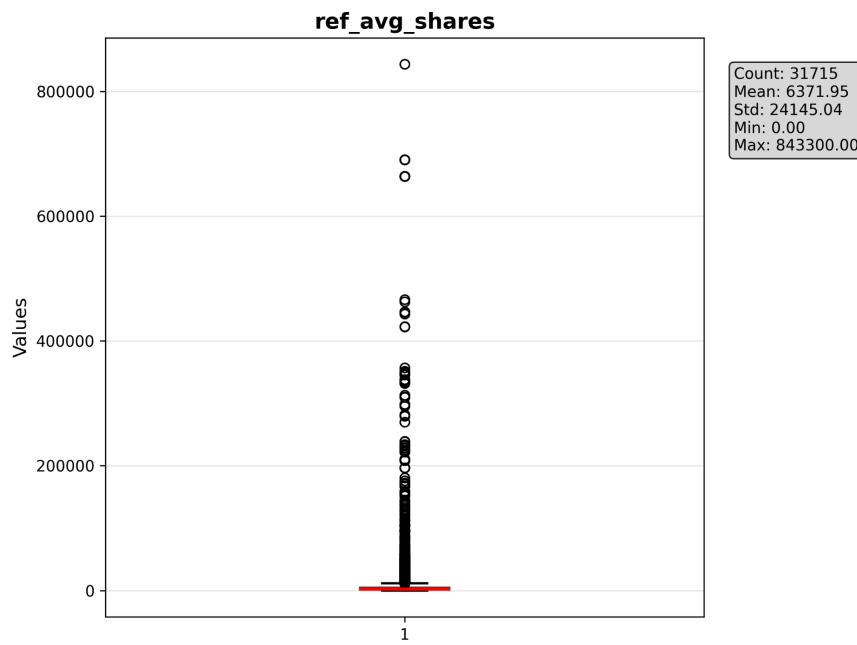
- **max_negative:** -0.11 ± 0.10 (surprinzător de slab negativ)
- **max_positive:** 0.76 ± 0.25 (puternic pozitiv în majoritatea postărilor)
- **min_negative:** -0.52 ± 0.29 (distribuție largă)
- **min_positive:** 0.10 ± 0.07 (consistent pozitiv minim)

Social media content = în general pozitiv (0.76 avg pozitiv vs -0.11 avg negativ)

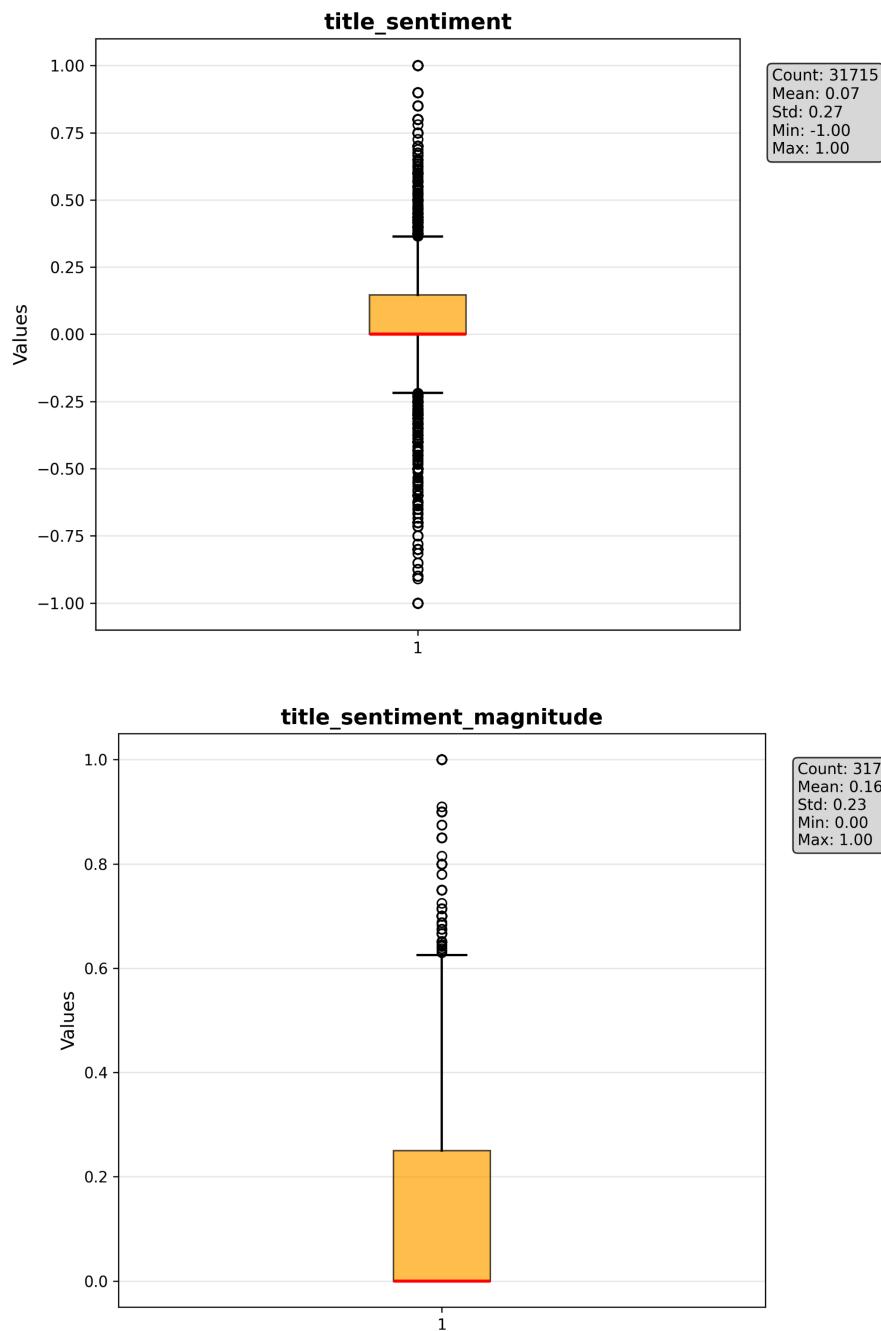




- **negative_word_rate:** 0.02 ± 0.01 (doar 2% cuvinte negative)
- **positive_word_rate:** 0.04 ± 0.02 (4% cuvinte pozitive - dublu)
- **non_neutral_negative:** 0.29 ± 0.16 (29% conținut negativ non-neutrul)
- **non_neutral_positive:** 0.68 ± 0.19 (68% conținut pozitiv - dominant)
- **non_stop_word_ratio:** 1.00 ± 5.85 (outlieri extremi)

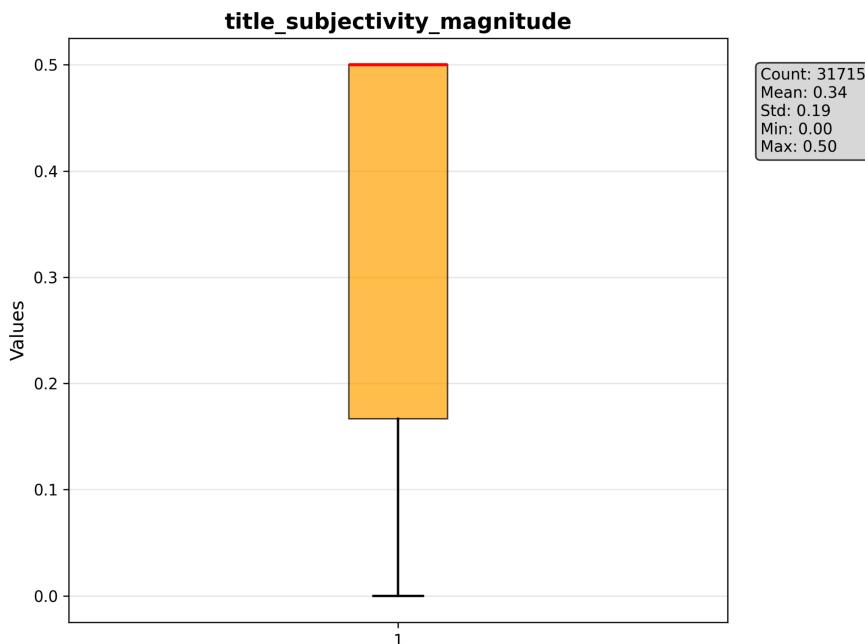
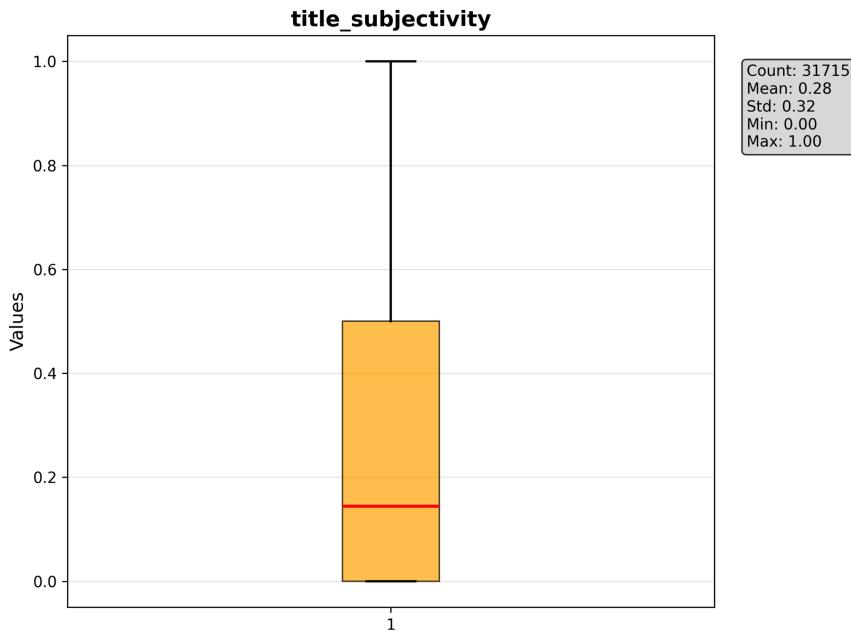


- **ref_avg_shares:** majoritatea articolelor rămân nevăzute
- **ref_max_shares:** anumite referințe din articole pot performa mai bine decât articolul în ansamblu
- **ref_min_shares:** indică că există totuși un nivel de bază de angajament pentru majoritatea conținutului

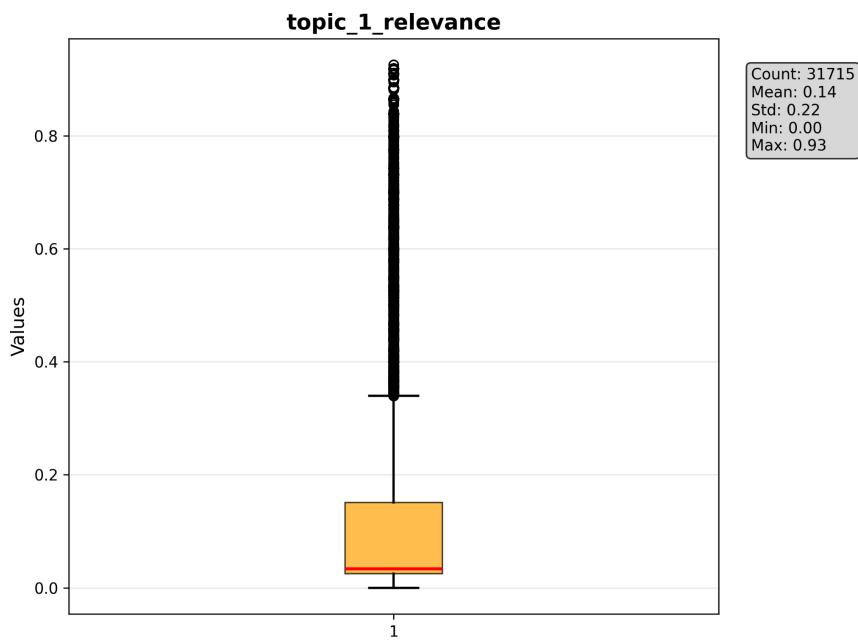
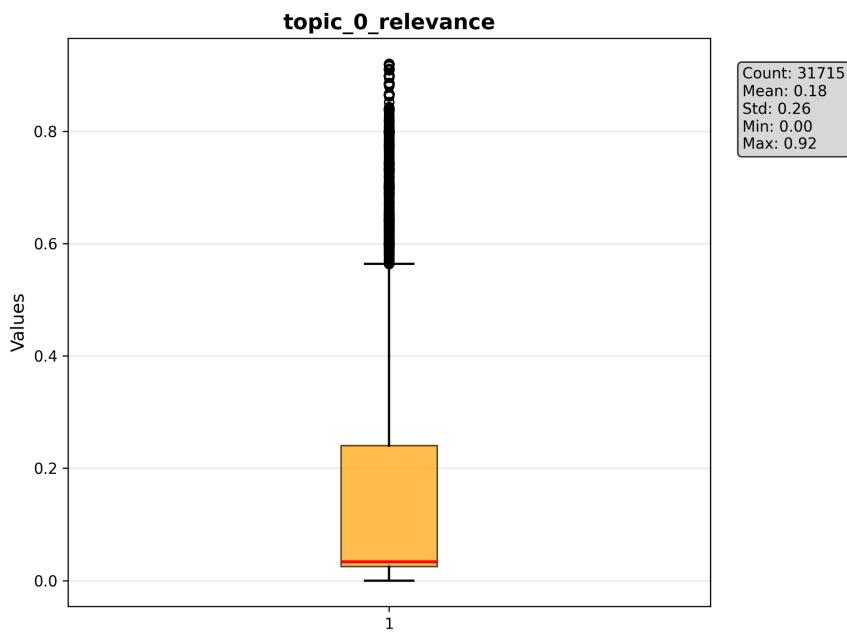
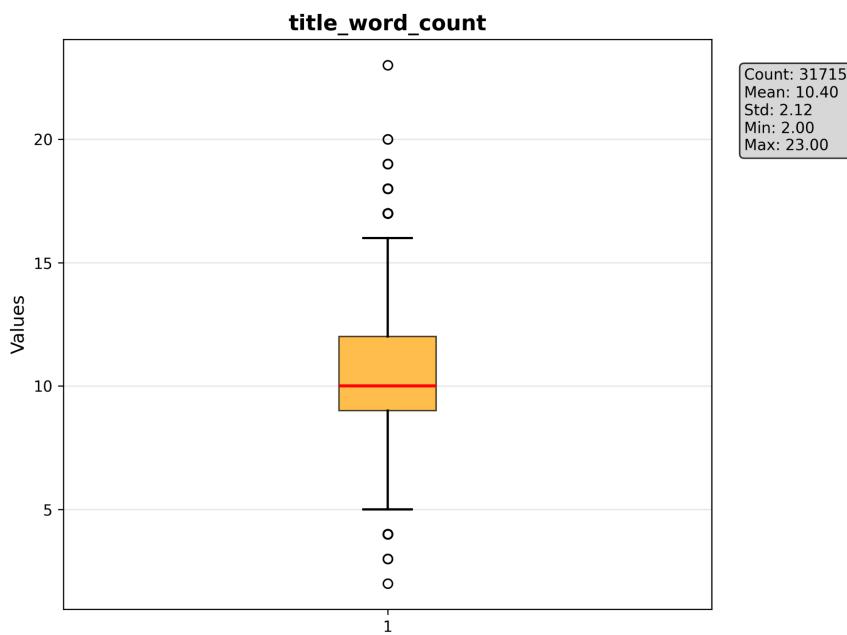


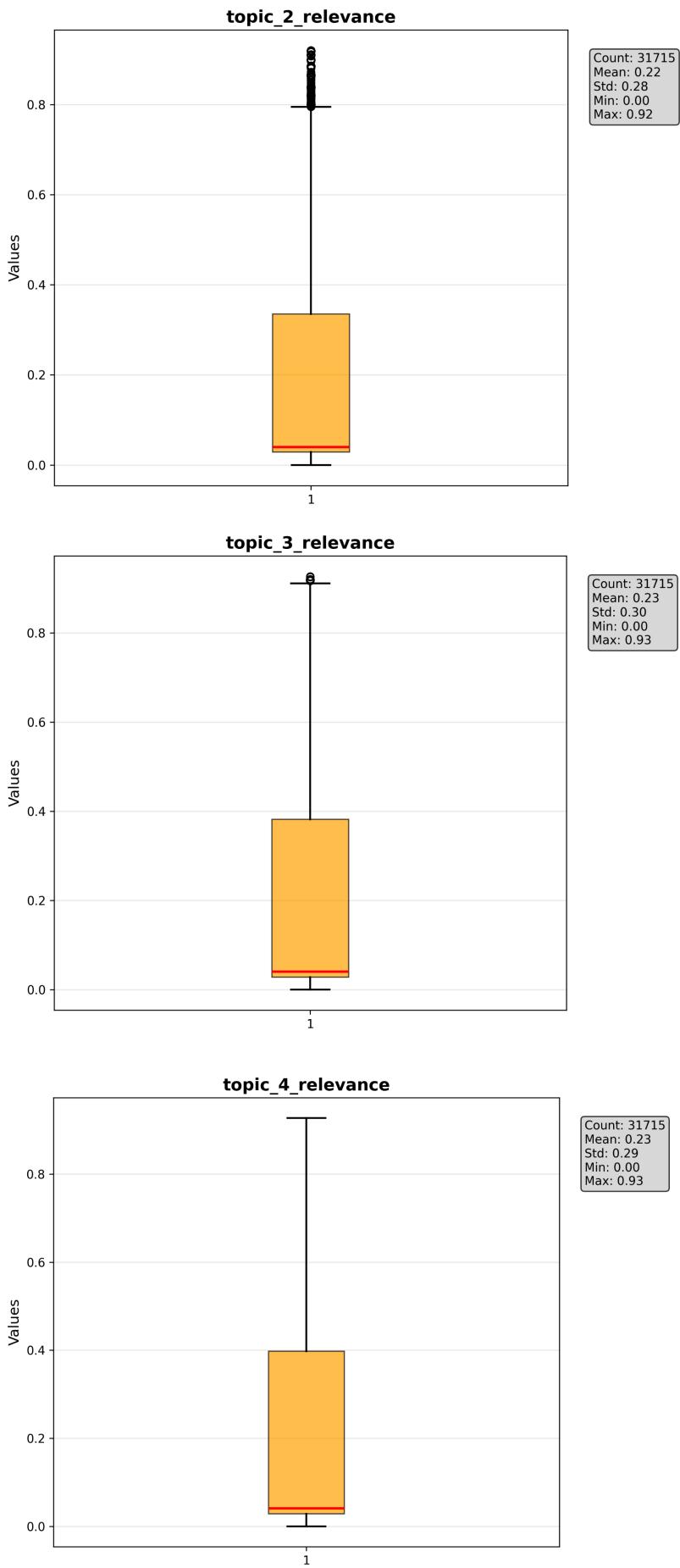
- **Ușoară tendință pozitivă în general** - media sentimentului este 0.07, indicând că titlurile tend să fie ușor pozitive în loc de negative. Totuși, distribuția este destul de echilibrată în jurul neutralității.

- **Intensitate emoțională scăzută** - magnitudinea sentimentului are media de doar 0.16, ceea ce înseamnă că majoritatea titlurilor sunt relativ neutre din punct de vedere emoțional, mai degrabă decât puternic pozitive sau negative.

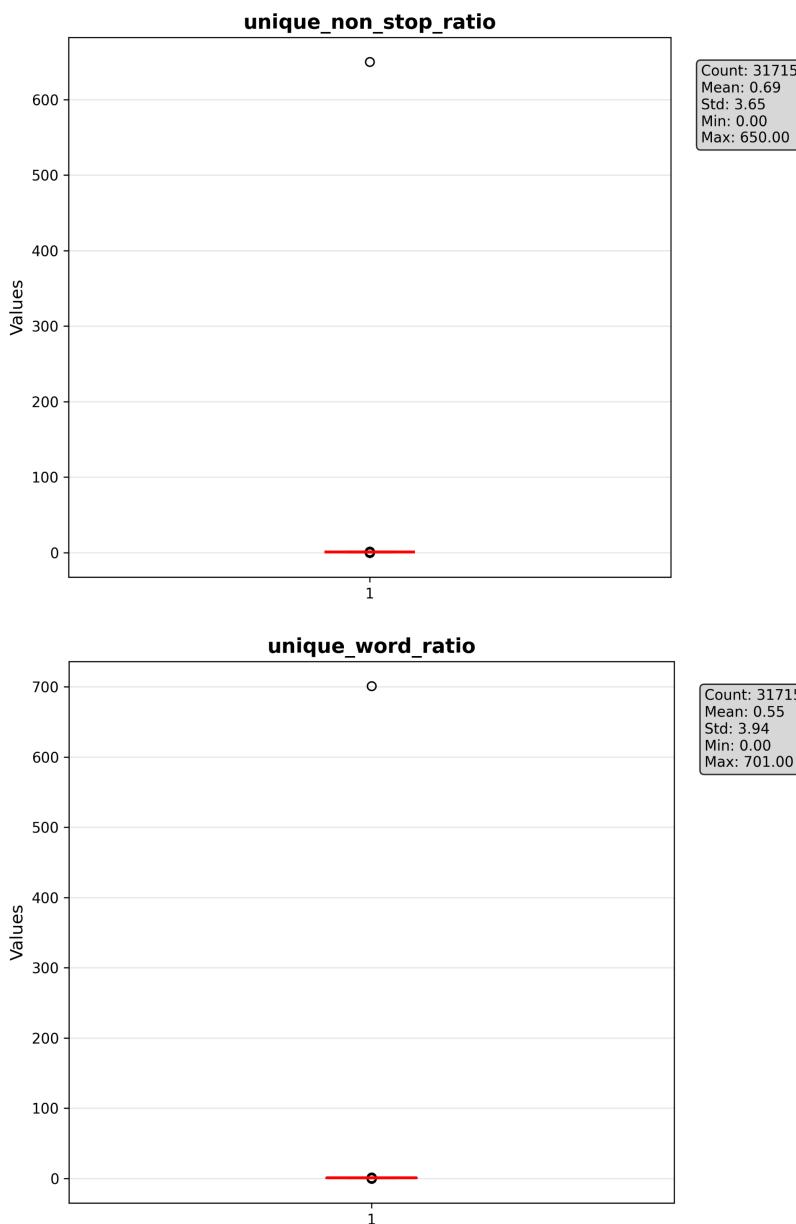


- **Predominant obiective** - cu o medie a subiectivității de 0.28, majoritatea titlurilor prezintă informațiile într-un mod factual, mai degrabă decât bazat pe opinii.
- **Obiectivitate consistentă** - metrica magnitudinii subiectivității (media 0.34) arată o grupare relativ strânsă, indicând standarde editoriale consistente în întreg setul de date.





- **Analiza lungimii titlurilor:** standarde profesionale consistente
- **Topicuri secundare (0 și 1):** topic 0 - media 0.18, concentrare puternică spre zero, sugerând că este un topic specializat care se aplică doar unei părți mici / topic 1 - cea mai mică medie (0.14), indicând cel mai puțin relevant topic din cele 5, posibil un subiect foarte specific
- **Topicuri dominante (2, 3, 4):** relevanță echilibrată - toate trei au medii similare (0.22-0.23), sugerând că acestea sunt categoriile principale de conținut, probabil acoperind subiecte majore de știri

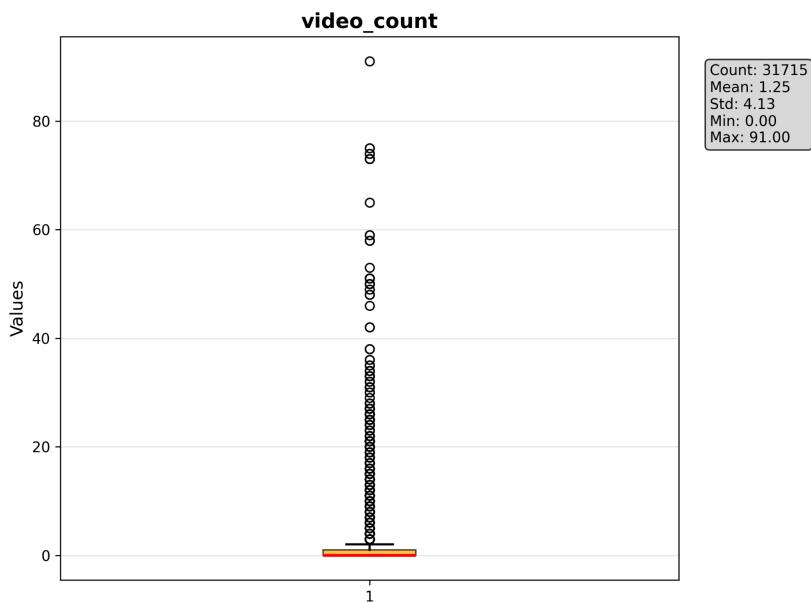


Rata cuvinte non-stop unice: majoritatea articolelor sunt convenționale

- **Outlieri excepționali** - valorile extreme până la 650 sugerează existența unor articole cu terminologie foarte specializată sau tehnică, posibil articole științifice.

Rata cuvinte unice totale

- Pattern similar, dar mai restrâns - media de 0.55 indică că atunci când se includ și cuvintele comune, unicitatea scade, ceea ce confirmă că majoritatea conținutului folosește structuri lingvistice standard.

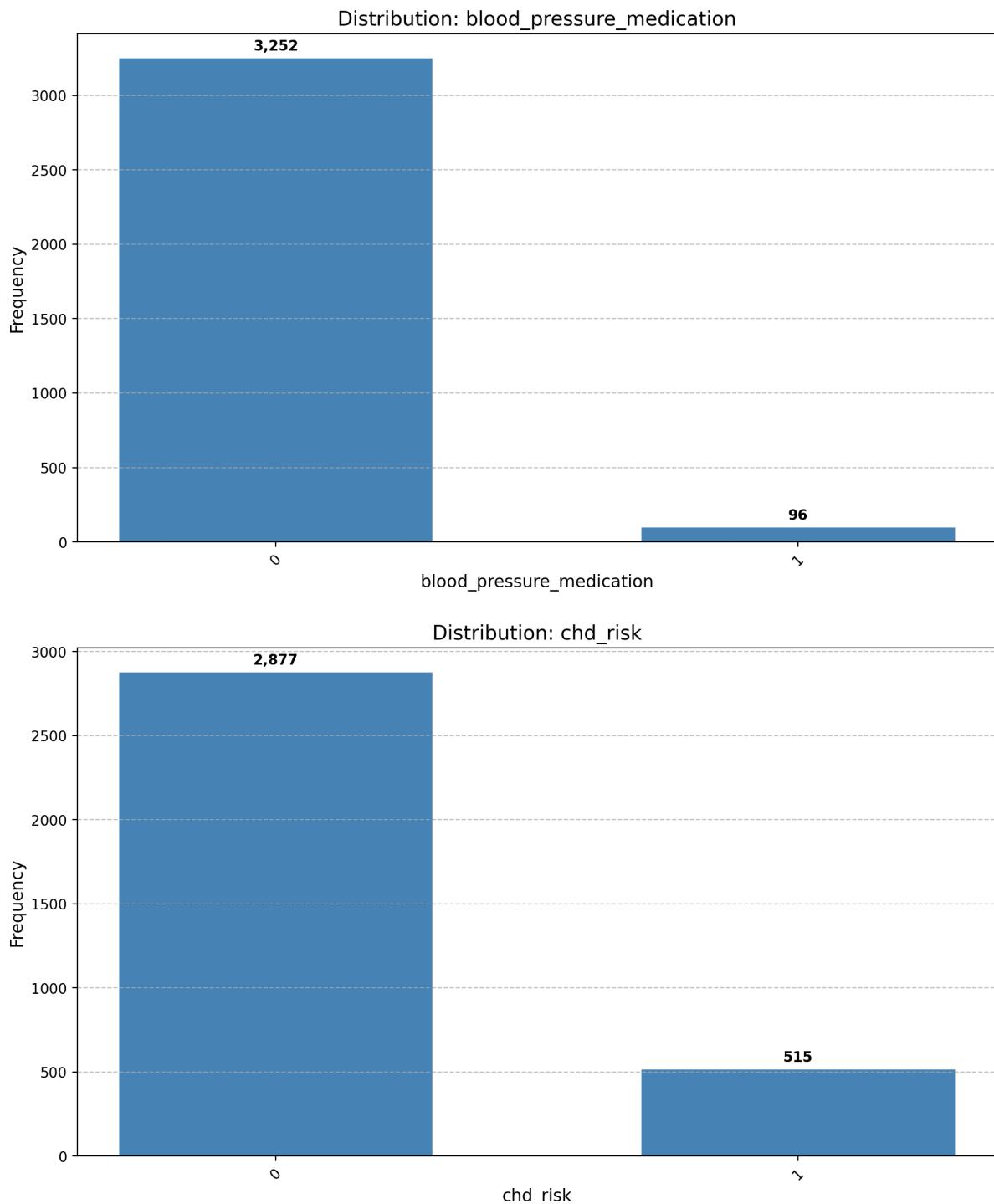


Numărul de videoclipuri

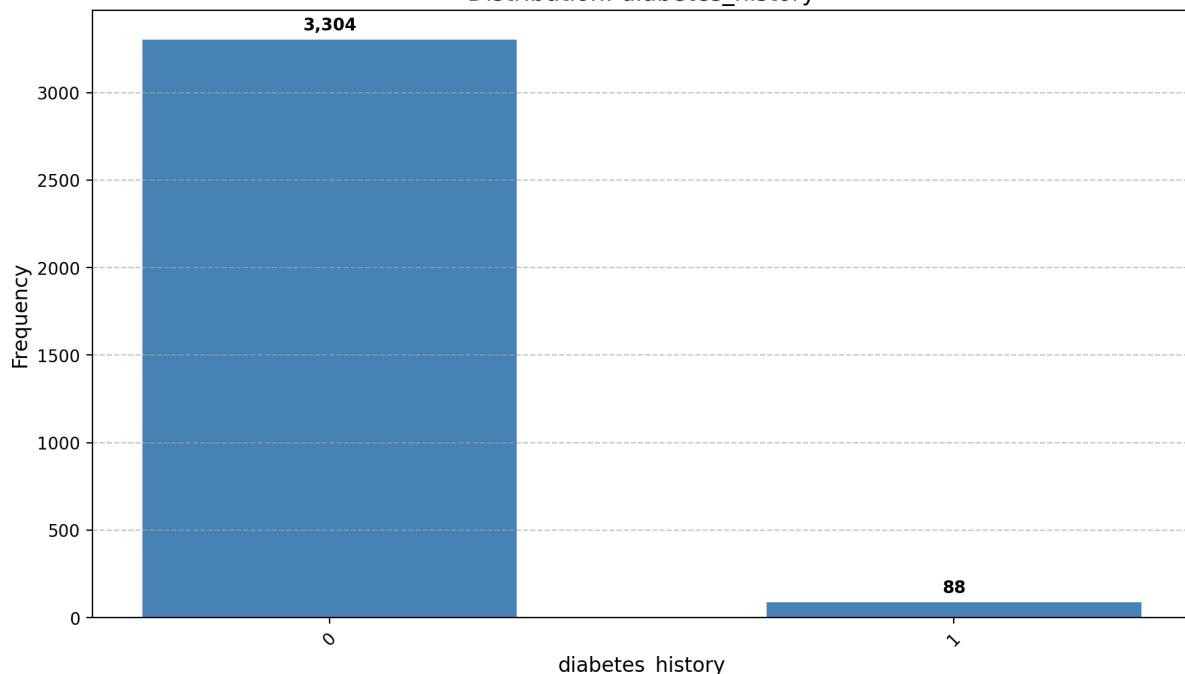
- Dominația textului - cu o medie de doar 1.25 videoclipuri per articol și majoritatea concentrată la 0-1, conținutul este predominant textual.

Funcția utilizata pentru analiza datelor categorice este `analyze_cat_data` care calculeaza statistici de baza pentru fiecare atribut discret, anume cate valori valide (nenule) exista, cate valori distincte sunt într-o coloana și generează cate o histogramă pentru fiecare atribut care compara cele două metrii.

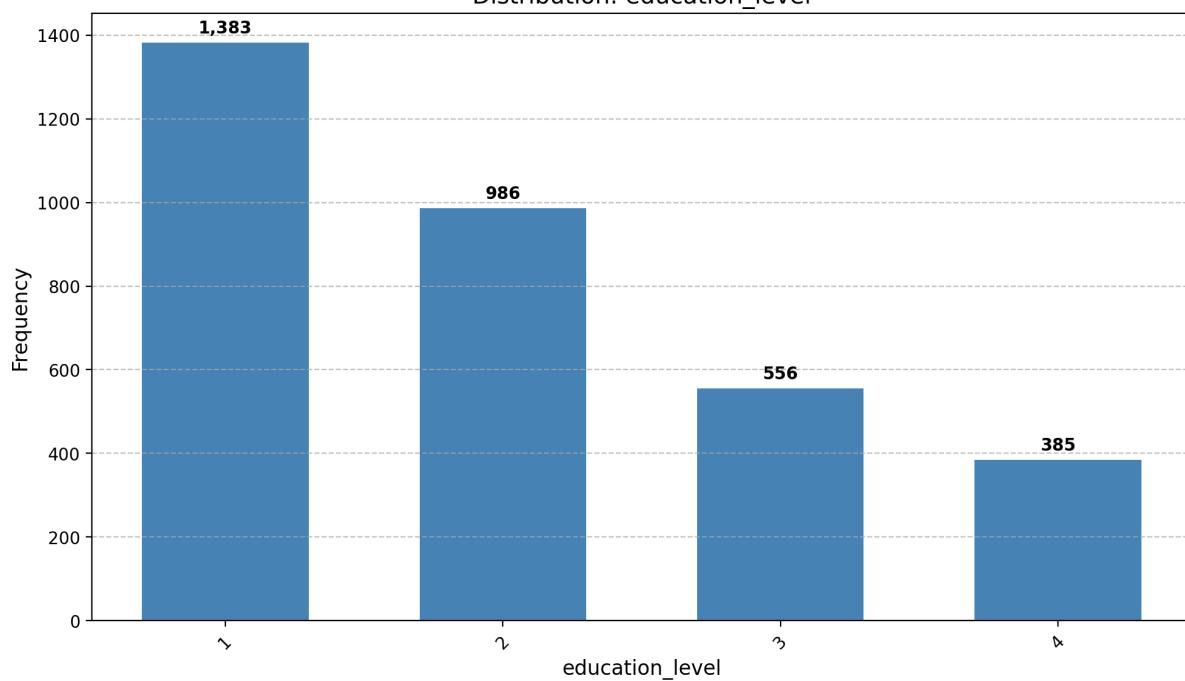
Grafice pentru analiza atributelor discrete referitoare la riscul dezvoltării unei boli coronariene:



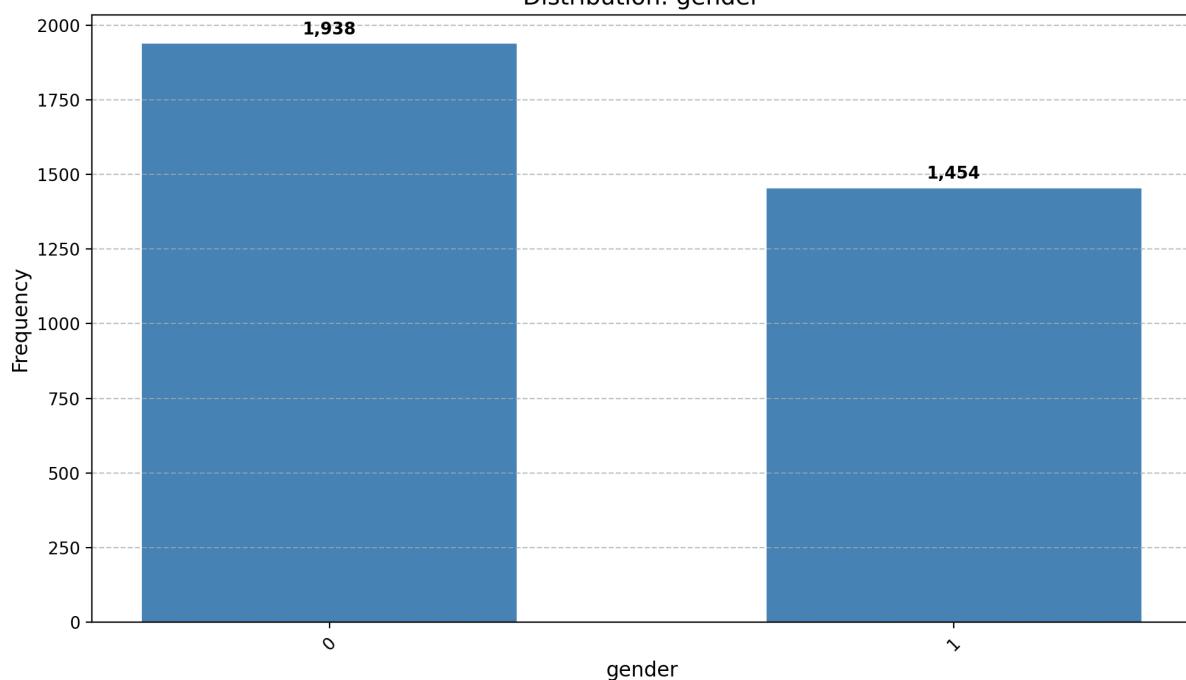
Distribution: diabetes_history



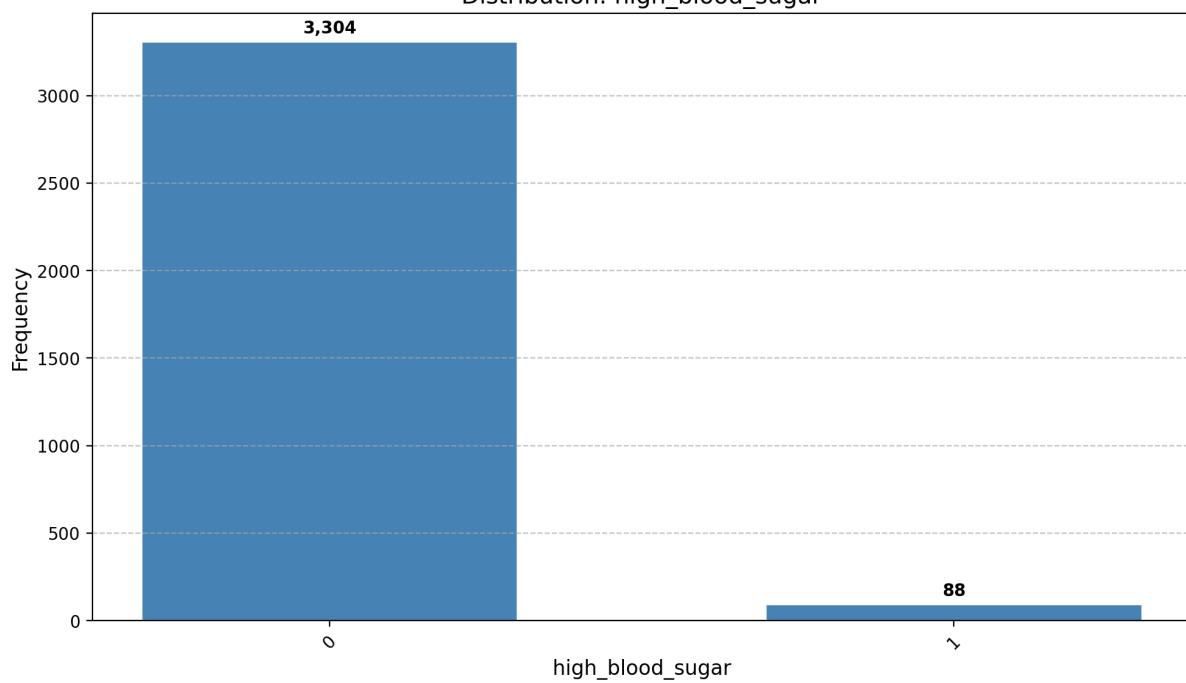
Distribution: education_level



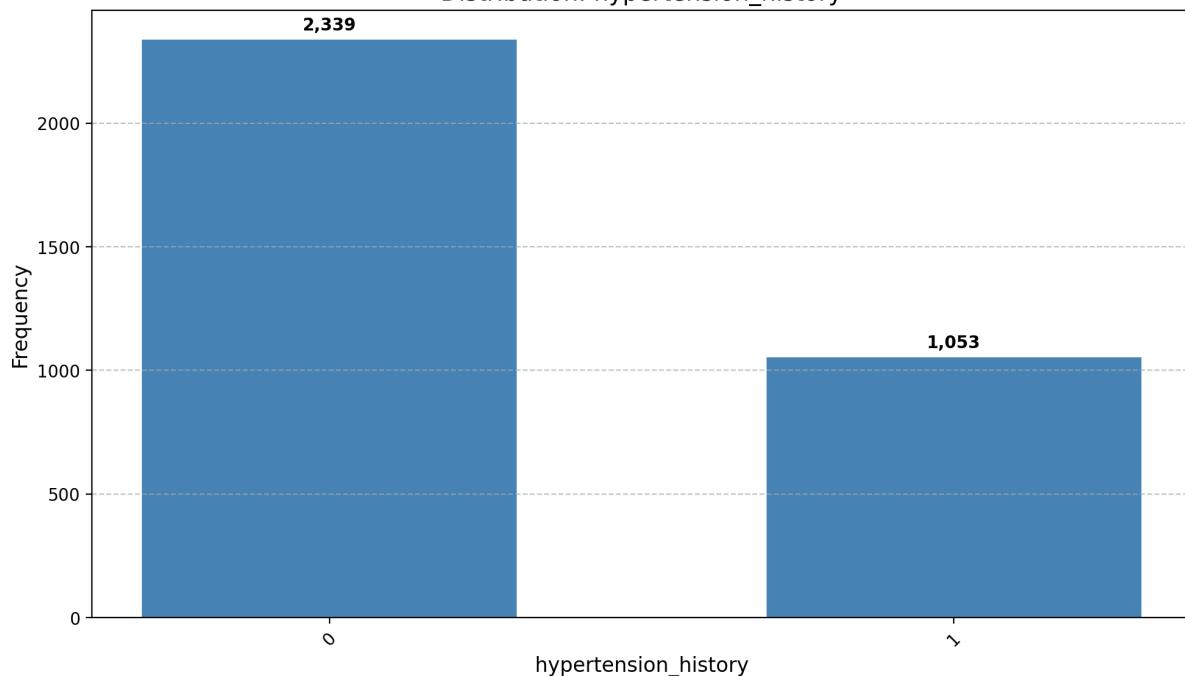
Distribution: gender



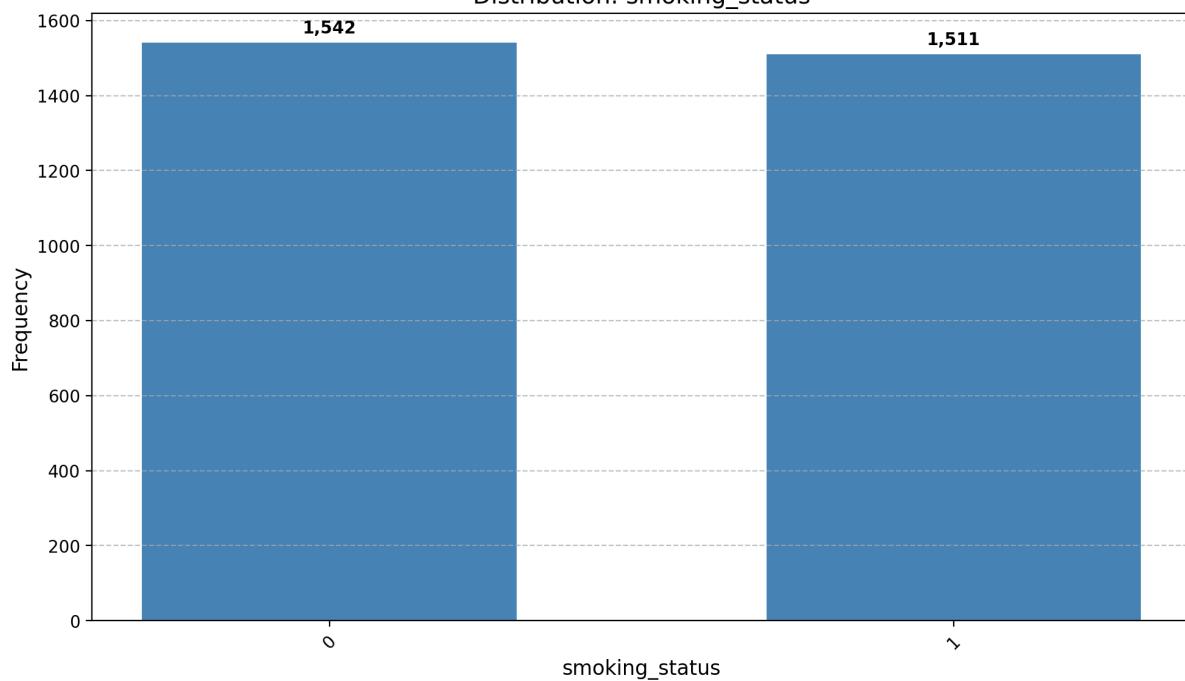
Distribution: high_blood_sugar

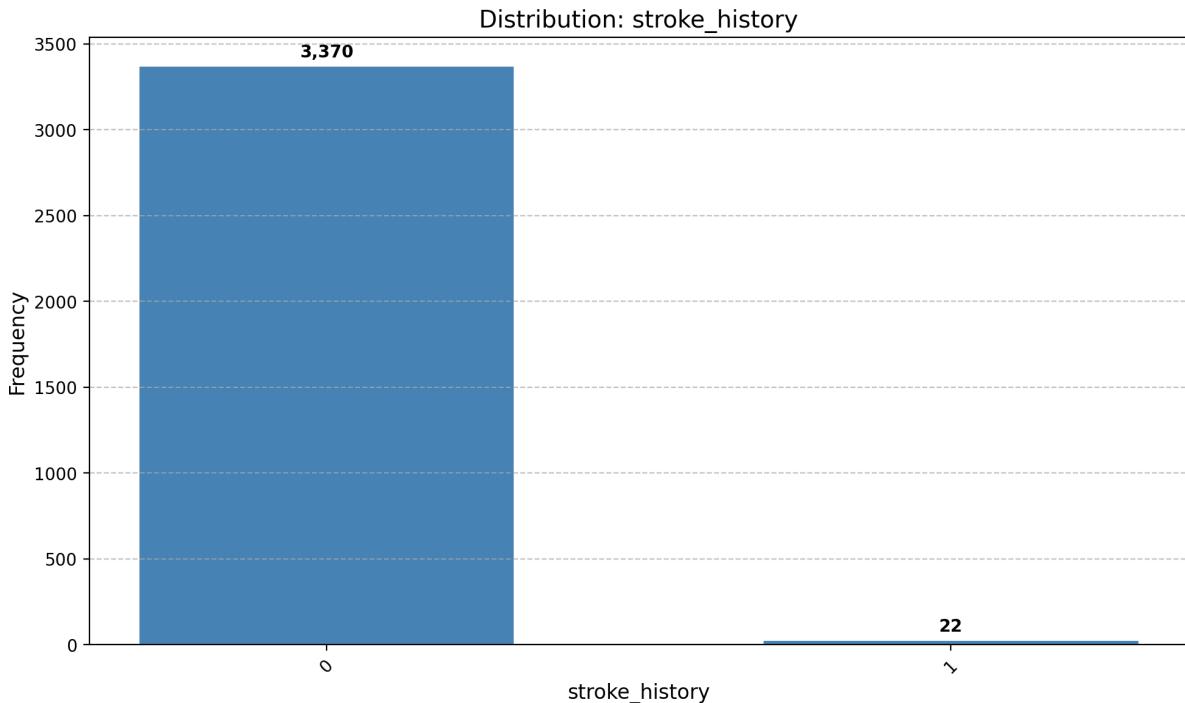


Distribution: hypertension_history



Distribution: smoking_status





Atribut complete (3,392 valori):

- chd_risk, diabetes_history, gender, high_blood_sugar, hypertension_history, stroke_history

Atribut cu date lipsă:

- smoking_status: 3,053 valori (lipsesc ~339, ~10.0%)
- education_level: 3,310 valori (lipsesc ~82, ~2.4%)
- blood_pressure_medication: 3,348 valori (lipsesc ~44, ~1.3%)

Atribut binare (2 categorii - 0/1):

- blood_pressure_medication, chd_risk, diabetes_history, gender, high_blood_sugar, hypertension_history, smoking_status, stroke_history

Excepție - atribut multinominal:

- education_level: 4 categorii distințe (1, 2, 3, 4 - niveluri educaționale ordonate)

Distribuția valorilor categorice

Dezechilibre extreme (>95% într-o categorie):

- stroke_history: 99.4% = 0 (3,370 vs 22)
- diabetes_history: 97.4% = 0 (3,304 vs 88)
- high_blood_sugar: 97.4% = 0 (3,304 vs 88)
- blood_pressure_medication: 97.1% = 0 (3,252 vs 96)

Distribuții echilibrate:

- smoking_status: 50.5% = 0, 49.5% = 1 (1,542 vs 1,511)
- gender: 57.1% = 0, 42.9% = 1 (1,938 vs 1,454)

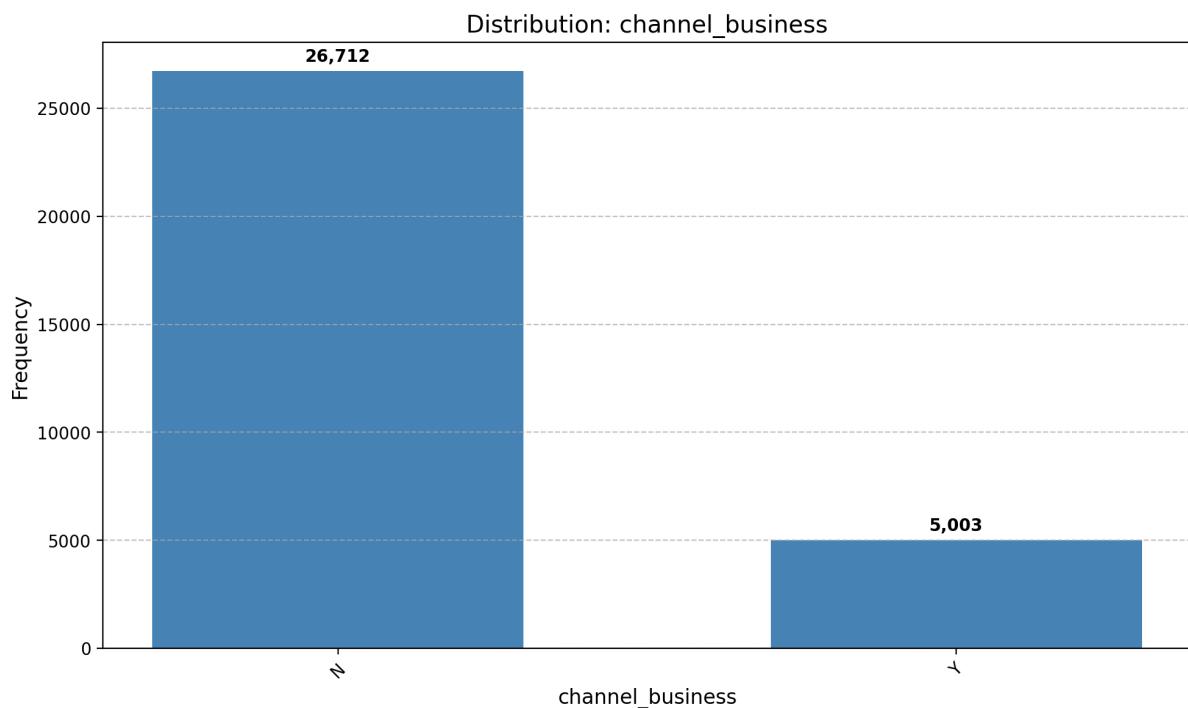
Distribuții moderate:

- chd_risk (țintă): 84.8% = 0, 15.2% = 1 (2,877 vs 515)
- hypertension_history: 69.0% = 0, 31.0% = 1 (2,339 vs 1,053)

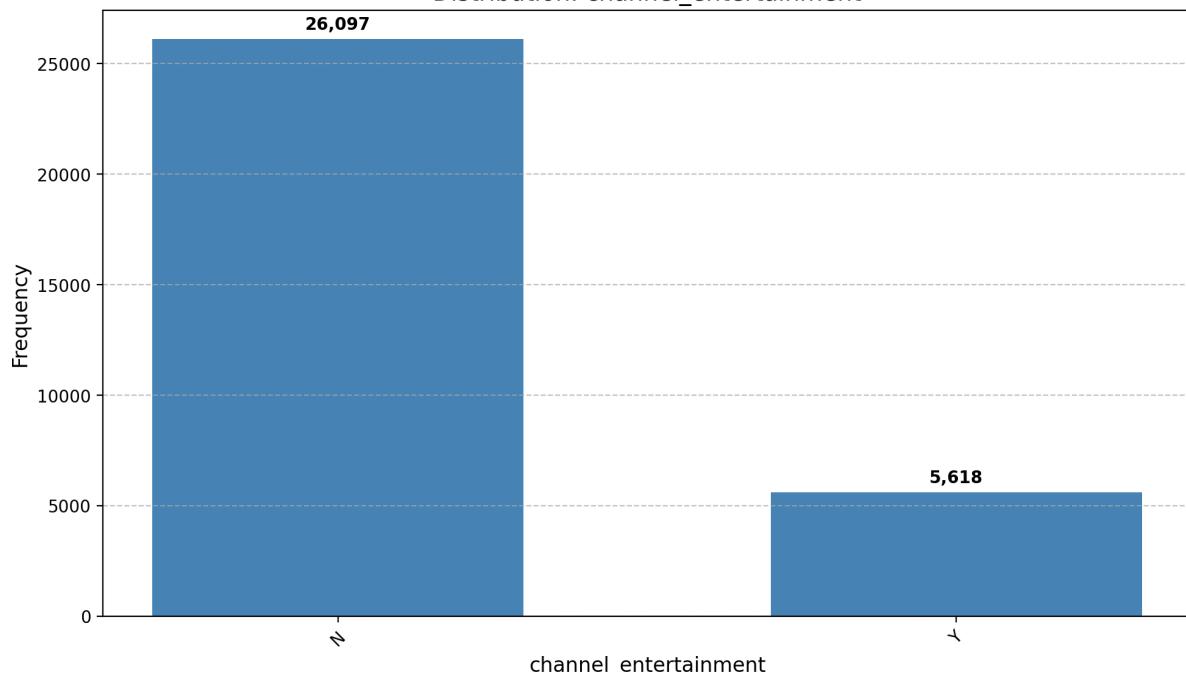
Distribuție multinomială:

- education_level:
 - Nivel 1: 41.8% (1,383)
 - Nivel 2: 29.8% (986)
 - Nivel 3: 16.8% (556)
 - Nivel 4: 11.6% (385)

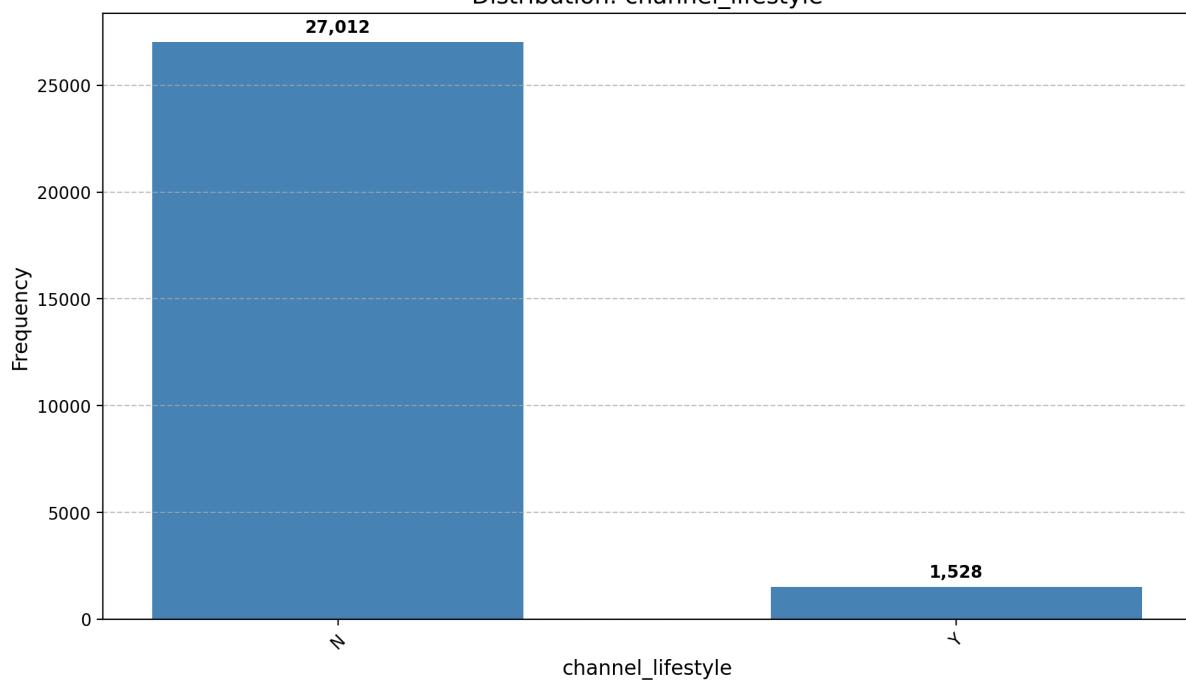
Grafice pentru analiza atributelor discrete referitoare la popularitatea știrilor în mediul online:



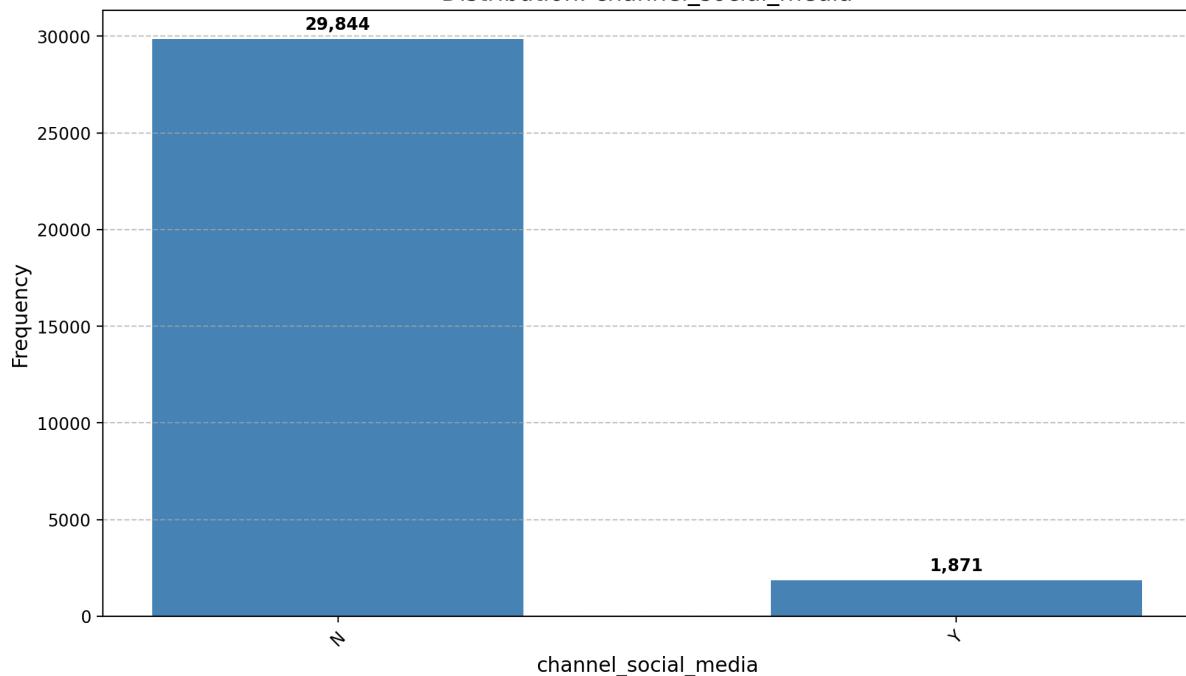
Distribution: channel_entertainment



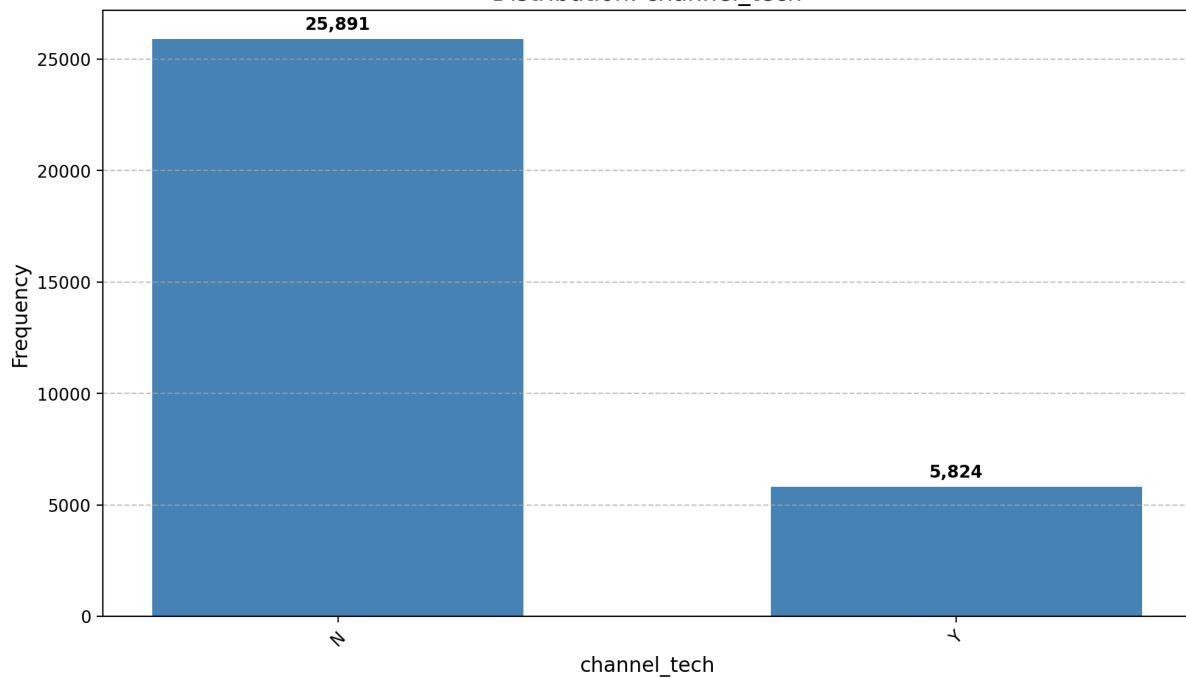
Distribution: channel_lifestyle

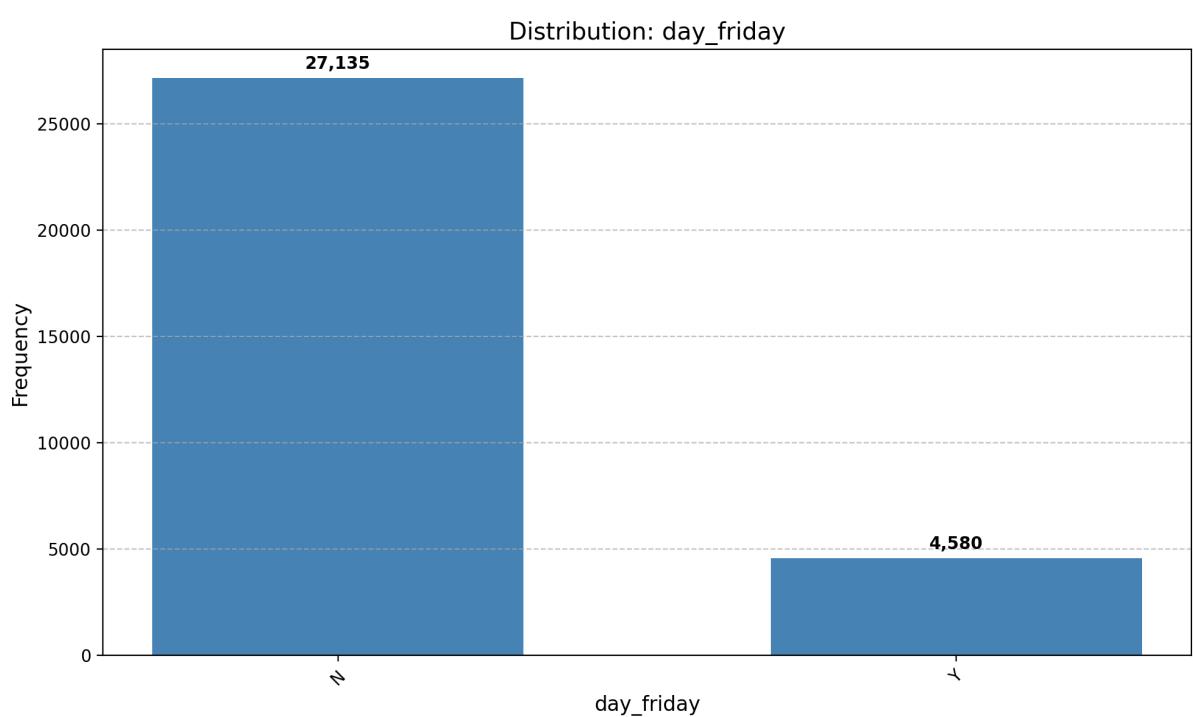
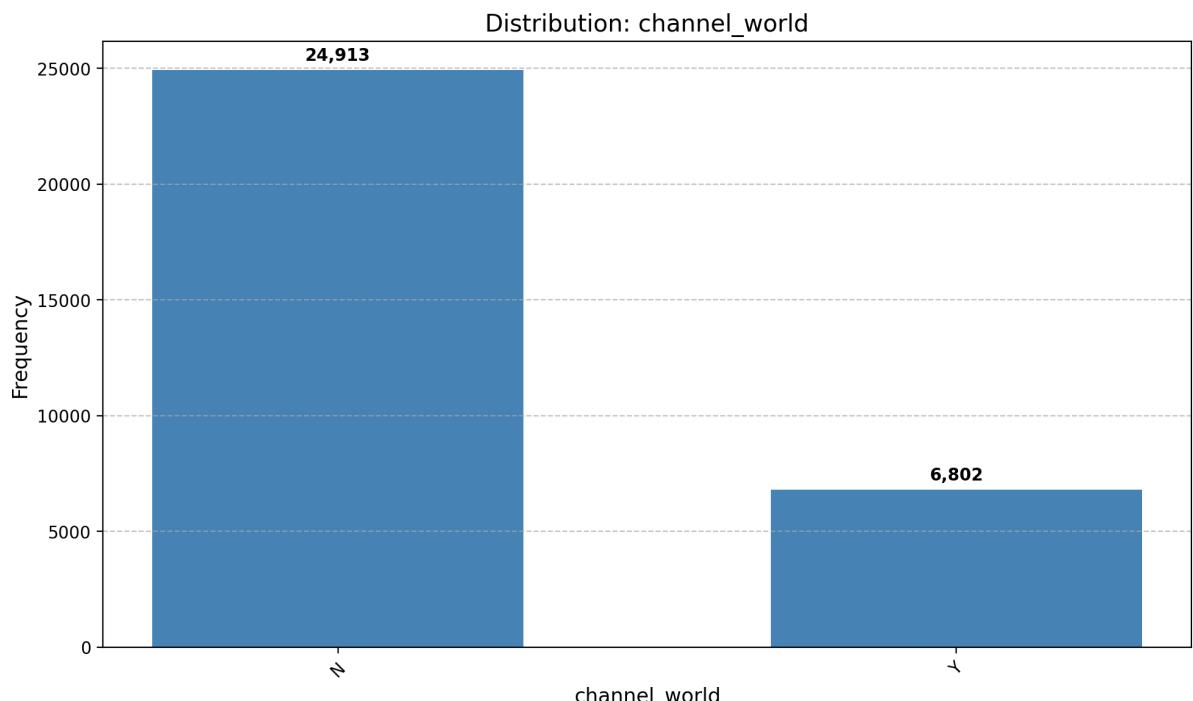


Distribution: channel_social_media

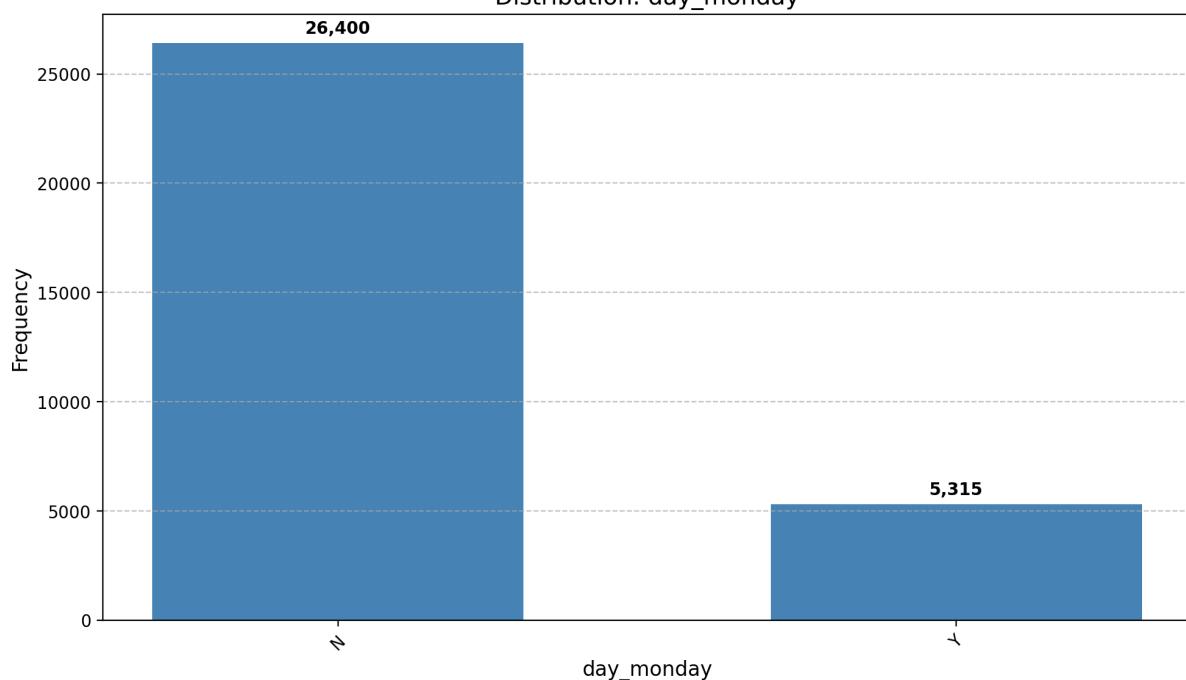


Distribution: channel_tech

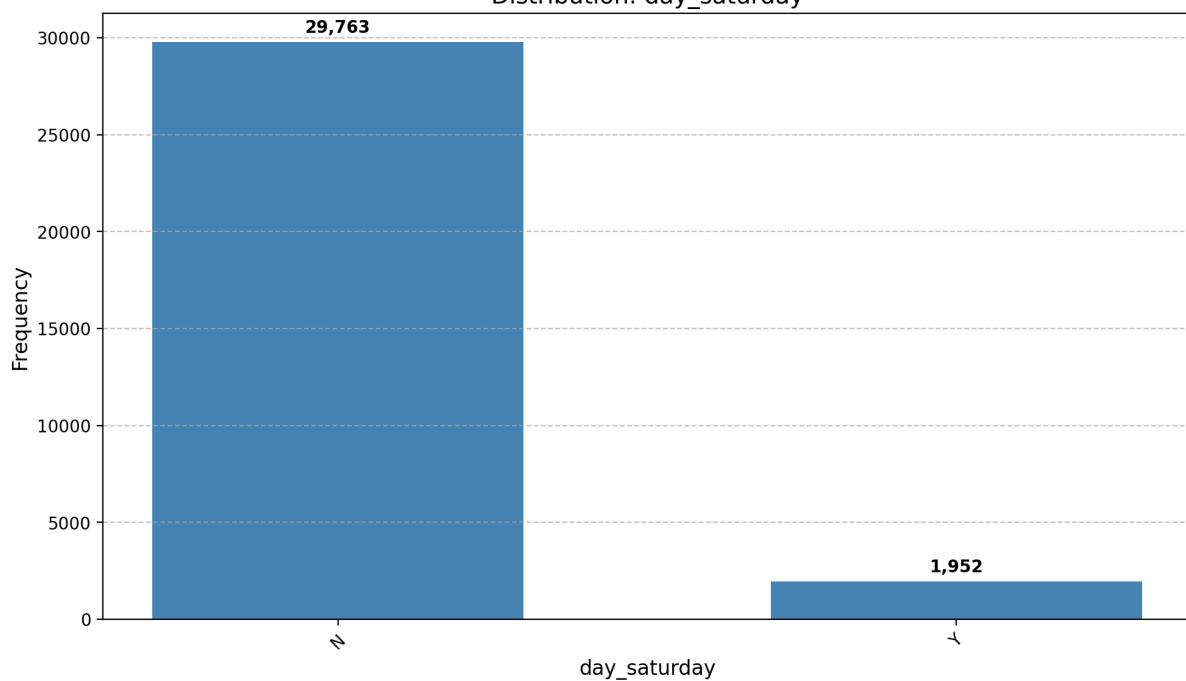




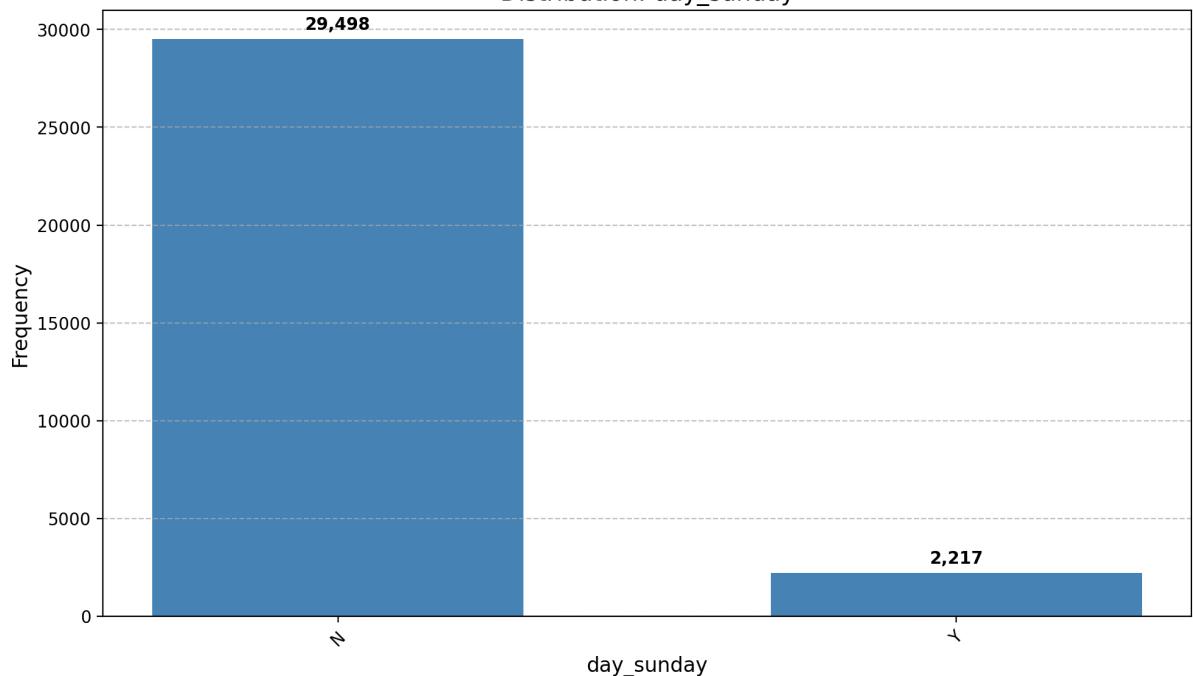
Distribution: day_monday



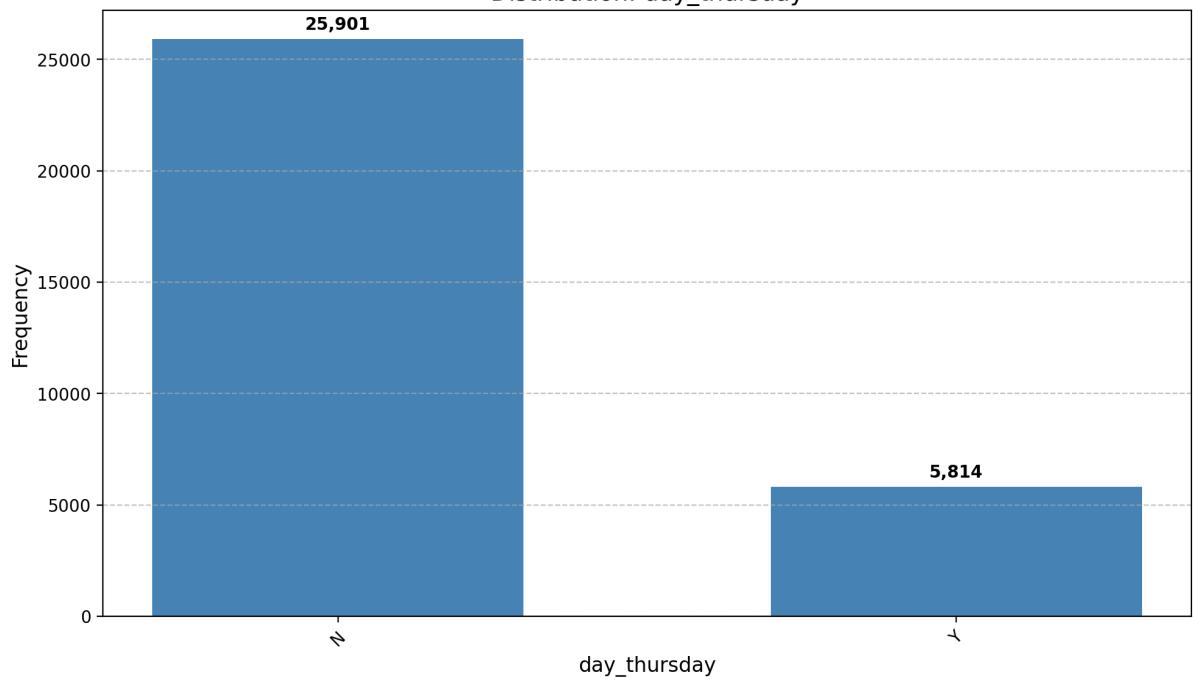
Distribution: day_saturday



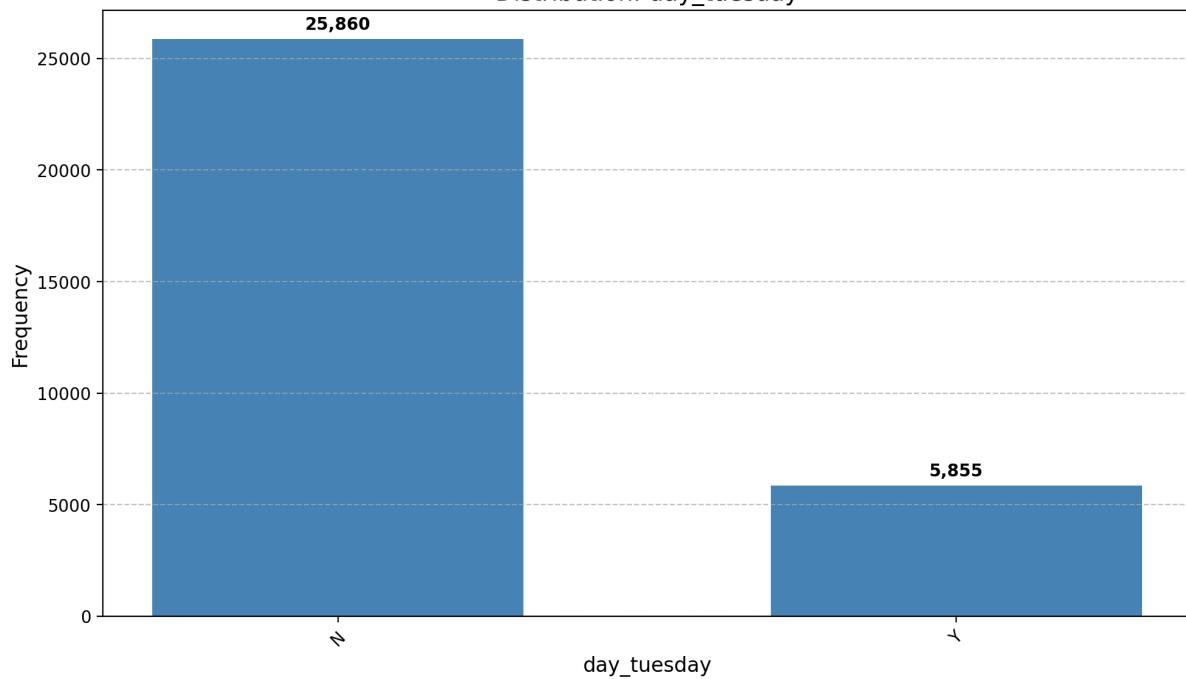
Distribution: day_sunday



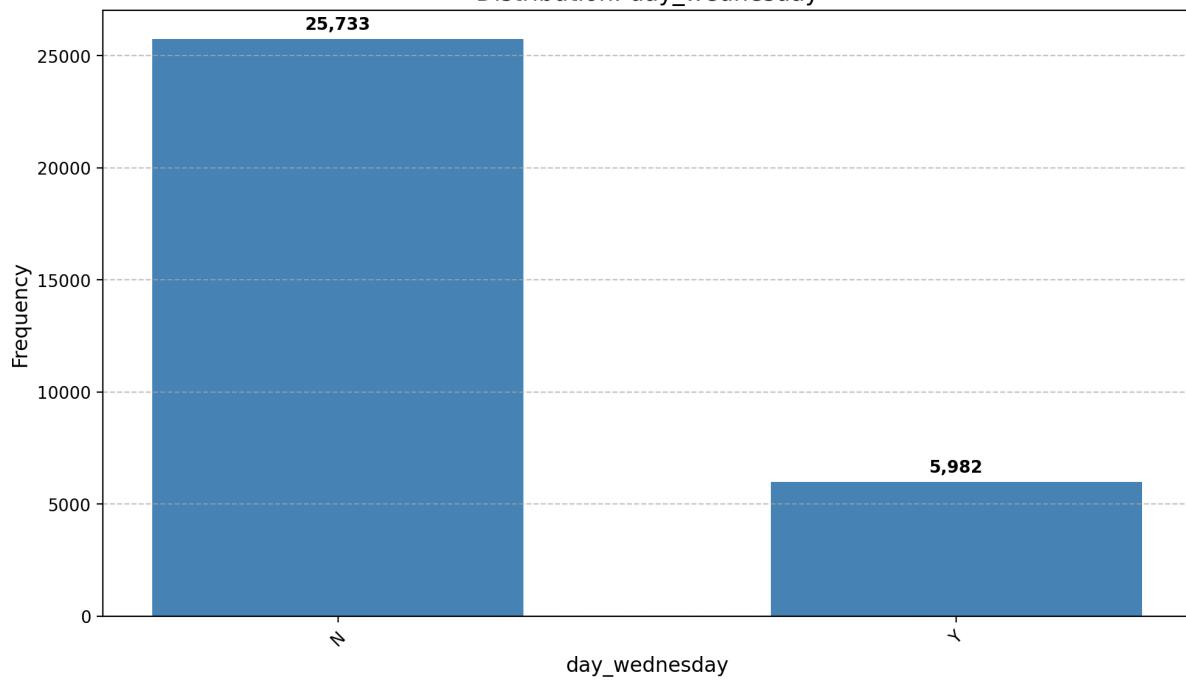
Distribution: day_thursday

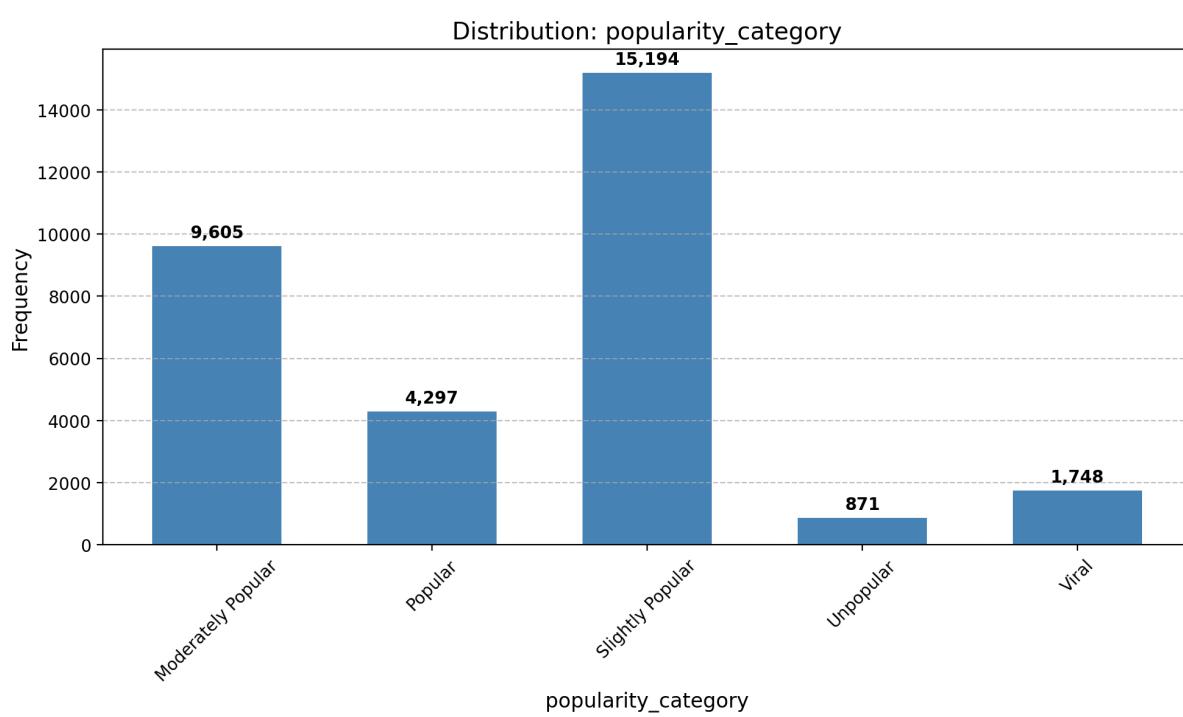
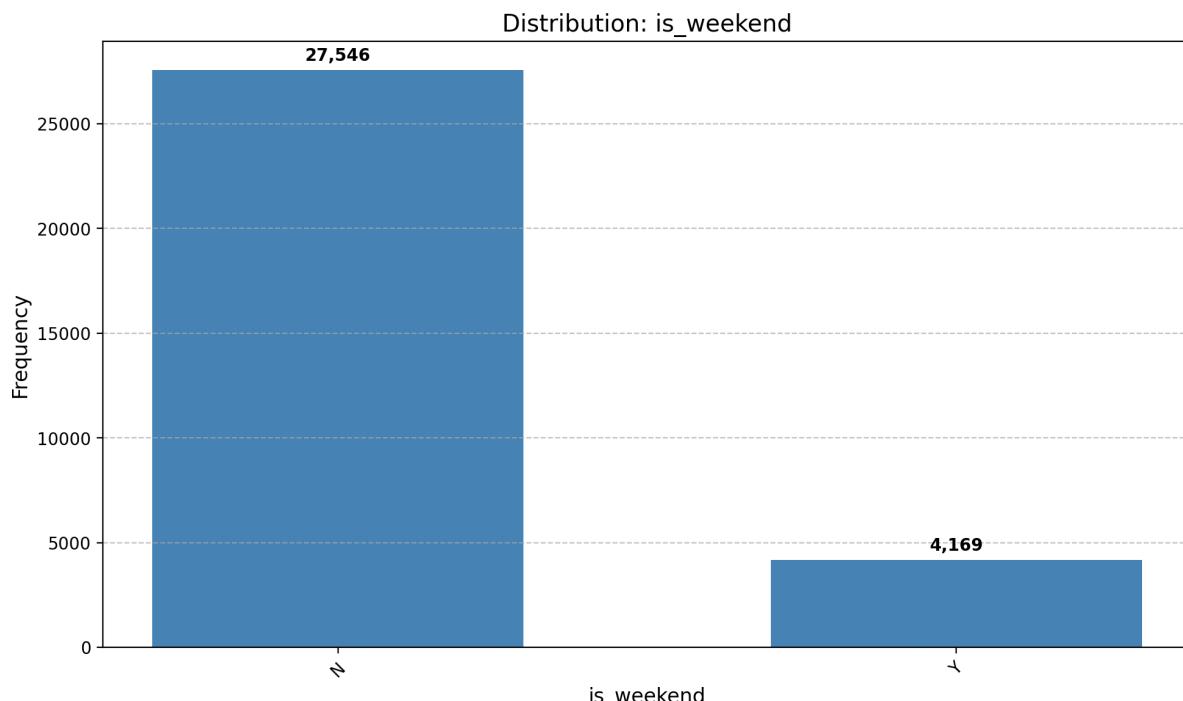


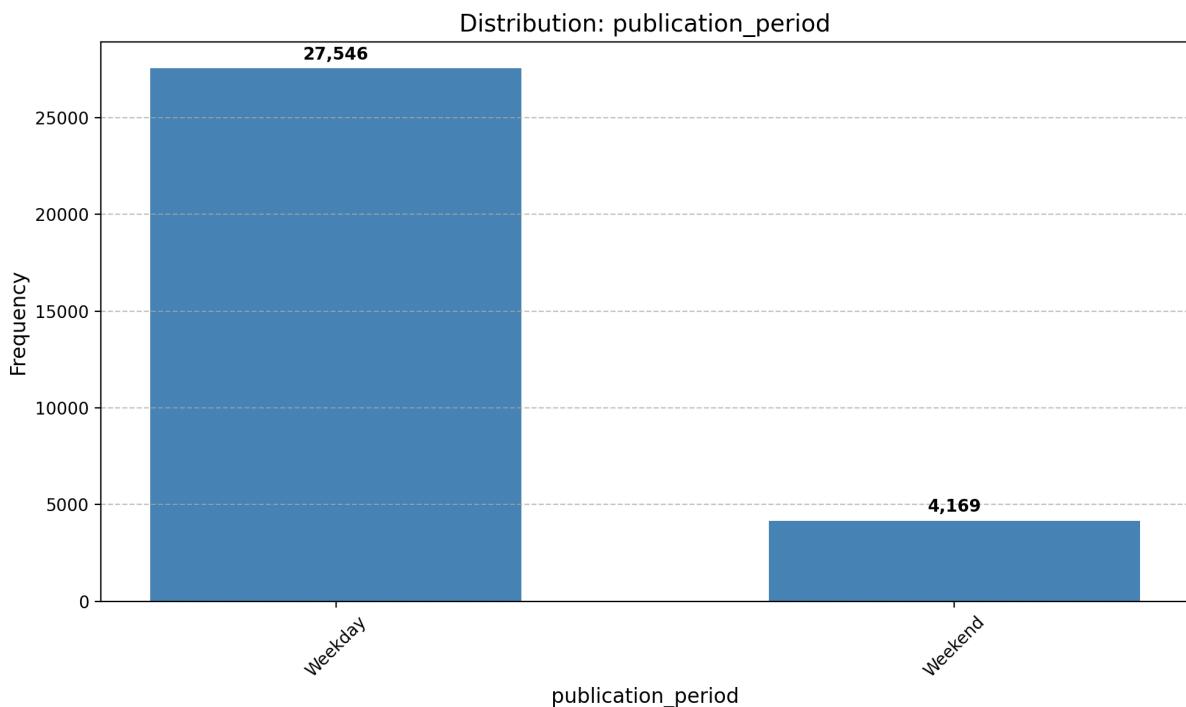
Distribution: day_tuesday



Distribution: day_wednesday







Distribuția valorilor categorice

Canale de conținut:

- channel_world: $78.5\% = 0, 21.5\% = 1 \Rightarrow$ dezechilibrat, dar cel mai echilibrat dintre toate
- channel_tech: $81.6\% = 0, 18.4\% = 1 \Rightarrow$ dezechilibrat
- channel_entertainment: $82.3\% = 0, 17.7\% = 1 \Rightarrow$ dezechilibrat
- channel_business: $84.2\% = 0, 15.8\% = 1 \Rightarrow$ dezechilibrat
- channel_social_media: $94.1\% = 0, 5.9\% = 1 \Rightarrow$ extrem dezechilibrat
- channel_lifestyle: $94.6\% = 0, 4.8\% = 1 \Rightarrow$ extrem dezechilibrat

Zilele săptămânii:

- day_wednesday: $81.1\% = 0, 18.9\% = 1 \Rightarrow$ dezechilibrat, dar cel mai echilibrat în comparație cu restul
- day_tuesday: $81.5\% = 0, 18.5\% = 1 \Rightarrow$ dezechilibrat, dar mai echilibrat în comparație cu restul
- day_thursday: $81.7\% = 0, 18.3\% = 1 \Rightarrow$ dezechilibrat, dar mai echilibrat în comparație cu restul
- day_monday: $83.2\% = 0, 16.8\% = 1 \Rightarrow$ dezechilibrat
- day_friday: $85.6\% = 0, 14.4\% = 1 \Rightarrow$ dezechilibrat
- day_saturday: $93.8\% = 0, 6.2\% = 1 \Rightarrow$ extrem de dezechilibrat
- day_sunday: $93.0\% = 0, 7.0\% = 1 \Rightarrow$ extrem de dezechilibrat

Variabile derivate:

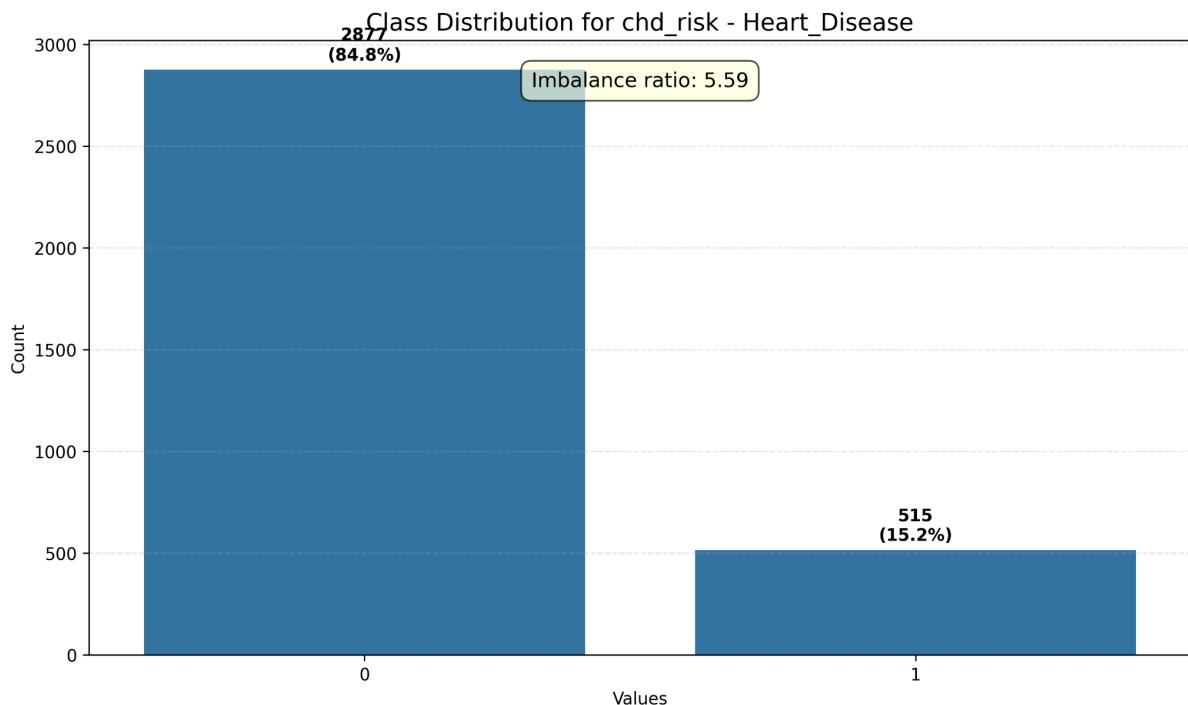
- is_weekend = publication_period: $86.9\% = 0, 13.1\% = 1 \Rightarrow$ identice, deci redundanță a datelor

Categorii de popularitate (multinomial):

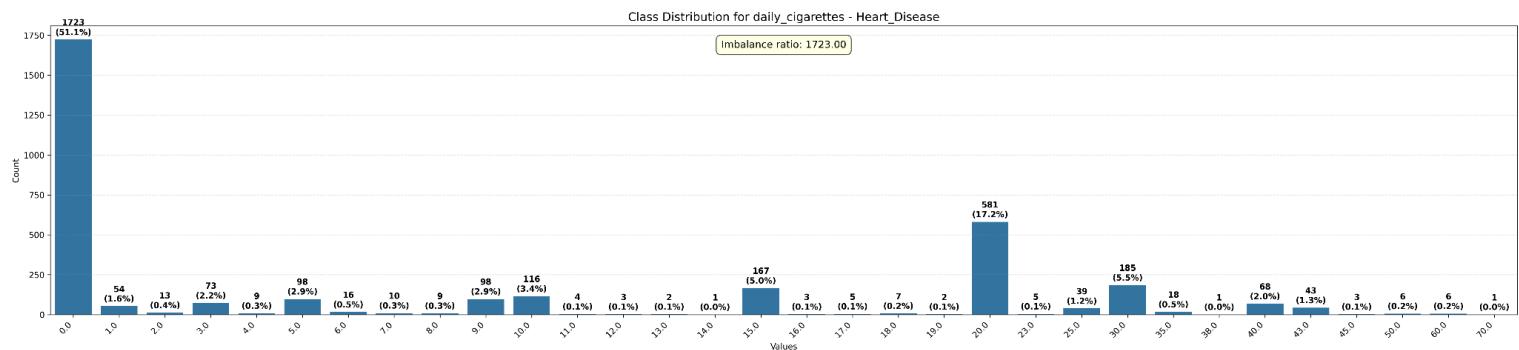
- Slightly Popular: 47.9% (15,194) - Majoritate
- Moderately Popular: 30.3% (9,605)
- Popular: 13.5% (4,297)
- Viral: 5.5% (1,748)
- Unpopular: 2.7% (871)

Analiza echilibrului de clase a fost realizata cu ajutorul functiei `analyze_class_balance` care primește ca input doar anumite coloane, intrucat setul de date pentru popularitatea în mediul online este unul prea mare și programul se blochează dacă toate atributele sunt prelucrate simultan. Funcția standardizează input-ul - fie string, fie listă de coloane, calculează statisticile pentru fiecare coloana, rata de dezechilibru și afișează rezultatele cu ajutorul unor ploturi.

Graficele de echilibru pentru probabilitatea dezvoltării unei boli coronariene:



Această variabilă binară prezintă un dezechilibru moderat cu 84.8% dintre pacienți clasificați ca având risc scăzut (0) și doar 15.2% cu risc înalt (1). Rata de dezechilibru de 5.59 indică faptul că clasa majoritară este de aproape 6 ori mai mare decât cea minoritară.

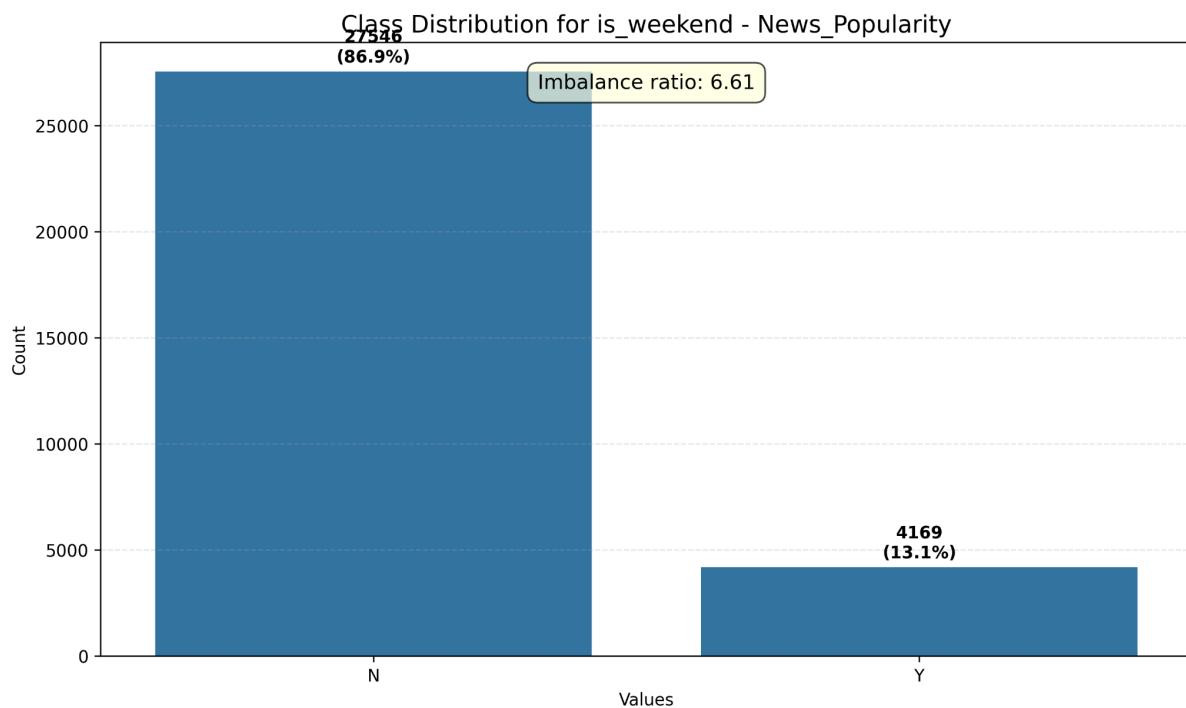


Această variabilă prezintă un dezechilibru extrem cu 51.1% dintre subiecți fiind nefumători (0 țigări/zi) și restul distribuit pe o gamă largă de 1-70+ țigări zilnic. Rata de dezechilibru de 1723.00 este excepțional de mare, reflectând faptul că majoritatea covârșitoare a populației nu fumează deloc, în timp ce fumătorii sunt distribuiți neuniform pe diferite niveluri de consum al țigărilor.

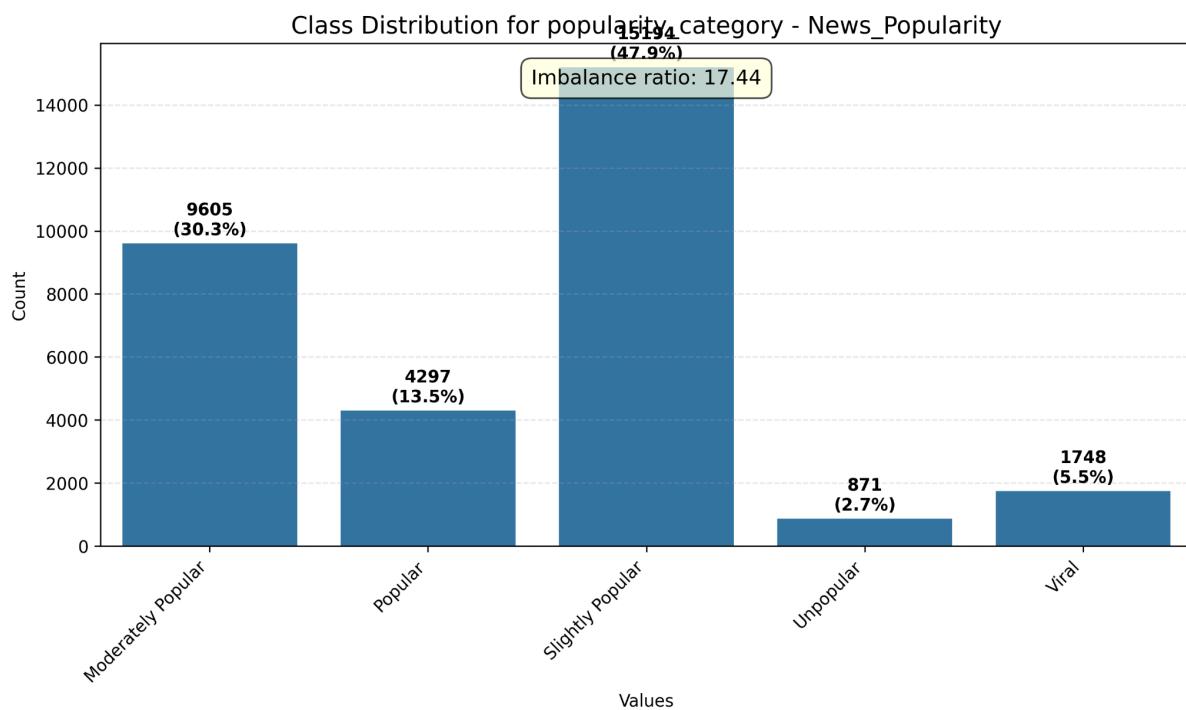
Restul atributelor sunt relativ dezechilibrate din cauza diversității acestora:

- blood_pressure_medication : 33.88
- cholesterol_level: 60.00
- stroke_history: 153.18
- systolic_pressure: 89.00
- hypertension_history: 2.22
- daily_cigarettes: 1723.00
- diastolic_pressure: 214.00
- heart_rate: 440.00
- smoking_status: 1.02
- diabetes_history: 37.55
- mass_index: 16.00
- blood_sugar_level: 144.00
- age : 144.00
- education_level: 3.59
- gender: 1.33
- glucose: 1.00
- total_cigarettes: 812.00
- high_blood_sugar: 37.55
- chd_risk: 5.59

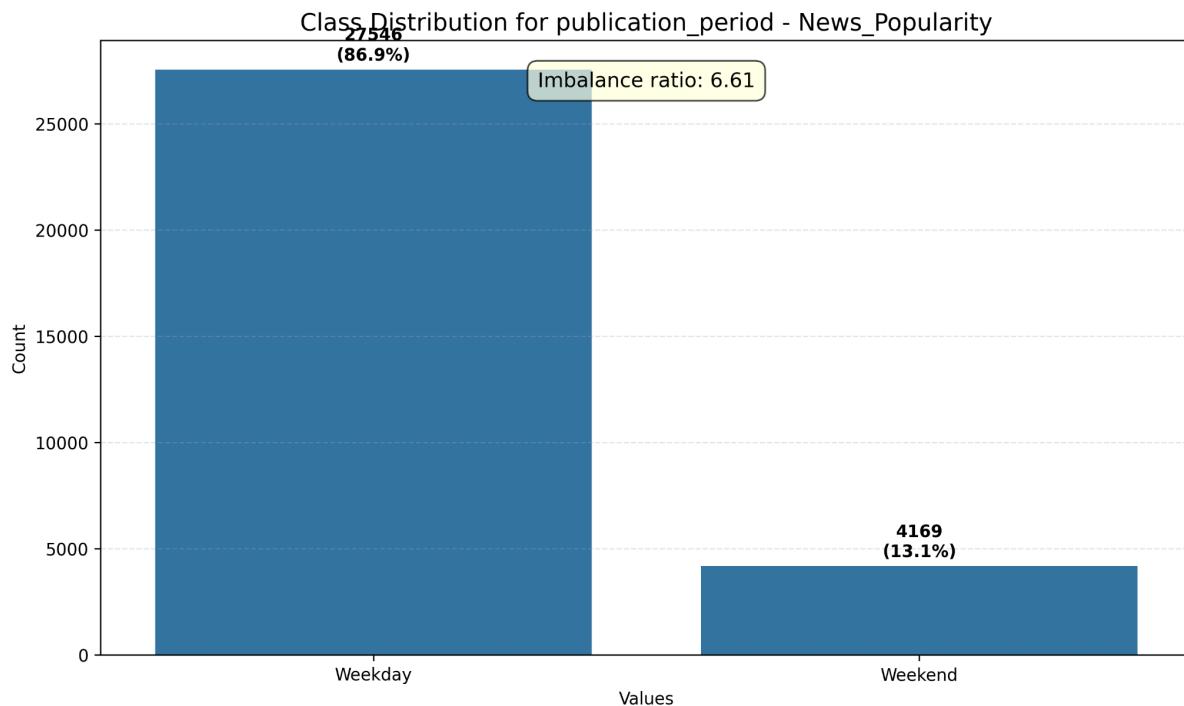
Graficele de echilibru pentru popularitatea știrilor în mediul online:



Graficul relevă o distribuție extrem de dezechilibrată între articolele publicate în timpul săptămânii și cele publicate în weekend. Cu 86.9% din articole publicate în zilele lucrătoare (N) și doar 13.1% în weekend (Y), se observă un raport de dezechilibru cu rata 6.61. Această distribuție sugerează că activitatea jurnalistică este concentrată predominant în timpul săptămânii.



Distribuția categoriilor de popularitate prezintă cea mai pronunțată dezechilibrare din toate graficele, cu o rata de 17.44. Categorie "Slightly Popular" domină cu 47.9% din articole, urmată de "Moderately Popular" cu 30.3%. Categoriile extreme sunt mult mai rare: "Popular" reprezentând 13.5%, "Viral" doar 5.5%, iar "Unpopular" este cea mai redusă cu 2.7%. Această distribuție sugerează că majoritatea articolelor de știri ating un nivel moderat de popularitate.



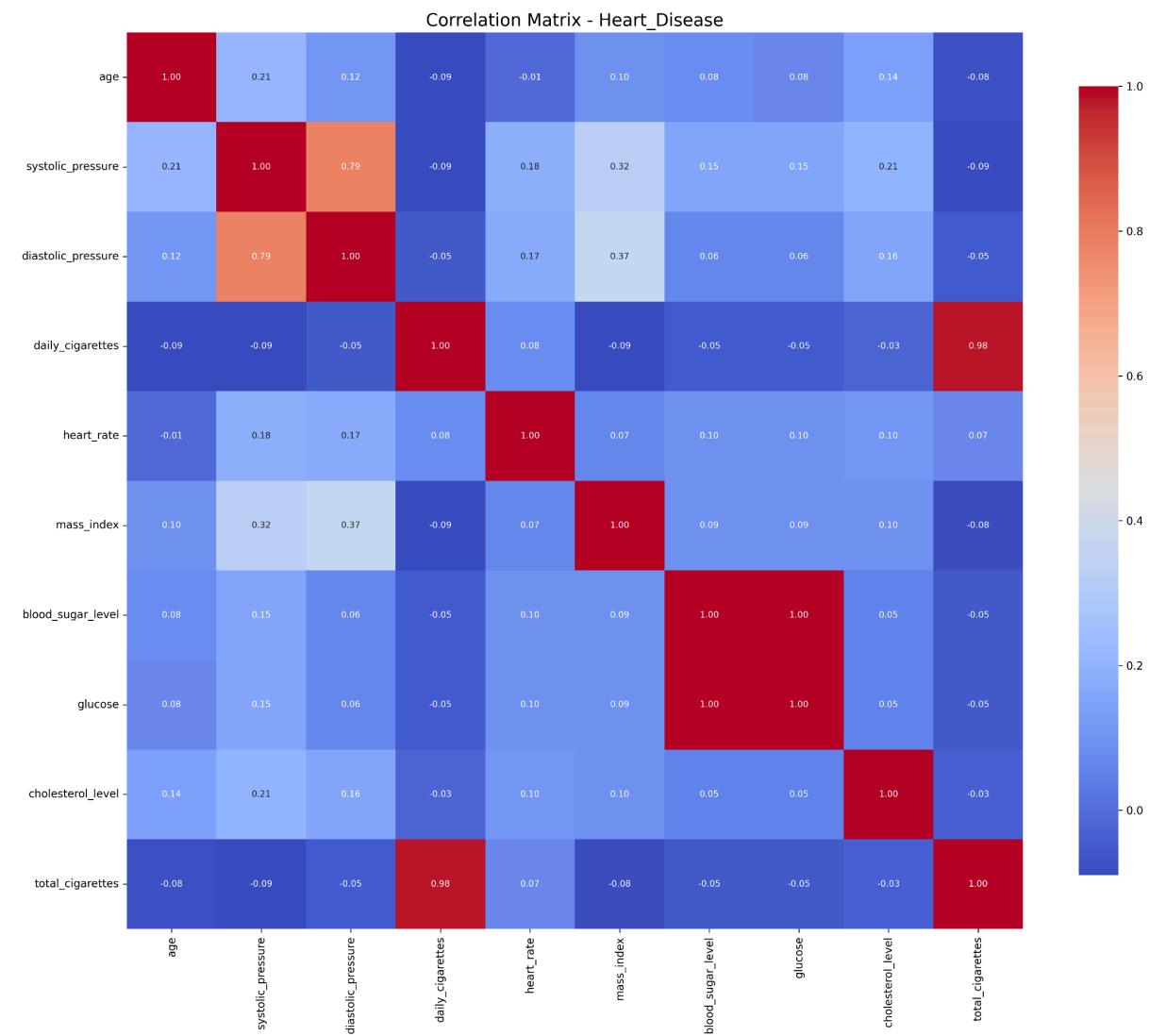
Graficul pentru perioada de publicare confirmă aceeași tendință observată în primul grafic, cu o distribuție identică: 86.9% articole publicate în "Weekday" și 13.1% în "Weekend", menținând același raport de dezechilibru de 6.61.

Pentru setul de date news_popularity echilibrul claselor este următorul, fapt ce relevă tot diversitatea plajelor de valori ale fiecărui atribut:

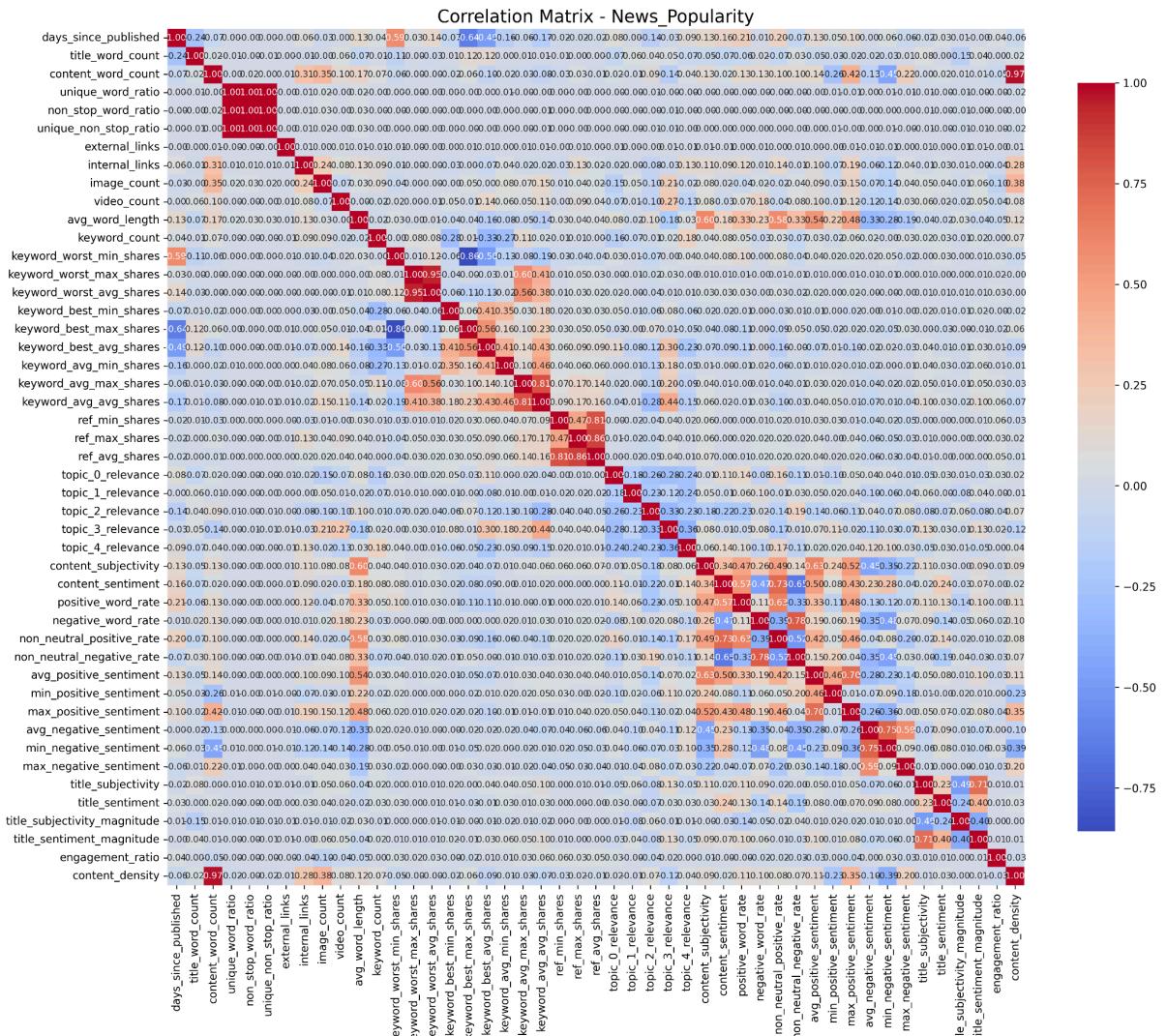
- url: 1.00
- channel_world: 3.66
- days_since_published: 14.00
- keyword_worst_min_shares: 18349.00
- title_word_count: 5918.00
- keyword_worst_max_shares: 1195.00
- content_word_count: 965.00
- keyword_worst_avg_shares: 555.00
- unique_word_ratio: 965.00
- title_subjectivity_magnitude: 16427.00
- title_sentiment_magnitude: 15918.00
- engagement_ratio: 858.00
- content_density: 1.00
- publication_period: 6.61
- popularity_category: 17.44
- and so on...

Analiza corelației dintre atrbute pentru a le elimina pe cele redundante a fost realizat cu funcțiile: `analyze_numerical_correlation` și `analyze_categorical_correlation` .

Funcția `analyze_numerical_correlation` analizează corelațiile Pearson între atrbutele numerice dintr-un dataset și identifică atrbutele care trebuie eliminate din cauza corelației ridicate. Funcția calculează matricea de corelație pentru toate atrbutele numerice, apoi creează un plot care este salvat ulterior ca imagine png. Pentru fiecare pereche de atrbute cu corelația mai mare de 0.8 în valoare absolută, funcția decide să eliminate atrbutul cu mai multe valori lipsă, evitând însă să eliminate atrbutul țintă (target).



Din analiza acestei matrice observam 3 perechi de atrbute puternic corelate: (total_cigarettes, daily_cigarettes), (high_blood_sugar, diabetes_history), (glucose, blood_sugar_level). Este evident cum un atrbut este influențat de celălalt din perechea sa. Deci, un atrbut este considerat redundant, neajutand semnificativ, deci va fi eliminat.

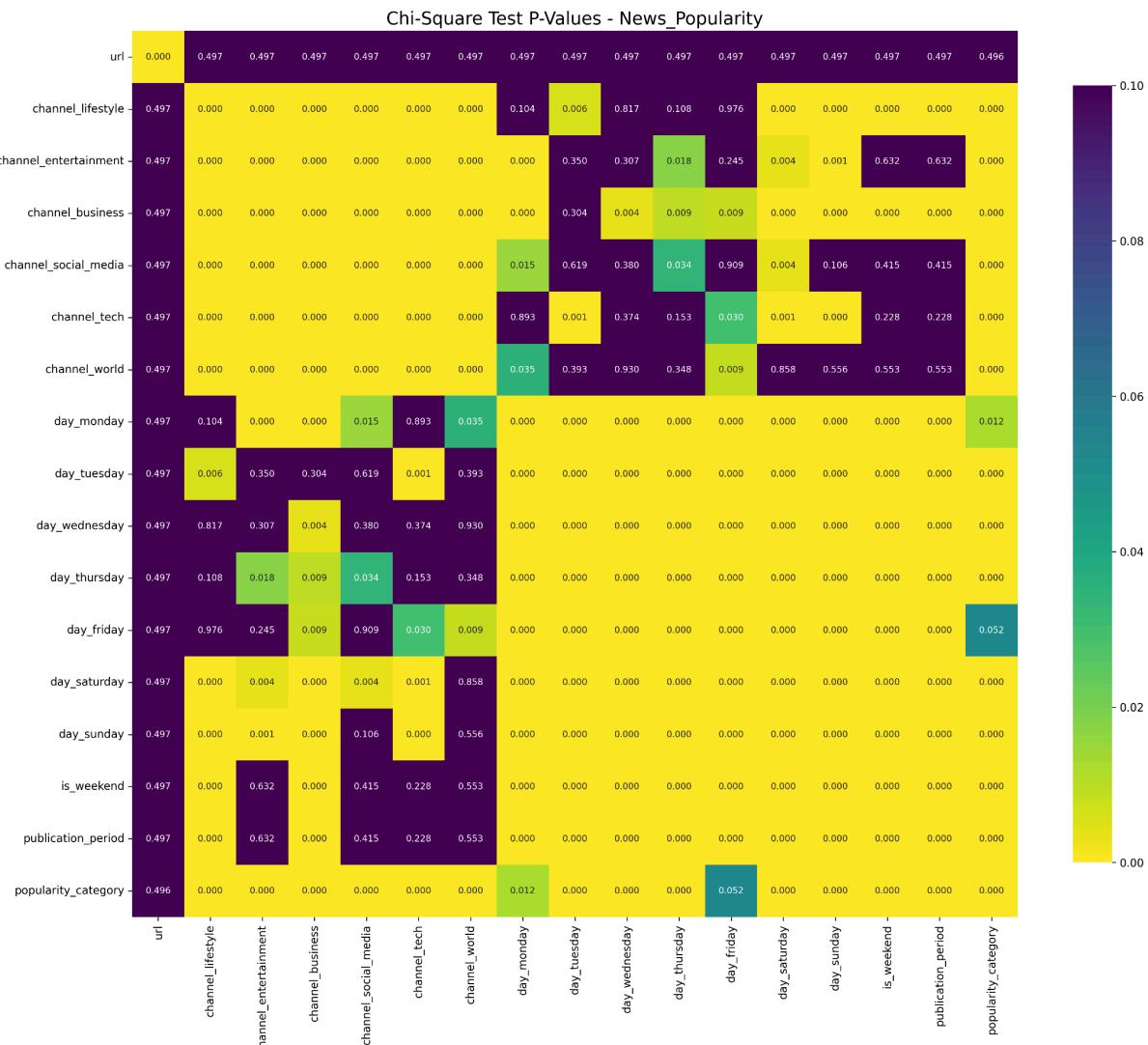


Prin analogie, avem următoarele perechi de attribute corelate: (non_stop_word_ratio, unique_word_ratio), (unique_non_stop_ratio, unique_word_ratio), (unique_non_stop_ratio, non_stop_word_ratio), (keyword_worst_avg_shares, keyword_worst_max_shares), (keyword_best_max_shares, keyword_worst_min_shares), (keyword_avg_avg_shares, keyword_avg_max_shares), (ref_avg_shares, ref_min_shares), (ref_avg_shares, ref_max_shares), (content_density, content_word_count), unul dintre attributele fiecărei perechi fiind necesar a fi eliminat.

Funcția `analyze_categorical_correlation` evaluează asocierea statistică între atributele categoriale folosind testul Chi-Square și identifică atributele corelate pentru eliminare. Funcția construiește o matrice de p-valori prin calcularea testului Chi-Square pentru fiecare pereche de atrbute categoriale, creând apoi o vizualizare a acestor p-valori. Perechile de atrbute cu $p \leq 0.05$ sunt considerate semnificativ corelate, iar din aceste perechi funcția elibera atrbutul cu mai multe valori lipsă, păstrând însă întotdeauna atrbutul său.



Din interpretarea acestei matrice se poate observa ca variabile cu asociere semnificativă cu chd_risk (boala cardiacă), respectiv variabile care au $p < 0.05$ sunt: gender, education_level, stroke_history, hypertension_history, diabetes_history, high_blood_sugar, blood_pressure_medication. Eliminarea se va face conform explicației de mai sus.



Aceasta matrice redă urmatoarele perechi corelate: (day_monday, channel_tech), (day_wednesday, channel_lifestyle), (day_wednesday, channel_world), (day_friday, channel_lifestyle), (day_friday, channel_social_media), (day_saturday, channel_world). Eliminarea se va face conform explicației de mai sus.

În cazul acestor matrice se observă perechile deloc corelate cu galben, iar cele cu o corelație puternică cu mov.

3.2 Preprocesarea Datelor ('data_preprocessing.py')

3.2.1 Date lipsă pentru un atribut într-un eșantion

Funcții utilizate `process_news_popularity_missing_data` și `process_heart_disease_missing_data`. Scopul acestora este de a verifica pentru fiecare set dacă există valori lipsă. Dacă există, datele sunt împărțite în numerice și categorice, este realizată imputarea fiecărui tip de date și apoi datele sunt combinate. În caz contrar, datele sunt returnate direct.

Pentru imputare sunt folosite funcțiile: `impute_categorical_data` care primește datele inițiale și lista cu coloane categorice, folosește SimpleImputer(strategy='most_frequent') pentru a înlocui valorile lipsă cu valoarea care apare cel mai frecvent în fiecare coloană și returnează un DataFrame cu datele categorice completate și `impute_numerical_data` care primește datele inițiale și coloanele numerice, folosește IterativeImputer (un algoritm mai avansat care impută fiecare coloană în funcție de celelalte, iterând de până la max_iter=300 ori pentru a îmbunătăți estimările) și returnează un DataFrame cu datele numerice complete.

Funcția care combina datele numerice și categorice imputate este `combine_imputed_data`.

Datele imputate sunt stocate pentru fiecare set de date la cheia 'heart_train_filled' din heart_set, respectiv 'news_train_filled' din news_set.

3.2.2 Valori extreme pentru un atribut într-un eșantion

Funcția `replace_outliers_data` detectează și înlocuiește valorile extreme (outlieri) din attributele numerice ale unui set de date, folosind regula IQR (interquartile range) pentru a identifica aceste valori. În loc să eliminate outlierii, funcția îi înlocuiește temporar cu NaN, apoi aplică IterativeImputer pentru a completa aceste valori. La final, valorile imputate sunt reintroduse doar în pozițiile unde au fost detectați outlieri, păstrând astfel structura originală a datelor și evitând pierderea de informație.

3.2.3 Eliminarea atributelor redundante

Funcțiile `remove_redundant_news_attributes` și `remove_redundant_heart_attributes` au rolul de a elimina coloanele redundante din seturile de date pentru popularitatea știrilor și boala cardiacă. Ele primesc o listă `attributes_to_drop` cu numele coloanelor ce trebuie eliminate și un DataFrame numeric deja preprocesat (cu outlieri tratați). Dacă lista este goală, nu se elimină nimic și se returnează o copie a datelor. Dacă lista conține attribute, acestea sunt eliminate folosind drop(columns=...), iar funcția afișează în consolă câte și care coloane au fost eliminate, precum și dimensiunile DataFrame-ului înainte și după curățare. Output-ul funcțiilor ce susține acest fapt este:

Removing 8 redundant attributes from Heart Disease:

- diabetes_history
- education_level
- daily_cigarettes
- smoking_status
- stroke_history
- blood_pressure_medication
- blood_sugar_level

- hypertension_history
- Heart Disease: (3392, 19) → (3392, 11)

Removing 22 redundant attributes from News Popularity:

- unique_word_ratio
- day_sunday
- channel_tech
- channel_social_media
- day_friday
- channel_world
- channel_lifestyle
- content_density
- channel_entertainment
- day_saturday
- keyword_worst_min_shares
- keyword_worst_max_shares
- day_wednesday
- day_monday
- keyword_avg_max_shares
- ref_min_shares
- is_weekend
- ref_max_shares
- non_stop_word_ratio
- day_tuesday
- channel_business
- day_thursday

News Popularity: (31715, 64) → (31715, 42)

3.2.4 Plaje valorice de mărimi diferite pentru atributele numerice

Operația de standardizare a datelor (aducere la aceeași scară) a fost realizată cu ajutorul funcției `standardize_numerical_data` care face o copie a datelor originale pentru a evita modificarea directă a DataFrame-ului original, verifică dacă coloanele numerice există în datele primite, această filtrare este utilă în caz că lista num_data conține coloane care au fost eliminate anterior, aplica standardizarea dacă există coloane valide. Pentru aceasta ultima operație am folosit StandardScaler care calculează media și deviația standard pentru fiecare coloană numerică, transformă valorile folosind formula: $z = (x - \mu) / \sigma$ unde x e valoarea, μ e media și σ e deviația standard. Funcția returnează datele standardizate.

Output-ul celor 4 operații de procesare a datelor a fost salvat în fisierele: 'heart_final_cleaned.csv', 'news_final_cleaned.csv', 'replaced_extreme_heart_num_data.csv', 'replaced_extreme_news_num_data.csv', 'standardized_heart_data.csv', 'standardized_news_data.csv', 'filled_heart_data.csv', 'filled_news_data.csv'.

3.3. Utilizarea algoritmilor de învățare automată

În aceasta tema am implementat: Arbori de Decizie, Păduri Aleatoare și MLP.

Funcțiile `prepare_heart_data_for_rf` și `prepare_news_data_for_rf` au fost folosite inițial pentru pregatirea datelor pe care se va face antrenarea algoritmului Păduri Aleatoare. Însă, ulterior am observat ca acestea pot fi folosite și pentru ceilalți doi algoritmi pentru a asigura coerența preprocesării.

Arbori de Decizie (`decision_tree.py`)

Codul implementează un pipeline complet de antrenare și testare pentru un clasificator de tip arbore decizional pentru două seturi de date: Heart Disease - clasificare binară (chd_risk) și News Popularity - clasificare multi-clasă (popularity_category).

Etapele principale sunt:

1. Pregătirea datelor (`prepare_heart_data_for_tree`, `prepare_news_data_for_tree`): se elimină atributurile neimportante, se tratează valorile lipsă, outlierii și se standardizează coloanele numerice, se aplică label encoding sau one-hot encoding pentru variabilele categorice.
2. Antrenarea modelului (`train_heart_decision_tree`, `train_news_decision_tree`): se configerează un DecisionTreeClassifier cu hiperparametrii optimizați, se antrenează modelul pe setul de date de antrenament, se prezice pe setul de test și se calculează acuratețea.
3. Rularea experimentelor (`run_decision_tree_experiments`): se rulează tot pipeline-ul și se returnează modelele și rezultatele obținute.

Hiperparametrii utilizati de DecisionTreeClasifier:

1. Pentru Heart Disease:

- Adâncime maximă (max_depth): 10
→ Limitează numărul maxim de niveluri în arbore pentru a preveni supraînvățarea.
- Număr minim de exemple pentru o divizare (min_samples_split): 20
→ O divizare a unui nod este permisă doar dacă are cel puțin 20 de exemple.
- Număr minim de exemple într-o frunză (min_samples_leaf): 10
→ O frunză (nod final) trebuie să conțină minimum 10 exemple, ceea ce crește generalizarea.
- Criteriul de decizie (criterion): 'gini'
→ Folosește indicele Gini pentru a măsura impuritatea la fiecare împărțire.
- Ponderare a claselor (class_weight): 'balanced'
→ Automat ajustează greutățile claselor inverse proporțional cu frecvența acestora, pentru a trata un posibil dezechilibru între clase (ex: cazuri de boală vs. fără boală).

2. Pentru News Popularity:

- Adâncime maximă (max_depth): 15
→ Arborele are voie să fie puțin mai adânc, având un set de date mai complex.
- Număr minim de exemple pentru o divizare (min_samples_split): 50
→ Un nod poate fi divizat doar dacă conține cel puțin 50 de exemple, reducând riscul de overfitting.
- Număr minim de exemple într-o frunză (min_samples_leaf): 20
→ Frunzele vor conține cel puțin 20 de exemple, ceea ce stabilizează deciziile finale.
- Criteriul de decizie (criterion): 'entropy'
→ Folosește entropia informațională (Information Gain) pentru a decide cele mai bune împărțiri.
- Ponderare a claselor (class_weight): 'balanced'
→ Ajută în cazul în care unele categorii de popularitate sunt subreprzentate.

Funcția `run_decision_tree_experiments` antrenează ambele modele, afișează acuratețea și returnează modelele împreună cu predicțiile și valorile reale, fiind astfel o funcție utilă pentru evaluare comparativă.

Paduri Aleatoare (`random_forests.py`)

Codul construiește o versiune îmbunătățită a metodei de clasificare pe bază de arbori, folosind algoritmul Random Forest, pentru aceleași două seturi de date: Heart Disease - clasificare binară (chd_risk) și News Popularity - clasificare multi-clasă (popularity_category).

Etapele principale:

1. Pregătirea datelor (`prepare_heart_data_for_rf`, `prepare_news_data_for_rf`): se preprocesează datele în mod similar cu cel pentru arborele de decizie (eliminări, completări, standardizări), se asigură că datele de antrenare și testare sunt compatibile (aceleași coloane).
2. Antrenarea modelului (`train_heart_random_forest`, `train_news_random_forest`): se antrenează un RandomForestClassifier cu hiperparametrii optimizați, se utilizează ponderarea claselor pentru a contracara dezechilibrul între clase, se evaluatează performanța pe setul de test.
3. Rularea experimentelor (`run_random_forest_experiments`): se rulează întreg procesul pentru ambele seturi, se returnează modelele, scorurile și predicțiile.

Hiperparametrii utilizati de RandomForestClassifier:

1. Pentru Heart Disease:

- Adâncime maximă a unui arbore (max_depth): None
 - Fără limitare, arborii pot crește până la adâncimea maximă necesară pentru a separa complet datele.
- Număr minim de exemple într-o frunză (min_samples_leaf): 1
 - Frunzele pot conține și un singur exemplu, ceea ce permite modelului o granularitate foarte fină (dar poate duce la overfitting).
- Criteriul de decizie (criterion): 'gini'
 - Folosește indicele Gini pentru a decide împărțirea nodurilor.
- Ponderarea claselor (class_weight): 'balanced'
 - Ajustează automat greutățile claselor pentru a combate dezechilibrul între clase (ex: mulți pacienți sănătoși vs puțini bolnavi).
- Numărul de estimatori (n_estimators): 1000
 - Modelul este format din 1000 de arbori, ceea ce îl face foarte robust și reduce varianța.
- Dimensiunea setului folosit de fiecare arbore (max_samples): 1500
 - Fiecare arbore este antrenat pe un eșantion aleator de 1500 de exemple din setul complet.
- Proporția de atribute folosite de fiecare arbore: Implicit (max_features='sqrt')
 - Pentru clasificare, default-ul este $\sqrt{nr. total de atribute}$. Acest lucru introduce diversitate între arbori.

2. Pentru News Popularity:

- Adâncime maximă a unui arbore (max_depth): 30
 - Se limitează adâncimea arborilor pentru a controla complexitatea modelului și a evita supraînvățarea.
- Număr minim de exemple într-o frunză (min_samples_leaf): 2
 - Fiecare frunză trebuie să conțină cel puțin două exemple.
- Criteriul de decizie (criterion): 'entropy'
 - Utilizează câștigul de informație pentru a împărți datele (preferabil pentru probleme cu multe clase).
- Ponderarea claselor (class_weight): 'balanced'
 - Corecteazădezechilibrele între clase (ex: știri foarte populare vs mai puțin populare).

- Numărul de estimatori (n_estimators): 1000
→ Ca și la modelul cardiac, sunt antrenați 1000 de arbori de decizie.
- Dimensiunea setului folosit de fiecare arbore (max_samples): 1500
→ Fiecare arbore este construit pe un subset bootstrap de 1500 exemple.
- Proporția de atribute folosite de fiecare arbore: Implicit (max_features='sqrt')
→ Folosirea doar a unui subset de atribute la fiecare split ajută la diversificarea arborilor.

Analog ca la Arboari de Decizie, Funcția `run_random_forest_experiments` antrenează ambele modele, afișează acuratețea și returnează modelele împreună cu predicțiile și valorile reale.

MLP ([mlp.py](#))

Codul implementează un pipeline complet de pregătire a datelor și antrenare a unui clasificator de tip MLP (MLPClassifier) pentru două seturi de date: Heart Disease - clasificare binară a riscului de boli cardiovasculare (chd_risk) și News Popularity - clasificare multi-clasă a articolelor de știri în funcție de popularitate (popularity_category).

Procesul este împărțit în trei etape:

1. Pregătirea datelor (`prepare_mlp_data`): elimină coloane irelevante, înlocuiește valori lipsă și outlieri, standardizează coloanele numerice și aplică one-hot encoding pentru coloanele categorice în cazul setului de știri.
2. Antrenarea modelului MLP (`train_heart_mlp`, `train_news_mlp`): împarte datele în caracteristici (x) și etichete (y), configerează și antrenează clasificatorul MLP, calculează acuratețea pe setul de test.
3. Rularea experimentelor (`run_mlp_experiments`): rulează complet pipeline-ul pentru ambele seturi, afișează rezultatele de performanță, returnează modelele, predicțiile și scorurile.

Hiperparametrii utilizati de MLPClassifier:

I. Arhitectura rețelei

1. Pentru Heart Disease:

- Straturi ascunse (hidden_layer_sizes): (100, 50, 25)
→ Rețea cu 3 straturi complet conectate: 100 neuroni în primul strat, 50 neuroni în al doilea, 25 neuroni în al treilea
- Funcția de activare (activation): 'relu'
→ ReLU este aleasă pentru a permite învățarea rapidă și a evita problema gradientului mic.

2. Pentru News Popularity:

- Straturi ascunse (hidden_layer_sizes): (200, 100, 50)
→ Rețea mai mare, potrivită pentru complexitatea crescută a datelor: 200 neuroni în primul strat, 100 neuroni în al doilea, 50 neuroni în al treilea
- Funcția de activare: 'relu'
→ Folosită și aici pentru performanță sporită pe date sparse sau high-dimensional.

II. Optimizator și parametrii de învățare

Pentru ambele modele:

- Optimizator (solver): 'adam'
→ Algoritm adaptiv bazat pe moment, foarte eficient pentru rețele neuronale.
- Rata de învățare (learning_rate_init):
 - 0.01 pentru Heart Disease (antrenare mai rapidă pe date mai mici)
 - 0.001 pentru News Popularity (date mai multe, nevoie de stabilitate)
- Strategia de adaptare a ratei de învățare (learning_rate): 'adaptive'
→ Rata scade automat dacă nu se observă îmbunătățiri în performanță.
- Număr maxim de epoci (max_iter):
 - 500 pentru Heart Disease
 - 300 pentru News
- Dimensiunea batch-ului: Implicită
→ MLPClassifier folosește batch-uri de mărime automată (mini-batch learning), adaptate intern.

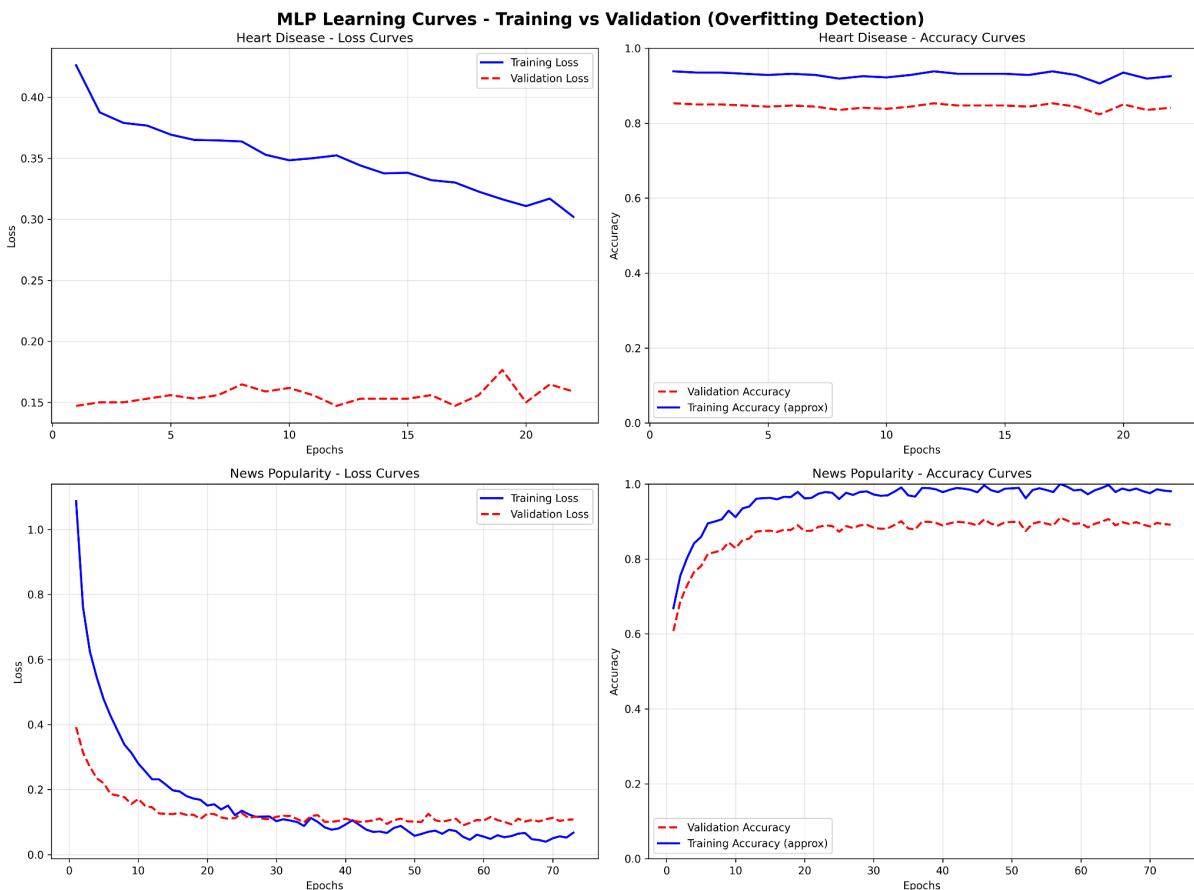
III. Regularizare și prevenirea overfitting-ului

Pentru ambele modele:

- Coeficient de regularizare L2 (alpha):
 - 0.001 pentru Heart
 - 0.0001 pentru News
→ Regularizare L2 aplicată pe ponderi (ridge penalty) pentru a controla complexitatea modelului.

- Early stopping (early_stopping): True
→ Antrenarea se oprește automat dacă performanța pe setul de validare nu se îmbunătățește.
- Proporție date de validare (validation_fraction): 0.1
→ 10% din datele de antrenament sunt rezervate pentru validare internă.
- Toleranță pentru oprirea antrenării (n_iter_no_change):
 - 20 epoci fără îmbunătățire pentru Heart
 - 15 epoci pentru News

Funcția `plot_mlp_learning_curves` are rolul de a vizualiza procesul de învățare al unui model MLP pe cele două seturi de date diferite. Mai specific, funcția generează o figură cu 4 subgrafice: stânga: curbe de pierdere (loss), dreapta: curbe de acuratețe (accuracy), sus pentru Heart Disease și jos pentru News Popularity, afișează evoluția antrenării: pierderea pe datele de antrenare (Training Loss), pierderea pe datele de validare (Validation Loss), acuratețea pe datele de validare (Validation Accuracy) și o aproximare a acurateței pe datele de antrenare (Training Accuracy (approx)), verifică automat overfitting-ul (dacă pierderea la validare crește sau rămâne semnificativ mai mare decât cea la antrenare, funcția semnalează potențialul overfitting (implicit în cod, deși fără text informativ). În final, graficul este salvat ca png.



Interpretare grafică:

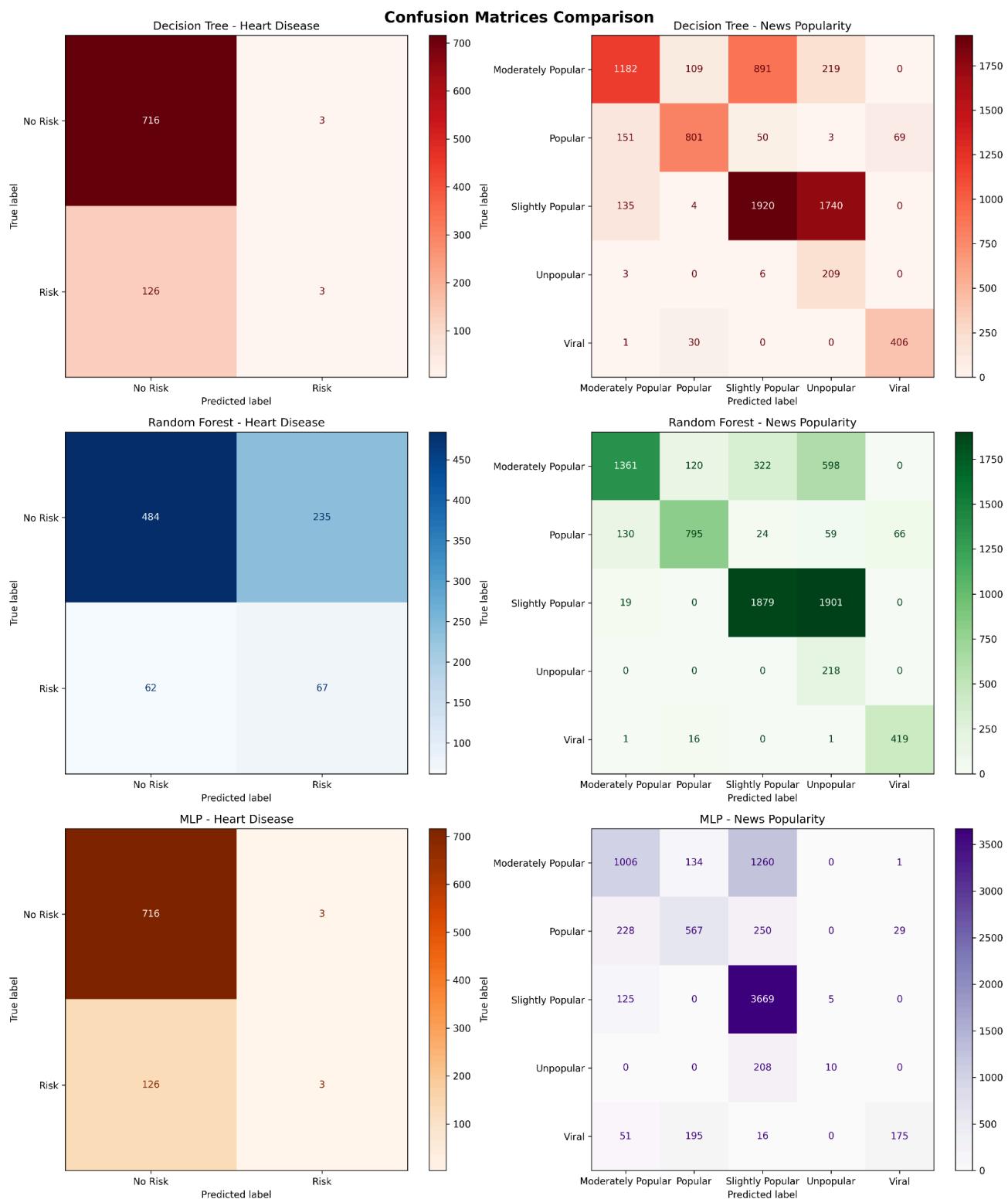
1. Pentru Heart Disease

- Loss: Pierderea la validare este constantă și joasă (sub 0.2), în timp ce pierderea la antrenare este mai mare. Asta poate indica un model stabil (underfitting ușor) sau faptul că validarea este simplificată ori pe un set ușor.
- Accuracy: Acuratețea de validare este stabilă (~0.85), puțin mai joasă decât cea de antrenare (~0.90), dar nu există semne clare de overfitting.

2. Pentru News Popularity

- Loss: Pierderea la antrenare scade rapid și ajunge foarte jos (~0.05), în timp ce validarea stagnează la ~0.1–0.15.
- Accuracy: Acuratețea de antrenare ajunge aproape de 100%, dar cea de validare stagnează sub 90%. Asta sugerează overfitting: modelul a învățat foarte bine datele de antrenare, dar nu generalizează la fel de bine pe date noi.

Matrice de confuzie



Analizând matricile pentru Heart Disease:

Decision Tree și MLP sunt foarte slabe la detectarea clasei "Risk":

- Decision Tree: doar 3 clasificări corecte din 129 de cazuri cu risc real (recall de doar 2.3%)
- MLP: identic, doar 3 clasificări corecte din 129 de cazuri cu risc real
- Ambele ratează masiv persoanele cu risc (126 de false negative), clasificându-le greșit ca "No Risk"

Random Forest este mult mai bun la detectarea riscului:

- 67 clasificări corecte din 129 de cazuri cu risc real (recall de 52%)
- Doar 62 de false negative

Analizând matricile pentru News Popularity:

MLP este cel mai slab algoritm pentru majoritatea claselor:

- Pentru "Moderately Popular": doar 1006 clasificări corecte din ~2401 cazuri (recall de ~42%)
- Pentru "Popular": doar 567 clasificări corecte din ~1074 cazuri (recall de ~53%)
- Pentru "Unpopular": doar 10 clasificări corecte din ~218 cazuri (recall de ~5%)
- Pentru "Viral": doar 175 clasificări corecte din ~437 cazuri (recall de ~40%)

Decision Tree și Random Forest sunt mult mai echilibrate:

- Random Forest excelează la "Moderately Popular" cu 1361 clasificări corecte și este perfect la "Unpopular" cu 218/218
- Decision Tree este foarte bun la "Viral" cu 406/437 clasificări corecte și la "Unpopular" cu 209/218

Tabel comparativ al algoritmilor pentru fiecare set de date:

Dataset	Algorithm	Accuracy	Precision	Recall	F1-Score
Heart Disease	Decision Tree	0.8479	0.7971	0.8479	0.7846
Heart Disease	Random Forest	0.6498	0.7853	0.6498	0.6961
Heart Disease	MLP	0.8479	0.7971	0.8479	0.7846
News Popularity	Decision Tree	0.5698	0.7287	0.5698	0.6222
News Popularity	Random Forest	0.5892	0.8428	0.5892	0.6711
News Popularity	MLP	0.6844	0.6925	0.6844	0.6523

Pentru Heart Disease, Decision Tree și MLP afișează valori identice pentru toate metricile (Accuracy=0.8479, F1=0.7846), ceea ce confirmă comportamentul lor similar observat în matricile de confuzie - ambii algoritmi clasifică aproape totul ca "No Risk", obținând o acuratețe înaltă prin simpla dominanță numerică a acestei clase. Însă această performanță este înselătoare, deoarece ratează masiv detectarea persoanelor cu risc cardiac (doar 3 din 129 cazuri detectate corect). Random Forest, deși

are o acuratețe globală mai mică (0.6498), oferă o abordare mult mai echilibrată și utilă clinic, detectând efectiv persoanele cu risc.

Pentru News Popularity, MLP pare să aibă cea mai bună acuratețe globală (0.6844), dar Random Forest excelează la precizia (0.8428), sugerând că atunci când Random Forest face o predicție, aceasta este de încredere. Metricile agregate nu reflectă însă provocările specifice ale fiecărei clase - de exemplu, dificultatea tuturor algoritmilor în detectarea articolelor "Unpopular" sau "Viral", care sunt clase minoritare dar importante.

Analiza performanța algoritmi:

Din perspectiva acurateței globale, Decision Tree și MLP domină pentru Heart Disease (84.79%), iar MLP pentru News Popularity (68.44%). Această performanță aparent se datorează unei strategii de clasificare conservatoare - algoritmii învață să exploateze dezechilibrul claselor, clasificând majoritatea cazurilor în clasa majoritară. Pentru Heart Disease, aceștia obțin acuratețe înaltă prin clasificarea a aproape tuturor pacienților ca "No Risk", ceea ce este statistic corect dar clinic periculos.

Din perspectiva utilității practice, Random Forest se dovedește superior prin **echilibrul său în detectarea tuturor claselor**. Pentru Heart Disease, Random Forest sacrifică acuratețea globală pentru a detecta efectiv persoanele cu risc cardiac (recall de 52% vs 2.3% pentru ceilalți), ceea ce este esențial într-un context medical. Pentru News Popularity, Random Forest obține cea mai înaltă precizie (84.28%), indicând că predicțiile sale sunt de încredere.

Random Forest excelează prin agregarea mai multor arbori de decizie antrenați pe subseturi diferite ale datelor, reușind să captureze mai bine variabilitatea și complexitatea datelor, evitând overfitting-ul și bias-ul către clasa majoritară. Această robustețe îl face mai potrivit pentru aplicații practice unde detectarea corectă a tuturor claselor este crucială.

****DISCLAIMER**:** utilizare Claude (<https://claude.ai/new>) și ChatGPT (<https://chatgpt.com/>) pentru generarea graficelor, analiza asupra graficelor mai complicate și pentru modul de utilizare al funcțiilor recomandate în cerința temei.

Alte resurse utilizate:

- link-urile puse la dispoziție în documentația temei