” Analiză predictivă și structurală a ecosistemului Github: o abordare Pyspark-ML pentru optimizarea stocării”

Abstract

Această lucrare prezintă o analiză aprofundată a unui set de date extins de metadate github, având ca scop evaluarea potențialului predictiv și a structurii interne a fișierelor. metodologia implementează un cadru scalabil utilizând pyspark ml pe baza caracteristicilor structurale (mărime, adâncime, duplicare și extensie).

Rezultatele cheie demonstrează o separabilitate aproape perfectă a datelor: clasificarea a confirmat o performanță **AUC** de **0.9918** (logistic regression), iar clustering-ul a fost validat cu un **Silhouette Score** de **0.9962**, identificând tipologii distincte de fișiere și evidențiind grupul "foarte duplicat" ca fiind ținta principală pentru reducerea costurilor de stocare prin deduplicare. concluziile oferă o bază solidă pentru decizii de infrastructură și strategii eficiente de gestionare a resurselor.

1. Introducere

Contextul actual al dezvoltării software se bazează pe repository-uri de cod din ce în ce mai mari, crescând exponențial costurile de stocare și mentenanță.

Lucrarea de față propune utilizarea instrumentelor de machine learning (ml) scalabile (pyspark ml) pentru a obține o înțelegere structurală a fișierelor dintr-un ecosistem github, depășind analiza pur cantitativă.

Obiectivele principale sunt: a valida capacitatea caracteristicilor structurale (mărime, adâncime, etc.) de a prezice tipul de fișier și de a grupa fișierele în tipologii acționabile din punct de vedere operațional.

2. Metodologia de analiză exploratorie a datelor

Analiza a fost efectuată pe un set de date conținând metadate despre fișiere (dimensiune, căi, tipuri de fișiere) și a implicat 11 măsurători cheie.

aspect	constatare cheie	implicație
distribuție mărime	outlieri extremi (fișiere gigant) domină volumul total de stocare, deși sunt rare	necesitatea pre-procesării robuste (logaritmi) pentru ml și concentrarea pe fișierele binare.
tipuri de fișiere	fișierele binare ocupă cel mai mare spațiu de stocare (~10% din total fișiere vs. ~90% text)	justifică sarcina de clasificare ca o problemă de separare binară critică.
redundanta structurală	prezența unui număr extrem de mare de copii pentru anumite fișiere (analiza 8).	confirmă necesitatea imediată de implementare a strategiilor de deduplicare a datelor.

3. Fundamentarea machine learning și clasificarea supervizată

Caracteristicile size, copies, depth au fost extrase și normalizate.

Datele au fost asamblate într-o coloană unică de features vectoriale și împărțite în seturi de antrenare (80%) și testare (20%).

4. Aplicarea și evaluarea algoritmilor de clasificare

Obiectivul clasificării a fost acela de a prezice clasa țintă binară (binary: text vs. binar) pe baza caracteristicilor structurale.

4.1. Rezultate comparative și selecția modelului

algoritm	auc (area under roc)	acuratețe	timp de antrenare (secunde)
logistic regression	**0.9918**	**0.9692**	**12.55**
random forest	0.9894	0.9467	22.65
decision tree	0.7137	0.9626	27.24

4.2. Interpretarea rezultatelor

Scorul auc de 0.9918 obținut de logistic regression demonstrează un potențial predictiv excelent și cvasitotal. acest scor, extrem de apropiat de 1.0, sugerează că separarea între fișierele text și cele binare este, în mare măsură, liniară în spațiul vectorial al caracteristicilor.

Modelul optim este logistic regression, deoarece a oferit cea mai înaltă performanță în cel mai scurt timp, fiind eficient și scalabil. scorul redus al arborelui de decizie (0.7137) confirmă că o modelare mai complexă nu era necesară, ci o relație liniară puternică.

5. Aplicarea algoritmilor de clustering

Obiectivul clustering-ului a fost descoperirea tipologiilor naturale de fișiere în setul de date. s-au aplicat k-means și bisecting k-means, evaluate prin silhouette score (măsura coeziunii și separării).

algoritm	timp de antrenare (secunde)	silhouette score
k-means	**7.07**	**0.9962**
bisecting k-means	24.06	0.9914

5.1. selecția modelului de clustering

Scorul silhouette de 0.9962 (k-means) confirmă o separare aproape perfectă a datelor.

K-means este ales ca algoritm optim datorită performanței sale marginal superioare și a eficienței sale de patru ori mai mari în pySpark (7.07 secunde vs. 24.06 secunde).

5.2. Identificarea tipologiilor

Analiza centrelor clusterelor (pentru k=5) a dezvăluit tipologii stabile bazate pe intersecția mărime, duplicare și extensie:

tipologie	caracteristici cheie	implicații operaționale
foarte duplicat	size ≈25 kb, copies ≈72	risc de redundanță maximă. este reprezentat de fișiere boilerplate (license, .gitignore, config) și necesită deduplicare imediată.
gigant, unic	size ≈73 mb, copies ≈1.0	risc de volum pur. sunt fișiere media sau arhive unice care necesită alocarea către soluții de stocare optimă.
mare, moderat duplicat	size ≈38 mb, copies ≈6.7	risc combinat. necesită inspecție și optimizare/compresie.

6. Discuții

Rezultatele validează ipoteza că atributele structurale ale fișierelor dețin o capacitate informațională extrem de mare. Faptul că atât clasificarea, cât și clustering-ul au obținut scoruri aproape de perfecțiune, AUC > 0.99, Silhouette > 0.99, sugerează că setul de date este liniar separabil și extrem de bine structurat.

7. Optimizarea stocării: traducerea ML în decizii de infrastructură

Optimizarea stocării nu este un algoritm separat, ci este concluzia operațională directă și beneficiul economic care rezultă din analiza de clustering. Segmentarea realizată de k-means permite implementarea următoarelor strategii de reducere a costurilor și îmbunătățire a eficienței:

1. Strategia de deduplicare (reducerea redundanței):

a) acțiune: se implementează un proces automatizat de deduplicare (eliminarea copiilor multiple, păstrând o singură referință) pe grupul "foarte duplicat" (clusterul 0).

b) rezultat: economie de costuri masivă prin recuperarea spațiului irosit de cele approx 72 de copii per fișier.

2. Alocarea resurselor (reducerea costului/gb):

a) acțiune: fișierele din clusterul "gigant, unic" sunt mutate pe soluții de stocare mai ieftine, numite cold storage (stocare rece/arhivare), deoarece nu sunt accesate frecvent.

b) rezultat: optimizarea costului: reducerea costului de stocare per gigabyte, fără a compromite disponibilitatea datelor critice.

3. Monitorizare proactivă: modelele ml pot fi integrate într-un sistem de monitorizare live pentru a clasifica și a aloca automat fișierele nou adăugate în cel mai potrivit tip de stocare, asigurând o gestionare proactivă.

8. Concluzie generală

Această analiză a demonstrat cu succes utilitatea aplicării algoritmilor de machine learning, scalabili, asupra metadatelor github. rezultatele confirmă capacitatea predictivă excepțională a caracteristicilor structurale în clasificarea fișierelor și existența unor tipologii clare care pot fi exploatate pentru optimizarea stocării.

Selecția modelului logistic regression și a algoritmului k-means oferă soluțiile cele mai eficiente și performante pentru acest ecosistem de date.

9. Contribuții și direcții viitoare

Contribuție: a fost oferit un model robust (logistic regression) pentru predicția naturii fișierului, util pentru sistemele de securitate sau analiză statică, și o metodologie bazată pe clustering pentru optimizarea economică a infrastructurii de stocare.

Cercetare viitoare: se recomandă explorarea algoritmilor de anomalii (ex: isolation forest) pe caracteristicile structurale pentru a identifica fișierele care nu se încadrează în tiparele normale, semnalând potențiale probleme de securitate sau gestionare defectuoasă.