Progetto per il corso di Reti Geografiche

Analisi delle Top-Repo di Github

Riproduzione e approfondimento di un'analisi svolta nel 2017:

"Analyzing 🔝 20k Github Repositories"

Partecipanti:

Antonio Giulio: 0522500732

Michele Delli Paoli: 0522500797

Sommario

Introduzione

Ogni sviluppatore che si rispetti ha utilizzato almeno una volta nella vita GitHub, una delle piattaforme di software versioning e hosting più famose al mondo.
GitHub ospita più di sette milioni di progetti open source, organizzati in repositories, a cui partecipano un numero ancora maggiore di persone, ma vi siete mai chiesti quali sono i progetti più famosi e popolari sulla piattaforma?
Cosa potrebbero dirci riguardo a trends nascenti nello sviluppo software o su GitHub stesso?!
Questo lavoro si propone in primis di riprodurre un'analisi svolta nel 2017 da Siddharth Kothari sulle repositories più popolari di GitHub chiamata "Analyzing 🔝 20k Github Repositories", al fine di ottenere informazioni utili riguardo l'evoluzione, nel corso di due anni (dal 2017 al 2019), dell' Open Source Software (OSS).
In più è stata svolta un'ulteriore analisi sugli utenti che hanno contribuito allo sviluppo di tali progetti, in particolare sul gender, informazione non fornita dalla piattaforma, e sulla distribuzione di questi ultimi tra le repositories.
Inoltre, per rendere riproducibile l'analisi e mantenere il lavoro aggiornato è stata sviluppata una semplice Web-App disponibile a questo indirizzo.
Il codice è disponibile sulla repository GitHub a questo indirizzo.

Analisi del 2017

Lo sviluppatore Americano Siddharth Kothari ha pubblicato "Analyzing 🔝 20k Github Repositories" il 24 Agosto del 2017.
Il lavoro di Siddharth è stato incentrato sull'analisi di 29,577 repositories presenti su Github etichettate come famose.
Ma con quale criterio affermiamo che una repo è "famosa"?
È stato scelto da Siddharth come criterio di popolarità il *numero di Stars* (concetto analogo ai Like/Mi Piace sui social media), precisamente il suo dataset è formato da tutti i progetti che hanno un numero di Stars maggiore o uguale a 500.
Le analisi svolte nel 2017 sono:

1. **Distribuzione delle Stars**
   Come prima cosa sono stati identificati dei range di popolarità al fine di esaminare la distribuzione delle repositories in base al numero di Stars, da 500 fino ad arrivare a 300,000 Stars.
   E' stata ottenuta, come ci si aspettava, una Long Tail distribution, ossia il 90% delle repo avevano meno di 5,000 Stars e la percentuale restante si distribuiva sui range di fascia superiore.

2. **Distribuzione delle Fork**

   Il numero di Stars rappresenta la popolarità ma non sempre corrisponde al valore intrinseco del software. Le Forks d'altra parte indicano quanti membri della community hanno scaricato una copia personale di una repo al fine di utilizzare quel software per i loro scopi o di contribuire al suo sviluppo. Ancora una volta l'analisi è stata organizzata in range di Forks a cui è stato associato il numero di repo rientranti nel range.

3. **Linguaggi di Programmazione**

   Senza dubbio è interessante constatare quali sono i linguaggi di programmazione più utilizzati nei Progetti presi in esame. L'autore ha analizzato i linguaggi più utilizzati di tutti i tempi e quelli più utilizzati nelle repositories attive nell'ultimo mese (2017).

4. **Repositories per anno di creazione**

   Ai progetti è associato anche l'anno di creazione, informazione che ci permette quindi di andare ad analizzare l'andamento della popolarità del sito stesso negli anni, precisamente dal 2008 (anno di creazione della piattaforma stessa) e di osservare quante delle repository più popolari siano state create in uno specifico anno.

5. **Topics**

   Nel Gennaio del 2017 GitHub ha introdotto i *Topics*, un nuovo modo di taggare e scoprire le repositories. L'autore ha svolto l'analisi dei topic per capire quali fossero quelli più popolari del 2017 e quali quelli di tendenza (trending) dell'ultimo semestre.

Analisi del 2019

Il primo passo da compiere per riprodurre il lavoro di Siddharth Kothari è stato, dunque, quello di avere a disposizione un dataset di repositories il cui numero di Stars fosse superiore a 500.
Il numero attuale dei progetti che rispettano questo criterio è **43,405**,
a confronto con i 29,577 del 2017. Quindi, in circa 2 anni, il numero di repositories ha registrato un aumento del **46,8%** dovuto a vecchi progetti che hanno guadagnato popolarità e a numerosi nuovi che sono stati creati e sono diventati gettonati.
In seguito, sono state riprodotte tutte le analisi effettuate nel 2017, esposte nella sezione "**Risultati**".
In più, è stata aggiornata la lista dei Trending Topic e poi svolta un'ulteriore analisi sugli UNISA Topic, ossia gli argomenti più rinomati (tra linguaggi di programmazione e tools) dei corsi di laurea Magistrale in Informatica offerti dalla nostra università.

Il lavoro del 2017 si limita esclusivamente all'analisi delle repositories, per ampliare la ricerca di S. Kothari è stato deciso di ispezionare ed analizzare anche i **Contributors**, ossia gli utenti di GitHub che hanno contribuito con vari *commit* allo sviluppo dei softwares. È stato ottenuto, dopo attività di scraping e data cleaning, un dataset di **22,500** nomi, precisamente i nomi dei contributors delle prime 15,000 repositories (circa) più popolari.

Su questo dataset sono state svolte due differenti analisi:

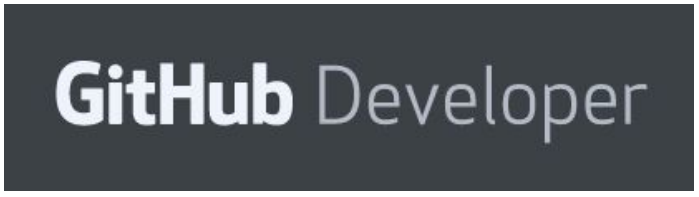
1. **Gender Detection**
   Purtroppo GitHub non fornisce nessuna informazione riguardo il sesso dei propri utenti, quindi ci si è posto l'obiettivo di scoprirlo con l'interessante proposito di ricavare la percentuale di donne presenti tra gli sviluppatori.

2. **Contributors Distribution**
   Prendendo in esame un campione del nostro dataset, formato dalle prime 22,730 repo più popolari, è stato ricavato, per ognuna di esse, il numero di contributors. Successivamente, è stata organizzata l'analisi in range di Stars, ciascuno dei quali formato da 2000 repo, ed è stato calcolato il numero medio dei Contributors delle repositories appartenenti a ciascun range e la distribuzione dei Contributors delle repositories in base ai 7 linguaggi di programmazione più famosi sulla piattaforma.

Di seguito viene descritta la tecnica con la quale sono stati raccolti i dati da GitHub.

Raccolta dei Dati

La sezione di GitHub dedicata agli sviluppatori:



mette a disposizione delle API per raccogliere dati relativi alle repositories e agli utenti della piattaforma: GitHub Search API.

GitHub Search API

L'API ***Search*** consente di cercare item specifici. Per esempio si può cercare un utente, una repository, un file e così via.
Per effettuare la ricerca si utilizzano delle *query* che seguono una determinata sintassi. Inoltre, viene offerta la possibilità di utilizzare questa API tramite richieste HTTP all'endpoint

`https://api.github.com`

In particolare, per riprodurre l'analisi del 2017 è stato utilizzato l'endpoint

`https://api.github.com/search/repositories`

specifico per la ricerca di repositories.
All'endpoint vanno aggiunti i parametri della ricerca, ad esempio per ottenere tutte le repositories che hanno un numero di Stars maggiore a 500 scriviamo:

`https://api.github.com/search/repositories?q=stars%3A>500`

In risposta ad ogni richiesta simile si ottiene un file JSON che riporta tutte le informazioni richieste.

```
total_count:          43406
incomplete_results:   false
▼ items:
  ▼ 0:
      id:             28457823
      node_id:        "MDEwOlJlcG9zaXRvcnkyODQ1NzgyMw=="
      name:           "freeCodeCamp"
      full_name:      "freeCodeCamp/freeCodeCamp"
      private:        false
    ▶ owner:          {…}
      html_url:       "https://github.com/freeCodeCamp/freeCodeCamp"
    ▶ description:    "The https://www.freeCode…with millions of people."
      fork:           false
    ▶ url:            "https://api.github.com/r…reeCodeCamp/freeCodeCamp"
    ▶ forks_url:      "https://api.github.com/r…eCamp/freeCodeCamp/forks"
    ▶ keys_url:       "https://api.github.com/r…eeCodeCamp/keys{/key_id}"
    ▶ collaborators_url: "https://api.github.com/r…aborators{/collaborator}"
    ▶ teams_url:      "https://api.github.com/r…eCamp/freeCodeCamp/teams"
    ▶ hooks_url:      "https://api.github.com/r…eCamp/freeCodeCamp/hooks"
    ▶ issue_events_url: "https://api.github.com/r…p/issues/events{/number}"
    ▶ events_url:     "https://api.github.com/r…Camp/freeCodeCamp/events"
    ▶ assignees_url:  "https://api.github.com/r…odeCamp/assignees{/user}"
    ▶ branches_url:   "https://api.github.com/r…deCamp/branches{/branch}"
    ▶ tags_url:       "https://api.github.com/r…deCamp/freeCodeCamp/tags"
    ▶ blobs_url:      "https://api.github.com/r…CodeCamp/git/blobs{/sha}"
    ▶ git_tags_url:   "https://api.github.com/r…eCodeCamp/git/tags{/sha}"
    ▶ git_refs_url:   "https://api.github.com/r…eCodeCamp/git/refs{/sha}"
    ▶ trees_url:      "https://api.github.com/r…CodeCamp/git/trees{/sha}"
    ▶ statuses_url:   "https://api.github.com/r…eCodeCamp/statuses/{sha}"
    ▶ languages_url:  "https://api.github.com/r…p/freeCodeCamp/languages"
    ▶ stargazers_url: "https://api.github.com/r…/freeCodeCamp/stargazers"
    ▶ contributors_url: "https://api.github.com/r…reeCodeCamp/contributors"
    ▶ subscribers_url: "https://api.github.com/r…freeCodeCamp/subscribers"
    ▶ subscription_url: "https://api.github.com/r…reeCodeCamp/subscription"
    ▶ commits_url:    "https://api.github.com/r…eeCodeCamp/commits{/sha}"
```

Limitazioni di GitHub Search API

Durante la fase di raccolta dei dati sono state riscontrate diverse difficoltà dovute ai limiti imposti dalla Search API.
La limitazione più significativa viene posta sul numero di richieste effettuabili all'endpoint *../search*, infatti, senza autenticazione, è possibile fare solo **10** richieste al minuto. Un limite troppo proibitivo per il lavoro da svolgere.
Tuttavia, effettuando richieste con autenticazione la soglia sale a 5000 all'ora. Per fare ciò bisogna allegare negli Header di ogni richiesta HTTP il Nome Utente del proprio profilo GitHub e un ***Personal Access Token*** generato appositamente per i propri scopi nella sezione "*..Settings/DeveloperSettings/PersonalAccessToken"* dell'area utente del sito.
Tutte le richieste vengono svolte in maniera automatica dai metodi della classe Java ***RequestGenerator*** sviluppata appositamente, la quale offre metodi per effettuare scraping dalla piattaforma e ricerche specifiche.

Altre limitazioni riguardano i file JSON ottenuti dalle richieste.
I risultati vengono organizzati in pagine che mostrano un massimo di 100 elementi (repositories) fino ad un limite di 10 pagine (quindi un totale di 1000 elementi per richiesta). Per ottenere pagine diverse bisogna fare richieste diverse, per gestire le pagine e gli elementi per pagina bisogna aggiungere ai propri target di ricerca i valori di "*page" e "per_page"*.
Per aggirare il limite dei 1000 elementi per ogni target basta organizzare la ricerca in range;
ad esempio "*stars:500..1000 - stars:1000..1500 - ...".*

Gender detection

La rappresentazione in formato JSON di ogni repository offre un campo chiamato "*contributors_url"* che fornisce il link ad un altro file JSON in cui vi è la lista dei Contributors che, a sua volta, offre il campo "*url*" che porta ad un ulteriore file dedicato alle informazioni di un singolo utente. Da notare dunque, che per ottenere un singolo nome bisogna fare almeno due richieste, il tutto si traduce in una computazione onerosa, soprattutto in termini di complessità. Nonostante ciò, abbiamo ricavato circa 33,000 nomi tra i quali però erano presenti diverse istanze "*null"* e numerosissime istanze che non corrispondevano a nomi reali. Ripulendo il dataset il numero di nomi si è ridotto a circa 22,500 istanze.
Successivamente sono state utilizzate due API esterne, GenderAPI e Genderize, per ottenere il gender di ogni nome del dataset ottenuto, in relazione ad una determinata soglia di attendibilità dei risultati.

Limiti e soluzioni

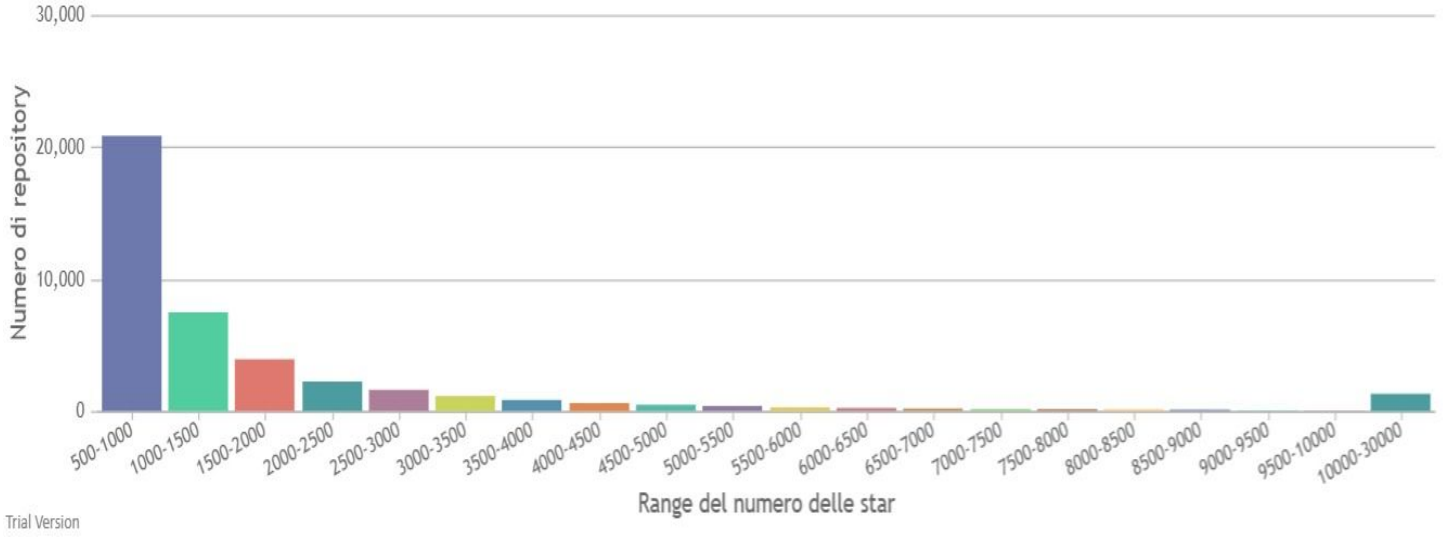
Anche nell'analisi del genere degli utenti sono state riscontrate diverse limitazioni.
Con Genderize è possibile effettuare 1000 richieste al giorno per indirizzo IP, invece per GenderAPI 500 al mese per account.
Per aggirare le limitazioni e quindi riuscire ad analizzare 22,500 nomi, sono stati creati diversi account per quanto riguarda GenderAPI.
Invece, per sfruttare al meglio Genderize, è stato fatto uso di tre diverse Virtual Private Network (VPN) (ProtonVPN, WindScribe e Hide.me) in maniera tale da cambiare indirizzo IP e poter effettuare circa 6000 richieste al giorno.

Risultati

Di seguito sono illustrati i risultati delle diverse analisi attraverso dei grafici che permettono di fare anche il confronto con i risultati ottenuti nel 2017.

Distribuzione delle Stars

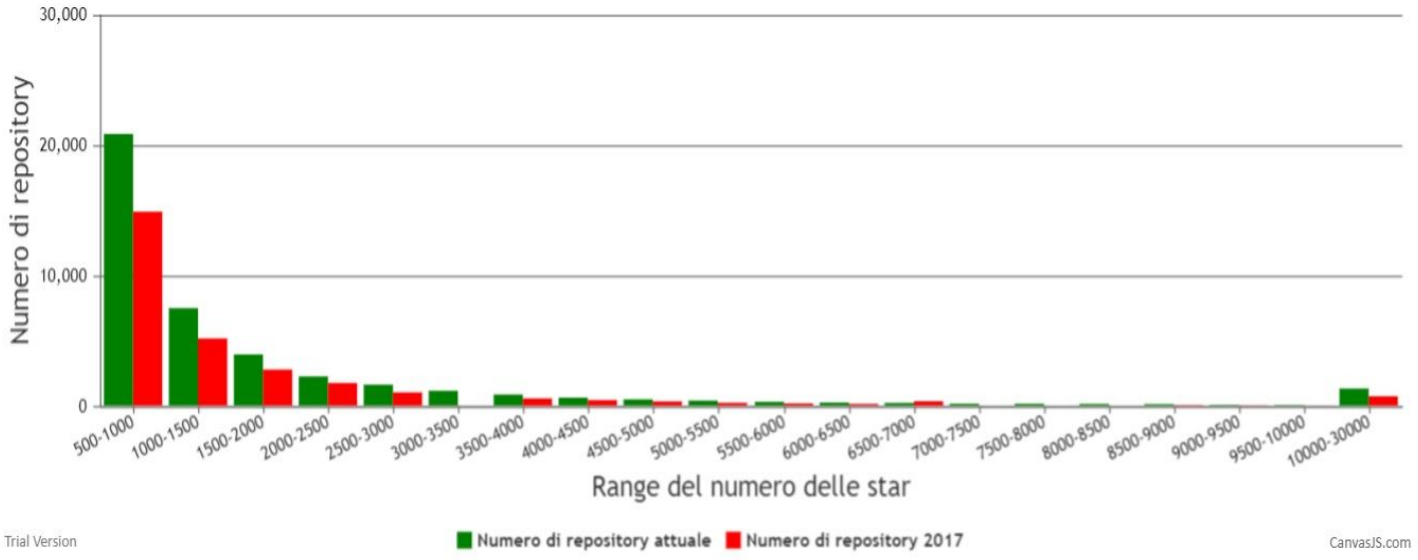
Com'è stato precedentemente spiegato, il numero di Stars di una repository può essere considerato un indice di gradimento da parte degli utenti della piattaforma.
Quest'analisi osserva la distribuzione delle repositories in base al numero di Stars, stabilendo dei **range** di **500** unità alla volta.



Nel grafico a barre di cui sopra si nota, come ci si aspettava, una **Long Tail** distribution.
Il **90%** delle repositories ha **meno di 5,000 Stars,** mentre per i range di fascia superiore si ha una distribuzione più o meno uniforme delle repositories.

Confronto con l'analisi del 2017

Il grafico a barre affianca i risultati del 2019 (in verde) a quelli del 2017 (in rosso). Possiamo constatare che per tutti i range di Stars c'è stato un incremento del numero di repositories nell'arco di 2 anni.



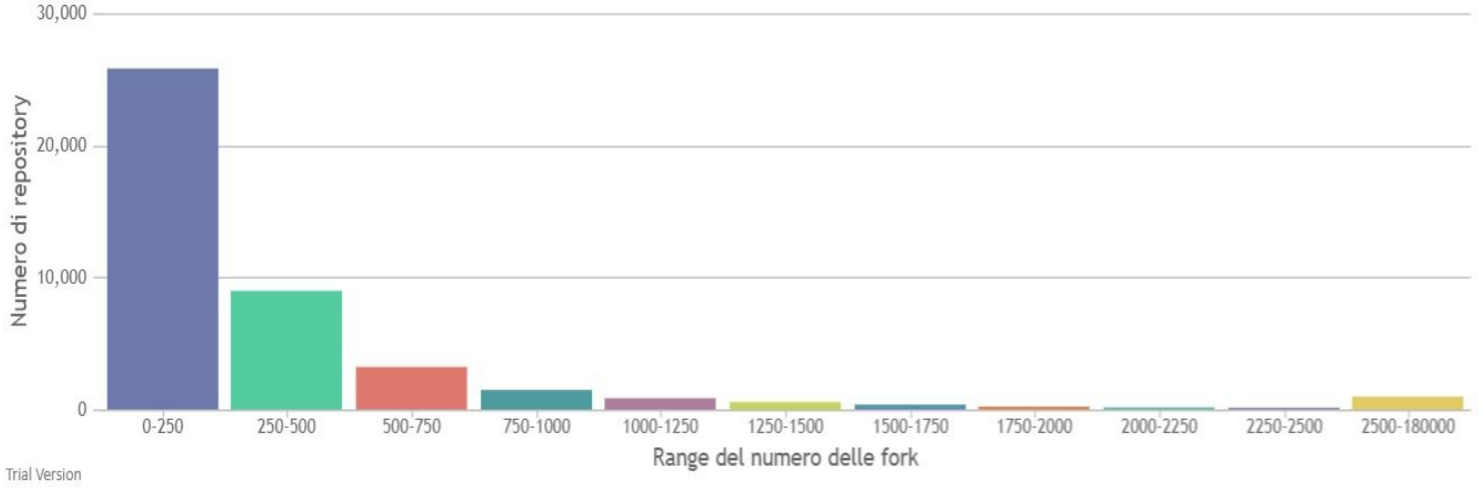
L'unico range che ha riscontrato una **diminuzione** del **33.8%** nel numero di repositories è quello da 6,500 a 7,000 Stars.

Di seguito è esposta una tabella con tutti i dati inerenti all'analisi.

Range di star	Numero di repository attuali	Numero di repository nel 2017	Differenza in percentuale
500-1000	20924	14954	▲ 39,9 %
1000-1500	7543	5217	▲ 44,6 %
1500-2000	3988	2825	▲ 41,2 %
2000-2500	2302	1800	▲ 27,9 %
2500-3000	1671	1073	▲ 55,7 %
3000-3500	1202	Dato non disponibile	0 %
3500-4000	908	608	▲ 49,3 %
4000-4500	668	480	▲ 39,2 %
4500-5000	536	380	▲ 41,1 %
5000-5500	445	275	▲ 61,8 %
5500-6000	350	226	▲ 54,9 %
6000-6500	294	187	▲ 57,2 %
6500-7000	263	397	▼ -33,8 %
7000-7500	202	38	▲ 431,6 %
7500-8000	203	44	▲ 361,4 %
8000-8500	185	44	▲ 320,5 %
8500-9000	170	98	▲ 73,5 %
9000-9500	125	84	▲ 48,8 %
9500-10000	104	68	▲ 52,9 %
10000-30000	1372	779	▲ 76,1 %

Distribuzione delle Forks

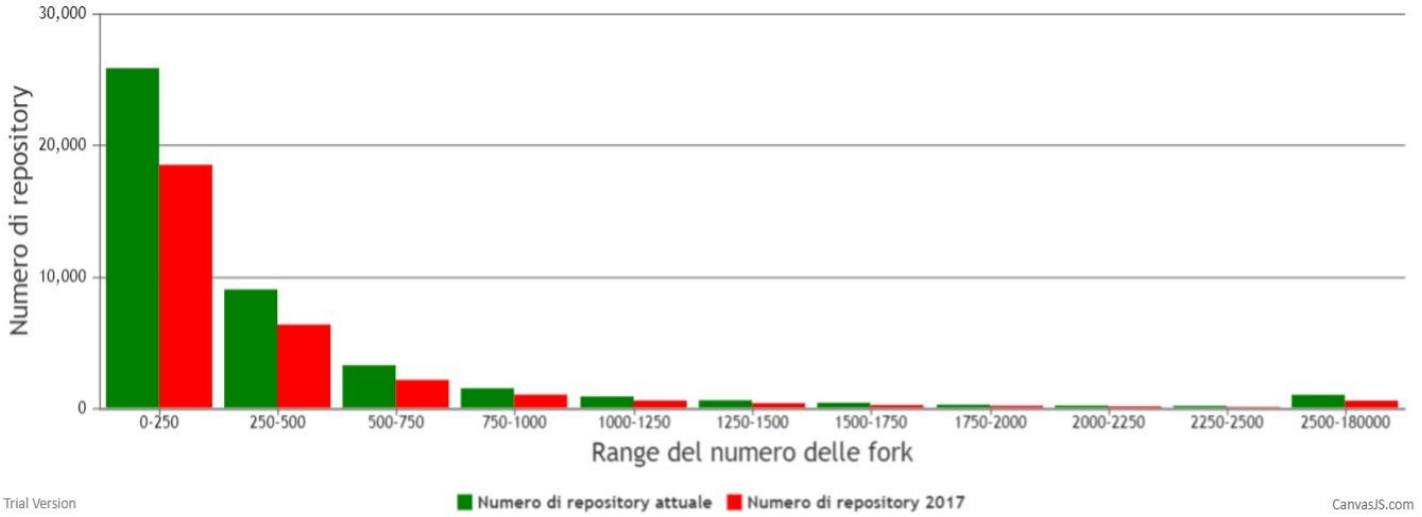
Il numero di **Forks** associato ad un progetto sulla piattaforma rappresenta il numero di utenti (loggati e non) che ne mantengono una copia personale in locale, al fine di utilizzare quel software o eventualmente di contribuire al suo sviluppo committando delle modifiche.
Nel seguente grafico a barre viene mostrata la distribuzione delle repositories più popolari in base al numero di Forks, suddivisi in **range** di **250** unità alla volta.
Osserviamo, anche stavolta, una **Long Tail** distribution, con il **60%** delle repositories che ha meno di 250 Fork.



Da questi risultati si evince, dunque, che i progetti che hanno più Stars hanno un maggior numero di Forks, da qui si intuisce che vi lavorano probabilmente molte più persone come vedremo in seguito.

Confronto con l'analisi del 2017

Il seguente grafico mostra i risultati del 2019 confrontati con quelli del 2017.
Si evince che per tutti i range c'è stato un incremento nel numero di repositories nell'arco temporale considerato.



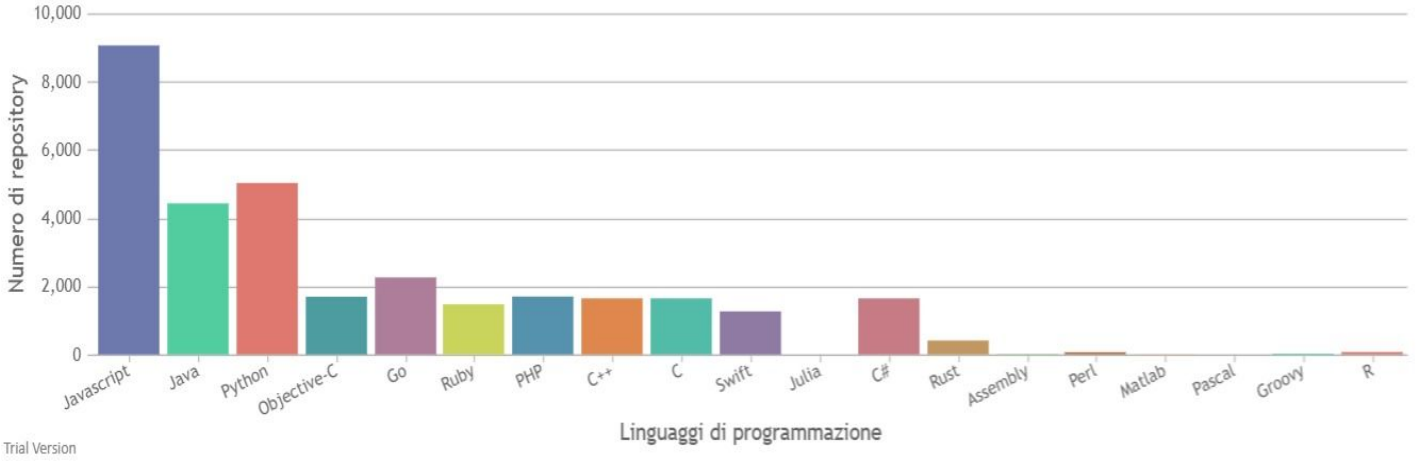
Il range che ha riscontrato l'incremento più significativo (del **75.9%**) è quello che raccoglie le repositories che hanno più di 2500 Forks.

La seguente tabella mostra tutti i dati inerenti a questa analisi.

Range di fork	Numero di repository attuali	Numero di repository nel 2017	Differenza in percentuale
0-250	25901	18538	▲ 39,7 %
250-500	9050	6379	▲ 41,9 %
500-750	3285	2166	▲ 51,7 %
750-1000	1534	1048	▲ 46,4 %
1000-1250	910	598	▲ 52,2 %
1250-1500	619	395	▲ 56,7 %
1500-1750	428	254	▲ 68,5 %
1750-2000	283	204	▲ 38,7 %
2000-2250	214	156	▲ 37,2 %
2250-2500	193	117	▲ 65 %
2500-180000	1034	588	▲ 75,9 %

Top Languages di tutti i tempi

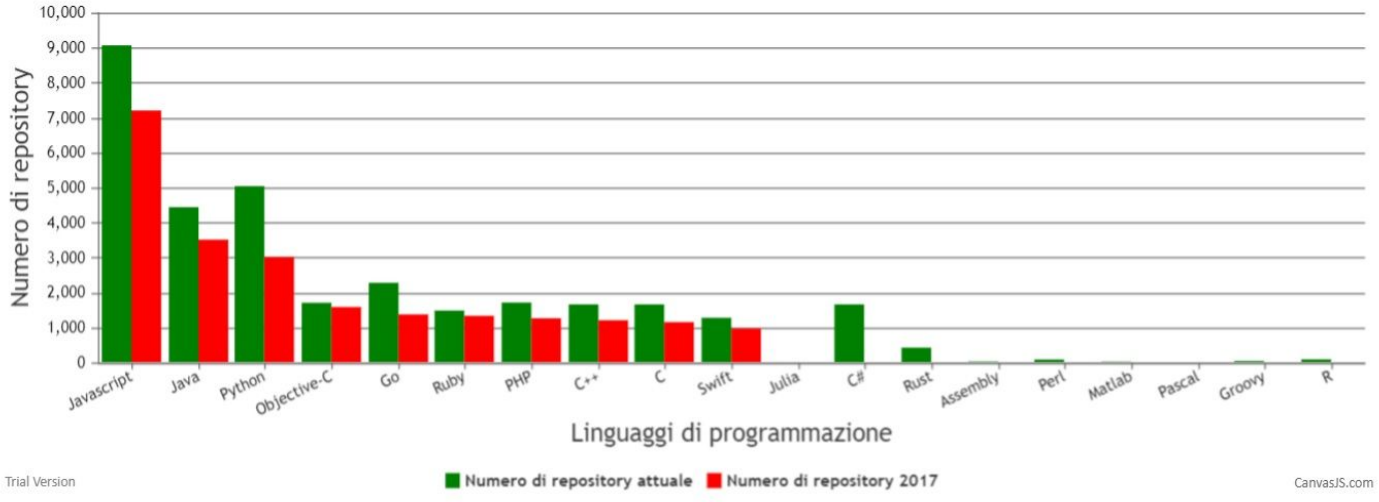
Questa sezione dell'analisi è dedicata alla distribuzione delle repositories in base ai linguaggi di programmazione principali. Sulla base di questi dati è possibile capire quali siano i linguaggi più utilizzati dai developers dall'apertura della piattaforma.



Dal grafico si evince che **JavaScript** è il linguaggio di programmazione più utilizzato tra le repositories più popolari, seguito da **Python** e da **Java.**

Confronto con l'analisi del 2017

Anche in questo caso constatiamo che il numero di repositories è aumentato per tutti i linguaggi di programmazione.



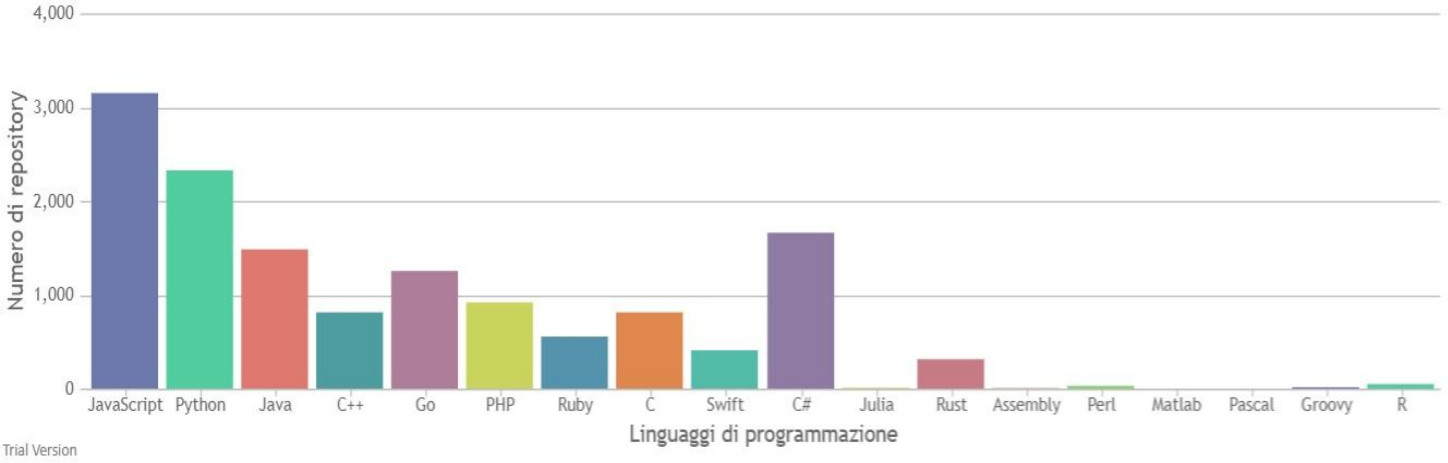
L'aumento più significativo si è verificato per il linguaggio **Python,** con un incremento del **67.2%** rispetto al 2017. Questo è probabilmente dovuto allo sviluppo e al sempre crescente interesse per il Machine Learning, come vedremo successivamente nell'analisi dei topic.

Nella seguente tabella sono esposti i dati inerenti all'analisi.

Linguaggio di programmazione	Numero di repository attuali	Numero di repository nel 2017	Differenza in percentuale
JavaScript	9087	7221	▲ 25,8 %
Java	4455	3523	▲ 26,5 %
Python	5054	3023	▲ 67,2 %
Objective-C	1716	1595	▲ 7,6 %
Go	2289	1383	▲ 65,5 %
Ruby	1494	1343	▲ 11,2 %
PHP	1723	1273	▲ 35,3 %
C++	1670	1217	▲ 37,2 %
C	1670	1162	▲ 43,7 %
Swift	1287	980	▲ 31,3 %
Julia	22	Dato non disponibile	0 %
C#	1561	Dato non disponibile	0 %
Rust	433	Dato non disponibile	0 %
Assembly	38	Dato non disponibile	0 %
Perl	92	Dato non disponibile	0 %
Matlab	34	Dato non disponibile	0 %
Pascal	13	Dato non disponibile	0 %
Groovy	52	Dato non disponibile	0 %
R	96	Dato non disponibile	0 %

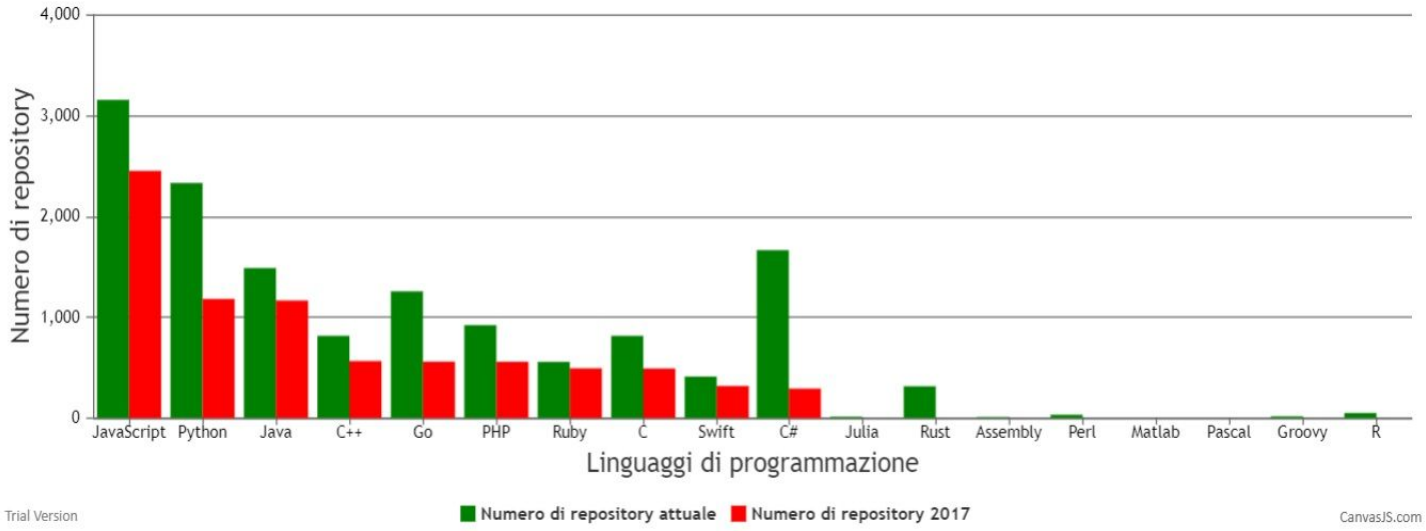
Top Languages dell'ultimo mese

A differenza dell'analisi precedente consideriamo esclusivamente le repositories attive nell'ultimo mese, cioè quelle che hanno ricevuto un *commit* da parte di uno o più contributors negli ultimi trenta giorni, e le distribuiamo in base ai linguaggi di programmazione.
Dunque, i risultati mostrano, come potevamo aspettarci, che anche nell'ultimo mese (Dicembre 2019) **JavaScript** è il linguaggio più utilizzato, seguito da **Python,** mentre questa volta al terzo posto troviamo **C#.**



Confronto con l’analisi del 2017

Ancora una volta confrontiamo i risultati con la precedente analisi che mostra la distribuzione per il mese di Luglio 2017.
Per tutti i linguaggi c’è stato un incremento del numero di repositories.
Notiamo un incredibile incremento del **460%** da parte di **C#** che evidentemente ha acquistato molta più rilevanza in quest’ultimo mese (Dicembre 2019) rispetto a 2 anni fa (Luglio 2017).

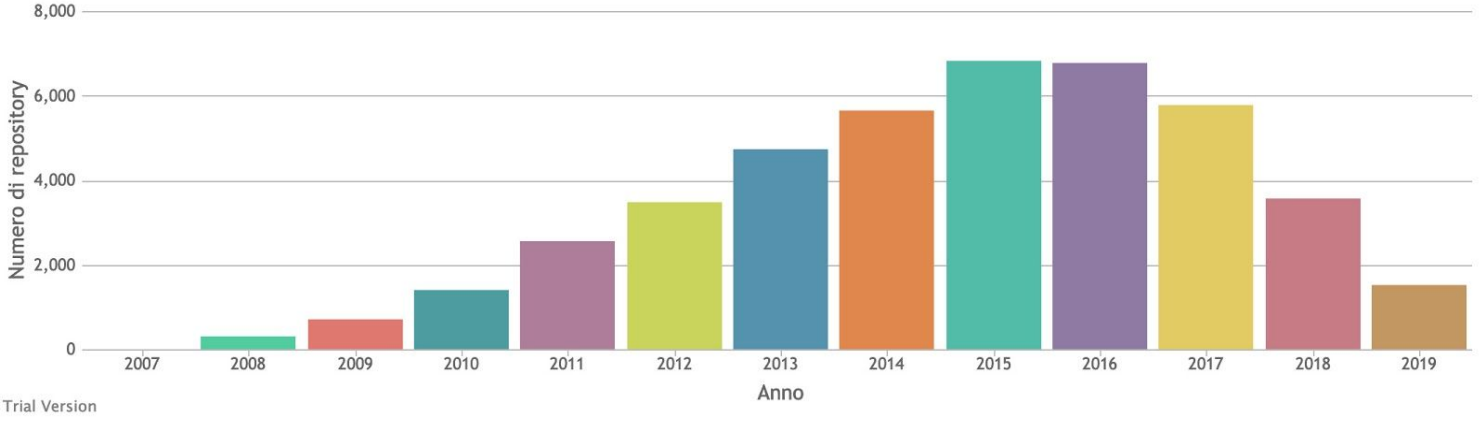


Nella tabella vengono mostrati i dati dell’analisi.

Linguaggio di programmazione	Numero di repository attive nell'ultimo mese (dal 10-11-2019 al 10-12-2019)	Numero di repository attive dal 15-07-17 al 15-08-17	Differenza in percentuale
JavaScript	3160	2456	▲ 28,7 %
Python	2336	1187	▲ 96,8 %
Java	1493	1171	▲ 27,5 %
C++	821	571	▲ 43,8 %
Go	1262	565	▲ 123,4 %
PHP	927	564	▲ 64,4 %
Ruby	563	499	▲ 12,8 %
C	821	497	▲ 65,2 %
Swift	417	324	▲ 28,7 %
C#	1670	298	▲ 460,4 %
Julia	19	Dato non disponibile	0 %
Rust	321	Dato non disponibile	0 %
Assembly	16	Dato non disponibile	0 %
Perl	39	Dato non disponibile	0 %
Matlab	4	Dato non disponibile	0 %
Pascal	6	Dato non disponibile	0 %
Groovy	23	Dato non disponibile	0 %
R	56	Dato non disponibile	0 %

Distribuzione delle Repositories in base all'anno di creazione

Tramite questo tipo di analisi possiamo renderci conto dell'andamento della popolarità di GitHub stesso e dell'Open Source Software negli anni in quanto vediamo il numero di repositories, sempre fra le più famose, create ogni anno dal 2008 in poi (anno di apertura del sito).
Dunque è evidente, dal grafico a barre, che vi è una notevole ascesa dal 2011 al 2016, per poi riscontrare un calo negli ultimi due anni.
Precisamente il **2015** è l'anno in cui sono state create più repositories (**6840**) che attualmente sono diventate popolari, circa il **15%** di quelle prese ad esame.



La tabella mostra nel dettaglio i risultati dell'analisi.

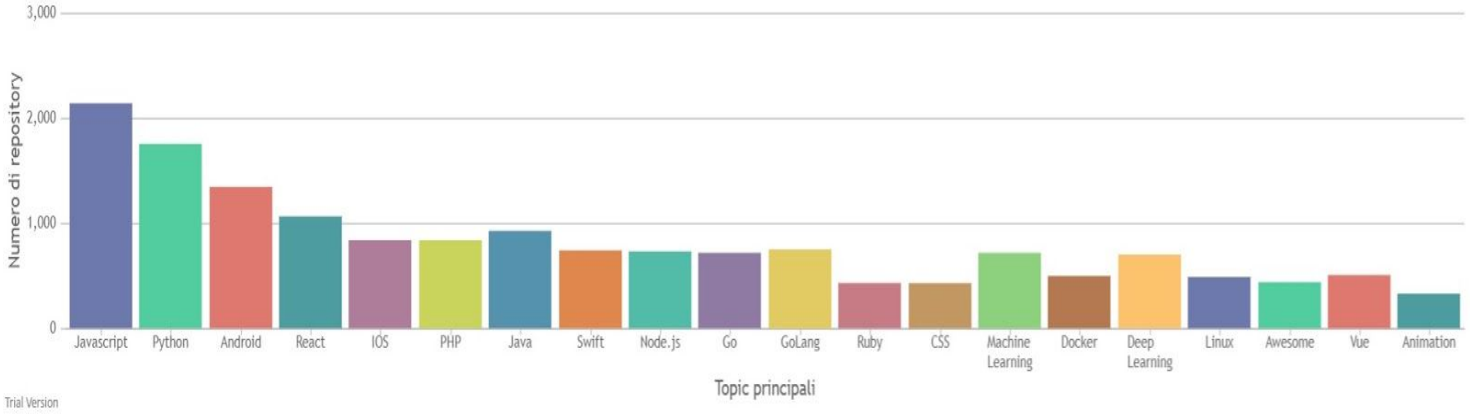
Anno	Numero di repository create	Percentuale rispetto al numero totale di repository
2007	1	0%
2008	319	0%
2009	723	1%
2010	1418	3%
2011	2572	5%
2012	3494	8%
2013	4748	10%
2014	5664	13%
2015	6840	15%
2016	6789	15%
2017	5793	13%
2018	3583	8%
2019	1537	3%

Topics

Com'è stato descritto nel capitolo precedente i Topics, introdotti nel 2017, rappresentano un nuovo modo di etichettare e scoprire repositories, in maniera molto simile agli hashtag sui social media come Instagram o Twitter.

Top Topics

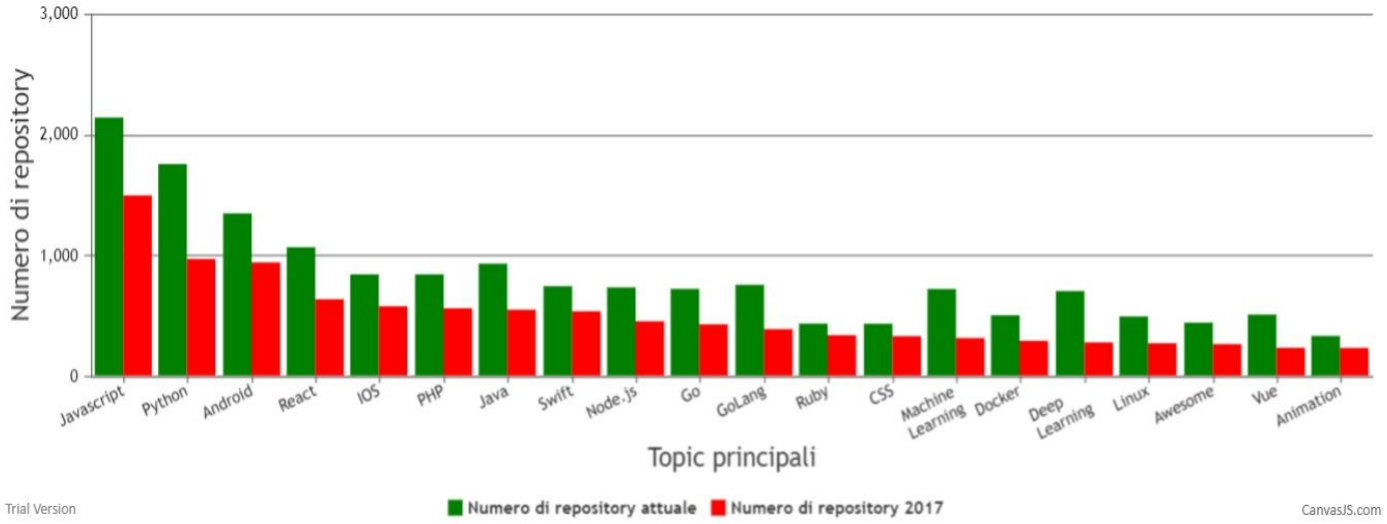
La prima analisi riguardante questo tema va a raccogliere quelli che sono i topic principali della piattaforma al fine di scoprire quali sono quelli più utilizzati.



Osservando dunque la distribuzione delle repositories in base ai Topic notiamo che **JavaScript** è il tag più utilizzato, come ci si aspettava dai risultati delle analisi sui linguaggi. La cosa più interessante da notare è il notevole utilizzo dei Topic **Android** e **IOS** che non sono disponibili come linguaggi su GitHub ma solo come Topics.

Confronto con i risultati del 2017

Nel seguente grafico a barre vediamo il confronto con i risultati ottenuti da S. Kothari.
Si evince che tutti i topic hanno registrato un incremento nel numero di repositories, dunque, possiamo dedurre che l'utilizzo dei Topics è stato apprezzato e sempre più utilizzato dagli utenti GitHub.



Nella seguente tabella sono raccolti i dati relativi all'analisi, dalla quale si evince che **Deep Learning** è il topic che ha registrato l'incremento più significativo nel numero di repository, con il **151%** in più rispetto al 2017.
Al secondo posto troviamo **Machine Learning** con un incremento del **128%** rispetto al 2017.
Questi risultati indicano che in questi due anni le repository che trattano di questi due topic hanno subito un forte sviluppo.

Topic	Numero di repository attuali	Numero di repository nel 2017	Differenza in percentuale
JavaScript	2146	1500	▲ 43,1 %
Python	1760	973	▲ 80,9 %
Android	1351	944	▲ 43,1 %
React	1070	639	▲ 67,4 %
IOS	845	581	▲ 45,4 %
PHP	845	564	▲ 49,8 %
Java	935	552	▲ 69,4 %
Swift	747	539	▲ 38,6 %
Node.js	737	456	▲ 61,6 %
Go	725	431	▲ 68,2 %
GoLang	758	391	▲ 93,9 %
Ruby	437	341	▲ 28,2 %
CSS	436	333	▲ 30,9 %
Machine Learning	724	317	▲ 128,4 %
Docker	506	294	▲ 72,1 %
Deep Learning	707	281	▲ 151,6 %
Linux	496	274	▲ 81 %
Awesome	445	267	▲ 66,7 %
Vue	512	236	▲ 116,9 %
Animation	335	235	▲ 42,6 %

Trending Topics

Successivamente, sono stati analizzati tutti i topic di GitHub (escludendo quelli creati dagli utenti) concentrando la ricerca sulle repositories che sono state **attive negli ultimi sei mesi**, al fine di capire quali sono i Topic di tendenza nell'ultimo periodo. Una volta ottenuti i dati abbiamo utilizzato un grafico a bolle per visualizzare i risultati.

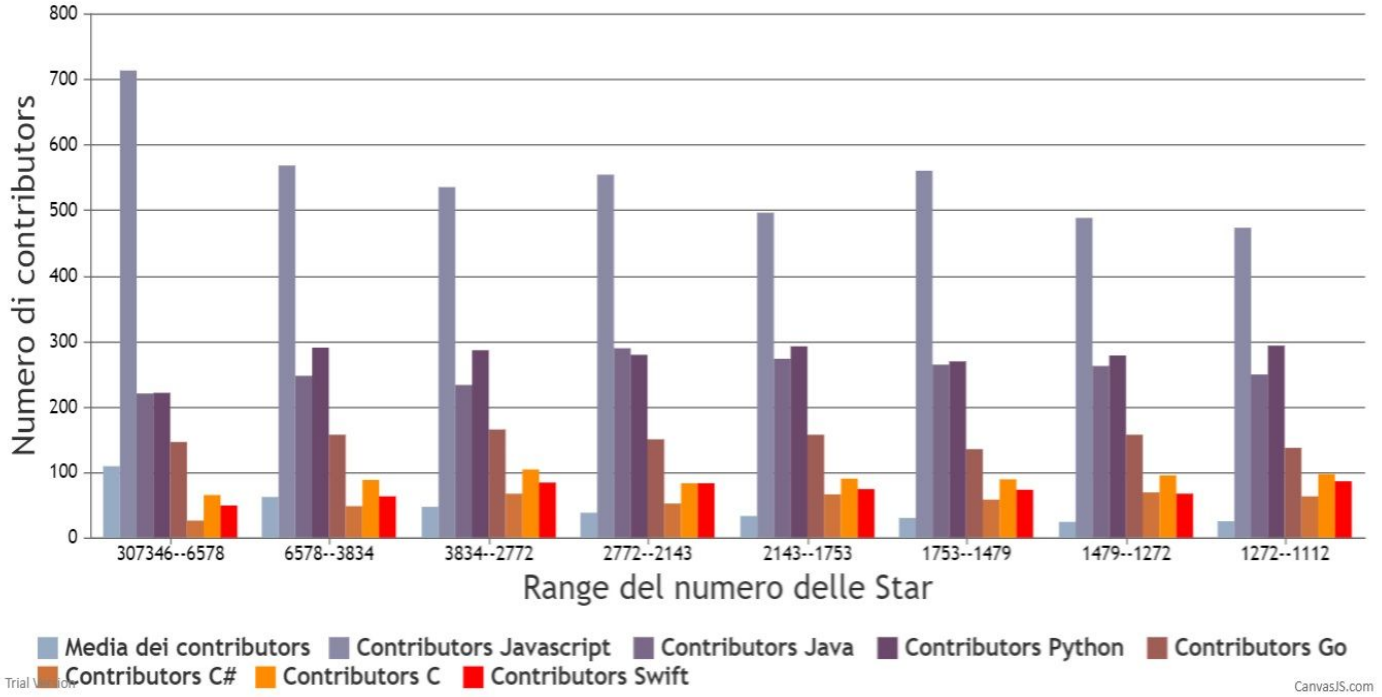
JavaScript dimostra di essere ancora l'argomento più curato sulla piattaforma, infatti ben 1752 repositories lo hanno utilizzato negli ultimi sei mesi. Al secondo posto troviamo **Python** e poi ci sono i linguaggi di programmazione per le applicazioni mobile, **Android** e **IoS**. In generale, osservando le bolle più grandi, ci rendiamo conto di quali sono i Topic più utilizzati.

UNISA Trending Topics

Questa parte del lavoro si concentra sugli UNISA Topics, ossia gli argomenti protagonisti (tra linguaggi di programmazione e tools) dei corsi di Laurea magistrale in Informatica offerti dalla nostra università, al fine di scoprire quali di questi sono argomenti popolari su GitHub.
I risultati sono stati raccolti nel seguente grafico a bolle.
Risalta agli occhi come l'ambito della **Sicurezza** e del **Deep-Learning** siano quelli più seguiti sul sito. Invece, per quanto riguarda i tools si deduce come vincitore la Container Platform **Docker**.

Analisi dei Contributors

I Contributors di una repository sono gli utenti GitHub che effettuano un *commit* a tale progetto, contribuendo quindi al suo sviluppo.
Il dataset di utenti ottenuto, che copre 20,731 repositories, è stato suddiviso in 8 range da 2,000 repositories ciascuno, in base al numero di Stars dei progetti.
Per ciascun range è stato ottenuto il **numero medio di Contributors** per repo, a cui è stato affiancato un prospetto del **numero di Contributors** delle repo **che utilizzano i 7 linguaggi** di programmazione risultati più gettonati nelle precedenti analisi.



Dal grafico di cui sopra si può osservare che per le repositories più famose, quindi con un numero di Stars elevato, il numero medio di Contributors è maggiore di 100. Alcuni progetti in questo range arrivano anche a 400 collaboratori.
Quindi, un numero elevato di utenti lavora alle repositories di GitHub più popolari.

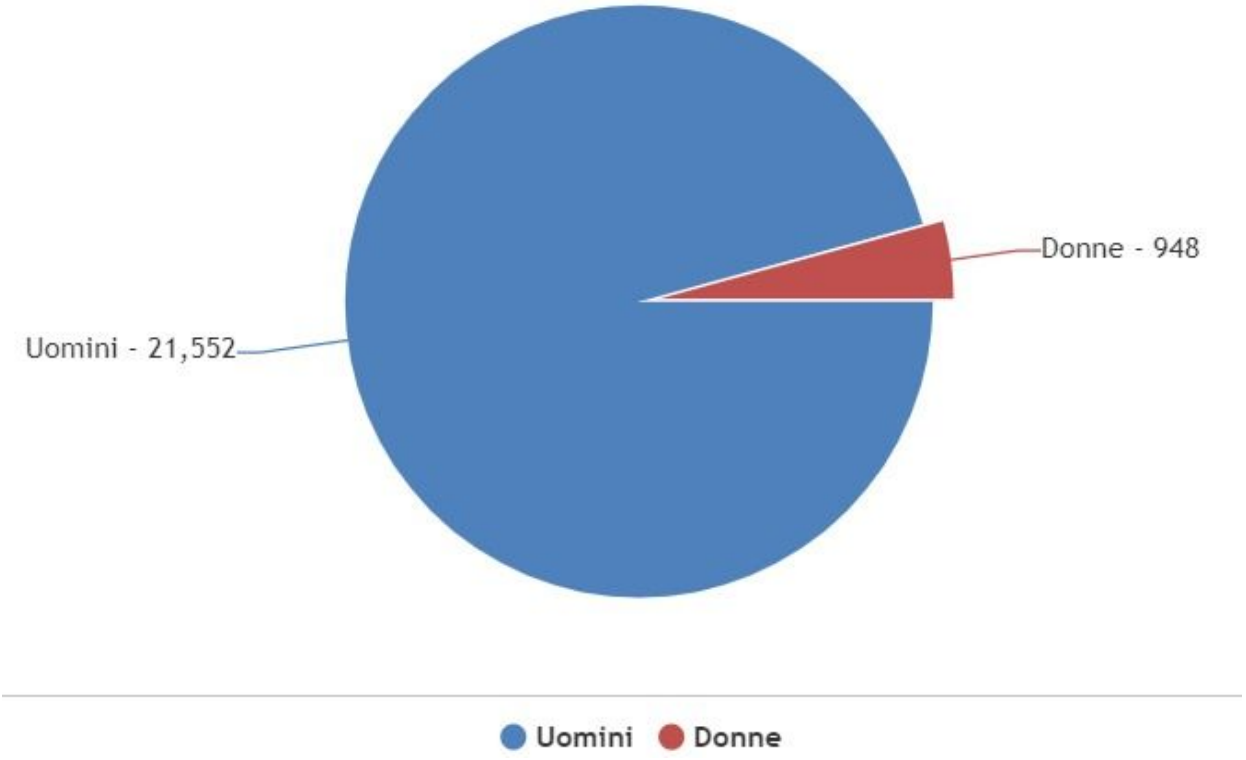
Per quanto riguarda i linguaggi utilizzati da questi Top Developers vediamo un podio composto da JavaScript, Java e Python per tutti i range.

Nel seguente grafico sono riportati, nel dettaglio, i risultati dell'analisi.

Range di Star	**Media dei contributors**	Contributors delle repository che utilizzano **Javascript**	Contributors delle repository che utilizzano **Java**	Contributors delle repository che utilizzano **Python**	Contributors delle repository che utilizzano **Go**	Contributors delle repository che utilizzano **C#**	Contributors delle repository che utilizzano **C**	Contributors delle repository che utilizzano **Swift**
307346--6578	110	714	221	222	147	27	66	50
6578--3834	63	569	248	291	158	49	89	64
3834--2772	48	536	234	287	166	68	105	85
2772--2143	39	555	290	280	151	53	84	84
2143--1753	34	497	274	293	158	67	91	75
1753--1479	31	561	265	270	136	59	90	74
1479--1272	25	489	263	279	158	70	96	68
1272--1112	26	474	250	294	138	64	98	87

Gender Analysis

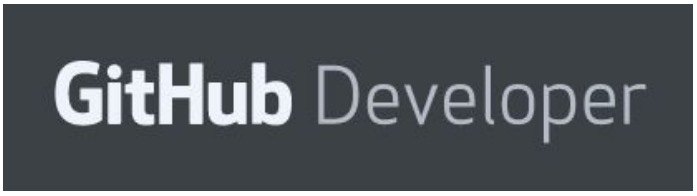
Questa parte conclusiva dell'analisi si concentra sul genere degli utenti di GitHub che hanno contribuito allo sviluppo delle repositories più famose.
Sono stati analizzati 22,500 nomi utilizzando diverse API per la Gender Detection ed è risultato, con un determinato indice di precisione, che sul totale solo 948 utenti sono donne, il **4%.**



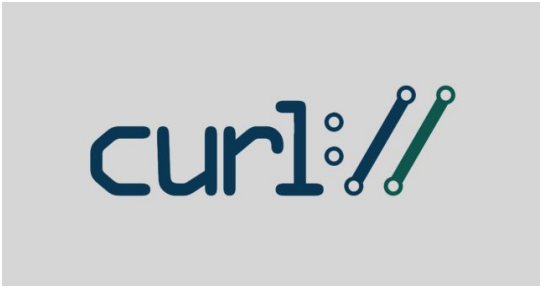
Strumenti utilizzati

Abbiamo utilizzato la IDE Eclipse per sviluppare la Web Application.

Per la raccolta dei dati:



Per fare le richieste HTTP agli endpoint:



Per l'analisi del genere:

Virtual Private Networks:

Per la generazione dei grafici: