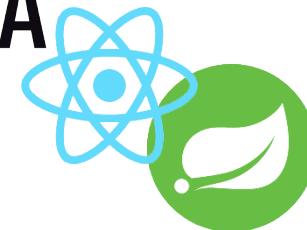




UNIVERSITA' degli STUDI di ROMA
T O R V E R G A T A



Presentazione Finale - Ticketing System

Pelella, Perrone, Pusceddu, Scarlino, Scarpitta, Tranzocchi, Trotta,

Consegna finale

5° Sprint: cosa abbiamo fatto?

Gruppo Operativo

- Deploy dell'intero sistema su Cloud:
 - Kubernetes per Spring e React sulla piattaforma cloud di Google.
 - AWS S3 per lo storage di file.
- Kubernetes è uno strumento open source di orchestrazione e gestione di container per automatizzare il rilascio e la scalabilità dell'applicazione.
- Paginazione lato Backend.
- Ricerca ticket per target e per categoria.



Ticketing system

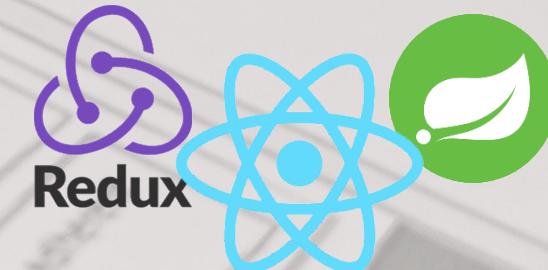
Funzionalità e tecnologie

Riepilogo delle funzionalità del sistema

- Ticket
- Gestione Utenti
- Gestione Target
- Gestione Team
- Login/Logout
- Configuration File
- Machine Learning

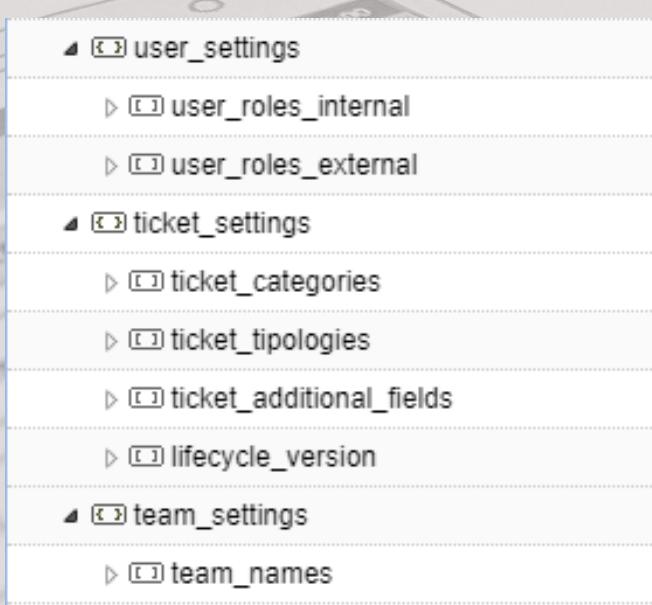
Riepilogo delle tecnologie adottate

- Database: MongoDB
- Backend: Spring
- Frontend: React, Redux



Configuration file

Flessibilità del sistema



- Il configFile è un file in formato Json composto da tre sezioni:
 - User settings, dove sono definiti i ruoli interni ed esterni degli utenti del sistema.
 - Ticket settings, dove sono definiti i possibili campi aggiuntivi dei ticket
 - Team settings, dove sono definiti i tipi di team

Flessibilità ticket

Gestione dei campi aggiuntivi

ticket_tipologies	Array[2]
0	{ 2 fields }
id	1
additional_field	Array[3]
0	{ 5 fields }
id	1
name	name
placeholder	placeholder
regularExp	^[a-zA-Z_-]{3,16}\$
additional_fields_ref	{ 2 fields }
1	{ 5 fields }
2	{ 5 fields }
1	{ 2 fields }

- L'amministratore può creare tipologie di campi aggiuntivi ed assegnare una tipologia ad un target.
- Tutti i ticket di un determinato target avranno, oltre ai campi base, dei campi aggiuntivi definiti dalla tipologia assegnata al target.
- Una tipologia di campo aggiuntivo comprende uno o più campi aggiuntivi che devono essere compilati da un cliente.
- Ogni campo aggiuntivo ha un nome, placeholder, un'espressione regolare ed un tipo.

Gestione Team

- L'amministratore può creare e modificare i team ed assegnarvi un tipo ed un target, creati, anche, in runtime.
- La creazione è possibile solo se contestualmente viene assegnato ad esso almeno un membro, il quale, se unico, diventerà team leader.
- Per eliminare un team è necessario rimuovere tutti i membri; finché l'ultimo membro ha dei ticket pendenti non sarà possibile eliminare il team.
- L'eliminazione di un utente dal sistema o un membro dal team comporta la ridistribuzione dei ticket agli altri membri con una funzione arbitraria (attualmente random).
- L'eliminazione di un utente sarà impedita qualora esso sia l'unico componente di un team ed abbia ticket pendenti.

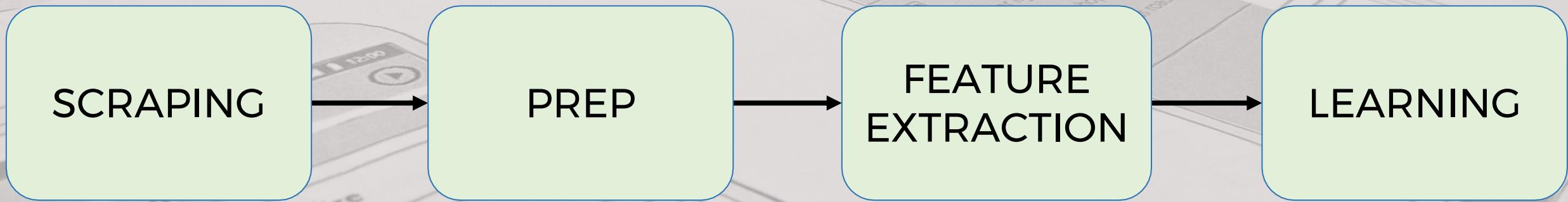
Gestione Utenti



- L'utente può effettuare registrazione, login e logout dalla piattaforma.
- Può modificare i dati essenziali del suo profilo eccetto ruolo e username.
- L'amministratore può creare e modificare gli utenti ed assegnargli un ruolo (ruoli presi dal **configFile**).

Pipeline di machine learning

Dallo scraping alla classificazione



- Salvataggio sul database nel formato ticket
- Sorgente ticket: Stack Overflow
- Stemming
- Rimozione delle stopword
- Modello vettoriale: matrice termini x documenti e pesatura tf-idf
- Rappresentazione nello spazio LSA
- **Clustering** con algoritmo K-means
- **Multiclassificazione** con algoritmo C-SVM e funzione Kernel lineare e polinomiale, schema One-vs-All



Scraping

Servizio REST /scraping/{page}/{name}

- Invochiamo le API esposte pubblicamente da Stack Overflow per ottenere i dati che costituiranno il training dataset.
- Otteniamo le domande relative ad un *name*, ad es. ‘mongo’, per mezzo di un client HTTP implementato dalla libreria **Retrofit2**



Preparation

Servizio REST `prepareDatasetByTarget/{target}/{reduction}`

1. Ottiene i ticket dal database mongoDB.
2. Applica le tecniche di stemming e rimozione delle stopword a ciascun ticket.
3. Genera i vettori di feature associati ai ticket tramite la funzione tf-idf e crea la matrice [*Terms* × *Docs*].
4. Applica la trasformazione Latent Semantic Analysis alla matrice sfruttando SVD.
5. Genera il file di dataset in formato .klp.
6. Crea una directory nel cloud S3 per ospitare i dataset di ciascun target così creati.
7. Esegue l'upload del file di dataset in cloud.

Stemming

Modulo Stemmer

- Lo **stemming** è un processo dell'Information Retrieval che permette di ridurre una parola da una sua forma flessa alla radice, o tema. Si parla di
- La taglia del dizionario di training si riduce e così anche la dimensione dello spazio dei vettori associati ai documenti.
- La recall migliora ma la precision potrebbe diminuire.
- Impieghiamo l'algoritmo di stemming di Porter.

Rule		
SSES	→	SS
IES	→	I
SS	→	SS
S	→	

Example		
caresses	→	caress
ponies	→	poni
caress	→	caress
cats	→	cat

Stopword

Modulo StopWordsRemover

- Le **stopword** sono parole di un documento che non riguardano un argomento specifico.
- Le preposizioni (es. after, for, in, to in inglese) e le congiunzioni (es. and, or, but, as, if) sono esempi tipici di stopword.
- Non aggiungono valore semantico al testo e rimuoverle permette di risparmiare risorse computazionali.
- Il Data Analyst dispone di un'interfaccia per inserire e rimuovere stopword.

a	been	get
about	before	getting
after	being	go
again	between	goes
age	but	going
all	by	gone
almost	came	got
also	can	gotten
am	cannot	had
an	come	has
and	could	ha

tf - idf

Modulo TFIDFElaborator

- I termini poco frequenti sono molto informativi → peso maggiore.
- I termini molto frequenti hanno poca rilevanza → peso minore.
- Definiamo la **term frequency** $tf_t = \# di occorrenze del termine t nel documento$.
- Definiamo la **inverse document frequency**

$$idf_t = \frac{\# di documenti di training}{\# di documenti in cui occorre il termine t} .$$

- Associamo ad ogni documento un vettore le cui componenti sono pesi associati ai termini che lo compongono.
- Calcoliamo il peso del termine t con la funzione $tf_t \cdot idf_t$

Latent Semantic Analysis (LSA)

Modulo LSAElaborator

- **LSA** è una tecnica di Information Retrieval per identificare un modello delle relazioni tra parole e concetti contenuti in un documento.
- LSA è una applicazione della tecnica matematica della Singular Value Decomposition (**SVD**).
- La decomposizione in valori singolari della matrice W [*Terms* × *Docs*] è $W = U \Sigma V^T$.
- Questa decomposizione si può approssimare con una matrice $W' = U_k \Sigma_k V_k^T$ di rango k molto minore del rango di W .
- Un Data Analyst può scegliere di quanto ridurre il rango della matrice [*Terms* × *Docs*] di partenza.

LSA in azione

Riduzione del rango

$$\begin{matrix} M \times N \\ W' \end{matrix} = \begin{matrix} M \times k \\ U_k \end{matrix} \begin{matrix} k \times k \\ S_k \end{matrix} \begin{matrix} k \times N \\ V_k^T \end{matrix}$$

Singular Value Decomposition (SVD)

Modulo SVDElaborator

- La decomposizione in valori singolari della matrice W [*Terms* × *Docs*] è $W = U \Sigma V^T$.
- Σ è la matrice diagonale dei **valori singolari** di W , cioè le radici degli autovalori di WW^T .
- I valori singolari identificano le dimensioni principali del dataset di partenza.
- Queste direzioni corrispondono ai **concetti** di cui si parla nei documenti.
- Un concetto è determinato dal ricorrere degli stessi termini in documenti differenti.

Clustering

Servizio REST /elaborateCluster/{target}/{K}

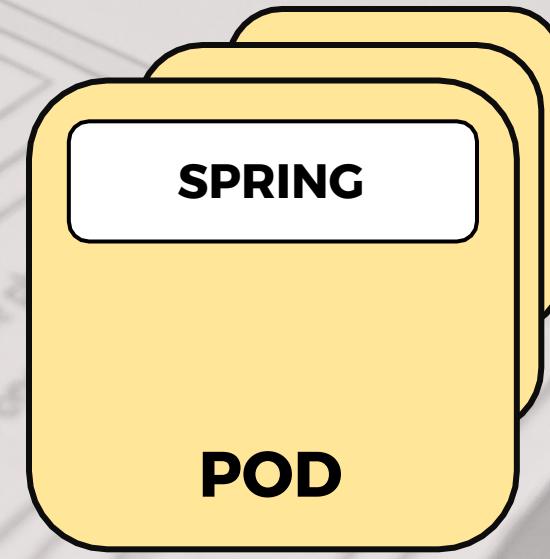
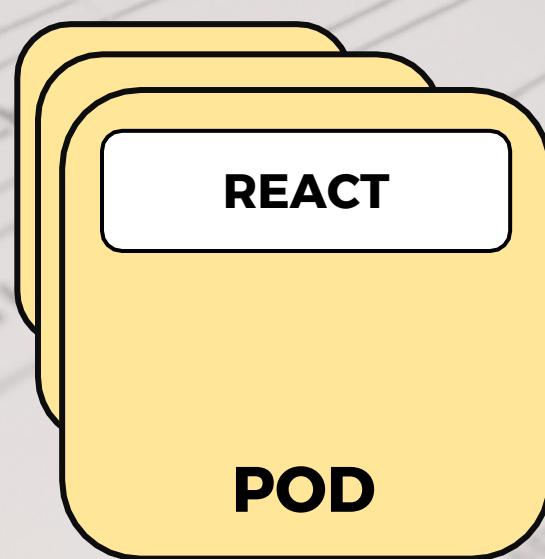
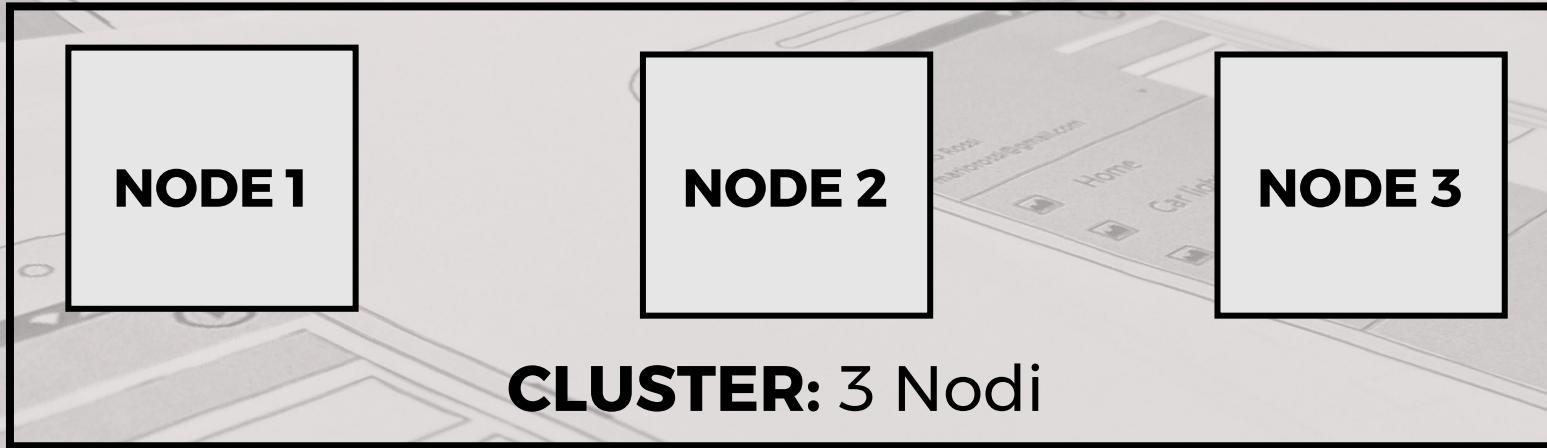
- Algoritmo K-Means per la clusterizzazione
- K è il numero di cluster che supponiamo di creare
- {target} è il target su cui andremo ad eseguire la clusterizzazione partendo dal file .klp generato nelle fasi precedenti.

Classificazione

Servizio REST /elaborateCluster/{target}/{K}

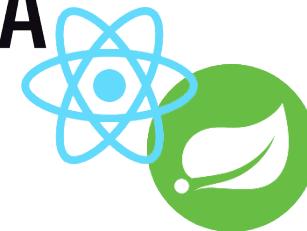
- Algoritmo C-Support Vector Machine
- Funzione Kernel lineare e polinomiale (lin, poly)
- Multiclassificazione tramite schema One-vs-All
- Sviluppi futuri:
 - Implementazione di altri metodi di pesatura dei termini sfruttando rappresentazione sparse vector
 - Classificazione: passive-aggressive

KUBERNETES





UNIVERSITA' degli STUDI di ROMA
T O R V E R G A T A



Grazie per l'attenzione

Pelella, Perrone, Pusceddu, Scarlino, Scarpitta, Tranzocchi, Trotta,