



Corso di Laurea in Informatica

Classificazione delle Domande sulla Sicurezza dei Post di Stack Overflow

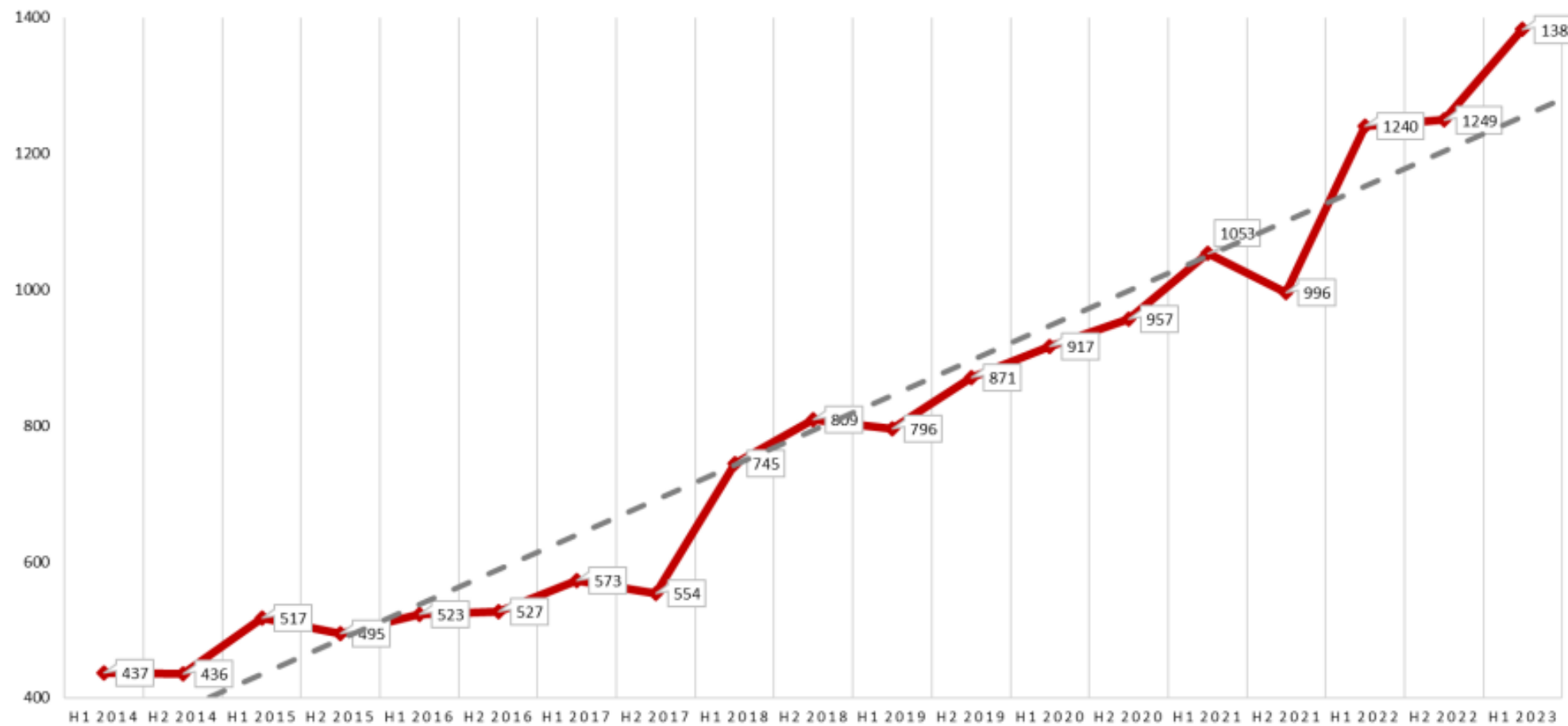
Prof. Fabio Palomba
Dott. Emanuele Iannone

Giuseppe Grano
Mat.: 0512110454



Introduzione e Background

Attacchi per semestre H1 2014 - H1 2023




© Clusit - Rapporto 2023 sulla Sicurezza ICT in Italia - aggiornamento giugno 2023

Introduzione e Background



Fonte dei dati testuali



The screenshot shows a Stack Overflow question titled "Why is char[] preferred over String for passwords?". The question was asked 11 years, 11 months ago and has 511k views. It has 3837 votes and 17 answers. The top answer, by Rakete1111, has 47.5k votes and explains that Strings are immutable and can be accessed from memory dumps, while char arrays can be wiped. The second answer, by Ahamed, has 39.4k votes and explains that arrays can be overwritten. The question is tagged with java, string, security, passwords, and char.

Why is char[] preferred over String for passwords?
Asked 11 years, 11 months ago · Modified 10 months ago · Viewed 511k times

In Swing, the password field has a `getPassword()` (returns `char[]`) method instead of the usual `getText()` (returns `String`) method. Similarly, I have come across a suggestion not to use `String` to handle passwords.

Why does `String` pose a threat to security when it comes to passwords? It feels inconvenient to use `char[]`.

java string security passwords char

edited Jan 13, 2017 at 11:48 by Rakete1111 (47.5k ● 17 ● 124 ● 164)

asked Jan 16, 2012 at 14:20 by Ahamed (39.4k ● 13 ● 41 ● 68)

Share Follow

Add a comment

17 Answers

Sorted by: Highest score (default)

Strings are immutable. That means once you've created the `String`, if another process can dump memory, there's no way (aside from [reflection](#)) you can get rid of the data before [garbage collection](#) kicks in.

With an array, you can explicitly wipe the data after you're done with it. You can overwrite the array with anything you like, and the password won't be present anywhere in the system, even before garbage collection.

So yes, this *is* a security concern - but even using `char[]` only reduces the window of opportunity for an attacker, and it's only for this specific type of attack.

As noted in the comments, it's possible that arrays being moved by the garbage collector will leave stray copies of the data in memory. I believe this is implementation-specific - the garbage collector *may* clear all memory as it goes, to avoid this sort of thing. Even if it does, there's still the time during which the `char[]` contains the actual characters as an attack window.

The Overflow Blog

- Three types of AI-assisted programmers
- What Gemini means for the GenAI boom

Featured on Meta

- Seeking feedback on tag colors update
- Site maintenance - Wednesday, December 13, 2023 @ 01:00 UTC (Tuesday,...)
- Collectives updates: new features and ways to get started with Discussions
- OverflowAI Alpha invitation emails were distributed in error Nov 28th
- Temporary policy: Generative AI (e.g., ChatGPT) is banned

Linked

- 24 Why we read password from console in char array instead of String
- 15 Is it a good practice to nullifying String in java
- 13 Java storing sensitive 'key' as String or char[]?
- 3 Why we should not use String for Storing password in Java but can use String for Storing password in C language?
- 4 How to pass SSL keystore password?
- 3 Clearing a memory in java heap space from JNI

Introduzione e Background

Classificazione manuale

-lenta

-poco scalabile

Classificazione automatica

-veloce

-scalabile

Introduzione e Background

Classificazione manuale

- lenta
- poco scalabile

Classificazione automatica

- veloce
- scalabile

Uso del modello
LDA (**Latent
Dirichlet
Allocation**)

In letteratura scientifica
esistono diversi lavori

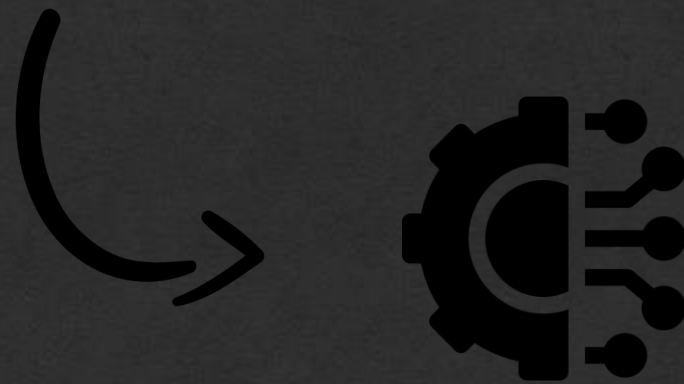
Obiettivo: realizzare un classificatore di domande, relative al tema sicurezza, presenti su Stack Overflow e Security Stack Exchange

Due problemi di ricerca:

- scelta del modello;
- scelta dell'algoritmo di bilanciamento del training set;

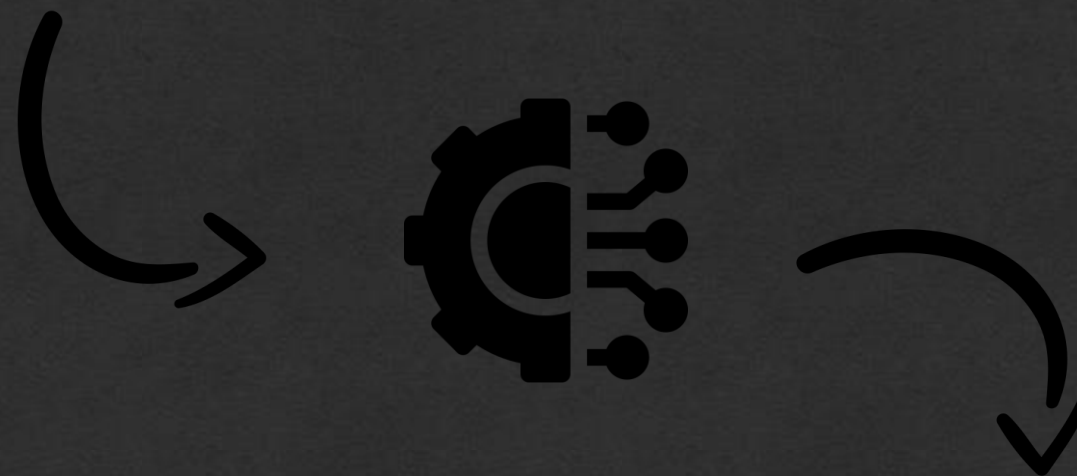
Obiettivo: realizzare un classificatore di domande, relative al tema sicurezza, presenti su Stack Overflow e Security Stack Exchange

I need to implement 256 bit AES encryption, but all the examples I have found online use a "KeyGenerator" to generate a 256 bit key, but I would like to use my own passkey. How can I create my own key? I have tried padding it out to 256 bits, but then I get an error saying that the key is too long. I do have the unlimited jurisdiction patch installed, so thats not the problem :)



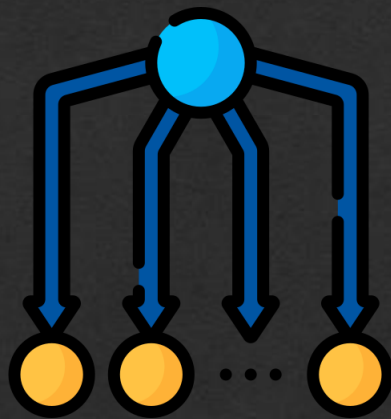
Obiettivo: realizzare un classificatore di domande, relative al tema sicurezza, presenti su Stack Overflow e Security Stack Exchange

I need to implement 256 bit AES encryption, but all the examples I have found online use a "KeyGenerator" to generate a 256 bit key, but I would like to use my own passkey. How can I create my own key? I have tried padding it out to 256 bits, but then I get an error saying that the key is too long. I do have the unlimited jurisdiction patch installed, so thats not the problem :)

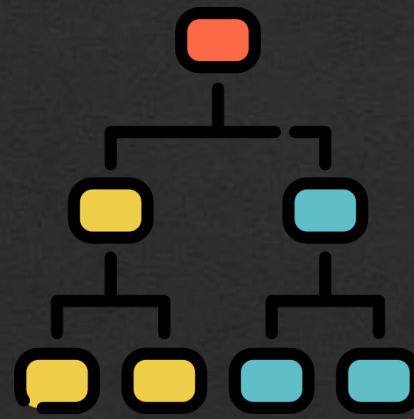


Encryption Errors

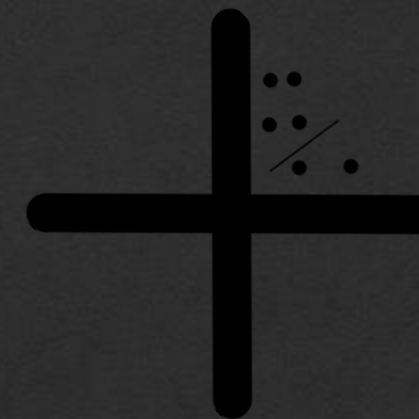
RQ1 . Quale è il modello di classificazione più adatto per determinare la tipologia di domanda relativa alla sicurezza?



Multinomial
Naive Bayes

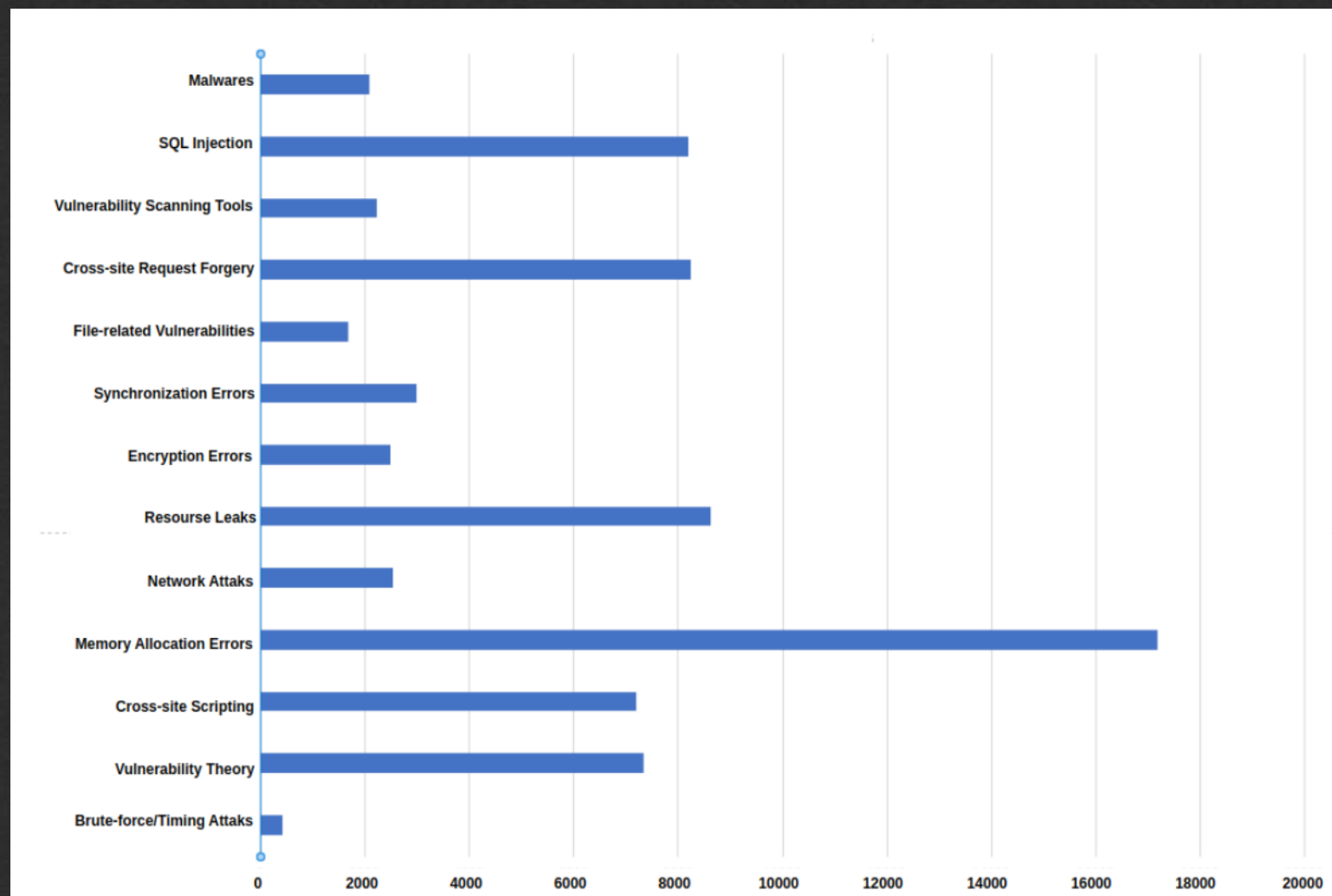


Decision Tree



LinearSVC

RQ2 . Qual è l'algoritmo di bilanciamento più adatto per determinare la tipologia di domanda relativa alla sicurezza?



Problema durante l'apprendimento del modello, utilizzando il training set sbilanciato

Algoritmi di bilanciamento:

-SMOTE

-Random Undersampling

Preparazione
dei dati



Sono stati utilizzati
13 Dataset

Dati presenti nei 13
Dataset sono stati
uniti in un unico
Dataset, formato da
due campi: **Text** e
Label

Testo della domanda

Classe di appartenenza

Preparazione
dei dati Preprocessing



Pipeline di pulizia

Lowercasing
Punctuation removal
Lemmatization
Stopwords removal

Preparazione
dei dati

Preprocessing

Vettorizzazione



Utilizzo del modello
Bag of Word



Preparazione
dei dati

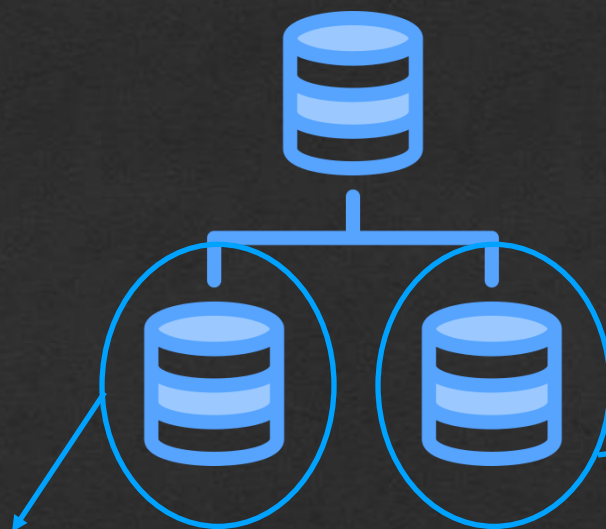
Preprocessing

Vettorizzazione

Divisione del
Dataset



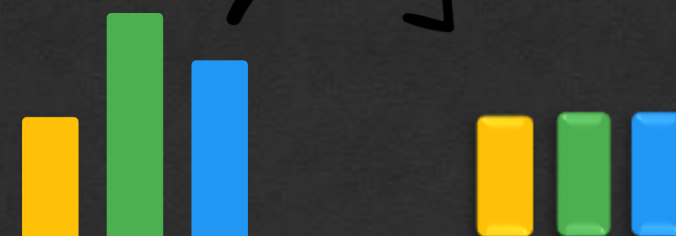
Divisione del dataset



Test set (20%)

Training set (80%)

Bilanciamento attraverso
Random Undersampling e
SMOTE



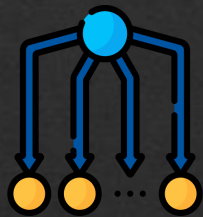
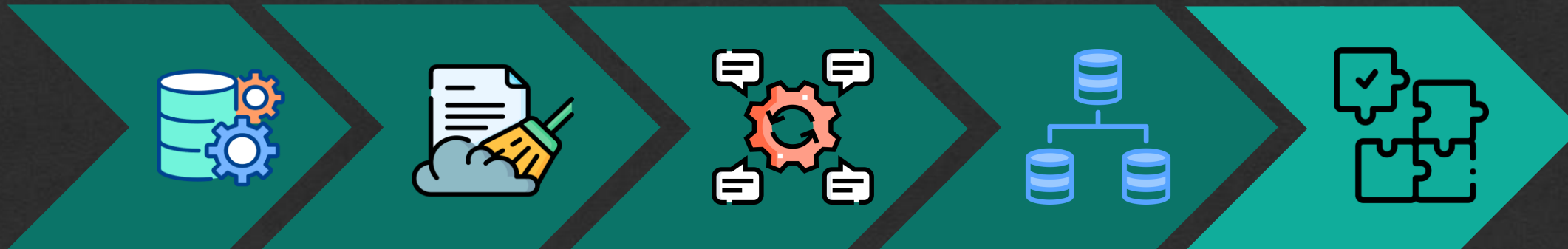
Preparazione
dei dati

Preprocessing

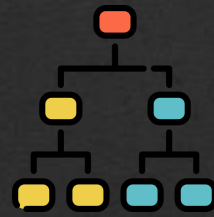
Vettorizzazione

Divisione del
Dataset

Addestramento
dei modelli



Multinomial
Naive Bayes



Decision
Tree



LinearSVC

Nella prima fase, i tre modelli sono stati addestrati sul training set non bilanciato.

Il modello risultato più performante, nella seconda fase è stato addestrato con il training set bilanciato.

Preparazione
dei dati

Preprocessing

Vettorizzazione

Divisione del
Dataset

Addestramento
dei modelli

Valutazione

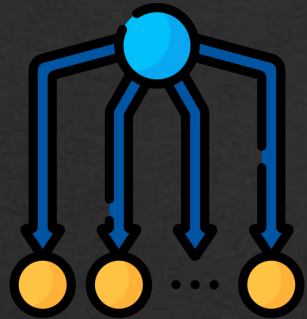


Metriche utilizzate:

Micro F1 score

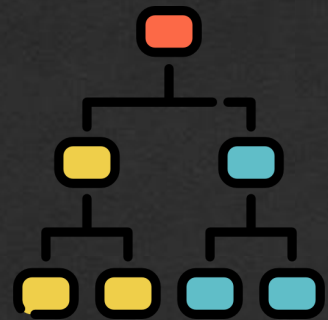
Micro Precision

Micro Recall



Multinomial Naive Bayes

micro F1 score = 0.78



Decision Tree

micro F1 score = 0.57

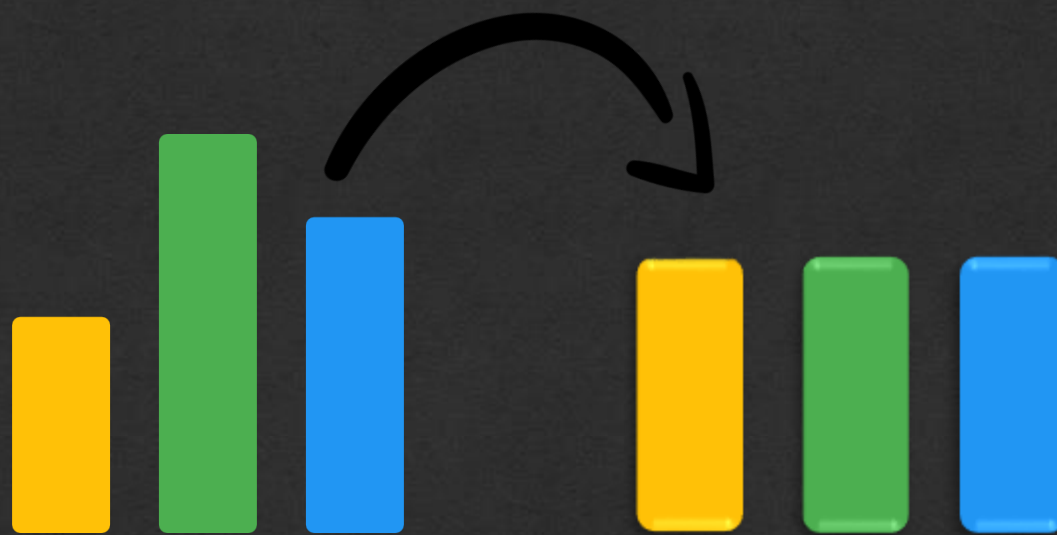


LinearSVC

micro F1 score = 0.76

RQ1 – Il modello che si adatta meglio al task della classificazione delle domande di sicurezza è il Multinomial Naive Bayes

Modello utilizzato: Multinomial Naive Bayes



Random Undersampling

Micro F1 score = 0.71



SMOTE

Micro F1 score = 0.80

RQ2 – La tecnica di bilanciamento del training set è l'oversampling, in particolare nel nostro caso, con l'algoritmo SMOTE



g.grano3@studenti.unisa.it



<https://github.com/Grano14>



www.linkedin.com/in/giuseppe-grano-215656279

Classificazione delle domande sulla sicurezza dei post di Stack Overflow

Giuseppe Grano

Università degli Studi di Salerno

Questo lavoro di ricerca ha permesso di aggiungere alcune considerazioni sui modelli che si adattano meglio al task della classificazione dei testi.

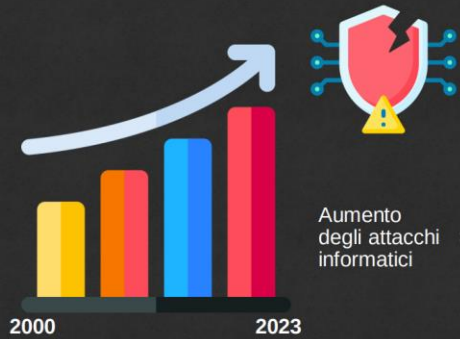
Inoltre è stato messo in evidenza come le tecniche di oversampling sono da preferire alle tecniche di undersampling.

Sviluppi futuri:

- valutazione approfondita delle metriche per singola classe;
- considerazione di diverse tecniche di oversampling confrontate allo SMOTE;

Introduzione e Background

seso^{lab}
SOFTWARE ENGINEERING
SALERNO



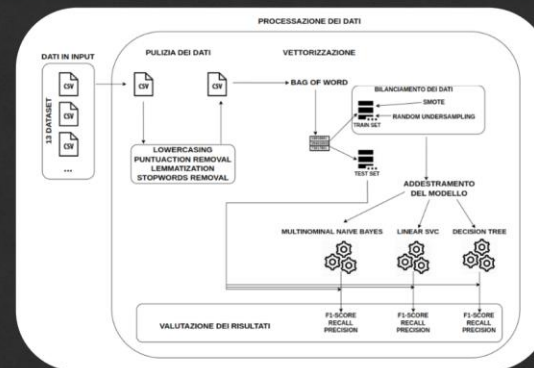
Aumento degli attacchi informatici = Aumento delle analisi sul tema sicurezza

✉ g.grano3@studenti.unisa.it
🌐 <https://github.com/Grano14>
🌐 www.linkedin.com/in/giuseppe-grano-215656279

Classificazione delle domande sulla sicurezza dei post di Stack Overflow
Giuseppe Grano
Università degli Studi di Salerno

Metodologia

seso^{lab}
SOFTWARE ENGINEERING
SALERNO



✉ g.grano3@studenti.unisa.it
🌐 <https://github.com/Grano14>
🌐 www.linkedin.com/in/giuseppe-grano-215656279

Classificazione delle domande sulla sicurezza dei post di Stack Overflow
Giuseppe Grano
Università degli Studi di Salerno

Analisi dei Risultati

seso^{lab}
SOFTWARE ENGINEERING
SALERNO



Multinomial Naive Bayes micro F1 score = 0.78

Decision Tree micro F1 score = 0.57

LinearSVC micro F1 score = 0.76

✉ g.grano3@studenti.unisa.it
🌐 <https://github.com/Grano14>
🌐 www.linkedin.com/in/giuseppe-grano-215656279

Classificazione delle domande sulla sicurezza dei post di Stack Overflow
Giuseppe Grano
Università degli Studi di Salerno

Conclusioni

seso^{lab}
SOFTWARE ENGINEERING
SALERNO

RQ1 – Il modello che si adatta meglio al task della classificazione delle domande di sicurezza è il Multinomial Naive Bayes

RQ2 – La tecnica di bilanciamento del training set è l'oversampling, in particolare nel nostro caso, con l'algoritmo SMOTE

✉ g.grano3@studenti.unisa.it
🌐 <https://github.com/Grano14>
🌐 www.linkedin.com/in/giuseppe-grano-215656279

Classificazione delle domande sulla sicurezza dei post di Stack Overflow
Giuseppe Grano
Università degli Studi di Salerno

Classificazione delle domande sulla sicurezza dei post di Stack Overflow

Grazie!



Questa tesi ha contribuito a piantare un albero in Kenya



Giuseppe Grano

g.grano3@studenti.unisa.it ✉
<https://github.com/Grano14> 🌐
www.linkedin.com/in/giuseppe-grano-215656279 🌐