

ISW2 ML

Francesco Bernardini

Matricola:0338264

Agenda

- 1) Contesto e Obiettivo
- 2) Metodologia
- 3) Risultati e conclusioni
- 4) Minacce alla validità

Contesto

In ogni progetto di ingegneria del software c'è una parte di testing.

Grazie a lei si possono rilevare e fixare bug.

Il problema però è che l'attività di testing è molto costosa, perciò non è possibile testare in modo esaustivo tutto.

Idea: Riuscire ad individuare le parti di un progetto di ingegneria del software che possono avere con più probabilità dei bug.

Obiettivo

Partendo dall'idea iniziale si vuole predire la bugginess delle classi di due progetti Apache, Bookkeeper e Storm, tramite delle tecniche di Machine Learning.

Ottimizzando così la fase di testing.

Verranno usati vari classificatori con varie tecniche di utilizzo e si vedrà l'andamento delle predizioni a seconda del classificatore e tecnica scelta.

- **Classificatori usati:** Random Forest, Naive Bayes e IBk.
- **Tecniche di utilizzo:** Feature Selection, Sampling e Cost Sensivity.

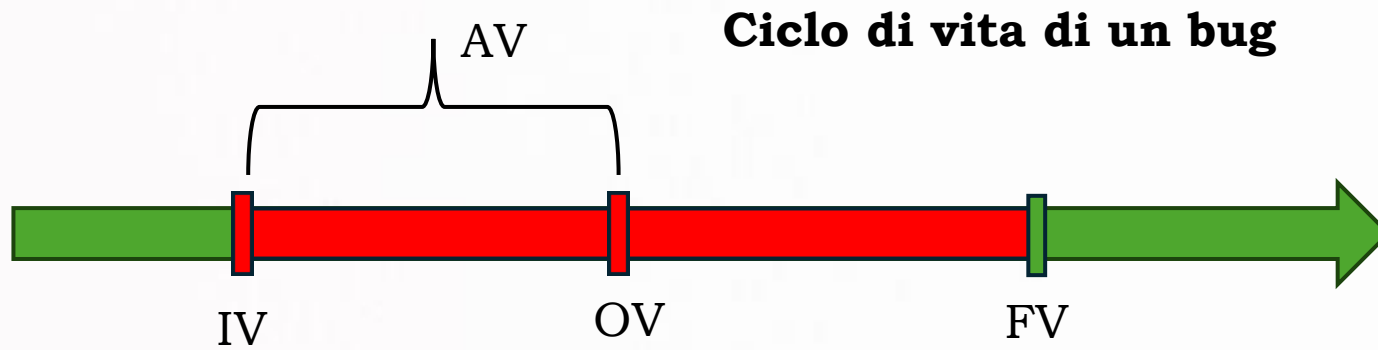
Metodologia: Acquisizione Dati

Jira: Software utilizzato per raccogliere i ticket dei progetti. Tramite le API di Json, si ricaveranno ticket fixati e non.

Git: Sistema per la raccolta di commit. Verrà usato in particolare il plugin JGit.



Metodologia: Creazione dei ticket



Le IV, OV e FV sono state recuperate tramite le issue di Jira

Problema: Non tutti i ticket di Jira hanno una IV associata.

Metodologia:

Creazione dei ticket (2)

La soluzione è stimare la IV dei ticket che non ce l'hanno.

Proportion: una tecnica che calcola la costante di proporzionalità p sui ticket che hanno una IV e permette di stimare la IV dei ticket che non ce l'hanno.

$$P = \frac{FV - IV}{FV - OV}$$

$$IV = FV - (FV - OV) * P$$

Metodologia:

Creazione dei ticket (3)

Per il calcolo di Proportion sono stati utilizzati due metodi:

Cold Start: calcolo la P del bug corrente come la P media dei bug di altri progetti. È utile per i primi bug di un progetto. Il problema qua è definire cosa è simile.

Incremental, calcolo P corrente come la P media dei bug passati, cioè su tutto quello che so dalle release precedenti. Questo approccio ha il vantaggio di utilizzare il maggior numero di informazioni, all'interno dello stesso progetto.

Metodologia:

Assunzioni

- Per evitare lo snoring sono state tolte l'ultima metà delle release.
- Sono stati rimossi tutti i Ticket dove la OV e/o la FV non erano presenti.
- Sono stati rimossi tutti i Ticket dove non c'era consistenza dei dati, nello specifico $OV > FV$ e/o $IV > OV$

Metodologia: Metriche

Metrica	Descrizione
Size	Numero di linee di codice
nR	Numero di revisioni
nAuth	Numero di autori
LOC Touched	Somma delle linee di codice aggiunte e rimosse nelle revisioni di una release
LOC Added	Somma delle linee di codice aggiunte nelle revisioni di una release
Max LOC Added	Numero massimo di linee aggiunte nelle revisioni di una release
Avg LOC Added	Numero medio di linee aggiunte nelle revisioni di una release
Churn	Somma di $ Loc\ Added - Loc\ Removed $ nelle revisioni di una release
Max Churn	Numero massimo di $ Loc\ Added - Loc\ Removed $ nelle revisioni di una release
Avg Churn	Numero medio di $ Loc\ Added - Loc\ Removed $ nelle revisioni di una release

Metodologia: Valutazione classificatori



Walk Forward

È una tecnica time-series molto utilizzata.

Qui il set di dati viene diviso in parti, ognuna è una release del progetto.

Le parti vengono ordinate cronologicamente e, in ogni esecuzione, tutti i dati disponibili prima della parte da prevedere vengono utilizzati come training set e la parte da prevedere viene utilizzata come test set.

Run	Parte				
	1	2	3	4	5
1	Testing	Training	Training	Training	Training
2	Training	Testing	Training	Training	Training
3	Training	Training	Testing	Training	Training
4	Training	Training	Training	Testing	Training
5	Training	Training	Training	Training	Testing

 Training
 Testing

Metodologia:

Classificatori e tecniche di utilizzo

I classificatori utilizzati sono:

- Random Forest
- Naive Bayes
- IBk

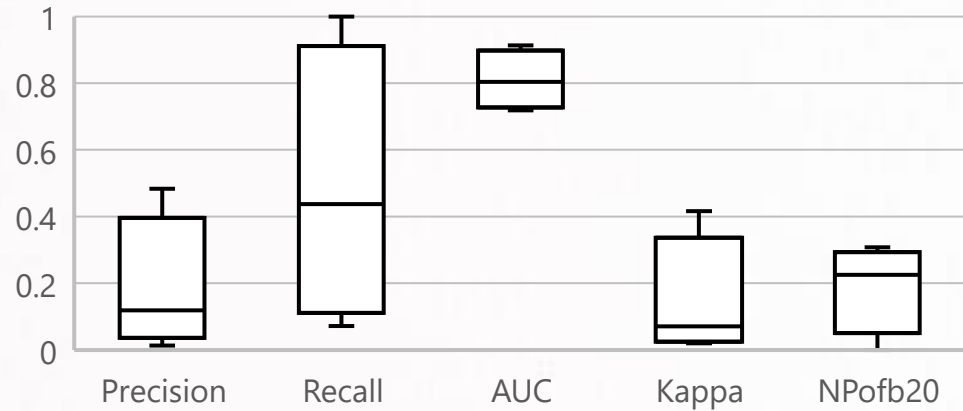
Le tecniche di utilizzo sono:

- Feature Selection
- Balancing (Undersampling e SMOTE)
- Sensitive Learning ($CFN = 10 * CFP$)

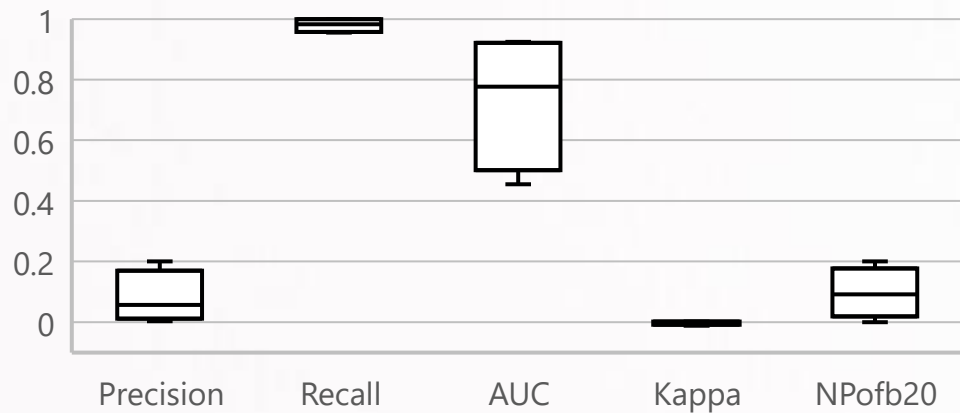
Risultati:

Bookkeeper Senza Filtri

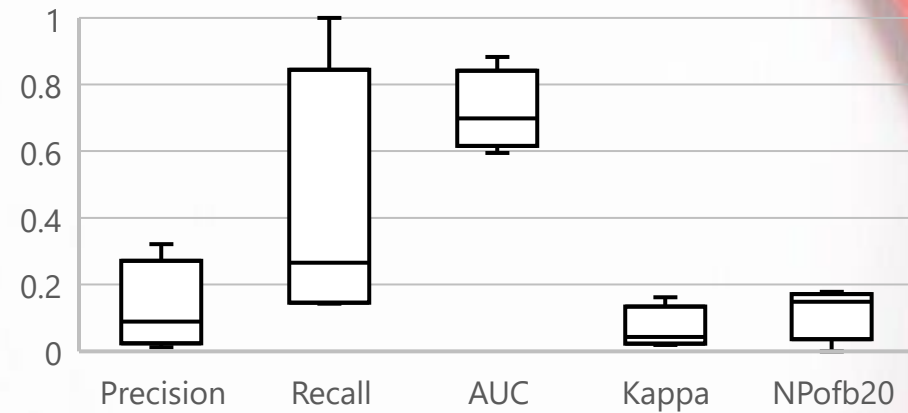
Random Forest



Naive Bayes



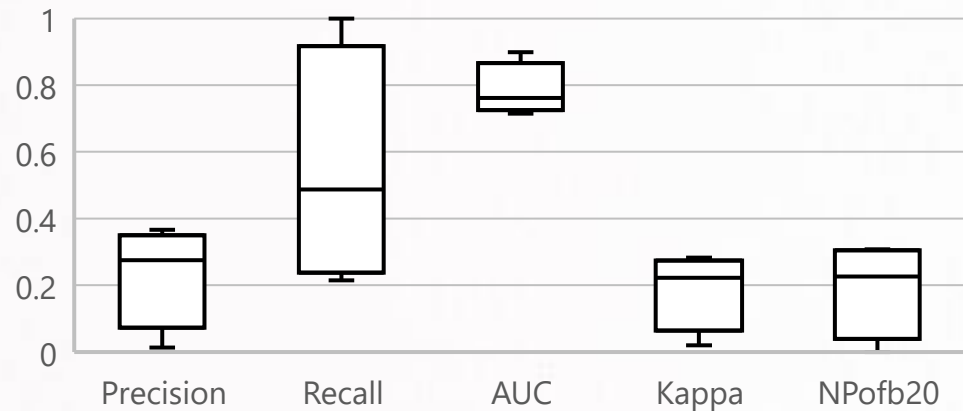
IBk



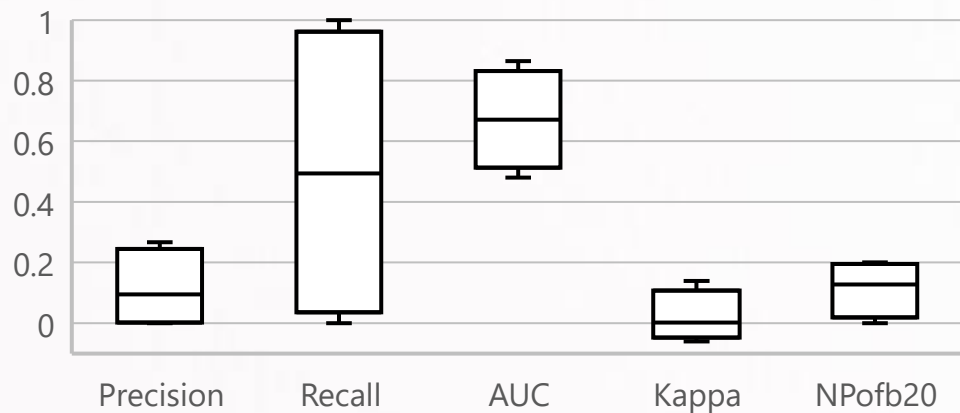
Risultati:

Bookkeeper Feature Selection

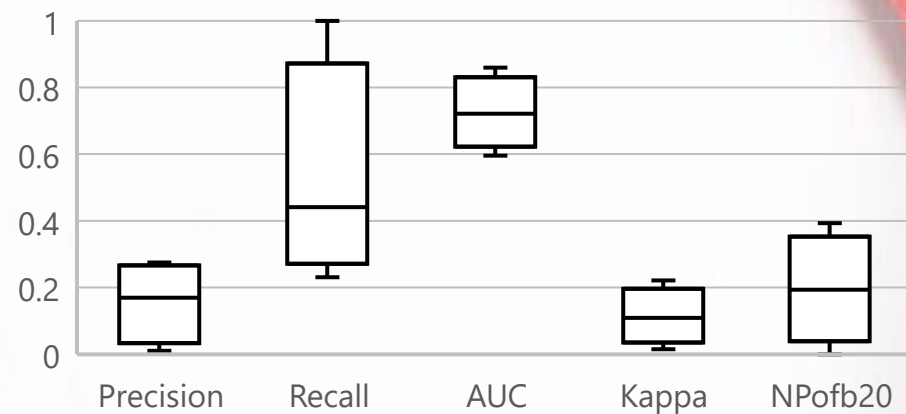
Random Forest



Naive Bayes



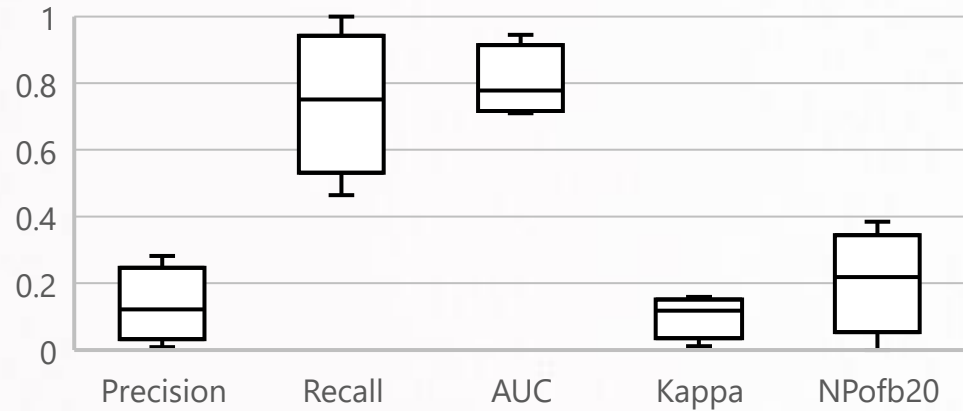
IBk



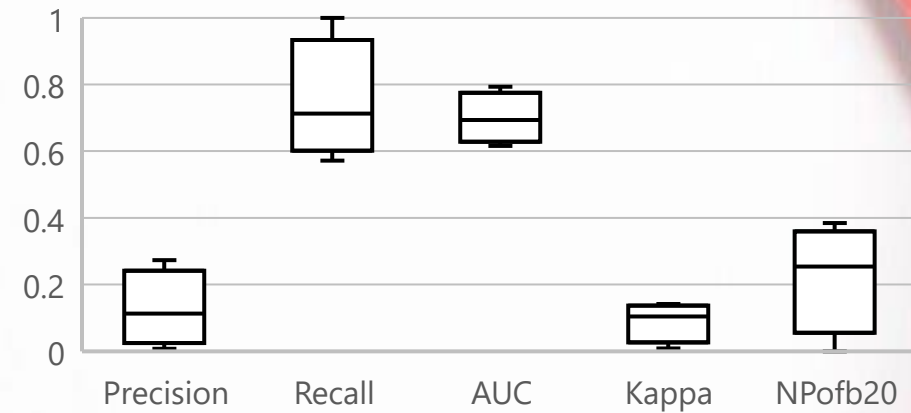
Risultati:

Bookkeeper Feature Selection e Undersampling

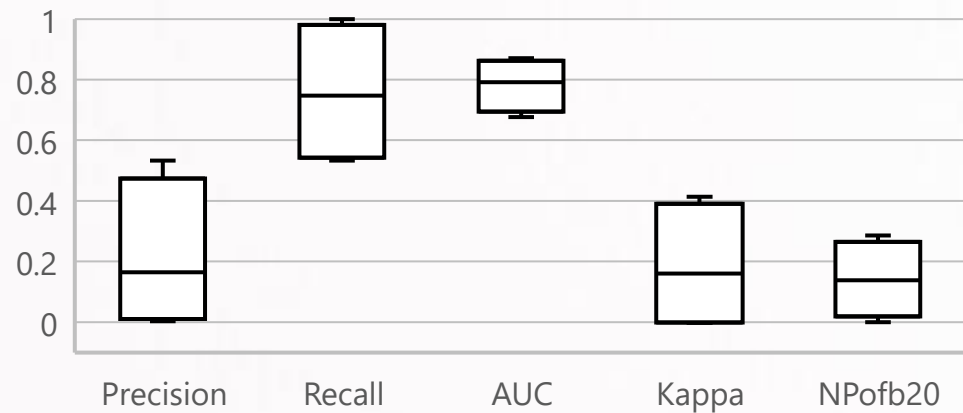
Random Forest



IBk



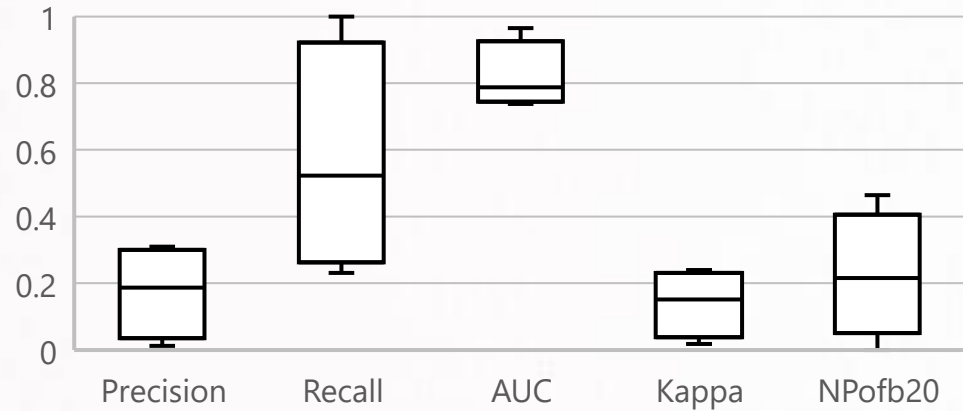
Naive Bayes



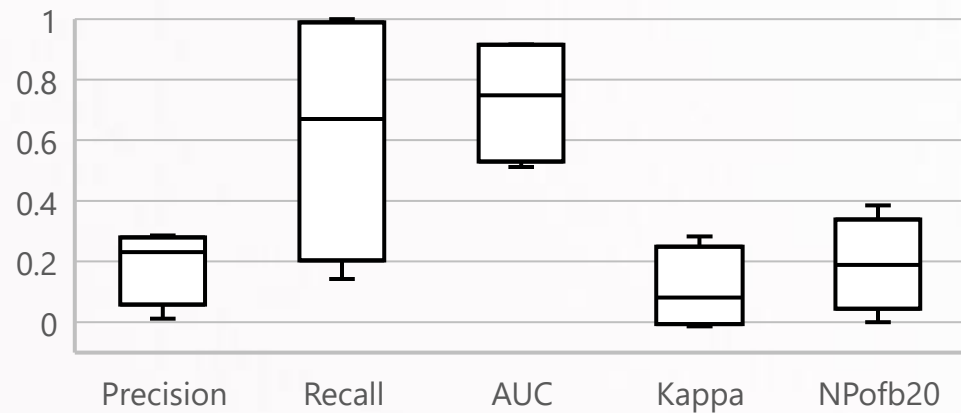
Risultati:

Bookkeeper Feature Selection e SMOTE

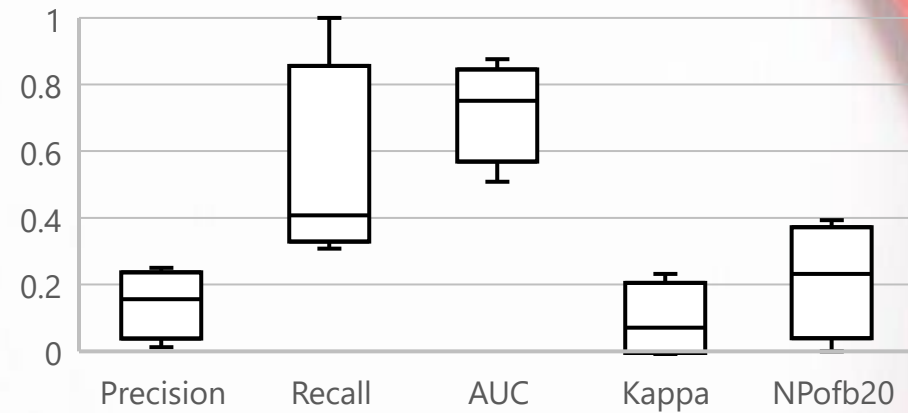
Random Forest



Naive Bayes



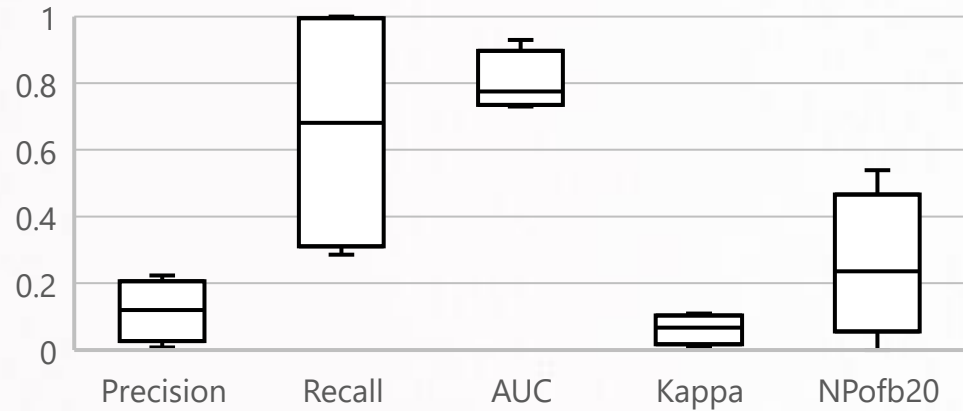
IBk



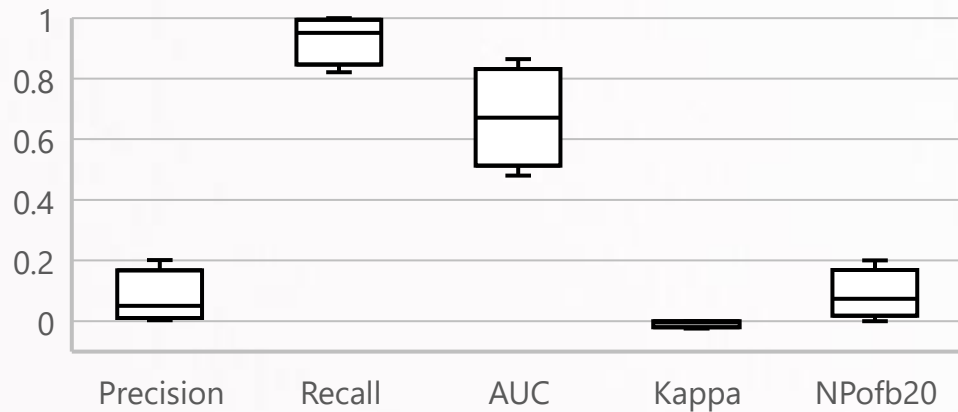
Risultati: Bookkeeper

Feature Selection e Sensitive Learning

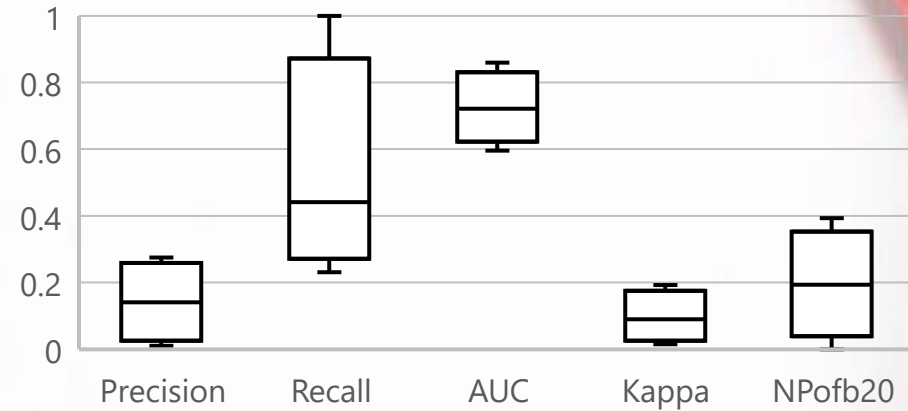
Random Forest



Naive Bayes



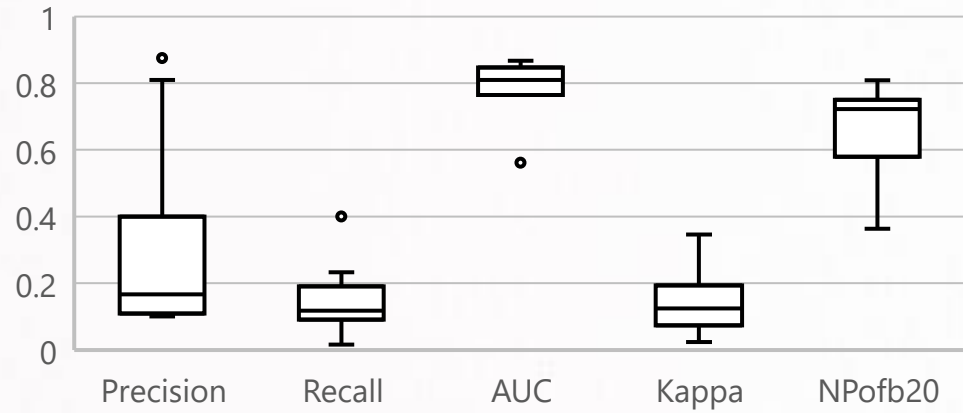
IBk



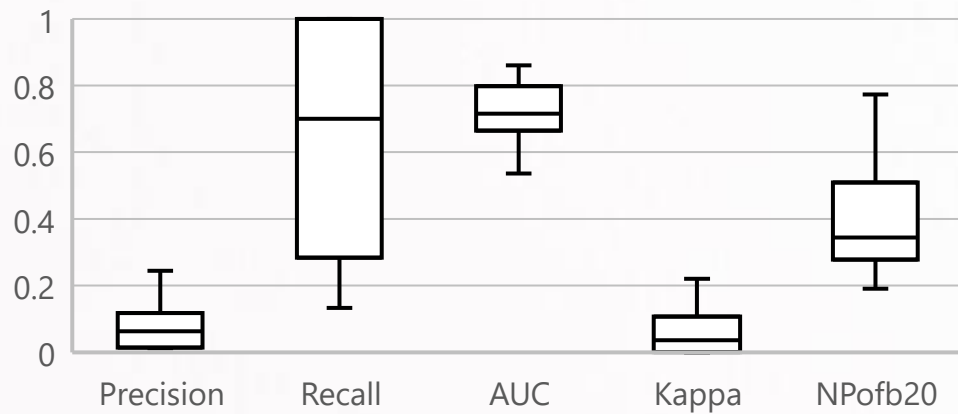
Risultati:

Storm Senza Filtri

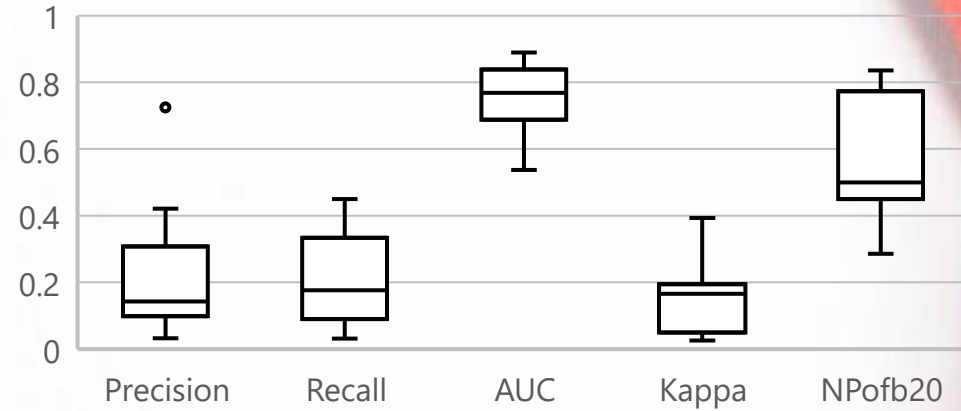
Random Forest



Naive Bayes



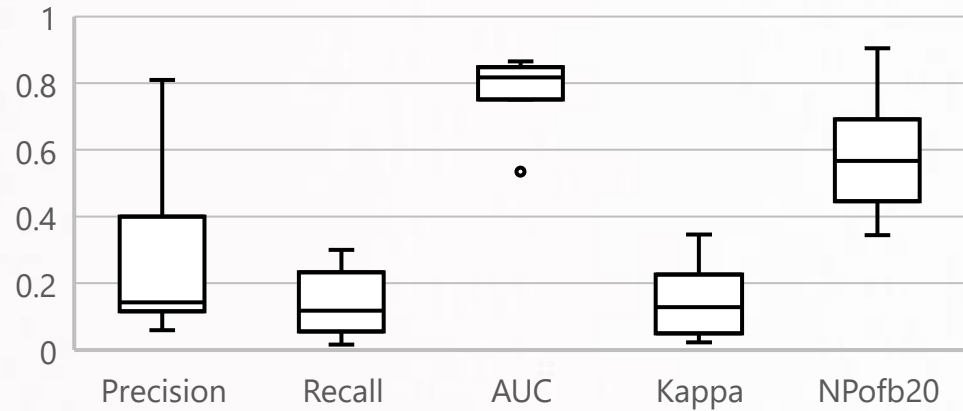
IBk



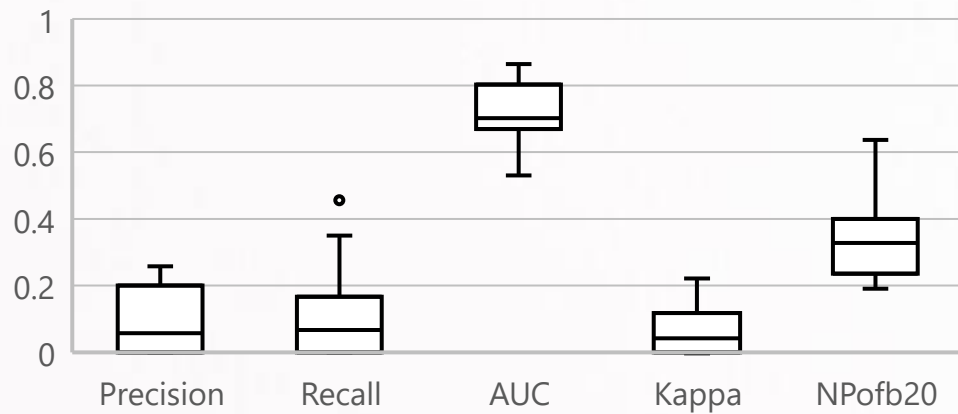
Risultati:

Storm Feature Selection

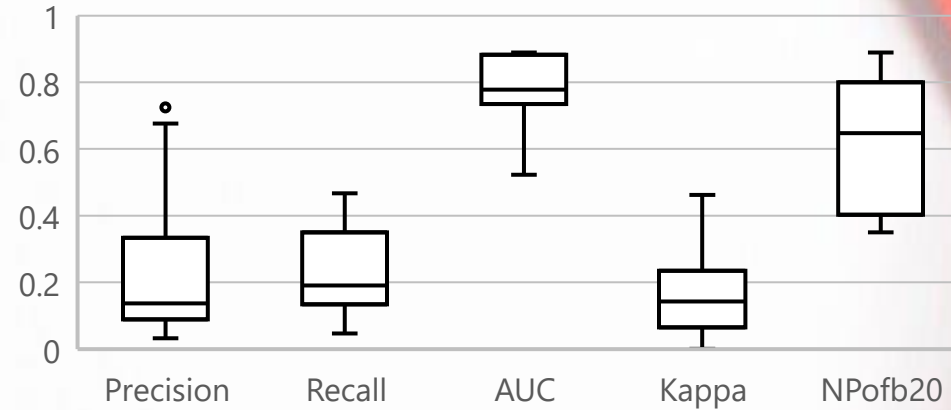
Random Forest



Naive Bayes



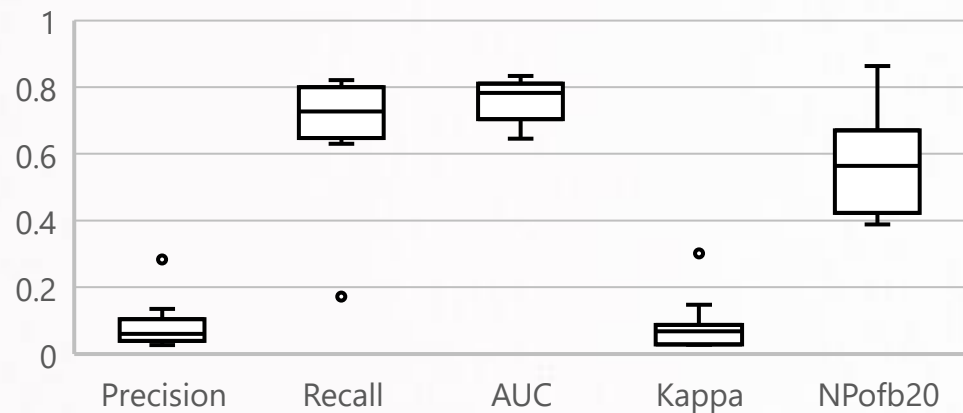
IBk



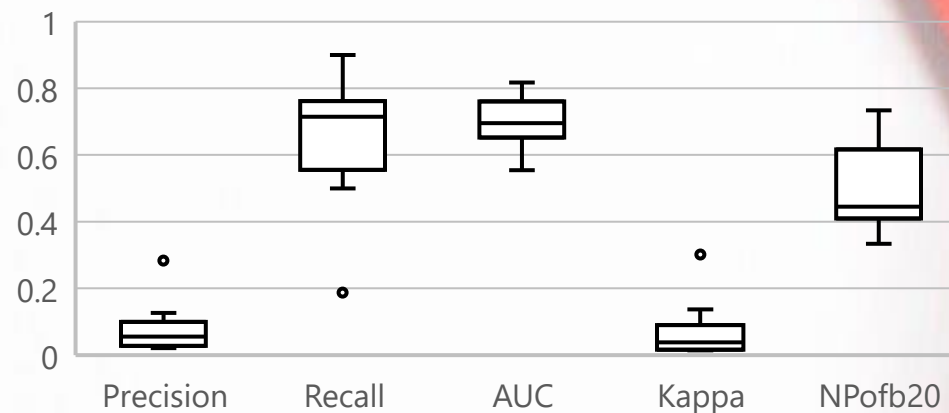
Risultati:

Storm Feature Selection e UnderSampling

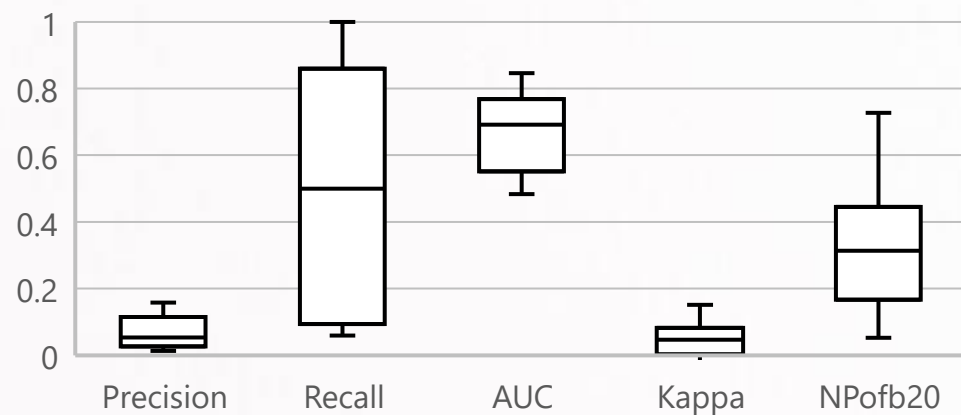
Random Forest



IBk



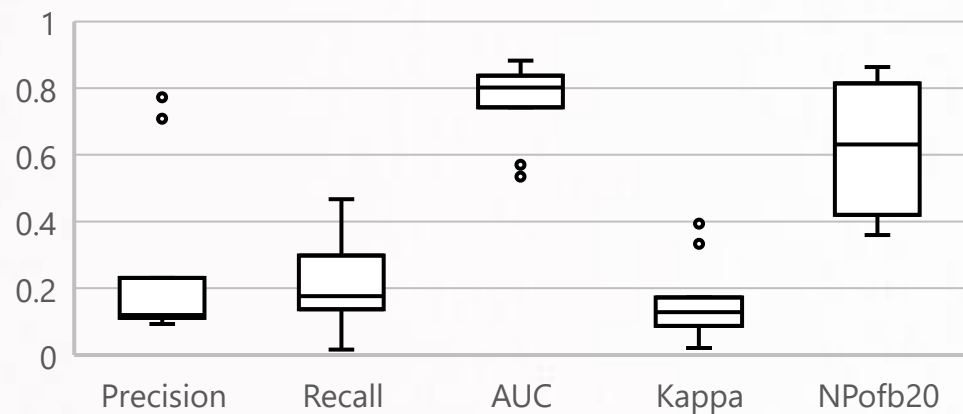
Naive Bayes



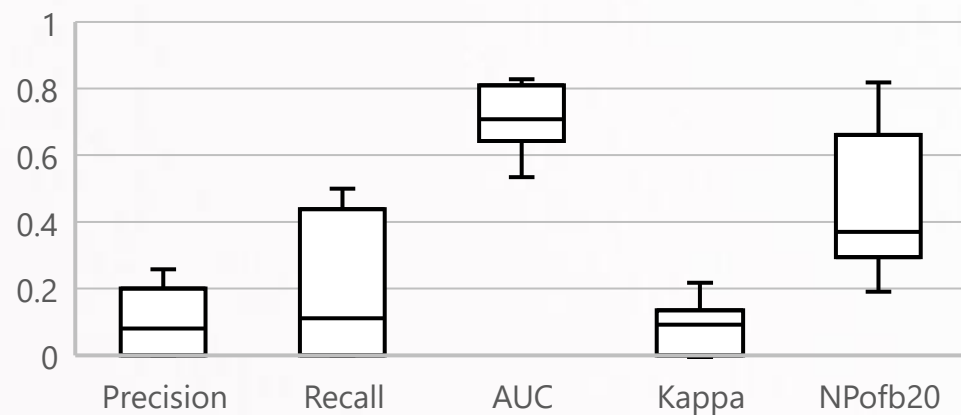
Risultati:

Storm Feature Selection e SMOTE

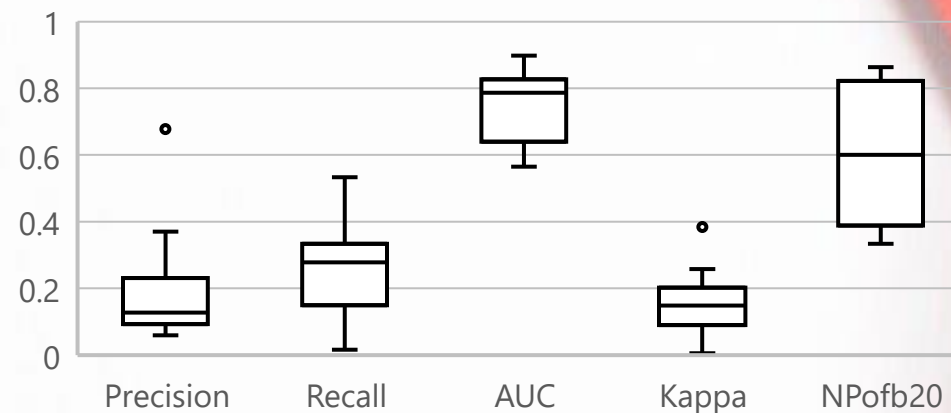
Random Forest



Naive Bayes



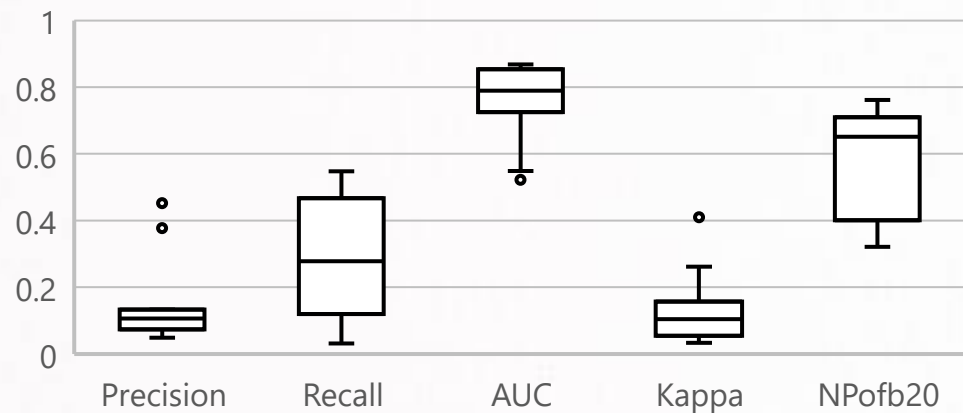
IBk



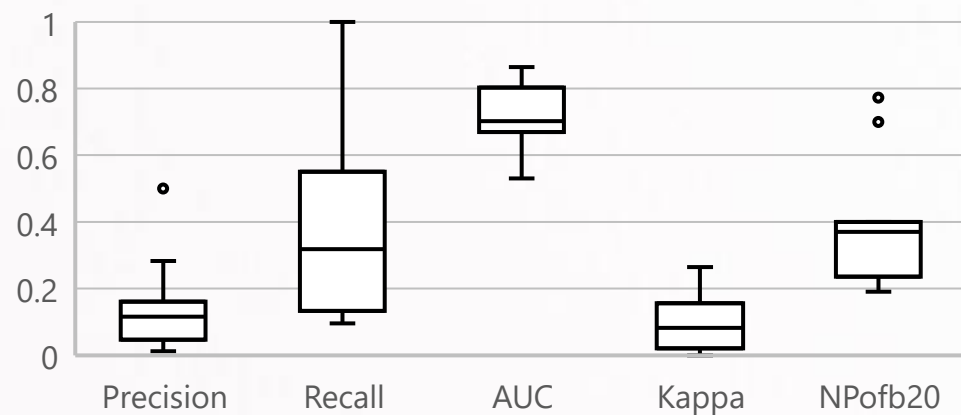
Risultati: Storm

Feature Selection e Sensitive Learning

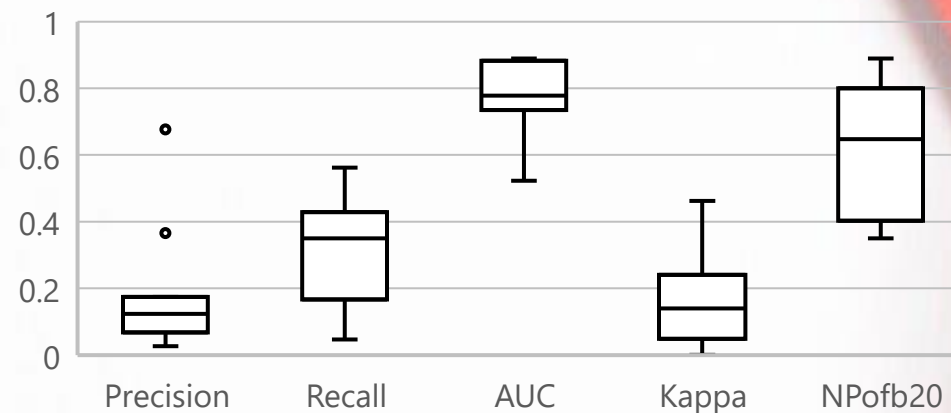
Random Forest



Naive Bayes



IBk



Conclusioni

Non c'è una configurazione migliore in assoluto, però:

Bookkeeper: il classificatore con prestazioni migliori è Naive Bayes con Feature Selection e Undersampling

Storm: il classificatore con prestazioni migliori è Random Forest con Feature Selection

Minacce alla validità

- La FV dei ticket è stata ricavata tramite la resolution date, mentre la OV con la creation date.
- Per ogni ticket è stata assegnata la prima release della AV come IV.
- Per fare Cold Start, è stato assunto che gli altri progetti Apache sono simili ai due progetti analizzati.
- In Jira non sono presenti le release che non hanno una data.
- Sono stati usati per la Proportion, solo i ticket che avevano la AV.

Grazie per l'attenzione!

Link utili:

Repository GitHub: <https://github.com/Berna1998/isw2-ML>

SonarCloud: https://sonarcloud.io/project/overview?id=Berna1998_isw2-ML