MACHINE LEARNING FOR SOFTWARE ENGINEERING

PROGETTO ISW2 SEZIONE FALESSI

Sara Da Canal – matricola 0316044
Università degli Studi di Roma Torvergata
Anno di corso 2021-2022

INTRODUZIONE

In questo progetto andremo a valutare come diverse tecniche di sampling, feature selection o cost sensitivity vanno a migliorare l'accuratezza di diversi classificatori disponibili su Weka.

Prenderemo in considerazione due progetti apache da cui estrarre il dataset per i classificatori, Syncope e Bookkeeper.

I classificatori considerati saranno:

- Random Forest
- Naive Bayes
- IBk con k=1

DATASET

I diversi classificatori dovranno valutare la probabilità che una classe sia buggy o meno. Vengono considerate solo classi Java, e non vengono considerate le classi di test. Per ottenere un dataset su cui testare i classificatori, siamo dovuti andare a calcolare, per ogni progetto, quali classi fossero buggy.

- È stato usato Jira per ottenere affected version, fixed version e opening version per ogni bug conosciuto del progetto
- Dove l'affected version non fosse presente o fosse inconsistente è stata calcolata la più probabile affected version con un metodo di proportion incrementale
- Ogni bug è stato associato ai file interessati usando il log dei commit ottenuto da git

Per evitare che il dataset sia reso invalido da troppe classi buggy ma non riconosciute tali, viene considerata solo la metà meno recente delle versioni, dove la buggyness può essere assunta abbastanza corretta

PROGETTI

- La buggyness dei due progetti è stata calcolata nello stesso modo per quanto riguarda ticket validi, quindi con affected e fixed version consistenti
- Per i ticket invalidi, su Bookkeeper ho usato un metodo di proportion incrementale, su Syncope cold start. Questo è dovuto al fatto che Bookkeeper ha dati consistenti per quanto riguarda le opening version, Syncope ne ha molto pochi e quindi i risultati ottenuti con la proportion incrementale non erano sensati

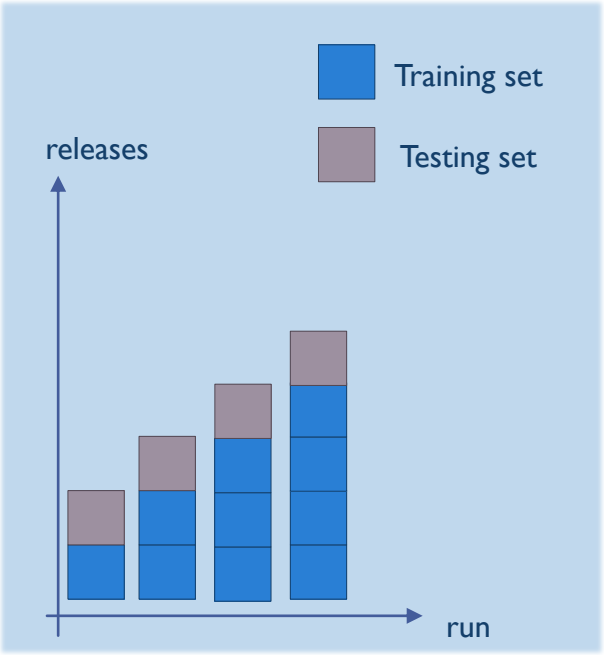
- Syncope è un progetto con dati molto meno coerenti, non si hanno informazioni sulla buggyness fino alla settima versione e anche dopo la media delle classi buggy sul totale è circa il 3,5% per il training e il 5% per il testing. Questi numeri hanno una varianza molto elevata dato che la maggior parte dei bug sono relativi alle versioni 1.1.4 e 1.1.5
- Bookkeeper ha dati più coerenti anche se molti di meno con circa il 10% di classi buggy nel training set e circa l'8% nel testing set. Solo la prima versione ha un training set privo di bug

Da questo si può supporre che i dati ottenuti da Bookkeeper abbiano validità maggiore nonostante la loro minore quantità

DIVISIONE DEL DATASET

Il dataset è stato diviso con la tecnica walk-forward:

- Inizialmente la prima versione, con la buggyness calcolata con le informazioni conosciute al momento della release fa da training set e la seconda, con la buggyness aggiornata ad oggi, da testing set
- All'iterazione n, le prime n versioni costituiscono il training set e la (n+1)-esima il testing set. La buggyness del training set è calcolata alla data dell'n-esima release.
- L'accuratezza è calcolata come la media sulle varie run



VARIABILI

Le variabili considerate per determinare la buggyness di una classe sono le seguenti. L'ipotesi più una classe sia grande, subisca modifiche o venga toccata da persone diverse più sia buggy

- Size: taglia della classe al momento della release
- Commit number: numero di commit relativi alla classe all'interno della release
- LOC touched: linee di codice modificate dai commit
- LOC added: linee di codice aggiunte alla classe
- Max LOC added: maggior numero di linee di codice aggiunte in un unico commit
- Avg LOC added: media delle linee di codice aggiunte per commit
- Churn: differenza tra le linee aggiunte o rimosse dalla classe all'interno della release
- Max churn: massimo churn ottenuto in unico commit
- Avg churn: churn medio per commit
- Authors' number: numero di autori che hanno lavorato sulla classe

FILTRI CONSIDERATI

Per ogni dataset sono state considerate diverse combinazioni di filtri:

Sampling	No sampling	Under sampling	Over sampling
Feature selection	No feature selection	Best first	
Cost sensitivity	No cost sensitive	Sensitive threshold	Sensitive learning

SAMPLING

Il sampling è necessario per evitare che ci siano forti sbilanciamenti tra le istanze buggy e non buggy presentate al classificatore, che potrebbero portarlo ad avere una maggiore accuratezza nel predirre istanze della classe prevalente e una accuratezza minore per la classe minoritaria

Under sampling

Vengono considerate tutte le istanze minoritarie e solo una parte di quelle maggioritarie

Over sampling

Le istanze minoritarie vengono replicate fino ad avere un numero pari di istanze maggioritarie o minoritarie

Su Syncope, l'under sampling porta i risultati migliori per tutti i classificatori tranne Naive Bayes, per cui la cosa migliore è non applicare il sampling

Su Bookkeeper, per Naive Bayes e Random Forest l'over sampling porta precision leggermente migliore e l'under sampling leggermente peggiore, per IBk non portano cambiamenti significativi

FEATURE SELECTION

Lo scopo della feature selection è considerare solo il più piccolo sottoinsieme di attributi significativo invece che l'intero insieme di variabili nel fare le predizioni, in modo da poter ridurre i costi dell'apprendimento e migliorare le predizioni.

Per effettuare feature selection è necessario selezionare un metodo di ricerca degli attributi e un algoritmo per la loro valutazione.

Attribute evaluator: CfsSubsetEval

Search Method: Best First

Su Syncope, applicare feature selection non porta cambiamenti significativi tranne che per Naive Bayes per cui le prestazioni peggiorano

Su Bookkeeper, per ognuno dei classificatori considerati, usare la feature selection migliora la precisione

COST SENSITIVITY

Un classificatore cost sensitive è un classificatore che assegna costi diversi agli errori. Nel nostro caso, ogni falso negativo costa 10 volte in più rispetto a un falso positivo, essendo un errore più grave per i nostri scopi.

Sensitive threshold

La soglia sulla probabilità di appartenere a una classe invece che essere lo standard 0.5 viene calcolata come CFP/(CFP+CFN), ovvero 0.09

Sensitive learning

Le istanze vengono replicate tante volte quant'è il loro costo

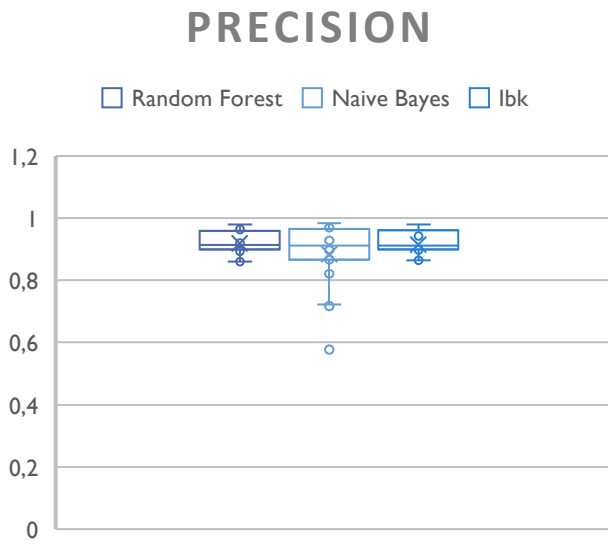
Su Syncope, il sensitive threshold migliora la precisione di Random Forest e il sensitive learning di Naive Bayes, mentre su Ibk non si notano cambiamenti

Su Bookkeeper, per Naive Bayes ed Ibk la precisione diminuisce introducendo una tecnica cost sensitive, mentre per Random Forest il sensitive learning ha precisione maggiore

PRECISION

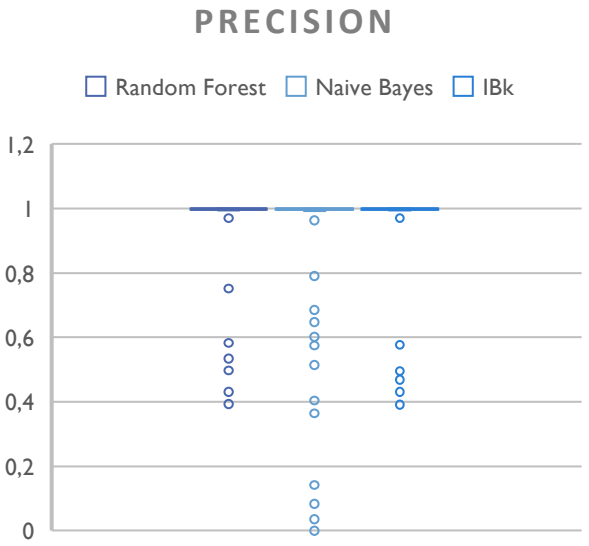
Bookkeeper

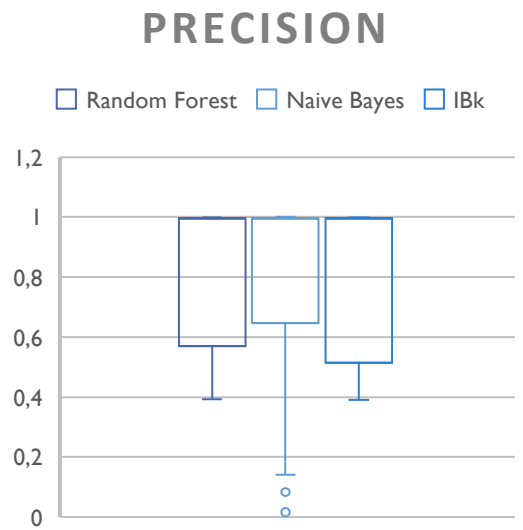
L'analisi della precisione mostra Random Forest e IBk abbiano varianze simili, Random Forest ha una media leggermente migliore. Naive Bayes ha media inferiore e varianza molto più elevata, può anche essere notato un outlier (0,57) che corrisponde a una politica di cost sensitive threshold e under sampling



Syncope

I testing set per cui non abbiamo bug vanno a portare media e varianza molto vicine a uno per ogni classificatore, ma possiamo comunque vedere come Naive Bayes sia il più variabile con addirittura alcuni valori sullo zero. Il grafico di destra è stato ottenuto eliminando le release con i testing set non significativi e mostra, come per Bookkeeper risultati simili per IBk e Random Forest e risultati peggiori per Naive Bayes

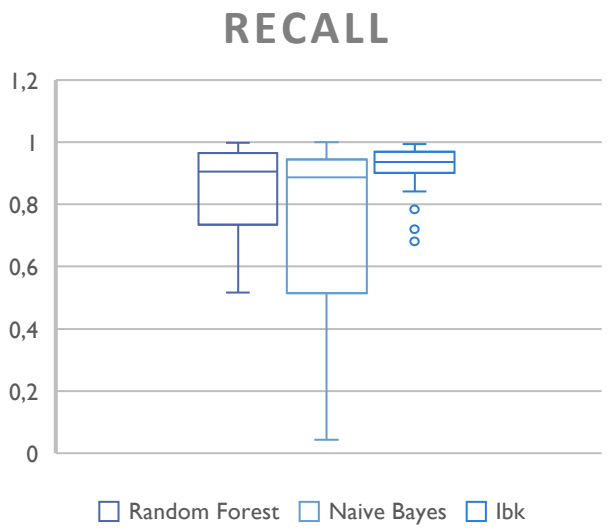




RECALL

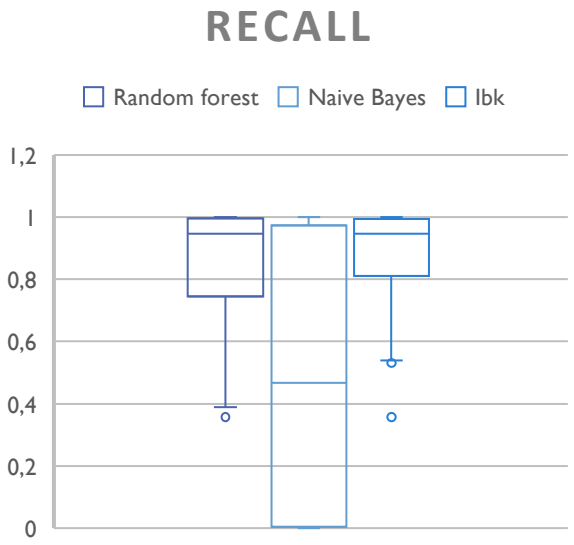
Bookkeeper

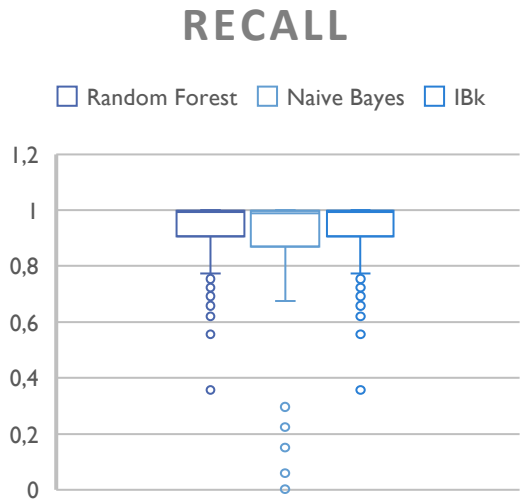
La recall maggiore la ha IBk, che mostra poca variabilità e qualche outliers, Naive Bayes ha la recall peggiore e molto variabile, arriva fino a valori molto vicini allo zero



Syncope

Anche con Syncope Naive Bayes mostra la recall peggiore, con una varianza enorme quando consideriamo ogni testing set e molti outliers, fino ad arrivare a recall zero, quando eliminiamo i testing set non validi. Eliminando questi testing set, forse a causa dei pochi dati disponibili, ogni classificatore presenta outliers

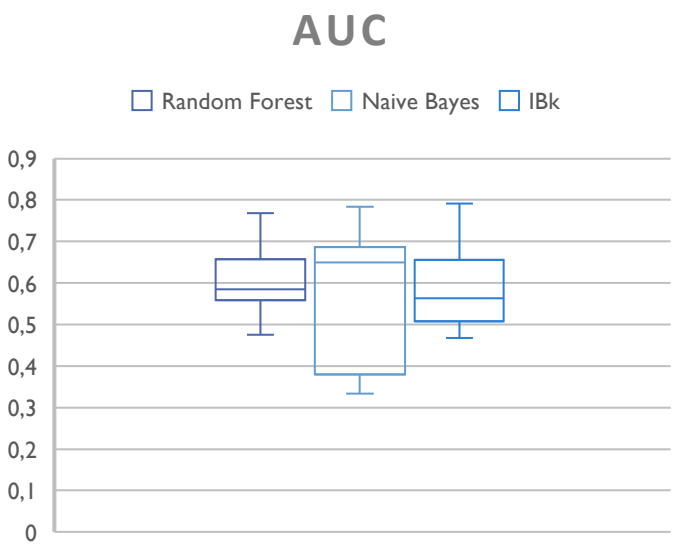




AREA UNDER ROC

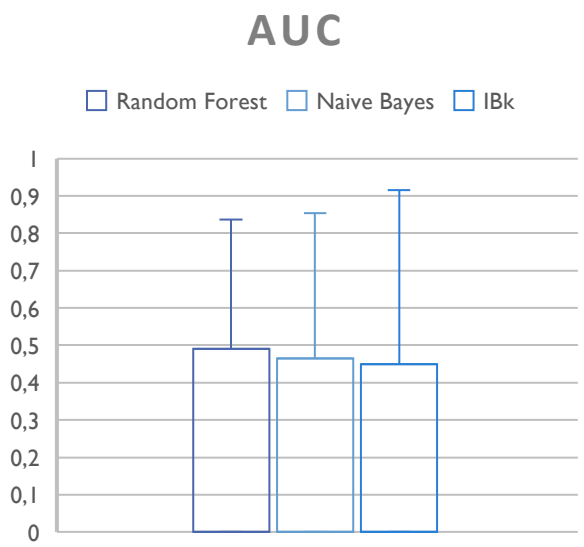
Bookkeeper

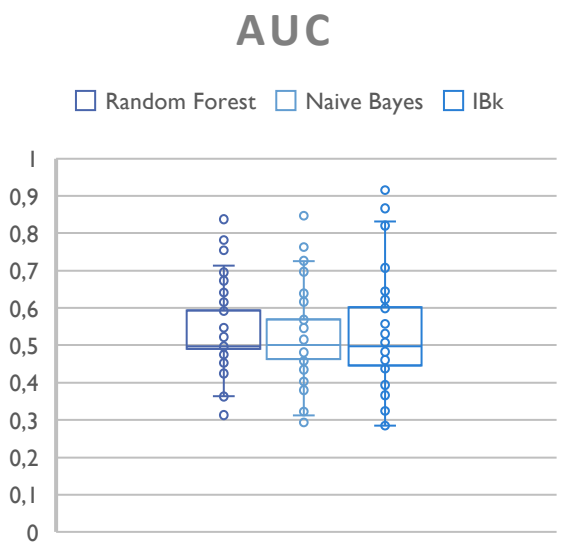
Naive Bayes ha una media molto alta per l'area under ROC, ma varianza elevata, mentre Random Forest ha una media minore ma tutti valori vicini alla media. IBk ha la media peggiore di tutti



Syncope

Dall'AUC di Syncope si vede come solo una metrica non sia rappresentativa di un buon classificatore: se la precision tendeva ad uno per i testing set privi di buggyness, l'AUC tende a zero, mostrando che il classificatore non sta facendo il giusto lavoro. Togliendo i dati non validi, l'AUC media è circa la stessa per ogni classificatore e Random Forest ha la minore varianza

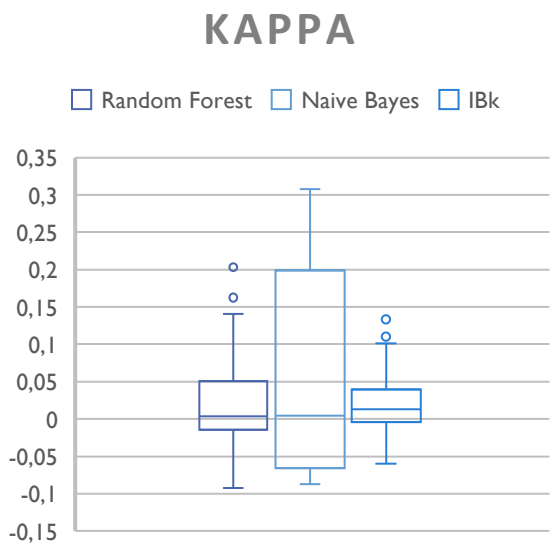




KAPPA

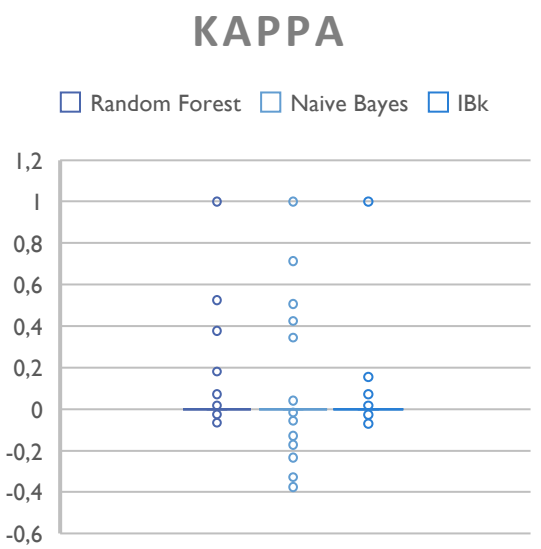
Bookkeeper

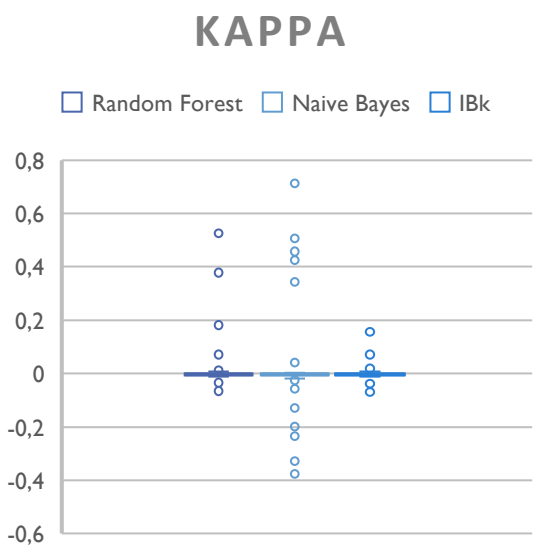
Forse a causa dei pochi dati, le medie dei valori di kappa sono molto vicine a zero, il che significa classificatori con la stessa accuratezza di un classificatore dummy. IBk sembra avere la media migliore ed è anche il classificatore con i valori meno negativi.



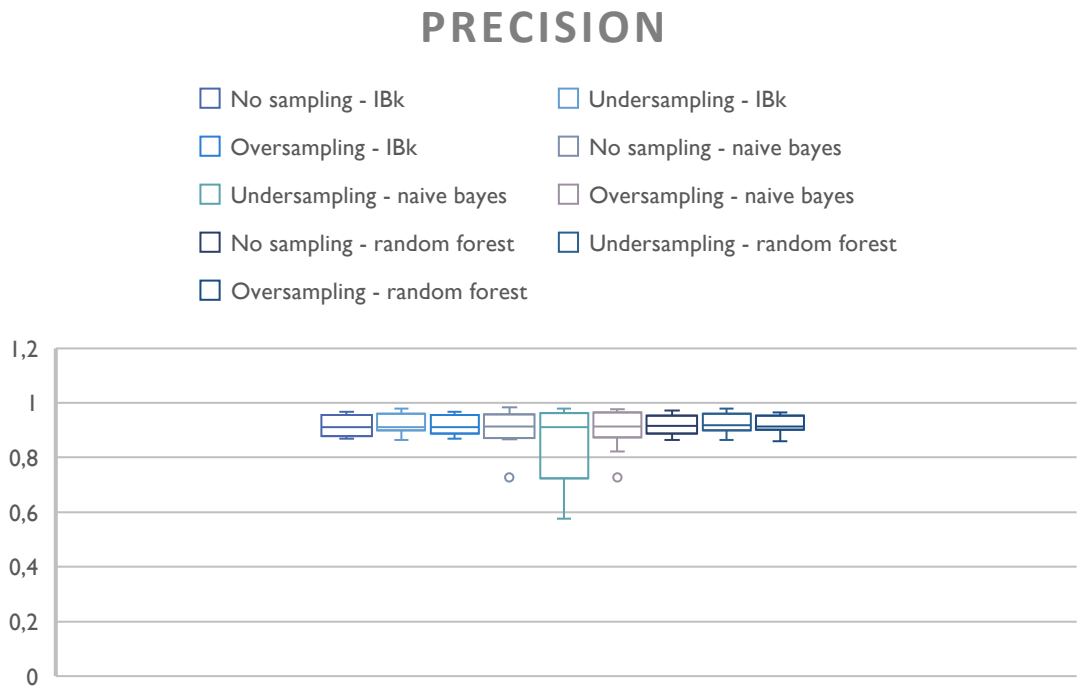
Syncope

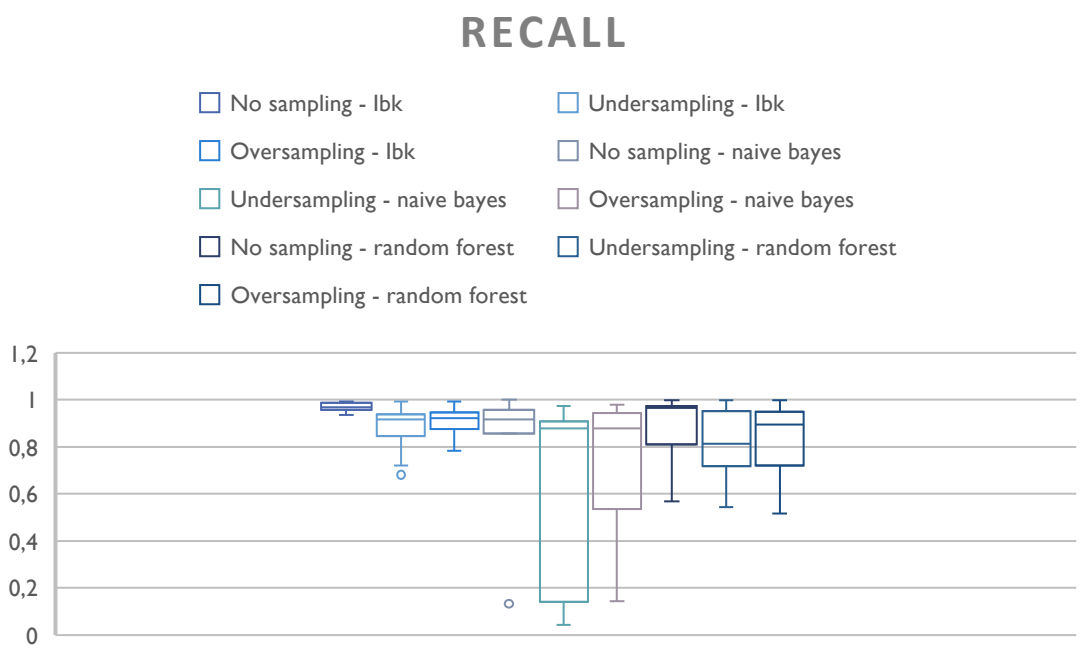
Per Syncope, la media di kappa è intorno a zero per ogni classificatore, con outliers pari a uno. Togliendo le istanze di test invalide si vede come le medie non migliorano ma si eliminano i valori di kappa = 1, abbastanza irreali da essere segno di una run che non è significativa. Naive Bayes è quello con i valori peggiori tra i classificatori, Random Forest, anche se mostra una media comunque molto bassa, sembra il migliore





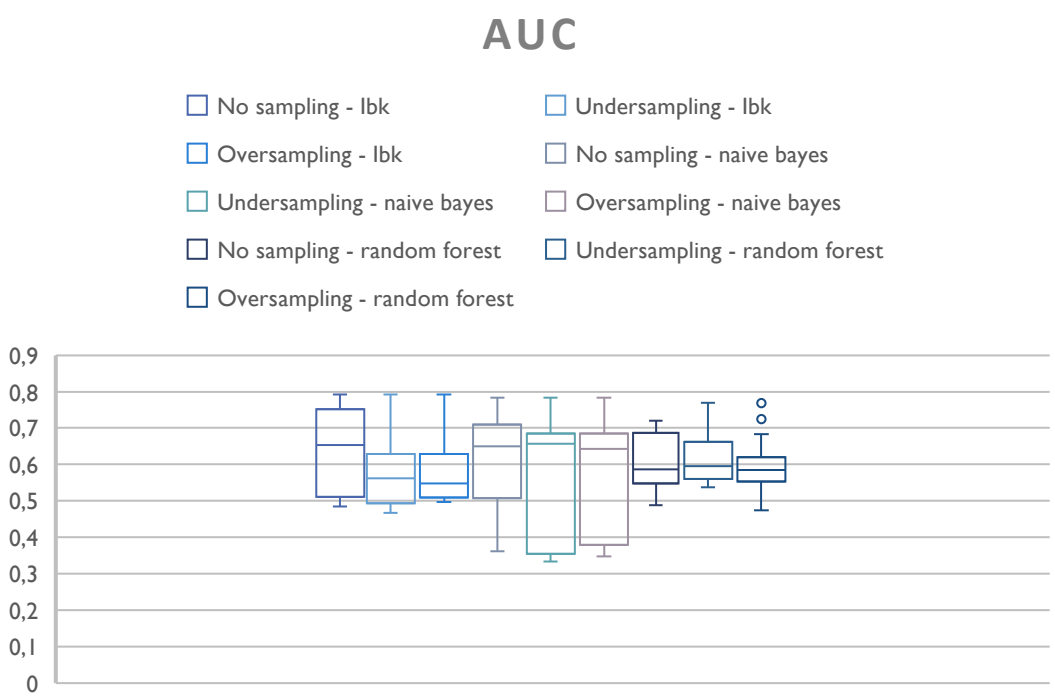
CONFRONTI SUL SAMPLING - 1

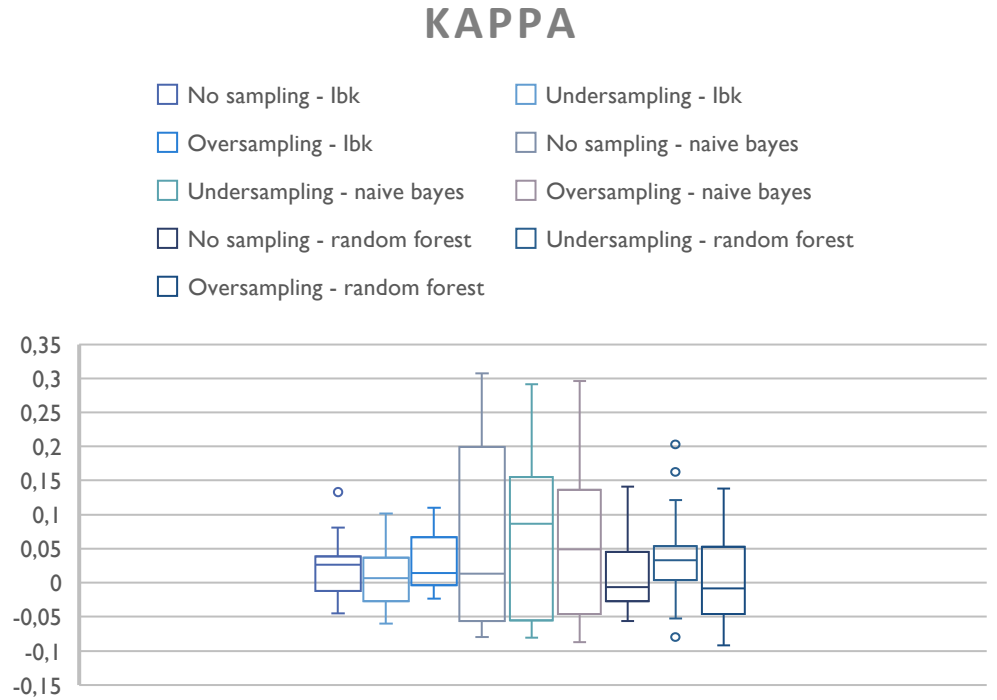




La precision media non sembra molto influenzata dal sampling, Naive Bayes acquista invece una varianza molto maggiore quando viene effettuato l'under sampling. La recall è molto più influenzata e risulta maggiore quando il sampling è assente, condizione che porta anche a valori meno variabili. Naive Bayes si dimostra nuovamente un classificatore peggiore degli altri due. Almeno per quanto riguarda queste due metriche, il sampling, contrariamente alle aspettative, non sembra vantaggioso.

CONFRONTI SUL SAMPLING - 2





Per quanto riguarda l'AUC, se per IBk evitare il sampling sembra una scelta migliore, lo stesso non si può dire di Naive Bayes e Random Forest, che migliorano leggermente con l'under sampling. Anche per la kappa si può dire la stessa cosa, si nota anche che in questo caso l'under sampling resta la scelta migliore ma l'over sampling risulta migliore della situazione dove il sampling non è previsto

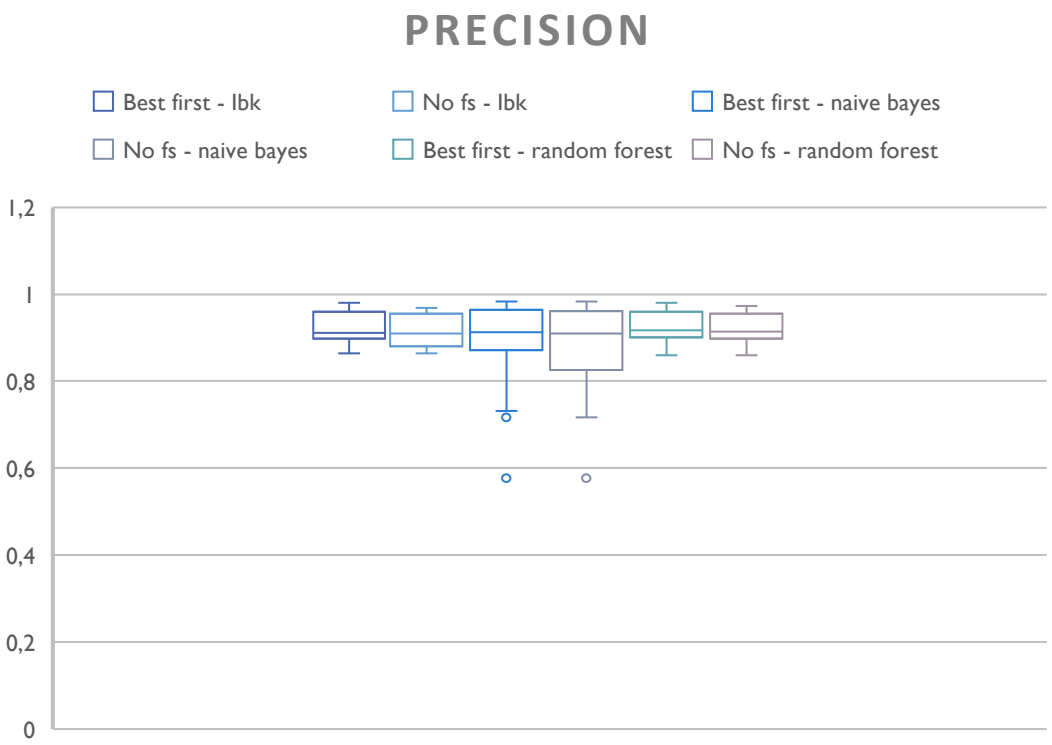
CONFRONTI SUL SAMPLING - 3

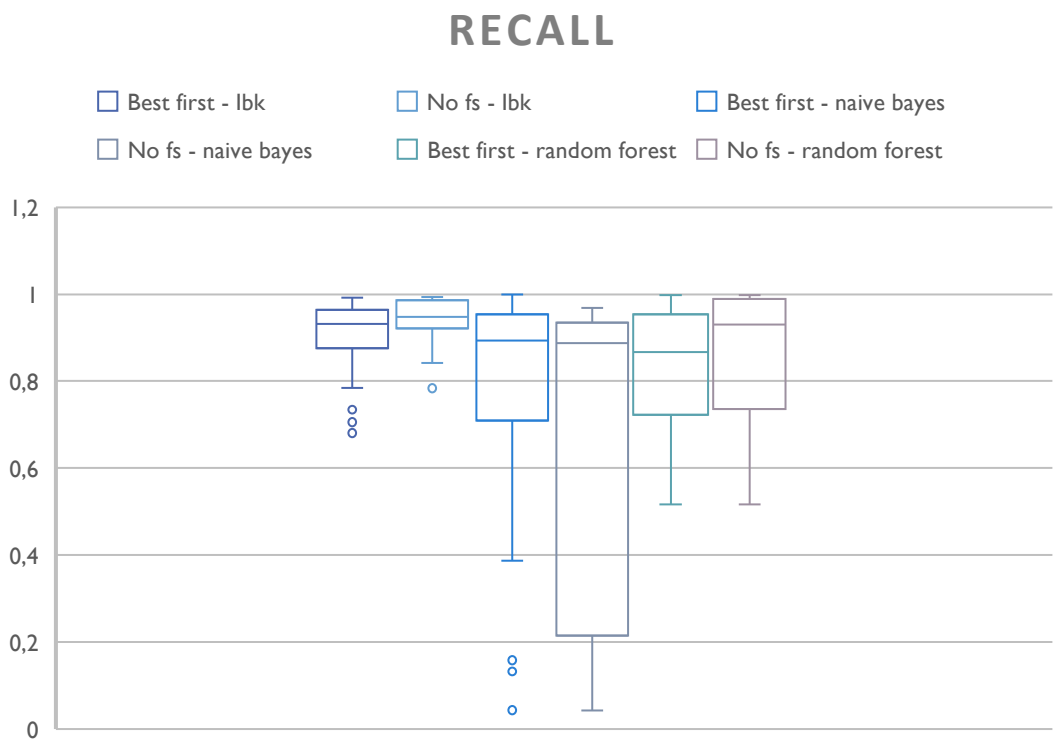
Valori medi per Syncope

Sampling	precision	recall	AUC	kappa
No sampling-ibk	0,83917033	0,99540671	0,52315021	-0,001747
Undersampling-ibk	0,84533624	0,8593525	0,53605817	0,01471267
Oversampling-ibk	0,83782914	0,97683743	0,53081376	-0,0090591
No sampling-naive bayes	0,8853376	0,98209129	0,56771455	0,11969669
Undersampling-naive bayes	0,78912583	0,73389536	0,49938348	0,00531195
Oversampling-naive bayes	0,80729369	0,74343731	0,52758919	0,01557033
No sampling-random forest	0,83909807	0,99435055	0,5244536	-0,0025044
Undersampling-random forest	0,86039595	0,79352919	0,5738569	0,04585457
Oversampling-random forest	0,83873714	0,97479164	0,5157129	-0,0053077

- I confronti precedenti sono stati effettuati con i dati di Bookkeeper.
- Era atteso un miglioramento di prestazioni con l'introduzione del sampling, si può invece affermare che l'over sampling non è una scelta vantaggiosa
- Per IBk il sampling va evitato del tutto, per Naive Bayes e Random Forest può essere sensato considerare l'under sampling, ma dipende comunque da quali sono le metriche di maggiore interesse
- I dati relativi a Syncope sembrano suggerire che per Naive Bayes il no sampling sia migliore, mentre per Random Forest l'under sampling porta a maggior accuratezza
- Sempre dai dati di Syncope, questa volta l'under sampling sembra essere una scelta suggerita anche per IBk, nonostante il no sampling resta quello a recall maggiore

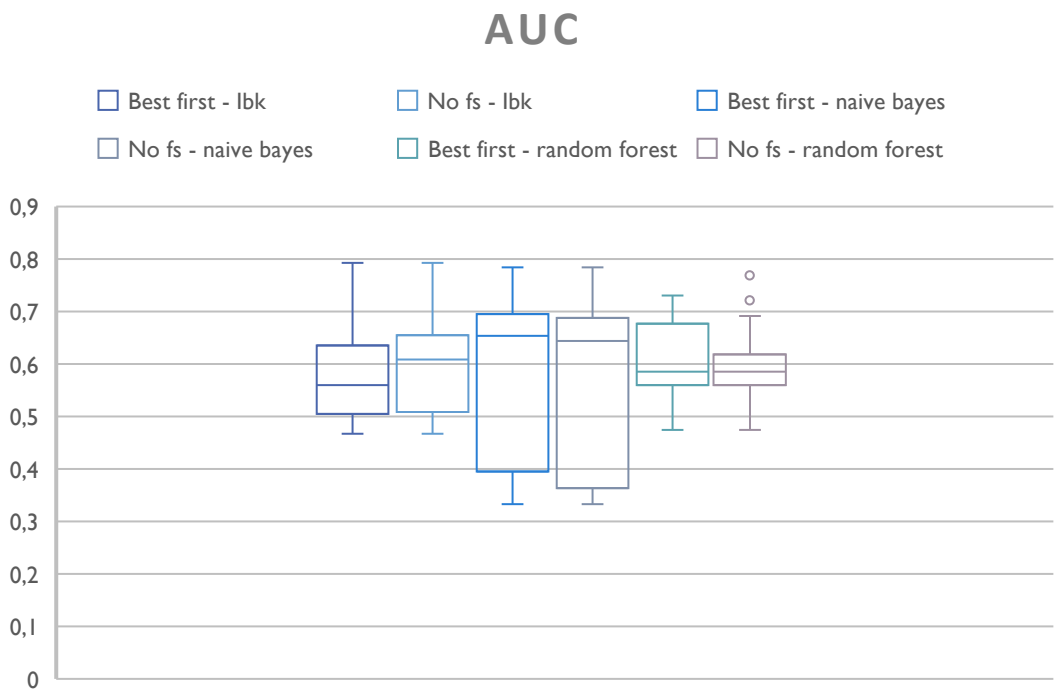
CONFRONTI SUL FEATURE SELECTION - I

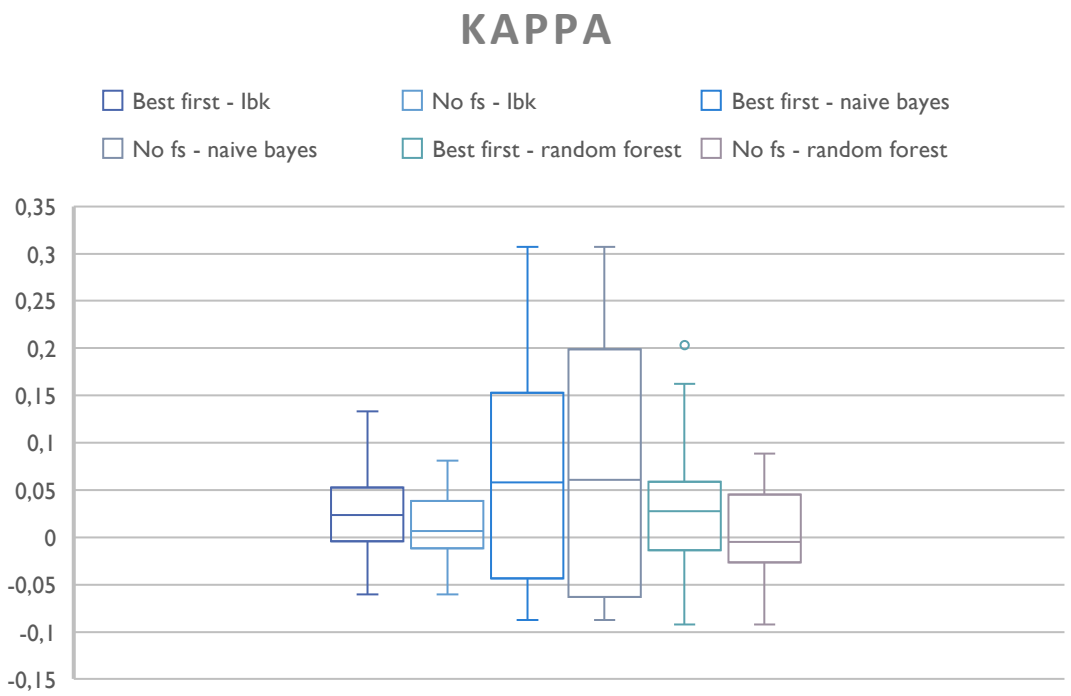




La precision media sembra migliorare con la feature selection rispetto alle esecuzioni senza feature selection, la recall peggiora significativamente per Random Forest e IBk mentre migliora per Naive Bayes. Il miglioramento è poco visibile per quanto riguarda la media, ma c'è un netto miglioramento per la varianza

CONFRONTI SUL FEATURE SELECTION- 2





La kappa sembra essere migliorata dal feature selection, in modo netto per IBk e Random Forest, meno per Naive Bayes la cui media peggiora leggermente ma che comunque vede un miglioramento per quanto riguarda la varianza. Per l'AUC avviene l'effetto opposto, IBk e Random Forest peggiorano con l'introduzione del best first, Naive Bayes ha un leggero miglioramento

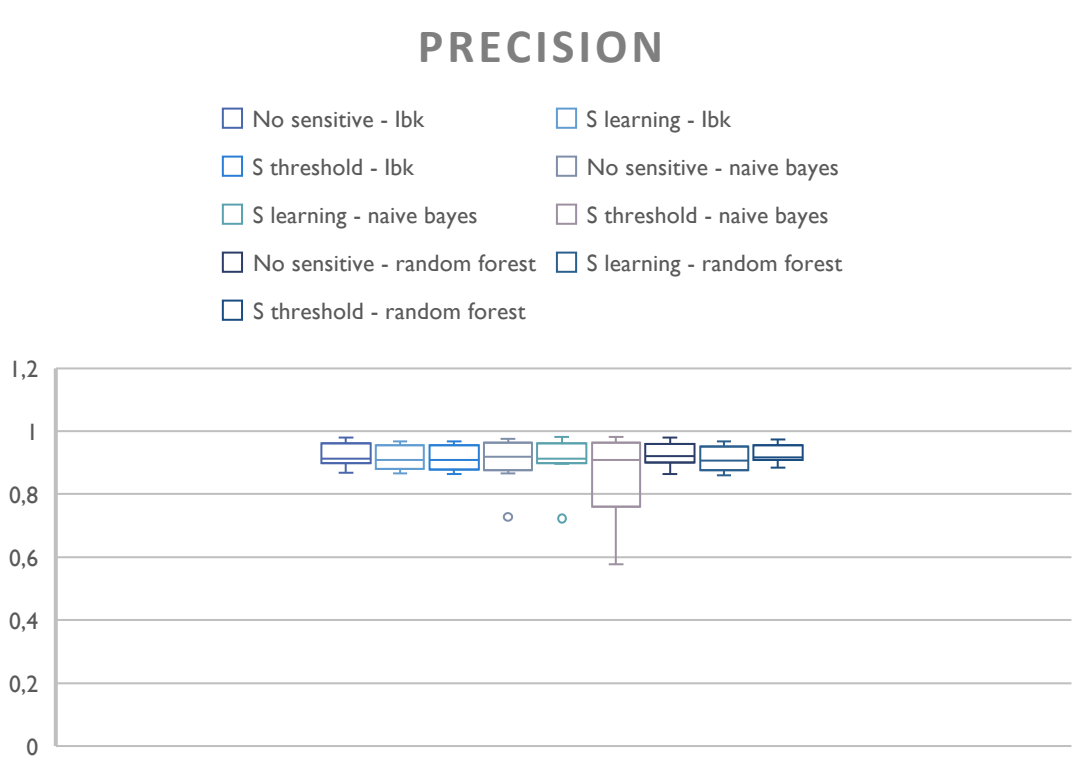
CONFRONTI SUL FEATURE SELECTION - 3

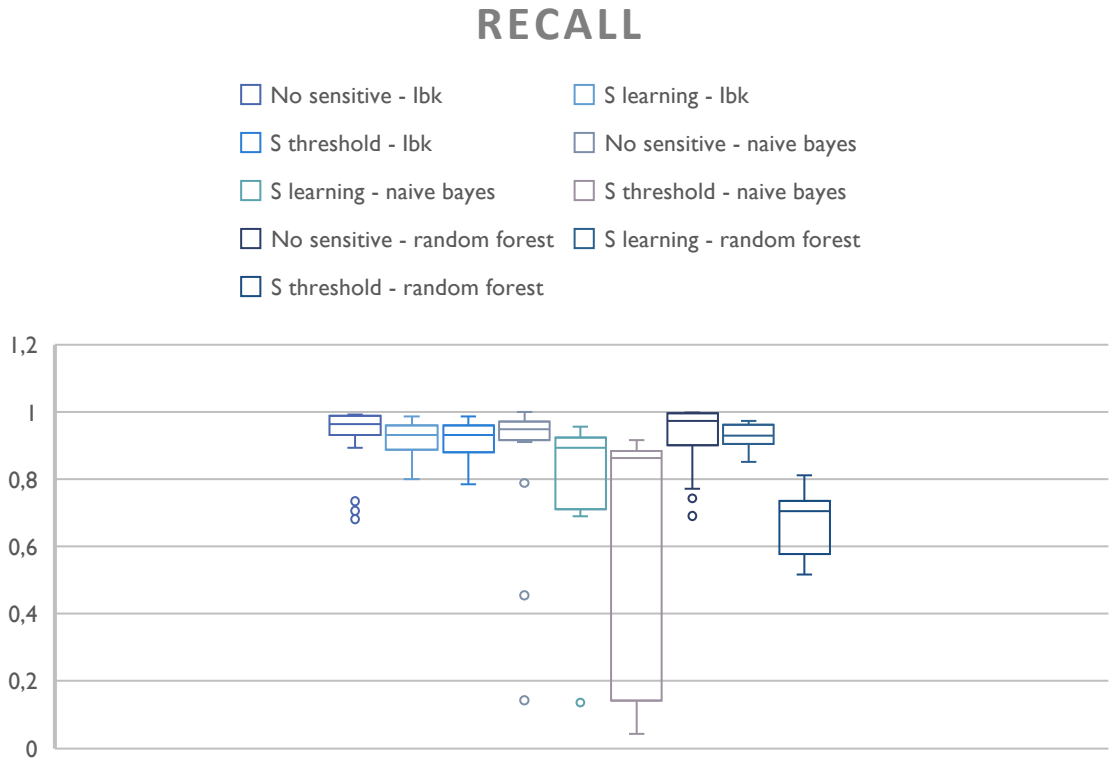
Valori medi per Syncope

Feature selection	precision	recall	AUC	kappa
Best first-ibk	0,84131978	0,9201321	0,52597806	0,00228594
No fs-ibk	0,84023737	0,967599	0,5340367	0,00079214
Best first-naive bayes	0,81271554	0,78770371	0,52025359	0,02423043
No fs-naive bayes	0,84178921	0,85191225	0,54287122	0,06948889
Best first-random forest	0,8467569	0,89741773	0,53504924	0,01385206
No fs-random forest	0,84539721	0,94436319	0,54096637	0,01150962

- I confronti precedenti sono stati effettuati con i dati di Bookkeeper.
- Il miglioramento di accuratezza atteso con l'introduzione del feature selection non è del tutto presente, alcune metriche migliorano altre no
- Naive Bayes è l'unico classificatore dove, secondo i dati di Bookkeeper, introdurre la feature selection è un miglioramento decisivo, ma i dati di Syncope mostrano esattamente l'opposto
- I dati relativi a Syncope confermano invece quanto già affermato per IBk e Random Forest, ovvero un miglioramento di precision e kappa e un peggioramento di AUC e recall
- L'inconsistenza relativa a Naive Bayes può essere dovuta al fatto che nella tabella consideriamo i valori medi per Syncope, non molto significativi per questo classificatore che ha una variabilità molto elevata

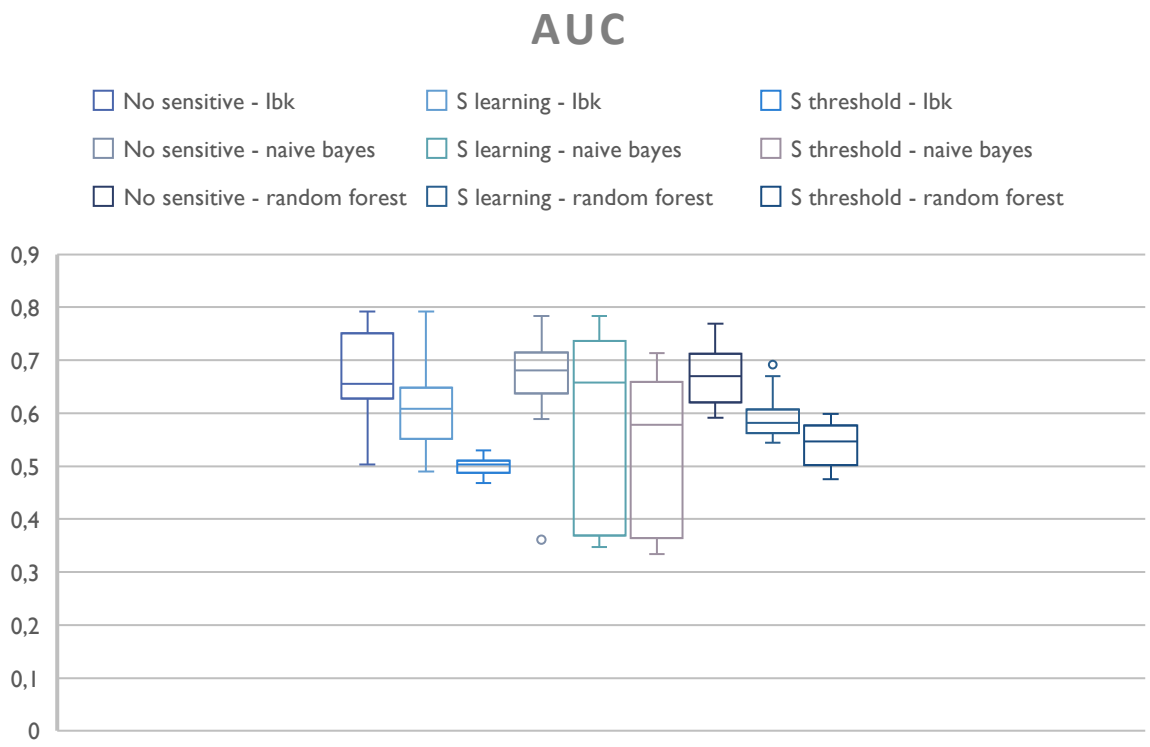
CONFRONTI SUL COST SENSITIVE- I

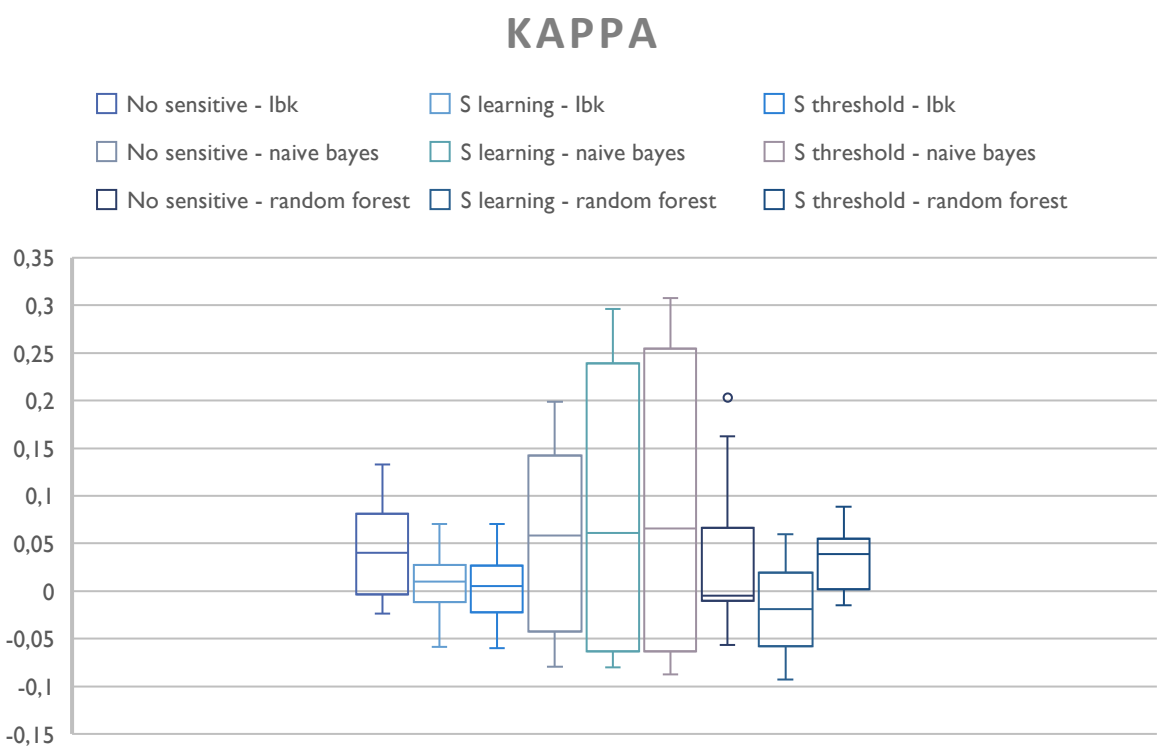




La precision non varia molto con l'introduzione del cost sensitive, tra le distribuzioni l'unica a differenziarsi significativamente sembra essere Naive Bayes con il sensitive threshold. La recall, come atteso, diminuisce, con entrambe le strategie di cost sensitivity, ma peggiora molto di più con il sensitive threshold rispetto al sensitive learning per ogni classificatore

CONFRONTI SUL COST SENSITIVE- 2





L'AUC si comporta in modo molto diverso rispetto alle aspettative, dato che non dovrebbe variare rispetto alla cost sensitivity, ma è la metrica che in realtà mostra più variazioni in assoluto, con i valori migliori senza cost sensitive e i peggiori con il sensitive threshold. La kappa sembra essere migliore senza cost sensitive per IBk, migliorare invece con il sensitive threshold per Naive Bayes e Random Forest

CONFRONTI SUL COST SENSITIVE - 3

Valori medi per Syncope

Sensitivity	precision	recall	AUC	kappa
No sensitive-ibk	0,84116348	0,92835536	0,53279862	0,00311626
S learning - Ibk	0,84071048	0,95269271	0,55581052	0,00125105
S threshold - Ibk	0,84046176	0,95054857	0,501413	0,00024981
No sensitive-naive bayes	0,83876921	0,88756233	0,54323426	0,05259402
S learning - naive bayes	0,844424	0,87068143	0,55142443	0,0642979
S threshold - naive bayes	0,7985639	0,70118019	0,50002852	0,02368705
No sensitive - random forest	0,84118983	0,92935967	0,53745188	0,00321781
S learning - random forest	0,8373949	0,95664962	0,55705838	-0,0069614
S threshold - random forest	0,85964643	0,8766621	0,51951314	0,0417861

- I confronti precedenti sono stati effettuati con i dati di Bookkeeper.
- Il peggioramento di alcune metriche introducendo il cost sensitive si è verificato come atteso, mentre il comportamento dell'AUC è diverso dalle aspettative
- Su Syncope, il sensitive learning non sembra cambiare molto l'AUC mentre il sensitive threshold la diminuisce di molto
- I dati di Syncope mostrano inconsistenze rispetto a quanto trovato da Bookkeeper per quanto riguarda la recall, che sembra avere i valori migliori per il sensitive learning
- IBk conferma avere la kappa migliore per il no sensitive, Random Forest per il sensitive threshold e Naive Bayes per il sensitive learning
- Il cost sensitive sembra essere il parametro per cui è più complicato selezionare una configurazione migliore delle altre ma dipende dal dataset e dalle metriche di maggiore interesse

CONCLUSIONI

- Non sembra esserci un classificatore nettamente migliore degli altri o una combinazione di filtri nettamente migliore
- Solitamente cambiando i filtri alcune metriche migliorano altre peggiorano, non c'è una soluzione unica ma dipende dallo scopo della predizione
- Naive Bayes è il classificatore che si comporta meglio quando ci sono molti dati, ma ha altissima variabilità dato che nelle run iniziali si ottengono valori bassissimi, Random Forest e IBk sono molto simili tra loro e più stabili
- Quello che si evince da questo studio è che il tipo di classificatore da usare dipende ancora molto dal dataset, e che data la scarsità di dati disponibili non è semplice ottenere risultati ottimi e con validità assoluta. Nel momento in cui si vanno a fare predizioni su uno specifico dataset è invece possibile fare un analisi per trovare il classificatore migliore in quella specifica situazione.
- Per Bookkeeper la mia scelta ricadrebbe su Naive Bayes con under sampling, no feature selection e no cost sensitive, per Syncope su IBk con no sampling, no feature selection e sensitive learning

LINK

- Sonarcloud: https://sonarcloud.io/project/overview?id=Sara-DaCanal_ISW2-Progetto_Falessi