

MACHINE LEARNING FOR SOFTWARE ENGINEERING

CORTESE ALESSANDRO - 0350035

Programma

- Contesto
- Obiettivi
- Assunzioni e Scelte
- Metriche Utilizzate
- Metodi di Valutazione
- Risultati e Valutazioni
- Link GitHub e SonarCloud

Contesto

- Oggigiorno, ogni progetto nell'ambito del Software Engineering richiede attività di software testing per far emergere e correggere eventuali bug. Tale attività è molto dispendiosa dal punto di vista delle risorse che si hanno a disposizione, è quindi necessario individuare un sottoinsieme di classi del progetto su cui poter concentrare le attività di testing.
- In che modo si possono individuare queste classi su cui concentrare i nostri sforzi per l'attività di testing?
- Per rispondere alla domanda, sarebbe ideale avere un quadro completo di quali classi sono state caratterizzate da bug andati in produzione nel passato e all'interno di quale release sono presenti queste classi.
- Conoscendo queste informazioni, è possibile sfruttare dei modelli di **Machine Learning** per poter predire quali siano le classi che possano contenere questi difetti.
- In questo modo, con l'utilizzo di particolari algoritmi detti **classificatori**, è possibile predire le classi su cui concentrare le attività di testing.

Obiettivi (1)

L'obiettivo è quello di analizzare due progetti di Apache Software Foundation, nello specifico BookKeeper e ZooKeeper, misurando l'accuratezza di tre classificatori usando differenti tecniche per migliorare la qualità delle predizioni, per poi stabilire come e quanto le tecniche utilizzate hanno impattato le predizioni sulle classi.

I classificatori utilizzati sono:

- Random Forest
- Naive Bayes
- IBK

Le tecniche utilizzate sono:

- Feature Selection
- Sampling
- Cost Sensitive

Obiettivi (2)

Per poter raggiungere gli obiettivi predisposti, bisogna prima considerare delle operazioni preliminari:

- Raccolte delle informazioni sulla storia passata dei progetti considerati.
- Costruzione del dataset e la scelta delle metriche utilizzate.
- Analisi e valutazione dei classificatori utilizzati per le predizioni.

Scelte ed Assunzioni (1)

Una delle prime scelte da fare riguarda come stabilire per quante versioni le classi sono state buggy: una classe è considerata affetta da bug nelle release che vanno dalla **injected version** (inclusa) fino alla **fixed version** (esclusa).



Scelte ed Assunzioni (2)

Le informazioni relativa a quali siano le opening version e la fixed version sono state ottenute tramite l'utilizzo di **Jira**, software specializzato nella gestione dei ticket.

Con l'utilizzo di **Jira** è sempre possibile accedere all'OV e alla FV, ma questo non vale per l'IV; infatti, l'istante di rilevazione ed eliminazione dei bug sono sempre noti, ma ciò non vale con l'istante di introduzione dei bug.

Come fare in questi casi?

Scelte ed Assunzioni (3)

Per rispondere alla domanda precedente, utilizziamo **Proportion**.

Proportion

Proportion è una tecnica utilizzata per poter stimare l'IV di un bug di cui non si sa quale sia la release in cui tale difetto è stato introdotto.

L'intuizione alla base di proportion è che la distanza in release tra OV e FV è **proporzionale** alla distanza tra la IV e la FV. In questo modo è possibile calcolare il valore di **Proportion** p per tutti quei bug di cui è nota l'IV, per poi predire l'IV in quei casi dove non è disponibile proprio grazie al valore di p.

$$p = (FV - IV) / (FV - OV)$$

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

Scelte ed Assunzioni (4)

- Nel caso in cui i bug con IV disponibile fossero meno di 5, si procede con il **cold start**, che consiste nell'utilizzare i bug di altri progetti di Apache per il calcolo di Proportion.
- Un altro problema da non sottovalutare è il problema dello **Snoring**: un bug, dopo essere introdotto, può rimanere “dormiente” per molto tempo; questo determina che le ultime release siano piene di bug dormienti e quindi i relativi dataset sulle classi buggy non siano affidabili.
- La soluzione attuata è quella di **scartare** la seconda metà delle release così da avere una elevata affidabilità sui dati.

Metriche Considerate (1)

- **Size**: Numero di linee di codice della classe;
- **LOC_Added**: Numero di linee di codice aggiunte alla classe durante la revisioni di questa release;
- **AVG_LOC_Added**: Numero medio di linee di codice aggiunte alla classe;
- **LOC_deleted**: Numero di linee di codice rimosse alla classe durante la revisione della classe;
- **AVG_LOC_Deleted**: Numero medio di linee di codice rimosse dalla classe;
- **Churn**: Somma dei moduli delle differenze tra le LOC aggiunte e quelle rimosse in una classe nella release considerata;
- **AVG_Churn**: Valore medio dei churn sulle revisioni relative alla release;

Metriche Considerate (2)

- **Fixed_Defects**: Numero di difetti risolti in una classe durante la release considerata;
- **NUMBER_OF_COMMITS**: Numero di commit all'interno di questa release considerata;
- **NUMBER_OF_AUTHORS**: Numero di sviluppatori che hanno toccato quella specifica classe considerata.

Metodi di Valutazione (1)

Durante l'utilizzo di uno dei classificatori scelti, il dataset viene diviso in porzioni differenti, dette **Training Set** e **Testing Set**: il primo rappresenta il set con cui viene addestrato il classificatore, mentre il secondo rappresenta i dati su cui fare le predizioni e verificare i risultati.

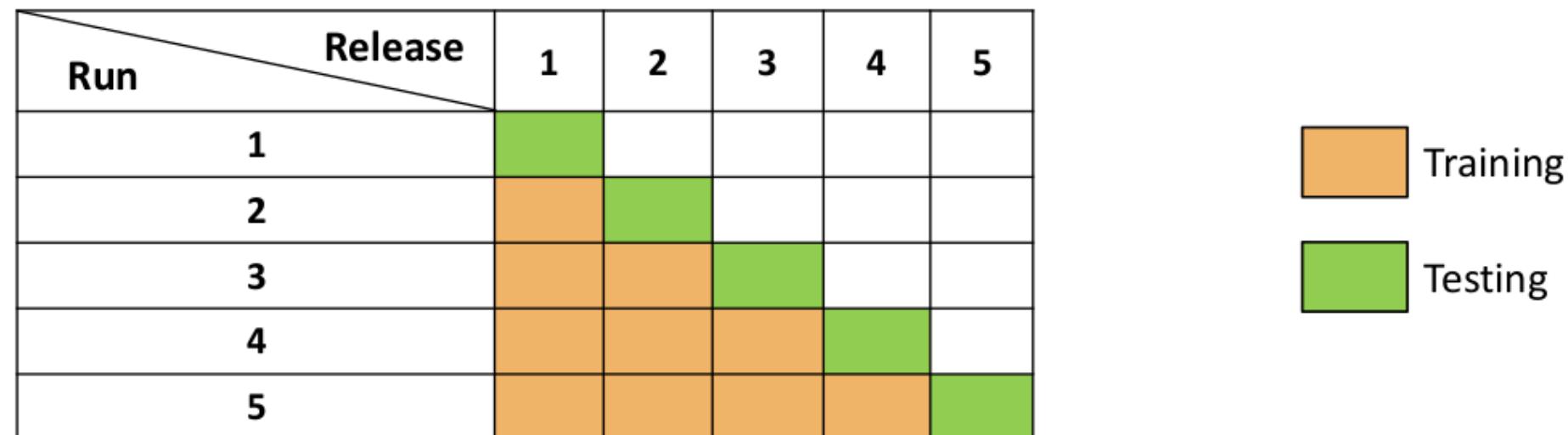
Il **Training Set** deve essere il più fedele possibile all'utilizzo reale del classificatore, può non sfruttare i dati del passato ed è affetto da snoring.

Il **Testing Set** deve essere il più fedele possibile alla realtà, idealmente potrebbe essere anche tramite un oracolo, deve sfruttare tutti i dati che ha a disposizione e non è affetto da snoring.

Metodi di Valutazione (2)

Uno degli obiettivi è quello di valutare le prestazioni di un classificatore e con quale tecnica di utilizzo.

Per la valutazione dei classificatori è stata usata la tecnica basata su times series, **Walk Forward**, che quindi tiene conto dell'ordine temporale dei dati ed è quindi impensabile che all'interno del training set vengano sfruttate informazioni future rispetto al training set stesso.



Risultati

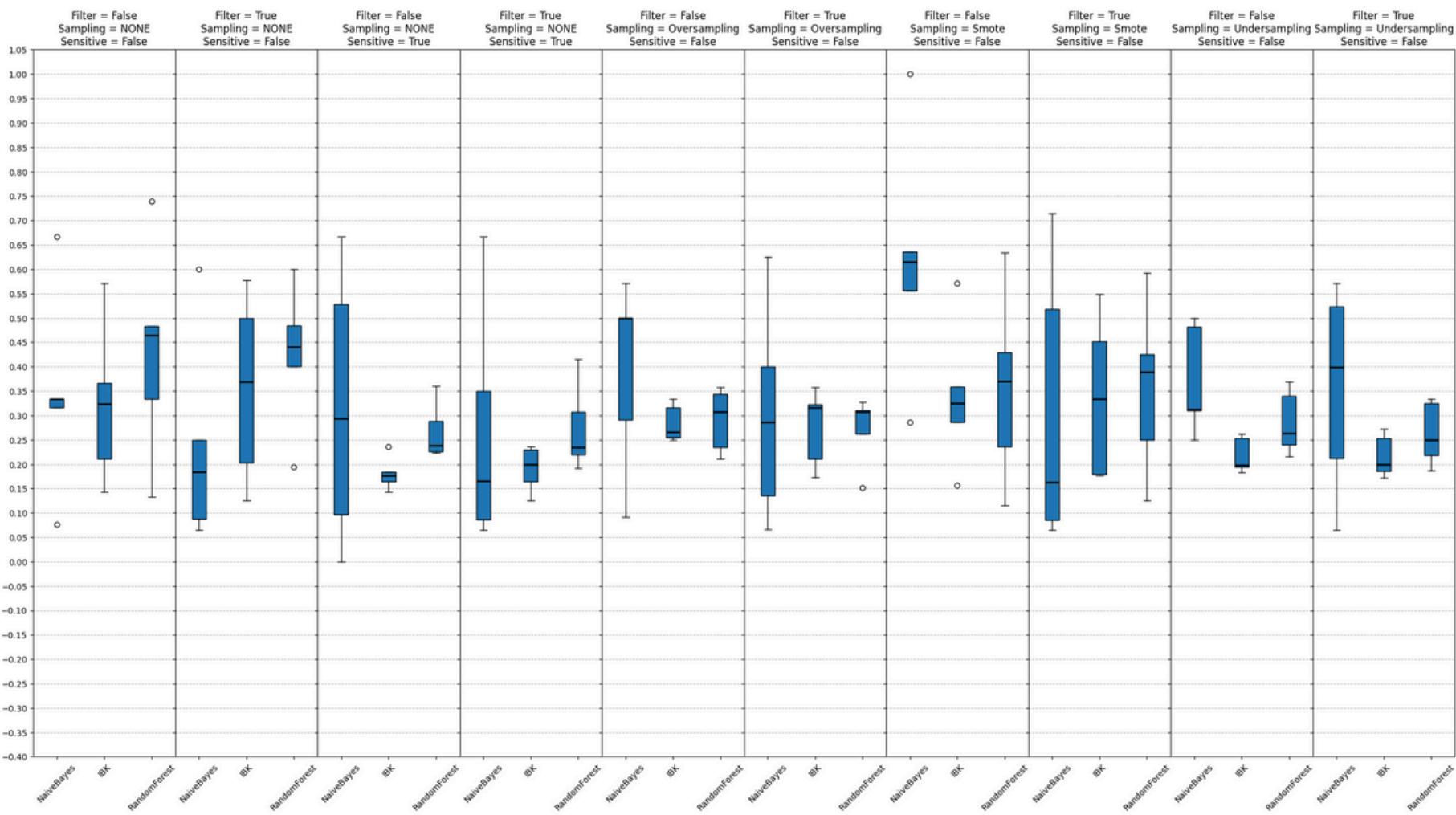
I risultati seguenti paragonano i valori di **Precision**, **Recall**, **KAPPA** e **AUC** ottenuti dai vari classificatori usando diverse tecniche. In questo modo, le differenze risultano più evidenti, in particolar modo quelle relative alle tecniche utilizzate.

Ricordando che i classificatori che vengono utilizzati sono **Random Forest**, **Naïve Bayes** e **IBK**, mentre per ognuna delle tecniche utilizzate sono state considerate alcune particolari tipologie:

- **Feature Selection**: Best First Backword
- **Sampling**: Undersampling, Oversampling e SMOTE
- **Cost Sensitive**: Sensitive Learning

Risultati - BookKeeper

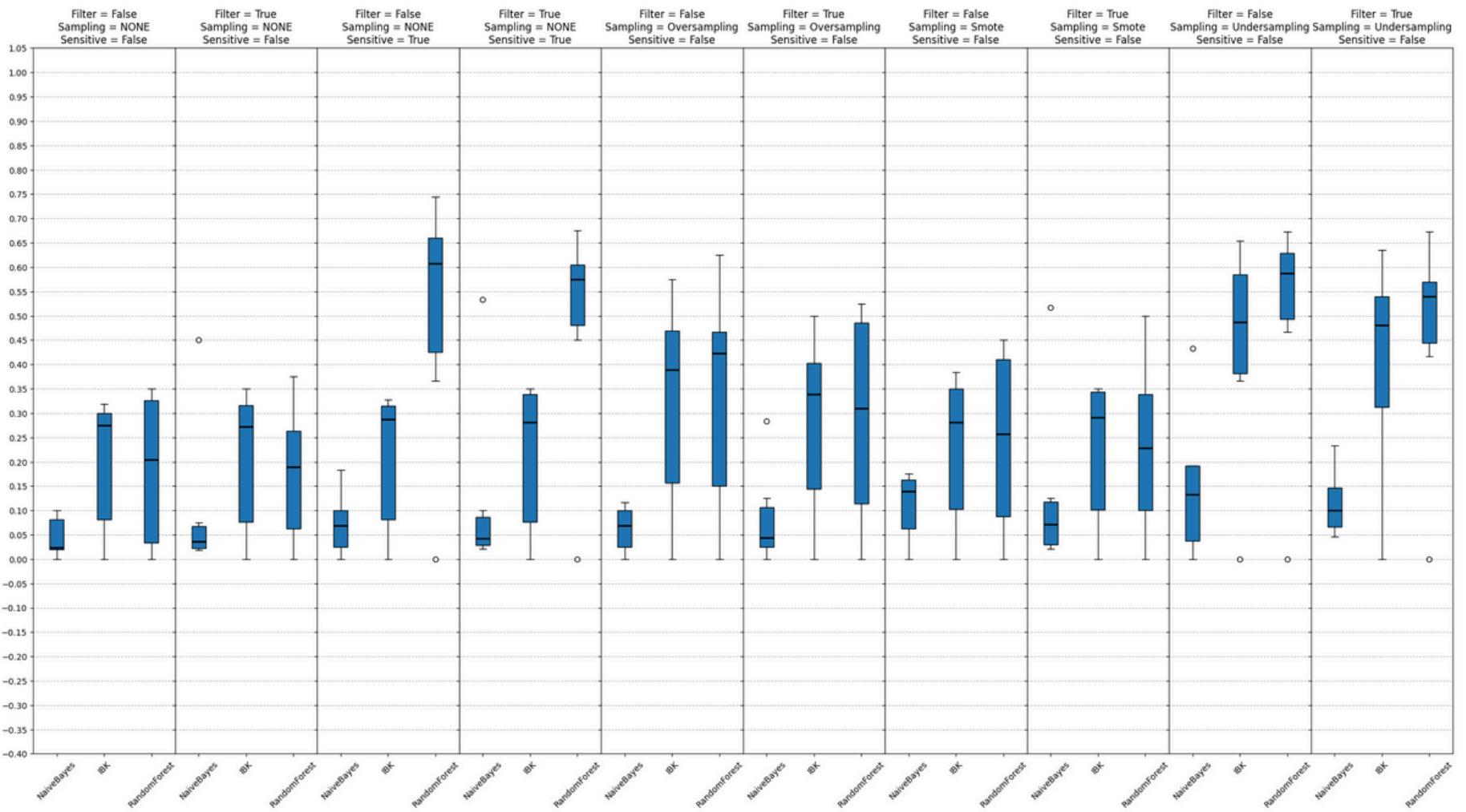
Precision



- **Feature Selection:** Applicando Feature Selection, migliora il valore della Precision nel caso del classificatore IBK, mentre negli altri casi in generale peggiora.
- **Oversampling:** Applicando solo Oversampling migliora solo il valore del classificatore Naive Bayes, mentre gli altri valori di Precision degli altri due classificatori peggiorano; in combinazione con il Feature Selection, invece, i valori peggiorano per tutti i classificatori.
- **Smote:** Applicando solamente Smote si ottiene un notevole miglioramento del valore della Precision per Naive Bayes, mentre gli altri due classificatori peggiorano. Con la combinazione dell'applicazione del Feature Selection peggiorano tutti i classificatori considerati.
- **Undersampling:** Applicando Undersampling i valori della Precision peggiorano per tutti i classificatori considerati. Con la combinazione dell'applicazione del Feature Selection si ottiene un piccolo miglioramento del valore della Precision per il classificatore Naive Bayes, mentre per gli altri classificatori peggiora.
- **Sensitive Learning:** Con l'applicazione del Sensitive Learning i valori di Precision peggiorano, sia nel caso in cui venga applicato che non venga applicato Feature Selection.

Risultati - BookKeeper

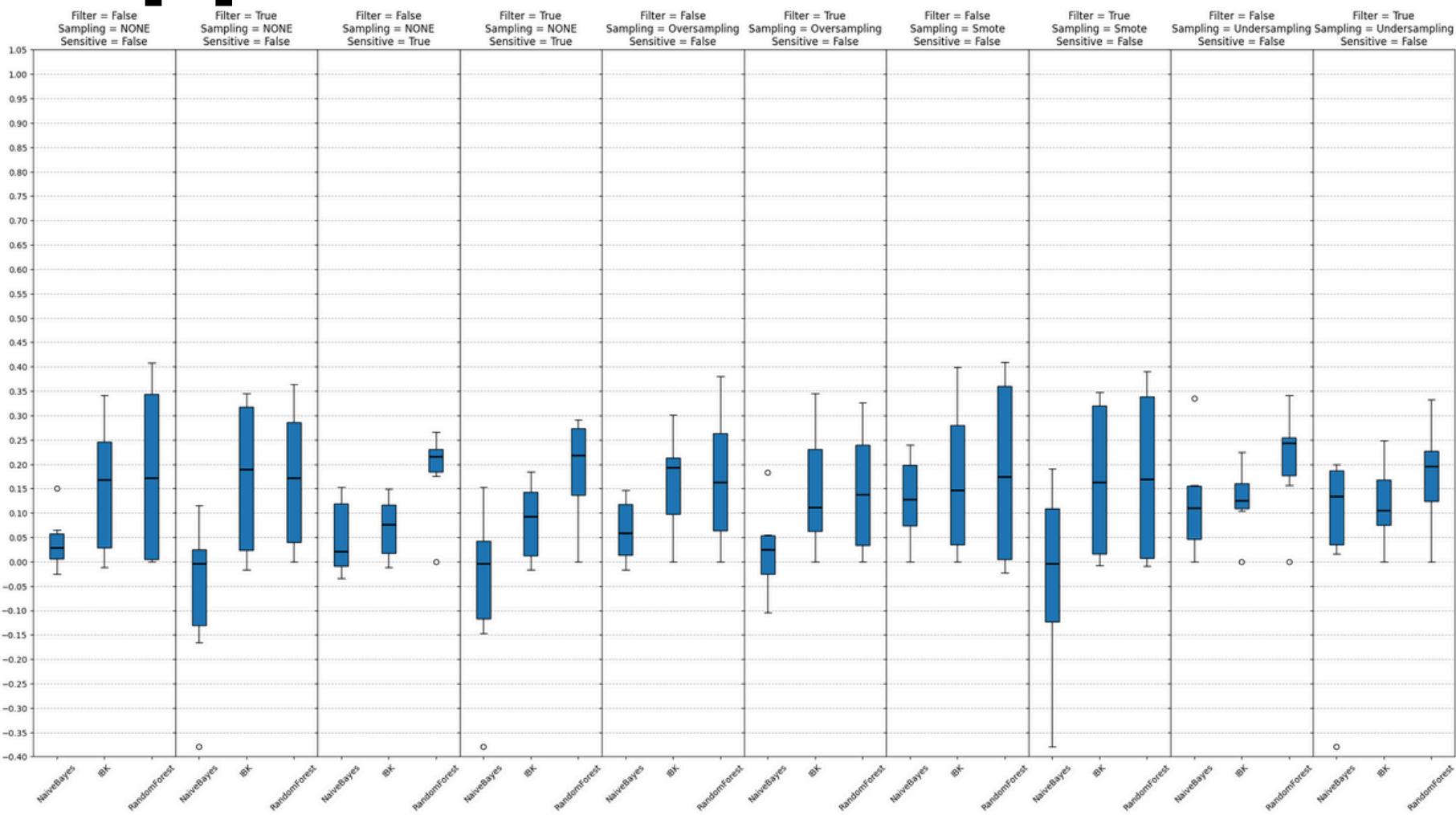
Recall



- **Feature Selection:** Con la sola applicazione del Feature Selection non si hanno miglioramenti nei valori della Recall. Con la combinazione del Sensitive Learning si ottengono notevoli miglioramenti per tutti i classificatori considerati.
- **Oversampling:** Applicando Oversampling, migliorano i valori di Recall per tutti i classificatori considerati, anche con la combinazione del Feature Selection ma i miglioramenti più grandi si ottengono con la sola applicazione di Oversampling.
- **Smote:** Con la sola applicazione di Smote, l'unico valore di Recall che migliora è Naive Bayes mentre per gli altri classificatori si ha un peggioramento. Lo stesso comportamento lo si ottiene combinandolo con il Feature Selection.
- **Undersampling:** Applicando Undersampling si hanno dei miglioramenti dei valori di Recall per tutti i classificatori, sia nel caso in cui venga combinato con Feature Selection sia nel caso in cui non venga combinato. I miglioramenti più grandi si ottengono con la sola applicazione di Undersampling.
- **Sensitive Learning:** Con l'applicazione del Sensitive Learning si ottengono miglioramenti nei valori della Recall per tutti i classificatori, sia con che senza la combinazione di Feature Selection. I miglioramenti significativi si ottengono con la sola applicazione del Sensitive Learning.

Risultati - BookKeeper

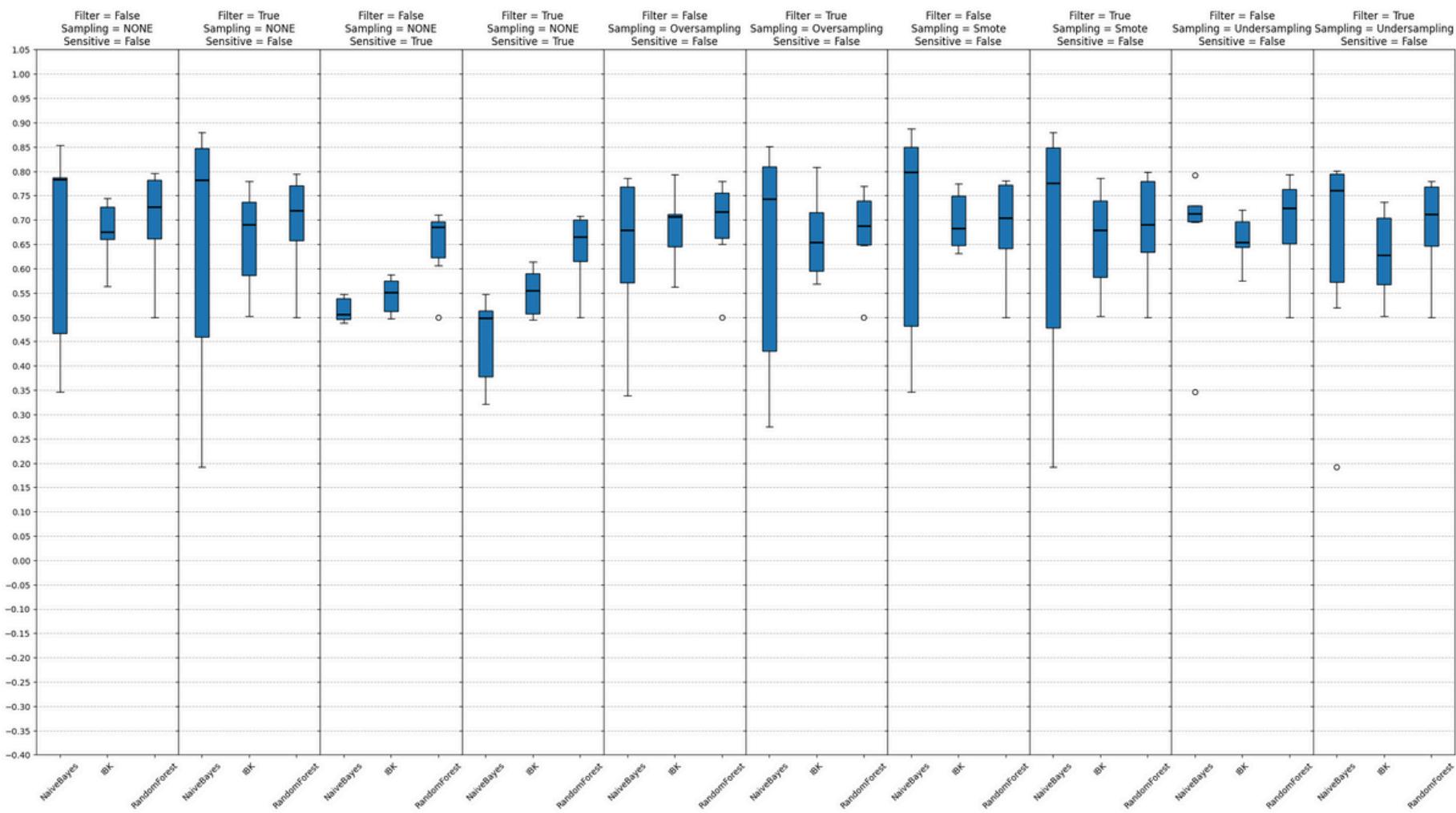
Kappa



- **Feature Selection:** Con la sola applicazione di Feature Selection non si ottengono notevoli miglioramenti per Random Forest e IBK, mentre nel caso di Naive Bayes peggiora il valore di Kappa. Con la combinazione del Sensitive Learning si nota un miglioramento solo per Random Forest di cui si ottiene il valore più elevato, mentre per gli altri classificatori peggiorano.
- **Oversampling:** Con la sola applicazione di Oversampling l'unico classificatore che peggiora è Random Forest, mentre gli altri classificatori migliorano. Con la combinazione del Feature Selection tutti e tre i classificatori peggiorano.
- **Smote:** Con la sola applicazione di Smote l'unico classificatore che migliora è Naive Bayes, mentre gli altri classificatori peggiorano. Con la combinazione del Feature Selection si ottiene un netto peggioramento per tutti e tre i classificatori considerati.
- **Undersampling:** Con la sola applicazione di Undersampling migliorano Naive Bayes e Random Forest, mentre IBK peggiora. Con la combinazione dell'applicazione di Feature Selection migliora solo Naive Bayes, mentre gli altri due classificatori peggiorano.
- **Sensitive Learning:** Con la sola applicazione di Sensitive Learning, l'unico classificatore che migliora è Random Forest, mentre gli altri due classificatori peggiorano. Questo comportamento lo si ottiene sia con che senza la combinazione di Feature Selection.

Risultati - BookKeeper

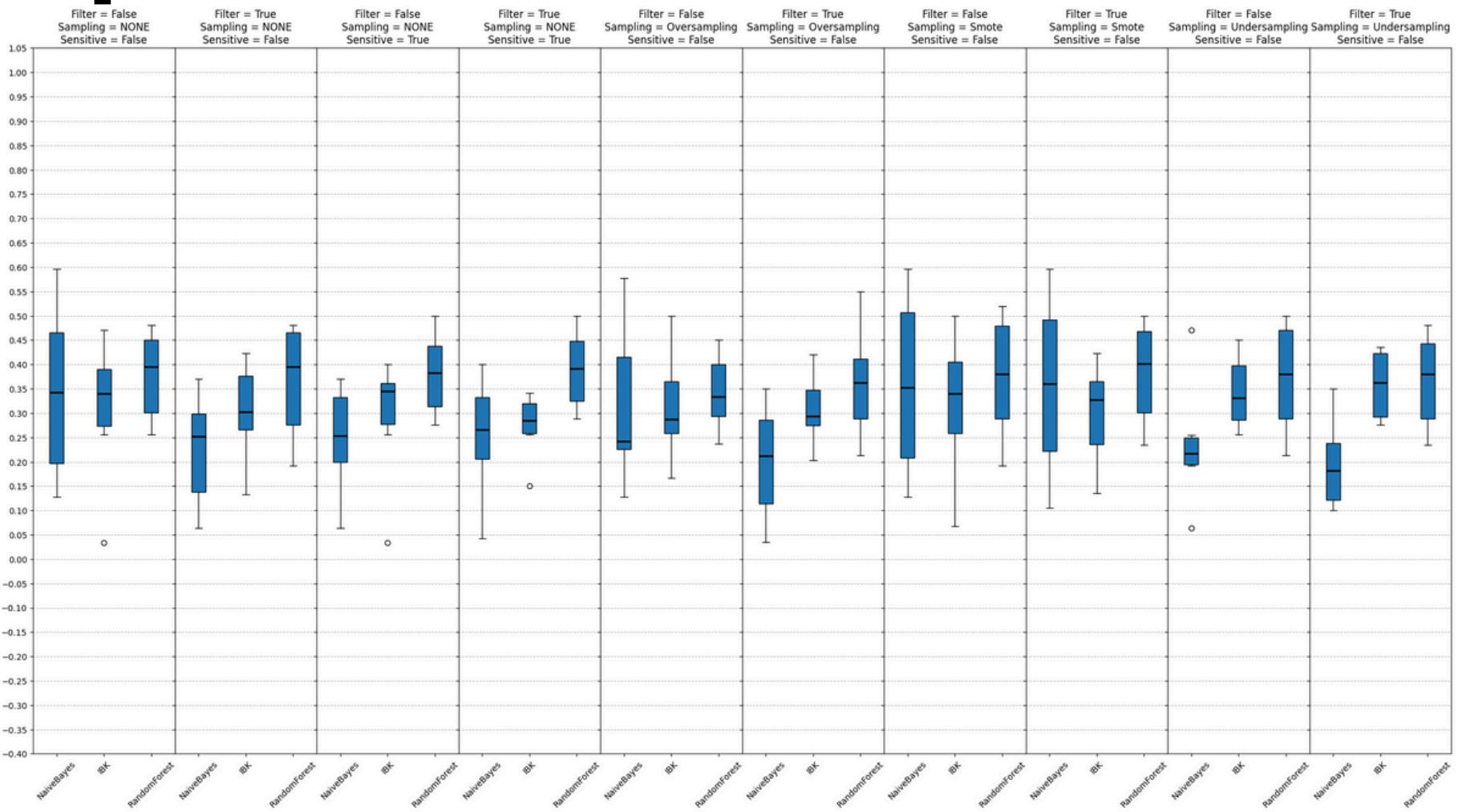
AUC



- **Feature Selection:** Con la sola applicazione di Feature Selection l'unico classificatore che migliora leggermente è Naive Bayes, mentre gli altri classificatori peggiorano. Con la combinazione dell'applicazione del Sensitive Learning si ha un peggioramento dei valori di AUC per tutti e tre i classificatori.
- **Oversampling:** Con la sola applicazione di Oversampling, l'unico classificatore che migliora è IBK, mentre gli altri classificatori peggiorano. Con la combinazione di Oversampling e Feature Selection peggiorano tutti e tre i classificatori.
- **Smote:** Con l'applicazione di Smote, l'unico classificatore che migliora è Naive Bayes, mentre gli altri classificatori peggiorano. Applicando sia Smote che Feature Selection peggiorano tutti i classificatori considerati.
- **Undersampling:** Nel caso dell'applicazione di Undersampling, sia in combinazione con il Feature Selection che non, si peggiora il valore di AUC per tutti i classificatori.
- **Sensitive Learning:** Con l'applicazione del Sensitive Learning, sia con la combinazione di Feature Selection che non, si ha un peggioramento del valore di AUC per tutti i classificatori considerati.

Risultati - BookKeeper

NpofB20



- **Feature Selection:** Applicando solo Feature Selection migliora il valore di NpofB20 nel caso di Naive Bayes e IBK, mentre nel caso Random Forest non si hanno variazioni. Combinando la Feature Selection con il Sensitive Learning si hanno miglioramenti per Naive Bayes e Random Forst, mentre IBK peggiora.
- **Oversampling:** Con la sola applicazione di Oversampling, si ottengono dei miglioramenti per tutti i classificatori. Lo stesso comportamento lo si nota anche con la combinando Feature Selection.
- **Smote:** Applicando solamente Smote, si ottengono dei piccoli peggioramenti per Naive Bayes, mentre per IBK e Random Forest si verifica un lieve miglioramento. Nel caso in cui si combini Smote con Feature Selection si ottengono peggioramenti per Naive Bayes e Random Forest, mentre si ottiene un lieve miglioramento per IBK.
- **Undersampling:** Nel caso della sola applicazione di Undersampling si ottengono miglioramenti del valore di NPofB20 in tutti i classificatori. Nel caso in cui si utilizzasse anche Feature Selection si notano miglioramenti per Naive Bayes e Random Forest, mentre un peggioramento nel caso di IBK.
- **Sensitive Learning:** Nel caso della sola applicazione del Sensitive Learning si hanno miglioramenti nel caso di Naive Bayes e Random Forest, mentre nel caso di IBK si notano peggioramenti. Nella combinazione di Feature Selection e Sensitive Learning si ottengono miglioramenti per tutti i classificatori considerati.

Risultati - BookKeeper

Valutazione

Come era prevedibile, non c'è un classificatore nettamente migliore degli altri considerando tutte le tecniche che si hanno a disposizione.

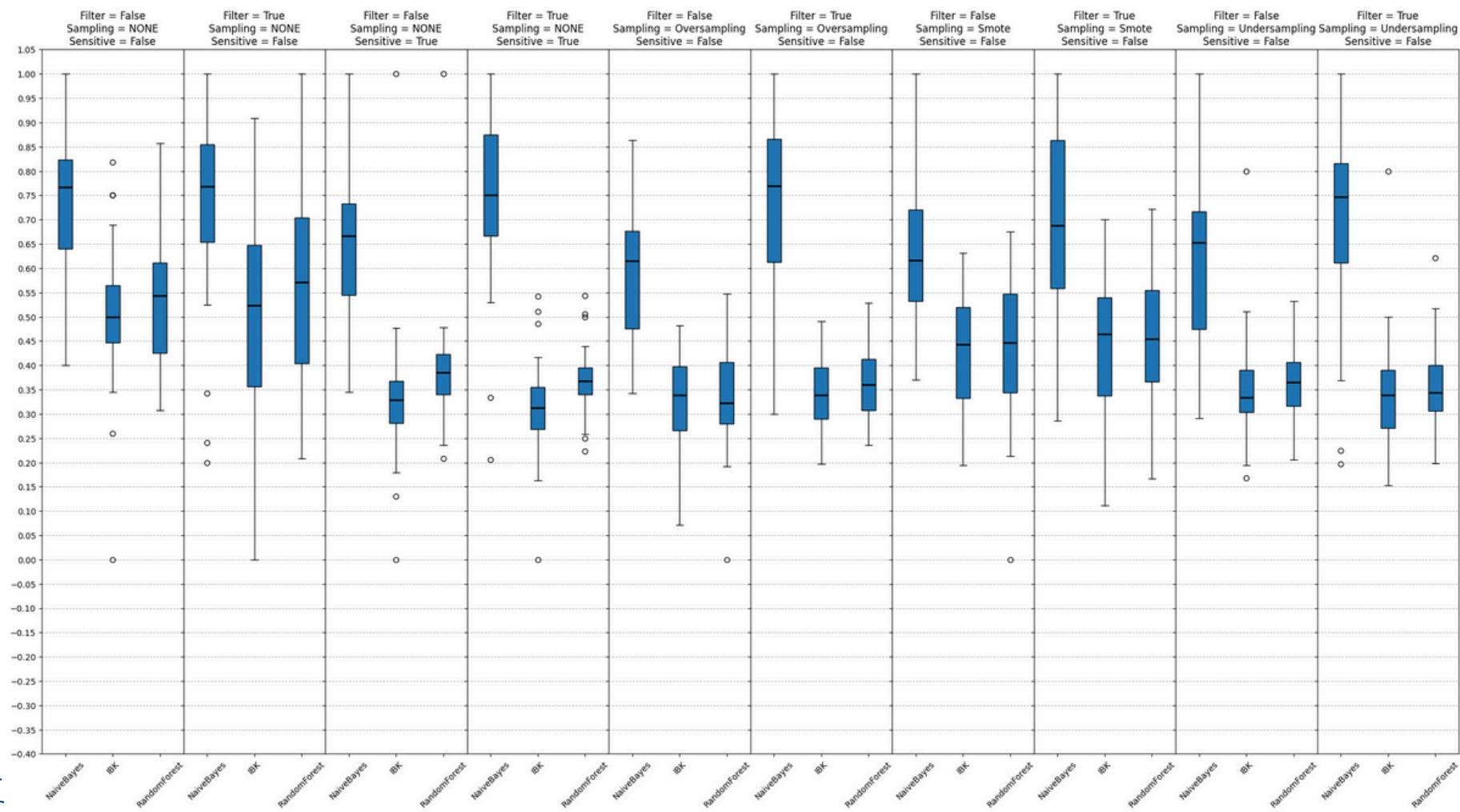
Si è visto come per una determinata tecnica un classificatore assuma valori migliori rispetto ad altri, mentre peggiori su altre tecniche rispetto ad altri classificatori.

Per questo motivo non c'è una scelta assoluta per la coppia composta da classificatore tecnica utilizzata.

Tuttavia, se si dovesse scegliere una classificatore e una tecnica da utilizzare, la coppia **Naive Bayes** e **Smote**, senza utilizzare Feature Selection e Sensitive Learning, raggiungono un giusto compromesso per i valori di Precision, Kappa e AUC.

Risultati - ZooKeeper

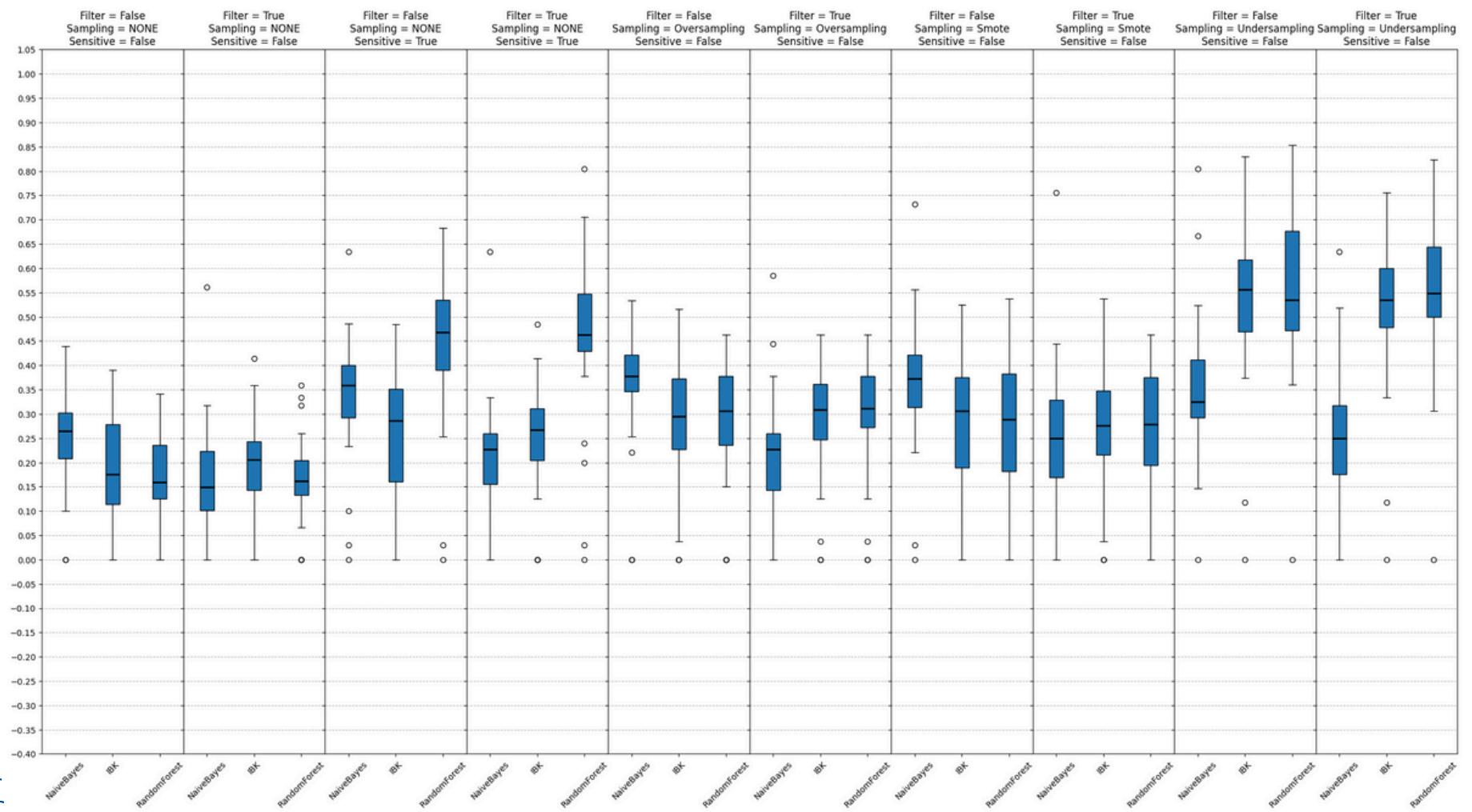
Precision



- **Feature Selection:** Applicando Feature Selection si ottengono miglioramenti per il valore della Recall per tutti i classificatori considerati. Applicando il Sensitive Learning in combinazione con Feature Selection si ottiene un lieve peggioramento per il valore della Recall per Naive Bayes, mentre si ottengono notevoli peggioramenti per IBK e Random Forest.
- **Oversampling:** Applicando solamente Oversampling si ottengono peggioramenti per il valore della Precision per tutti i classificatori considerati. Applicando sia Oversampling che Feature Selection nel caso di Naive Bayes non si ottengono né miglioramenti né peggioramenti, mentre IBK e Random Forest peggiorano drasticamente.
- **Smote:** Applicando Smote, sia con che senza Feature Selection, si ottengono valori peggiori per la Precision per tutti i classificatori considerati.
- **Undersampling:** Applicando Undersampling, sia con che senza Feature Selection, si ottengono valori peggiori per tutti i classificatori considerati.
- **Sensitive Learning:** Applicando solamente il Sensitive Learning si ottengono valori peggiori per la Precision per tutti i classificatori considerati. Combinando l'utilizzo di Sensitive Learning e Feature Selection si ha un lieve peggioramento per Naive Bayes, mentre per IBK Random Forest si nota di più il peggioramento del valore della Precision.

Risultati - ZooKeeper

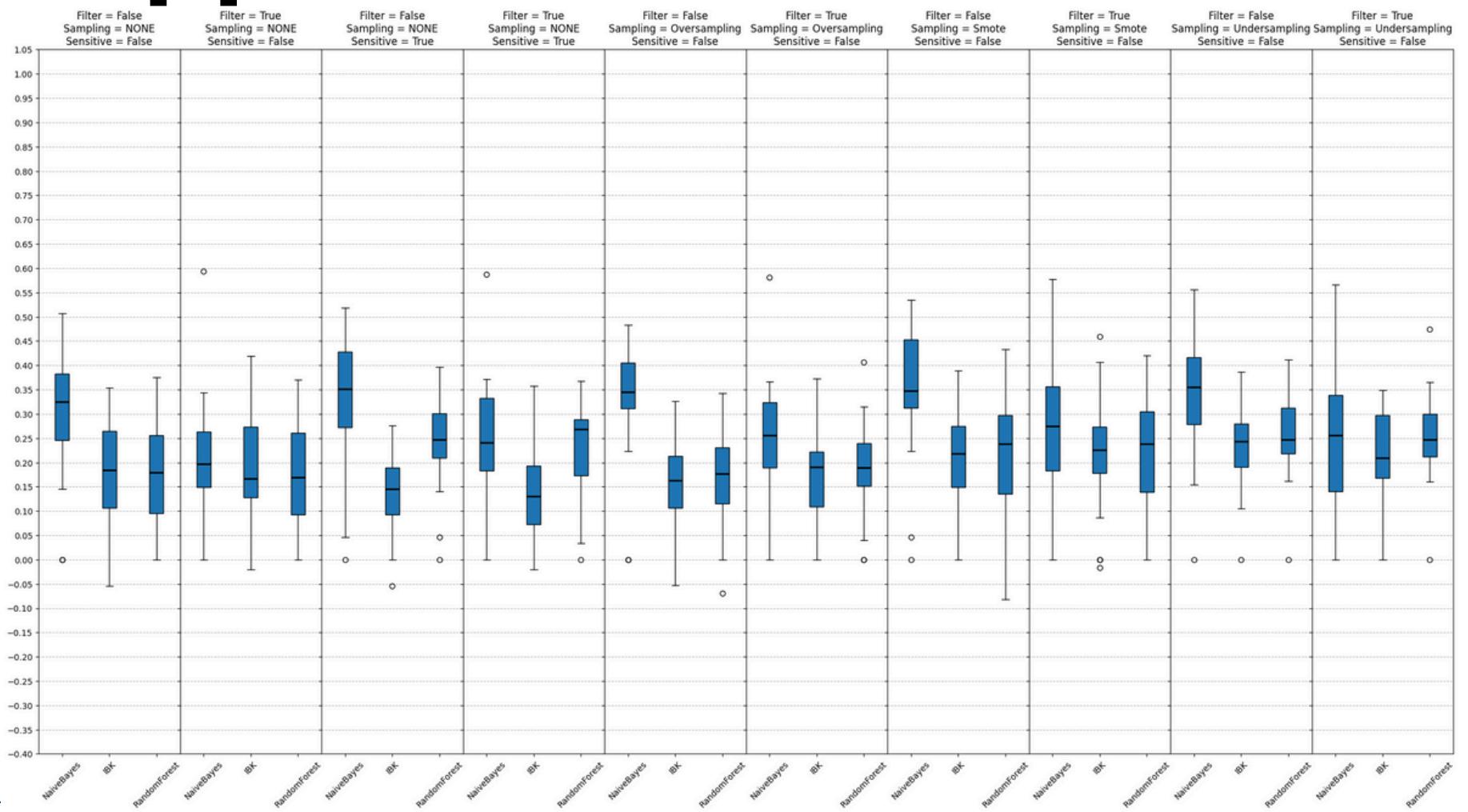
Recall



- **Feature Selection:** Con la sola applicazione di Feature Selection si nota un peggioramento per Naive Bayes, un miglioramento per IBK mentre per Random Forest non si notano né miglioramenti né peggioramenti. Combinando l'utilizzo dei Feature Selection e Sensitive Learning si ottengono notevoli miglioramenti per IBK e Random Forest, mentre peggiora nel caso di Naive Bayes.
- **Oversampling:** Nel caso del solo utilizzo di Oversampling si ottengono miglioramenti per tutti i classificatori considerati. L'utilizzo combinato di Oversampling e Feature Selection porta un miglioramento per IBK e Random Forest ma un peggioramento per Naive Bayes.
- **Smote:** Il solo utilizzo di Smote determina un miglioramento per tutti i classificatori considerati. Nel caso dell'applicazione combinata di Smote e Feature Selection si ottengono miglioramenti per IBK e Random Forest, ma un peggioramento per Naive Bayes.
- **Undersampling:** Applicando solamente Undersampling si ottengono miglioramenti per tutti i classificatori, applicando sia Undersampling che Feature Selection peggiora Naive Bayes, ma migliorano IBK e Random Forest.
- **Sensitive Learning:** Applicando solamente il Sensitive Learning si ottiene un miglioramento per tutti i classificatori considerati; applicando anche il Feature Selection migliorano solo IBK e Random Forest mentre Naive Bayes peggiora.

Risultati - ZooKeeper

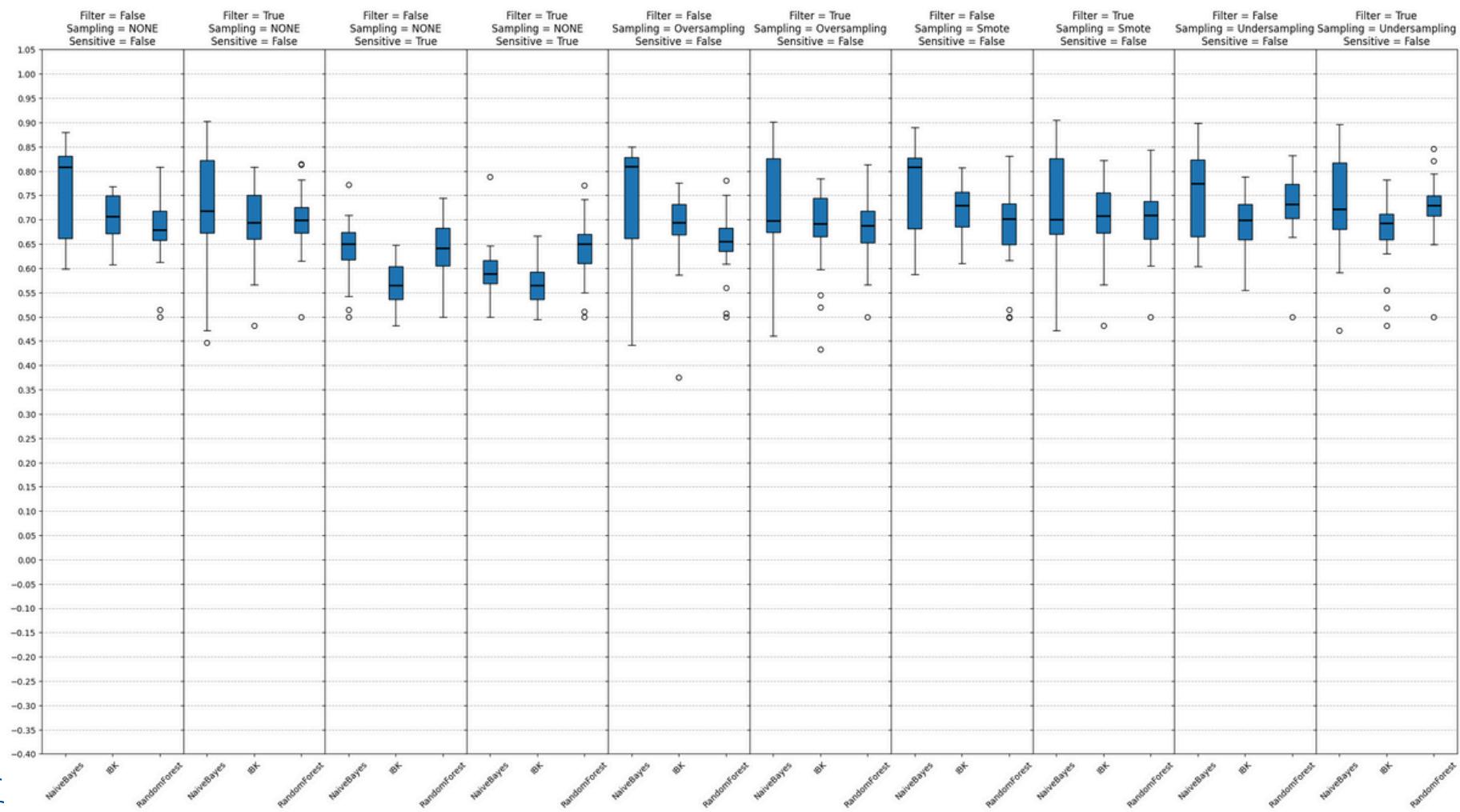
Kappa



- **Feature Selection:** Applicando solamente Feature Selection si ottengono peggioramenti per tutti i classificatori considerati. Applicando sia Feature Selection che Sensitive Learning migliora solo Random Forest, mentre sia Naive Bayes che IBK peggiorano.
- **Oversampling:** Applicando solo Oversampling migliora solo Naive Bayes, mentre gli altri classificatori peggiorano. Applicando sia Oversampling che Feature Selection si ottengono miglioramenti per IBK e Random Forest, mentre Naive Bayes peggiora.
- **Smote:** Applicando solamente Smote si ottengono miglioramenti per tutti i classificatori considerati. Applicando sia Smote che Feature Selection peggiora Naive Bayes mentre migliorano IBK e Random Forest.
- **Undersampling:** Applicando solamente Undersampling, tutti i classificatori considerati migliorano. Applicando sia Undersampling che Feature Selection si ottiene un miglioramento solo per Random Forest, mentre sia Naive Bayes che IBK peggiorano.
- **Sensitive Learning:** Applicando solamente il Sensitive Learning si ottengono miglioramenti per Naive Bayes e Random Forest, mentre peggiora IBK. Applicando anche Feature Selection si ottiene un miglioramento solo per Random Forest, mentre Naive Bayes e IBK peggiorano.

Risultati - ZooKeeper

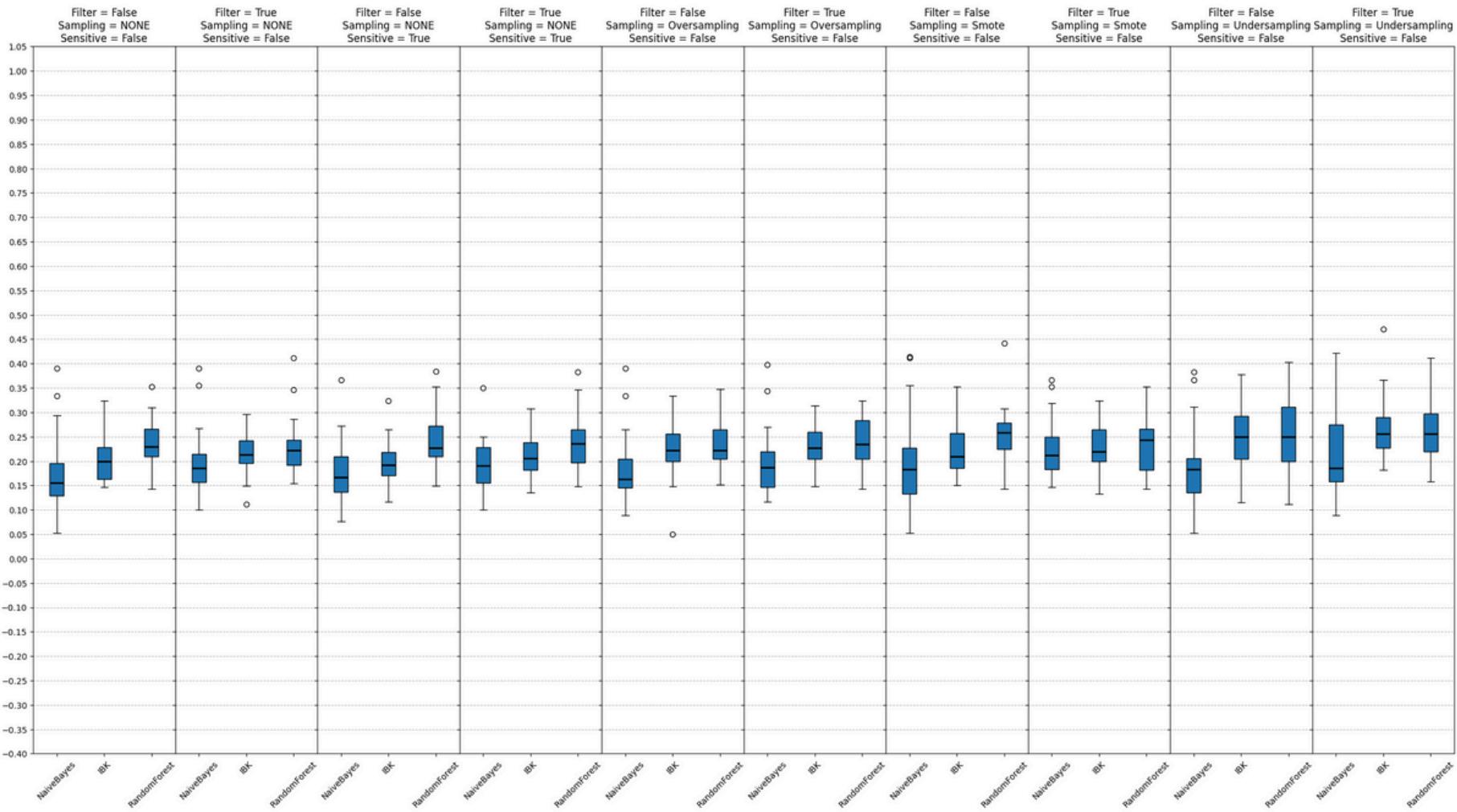
AUC



- **Feature Selection:** Applicando solo Feature Selection si ottengono dei peggioramenti sia per Naive Bayes che per IBK, mentre nel caso di Random Forest si ottiene un miglioramento. Applicando sia Feature Selection che Sensitive Learning si ottengono dei peggioramenti per tutti i classificatori considerati.
- **Oversampling:** Applicando solo Oversampling si ottengono miglioramenti per Naive Bayes e Random Forest, mentre si ottengono peggioramenti nel caso di IBK. Applicando sia Oversampling che Sensitive Learning peggiorano Naive Bayes e IBK ma migliora Random Forest.
- **Smote:** Applicando solo Smote si ottengono miglioramenti per tutti e tre i classificatori. Applicando sia Smote che Feature Selection peggiora Naive Bayes ma migliorano IBK e Random Forest.
- **Undersampling:** Applicando solo Undersampling si ottengono miglioramenti per IBK e Random Forest, mentre Naive Bayes peggiora. Applicando sia Undersampling che Feature Selection si ottiene un miglioramento solo nel caso di Random Forest, mentre Naive Bayes e IBK peggiorano.
- **Sensitive Learning:** Applicando solamente il Sensitive Learning si ottengono peggioramenti per tutti i classificatori considerati. Lo stesso comportamento lo si ottiene applicando sia Sensitive Learning che Feature Selection.

Risultati - ZooKeeper

NPofB20



- **Feature Selection:** Applicando solamente Feature Selection si ottiene un miglioramento solo nel caso di Random Forest, mentre per Naive Bayes e IBK si ottengono lievi peggioramenti. Applicando sia Feature Selection che Sensitive Learning si ottengono peggioramenti per tutti e tre i classificatori considerati.
- **Oversampling:** Applicando solamente Oversampling si ottiene un miglioramento solo per il caso di Random Forest, mentre per Naive Bayes e IBK si ottengono leggeri peggioramenti. Applicando sia Oversampling che Feature Selection si ottengono lievi peggioramenti per Random Forest e IBK, mentre Naive Bayes peggiora notevolmente.
- **Smote:** Con la sola applicazione di Smote si ottengono notevoli peggioramenti per Naive Bayes e Random Forest mentre IBK peggiora lievemente. Applicando sia Smote che Feature Selection si ottiene un peggioramento del valore di NPofB20 per tutti i classificatori considerati.
- **Undersampling:** Con la sola applicazione di Undersampling si ottengono dei peggioramenti per tutti i classificatori considerati. Applicando sia Undersampling che Feature Selection si ha che Naive Bayes peggiora lievemente, mentre sia IBK che Random Forest peggiorano notevolmente.
- **Sensitive Learning:** Applicando solamente il Sensitive Learning si nota come i tre classificatori considerati peggiorino lievemente. Con l'applicazione sia di Sensitive Learning che di Feature Selection i classificatori IBK e Random Forest peggiorano lievemente, mentre Naive Bayes peggiora notevolmente.

Risultati - ZooKeeper

Valutazione

A differenza del caso di BookKeeper, in questo progetto si nota come Naive Bayes assuma valori migliori rispetto agli altri classificatori considerati.

La combinazione migliore risulta essere **Naive Bayes con Oversampling**.

Anche in questo caso i valori si ottengono senza applicare Feature Selection e Sensitive Learning dato che il loro utilizzo non apportava dei miglioramenti.

Utilizzando questa combinazione si ottengono notevoli valori per la Precision, Kappa e AUC a discapito di un valore non troppo elevato di Recall.

GRAZIE PER L'ATTENZIONE

Links

GitHub: https://github.com/alessandro-cortese/ISW2_Project

SonarCloud: https://sonarcloud.io/summary/new_code?id=alessandro-cortese_ISW2_Project