

TECNICHE DI MACHINE LEARNING APPLICATE AL SOFTWARE ENGINEERING

DELIVERABLE MODULO : ML FOR SE

LUDOVICO ZARRELLI – MATRICOLA: 0316448

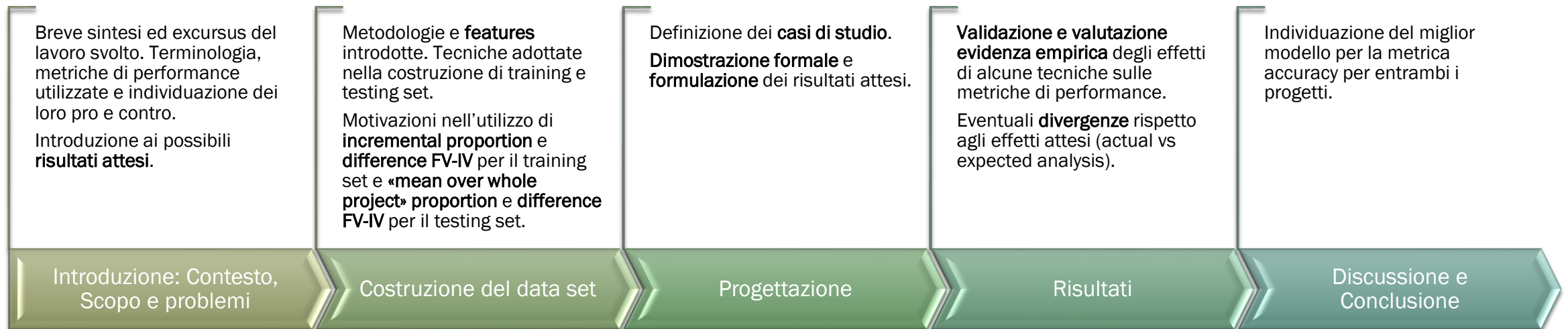
BOOKKEEPER:

[LUDOVICO99/BOOKKEEPERLEARNING \(GITHUB.COM\)](#)
[BookkeeperLearning - ludovico99 \(sonarcloud.io\)](#)

STORM:

[LUDOVICO99/STORMLEARNING \(GITHUB.COM\)](#)
[StormLearning - ludovico99 \(sonarcloud.io\)](#)

ROADMAP:



INTRODUZIONE: CONTESTO

- Applicazione di tecniche di ML come **feature selection**, **sampling**, **cost sensitive classifier**, **walk forward** come tecnica di validazione (il nostro data set è una serie temporale in quanto la correlazione tra il valore target di due istanze successive in release diverse) per l'apprendimento di un modello predittivo per l'individuazione di classi buggy applicato al contesto di due progetti Apache e open-source: Bookkeeper e Storm.
 - **Supervised Learning** : Ad ogni elemento del training set è associato un valore target.
 - **Classificazione binaria**: Il target assume soltanto due possibili valori : no (0) , yes (1).
 - Utilizzo di 4 predittori : **Naive Bayes**, **Random Forest**, **IBK** e **J48**.

INTRODUZIONE: SCOPO

- Il lavoro svolto ha un duplice scopo / obiettivo:
 - Individuazione del **miglior modello predittivo** con relativi filtri in base alla metrica di performance dell' accuratezza.
 - Valutare empiricamente e la conformità rispetto ai **risultati attesi** dell'effetto di tecniche di ML principalmente su tre metriche di performance: **Accuracy, Recall e Precision.**
- I risultati dello studio sono tratti dal software open-source Bookkeeper. Per Storm è possibile ottenere gli stessi risultati.

INTRODUZIONE: TERMINOLOGIA (CONCETTI)

- Introduzione e definizione descrittiva di alcuni concetti ricorrenti nella trattazione: Le metriche di performance.
 - **Recall:** Rappresenta il rate dei positivi classificati correttamente sul totale di istanze effettivamente positive (denominate come real o actual positives).
 - **Precision:** Rate di positivi classificati correttamente sul totale di istanze classificate come positive (denominate come Predictive positives).
 - **Accuracy:** Rate di istanze classificate correttamente sul totale di istanze classificate. E' utile quando le classi sono bilanciate e hanno la stessa importanza.
 - **Kappa:** E' una misura di quanto il nostro classificatore sia migliore di un dummy classifier.
 - **ROC AUC:** Al variare della soglia di threshold (i classificatori restituiscono una probabilità di appartenere ad una o all' altra classe, la threshold è un valore che mi permette di discriminare e stabilire l'appartenenza ad una classe) vengono calcolati due valori: true positive rate (Recall o specificità) e il false positive rate (1 – sensitivity).

- **Tpr** (True positive rate) : $\frac{TP}{TP+FN}$

- **Fpr** (False positive rate): $\frac{FP}{FP+TN}$

Al variare della soglia (da 0 a 1 compresi) viene individuato un punto nel piano con ordinata pari al tpr e con ascissa pari al fpr. Di quest' ultimo grafico bi-dimensionale ne calcolo l'area sotto la curva. Ottengo un valore compreso tra 0 e 1. ROC AUC è una metrica più robusta rispetto l'accuracy in quanto è threshold independent e nel caso in cui le due classi fossero sbilanciate l'accuracy porterebbe a preferire un modello che sceglie principalmente la classe maggioritaria.

INTRODUZIONE: CONCETTI E EXPECTED ANALYSIS - 1

- Tecniche di ML che sono state utilizzate per la realizzazione dello studio:
 - **Feature selection:** Vengono ridotte le features (numero di attributi) per diminuire contestualmente il costo dell'apprendimento. E' utile per data set che sono onerosi da un punto di vista computazionale.
 - **Pro:** Riduzione del costo di apprendimento, overfitting e incremento di accuracy. La presenza di features irrilevanti può comportare una diminuzione dell'accuratezza e portare il modello a basarsi su features poco rilevanti. Soprattutto nel caso di learner based feature selection o wrappers l'incremento di accuracy è evidente a causa del fatto che la scelta degli attributi avviene in base al concetto di highest predictive accuracy.
 - **Cons:** Purtroppo in alcuni casi è possibile avere anche un decremento dell'accuracy poiché ho meno features da utilizzare per l'apprendimento. Feature selection con wrapper è costosa computazionalmente.
 - E' statisticamente provato che esiste un **numero ottimo di features** che dovrebbe essere usato (figura). Se più features sono aggiunte del necessario, le performance del modello diminuiranno (si è aggiunto del rumore).

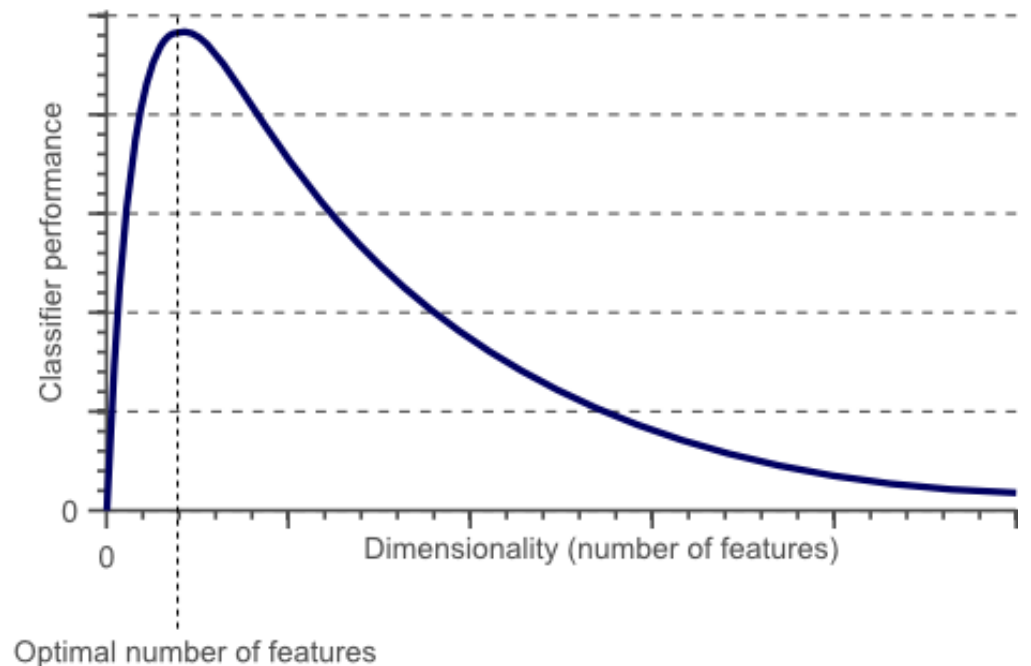
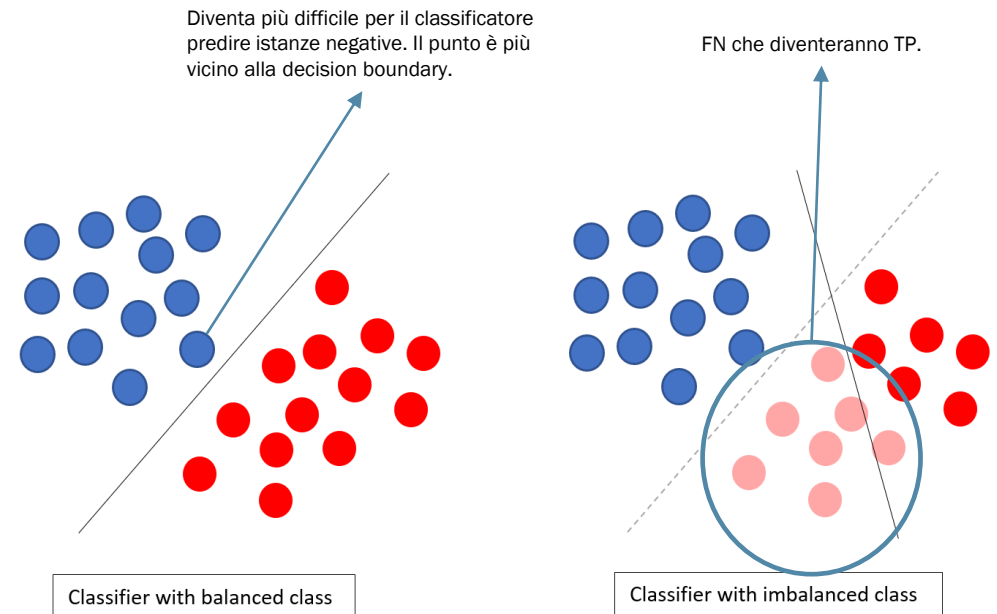


Figura 1: Impatto della dimensionalità sulle performance del classificatore.

INTRODUZIONE: CONCETTI E EXPECTED ANALYSIS - 2

- **Balancing o sampling:** Utilizzato nel caso di un data set sbilanciato, il balancing porta il training set ad avere un numero di istanze positive o negative equiparabile/confrontabile. Supponendo che la classe minoritaria sia quella dei positivi:
- **Pro:** Sicuramente la recall aumenta poiché il classificatore predirà correttamente più istanze positive.
 - **Cons:** La precision diminuisce poiché vengono predetti più falsi positivi. L' accuracy ha un comportamento imprevedibile poiché aumentano i TP e FP ma diminuiscono i FN e i TN.



INTRODUZIONE: CONCETTI E EXPECTED ANALYSIS - 3

- **Cost sensitivity:** In alcune situazioni l'errore commesso nel predire un'istanza negativa come positiva è maggiore di predire un'istanza positiva come negativa. A tal proposito basti pensare al covid, è preferibile dire ad un negativo che è positivo (FP), piuttosto che ad un positivo che è negativo (FN) (Quest' ultimo potrebbe contagiare altre persone). Per questo motivo si tende ad assegnare dei costi in base alla gravità dell' errore commesso CFP e CFN. Supponendo che $CFN = 10 * CFP$:
 - **Pro:** La recall aumenta considerevolmente poiché aumentano i true positive e diminuiscono (di tanto) i false negative (FN è il valore che voglio minimizzare).
 - **Cons:** La precision diminuisce poiché aumentano notevolmente i false positive. Sull'accuracy è difficile inferire un comportamento (come nel balancing) atteso poiché aumentano i TP e FP ma diminuiscono i FN e i TN.
- Nella figura accanto la **decision boundary**, continua, è quella ottenuta nel caso di costi uguali, che implica una threshold pari a 0.5. Il predittore ad ogni istanza assegna la classe più probabile. Quella tratteggiata è una possibile decision boundary, in seguito alla diminuzione della threshold per «favorire» per esempio l'assegnazione alla classe 1 (classe blu). x_1 e x_2 sono i valori delle features.

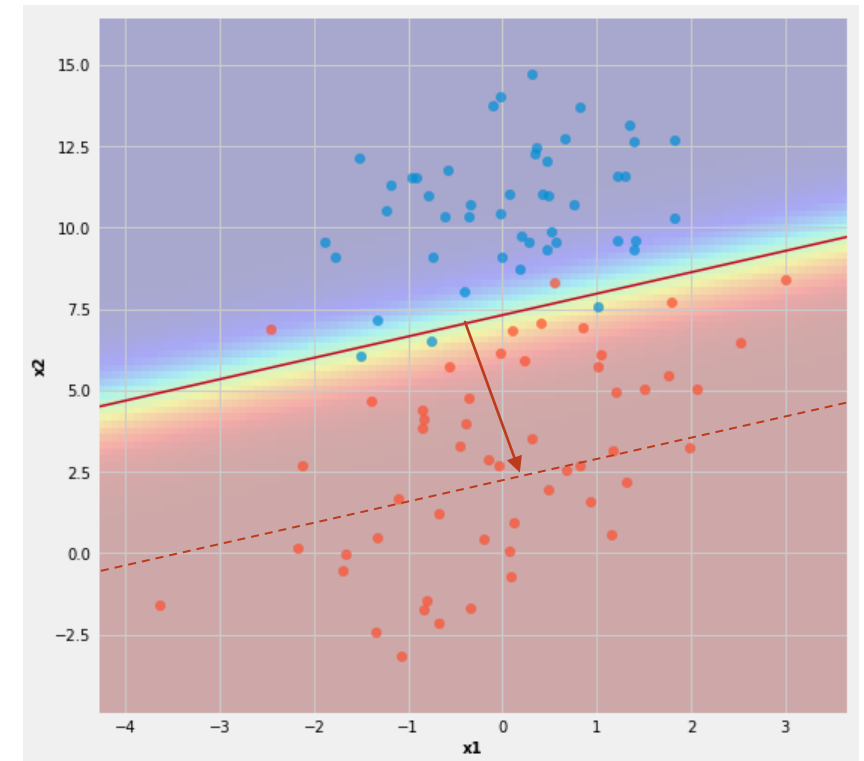


Figura 2: Impatto del Thresholding sulla decision boundary.

COSTRUZIONE DEL DATA SET: LE FEATURES

- Per la realizzazione del data set ho calcolato, attraverso l'unione delle informazioni fornite da jira e git, **18 features** (compreso il nome versione e il nome della classe, che sono stato opportunamente eliminate prima dell'apprendimento). Alcune sono **per release** e altre sono state calcolate su **tutte le release** del progetto.
 - Nome classe, size, age e numero di autori sono per progetto.
 - Nome versione, Loc touched, Numero di revisioni, Loc added, Max Loc added , Avg Loc added, churn, max churn, avg churn, Chg set size, max chg set size, avg chg set size e weighted age sono per release.





COSTRUZIONE DEL DATA SET: LABELING E TARGET VALUES

- Il labeling delle istanze del data set è basato sull'idea che, una classe interessata da commits, contrassegnati da tickets con issue type pari a «bug» e che hanno interessato una release tra le affected versions (versioni da IV a FV -1), precedentemente individuate, per quel ticket è buggy.
 - **IV:** L'earliest version tra le affected versions fornite da jira.
 - **OV:** La release immediatamente successiva alla data di creazione del ticket.
 - **FV:** La release immediatamente successiva alla data di risoluzione del ticket.
- Purtroppo, molto spesso, jira non fornisce la lista delle versioni affected (o è inconsistente) per quel ticket. A tal proposito si è scelto di utilizzare proportion per individuare l'IV per tutti i tickets interessati da questo problema. Si è scelto di adottare incremental proportion per il training e una media su tutto il progetto come proportion per il testing set.
 - P è calcolato solo su tickets, le cui IV, OV e FV rispettano stringenti criteri di consistenza. Non tutti i tickets con IV disponibile da jira sono da considerarsi validi.
 - **Incremental proportion** : Per ogni versione R, P è calcolato come media su tutti i difetti fixed dalla versione 1 a R-1. Così da evitare la contaminazione dei dati temporalmente precedenti con quelli futuri.
 - **«Mean over whole project» proportion** : P è calcolato come media di tutti i difetti fixed del progetto. In modo tale da avere il testing set il più accurato possibile.
 - Nel caso in cui l'OV == FV (Resolution date e creation date coincidono), per calcolare l'IV, ho utilizzato **incremental difference** tra FV e IV, calcolato in precedenza sui tickets «validi», nel caso del training set, mentre ho utilizzato **«mean over whole project» della differenza tra FV e IV** nel caso del testing set.
 - $IV = FV - \text{incremental FV} - IV$ per il training, $IV = FV - \text{mean FV} - IV$ per il testing.

COSTRUZIONE DEL DATA SET: CONCLUSIONI

- Il data set è stato successivamente «tagliato», per mitigare l'effetto di classi snoring, considerando la prima metà delle release.
- **Bookkeeper:**
 - Totale istanze: 2.281 classi (esclusi le classi di test).
 - Totale istanze positive nel training: 485.
 - Totale istanze positive nel testing: 495.
- **Storm:**
 - Totale istanze: 22.221 classi (escluse le classi di test)
 - Totale istanze positive nel training: 2651.
 - Totale istanze positive nel testing: 2964.

PROGETTAZIONE: INTRODUZIONE

- **Walk forward** è una tecnica di validazione del modello. Divido il training e testing set in parti corrispondenti alle istanze di ogni release del progetto. All' iterazione i-esima il training set comprende le istanze dalla release 1 a i -1, mentre il testing set è costituito da quelle relative all'i-esima release.
- La progettazione prevede dapprima, la formulazione, analisi e valutazione della raggiungibilità dei risultati attesi di 3 casi d'uso:
 - Walk forward senza filtri vs walk forward con feature selection.
 - Walk forward senza filtri vs walk forward con SMOTE sampling.
 - Walk forward senza filtri vs walk forward con cost sensitivity.
- Successivamente, verranno analizzate alcune permutazioni (Conta l'ordine di applicazione del filtro) che ritengo più rilevanti ed interessanti dal punto di vista dell'accuratezza.
- **Legenda dei risultati attesi (per le tabelle successive):**
 -  : caso rispetto al quale mi raffronto.
 -  : aumento del valore.
 -  : analiticamente non è possibile inferire alcun ipotesi.
 -  : diminuzione del valore.

PROGETTAZIONE: WALK FORWARD E FEATURE SELECTION

- Caso di studio 1: L'obiettivo è confrontare e validare empiricamente le aspettative su feature selection nei seguenti 2 casi:
 - Walk forward standard vs Walk forward con best first come tecnica di feature selection (E' un filtro, quindi correlation based feature selection technique).
 - Walk forward standard vs Walk forward con Wrapper forwards search (E' un wrapper, quindi learner based feature selection technique).
 - Si suppone che il totale di istanza, dopo l'applicazione del filtro, rimanga lo stesso → L'incremento è uguale al decremento.

$$\text{Recall} = \frac{TP}{TP+FN} \rightarrow \frac{TP+incr_{tp}}{TP+incr_{tp}+FN-decr_{fn}} = \frac{(TP+incr_{tp})}{TP+FN} = \text{Recall} + \frac{incr_{tp}}{TP+FN}$$

$$\text{Accuracy} = \frac{TX}{TX+FX} \rightarrow \frac{TP+incr_{tp}+TN+incr_{tn}}{TP+incr_{tp}+TN+incr_{tn}+FP-decr_{fp}+FN-decr_{fn}} = \text{Accuracy} + \frac{incr_{tp}+incr_{tn}}{TX+FX}$$

$$\text{Precision} = \frac{TP}{TP+FP} \rightarrow \frac{TP+incr_{tp}}{TP+incr_{tp}+FP-decr_{fp}}$$

Con Feature selection	Variazione attesa
TP	↑ = $incr_{tp}$
TN	↑ = $incr_{tn}$
FP	↓ = $decr_{fp}$
FN	↓ = $decr_{fn}$



CS 1:	Senza filtri	Best first	W. Forwards search
Accuracy	=	↑	↑
Recall	=	↑	↑
Precision	=	≠	≠

PROGETTAZIONE: WALK FORWARD E SAMPLING

- Caso di studio 2: L'obiettivo è confrontare e validare empiricamente le assunzioni su balancing nel seguente caso:
 - Walk forward standard vs Walk forward con SMOTE sampling.
- Nel caso di sampling è difficile inferire risultati attesi perché il numero di istanze del data set cambia. Di conseguenza gli incrementi con i rispettivi decrementi, sempre rispetto al caso senza filtri, potrebbero non essere uguali (Il numero di istanze positive e negative cambia applicando sampling). Se fossero uguali (Il totale di istanze rimane invariato) la situazione sarebbe stata la seguente:

- $Recall = \frac{TP}{TP+FN} \rightarrow \frac{TP+incr_{tp}}{TP+incr_{tp}+FN-decr_{fn}} = \frac{(TP+incr_{tp})}{TP+FN} = Recall + \frac{incr_{tp}}{TP+FN}$
- $Accuracy = \frac{TX}{TX+FX} \rightarrow \frac{TP+incr_{tp}+TN+incr_{tn}}{TP+incr_{tp}+TN+incr_{tn}+FP-decr_{fp}+FN-decr_{fn}} = Accuracy + \frac{incr_{tp}+incr_{tn}}{TX+FX}$
- $Precision = \frac{TP}{TP+FP} \rightarrow \frac{TP+incr_{tp}}{TP+incr_{tp}+FP-decr_{fp}}$

Con Sampling	Variazione attesa
TP	↑ = $incr_{tp}$
TN	↓ = $decr_{tn}$
FP	↑ = $incr_{fp}$
FN	↓ = $decr_{fn}$







CS 2:	Senza filtri	SMOTE sampling
Accuracy	=	≠
Recall	=	↑
Precision	=	↓

PROGETTAZIONE: WALK FORWARD E COST SENSITIVITY



- Caso di studio 3: L'obiettivo è confrontare e validare empiricamente le assunzioni su cost sensitive classifiers nel seguente caso:

➤ Walk forward standard vs Walk forward con sensitive threshold e $CFN = 10 * CFP$.

- $Recall = \frac{TP}{TP+FN} \rightarrow \frac{TP+incr_{tp}}{TP+incr_{tp}+FN-decr_{fn}} = \frac{(TP+incr_{tp})}{TP+FN} = Recall + \frac{incr_{tp}}{TP+FN}$
- $Accuracy = \frac{TX}{TX+FX} \rightarrow \frac{TP+incr_{tp}+TN-decr_{tn}}{TP+incr_{tp}+TN-decr_{tn}+FP+incr_{fp}+FN-decr_{fn}} = Accuracy + \frac{incr_{tp}-decr_{tn}}{TX+FX}$
- $Precision = \frac{TP}{TP+FP} \rightarrow \frac{TP+incr_{tp}}{TP+incr_{tp}+FP+incr_{fp}}$

Con Cost sensitivity	Variazione attesa
TP	 = $incr_{tp}$
TN	 = $decr_{tn}$
FP	 = $incr_{fp}$
FN	 = $decr_{fn}$



CS 3:	Senza filtri	Sensitive threshold
Accuracy		
Recall		
Precision		

RISULTATI: ACCURACY IN BOOKKEEPER

Considerando tutte le iterazioni possiamo dedurre le seguenti conclusioni:

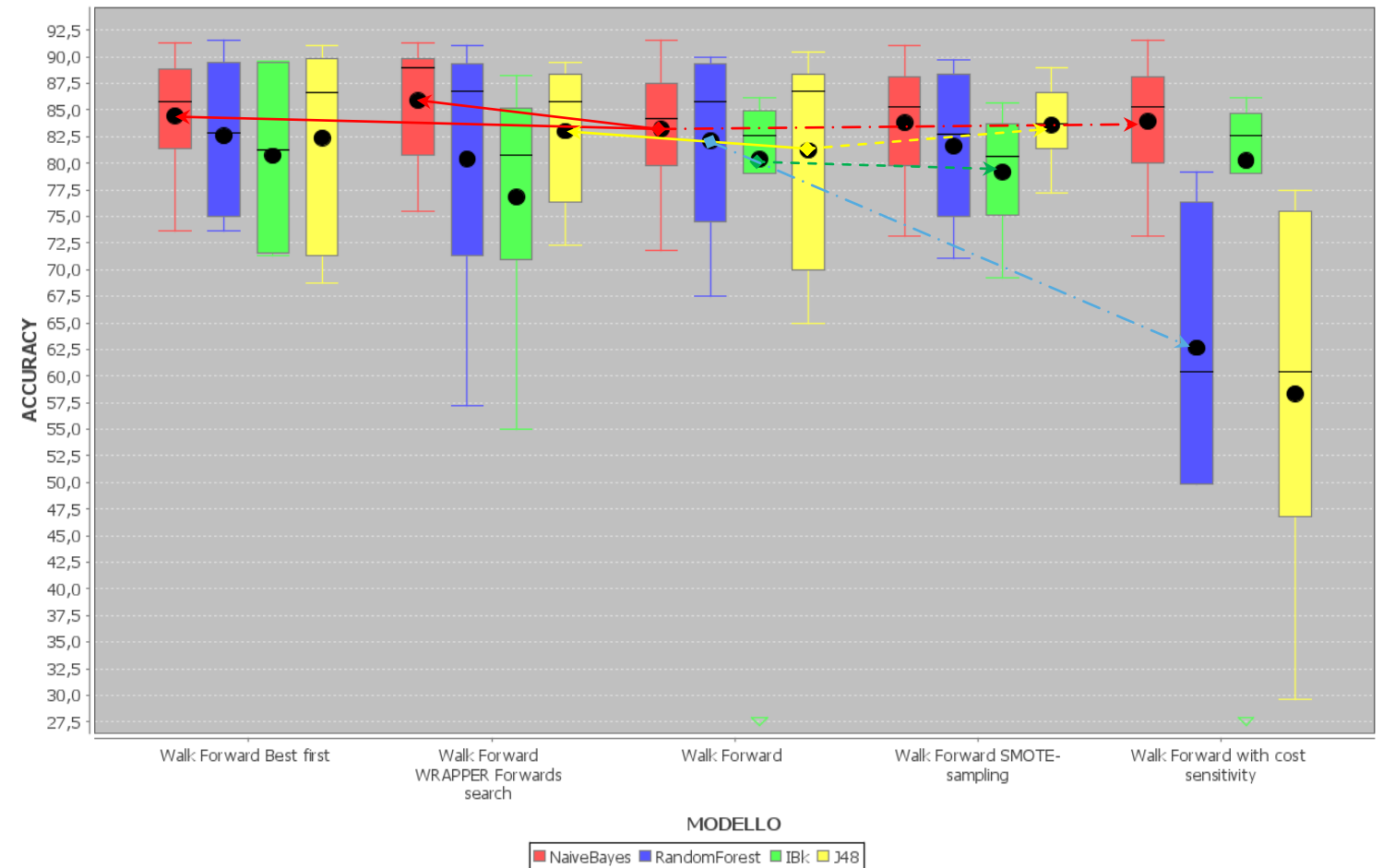
- Come motivato in precedenza, Wrapper forwards search porta ad un notevole aumento dell' accuracy, soprattutto nel caso di Naïve bayes.
- In generale, non è sempre vero che l'accuracy aumenti nel caso di feature selection. E' possibile che vengano tolte features rilevanti. Inoltre, ho meno informazioni dalle quali il classificatore deve apprendere.
- La mediana, spesso superiore alla media, è il risultato di una distribuzione di frequenze inclinata a sinistra o asimmetrica negativa. Ciò implica una coda più lunga a sinistra e più alta a destra. I limiti superiori dei dati sono estremamente alti, in corrispondenza presumibilmente nelle ultime iterazioni.
- L'accuratezza, in 2 di 4 casi, diminuisce leggermente per SMOTE e notevolmente per cost sensitivity.

Legenda per i primi tre casi di studio:

- —◆— per cs 1
- - - -◆- - - per cs 2
- ·····◆···· per cs 3

Classifiers

Which classifier is the best one?



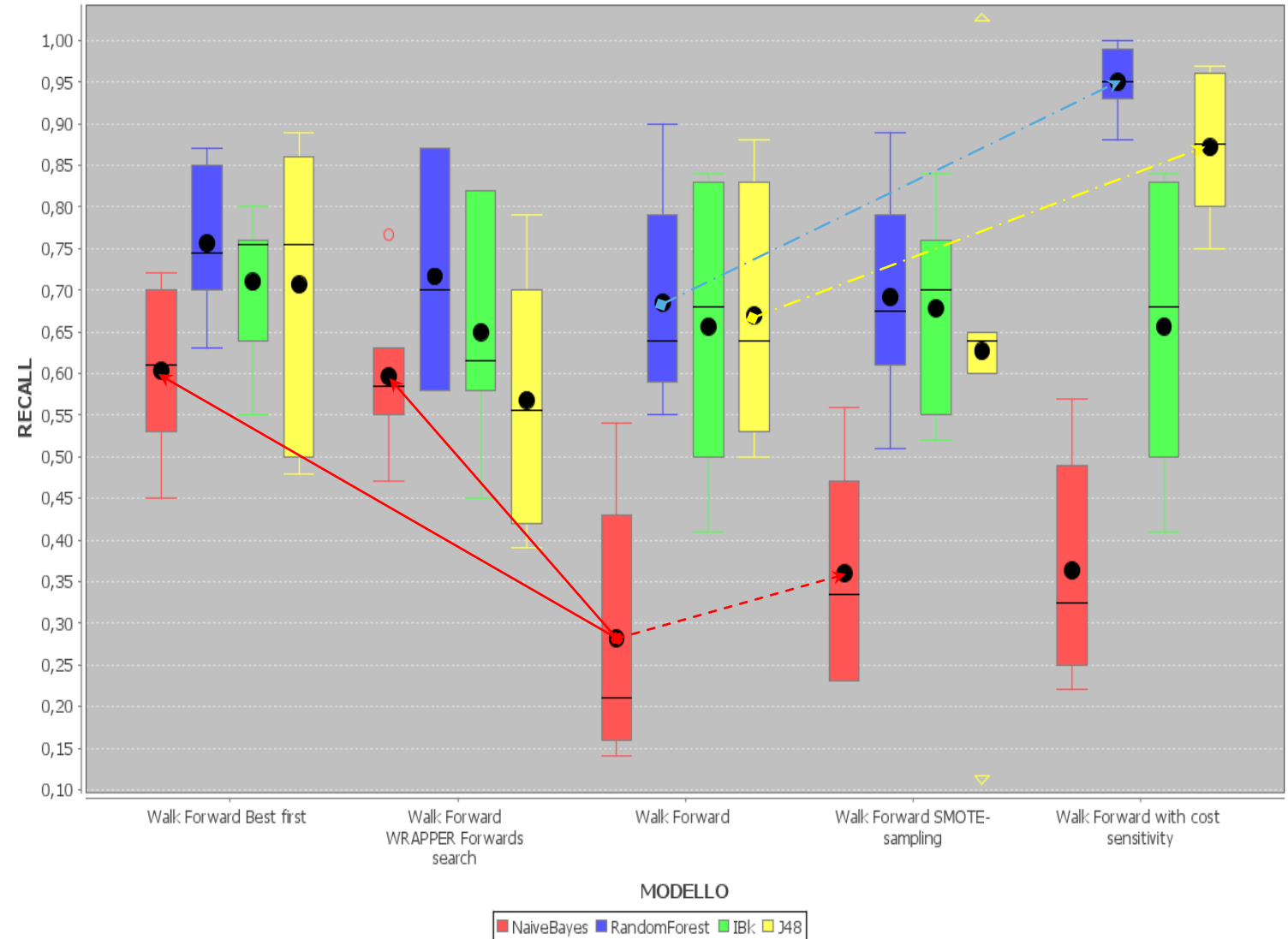
RISULTATI: RECALL IN BOOKKEEPER

E' possibile giungere alle seguenti conclusioni:

- La recall aumenta leggermente per smote sampling.
- Per cost sensitive classifier l'aumento, poichè la differenza di costo tra CFN e CFP è notevole, del valore della recall è considerevole.
- La recall aumenta nel caso di feature selection.

Classifiers

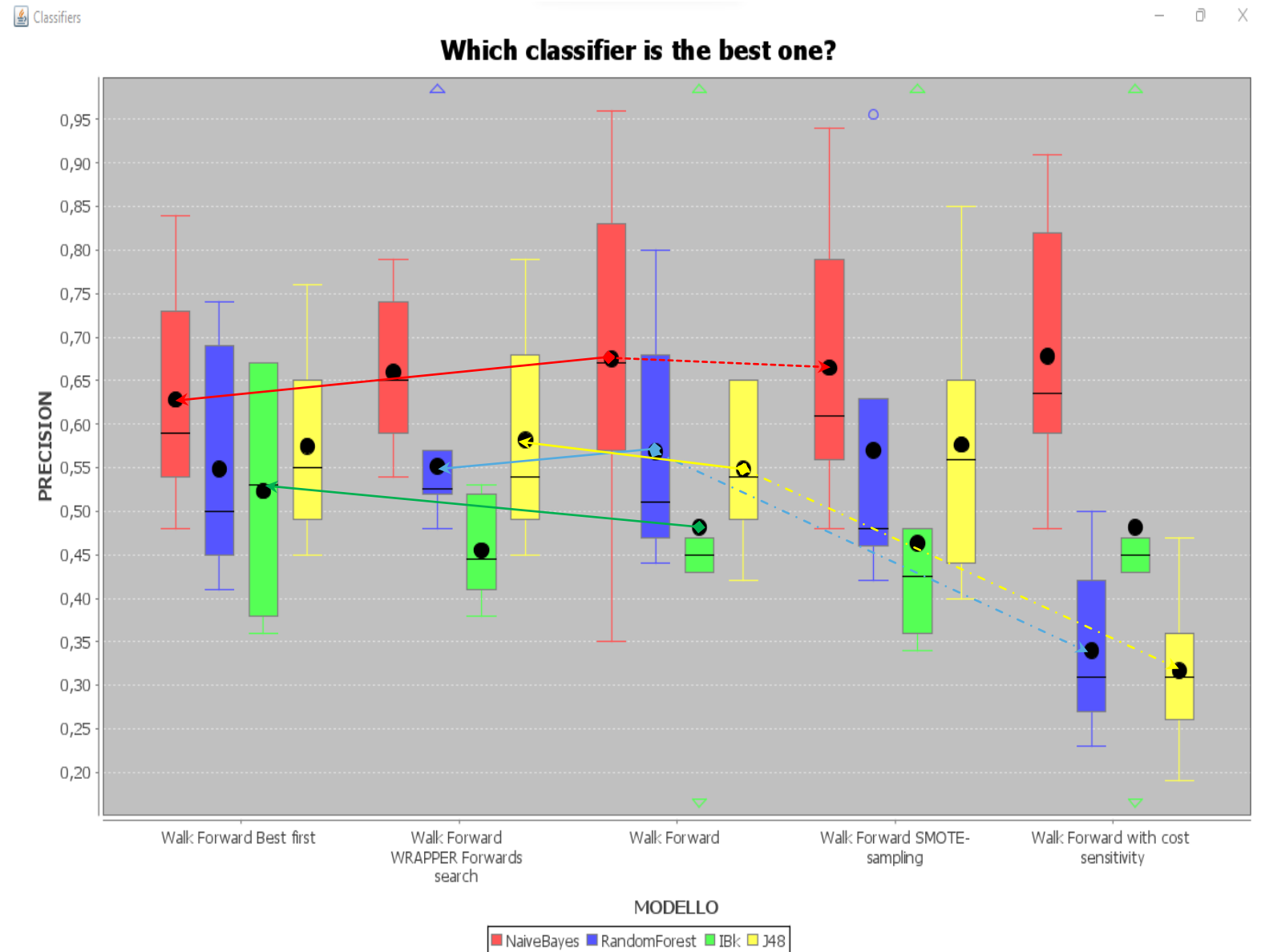
Which classifier is the best one?



RISULTATI: PRECISION IN BOOKKEEPER

E' possibile giungere alle seguenti conclusioni:

- La precision diminuisce leggermente per smote sampling in 2 casi su 4 (IBK, Naïve Bayes).
- Per cost sensitive classifier la diminuzione del valore della precision è considerevole e rapportabile alla variazione della recall (La precision diminuisce di più nei casi in cui la recall è aumentata maggiormente).
- Per feature selection la precision diminuisce per Random Forest e Naïve Bayes ma aumenta per IBK e J48.



PROGETTAZIONE: CASI DI STUDIO COMPLESSI

- Ho deciso di applicare in tutti i casi feature selection (come secondo filtro) poiché in linea teorica dovrebbe aumentare l'accuracy.
- Cosa succede se feature selection è applicato precedentemente al balancing o a cost sensitivity ?
- L'obiettivo è confrontare e validare empiricamente alcune permutazioni (conta l'ordine e senza ripetizione) di tecniche di ML:
 - CS 4: Walk forward balancing vs Walk forward con Feature selection e balancing (BEST FIRST).
 - CS 5: Walk forward con cost sensitivity vs Walk forward con Feature selection e cost sensitivity (BEST FIRST).
- In entrambi i casi mi aspetto di notare gli effetti prodotti dall'utilizzo di feature selection.

PROGETTAZIONE: ANALISI DEI RISULTATI ATTESI (CS4, CS5)

Con Feature selection:	Variazione attesa
TP	↑
TN	↑
FP	↓
FN	↓

Con cost sensitivity o balancing:	Variazione attesa
TP	↑
TN	↓
FP	↑
FN	↓



CS 4 :	Senza filtri	Solo SMOTE	Best first + SMOTE
Accuracy	= ↔ ≠	↔	↑
Recall	= ↔	↑	↔
Precision	= ↔	↓	↔ ≠

CS 5 :	Senza filtri	Solo cost sensitivity	Best first + Cost sensitivity
Accuracy	= ↔ ≠	↔	↑
Recall	= ↔	↑	↔
Precision	= ↔	↓	↔ ≠

RISULTATI: ACCURACY IN BOOKKEEPER

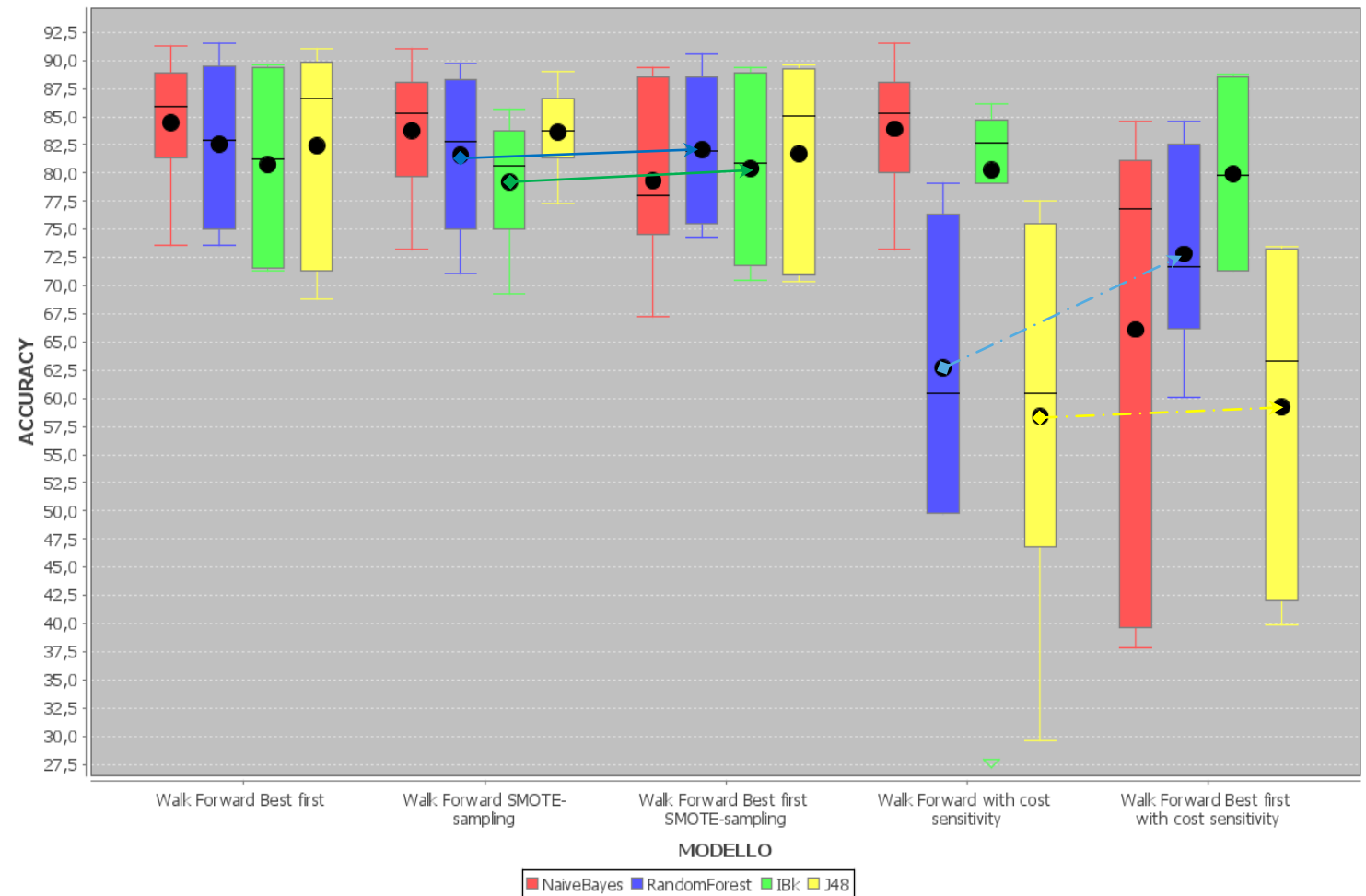
- Soltanto nella metà dei casi è possibile notare un aumento dell'accuracy in seguito all'utilizzo di best first.
- Nel caso di Naïve bayes non è visibile avere un riscontro empirico delle considerazioni precedenti.
- Non avendo conoscenze sul funzionamento dei modelli utilizzati non riesco a capire il motivo di questa discrepanza tra teoria ed esperienza.

Legenda per gli ultimi due casi di studio:

- —◆— per cs 4
- - -◆- - per cs 5

Classifiers

Which classifier is the best one?

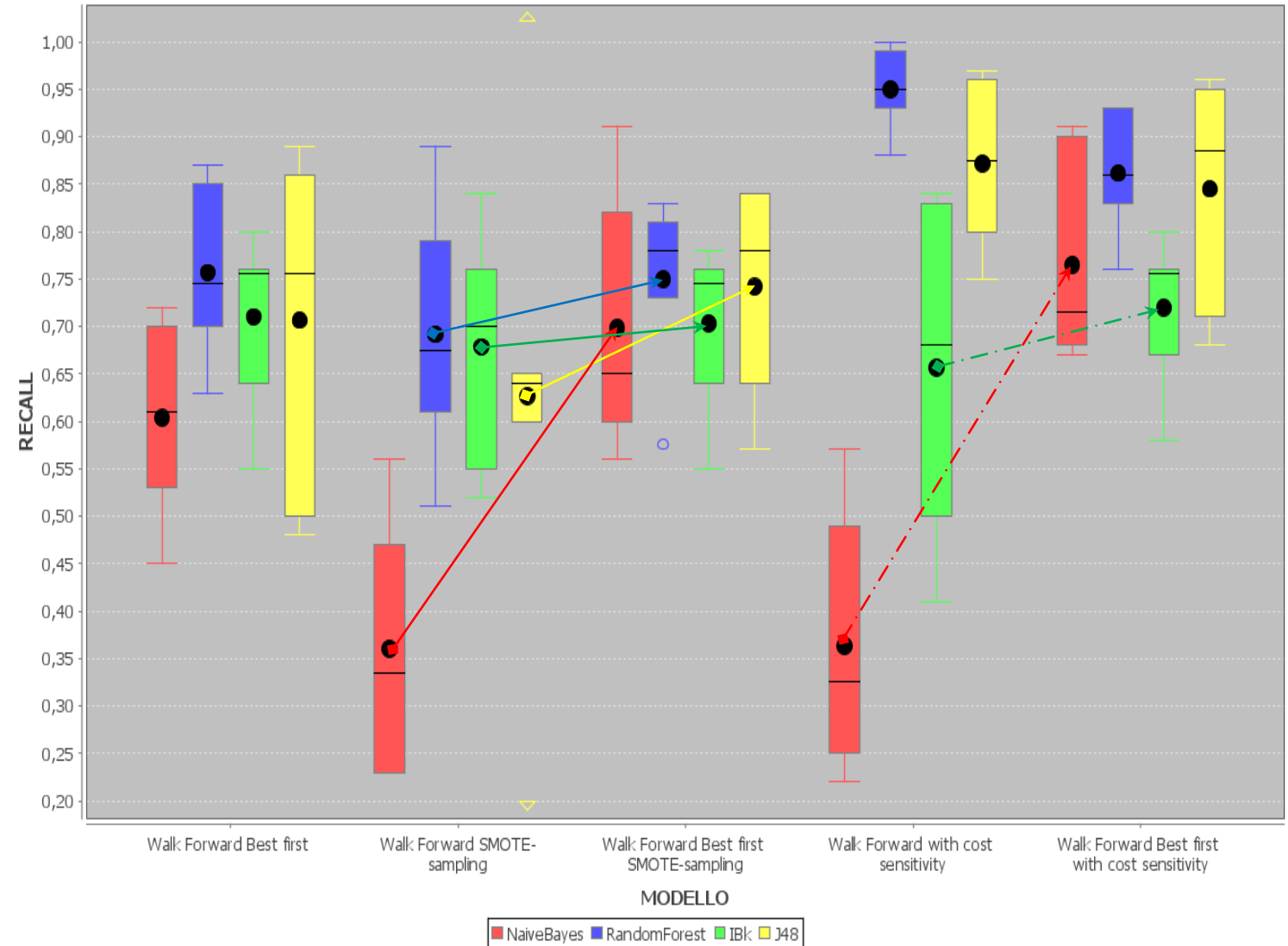


RISULTATI: RECALL IN BOOKKEEPER

- Al contrario dell'accuratezza, è evidente in questo caso un aumento della recall.
- Soltanto nel caso di Random forest non si ha un miglioramento delle prestazioni.
- Si può evincere che best first migliora spesso la recall.
- Rispetto al caso solo SMOTE, SMOTE + best first evidenzia un aumento di recall in tutti e 4 i casi.

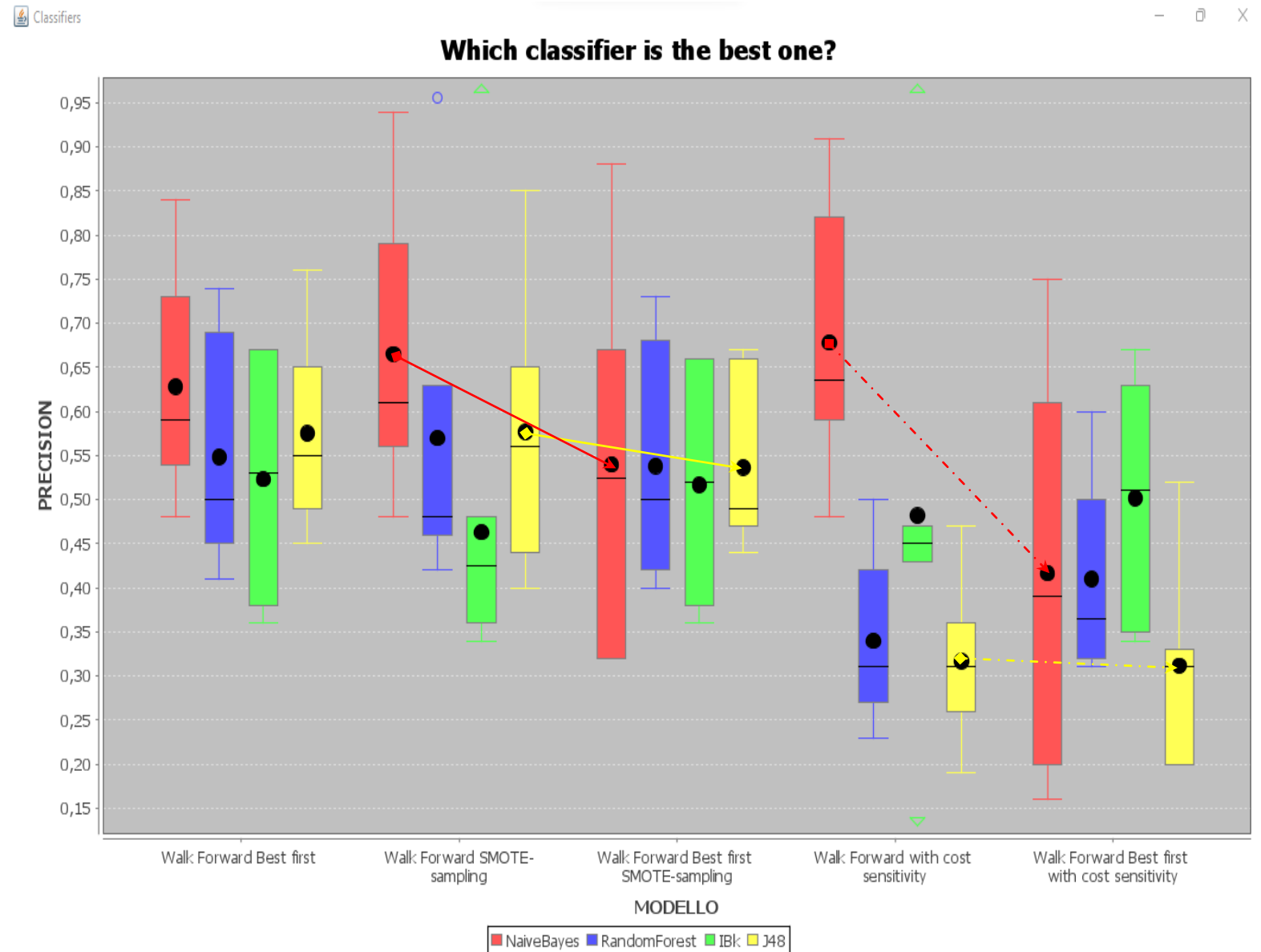
Classifiers

Which classifier is the best one?



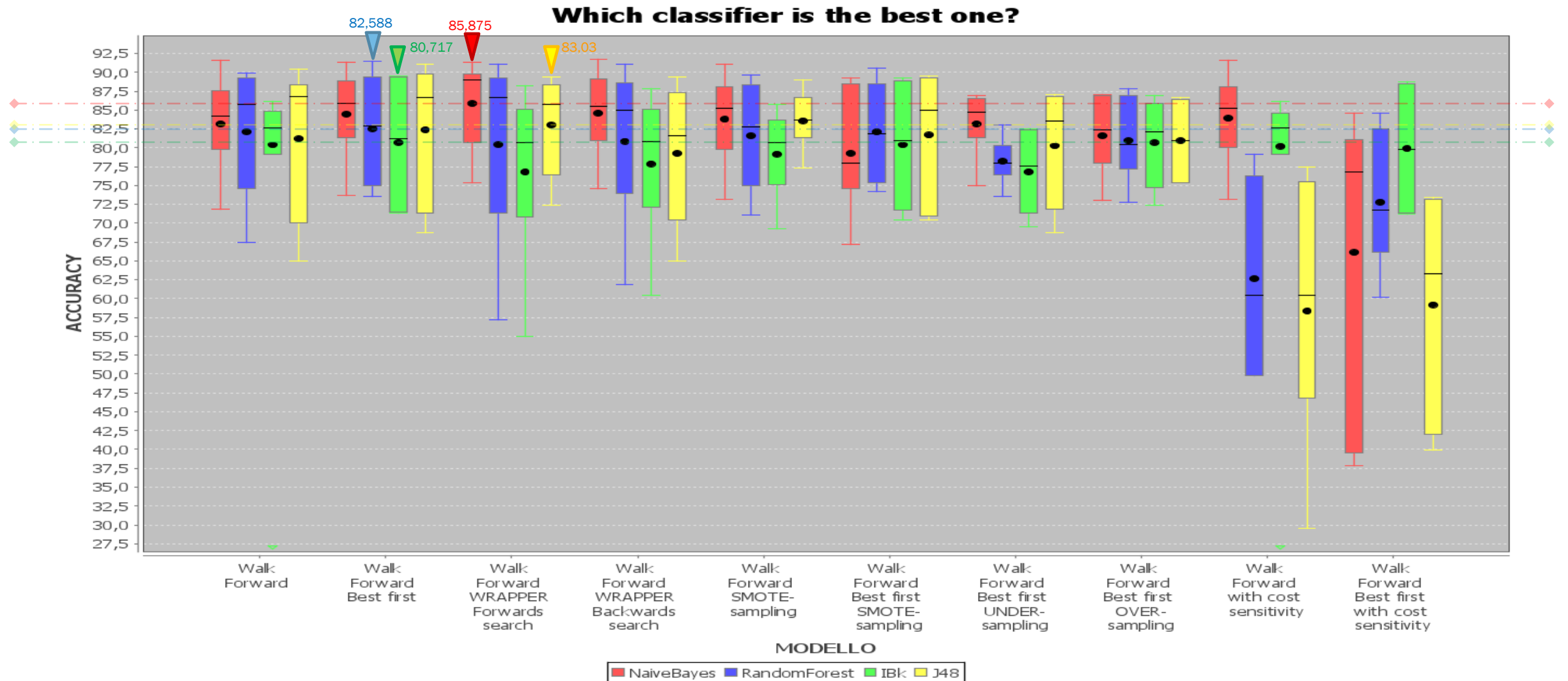
RISULTATI: PRECISION IN BOOKKEEPER

- Come nel caso dell'accuracy, è possibile validare empiricamente un comportamento altalenante della precision.
- Soltanto in Naïve bayes e J48 si ha diminuzione di precision.



CONCLUSIONE BOOKKEEPER: QUALE CLASSIFICATORE HA IL MIGLIOR VALORE (IN MEDIA) DI ACCURATEZZA?

Classifiers



CONCLUSIONE STORM : QUALE CLASSIFICATORE HA IL MIGLIOR VALORE (IN MEDIA) DI ACCURATEZZA?

Classifiers

— □ ×

Which classifier is the best one?

