Facoltà di Ingegneria Informatica
Università di Roma Tor Vergata

Relazione Ingegneria del Software 2

Machine Learning for Software Engineering

Deliverable 2

Camilli Michela 0286047

Indice

Introduzione
Descrizione del problema

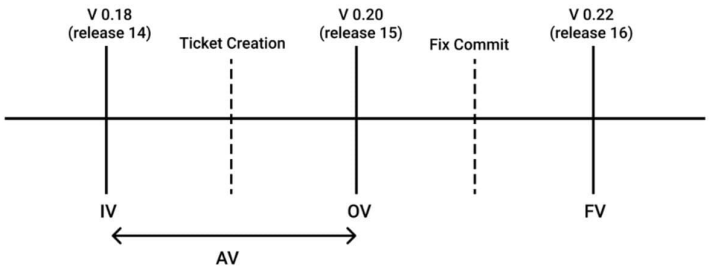
Questa Deliverable ha lo scopo di fornire un'analisi di dati, su due repository Apache open-source e di valutare, in base a delle metriche calcolate, se esista una correlazione tra quest'ultime e l'insorgenza di bug nel codice. I progetti presi in cosiderazione sono stati:

- Apache BookKeeper
- Apache ZooKeeper

Tale studio è consistito di due fasi:

- Una prima fase devoluta al corretto reperimento dei dati, all'analisi della buggyness e al calcolo delle metriche necessarie;
- Una seconda fase in cui si è valutata l'accuratezza di alcuni modelli predittivi per la previsione della buggyness, tramite la combinazione di alcune tecniche date.

I dati presi in considerazione sono relativi a progetti open-source Apache e hanno necessitato di un'analisi approfondita per riuscire ad individuare quali classi, all'interno di tali progetti, fossero affette da bug.
Si è preso, dapprima, in considerazione il ciclo di vita di un bug:



Andando, quindi, a reperire i valori necessari per l'analisi di tale life-cycle da Jira o tramite l'ausisilio di Git, qualore nel primo non fossero disponibili.

In particolare, per andare a sopperire alla mancanza di dati relativi all'AV su Jira, si è sfruttato il metodo *Proportional* di tipo Increment, il quale utilizza l'intuizione per cui esiste una proporzione costante tra le versioni che passano tra l'IV e l'OV e tra quelle tra OV e FV (che, al contrario dei primi, sono noti).
Nello specifico, tale proporzione è pari a:

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

Tramite cui è stato possibile calcolare il valore di IV predetto uguale a:

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

Per entrambi i progetti, sono state considerate 11 metriche, necessarie per consentire ai classificatori di stimare la presenza o meno di bug per una classe, in una certa release. In particolare sono state selezionate:

- **NR**: Numero di revisioni;
- **NAuth**: Numero di autori;
- **LOC_added**: Somma sulle revisioni di LOC aggiunte (per una release);
- **MAX_LOC_added**: Massimo sulle revisioni di LOC aggiunte (per una release);
- **AVG_LOC_added**: Media sulle revisioni di LOC aggiunte (per una release);

- **Churn**: Somma sulle revisioni di LOC aggiunte - rimosse;
- **MAX_Churn**: Valore massimo di Churn sulle revisioni;
- **AVG_Churn**: Valore medio di Churn sulle revisioni;
- **ChgSetSize**: Numero di file committati insieme con la classe considerata;
- **MAX_ChgSet**: Valore massimo di ChgSet sulle revisioni;
- **AVG_ChgSet**: Valore medio di ChgSet sulle revisioni;

I dati ottenuti sono stati, quindi, inseriti in un apposito file con nome *Project*Dataset.csv.

Per lo sviluppo di questa parte della Deliverable è stato utilizzato il tool *weka*.

A seguito del calcolo delle metriche nella fase precedente, è stato, quindi, possibile creare dei training e testing set, con i dati contenuti nei file *csv*, convertiti in file *arff*.

Sono stati, in particolare, creati validi fold tramite l'utilizzo della tecnica *Walk-Forward*. Questa consiste nel dividere il data-set in parti, cronologicamente, in modo che, per predire le release successive, i classificatori siano addestrati su quelle, volta per volta, precedenti.
Ad ogni iterazione, infatti, tutti i dati presenti nelle release precedenti al fold da predire, sono usati come training set, mentre i dati relativi alla release attuale sono usati come testing set.

I classificatori considerati nei due progetti sono stati:

- **RandomForest**
- **NaiveBayes**
- **Ibk**

Per aumentarne l'accuratezza, misurata in termini di *Precision*, *Recall*, *AUC* e *Kappa*, sono state utilizzate le seguenti tecniche:

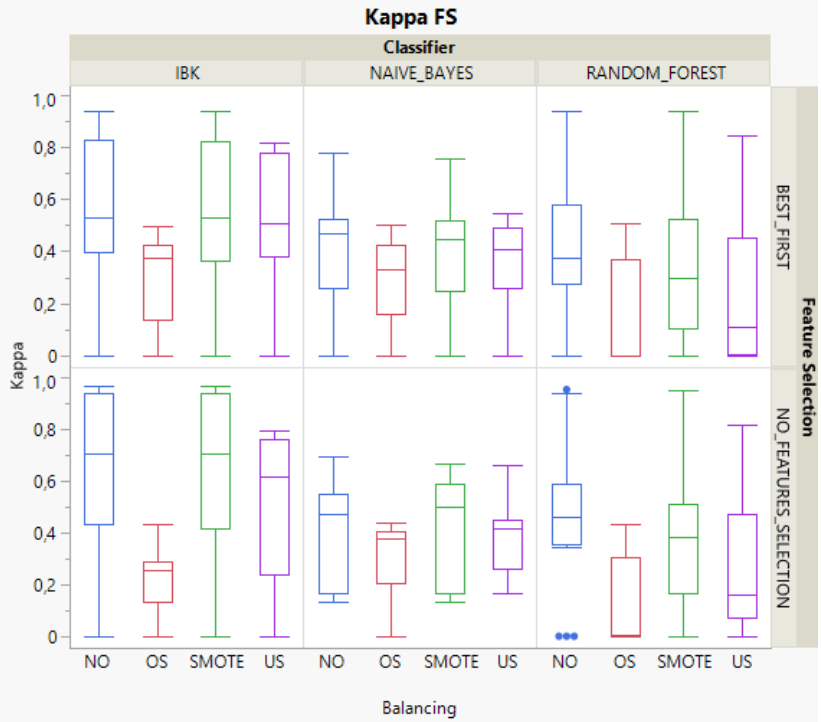
- **No Selection** e **Best First** come *Feature Selection*;
- **No Sampling**, **Oversampling**, **Undersampling** e **SMOTE** come *Balancing*;
- **No Cost Sensitive**, **Sensitive Threshold** e **Sensitive Learning** come *Sensitive Valuation*, in cui la matrice dei costi prevede, in entrambi i casi, un peso 10 volte maggiore per un falso negativo rispetto a quello per un falso positivo.

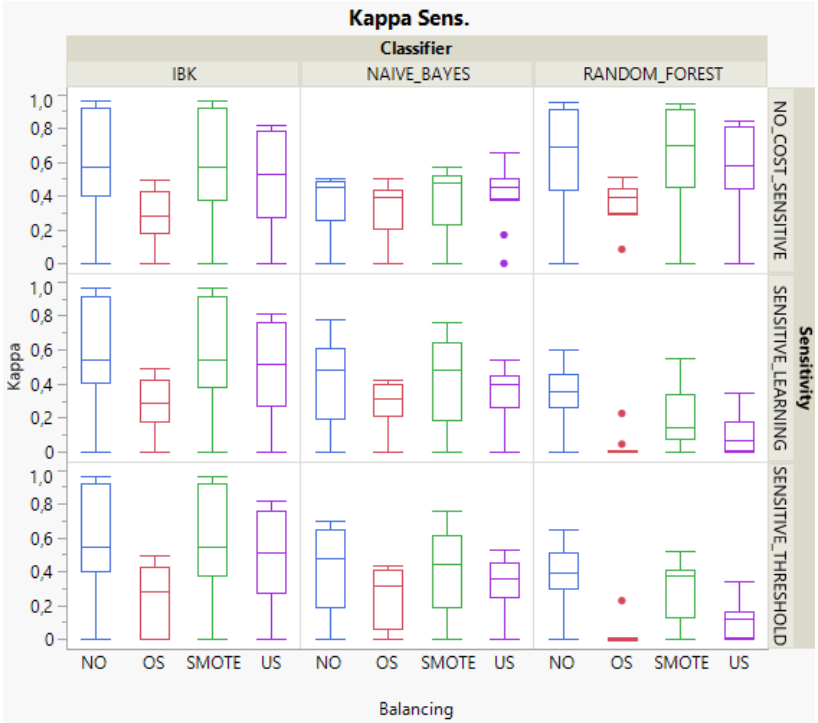
Risultati Ottenuti
Scelte d'analisi

Per l'analisi dei risultati ottenuti, è stata graficata, tramite box-plot, l'accuratezza raggiunta con tutte le possibili configurazioni delle tecniche sovraelencate, in modo da capire quali combinazioni hanno portato ai modelli di predizione più accurati per il problema di classificazione considerato.

Risultati Ottenuti
BOOKKEEPER-KAPPA

Le migliori configurazioni ottenute per *Kappa* sono state:



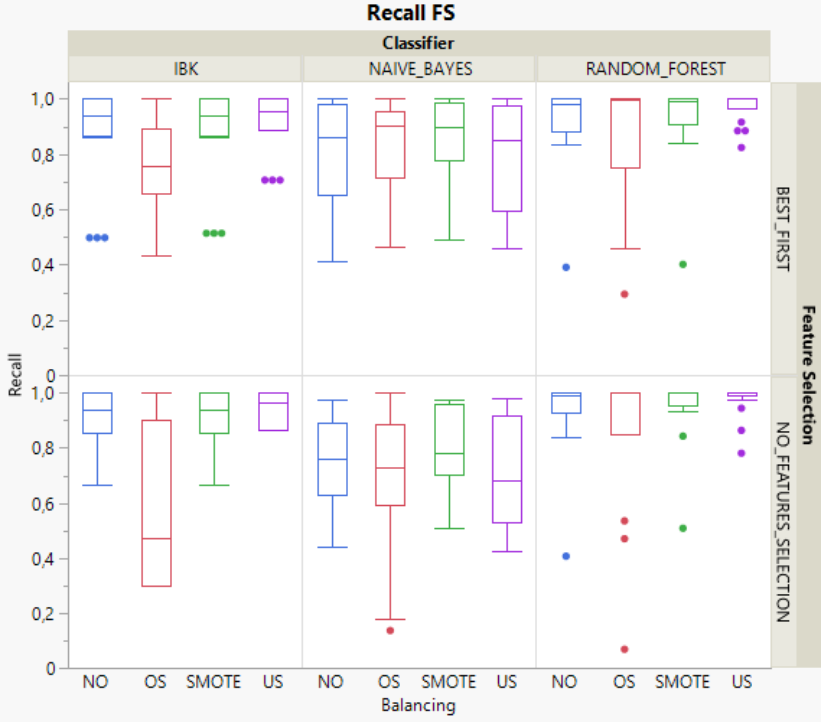


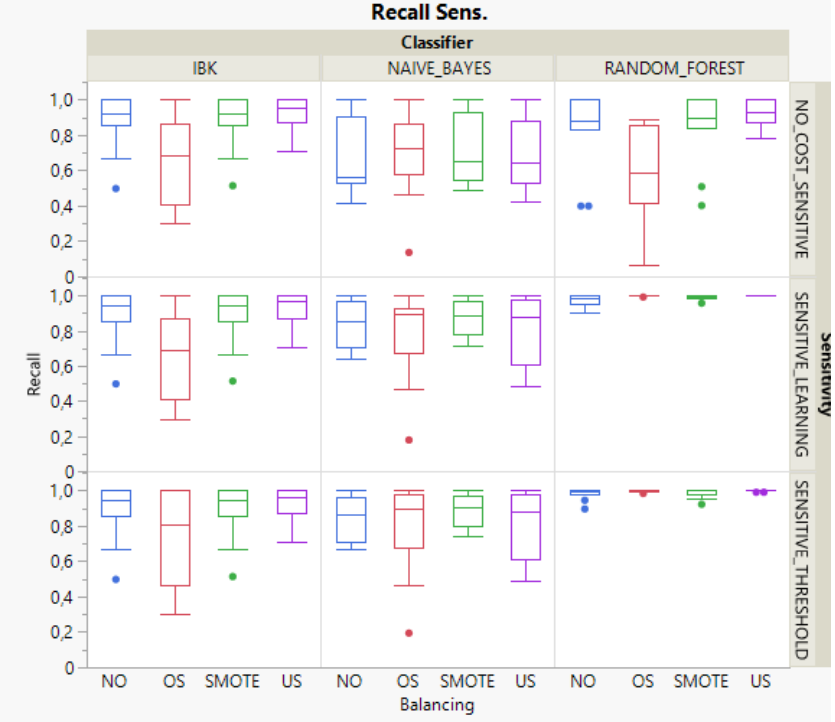
- IBK, {No_Sampling, SMOTE}, Best_First

- IBK, {No_Sampling, SMOTE}, {No_Cost_Sensitive, Sensitive_Learning, Sensitive_Threshold}
- Random_Forest, {No_Sampling, SMOTE}, No_Cost_Sensitive

Risultati Ottenuti
BOOKKEEPER-RECALL

Le migliori configurazioni ottenute per *Recall* sono state:



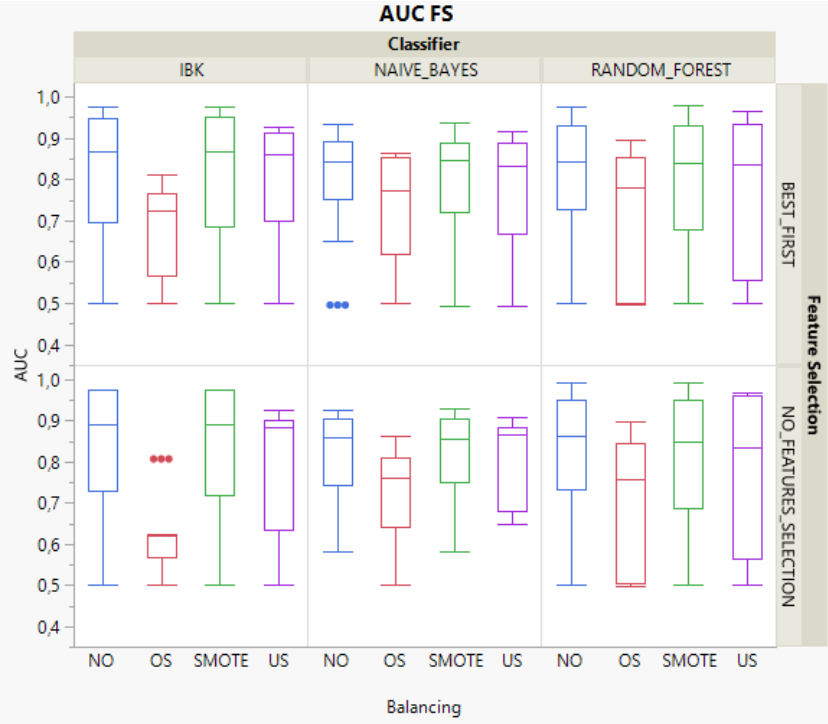


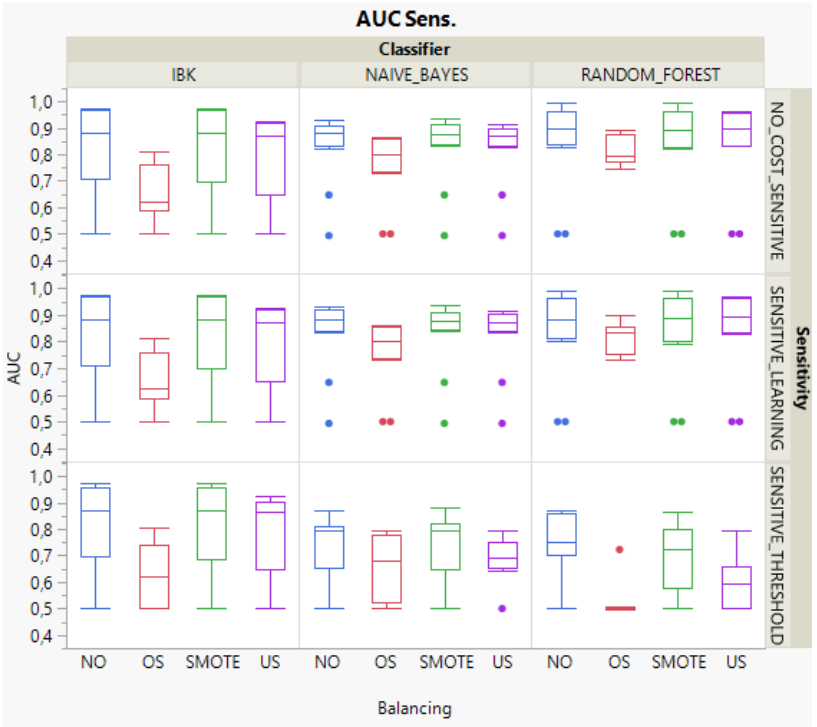
- IBK, {No_Sampling, SMOTE, Under_Sampling}, Best_First
- Random_Forest, {No_Sampling, SMOTE, Under_Sampling}, No_Features_Selection

- Random_Forest, {No_Sampling, SMOTE, Under_Sampling}, {Sensitive_Learning, Sensitive_Threshold}

Risultati Ottenuti
BOOKKEEPER-ROC AREA

Le migliori configurazioni ottenute per *AUC* sono state:



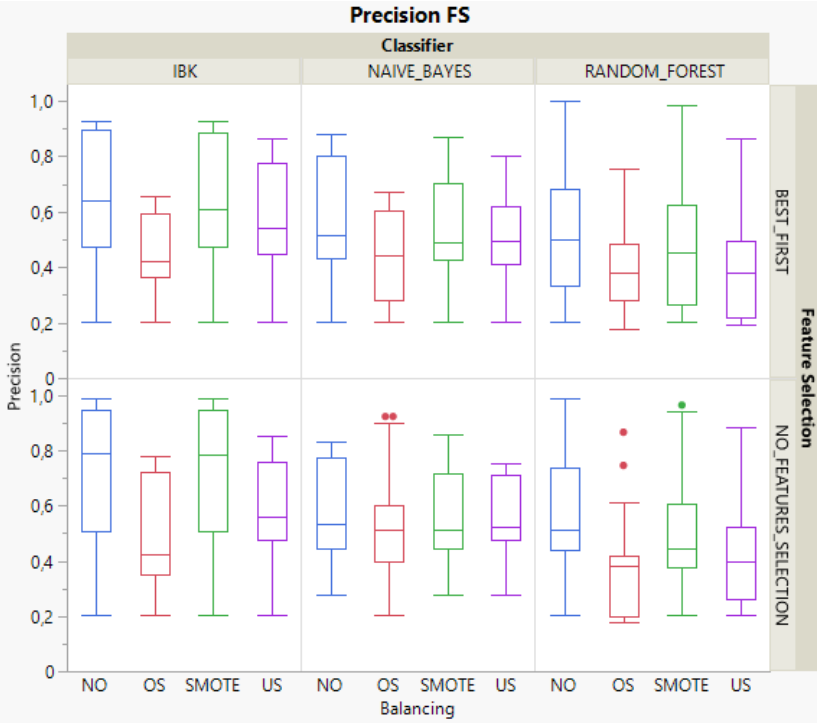


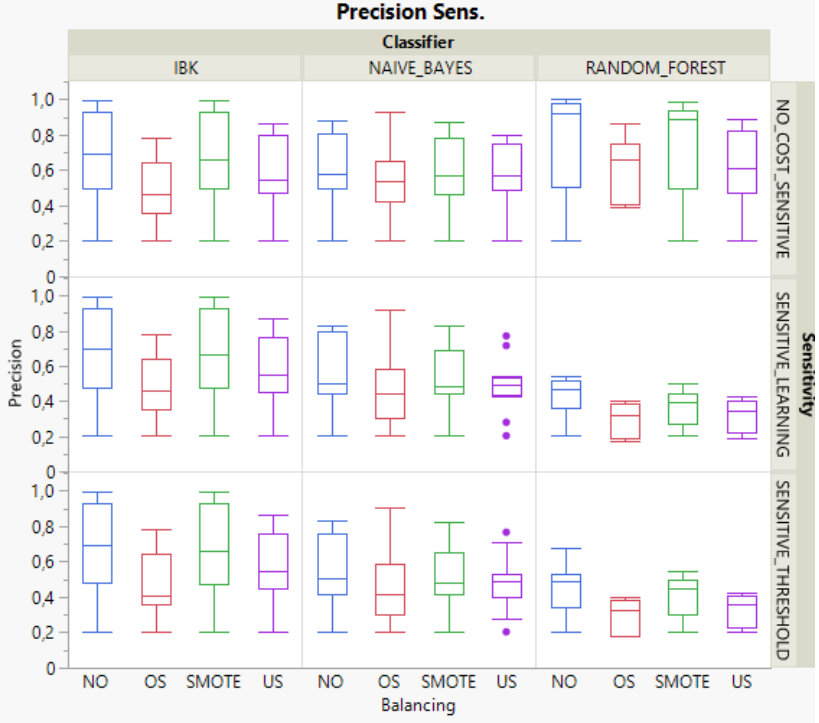
- IBK, {No_Sampling, SMOTE}, Best_First
- IBK, {No_Sampling, SMOTE}, No_Features_Selection
- Random_Forest, No_Sampling, No_Features_Selection

- Naive_Bayes, {No_Sampling, SMOTE}, {No_Cost_Sensitive, Sensitive_Learning}
- Random_Forest, {No_Sampling, SMOTE}, {No_Cost_Sensitive, Sensitive_Learning}

Risultati Ottenuti
BOOKKEEPER-PRECISION

Le migliori configurazioni ottenute per *Precision* sono state:





- IBK, {No_Sampling, SMOTE}, {Best_First, No_Features_Selection}
- Random_Forest, {No_Sampling, SMOTE}, {No_Cost_Sensitive}

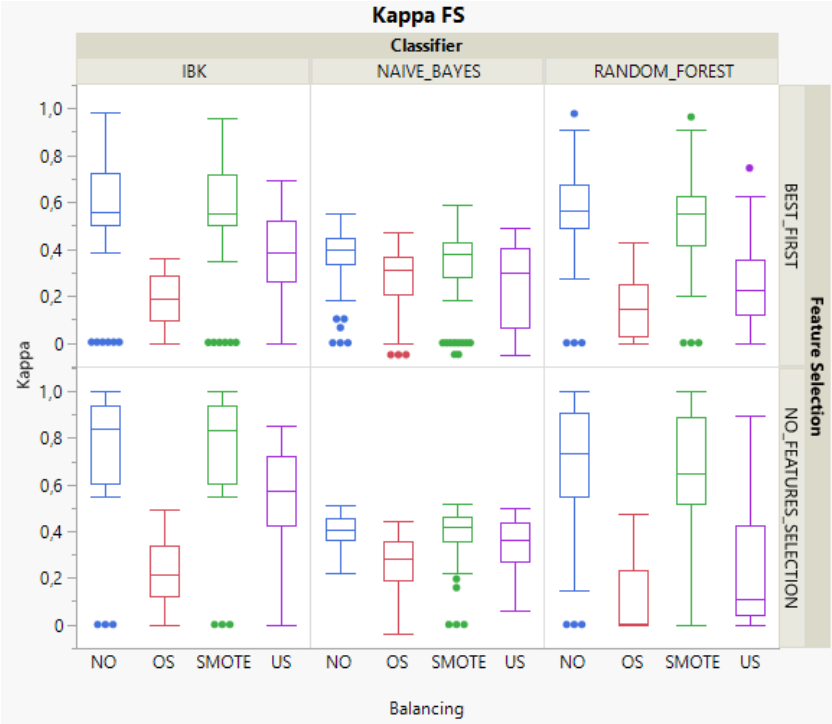
Risultati Ottenuti
Considerazioni BOOKKEEPER

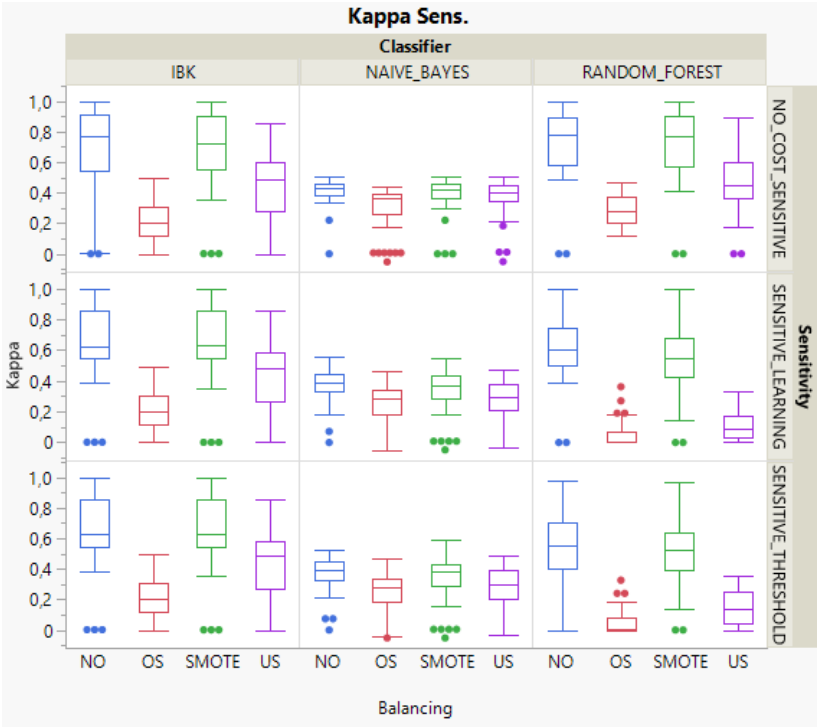
Dai box-blot precedenti, emergono alcuni risultati:

- Per qualsiasi metrica di accuratezza, il classificatore *IBK* risulta presente nelle migliori configurazioni, in combinazione con le tecniche considerate;
- Le tecniche di balancing che, prevalentemente, hanno fornito i migliori risultati sono state, in egual misura, *SMOTE* e *No_Sampling*;
- La metrica di accuratezza ad aver maggiormente beneficiato delle tecniche di Sensitive_Valuation è stata *Recall*;
- *AUC* è stata la metrica con il più ampio range interquartile.

Risultati Ottenuti
ZOOKEEPER-KAPPA

Le migliori configurazioni ottenute per *Kappa* sono state:

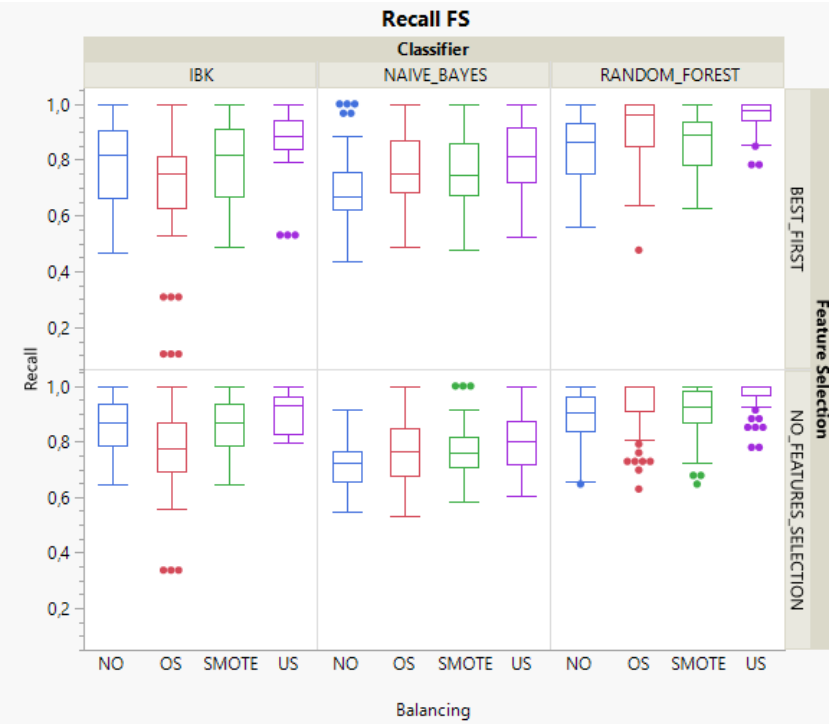


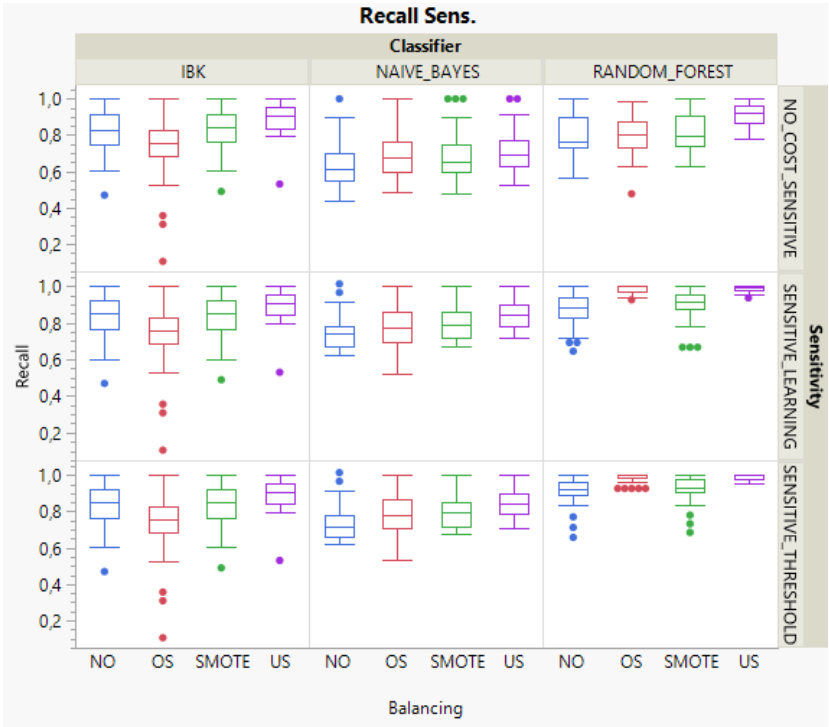


- IBK, {No_Sampling, SMOTE}, No_Features_Selection
- Random_Forest, {No_Sampling, SMOTE}, No_Cost_Sensitive

Risultati Ottenuti
ZOOKEEPER-RECALL

Le migliori configurazioni ottenute per *Recall* sono state:



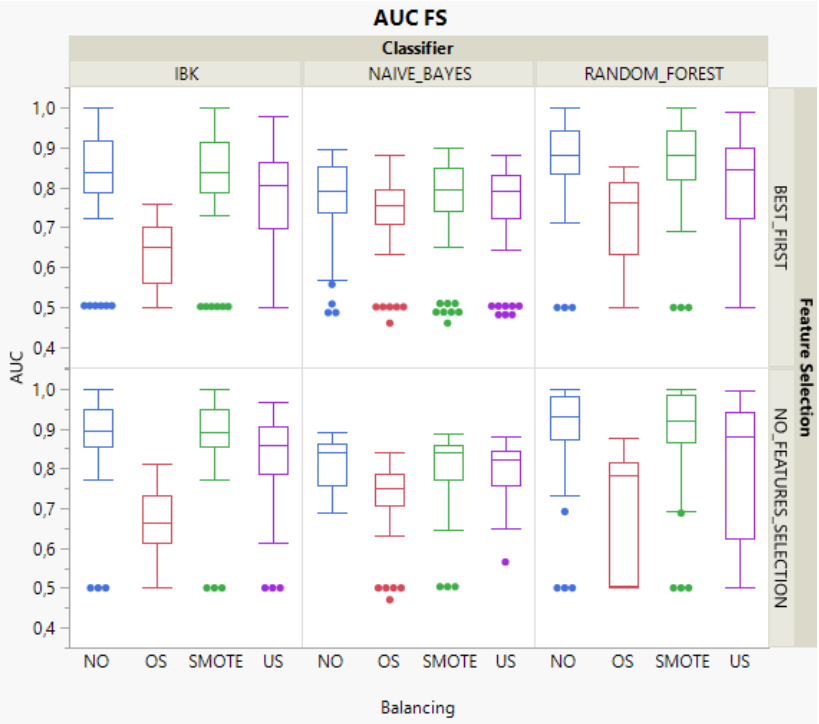


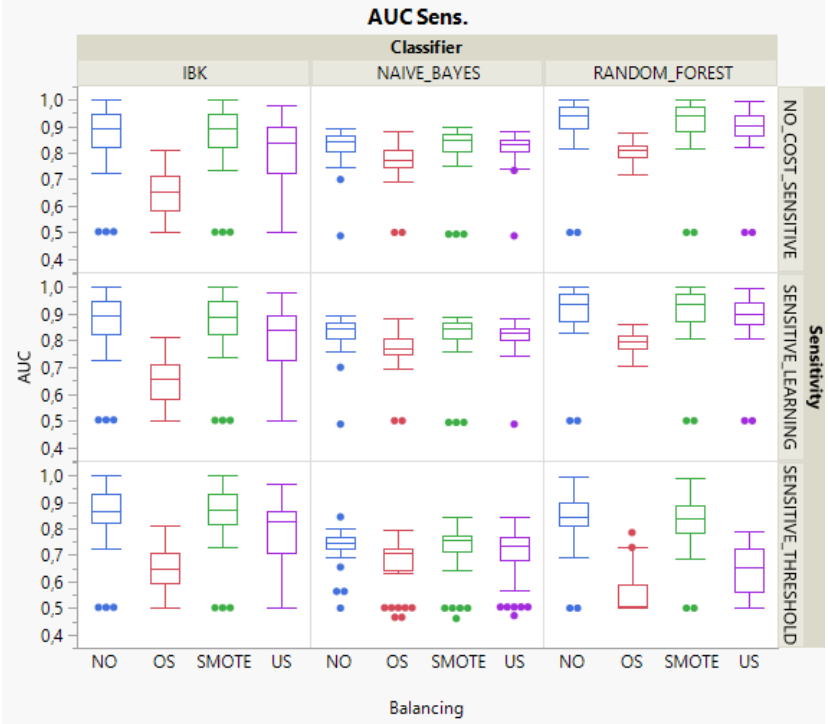
- IBK, Under_Sampling, {Best_First, No_Features_Selection}
- Random_Forest, Under_Sampling, {Best_First, No_Features_Selection}

- IBK, Under_Sampling, {No_Cost_Sensitive, Sensitive_Learning, Sensitive_Threshold}
- Random_Forest, Under_Sampling, {No_Cost_Sensitive, Sensitive_Learning, Sensitive_Threshold}

Risultati Ottenuti
ZOOKEEPER-ROC AREA

Le migliori configurazioni ottenute per *AUC* sono state:



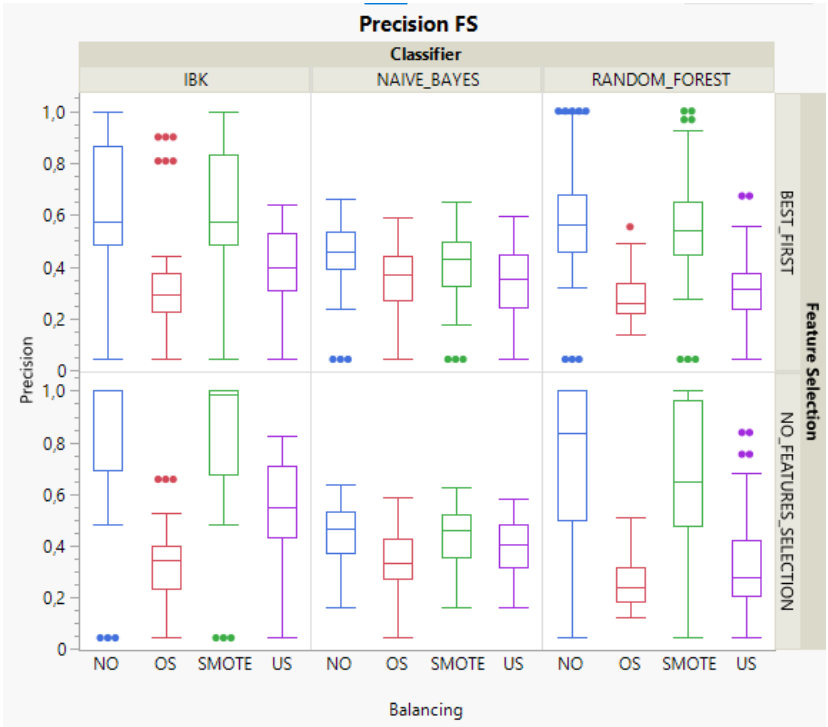


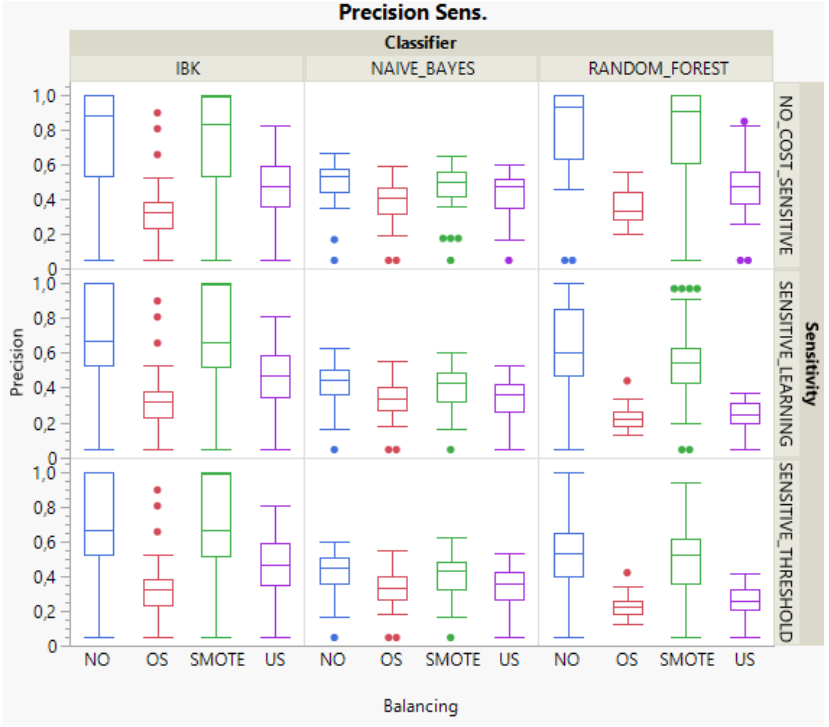
- IBK, {No_Sampling, SMOTE}, {Best_First, No_Features_Selection}
- Random_Forest, {No_Sampling, SMOTE}, {Best_First, No_Features_Selection}

- IBK,{No_Sampling, SMOTE}, {No_Cost_Sensitive, Sensitive_Learning, Sensitive_Threshold}
- Random_Forest, {No_Sampling, SMOTE, Under_Sampling}, {No_Cost_Sensitive, Sensitive_Learning}

Risultati Ottenuti
ZOOKEEPER-PRECISION

Le migliori configurazioni ottenute per *Precision* sono state:





- IBK, {No_Sampling, SMOTE}, No_Features_Selection

- IBK,{No_Sampling, SMOTE}, No_Cost_Sensitive
- Random_Forest, {No_Sampling, SMOTE}, No_Cost_Sensitive

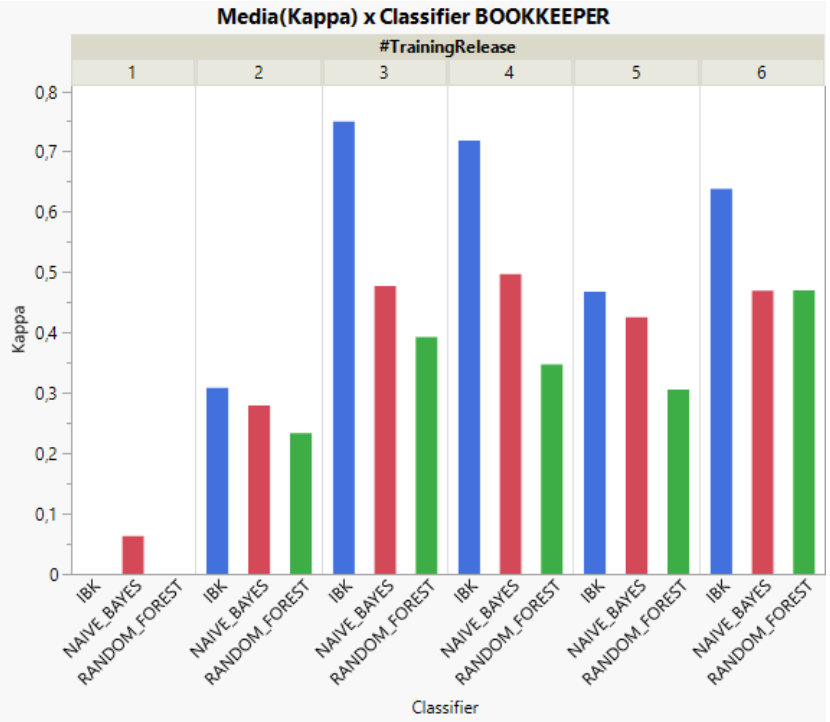
Risultati Ottenuti
Considerazioni ZOOKEEPER

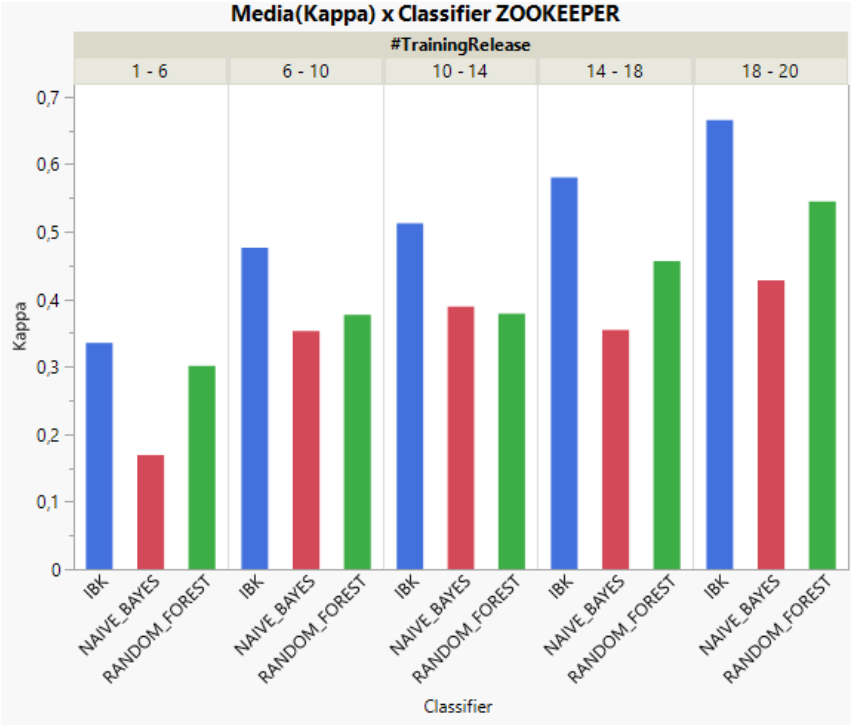
Dai box-blot precedenti, emergono alcuni risultati:

- Per qualsiasi metrica di accuratezza, i classificatori *IBK* e *Random_Forest* risultano presenti nelle migliori configurazioni, in combinazione con le tecniche considerate;
- Le tecniche di balancing che, prevalentemente, hanno fornito i migliori risultati sono state, in egual misura, *SMOTE* e *No_Sampling*;
- La metrica di accuratezza ad aver maggiormente beneficiato delle tecniche di Sensitive_Valuation è stata *Recall*;
- *Precision* è stata la metrica con il più ampio range interquartile.

Risultati Ottenuti
Confronto valori medi KAPPA

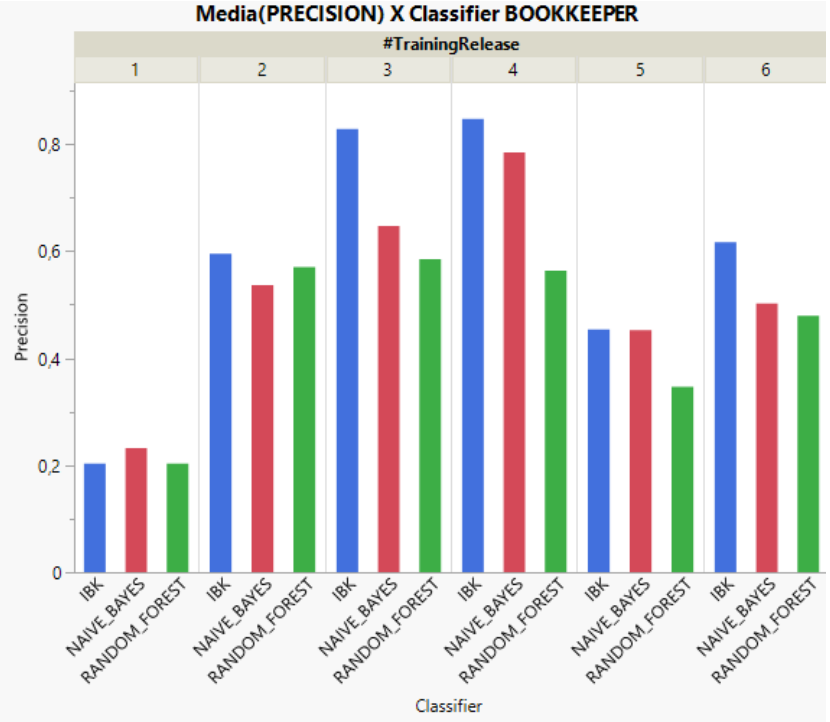
Si pongono, quindi, a confronto, i valori medi di *Kappa* ottenuti sui diversi fold di dataset utilizzati secondo Walk-Forward per ciascun classificatore, nei due progetti considerati.

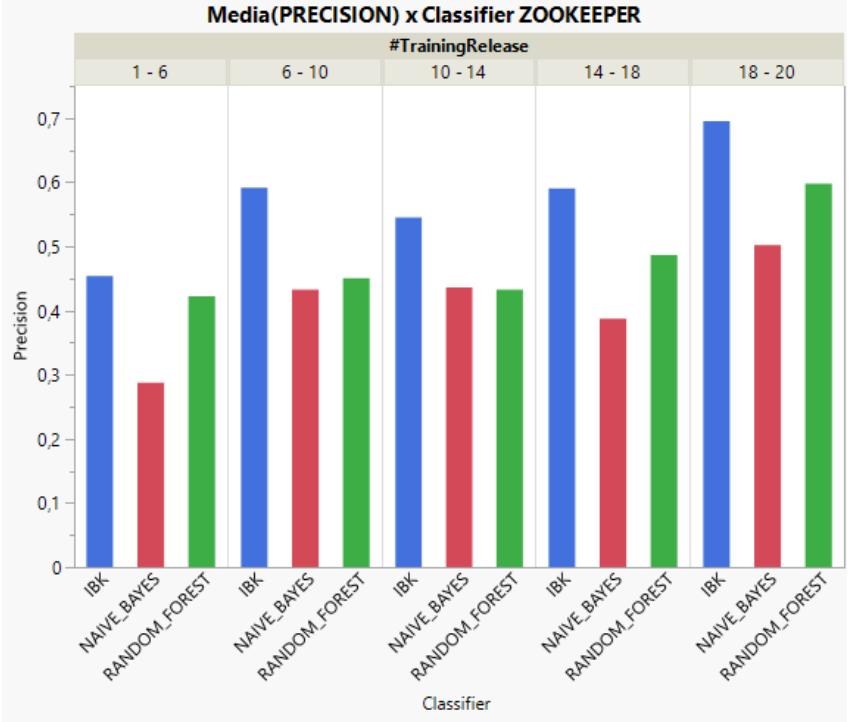




Risultati Ottenuti
Confronto valori medi PRECISION

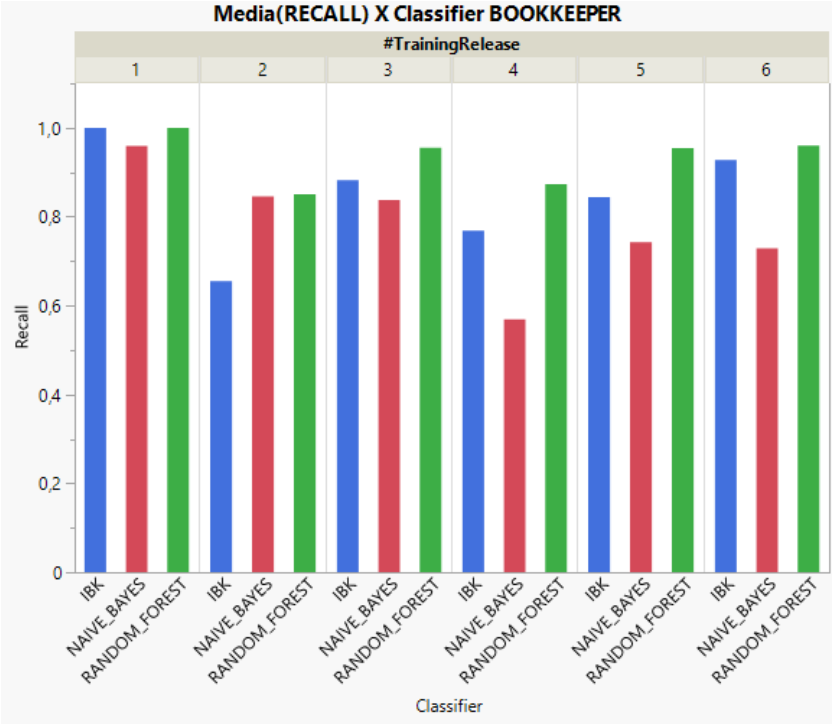
Si pongono, quindi, a confronto, i valori medi di *Precision* ottenuti sui diversi fold di dataset utilizzati secondo Walk-Forward per ciascun classificatore, nei due progetti considerati.

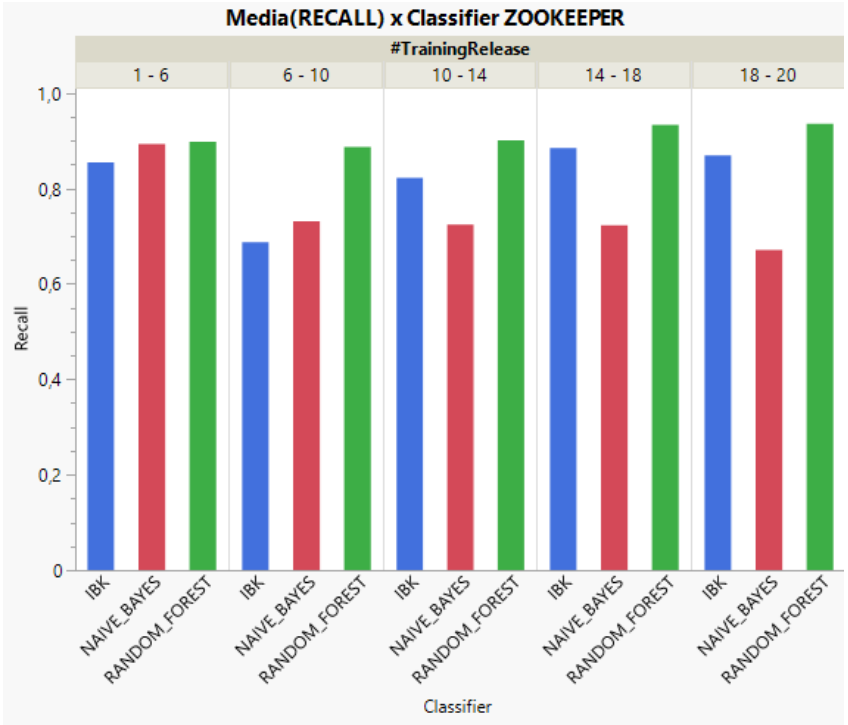




Risultati Ottenuti
Confronto valori medi RECALL

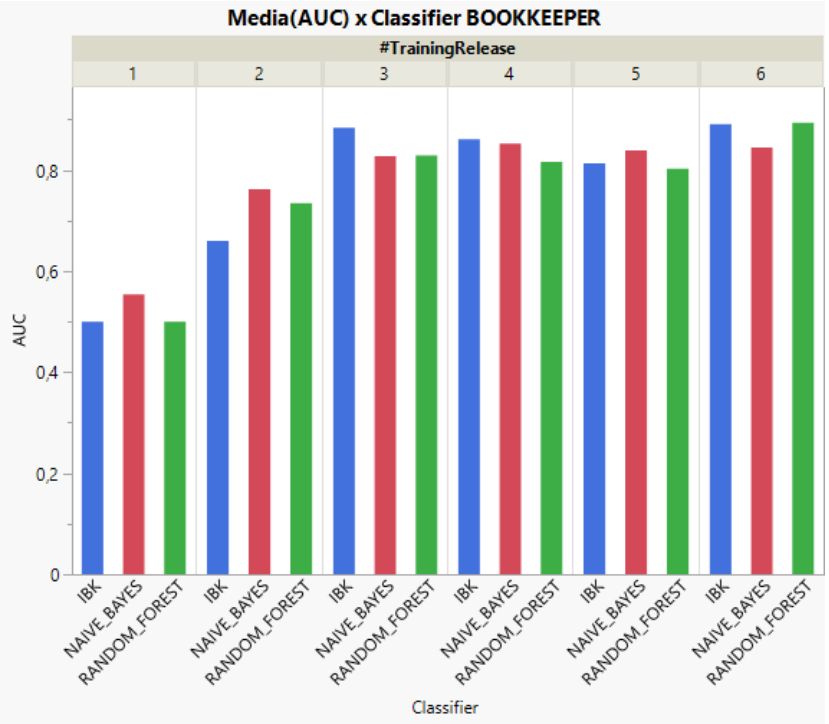
Si pongono, quindi, a confronto, i valori medi di *Recall* ottenuti sui diversi fold di dataset utilizzati secondo Walk-Forward per ciascun classificatore, nei due progetti considerati.

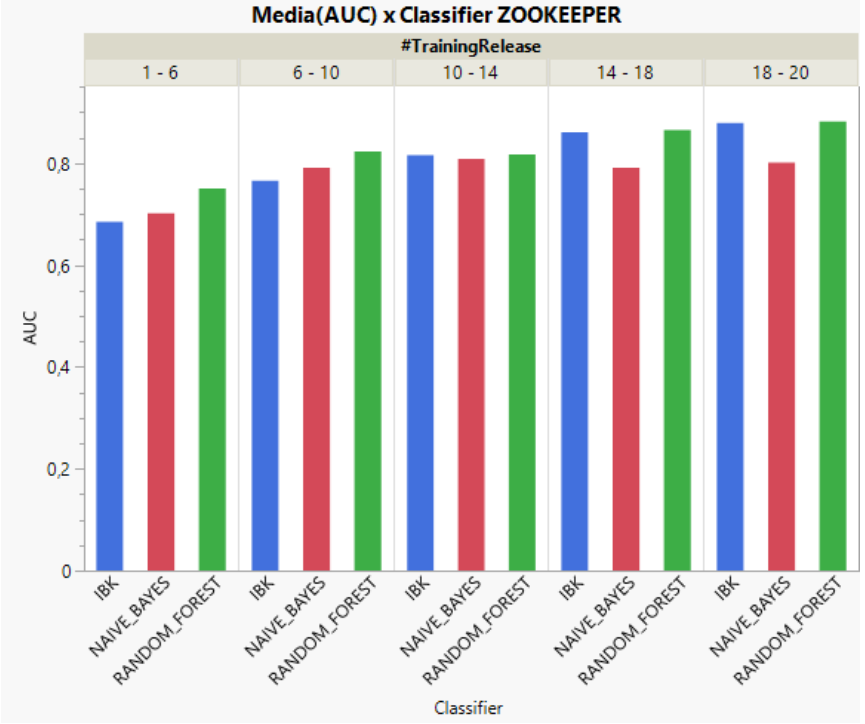




Risultati Ottenuti
Confronto valori medi AUC

Si pongono, quindi, a confronto, i valori medi di *AUC* ottenuti sui diversi fold di dataset utilizzati secondo Walk-Forward per ciascun classificatore, nei due progetti considerati.





Link GitHub e SonarCloud

Di seguito sono riportati i link della repository Github del progetto e dell'analisi dello stesso su SonarCloud.

- Repository GitHub
- Analisi Sonar