Machine Learning for Software Engineering

Indice

- Introduzione
- Progettazione
- Risultati
- Conclusioni
- Minacce alla validità
- Link

Introduzione

- **Contesto**
  - I bug costano
  - Con il testing si possono trovare e dare garanzie sulla qualità dei prodotti
  - Ma anche il testing è costoso
- **Scopo**
  - Rendere il testing più efficiente
  - Verrà effettuato predicendo quali classi conterranno bug
  - Nello studio si analizzeranno le differenze tra i modelli di predizione

Progettazione

1) Recupero informazioni

- Con **Git** sono state recuperate le release e le revisioni
- Con **Jira** sono stati recuperati i ticket di tipo bug e risolti

E’ di interesse stabilire quali classi e in quali versioni siano affette dai difetti.

Per stabilire le versioni dai ticket viene recuperato:

- Opening version (OV)
- Fixed vesion (FV)
- Affected versions (AV), se disponibili

Progettazione

2) Calcolo infected version (IV)

- Se le AV erano disponibili, la prima viene considerata come IV
- Se non lo erano, è stato utilizzato **proportion**:

$$P = \frac{FV - IV}{FV - OV} \Leftrightarrow IV = FV - (FV - OV) * P$$

  - Per il calcolo di P
    - Se erano disponibili meno di 5 ticket per la versione è stato usato **Cold Start** calcolato sugli altri progetti di Apache
    - Altrimenti viene calcolato tramite **Increment**

Progettazione

3) Individuazione classi affette

- Dalle versioni di Jira sono state ricavate le revisioni di Git
- Dalle revisioni di Git sono state ricavate le classi modificate

Progettazione

4) Scelta metriche

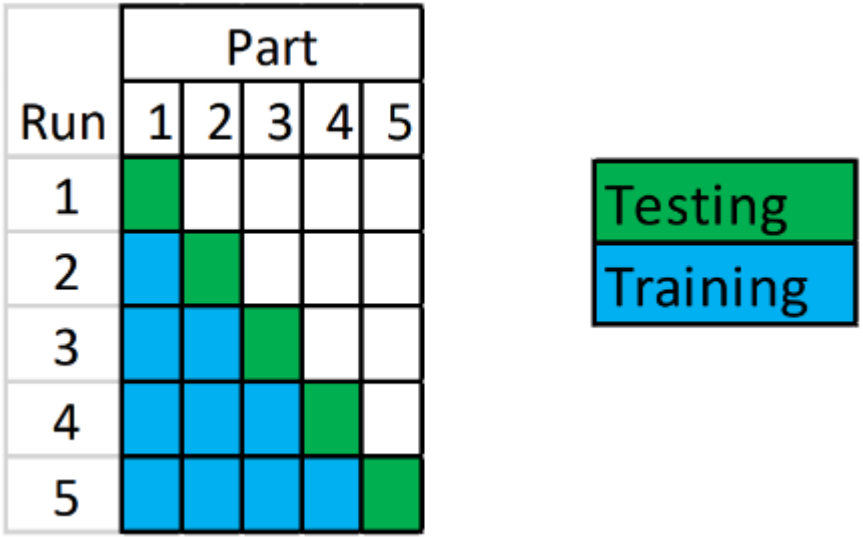
- Sono state usate metriche di classe

LoC	Loc Touched
NR	Nfix
Nauth	Avarege LoC Added
Max Loc ADded	Churn
Average Churn	Max Churn

Progettazione

5) Dataset

- Per preservare l'ordine temporale ed evitare che il testing set abbia dati antecedenti al traning set, è stato creato usando **walk-foward**



Progettazione

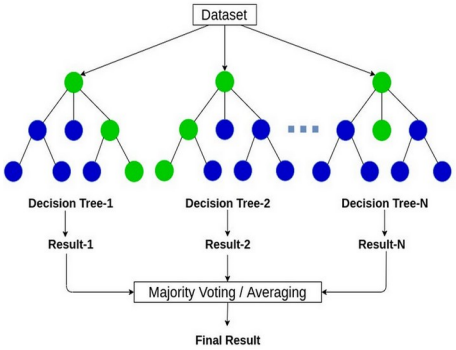
6) Metriche di performance:

- Precision $\frac{TP}{TP+FP}$
- Recall $\frac{TP}{TP+FN}$
- AUC
  - Area sotto la ROC curve
- Kappa
  - Quante volte il modello è stato più accurato di un classificatore dummy

Progettazione

7) Classificatori:

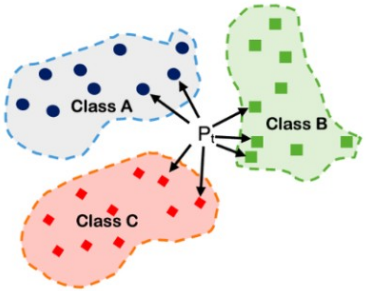
- Random forest



- Naive Bayes

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

- Ibk

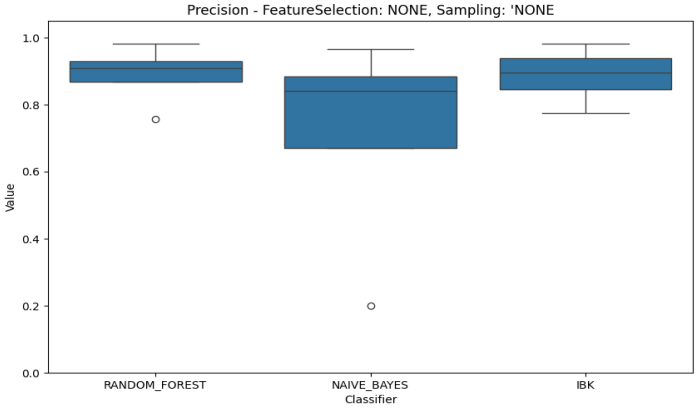


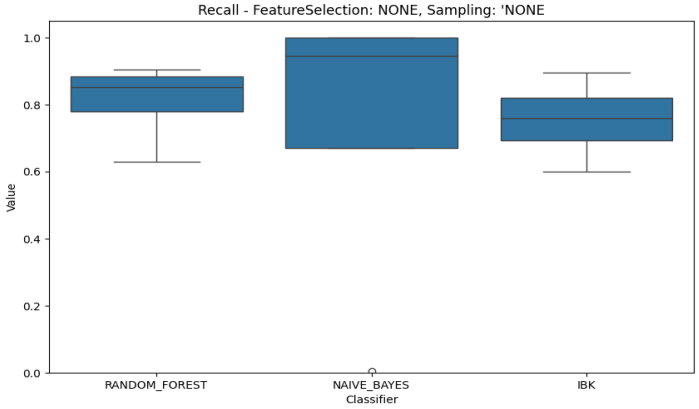
Progettazione

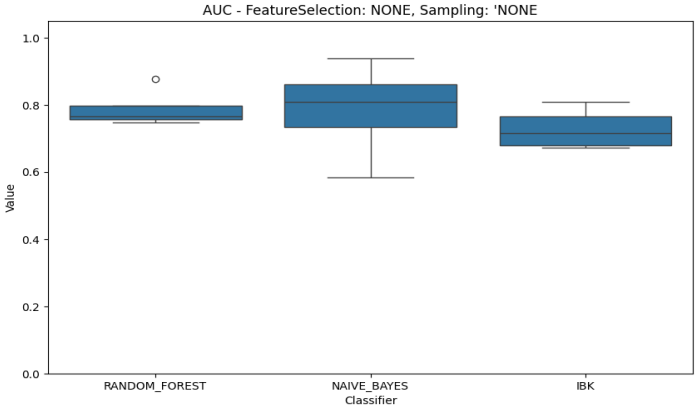
8) Variabili validate empiricamente:

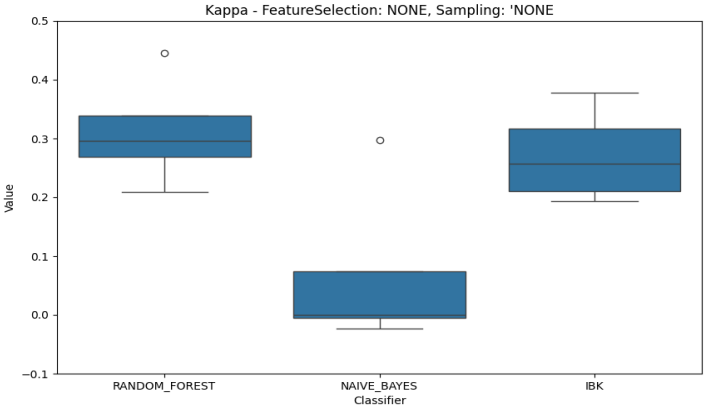
- No selection e no sampling
- Best first e no sampling
- Best first e oversampling
- Best first e undersampling
- Best first e SMOTE

Risultati BookKeeper

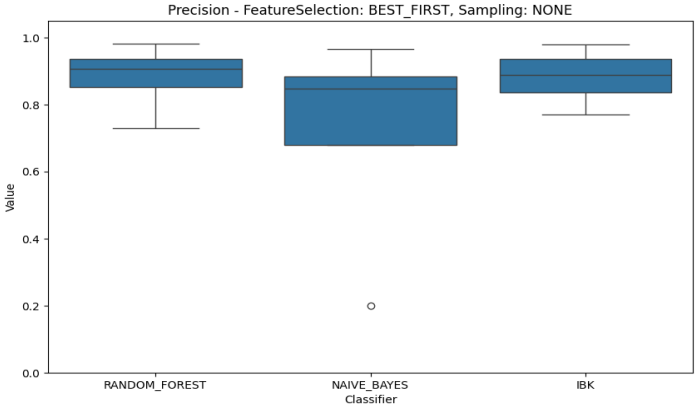


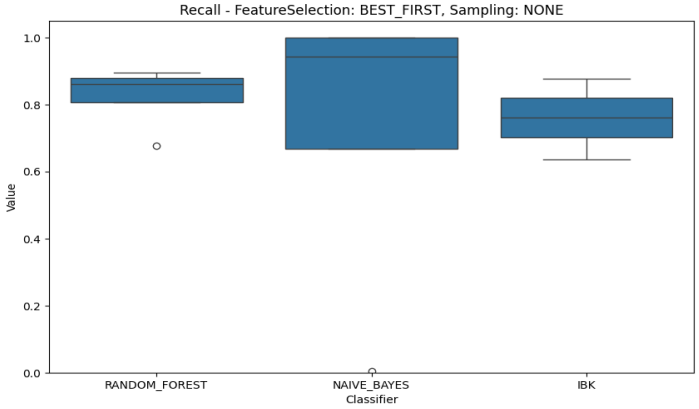


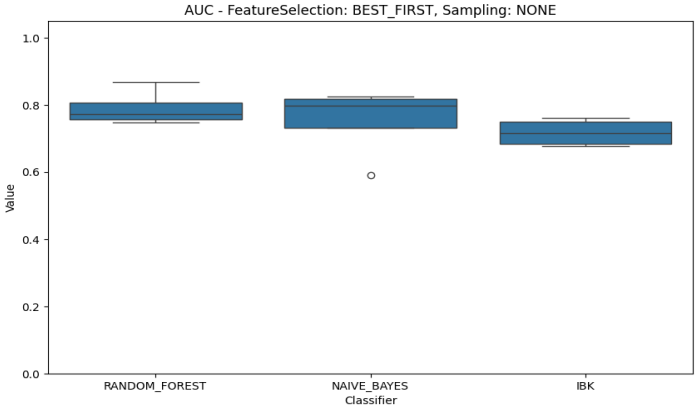


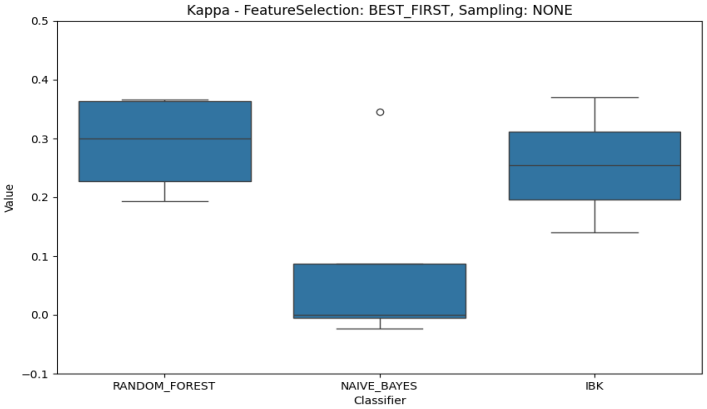


Risultati BookKeeper

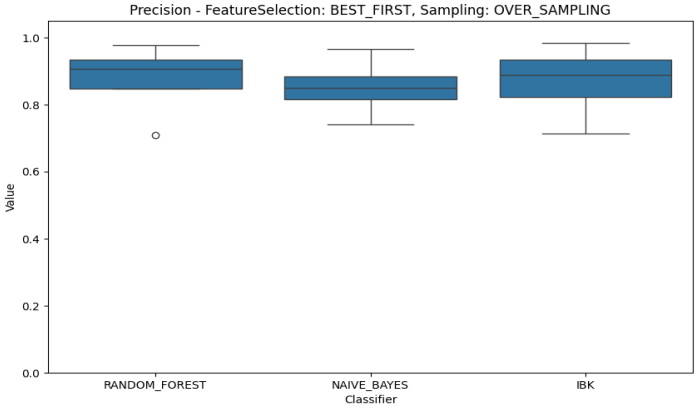


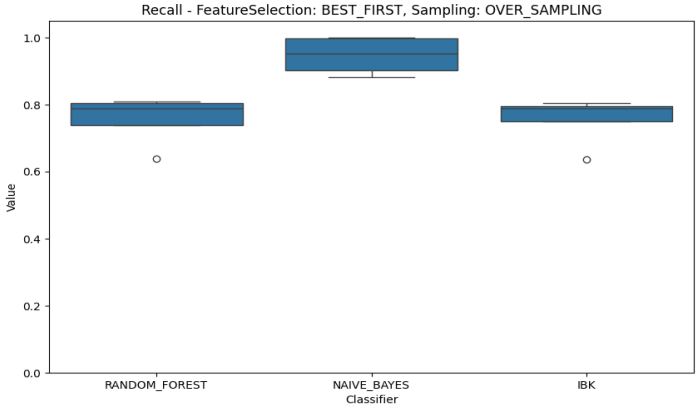


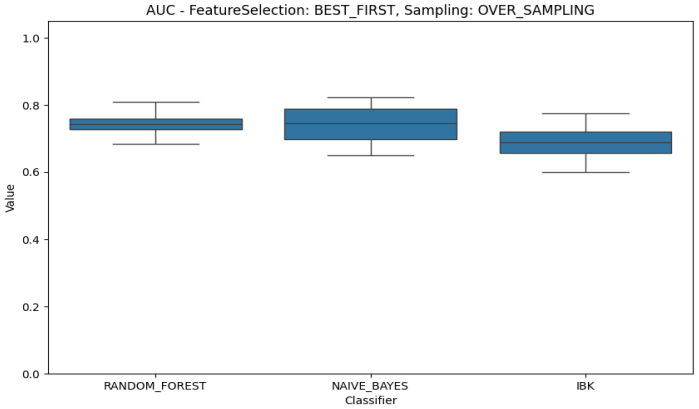


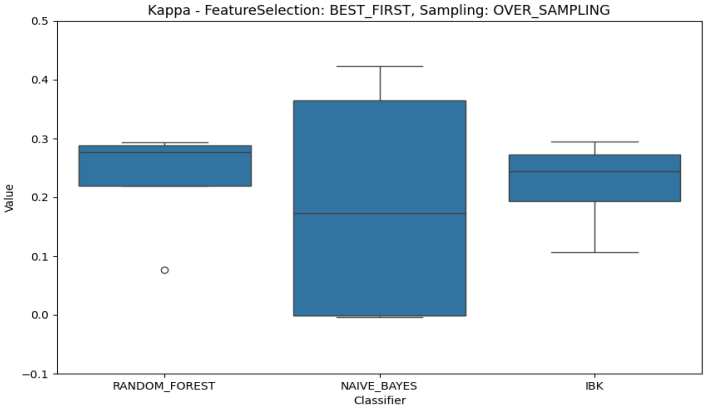


Risultati BookKeeper

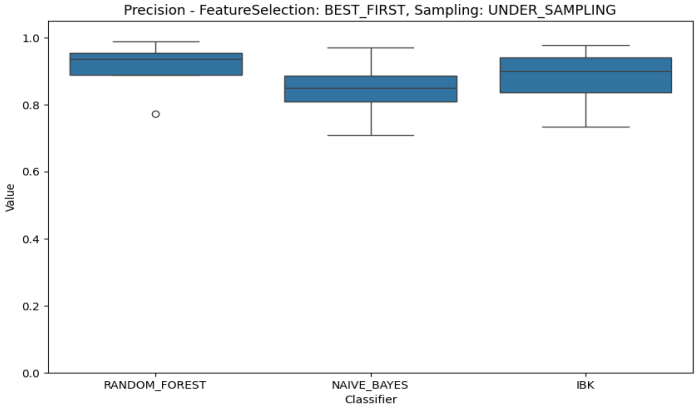


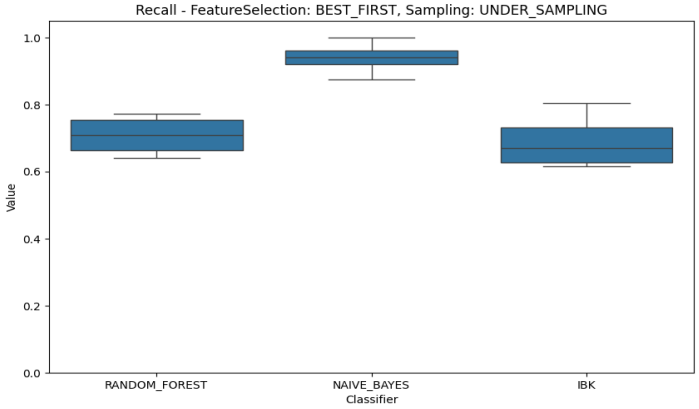


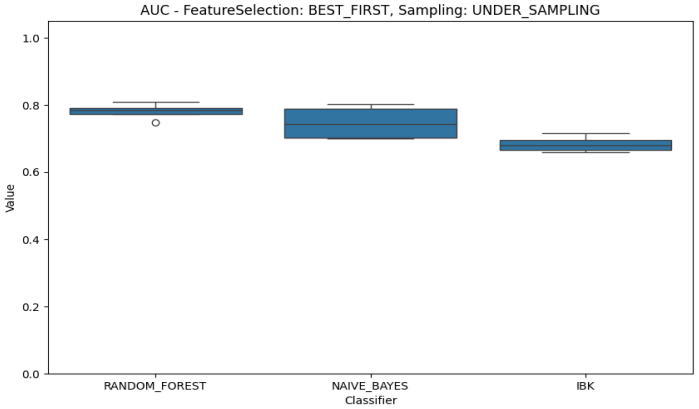


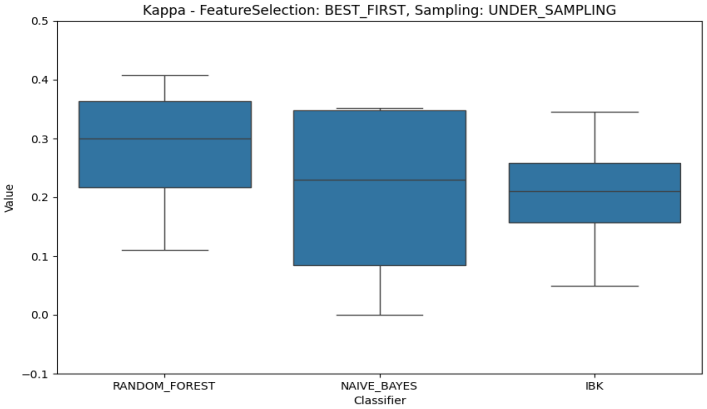


Risultati BookKeeper

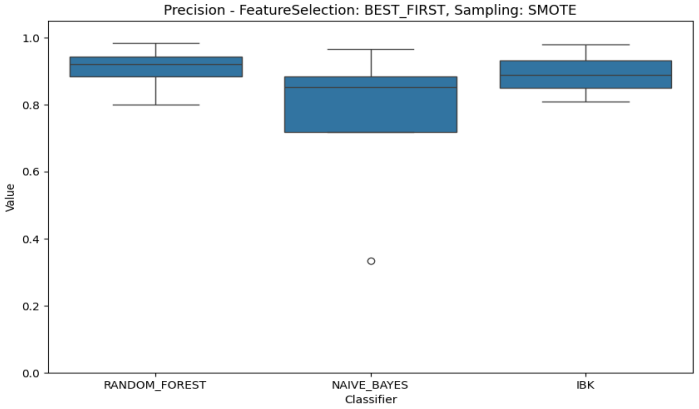


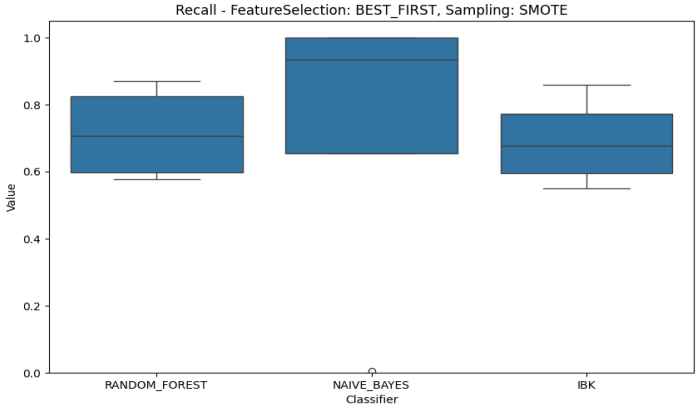


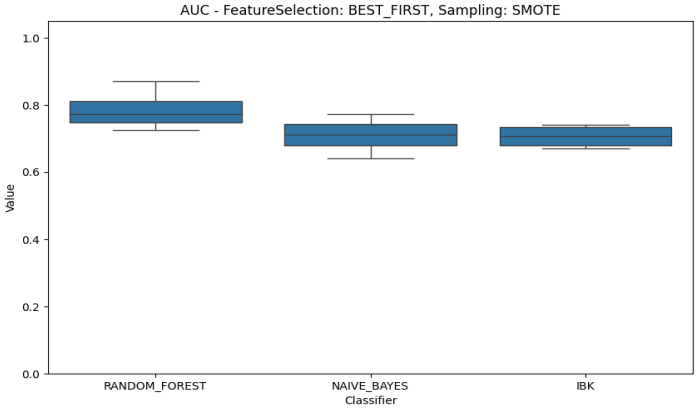


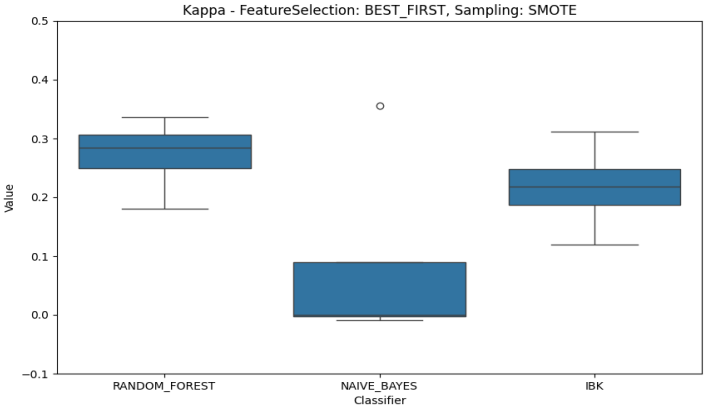


Risultati BookKeeper





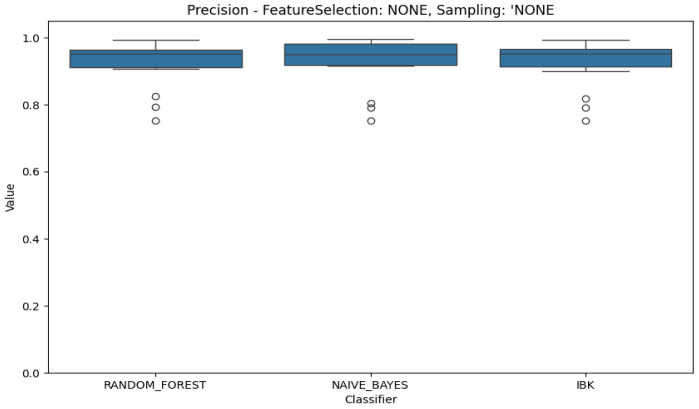


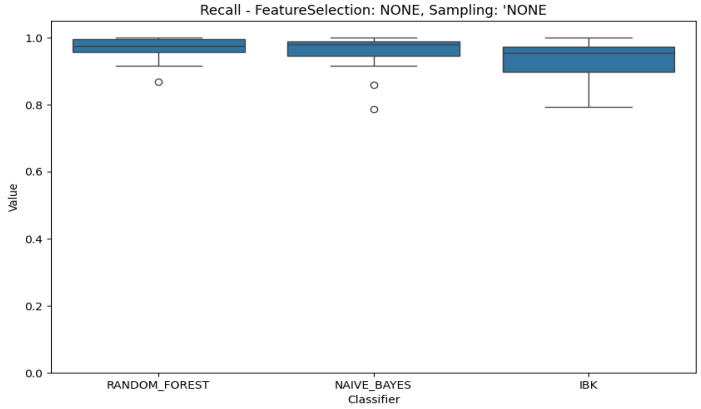


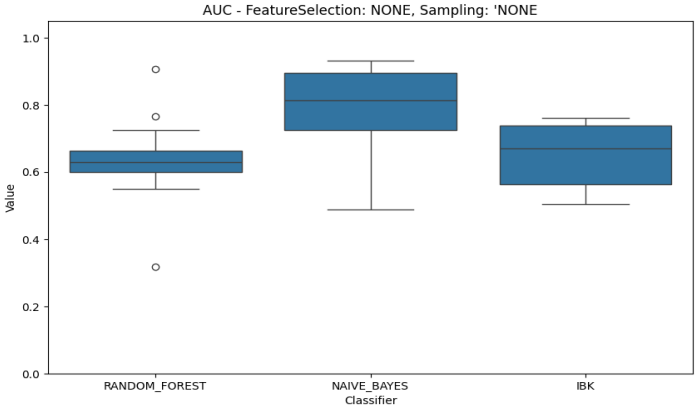
Conclusioni BookKeeper

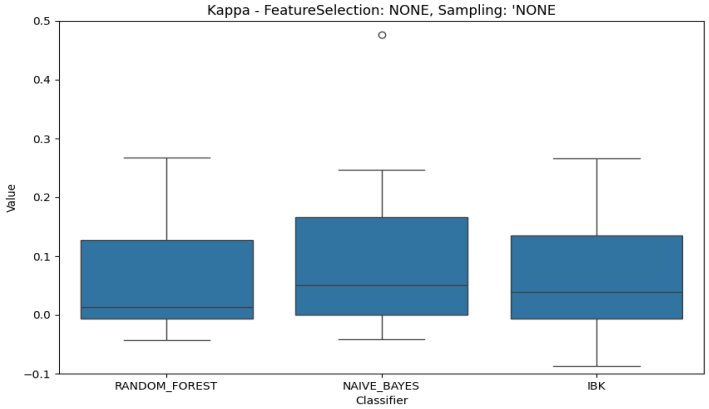
- Miglior classificatore:
  - No feature selection e no sampling **Random forest**
  - Best first e no sapling **Random forest**
  - Best first e oversampling **Naive Bayes**
  - Best first e undersampling **Naive Bayes**
  - Best first e SMOTE **Random forest**
  - Migliore combinazione in assoluto: **Best first e no sampling con Random forest**

Risultati Avro

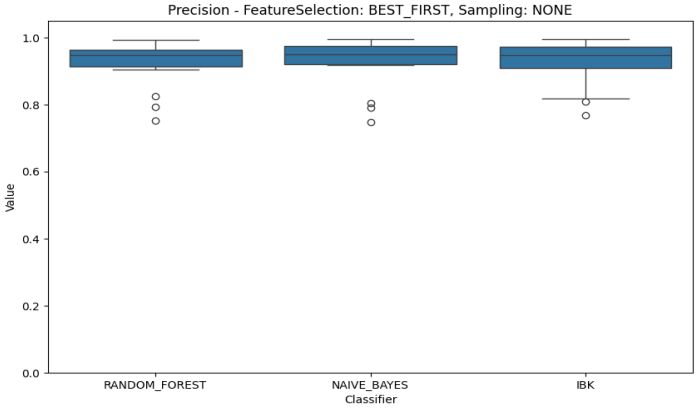


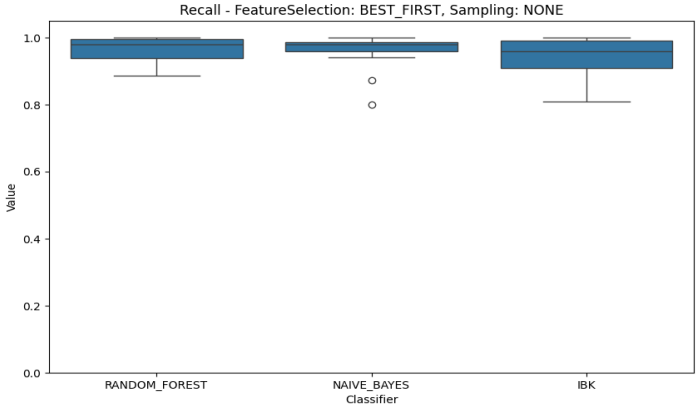


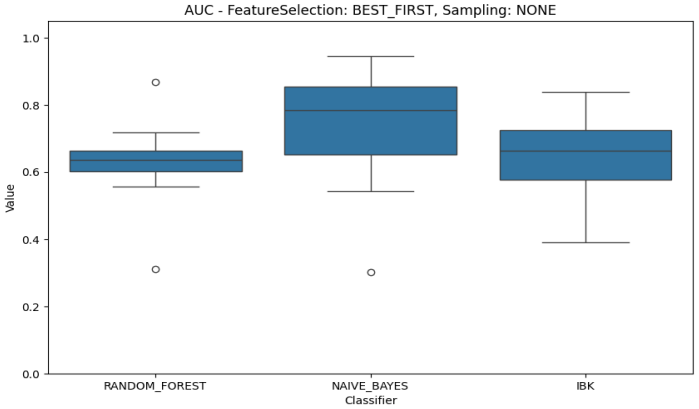


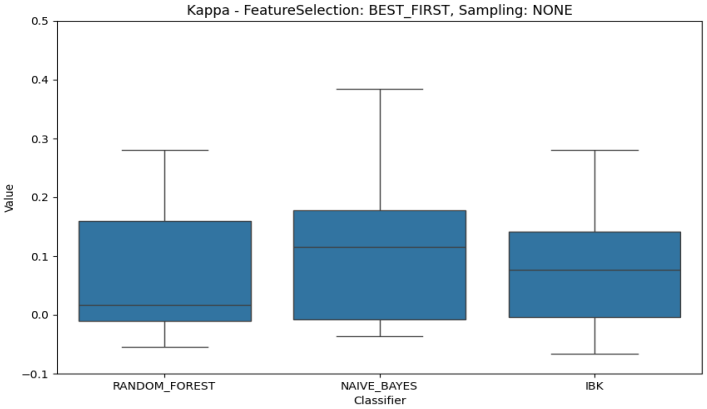


Risultati Avro

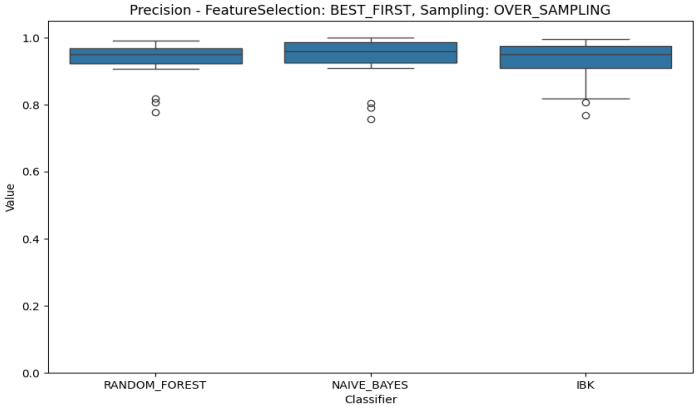


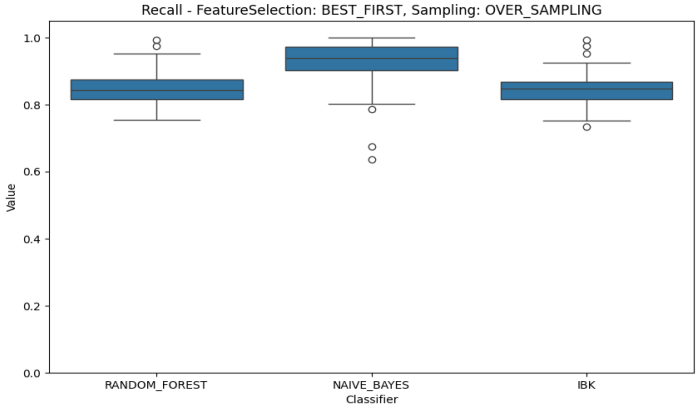


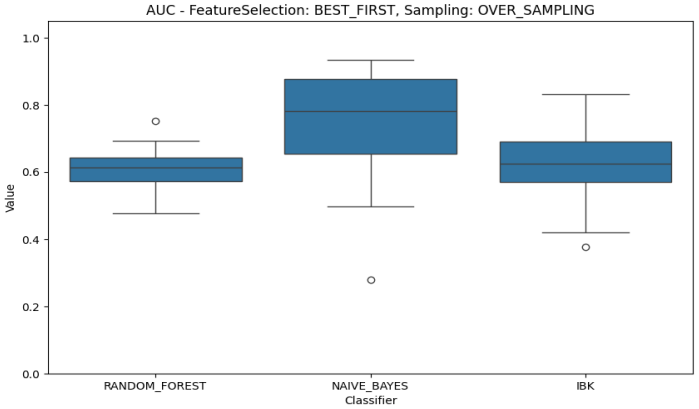


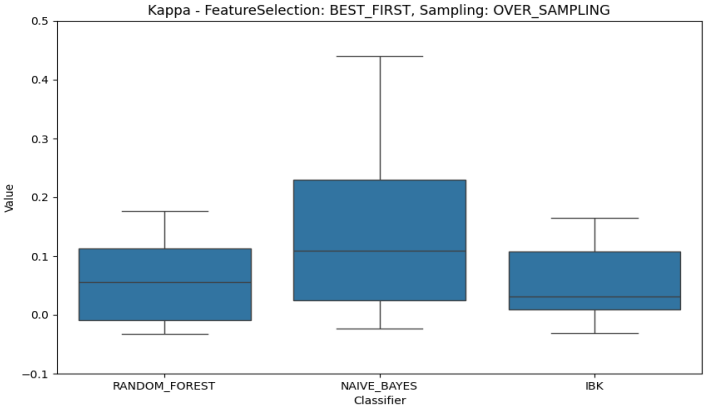


Risultati Avro

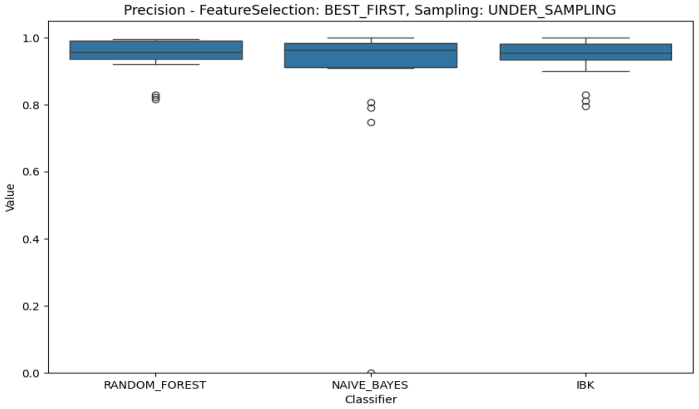


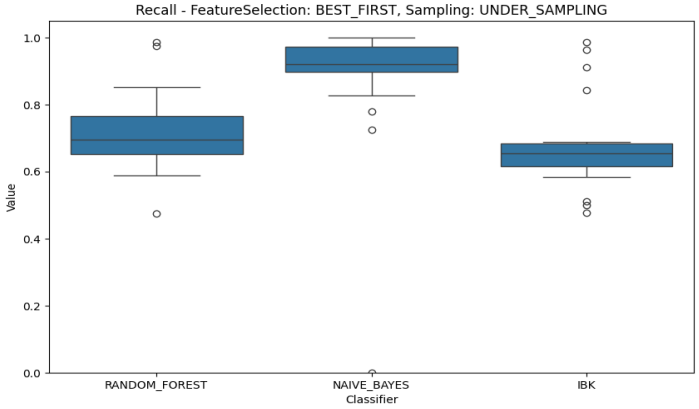


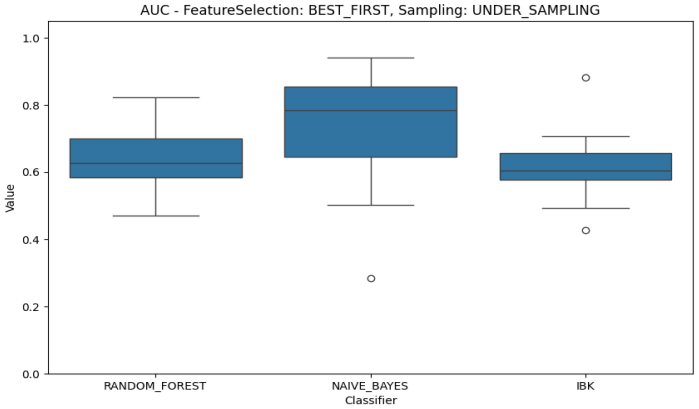


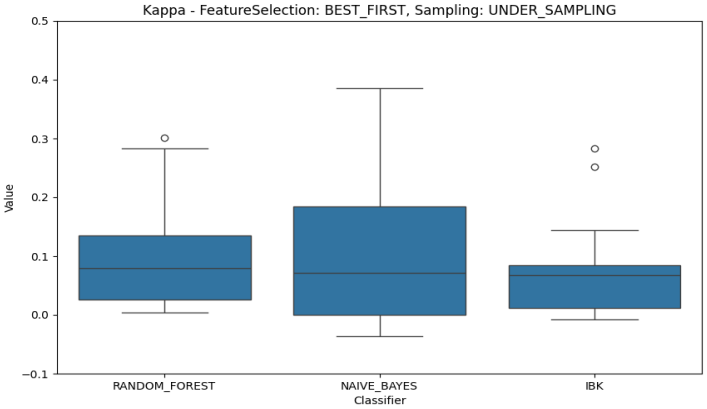


Risultati Avro

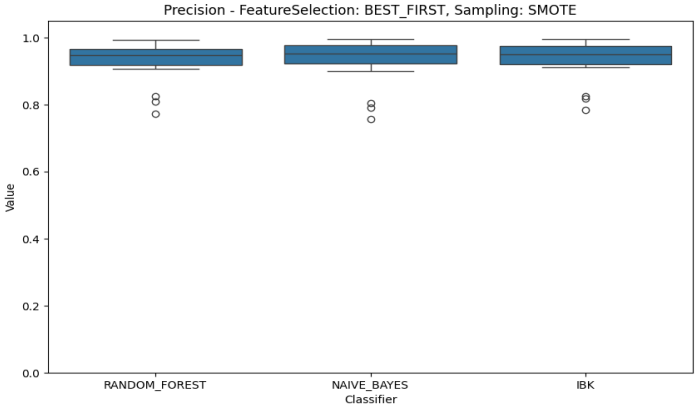


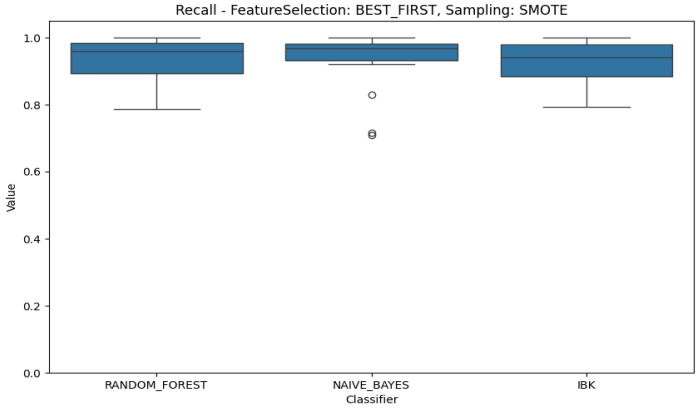


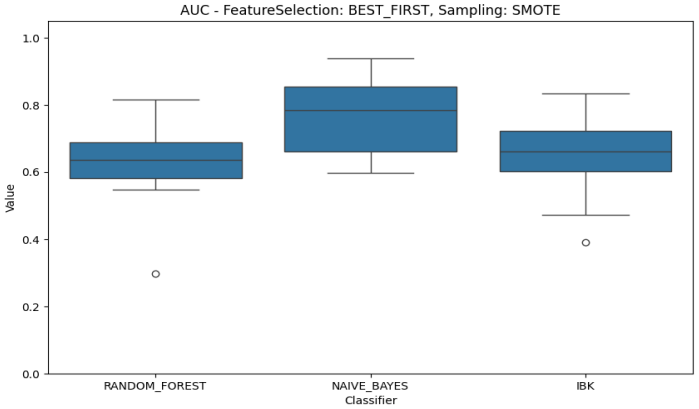


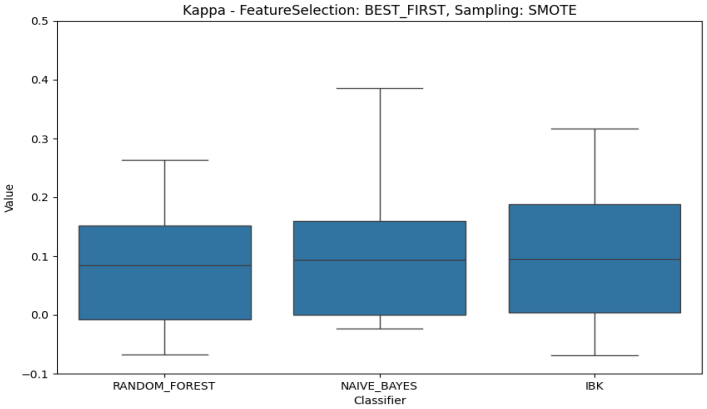


Risultati Avro









Conclusioni Avro

- Miglior classificatore:
  - No feature selection e no sampling **Naive Bayes**
  - Best first e no sapling **Naive Bayes**
  - Best first e oversampling **Naive Bayes**
  - Best first e undersampling **Naive Bayes**
  - Best first e SMOTE **Naive Bayes**
  - Migliore combinazione in assoluto: **Best first e oversamplin sampling con Naive Bayes**

Minacce alla validità

- Considerando il numero di versioni e il cambiamento frequente delle classi in Avro, potrebbe essere più appropriato **Moving Window** per il calcolo di proportion

- Con Cold Start vengono considerati pochi progetti e il loro unico legame è l'appartenenza al gruppo Apache

Link

- GitHub:
  - https://github.com/GRonz00/ispw2

- SonarClouad:
  - https://sonarcloud.io/project/overview?id=GRonz00_ispw22