

# GAME RATINGS PREDICTOR

Classificazione di videogiochi in base alla fascia  
d'età assegnata dall'ESRB  
(Entertainment Software Rating Board)

Ingegneria Informatica – Università di Roma “La Sapienza”  
Corso di Metodi Quantitativi per l’Informatica

A cura di Roberto Falconi e Federico Guidi

# INTRODUZIONE

- ▶ Applicare gli insegnamenti ricevuti a lezione
- ▶ Sviluppare un progetto pratico che implementi i metodi del Machine Learning
- ▶ Valutare l'operato e renderlo pubblico

Roberto Falconi  
Federico Guidi

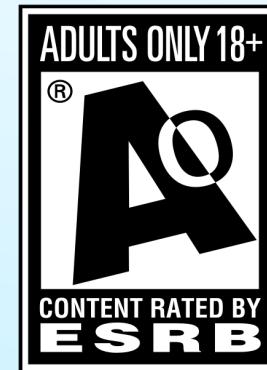
# PROCEDIMENTO SEGUITO

- ▶ Scegliere un campo d'azione e un dataset (industria videoludica)
- ▶ Definire un problema da affrontare (multiclass classification)
  - ▶ Applicare tecniche di discretizzazione (one-hot encoding)
  - ▶ Ritornare al problema multiclass  
(normalizzando i risultati finali ed applicando la majority rule)
- ▶ Individuare i classificatori opportuni (Random Forest, Logistic Regression, k-NN)
  - ▶ Impiegare algoritmi di inferenza (coarse-to-fine)
- ▶ Optare per un linguaggio di programmazione opportuno (Python)
- ▶ Prediligere dei tipi di stima e dei metodi di validazione  
(confidence, cross-validation, misclassification rate e accuracy score)

Roberto Falconi  
Federico Guidi



ENTERTAINMENT SOFTWARE RATING BOARD



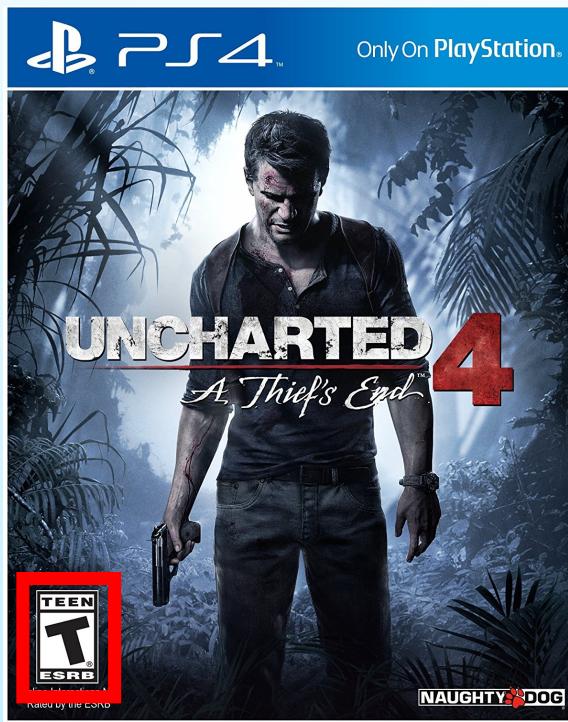
Roberto Falconi  
Federico Guidi



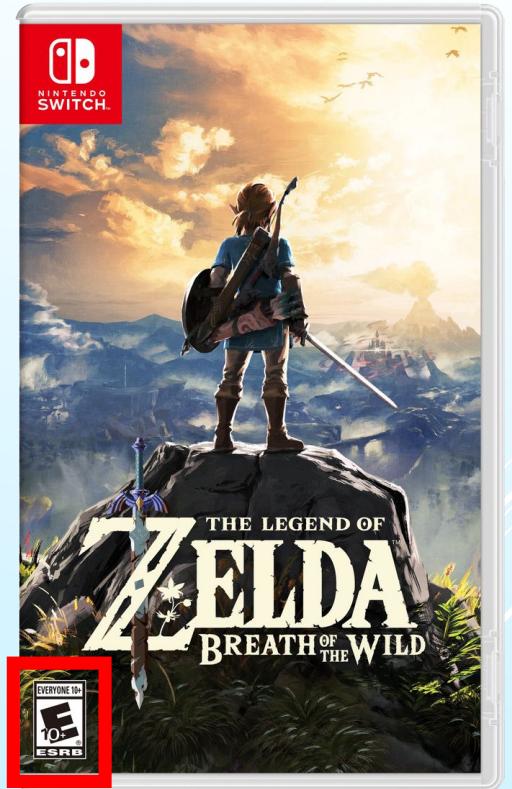
Super Mario Sunshine  
2002 (E)



Grand Theft Auto V  
2013 (M)



Uncharted 4  
2016 (T)



The Legend of Zelda  
Breath of the Wild  
2017 (E10+)

Roberto Falconi  
Federico Guidi

Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	Developer	Rating
FIFA 17	PS4	2016	Sports	Electronic Arts	0.66	5.75	0.08	1.11	7.59	85	41	5	398	EA Sports, EA Vancouver	E
Uncharted 4: A Thief's End	PS4	2016	Shooter	Sony Computer Entertainment	1.85	2.5	0.19	0.85	5.38	93	113	7.9	7064	Naughty Dog	T
Call of Duty: Infinite Warfare	PS4	2016	Shooter	Activision	1.61	2	0.15	0.71	4.46	77	82	3.4	1129	Infinity Ward	M
Battlefield 1	PS4	2016	Shooter	Electronic Arts	1.1	2.15	0.21	0.61	4.08	88	31	8.4	809	EA DICE	M
Tom Clancy's The Division	PS4	2016	Shooter	Ubisoft	1.35	1.7	0.15	0.6	3.8	80	64	7	2219	Massive Entertainment	M
FIFA 17	XOne	2016	Sports	Electronic Arts	0.43	2.05	0	0.17	2.65	84	50	5.5	201	EA Sports, EA Vancouver	E
Call of Duty: Infinite Warfare	XOne	2016	Shooter	Activision	1.46	0.74	0	0.22	2.42	78	17	3.1	290	Infinity Ward	M
Far Cry: Primal	PS4	2016	Action	Ubisoft	0.6	1.25	0.06	0.35	2.26	76	91	6.3	635	Ubisoft Montreal	M
Battlefield 1	XOne	2016	Shooter	Electronic Arts	1.28	0.77	0	0.2	2.25	87	37	8.2	440	EA DICE	M
Tom Clancy's The Division	XOne	2016	Shooter	Ubisoft	1.29	0.68	0	0.2	2.16	80	33	6.9	614	Massive Entertainment	M
Overwatch	PS4	2016	Shooter	Activision	0.81	0.85	0.15	0.33	2.14	90	31	6.1	1358	Blizzard Entertainment	T
NBA 2K17	PS4	2016	Sports	Take-Two Interactive	1.25	0.27	0.02	0.34	1.88	88	47	6.7	162	Visual Concepts	E
Mafia III	PS4	2016	Action	Take-Two Interactive	0.42	1.08	0.03	0.28	1.81	68	66	5.1	1147	Hangar 13	M
Madden NFL 17	PS4	2016	Sports	Electronic Arts	1.25	0.17	0	0.32	1.75	82	35	4.9	83	EA Sports	E
No Man's Sky	PS4	2016	Action	Hello Games	0.63	0.76	0.03	0.27	1.7	71	94	4.5	5096	Hello Games	T
Dark Souls III	PS4	2016	Role-Playing	Namco Bandai Games	0.65	0.45	0.34	0.22	1.65	89	69	8.8	1940	From Software	M

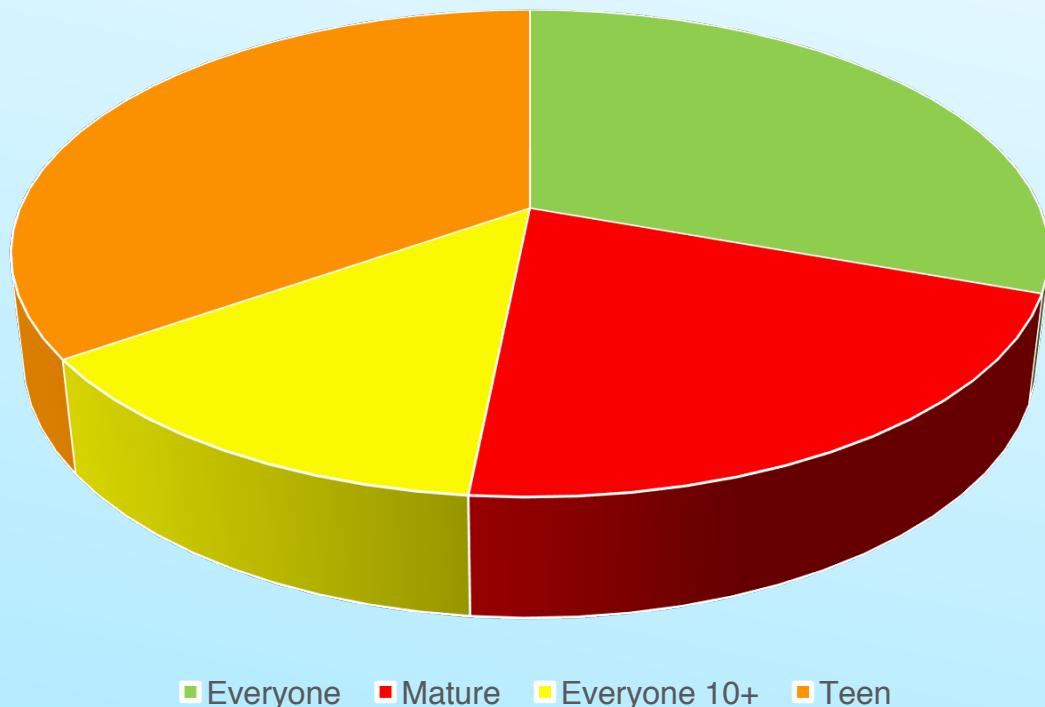
# DATASET ANALYSIS

## PARTE DEL DATASET

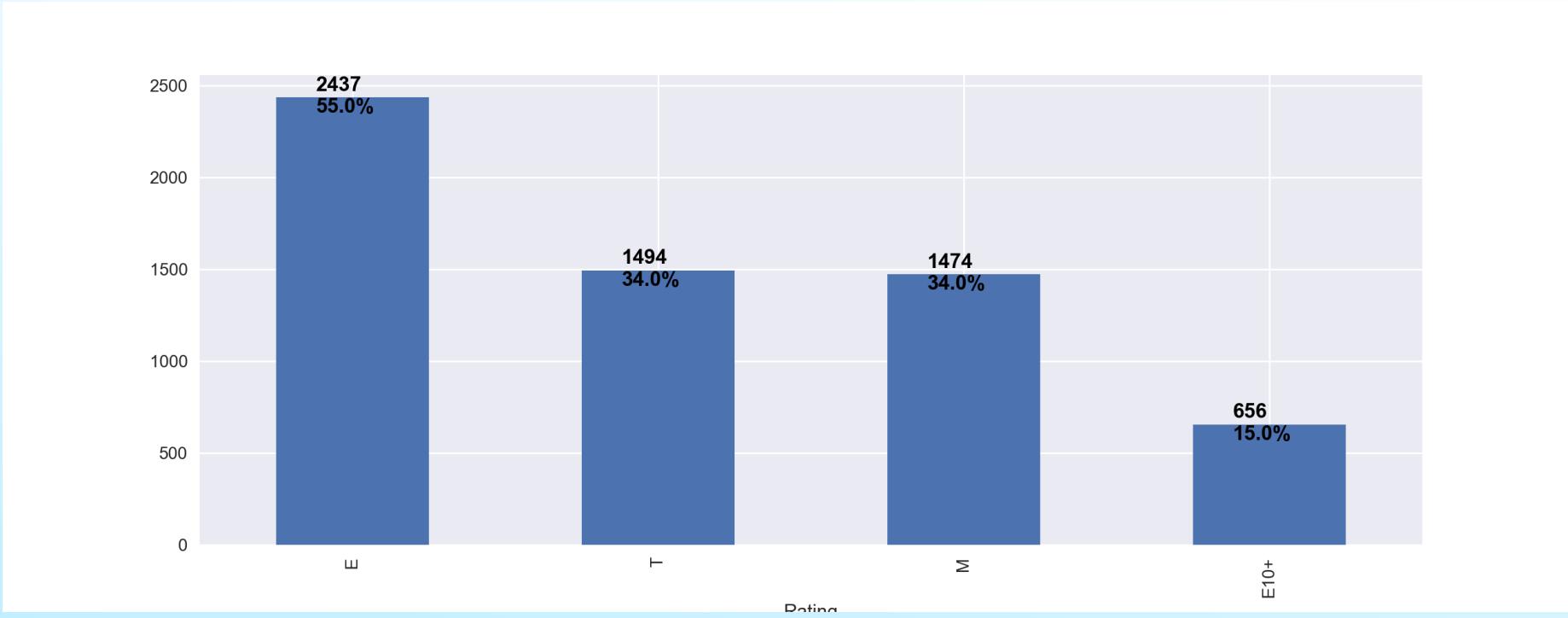
Roberto Falconi  
Federico Guidi

# RIPARTIZIONE DELLE CLASSI NEL DATASET

Numero di elementi per classe



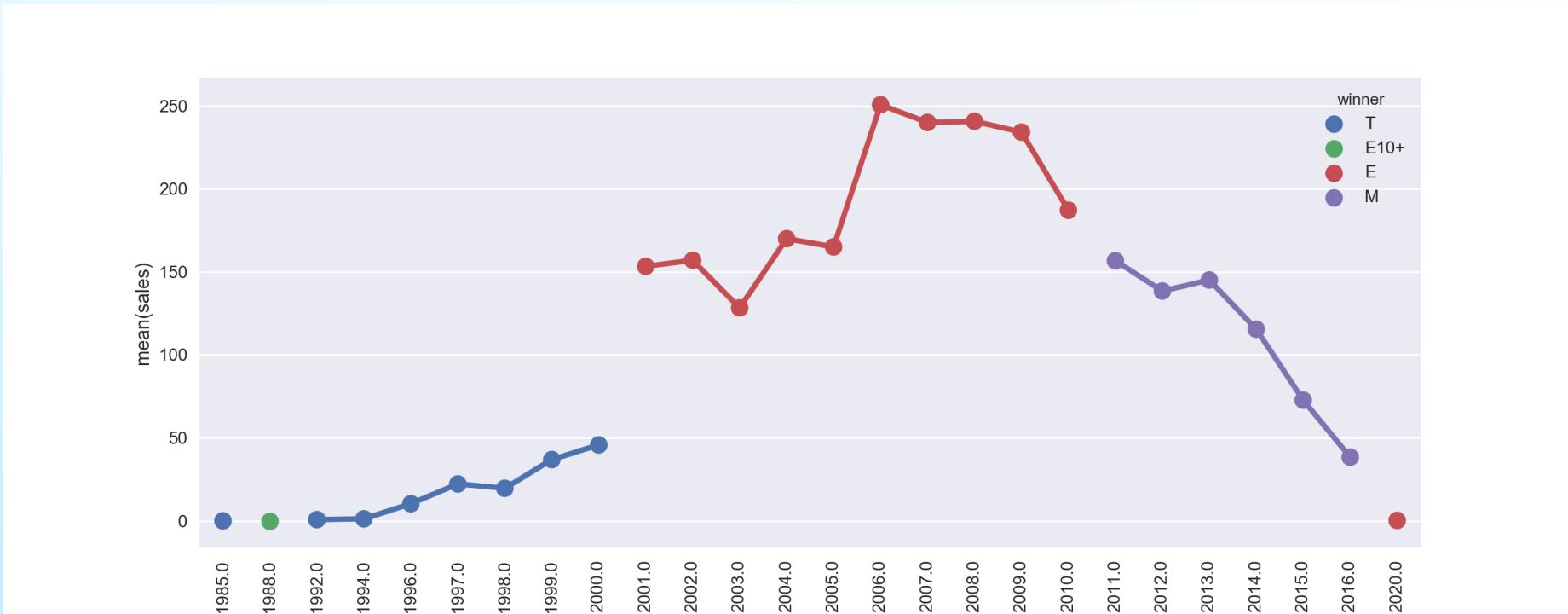
Roberto Falconi  
Federico Guidi



# DATASET ANALYSIS

## L'IMPORTANZA DEL RATING NELLE VENDITE

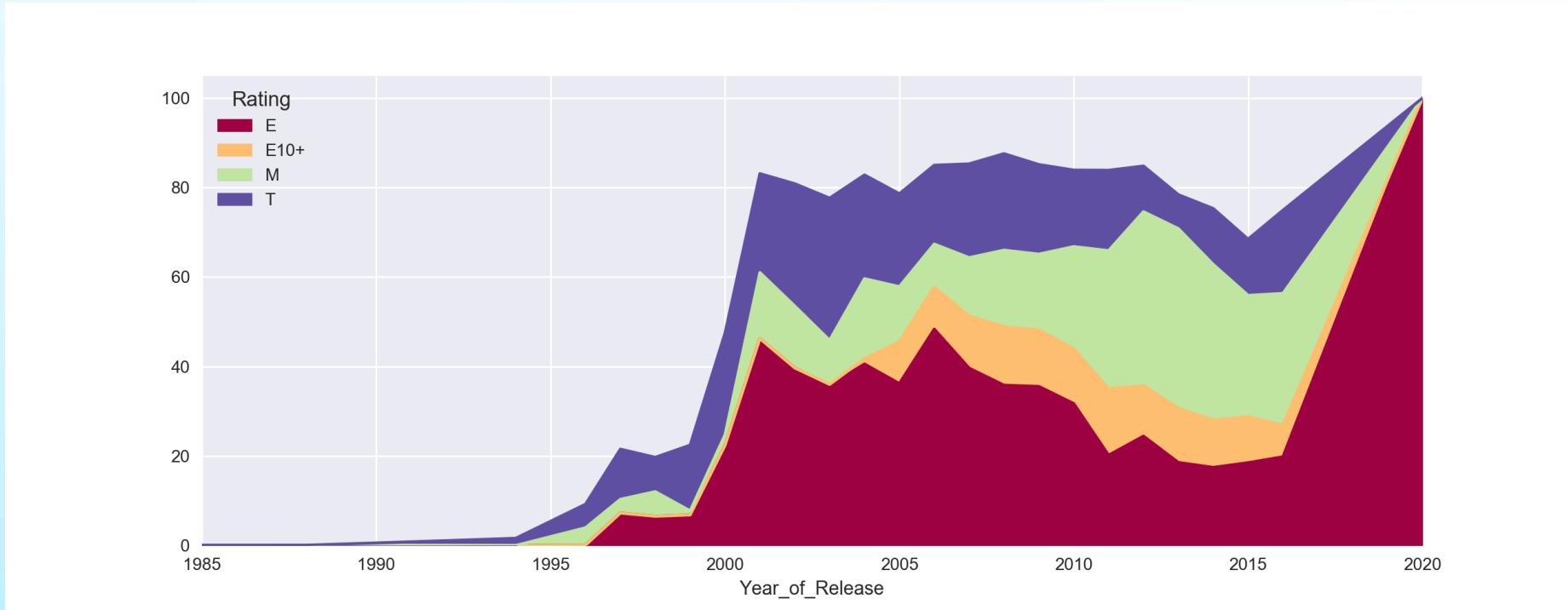
Roberto Falconi  
Federico Guidi



# DATASET ANALYSIS

## L'IMPORTANZA DEL RATING NELLE VENDITE

Roberto Falconi  
Federico Guidi



# DATASET ANALYSIS

## L'IMPORTANZA DEL RATING NELLE VENDITE

Roberto Falconi  
Federico Guidi

# INSTALLAZIONE

UBUNTU, DEBIAN E MACOS

Necessario: **Python Developer** e **pip**

```
>> sudo apt-get install python-dev
```

Assicuriamoci di avere pip alla sua ultima versione

```
>> pip install --upgrade pip
```

Cloniamo la repository in una cartella a piacere

```
>> git clone https://gitlab.com/mqpi_2016_17/GameRatingsPredictor
```

Spostandoci nella cartella GameRatingsPredictor, installiamo i package Python richiesti dall'applicazione

```
>> sudo pip install -r requirements.txt
```

Possiamo far partire il programma direttamente con il comando

```
>> python algoritmo-runner.py
```

In alternativa, possiamo decidere di installarlo permanentemente sul nostro dispositivo tramite

```
>> sudo python setup.py install --record files.txt
```

disinstallarlo con

```
>> uninstall it with cat files.txt | xargs rm -rf
```

e infine farlo partire ovunque nel termine tramite il semplice comando

```
>> algoritmo
```

(Per l'installazione su Windows si veda la relazione)

Roberto Falconi  
Federico Guidi

# PREPARAZIONE DEL DATASET

ELIMINAZIONE ELEMENTI INCOMPLETI

Name	Rating
Super Mario	E
FIFA	T
Pokémon	E10
Tetris	NaN



Name	Rating
Super Mario	E
FIFA	T
Pokémon	E10

Roberto Falconi  
Federico Guidi

# PREPARAZIONE DEL DATASET

APPLICAZIONE ONE-HOT ENCODING (DISCRETIZZAZIONE)

Name	Rating
Super Mario	E
FIFA	T
Pokémon	E10



Name	Rating_E	Rating_E10	Rating_T
Super Mario	1	0	0
FIFA	0	0	1
Pokémon	0	1	0

# PREPARAZIONE DEL DATASET

SEPARAZIONE IN TRAINING SET E TEST SET

Name	Rating_E	Rating_E10	Rating_T
Super Mario	1	0	0
FIFA	0	0	1
Pokémon	0	1	0



Name	Rating_E	Rating_E10	Rating_T
Super Mario	1	0	0
FIFA	0	0	1

Name	Rating_E	Rating_E10	Rating_T
Pokémon	0	1	0

Roberto Falconi  
Federico Guidi

# PREPARAZIONE DEL DATASET

## SEPARAZIONE IN TRAINING SET E TEST SET

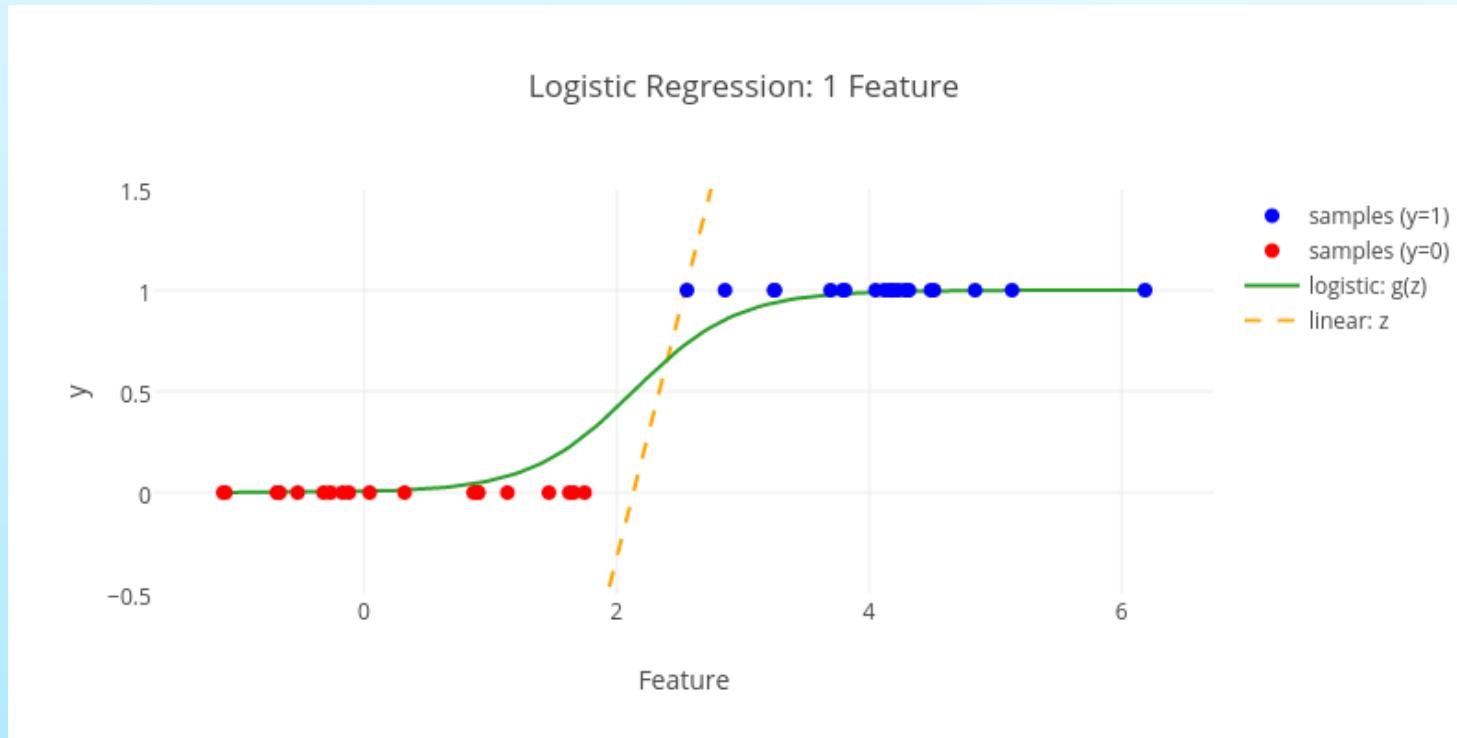
- ▶ Il dataset viene inizialmente mescolato
- ▶ Viene poi scelto, casualmente, l'**80%** degli elementi del dataset per comporre il **training set**
- ▶ Il restante **20%** degli elementi rappresenta il **test set**

# LOGISTIC REGRESSION

- ▶ È un **modello di regressione** dove la variabile dipendente è di tipo categorico, ovvero può assumere determinati valori corrispondenti a determinate classi. Nel nostro caso, la variabile dipendente è binaria e può assumere solo due valori, ovvero se l'elemento (videogioco) in analisi appartiene o meno a una determinata classe (fascia d'età assegnata dall'ESRB).

Roberto Falconi  
Federico Guidi

# LOGISTIC REGRESSION



► Grafico della curva della Logistic Regression; mostra la probabilità che un evento si verifichi

# LOGISTIC REGRESSION

IL CODICE IN PYTHON

```
LogisticRegression(penalty='l1', dual=False,  
C=1.0, fit_intercept=True, intercept_scaling=1,  
class_weight=None, random_state=None,  
solver='liblinear', max_iter=100,  
multi_class='ovr', verbose=0, warm_start=False,  
n_jobs=-1)
```

Roberto Falconi  
Federico Guidi

# LOGISTIC REGRESSION

## Pro

- ▶ Può gestire andamenti non lineari
- ▶ Fornisce in output una probabilità, che può essere utilizzata per misurare la confidenza della predizione
- ▶ Le variabili indipendenti non devono avere uguale varianza per ogni gruppo
- ▶ Non assume una relazione lineare tra variabili dipendenti e indipendenti

## Contro

- ▶ Richiede un dataset di larghe dimensioni
- ▶ I dati in input devono essere strettamente correlati tra loro
- ▶ Vulnerabile alla presenza di dati non significativi
- ▶ Vulnerabile alla overconfidence



Roberto Falconi  
Federico Guidi

# RANDOM FOREST

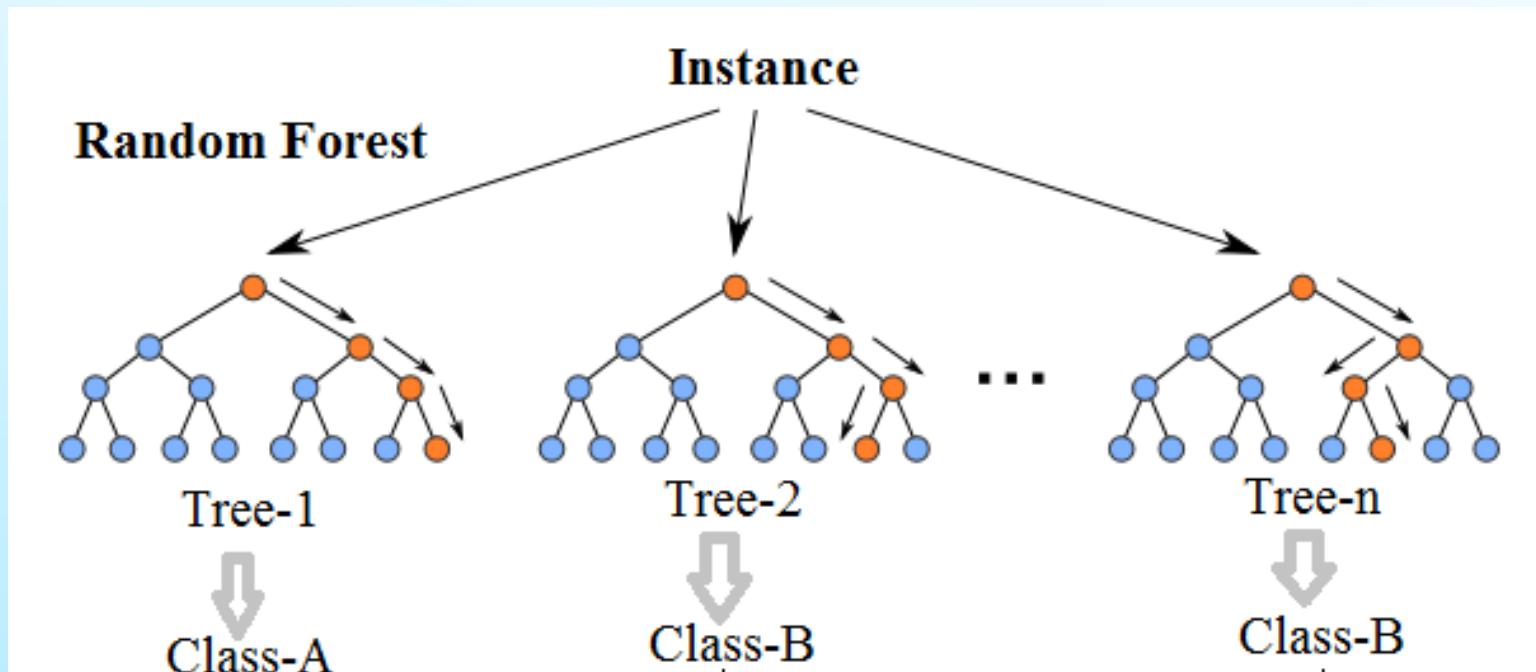
- Le **Random Forest** sono un metodo di apprendimento d'insieme (utilizzato per la classificazione, regressione e altro) che operano costruendo una moltitudine di alberi di decisione durante l'allenamento del modello e dando come output la classe che è la moda delle classi (nel caso della classificazione) oppure la predizione della media (nel caso della regressione) dei singoli alberi.



Roberto Falconi  
Federico Guidi

# RANDOM FOREST

- ▶ Esempio generale semplificato dell'operato del Random Forest; mostra i procedimenti seguiti per effettuare una classificazione



# RANDOM FOREST

IL CODICE IN PYTHON

```
RandomForestClassifier(n_estimators=240,  
criterion='gini', max_depth=None, min_samples_split=2,  
min_samples_leaf=1, min_weight_fraction_leaf=0.0,  
max_features='auto', max_leaf_nodes=None,  
min_impurity_split=1e-07, bootstrap=True, oob_score=True,  
n_jobs=1, random_state=None, verbose=0, warm_start=False,  
class_weight=None)
```

Roberto Falconi  
Federico Guidi

# RANDOM FOREST

## Pro

- ▶ Uno degli algoritmi di apprendimento più accurati disponibili
- ▶ Si rivela efficiente anche su grandi basi di dati
- ▶ Può gestire senza problemi migliaia di variabili di ingresso
- ▶ Efficace anche in presenza di dati non significativi e/o non correlati tra loro
- ▶ Fornisce stime sull'importanza delle variabili
- ▶ All'aumentare di dimensioni della foresta, non aumentano le distorsioni sulla precisione
- ▶ Bilancia l'errore negli insiemi di popolazioni con dati non strettamente correlati

## Contro

- ▶ In alcuni dataset potrebbe causare overfitting



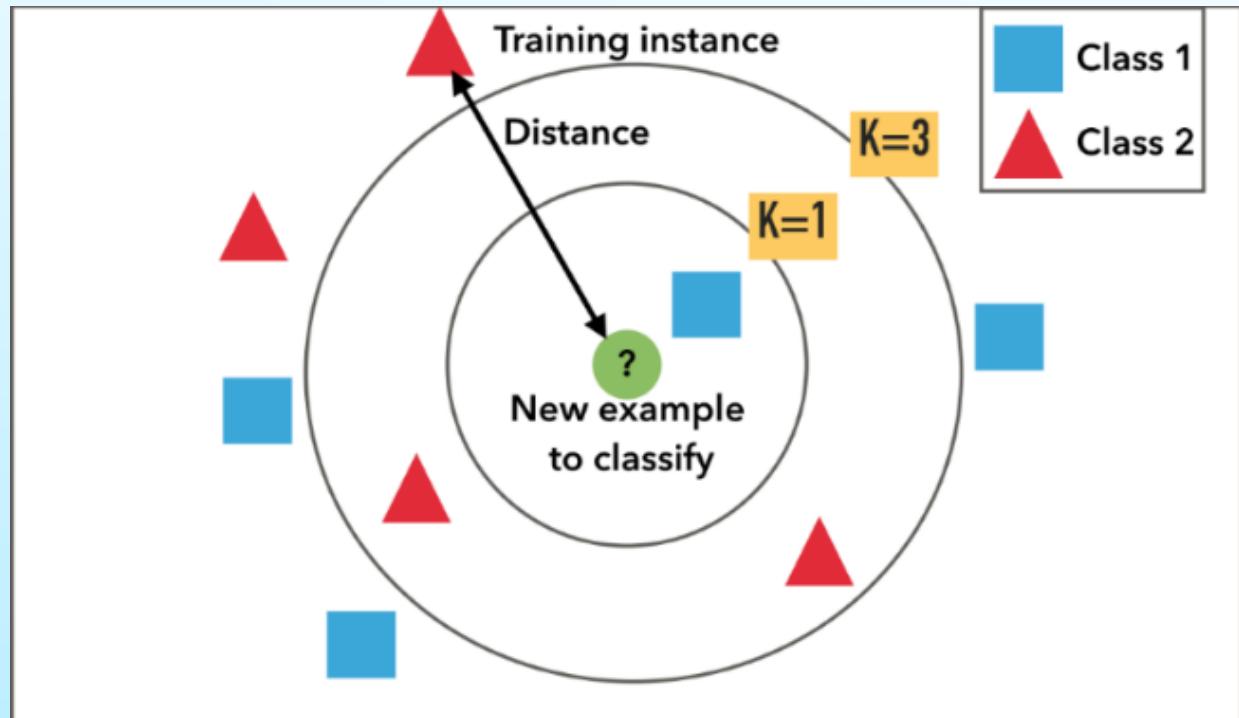
Roberto Falconi  
Federico Guidi

# K-NN

- ▶ Il **k-nearest neighbor** (k-NN) è un algoritmo utilizzato nel riconoscimento di pattern per la classificazione di oggetti basandosi sulle caratteristiche degli oggetti vicini a quello considerato: un oggetto viene classificato in base alla maggioranza dei voti dei suoi  $k$  vicini. È l'algoritmo più semplice fra quelli utilizzati nel Machine Learning.

# K-NN

- ▶ Esempio generale semplificato dell'operato del k-NN; mostra i procedimenti seguiti per effettuare una classificazione



# K-NN

IL CODICE IN PYTHON

```
KNeighborsClassifier(n_neighbors=10,  
weights='uniform', algorithm='auto', leaf_size=30,  
n_jobs=-1)
```

Roberto Falconi  
Federico Guidi

# K-NN

## Pro

- ▶ Non richiede training
- ▶ Efficace anche su dataset vasti
- ▶ Mantiene l'efficienza in presenza di dati non correlati tra di loro
- ▶ Resiste alla presenza di dati non significativi

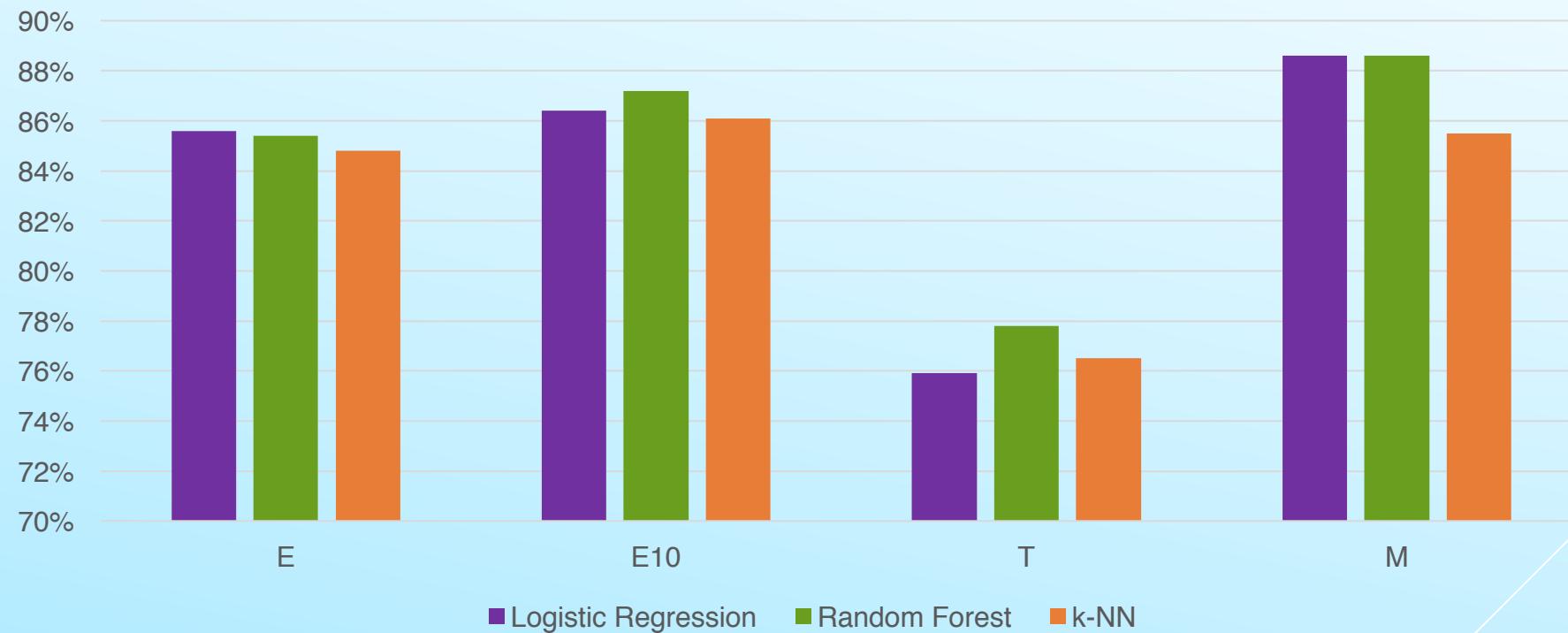
## Contro

- ▶ Richiede l'esplicitazione del k
- ▶ Non è sempre semplice determinare il tipo di distanza da utilizzare
- ▶ Non gestisce in maniera sempre opportuna le variabili categoriche
- ▶ Non è in grado di gestire i “confini poco definiti”
- ▶ I costi di computazione sono relativamente elevati

Roberto Falconi  
Federico Guidi

# APPLICAZIONE DEI CLASSIFICATORI

Accuracy Score / Cross-validation



Roberto Falconi  
Federico Guidi

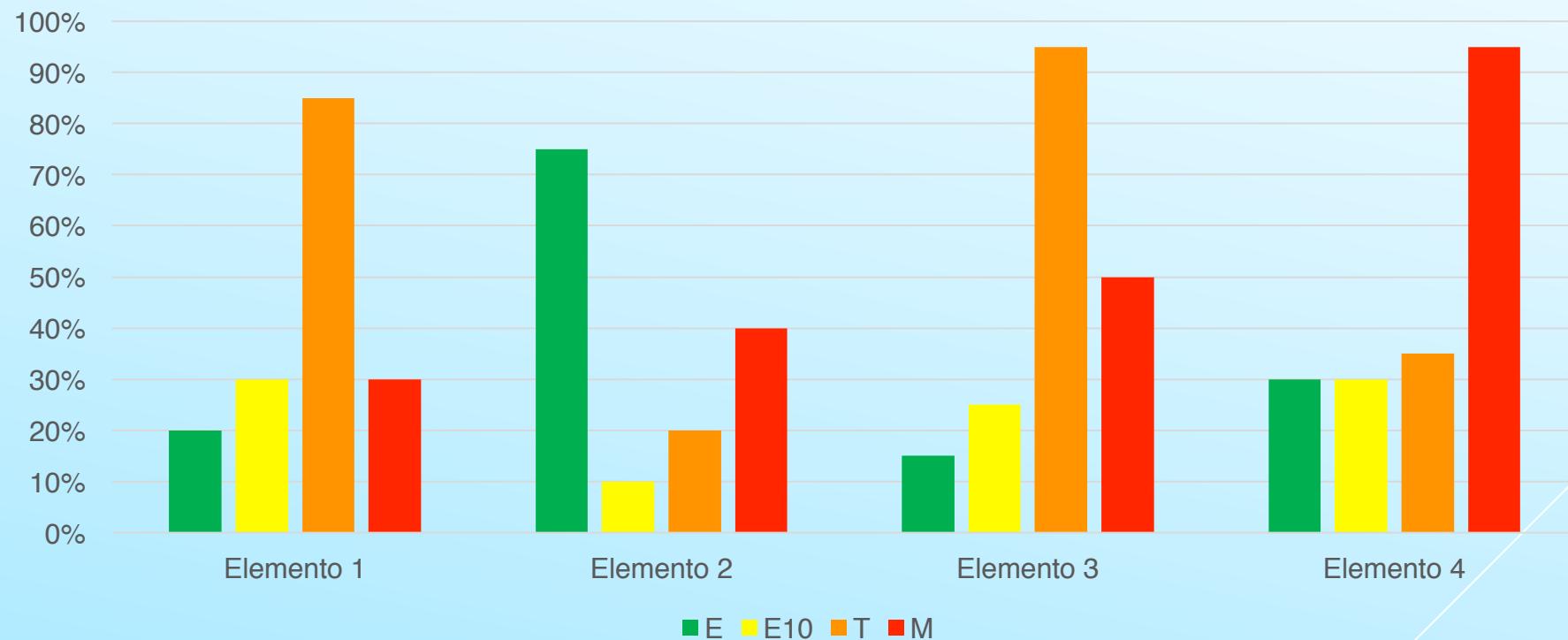
# APPLICAZIONE DEI CLASSIFICATORI

- ▶ Abbiamo visto nella slide precedente che i tre classificatori ottengono mediamente dei buoni accuracy score.
- ▶ Per decretare un classificatore “vincitore”, tuttavia, occorre tener conto della confidenza che hanno tali classificatori con l'output che hanno prodotto.
- ▶ Di seguito vedremo i primi 4 output prodotti dai classificatori sui primi 4 elementi del dataset.

Roberto Falconi  
Federico Guidi

# APPLICAZIONE DEI CLASSIFICATORI

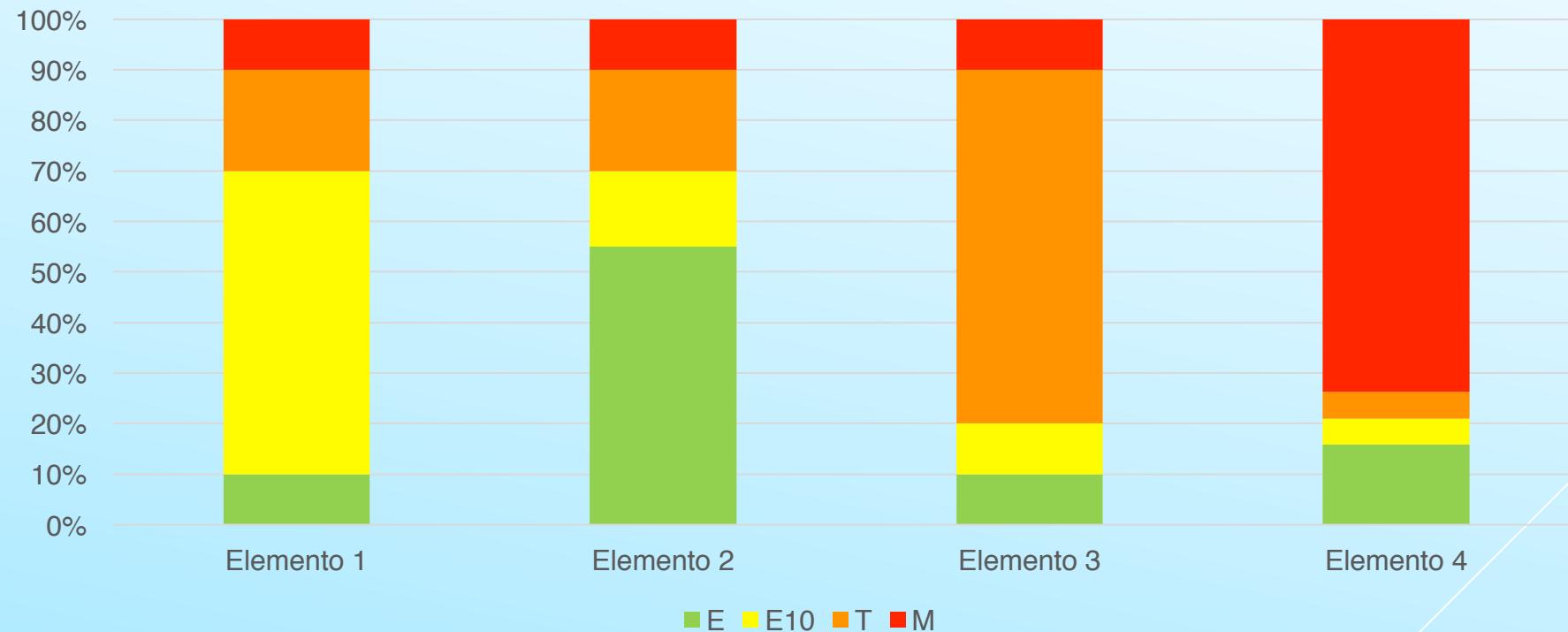
Random Forest - confidenza  
(probabilità che un elemento appartenga ad una classe)



Roberto Falconi  
Federico Guidi

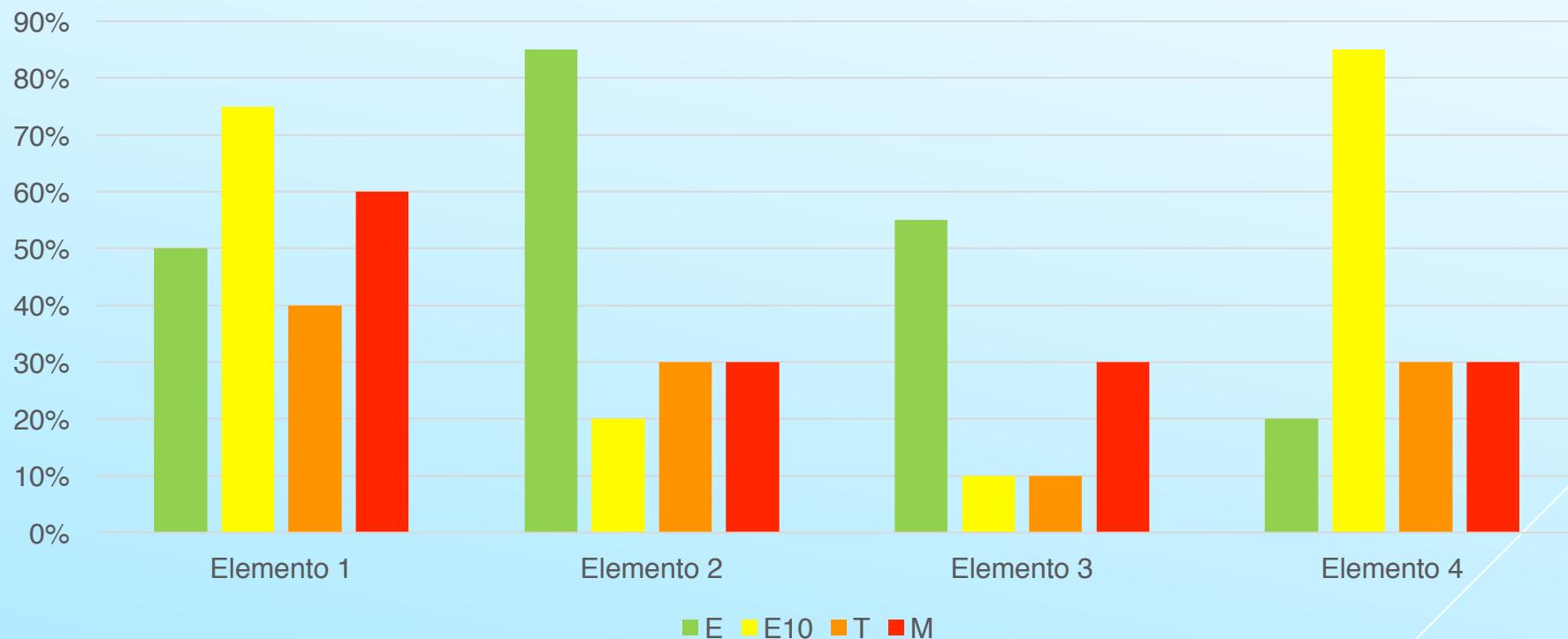
# APPLICAZIONE DEI CLASSIFICATORI

Random Forest - confidenza normalizzata  
(probabilità che un elemento appartenga ad una classe)



# APPLICAZIONE DEI CLASSIFICATORI

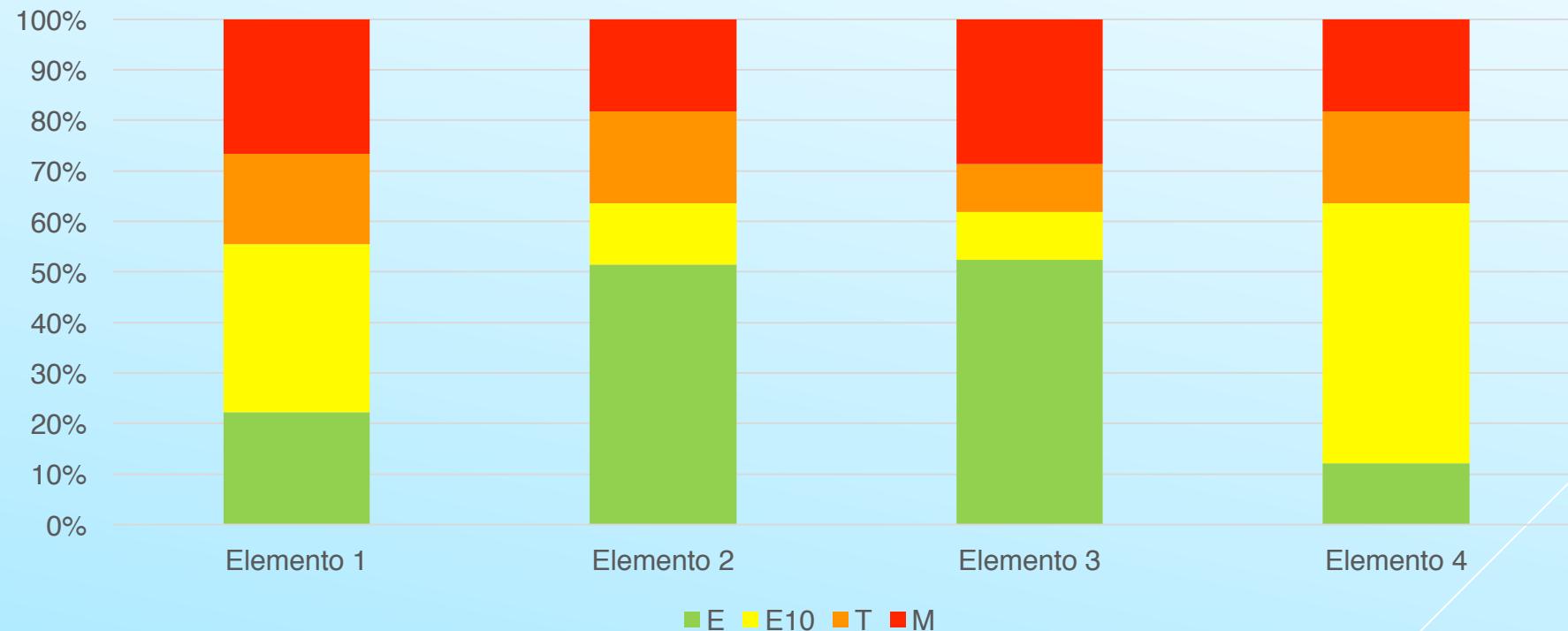
Logistic Regression - confidenza  
(probabilità che un elemento appartenga ad una classe)



Roberto Falconi  
Federico Guidi

# APPLICAZIONE DEI CLASSIFICATORI

Logistic Regression - confidenza normalizzata  
(probabilità che un elemento appartenga ad una classe)

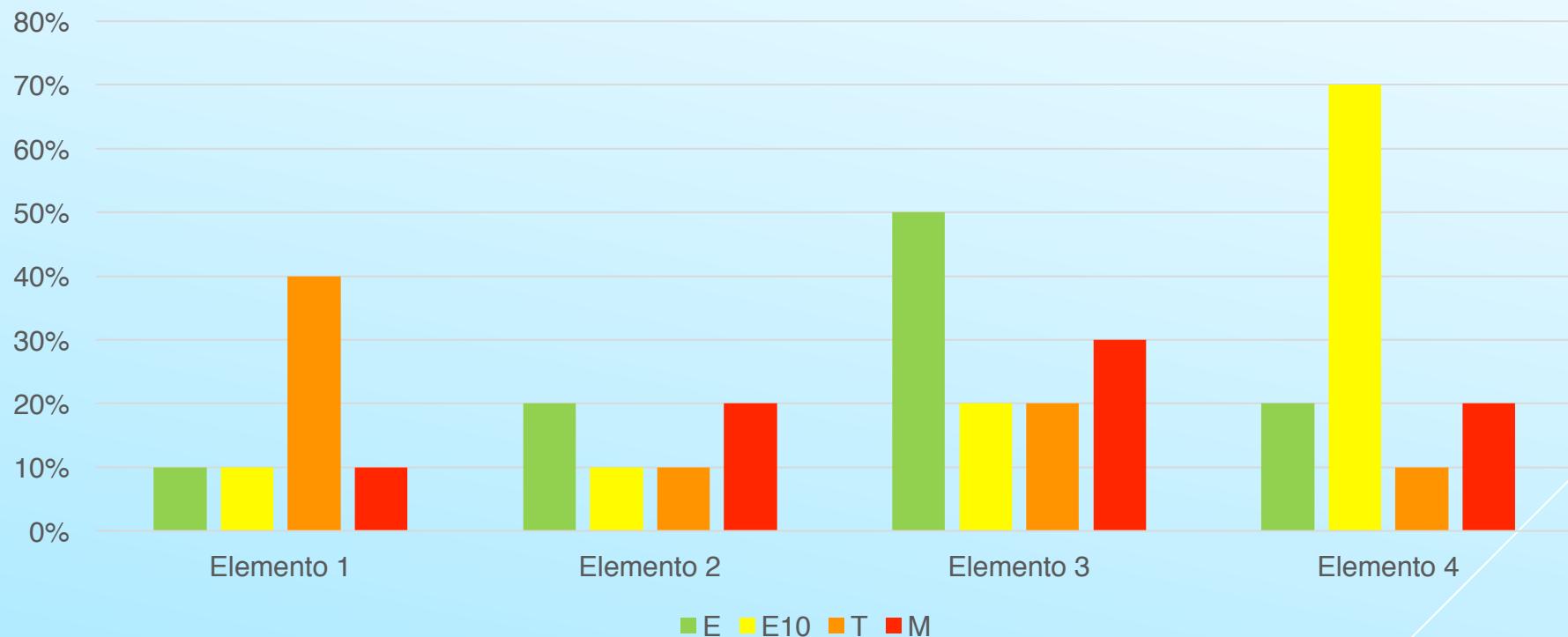


In questa situazione la Logistic Regression ha prodotto una misclassification sull'elemento 1 (nessun classificatore binario supera la majority rule)

Roberto Falconi  
Federico Guidi

# APPLICAZIONE DEI CLASSIFICATORI

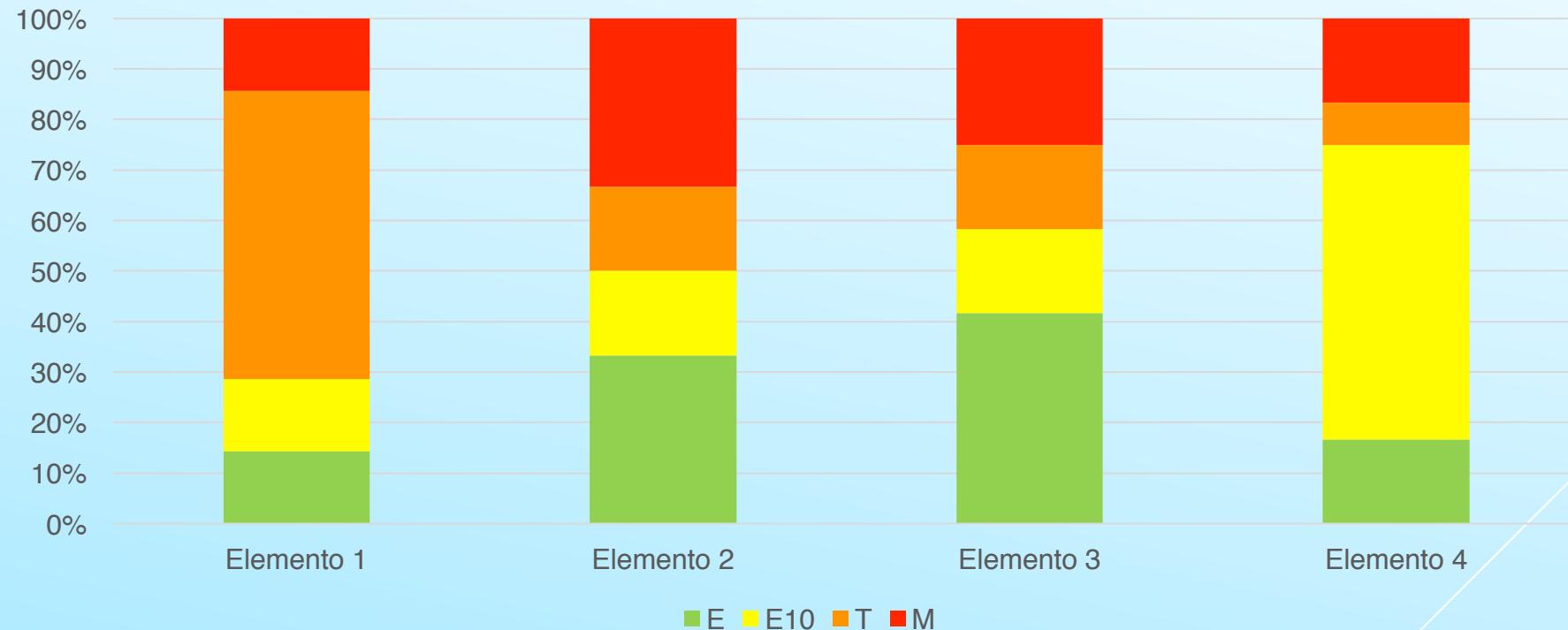
k-NN - confidenza  
(probabilità che un elemento appartenga ad una classe)



Roberto Falconi  
Federico Guidi

# APPLICAZIONE DEI CLASSIFICATORI

k-NN - confidenza normalizzata  
(probabilità che un elemento appartenga ad una classe)

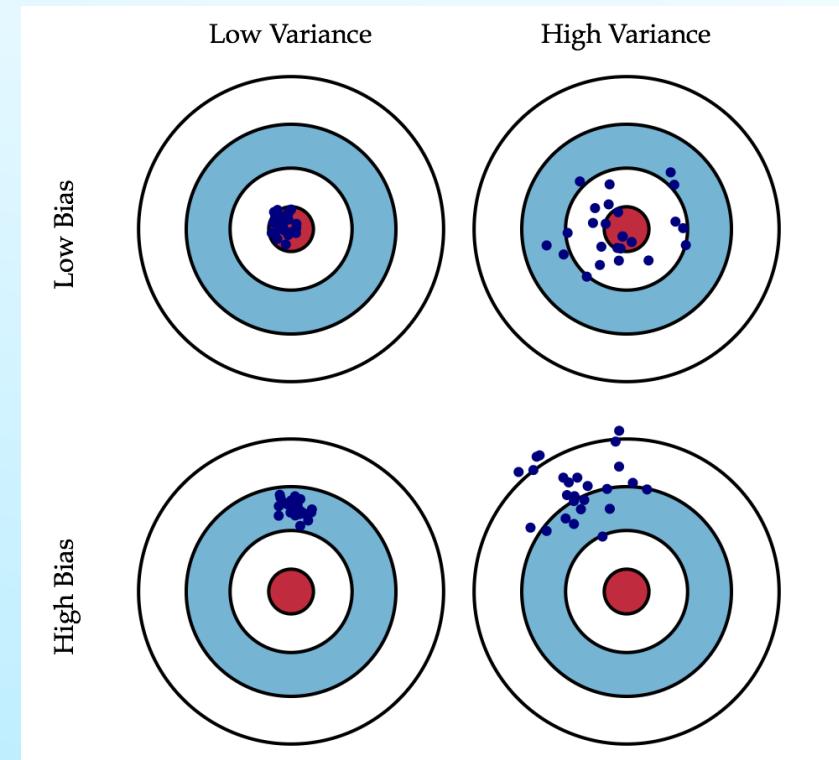


In questa situazione il k-NN ha prodotto una misclassification per il 2° e per il 3° elemento (nessun classificatore binario supera la majority rule)

# BIAS-VARIANCE TRADEOFF

## OSSERVAZIONI

- ▶ Gli errori che possono influire negativamente sui risultati ottenuti sono principalmente di due tipi: dovuto al **bias** e dovuto alla **varianza**.
- ▶ **Errore dovuto al bias:** considerato come la differenza tra la previsione (o media) attesa del nostro modello e il valore corretto che stiamo cercando di prevedere.
- ▶ **Errore dovuto alla varianza:** considerato come la variabilità di una predizione del modello per un determinato punto.



Roberto Falconi  
Federico Guidi

# BIAS-VARIANCE TRADEOFF

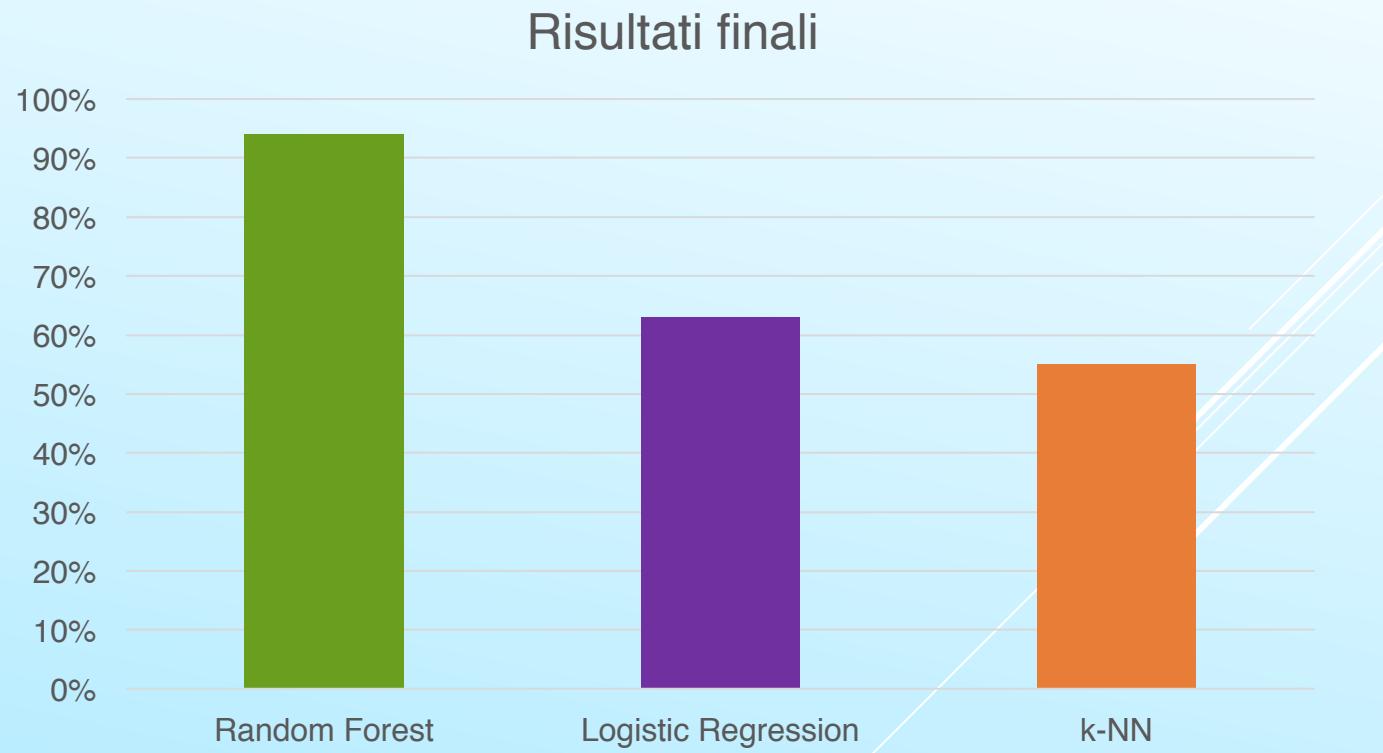
## OSSERVAZIONI

Nel nostro caso abbiamo notato che:

- ▶ **Random Forest:** con un n\_estimator pari a 240 si hanno i risultati migliori; aumentandone la quantità, i risultati peggiorano.
- ▶ **K-NN:** il parametro n\_neighbors, ovvero k, non può essere troppo grande, altrimenti aumenta la varianza e peggiorano di conseguenza i risultati. Il valore ottimale è stato trovato in k = 30.
- ▶ **Logistic Regression:** non ha parametri numerici, pertanto non possono essere effettuate le considerazioni precedenti.

# CONCLUSIONI

- ▶ Con un **accuracy score elevato**, che nel caso di Logistic Regression e k-NN è **diminuito** una volta ponderato sulla base della **confidenza** dei vari classificatori binari, il classificatore multiclass finale che è riuscito a raggiungere nel migliore dei modi il nostro obiettivo è stato senza dubbio **Random Forest**.



Roberto Falconi  
Federico Guidi

# CONCLUSIONI

Name	Rating
Madden NFL	E
Mafia III	M
No Man's Sky	T
NBA 2K17	E

Parte di dataset



**Random Forest: 94.71%**

```
Madden NFL 17 1 1
Mafia III 4 4
No Man's Sky 3 3
NBA 2K17 1 1
Mafia III 4 4
```

Output prodotto

Roberto Falconi  
Federico Guidi