PROGETTO DI MACHINE LEARNING

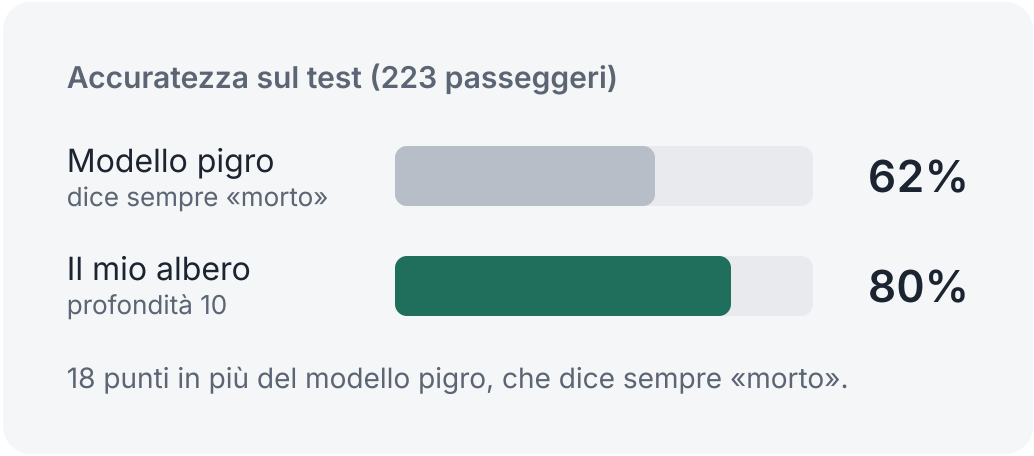
Chi si salva sul Titanic?

di **Nicola Silvestre**

Ho insegnato a un **albero di decisione** a indovinare se un passeggero si è salvato, guardando solo quattro cose: **sesso, età, classe e porto d'imbarco**.

Ho provato 5 profondità e ho scelto la **10** con il validation. All'esame finale l'albero ha indovinato **179 passeggeri su 223**.

Sul test (0,803) è andato quasi come sul validation (0,814).



Notebook completo: https://drive.google.com/file/d/1BFk2OyuDFSi6ylLBy2tqjwmfVjxvNPVg/view?usp=sharing

Il percorso in 11 passi

Le slide seguono questi passi, nello stesso ordine del notebook, più uno sguardo dentro un albero.

PREPARAZIONE DEL DATASET
ho reso la tabella leggibile per il modello

1 **Analisi del DataFrame**
cosa c'è, cosa manca

2 **Scelta dell'indice**
PassengerId

3 **Separazione di X e y**
indizi e soluzione

4 **Parole in numeri**
Sex ed Embarked

SUDDIVISIONE DEI DATI
ho diviso i passeggeri in train, validation e test

5 **Train, validation, test**
501 · 167 · 223 passeggeri

6 **Età mancanti**
mediana del solo train

ADDESTRAMENTO E SCELTA DEL MODELLO
ho allenato 5 alberi e ho tenuto il migliore

7 **5 alberi**
profondità 2, 5, 10, 25, None

8 **Prova sul validation**
un voto per albero

9 **Scelta del migliore**
profondità 10

VALUTAZIONE FINALE E CONCLUSIONI
una sola prova sul test, poi risultati e limiti

10 **Prova finale sul test**
una volta sola

11 **Conclusioni**
risultati e limiti

PREPARAZIONE DEL DATASET · PASSO 1 DI 11

1 Analisi del DataFrame

Prima di addestrare il modello ho guardato bene i dati: cosa c'è, cosa manca, quanti si sono salvati.

La tabella: 891 passeggeri × 6 colonne

Colonna	Cosa contiene	Tipo	Vuoti
PassengerId	numero del passeggero	numero	0
Sex	male / female	parola	0
Age	età, da 0,42 a 80 anni	numero	177
Pclass	classe 1, 2, 3	numero	0
Embarked	porto: S, C, Q	parola	2
Survived	0 = morto, 1 = vivo	soluzione	0

Da sistemare dopo: **177 età** e **2 porti** mancanti. Le parole (Sex, Embarked) andranno tradotte in numeri.

Survived: chi è morto e chi si è salvato



62% 38%

C'è già la soluzione (Survived) → addestramento **supervisionato**.

Vivo o morto è una categoria, non un numero → **classificazione**.

Chi dice sempre «morto» indovina il **62%**: è la soglia che il modello doveva battere.

Le classi non sono troppo sbilanciate (62 / 38): ho potuto usare l'**accuratezza**, confrontandola sempre con il 62%.

2 Scelta dell'indice

Ho usato il numero del passeggero come indice della tabella.

indice	PassengerId	Sex	Age	...
0	1	male	22	...
1	2	female	38	...
2	3	female	26	...

Prima: pandas numera da solo 0, 1, 2... · 6 colonne

→

PassengerId	Sex	Age	Pclass	...
1	male	22	3	...
2	female	38	1	...
3	female	26	3	...

Dopo: **df.set_index("PassengerId")** · 5 colonne

Perché PassengerId

> È **unico**: is_unique → True, ogni passeggero ha il suo numero.

> **Non ha vuoti**: 0 valori mancanti.

> È solo un'**etichetta**: il passeggero 500 non vale più del 499.

Cosa sarebbe successo lasciandolo come colonna

Il modello impara **solo dalle colonne**. Se PassengerId fosse rimasto lì, l'albero lo avrebbe usato come un indizio, con domande senza senso tipo «PassengerId ≤ 400?». Come indice, invece, il modello non lo vede: serve solo a riconoscere ogni passeggero.

3 Separazione di X e y

Ho tagliato la tabella in due: gli indizi da una parte, la soluzione dall'altra.

X gli indizi · 891 righe × 4 colonne

indice	Sex	Age	Pclass	Embarked
1	male	22	3	S
2	female	38	1	C
3	female	26	3	S

y la soluzione · 891

Survived
0
1
1

Stesse righe, stesso indice: il passeggero 1 sta in X e la sua soluzione sta in y, sempre alla riga 1.

Perché separarli

Il modello deve imparare a passare **dagli indizi alla soluzione**. Se Survived fosse rimasta dentro X, il modello l'avrebbe copiata invece di imparare a indovinarla.

Il momento giusto

Le modifiche successive (parole in numeri, età mancanti) riguardano **solo gli indizi**. Separando prima, ho lavorato solo su X senza rischiare di toccare la soluzione.

4 Traduzione delle parole in numeri

L'albero sa fare solo confronti tra numeri, tipo «Age ≤ 30?». Le parole andavano tradotte.

Sex LabelEncoder · scikit-learn

female	→	0
male	→	1

fit mette le parole in ordine alfabetico, **transform** le sostituisce con 0 e 1.

Va bene perché i valori sono solo 2: non c'è niente in mezzo, quindi non si inventa nessun ordine.

Embarked get_dummies · pandas

porto	Embarked_C	Embarked_Q	Embarked_S
C	1	0	0
Q	0	1	0
S	0	0	1
vuoto (2 passeggeri)	0	0	0

I 3 porti non hanno un ordine: con 0, 1, 2 il modello penserebbe che S valga più di C. Così ogni porto diventa una domanda sì/no, e i 2 vuoti restano «non so».

Ecco X dopo la traduzione: 6 colonne, tutte numeri (le età mancanti le ho riempite dopo la divisione)

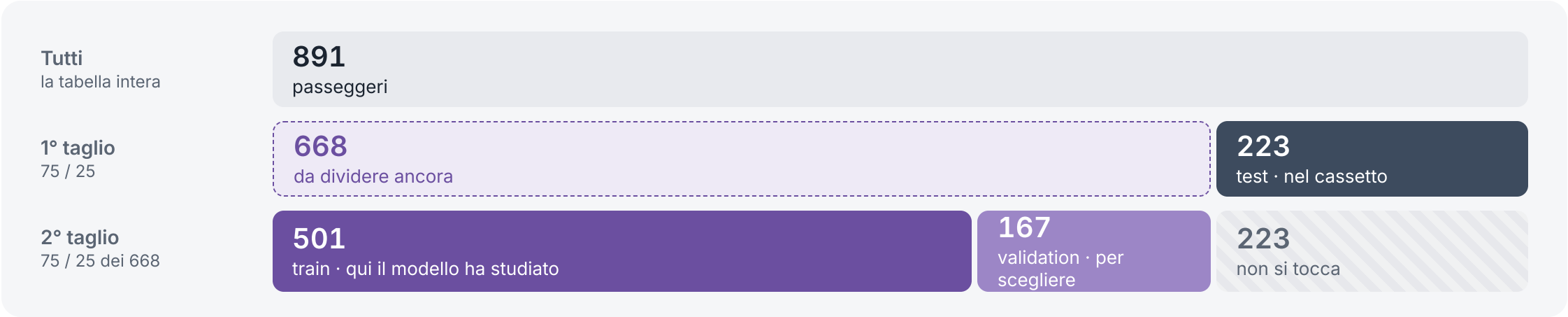
Pclass e Age erano già numeri

PassengerId	Sex	Age	Pclass	Embarked_C	Embarked_Q	Embarked_S
1	1	22	3	0	0	1
2	0	38	1	1	0	0

Passeggero 1: uomo (1), 22 anni, 3ª classe, salito a Southampton. Passeggero 2: donna (0), 38 anni, 1ª classe, salita a Cherbourg.

5 Divisione in train, validation e test

Ho diviso i passeggeri con due tagli 75/25, sempre con seed 0.



Perché due tagli

Dovevo scegliere tra 5 alberi. Per sceglierli mi serviva un gruppo diverso **sia da quello su cui studiavano sia dal test**, altrimenti avrei sbirciato l'esame e il voto finale sarebbe stato gonfiato.

Il seed 0

random_state=0 fissa la mescolata: chiunque rifaccia il notebook trova gli stessi gruppi e gli stessi numeri.

6 Riempimento delle età mancanti

A 177 passeggeri mancava l'età. Ho messo la mediana, calcolata solo sul train.

mediana di Age sui 501 del train **29 anni**

→ la stessa età è andata nei vuoti di tutti e tre i gruppi

Train · 501	Validation · 167	Test · 223
102 vuoti	**31** vuoti	**44** vuoti
→ 29 · poi 0 vuoti	→ 29 · poi 0 vuoti	→ 29 · poi 0 vuoti

Mediana, non media

È l'età che sta **in mezzo**: metà dei passeggeri è più giovane, metà più vecchia. A differenza della media, non si fa spostare da pochi passeggeri molto anziani.

Solo sul train: il *data leakage*

29
solo train: corretto

~~**28**~~
tutti gli 891: userebbe le età del test

Se avessi usato anche le età del test, il modello avrebbe avuto un'informazione sui passeggeri su cui poi l'ho provato.

7 Costruzione di 5 alberi

Ho allenato cinque alberi sugli stessi 501 passeggeri: cambia solo quante domande possono fare di fila.

albero	max_depth	domande di fila	gruppi finali
albero_2	2	2	4
albero_5	5	5	23
albero_10	10	10	92
albero_25	25	14	107
albero_max	None	14	107

Come lavora un albero

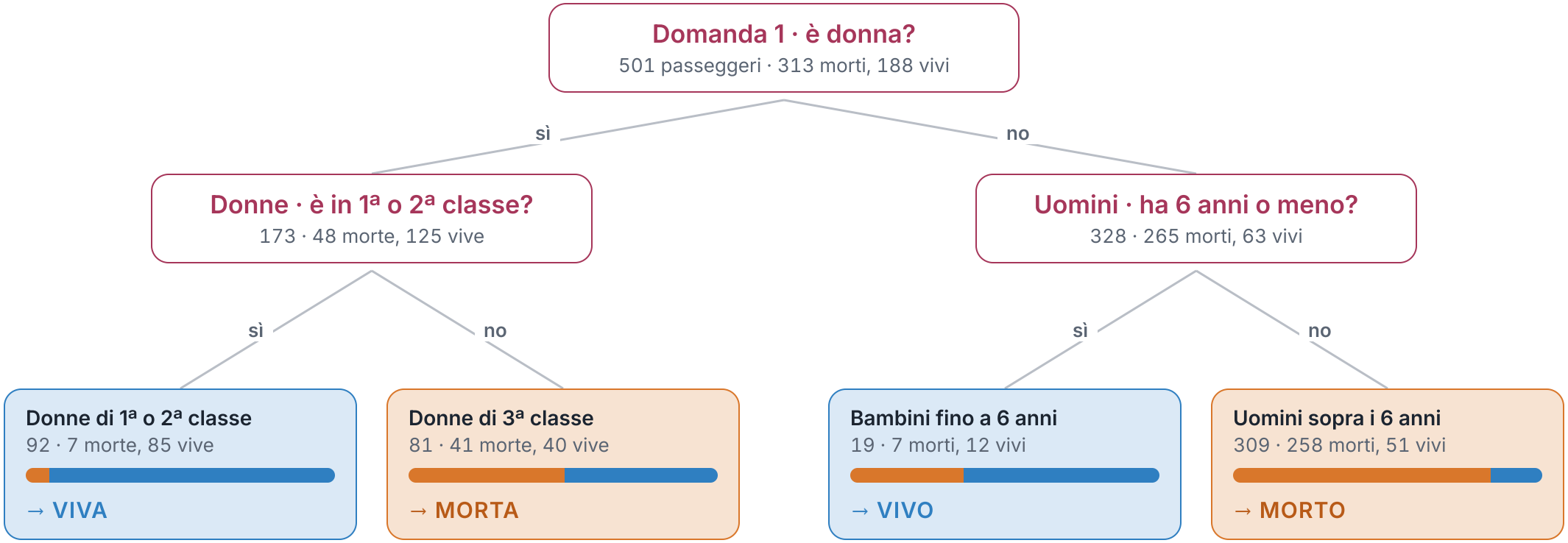
Ogni domanda è un sì/no che **taglia un gruppo in due**, per esempio «è donna?» o «ha 6 anni o meno?». Alla fine ogni gruppo risponde con la maggioranza: vivo o morto. Con più domande di fila, i gruppi diventano più piccoli.

25 e None sono lo stesso albero

Tutti e due si fermano a **14 domande di fila**: dopo non c'è più niente da dividere, perché in ogni gruppo i passeggeri hanno la stessa soluzione o gli stessi identici indizi.

Un albero di esempio: albero_2

Due domande di fila, quattro gruppi finali. Ogni gruppo risponde con la maggioranza. Numeri veri, sui 501 del train.

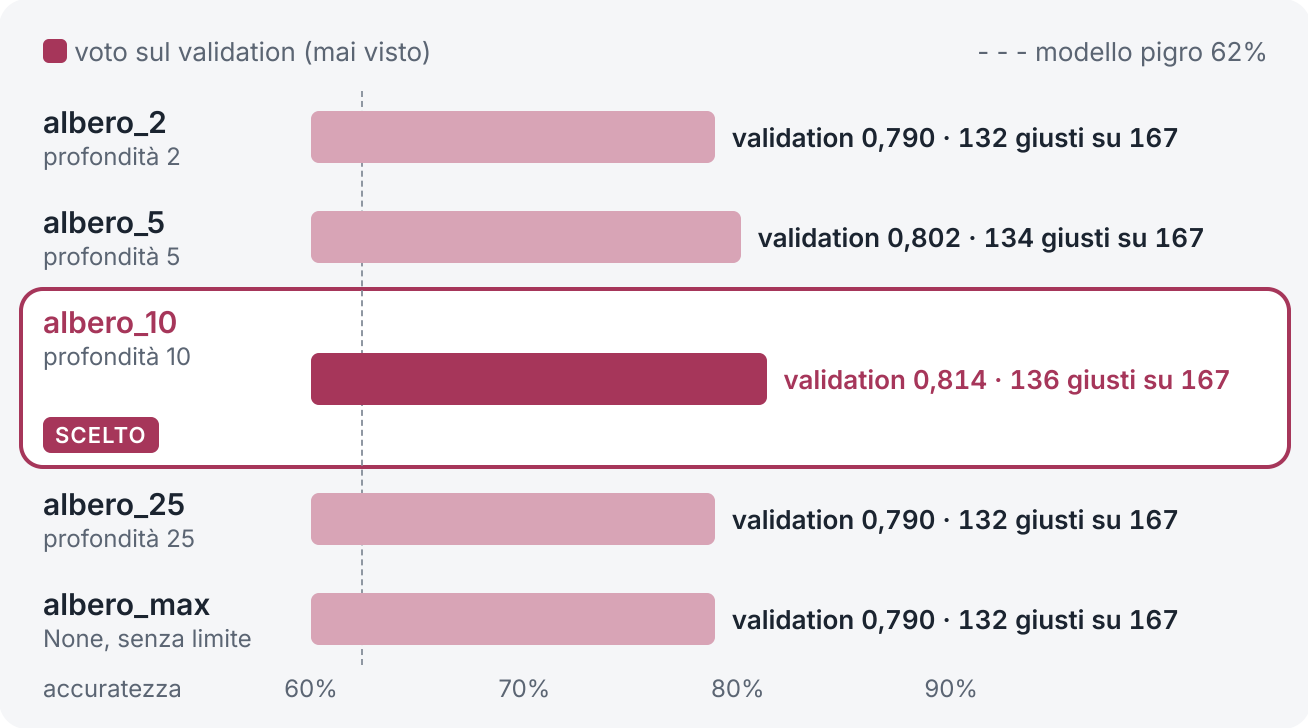


Le donne di 3ª classe sono quasi metà e metà (41 contro 40): la maggioranza «morta» vince per un solo passeggero di differenza.

albero_10 parte dalle **stesse due domande**, poi continua a dividere.

8–9 Prova sul validation e scelta

Ogni albero ha risposto sui 167 passeggeri del validation, mai visti. Ho scelto quello con il voto più alto.



Come si legge

> **Il voto più alto sul validation** è di albero_10: 136 giusti su 167, cioè 0,814. Solo 2 passeggeri più di albero_5.

> Tutti battono il **62%** del modello pigro.

> Il voto **sale fino a 10 e poi scende**: con 2 domande di fila l'albero è troppo semplice (**underfitting**), con 25 o None si adatta troppo ai passeggeri del train e con quelli nuovi va peggio (**overfitting**).

Le barre partono dal 60%, non da 0, per far vedere le differenze di pochi punti.

10 Prova finale sul test

Alla fine ho provato l'albero scelto sui 223 passeggeri del test, una volta sola.

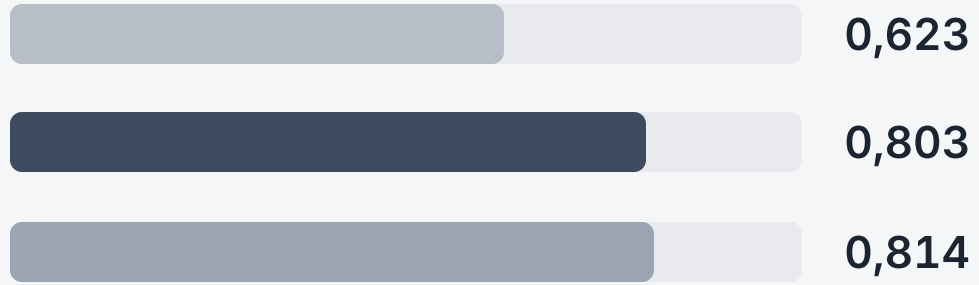
Accuratezza sul test

0,803

179 giusti su **223**

cioè 8 passeggeri su 10

Confronto



> **18 punti** in più del modello pigro: l'albero ha trovato regole utili.

> Test e validation quasi uguali: il voto del validation era **una stima affidabile**.

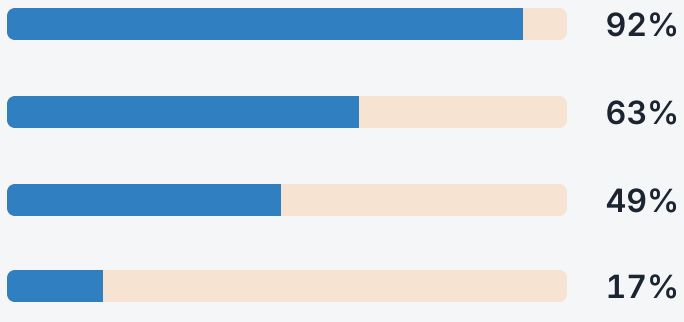
Dopo questo voto non ho più cambiato albero. Se l'avessi fatto guardando il test, il test sarebbe diventato un secondo validation e 0,803 non sarebbe più stato affidabile.

11 Conclusioni

La risposta alla domanda iniziale e il confronto tra le 5 profondità.

Chi si salva sul Titanic?

Le donne di 1ª e 2ª classe quasi tutte, i bambini in buona parte. Le donne di 3ª classe metà e metà. Gli uomini adulti quasi mai. Sono i 4 gruppi dell'albero di esempio, sui 501 del train:



Le 5 profondità a confronto

Voto sul validation (167 passeggeri):

2	5	10	25	None
0,790	0,802	0,814	0,790	0,790
troppo semplice		**scelta · test 0,803**	troppo complicato	uguale alla 25

Tra le 5 la 10 ha avuto il voto più alto e sul test l'ha confermato: **0,803, 8 passeggeri su 10**.

Notebook completo: https://drive.google.com/file/d/1BFk2OyuDFSi6ylLBy2tqjwmfVjxvNPVg/view?usp=sharing