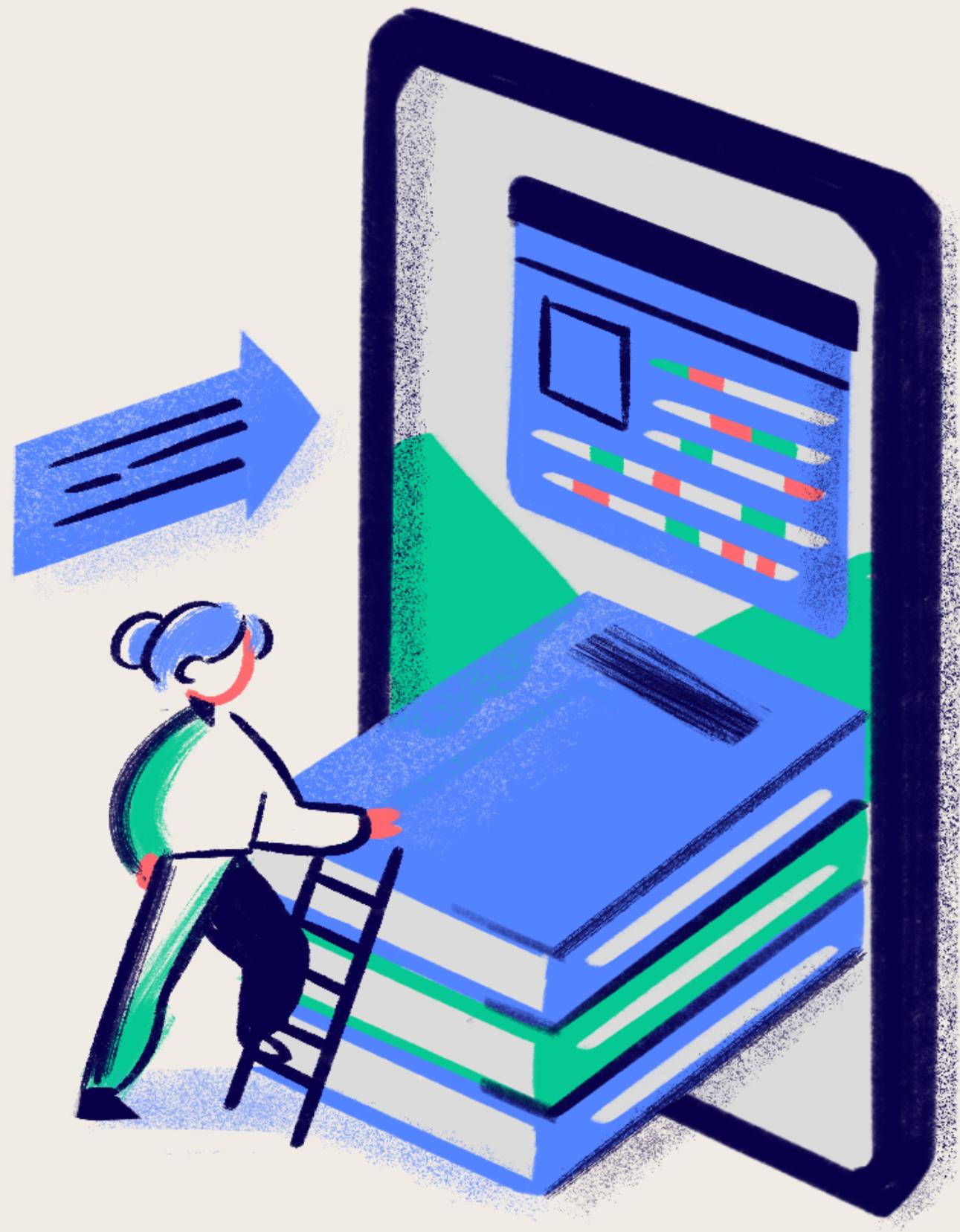


PRESENTATO DA ALESSANDRO SMAJLOVIC

ANALISI DELLE SPECIFICHE DEGLI SMARTPHONE

UN'ANALISI DETTAGLIATA
DELLE CARATTERISTICHE
DEI MODELLI DI
SMARTPHONE



INTRODUZIONE

Obiettivo della Presentazione:

- Questa presentazione mira a fornire un'analisi dettagliata delle specifiche degli smartphone, esplorando diverse caratteristiche come capacità della RAM, velocità del processore, capacità della batteria e disponibilità della ricarica rapida.

Importanza dell'Analisi delle Specifiche degli Smartphone:

- Le specifiche hardware sono fondamentali nella scelta di uno smartphone, influenzando direttamente le prestazioni, l'efficienza energetica e l'esperienza utente.
- Comprendere queste specifiche aiuta i consumatori a fare scelte informate e i produttori a identificare le tendenze di mercato e le preferenze degli utenti.



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Carica i dati dal CSV in un DataFrame
df = pd.read_csv('C:/Users/ale_s/smартфон_analyze/smartphones_cleaned_v6.csv')

# Visualizza le colonne del DataFrame
print("Colonne del DataFrame:", df.columns)
```

```
C:\Users\ale s\AppData\Local\Temp\ipykernel_16856\1021199404.py:1: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466
```

```
import pandas as pd
Colonne del DataFrame: Index(['brand_name', 'model', 'price', 'rating', 'has_5g', 'has_nfc',
   'has_ir_blaster', 'processor_brand', 'num_cores', 'processor_speed',
   'battery_capacity', 'fast_charging_available', 'fast_charging',
   'ram_capacity', 'internal_memory', 'screen_size', 'refresh_rate',
   'num_rear_cameras', 'num_front_cameras', 'os', 'primary_camera_rear',
   'primary_camera_front', 'extended_memory_available', 'extended_upto',
   'resolution_width', 'resolution_height'],
  dtype='object')
```

Panoramica delle colonne presenti nel dataset

```
df.head(10)
```

Python

	brand_name	model	price	rating	has_5g	has_nfc	has_ir_blower	processor_brand	num_cores	processor_speed	...	refresh_rate	num_rear_cameras	num_front_cameras	os	primary_car
0	oneplus	OnePlus 11 5G	54999	89.0	True	True	False	snapdragon	8.0	3.20	...	120	3	1.0	android	
1	oneplus	OnePlus Nord CE 2 Lite 5G	19989	81.0	True	False	False	snapdragon	8.0	2.20	...	120	3	1.0	android	
2	samsung	Samsung Galaxy A14 5G	16499	75.0	True	False	False	exynos	8.0	2.40	...	90	3	1.0	android	
3	motorola	Motorola Moto G62 5G	14999	81.0	True	False	False	snapdragon	8.0	2.20	...	120	3	1.0	android	
4	realme	Realme 10 Pro Plus	24999	82.0	True	False	False	dimensity	8.0	2.60	...	120	3	1.0	android	
5	samsung	Samsung Galaxy F23 5G (6GB RAM + 128GB)	16999	80.0	True	True	False	snapdragon	8.0	2.20	...	120	3	1.0	android	
6	apple	Apple iPhone 14	65999	81.0	True	True	False	bionic	6.0	3.22	...	60	2	1.0	ios	

Cell 1 of 28

Cell 1 of 28

df.info

```
<bound method DataFrame.info of      brand_name          model  price  rating  has_5g  \
0      oneplus        OnePlus 11 5G  54999    89.0   True
1      oneplus  OnePlus Nord CE 2 Lite 5G  19989    81.0   True
2      samsung  Samsung Galaxy A14 5G  16499    75.0   True
3      motorola  Motorola Moto G62 5G  14999    81.0   True
4      realme  Realme 10 Pro Plus  24999    82.0   True
...
975     motorola  Motorola Moto Edge S30 Pro  34990    83.0   True
976      honor      Honor X8 5G  14990    75.0   True
977      poco  POCO X4 GT 5G (8GB RAM + 256GB)  28990    85.0   True
978     motorola  Motorola Moto G91 5G  19990    80.0   True
979      samsung  Samsung Galaxy M52s 5G  24990    74.0   True
```

```
has_nfc  has_ir_blower processor_brand  num_cores  processor_speed  ...  \
0       True            False  snapdragon      8.0        3.20  ...
1      False            False  snapdragon      8.0        2.20  ...
2      False            False      exynos      8.0        2.40  ...
3      False            False  snapdragon      8.0        2.20  ...
4      False            False  dimensity      8.0        2.60  ...
...
975     False            False  snapdragon      8.0        3.00  ...
976     False            False  snapdragon      8.0        2.20  ...
977     True             True  dimensity      8.0        2.85  ...
978     True             False  snapdragon      8.0        2.20  ...
979     False            False        NaN      8.0        NaN  ...
...
977      NaN           1080        2460
978  1024.0           1080        2400
979  1024.0           1080        2400
```

```
# Conta i diversi tipi dispositivi per brand
conteggio_brand = df['brand_name'].value_counts()

# Stampa il conteggio dei dispositivi per brand
print("Conteggio generale dei brand per numero di dispositivi:")
print(conteggio_brand)
```

Conteggio generale dei brand per numero di dispositivi:

```
brand_name
xiaomi      134
samsung     132
vivo        111
realme       97
oppo         88
motorola     52
apple         46
oneplus      42
poco          41
tecno         33
iqoo          32
infinix       29
huawei        16
google        14
nokia         13
honor         13
itel           10
sony            9
asus            7
nubia           6
nothing          5
lava             4
jio              4
...
tesla            1
cola              1
blackview        1
Name: count, dtype: int64
```

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output [settings](#)...

Conteggio dei brand per numero di dispositivi

```
# Conta il numero di brand unici nella colonna 'brand_name'  
numero_di_brand = df['brand_name'].nunique()  
  
# Stampa il numero di brand unici  
print("Numero di brand unici:", numero_di_brand)  
[13] Python  
... Numero di brand unici: 46
```

Conteggio dei brand unici

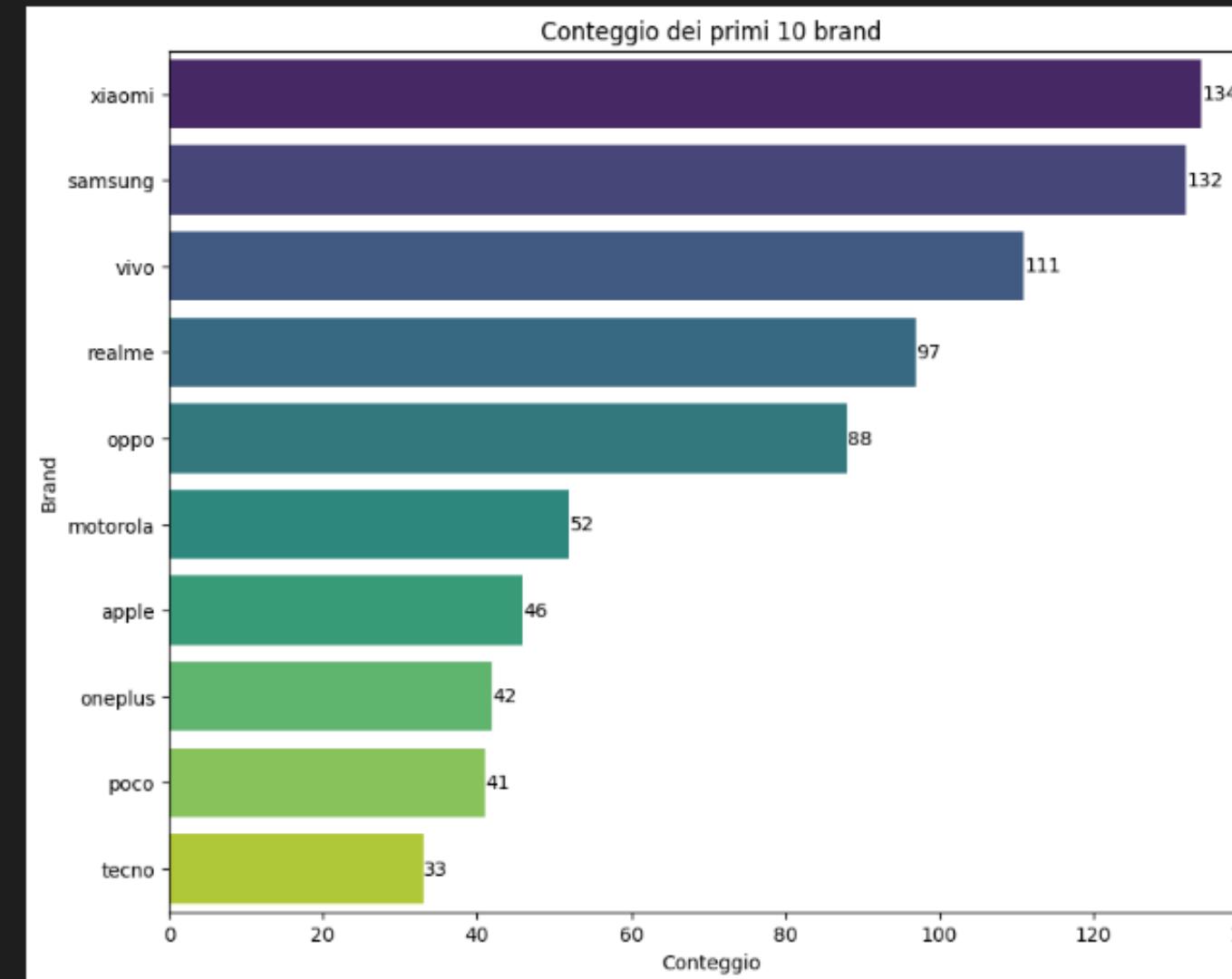
```
conteggio_brand = df['brand_name'].value_counts().head(10)

# Crea un grafico a barre orizzontali per i primi 10 brand
plt.figure(figsize=(10, 8))
sns.barplot(y=conteggio_brand.index, x=conteggio_brand.values, palette='viridis')
plt.title('Conteggio dei primi 10 brand')
plt.xlabel('Conteggio')
plt.ylabel('Brand')
# Aggiungi il numero alla fine di ogni barra
for index, value in enumerate(conteggio_brand.values):
    plt.text(value, index, str(value), va='center')
plt.show()
```

C:\Users\ale s\AppData\Local\Temp\ipykernel_13468\2172213823.py:6: FutureWarning:

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'y' variable to 'hue' and set 'legend=False' for the same effect.

```
sns.barplot(y=conteggio_brand.index, x=conteggio_brand.values, palette='viridis')
```



Conteggio dei primi 10 brand

```

# Ordina i dati per prezzo e seleziona i primi 20 modelli
modelli_top_20 = df.sort_values(by='price', ascending=False).head(20)

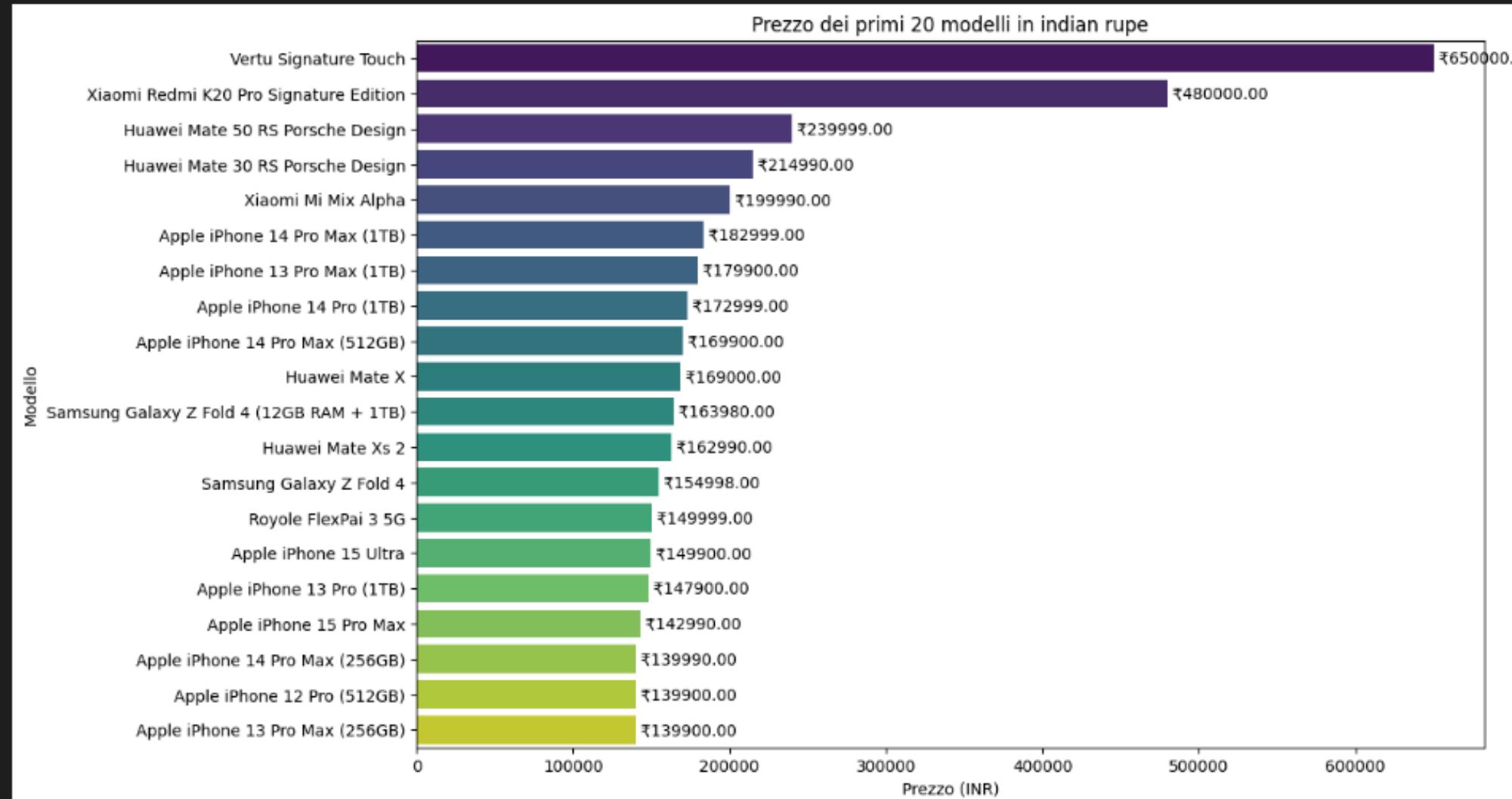
# Crea un grafico a barre orizzontali per i primi 20 modelli in base al prezzo
plt.figure(figsize=(12, 8))
sns.barplot(x='price', y='model', data=modelli_top_20, palette='viridis')
plt.title('Prezzo dei primi 20 modelli in indian rupe')
plt.xlabel('Prezzo (INR)')
plt.ylabel('Modello')
# Aggiungi il prezzo alla fine di ogni barra
for index, value in enumerate(modelli_top_20['price']):
    plt.text(value, index, f' ₹{value:.2f}', va='center')
plt.show()

```

C:\Users\ale s\AppData\Local\Temp\ipykernel_5368\3857378594.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x='price', y='model', data=modelli_top_20, palette='viridis')
```



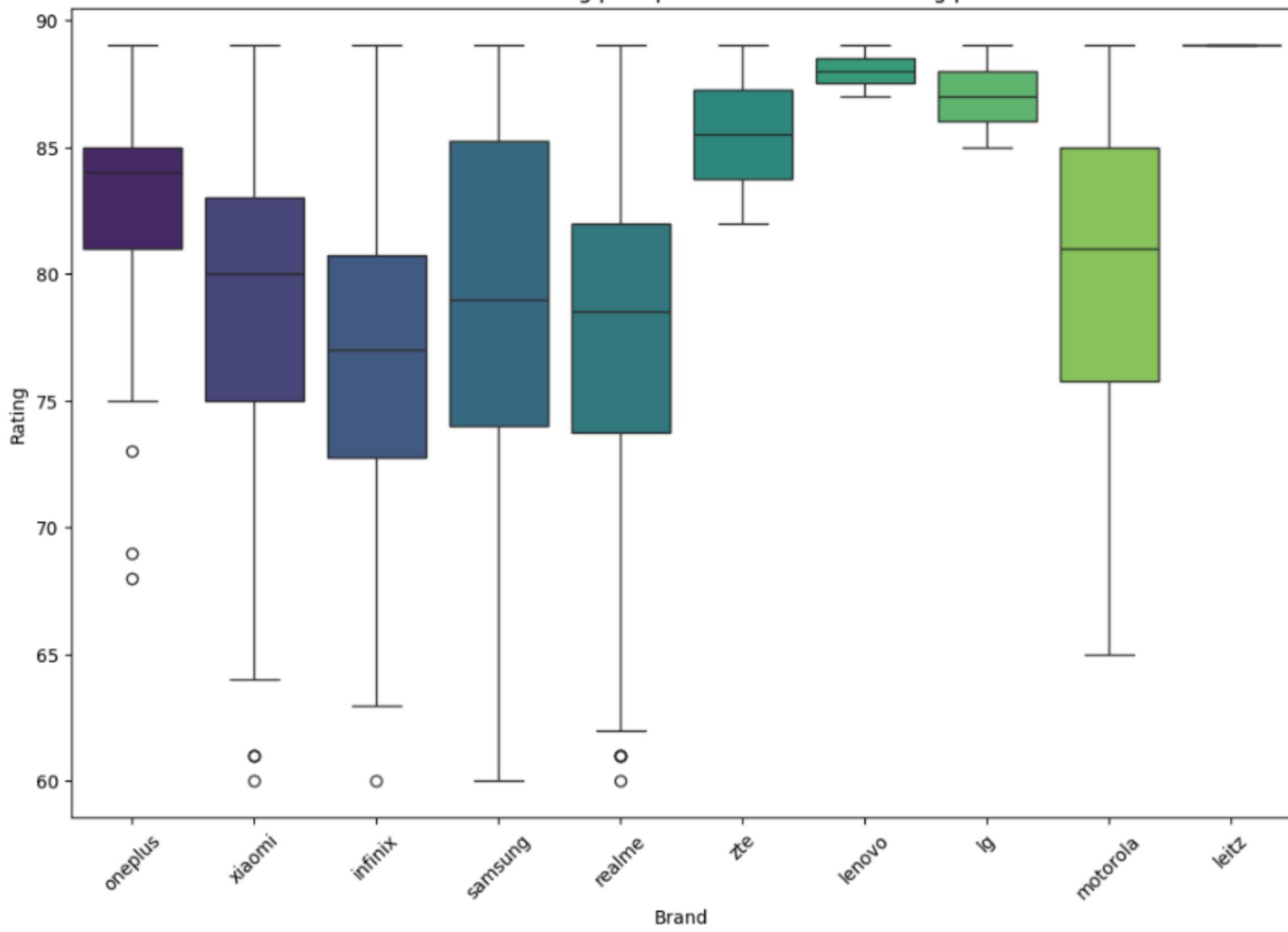
Prezzo dei primi 10 modelli Rupee Indiani

```
sns.catplot(data=df, x="price", y="brand_name", jitter=False)
plt.title('Distribuzione del prezzo per brand')
plt.xlabel('Prezzo')
plt.ylabel('Brand')
plt.show()
```



Distribuzioni del prezzo per brand

Distribuzione dei rating per i primi 10 brand con i rating più alti



Distribuzione dei rating più alti

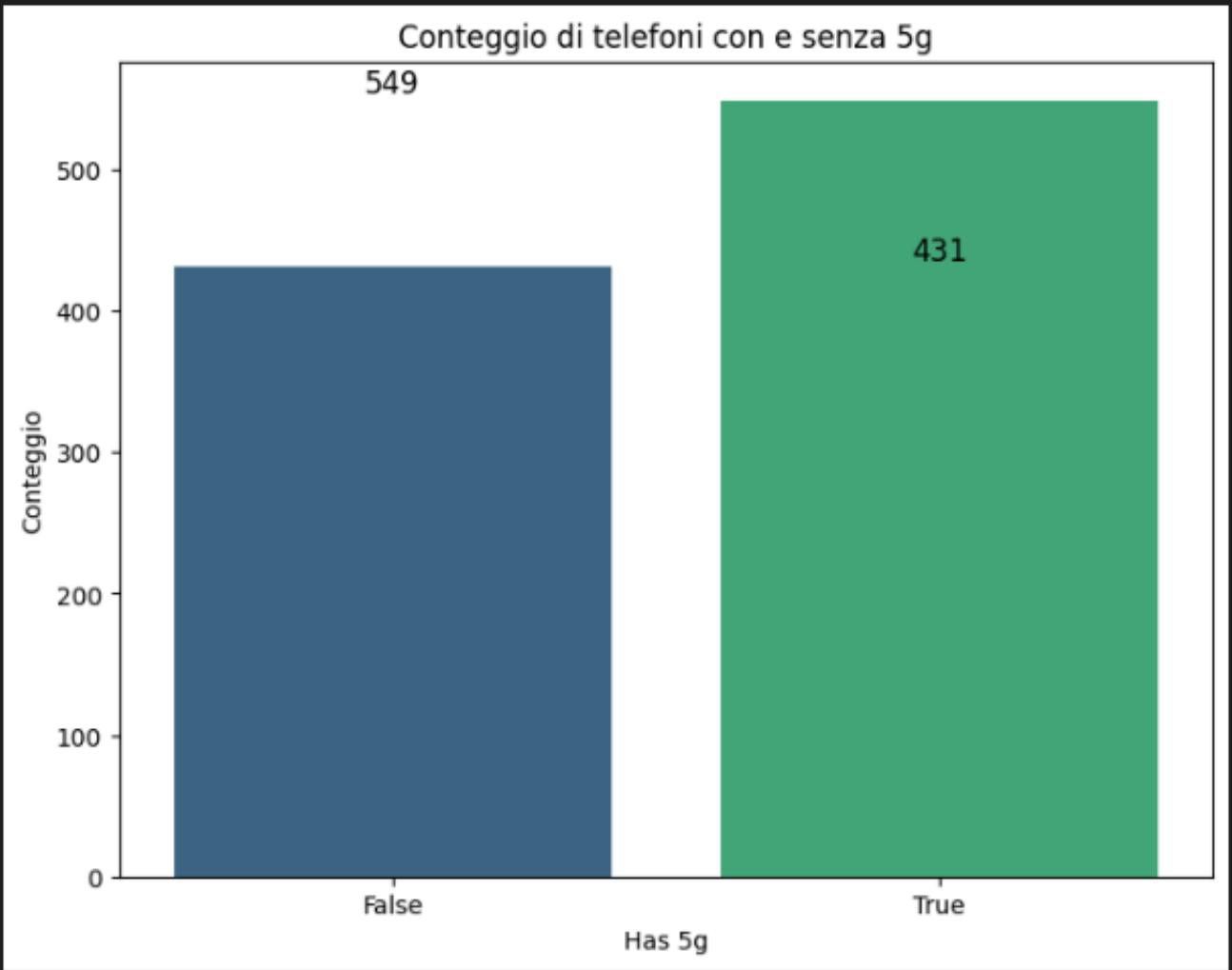
```
# Calcola i conteggi di telefoni con e senza 5G
conteggio_5g = df['has_5g'].value_counts()

# Crea un grafico a barre per i conteggi di telefoni con e senza 5G
plt.figure(figsize=(8, 6))
sns.barplot(x=conteggio_5g.index, y=conteggio_5g.values, palette='viridis')
plt.title('Conteggio di telefoni con e senza 5g')
plt.xlabel('Has 5g')
plt.ylabel('Conteggio')
plt.xticks(rotation=0) # Mantieni le etichette dell'asse x dritte
# Aggiungi i numeri alla fine di ogni barra
for index, value in enumerate(conteggio_5g.values):
    plt.text(index, value, str(value), ha='center', va='bottom', fontsize=12)
plt.show()
```

C:\Users\ale s\AppData\Local\Temp\ipykernel_14356\3945825212.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=conteggio_5g.index, y=conteggio_5g.values, palette='viridis')
```



Conteggio dei dispositivi con e senza 5g

```
# Conta i modelli con e senza NFC
conteggio_nfc = df['has_nfc'].value_counts().sort_index()

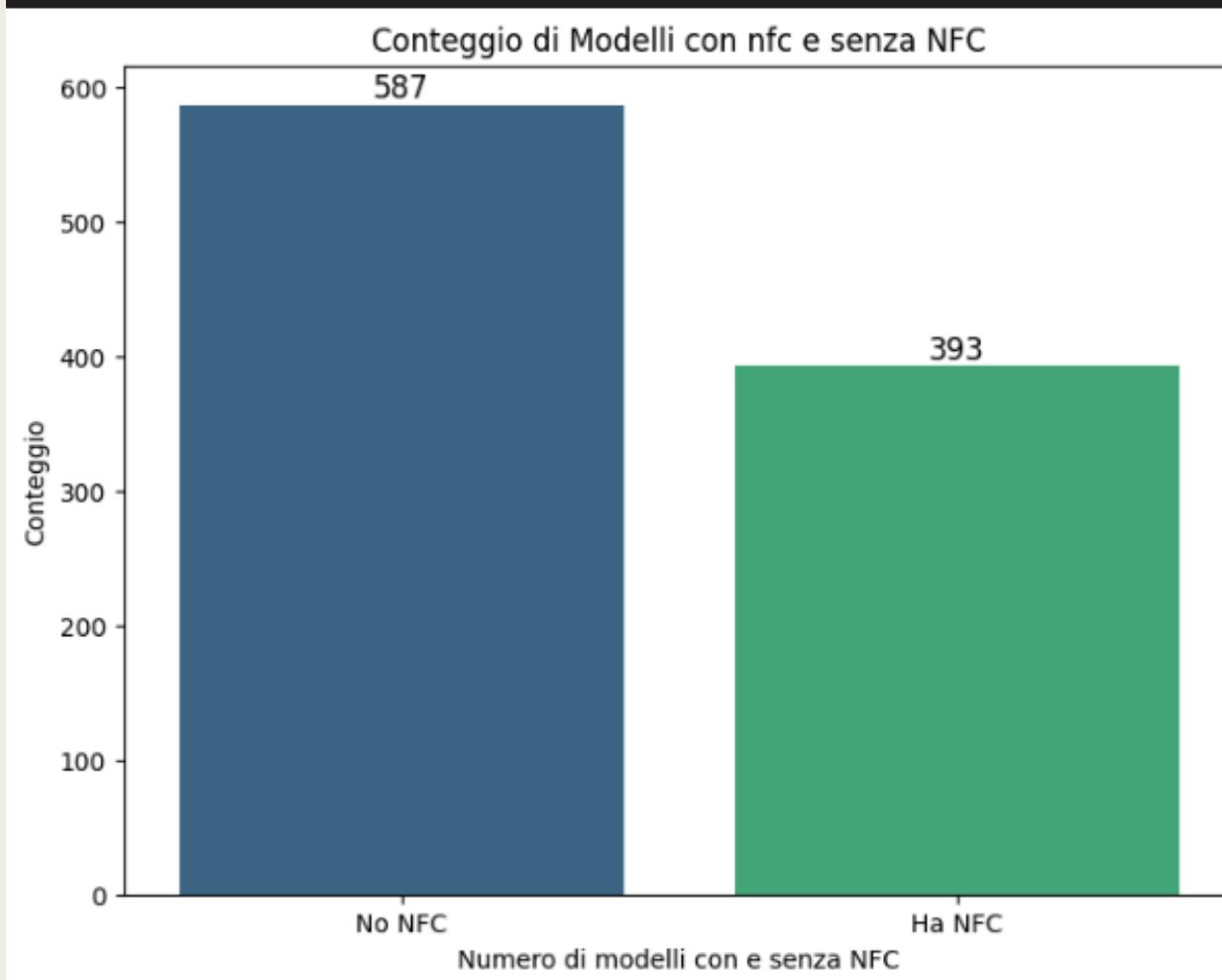
# Crea un grafico a barre per i conteggi di modelli con e senza NFC
plt.figure(figsize=(8, 6))
sns.barplot(x=conteggio_nfc.index, y=conteggio_nfc.values, palette='viridis')
plt.title('Conteggio di Modelli con nfc e senza NFC')
plt.xlabel('Numero di modelli con e senza NFC')
plt.ylabel('Conteggio')
plt.xticks(ticks =[0,1], labels=['No NFC', 'Ha NFC'])
# Aggiungere i numeri alla fine di ogni barra
for index, value in enumerate(conteggio_nfc.values):
    plt.text(index, value, str(value), ha='center', va='bottom', fontsize=12)

plt.show()
```

:\\Users\\ale s\\AppData\\Local\\Temp\\ipykernel_14356\\288486451.py:6: FutureWarning:

assing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=conteggio_nfc.index, y=conteggio_nfc.values, palette='viridis')
```



Conteggio modelli con e senza NFC

```

conteggio_ir_blaster = df['has_ir_blaster'].value_counts().sort_index()

# Crea un grafico a barre orizzontali per i conteggi di modelli con e senza IR blaster
plt.figure(figsize=(10, 6))
sns.barplot(y=conteggio_ir_blaster.index, x=conteggio_ir_blaster.values, palette='viridis', orient='h')
plt.title('Conteggio di Modelli con IR Blaster e senza IR Blaster')
plt.xlabel('Conteggio')
plt.ylabel('Numero di modelli con e senza IR blaster')
plt.yticks(ticks=[0, 1], labels=['No IR', 'Ha IR'])

# Aggiungere i numeri alla fine di ogni barra
for index, value in enumerate(conteggio_ir_blaster.values):
    plt.text(value, index, str(value), va='center', ha='left', fontsize=12)

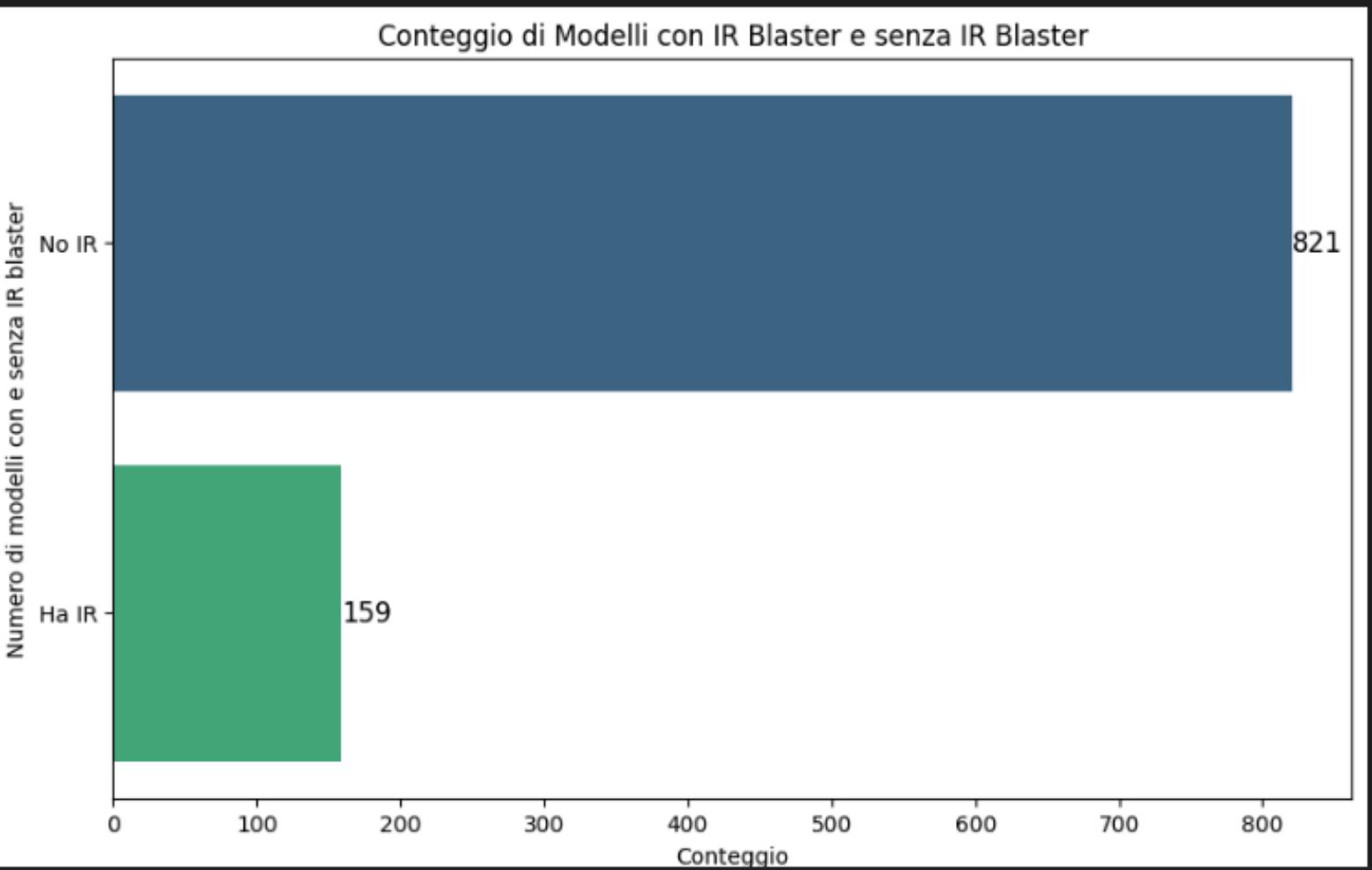
plt.show()

```

\Users\ale s\AppData\Local\Temp\ipykernel_5636\3487199670.py:5: FutureWarning:

Missing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(y=conteggio_ir_blaster.index, x=conteggio_ir_blaster.values, palette='viridis', orient='h')
```



Conteggio modelli con IR blaster e senza IR blaster

```

# Conteggio del numero di smartphone per brand
conteggio_processor_brand = df['processor_brand'].value_counts()

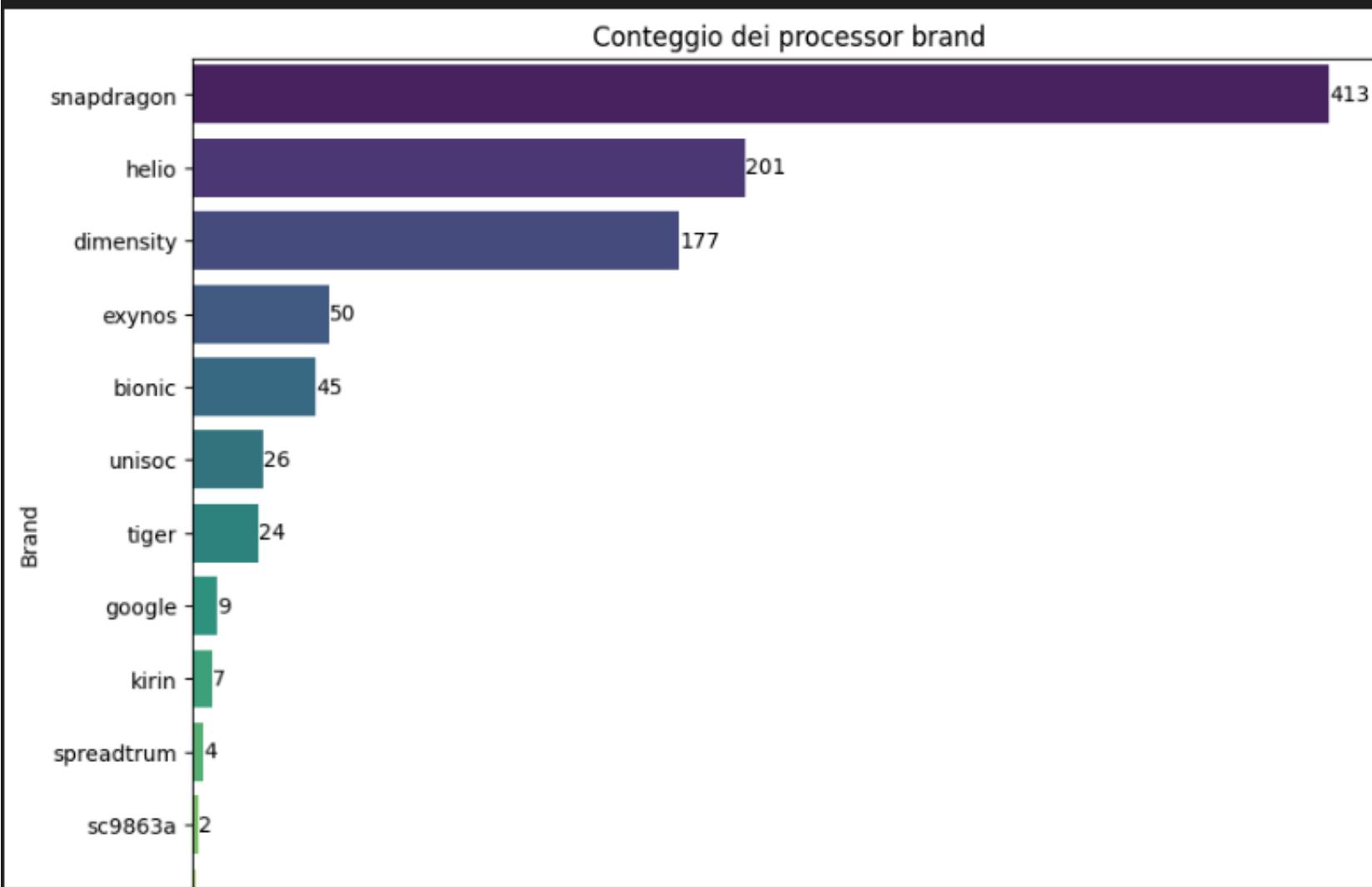
# Crea un grafico a barre orizzontali per i primi 10 brand
plt.figure(figsize=(10, 8))
sns.barplot(y=conteggio_processor_brand.index, x=conteggio_processor_brand.values, palette='viridis')
plt.title('Conteggio dei processor brand')
plt.xlabel('Conteggio')
plt.ylabel('Brand')
# Aggiungi il numero alla fine di ogni barra
for index, value in enumerate(conteggio_processor_brand.values):
    plt.text(value, index, str(value), va='center')
plt.show()

```

`:\Users\ale_s\AppData\Local\Temp\ipykernel_5636\978212636.py:6: FutureWarning:`

`assing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.`

```
sns.barplot(y=conteggio_processor_brand.index, x=conteggio_processor_brand.values, palette='viridis')
```



Conteggio Processor Brand

```
conteggio_num_cores = df['num_cores'].value_counts().sort_index()

# Crea un grafico a barre orizzontali per i num_cores
plt.figure(figsize=(10, 6))
sns.barplot(y=conteggio_num_cores.index, x=conteggio_num_cores.values, palette='viridis', orient='h')
plt.title('Conteggio dei num_cores')
plt.xlabel('Conteggio')
plt.ylabel('Numero di core')

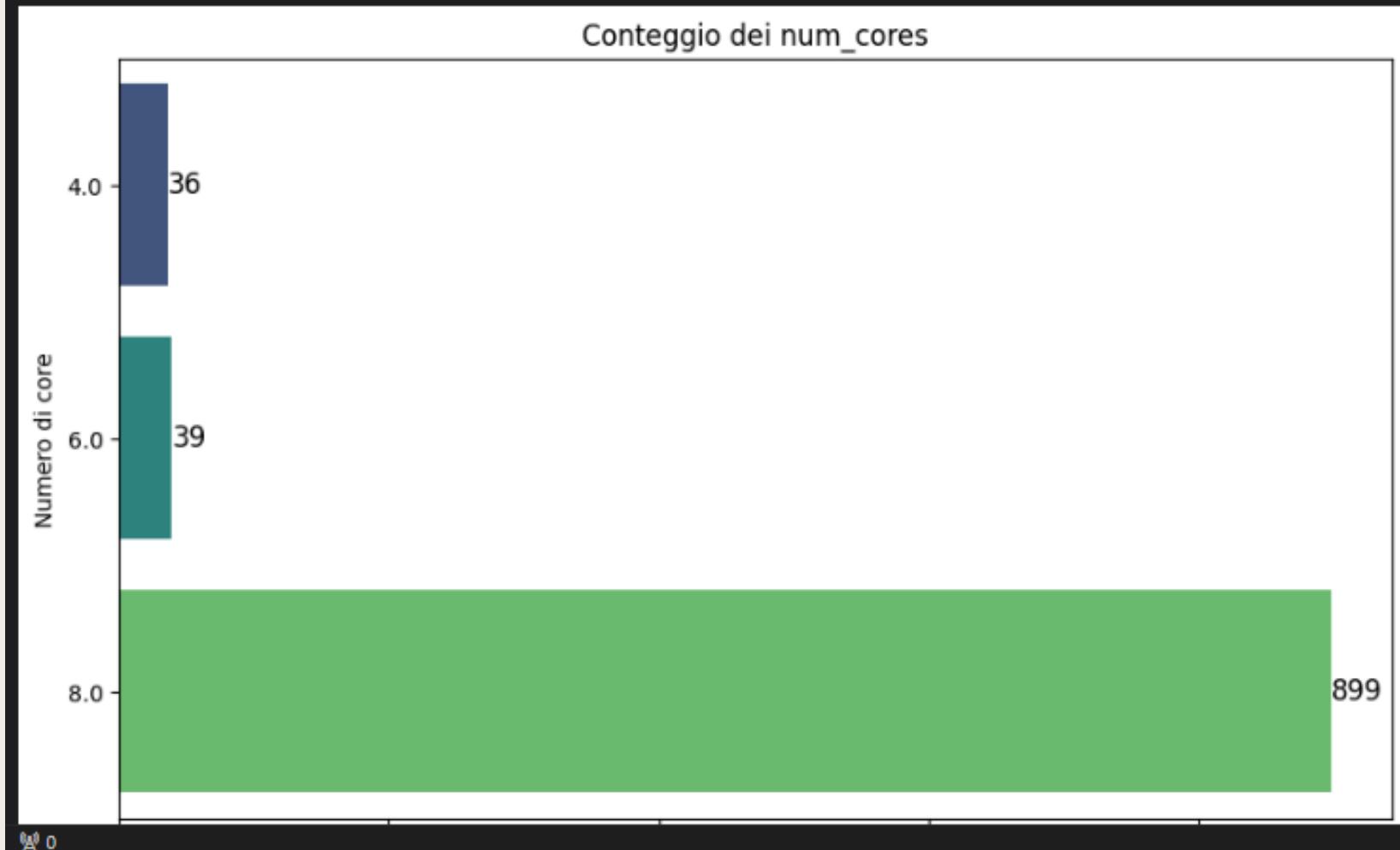
# Aggiungi il numero alla fine di ogni barra
for index, value in enumerate(conteggio_num_cores.values):
    plt.text(value, index, str(value), va='center', ha='left', fontsize=12)

plt.show()
```

C:\Users\ale s\AppData\Local\Temp\ipykernel_8772\3795722964.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(y=conteggio_num_cores.index, x=conteggio_num_cores.values, palette='viridis', orient='h')
```



Conteggio dei num cores

```

# contiene la colonna 'processor_speed'
conteggio_processor_speed = df['processor_speed'].value_counts().sort_index()

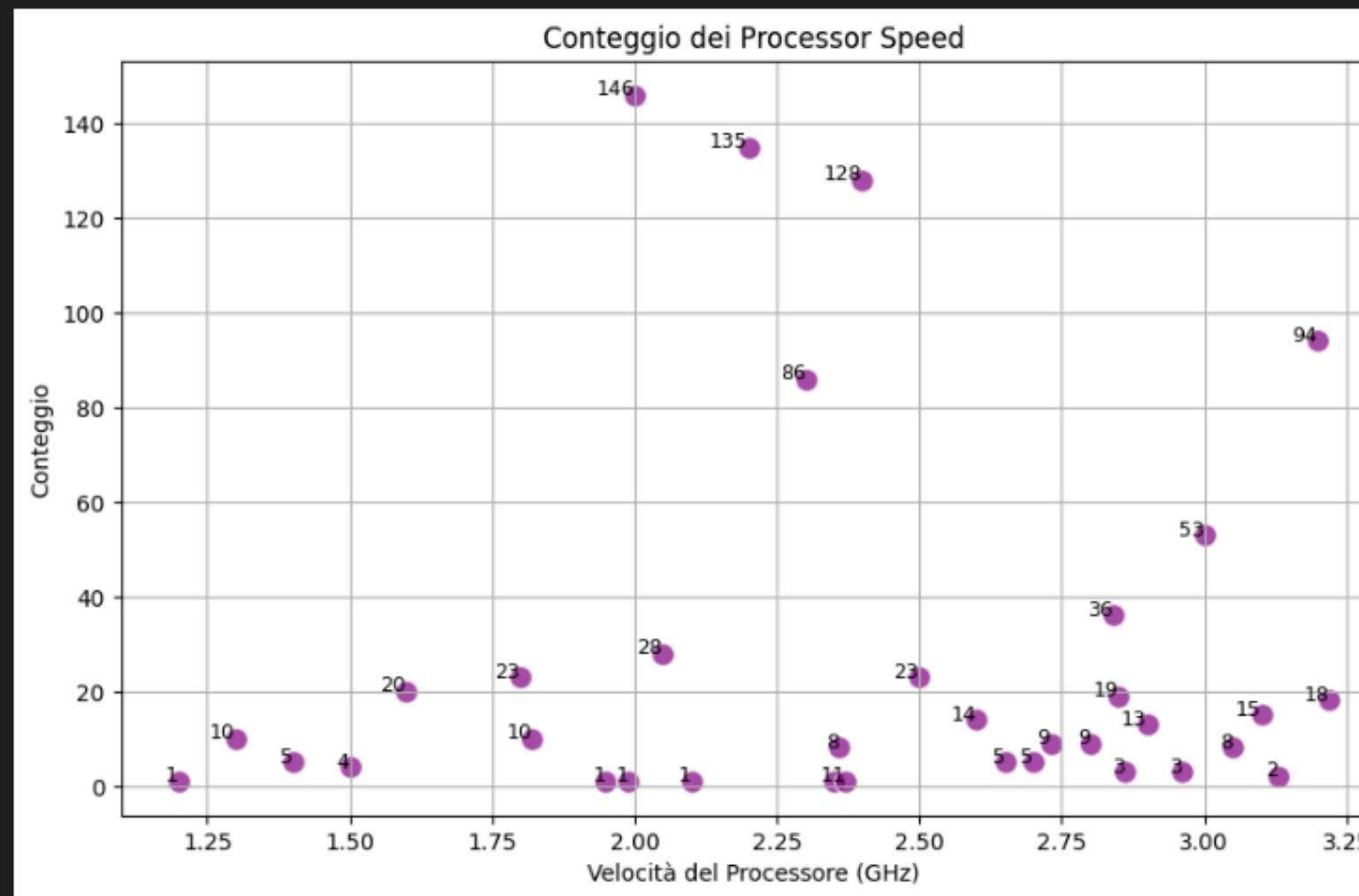
# Creare i dati per il grafico scatter
processor_speeds = conteggio_processor_speed.index
counts = conteggio_processor_speed.values

# Creare un grafico a dispersione (scatter plot) per processor speed
plt.figure(figsize=(10, 6))
plt.scatter(processor_speeds, counts, color='purple', alpha=0.7, edgecolors='w', s=100)
plt.title('Conteggio dei Processor Speed')
plt.xlabel('Velocità del Processore (GHz)')
plt.ylabel('Conteggio')

# Aggiungi i numeri vicino ai punti
for i, count in enumerate(counts):
    plt.text(processor_speeds[i], count, str(count), fontsize=9, ha='right')

plt.grid(True)
plt.show()

```



Conteggio dei Processor Speed

```

# Conteggio capacità batteria
conteggio_batteria = df['battery_capacity'].value_counts().head(10)

# Crea un grafico a barre orizzontali per i primi 10 modelli con capacità della batteria
plt.figure(figsize=(10, 8))
sns.barplot(y=conteggio_batteria.index, x=conteggio_batteria.values, palette='viridis', orient='h')
plt.title('Conteggio Capacità Batteria')
plt.xlabel('Conteggio')
plt.ylabel('Capacità Batteria (mAh)')

# Aggiungi il numero alla fine di ogni barra
for index, value in enumerate(conteggio_batteria.values):
    plt.text(value, index, str(value), va='center', ha='left', fontsize=12)

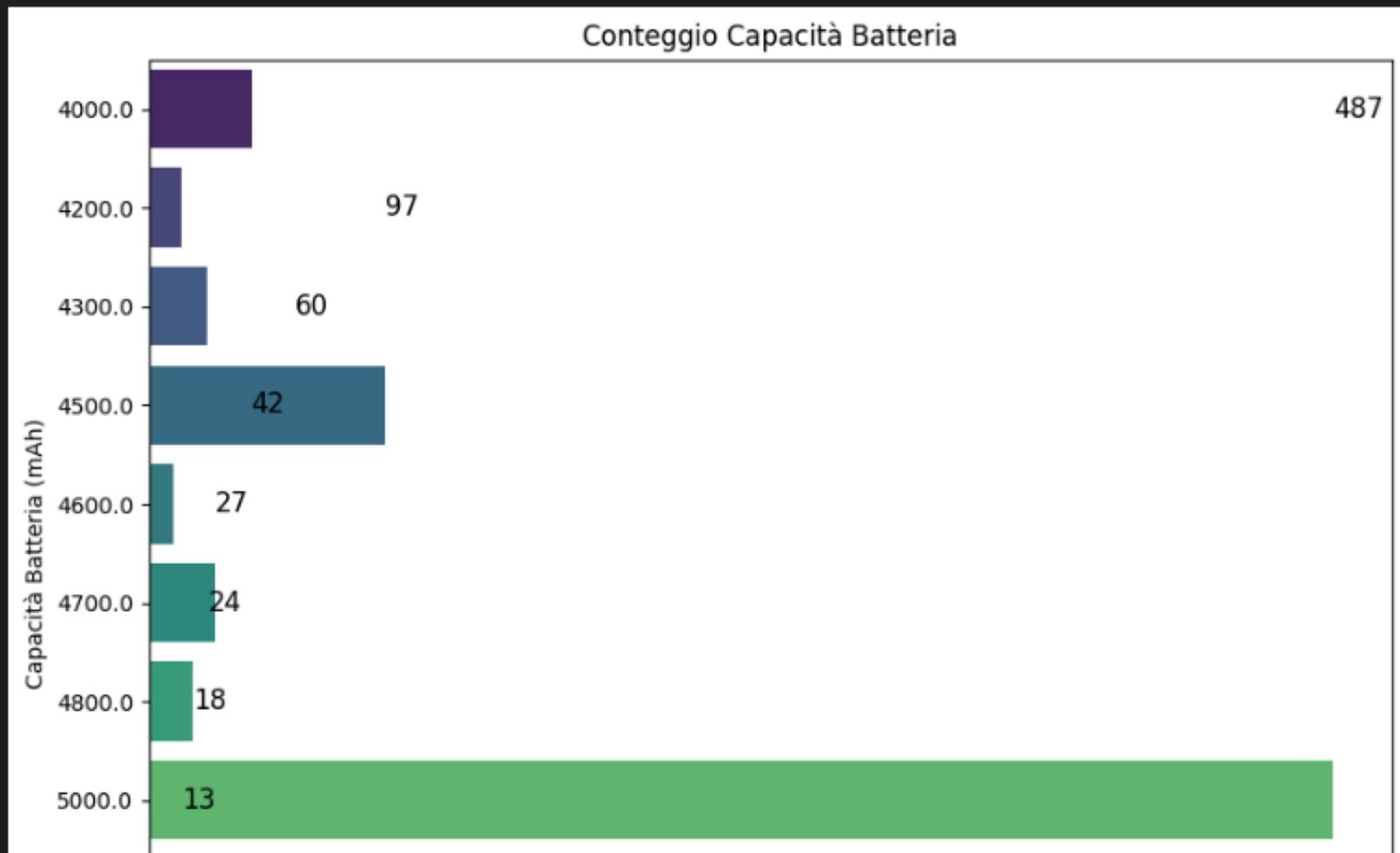
plt.show()

```

[C:\Users\ale s\AppData\Local\Temp\ipykernel_4796\3295017952.py:6: FutureWarning:](#)

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(y=conteggio_batteria.index, x=conteggio_batteria.values, palette='viridis', orient='h')
```



Capacità batteria

```

conteggio_fast_charging = df['fast_charging_available'].value_counts().sort_index()

# Crea un grafico a barre orizzontali per fast_charging_available
plt.figure(figsize=(10, 6))
sns.barplot(y=conteggio_fast_charging.index, x=conteggio_fast_charging.values, palette='viridis', orient='h')
plt.title('Conteggio di Modelli con e senza Ricarica Rapida')
plt.xlabel('Conteggio')
plt.ylabel('Disponibilità di Ricarica Rapida')
plt.yticks(ticks=[0, 1], labels=['No', 'Si'])

# Aggiungi il numero alla fine di ogni barra
for index, value in enumerate(conteggio_fast_charging.values):
    plt.text(value, index, str(value), va='center', ha='left', fontsize=12)

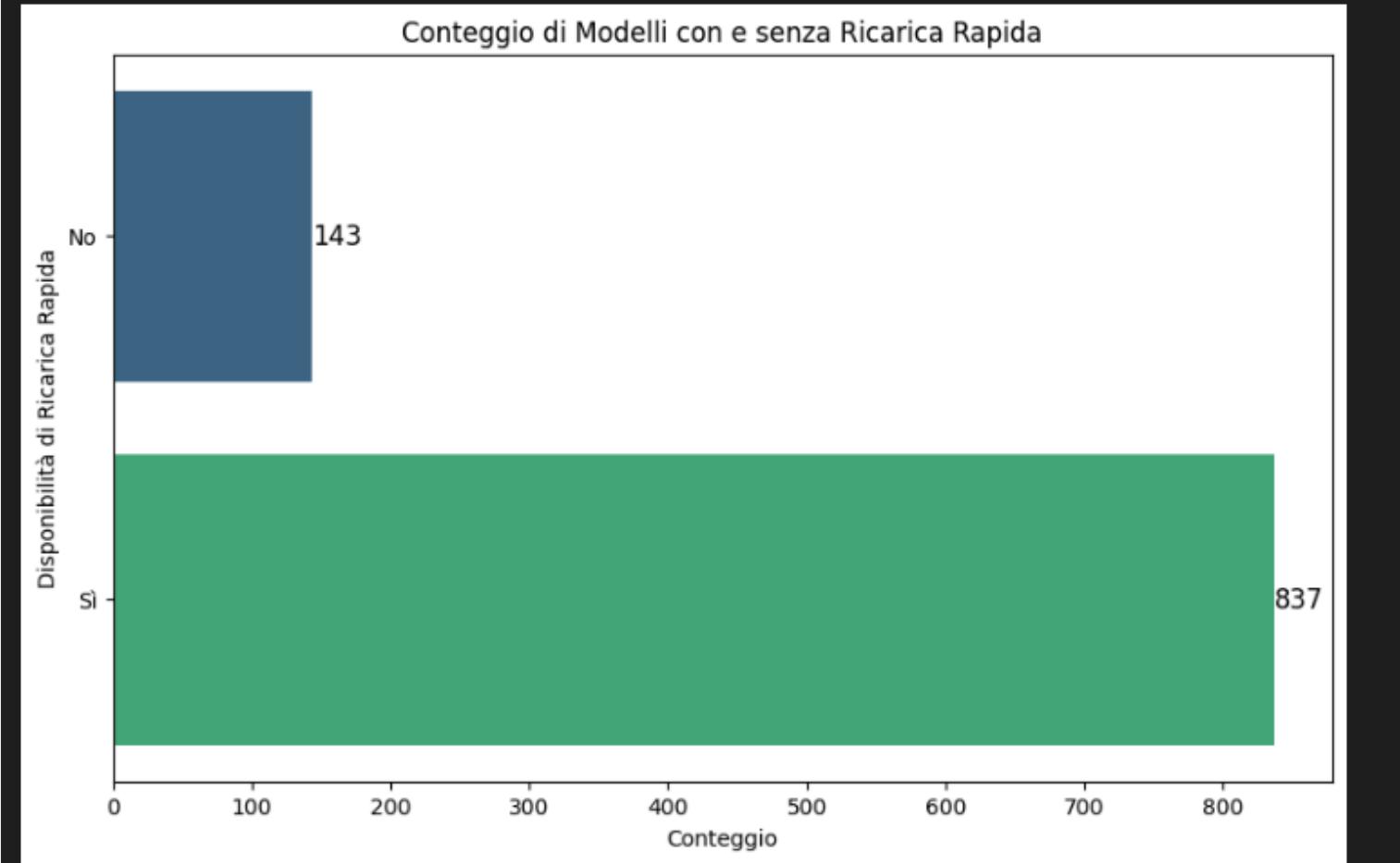
plt.show()

```

C:\Users\ale s\AppData\Local\Temp\ipykernel_4796\1603836255.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(y=conteggio_fast_charging.index, x=conteggio_fast_charging.values, palette='viridis', orient='h')
```



Conteggio di modelli con ricarica rapida

```

# Raggruppa i dati per brand e fast_charging_available
conteggio_per_brand = df.groupby(['brand_name', 'fast_charging_available']).size().unstack(fill_value=0)

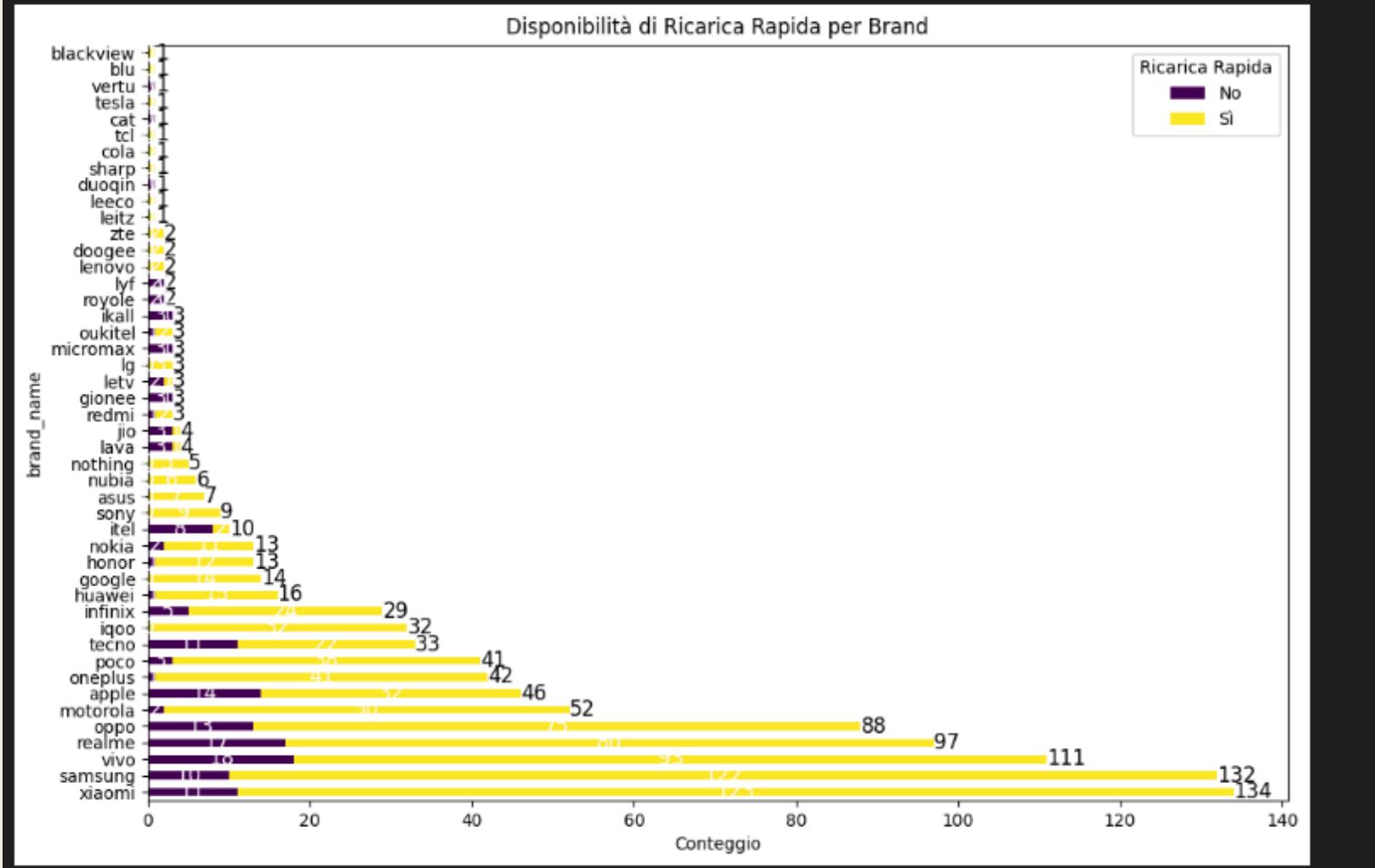
# Ordina i brand per numero totale di modelli (sommando le due colonne)
conteggio_per_brand['total'] = conteggio_per_brand.sum(axis=1)
conteggio_per_brand = conteggio_per_brand.sort_values(by='total', ascending=False).drop(columns='total')

# Crea un grafico a barre orizzontali per la ricarica rapida per ciascun brand
conteggio_per_brand.plot(kind='barh', stacked=True, figsize=(12, 8), colormap='viridis')
plt.title('Disponibilità di Ricarica Rapida per Brand')
plt.xlabel('Conteggio')
plt.ylabel('brand_name')
plt.legend(title='Ricarica Rapida', labels=['No', 'Sì'])

# Aggiungi il numero alla fine di ogni barra
for index, (no_fast, yes_fast) in enumerate(zip(conteggio_per_brand[0], conteggio_per_brand[1])):
    plt.text(no_fast + yes_fast, index, str(no_fast + yes_fast), va='center', ha='left', fontsize=12)
    plt.text(no_fast / 2, index, str(no_fast), va='center', ha='center', fontsize=12, color='white')
    plt.text(no_fast + yes_fast / 2, index, str(yes_fast), va='center', ha='center', fontsize=12, color='white')

plt.show()

```



Disponibilità di ricarica rapida per brand

```

# Ordina i dati in base alla capacità della RAM e seleziona i primi 20 modelli
top_20_models_ram = df[['model', 'ram_capacity']].sort_values(by='ram_capacity', ascending=False).head(20)

# Crea un grafico a barre orizzontali per i primi 20 modelli con la RAM più alta
plt.figure(figsize=(12, 8))
sns.barplot(y=top_20_models_ram['model'], x=top_20_models_ram['ram_capacity'], palette='viridis')
plt.title('Capacità di RAM per i Primi 20 Modelli')
plt.xlabel('Capacità di RAM (GB)')
plt.ylabel('Modello')

# Aggiungi il numero alla fine di ogni barra
for index, value in enumerate(top_20_models_ram['ram_capacity']):
    plt.text(value, index, str(value), va='center', ha='left', fontsize=12)

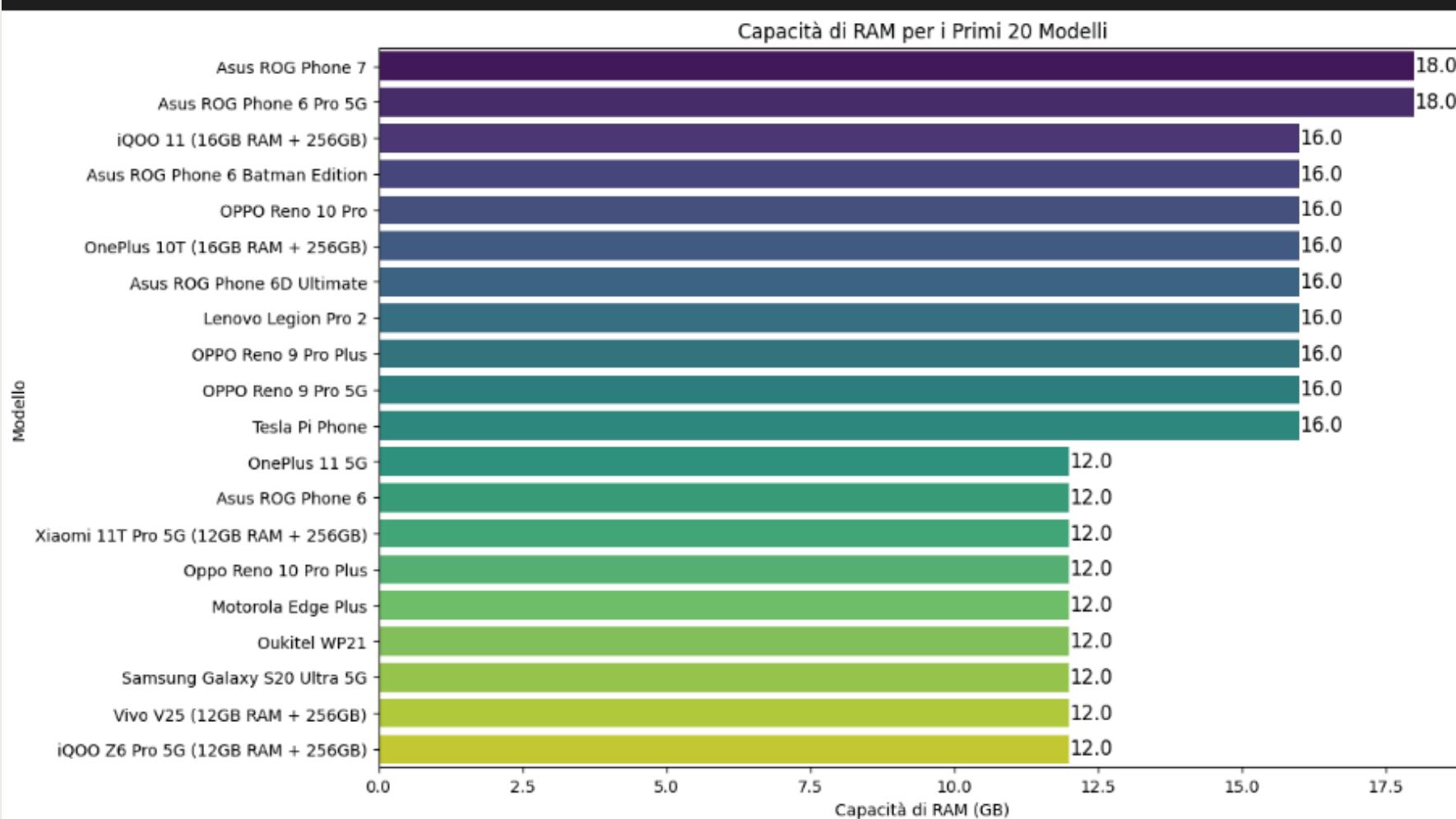
plt.show()

```

:\\Users\\ale_s\\AppData\\Local\\Temp\\ipykernel_16856\\2493363397.py:6: FutureWarning:

assing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'y' variable to 'hue' and set 'legend=False' for the same effect.

```
sns.barplot(y=top_20_models_ram['model'], x=top_20_models_ram['ram_capacity'], palette='viridis')
```



Capacità RAM per i primi 20 modelli

```

# Ordina i dati in base alla memoria interna e seleziona i primi 20 modelli
top_20_models_memory = df[['model', 'internal_memory']].sort_values(by='internal_memory', ascending=False).head(20)

# Crea un grafico a barre orizzontali per i primi 20 modelli con la memoria interna più alta
plt.figure(figsize=(12, 8))
sns.barplot(y=top_20_models_memory['model'], x=top_20_models_memory['internal_memory'], palette='viridis')
plt.title('Capacità di memoria interna per i Primi 20 Modelli')
plt.xlabel('Capacità di memoria interna (GB)')
plt.ylabel('Modello')

# Aggiungi il numero alla fine di ogni barra
for index, value in enumerate(top_20_models_memory['internal_memory']):
    plt.text(value, index, str(value), va='center', ha='left', fontsize=12)

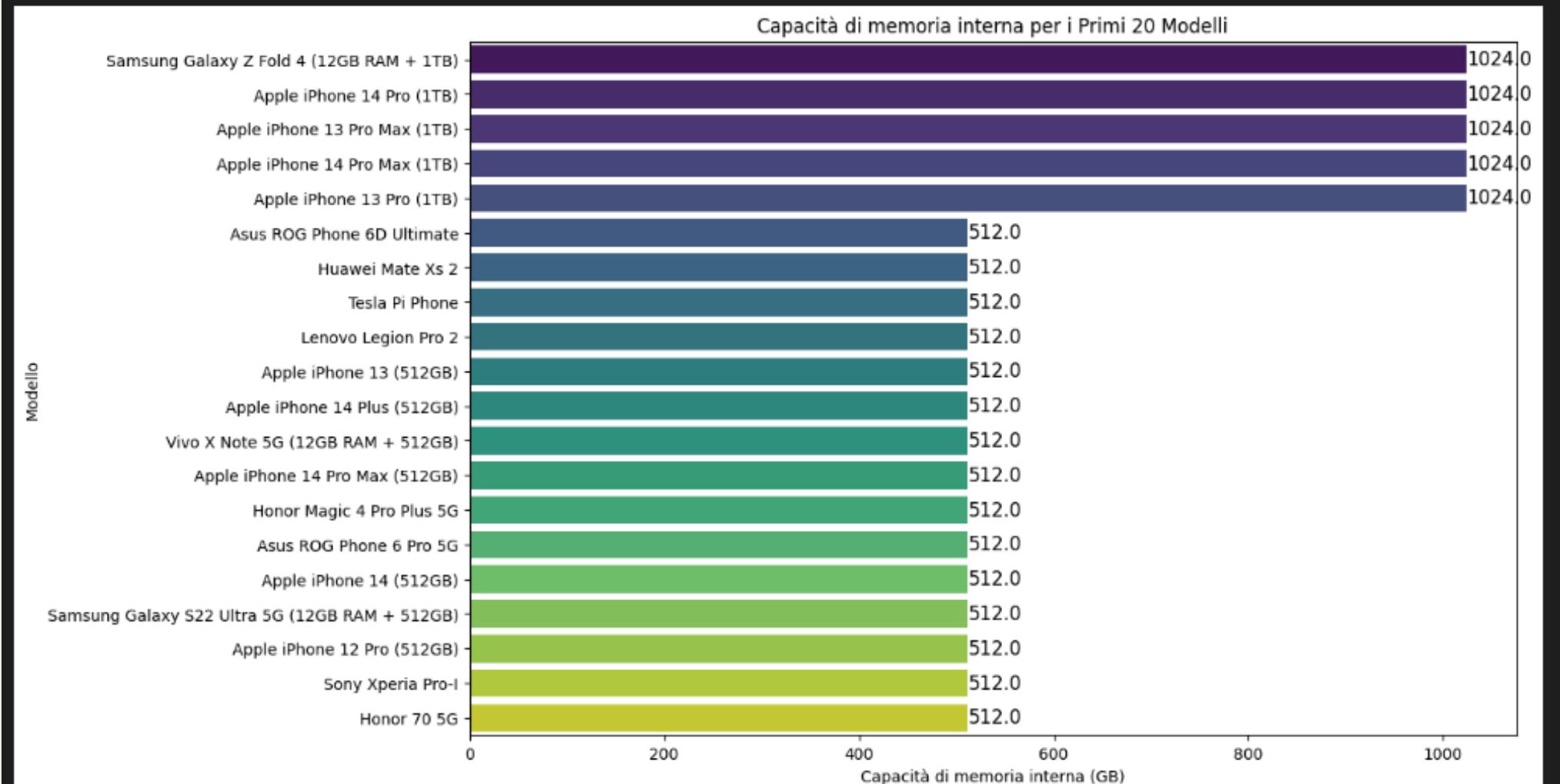
plt.show()

```

C:\Users\ale_s\AppData\Local\Temp\ipykernel_16856\1896306557.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(y=top_20_models_memory['model'], x=top_20_models_memory['internal_memory'], palette='viridis')
```



Capacità di memoria interna per i primi 20 modelli

```

# Ordina i dati in base alle dimensioni dello schermo e seleziona i primi 20 modelli
top_20_models_screen = df[['model', 'screen_size']].sort_values(by='screen_size', ascending=False).head(20)

# Crea un grafico a barre orizzontali per i primi 20 modelli con lo schermo più grande
plt.figure(figsize=(12, 8))
sns.barplot(y=top_20_models_screen['model'], x=top_20_models_screen['screen_size'], palette='viridis')
plt.title('Dimensione schermo per i Primi 20 Modelli')
plt.xlabel('Dimensione schermo')
plt.ylabel('Modello')

# Aggiungi il numero alla fine di ogni barra
for index, value in enumerate(top_20_models_screen['screen_size']):
    plt.text(value, index, str(value), va='center', ha='left', fontsize=12)

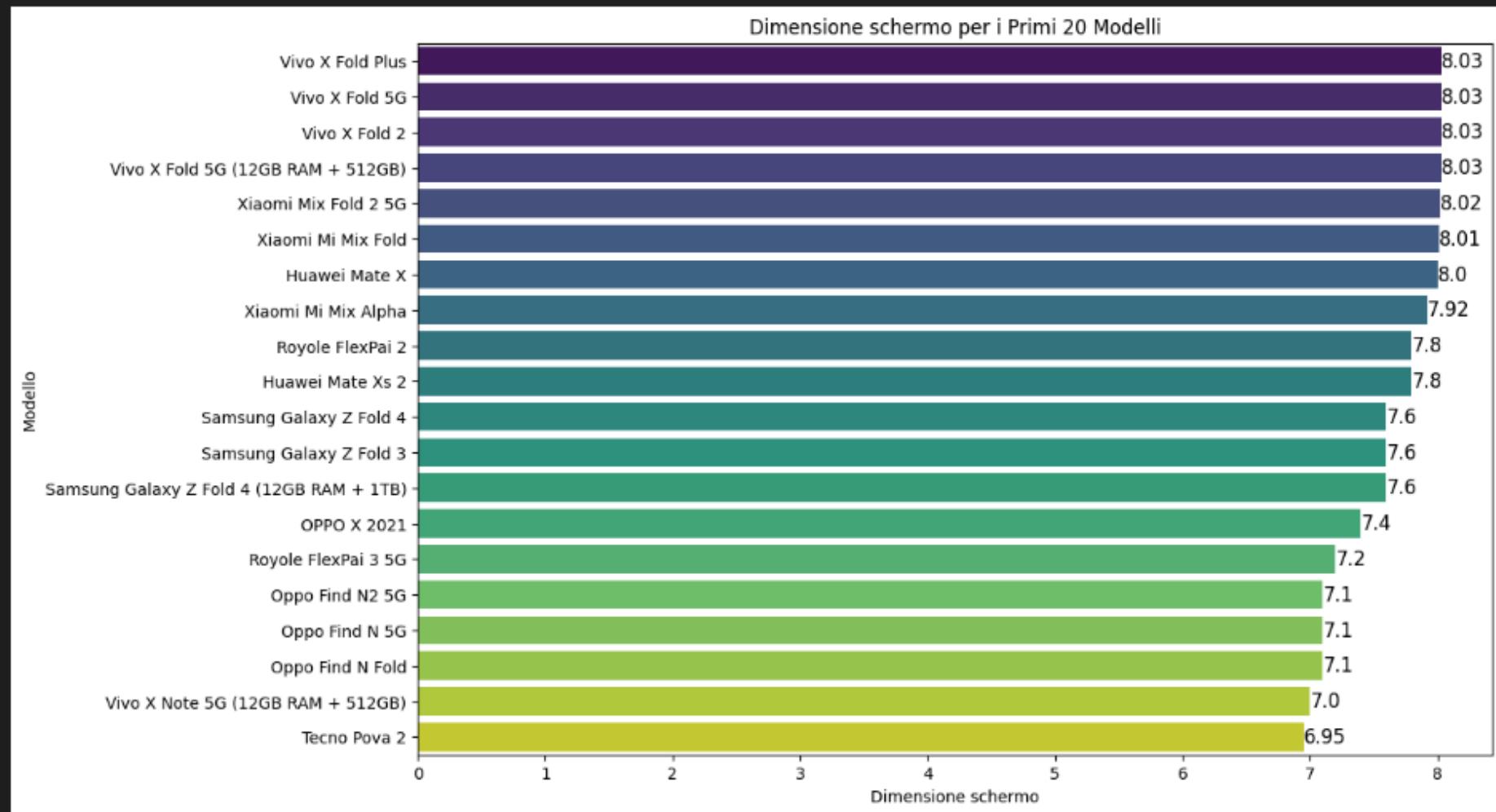
plt.show()

```

C:\Users\ale s\AppData\Local\Temp\ipykernel_16856\449055659.py:6: FutureWarning:

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'y' variable to 'hue' and set 'legend=False' for the same effect.

```
sns.barplot(y=top_20_models_screen['model'], x=top_20_models_screen['screen_size'], palette='viridis')
```



Dimensione schermo per i primi 20 modelli

