

Box Plot e Analisi Esplorativa dei Dati per Machine Learning

Introduzione ai Box Plot

I box plot, noti anche come diagrammi a scatola e baffi, sono strumenti grafici fondamentali per l'analisi esplorativa dei dati (EDA). Questi grafici offrono una rappresentazione sintetica ma completa della distribuzione di un insieme di dati numerici, evidenziando caratteristiche essenziali come:

- Tendenza centrale
- Dispersione
- Simmetria/asimmetria
- Presenza di outlier

Anatomia di un Box Plot

Un box plot si compone dei seguenti elementi:

1. Scatola (Box):

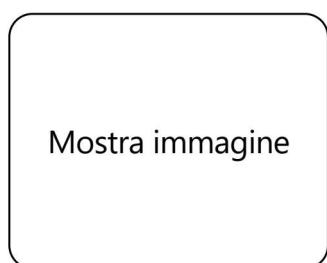
- Rappresenta i quartili dei dati
- Il bordo inferiore corrisponde al primo quartile (Q_1 , 25° percentile)
 - La linea centrale rappresenta la mediana (Q_2 , 50° percentile)
 - Il bordo superiore corrisponde al terzo quartile (Q_3 , 75° percentile)

2. Baffi (Whiskers):

- Estensioni della scatola che indicano la variabilità al di fuori dei quartili
- Tipicamente si estendono fino a $1.5 \times \text{IQR}$ (dove $\text{IQR} = Q_3 - Q_1$)
 - Il limite inferiore è $\max(\min(\text{dati}), Q_1 - 1.5 \times \text{IQR})$
 - Il limite superiore è $\min(\max(\text{dati}), Q_3 + 1.5 \times \text{IQR})$

3. Outlier:

Punti oltre i baffi, rappresentati come punti individuali



Mostra immagine

Box Plot nell'Analisi Esplorativa dei Dati (EDA)

Vantaggi dei Box Plot in EDA

- **Compattezza:** Riassumono grandi quantità di dati in una rappresentazione visiva semplice
- **Confrontabilità:** Permettono di confrontare facilmente più distribuzioni
- **Robustezza:** Basati su statistiche robuste (mediana e quartili) resistenti agli outlier
- **Interpretabilità:** Forniscono informazioni immediate sulla forma della distribuzione

Cosa ci raccontano i Box Plot

1. Posizione della mediana all'interno della scatola:

- Mediana centrata: suggerisce simmetria della distribuzione
- Mediana spostata: indica asimmetria (skewness) nella distribuzione

2. Dimensione della scatola:

- Box ampio: alta variabilità interquartile
- Box stretto: bassa variabilità interquartile

3. Lunghezza dei baffi:

- Baffi simmetrici: distribuzione tendenzialmente simmetrica
- Baffi asimmetrici: distribuzione asimmetrica (coda lunga)

4. Presenza di outlier:

- Molti outlier: possibile presenza di anomalie o distribuzione a coda pesante
- Assenza di outlier: distribuzione più compatta o uniformemente distribuita

Box Plot e Feature Engineering

Il Feature Engineering è il processo di trasformazione dei dati grezzi in feature che migliorano le prestazioni dei modelli di machine learning. I box plot giocano un ruolo cruciale in questa fase.

Identificazione e Gestione degli Outlier

Gli outlier possono influenzare negativamente le prestazioni dei modelli ML. I box plot aiutano a:

1. Rilevare automaticamente gli outlier: Valori al di fuori dei baffi sono potenziali candidati per trattamenti speciali

2. Quantificare la gravità degli outlier: La distanza dai baffi fornisce una misura dell'anomalia

3. Guidare strategie di gestione:

- Rimozione: eliminare outlier estremi
- Winsorizzazione: limitare valori estremi ai limiti dei baffi
- Trasformazione: applicare trasformazioni (es. logaritmica) per ridurre l'impatto

Normalizzazione e Standardizzazione

I box plot aiutano a identificare quali feature necessitano di normalizzazione:

1. Ampia variabilità: Feature con box plot ampi potrebbero dominare quelle con box plot stretti

2. Asimmetria: Distribuzione asimmetrica potrebbe beneficiare di trasformazioni specifiche

Binning e Discretizzazione

Per convertire variabili continue in categorie, i box plot aiutano a:

- 1. Definire punti di taglio naturali:** I quartili possono suggerire soglie significative
- 2. Identificare cluster:** Le densità di punti visibili nei box plot possono suggerire gruppi naturali

Box Plot e Correlazione tra Variabili

Box Plot Condizionali

I box plot condizionali mostrano la distribuzione di una variabile numerica rispetto a categorie di un'altra variabile, rivelando relazioni significative:

- 1. Differenze nelle mediane:** Indicano effetti della variabile categorica sulla tendenza centrale
- 2. Differenze nella dispersione:** Suggeriscono eteroschedicità (varianza non costante)

Analisi di Correlazione Multivariata

Combinando box plot con altre tecniche:

- 1. Box plot a griglia:** Matrici di box plot che mostrano relazioni tra più variabili
- 2. Box plot + scatter plot:** Complementari per comprendere relazioni non lineari

Identificazione di Interazioni

- 1. Box plot stratificati:** Mostrano come la relazione tra due variabili cambia in base a una terza
- 2. Pattern nelle mediane:** Trend non paralleli nelle mediane suggeriscono interazioni

Interpretazione Avanzata dei Box Plot per ML

Rilevazione di Multimodalità

Box plot con "strozzature" (restringimenti nella densità) possono suggerire distribuzioni multimodali, che potrebbero richiedere:

- 1. Clustering:** Separazione dei dati in sottopopolazioni
- 2. Modelli multipli:** Training di modelli specializzati per ciascuna modalità

Valutazione della Stabilità delle Feature

Confrontando box plot tra set di training e validation:

- 1. Consistenza:** Box plot simili indicano feature stabili
- 2. Differenze significative:** Suggeriscono potential shift o problemi di generalizzazione

Box Plot per Feature Importance

Analizzando box plot delle feature rispetto alla variabile target:

1. **Separazione netta**: Feature con box plot ben separati per diverse classi target sono potenzialmente più discriminative

2. **Sovrapposizione**: Suggerisce minore potere predittivo

Implementazione Pratica

Python con Matplotlib/Seaborn

python

Copia

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

# Caricamento dataset
df = pd.read_csv('dataset.csv')

# Box plot base
plt.figure(figsize=(10, 6))
sns.boxplot(x=df['feature_name'])
plt.title('Box Plot di Feature Name')
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()

# Box plot condizionale
plt.figure(figsize=(12, 7))
sns.boxplot(x='categorical_feature', y='numeric_feature', data=df)
plt.title('Box Plot Condizionale')
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()

# Box plot con punti sovrapposti (per visualizzare distribuzione)
plt.figure(figsize=(12, 7))
sns.boxplot(x='categorical_feature', y='numeric_feature', data=df)
sns.stripplot(x='categorical_feature', y='numeric_feature', data=df,
              size=4, color='.3', alpha=0.6)
plt.title('Box Plot con Distribuzione dei Punti')
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```

Identificazione programmatica di outlier

```
def identify_outliers(df, column):
    """Identifica outlier usando il metodo IQR."""
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)]
    return outliers, lower_bound, upper_bound

# Utilizzo
outliers, lower, upper = identify_outliers(df, 'feature_name')
print(f"Trovati {len(outliers)} outlier")
print(f"Limite inferiore: {lower}, Limite superiore: {upper}")
```

Best Practices per Box Plot in Machine Learning

1. Utilizzare box plot all'inizio del processo di EDA:

- Creare box plot per tutte le feature numeriche
- Identificare immediatamente anomalie e pattern distributivi

2. Combinare box plot con istogrammi e density plot:

- Box plot forniscono statistiche sommarie
- Iistogrammi e density plot mostrano la forma completa della distribuzione

3. Box plot per feature categoriche:

- Utilizzare diagrammi a barre per frequenze
- Box plot per metriche aggregate per categoria

4. Box plot prima e dopo le trasformazioni:

- Confrontare distribuzioni pre/post-processing
- Verificare l'efficacia delle tecniche di feature engineering

5. Box plot per analisi residui:

- Verificare assunzioni del modello
- Identificare pattern nei residui condizionati alle feature

Conclusioni

I box plot rappresentano uno strumento fondamentale nell'arsenale di un data scientist, particolarmente prezioso nella fase di esplorazione e preparazione dei dati per il machine learning. La loro capacità di

condensare informazioni distribuzionali in una rappresentazione compatta li rende ideali per:

1. Comprendere rapidamente la struttura dei dati
2. Individuare anomalie e valori estremi
3. Guidare decisioni informate nel feature engineering
4. Valutare l'efficacia delle trasformazioni applicate
5. Comunicare in modo efficace le caratteristiche dei dati

L'integrazione sistematica dei box plot nel flusso di lavoro di machine learning contribuisce significativamente al miglioramento delle prestazioni dei modelli, facilitando la pulizia dei dati, l'ingegnerizzazione delle feature e l'interpretazione dei risultati.