# PREDIZIONE DEL LIVELLO DI PROGRESSIONE DEL DIABETE:

# Trainign di un modello ML

PROGETTO FINALE DEL CORSO MACHINE LEARNING

GIULIA COSENZA

# 1-Lo scopo del progetto

L'obiettivo di questo progetto è addestrare un modello di machine learning per predire il livello di progressione della malattia diabetica ad un anno di distanza, utilizzando un set di dati medici.

GIULIA COSENZA

# 2-I dati

Il dataset utilizzato per l'addestramento del modello è **Diabetes** della libreria **scikit-learn**.
Si tratta di un insieme di dati medici utilizzato comunemente per problemi di regressione. Esso contiene informazioni relative a pazienti affetti da diabete, con l'obiettivo di predire il livello di progressione della malattia ad un anno di distanza.

## Il dataset è strutturato in:

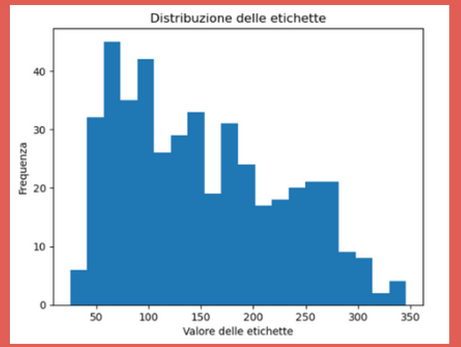
- **442** campioni di pazienti
- **10** variabili numeriche indipendenti, che rappresentano caratteristiche cliniche

## Le variabili (o features):

- **Age**: Età del paziente in anni
- **Sex**: Sesso del paziente (0 per femmina, 1 per maschio)
- **BMI** (Body Mass Index): Indice di massa corporea
- **BP** (Blood Pressure): Pressione sanguigna media
- **S1**: Livello totale di colesterolo sierico
- **S2** : Livello di colesterolo LDL
- **S3** : Livello di colesterolo HD
- **S4**: Rapporto tra il colesterolo totale e l'HDL
- **S5** : Valore logaritmico dei livelli di trigliceridi sierici
- **S6** : Livelli di zucchero nel sangue

## Le labels:

La variabile y rappresenta il livello di progressione della malattia a un anno di distanza e si tratta di un valore continuo. La distribuzione spazia da 25 a 346 ed è asimmetrica verso sinistra.

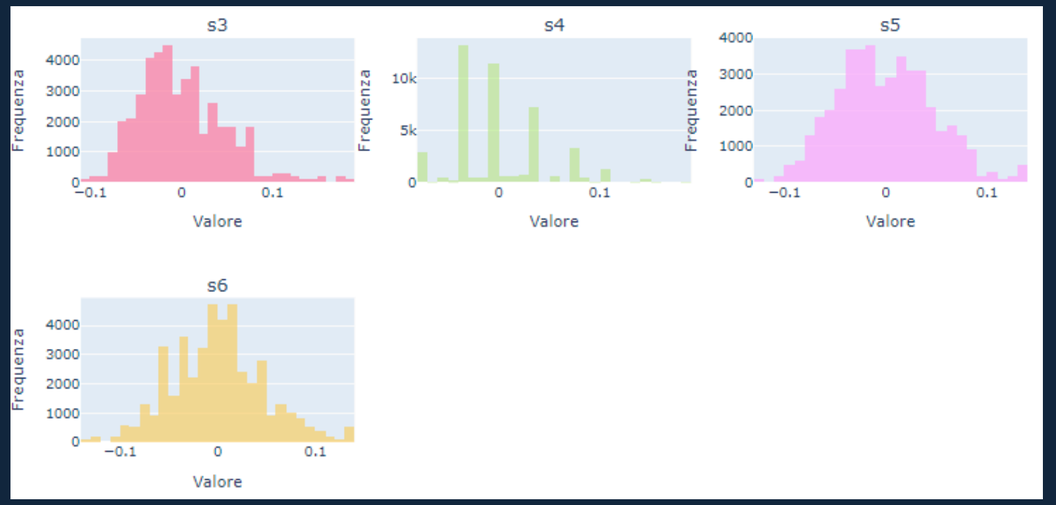


# 3-Analisi esplorativa dei dati

| | |
|---|---|
| **1 .Presenza di valori nulli** | **Non si osservano valori NaN** |

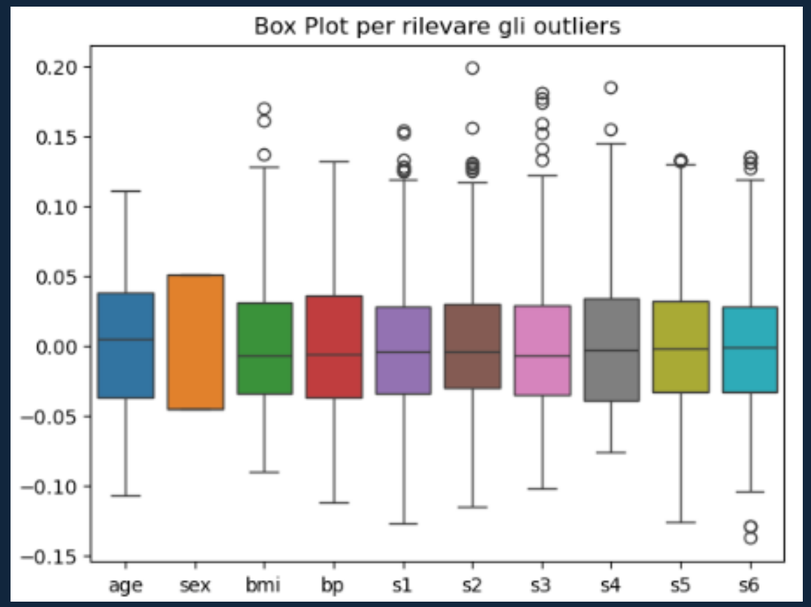
# 3-Analisi esplorativa dei dati

| | |
|---|---|
| **2. Distribuzione delle features** | **Le feature mostrano una distribuzione con media 0 e varianza uniforme. La magnitudine complessiva di ogni feature è bilanciata rispetto al dataset. Non è necessaria ulteriore standardizzazione o normalizzazione.** |

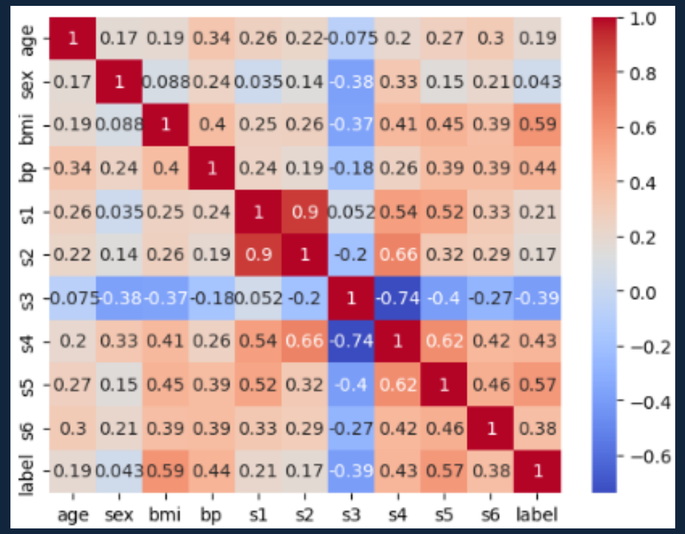
# 3-Analisi esplorativa dei dati

| | |
|---|---|
| **3. Presenza di valori outliers** | **Si osservano outliers superiori per le features bmi, s1, s2, s3, s4,s5 e s6.<br>Si osservano outliers inferiori per la feature s6.** |



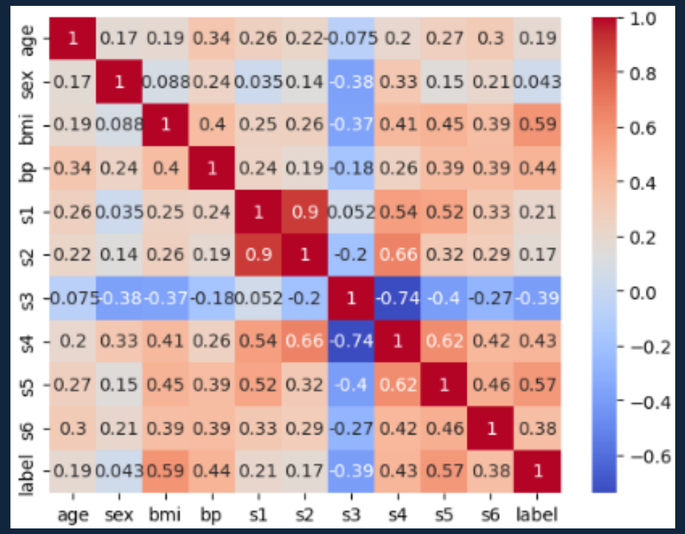
# 3-Analisi esplorativa dei dati

| | |
|---|---|
| 4. Correlazione tra le variabili | Correlazione positiva:<br>• s1 e s2: Correlazione positiva molto forte<br>• s2 e s4, s5 e s4: Correlazione positiva forte<br>• s1 e s4, s1 e s5: Correlazione positiva moderatamente forte<br>Correlazione negativa:<br>• s3 e s4: Correlazione negativa molto forte |



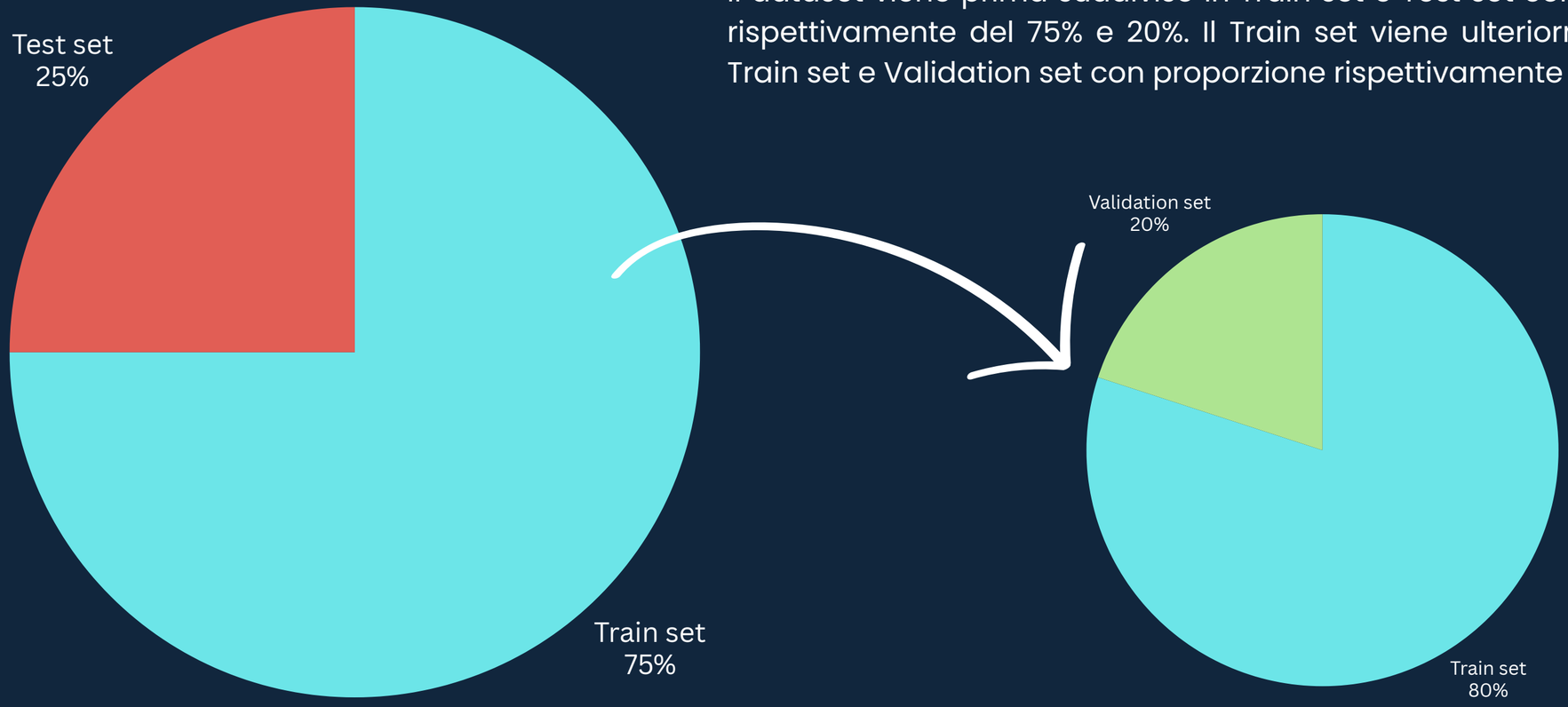
# 3-Analisi esplorativa dei dati

| | |
|---|---|
| **5. Correlazione tra feature e labels** | • **s5 e bmi mostrano una correlazione positiva più evidente rispetto alle altre feature con la variabile target;**<br>• **bp, s3, s4 e s6 hanno una correlazione più moderata;**<br>• **age, sex, s1, s2 mostrano correlazioni più deboli e quasi assenti con la variabile target.** |



# 4-Suddivisione del set

Il dataset viene prima suddiviso in Train set e Test set con una proporzione rispettivamente del 75% e 20%. Il Train set viene ulteriormente splittato in Train set e Validation set con proporzione rispettivamente del 80% e 20%



# 5-Baseline model: Regressione lineare

Considerando le correlazioni moderate di alcune feature con la label,  viene scelta la Regressione lineare come Baseline Model.

Il modello viene allenato sui dati di train e testato sul set di validazione per **valutarne le performance**.

```
-- Risultati sui dati di addestramento --
Mean Absolute Error (MAE) - Train: 43.297615418575
Mean Squared Error (MSE) - Train: 2872.1025369384
R^2 Score - Train: 0.5218122703028829

-- Risultati sui dati di validazione --
Mean Absolute Error (MAE) - Val: 47.66031147285695
Mean Squared Error (MSE) - Val: 3280.9353227550155
R^2 Score - Val: 0.4527758593380576
```

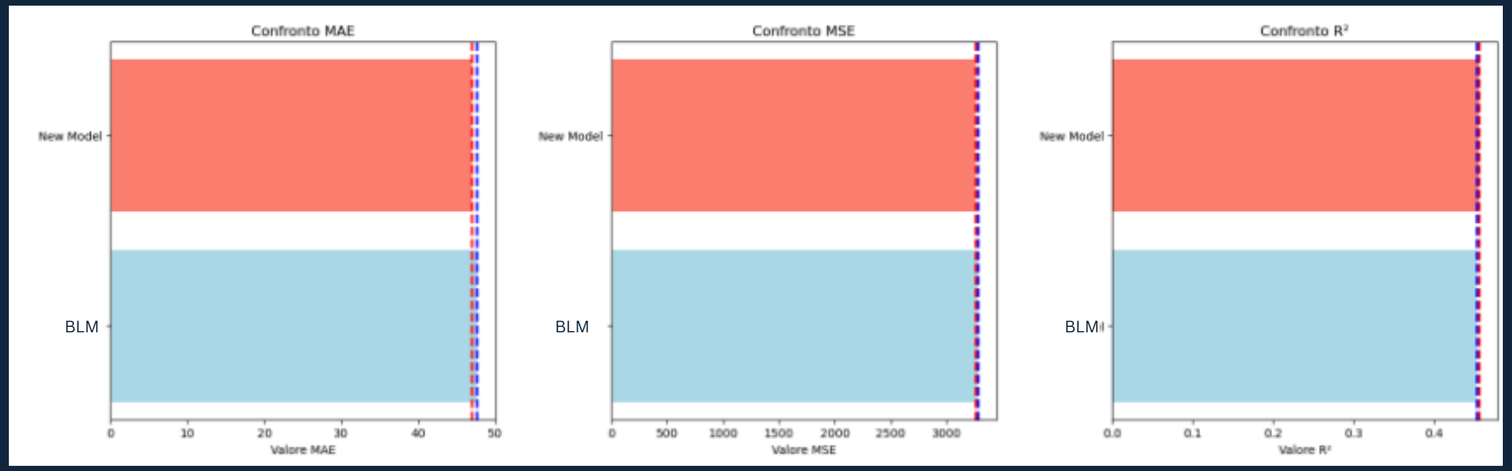
Il modello viene allenato sui dati di train e testato sullo stesso set di train per **valutarne eventuale overfit**.

**Il modello sembra moderatamente bilanciato, con performance simili su entrambi i set (train e test)**

# 6-Ottimizzazione del BLM

Per analizzare l'impatto delle caratteristiche del train set sulle performance del baseline model (BLM) si valuta se alcuni accorgimenti applicati al train set possano migliorare le performance del modello di Regressione Lineare.

## 6.1-Rimozione degli outliers

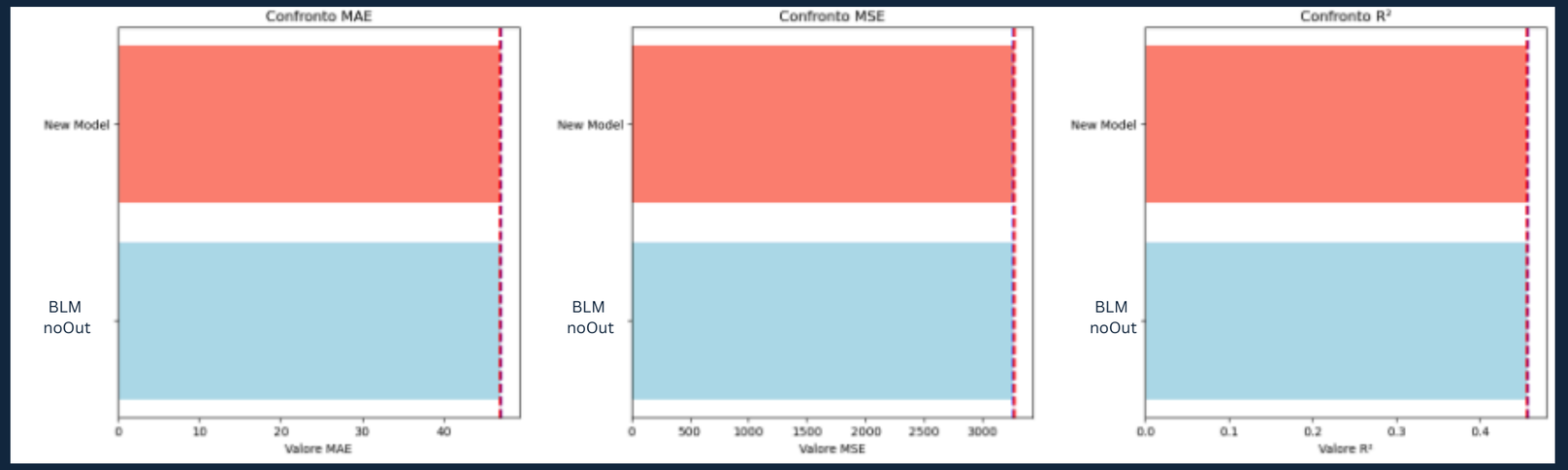



**Rimuovere gli outliers dal training set migliora le prestazioni: questo suggerisce che gli outliers stavano influenzando negativamente il modello.**

# 6-Ottimizzazione del BLM

Per analizzare l'impatto delle caratteristiche del train set sulle performance del baseline model (BLM) si valuta se alcuni accorgimenti applicati al train set possano migliorare le performance del modello di Regressione Lineare.

## 6.2-Rimozione delle feature tra loro correlate

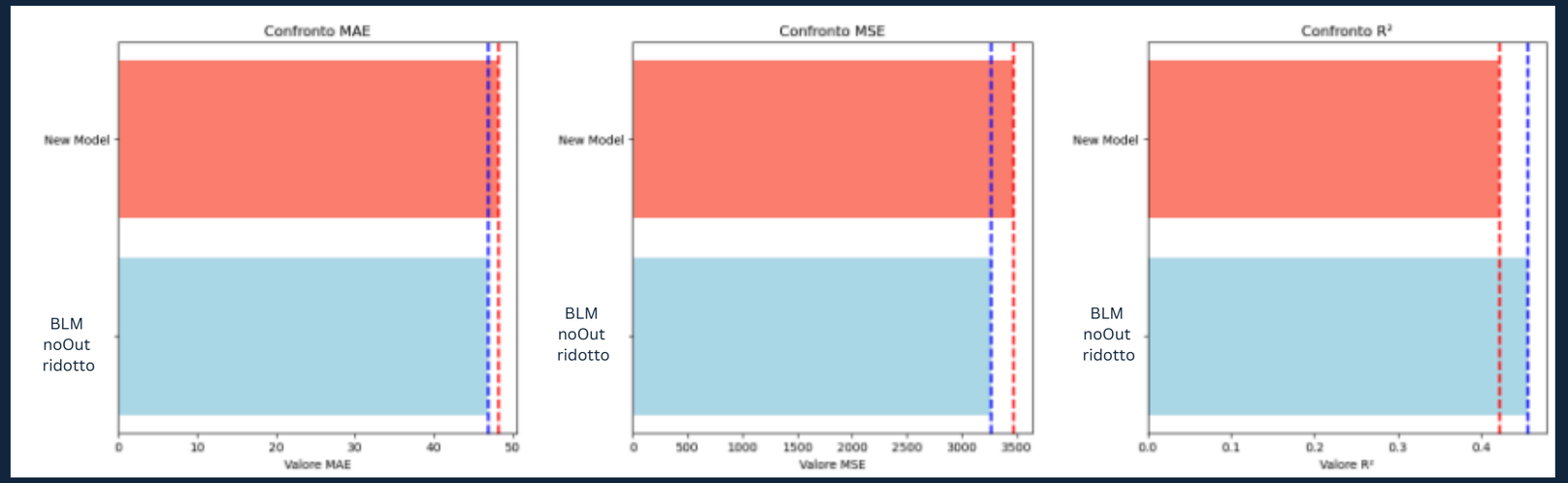



**Rimuovere anche le feature altamente correlate tra loro porta a un lieve miglioramento delle prestazioni del modello in termini di MAE ma non in termini di MSE, rispetto al BLM senza outliers.**

# 6-Ottimizzazione del BLM

Per analizzare l'impatto delle caratteristiche del train set sulle performance del baseline model (BLM) si valuta se alcuni accorgimenti applicati al train set possano migliorare le performance del modello di Regressione Lineare.

## 6.3-Selezione delle feature più correlate alle labels



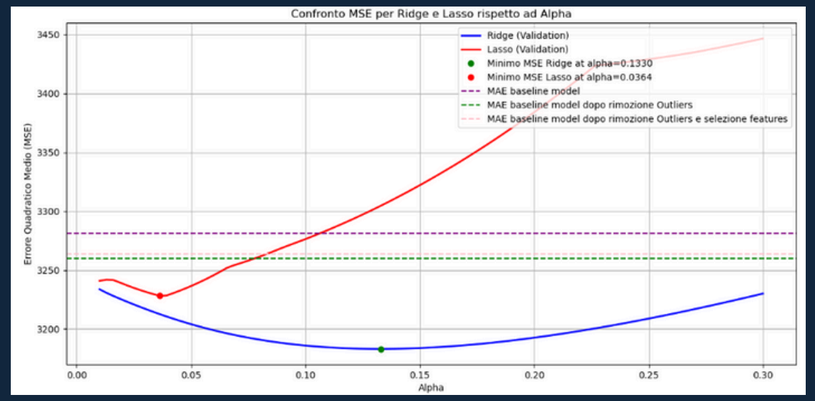

**Rimuovere ulteriormente le feature poco correlate con le labels peggiora le prestazioni.**

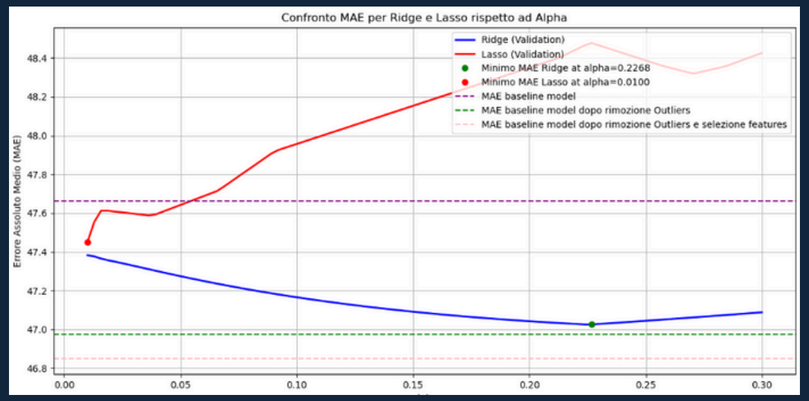
# 7-Scelta del modello: Ridge e Lasso

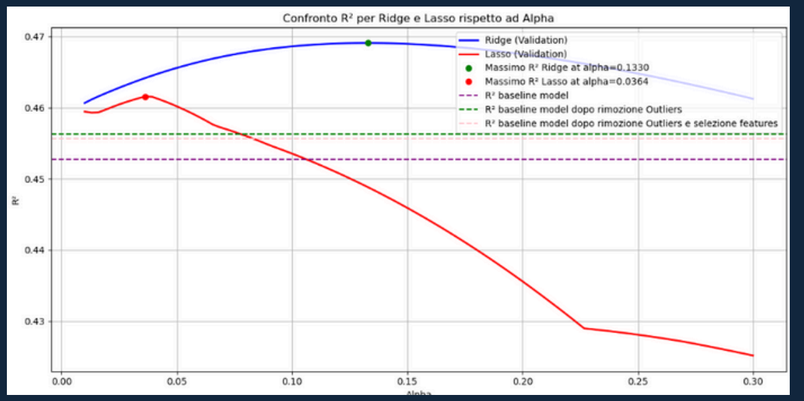
La presenza di correlazioni lineari tra alcune features e le etichette suggerisce che **i modelli lineari potrebbero essere efficaci** per la previsione delle etichette.
Tuttavia, a causa della **multicollinearità nei dati** e dei risultati osservati nel modello baseline, che indicano come la sua riduzione possa migliorare le performance, si è deciso di esplorare due modelli che, grazie alla loro struttura, sono in grado di mitigare l'effetto delle correlazioni tra le variabili: **Lasso e Ridge Regression.**

# 7-Scelta del modello: Ridge e Lasso

Il modello viene  allenato sui dati di train e testato sul set di validazione per valutarne le performance al variare della penalizzazione alpha.




- **Ridge vs Lasso**

**T**ra Ridge e Lasso, Ridge mostra generalmente prestazioni migliori. $R^2$ risulta superiore, il che suggerisce che riesce a spiegare meglio la varianza dei dati rispetto a Lasso. Inoltre, il MSE e il MAE di Ridge sono più bassi, indicandone una migliore capacità di previsione e minore errore assoluto, il che implica una previsione più precisa in media.

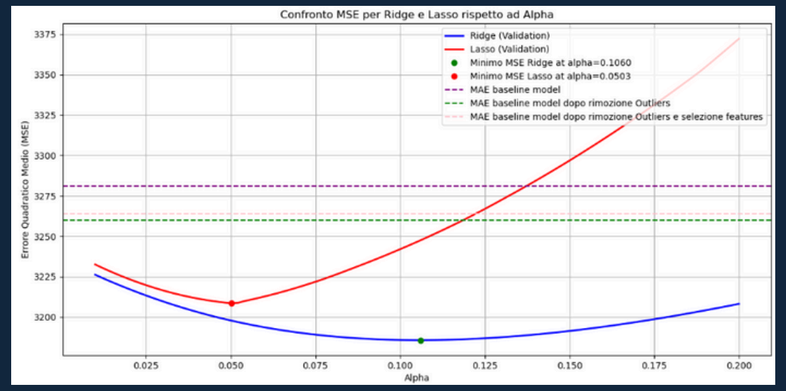
- **Ridge e Lasso vs BLM, BLM senza outlier e BLM senza outlier e dopo riduzione della multicolinearità**

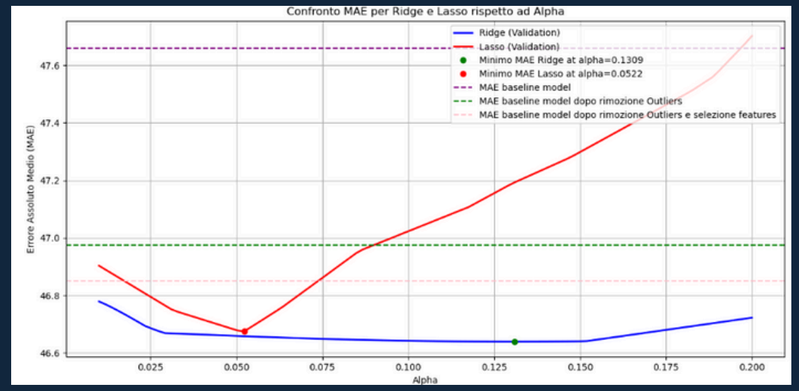
**In** presenza di outliers, sia Ridge che Lasso ottengono performance superiori rispetto a BLM in termini di R2 e MSE. In relazione a MAE la Regressione lineare allenata su un train set senza outliers risulta però migliore. La Regressione lineare allenata su un train set senza outliers potrebbe essere più precisa nel catturare la tendenza complessiva dei dati, ma potrebbe fare previsioni meno accurate su singoli punti.

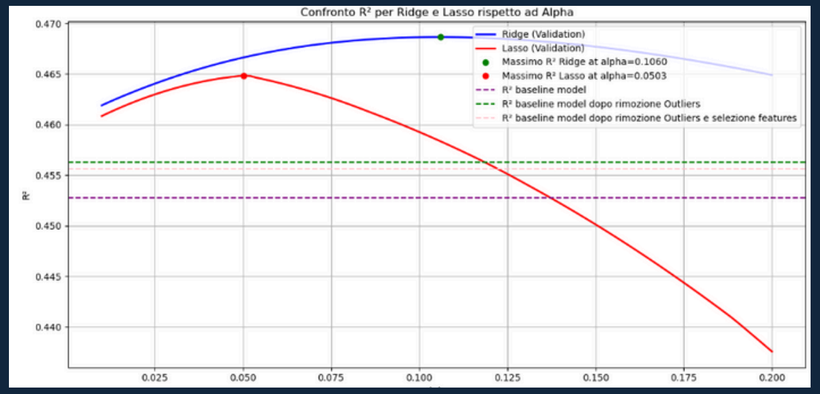
# 8-Ottimizzazione di Ridge e Lasso

## 8.1-Rimozione degli outliers

Poiché, come evidenziato dal modello BLM, gli outliers sembrano influire negativamente sulle predizioni e, dato che Lasso e Ridge da soli non gestiscono efficacemente questi valori (anzi, li amplificano in alcuni casi),  viene addestrato il modello dopo aver rimosso gli outliers dal training.




- **Ridge senza outliers vs Lasso senza outliers**

La differenza tra Ridge e Lasso è minima, ma Ridge continua a dominare leggermente.

- **Ridge e Lasso senza outliers vs BLM, , BLM senza outlier e BLM senza outlier e dopo riduzione della multicolinearità**

Sia Ridge che Lasso senza outliers continuano a performare meglio rispetto agli altri modelli. I miglioramenti si percepiscono in relazione a tutte le metriche.

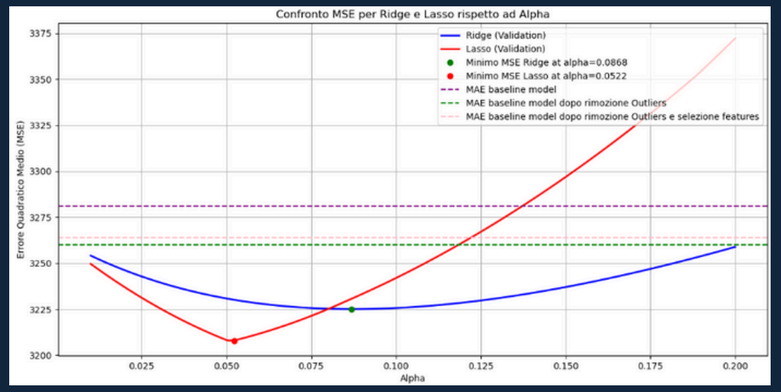
- **2.3 Ridge e Lasso senza outliers vs Ridge e Lasso con outliers**

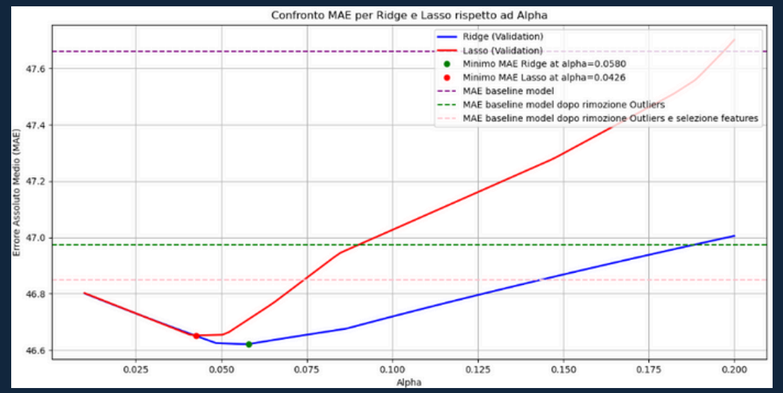
La rimozione degli outliers sembra migliorare entrambi i modelli, si nota esclusivamente un leggero incremento della MSE per Ridge, di appena lo 0.06%. Sicuramente più apprezzabile è il decremento dello 0.9% di MAE.

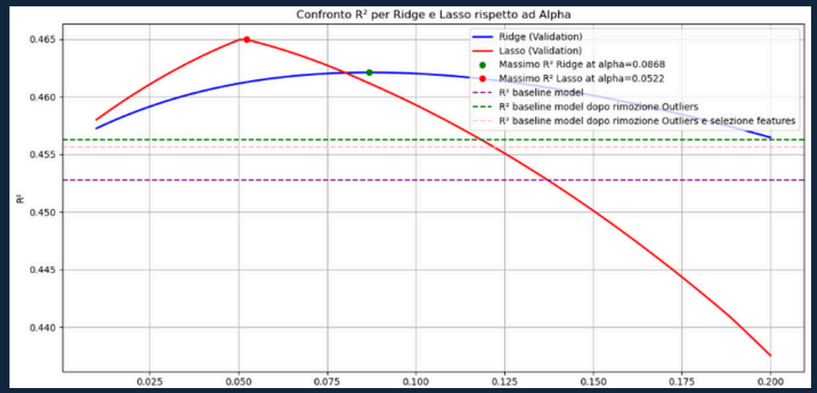
# 8-Ottimizzazione di Ridge e Lasso

## 8.2-Rimozione features correlate tra loro

Si prova a forzare ulteriormente la selezione delle features rimuovendo manualmente le feature correlate s1 e s4 come visto per il baseline model.




- **Ridge senza outliers e con features ridotte vs Lasso senza outliers e con features ridotte**

Con la riduzione delle features, le prestazioni di Ridge e Lasso sono simili, con lievi differenze in MSE e MAE. Il $R^2$ simile indica che entrambi i modelli si adattano in modo analogo ai dati ridotti. Il miglioramento di MSE per Lasso suggerisce che la riduzione delle features può portare a una previsione leggermente più precisa.

- **Ridge e Lasso senza outliers e con features ridotte vs BLM, NoOut e Ridotto_noOut**

L'introduzione della riduzione delle features porta i modelli Ridge e Lasso a mantenere una superiorità rispetto ai modelli precedenti.

- **Ridge e Lasso senza outliers e con features ridotte vs Ridge e Lasso senza outliers**

Il confronto tra Ridge e Lasso con e senza riduzione delle features mostra che la riduzione porta a una leggera perdita di adattamento (riflessa nel calo di $R^2$). La riduzione porta anche a un MSE più elevato per Ridge, suggerendo che la capacità predittiva complessiva diminuisce con meno variabili. Tuttavia, il MAE risulta leggermente inferiore (-0.04%).

# 9- Addestramento del modello

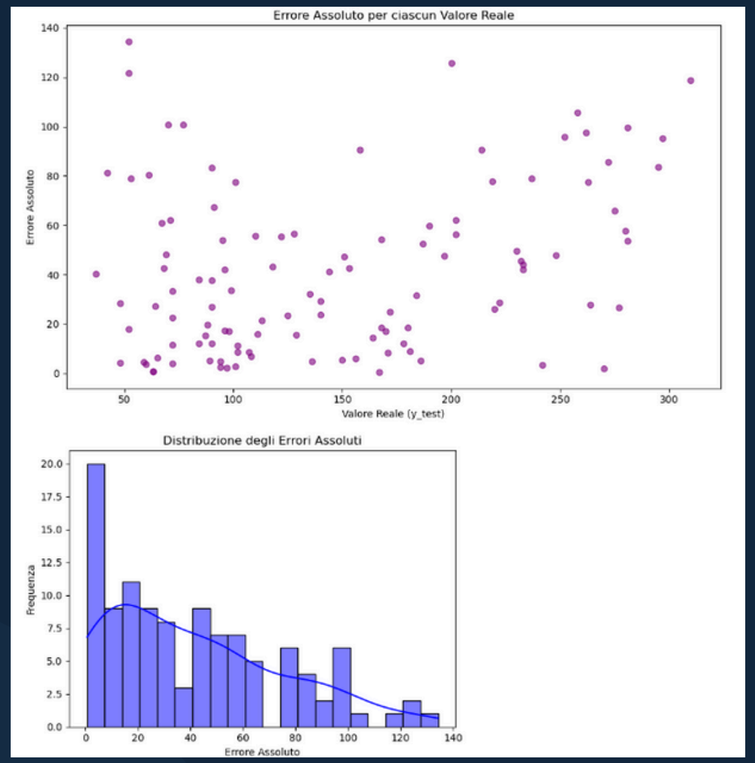
In generale, Ridge addestrato su un train set senza outliers si rivela essere il miglior modello.

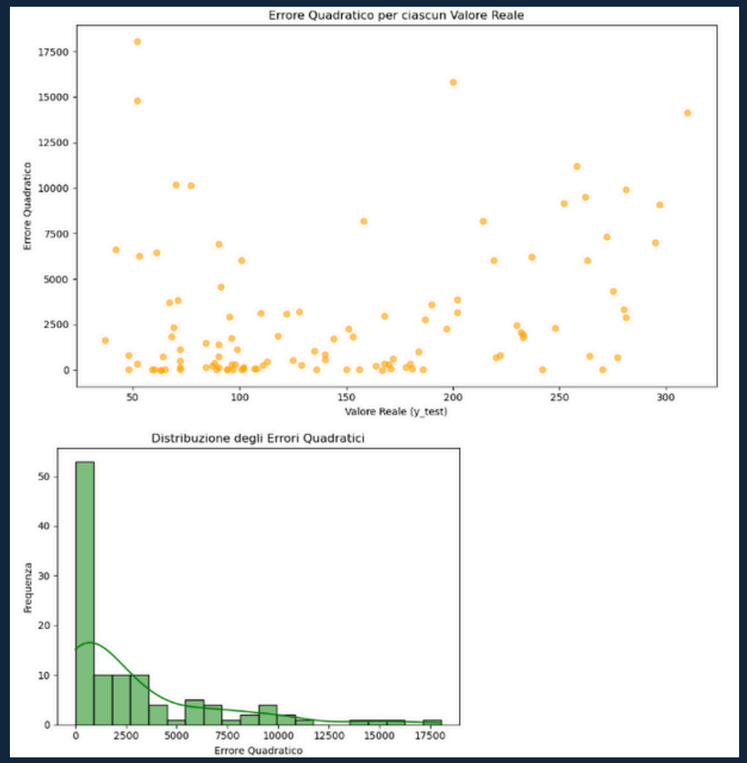
Prima dell'addestramento vero e proprio vengono ottimizzati altri parametri utili per il modello, valutando le seguenti combinazioni:

| | |
|---|---|
| **Positivity** | **False** |
| **Solvers** | **['auto', 'svd', 'cholesky', 'lsqr', 'saga']** |
| **Max_iters** | **[100, 500, 1000, 1500]** |
| **Tol** | **[1e-2, 1e-3, 1e-4, 1e-5]** |
| **Alpha** | **alpha_values = np.linspace(0.01, 0.2, 20)** |

# 10-Risultati

- **Mean Absolute Error (MAE) del modello: 41.29**
- **Mean Squared Error (MSE) del modello: 2819.53**
- **R^2 - Test: 0.49**




- **Non emerge una relazione evidente tra il valore reale e l'entità dell'errore, suggerendo che l'errore è distribuito casualmente e non dipende dai valori del target**

- **La maggior parte degli errori è concentrata nella fascia bassa**

# 11-Conclusioni

**Caratteristiche di y_test:**

- Media: 145.54
- Deviazione standard: 74.7
- Minimo: 37
- Massimo: 310

- Il MAE di 42.19 rappresenta circa 28.4% della media dei valori reali.
- Rispetto alla deviazione standard, un MAE di 41.29 è circa 0.55 volte la deviazione standard, il che indica un errore che può essere considerato relativamente significativo, ma non estremo.
- Il MSE è molto maggiore del MAE, il che suggerisce che ci sono alcuni errori particolarmente grandi che influiscono molto sul valore complessivo.
- Un $R^2$ di 0.49 implica che circa il 51% della variabilità nelle etichette non è spiegato dal modello, il che suggerisce che ci potrebbero essere altri fattori non catturati.

# Link al codice

https://drive.google.com/file/d/1OzTgNdFbwHNhU0APmW15HIvPQRqKDMdU/view?usp=sharing