PREDIZIONE DEL LIVELLO DI PROGRESSIONE DEL DIABETE:

Trainign di un modello ML

PROGETTO FINALE DEL CORSO MACHINE LEARNING

GIULIA COSENZA

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 a un anno di distanza, utilizzando un set di dati medici.

GIULIA COSENZA

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
- **Nessun** valore mancante
- Dati già **scalati e centrati**

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.

Pre-processing dei dati

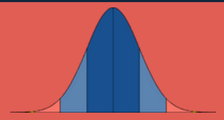
NaN
Missing data

E' stata controllata l'eventuale presenza di missing values.

```
X.isnull().any()

age     False
sex     False
bmi     False
bp      False
s1      False
s2      False
s3      False
s4      False
s5      False
s6      False
dtype: bool
```

Il dataset non contiene valori nulli



Scaling and centering

Quando le variabili hanno scale e distribuzioni diverse , alcune possono dominare sulle altre, e ciò può distorcere i risultati di molti modelli di machine learning. Lo scaling e il centering permettono di ovviare a questo problema.

Nel dataset i dati forniti sono gia stati sottoposti a scaling e centering.

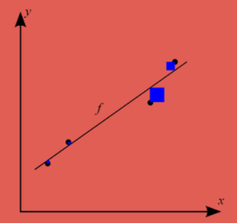
Test_train_split

La funzione train_test_split di scikit-learn è stata utilizzata per suddividere il dataset in due set distinti: per l'addestramento e per il test.
I parametri utilizzati:

- **X: le feature (variabili indipendenti) utilizzate dal modello per fare previsioni;**
- **y: la variabile target (label) che il modello deve predire;**
- **test_size=0.25: indica che il 25% dei dati sarà destinato al test e il restante 75% all'addestramento;**
- **random_state=42: assicura che la divisione dei dati sia riproducibile, fissando il generatore di numeri casuali.**

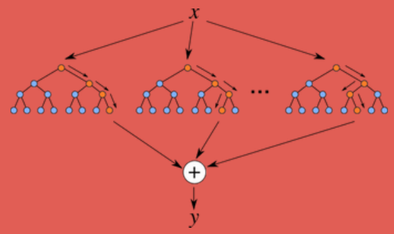
Scelta del modello

Per predire il valore di y nel dataset Diabetes di scikit-learn possiamo utilizzare diversi modelli di regressione. Si utilizza la regressione perchè il target (progressione della malattia) è una variabile continua.



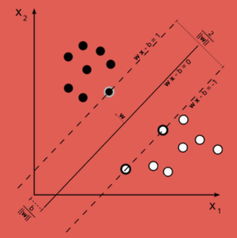
Regressione lineare

Modello di base che cerca di predire la variabile target come una combinazione lineare delle variabili indipendenti.



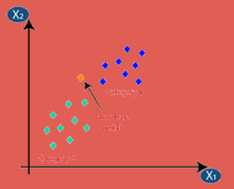
Random Forest Regression

Modello di regressione basato su un ensemble di alberi decisionali. Random Forest è noto per gestire bene la non linearità e l'overfitting



SVR

Il Support Vector Regressor può modellare relazioni non lineari molto complesse.



KNN Regressor

Calcola la media delle etichette dei k vicini più prossimi per fare la previsione.

Scelta del modello

La cross-validazione

La cross-validazione è stata essenziale per la scelta del miglior modello.
Consiste nel suddividere il dataset in più sottoinsiemi (fold), addestrando il modello su alcuni di questi e testandolo su altri, per ottenere una stima più affidabile delle sue performance.

Modelli Testati:

- Regressione lineare:
  - Ridge Regression
  - Lasso Regression
- Random Forest
- SVM
- KNN

Cross-Validazione (5-fold):

- Ogni modello è stato valutato utilizzando 5 fold
- La metrica di valutazione scelta è stata l'Errore Quadratico Medio (MSE)

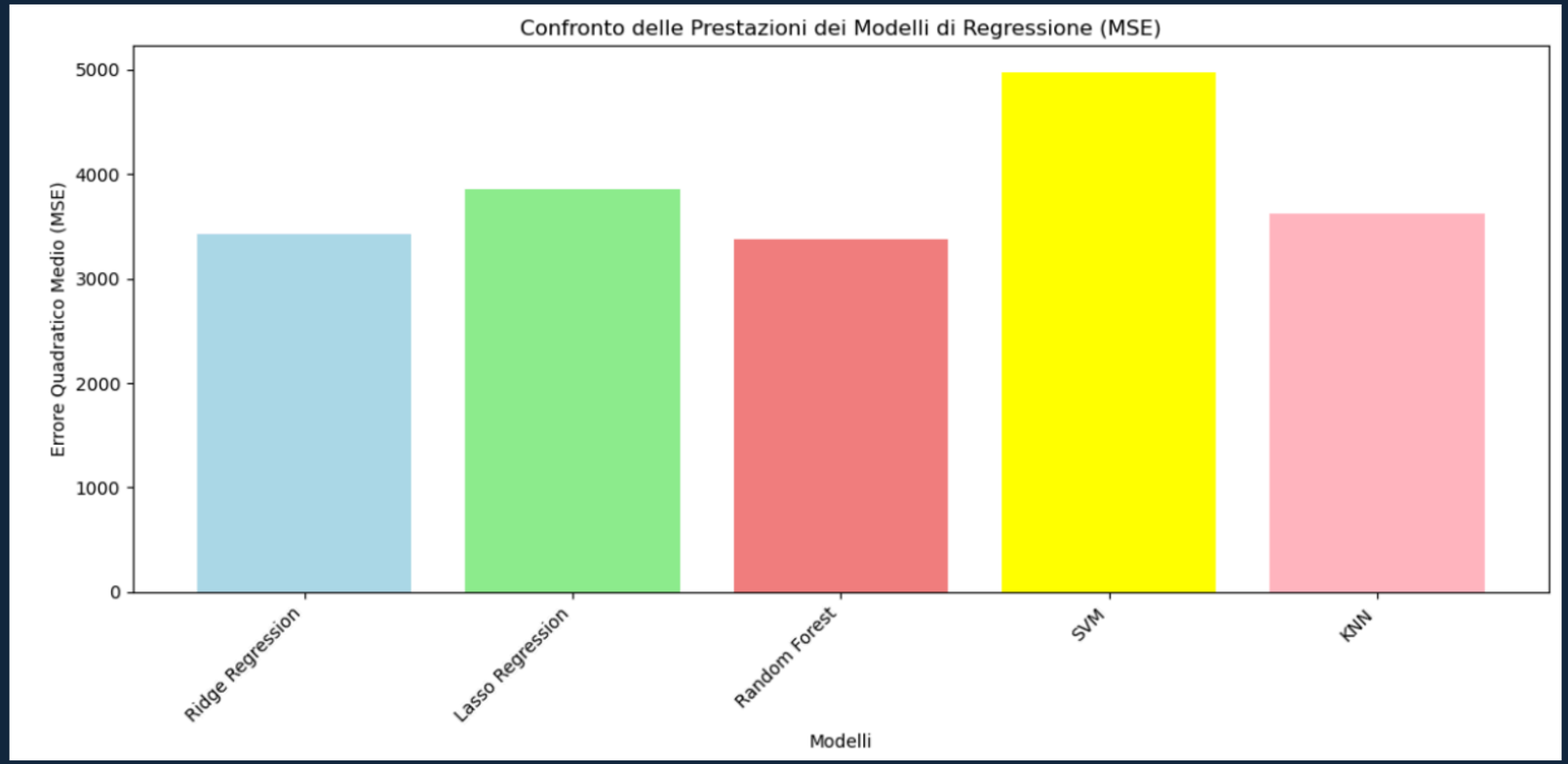
Scelta del Modello:

- I punteggi MSE ottenuti sono stati mediati e confrontati tra i modelli.
- Il modello con il MSE più basso è stato considerato quello con le migliori prestazioni complessive e la maggiore capacità di generalizzazione.

Scelta del modello

La cross-validazione

Sulla base dei risultati ottenuti dalla cross validazione è stato deciso di addestrare sui dati il modello Random Forest Regressor.



Validazione del modello

Metodo

- È stato utilizzato GridSearchCV per eseguire una ricerca esaustiva sugli iperparametri del modello, in particolare su n_estimators (numero di alberi) e max_depth (profondità massima degli alberi).
- È stata applicata una cross-validazione 5-fold per garantire una stima robusta delle prestazioni del modello e prevenire overfitting.

```
rf = RandomForestRegressor(random_state=42)

param_grid = {
    'n_estimators': [50, 100, 200, 300],
    'max_depth': [None, 1, 2, 3, 4, 5, 10, 20],
}
grid_search = GridSearchCV(rf, param_grid, cv=5, scoring='neg_mean_squared_error', verbose = 0)
grid_search.fit(X_train, y_train.values.ravel())
```

Risultato

- La Grid Search ha identificato i parametri ottimali per il modello, minimizzando l'errore quadratico medio (MSE) su diverse partizioni dei dati.

```
# Visualizzo i migliori iperparametri
print(f"Migliori parametri: {grid_search.best_params_}")
```

```
Migliori parametri: {'max_depth': 5, 'n_estimators': 300}
```

Costruzione del modello e valutazione delle sue performance

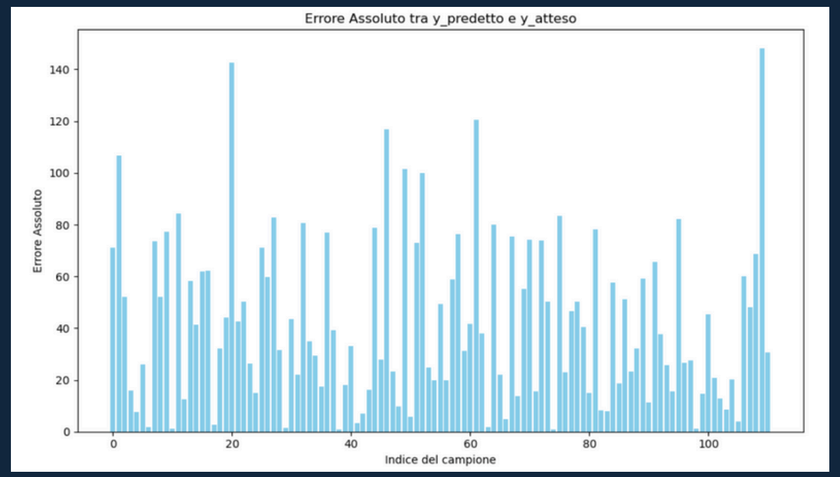
- **Addestramento del Modello:** È stato creato un modello RandomForestRegressor con iperparametri specifici (max_depth=5, n_estimators=300) per limitare l'overfitting e migliorare la stabilità. Il modello è stato addestrato sui dati di addestramento (X_train, y_train) per apprendere le relazioni tra le variabili indipendenti e la variabile target.

- **Generazione delle Previsioni:** Una volta addestrato, il modello ha effettuato previsioni sui dati di test (X_test) utilizzando il metodo predict(), stimando i valori di progressione della malattia.

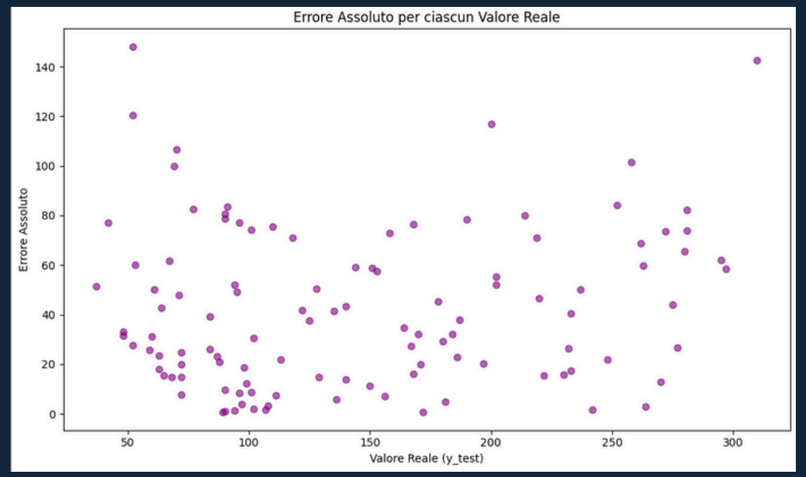
- **Valutazione delle Performance:** È stato utilizzato l'Errore Assoluto Medio (MAE) per valutare la qualità delle previsioni. Questo perché, quando si lavora con un modello di regressione, l'obiettivo è minimizzare l'errore tra le predizioni del modello e i valori reali.

Risultati

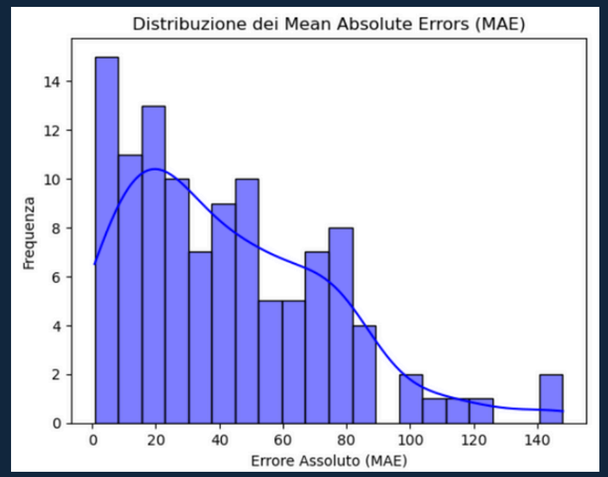
Il MAE ottenuto dal modello è di **42.19**, che rappresenta la distanza media tra le previsioni e i valori reali della progressione della malattia nel dataset Diabetes.



Gli errori sono distribuiti in modo eterogeneo, con picchi significativi che superano i 100 e molti valori più bassi. Indica che il modello può essere accurato per alcune osservazioni ma non generalizza bene su altre, evidenziando la presenza di outlier o variazioni elevate nei dati.



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 (tra 0 e 50), indicando che il modello è mediamente preciso.

GIULIA COSENZA

Interpretazione dei risultati e conclusione

Caratteristiche di y_test:

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

- Il MAE di 42.19 rappresenta circa il 29% della media dei valori (145.54), indicando un errore medio accettabile.
- Rispetto alla deviazione standard (74.7), il MAE appare contenuto, suggerendo che il modello ha catturato bene le variazioni principali, pur avendo alcuni errori più grandi.

Il modello ha un errore medio accettabile, ma c'è spazio per miglioramenti. L'ottimizzazione del modello potrebbe ridurre ulteriormente l'errore, aumentando la precisione delle previsioni.

Link al codice

https://drive.google.com/file/d/1Av7TRa80CgK9zoq0sXexp6_TD3H92qUu/view?usp=sharing