



Computational e Statistical Learning

Comparazione Metodi Supervised



Tabella dei contenuti

01

Struttura dataset

03

**Comparativa metodi
Supervised**

02

**Suddivisione
dataset**

04

Conclusioni





01

Struttura Dataset



The “Boston Housing Dataset”



Storia

Il Boston Housing Dataset è stato originariamente raccolto dalla U.S. Census Bureau e contiene informazioni sul valore delle abitazioni in vari sobborghi di Boston.



Obiettivi

Problemi di regressione: Predire il valore mediano delle abitazioni sulla base delle caratteristiche del quartiere.

Benchmarking di modelli di machine learning: Valutare e confrontare le performance di vari algoritmi di regressione.





506 istanze

14 attributi per ogni sample



Attributi dataset

Activity	Start date
CRIM	Tasso di criminalità per capita per città.
ZN	Proporzione di terreno residenziale suddivisa in lotti superiori a 25.000 piedi quadrati.
INDUS	Proporzione di acri di attività commerciali non al dettaglio per città.
CHAS	Variabile fittizia che è uguale a 1 se il tratto confina con il fiume Charles, altrimenti è 0.
NOX	Concentrazione di ossido nitrico (parti per 10 milioni).

Attributi dataset

Activity	Start date
AGE	Proporzione delle unità occupate dai proprietari costruite prima del 1940.
DIS	Distanza pesata verso cinque centri lavorativi di Boston.
RAD	Indice di accessibilità alle autostrade radiali.
TAX	Aliquota fiscale sulla proprietà per \$10.000.
PTRATIO	Rapporto alunni-insegnanti per città.
B	$1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town

Attributi dataset

Activity	Start date
RM	Numero medio di stanze per abitazione.
LSTAT	Percentuale di popolazione a basso reddito.
MEDV	Valore mediano delle abitazioni, in migliaia di dollari



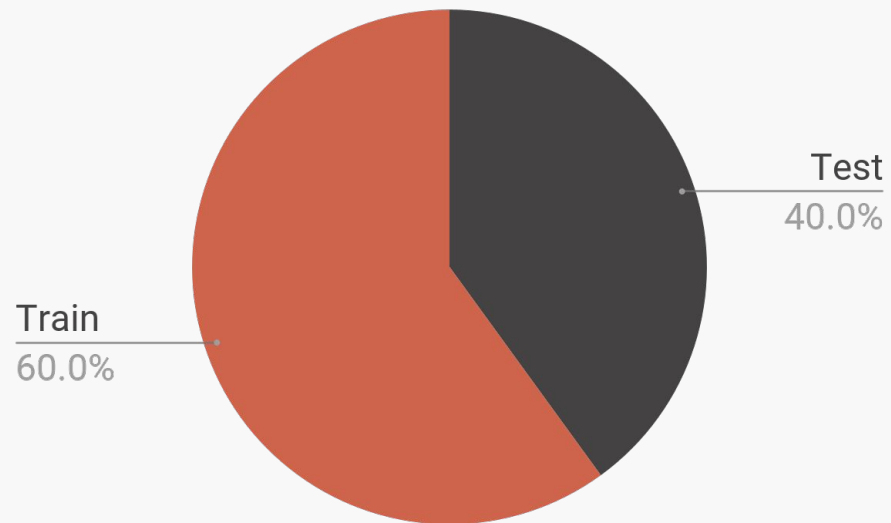
02

Suddivisione dataset



■ Test

■ Train



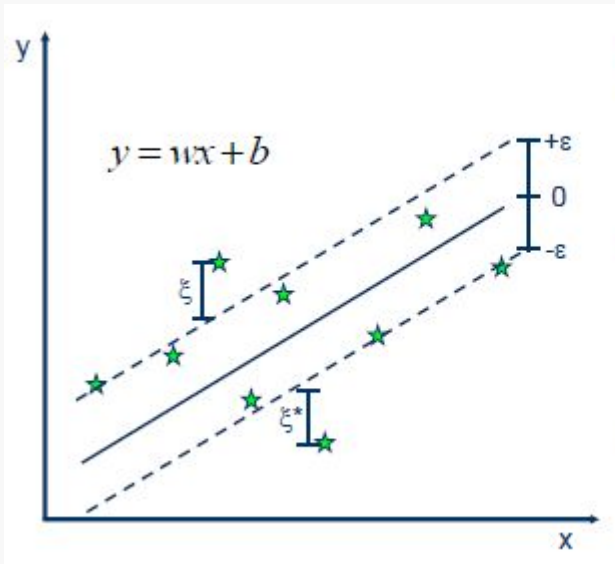


03

Comparativa metodi Supervised



SVR problem



$$\text{Min } \frac{1}{2} \|w\|^2 + c \sum (\xi + \xi^*)$$

Soggetto a

$$Y_i - w x_i - b \leq \epsilon + \xi$$

$$w x_i + b - Y_i \leq \epsilon + \xi^*$$

SVR method

Kernel	C	MSE
Lineare	10	23.69
Lineare	0.1	24.14
Lineare	1	25.21
Polinomiale	100	36.49
Gaussiano	100	38.99
Polinomiale	10	49.32

Applicazione metodo di **GridSearchCV**, fornito dalla libreria Scikit Learn.

Parametri possibili:

- 'kernel': ['linear', 'poly', 'rbf'],
- 'C': [0.1, 1, 10, 100]

Codice Python:

```
svr = SVR(kernel='linear', C=10)
```



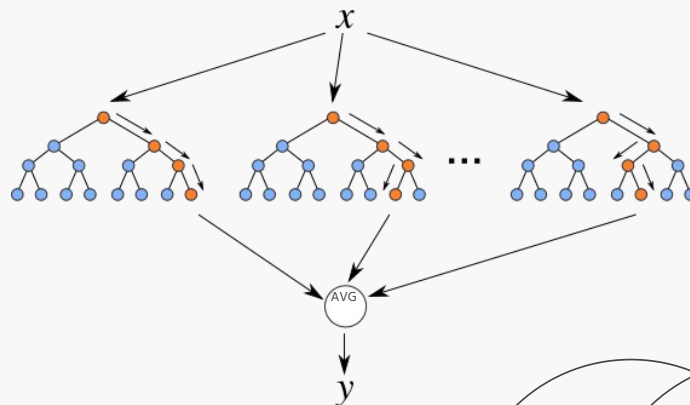
Linear Regression

$$F(x) = \min \frac{1}{n} \sum (y_i - \hat{y}^i)^2$$

- **N** = numero di samples
- **\hat{y}^i** = valore predetto dal modello
- **y_i** = valore osservato

Random Forest Regression

Ensemble method che unisce classificatori deboli per ottenere un classificatore forte.





RFR method

N Estimators	Max depth	MSE
200	10	10.41
300	20	10.43
400	30	10.47
200	30	10.61
400	None	10.66

Applicazione metodo di **GridSearchCV**,
fornito dalla libreria Scikit Learn.

Parametri possibili:

- 'n_estimators': [100,200,300,400],
- 'max_depth': [None, 10, 20, 30]

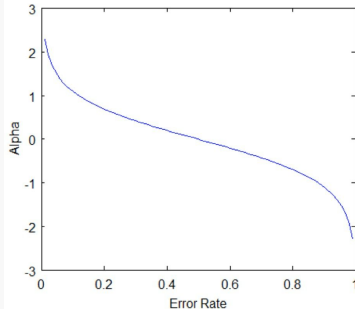
Codice Python:

```
rf =  
RandomForestRegressor(n_estimators=200  
, max_depth = 10)
```


Gradient Boosting Regression

Metodo di Boosting, che ad ogni iterazione aggiorna i pesi (W) in base agli errori, migliorando gli alberi decisionali successivi.

$$\alpha^t = \frac{1}{2} \ln \left(\frac{1 - \epsilon^t}{\epsilon^t} \right)$$



$$W_{t+1} = W_t + e^{-\alpha} y h(x)$$

$$\epsilon(h_j^t) = \sum_{i=1}^n w_i^t \cdot I(h_j^t(x_i) \neq y_i)$$

GBR method

N Estimators	Max depth	MSE
300	10	19.18
200	10	19.46
100	10	19.52
400	10	19.52
100	30	19.59

Applicazione metodo di **GridSearchCV**,
fornito dalla libreria Scikit Learn.

Parametri possibili:

- 'n_estimators': [100,200,300,400],
- 'max_depth': [None, 10, 20, 30]

Codice Python:

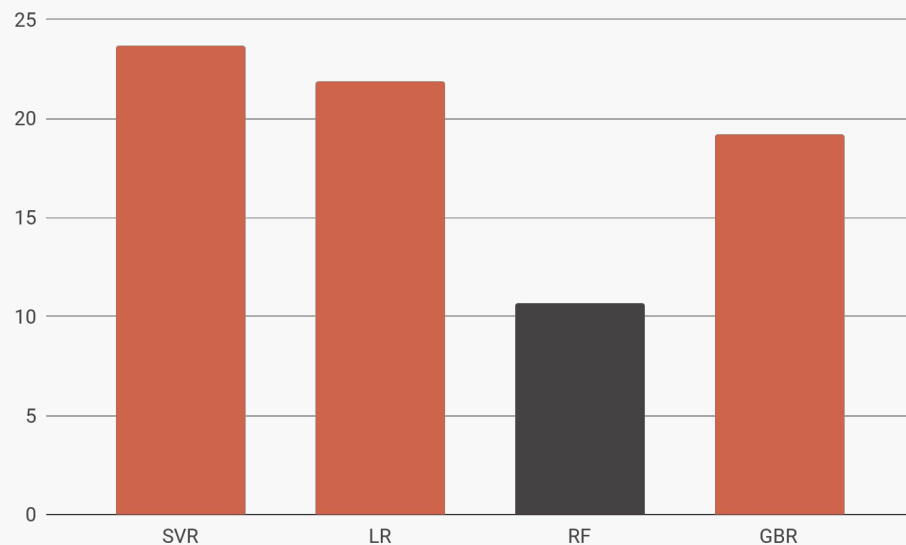
```
gbr =  
GradientBoostingRegressor(n_estimators  
= 300 ,max_depth = 10)
```

Results

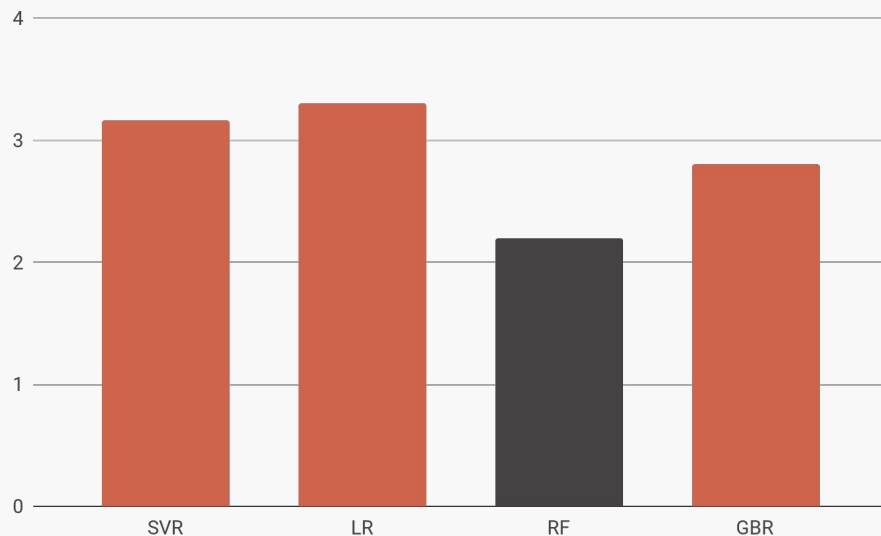
Mean Squared Error (MSE)

Misura statistica utilizzata per valutare la qualità di un modello di regressione.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$



Results



Mean Absolute Error (MAE)

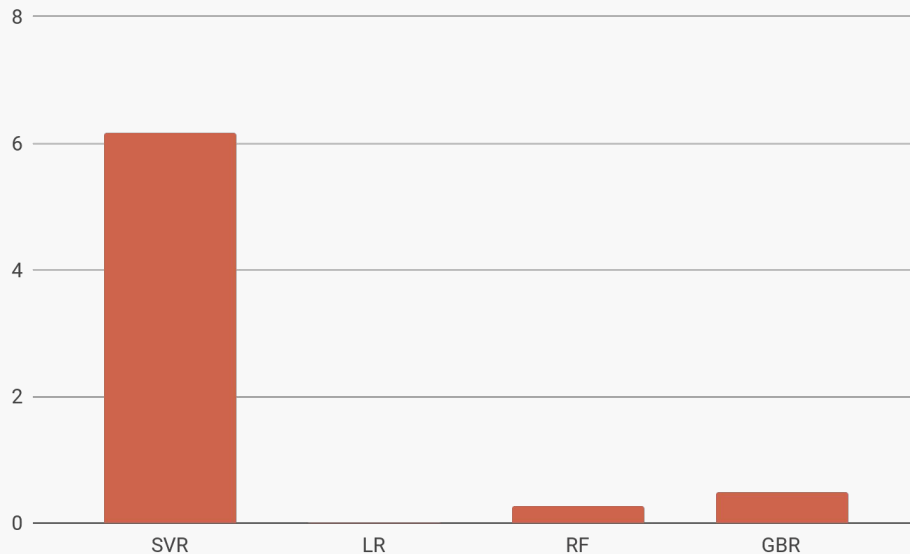
Misura statistica utilizzata per valutare la qualità di un modello di regressione.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - f(x_i)|$$

Tempo esecuzione

Nella seguente tabella sono riportati i tempi medi di esecuzione (in secondi) per vari algoritmi di regressione.

I tempi sono stati misurati utilizzando il modulo **time** di Python.







04

Conclusioni



Random Forest per Regressione

Basandosi sui risultati ottenuti, il Random Forest si è dimostrato il migliore tra gli algoritmi di regressione esaminati. Questo modello ha realizzato delle **previsioni** più coerenti con le “label” date, mantenendo **costi computazionali** ridotti.





**Grazie per
l'attenzione**