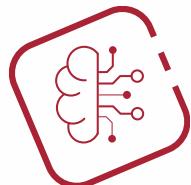




Algoritmi per il Machine Learning

Dott. Antonio Giovanni Lezzi



MACHINE LEARNING

Dott. Antonio Giovanni Lezzi



$(x) \cos(x)$

$$f(x) = 3 \sin(2x) \cos(x)$$

 $n = 4$ 

ALGORITMI DI MACHINE LEARNING

$$\text{lor}(x) = 6x - 42 \cdot \frac{x^3}{3!}$$

- **Regressione multipla**
- Esempio di regressione multipla e algebra lineare
- Errore quadratico medio per regressione multipla
- Indice di bontà per la regressione
- Criterio informativo di AKAIKE
- Regressione multipla in SciKit-Learn

 $x_0 = 0$ $\epsilon = 0.785$



REGRESSIONE LINEARE MULTIPLA

- L'analisi di regressione multipla è un metodo usato per identificare la forza dell'effetto che le variabili indipendenti hanno su una variabile dipendente.
- Capire quanto cambierà la variabile dipendente quando cambiamo le variabili indipendenti permette di **prevedere effetti o impatti dei cambiamenti** delle situazioni reali.
- Ciò non significa che sapremo cosa accadrà, piuttosto avremo un'idea di cosa potrà accadere.



REGRESSIONE LINEARE MULTIPLA

- Per esempio, utilizzando la regressione lineare multipla si può valutare come si modifica la pressione sanguigna al modificarsi dell'indice di massa corporea considerando gli altri fattori costanti (età, sesso, ecc.) e ipotizzare cosa probabilmente può succedere.
- In secondo luogo, è possibile ottenere stime puntuali cercando di **prevedere tendenze e valori futuri**.
- Ad esempio, sarà possibile trovare risposte alla domanda “quale sarà il prezzo dell’oro tra 6 mesi?”
- La regressione lineare multipla permette di ottenere modelli di apprendimento performanti anche nel caso di elevato numero di record da analizzare.



REGRESSIONE LINEARE MULTIPLA

- \hat{y} è la risposta ai valori, ossia rappresenta il risultato previsto dal modello;
- β_0 è l'intercetta, ossia il valore di \hat{y} quando gli x sono tutti uguali a 0;
- β_1 è il coefficiente di X_1 (la prima caratteristica);
- β_n è il coefficiente di X_n (l'ennesima caratteristica);
- x_1, x_2, \dots, x_n sono le variabili indipendenti del modello.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$



REGRESSIONE LINEARE MULTIPLA

- L'equazione spiega la relazione tra una **variabile dipendente continua** (\hat{y}) e due o più **variabili indipendenti** ($x_1, x_2, x_3 \dots$ e così via).
- Supponiamo di stimare l'emissione di CO₂ di un'automobile (variabile dipendente \hat{y}) considerando l'ampiezza del motore, il numero dei cilindri e il consumo di carburante.
- Questi ultimi fattori sono le variabili indipendenti x_1, x_2 e x_3 rispettivamente.
- I beta sono numeri reali e vengono chiamati coefficienti di regressione stimati del modello.



REGRESSIONE LINEARE MULTIPLA

- La \hat{y} è una **variabile dipendente continua**, in quanto essendo la somma di beta, $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots$, risulta essere un numero reale.
- Ne consegue che una variabile di tipo categoriale, che come risultato genererebbe una risposta del tipo “si” / “no”, non può essere rappresentata da un modello di regressione lineare multipla.



ALGORITMI DI MACHINE LEARNING

- Regressione multipla
- **Esempio di regressione multipla e algebra lineare**
- Errore quadratico medio per regressione multipla
- Indice di bontà per la regressione
- Criterio informativo di AKAIKE
- Regressione multipla in SciKit-Learn

ESEMPIO DI REGRESSIONE MULTIPLA E ALGEBRA LINEARE



- Ipotizziamo di voler prevedere l'emissione di CO₂ di un'automobile considerando variabili indipendenti come l'ampiezza del motore, numero di cilindri e consumo di carburante

Campione	Aampiezza motore	Cilindri	Consumo carburante	Emissioni CO2
0	2	4	8,5	196
1	2,4	4	9,6	221
2	1,5	4	5,9	136
3	3,5	6	11,1	255
4	3,5	6	10,6	244
5	3,5	6	10	230
6	3,5	6	10,1	232
7	3,7	6	11,1	255
8	3,7	6	11,6	267
9	2,4	4	9,2	?



ESEMPIO DI REGRESSIONE MULTIPLA E ALGEBRA LINEARE

- L'equazione di regressione è:

$$CO_2em = \beta_0 + \beta_1 * \text{ampiezza motore} + \beta_2 * \text{cilindri} + \beta_3 * \text{consumo carburante}$$

- Matematicamente, si può scrivere l'equazione del modello come:

$$\hat{y} = \beta^T X$$

- Il termine T indica il vettore Beta trasposto, che è pari alla seguente espressione:

$$\beta^T = [\beta_0, \beta_1, \beta_2, \dots]$$



ESEMPIO DI REGRESSIONE MULTIPLA E ALGEBRA LINEARE

- Il vettore X invece è esprimibile nel suddetto modo:

$$X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \dots \end{bmatrix}$$

- Significa che esso è assimilabile al prodotto scalare di due vettori, il **vettore Beta**, detto vettore peso, e il **vettore X**, detto vettore caratteristica.

ESEMPIO DI REGRESSIONE MULTIPLA E ALGEBRA LINEARE

- Nell'esempio proposto X_1 rappresenta l'**ampiezza motore**, mentre X_2 il **numero dei cilindri** e x_3 il **consumo di carburante**.
- Il primo elemento del vettore X è impostato di default pari a 1, per evitare di annullare β_0 .
- Il prodotto del vettore beta trasposto per il vettore caratteristica ci da, per definizione di prodotto matriciale, l'equazione del modello dell'emissione di CO₂, quindi scrivere l'equazione è un modo più compatto per scrivere l'equazione.



ESEMPIO DI REGRESSIONE MULTIPLA E ALGEBRA LINEARE

- Nello spazio monodimensionale la regressione lineare semplice è anch'essa esprimibile attraverso l'equazione $\hat{y} = \beta^T X$, che però di fatto è l'equazione di una retta.
- Nella regressione lineare multipla, invece, quando ci sono più valori di input (o x), l'equazione $\hat{y} = \beta^T X$ crea quello che si chiama **piano** o **iperpiano**.
- Lo scopo della regressione lineare multipla è quello di determinare il miglior iperpiano che viene coperto dai dati, occorre stimare i valori del vettore Beta che meglio predicono i valori del target in ogni riga.
- Per comprendere come stimare tali coefficienti, è bene prima definire l'**errore quadratico medio**.



ALGORITMI DI MACHINE LEARNING

- Regressione multipla
- Esempio di regressione multipla e algebra lineare
- **Errore quadratico medio per regressione multipla**
- Indice di bontà per la regressione
- Criterio informativo di AKAIKE
- Regressione multipla in SciKit-Learn



ERRORE QUADRATICO MEDIO

- Supponiamo di conoscere i valori di Beta, ossia il vettore peso, sostituendo i valori del vettore caratteristica dentro al modello si trova \hat{y} , ossia il valore previsto delle emissioni di CO₂ di una determinata automobile.
- Supponiamo che venga per la prima auto un valore previsto di emissioni pari a 140 g/km (il valore reale è però 196 g/km).

Campione	Aampiezza motore	Cilindri	Consumo carburante	Emissioni CO2
0	2	4	8,5	196



ERRORE QUADRATICO MEDIO

- In questo caso l'errore definito **residuo** che si ricava è pari alla differenza tra il valore reale e il valore previsto:
- $196 - 140 = 56$ errore residuo
- 56 rappresenta l'errore del modello solamente per la prima auto.
- Se eleviamo al quadrato tale errore residuo ed eseguiamo per tutti gli altri campioni la stessa procedura, dividendo poi il risultato per il numero di campioni n, troviamo quello che è definito l'**errore quadratico medio o MSE (Mean squared error)**.



ERRORE QUADRATICO MEDIO

- L'MSE mostra quanto bene o male il modello descrive il set di dati analizzato. La formula per calcolarlo è la seguente:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- Dove i campioni n di auto analizzate sono 9.
- Più l'MSE è elevato e meno il modello rappresenta ciò che stiamo studiando e viceversa. Pertanto è bene cercare di ottenere un errore quadratico medio più piccolo possibile.



ERRORE QUADRATICO MEDIO

- Per raggiungere tale risultato si possono utilizzare i seguenti metodi, che tra l'altro ci permettono di determinare i migliori beta per il nostro modello.
- Metodo dei minimi quadrati
- Metodo della discesa del gradiente



ALGORITMI DI MACHINE LEARNING

- Regressione multipla
- Esempio di regressione multipla e algebra lineare
- Errore quadratico medio per regressione multipla
- **Indice di bontà per la regressione**
- Criterio informativo di AKAIKE
- Regressione multipla in SciKit-Learn



R² RETTIFICATO

- L' aggiunta di predittori a un modello farà aumentare R² anche se le prestazioni del modello non migliorano.
- Una soluzione è l'R² rettificato come misura delle prestazioni del modello, ci sono due variabili aggiuntive: **n** e **k** .
- Il primo rappresenta il numero di punti dati nel modello, mentre il secondo rappresenta il numero di variabili nel modello, escluso il termine costante.

$$R_{adj}^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1}$$

- n = il numero di punti dati nel campione
- k = include il numero di variabili nel modello, escluso il termine costante (l'intercetta)



R² RETTIFICATO

- Se il modello della regressione lineare avesse la forma:

$$\hat{y} = a_0 + a_1 * x_1 + a_2 * x_2$$

- Allora si ha **k = 2**, poiché hai due predittori: **a1** e **a2**.

- Perché R_{adj}^2 è migliore dell' R^2 ? Considerando i seguenti due modelli:

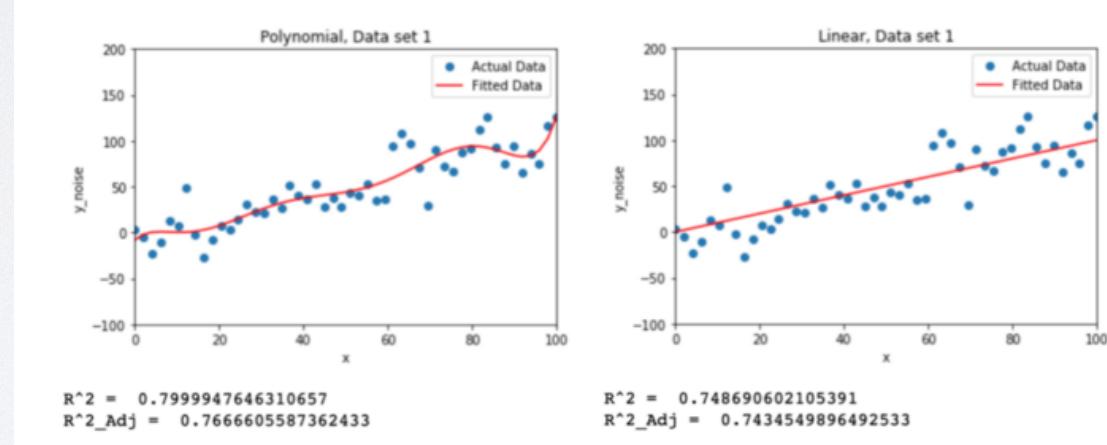
$$\hat{y} = x$$

$$\hat{y} = a_0 + a_1 * x_1 + a_2 * x^{22} + a_3 * x^{33} + a_4 * x^{44} + a_5 * x^{55} + a_6 * x^{66} + a_7 * x^{77}$$



R² RETTIFICATO

- Gli stessi dati, basati su $\mathbf{y} = \mathbf{x} + \mathbf{e}$ sono stati previsti dai modelli, il risultato è mostrato dai grafici
- L'R² del modello di sinistra (che ha più termini) è più alto di quello del modello di destra, il che farebbe pensare che sia un modello migliore.
- Sappiamo che questo non è vero, poiché i dati sono costruiti su $\mathbf{y} = \mathbf{x} + \mathbf{e}$.





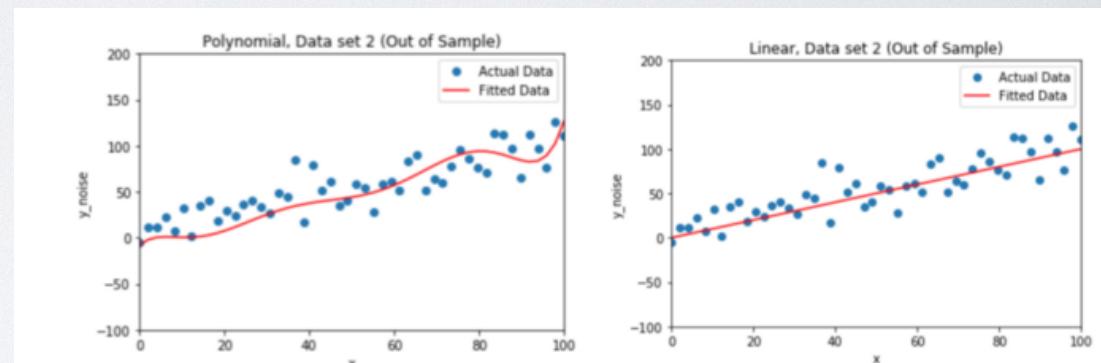
R² RETTIFICATO

- Con i valori di R² rettificato, si nota che quello per il modello più a destra è rimasto più o meno lo stesso, mentre quello del modello più a sinistra è cambiato in modo significativo, mostrando l'impatto che l'aumento del numero di termini può avere sul valore di R² .
- In questo caso particolare, si potrebbe scegliere il modello più a sinistra, poiché anche dopo aver considerato i termini extra, ha un R² aggiustato più alto.
- Sappiamo che questo è falso, potrebbe essere solo il risultato di errori casuali.
- **Sappiamo che un R² più alto non significa che il modello sia migliore!**



R² RETTIFICATO

- Possiamo esaminarlo ulteriormente utilizzando lo stesso modello per adattare un "nuovo set di dati", in modo da rimuovere il "bias di addestramento".
- Vediamo qui che il modello lineare ha un adattamento significativamente migliore di quello del modello polinomiale (a sinistra), con valori R² e R^2_{adj} paragonabili a quelli del dataset precedente.





R² RETTIFICATO

- Sia con R² che con R_{adj}^2 , è buona norma abbozzare i modelli risultanti per controllare visivamente che il risultato abbia senso
- nei casi in cui il risultato non ha senso, l'aggiunta di punti dati aggiuntivi o l'utilizzo di un set di dati di "prova" diverso potrebbe fornire maggiori informazioni.



R² RETTIFICATO

- L'R² corretto migliora l'R² fornendo informazioni sul fatto che il valore R² di un modello sia dovuto a quanto è buono l'adattamento o piuttosto a causa della sua complessità
1. Fornisce più informazioni sulla questione dell'overfitting
 2. Diminuisce l'effetto della casualità sul valore di R² (cioè se è alto a causa della casualità, R² aggiustato lo rifletterà)
 3. Ha ancora gli altri problemi associati a R²



MAE

- Il MAE è la somma di tutte le grandezze di errore divisa per il numero di punti, quindi essenzialmente l'errore medio.
- Pertanto, minore è il MAE, minore sarà l'errore nel tuo modello.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- n = numero di punti
- y = punto effettivo
- \hat{y} = punto previsto



ERRORE QUADRATICO MEDIO (MSE)

- Il MSE è la somma dei **quadrati** di tutti gli errori divisa per il numero di punti.
- Si noti che, poiché in ogni caso l'errore è effettivamente al quadrato, non può essere confrontato direttamente con il MAE, perché sarà sempre di ordine superiore.
- Pertanto, come con MAE, minore è il MSE, minore è l'errore nel modello.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- n = numero di punti
- y = punto effettivo
- y_hat = punto previsto



ROOT MEAN SQUARED ERROR (RMSE)

- RMSE è la radice quadrata del MSE.
- Questa è in un certo senso una metrica più utile e ora poiché sia MAE che RMSE hanno lo stesso "ordine" di errore, possono essere confrontati tra loro.
- Come con MAE e MSE, MSAE inferiore → errore inferiore.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

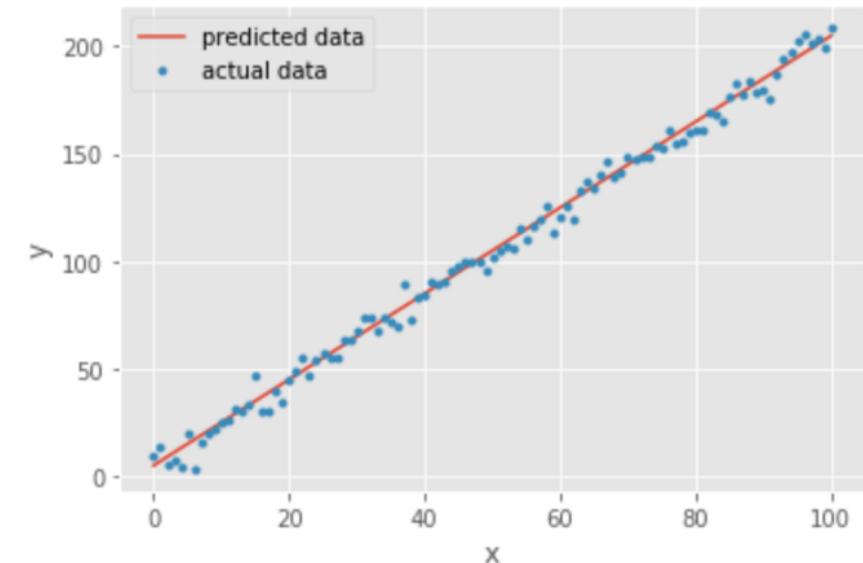
- n = numero di punti
- y = punto effettivo
- y_hat = punto previsto



ALLORA, COM'È QUESTO IN PRATICA?

- Dati due esempi $\hat{y} = 2x + 5$ e una con rumore, quindi $y = 2x + 5 + \epsilon$
- Sia MAE e RMSE sono molto vicini tra loro, indicando entrambi che il modello ha un errore abbastanza basso (ricorda, minore MAE o RMSE, minore è l'errore!).
- Ma allora qual è la differenza tra MAE e RMSE? Perché il MAE è più basso?

MAE = 3.7301886749473785
MSE = 22.527408286222677
RMSE = 4.746304697996398





ALLORA, COM'È QUESTO IN PRATICA?

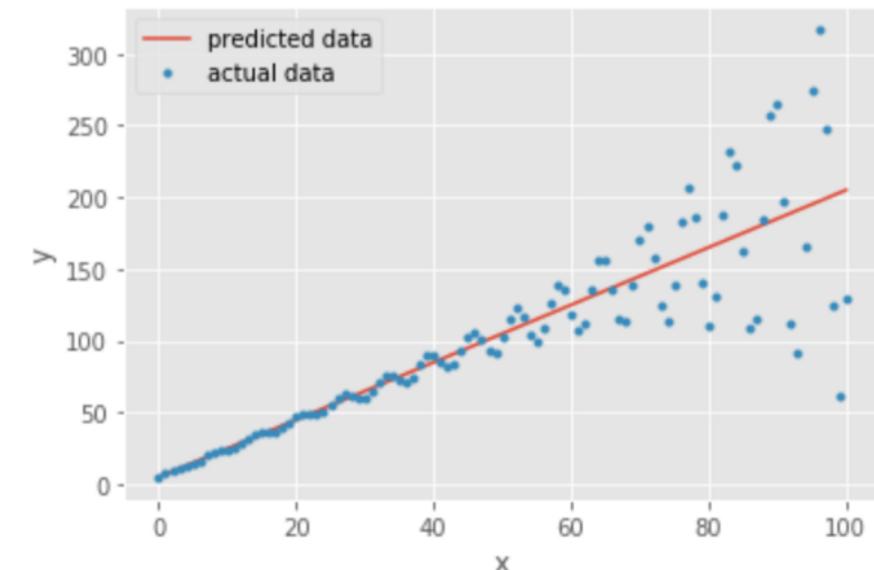
- Quando guardiamo le equazioni per MAE e RMSE, notiamo che RMSE ha un termine al quadrato... quindi: grandi errori saranno al quadrato, e quindi aumenterebbero il valore di RMSE.
- Possiamo concludere che RMSE è migliore nell'acquisizione di grandi errori nei dati, mentre MAE fornisce semplicemente l'errore medio.
- **Poiché l'RMSE somma anche i quadrati prima di prendere una media, sarà sempre intrinsecamente superiore al MAE.**



ALLORA, COM'È QUESTO IN PRATICA?

- La linea arancione rappresenta l'equazione
 $\hat{y} = 2x + 5$
e la 'y' però ha la forma:
 $y = y + \sin(x) * \exp(x / 20) + e$
- dove **exp** () rappresenta la funzione esponenziale (e quindi vediamo un aumento nella deviazione dei punti).
- L'RMSE è quasi il doppio del valore MAE, perché ha catturato la "grandezza" degli errori (in particolare quelli da **x = 80** in poi).

MAE = 19.138201842683475
MSE = 1147.8294224102203
RMSE = 33.87963137949143





ALGORITMI DI MACHINE LEARNING

- Regressione multipla
- Esempio di regressione multipla e algebra lineare
- Errore quadratico medio per regressione multipla
- Indice di bontà per la regressione
- **Criterio informativo di AKAIKE**
- Regressione multipla in SciKit-Learn



CRITERIO INFORMATIVO DI AKAIKE (AIC)

- L'AIC **misura** sia quanto bene i **dati si adattano al modello**, sia quanto sia **complesso**.
- È una miscela di R^2 e R^2 corretto.
- Ciò che fa è **penalizzare** un modello per la sua complessità, ma **premiarlo** per quanto bene si adatta ai dati.
- È valore quasi sempre negativo.

$$AIC = 2 \cdot \ln\left(\frac{e^k}{\hat{L}}\right)$$

k = numero di predittori (compreso il coefficiente!)

L = massima verosimiglianza logaritmica



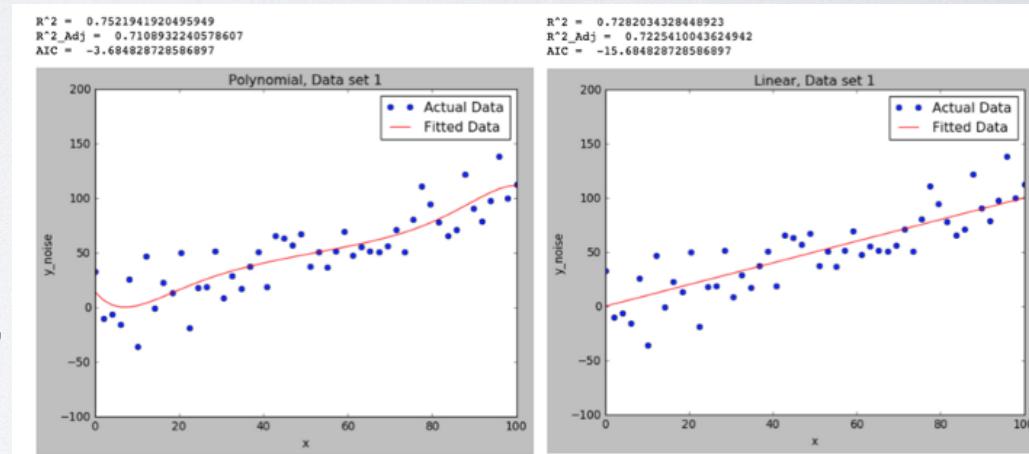
CRITERIO INFORMATIVO DI AKAIKE (AIC)

- In sostanza, più **basso è** l'AIC (cioè più negativo), **migliore è** il modello nel modo in cui si **adatta** ai dati e come **evita l'overfitting**
- Ricordarsi: complessità → overfitting, quindi se l'AIC penalizza la complessità, allora penalizza l'overfitting.



ESEMPIO AIC

- Con l'esempio usato per R^2 aggiustato, dove avevamo un modello incredibilmente complesso per modellare una linea lineare con rumore.
- È stato rieseguito il modello, questa volta aggiungendo anche un punteggio AIC.
- L' R^2 del grafico di sinistra è maggiore di quello di destra, ma **sappiamo** che quello più a destra è corretto.
- Questo è un sintomo di R^2 che si ingrandisce per i modelli più complessi.



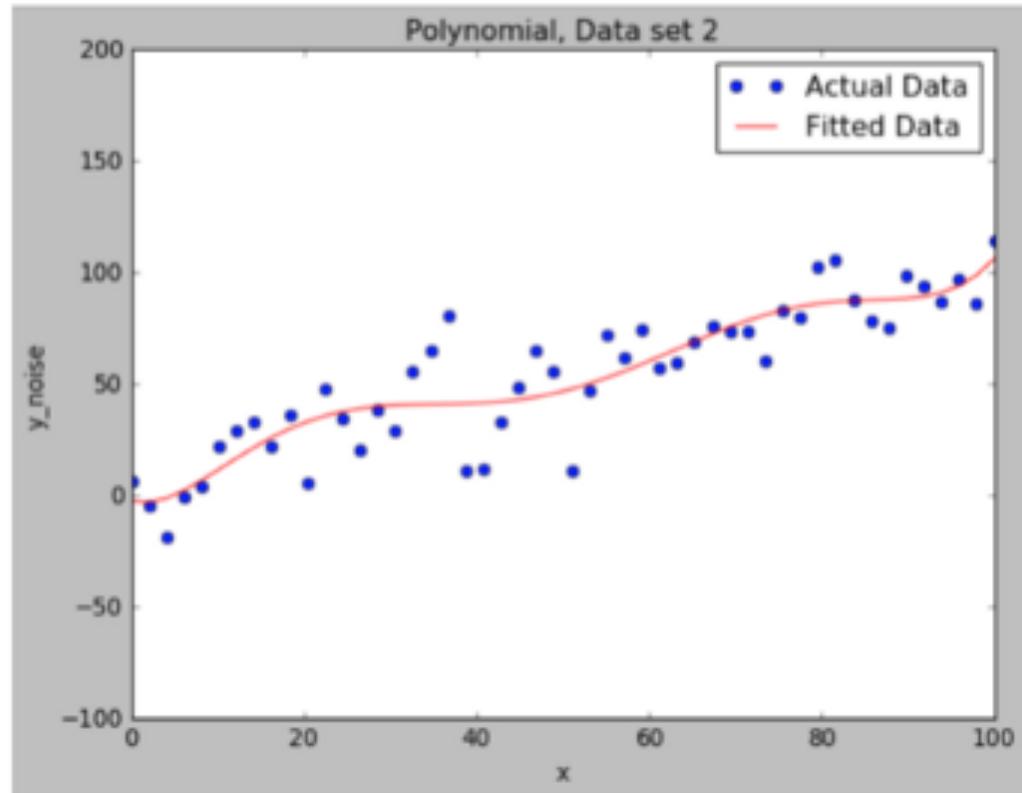


CRITERIO INFORMATIVO DI AKAIKE (AIC)

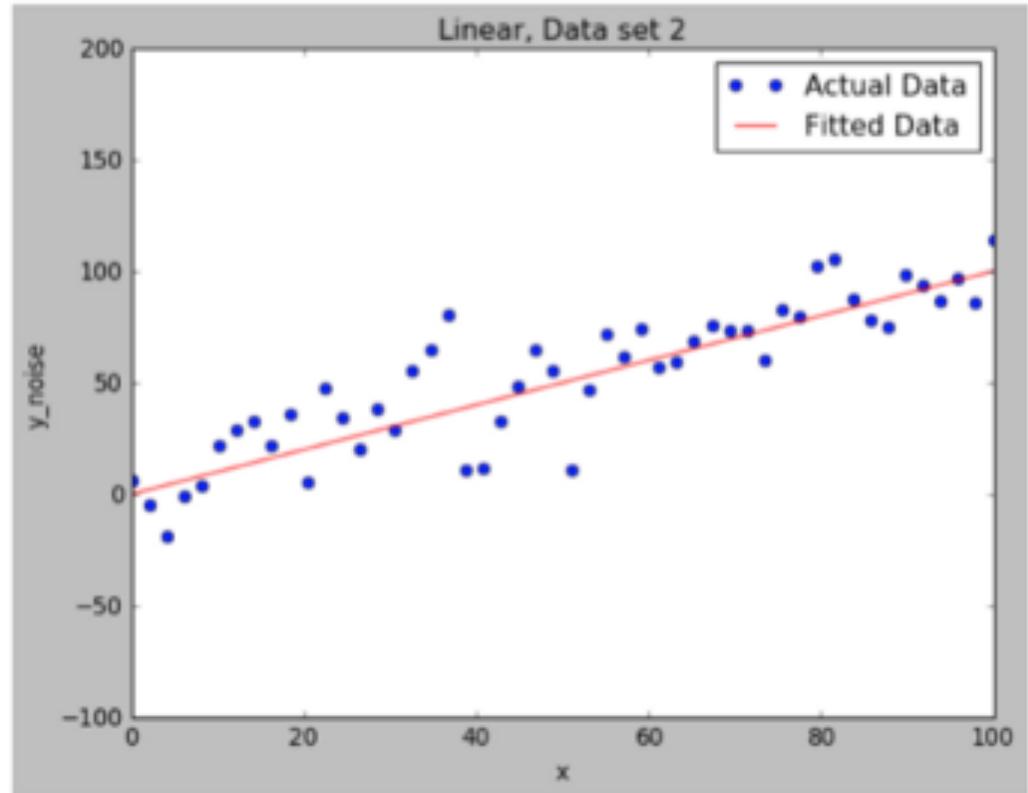
- Più l'AIC è negativo, migliore è la vestibilità e la mancanza di overfit.
- Quindi dal solo parametro AIC, possiamo concludere che il modello più semplice è migliore (ricordarsi di abbozzare sempre i grafici e a ragionarci sopra, non fidarsi solo dei numeri!).



R^2 = 0.7984722822654993
R^2_Adj = 0.7648843293097491
AIC = -2.894266214575687



R^2 = 0.7640874077612184
R^2_Adj = 0.7591725620895772
AIC = -14.894266214575687





RIEPILOGO AIC

- Più basso è l'AIC, migliore è il modello in termini di vestibilità e di evitare l'eccessivo adattamento.
- L'AIC è un buon indicatore della qualità del modello, poiché tiene conto sia della vestibilità, ma anche di quanto poco il modello vada bene
- Matematicamente, AIC è valido solo per un set di dati infinito. A livello computazionale, l'errore può essere compensato avendo una dimensione del campione molto grande. Per campioni più piccoli, è necessario aggiungere un fattore di correzione.



RIEPILOGO AIC

```
from sklearn import linear_model  
reg = linear_model.LassoLarsIC(criterion='aic')  
reg.fit([[-1, 1], [0, 0], [1, 1]], [-1.1111, 0, -1.1111])  
print(reg.coef_)
```



MATLAB Model	Mathematical expression	R^2	R_a^2	$MSEc$	AIC
linear	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$	0.4539	0.4517	0.0342	-266.1359
interactions	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$	0.4547	0.4514	0.0342	-264.9174
purequadratic	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$	0.4550	0.4506	0.0342	-263.1331
quadratic	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$	0.4559	0.4504	0.0343	-261.9933
poly13	$Y = \beta_0 + \beta_1 X_1 + \sum_{i=1}^3 \beta_{i+1} X_2^i + \sum_{i=1}^3 \beta_{i+4} X_1 X_2^i$	0.4577	0.4511	0.0342	-261.6155
poly31	$Y = \beta_0 + \beta_1 X_2 + \sum_{i=1}^3 \beta_{i+1} X_1^i + \sum_{i=1}^3 \beta_{i+4} X_2 X_1^i$	0.4599	0.4533	0.0341	-263.6400



ALGORITMI DI MACHINE LEARNING

- Regressione multipla
- Esempio di regressione multipla e algebra lineare
- Errore quadratico medio per regressione multipla
- Indice di bontà per la regressione
- Confronto fra gli indici
- Criterio informativo di AKAIKE
- **Analisi dei residui in SciKit-Learn**



SCELTA DEL DATASET

- Il set di dati contiene le informazioni meteorologiche di base come temperatura, velocità del vento, pressione e condizioni meteorologiche.
- L'obiettivo del problema è tentare di capire se esiste una relazione di tipo lineare tra umidità e temperatura, dove la **temperatura** è la variabile dipendente (y) mentre **l'umidità** è la variabile indipendente (X).
- Per verificare l'esistenza di questa relazione o meno creiamo un **modello di regressione** per prevedere i dati di temperatura dall'umidità.



IMPORTARE LE LIBRERIE

- Oltre a importare pandas, numpy e matplotlib importiamo seaborn, una libreria di visualizzazione dei dati. Inoltre importiamo alcune importanti classi e moduli di Sklearn:
- il **LabelEncoder**, per codificare i valori stringa in campi numerici;
- le **metriche** (R2, il coefficiente di correlazione) per valutare il modello di regressione lineare;
- il **Train_test_split** per suddividere la fase di addestramento con quella di test;
- il **linear_model**, il modulo che include i vari tipi di regressione.

```
7 import pandas as pd
8 import numpy as np
9 import matplotlib.pyplot as plt #Data visualisation libraries
10 import seaborn as sns
11 from sklearn.preprocessing import LabelEncoder
12 from sklearn.metrics import r2_score
13 from sklearn.model_selection import train_test_split
14 from sklearn import linear_model
15
16
```



ANALIZZIAMO IL DATASET

- Quando si eseguono analisi su dataset è sempre opportuno verificare e preparare i dati per eliminare righe vuote o colonne prive di dati.
- Guardando bene al dataset, si può notare che alcune colonne non contengono dati, come Loud Cover, e dovrebbero essere rimosse dal set di dati, tuttavia, poiché vengono usati solo dati di umidità e temperatura, la colonna non verrà eliminata.

Formatted Date	False
Summary	False
Precip Type	True
Temperature (C)	False
Apparent Temperature (C)	False
Humidity	False
Wind Speed (km/h)	False
Wind Bearing (degrees)	False
Visibility (km)	False
Loud Cover	False
Pressure (millibars)	False
Daily Summary	False
dtype: bool	



ANALIZZIAMO IL DATASET

- Verifichiamo ora la presenza di celle nulle e se una riga qualsiasi contiene null, elimineremo la riga. Per farlo eseguiamo la seguente riga di codice:

Tempo.isna().any()

- Otteniamo il seguente risultato, che ci indica che la colonna delle precipitazioni (Precip Type) ha righe nulle, dal valore True ottenuto di fianco
- Pertanto provvediamo ad eliminare tali righe attraverso la **funzione dropna()**

Formatted Date	False
Summary	False
Precip Type	True
Temperature (C)	False
Apparent Temperature (C)	False
Humidity	False
Wind Speed (km/h)	False
Wind Bearing (degrees)	False
Visibility (km)	False
Loud Cover	False
Pressure (millibars)	False
Daily Summary	False
dtype: bool	



CREIAMO LA MAPPA DI CALORE

- Prima di creare il modello di regressione, è buona norma verificare la correlazione tra le variabili, magari facendoci aiutare da una mappa di calore che mostra le varie correlazioni come mostrato di seguito.
- In questo script prima di tutto ci copiamo di dati nella variabile `modeling_data`. E in questo dataset copiato eliminiamo le colonne “Daily Summary” e “Loud Cover”.
- Dopodichè trasformiamo, attraverso la **classe Label Encoder**, le colonne “Summary” e “Precip Type” da variabili stringa a variabili numeriche in modo da facilitare i calcoli al modello.

```
28
29 modeling_data=tempo.copy()
30 modeling_data=modeling_data.drop(['Daily Summary','Loud Cover'], axis=1)
31 le = LabelEncoder()
32 modeling_data['Summary']=le.fit(modeling_data['Summary']).transform
33 (modeling_data['Summary'])
34 le2 = LabelEncoder()
35 modeling_data['Precip Type']=le2.fit(modeling_data['Precip Type']).transform
36 (modeling_data['Precip Type'])
37
```



CREIAMO LA MAPPA DI CALORE

- Il Label Encoder ci aiuterà a convertire questo tipo di dati di testo categoriali in dati numerici comprensibili al modello: tutto ciò che dobbiamo fare, per etichettare le colonne, è importare la classe LabelEncoder dalla libreria sklearn, adattare e trasformare le colonne, e quindi sostituire i dati di testo esistenti con i nuovi dati codificati.
- Siamo pronti ora a creare la mappa di calore.
(Script qui a fianco)

```
38
39 corr = modeling_data.corr()
40 mask = np.zeros_like(corr, dtype=np.bool)
41 mask[np.triu_indices_from(mask)] = True
42 f, ax = plt.subplots(figsize=(11, 9))
43 cmap = sns.diverging_palette(0, 150, as_cmap=True)
44 sns.heatmap(corr, mask=mask, cmap=cmap, center=0,
45             square=True, linewidths=.5, cbar_kws={"shrink": .5})
46
```



CREIAMO LA MAPPA DI CALORE

- Come possiamo vedere, viene utilizzato il dataset copiato **modeling_data**. Esso ci crea l'array corr. Successivamente, viene creato un array di zeri con la stessa dimensione di quello appena creato, come prevede la **funzione zeros_like** di numpy, mentre con la **funzione triu_indices_from** troviamo gli indici dell'array del triangolo in alto necessari a creare la mappa di calore.
- Anche le altre righe di codice servono per impostare la mappa di calore. Grazie alle librerie **matplotlib** e **seaborn**, si definisce la struttura e la forma del grafico (per maggiori informazioni si veda [questo link](#)).



CREIAMO LA MAPPA DI CALORE

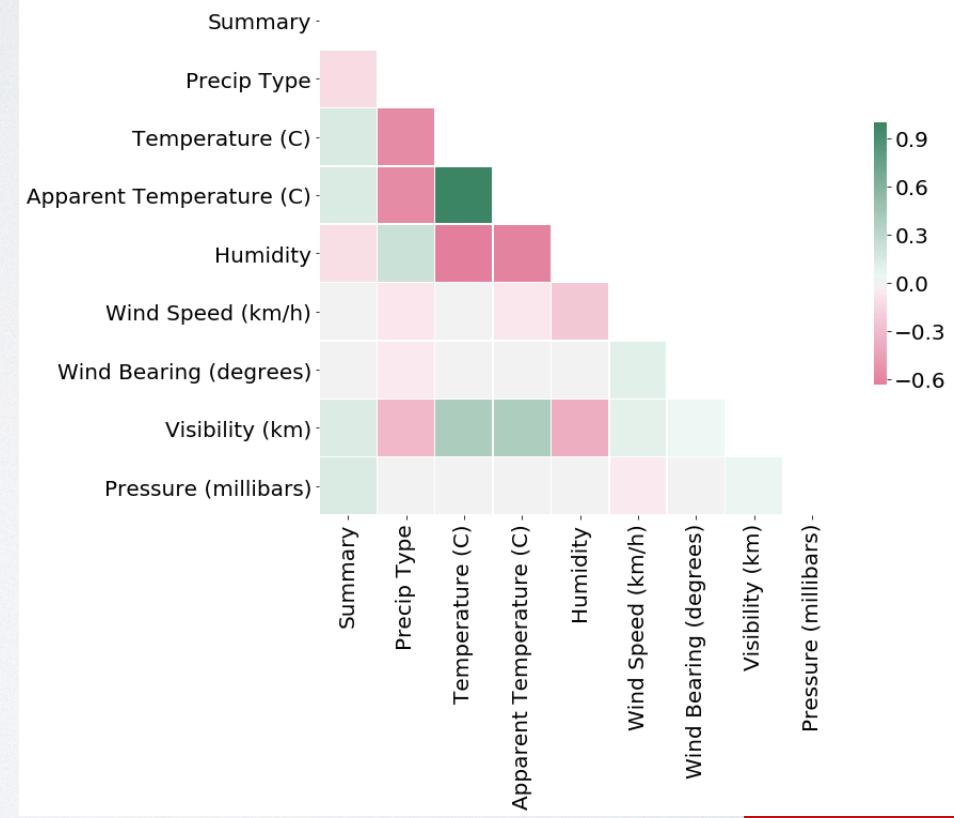
- Come risultato otteniamo il seguente grafico. Sia sull'asse x che sull'asse y abbiamo i campi del dataset copiato (`modeling_data`). I colori a forma di quadrato che si formano ci indicano il valore di correlazione tra le variabili in essere. In particolare:
 - più ci si avvicina al verde più le variabili sono correlate tra loro,
 - variabili con correlazione nulla avranno colore quasi vicino al bianco;
 - variabili con correlazione inversa avranno colore tendente al rosa/porpora.



CREIAMO LA MAPPA DI CALORE

Guardando attentamente il grafico possiamo notare che:

- I dati di temperatura e temperatura apparente hanno dati simili poiché la correlazione tra loro è elevata;
- I dati di pressione hanno una correlazione molto scarsa con i dati di temperatura e ciò può essere dovuto all'esistenza di righe con valore di pressione pari a zero;
- La temperatura e l'umidità hanno una correlazione inversa.





TEST TRAIN DATA

- Prima di creare il modello cancelliamo anche le colonne “Apparent Temperature”, “Formatted Date” e “Summary” dal dataset.
- E consideriamo solo i valori della colonna “Humidity” che sono maggiori di zero.

```
47
48 modeling_data=modeling_data.drop(['Apparent Temperature (C)',  
49           'Formatted Date','Summary'],axis=1)  
50 modeling_data=modeling_data[modeling_data['Humidity']>0]  
51
```



TEST TRAIN DATA

- Una volta fatto ciò, possiamo provvedere a suddividere il dataset in set di dati di test e di allenamento.
- In questo caso consideriamo il 33% del set di dati come dato di test, mentre il 66% sarà il set di dati di addestramento.

```
53  
54 X_train, X_test, y_train, y_test = train_test_split (modeling_data  
55             ['Humidity'], modeling_data['Temperature (C)'],  
56             test_size=0.33, random_state=42)  
57
```



CREIAMO IL MODELLO DI REGRESSIONE

- Assegniamo alla variabile reg il valore del modello di regressione.
Eseguiamo poi l'adattamento del modello di regressione ai dati.
- Col termine reg.coef invece stimiamo i coefficienti del problema di regressione.

```
58 reg = linear_model.LinearRegression(copy_X=True,
59     fit_intercept=True, n_jobs=None, normalize=False)
60 reg.fit(X_train.values.reshape(-1, 1),y_train.values.reshape(-1, 1))
61 reg.coef_
62
63
64 print ('Coefficiente di determinazione del dataset di addestramento:'
65     + str(reg.score(X_train.values.reshape(-1, 1),
66                 y_train.values.reshape(-1, 1))))
67
68 print ('Coefficiente di determinazione del dataset di test:'
69     + str(reg.score(X_test.values.reshape(-1, 1),
70                 y_test.values.reshape(-1, 1))))
71
```



CREIAMO IL MODELLO DI REGRESSIONE

- Vediamo poi stampati dalle successive righe di codice il coefficiente di determinazione R2 (la prima mi dice il coefficiente di determinazione relativo al dataset di addestramento, mentre la seconda il coefficiente previsto del dataset di test).

```
Coefficiente di determinazione del dataset di addestramento:0.40504986225298323  
Coefficiente di determinazione del dataset di test:0.4033434163300134
```

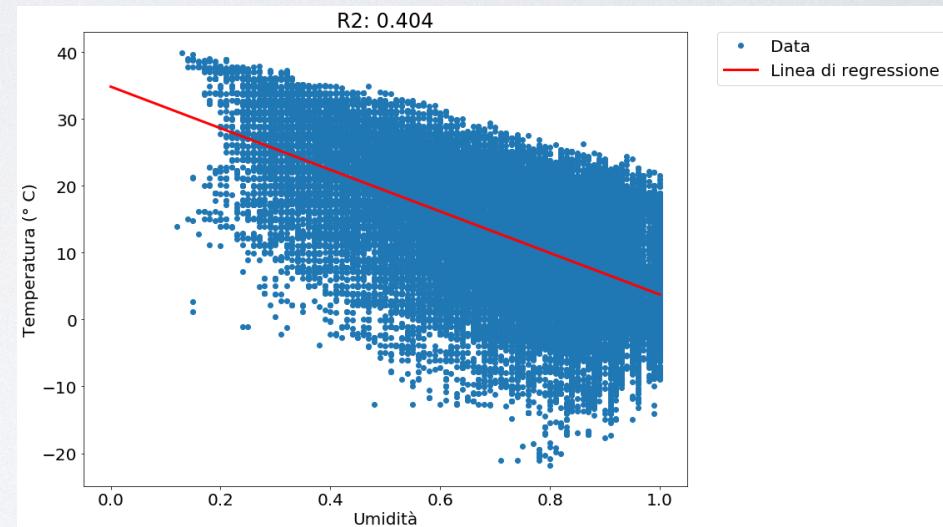


```
72
73 font = {'size' : 20}
74 plt.rc('font', **font)
75 plt.figure(figsize=(13,10))
76 plt.plot(modeling_data['Humidity'],
77           modeling_data['Temperature (C)'],
78           'o',label='Data')
79 I=np.linspace(np.floor(min(modeling_data['Humidity']))*0.95,
80                 np.ceil(max(modeling_data['Humidity']))*0.11),50)
81 plt.plot(I,reg.predict(I.reshape(-1, 1)),color='r',
82           linewidth=3,label='Linea di regressione')
83 plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
84 plt.xlabel('Umidità');
85 plt.ylabel('Temperatura (° C)')
86 Preds=reg.predict( modeling_data['Humidity'].values.reshape(-1, 1))
87 R2=r2_score(modeling_data['Temperature (C)'],Preds )
88 plt.title('R2: '+ str(np.round(R2,decimals=3)))
89 plt.show()
90
```



CREIAMO IL MODELLO DI REGRESSIONE

- Vengono impostati i caratteri, le etichette, i titoli e le varie dimensioni del grafico ed otteniamo il seguente risultato (dove i dati del modello sono mostrati dai pallini blu, mentre la retta di regressione dalla linea rossa):





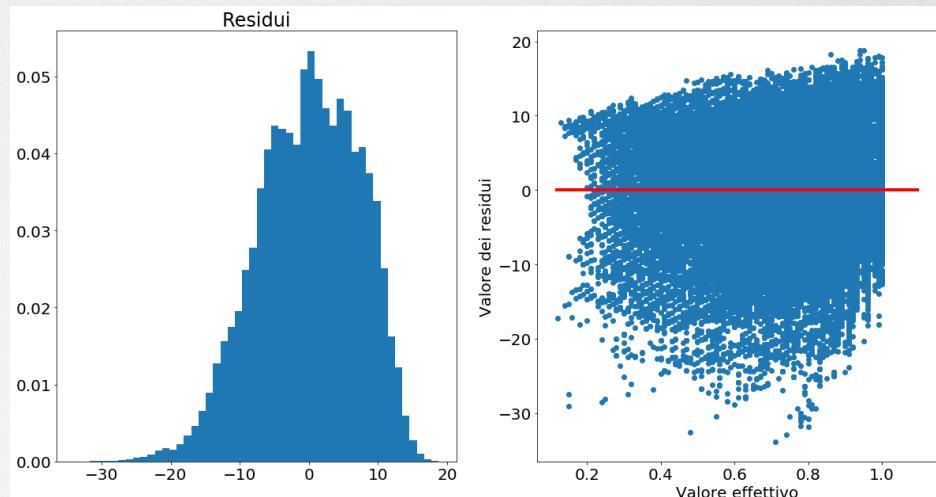
CREIAMO IL MODELLO DI REGRESSIONE

- Possiamo notare che il valore che otteniamo è un basso valore del coefficiente di correlazione ($R^2 = 0,404$). Da ciò si può dedurre che l'umidità influenza poco la variabile temperatura.
- Infine se volessimo calcolare il valore dei residui, ossia una stima osservabile dell'errore statistico, si può eseguire il seguente codice:

```
92 Residuals=modeling_data['Temperature (C)'].values.reshape(-1, 1)-Preds
93 font = {'size' : 20}
94 plt.rc('font', **font)
95 fig, ax = plt.subplots(1,2,figsize=(20,10))
96 num_bins = 50
97 n, bins, patches = ax[0].hist(Residuals, num_bins, density=1)
98 ax[0].title.set_text('Residui');
99
100
101 ax[1].plot(modeling_data['Humidity'],Residuals,'o')
102 ax[1].set(xlabel='Valore effettivo', ylabel='Valore dei residui')
103 ax[1].hlines(0, np.min(modeling_data['Humidity'])*0.95,
104   np.max(modeling_data['Humidity'])*1.1, colors='r',
105   linestyles='solid',zorder=10, linewidth=4 )
106 plt.show()
107
```

CREIAMO IL MODELLO DI REGRESSIONE

- Anche in questo caso vengono definite le caratteristiche dei grafici: il primo rappresenta la distribuzione dei residui per la temperatura mentre il secondo il grafico dei residui per quanto riguarda l'umidità.
- Tale script da come output il seguente grafico:





CONCLUSIONE

- La regressione è un metodo molto comune per predire una variabile conoscendone la relazione con altre, ed è molto utile per poter prevedere effetti o impatti nei cambiamenti delle situazioni reali.
- Per il dataset analizzato, il modello di regressione creato non è un buon predittore siccome solamente il 40% della varianza della temperatura è prevedibile dall'umidità.