Linear Regression Tool

Relazione progetto: Metodi Matematici e Statistici (6 CFU)

Pietro Biondi

Indice

Introduzione

Al giorno d'oggi in giro per il web vi sono sempre più dati disponibili su cui poter effettuare delle statistiche interessanti.

Attraverso l'uso di linguaggi di programmazione come Python / MATLAB è possibile manipolare i dati e ricavare tutte le specifiche che si desiderano.

Linear Regression Tool (LRT) consente di calcolare attributi e statistiche, attraverso le funzioni sviluppate in Python2.7, come: regressione lineare (coefficienti m e q), covarianza, coefficiente di Pearson.

A questo proposito grazie al coefficiente di Pearson possiamo capire se le due variabili sono correlate o meno.

Regressione Lineare

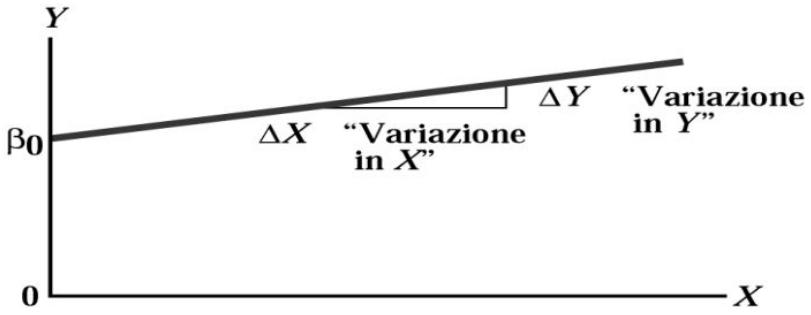
La regressione formalizza e risolve il problema di una relazione funzionale tra variabili misurate sulla base di dati campionari estratti da un'ipotetica popolazione infinita. Originariamente Galton utilizzava il termine come sinonimo di correlazione, tuttavia oggi in statistica l'analisi della regressione è associata alla risoluzione del modello lineare. Per la loro versatilità, le tecniche della regressione lineare trovano impiego nel campo delle scienze applicate: chimica, geologia, biologia, fisica, ingegneria, medicina, nonché nelle scienze sociali: economia, linguistica, psicologia e sociologia.

Più formalmente, in statistica la regressione lineare rappresenta un metodo di stima del valore atteso condizionato di una variabile dipendente, **Y**, dati i valori di altre variabili indipendenti, $X_1, \ldots, X_k$.
La retta di regressione si ottiene applicando il metodo dei minimi quadrati.
Il modello di regressione lineare è quindi

$$Y_i = \beta_0 + \beta_1 X_i + u_i$$

dove:

- i varia tra le osservazioni, i = 1, ..., n
- Y_i è la variabile dipendente
- X_i è la variabile indipendente
- $\beta_0 + \beta_1 X$ è la retta di regressione o funzione di regressione della popolazione;
- β_0 è l'intercetta della retta di regressione della popolazione, ovvero il punto dove la retta incontra l'asse delle ordinate;
- β_1 è il coefficiente angolare della retta di regressione della popolazione, esso indica quando varia la y al variare di x;
- u_i è l'errore statistico.



L'inclinazione β_1 indica come varia Y in corrispondenza di una variazione unitaria di X. Il segno di β_1 indica se la relazione lineare è positiva o negativa.
L'intercetta β_0 corrisponde al valore medio di Y quando X è uguale a 0.

Per ogni osservazione campionaria si dispone di una determinazione **Y** e di **K** determinazioni non dipendenti (non stocastiche) $X_1, X_2, \ldots, X_k$. Si cerca quindi una relazione di tipo lineare tra la variabile Y e le k variabili deterministiche. Per cui possiamo dire che la regressione viene utilizzata per costruire un modello attraverso cui prevedere i valori di una variabile dipendente o risposta (quantitativa) a partire dai valori di una o più variabili indipendenti o esplicative.

Funzionamento

L'applicazione è interamente sviluppata in Python 2.7 per il funzionamento è necessario importare delle librerie, nello specifico:

- Numpy (libreria per la computazione scientifica che include la possibilità di effettuare operazioni su array e matrici)
- Matplotlib (libreria per la creazione di grafici 2D)
- SciPy (libreria di algoritmi, strumenti matematici e strutture dati. Contiene moduli per l'ottimizzazione per algebra lineare, l'integrazione, FFT, elaborazione di segnali)
- Sklearn (libreria per il machine learning che contiene algoritmi di classificazione regressione e clustering. Sklearn è stata progettata per operare con Numpy e SciPy)
- Statistics (libreria che fornisce metodi per calcolo statistico)

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats as st
from sklearn.linear_model import LinearRegression
from statistics import mode,mean,stdev
```

Definizione e inizializzazione array:

```
array_X = np.array([29, 44, 36, 37, 53, 68, 75, 18, 31])
array_Y = np.array([30, 38, 36, 29, 28, 33, 35, 28, 30])
```

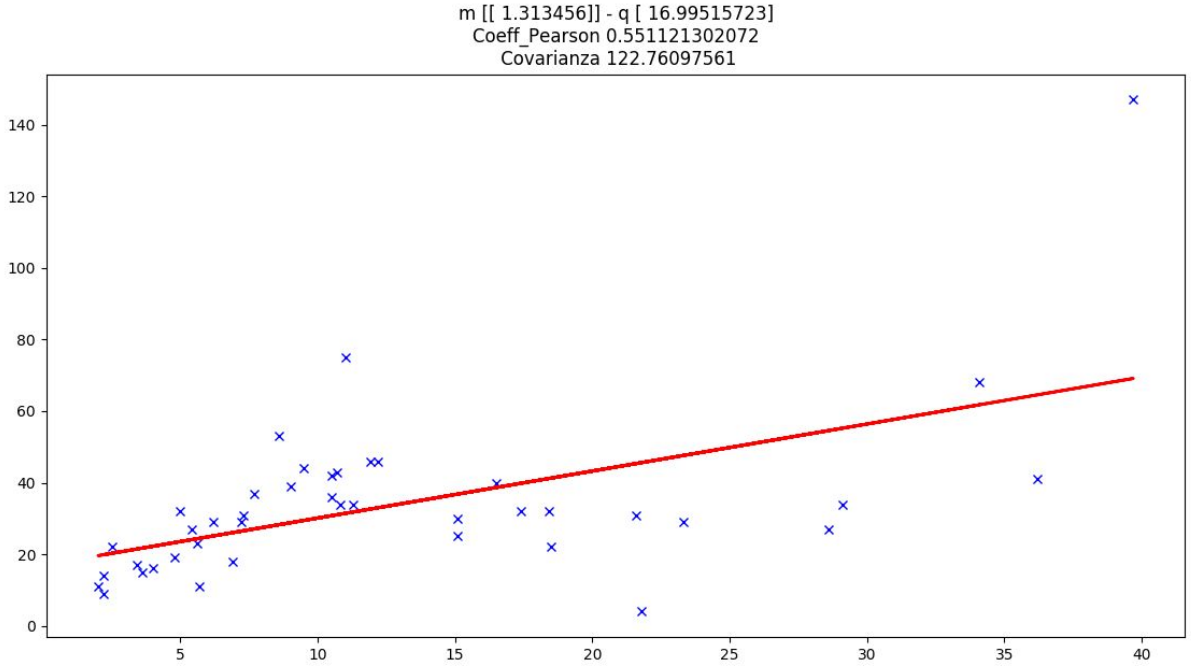
Creazione e allenamento del regressore e calcolo dati:

```
regr = LinearRegression()
regr.fit(X, Y)

m = regr.coef_
q = regr.intercept_
covarianza= np.cov(array_X,array_Y)
coefPears = st.pearsonr(array_X,array_Y)
```

Plotting dei dati (grafico):

```
plt.figure()
plt.plot(X, Y, 'bx')
plt.plot(X, regr.predict(X), color='Red', linewidth=2)
plt.show()
```



Spiegazione

L'applicazione richiede l'inserimento dei dati tramite array di numpy come visualizzato nel punto precedente.
Una volta inseriti i dati, l'oggetto LinearRegression() di Sklearn verrà allenato con i rispettivi parametri X , Y. Successivamente sarà possibile ottenere il coefficiente angolare (m) e il termine noto (q).
Avendo a disposizione i due array, è inoltre possibile ottenere altri dati statistici come covarianza e coefficiente di Pearson, per vedere la correlazione tra i due dati.

Risultati

Avendo inserito come input gli array X e Y, rispettivamente pari a :

X = np.array([6.2, 9.5, 10.5, 7.7, 8.6, 34.1, 11, 6.9, 7.3, 15.1, 29.1, 2.2, 5.7, 2, 2.5, 4, 5.4, 2.2, 7.2, 15.1, 16.5, 18.4, 36.2, 39.7, 18.5, 23.3, 12.2, 5.6, 21.8, 21.6, 9, 3.6, 5, 28.6, 17.4, 11.3, 3.4, 11.9, 10.5, 10.7, 10.8, 4.8])

Y = np.array([29, 44, 36, 37, 53, 68, 75, 18, 31, 25, 34, 14, 11, 11, 22, 16, 27, 9, 29, 30, 40, 32, 41, 147, 22, 29, 46, 23, 4, 31, 39, 15, 32, 27, 32, 34, 17, 46, 42, 43, 34, 19])

Abbiamo effettuato delle operazioni su essi.
Tramite le varie librerie abbiamo calcolato: regressione lineare, covarianza, coefficiente di Pearson, coefficiente angolare (m) e termine noto (q).

Nell'immagine vengono riportati i risultati calcolati tramite l'applicazione in Python 2.7

```
pietro@pietro-biondi ~/Scrivania/Progetto Metodi Mat e Statistici  $ python regressoreStatistico.py
DATI

Covarianza
122.76097561

Coefficiente di Pearson
0.551121302072

Coefficiente angolare (m)
[[ 1.313456]]

Termine noto (q)
[ 16.99515723]
```

Conclusioni

Avendo ottenuto un coefficiente di Pearson pari a 0.55, possiamo concludere che le due variabili X e Y sono correlate positivamente.
Inoltre possiamo affermare che si ha una correlazione moderata tra di esse poiché Pxy è compreso tra 0.3 e 0.7 .

Correlazione e indipendenza [modifica | modifica wikitesto]

Nella pratica si distinguono vari "tipi" di correlazione.

- Se $\rho_{XY} > 0$, le variabili X e Y si dicono *direttamente correlate*, oppure *correlate positivamente*;
- se $\rho_{XY} = 0$, le variabili X e Y si dicono *incorrelate*;
- se $\rho_{XY} < 0$, le variabili X e Y si dicono *inversamente correlate*, oppure *correlate negativamente*.

Inoltre per la correlazione diretta (e analogamente per quella inversa) si distingue:

- se $0 < \rho_{XY} < 0,3$ si ha *correlazione debole*;
- se $0,3 < \rho_{XY} < 0,7$ si ha *correlazione moderata*;
- se $\rho_{XY} > 0,7$ si ha *correlazione forte*.

Referenze

1. Dati ricavati :
   http://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/slr/frames/frame.html
2. Regressione lineare:
   https://it.wikipedia.org/wiki/Regressione_lineare
3. Coefficiente di Pearson:
   https://it.wikipedia.org/wiki/Indice_di_correlazione_di_Pearson
4. Numpy:
   http://www.numpy.org/
5. Matplotlib
   http://matplotlib.org/
6. SciPy
   https://www.scipy.org/
7. Sklearn
   http://scikit-learn.org/stable/
8. Statistics
   https://docs.python.org/3/library/statistics.html