

Programming Lab

Parte 9

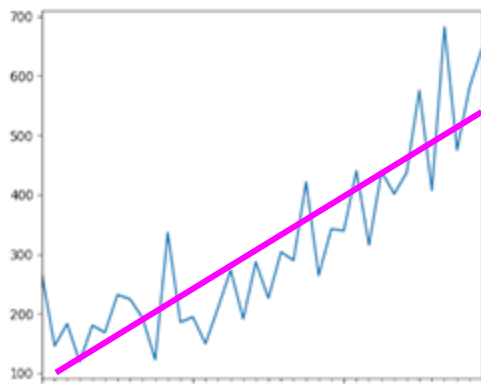
Lavorare veramente 2: fittiamo un modello

Laura Nenzi

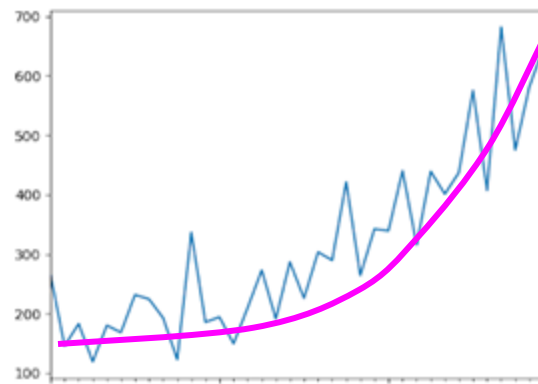
Many slides by Stefano Russo

Cosa vuol dire fare il “fit” di un modello

- Per “fit” di un modello si intende farlo “aderire” ai dati.
- Per esempio, stimare i coefficienti di una retta (detto anche regressione lineare) o di una curva (regressione esponenziale)



$$f(x) = a \cdot x + b$$

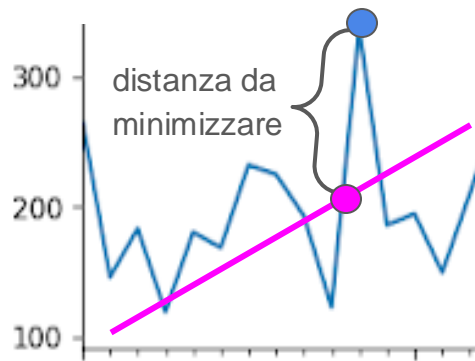


$$f(x) = a \cdot e^{(bx)} + c$$

Come si fa il “fit” di un modello

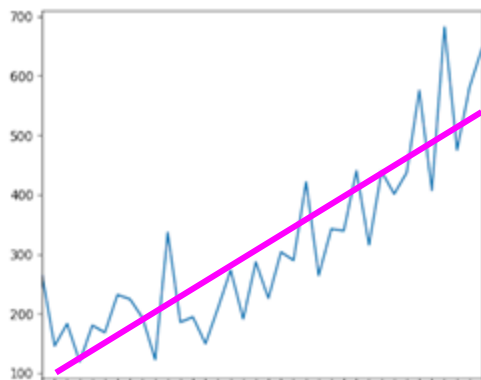
- Per calcolare i coefficienti di queste funzioni in modo che “aderiscano” al meglio ai dati si usano metodologie statistiche di minimizzazione della distanza, punto per punto, tra i dati e la curva (più o meno).

→ Non lo vedremo, ma lo studierete a breve.

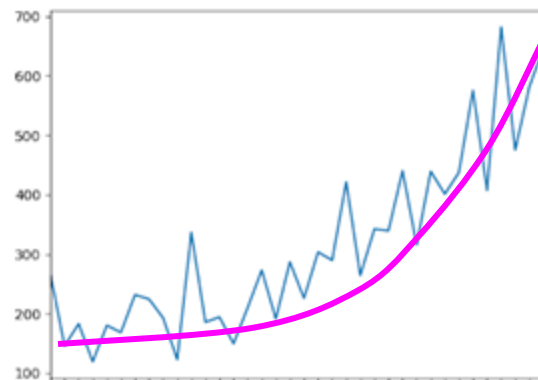


Cosa vuol dire fare il “fit” di un modello

- Minimizzando la distanza punto punto trovo i coefficienti che mi fanno aderire meglio la mia retta o curva e posso calcolare l'errore, che mi permette di valutare il modello (come vedremo la prossima ed ultima lezione)



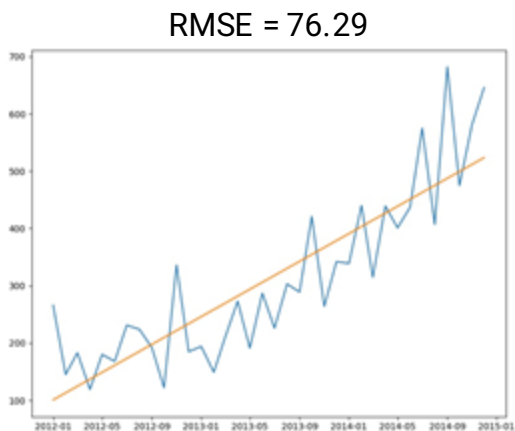
$$f(x) = a \cdot x + b$$



$$f(x) = a \cdot e^{(bx)} + c$$

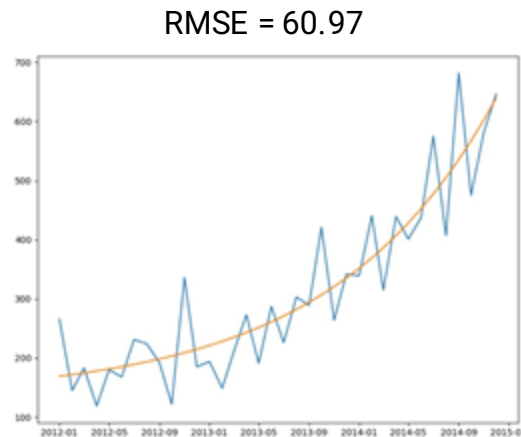
Cosa vuol dire fare il “fit” di un modello

- Minimizzando la distanza punto punto trovo i coefficienti che mi fanno aderire meglio la mia retta o curva e posso calcolare l'errore, che mi permette di valutare il modello (come vedremo la prossima ed ultima lezione)



$$f(x) = a \cdot x + b$$

$$a = 12.08, b = 89.14$$



$$f(x) = a \cdot e^{(bx)} + c$$

$$a = 31.41, b = 0.077, c = 135.81$$

Come faremo noi un “fit” di esempio

- Vediamo un esempio molto più semplice di una retta o una curva esponenziale, perchè in fin dei conti fare il “fit” di un modello vuol dire semplicemente sfruttare i dati per modellare meglio il fenomeno.
 - Prenderemo l’incremento medio delle vendite di shampoo su **tutto** il dataset, non solo sugli ultimi tre (o “n”) mesi come fatto l’altra volta.

Il nostro modello “fittabile”

Creiamo un nuovo modello per le vendite dello shampoo, tale per cui:

Le vendite dello shampoo al tempo $t+1$ sono date da

- la variazione media negli n mesi precedenti,
- applicata sulle vendite al tempo t .
- *mediata con la variazione media su tutto il dataset*

Esempio

| Mese | Passo temporale | Vendite |
|-----------|-----------------|---------|
| Maggio | non rilevante | 8 |
| Giugno | non rilevante | 19 |
| Luglio | non rilevante | 31 |
| Agosto | non rilevante | 41 |
| Settembre | t-2 | 50 |
| Ottobre | t-1 | 52 |
| Novembre | t (adesso) | 60 |
| Dicembre | t+1 | ? |

Uso tre mesi precedenti
per la predizione (n=3)

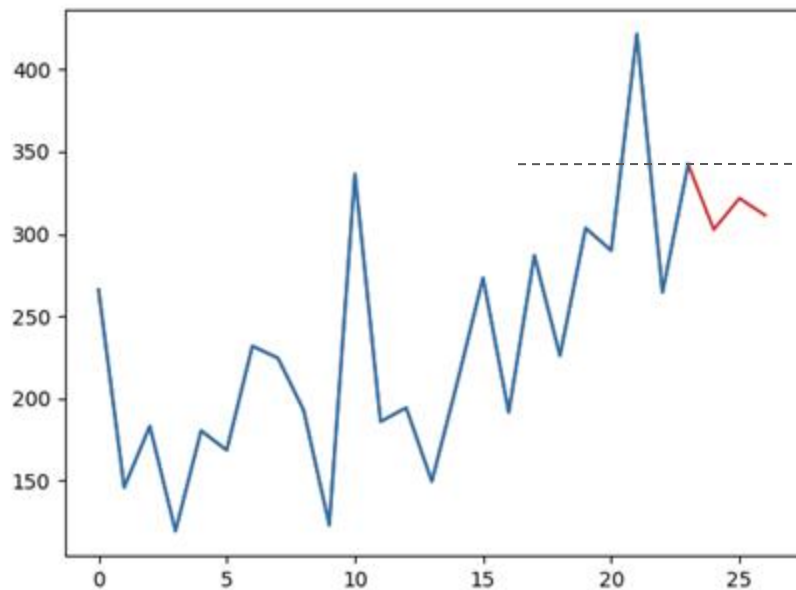
Esempio

| Mese | Passo temporale | Vendite |
|-----------|-----------------|---|
| Maggio | non rilevante | 8 |
| Giugno | non rilevante | 19 |
| Luglio | non rilevante | 31 |
| Agosto | non rilevante | 41 |
| Settembre | t-2 | 50 |
| Ottobre | t-1 | 52 |
| Novembre | t (adesso) | 60 |
| Dicembre | t+1 | $60 + (((11+12+10) / 3) + ((2+8) / 2)) / 2 = 68$ |

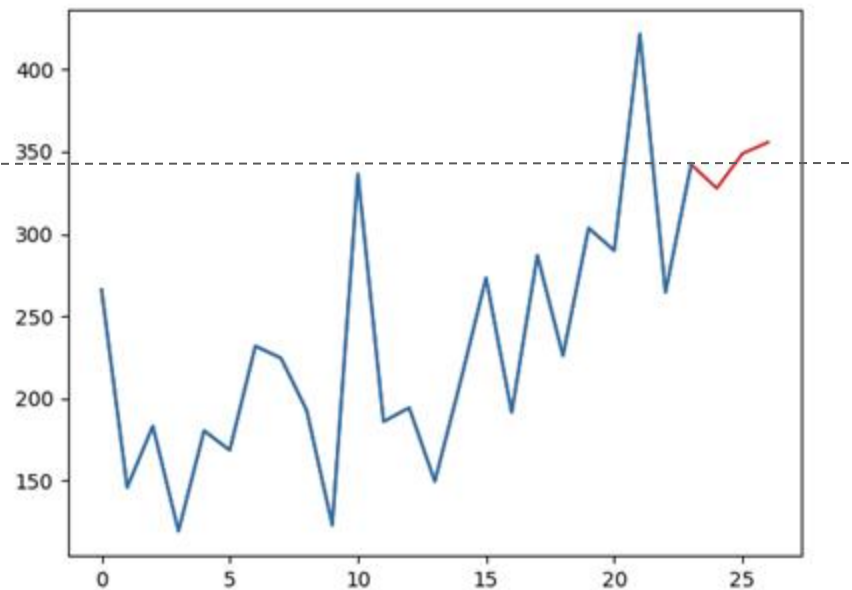
(con il modello dell'altra lezione era 65)

Predizioni a confronto sulle vendite di shampoo

Modello senza fit (quello dell'altra lezione)

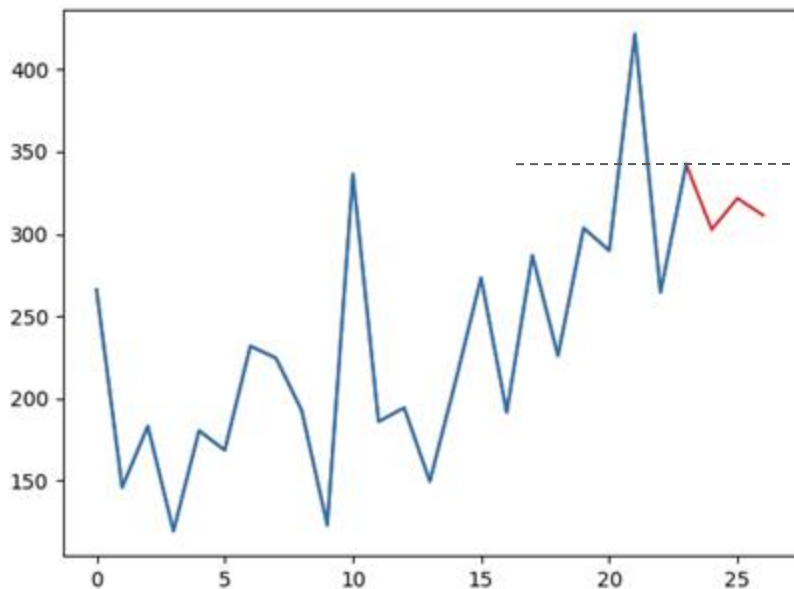


Modello con fit (quello appena visto)

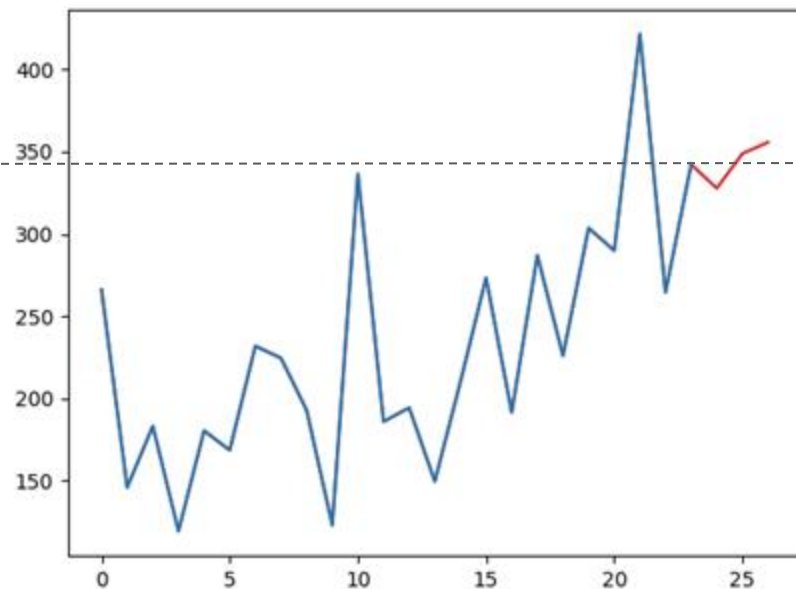


Predizioni a confronto sulle vendite di shampoo

Modello senza fit (quello dell'altra lezione)



Modello con fit (quello appena visto)



Capire quale va meglio in generale è argomento della prossima lezione!

Esercizio (preambolo)

Ci accorgiamo che andremmo a scrivere la stessa logica (di calcolo delle variazioni) più volte, e che in generale sto avendo più “tipi” di modelli.

- vorrò avere un metodo di supporto per le variazioni
→ (i.e. `compute_avg_variation`)
- ed una classe base

(nelle dispense è spiegato tutto molto meglio, ora ci interessa solo di inquadrare il problema)

L'oggetto Model di base

Creiamo un oggetto **Model** come base. Avrà due metodi principali:

- il metodo “fit” per fittare il modello su dei dati, e
- il metodo “predict” per ottenere delle previsioni a partire da altri dati.

models.py

```
class Model():  
  
    def fit(self, data):  
        # Fit non implementato nella classe base  
        raise NotImplementedError('Metodo non implementato')  
  
        def predict(self, data):  
            # Predict non implementato nella classe base  
            raise NotImplementedError('Metodo non implementato')
```

L'oggetto Model di base

Creiamo un oggetto **Model** come base. Avrà due metodi principali:

- il metodo “fit” per fittare il modello su dei dati, e
- il metodo “predict” per ottenere delle previsioni a partire da altri dati.

models.py

```
class Model():  
  
    def fit(self, data):  
        # Fit non implementato nella classe base  
        raise NotImplementedError('Metodo non implementato')  
  
        def predict(self, data):  
            # Predict non implementato nella classe base  
            raise NotImplementedError('Metodo non implementato')
```

Esercizio

Creiamo l'oggetto **Model** scritto nelle slide precedenti e modifichiamo la classe **TrendModel** in modo che estenda **Model** andando ad implementare la `predict`.

Definiamo una funzione di supporto **`compute_avg_variation`** che calcoli la variazione media. Definendola come metodo della classe **TrendModel**, tutte le sue sottoclassi la ereditano.

Esercizio

Estendete il modello della lezione precedente **TrendModel** come **FitTrendModel**, andando ad implementare il metodo *fit()*.

Il fit deve, come appena descritto, calcolare l'incremento medio su tutto il dataset e salvarlo da qualche parte (es: `self.historical_avg_variation`).

Poi, sovrascrivete il metodo *predict()* in modo che usi l'incremento medio su tutto il dataset come descritto nelle slides precedenti.

Usate l'esempio numerico delle slides prima di provare con i dati delle vendite di shampoo!!

P.S.: per graficare i dati e/o la predizione:

```
data = [8,19,31,41,50,52,60]
prediction = 68

import matplotlib.pyplot as plt
plt.plot(data + [prediction], color='tab:red')
plt.plot(data, color='tab:blue')
plt.show()
```

(una volta che si vede il grafico, premere assieme “crontol” e “c” nel terminale per “uccidere” lo script Python, liberando lo schermo e riprendendo il controllo del terminale)