



UNIVERSITÀ DEGLI STUDI DI FIRENZE  
SCUOLA DI INGEGNERIA - DIPARTIMENTO DI INGEGNERIA  
DELL'INFORMAZIONE

---

Tesi di Laurea Magistrale in Ingegneria Informatica

**PROGETTAZIONE E SVILUPPO DI COMPONENTI  
PER LA PIATTAFORMA AIRQINO DEDICATA AL  
MONITORAGGIO DELLA QUALITÀ DELL'ARIA**

*Candidato*  
Edoardo D'Angelis

*Relatori*  
Prof. Andrew D. Bagdanov  
Prof. Pietro Pala

*Correlatori*  
Dott. Walter Nunziati  
Dott. Alice Cavaliere

---

Anno Accademico 2020/2021

# Abstract

Air pollution is currently one of the main issues affecting urbanized areas worldwide. Local administrations monitor these harmful gases by means of reference monitoring stations provided by regional/national environmental protection agencies. These stations, however, have limitations due to coarse spatial coverage of the whole municipality, low time-frequency, and high costs. In this framework, the National Research Council of Italy (CNR-IBE) and the Tuscany Region Environmental Protection Agency (ARPAT) agreed to an initiative to create a low-cost network aimed at monitoring air quality over an Italian urban area. The rural town of Capannori, located in the Tuscany region (Italy), was chosen as a testing area since it lies within a critical area both affected by a variety of emission sources and weather conditions unfavourable to pollutant dispersion. The air quality analysis was carried out by means of several innovative low-cost stations named AIRQino, equipped with sensors for collecting air pollution (PM<sub>2.5</sub>, PM<sub>10</sub>, NO<sub>2</sub>, O<sub>3</sub>, CO, CO<sub>2</sub>) and meteorological parameters (air temperature and relative humidity). Concentrations of PM were mainly considered in this work for providing indicative air quality measurements to supplement fixed measurements collected by the official urban monitoring network. This work, still ongoing, has two main objectives: (i) to show the robustness of AIRQino at measuring PM concentrations; (ii) to investigate the PM concentrations dynamics at

higher spatial and time scale distribution compared to the reference station.

# Sommario

...

# Indice

<b>Abstract</b>	<b>i</b>
<b>Sommario</b>	<b>iii</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Contesto . . . . .	1
1.1.1 Descrizione del problema . . . . .	1
1.1.2 Motivazioni . . . . .	2
1.2 La piattaforma AirQino . . . . .	2
1.2.1 Hardware dei sensori . . . . .	2
1.2.2 Architettura e tecnologie . . . . .	3
1.2.3 Progetti correlati . . . . .	3
1.2.4 Progetti simili . . . . .	4
<b>2 Sviluppi tecnologici</b>	<b>6</b>
2.1 Replica del database di produzione . . . . .	6
2.1.1 Motivazioni . . . . .	6
2.1.2 Streaming Replication . . . . .	6
2.1.2.1 Preparazione del database primario . . . . .	7
2.2 Ottimizzazione di query temporali . . . . .	8
2.2.1 Motivazioni . . . . .	8

2.2.2	Continuous Aggregates . . . . .	8
2.2.3	Risultati ottenuti . . . . .	8
<b>3</b>	<b>Calibrazione</b>	<b>9</b>
3.1	I dati a disposizione . . . . .	10
3.1.1	Dataset NO <sub>2</sub> . . . . .	12
3.1.2	Dataset PM <sub>2.5</sub> e PM <sub>10</sub> . . . . .	14
3.1.3	Preprocessamento . . . . .	16
3.2	Regressione . . . . .	16
3.2.1	Introduzione . . . . .	16
3.2.2	Correlazione e coefficiente di determinazione . . . . .	18
3.2.3	Analisi dei residui . . . . .	22
3.2.3.1	Distribuzione degli errori . . . . .	23
3.2.3.2	Correlazione tra errore e variabili . . . . .	23
3.2.3.3	Omogeneità della varianza dei residui . . . . .	24
3.2.3.4	Influenza di outliers . . . . .	24
3.2.4	Modelli di regressione . . . . .	25
3.2.4.1	Regressione lineare . . . . .	25
3.2.4.2	Regressione lineare robusta (Huber) . . . . .	27
3.2.4.3	Regressione lineare avanzata . . . . .	28
3.2.4.4	Regressione Ridge . . . . .	31
3.2.4.5	Regressione Lasso . . . . .	32
3.2.4.6	Regressione polinomiale . . . . .	33
3.2.4.7	Regressione con Random Forest . . . . .	35
3.2.4.8	Regressione con Gradient Boosting . . . . .	36
3.2.4.9	Regressione con SVR . . . . .	36
3.2.4.10	Regressione con KernelRidge . . . . .	38
3.3	Esperimenti e risultati ottenuti . . . . .	38

3.3.1	NO <sub>2</sub>	38
3.3.2	PM <sub>2.5</sub>	38
3.3.3	PM <sub>10</sub>	39
3.4	Validazione	39
3.4.1	PM <sub>2.5</sub>	39
3.4.2	PM <sub>10</sub>	39
3.5	Discussione	40
<b>4</b>	<b>Interfaccia di calibrazione</b>	<b>41</b>
4.1	Motivazioni	41
4.2	Tecnologie	41
4.2.1	Backend	41
4.2.2	Frontend	41
4.3	Funzionamento	41
4.4	Autenticazione	42
4.5	CI e deploy automatico	42
<b>Conclusioni e sviluppi futuri</b>		<b>44</b>
<b>Bibliografia</b>		<b>45</b>

# Capitolo 1

## Introduzione

tesi realizzata in collaborazione con magenta e ibe cnr + foto ecc

### 1.1 Contesto

Il monitoraggio della qualità dell'aria è una delle attività più importanti per la tutela della salute pubblica. La qualità dell'aria può essere influenzata da molte sorgenti di emissione, tra cui le automobili, le centrali elettriche, gli impianti di riscaldamento e le fabbriche. I principali inquinanti atmosferici sono il biossido di zolfo, gli idrocarburi policiclici aromatici, il monossido di carbonio e gli ozono. Gli effetti dell'inquinamento atmosferico sulla salute sono molteplici e possono essere a breve o a lungo termine. I principali rischi sono l'asma, le malattie cardiovascolari, il cancro e le malattie respiratorie. Il monitoraggio della qualità dell'aria permette di individuare le sorgenti di emissione e di intervenire per ridurre l'inquinamento atmosferico.

#### 1.1.1 Descrizione del problema

...

### 1.1.2 Motivazioni

...

## 1.2 La piattaforma AirQino

AirQino è una piattaforma di monitoraggio ambientale ad alta precisione, realizzata dal Consiglio Nazionale delle Ricerche (CNR) in collaborazione con TEA Group e Quanta Srl. Il progetto nasce dall'esigenza di realizzare una rete di stazioni mobile per un monitoraggio più completo della qualità dell'aria in ambito urbano, in linea con la Direttiva 2008/50/EC, che riconosce e regolamenta l'importanza di misure aggiuntive rispetto a quelle delle stazioni fisse.

Nonostante infatti l'attività svolta da ARPA, a causa del numero limitato di stazioni e/o di sorgenti monitorate, ad oggi, la conoscenza sullo stato dell'inquinamento dell'aria da parte degli Enti Locali rimane molto limitata.

### 1.2.1 Hardware dei sensori

Per quanto riguarda la caratteristiche dei sensori, i sensori di tipo MOS sono costituiti da un film (credo allumina? Per fabbricare gli strati sensibili del film, si prepara una pasta viscosa: al materiale funzionale, sotto forma di polvere, viene aggiunta una miscela di agenti reologici in solventi volatili) depositato su una piastra di elementi riscaldanti la cui temperatura operativa è generalmente compresa tra 300 e 500°C. Di solito il materiale funzionale del film più adatto per la rilevazione di biossido di azoto è l'ossido di ferro e lantanio ( $\text{LaFeO}_3$ ) che oltre ad avere una buona sensibilità agli ossidi di azoto ha una bassa sensibilità al monossido di carbonio. Per la rilevazione dell'ozono viene invece utilizzato triossido di tungsteno ( $\text{WO}_3$ ). Questo ti-

po di materiale funzionale risulta molto sensibile ai gas ossidanti come O<sub>3</sub> e NO<sub>2</sub>. Qualsiasi sia il materiale funzionale, il principio di funzionamento per tutti i MOS nella rilevazione di gas è quello di interagire con il gas presente all'interno dell'atmosfera tramite reazioni di ossidoriduzione, portando a un cambiamento di conduttività, che viene rilevato da un circuito apposito. Le variazioni della conduttività dei sensori è fortemente influenzata dalle variazioni di umidità e temperatura, come rilevato dalla letteratura sull'argomento [ref]. Nel caso dei sensori Mics che noi utilizziamo, il produttore non rilascia informazioni sull'influenza nella lettura dovuta alla temperatura/umidità ma che queste influiscono può essere ipotizzato come può essere ipotizzato che ci sia una influenza introdotta dalla temperatura nel circuito ADC del microcontrollore.

Questo segnale viene passato al convertitore analogico digitale del controllore che lo trasforma in counts (10 bit da 0 a 2 alla 10).

ossidoriduzione → piastra che si scalda a seconda dell'inquinante genera corrente

il segnale viene passato attraverso un convertitore analogico digitale e l'uscita è a 10 bit (questa unità la chiamo counts)

### **1.2.2 Architettura e tecnologie**

...

### **1.2.3 Progetti correlati**

...

### 1.2.4 Progetti simili

- **Airly** (<https://airly.org/>) è una piattaforma che consente di dividere informazioni ambientali in tempo reale, grazie alla quale è possibile monitorare la qualità dell'aria e i livelli di inquinamento;
- **Aqicn** (<https://aqicn.org>) è un progetto open source lanciato nel 2010 che consente di monitorare l'inquinamento atmosferico in tempo reale;
- **IQAir** (<https://aqicn.org>) è una società svizzera che produce e vende purificatori d'aria per uso residenziale e commerciale. La loro applicazione fornisce un rapporto in tempo reale sulla qualità dell'aria e previsione dell'inquinamento atmosferico;
- **Decentlab** (<https://decentlab.com>) è un'azienda svizzera che fornisce dispositivi e servizi di sensori wireless per soluzioni di monitoraggio distribuite ed economiche;
- **SMART Treedom** (<https://smart.treedom.net>) è il frutto dalla collaborazione tra Treedom e l'Istituto di Biometeorologia del Consiglio Nazionale delle Ricerche. La finalità del progetto è stata quella di prototipare un sistema integrato che possa essere modulato con diversi sensori in base al tipo di grandezza fisica che si vuole misurare e una tecnologia laser per la misura delle polveri sottili;
- **PlanetWatch** (<https://planetwatch.io>) è una piattaforma decentralizzata che consente di monitorare e proteggere il pianeta attraverso la condivisione di informazioni. Gli utenti possono condividere informazioni sull'ambiente, la sostenibilità e la responsabilità sociale;

- **HackAIR** (<https://hackair.eu>) è una piattaforma open source che consente ai cittadini di monitorare la qualità dell'aria nei propri quartieri. Gli utenti possono interagire con la piattaforma per segnalare la qualità dell'aria nel proprio quartiere, visualizzare i dati relativi alla qualità dell'aria e condividere informazioni e dati con altri utenti.

# Capitolo 2

## Sviluppi tecnologici

### 2.1 Replica del database di produzione

...

#### 2.1.1 Motivazioni

...

#### 2.1.2 Streaming Replication

La streaming replication di PostgreSQL è una funzionalità che consente di replicare i dati in tempo reale da una istanza di PostgreSQL a un'altra. Questo significa che, se si modificano i dati in una delle istanze, questi saranno immediatamente replicati anche nell'altra istanza. La streaming replication di PostgreSQL offre diversi vantaggi:

- maggiore disponibilità dei dati: se una delle istanze di PostgreSQL viene a mancare, i dati saranno comunque disponibili nell'altra istanza;

- maggiore velocità di replica: i dati vengono replicati in tempo reale, senza dover attendere il completamento delle operazioni di replica;
- riduzione del carico sulle risorse: la replica in tempo reale riduce il carico sulle risorse della infrastruttura di storage.

La replica si basa sulle transazioni WAL (Write Ahead Log) e utilizza il protocollo TCP per garantire una connessione sicura tra i server. —

TimescaleDB può gestire la replica utilizzando la streaming replication integrata di PostgreSQL (vedi docs ufficiali: <https://docs.timescale.com/timescaledb/latest/how-to-guides/replication-and-ha/replication/>).

#### 2.1.2.1 Preparazione del database primario

- Creare un utente PostgreSQL con un ruolo adatto ad avviare la streaming replication:

```
1 SET password_encryption = 'scram-sha-256';
2 CREATE ROLE repuser WITH REPLICATION PASSWORD 'SOME_SECURE_PASSWORD' LOGIN;
```

Estratto 2.1: TODO

- Aggiungere i seguenti parametri al file `/var/lib/postgresql/data/postgresql.conf`:

```
1 listen_addresses= '*'
2 wal_level = replica
3 max_wal_senders = 2
4 max_replication_slots = 2
5 synchronous_commit = off
```

Estratto 2.2: TODO

## 2.2 Ottimizzazione di query temporali

...

### 2.2.1 Motivazioni

...

### 2.2.2 Continuous Aggregates

I continuous aggregate sono una funzionalità integrata in TimescaleDB che consente di aggregare i dati in tempo reale, senza la necessità di eseguire query aggiuntive. Questa funzionalità utilizza i contatori per tenere traccia dei dati aggregati in tempo reale e fornisce una rappresentazione dei dati aggregati in tempo reale.

I continuous aggregate offrono numerosi vantaggi, tra cui:

- Flessibilità: è possibile aggregare dati in tempo reale in base a qualsiasi criterio desiderato.
- Risparmio di tempo: non è necessario eseguire query aggiuntive per ottenere informazioni aggregate in tempo reale.
- Risparmio di spazio: i dati aggregati in tempo reale occupano meno spazio rispetto ai dati non aggregati.
- Maggiore efficienza: i continuous aggregate sono più efficienti dei query batch per l'aggregazione dei dati in tempo reale.

### 2.2.3 Risultati ottenuti

...

# Capitolo 3

## Calibrazione

Questo capitolo riguarda la parte di tesi incentrata sulla calibrazione delle centraline AirQino (1.2). Nella sezione 3.1 vengono presentati i dataset a disposizione, la loro struttura e il lavoro di preprocessamento fatto.

La sezione 3.2 racchiude una breve panoramica teorica sui concetti di regressione, correlazione, metriche (coefficiente di correlazione, coefficiente di determinazione, scarto quadratico medio), analisi dei residui e modelli di regressione (sia lineare che non lineari).

La sezione 3.3 presenta i risultati ottenuti, mentre in 3.4 sono elencati gli esperimenti svolti in fase di validazione.

Infine, la sezione 3.5 viene riassunto tutto il lavoro svolto e fornita un'analisi qualitativa dei risultati ottenuti.

### 3.1 I dati a disposizione

I dataset messi a disposizione sono due:

- Dataset delle misurazioni di concentrazione di NO<sub>2</sub> nell'aria relative alla centralina SMART16 AirQino, da confrontare con i dati NO<sub>2</sub> ARPAT<sup>1</sup> della stazione Capannori (Lucca);
- Dataset delle misurazioni di concentrazione di PM<sub>2.5</sub> e PM<sub>10</sub> nell'aria relative alla centralina SMART16 AirQino, da confrontare con i dati PM<sub>2.5</sub> e PM<sub>10</sub> ARPAT della stazione Capannori (Lucca).

In entrambi i casi la centralina SMART16 è stata in co-locazione con la stazione ARPAT Capannori per tutto periodo di interesse.



Figura 3.1: Una centralina AirQino

Fonte: [airqino.magenta-lab.it](http://airqino.magenta-lab.it)

<sup>1</sup>Agenzia regionale per la protezione ambientale della Toscana



Figura 3.2: Centralina ARPAT in sede Capannori (provincia di Lucca)

Fonte: [arpat.toscana.it](http://arpat.toscana.it)

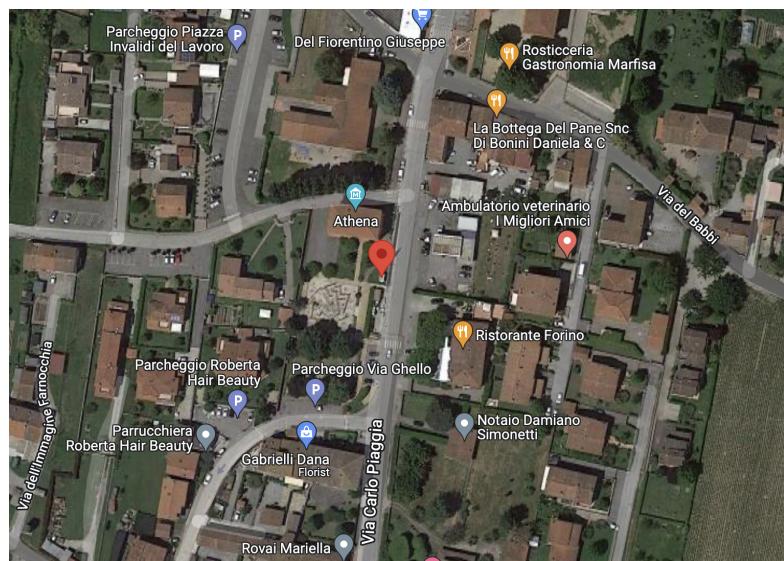


Figura 3.3: Posizione della centralina SMART16 (AirQino)  
e ARPAT (Capannori) a Lucca

### 3.1.1 Dataset NO<sub>2</sub>

Il dataset di misurazioni NO<sub>2</sub>, comprende sia i dati della centralina AirQino SMART16 che i dati di ARPAT (Capannori). Ci sono però delle differenze sostanziali tra i due set di dati:

	<b>Periodo</b>	<b>Unità</b>	<b>Frequenza dati</b>
<b>SMART16</b>	01/01/2020 - 31/12/2020	<i>counts</i>	ogni 1/2 minuti
<b>ARPAT</b>	01/01/2020 - 31/12/2020	$\mu\text{g}/\text{m}^3$	medie giornaliere

Tabella 3.1: Differenze tra i dati di SMART16 e ARPAT (per NO<sub>2</sub>)

Da notare che le centraline AirQino misurano la concentrazione di NO<sub>2</sub> con il sensore **MiCS-2714** (come già accennato in 1.2.1) che fornisce output in *counts* (unità di misura del segnale convertito da analogico a digitale, con uscita a 10 bit). Questo significa che sarà compito della fase di calibrazione convertire l'output direttamente in unità ingegneristica (in questo caso  $\mu\text{g}/\text{m}^3$ ).

Nella figura 3.4 sono riportate la frequenza di misurazione (inteso come il numero di misurazioni effettuate ogni ora) e l'andamento della concentrazione di NO<sub>2</sub> nell'aria (in *counts*) per come sono state misurate dalla centralina SMART16 nel periodo di interesse (01/01/2020 - 31/12/2020).

La figura 3.5 invece riporta l'andamento della concentrazione di NO<sub>2</sub> nell'aria (in  $\mu\text{g}/\text{m}^3$ ) misurato dalla stazione ARPAT di Capannori nello stesso periodo.

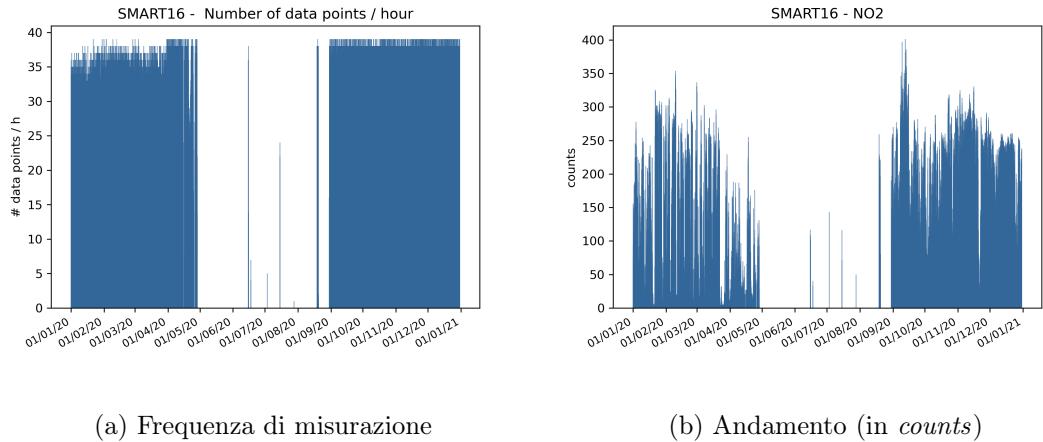


Figura 3.4: Frequenza di misurazione e andamento NO<sub>2</sub> (SMART16)  
nel periodo 01/01/2020 - 31/12/2020

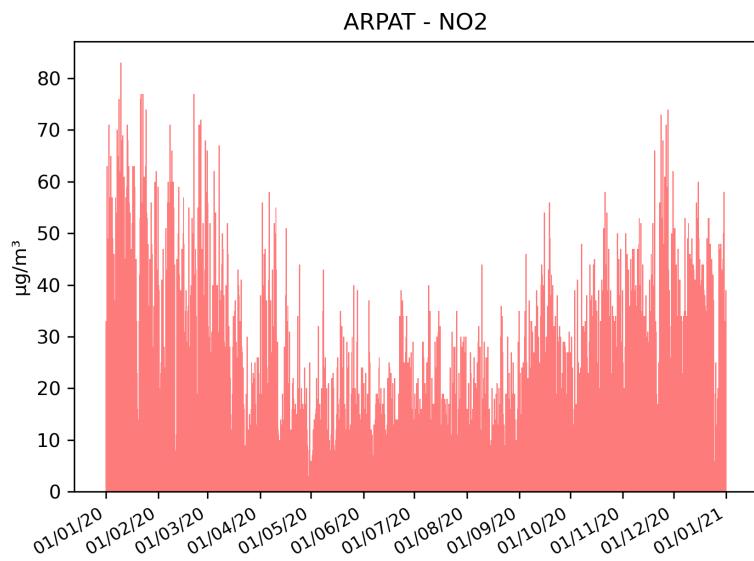


Figura 3.5: Andamento NO<sub>2</sub> (in  $\mu\text{g}/\text{m}^3$ ) misurato dalla stazione ARPAT di Capannori nel periodo 01/01/2020 - 31/12/2020

### 3.1.2 Dataset PM<sub>2.5</sub> e PM<sub>10</sub>

Il dataset di misurazioni PM<sub>2.5</sub> e PM<sub>10</sub>, che mette a confronto i dati della centralina AirQino SMART16 e i dati di ARPAT (Capannori), risulta invece così strutturato:

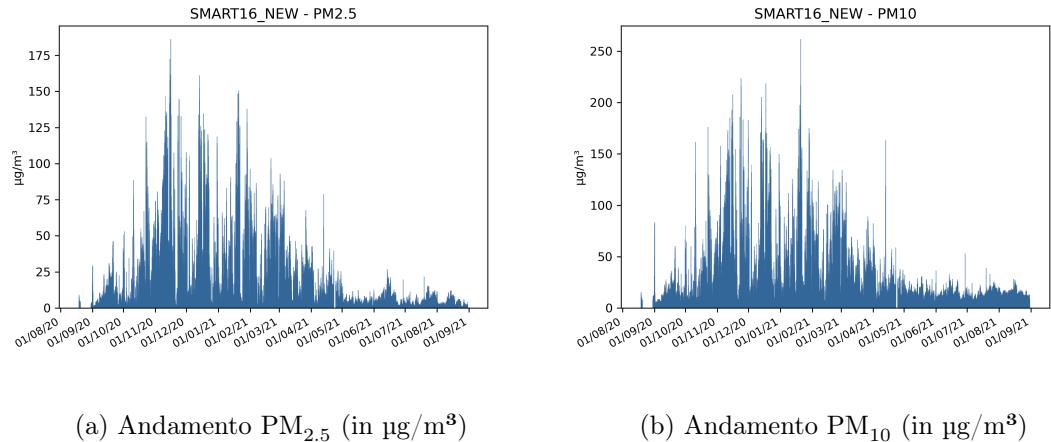
	<b>Periodo</b>	<b>Unità</b>	<b>Frequenza dati</b>
<b>SMART16</b>	01/09/2020 - 31/08/2021	$\mu\text{g}/\text{m}^3$	ogni 1/2 minuti
<b>ARPAT</b>	01/09/2020 - 31/08/2021	$\mu\text{g}/\text{m}^3$	medie ogni 8h

Tabella 3.2: Differenze tra i dati di SMART16 e ARPAT (per PM<sub>2.5</sub> e PM<sub>10</sub>)

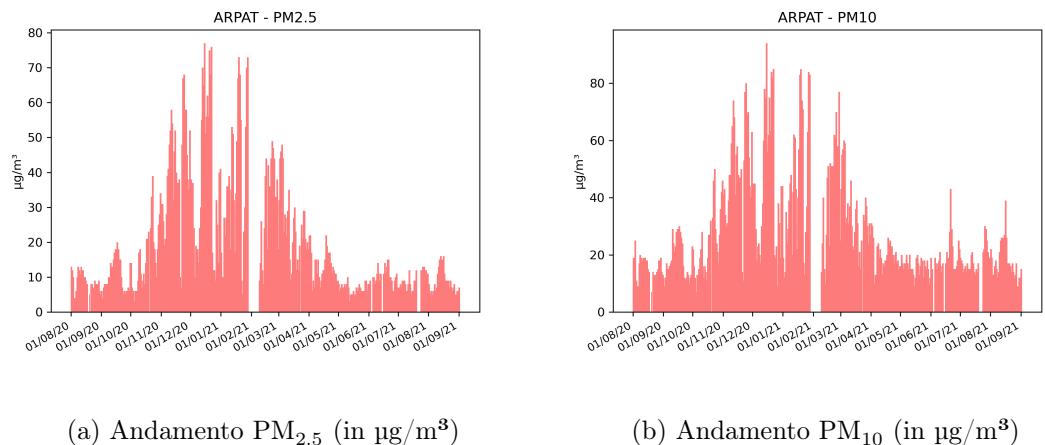
Il sensore utilizzato dalle centraline SMART è il **SDS011** (vedi 1.2.1) che ha la caratteristica di fornire l'uscita direttamente in unità ingegneristica ( $\mu\text{g}/\text{m}^3$ ), quindi in questo caso non c'è bisogno di convertire l'unità di misura nella fase di calibrazione (a differenza di quanto accade con NO<sub>2</sub>).

Nella figura 3.6 sono riportati gli andamenti della concentrazione di PM<sub>2.5</sub> e PM<sub>10</sub> nell'aria (in  $\mu\text{g}/\text{m}^3$ ) per come sono state misurate dalla centralina SMART16 nel periodo di interesse (01/09/2020 - 31/08/2021).

La figura 3.7 invece riporta gli andamenti della concentrazione di PM<sub>2.5</sub> e PM<sub>10</sub> nell'aria (in  $\mu\text{g}/\text{m}^3$ ) misurato dalla stazione ARPAT di Capannori nello stesso periodo.

Figura 3.6: Andamento PM<sub>2.5</sub> e PM<sub>10</sub> (SMART16)

nel periodo 01/09/2020 - 31/08/2021

Figura 3.7: Andamento PM<sub>2.5</sub> e PM<sub>10</sub> (ARPAT)

nel periodo 01/09/2020 - 31/08/2021

### 3.1.3 Preprocessamento

...

## 3.2 Regressione

Nella statistica applicata come nelle scienze sperimentali si osserva (o si ipotizza) l'esistenza di relazioni fra due o più grandezze.

Sorge allora il problema di determinare una funzione che, in base ai dati ricavati mediante esperimenti o rilevazioni statistiche, rappresenti queste relazioni permettendo, in questo modo, di analizzare meglio i fenomeni osservati.

### 3.2.1 Introduzione

Limitando lo studio a problemi che stabiliscono relazioni fra due sole variabili, si tratta, partendo dalle coppie  $(x_i, y_i)$  di dati corrispondenti rilevati, di determinare una funzione  $y = f(x)$  che rappresenti il fenomeno.

Per trovare una funzione che rappresenti il fenomeno si può procedere in due modi:

- determinare una funzione che assuma esattamente i valori  $(x_i, y_i)$  rilevati; questo procedimento viene detto interpolazione per punti noti;
- determinare una funzione che si accosti il più possibile ai punti  $(x_i, y_i)$ ; questo procedimento viene detto interpolazione fra punti noti.

La ricerca di una funzione, generalmente espressa da un polinomio, che passi esattamente per i punti  $(x_i, y_i)$  è piuttosto laboriosa; nelle applicazioni statistiche si preferisce determinare una funzione il cui grafico si avvicini ai punti rilevati.

Osservando l'andamento del fenomeno si sceglie il tipo di funzione interpolatrice: lineare, quadratica, esponenziale, ecc. e quindi si procede alla determinazione dei parametri, ossia delle costanti che compaiono nella funzione scelta in modo che sia soddisfatta una condizione di accostamento prefissata.

Per conseguire questo scopo il metodo più utilizzato è il metodo dei **minimi quadrati** che costituisce un'applicazione della ricerca del minimo di una funzione di più variabili mediante gli strumenti dell'analisi infinitesimale.

Si considerino due variabili  $X$  e  $Y$  sulle quali si sono effettuate  $n$  rilevazioni:

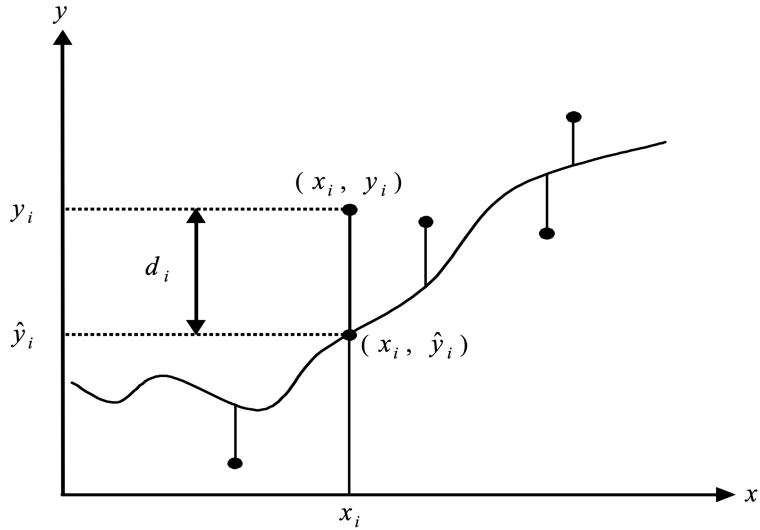
$$(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_n, y_n)$$

Sia  $y = f(x; a, b, c, \dots, k)$  la funzione interpolatrice scelta. Siano inoltre  $\hat{y}_i$  valori predetti sulla curva corrispondenti ai valori  $x_i$  rilevati.

La condizione di accostamento data dal metodo dei minimi quadrati è quella di determinare i valori dei parametri in modo che sia minima la somma dei quadrati delle differenze fra i valori osservati  $y_i$  e i valori predetti  $\hat{y}_i$  (figura 3.8), ovvero:

$$\varphi(a, b, c, \dots, k) = \sum_{i=1}^n [y_i - f(x_i; a, b, c, \dots, k)]^2$$

dove i valori  $x_i$  e  $y_i$  sono noti, mentre sono incogniti i parametri  $a, b, c, \dots, k$  della funzione. [1]

Figura 3.8: Condizione dei *minimi quadrati* [1]

### 3.2.2 Correlazione e coefficiente di determinazione

Quando la dipendenza tra le due variabili è lineare, si parla di correlazione lineare, che può essere valutata mediante il coefficiente di correlazione lineare ( $r$ ):

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

dove il termine al numeratore rappresenta la *covarianza* di  $X$  ed  $Y$  cioè la variabilità congiunta delle coppie  $(x_i, y_i)$  di valori corrispondenti rispetto al proprio valor medio; mentre il denominatore rappresenta il prodotto delle deviazioni standard di  $X$  ed  $Y$ .

Il coefficiente di correlazione lineare gode di importanti proprietà:

- $-1 \leq r \leq 1$ ;

- si ha  $r = 1$  quando tutti i dati sono allineati lungo una retta crescente (figura 3.9);

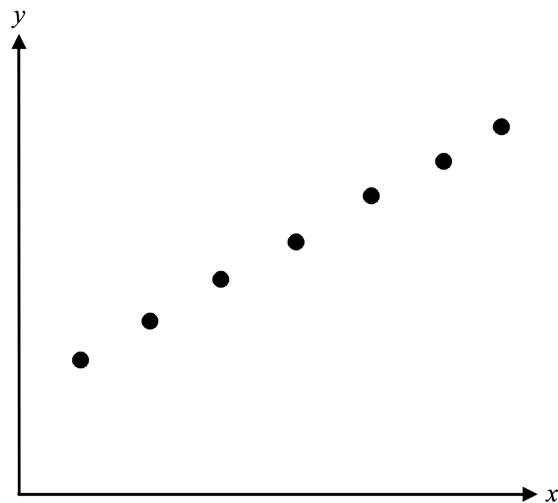


Figura 3.9: Correlazione lineare positiva

- si ha  $r = -1$  quando tutti i dati sono allineati lungo una retta decrescente (figura 3.10);

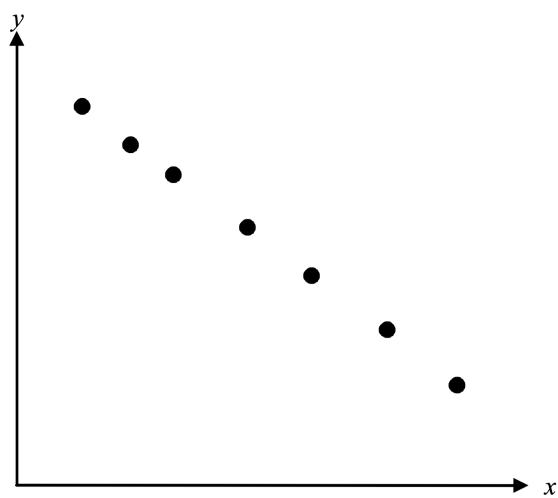


Figura 3.10: Correlazione lineare negativa

- si ha  $r = 0$  quando non esiste una relazione lineare tra i dati (figura 3.11).

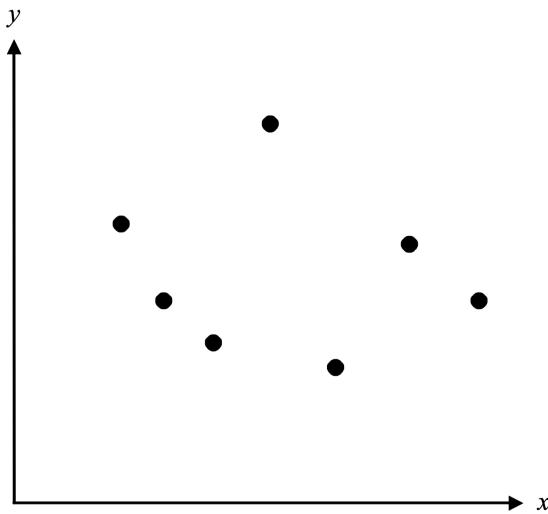


Figura 3.11: Nessuna correlazione

Sapendo che la varianza ( $\sigma_y^2$ ) della variabile  $Y$  si può scomporre in una parte ( $\sigma_{\hat{y}}^2$ ), detta varianza spiegata, in quanto la variabilità della  $Y$  è dovuta alla dipendenza di  $Y$  dalla variabile  $X$ , e in una parte ( $\sigma_e^2$ ), detta varianza non spiegata, in quanto la variabilità della  $Y$  non dipende dalla variabile  $X$ , ma da altri fattori; si può introdurre un secondo indicatore, dato dal rapporto tra la varianza spiegata e la varianza totale, chiamato **coefficiente di determinazione**:

$$r^2 = \frac{\sigma_{\hat{y}}^2}{\sigma_y^2}$$

che indica quale frazione di varianza totale è dovuta alla dipendenza fra le variabili  $Y$  e  $X$ , ossia quale frazione della variazione della variabile  $Y$  è spiegata dalle variazioni della variabile  $X$ .

Sapendo che:

$$\sigma_y^2 = \sigma_{\hat{y}}^2 + \sigma_e^2$$

allora:

$$r^2 = \frac{\sigma_{\hat{y}}^2}{\sigma_y^2}$$

è evidente, quindi, che se la variabilità non spiegata è trascurabile,  $\sigma_e^2$  tende ad annullarsi ed  $r^2$  avrà un valore prossimo ad 1, mentre diverrà via via minore di 1 al diminuire dell'accordo tra la funzione calcolata e le osservazioni sperimentali.

Minore è la somma residua rispetto alla somma totale dei quadrati, maggiore sarà il valore del coefficiente di determinazione,  $r^2$ , il quale è un indicatore del livello di precisione con cui l'equazione ottenuta dall'analisi di regressione spiega la relazione tra le variabili. [2]

Un'altra metrica utile in ambito delle regressioni è l'errore quadratico medio (in inglese *Mean Squared Error*, MSE) che indica la discrepanza quadratica media fra i valori dei dati osservati ed i valori dei dati stimati:

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

La sua radice quadrata fornisce un ulteriore indice statistico, la cosiddetta radice dell'errore quadratico medio (in inglese *root-mean-square error*, RMSE). L'RMSE può essere anche calcolato come deviazione standard degli scarti. Da notare che l'MSE ed RMSE non sono quantità a-dimensionali,

bensì assumono l'unità di misura della grandezza considerata (RMSE) ed il suo quadrato (MSE).

### 3.2.3 Analisi dei residui

Esistono metodi utili per diagnosticare le violazioni delle ipotesi di regressione di base: questi si basano principalmente sullo studio dei residui del modello. Spesso infatti la retta di regressione è infatti una semplificazione della realtà e non coglie tutta la variabilità presente in un insieme di dati. [3]

Si definiscono i residui come:

$$e_i = y_i - \hat{y}_i, \quad i = 1, 2, \dots, n$$

dove  $y_i$  è il valore osservato e  $\hat{y}_i$  è il valore predetto.

Poiché un residuo può essere visto come la deviazione tra i dati e l'adattamento, è anche una misura della variabilità nella variabile di risposta non spiegata dal modello di regressione. [4]

Eventuali scostamenti dalle ipotesi sugli errori dovrebbero quindi manifestarsi nei residui. L'analisi grafica dei residui è un modo efficace per scoprire diversi tipi di inadeguatezze del modello, tra cui:

- se i residui hanno distribuzione normale (3.2.3.1);
- se le varieabili indipendenti sono correlate con l'errore (3.2.3.2);
- se la varianza dei residui è omogenea (3.2.3.3);
- se ci sono degli outliers che influenzano la pendenza della retta (3.2.3.4).

### 3.2.3.1 Distribuzione degli errori

La distribuzione normale degli errori può essere verificata attraverso un grafico dei quantili, detto anche q-q plot. In questa tipologia di grafico, i quantili teorici di una distribuzione Normale sono riportati sull'asse orizzontale. I quantili dei residui standardizzati sono invece riportati sull'asse verticale. L'idea è che se i residui hanno una distribuzione normale, i loro quantili dovrebbero coincidere con quelli della distribuzione normale. A livello visivo, questo significa che i punti dovrebbero disporsi lungo la *bisettrice*, indicata dalla retta presente nel grafico (figura TODO).

Nella pratica, non capita quasi mai che i punti si dispongano esattamente lungo la bisettrice. Per poter dire che gli errori hanno una distribuzione normale ci si accontenta quindi che i punti siano vicino alla linea presente nel grafico. Tuttavia in generale le stime sui coefficienti di regressione sono abbastanza robuste a violazioni della normalità distributiva dei residui.

### 3.2.3.2 Correlazione tra errore e variabili

Se una variabile esplicativa è correlata con il termine d'errore, è possibile utilizzare questa variabile esplicativa per predire quale sarà l'errore del modello di regressione. Questo in generale non è un buon segno, perché la componente di errore di un modello di previsione deve essere imprevedibile.

Per verificare la non correlazione tra la variabile indipendente ( $x$ ) e i residui è utile osservare un grafico di dispersione come quello riportato in figura TODO, in sull'asse orizzontale si mettono i valori della  $x$ , mentre sull'asse verticale i valori dei residui.

L'ipotesi è confermata se non è individuabile nessuna relazione tra le due variabili.

### 3.2.3.3 Omogeneità della varianza dei residui

Per verificare l’ipotesi di omogeneità delle varianze dei residui, è necessario creare un grafico a dispersione. I valori stimati della  $y$  si riportano sull’asse orizzontale delle  $x$ . Sull’asse verticale delle  $y$  invece si indicano i valori dei residui (figura 3.12).

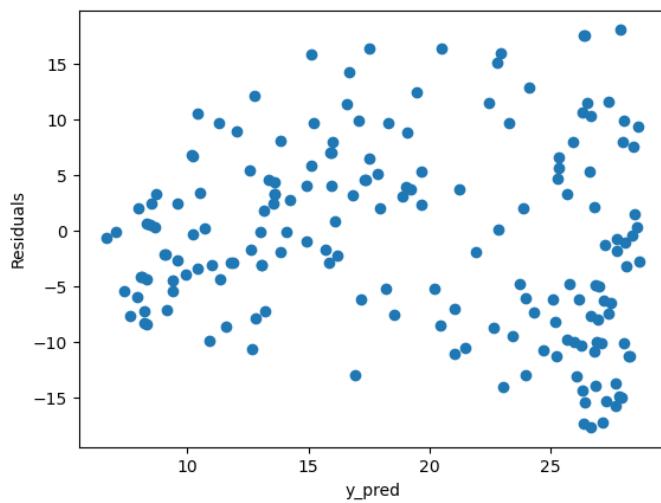


Figura 3.12: Esempio di distribuzione dei residui

Se c’è omogeneità della varianza dei residui, i punti saranno dispersi in modo simile sia nella parte sinistra che in quella destra del grafico. Questa proprietà se verificata prende il nome di **omoschedasticità**.

### 3.2.3.4 Influenza di outliers

Il grafico a dispersione tra valori predetti e residui permette di individuare anche i possibili outliers, ovvero i punti isolati nel grafico (quelli con residui maggiori). Tuttavia, per verificare se ci sono outliers in un modello di regressione, spesso si utilizzano altre tecniche (ad esempio eliminando i punti problematici tramite la distanza di Cook, descritta in 3.2.4.3, oppure

applicando stime robuste meno sensibili alle osservazioni problematiche, ad esempio con la funzione peso di Huber descritta in 3.2.4.2). Nel primo caso è utile anche provare a rifare le analisi di regressione escludendo le osservazioni potenzialmente problematiche e vedere se ci sono differenze nei coefficienti del modello.

Nei modelli di regressione infatti anche un singolo outlier può influenzare in maniera sostanziale la capacità di adattamento del modello ai dati, soprattutto se il campione non è molto numeroso.

### 3.2.4 Modelli di regressione

I modelli di regressione sono ampiamente utilizzati sia per la previsione o la descrizione dei dati che la stima e il controllo dei parametri.

#### 3.2.4.1 Regressione lineare

Si considera una funzione lineare a due variabili:

$$y = a + b * x$$

In questo caso si deve rendere minima la funzione:

$$\varphi(a, b) = \sum_{i=1}^n [y_i - (a + bx_i)]^2$$

Annullando le derivate parziali prime rispetto ad  $a$  e  $b$  si ha il sistema:

$$\begin{cases} \sum_{i=1}^n 2 [y_i - (a + bx_i)] (-1) = 0 \\ \sum_{i=1}^n 2 [y_i - (a + bx_i)] (-x_i) = 0 \end{cases}$$

che risolto, fornisce i valori dei parametri:

$$\begin{cases} \hat{a} = \bar{y} - b\bar{x} \\ \hat{b} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \end{cases}$$

dove  $\bar{x}$  e  $\bar{y}$  indicano le *medie aritmetiche*, rispettivamente di  $x_i$  e  $y_i$ .

La stima del parametro  $b$ , *coefficiente angolare* della funzione lineare, può essere rappresentato nella forma:

$$\hat{b} = \frac{\sum_{i=1}^n \frac{(x_i - \bar{x})(y_i - \bar{y})}{n}}{\sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n}}$$

dove il denominatore è la *varianza* di  $X$  ( $\sigma_X^2$ ), mentre il numeratore è detto *covarianza* di  $X$  e  $Y$  ( $\sigma_{XY}$ ) e misura la variabilità congiunta delle coppie  $(x_i, y_i)$  di valori corrispondenti rispetto al proprio valor medio; quindi, il coefficiente  $b$  della retta interpolante esprime la variabilità congiunta di  $X$  e  $Y$  rapportata alla variabilità della sola  $X$ .

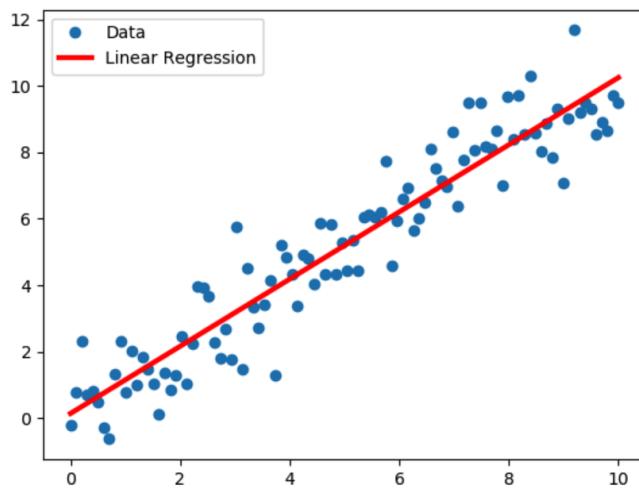


Figura 3.13: Esempio di regressione lineare

La precisione della retta calcolata dalla regressione lineare dipende dal grado di dispersione nei dati. Più i dati sono lineari, più il modello risulterà accurato.

### 3.2.4.2 Regressione lineare robusta (Huber)

La regressione Huber (in inglese Huber regression, anche detta regressione robusta) è una metodologia statistica per la stima dei parametri di un modello lineare in presenza di *outliers*.

Ci sono situazioni in cui si verifica presenza di valori anomali che influiscono sul modello di regressione, nel senso che possono avere una forte influenza sul metodo dei minimi quadrati, di fatto *deviando* troppo l'equazione di regressione nella loro direzione. Il metodo dei minimi quadrati, infatti, in questi casi ha lo svantaggio di avere la tendenza a essere dominato da questi valori — infatti sommando il quadrato dei residui ( $\sum_{i=1}^n a_i^2$  dove  $a_i$  è il residuo i-esimo), la media risulta troppo influenzata da pochi valori  $a_i$  particolarmente grandi.

Ci sono due modi per affrontare questa situazione:

- Scartare le osservazioni *scomode* (vedi regressione lineare avanzata 3.2.4.3);
- Applicare procedure di stime robuste in modo che siano meno sensibili alle osservazioni troppo influenti (figura 3.14).

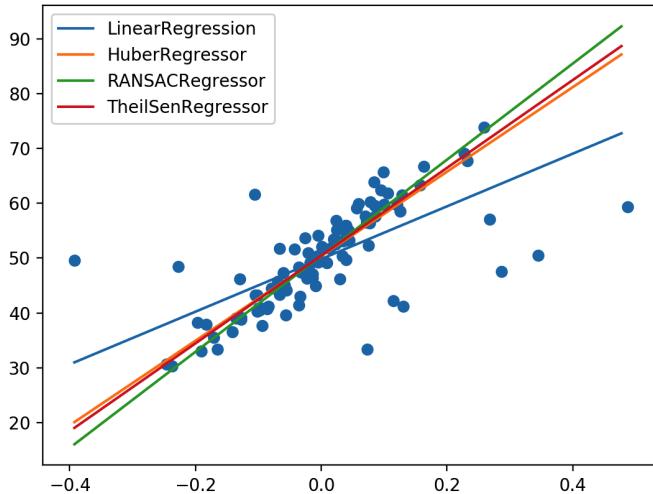


Figura 3.14: Comportamento di modelli di regressione robusta in presenza di outliers

Una delle funzioni di stima robusta, comunemente usata in diversi metodi di regressione per ridurre la sensibilità dei parametri alla presenza di outliers, è la **funzione di Huber**, che risulta quadratica per piccoli valori di  $x$ , e lineare per valori più grandi. È definita come:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{per } |a| \leq \delta \\ \delta(|a| - \frac{1}{2}\delta), & \text{altrimenti} \end{cases}$$

Dove la variabile  $a$  fa riferimento al residuo, cioè la differenza tra valore osservato e valore predetto ( $a = y - f(x)$ ).

### 3.2.4.3 Regressione lineare avanzata

Come accennato in 3.2.4.2, un'altra tecnica per la gestione di outlier è quella di applicare il modello sul dataset dopo aver rimosso i valori anomali. Esistono molte metriche su cui basarsi per rimuovere gli outlier da un set di

dati: un metodo che viene spesso utilizzato nella regressione è la **distanza di Cook**.

La distanza di Cook è una stima dell'*influenza* di una osservazione in un dataset, in termini di residuo (outlier) o di elevato *leverage*: è un riepilogo di quanto cambierebbe un modello di regressione nel caso in cui venga rimossa l'i-esima osservazione.

In presenza di outliers la distanza di Cook aumenta, e quindi questi dati ad alta influenza hanno un maggiore impatto sulle stime dei parametri della regressione.

La distanza di Cook [5] dell'osservazione  $i$  ( $\forall i = 1, \dots, n$ ) è definita come:

$$D_i = \frac{\sum_{j=1}^n (\hat{y}_j - \hat{y}_{j(i)})^2}{ps^2}$$

dove:

- $n$  è il numero di osservazioni;
- $\hat{y}_j$  è il valore predetto;
- $\hat{y}_{j(i)}$  è la risposta ottenuta escludendo l'i-esima osservazione.

Oppure, in modo equivalente:

$$D_i = \frac{e_i^2}{ps^2} \left[ \frac{h_i}{(1 - h_i)^2} \right]$$

dove:

- $e_i = y_i - \hat{y}_i$  è l'i-esimo residuo;
- $p$  è il numero di coefficienti della regressione;

- $s^2$  è l'errore quadratico medio (MSE);
- $h_i$  è il peso che l'i-esimo osservazione ha sul valore della regressione (*leverage*).

Un esempio di rilevazione grafica di outlier tramite distanza di Cook è riportato in figura 3.15.

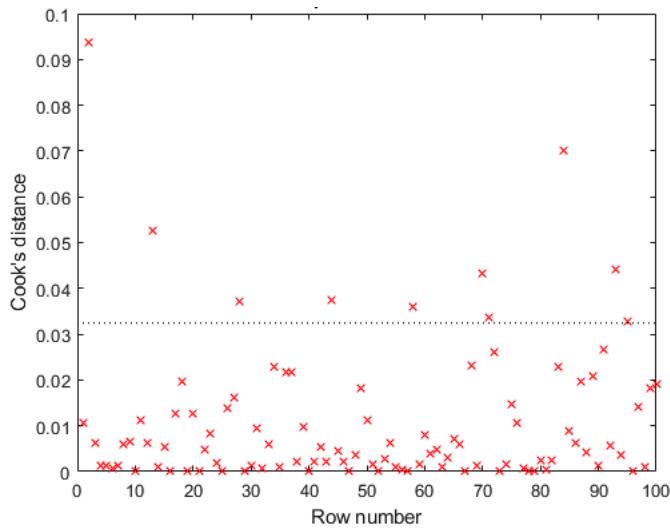


Figura 3.15: Riconoscimento di outlier tramite distanza di Cook

Vi sono diverse opinioni riguardo al valore di soglia di *cut-off*, oltre la quale un dato può essere considerato un outlier. In [6] viene proposta:

$$D_i > \frac{4}{n}$$

dove  $n$  è il numero di osservazioni. La distanza di Cook può anche essere utilizzata per individuare regioni dello spazio nelle quali sarebbe necessario effettuare una validazione, ad esempio acquisendo più dati.

### 3.2.4.4 Regressione Ridge

Nella statistica e nel Machine Learning, la regressione Ridge è un metodo di analisi di regressione che applica una fase di **regolarizzazione** al fine di migliorare l'accuratezza della previsione, prevenire l'*overfitting* e penalizzare la complessità del modello. Insieme al LASSO (vedi 3.2.4.5) è un modello di regressione che viene ripreso anche da tecniche di Boosting di Machine Learning.

Parlando di regolarizzazione in generale esistono due tipi di penalizzazione:

- **L1**: penalizza il valore assoluto dei coefficienti del modello (es. Lasso);
- **L2**: penalizza il quadrato del valore dei coefficienti del modello (es. Ridge).

La regressione Ridge usa la penalità L2: in pratica questo produce coefficienti piccoli, ma nessuno di loro è mai annullato (*feature shrinkage*).

Richiamando il metodo dei minimi quadrati (3.2.1) si deve minimizzare la somma dei quadrati dei residui (RSS):

$$\text{RSS} = \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2$$

Nella regressione Ridge si aggiunge anche un termine di penalità, ottenendo quindi:

$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2$$

Dove  $\lambda$  è un parametro di *tuning* che serve proprio a controllare l'effetto della penalità: un valore  $\lambda = 0$  infatti non avrà effetto sul risultato finale (l'equazione viene ricondotta a quella dei minimi quadrati), al contrario per  $\lambda \rightarrow \infty$  invece i coefficienti di regressione stimati tenderanno a zero poiché si darà molto peso alla penalità del modello. [7]

Il modello di Ridge Regression presenta dei vantaggi rispetto a quello dei minimi quadrati, soprattutto per quanto riguarda il *bias-variance trade-off*: in generale, quando c'è una relazione lineare tra i predittori e la variabile risposta, il modello dei minimi quadrati comporta poco bias ma alta varianza. Questo si traduce nel fatto che una piccola variazione nel training data può generare un cambiamento notevole nei coefficienti stimati; di contro la Ridge regression lavora bene nelle situazioni dove il modello dei minimi quadrati genera ampia varianza nelle stime. [8]

### 3.2.4.5 Regressione Lasso

Lo svantaggio della regressione Ridge è il fatto di considerare tutte le variabili per la predizione nel modello finale. Il termine di regolarizzazione  $\lambda \sum_{j=1}^p \beta_j^2$  tende ad assegnare ai coefficienti valori vicini allo zero, ma non perfettamente zero, a meno che  $\lambda = 0$ . Questo non crea problemi per l'accuratezza della predizione quanto per l'interpretazione delle varabili, soprattutto quando il numero delle variabili diventa alto.

La regressione Lasso (acronimo di *least absolute shrinkage and selection operator*, ovvero operatore di restringimento e selezione minimo assoluto) è un'alternativa alla regressione ridge utilizzata proprio per superare questo problema. L'unica differenza sta nel termine di regolarizzazione, ovvero:

$$\lambda \sum_{j=1}^p |\beta_j|$$

Per cui l'equazione del modello diventa:

$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| = RSS + \lambda \sum_{j=1}^p |\beta_j|$$

Anche nel caso di Lasso regression il parametro di regolarizzazione tende a stimare i valori dei coefficienti verso lo zero ma, a differenza della regressione ridge, la penalità  $\lambda \sum_{j=1}^p |\beta_j|$  costringe uno o più coefficienti ad essere esattamente zero per certi valori di  $\lambda$ . [8]

### 3.2.4.6 Regressione polinomiale

La regressione polinomiale è una generalizzazione della regressione lineare, infatti utilizza lo stesso metodo matematico della variante lineare, ma assume che la relazione di funzione che caratterizza i dati sia meglio descritta, anzichè da una retta, da un polinomio. In questo caso il metodo dei minimi quadrati può essere utilizzato anche per adattare una funzione polinomiale a un insieme di dati. Considerato un polinomio di grado  $k$ :

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_k x^k$$

In questo caso il sistema di equazioni da risolvere è:

$$\begin{cases} na_0 + a_1 \sum_{i=1}^n x_i + a_2 \sum_{i=1}^n x_i^2 + \dots + a_k \sum_{i=1}^n x_i^k = \sum_{i=1}^n y_i \\ a_1 \sum_{i=1}^n x_i + a_2 \sum_{i=1}^n x_i^2 + \dots + a_k \sum_{i=1}^n x_i^k = \sum_{i=1}^n x_i y_i \\ \dots \\ a_1 \sum_{i=1}^n x_i^k + a_2 \sum_{i=1}^n x_i^{k+1} + \dots + a_k \sum_{i=1}^n x_i^{2k} = \sum_{i=1}^n x_i^k y_i \end{cases}$$

che, risolto, permette di ricavare i parametri  $a_0, a_1, a_2, \dots, a_k$ .

I polinomi sono ampiamente utilizzati in situazioni in cui la risposta è curvilinea, poiché anche relazioni non lineari complesse possono essere adeguatamente modellate da polinomi su intervalli ragionevolmente piccoli delle  $x$ .

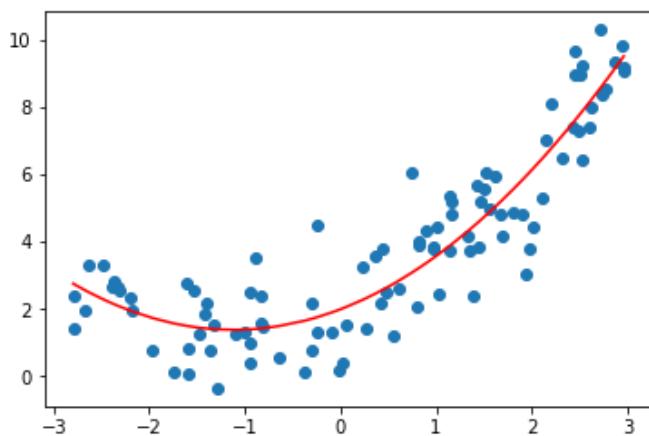


Figura 3.16: Esempio di regressione polinomiale

Ci sono diverse considerazioni importanti che emergono quando si adatta un polinomio in una variabile: una di queste riguarda la scelta dell'ordine del modello. Come regola generale, l'uso di polinomi di ordine elevato ( $k > 2$ ) dovrebbe essere evitato: un modello di ordine basso è quasi sempre preferibile a un modello di ordine elevato per ragioni di minore complessità, di coerenza con i dati e per evitare *overfitting*.

Come caso estremo, è sempre possibile trovare un polinomio di grado  $n-1$  ad  $n$  punti che risulti in un buon adattamento dei dati. Nella maggior parte dei casi, però, questo non farebbe nulla per migliorare la comprensione della funzione sconosciuta, né sarà probabilmente un buon predittore.

### 3.2.4.7 Regressione con Random Forest

La regressione Random Forest è un algoritmo di apprendimento supervisionato che utilizza il metodo di apprendimento *ensemble* per la regressione tramite alberi di decisione. Il metodo di apprendimento ensemble è una tecnica che combina le previsioni di più algoritmi di apprendimento automatico per effettuare una previsione più accurata rispetto a un singolo modello. [9]

In particolare, una foresta casuale (random forest) opera adattando una serie di alberi decisionali su vari sottocampioni del set di dati e utilizza la media dei risultati per migliorare l'accuratezza predittiva e controllare l'overfitting. In breve, l'algoritmo funziona esegue i seguenti passi:

1. Sceglie a caso  $k$  osservazioni dati dal training set;
2. Costruisce un albero decisionale associato a queste  $k$  osservazioni;
3. Sceglie il numero  $N$  di alberi da costruire e ripete i passaggi 1 e 2 per ciascuno;
4. Per una nuova osservazione, fa in modo che ciascuno degli  $N$  alberi preveda il valore di  $y$ , e assegna il nuovo punto alla media su tutti i valori  $y$  previsti.

Uno dei principali svantaggi degli alberi decisionali è che sono molto inclini a fare *overfitting*: funzionano bene sui dati di training, ma non sono così flessibili per fare previsioni su campioni invisibili. Sebbene ci siano soluzioni alternative per questo, come ad esempio ridurre gli alberi, questo riduce il loro potere predittivo. Generalmente sono modelli con bias medio e varianza alta, ma sono semplici e di facile interpretazione.

### 3.2.4.8 Regressione con Gradient Boosting

Il Gradient Boosting è una tecnica di Machine Learning che ha alla base la stima iterativa di alberi sui residui ottenuti ad ogni passo e l'aggiornamento in maniera adattiva delle stime. Questa tecnica riprende il concetto matematico del *Gradient Descent*, per cui lo split scelto sarà quello che favorisce l'avvicinamento al punto di minimo della funzione obiettivo.

Il Gradient Descent è un algoritmo di ottimizzazione che consente di individuare il valore minimo di una funzione di costo per sviluppare un modello con una previsione accurata.

L'algoritmo Gradient Boosting applicato a problemi di regressione può essere descritto nei seguenti passi:

1. Si inizializza il modello con un valore noto;
2. Si considera sul training set una *loss function*, ovvero una funzione differenziabile che esprima una valutazione della predizione (es.  $\frac{1}{2}(y_i - \hat{y}_i)$  dove  $y_i$  è l'osservazione e  $\hat{y}_i$  è la predizione);
3. Scelto un numero massimo, si itera modellando un albero di regressione seguendo una procedura di discesa del gradiente, in modo da minimizzare la *loss function*. L'albero ottenuto viene aggiunto alla sequenza di alberi già esistente, nel tentativo di correggere o migliorare l'output finale del modello.

### 3.2.4.9 Regressione con SVR

Un altro modello per la regressione è SVR (*support-vector regression*), basato sui modelli SVM (support-vector machines) di apprendimento supervisionato spesso utilizzati per il problema della classificazione. [10]

Rispetto alla regressione lineare e al metodo dei minimi quadrati, SVR presenta più flessibilità perchè consente di definire una soglia di accettazione dell'errore nel modello. Infatti l'algoritmo SVR si propone di minimizzare non l'errore quadratico, come nel metodo dei minimi quadrati, ma i coefficienti (nello specifico, la norma al quadrato del vettore dei coefficienti). La funzione obiettivo quindi diventa:

$$\min \frac{1}{2} \|\mathbf{w}\|^2$$

Il termine di errore invece è gestito nei vincoli, dove si imposta l'errore assoluto minore o uguale a un margine specificato, chiamato errore massimo ( $\varepsilon$ ):

$$|y_i - w_i x_i| \leq \varepsilon$$

Il tuning del parametro  $\varepsilon$  consente di ottenere la precisione desiderata del modello di regressione. Solitamente alla funzione obiettivo si aggiunge anche delle variabili di *slack* ( $\xi_i$ ), che indicano la deviazione dal margine di ciascun valore che supera la soglia  $\varepsilon$  (lo scopo è di minimizzarle il più possibile). La funzione obiettivo in questo caso diventa:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n |\xi_i|$$

dove  $C$  è un altro iperparametro regolabile: all'aumentare di  $C$ , aumenta anche la tolleranza per i punti al di fuori di  $\varepsilon$ . Quando invece  $C$  si avvicina a 0, la tolleranza si avvicina a 0 e l'equazione ricade nel caso semplificato.

I modelli di regressione SVM possono anche eseguire una regressione non lineare, applicando il *kernel trick* per mappare i dati in uno spazio di caratteristiche multidimensionale. Il tipo di kernel da usare nell'algoritmo è un altro parametro da definire nel modello. I kernel più comuni sono quello lineare, polinomiale (di grado  $n$ ) o rbf (basato su *funzione di base radiale*).

#### 3.2.4.10 Regressione con KernelRidge

La regressione Kernel Ridge (o KRR, *Kernel Ridge Regression*) è un altro modello che combina la regressione Ridge (descritta in 3.2.4.4) con il *kernel trick*, imparando così una funzione lineare nello spazio indotto dal rispettivo kernel e dai dati. Per i kernel non lineari, questo corrisponde a una funzione non lineare nello spazio originale. [11]

La forma del modello di regressione KRR è identica a quello basato su SVR (descritto in 3.2.4.9), ma vengono utilizzate diverse funzioni di *loss*: KRR minimizza l'errore quadratico mentre SVR minimizza i coefficienti in base alla soglia  $\varepsilon$ . Il modello KRR in genere risulta più veloce per dataset di medie dimensioni.

### 3.3 Esperimenti e risultati ottenuti

...

#### 3.3.1 NO<sub>2</sub>

...

#### 3.3.2 PM<sub>2.5</sub>

...

### 3.3.3 PM<sub>10</sub>

...

## 3.4 Validazione

Poiché l'adattamento del modello ai dati disponibili costituisce la base per molte delle tecniche utilizzate nel processo di sviluppo del modello (come la selezione delle variabili), si è tentati di concludere che un modello che si adatta bene ai dati avrà successo anche nel applicazione finale. Non è necessariamente così. Ad esempio, un modello potrebbe essere stato sviluppato principalmente per prevedere nuove osservazioni.

Non vi è alcuna garanzia che l'equazione che fornisce il miglior adattamento ai dati esistenti sarà un predittore di successo. Fattori influenti che erano sconosciuti durante la fase di costruzione del modello possono influenzare in modo significativo le nuove osservazioni, rendendo le previsioni quasi inutili.

La corretta convalida di un modello sviluppato per prevedere nuove osservazioni dovrebbe implicare una fase di validazione fatta sul campo prima di rilasciare il modello.

### 3.4.1 PM<sub>2.5</sub>

...

### 3.4.2 PM<sub>10</sub>

...

### **3.5 Discussione**

...

# Capitolo 4

## Interfaccia di calibrazione

### 4.1 Motivazioni

...

### 4.2 Tecnologie

...

#### 4.2.1 Backend

...

#### 4.2.2 Frontend

...

### 4.3 Funzionamento

...

## 4.4 Autenticazione

Keycloak è un'identità federata open source, sviluppata da Red Hat. Può essere utilizzata per gestire l'autenticazione di utenti e servizi in ambienti cloud e on-premise. I principali vantaggi di Keycloak sono la scalabilità, l'affidabilità e la flessibilità.

Keycloak include un server e un agente. L'agente è installato sulle applicazioni che richiedono l'autenticazione, mentre il server gestisce tutte le richieste di autenticazione. Quando un utente tenta di accedere a una applicazione protetta da Keycloak, l'agente verifica se l'utente è autenticato e, in caso affermativo, fornisce le credenziali appropriate all'applicazione.

## 4.5 CI e deploy automatico

Continuous integration è una metodologia di sviluppo software che prevede il continuo e costante integrazione dei cambiamenti effettuati dai developer all'interno di un codice sorgente.

La continuous integration ha lo scopo di evitare problemi di sincronizzazione tra gli sviluppatori, riducendo il numero di bug rilevati in fase di testing e aumentando la qualità del codice prodotto.

I principali vantaggi della continuous integration sono:

- riduzione del numero di bug rilevati in fase di testing;
- aumento della qualità del codice prodotto;
- maggiore sincronizzazione tra gli sviluppatori;
- minor rischio di collisioni tra i cambiamenti effettuati dagli sviluppatori.

Jenkins è uno strumento open source di continuous integration. Jenkins permette di automatizzare il processo di integrazione dei cambiamenti effettuati dai developer.

tuati dai developer all'interno di un codice sorgente, eseguendo una serie di controlli per verificarne la correttezza.

Jenkins può essere utilizzato per gestire una varietà di progetti, tra cui sviluppo software, testing, build, deployment e automazione dei processi.

## **Conclusioni e sviluppi futuri**

...

# Bibliografia

- [1] E. Belluco, *Excel per la statistica*. Franco Angeli, 2005.
- [2] M. S. Paoletta, *Linear Models and Time-Series Analysis*. Wiley, 2019.
- [3] P. Pozzolo, *Analisi dei residui del modello di regressione lineare*. 2020.
- [4] D. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. Wiley, 2012.
- [5] R. D. Cook, *Detection of Influential Observation in Linear Regression*. Taylor and Francis, Ltd., 1977.
- [6] J. Fox, *Applied Regression Analysis and Generalized Linear Models*. Sage Publications, 2015.
- [7] G. Spanò, *Lasso vs Ridge Regression*. 2017.
- [8] A. Felice, *Applicazione di modelli di Machine Learning ad Albero Decisionale con R per il Credit Scoring nel settore elettronico*. 2021.
- [9] T. K. Ho, *Random decision forests*. 1995.
- [10] C. Cortes, *Support-Vector Networks*. 1995.
- [11] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.