



Analisi dataset relativo all'inquinamento dell'aria

Analisi con Python

Autori:

Chiara Amalia Caporusso
Margherita Galeazzi
Simone Scalella
Zhang Yihang



Sommario

1	Introduzione.....	3
2	Dataset.....	3
3	Fase di ETL.....	5
4	Analisi descrittive	6
5	Distribuzioni.....	8
6	Analisi temporali.....	12
6.1	BoxPlot.....	13
6.2	StripPlot	14
7	Analisi geografiche	15
7.1	Correlazioni.....	16
8	Clustering.....	18
8.1	Fase di ETL per il clustering.....	18
8.2	Clustering bidimensionale	19
8.3	PCA.....	24
8.4	DBSCAN.....	26
9	Classificazione.....	27
9.1	Classificazione binaria in base a PM10	27
9.1.1	Classificazione con Grid Search	31
9.1.2	Testing del modello	33
9.2	Classificazione multi-classe in base all' NO2	35
9.2.1	Classificazione con GridSearch	38
9.2.2	Testing del modello	40
10	Serie temporale	41
10.1	ETL	42
10.2	Analisi della serie	42
10.3	Stazionarietà.....	42
10.4	Autocorrelazione e autocorrelazione parziale	43
10.5	Modello ARIMA	44
10.6	Predizione in-sample	45
10.7	Metriche di valutazione.....	47
10.7.1	MAE	47
10.7.2	MAPE	48
10.7.3	MSE.....	48
10.7.4	R ²	48
10.8	Predizione out-sample.....	48



10.9 Miglioramenti	49
10.9.1 Stazionarietà	50
10.9.2 Autocorrelazione e autocorrelazione parziale	50
10.9.3 Modello ARIMA	51
10.9.4 Predizione in-sample	51
10.9.5 Predizione out-sample.....	52



1 Introduzione

Il seguente progetto ha come scopo l'analisi di un dataset contenente informazioni relative ad agenti inquinanti, nel periodo di tempo compreso tra il 2008 e il 2018, presenti nella città di Madrid. I dati presenti nel dataset sono stati raccolti tramite delle stazioni per il monitoraggio della qualità dell'aria. È possibile scaricare il dataset dal seguente link:

<https://www.kaggle.com/datasets/decide-soluciones/air-quality-madrid>

Per realizzare questo progetto e tutte le analisi in esso contenute faremo uso di un linguaggio di programmazione open source ampiamente utilizzato nell'ambito della Data Science, stiamo parlando di Python. Il suo utilizzo è dovuto al numero di librerie messe a disposizione per poter eseguire diversi tipi di task. Utilizzeremo librerie quali Pandas, Seaborn, Matplotlib, Sklearn e Statsmodel, che sono gratuite e vengono mantenute sempre aggiornate. Python è un linguaggio cross Platform e ha una sintassi semplificata. Tutte queste caratteristiche rendono Python un linguaggio molto importante e molto usato nella Data Science. In questo progetto faremo uso delle librerie sopracitate e anche di altre, al fine di effettuare operazioni di estrazione, pulizia e visualizzazione dei dati. Realizzeremo task di Data mining e studi specifici su serie temporali d'interesse.

2 Dataset

Il dataset che andremo ad utilizzare è un dataset molto ricco di informazioni, ogni file, eccetto l'anno 2018, contiene oltre 200.000 righe. Quindi, abbiamo deciso di lavorare su una porzione dei file, prendendo in considerazione l'intervallo temporale che va dal 2008 al 2018. I campi che compongono il dataset sono:

- **Date**: questo campo contiene la data relativa al momento in cui è stato fatto il rilevamento. Il formato della data è composto dall'anno, il mese, il giorno, e l'ora.
- **BEN**: questo campo contiene il rilevamento del benzene. Il livello di benzene è misurato in $\mu\text{g}/\text{m}^3$. Il benzene è un irritante per gli occhi e la pelle, lunghe esposizioni possono causare diversi tipi di cancro, leucemia e anemie. Il benzene è considerato un cancerogeno per l'uomo di gruppo 1 dalla IARC.
- **EBe**: questo campo contiene il rilevamento dell'etilbenzene. Il livello di etilbenzene è misurato in $\mu\text{g}/\text{m}^3$. L'esposizione a lungo termine può causare problemi all'udito o ai reni e l'IARC ha concluso che l'esposizione a lungo termine può produrre il cancro.
- **CO**: questo campo contiene il rilevamento del monossido di carbonio. Il livello di monossido di carbonio è misurato in mg/m^3 . L'avvelenamento da monossido di carbonio comporta mal di testa, vertigini e confusione in brevi esposizioni e può provocare perdita di coscienza, aritmie, convulsioni o persino la morte a lungo termine.
- **NMHC**: questo campo contiene il rilevamento di idrocarburi non metanici (composti organici volatili). Il livello di idrocarburi non metanici è misurato in mg/m^3 . L'esposizione prolungata ad alcune di



queste sostanze può causare danni al fegato, ai reni e al sistema nervoso centrale. Si sospetta che alcuni di loro causino il cancro negli esseri umani.

- **NO:** questo campo contiene il rilevamento dell'ossido nitrico. Il livello di ossido nitrico è misurato in $\mu\text{g}/\text{m}^3$. Questo è un gas altamente corrosivo generato, tra l'altro, dai veicoli a motore e dai processi di combustione del carburante.
- **NO₂:** questo campo contiene il rilevamento del biossido di azoto. Il livello di biossido di azoto è misurato in $\mu\text{g}/\text{m}^3$. L'esposizione a lungo termine è causa di malattie polmonari croniche e sono dannose per la vegetazione.
- **O₃:** questo campo contiene il rilevamento dell'ozono. Il livello di ozono è misurato in $\mu\text{g}/\text{m}^3$. Livelli elevati possono produrre asma, bronchite o altre malattie polmonari croniche in gruppi sensibili o lavoratori all'aperto.
- **PM10:** questo campo contiene il rilevamento del materiale particolato aerodisperso con particelle inferiori a 10 μm . Il livello di materiale particolato aerodisperso è misurato in $\mu\text{g}/\text{m}^3$. Anche se non possono penetrare nell'alveolo, possono comunque penetrare attraverso i polmoni e colpire altri organi. L'esposizione a lungo termine può causare cancro ai polmoni e complicazioni cardiovascolari.
- **PM25:** questo campo contiene il rilevamento del materiale particolato aerodisperso, di diametro maggiore rispetto al precedente. Il livello di materiale particolato aerodisperso è misurato in $\mu\text{g}/\text{m}^3$. Le dimensioni di queste particelle consentono loro di penetrare nelle regioni di scambio gassoso dei polmoni (alveoli) e persino di entrare nelle arterie. È stato dimostrato che l'esposizione a lungo termine è correlata al basso peso alla nascita e all'ipertensione nei neonati.
- **SO₂:** questo campo contiene il rilevamento dell'anidride solforosa. Il livello di anidride solforosa è misurato in $\mu\text{g}/\text{m}^3$. Alti livelli di anidride solforosa possono produrre irritazione della pelle e delle membrane e peggiorare l'asma o le malattie cardiache nei gruppi sensibili.
- **TCH:** questo campo contiene il rilevamento del livello totale di idrocarburi. Questo livello totale di idrocarburi è misurato in mg/m^3 . Questo gruppo di sostanze può essere responsabile di diverse malattie del sangue, del sistema immunitario, del fegato, della milza, dei reni o dei polmoni.
- **TOL:** questo campo contiene il rilevamento del toluene (metilbenzene). Il livello di toluene è misurato in $\mu\text{g}/\text{m}^3$. L'esposizione a lungo termine a questa sostanza (presente anche nel fumo di tabacco) può causare complicazioni renali o danni permanenti al cervello.
- **Station:** questo campo contiene il codice identificativo della stazione che ha effettuato quel rilevamento. Nel dataset abbiamo una tabella `Station`, all'interno della quale sono contenuti i codici delle stazioni più altre informazioni molto importanti. Queste informazioni sono il nome della stazione, il nome del luogo dov'è situata e le indicazioni geografiche di questo luogo.
- **CH₄:** questo campo contiene il rilevamento del livello di metano. Il livello di metano è misurato in mg/m^3 . Questo gas è un asfissiante, che sostituisce l'ossigeno di cui gli animali hanno bisogno per respirare. Il metano può provocare vertigini, debolezza, nausea e perdita di coordinazione.
- **MXY:** questo campo contiene il rilevamento del livello di m-xilene. Il livello di m-xilene è misurato in $\mu\text{g}/\text{m}^3$. Gli xileni possono influenzare non solo l'aria, ma anche l'acqua e il suolo e una lunga esposizione



a livelli elevati di xilene può provocare malattie che colpiscono il fegato, i reni e il sistema nervoso (in particolare la memoria e la reazione allo stimolo alterata).

- **PXY:** questo campo contiene il rilevamento del livello di p-xilene. Il livello di p-xilene è misurato in $\mu\text{g}/\text{m}^3$. Vedere MXY per gli effetti dell'esposizione allo xilene sulla salute.
- **OXY:** questo campo contiene il rilevamento del livello di o-xilene. Il livello di o-xilene è misurato in $\mu\text{g}/\text{m}^3$. Vedere MXY per gli effetti dell'esposizione allo xilene sulla salute.
- **NOX:** questo campo contiene il rilevamento del livello di ossidi di azoto. Il livello di ossidi di azoto è misurato in $\mu\text{g}/\text{m}^3$. Colpiscono il sistema respiratorio umano peggiorando l'asma o altre malattie, e sono responsabili del colore bruno-giallastro dello smog fotochimico.

3 Fase di ETL

Per poter proseguire con il nostro progetto abbiamo dovuto eseguire una fase di ETL (Extraction, Transformation, Load).

Come prima cosa abbiamo analizzato i dati visivamente, per capire se tutti i file del nostro dataset erano conformi tra di loro e non presentavano anomalie. Il dataset possiede un file CSV per ogni anno, quindi, per una maggiore comodità abbiamo bisogno di unificare le tabelle.

Step successivo è stato quello di controllare le strutture delle tabelle che vogliamo unificare. Abbiamo osservato come non tutte le tabelle hanno le stesse colonne, di conseguenza abbiamo approfondito l'analisi su quelle colonne. Le colonne problematiche sono la CH4, NOX, OXY, PXY, MXY e NO.

Per le colonne PXY, MXY e OXY osserviamo come siano presenti solo all'interno delle tabelle 2008, 2009, 2010. Inoltre, sono colonne poco valorizzate con tantissimi valori vuoti. Quindi, abbiamo deciso di eliminarli.

Per la colonna CH4, abbiamo osservato che è presente solo all'interno delle tabelle 2017, 2018. Inoltre, la tabella del 2018 è la meno popolata. Quindi, siccome stiamo lavorando su 11 tabelle, abbiamo deciso di eliminarla, in quanto rappresenta un valore con una presenza marginale nel dataset complessivo.

La colonna NOX è una misura che è presente solo in alcune tabelle del nostro intervallo, inoltre, rappresenta la somma degli ossidi di azoto, nel nostro dataset corrisponde, all'incirca, alla somma del monossido di azoto (NO) e del biossido di azoto (NO₂). Questo tipo di dipendenze creano problemi nei task di classificazione e di regressione, quindi abbiamo deciso di eliminarla.

L'ultima colonna che abbiamo analizzato è quella relativa a NO, essa non è presente all'interno delle tabelle del 2008, 2009, 2010, in tutte le altre sì. Questo dato è molto presente nel dataset, quindi abbiamo deciso di aggiungere questa colonna alle tabelle che non la possedevano. Per risolvere il problema dei valori nulli abbiamo deciso di utilizzare la media troncata. Infatti, le tre tabelle avrebbero avuto una colonna piena di valori nulli, e questo la rendeva inutile per qualsiasi tipo di analisi. Di conseguenza ci siamo calcolati la media troncata di quel valore utilizzando tutti i valori di quella colonna ottenuti dall'unione di tutti i dataset, tranne quelle tre tabelle. La media troncata è stata calcolata con una percentuale di troncamento pari al 20 %, con questa soluzione evitiamo il problema degli outlier nel calcolo della media. Di seguito riportiamo l'immagine del codice implementato:

```
# adesso devo estrarre la media troncata del valore NO che andremo a sostituire nella tabella dei 3 anni precedenti
NO = madrid1118[['NO']]

#convert pandas dataframe to numpy array
arr = NO.to_numpy()
NO_list = [item for sublist in list(arr) for item in sublist]

💡Calcolo la media troncata
newValue = stats.trim_mean(NO_list,0.1)
```



Al termine di questa fase di ETL abbiamo proceduto con l'unificazione dei dataset. Di seguito riportiamo l'immagine del codice implementato.

```
# Inserisco il nuovo valore all'interno del dataset madrid8910 e ordino le colonne
madrid8910['NO'] = newValue
madrid8910 = madrid8910.loc[:, ['date','BEN','CO','EBE','NMHC','NO','NO_2','O_3','PM10','PM25','SO_2','TCH','TOL','station']]
madrid8910.head()



|   | date                | BEN  | CO   | EBE | NMHC | NO        | NO_2       | O_3       | PM10      | PM25 | SO_2  | TCH  | TOL  | station  |
|---|---------------------|------|------|-----|------|-----------|------------|-----------|-----------|------|-------|------|------|----------|
| 0 | 2008-06-01 01:00:00 | NaN  | 0.47 | NaN | NaN  | 11.820078 | 83.089996  | 16.990000 | 16.889999 | 10.4 | 8.98  | NaN  | NaN  | 28079001 |
| 1 | 2008-06-01 01:00:00 | NaN  | 0.59 | NaN | NaN  | 11.820078 | 94.820000  | 17.469999 | 19.040001 | NaN  | 5.85  | NaN  | NaN  | 28079003 |
| 2 | 2008-06-01 01:00:00 | NaN  | 0.55 | NaN | NaN  | 11.820078 | 75.919998  | 13.470000 | 20.270000 | NaN  | 6.95  | NaN  | NaN  | 28079004 |
| 3 | 2008-06-01 01:00:00 | NaN  | 0.36 | NaN | NaN  | 11.820078 | 61.029999  | 23.110001 | 10.850000 | NaN  | 5.96  | NaN  | NaN  | 28079039 |
| 4 | 2008-06-01 01:00:00 | 1.68 | 0.80 | 1.7 | 0.3  | 11.820078 | 105.199997 | 12.120000 | 37.160000 | 21.9 | 10.92 | 1.53 | 6.67 | 28079006 |


```
Adesso facciamo l'unione di tutti i dataset
allMadrid = pd.concat([madrid8910,madrid1118])
allMadrid.head()
```



|   | date                | BEN | CO   | EBE | NMHC | NO        | NO_2      | O_3       | PM10      | PM25 | SO_2 | TCH | TOL | station  |
|---|---------------------|-----|------|-----|------|-----------|-----------|-----------|-----------|------|------|-----|-----|----------|
| 0 | 2008-06-01 01:00:00 | NaN | 0.47 | NaN | NaN  | 11.820078 | 83.089996 | 16.990000 | 16.889999 | 10.4 | 8.98 | NaN | NaN | 28079001 |
| 1 | 2008-06-01 01:00:00 | NaN | 0.59 | NaN | NaN  | 11.820078 | 94.820000 | 17.469999 | 19.040001 | NaN  | 5.85 | NaN | NaN | 28079003 |
| 2 | 2008-06-01 01:00:00 | NaN | 0.55 | NaN | NaN  | 11.820078 | 75.919998 | 13.470000 | 20.270000 | NaN  | 6.95 | NaN | NaN | 28079004 |


```

2 - ETL: secondo passo

4 Analisi descrittive

Durante lo step successivo andremo a realizzare delle analisi di tipo descrittivo. Le analisi descrittive, così come suggerisce il nome, sintetizzano o descrivono i dati e creano dei risultati che sono interpretabili dagli esseri umani. Di seguito riportiamo le analisi che abbiamo effettuato con relativa descrizione.

Nel dataset utilizzato dobbiamo lavorare con tutto un insieme di agenti inquinanti, quindi, la prima analisi che abbiamo fatto è stata quella di calcolare, per ogni anno, il valore medio di ogni agente inquinante. Lo step successivo è stato quello di andare a cercare dei valori di soglia annuali, da poter utilizzare per capire quale agente inquinante fosse più presente e quindi più pericoloso e interessante. I valori di soglia sono stati cercati sui siti dell'arpa, Wikipedia, e altri documenti che riportano le leggi in vigore sul territorio europeo che descrivono le normative relative alla presenza di tali agenti inquinanti nell'aria. Le soglie individuate hanno la stessa dimensione delle misure del dataset. Rappresentano valori importanti per la salvaguardia della salute delle persone. Purtroppo, di alcuni agenti chimici non siamo riusciti a trovare dei valori di soglia, in quanto sono ancora in fase di studio e realizzazione, oppure, non esiste ancora una normativa europea che prevede dei valori di soglia all'interno delle grandi città. Di seguito riportiamo i risultati delle prime analisi.

mean value BEN	tollerance BEN	mean value EBE	tollerance EBE	mean value CO	tollerance CO	mean value PM25	tollerance PM25	mean value PM10	tollerance PM10
2018 0.5558643116351426	5.0	0.30053100835570856	5.0	0.3444331835704493	10.0	7.7180972927242	25.0	13.519682479477662	40.0
2017 0.5956912391833159	5.0	0.39484901580666504	5.0	0.3639084667461865	10.0	9.950271995079099	25.0	19.94766610011619	40.0
2016 0.6323764709124132	5.0	0.37440054367326386	5.0	0.35492855211554114	10.0	10.3856505034026646	25.0	19.14971718467549	40.0
2015 0.757126052902769	5.0	0.495404331730131	5.0	0.36664056669061856	10.0	11.474925709601393	25.0	21.02113971482797	40.0
2014 0.6820315200792075	5.0	0.47067825553699033	5.0	0.36815095463652686	10.0	10.618973381834978	25.0	19.28492680541841	40.0
2013 0.7131326573209409	5.0	0.8117333512566551	5.0	0.3287667986702723	10.0	9.93628185840708	25.0	18.372364071919415	40.0
2012 0.8291568817584438	5.0	0.9520550891903181	5.0	0.35509885996104507	10.0	11.817211103442988	25.0	23.600717297163243	40.0
2011 0.8152153975329459	5.0	0.9702249263644471	5.0	0.36781241971905	10.0	12.2803404645357906	25.0	23.284602579439518	40.0
2010 0.773164584176596	5.0	1.0754774492684233	5.0	0.35717645811854815	10.0	11.991911478159166	25.0	21.840783004728475	40.0
2009 0.7577792386210384	5.0	1.2206809023540703	5.0	0.39361716824984494	10.0	12.805031529120837	25.0	24.522735158154028	40.0
2008 0.892794010317489	5.0	1.2732848575010212	5.0	0.4120522415797601	10.0	14.40267834472691	25.0	26.611135257005163	40.0

Figura 3 - valori medi annuali e soglie BEN, EBE, CO, PM25, PM10

	mean value O_3	tollerance O_3	mean value NO_2	tollerance NO_2	mean value NO	tollerance NO	mean value SO_2	tollerance SO_2
2018	44.86356713026542	120.0	38.63202946560895	40.0	19.893252549908464	40.0	5.388721751906528	20.0
2017	48.31968664794023	120.0	41.58600435271327	40.0	23.41512448281635	40.0	6.852585740124189	20.0
2016	49.072237165629794	120.0	38.55880238143173	40.0	22.053956110328166	40.0	7.690954744661312	20.0
2015	50.582213302186325	120.0	40.98755298005316	40.0	26.75605469217691	40.0	6.882333310317844	20.0
2014	51.64660053747093	120.0	35.04177304761085	40.0	19.952542145978562	40.0	5.022377526078425	20.0
2013	50.015460618096476	120.0	34.71264131453603	40.0	20.17230330738183	40.0	4.4200758882373234	20.0
2012	41.16777969412861	120.0	38.65991175573683	40.0	24.75749865395038	40.0	4.3740281451455925	20.0
2011	44.98963293732507	120.0	44.87843525983914	40.0	28.133595759592325	40.0	6.9265800165209575	20.0
2010	47.65975143033252	120.0	44.27004817335597	40.0	11.82007782455046	40.0	9.702223905922294	20.0
2009	44.69399855537345	120.0	54.34899234934374	40.0	11.820077824550461	40.0	10.281263599254295	20.0
2008	39.03918540990447	120.0	55.357186390244046	40.0	11.820077824550461	40.0	10.564374614593264	20.0

Figura 4 - valori medi annuali e soglie O3, NO2, NO, SO2

Nelle tabelle abbiamo inserito in una colonna il valore medio, e nella colonna affianco la tolleranza. Abbiamo utilizzato tre colori per identificare un certo tipo di valore. La **legenda** è la seguente:

- Con il colore giallo abbiamo evidenziato i valori medi che sono maggiori o uguali rispetto a un terzo della tolleranza.
- Con il colore arancione abbiamo evidenziato i valori medi che sono maggiori o uguali rispetto alla metà della tolleranza.
- Con il colore rosso abbiamo evidenziato i valori medi che sono maggiori o uguali rispetto alla tolleranza.

I valori più interessanti da studiare sono quelli relativi al biossido di azoto, ossido nitrico, PM10, PM25 e anidride solforosa. Osserviamo come il biossido di azoto sia un agente inquinante che negli anni ha superato più volte il limite. Tutti gli altri non superano mai le soglie, però, spesso assumono un valore significativo. Tutti gli altri agenti inquinanti saranno comunque presi in considerazione per successive analisi e task.

I cinque agenti su cui vogliamo eseguire approfondimenti vengono misurati tutti con la stessa dimensione, quindi, una volta calcolato il valore medio, dell'intervallo temporale, per ogni agente, andiamo a realizzare un grafico a torta. La seguente misura mostra i risultati ottenuti.

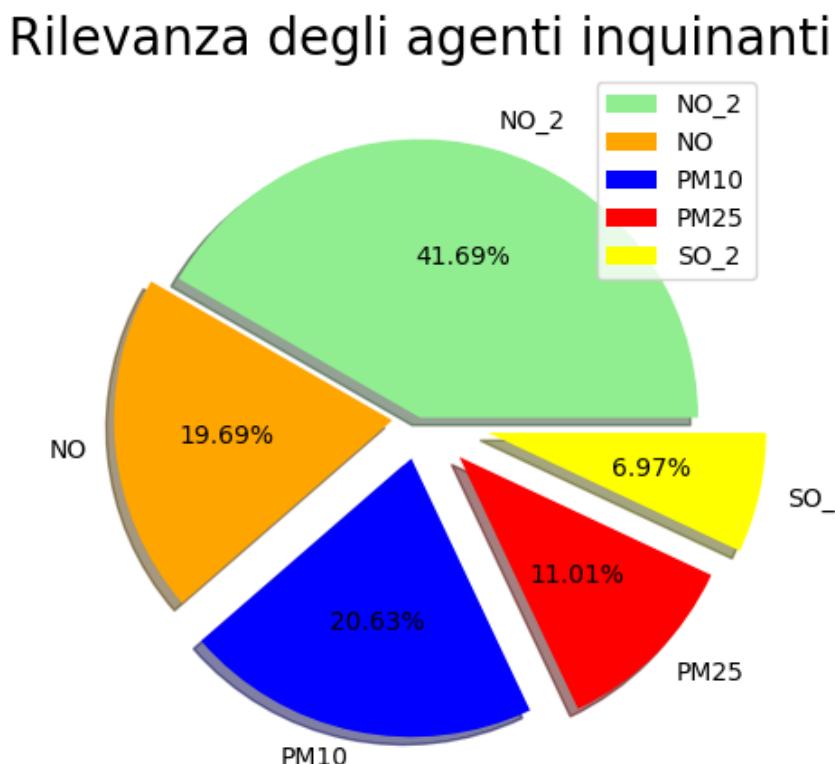


Figura 5 - rilevanza degli agenti inquinanti

Nell'analisi abbiamo suddiviso gli agenti inquinanti per colore, in modo da rendere più evidente la loro percentuale. Abbiamo comunque inserito una legenda che riporta le varie associazioni colore - inquinante. Da questo diagramma osserviamo come l'agente inquinante più presente sia il biossido di azoto, seguito da PM10, NO, PM25 e SO₂. Questo conferma il risultato precedente, infatti abbiamo già osservato come il NO₂ sia un inquinante che spesso supera il valore di soglia. Percentuali significative corrispondono al PM10 e al NO, i quali, insieme, raggiungono il 40% del totale.

Con una successiva analisi vogliamo esplorare la relazione che c'è tra il biossido e il monossido di azoto, con la seguente immagine riportiamo i risultati ottenuti.

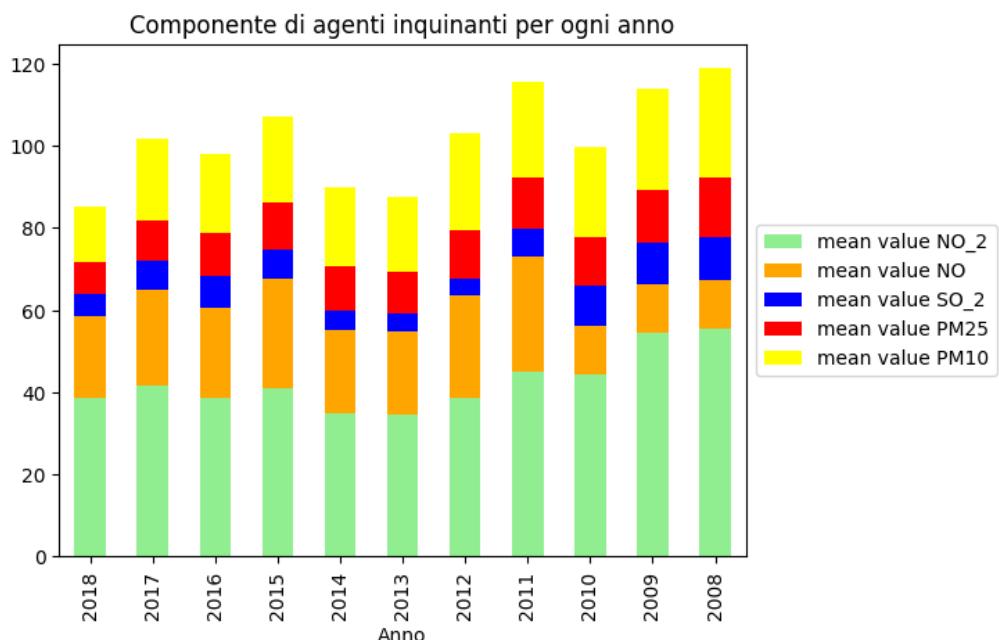


Figura 6 - Componente di agenti inquinanti per ogni anno

Continuando a lavorare con i valori medi degli agenti inquinanti abbiamo deciso di analizzare, anno per anno, la presenza delle varie componenti. Il diagramma migliore per realizzare quest'analisi è il diagramma a barre, dove ogni barra rappresenta l'intero totale delle medie, suddivisa per agente inquinante. Sull'asse delle ascisse abbiamo messo gli anni relativi all'intervallo d'interesse e sull'asse delle ordinate abbiamo messo i valori. Osserviamo come, durante ogni anno, l'agente inquinante più presente è sempre il biossido di azoto, i due elementi che occupavano il secondo e terzo posto, cioè, il monossido di azoto e il PM10. Osserviamo come il monossido di azoto aveva un valore mediamente basso durante il 2008, e poi è cresciuto nel tempo, arrivando ad assumere valori mediamente più alti nel 2018. Il PM10, invece, segue un andamento inverso, cioè, parte con un valore mediamente alto nel 2008 e finisce con un valore mediamente più basso nel 2018. Anche il PM25 si riduce leggermente durante gli anni. Infine, l'anidride solforosa prima si abbassa e poi ricresce, rimane comunque l'agente inquinante meno presente rispetto agli altri.

5 Distribuzioni

Lo step successivo è stato quello di valutare la distribuzione della concentrazione degli agenti inquinanti più significativi, presenti all'interno del dataset. Di seguito riportiamo i risultati della prima analisi.

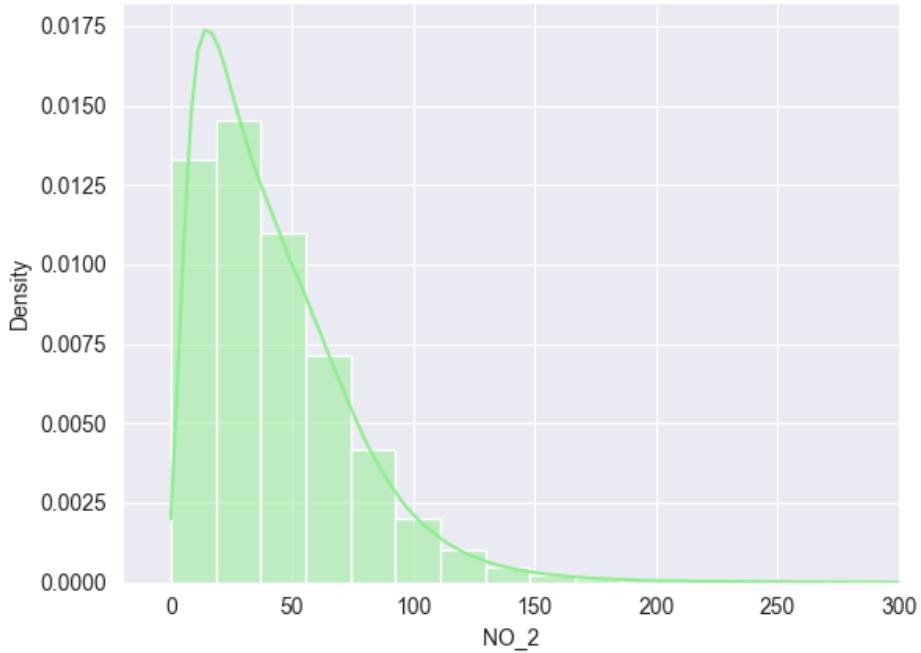


Figura 7.1 - distribuzione biossido di azoto

Da quest'analisi osserviamo come il biossido di azoto, non solo assume valori superiori alla soglia limite annuale, ma ci sono dei rilevamenti che riportano delle misure molte alte, con valori che superano il doppio della soglia. Fortunatamente, solo molti meno rispetto ai valori che sono inferiori alla soglia. La maggioranza delle misure ha un valore che è inferiore, o molto inferiore rispetto alla soglia di sicurezza annuale.

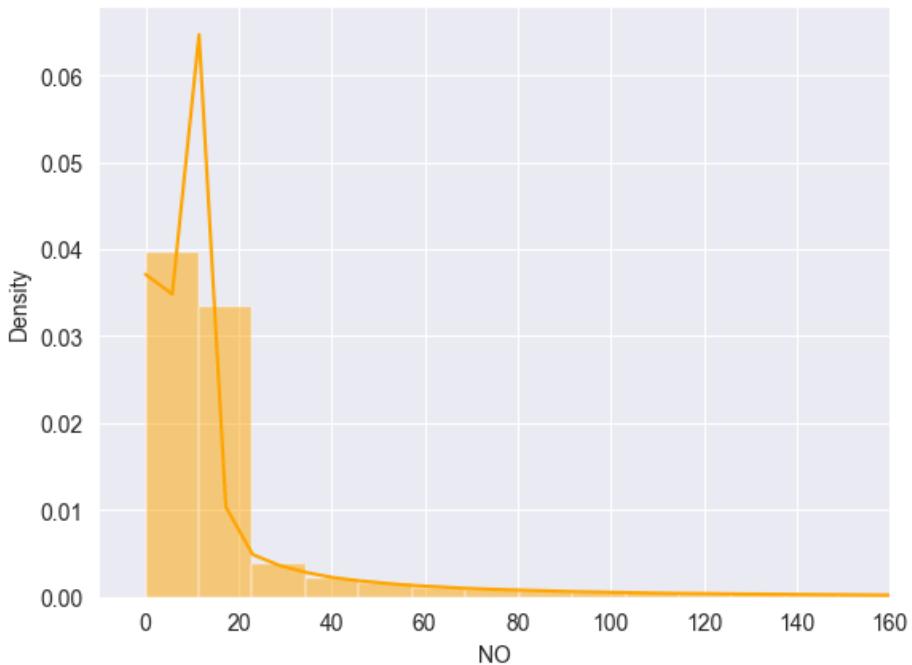


Figura 7.2 - distribuzione monossido di azoto

Questa è l'analisi sulla distribuzione del monossido di azoto. Quest'agente inquinante non supera il valore della soglia di sicurezza, però, per alcuni anni ha avuto un valore significativo, superiore rispetto alla metà del valore di soglia. La distribuzione conferma questa cosa, infatti, quasi la maggior parte delle misurazioni

hanno un valore che all'incirca è pari a 20. Ricordiamo che il valore di soglia era 40, il valore a cui facciamo riferimento è 20, cioè, la metà.

Di seguito carichiamo il risultato sull'analisi dell'anidride solforosa.

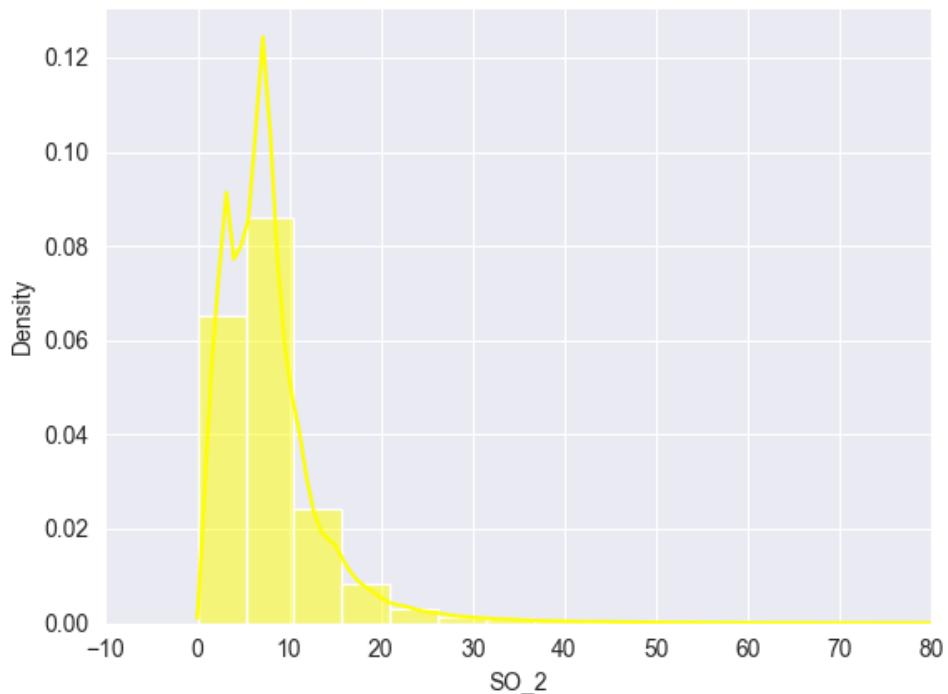


Figura 7.3 - distribuzione anidride solforosa

Osserviamo come la maggior parte delle misurazioni hanno un valore al di sotto delle fasce di soglia, infatti, con l'analisi precedente avevamo già constatato come questo agente inquinante sia poco presente.

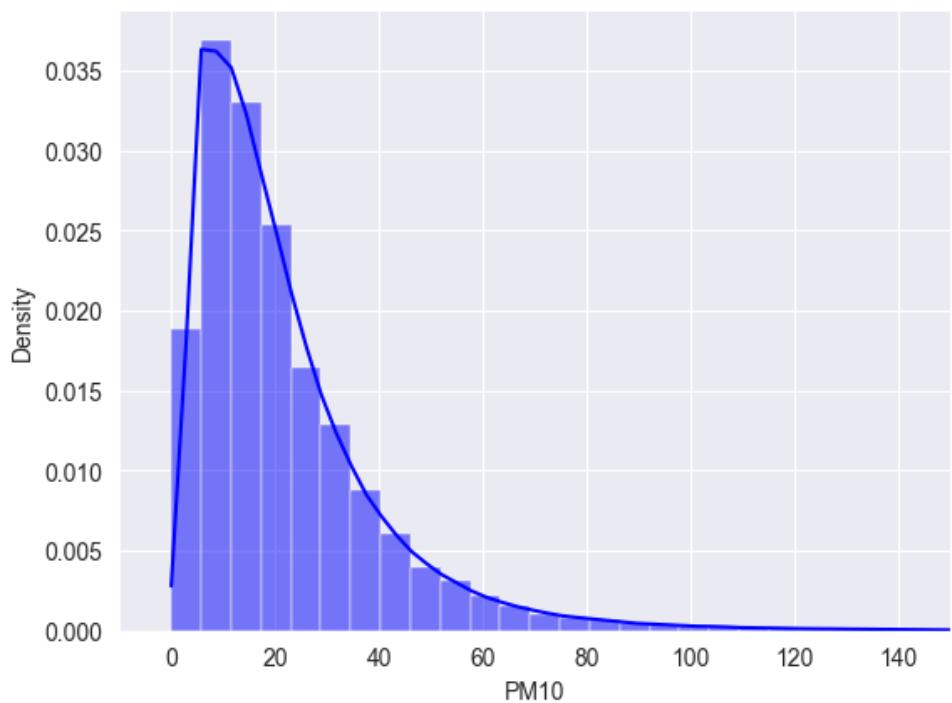


Figura 7.4 - distribuzione PM10

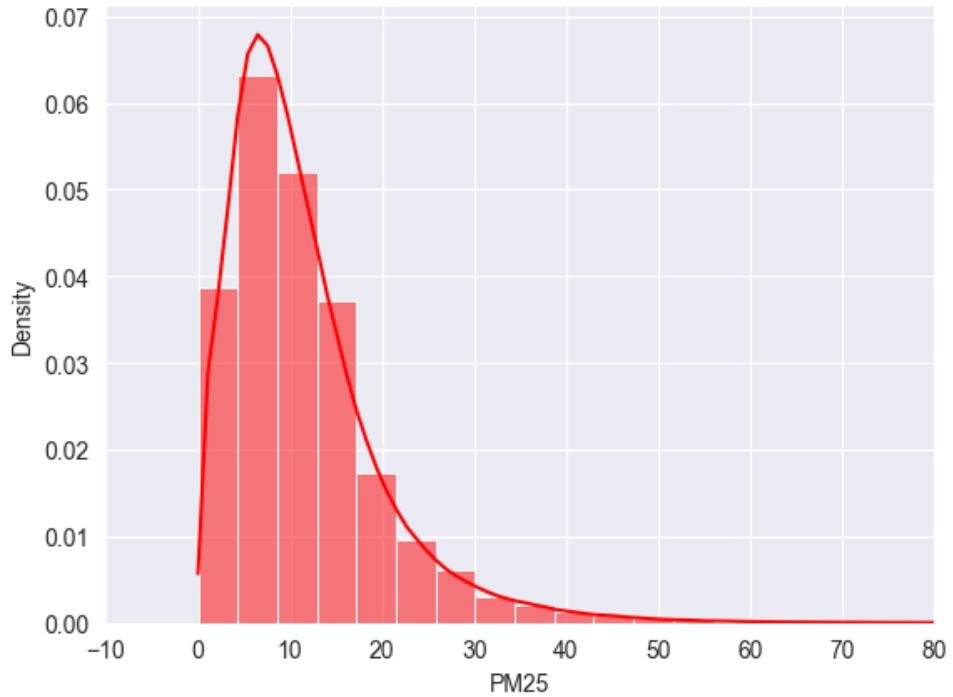


Figura 7.5 - distribuzione PM25

Infine, carichiamo le analisi fatte sulla distribuzione relativa alle misure del PM10 e del PM25. Così come per le analisi precedenti osserviamo una corrispondenza tra i valori delle misure e le analisi precedenti. Infatti, la maggior parte delle misure hanno valori al di sotto della soglia di sicurezza, con la differenza, rispetto alle due analisi precedenti, che in questo caso i valori sono più distribuiti, cioè, nell'analisi dell'anidride solforosa e del monossido d'azoto, avevamo tante misure comprese, principalmente, in due intervalli. Queste misure, invece, hanno valori che appartengono a più intervalli, quindi abbiamo una maggiore distribuzione, questo ci indica che negli anni i valori sono cambiati spesso. Questo può essere dovuto a politiche per la riduzione dell'inquinamento, oppure, può essere dovuto alla nascita di nuove aziende o complessi industriali. Di seguito procediamo con l'analisi temporale.

6 Analisi temporali

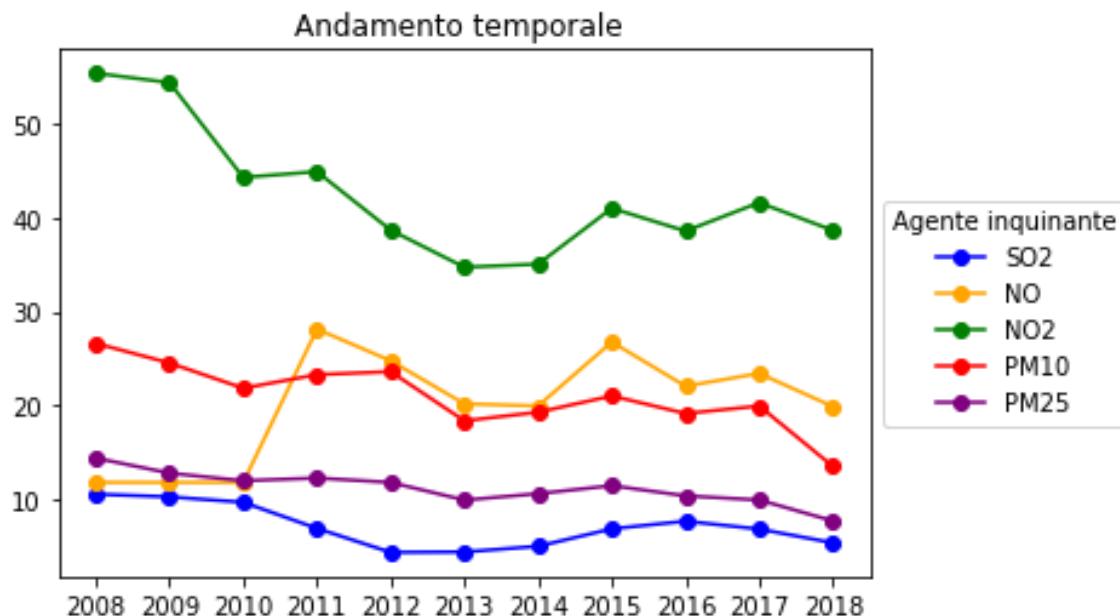


Figura 8 - andamento temporale agenti inquinanti

Si è quindi deciso di valutare come gli agenti inquinanti variassero nel tempo. Prima di effettuare queste analisi è stata necessaria un'operazione di pulizia del dataset. Per quanto concerne il grafico sopra riportato si è ritenuto necessario andare a considerare il valore medio annuo dei cinque agenti inquinanti più rilevanti per la nostra analisi.

Possiamo infatti notare che l'NO₂ è l'indice inquinante più importante, in quanto assume valori decisamente più alti rispetto agli altri ma, come possiamo notare, negli ultimi anni è diminuito, soprattutto tra il 2009 e il 2013. Mentre per quanto riguarda l'NO ha avuto un netto aumento tra il 2010 e il 2011.

Come riportato dall'articolo di Sicurauto¹, le emissioni di NO₂ in Europa sono calate drasticamente nelle settimane dell'emergenza Coronavirus. Una conferma arriva dai dati rilevati dall'European Enviroement Agency, in quanto le emissioni di biossido di azoto sono calate drasticamente. A Madrid le emissioni medie di NO₂ sono diminuite del 56% tra la settimana del 16-22 marzo 2020 rispetto al 2019.

Rispetto al forte calo dell'inquinamento da biossido di azoto, lo studio dell'ENEA² evidenzia una riduzione più modesta del livello di PM10 e PM2.5 mentre in alcune città, le polveri sottili hanno fatto registrare un leggero aumento.

Lo studio ha quantificato anche il numero di morti premature evitate a seguito della riduzione dell'inquinamento per effetto delle misure adottate dai governi UE contro la pandemia.

¹ <https://www.sicurauto.it/news/attualita-e-curirosita/emissioni-di-no2-50-in-europa-con-l'emergenza-coronavirus/>

² <https://www.enea.it/>

6.1 BoxPlot

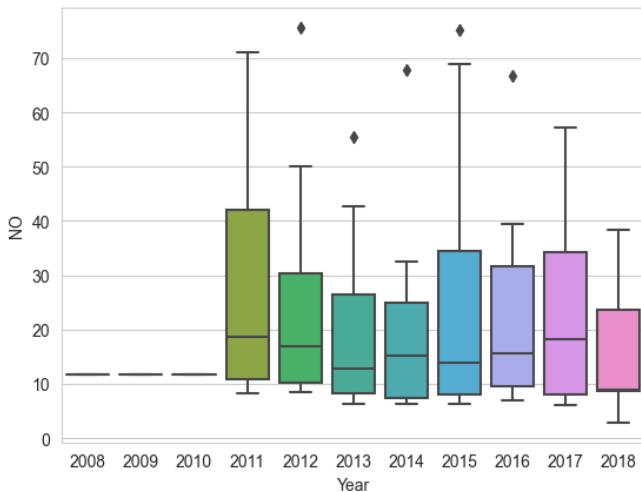


Figura 9.1 - BoxPlot NO

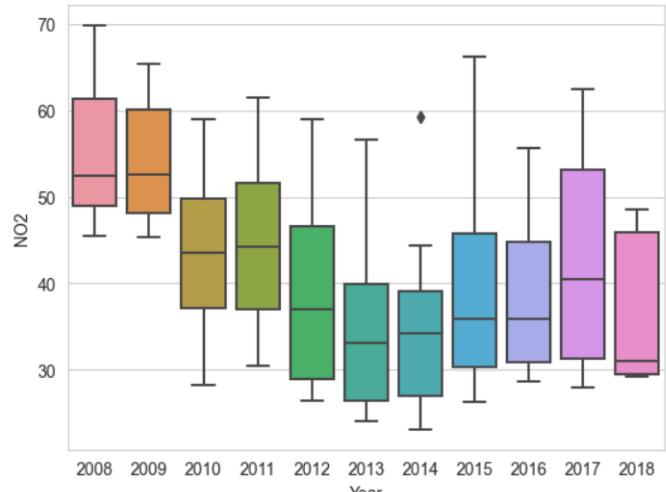


Figura 9.2 - BoxPlot NO₂

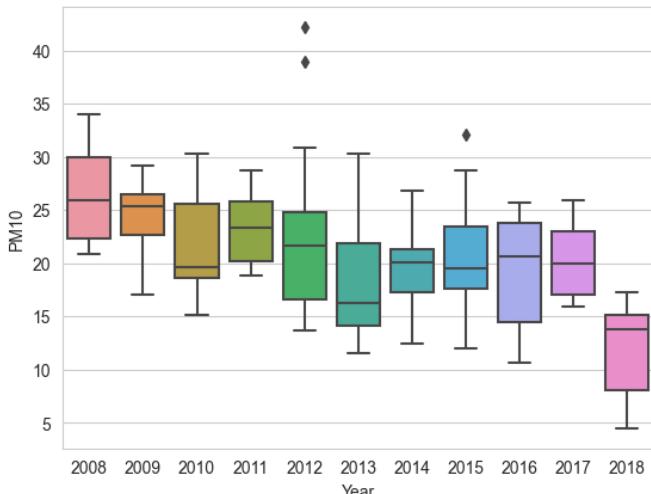


Figura 9.3 - BoxPlot PM10

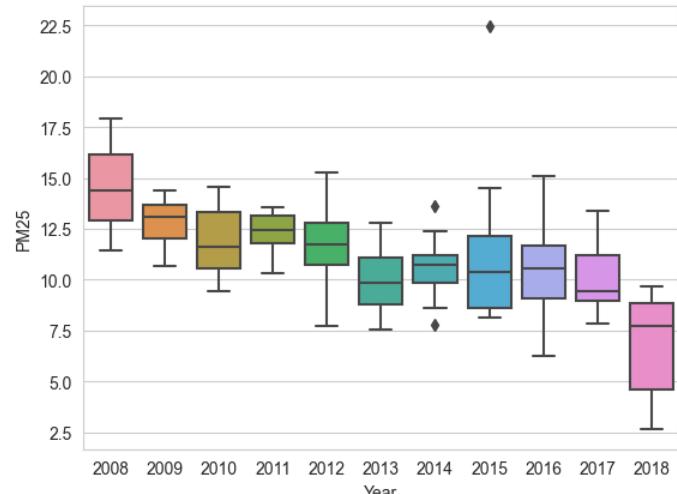


Figura 9.4 - BoxPlot PM25

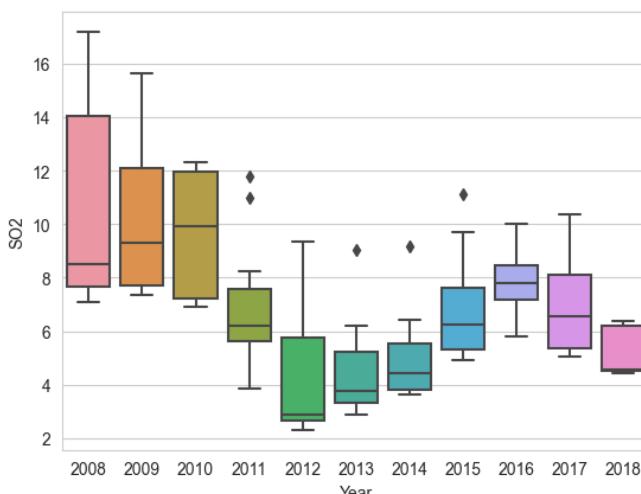


Figura 9.5 - BoxPlot SO₂

Come possiamo osservare dal secondo boxplot, l' NO_2 è l'agente inquinante assume i valori più alti nel corso degli anni. Queste analisi confermano le analisi temporali fatte in precedenza, gli ossidi di azoto stanno diminuendo in quanto, oltre al Corona Virus, si sta optando sempre più per mezzi eco-sostenibili, riducendo nettamente l'inquinamento, rispetto agli anni precedenti. Gli ossidi di azoto provengono principalmente dai gas di scarico delle macchine e da apparecchi per uso domestico come il riscaldamento a kerosene. Secondo la European Environmental Agency, un'esposizione prolungata a queste sostanze può

interferire con la capacità del sangue di trasportare l'ossigeno, causando insufficienza respiratoria. Gli ossidi di zolfo sono anche dannosi per la vegetazione e l'ambiente, essendo alla base del fenomeno dell'acidificazione delle piogge.

Il PM2.5 è una sottocategoria del PM10 ed è molto più dannoso del PM10 per via del suo diametro estremamente ridotto (inferiore ai 2.5 micron).

Per quanto riguarda le polveri sottili, la Spagna si posiziona al quarto posto con 125,31 migliaia di tonnellate prodotte di polveri sottili PM2.5. Come però possiamo vedere dai Boxplot riportati, nel tempo queste polveri sono diminuite, comportando un netto miglioramento della qualità dell'aria.

6.2 StripPlot

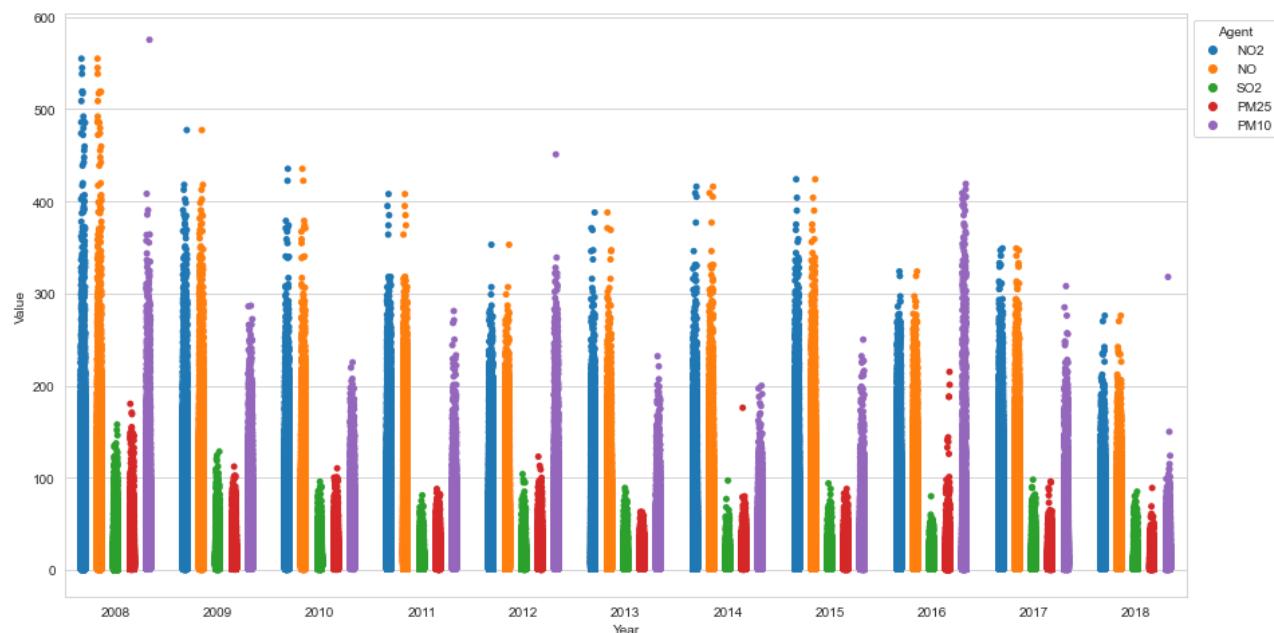


Figura 10 - StripPlot agenti inquinanti

Si può vedere che con il passare del tempo la quantità di agenti inquinanti nell'aria ha subito un drastico calo, soprattutto per quanto riguarda i composti NOx e il materiale particolato aerodisperso con particelle inferiori a $10 \mu\text{m}$, questo è sicuramente un dato incoraggiante in quanto sottolinea una riduzione dell'inquinamento dell'aria. Questo calo è stato probabilmente dettato dall'introduzione a livello europeo di normative atte a contrastare appunto il fenomeno dell'inquinamento atmosferico. Inoltre, in Spagna per porre un freno al cambiamento atmosferico si è deciso di introdurre un segnale stradale nuovo³ per delimitare l'accesso ai centri urbani ai veicoli molto inquinanti con multe fino ai 6000€.

Inoltre, il calo di NO2 che era già stato notevole ha subito un ulteriore abbassamento nel 2018, probabilmente dovuto anche all'avvertimento ricevuto il 15 febbraio 2017, nel quale l'Unione Europea esortava la Spagna⁴, l'Italia, la Germania, la Francia e UK a ridurre le emissioni di biossido di azoto (NO2) affinché queste rientrassero nei limiti consentiti stabiliti dalla Direttiva Europea 2008/50/CE⁵.

³ https://www.repubblica.it/motori/sezioni/attualita/2022/02/15/news/spagna_segnale_auto_inquinanti-337826326/

⁴ https://ec.europa.eu/commission/presscorner/detail/en/IP_17_238

⁵ <https://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:32008L0050>

7 Analisi geografiche

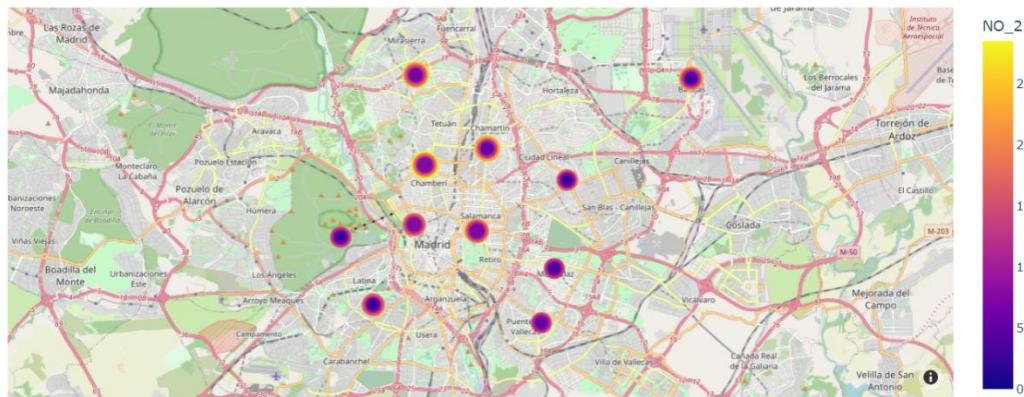


Figura 11.1 - ScatterPlot NO₂

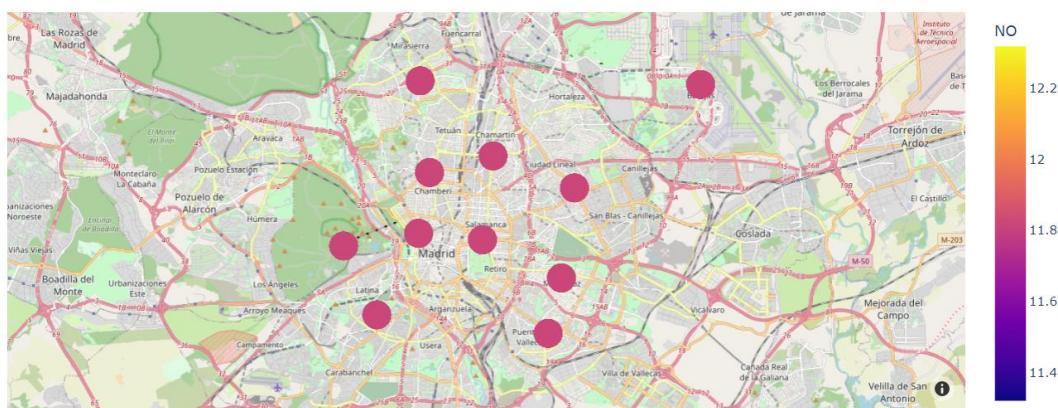


Figura 11.2 – SatterPlot NO

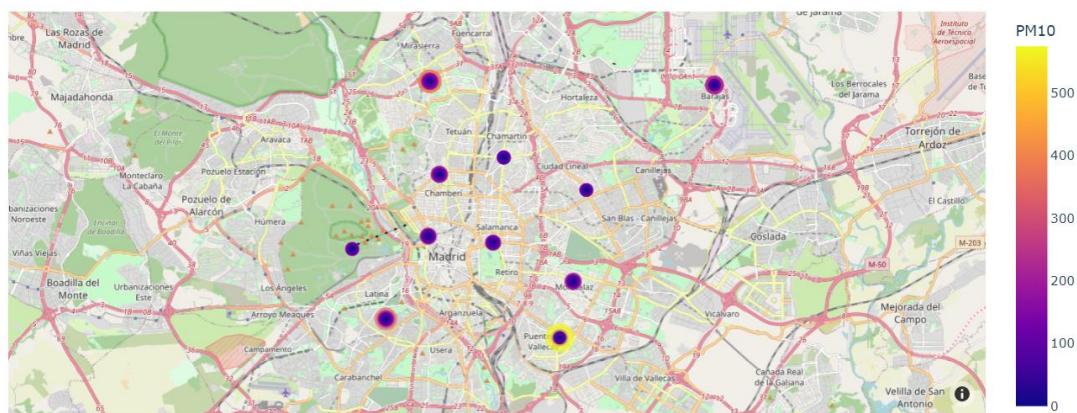


Figura 11.3- ScatterPlot PM10

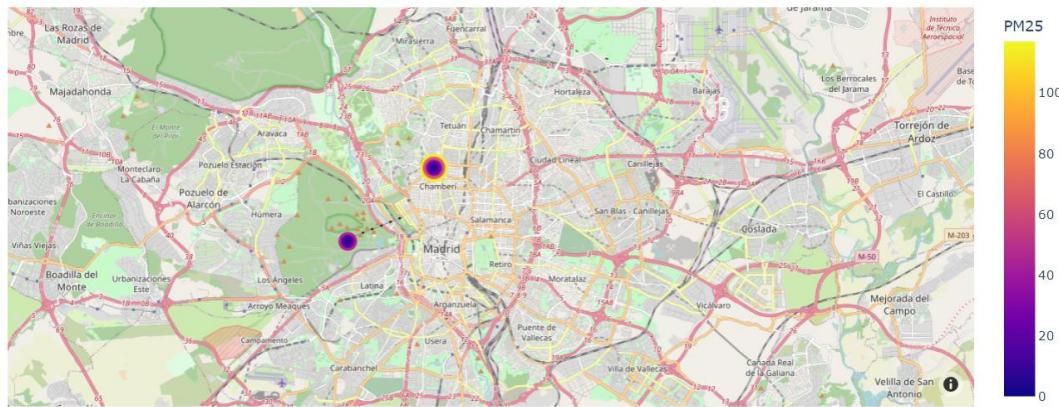


Figura 11.4 - ScatterPlot PM25

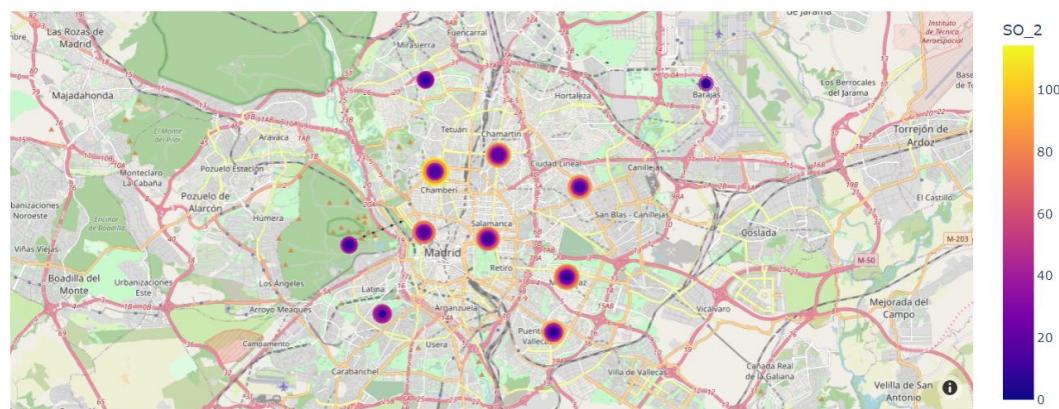


Figura 11.5 - ScatterPlot SO₂

Le 5 mappe sopra presentate mostrano quali stazioni hanno eseguito il rilevamento dei vari fattori inquinanti e l'intensità della rilevazione mediante una colorazione, che nel caso di valori bassi tende al blu, mentre per valori alti tende al giallo.

7.1 Correlazioni

Si è quindi analizzata la correlazione tra gli attributi, ovvero quanto le colonne del dataset dipendono l'una dall'altra.

Si è innanzitutto utilizzato il metodo “pairplot” messo a disposizione dalla libreria “seaborn”, che permette di graficare a coppie i valori degli attributi in un piano cartesiano.

Si sono valutate le correlazioni relative ai campi: “NO”, “NO2”, “SO2”, “PM25” e “PM10”

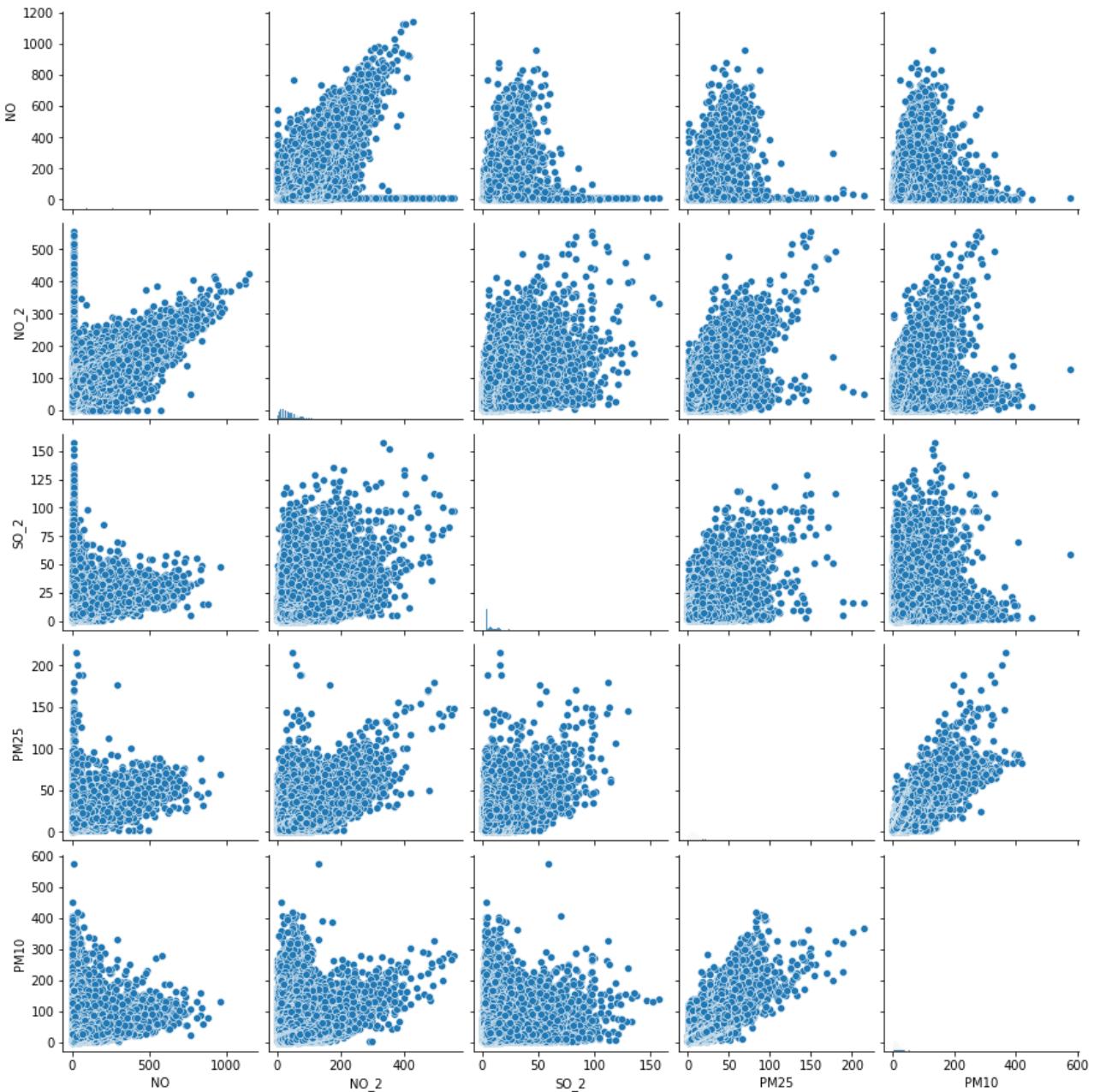


Figura 12 - PairPlot agenti inquinanti

Come previsto, sembra esserci una certa correlazione tra la concentrazione delle sostanze e il corrispettivo indice PM25.

I restanti attributi sembrano comunque abbastanza correlati tra loro, meno rispetto all'affermazione precedente, con valori che si distribuiscono abbastanza uniformemente sul piano cartesiano.

Per confermare ulteriormente quanto appena detto, si è utilizzato il metodo “heatmap”, sempre della libreria “seaborn”, per rappresentare esplicitamente la matrice di correlazione tra gli attributi.

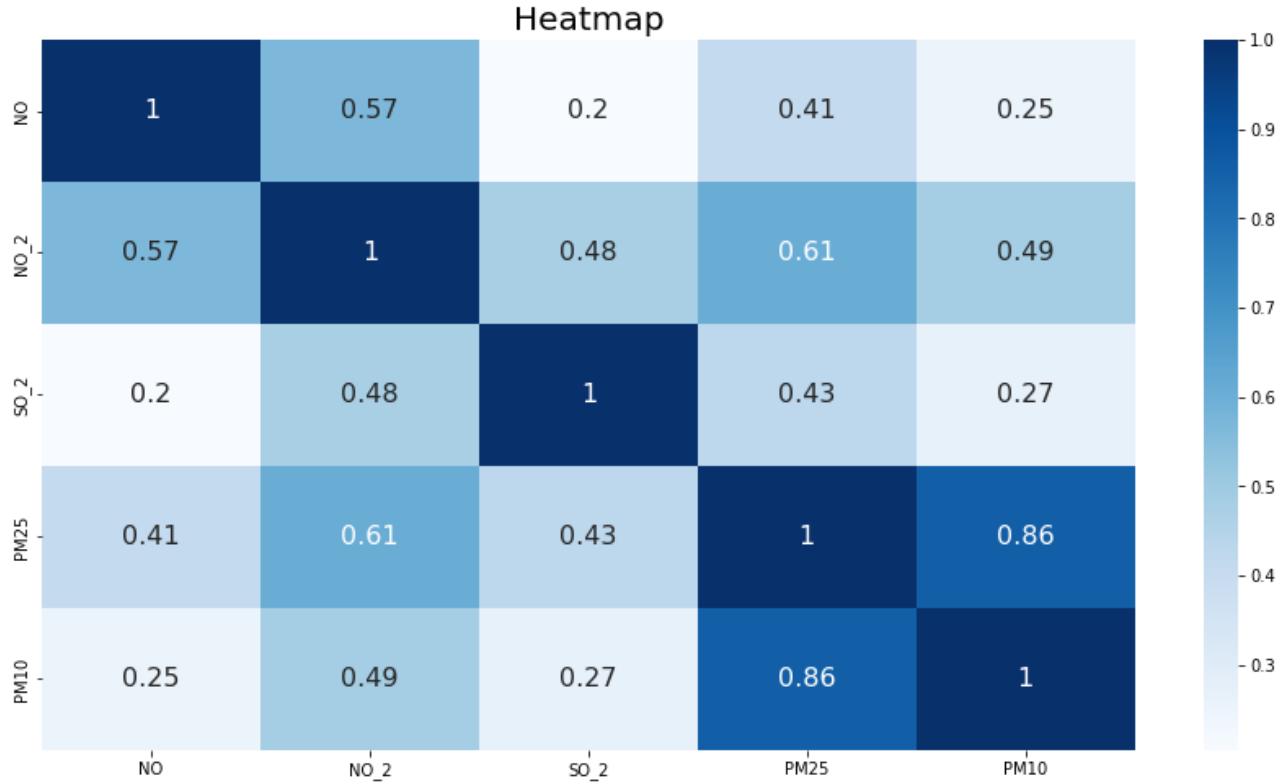


Figura 13 - heatmap agenti inquinanti

Questo grafico conferma quanto precedentemente affermato. Vi è infatti un alto valore di correlazione tra le coppie “PM25/NO₂” e “PM25/PM10”, abbiamo comunque una correlazione abbastanza alta tra “NO/NO₂”.

8 Clustering

In questa fase del progetto abbiamo realizzato i vari task di clusterizzazione, per individuare all’interno del dataset elementi con caratteristiche comuni. Infatti, il clustering è un insieme di tecniche di analisi dei dati volte alla selezione e al raggruppamento di elementi omogenei in un insieme di dati.

Le tecniche di clustering si basano su misure relative alla somiglianza tra gli elementi.

8.1 Fase di ETL per il clustering

Prima di descrivere i risultati ottenuti tramite le varie tecniche di clustering, occorre specificare quali sono state le azioni fatte sul dataset per ottenere tali risultati. Il primo algoritmo utilizzato è stato i K-means, il quale, lavorando sulle distanze non ammetteva valori Nan. Abbiamo valutato se eliminare direttamente tali valori, oppure, se sostituirli con una media troncata. Il dataset presenta alcune problematiche relative alla presenza di valori Nan, ad esempio abbiamo il problema che alcune stazioni, per un mese, o per lunghi intervalli di tempo, non hanno misurato un certo agente inquinante, oppure, lo hanno misurato pochissime volte. Nel calcolo della media troncata abbiamo operato due suddivisioni per ottenere un valore molto preciso. Noi calcolavamo per ogni agente inquinante usato per l’analisi, abbiamo preso i singoli anni, e per ogni anno abbiamo considerato le singole stazioni.

A causa della presenza di troppi valori Nan, o solo valori Nan, la funzione usata per il calcolo della media troncata restituiva un valore Nan. Quindi, abbiamo dovuto utilizzare un valore medio più ampio, che era quello annuale. Questa soluzione generava una nuova problematica, infatti, i punti non erano più distribuiti

in maniera omogenea, avevamo dei fasci di punti con un parametro costante, e questo sbilanciava i cluster. Di seguito riportiamo un'immagine di questo esempio.

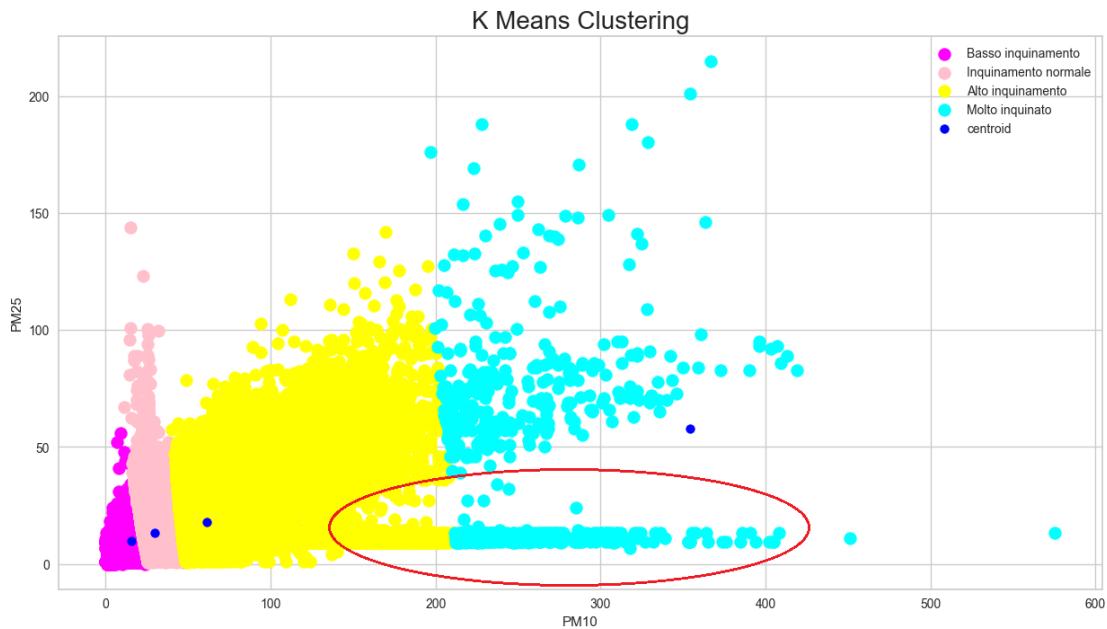


Figura 14 - K-Means Clustering PM25/PM10

La parte cerchiata in rosso è quella relativa ai Nan sostituiti con le medie annuali.

Di conseguenza abbiamo scelto di procedere con un approccio ibrido, cioè, dove possibile sostituendo i Nan con le medie troncate. Dove ciò non è possibile procediamo con la rimozione del valore.

La seconda operazione che è stata fatta è relativa solo a due algoritmi di clustering, il gerarchico e il DBSCAN. Abbiamo dovuto ridurre le dimensioni del dataset per motivi computazionali e di tempo. Infatti, lavorando in un ambiente locale non è possibile eseguire quest'analisi su dataset con oltre un milione di righe, quindi abbiamo estratto casualmente elementi da ogni singolo anno del dataset e, mettendoli insieme, abbiamo ottenuto un nuovo dataset.

8.2 Clustering bidimensionale

Come prima operazione di clustering abbiamo preso in considerazione i campi relativi al biossido di azoto (NO_2) e l'anidride solforosa (SO_2). Questa scelta è stata fatta prendendo in considerazione il fatto che, dal grafico delle correlazioni, creato precedentemente, i valori di questi due agenti sembrano essere distribuiti in maniera uniforme nel piano cartesiano.

Come primo algoritmo abbiamo deciso di utilizzare il K-means. Tale algoritmo minimizza la varianza totale intra-gruppo e ogni gruppo viene identificato tramite un centroide o punto medio. Vengono quindi determinati k punti di riferimento casuali, per poi iterare le seguenti:

1. Si associa ogni elemento al punto di riferimento più vicino, formando così k cluster
2. Per ogni cluster individuato si determina il nuovo centroide
3. Si aggiornano i centroidi

Questa procedura è reiterata, finché le posizioni dei centroidi non convergono.

L'incognita prevista per utilizzare questo algoritmo è quella relativa al numero di cluster. Quindi, per poterlo utilizzare, abbiamo scelto questo parametro mediante l'elbow method.

Di seguito riportiamo i risultati ottenuti.

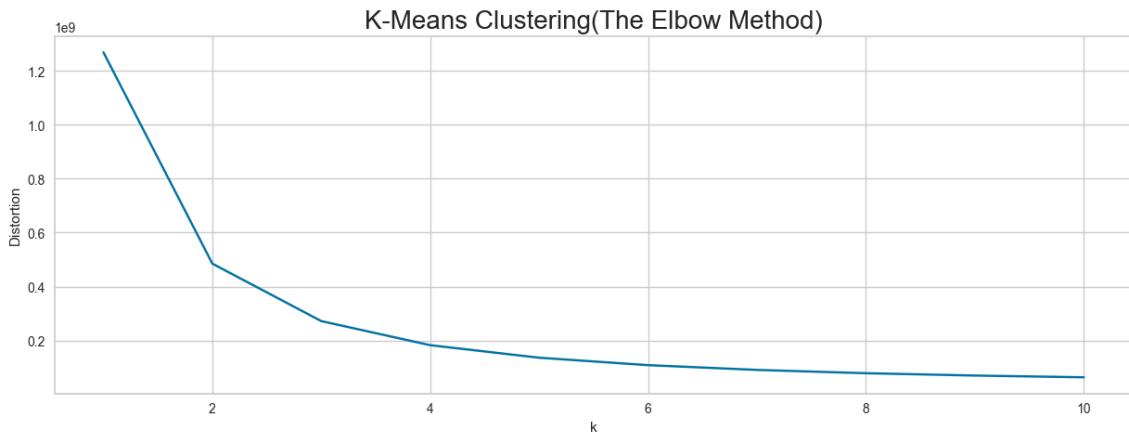


Figura 15 - The Elbow Method

Questo è il grafico che abbiamo ottenuto applicando l'Elbow Method. Il numero di cluster va scelto nel punto in cui la curva inizia a deviare. Quindi, abbiamo scelto come valore del parametro K il 4, cioè, ipotizziamo di avere quattro cluster.

A questo punto è stato possibile applicare l'algoritmo ottenendo i seguenti risultati.

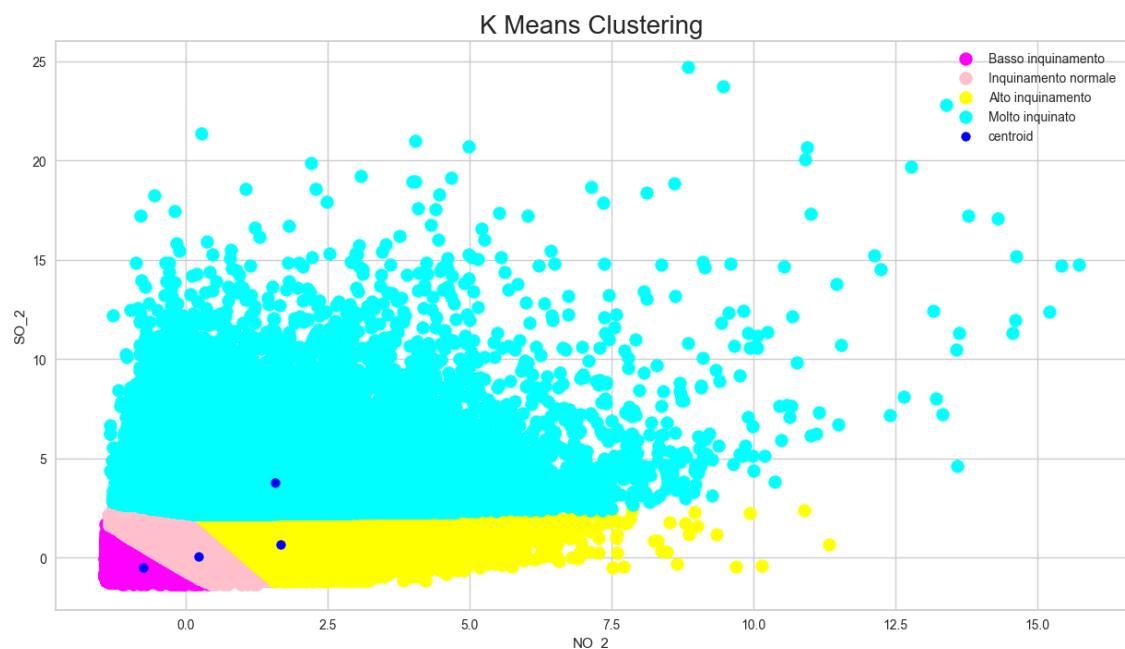


Figura 16 - K Means Clustering SO₂/NO₂

Come si può osservare dal grafico abbiamo individuato 4 gruppi, che sono stati definiti nel seguente modo:

- Il cluster magenta corrisponde a record con valori bassi di NO₂ e di SO₂, quindi a un inquinamento basso.
- Il cluster rosa corrisponde a record con valori un po' più alti di NO₂ e di SO₂, quindi a un inquinamento 'normale'.
- Il cluster giallo corrisponde a record con valori alti di NO₂ e valori più alti di SO₂ rispetto ai due cluster precedenti, quindi corrisponde a un inquinamento alto.
- Il cluster ciano corrisponde a record con valori molto alti di NO₂ e di SO₂, quindi a un inquinamento molto alto.



Per valutare il grafico, successivamente, abbiamo utilizzato il metodo “silhouette” che misura quanto un elemento è affine agli altri appartenenti allo stesso cluster. Questa misura è valutata in un intervallo [-1,1], più la misura tende a 1, più l'oggetto è adatto al cluster di appartenenza.

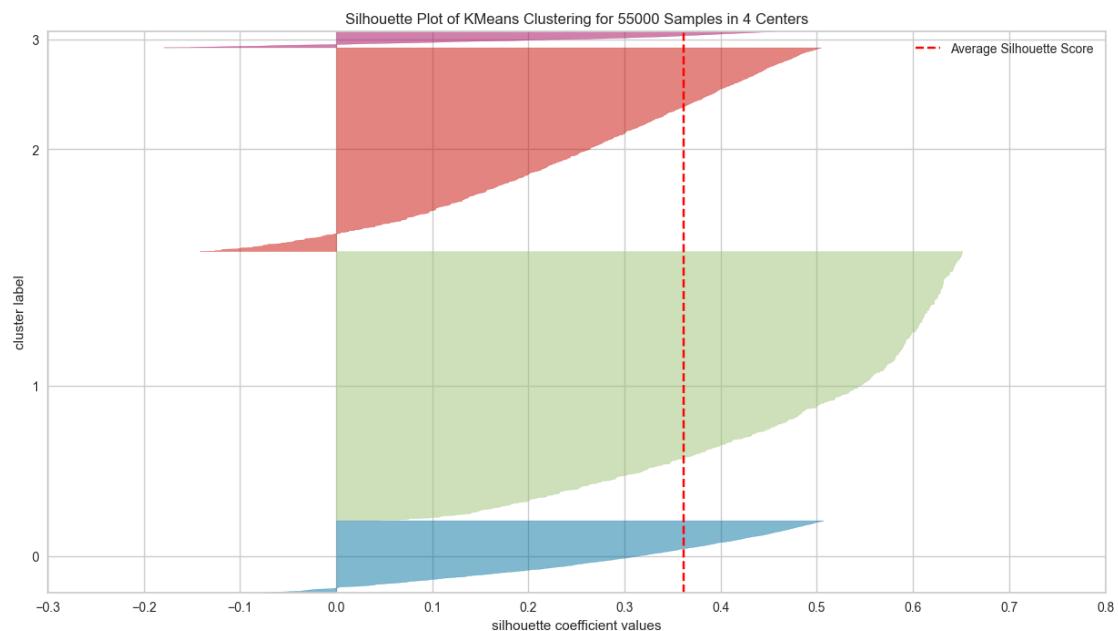


Figura 17 - Silhouette Plot of K Means Clustering SO₂/NO₂

Dal grafico osserviamo come il valore medio non sia molto alto. Abbiamo, all'incirca, un valore medio della silhouette pari a 0.37. Però, osserviamo che molti elementi superano tale valore medio e lavoriamo con un dataset che ha oltre un milione di elementi. Di conseguenza, abbiamo ritenuto il risultato di clustering accettabile.

Successivamente, è stata effettuata un'operazione di clustering gerarchico, utilizzando sempre il dataset precedente e con un numero di cluster pari a 4.

Di seguito riportiamo il risultato ottenuto.

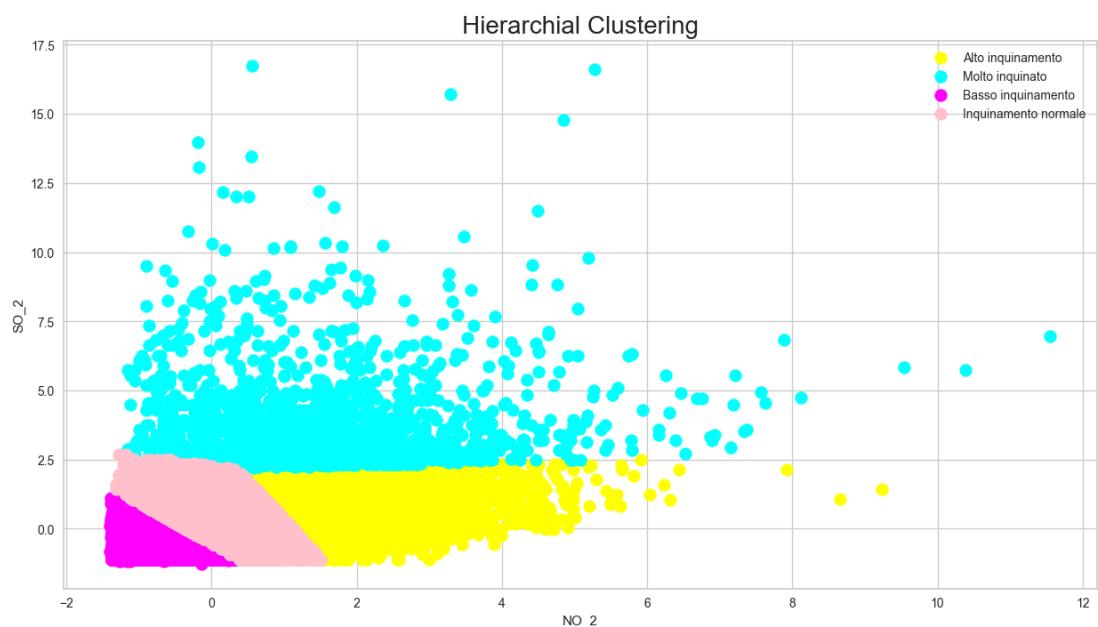


Figura 18 - Hierarchical Clustering SO₂/NO₂

Questo risultato presenta forti somiglianze con quello precedente, e questo è un aspetto positivo. Osserviamo come, a parità di cluster, la suddivisione degli elementi sia stata quasi identica, le uniche differenze che abbiamo sono quella relativa al cluster 'Rosa', il quale contiene più elementi, rispetto al risultato precedente e quella relativa al cluster 'Ciano', il quale contiene meno elementi, rispetto al risultato precedente. Si è deciso di effettuare un ulteriore operazione di clustering, prendendo in considerazione una nuova coppia di agenti inquinanti, che sono PM10 e SO₂. La scelta è stata fatta osservando la distribuzione dei valori sul piano cartesiano.

Iniziamo applicando l'Elbow Method sul nuovo dataset. Di seguito riportiamo i risultati ottenuti.

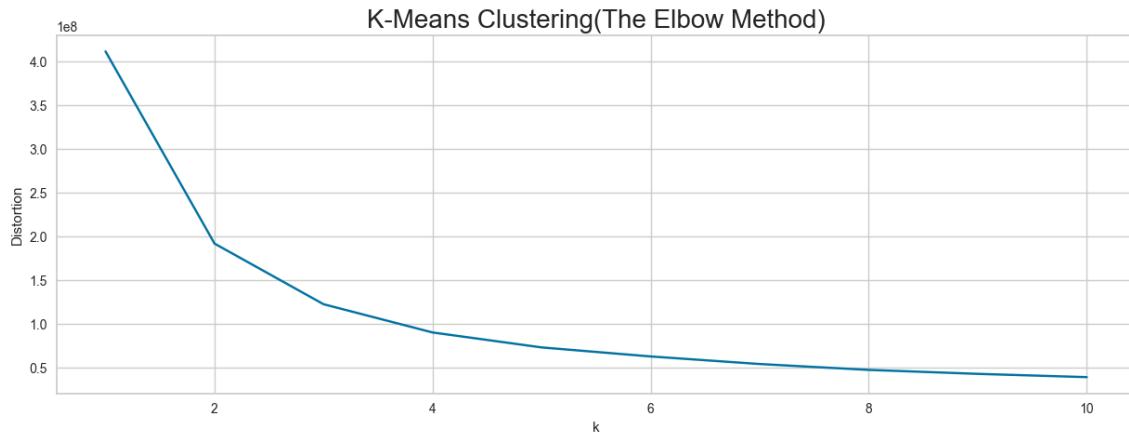


Figura 19 - The Elbow Method

Anche in questo caso osserviamo come il numero di cluster da cercare sia 4. Il passo successivo è stato quello di applicare l'algoritmo k-means. Di seguito riportiamo i risultati ottenuti.

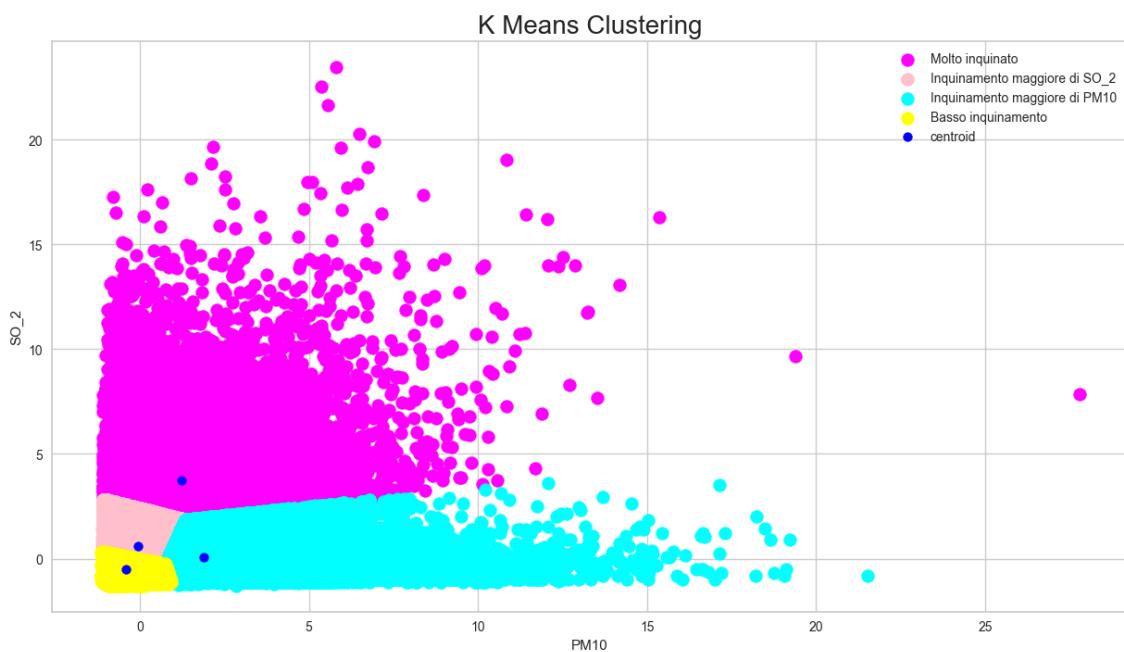


Figura 20 - K Means Clustering SO₂/PM10

Come si può osservare dal grafico, sono stati identificati quattro cluster, definiti nel seguente modo:

- Il cluster magenta corrisponde a record con valori molto alti di PM10 e di SO₂, quindi, corrisponde a un inquinamento molto alto.



- Il cluster rosa corrisponde a record con valori alti di SO₂, e valori normali di PM10, quindi, corrisponde a un inquinamento medio/alto.
- Il cluster ciano corrisponde a record con valori molto alti di PM10, e valori normali di SO₂, quindi, corrisponde a un inquinamento medio/alto.
- Il cluster giallo corrisponde a record con valori molto bassi di PM10 e di SO₂, quindi, corrisponde a un basso inquinamento.

Per valutare il grafico, successivamente, abbiamo utilizzato il metodo “silhouette”. Tale metodo è stato descritto precedentemente e di seguito riportiamo i risultati ottenuti.

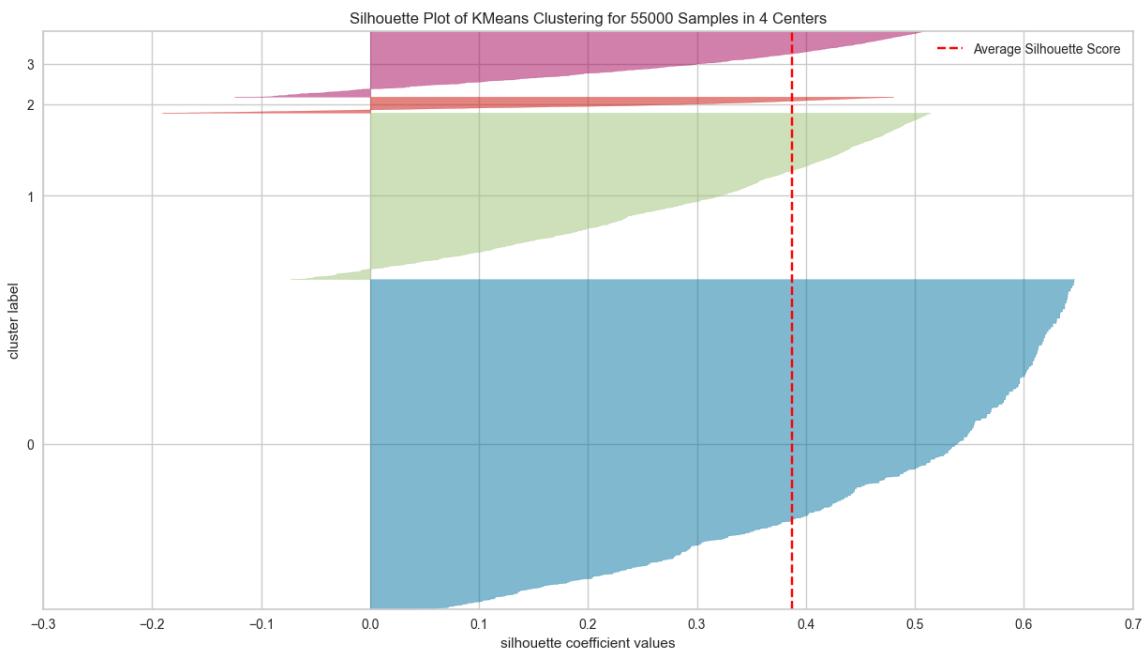


Figura 21 - Silhouette Plot KMeans Clustering SO2/PM10

Il risultato ottenuto in questo caso risulta essere migliore rispetto a quello precedente. Osserviamo come il valore medio della silhouette, all'incirca, è pari a 0.39. Un valore maggiore rispetto a quello ottenuto precedentemente, che, all'incirca, era pari a 0.37. I due risultati sono leggermente diversi tra loro, perché, comunque hanno una feature in comune è la SO₂. Però, usare come elemento di clusterizzazione l'agente inquinante PM10 al posto dell'NO₂ ci ha permesso di ottenere un risultato che, ha sempre quattro cluster, ha molti elementi che sono stati inseriti nel cluster corretto in quanto hanno un valore della silhouette maggiore rispetto al valore medio, e, infine, ha un valore medio di questa misura leggermente superiore rispetto alla clusterizzazione precedente.

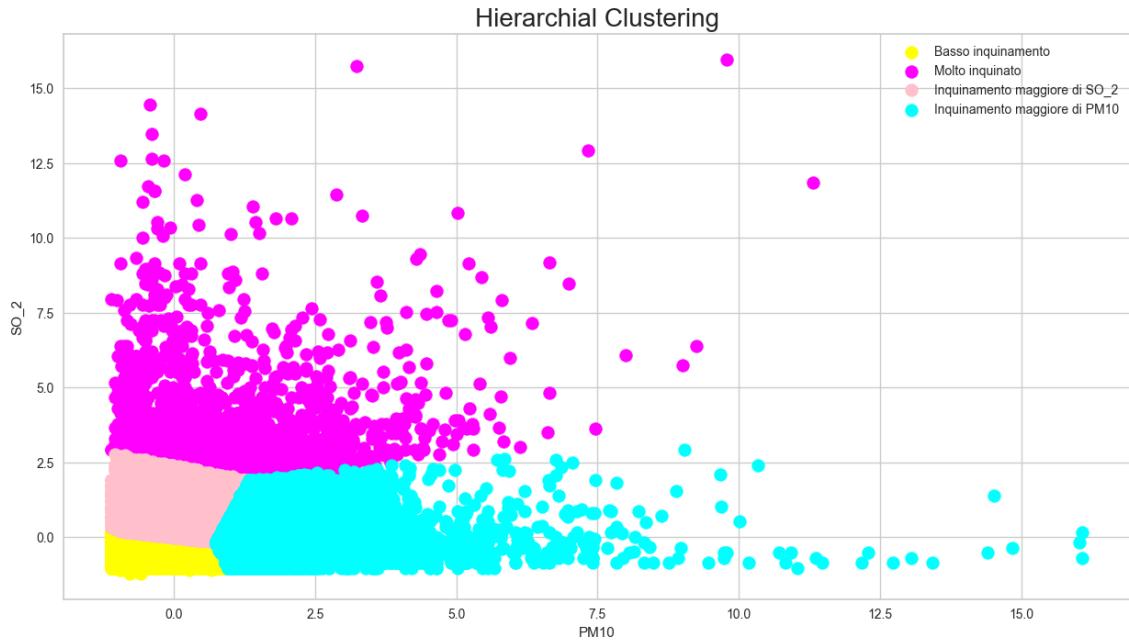


Figura 22 - Hierarchial Clustering SO₂/PM10

Lo step successivo, in maniera analoga al caso precedente, è stato quello di utilizzare il cluster gerarchico. Il risultato ottenuto è molto soddisfacente, e conferma il risultato ottenuto tramite il k-means. Infatti, abbiamo ottenuto quattro cluster che hanno all’incirca lo stesso insieme di elementi. Le uniche differenze che abbiamo sono relative al cluster rosa e ciano, i quali hanno una dimensione leggermente più grande. Il cluster magenta è leggermente più piccolo, e gli elementi che prima gli appartenevano adesso sono stati inseriti negli altri due, cioè, nel rosa e nel magenta.

8.3 PCA

Si è inoltre effettuata un’operazione di clustering sfruttando tutti i principali agenti inquinanti. Per farlo abbiamo utilizzato La PCA, che sta per “principal component analysis”. Questa è una tecnica per la semplificazione dei dati utilizzata nell’ambito del machine learning. Questa tecnica ha lo scopo di ridurre il numero di features che descrivono un insieme di dati. Queste tecniche vengono implementate limitando il più possibile la perdita di informazioni.

Anche in questo caso è stato utilizzato come algoritmo il k-means. Prima di utilizzare questo algoritmo dobbiamo utilizzare l’Ebow Method. Di seguito riportiamo i risultati ottenuti.

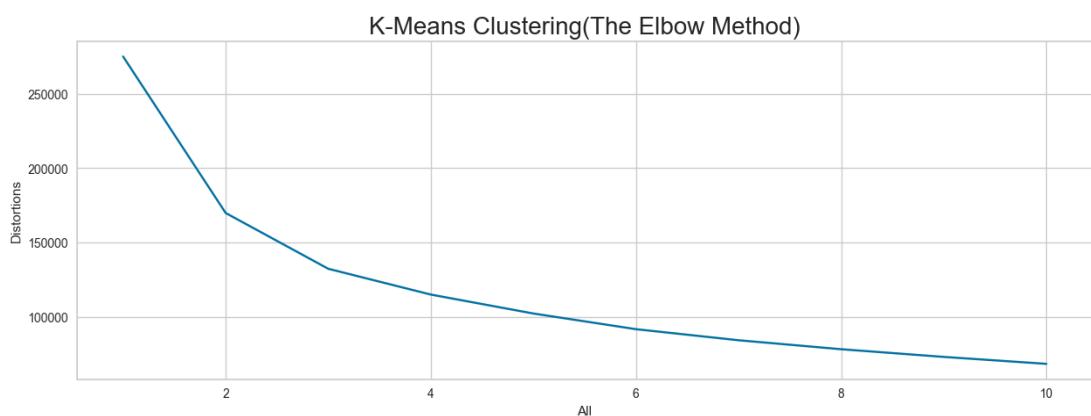


Figura 23 - The Elbow Method

Anche in questo caso sceglieremo come numero di cluster il valore quattro, quindi K=4. Arrivati a questo punto applichiamo l'algoritmo k-means.

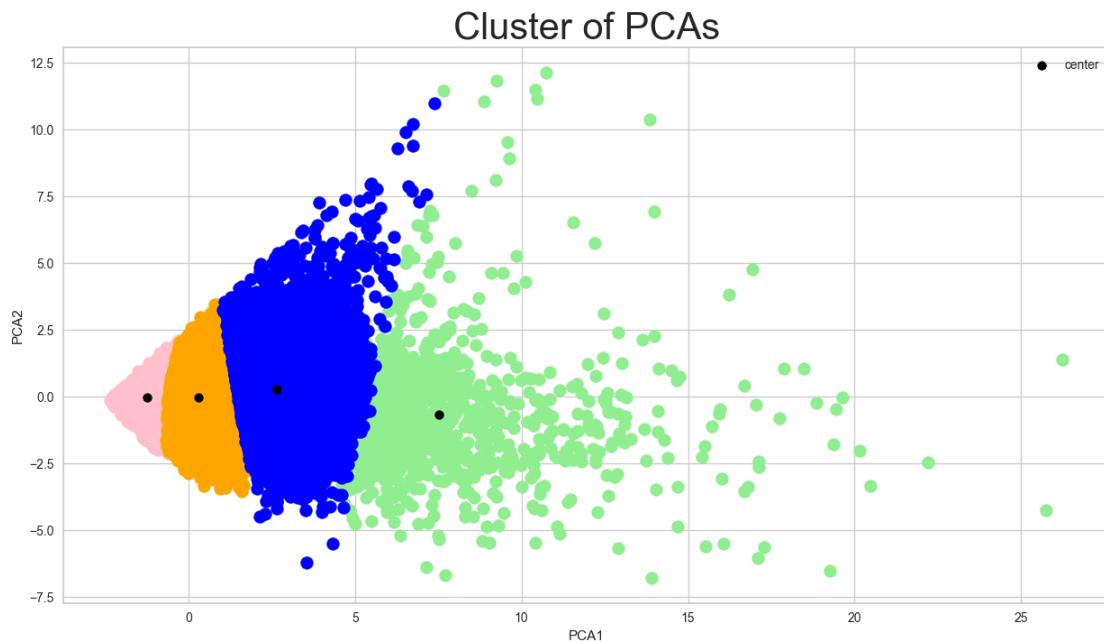


Figura 24 - PCA Cluster

Questo è il risultato che abbiamo ottenuto. Possiamo osservare come i cluster siano suddivisi in maniera abbastanza soddisfacente, cioè, possiamo distinguere i cluster tra di loro, e seguono un certo andamento. Le nuove features sono il risultato di una combinazione lineare di quelle precedenti, quindi, possiamo ipotizzare che i cluster rappresentino quattro insiemi di elementi, che sono l'insieme degli elementi con valori bassi degli agenti inquinanti, l'insieme degli elementi con valori normali degli agenti inquinanti, l'insieme degli elementi con valori alti degli agenti inquinanti e, infine, l'insieme degli elementi con valori molto alti degli agenti inquinanti.

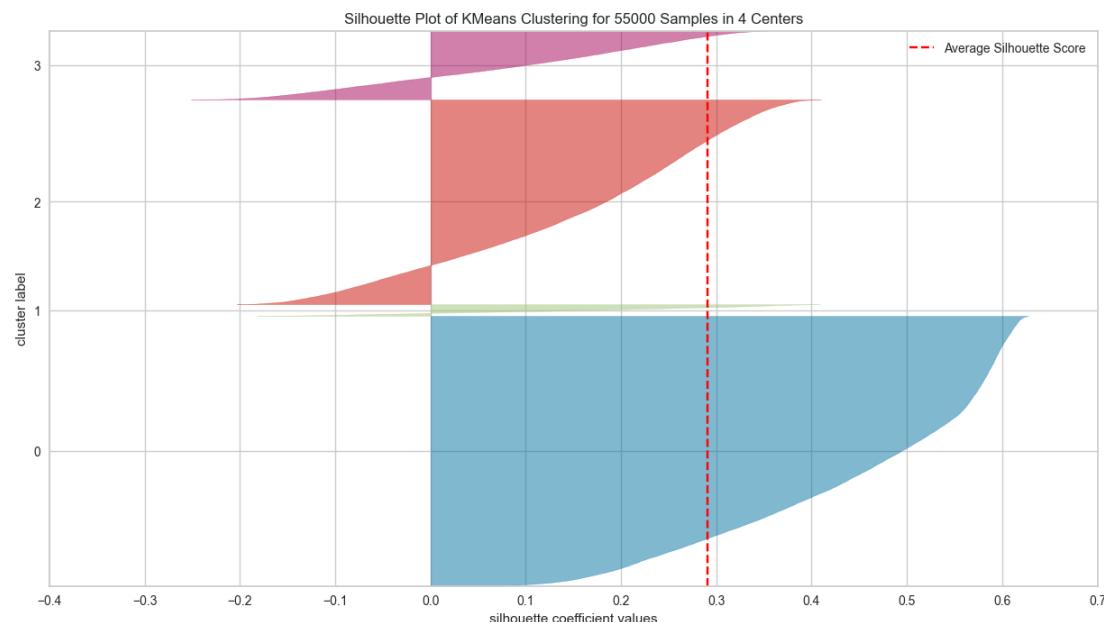


Figura 25 - Silhouette Plot of K Means Clustering



Successivamente è stata realizzata una misura della silhouette dei cluster ottenuti. Rispetto ai due risultati precedentemente ottenuti abbiamo un valore medio che è leggermente più basso, che vale all'incirca 0.29. Osserviamo come nel nuovo spazio delle features gli elementi del dataset rimangono comunque molto vicini, e questo è un problema in fase di clusterizzazione. Infatti, i cluster sono molto vicini tra di loro, e questo, sicuramente, è un fattore che ha influenzato negativamente il risultato finale.

8.4 DBSCAN

A seguito dei risultati ottenuti dalla riduzione della dimensionalità per mezzo della tecnica PCA, si è deciso di applicare la stessa, in combinazione con l'algoritmo DBSCAN, per cercare nuovi risultati.

Il DBSCAN è un algoritmo di clustering basato sulla densità, quindi, non richiede la conoscenza a priori del numero di cluster, come invece faceva l'algoritmo k-means. Però, richiede la conoscenza di altri due parametri che sono epsilon e il MinPts, il primo rappresenta la distanza tra i punti affinché questi siano considerati parte dello stesso cluster, e il secondo rappresenta il numero minimo di punti per formare una “regione densa”.

Per stimare la distanza epsilon, si è fatto uso della classe NearestNeighbors ed è stato realizzato un grafico che rappresenta l'epsilon al variare della distanza tra i punti, in relazione all'indice di questi ultimi. Il valore ideale di questo parametro di questo parametro è quello che corrisponde al punto della curva in cui la pendenza aumenta drasticamente. Di seguito riportiamo il risultato ottenuto.

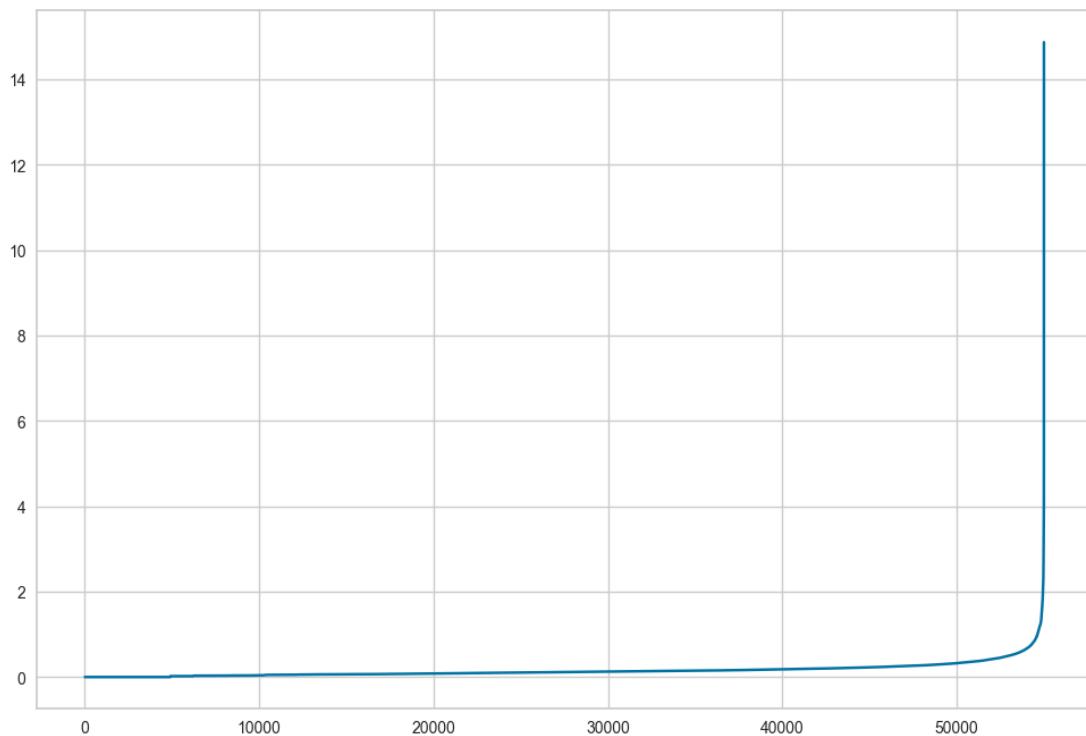


Figura 26 - risultato DBSCAN

Da questo grafico si è ricavato $\text{epsilon} = 1$.

Per definire il secondo parametro sono state effettuate delle ricerche, e da queste è emerso che solitamente si sceglie un valore maggiore rispetto al numero di features utilizzate. Quindi, si è posto $\text{MinPts}=6$.

Definiti i parametri si è proceduto all'applicazione dell'algoritmo DBSCAN, di seguito riportiamo i risultati ottenuti.

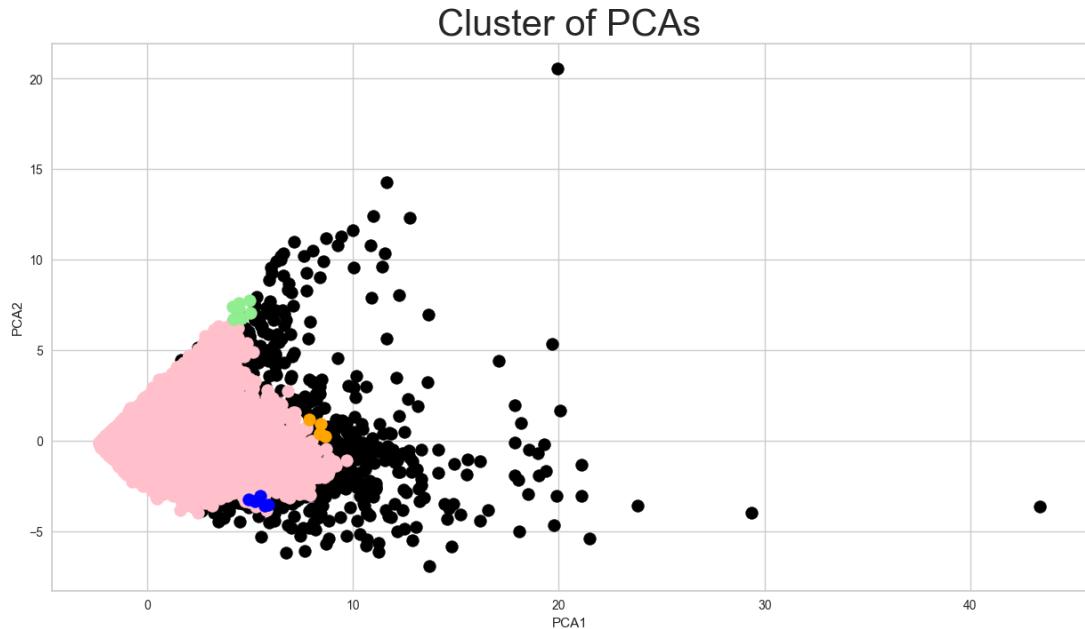


Figura 27 - PCA cluster

Osserviamo che l'algoritmo è riuscito a individuare quattro cluster, i quali però risultano essere estremamente sbilanciati. Infatti, si può vedere come il cluster rosa sia quello che contenga la maggior parte dei punti, mentre gli altri tre ne contengono pochissimi. Oltre a ciò, è possibile notare come molti punti non siano stati inseriti all'interno di un cluster, e perciò etichettati come rumore. Sembrerebbe, infine, che l'algoritmo DBSCAN non sia riuscito a lavorare bene con il dataset. Il motivo è analogo all'analisi precedente e fa riferimento alla distribuzione dei punti. Infatti, nel caso precedente, avere punti molto vicini e molto densi fra loro ci ha portato ad avere cluster vicinissimi, e questo ha inciso negativamente sul valore della silhouette. Per lo stesso motivo il DBSCAN fallisce, infatti costruisce un grande cluster, il quale racchiude tutti i punti più vicini e più densi, e con quello che rimane costruisce gli altri tre cluster. Gli altri punti avendo valori troppo alti risultano essere troppo isolati e distanti, e questo gli porta ad essere etichettati come rumore.

9 Classificazione

In questa sezione sono riportate le analisi effettuate attraverso le tecniche di classificazione, utilizzando modelli predittivi. L'obiettivo di questa analisi è quello di confrontare le prestazioni in termini di accuratezza di diversi modelli di classificazione, addestrati su un sottoinsieme del dataset originario, per poi testarlo sul restante sottoinsieme del dataset. Essendo il nostro dataset composto da dati raccolti dal 2008 al 2018, abbiamo deciso di addestrare il modello con i dati raccolti fino al 2016 e testarlo quindi con i dati dal 2017 al 2018.

L'accuratezza di un modello di classificazione è intesa come il rapporto tra il numero di dati classificati correttamente e il numero totale dei dati sottoposti a classificazione.

Per definire il modello sono stati utilizzati degli algoritmi messi a disposizione dalla libreria Scikit-Learn di Python. Si è deciso di effettuare due tipologie di classificazione:

- Binaria, relativamente al parametro PM10, in base alle classi Tollerante/Non Tollerante
- Multi-classe, relativamente al parametro NO_2, nelle classi Tollerante/Moderato/Non Tollerante per Persone Sensibili/Non Tollerante

9.1 Classificazione binaria in base a PM10

Si è deciso di effettuare una classificazione binaria sulla base della qualità dell'aria e della quantità di agenti chimici tolleranti presenti in essa. In particolare, si è deciso di effettuare un'analisi sulla quantità di polveri



sottili presenti nell'aria giornalmente, in quanto non dovrebbe superare il limite giornaliero di $20 \mu\text{g}/\text{m}^3$. Quindi si è deciso di individuare all'interno del dataset due classi, corrispondenti a quantità di PM10 tollerante/ Non Tollerante giornaliero. Essendo i dati stati raccolti da diverse stazioni, con cadenza oraria, si è deciso, data la grande quantità di dati a disposizione, di raggrupparli per giorno, effettuando una media dei valori.

	NO	NO_2	PM10	PM25	SO_2
0	11.820078	67.084565	39.855419	25.650870	19.815357
1	11.820078	66.552372	21.005401	15.767895	14.897970
2	11.820078	50.410930	9.173423	5.617396	10.923919
3	11.820078	58.247292	21.390939	13.236000	12.875144
4	11.820078	47.282292	18.745682	12.485729	11.804503

Figura 28 - Valori medi giornalieri agenti chimici

Da questa figura possiamo osservare come i valori numerici presenti appartengano ad insiemi numerici molto diversi, per questo motivo si è deciso di procedere con un'operazione di **normalizzazione** dei dati affinché tutti fossero compresi nell'intervallo [0,1]. In questo caso si è fatto uso della funzione *MinMaxScaler* della libreria *Sklearn*, ottenendo i risultati della seguente tabella:

	NO	NO_2	PM25	SO_2
0	0.041667	0.526316	0.403509	0.642857
1	0.041667	0.517544	0.228070	0.464286
2	0.041667	0.377193	0.052632	0.321429
3	0.041667	0.447368	0.192982	0.392857
4	0.041667	0.350877	0.175439	0.357143

Figura 29 - Risultato ottenuto dopo l'applicazione di *MinMaxScaler*

A questo punto siamo andati a controllare quanto il dataset fosse sbilanciato, secondo la seguente idea: appartengono alla classe 0 (Tollerante) i valori giornalieri con $\text{PM10} \leq 20$, mentre i restanti nella classe Non Tollerante.

```
condition_list = [(Livello <= 20), (Livello > 20)]  
  
choicelist = [0,1]  
  
Livello = np.select(condition_list, choicelist, default='Non Specificato')  
  
Livello = Livello.astype(int)
```

Si è quindi ottenuto il seguente risultato:

Tollerabili: 1673
Non tollerabili: 1615

Il passo successivo è stato quindi quello di controllare se il dataset fosse bilanciato o meno. Facendo questo si è visto che attualmente nel dataset sono presenti 1673 giorni con un valore “TOLLERABILE” e 1615 giorni con un valore “NON TOLLERABILE”. Alla luce di ciò si è deciso di bilanciare il dataset considerando tutti i 1615 giorni “NON TOLLERABILI”, ma scegliendo un sample della stessa misura di tipo “TOLLERABILI”.

Dopo aver proceduto con queste operazioni preliminari si è potuto procedere con l'**analisi delle correlazioni** del dataset, con la speranza di trovare attributi che non fossero troppo correlati tra loro.

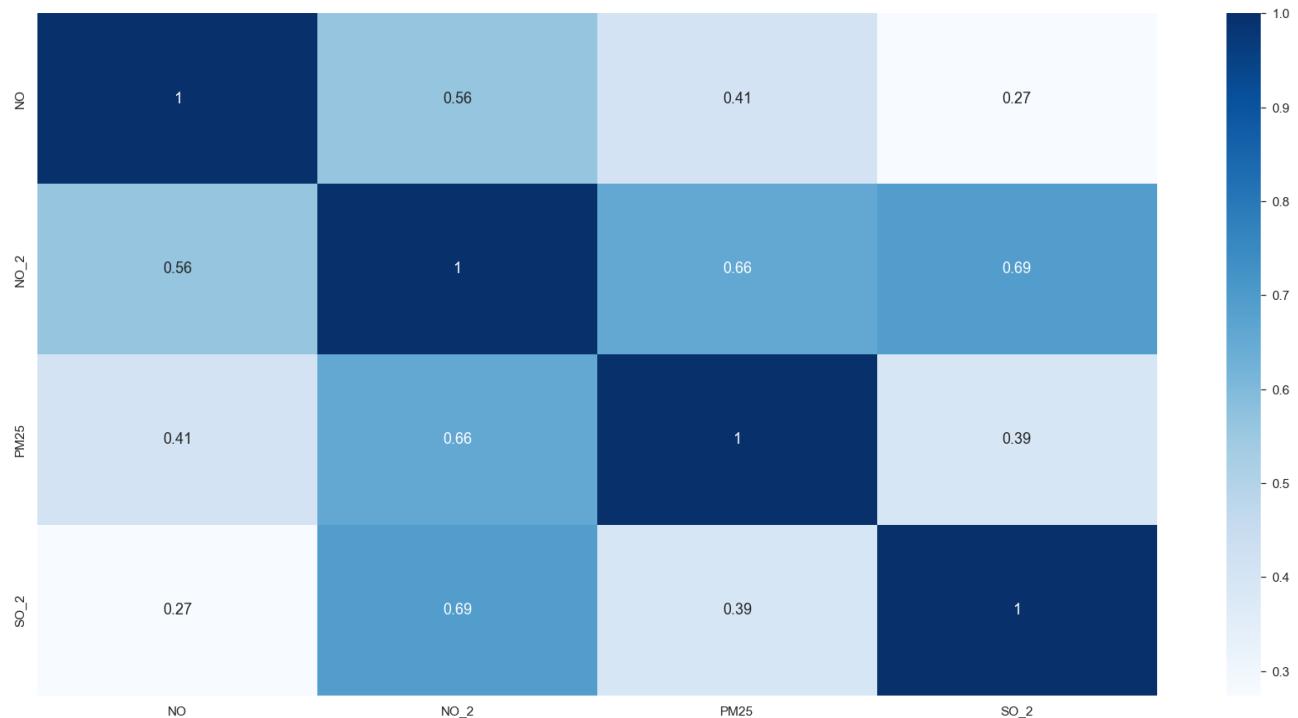


Figura 30 - Correlazione agenti inquinanti

Scongiurata tale ipotesi, si è passati alla fase di classificazione binaria, utilizzando i seguenti algoritmi di classificazione:

- LogisticRegression
- DecisionTreeClassifier
- SVC
- RandomForestClassifier
- AdaBoostClassifier
- GradientBoostingClassifier
- LinearDiscriminantAnalysis

Per effettuare l’addestramento ed il test degli algoritmi, è necessario procedere alla suddivisione del dataset bilanciato in due parti. A tal proposito si è deciso di utilizzare il metodo dell’holdout, dove il training-set rappresenta l’80% dei dati mentre il restante 20% costituisce il test-set.

I risultati di accuratezza ottenuti dai diversi modelli sono stati:

- LogisticRegression: 0.87
- DecisionTreeClassifier: 0.86
- SVC: 0.90
- RandomForestClassifier: 0.88



- AdaBoostClassifier: 0.89
- GradientBoostingClassifier: 0.91
- LinearDiscriminantAnalysis: 0.86

Per visualizzare meglio i risultati dell'accuratezza ottenuti, viene riportata la rappresentazione grafica indicando anche per ogni risultato la media della varianza e la relativa deviazione standard per comprendere il range in cui il valore può variare o meno.

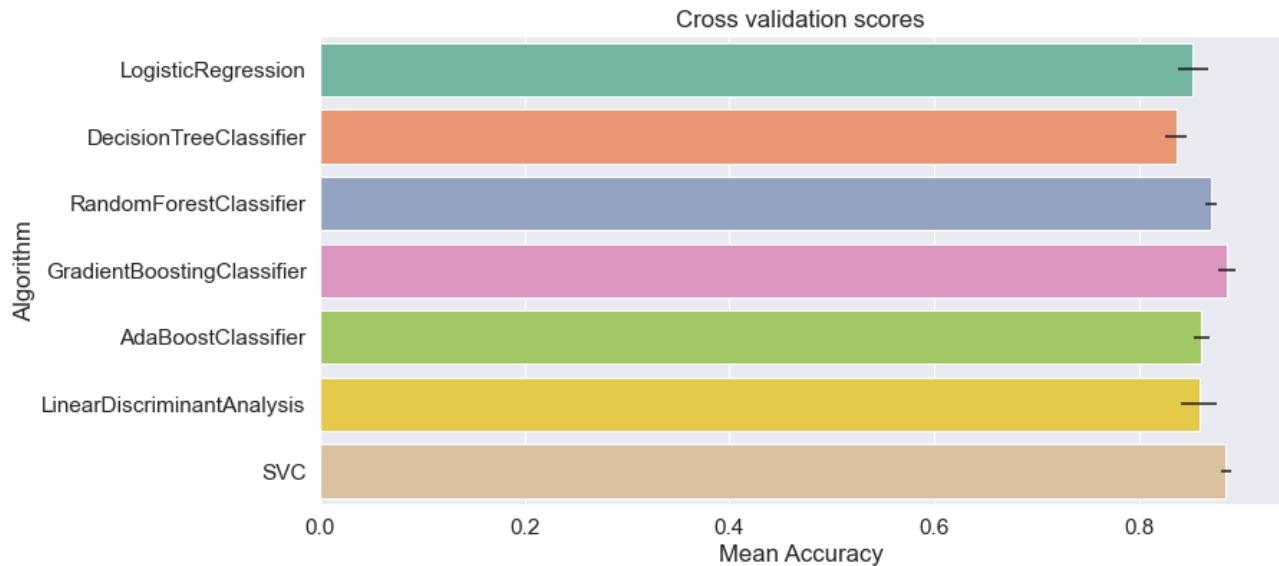


Figura 31 - BarPlot per la visualizzazione dei risultati dei classificatori

Sempre sulla base dei risultati ottenuti, vengono riportate le diverse matrici di confusione associate ad ogni classificatore.

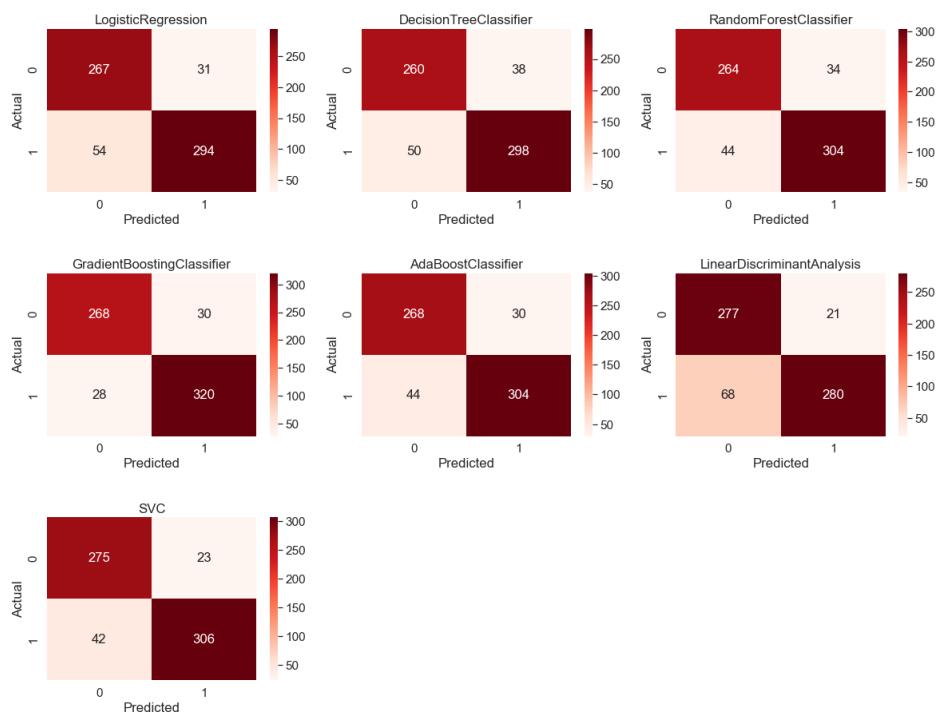


Figura 32 - Heatmap per la rappresentazione delle matrici di confusione per i diversi classificatori

Si osserva che il risultato migliore di accuratezza è stato ottenuto con il classificatore GradientBoostingClassifier.

	Precision	Recall	F1-Score	Support
Tollerabile	0.91	0.90	0.90	298
Non Tollerabile	0.91	0.92	0.92	348
Accuracy			0.91	646
Macro avg	0.91	0.91	0.91	646
Weighted avg	0.91	0.91	0.91	646

Figura 33 - Report del classificatore GradientBoostingClassifier

Si osserva che il classificatore ha un'affidabilità praticamente identica nel riconoscere elementi appartenenti alla categoria "Tollerabile" e "Non Tollerabile", infatti misura un valore di F1-Score nel primo caso pari a 0.90, mentre nel secondo caso pari a 0.92.

Un altro metodo che è stato utilizzato per valutare il modello ottenuto è mediante la **curva ROC**. Essa rappresenta un grafico che mostra le performance di un modello di classificazione considerando tutte le possibili soglie di classificazione. Tale curva rappresenta due parametri: *True Positive Rate* e *False Positive Rate*. Nel caso del modello di classificazione ottenuto, la ROC curve che si ottiene è rappresentata come:

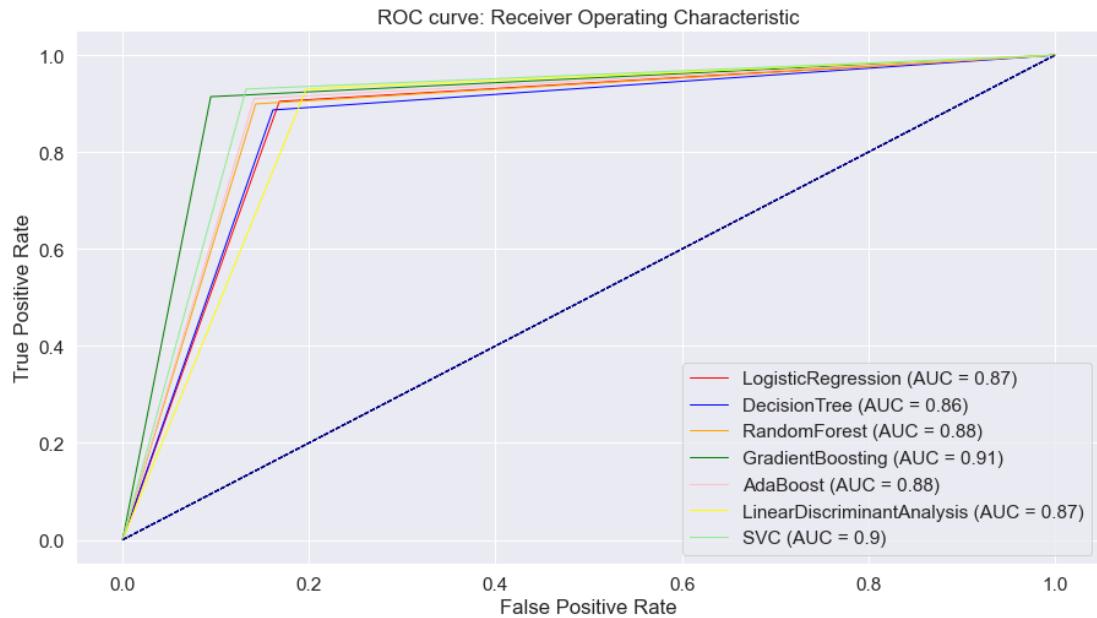


Figura 34 - ROC Curve

9.1.1 Classificazione con Grid Search

La valutazione dei vari classificatori permette di definire una heatmap, nella quale si possono mostrare le correlazioni fra i modelli utilizzati. Quindi, tramite la tabella riportata, si riesce a vedere quali classificatori restituiscono predizioni simili tra loro.

Come si può notare i classificatori *LogisticRegression* e *LinearDiscriminantAnalysis* hanno correlazioni molto alte, così come il *GradientBoostClassifier* e l'*SVC*; quindi, utilizzare uno piuttosto che l'altro non cambia molto le predizioni fatte.

Mediante tale grafico è anche possibile individuare i modelli che verranno utilizzati nell'analisi successiva. Infatti, un'ulteriore modalità di classificazione è mediante la tecnica **Grid Search**. È una tecnica che tenta di calcolare i valori ottimali degli iperparametri, mediante una ricerca esaustiva che viene eseguita sui valori dei parametri specifici di un modello. Alla luce della heatmap in Figura 35 si è deciso di utilizzare come classificatori il *DecisionTree* e il *GradientBoosting*, poiché tra loro si ha una bassa correlazione (0.81).

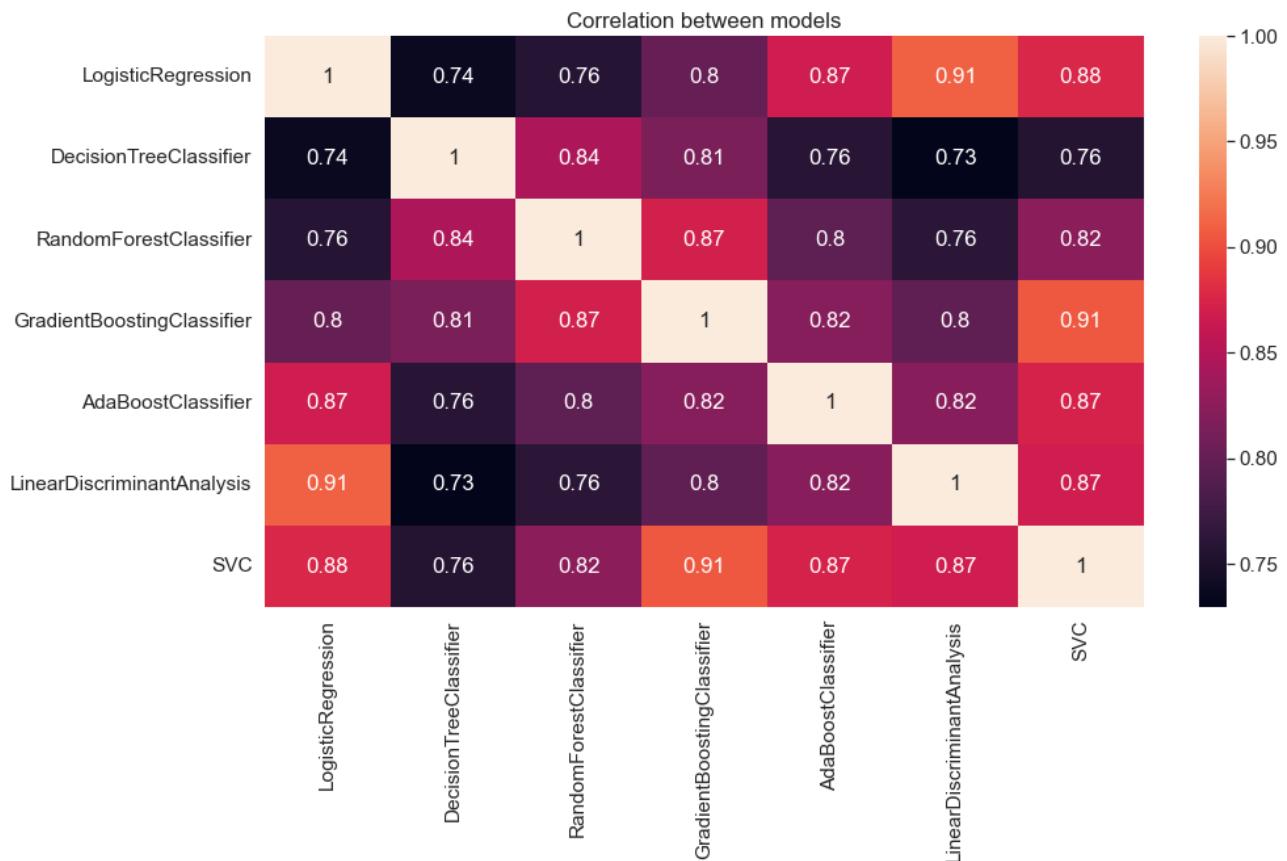


Figura 35 - Correlazioni tra modelli

Di seguito sono riportati gli iperparametri utilizzati:

```
#DecisionTree Iperparametri
DT_param = {"max_depth": [2,3,8,10],
            "max_features": [0.3, 0.7, 1],
            "min_samples_split": [2, 3, 10],
            "min_samples_leaf": [1, 3, 10],
            "criterion": ["gini"]}

#GradientBoosting Iperparametri
GB_param = {'loss' : ["deviance"],
            'n_estimators' : [100,200,300],
            'learning_rate': [0.1, 0.05, 0.01],
            'max_depth': [4, 8],
            'min_samples_leaf': [100,150],
            'max_features': [0.3, 0.1]}
```



In questo caso si è deciso di effettuare una suddivisione del dataset bilanciato utilizzando la “*k-fold cross validation*”, operazione che consiste nella suddivisione dell’insieme di dati totale in un numero k di parti di uguale numerosità e, ad ogni passo, la k-esima parte dell’insieme di dati utilizzata come set di test, mentre la restante costituisce sempre l’insieme di addestramento. Successivamente tutti i risultati vengono mediati. Questa operazione permette di diminuire drasticamente le possibilità di incorrere in problemi di overfitting. Eseguito l’addestramento sono stati valutati i modelli con e senza grid-search in termini di accuracy ed il risultato è stato il seguente:

	DecisionTreeClassifier	GradientBoostingClassifier
Accuratezza senza GridSearchCV	0.836	0.886
Accuratezza con GridSearchCV	0.879	0.887

Si può osservare che i modelli ottenuti sono migliori rispetto a quelli ottenuti effettuando un normale training, soprattutto per quanto riguarda il DecisionTree, infatti in entrambi i casi l’accuratezza è prossima all’88%, che rappresenta il risultato migliore che si era ottenuto al primo addestramento.

Infine, come ultima analisi, si è deciso di effettuare un **ensemble** dei modelli. Esso consiste nell’andare ad utilizzare più classificatori contemporaneamente per ottenere delle performance migliori rispetto ad utilizzare i singoli classificatori da soli. In questo caso, sempre alla luce di quanto fatto con la modalità di classificazione GridSearch, si è deciso di effettuare un ensemble dei classificatori DecisionTree e GradientBoosting. In questo caso, il livello di accuratezza che si ottiene è 0.902, quindi migliore rispetto a quello ottenuto con GridSearch, ma più affidabile dato che è stato ottenuto valutando più classificatori contemporaneamente.

9.1.2 Testing del modello

A questo punto procediamo con la fase di testing del modello, andando ad utilizzare la restante parte del dataset che non è stata per niente impiegata per la costruzione del modello ottenuto. Andiamo quindi a prendere tutti i dati sugli agenti inquinanti dal 2017 al 2018, seguendo le stesse procedure fatte durante la fase di addestramento. Sulla base di ciò, i risultati che otteniamo, dando in pasto al modello addestrato i nuovi dati:

Tollerabili: 234

Non tollerabili: 252

Si nota in questo caso che il numero di NON TOLLERABILI supera, se anche di pochissimo, il numero di TOLLERABILI. Abbiamo quindi proceduto con l’utilizzo della libreria *yellowbrik*, estensione della libreria Scikit-Learn per facilitare la selezione del modello e la messa a punto degli iperparametri. Di seguito viene riportata la ROC curve:

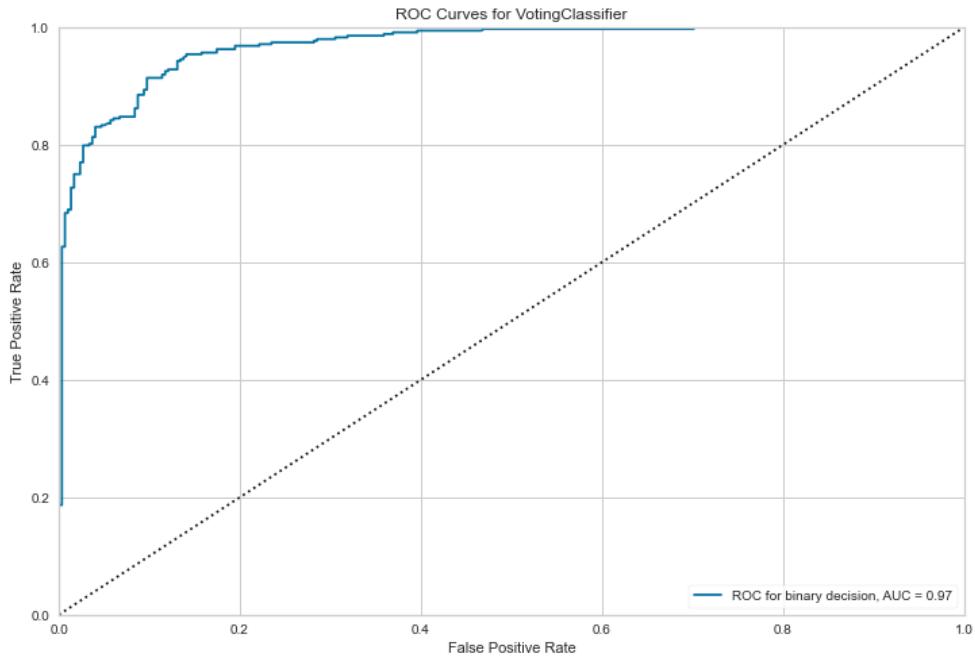


Figura 36 - ROC Curve for VotingClassifier

Di seguito viene riportata la Precision-Recall curve:

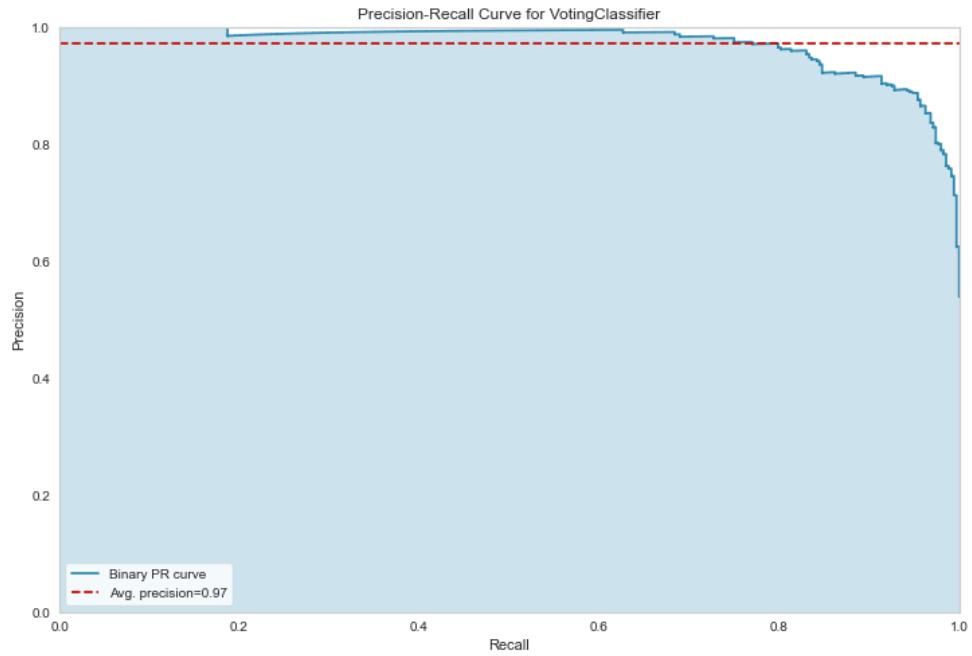


Figura 37 - Precision-Recall Curve for VotingClassifier

Riportiamo anche la Learning Curve:

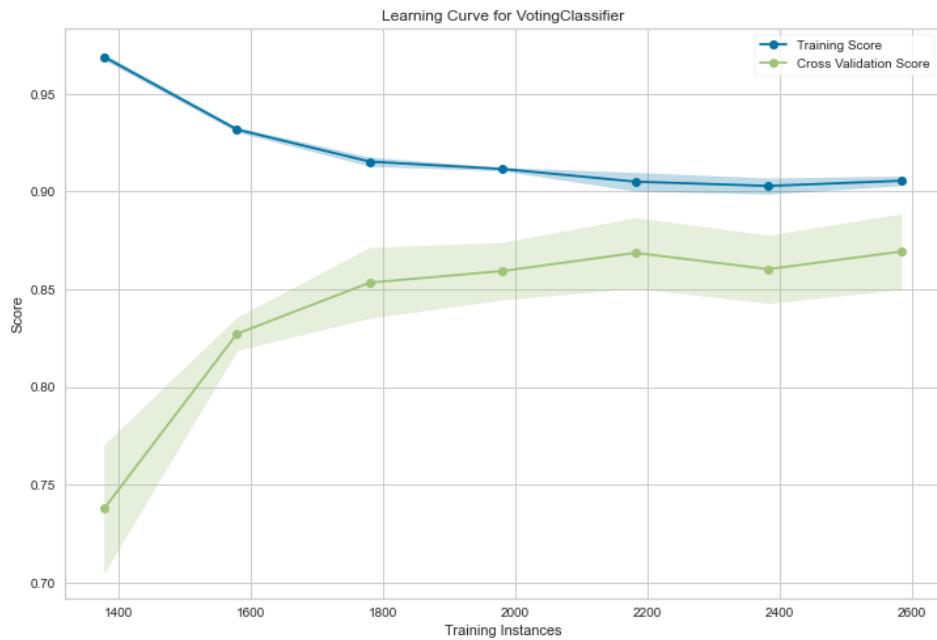


Figura 38 - Learning Curve for VotingClassifier

9.2 Classificazione multi-classe in base all' NO2

Si è scelto poi di effettuare una classificazione multi-classe sulla base del parametro NO2. In base a quanto visto precedentemente, l'NO2 è un gas irritante per l'apparato respiratorio e per gli occhi che può causare bronchiti fino anche a edemi polmonari e decesso.

Anche in questo caso abbiamo proceduto a dividere il dataset, utilizzando i dati raccolti fino al 2016 per addestrare il modello e, quelli raccolti nel periodo dal 2017 al 2018 per la fase di test. Anche in questo caso si è deciso di raggruppare i dati per giorno, essendo questi stati raccolti da diverse stazioni, con cadenza oraria. Si è poi proceduto nel seguente modo:

```
condition_list =[(Livello <= 30),((Livello > 30) & (Livello <= 50)),(Livello > 50)]
choicelist = [0,1,2]
Livello = np.select(condition_list, choicelist, default='Non Specificato')
```

Si è quindi ottenuto il seguente risultato:

Tollerabili: 883

Parzialmente tollerabili: 1366

Non tollerabili: 1039

Il passo successivo è stato quindi quello di controllare se il dataset fosse bilanciato o meno. Facendo questo si è visto che attualmente nel dataset sono presenti 883 giorni con un valore "TOLLERABILE", 1366 giorni con un valore "PARZIALMENTE TOLLERABILE" e 1039 giorni con un valore "NON TOLLERABILE". Alla luce di ciò si

è deciso di bilanciare il dataset considerando tutti i 883 giorni “TOLLERABILI”, ma scegliendo un sample della stessa misura per tutte le altre classi.

Dopo aver proceduto con queste operazioni preliminari si è potuto procedere con l'**analisi delle correlazioni** del dataset, con la speranza di trovare attributi che non fossero troppo correlati tra loro.

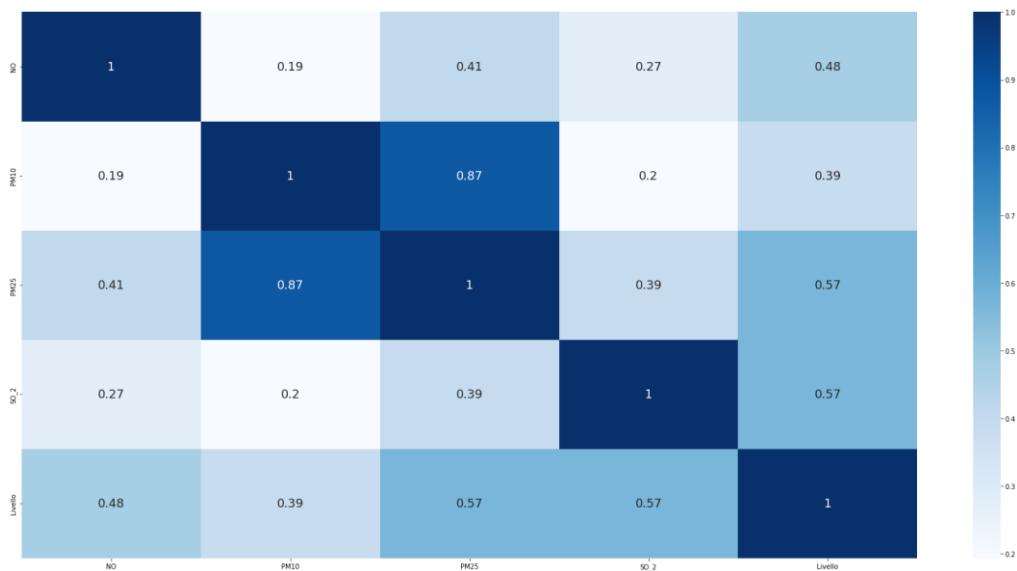


Figura 39 - Correlazione agenti inquinanti

Si è deciso di rimuovere l’agente inquinante PM25 in quanto presenta un’elevata correlazione con il PM10, ma comunque alte con i restanti agenti inquinanti. Il risultato è quindi il seguente:

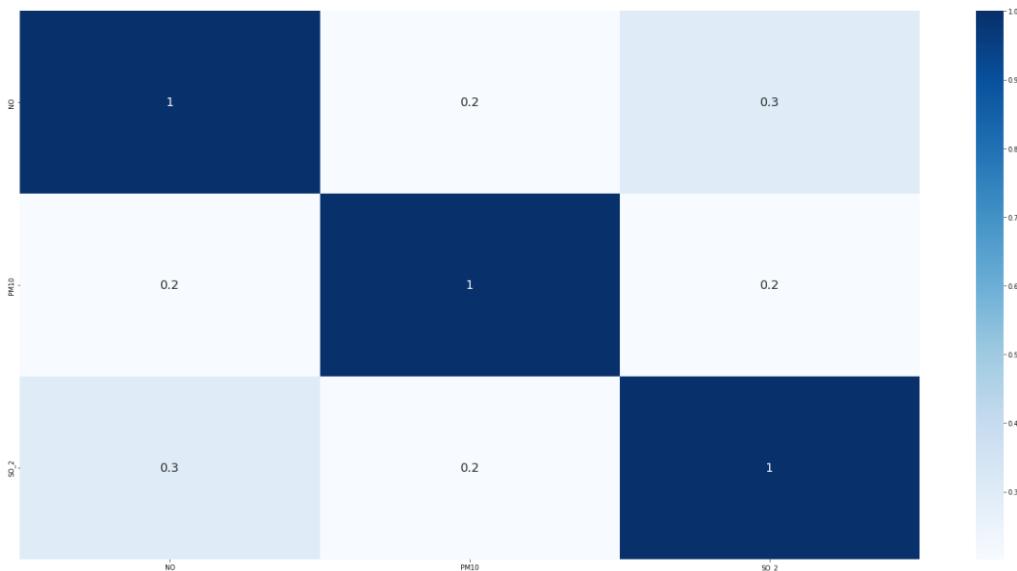


Figura 40 - Correlazione agenti inquinanti con rimozione di PM25

Si è quindi passati alla fase di classificazione, utilizzando i seguenti algoritmi:

- LogisticRegression
- DecisionTreeClassifier
- SVC
- RandomForestClassifier
- AdaBoostClassifier



- GradientBoostingClassifier
- LinearDiscriminantAnalysis

Per effettuare l'addestramento ed il test degli algoritmi, è necessario procedere alla suddivisione del dataset bilanciato in due parti. A tal proposito si è deciso di utilizzare il metodo dell'holdout, dove il training-set rappresenta l'80% dei dati mentre il restante 20% costituisce il test-set.

I risultati di accuratezza ottenuti dai diversi modelli sono stati:

- LogisticRegression: 0.73
- DecisionTreeClassifier: 0.74
- SVC: 0.80
- RandomForestClassifier: 0.76
- AdaBoostClassifier: 0.76
- GradientBoostingClassifier: 0.79
- LinearDiscriminantAnalysis: 0.68

Per visualizzare meglio i risultati dell'accuratezza ottenuti, viene riportata la rappresentazione grafica indicando anche per ogni risultato la media della varianza e la relativa deviazione standard per comprendere il range in cui il valore può variare o meno.

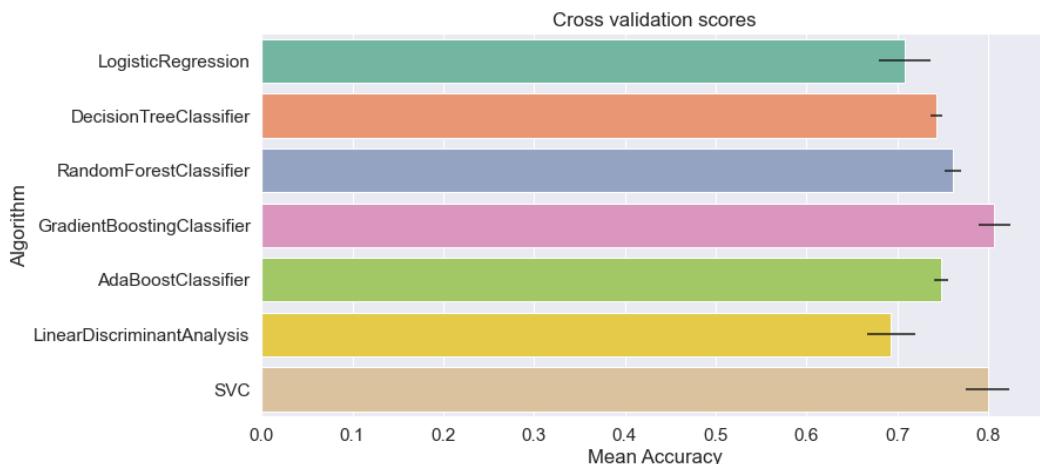


Figura 41 -BarPlot per la visualizzazione dei risultati dei classificatori

Sempre sulla base dei risultati ottenuti, vengono riportate le diverse matrici di confusione associate ad ogni classificatore.

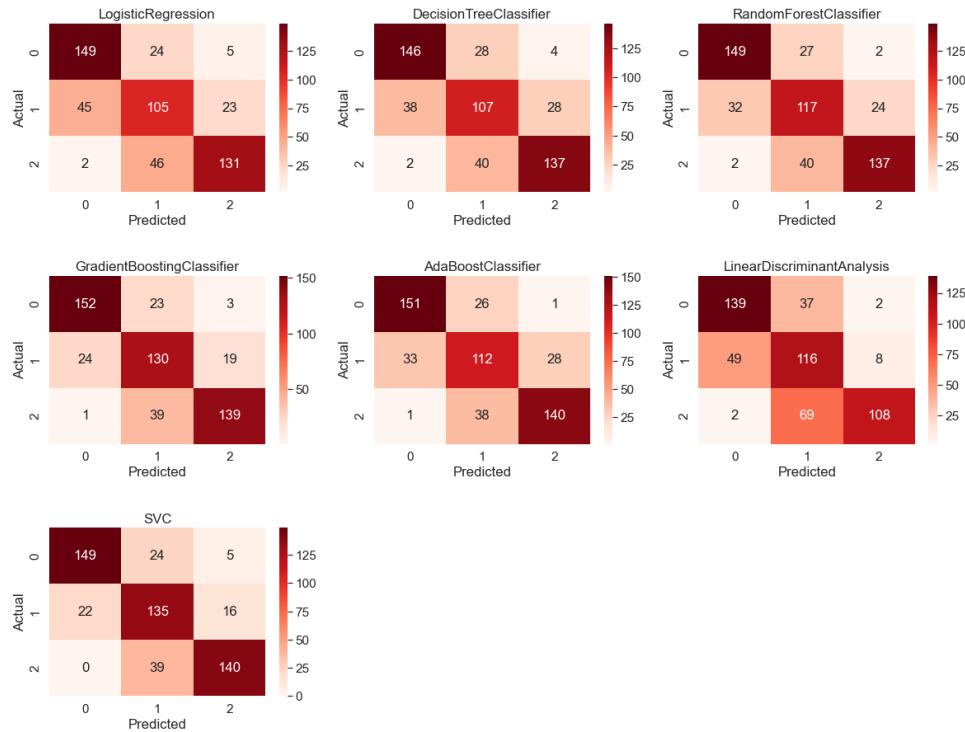


Figura 42 - Heatmap per la rappresentazione delle matrici di confusione per i diversi classificatori

Si osserva che il risultato migliore di accuratezza è stato ottenuto con il classificatore SVC.

	Precision	Recall	F1-Score	Support
Tollerabile	0.87	0.84	0.85	178
Parzialmente Tollerabile	0.68	0.78	0.73	173
Non Tollerabile	0.87	0.78	0.82	179
Accuracy			0.80	530
Macro avg	0.81	0.80	0.80	530
Weighted avg	0.81	0.80	0.80	530

Figura 43 - Report del classificatore SVC

Si osserva che il classificatore ha una buona affidabilità nel riconoscere gli elementi appartenenti alla classe "TOLLERABILE", minore invece nel caso delle altre due classi.

9.2.1 Classificazione con GridSearch

La valutazione dei vari classificatori permette di definire una heatmap, nella quale si possono mostrare le correlazioni fra i modelli utilizzati. Quindi, tramite la tabella riportata, si riesce a vedere quali classificatori restituiscono predizioni simili tra loro.

Come si può notare i classificatori *RandomForestClassifier* e *DecisionTreeClassifier* hanno correlazioni molto alte, così come il *GradientBoostingClassifier* e l'*SVC*; quindi, utilizzare uno piuttosto che l'altro non cambia molto le predizioni fatte.

Mediante tale grafico è anche possibile individuare i modelli che verranno utilizzati nell'analisi successiva. Infatti, un'ulteriore modalità di classificazione è mediante la tecnica **Grid Search**. È una tecnica che tenta di

calcolare i valori ottimali degli iperparametri, mediante una ricerca esaustiva che viene eseguita sui valori dei parametri specifici di un modello. Alla luce della heatmap in Figura 44 si è deciso di utilizzare come classificatori il DecisionTree e il GradientBoosting, poiché tra loro si ha una bassa correlazione (0.86).

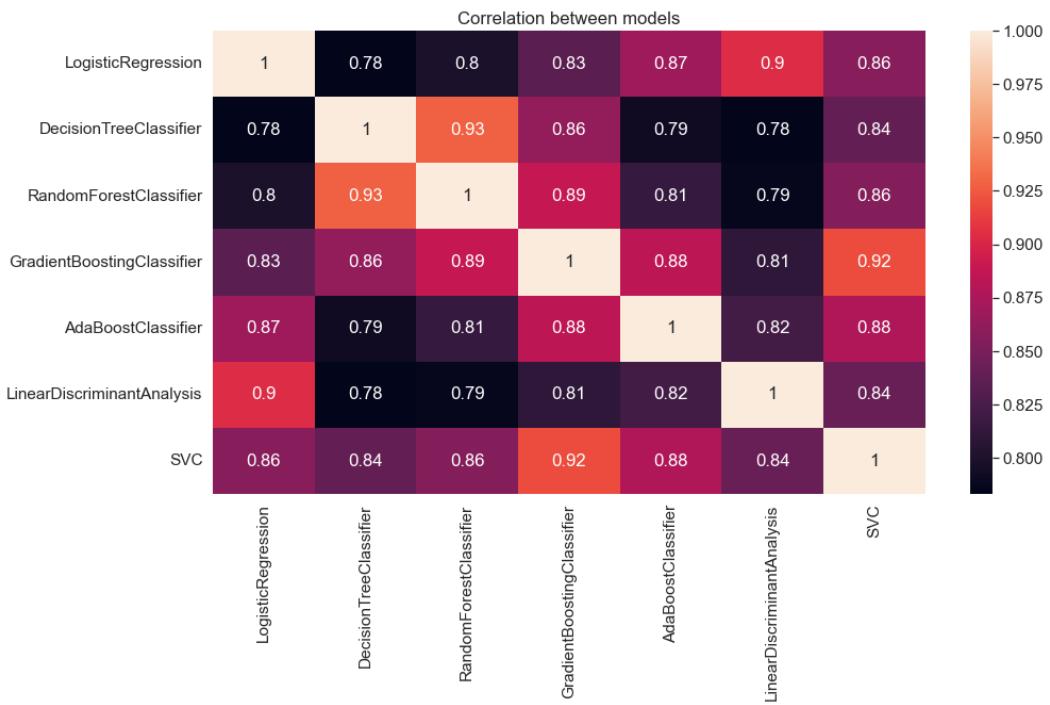


Figura 44 - Correlazioni tra modelli

Di seguito sono riportati gli iperparametri utilizzati:

```
#DecisionTree Iperparametri
DT_param = {"max_depth": [2,3,8,10],
            "max_features": [0.3, 0.7, 1],
            "min_samples_split": [2, 3, 10],
            "min_samples_leaf": [1, 3, 10],
            "criterion": ["gini"]}

#GradientBoosting Iperparametri
GB_param = {'loss' : ["deviance"],
            'n_estimators' : [100,200,300],
            'learning_rate': [0.1, 0.05, 0.01],
            'max_depth': [4, 8],
            'min_samples_leaf': [100,150],
            'max_features': [0.3, 0.1]}
```

In questo caso si è deciso di effettuare una suddivisione del dataset bilanciato utilizzando la “*k-fold cross validation*”, operazione che consiste nella suddivisione dell’insieme di dati totale in un numero k di parti di uguale numerosità e, ad ogni passo, la k-esima parte dell’insieme di dati utilizzata come set di test, mentre la restante costituisce sempre l’insieme di addestramento. Successivamente tutti i risultati vengono mediati. Questa operazione permette di diminuire drasticamente le possibilità di incorrere in problemi di overfitting. Eseguito l’addestramento sono stati valutati i modelli con e senza grid-search in termini di accuracy ed il risultato è stato il seguente:



	DecisionTreeClassifier	GradientBoostingClassifier
Accuratezza senza GridSearchCV	0.743	0.807
Accuratezza con GridSearchCV	0.797	0.811

Si può osservare che i modelli ottenuti sono migliori rispetto a quelli ottenuti effettuando un normale training, soprattutto per quanto riguarda il DecisionTree, infatti in entrambi i casi l'accuratezza è prossima all'80%, che rappresenta il risultato migliore che si era ottenuto al primo addestramento.

Infine, come ultima analisi, si è deciso di effettuare un **ensemble** dei modelli. Esso consiste nell'andare ad utilizzare più classificatori contemporaneamente per ottenere delle performance migliori rispetto ad utilizzare i singoli classificatori da soli. In questo caso, sempre alla luce di quanto fatto con la modalità di classificazione GridSearch, si è deciso di effettuare un ensemble dei classificatori DecisionTree e GradientBoosting. In questo caso, il livello di accuratezza che si ottiene è 0.798, quindi minore rispetto a quello ottenuto con GridSearch, ma più affidabile dato che è stato ottenuto valutando più classificatori contemporaneamente.

9.2.2 Testing del modello

A questo punto procediamo con la fase di testing del modello, andando ad utilizzare la restante parte del dataset che non è stata per niente impiegata per la costruzione del modello ottenuto. Andiamo quindi a prendere tutti i dati sugli agenti inquinanti dal 2017 al 2018, seguendo le stesse procedure fatte durante la fase di addestramento. Sulla base di ciò, i risultati che otteniamo, dando in pasto al modello addestrato i nuovi dati:

Tollerabili: 139

Parzialmente Tollerabili: 155

Non Tollerabili: 192

Si nota in questo caso che il numero di NON TOLLERABILI supera il numero dei PARZIALMENTE TOLLERABILI. Abbiamo quindi proceduto con l'utilizzo della libreria *yellowbrik*, estensione della libreria Scikit-Learn per facilitare la selezione del modello e la messa a punto degli iperparametri.

Di seguito viene riportata la curva Precision-Recall:

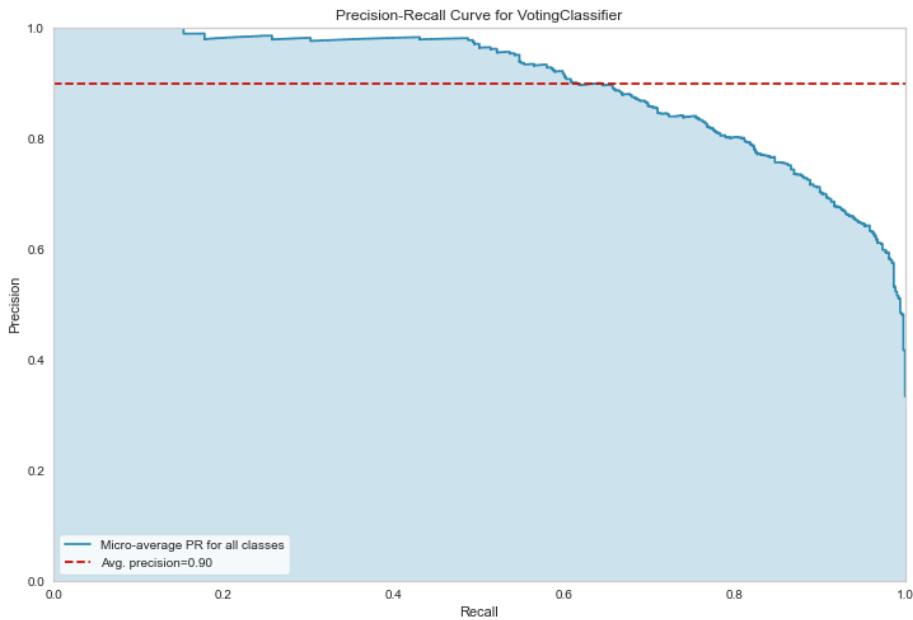


Figura 45 - Precision-Recall Curve for VotingClassifier

Riportiamo anche la Learning Curve:

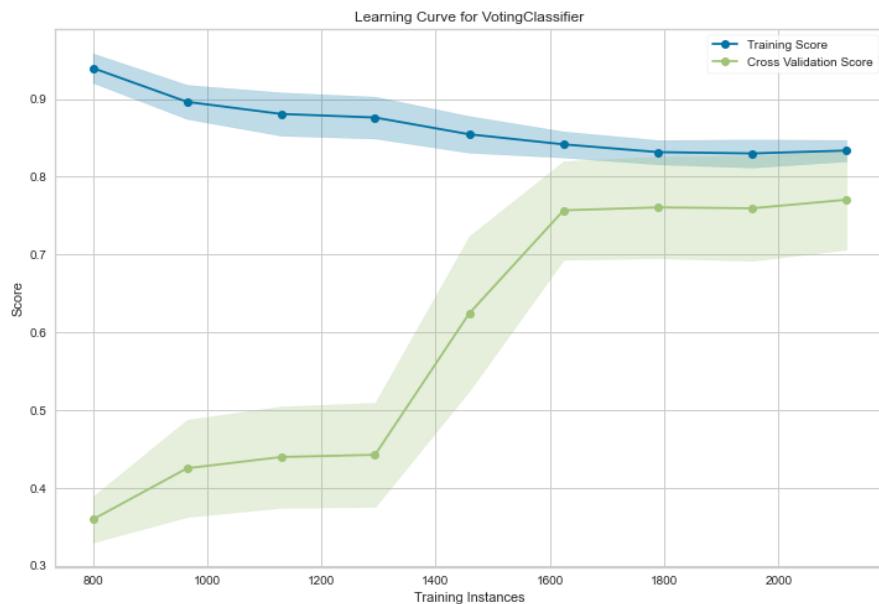


Figura 46 - Learning Curve for VotingClassifier

10 Serie temporale

Tratteremo in questo capitolo le serie temporali, i trend e le previsioni che possono essere realizzate con specifici algoritmi. Per fare ciò ci serviremo della libreria Statsmodels di Python, mediante la quale potremo esplorare dati, eseguire test statistici e stimare modelli statistici.



Continueremo nell'analisi del dataset introdotto nei primi capitoli in particolare le colonne che utilizzeremo per le analisi saranno ovviamente la colonna 'date' contenente il giorno nel quale è stata effettuata la rilevazione e le colonne 'SO_2', 'NO', 'NO_2', 'PM10' e 'PM25' relative alla quantità di agente presente nell'aria alla data.

10.1 ETL

Una passo che abbiamo dovuto compiere prima di iniziare ad analizzare il dataset è stato quello di eseguire una breve fase di ETL.

Abbiamo manipolato la colonna 'date' utilizzando la funzione di Pandas to_datetime, in maniera tale da convertire i valori nel tipo datetime, successivamente abbiamo normalizzato, con la funzione normalize, le date in maniera tale che gli orari risultassero sempre gli stessi (ovvero 00:00:00) e questo per eseguire poi un groupby in maniera tale da avere un dato medio riguardante una singola giornata.

```
allMadrid = pd.read_csv(path+"out.csv")
allMadrid['date'] = pd.to_datetime(allMadrid['date'])
allMadrid['date']=allMadrid['date'].dt.normalize()
allMadrid=allMadrid.groupby(['date']).mean().drop(columns=['station'])
display(allMadrid)
```

✓ 4.6s

	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL
date												
2008-01-01	1.314348	0.652554	1.040163	0.314783	11.820078	67.084565	9.280977	39.855419	25.650870	19.815357	1.656304	4.026522
2008-01-02	1.128177	0.531806	1.016719	0.270625	11.820078	66.552372	9.750819	21.005401	15.767895	14.897970	1.536750	3.857708
2008-01-03	0.727969	0.360000	0.942292	0.229083	11.820078	50.410930	30.313847	9.173423	5.617396	10.923919	1.400333	2.911875
2008-01-04	0.907969	0.458993	1.062031	0.257647	11.820078	58.247292	20.756432	21.390939	13.236000	12.875144	1.409706	4.055104
2008-01-05	0.823750	0.389635	0.785260	0.238250	11.820078	47.282292	20.405407	18.745682	12.485729	11.804503	1.482792	3.244219
...

Figura 47 - Fase di ETL

10.2 Analisi della serie

Ci siamo concentrati sul principale agente inquinante, ovvero il NO₂, abbiamo quindi estratto dal dataframe con tutti i dati, solamente quelli necessari, e abbiamo poi mostrato l'andamento generale, nel corso degli anni, attraverso un opportuno grafico.

Abbiamo estratto i dati relativi all'agente NO e mostrato il suo andamento generale.

```
forNo = allMadrid[ 'NO_2' ]
forNo.plot(figsize=(15, 2))
```

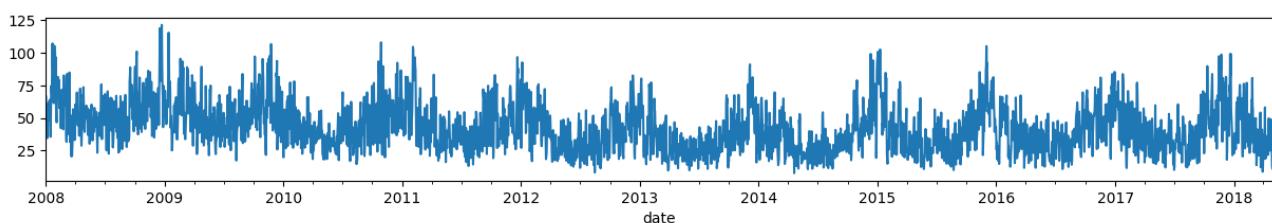


Figura 48 - Andamento generale NO₂

10.3 Stazionarietà

Abbiamo valutato la stazionarietà delle varie serie.

La stazionarietà è una proprietà delle serie temporali che indica se i valori non dipendono dal tempo, in una serie stazionaria la media e la varianza devono essere costanti nel tempo.

Per fare ciò abbiamo utilizzato l'augmented dickey fuller test. L'ipotesi nulla dell'ADF test è che la serie sia non stazionaria quindi se il valore p-value minore di 0.05 allora si rifiuta l'ipotesi nulla e si può affermare che la serie è stazionaria. Nel caso della serie in analisi i risultati del test sono i seguenti:

- ADF Statistic: -5.235168
- p-value: 0.000007
- Critical Values:
 - 1%: -3.432
 - 5%: -2.862
 - 10%: -2.567

Quindi la serie è stazionaria e si può procedere, prendendo il parametro $D=1$.

10.4 Autocorrelazione e autocorrelazione parziale

Per utilizzare il modello SARIMAX, è stato poi necessario ricavare i parametri p e q . Per stimare il parametro p si dovrà ricavare l'autocorrelazione parziale. Analogamente, per ricavare il parametro q , si è dovrà ricavare l'autocorrelazione.

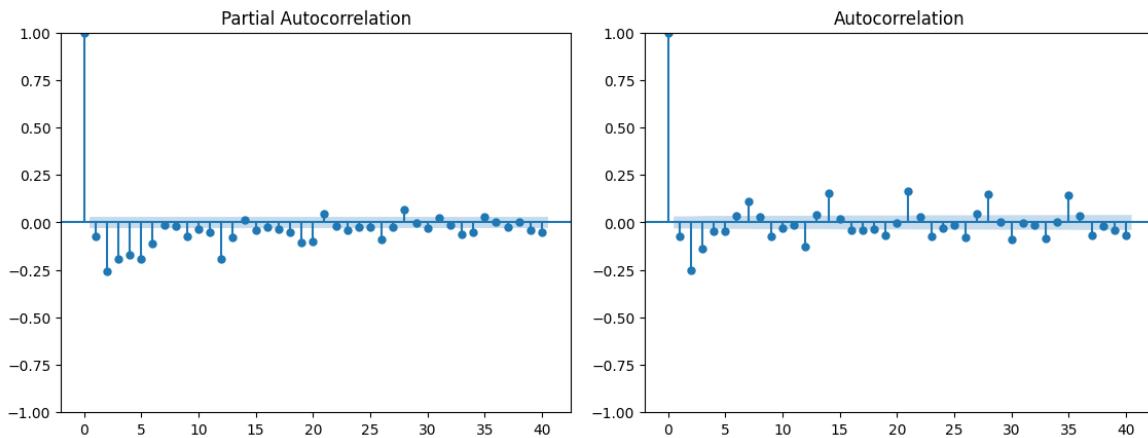


Figura 49 - Autocorrelazione parziale e Autocorrelazione No2

Dai seguenti grafici abbiamo quindi ricavato un parametro $p = 6$ ed un parametro $q = 5$, e per la stagionalità un parametro $P = 1$ e $Q = 2$.

E tramite una funzione messa a disposizione da statsmodel, la seasonal_decompose si evidenziano come mostrato nella figura a seguire le componenti trend e stagionalità che caratterizzano la serie.

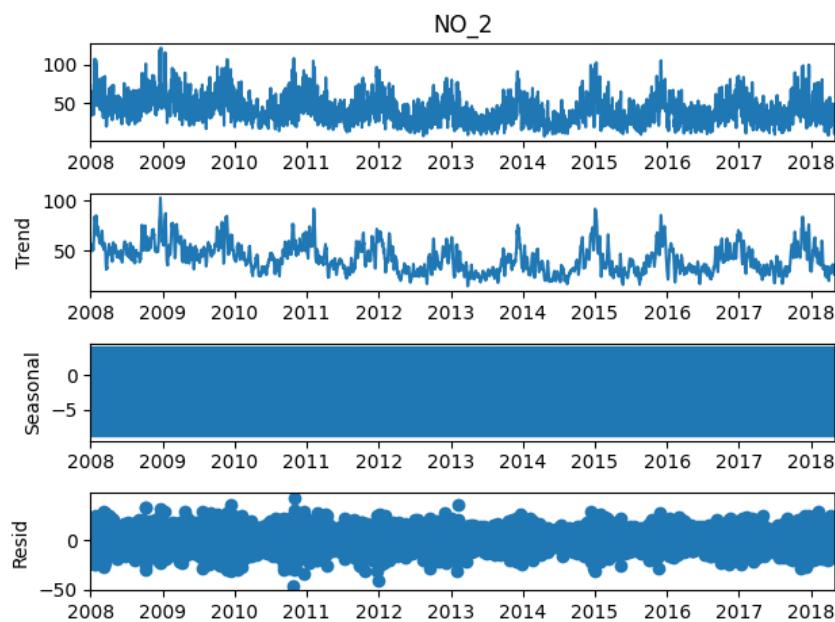


Figura 50 - Decomposizione della serie temporale relativa al No2

10.5 Modello ARIMA

Si è proceduto creando un modello ARIMA, utilizzando i valori, ottenuti osservando i grafici sopra.

- p : ordine del termine AR (Auto Regressive);
- q : ordine del termine MA (Moving Average);
- d : numero di differenziazioni necessarie per rendere la serie stazionaria.

Per l'allenamento di tale modello si è usato un training set che è costituito dal 80% del nostro dataset, mentre il restante 20% è stato utilizzato per testare il modello risultante.

Il modello ARIMA utilizzato è stato il seguente:

$$ARIMA: (1,0,2)\times(6,1,5,12)$$

Il modello risultante è il seguente:

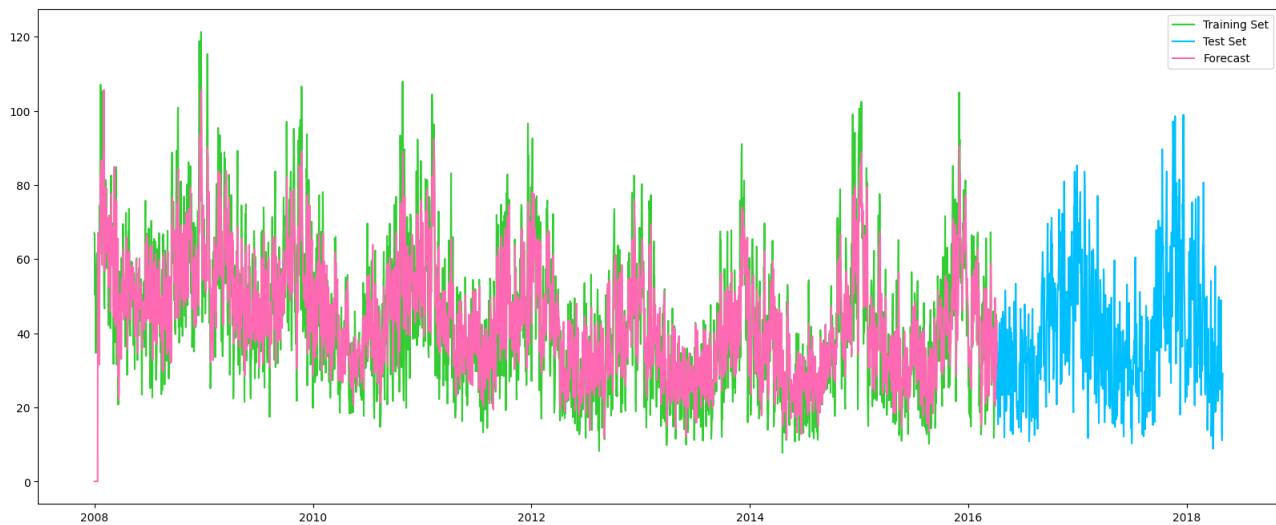


Figura 51 - MODELLO 1

Si è poi pensato di limitare il campione ad un periodo più recente in quanto nell'ultimo periodo sono stati presi numerosi provvedimenti per limitare le emissioni di sostanze inquinanti, e quindi rispetto ai primi elementi del campione, gli altri erano mediamente più bassi.

Il modello risultante è il seguente:

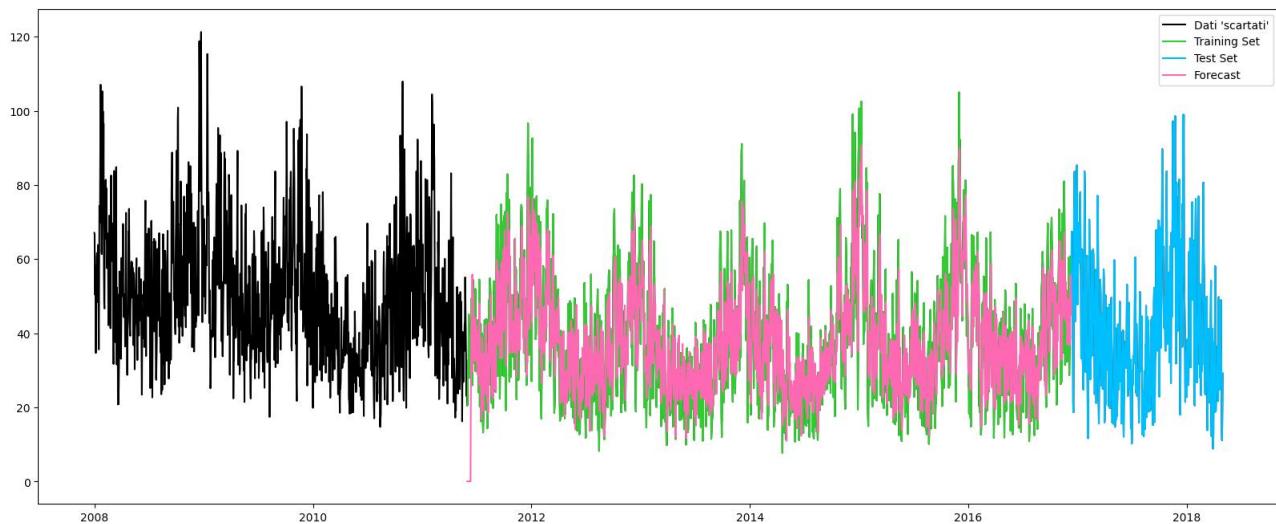


Figura 52 - MODELLO 2



Abbiamo testato poi anche altri parametri e quelli che ci è parso restituissero risultati leggermente migliori, sono quelli espressi nel seguente modello:

$$ARIMA: (0,0,0)\times(5,1,5,12)$$

E lo abbiamo allenato sia con il dataset completo, e il risultato è stato il seguente:

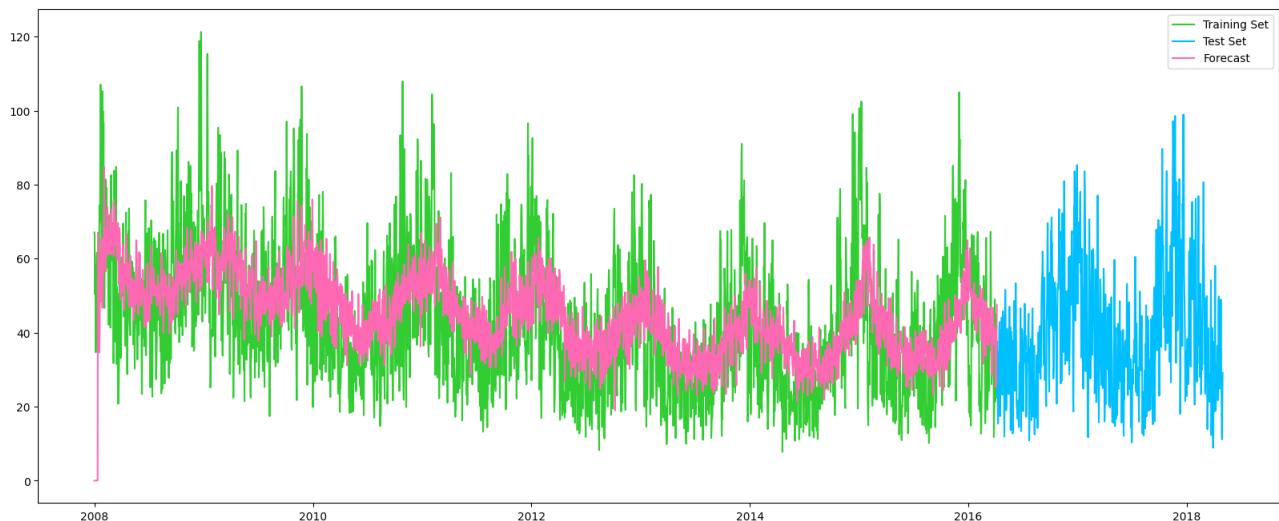


Figura 53 - MODELLO 3

Che con il dataset "parziale" e il risultato è stato il seguente:

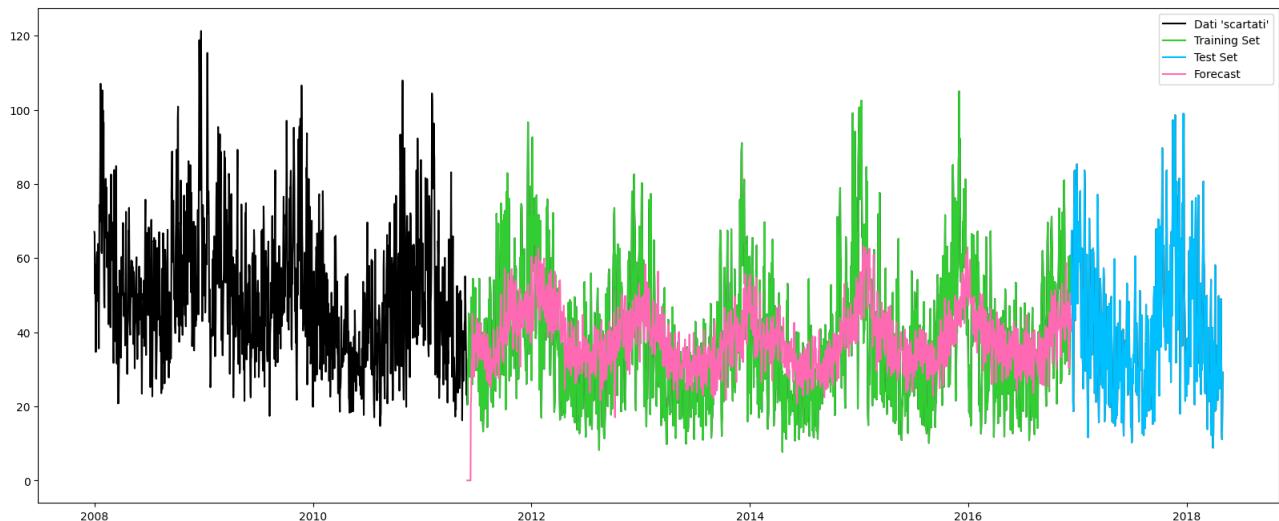


Figura 54 - MODELLO 4

10.6 Predizione in-sample

Siamo poi passati una volta estratto il modello ad effettuare una predizione in-sample, ovvero si vanno a predire valori di cui però si conosce anche il valore vero, per poter calcolare le metriche e capire quindi le prestazioni del modello.

La predizione in-sample effettuata con il primo modello restituisce il seguente risultato:

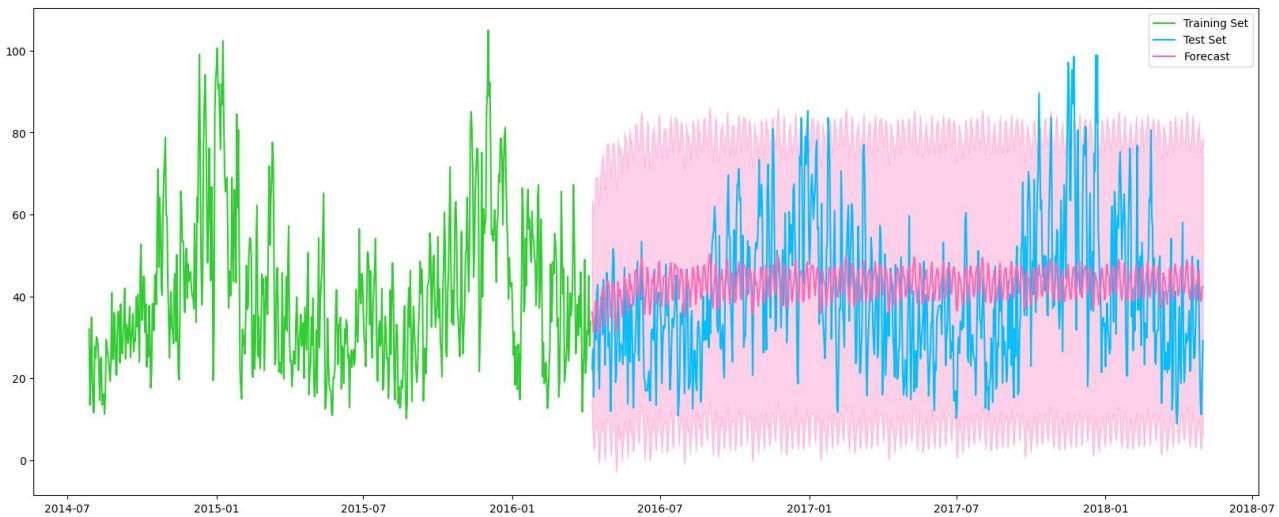
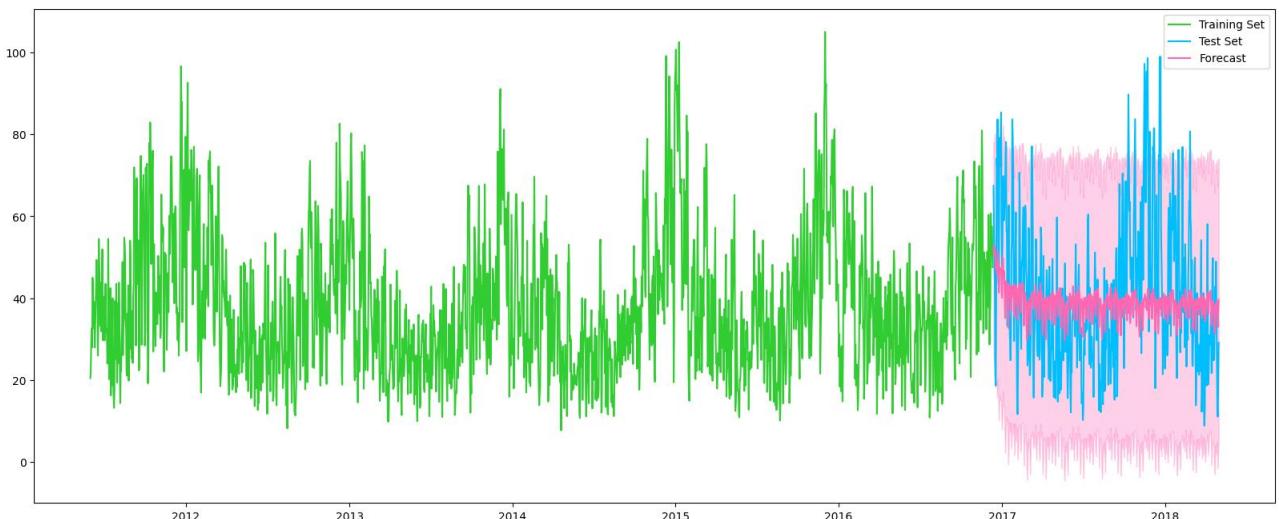


Figura 55 - MODELLO 1: predizione in-sample

La predizione in-sample effettuata con il secondo modello restituisce il seguente risultato:





La predizione in-sample effettuata con il terzo modello restituisce il seguente risultato:

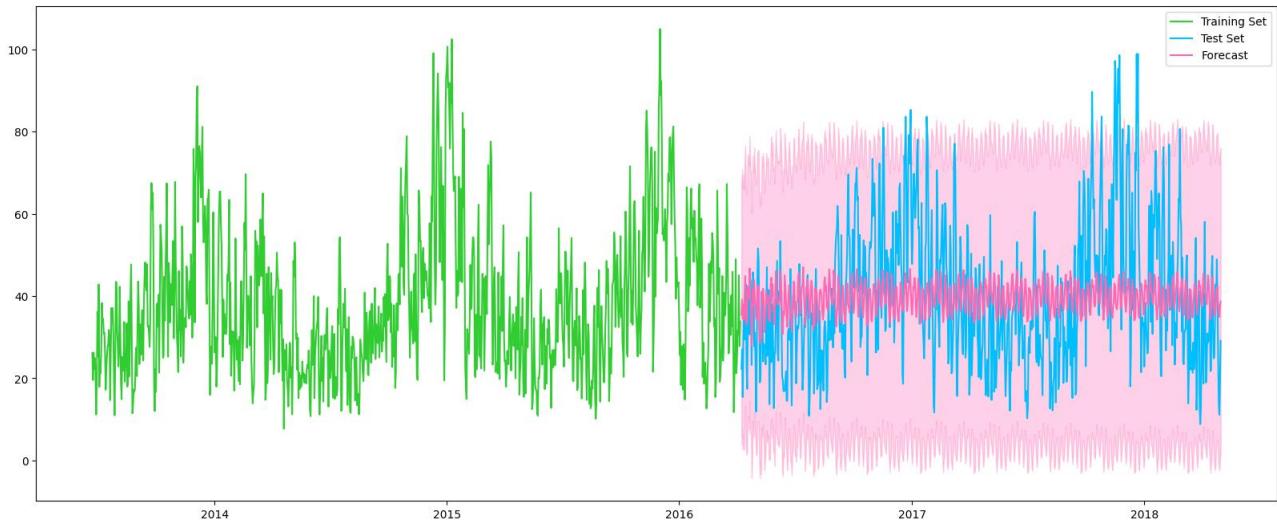


Figura 57 - MODELLO 3: predizione in-sample

La predizione in-sample effettuata con il quarto modello restituisce il seguente risultato:

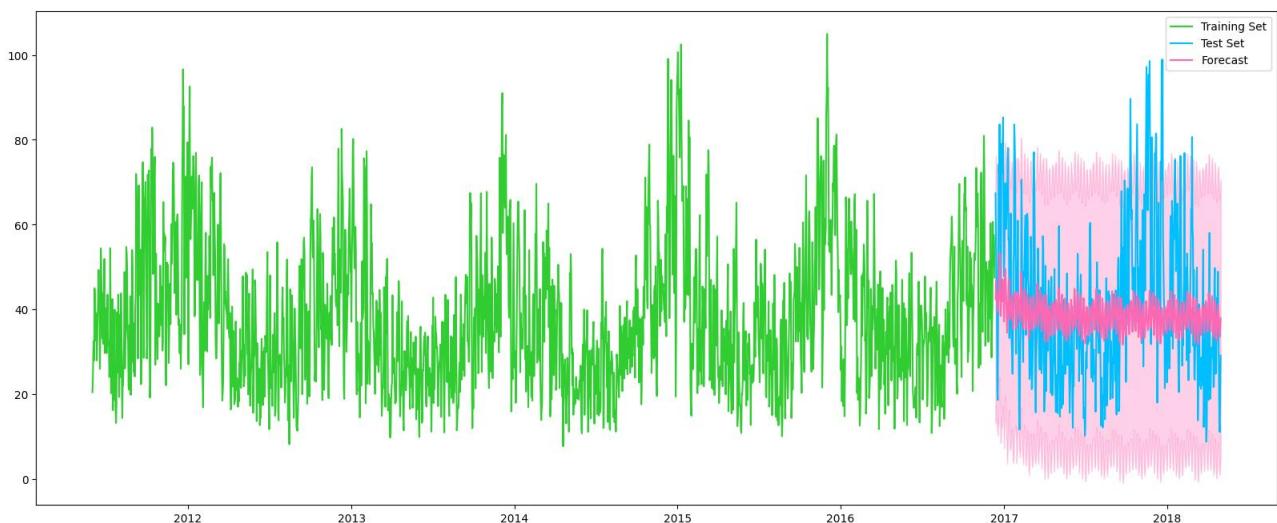


Figura 58 - MODELLO 4: predizione in-sample

10.7 Metriche di valutazione

Abbiamo proceduto a valutare le predizioni ottenute utilizzando le metriche che abbiamo calcolato mediante la libreria `sklearn.metrics`. In seguito, riporteremo le metriche risultanti dalle predizioni effettuate con entrambi i modelli e le commenteremo.

Nota che le prime tre metriche sono da minimizzare mentre la quarta è invece da massimizzare.

10.7.1 MAE

MODELLO 1	MODELLO 2	MODELLO 3	MODELLO 4
13.840837069935771	13.908937984029043	13.327795562584788	13.992743700583356

Tabella 1 - MAE



Rappresenta la media delle differenze assolute tra i valori predetti e i valori reali e da quindi un'idea della grandezza dell'errore, però non da informazioni sulla direzione di tale errore, ovvero se c'è una sovrastima o una sottostima.

Come si può vedere nella tabella sopra, *Tabella 1* si hanno risultati leggermente migliori con il modello 3.

10.7.2 MAPE

MODELLO 1	MODELLO 2	MODELLO 3	MODELLO 4
0.4602011561737929	0.40164587959687215	0.41104805432926617	0.40471265930951833

Tabella 2 – MAPE

Riporta ad un valore percentuale il valore del MAE.

Come si può vedere nella tabella sopra, *Tabella 2* si hanno risultati leggermente migliori con il modello 2.

10.7.3 MSE

MODELLO 1	MODELLO 2	MODELLO 3	MODELLO 4
284.2761235346021	318.310318343035	280.4565997370065	320.9132680250848

Tabella 3 – MSE

È molto simile all'errore assoluto medio (MAE) in quanto fornisce soltanto un valore sull'entità dell'errore.

Come si può vedere nella tabella sopra, *Tabella 3* si hanno risultati leggermente migliori con il modello 3.

10.7.4 R²

MODELLO 1	MODELLO 2	MODELLO 3	MODELLO 4
0.05278995719654589	0.0718756612435465	0.0655166373511279	0.06428602052739085

Tabella 41 - R²

La metrica R² (o R-Squared) fornisce un'indicazione sulla bontà di adattamento di un insieme di previsioni ai valori reali. Nella libreria sklearn.metrics può assumere anche valori negativi, ma generalmente ha un valore che va da 0 a 1, e più alto è il valore più l'andamento è perfetto.

Quindi quando il risultato di tale metrica è 0 il modello utilizzato non spiega per nulla i dati; quando è 1 il modello, invece, spiega perfettamente i dati.

Come si può vedere nella tabella sopra, *Tabella 4* si hanno risultati migliori con il modello 2, comunque tutti i modelli non spiegano in maniera eccelsa i dati, anzi i risultati sono alquanto deludenti.

10.8 Predizione out-sample

La predizione out-sample effettuata con il primo modello restituisce i seguenti risultati:

La predizione out-sample effettuata con il primo modello restituisce i seguenti risultati:

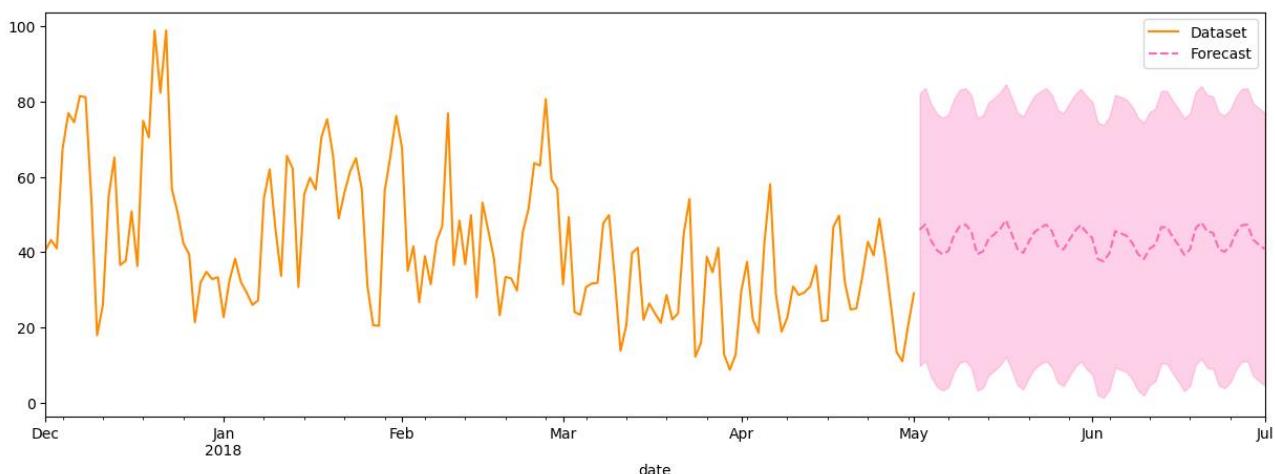


Figura 59 - MODELLO 1: predizione out-sample



La predizione out-sample effettuata con il secondo modello restituisce i seguenti risultati:

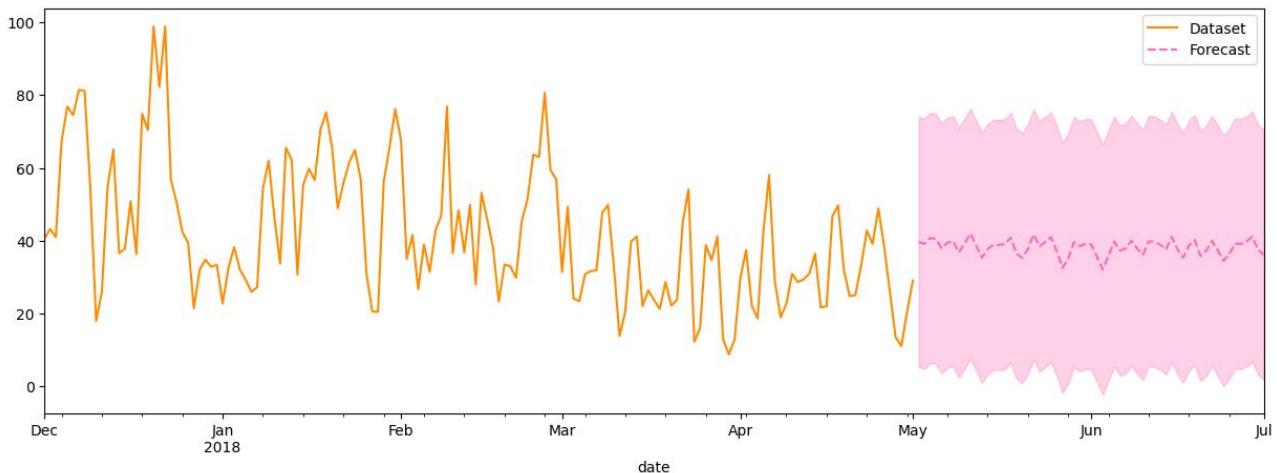


Figura 60 - MODELLO 2: predizione out-sample

La predizione out-sample effettuata con il terzo modello restituisce i seguenti risultati:

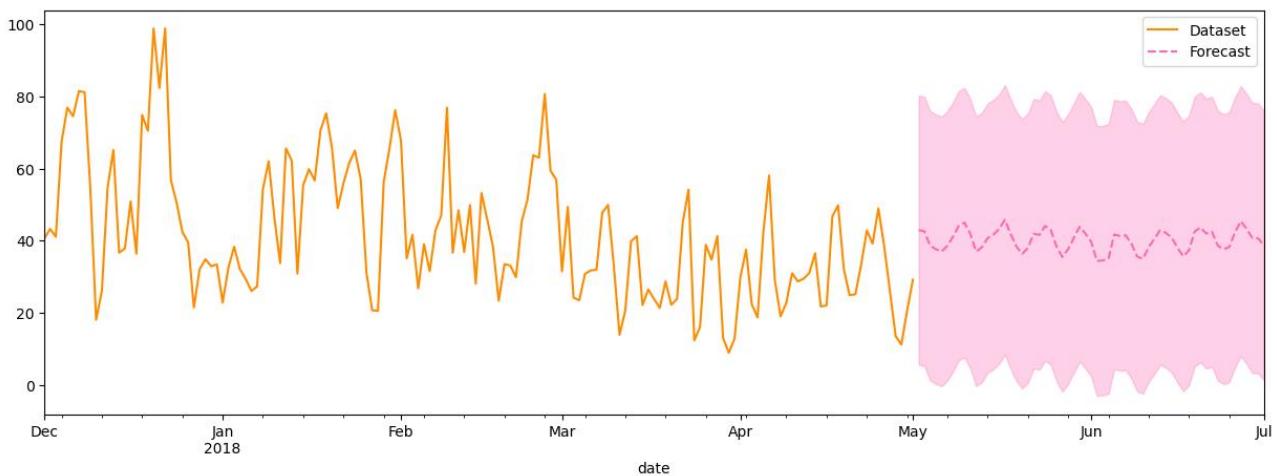


Figura 61 - MODELLO 3: predizione out-sample

La predizione out-sample effettuata con il quarto modello restituisce i seguenti risultati:

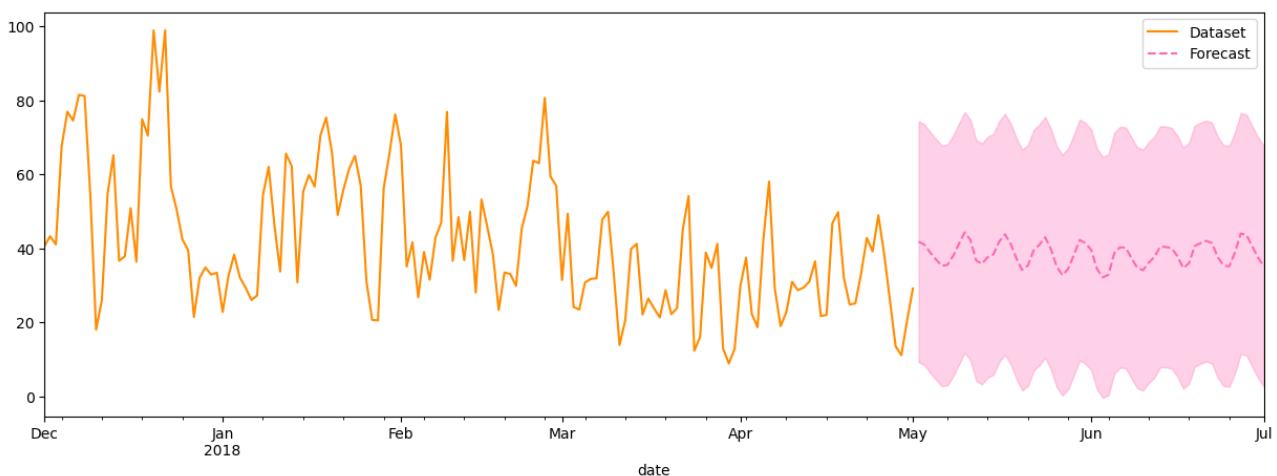


Figura 62 - MODELLO 4: predizione out-sample

10.9 Miglioramenti

Non soddisfatti dei risultati ottenuti abbiamo pensato di raggruppare mensilmente i dati per rendere più facile la modellazione e il forecasting, ed effettivamente i risultati sono stati notevolmente migliori.

L'andamento della serie in analisi è ora, il seguente:

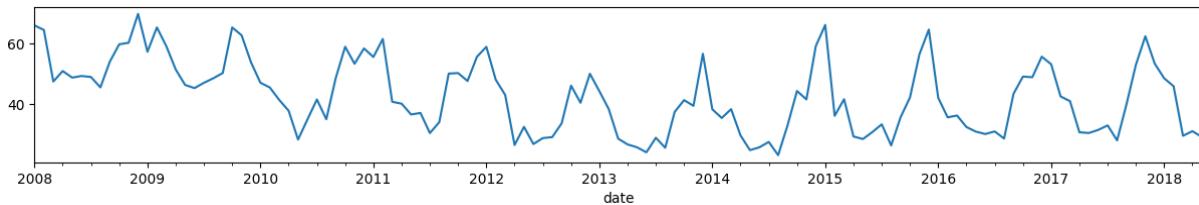


Figura 63 - Andamento generale No2 mensile

10.9.1 Stazionarietà

Anche in questo caso abbiamo effettuato l'augmented dickey fuller test, ed i risultati ottenuti sono stati i seguenti:

- ADF Statistic: -2.259714
- p-value: 0.185290
- Critical Values:
 - 1%: -3.491
 - 5%: -2.888
 - 10%: -2.581

Quindi essendo il p-value minore di 0.05 si rifiuta l'ipotesi nulla e si può affermare che la serie è stazionaria. Si è deciso quindi di prendere il parametro $D = 0$.

10.9.2 Autocorrelazione e autocorrelazione parziale

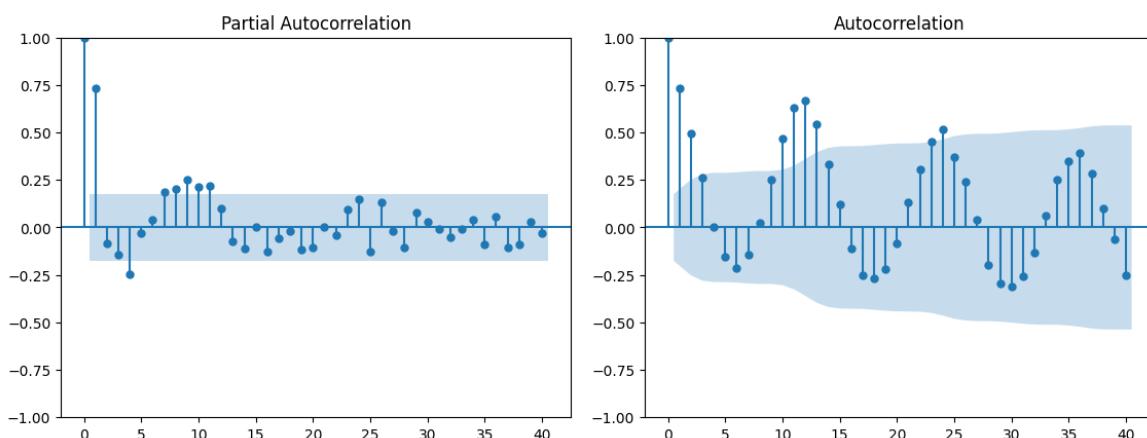


Figura 64 - Autocorrelazione parziale e Autocorrelazione No2 mensile

E da questi grafici riusciamo a dedurre i parametri p e q , che ci serviranno per implementare il modello. In particolare, dal grafico di correlazione ricaviamo il parametro $p = 1$, in quanto, guardando il grafico di autocorrelazione parziale (escluso il primo) vi è un solo lag fuori dalla soglia, prima del primo che vi è all'interno. In maniera analogia si prende $q = 2$, in quanto, guardando il grafico di autocorrelazione (escluso il primo) vi sono 2 lag fuori dalla soglia, prima del primo che vi è all'interno.

Inoltre, per la stagionalità si è preso $P = 1$ e $Q = 2$, in quanto il 12°lag è fuori dalla soglia sia per l'autocorrelazione che per l'autocorrelazione parziale.

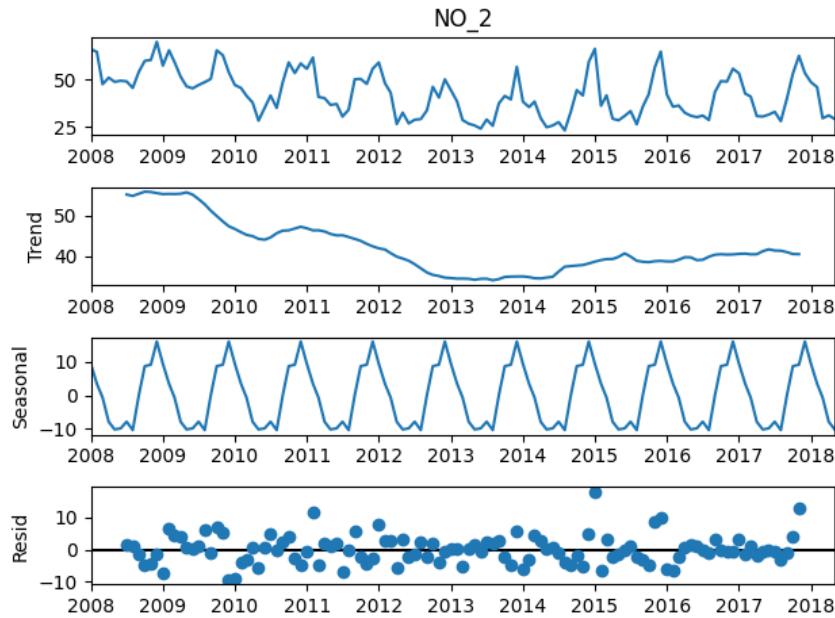


Figura 65 - Decomposizione della serie temporale relativa al No2 mensile

10.9.3 Modello ARIMA

Si è proceduto creando un modello ARIMA, utilizzando i valori, estratti prima, quindi il modello è:

$$ARIMA: (1,0,2)\times(1,0,2,12)$$

Che sono stati scelti a seguito del testing di diversi parametri.

Il modello risultante è il seguente:

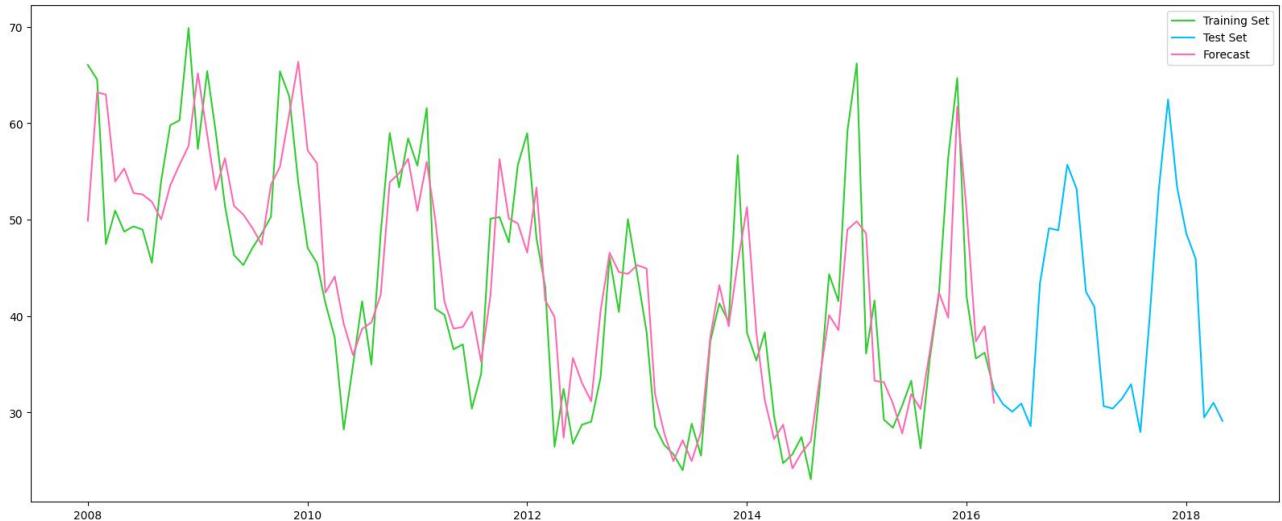


Figura 66 - MODELLO mensile

Anche in questo caso, come nel capitolo precedente, abbiamo diviso il dataset in training set e test set, destinando l'ottanta per cento degli elementi del dataset per il training e la restante parte per il test.

10.9.4 Predizione in-sample

Siamo poi passati una volta estratto il modello ad effettuare una predizione in-sample, ovvero si vanno a predire valori di cui però si conosce anche il valore vero, per poter calcolare le metriche e capire quindi le prestazioni del modello.

Sostanzialmente abbiamo testato il modello.



La predizione in-sample restituisce il seguente risultato:

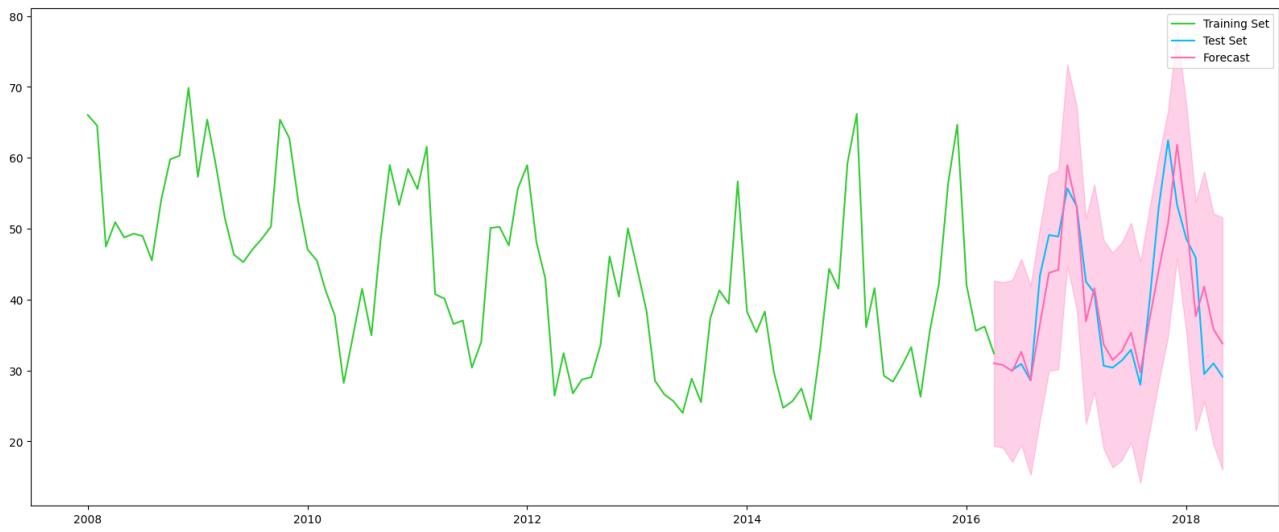


Figura 67 - MODELLO mensile: predizione in-sample

E le metriche risultanti sono le seguenti:

MAPE = 0.10400962051935408

MSE = 30.526704490809703

MAE = 4.274835325602601

R2 = 0.7209980518396807

10.9.5 Predizione out-sample

Infine, abbiamo effettuato la predizione out-sample, per cercare di capire come potrebbe essere l'andamento nei prossimi anni. Il risultato della predizione out-sample è il seguente:

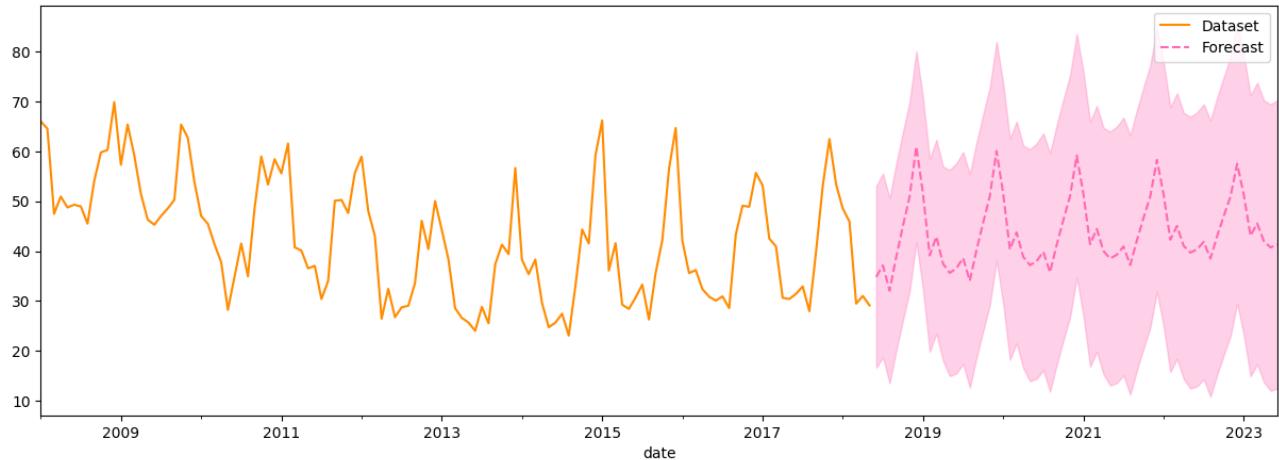


Figura 68 - MODELLO mensile: predizione out-sample



Link al repository GitHub:

<https://github.com/Simone-Scalella/DataScience2Project>