Introduction:

In the age of cloud computing, where data reigns supreme, ensuring data quality has never been more critical. Time series data, with its dynamic nature and diverse applications, demands special attention in cloud-based environments. In this blog post, we explore a Data Quality Management Framework tailored for time series data in the cloud,applied here to the UCI Air Quality dataset.

Motivation:

Our Data Quality Assessment and Improvement Framework for time series data emerged from the need to address data reliability challenges effectively. Time series data underpins crucial decisions in today's data-driven landscape. To ensure its accuracy, consistency, and completeness, we recognized the necessity for a specialized toolset. Our motivation is to provide data professionals with a user-friendly solution that not only detects issues but actively enhances data quality. By simplifying the process, we aim to make high-quality time series data accessible to all, fostering data-driven excellence across industries.

Framework Overview

At its core, the DataQuality framework comprises two essential components: the Checker and the Improver. These components work together to comprehensively assess data quality and enact meaningful improvements.

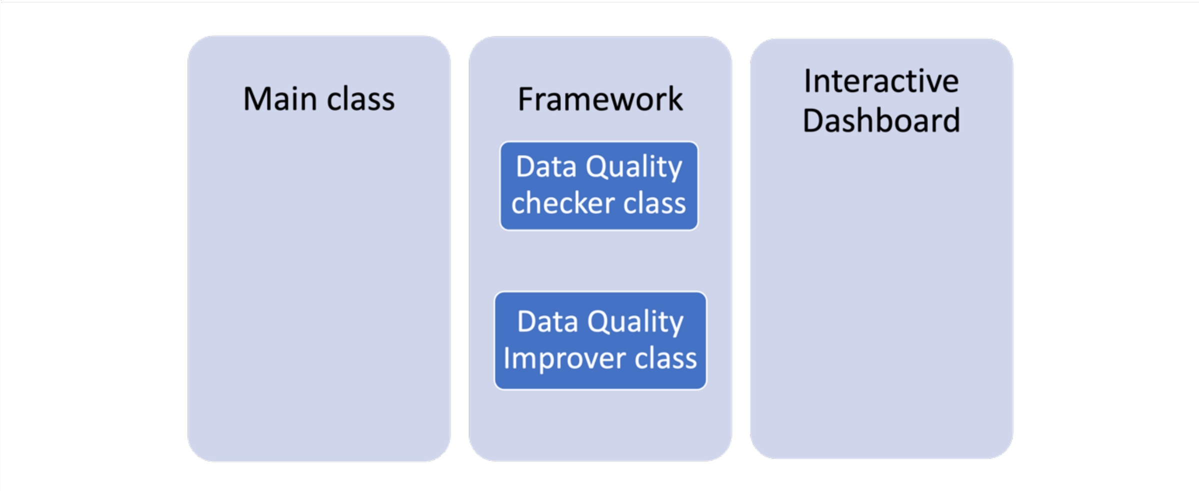


Figure 1 : Framework Overview

• **Checker**: The Checker component comprises a wide range of functions to inspect data for inconsistencies, missing values, outliers, and other quality-related issues. With its ability to analyze large datasets swiftly, the Checker ensures that your data meets the highest standards.

Here's a brief overview of its key features:

1. Completeness Assessment: It helps identify missing values within the dataset, providing insights into the extent of missing data. To do this, the framework calculates the number of missing values for each day, allowing users to see how complete or incomplete the data is on a daily basis. Moreover, it goes beyond just counting missing values; it also identifies periods where data coverage is notably poor. These 'poor coverage' periods are flagged , providing a clear indication of when the data may be less reliable.

2. Skewness Calculation: This class calculates skewness for numeric columns, aiding in understanding data distribution.
3. Bias Detection: It quantifies bias within a target column, ensuring data fairness and accuracy.
4. Stationarity Check: It assesses the stationarity of numeric columns, crucial for selecting appropriate time series analysis methods.
5. Time Shift Analysis: The class calculates time data reigns supreme for columns of interest, revealing temporal relationships between variables.

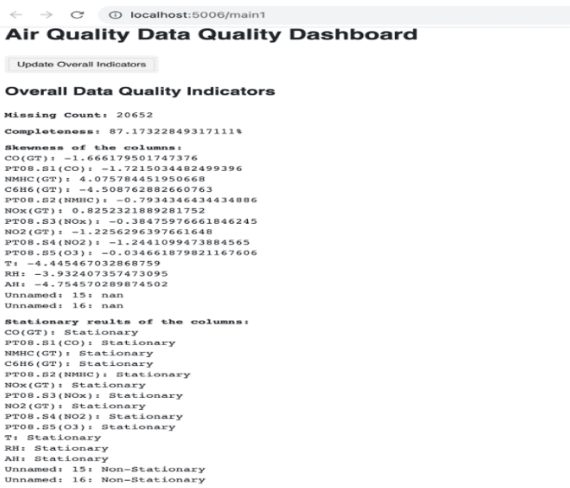
• **Improver:** Where the Checker identifies areas for improvement, the Improver component steps in to elevate your data quality. Unlike conventional solutions that offer only basic fixes, the Improver class provides a set of advanced options. For instance, when addressing missing values, you're not limited to simple imputation methods; you have the flexibility to choose from a range of techniques, including machine learning-based imputation models. This empowers to tailor data quality improvements according to specific needs.

1. Count Missing Values: It allows users to determine the number of missing values in the dataset. It's essential for understanding data completeness.
2. Impute Missing Values: This function offers various methods for imputing missing values in the dataset. It includes techniques like forward/backward filling (impute_forward_back_fill), linear interpolation (impute_linear_interpolation), and seasonal decomposition (impute_seasonal_decomposition). These methods help in replacing missing data with meaningful estimates.
3. Plot Air Quality Data: The plot_air_quality_data method assists in visualizing air quality data. Users can specify the timestamp and column name to plot. Additionally, it allows overlaying imputed and resampled data for comparison, aiding in data analysis and visualization.
4. Align Frequencies: The align_frequencies method is useful for aligning the frequencies of timestamped data. Users can specify the timestamp column, value column, and target frequency to resample the data. This feature is handy for ensuring data consistency.
5. Handle Duplicates: The handle_duplicates method helps identify and remove duplicate rows in the dataset, ensuring data integrity.
6. Smooth Outliers: The smooth_outliers method identifies outliers in numeric columns and smoothes the data by applying a moving average. This feature aids in handling noisy data.
7. Improve Stationarity: The improve_stationarity method supports making data stationary by differencing non-stationary time series data. This is crucial for time series analysis and modeling.
8. Improve Time Shifts: The improve_time_shifts method analyzes time shifts in a target column, helping users discover temporal relationships between variables. It automatically

corrects time shifts by interpolating data.

These quality scores and insights are presented on a user-friendly dashboard, empowering data professionals to make informed decisions and ensuring data-driven excellence across industries.

Fig 3: The Air Quality Data Quality Dashboard provides two key metrics when selecting specific columns of interest - the overall normalized consistency score and the overall normalized relevancy score.



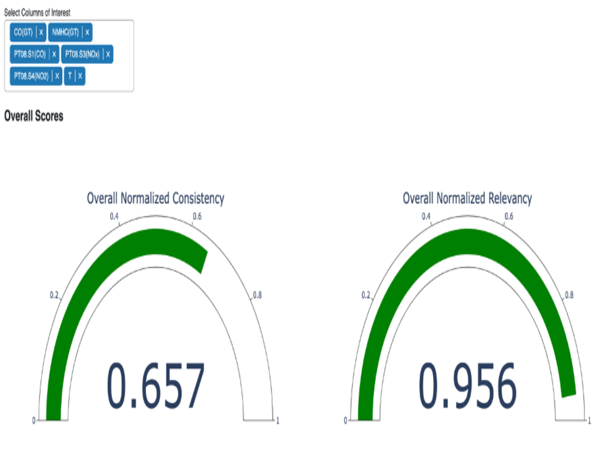


Figure 3 : Overall consistency and Relevancy score

Figure 2 : Dashboard Overview

Fig 2: By clicking the "Update Overall Consistency" button on the Air Quality Data Quality Dashboard, users can access a comprehensive overview of important data quality indicators. This includes information on missing data points, which currently stands at 20,652, indicating a data completeness level of 87.17%. The skewness of individual columns like CO(GT), PT08.S1(CO), and NMHC(GT) is also provided, giving insights into the distribution of the data. Moreover, users can determine the stationarity status of these columns, with most being found to be stationary. This ensures that the data is reliable for further analytical processes. With this complete snapshot, users can confidently evaluate the quality of their time series data and make informed decisions accordingly.

The created framework when put to test on the UCI Air quality dataset yielded the following results. The data had a 87.17% complteness rate before improvement and 99.93% completeness rate after. The missing count of the data identified as 20652 was later filled with imputation methods in the Improver class.Using check_missing_data(), temporal analysis revealed days with low coverage, such as "2004-03-11 were identified." Statistical analysis such as skewness showed that columns CO(GT), PT08.S1(CO), C6H6(GT), PT08.S2(NMHC), PT08.S3(NOx), NO2(GT), PT08.S4(NO2), PT08.S5(O3), T, RH, and AH all exhibit negative skewness which indicates a bias towards higher values while columns NMHC(GT) and NOx(GT) display positive skewness indicating bias towards lower values.While chemical sensor values(columsn like CO(GT),PT08.S1(CO),NMHC(GT) were non-stationary,meteorological characteristics (columns like (Temperature),RH (Relative Humidity),AH (Absolute Humidity)were indicated stationary.

The count of 20652 identified as missing values were imputed using a selection of three imputation methods.These result's are depicted in the figures.This filled in the gaps and enhanced consistency.

Fig 4 is the input for Impute method, where we can see the missing values are filled using forward_backward_fill in Fig 5. The method uses the last known value forward (ffill) and, if necessary, the next known value backward (bfill) to fill in gaps.

Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
10-03-2004	18:00:00	2.6	1360	150	11.9	1046	166	1056	113	1692	1268	13.6	48.9	0.7578
10-03-2004	19:00:00	2	1292	112	9.4	955	103	1174	92	1559	972	13.3	47.7	0.7255
10-03-2004	20:00:00		1402	88	9.0	939	131	1140	114	1555	1074	11.9	54.0	0.7502
10-03-2004	21:00:00	2.2	1376	80	9.2	948		1092	122	1584	1203	11.0	60.0	0.7867
10-03-2004	22:00:00	1.6	1272	51	6.5	836	131	1205	116		1110	11.2	59.6	0.7888
10-03-2004	23:00:00	1.2	1197	38	4.7	750	89	1337	96	1393	949	11.2	59.2	0.7848
11-03-2004	00:00:00	1.2	1185	31	3.6	690	62		77	1333		11.3	56.8	0.7603
11-03-2004	01:00:00	1	1136	31	3.3	672	62	1453	76	1333	730	10.7	60.0	0.7702
11-03-2004	02:00:00	0.9		24		609	45	1579	60	1276	620	10.7	59.7	0.7648
11-03-2004	03:00:00	0.6	1010	19	1.7	561	-200	1705	-200	1235	501	10.3	60.2	0.7517
11-03-2004	04:00:00	-200	1011	14	1.3	527	21	1818	34	1197	445	10.1	60.5	0.7465
11-03-2004	05:00:00	0.7	1066	8	1.1	512	16	1918	28	1182	422	11.0	56.2	0.7366
11-03-2004	06:00:00	0.7	1052	16	1.6	553	34	1738	48		472	10.5	58.1	0.7353
11-03-2004	07:00:00	1.1	1144	29	3.2	667		1490	82	1339	730	10.2	59.6	0.7417
11-03-2004	08:00:00	2	1333		8.0	900	174	1136		1517	1102	10.8	57.4	0.7408
11-03-2004	09:00:00	2.2	1351	87	9.5	960	129	1079	101	1583	1028	10.5	60.6	0.7691
11-03-2004	10:00:00	1.7	1233	77	6.3		112	1218	98	1446	860	10.8	58.4	0.7552
11-03-2004	11:00:00	1.5	1179	43	5.0	762	95	1328	92	1362	671	10.5	57.9	0.7352
11-03-2004	12:00:00	1.6	1236	61	5.2	774	104	1301	95	1401	664	9.5	66.8	0.7951
11-03-2004	13:00:00	1.9	1286	63	7.3	869	146	1162	112	1537	799	8.3	76.4	0.8393
11-03-2004	14:00:00	2.9	1371	164	11.5	1034		983	128	1730	1037	8.0	81.1	0.8736
11-03-2004	15:00:00	2.2	1310	79	8.8	933	184	1082	126	1647	946	8.3	79.8	0.8778
11-03-2004	16:00:00	2.2	1292	95	8.3	912	193	1103		1591	957	9.7	71.2	0.8569
11-03-2004	17:00:00	2.9	1383	150	11.2	1020	243	1008	135	1719	1104	9.8	67.6	0.8185

Figure 4: Sample input for impute methods

Date	Time	CO(GT)	PT08.S1(CO	NMHC(GT)	C6H6(GT)	08.S2(NMH	NOx(GT)	T08.S3(NO	NO2(GT)	T08.S4(NO	PT08.S5(O3	T
2004-03-10 00:00:00	18:00:00	2.6	1360	150	11.88172	1045.5	166	1056.25	113	1692	1267.5	13.6
2004-03-10 00:00:00	19:00:00	2	1292.25	112	9.397165	954.75	103	1173.75	92	1558.75	972.25	13.3
2004-03-10 00:00:00	20:00:00	2	1402	88	8.997817	939.25	131	1140	114	1554.5	1074	11.9
2004-03-10 00:00:00	21:00:00	2.2	1375.5	80	9.228796	948.25	131	1092	122	1583.75	1203.25	11
2004-03-10 00:00:00	22:00:00	1.6	1272.25	51	6.518224	835.5	131	1205	116	1583.75	1110	11.15
2004-03-10 00:00:00	23:00:00	1.2	1197	38	4.741012	750.25	89	1336.5	96	1393	949.25	11.175
2004-03-11 00:00:00	00:00:00	1.2	1185	31	3.624399	689.5	62	1336.5	77	1332.75	s	11.325
2004-03-11 00:00:00	01:00:00	1	1136.25	31	3.326677	672	62	1453.25	76	1332.75	729.5	10.675
2004-03-11 00:00:00	02:00:00	0.9	1136.25	24	3.326677	608.5	45	1579	60	1276	619.5	10.65
2004-03-11 00:00:00	03:00:00	0.6	1009.75	19	1.696658	560.75	-200	1705	-200	1234.75	501.25	10.25
2004-03-11 00:00:00	04:00:00	-200	1011	14	1.29362	526.75	21	1817.5	34	1196.75	445.25	10.075
2004-03-11 00:00:00	05:00:00	0.7	1066	8	1.133431	512	16	1918	28	1182	421.75	11
2004-03-11 00:00:00	06:00:00	0.7	1051.75	16	1.603768	553.25	34	1738.25	48	1182	471.5	10.45
2004-03-11 00:00:00	07:00:00	1.1	1144	29	3.243618	667	34	1489.75	82	1339	729.75	10.2
2004-03-11 00:00:00	08:00:00	2	1333.25	29	8.013773	899.75	174	1136	82	1517	1101.5	10.75
2004-03-11 00:00:00	09:00:00	2.2	1351	87	9.540643	960.25	129	1079	101	1582.75	1027.75	10.5
2004-03-11 00:00:00	10:00:00	1.7	1233.25	77	6.335782	960.25	112	1218	98	1445.75	859.75	10.8
2004-03-11 00:00:00	11:00:00	1.5	1178.75	43	4.971584	762	95	1327.5	92	1361.75	670.5	10.5
2004-03-11 00:00:00	12:00:00	1.6	1236	61	5.216919	774.25	104	1301.25	95	1401.25	664	9.525
2004-03-11 00:00:00	13:00:00	1.9	1285.5	63	7.269933	868.5	146	1162.25	112	1536.75	799	8.3
2004-03-11 00:00:00	14:00:00	2.9	1371	164	11.53901	1033.5	146	983.25	128	1730.25	1036.5	8
2004-03-11 00:00:00	15:00:00	2.2	1310	79	8.826223	932.5	184	1081.75	126	1646.5	946.25	8.325
2004-03-11 00:00:00	16:00:00	2.2	1291.75	95	8.301413	911.5	193	1102.5	126	1590.75	956.75	9.7
2004-03-11 00:00:00	17:00:00	2.9	1383	150	11.15158	1019.75	243	1008	135	1718.75	1104	9.775

Figure 5 : Output forward backward fill

Date	Time	CO(GT)	PT08.S1(CO	NMHC(GT)	C6H6(GT)	08.S2(NMH	NOx(GT)	T08.S3(NO	NO2(GT)	T08.S4(NO	PT08.S5(O3	T
2004-03-10 00:00:00	18:00:00	2.6	1360	150	11.88172	1045.5	166	1056.25	113	1692	1267.5	13.6
2004-03-10 00:00:00	19:00:00	2	1292.25	112	9.397165	954.75	103	1173.75	92	1558.75	972.25	13.3
2004-03-10 00:00:00	20:00:00	2.1	1402	88	8.997817	939.25	131	1140	114	1554.5	1074	11.9
2004-03-10 00:00:00	21:00:00	2.2	1375.5	80	9.228796	948.25	131	1092	122	1583.75	1203.25	11
2004-03-10 00:00:00	22:00:00	1.6	1272.25	51	6.518224	835.5	131	1205	116	1488.375	1110	11.15
2004-03-10 00:00:00	23:00:00	1.2	1197	38	4.741012	750.25	89	1336.5	96	1393	949.25	11.175
2004-03-11 00:00:00	00:00:00	1.2	1185	31	3.624399	689.5	62	1394.875	77	1332.75	839.375	11.325
2004-03-11 00:00:00	01:00:00	1	1136.25	31	3.326677	672	62	1453.25	76	1332.75	729.5	10.675
2004-03-11 00:00:00	02:00:00	0.9	1073	24	2.511668	608.5	45	1579	60	1276	619.5	10.65
2004-03-11 00:00:00	03:00:00	0.6	1009.75	19	1.696658	560.75	-200	1705	-200	1234.75	501.25	10.25
2004-03-11 00:00:00	04:00:00	-200	1011	14	1.29362	526.75	21	1817.5	34	1196.75	445.25	10.075
2004-03-11 00:00:00	05:00:00	0.7	1066	8	1.133431	512	16	1918	28	1182	421.75	11
2004-03-11 00:00:00	06:00:00	0.7	1051.75	16	1.603768	553.25	34	1738.25	48	1260.5	471.5	10.45
2004-03-11 00:00:00	07:00:00	1.1	1144	29	3.243618	667	104	1489.75	82	1339	729.75	10.2
2004-03-11 00:00:00	08:00:00	2	1333.25	58	8.013773	899.75	174	1136	91.5	1517	1101.5	10.75
2004-03-11 00:00:00	09:00:00	2.2	1351	87	9.540643	960.25	129	1079	101	1582.75	1027.75	10.5
2004-03-11 00:00:00	10:00:00	1.7	1233.25	77	6.335782	861.125	112	1218	98	1445.75	859.75	10.8
2004-03-11 00:00:00	11:00:00	1.5	1178.75	43	4.971584	762	95	1327.5	92	1361.75	670.5	10.5
2004-03-11 00:00:00	12:00:00	1.6	1236	61	5.216919	774.25	104	1301.25	95	1401.25	664	9.525
2004-03-11 00:00:00	13:00:00	1.9	1285.5	63	7.269933	868.5	146	1162.25	112	1536.75	799	8.3
2004-03-11 00:00:00	14:00:00	2.9	1371	164	11.53901	1033.5	165	983.25	128	1730.25	1036.5	8
2004-03-11 00:00:00	15:00:00	2.2	1310	79	8.826223	932.5	184	1081.75	126	1646.5	946.25	8.325
2004-03-11 00:00:00	16:00:00	2.2	1291.75	95	8.301413	911.5	193	1102.5	130.5	1590.75	956.75	9.7
2004-03-11 00:00:00	17:00:00	2.9	1383	150	11.15158	1019.75	243	1008	135	1718.75	1104	9.775

Figure 6 : Output linear interpolation

The impute_linear_interpolationfunction will use linear interpolation as an imputation method for missing columns. Initially, it copies the records from the selected column and then applies linear interpolation, which estimates missing values through creating a linear relationship between the points. Then fills in the missing values by plotting the missing data over the line.Fig 6 shows the results of linear interpolation. The impute_seasonal_decomposition function first preprocess the the dataset, converting ‘Date’ and ‘Time’ strings to datetime objects and merging them into one ‘Datetime’ column. Then it performs seasonal decomposition on the selected column. Seasonal decomposition is a technique for isolating seasonal patterns from the data. The missing values are then imputed with the seasonal component, aligning with the seasonality. Figure 7 shows the results of seasonal decomposition.

Date	Time	CO(GT)	T08.S1(CO	NMHC(GT)	C6H6(GT)	08.S2(NMH	NOx(GT)	T08.S3(NO	NO2(GT)	T08.S4(NO	T08.S5(O3	T	RH	AH
2004-03-10	18:00:00	7.78	398.16	202.36	13.09	463.74	146.15	-473.46	57.17	584.56	549.73	2.07	48.88	0.76
2004-03-10	19:00:00	9.87	596.62	356.26	19.62	631.3	246.13	-566.45	77.23	832.75	844.53	1.53	47.7	0.73
2004-03-10	20:00:00	9.02	456.95	292.57	16.05	539.24	209.8	-518.82	68.55	683.07	787.67	1.68	53.98	0.75
2004-03-10	21:00:00	10.98	120.38	90.94	4.35	196.31	102.88	-292.84	44.59	180.43	400.78	1.25	60	0.79
2004-03-10	22:00:00	12.76	-234.34	-41.81	-4.15	-142.9	-37.83	88.03	-0.64	-209.01	-82.5	0.28	59.58	0.79
2004-03-10	23:00:00	12.21	-290.28	-82.1	-6.51	-271.16	-88.91	341.13	-34.2	-299.73	-369.94	0.22	59.18	0.78
2004-03-11	00:00:00	12.87	-123.91	-56.77	-3.54	-105.55	-17.24	43.15	8.36	-184.74	-122.67	0.1	56.77	0.76
2004-03-11	01:00:00	13.05	-159.08	-61.47	-3.21	-89.03	-7.35	45.69	20.9	-185.48	-116.65	-0.78	60	0.77
2004-03-11	02:00:00	12.54	-212.62	-76.18	-5.55	-204.02	-58.99	197.88	1.92	-259.94	-260.99	-1.71	59.67	0.76
2004-03-11	03:00:00	11.93	-310.76	-97.49	-7.95	-352.64	-341.47	486.68	-288.97	-376.93	-445.01	-2.19	60.2	0.75
2004-03-11	04:00:00	-188.89	-369.76	-112.16	-8.96	-428.46	-120.28	705.61	-56.85	-433.51	-580.27	-3.35	60.47	0.75
2004-03-11	05:00:00	11.7	-351.6	-116.27	-9.11	-436.76	-109.03	710.5	-44.39	-411.63	-571.22	-3.45	56.18	0.74
2004-03-11	06:00:00	8.09	-215.11	-77.44	-6.3	-304.18	-73.93	457.23	-27.24	-299.3	-416.31	-1.33	58.12	0.74
2004-03-11	07:00:00	4.5	-84.55	-48.06	-3.23	-145.09	-76.55	180.64	-5.93	-128.11	-116.1	0.46	59.6	0.74
2004-03-11	08:00:00	5.21	89.67	-61.85	0.85	66.87	53.03	-155.01	-8.66	20.5	228.32	1.13	57.43	0.74
2004-03-11	09:00:00	5.29	103.77	-12.81	1.99	115.02	1.17	-200.93	8.98	70.43	140.8	0.97	60.6	0.77
2004-03-11	10:00:00	4.76	-6.26	-25.45	-1.25	114.17	-17.33	-61.63	6.34	-63.89	-22.88	1.37	58.35	0.76
2004-03-11	11:00:00	4.56	-48.43	-59.41	-2.53	-79.75	-32.68	40.62	1.88	-139.73	-197.83	1.19	57.93	0.74
2004-03-11	12:00:00	4.66	16.93	-41.66	-2.29	-67.38	-24.05	11.35	5.36	-98.17	-193.26	0.34	66.77	0.8
2004-03-11	13:00:00	4.93	70.54	-40.62	-0.35	21.19	15.22	-121.63	21.23	35.5	-57.6	-0.76	76.43	0.84
2004-03-11	14:00:00	5.9	161.06	59.59	3.84	180.64	12.97	-292.52	35.88	227.48	176.28	-0.93	81.15	0.87
2004-03-11	15:00:00	5.19	105.67	-25.79	1.11	77.08	50.2	-189.68	33.23	144.15	84.3	-0.46	79.8	0.88
2004-03-11	16:00:00	5.18	93.69	-9.74	0.59	56.23	59.2	-169.98	33.28	90.71	95.86	1.07	71.15	0.86
2004-03-11	17:00:00	5.88	193.25	45.36	3.44	165.14	108.9	-265.56	41.98	220.59	244.96	1.32	67.62	0.82

Figure 7: Output seasonal decomposition

An **hourly frequency alignment** of CO(GT) allowed for a consistent temporal analysis.The **align_frequencies** method as shown in Fig 8, aligns values with specified target frequencies and making it easier to perform comparisons over a period of time interval. Noise in the measurements was reduced using outlier smoothing.**Smooth Outler method** selected the target columns and applied a moving average with a specified window size, designed to mitigate sudden fluctuations in the data.

Date	Time	CO(GT)	T08.S1(CO	NMHC(GT)	C6H6(GT)	08.S2(NMH	NOx(GT)	T08.S3(NO	NO2(GT)	T08.S4(NO	T08.S5(O3	T
03/10/2004	18:00:00	2.6	1360	150	11.88172	1045.5	166	1056.25	113	1692	1267.5	13.6
03/10/2004	19:00:00	2	1292.25	112	9.397165	954.75	103	1173.75	92	1558.75	972.25	13.3
03/10/2004	20:00:00	2.2	1402	88	8.997817	939.25	131	1140	114	1554.5	1074	11.9
03/10/2004	21:00:00	2.2	1375.5	80	9.228796	948.25	131	1092	122	1583.75	1203.25	11
03/10/2004	22:00:00	1.6	1272.25	51	6.518224	835.5	131	1205	116	1393	1110	11.15
03/10/2004	23:00:00	1.2	1197	38	4.741012	750.25	89	1336.5	96	1393	949.25	11.175
03/11/2004	00:00:00	1.2	1185	31	3.624399	689.5	62	1453.25	77	1332.75	ss	11.325
03/11/2004	01:00:00	1	1136.25	31	3.326677	672	62	1453.25	76	1332.75	729.5	10.675
03/11/2004	02:00:00	0.9	1009.75	24	1.696658	608.5	45	1579	60	1276	619.5	10.65
03/11/2004	03:00:00	0.6	1009.75	19	1.696658	560.75	-200	1705	-200	1234.75	501.25	10.25
03/11/2004	04:00:00	-200	1011	14	1.29362	526.75	21	1817.5	34	1196.75	445.25	10.075
03/11/2004	05:00:00	0.7	1066	8	1.133431	512	16	1918	28	1182	421.75	11
03/11/2004	06:00:00	0.7	1051.75	16	1.603768	553.25	34	1738.25	48	1339	471.5	10.45
03/11/2004	07:00:00	1.1	1144	29	3.243618	667	174	1489.75	82	1339	729.75	10.2
03/11/2004	08:00:00	2	1333.25	87	8.013773	899.75	174	1136	101	1517	1101.5	10.75
03/11/2004	09:00:00	2.2	1351	87	9.540643	960.25	129	1079	101	1582.75	1027.75	10.5
03/11/2004	10:00:00	1.7	1233.25	77	6.335782	762	112	1218	98	1445.75	859.75	10.8
03/11/2004	11:00:00	1.5	1178.75	43	4.971584	762	95	1327.5	92	1361.75	670.5	10.5
03/11/2004	12:00:00	1.6	1236	61	5.216919	774.25	104	1301.25	95	1401.25	664	9.525
03/11/2004	13:00:00	1.9	1285.5	63	7.269933	868.5	146	1162.25	112	1536.75	799	8.3
03/11/2004	14:00:00	2.9	1371	164	11.53901	1033.5	184	983.25	128	1730.25	1036.5	8
03/11/2004	15:00:00	2.2	1310	79	8.826223	932.5	184	1081.75	126	1646.5	946.25	8.325
03/11/2004	16:00:00	2.2	1291.75	95	8.301413	911.5	193	1102.5	135	1590.75	956.75	9.7
03/11/2004	17:00:00	2.9	1383	150	11.15158	1019.75	243	1008	135	1718.75	1104	9.775

Figure 8 : Output align frequencies

Additionaly, it also identified outliers by calculating z-scores, considering any data points exceeding a predetermined threshold as outliers. To facilitate further analysis, the method **introduced two new columns**: ’is_outlier,’ which flags data points as outliers (True), and ’smoothened_rows,’ which records the names of columns with outliers for each row. There

duplicates discovered were dropped. By **differencing techniques**, columns like CO(GT),PT08.S1(CO)stationarity was also enhanced. The framework's ability to detect and rectify data issues makes it a valuable asset for any organization relying on time series data for decision-making.

Future Work

The framework has a wide range of potential applications in the future. Extending the compatibility of data sources is one important option. While the framework currently supports data upload from zip files, it could be expanded to easily interface with various database systems, enhancing its use. More time series-specific validation techniques, such as change point identification, might be added to the existing toolkit of techniques. Robustness would be increased using various time indexes like DateTimeIndex with a range of frequencies. Based on evaluation, automated recommendations might be included. The framework's usefulness would be further demonstrated by testing it on additional varied datasets. These provide intriguing directions for further research.

Our experience with Databrick

Databricks played a crucial role in effectively conducting our research project. Our team collaborated through notebooks for interactive code development and data analysis, allowing us to efficiently track work progress, test multiple approaches, and share expertise in data quality improvement. Databricks Repos kept our code synced to the Git repository and facilitated change tracking and code versioning. With Databricks, we could effectively analyze large datasets without resource constraints.

Acknowledgement

The blog post was authored by Kavya Sasikumar with contributions from Aravind RK and Deepthy Paulose.

We would like to express our sincere gratitude to Prof. Dr. Frank Hopfgartner for his invaluable guidance, support, and mentorship throughout the course of this project. His expertise and insights were instrumental in shaping this report.

We would also like to extend our heartfelt thanks to Evgeny Chernyi from Databricks for his collaboration and assistance in providing valuable data and insights. His contributions significantly enriched the quality of this research.

Lastly, we would like to acknowledge the Universität Koblenz for providing the resources and the environment necessary for conducting this research.

Resources

[Github link for the source code]

[UCI Air quality dataset]