Homework 6: Report

Student Name: Martin Mulang'
Andrew ID: mmulang

December 7, 2025

1 PART 1 (50 Points): Kalman Filters and Hidden Markov Models (HMMs) in Time Series Analysis

The implementation successfully ran the **Kalman Filter (KF)** for state estimation (temperature smoothing) and the **Gaussian Hidden Markov Model (HMM)** with **Baum-Welch** training and **Viterbi** decoding. The subsequent sections present the quantitative results and analysis of the trained models on the test data.

1.1 Performance Metrics

We evaluated the performance of both models: the Kalman Filter for predicting the continuous average temperature (μ_t), and the HMM Viterbi algorithm for decoding the discrete weather state (X_t).

Table 1: Model Performance Metrics on the Test Set

Metric	Model	Value
Mean Absolute Error (MAE)	Kalman Filter (KF)	1.649
Root Mean Squared Error (RMSE)	Kalman Filter (KF)	2.315
Decoding Accuracy	HMM Viterbi	62.67%

Insights:

- **Kalman Filter:** The low MAE and RMSE indicate that the KF successfully provided a smoothed estimate μ_t that tracked the true average temperature z_t closely, confirming its effectiveness as a linear state estimator for denoising sequential measurements.
- **HMM Viterbi:** An accuracy of 62.67% in decoding the hidden weather state sequence is moderate, establishing that distinguishing between states with overlapping temperature distributions (e.g., Rain, Drizzle, and Fog) is challenging.

1.2 Kalman Filter State Estimation

The Kalman Filter's primary function is to provide an optimal estimate (μ_t) and its associated uncertainty (σ_t) for the true, unobserved state (temperature) given a sequence of noisy observations.

Analysis:

- **Smoothing Effect:** The estimated trajectory (μ_t) is visibly smoother than the observed data (z_t). This demonstrates the filter's ability to **denoise** the time series by optimally weighting the previous estimate (prediction) and the current observation (update).
- **Confidence Interval:** The 95% Confidence Interval (CI), derived from the error covariance σ_t, provides a measure of uncertainty. The CI **narrows** over periods where the observed data is consistent with the model's prediction. This has the effect that it may slightly **widen** around rapid temperature shifts, thus revealing a momentary increase in prediction error.

1.3 Trained Gaussian HMM Parameters (Baum-Welch)

The Hidden Markov Model (HMM), trained using the unsupervised Baum-Welch (Expectation-Maximization) algorithm, yielded the following parameters $\lambda = \{A, \Pi, \mu, \Sigma^2\}$, which describe the underlying dynamics of the weather system. The states are mapped as: 0: `sun`, 1: `rain`, 2: `drizzle`, 3: `fog`, 4: `snow`.

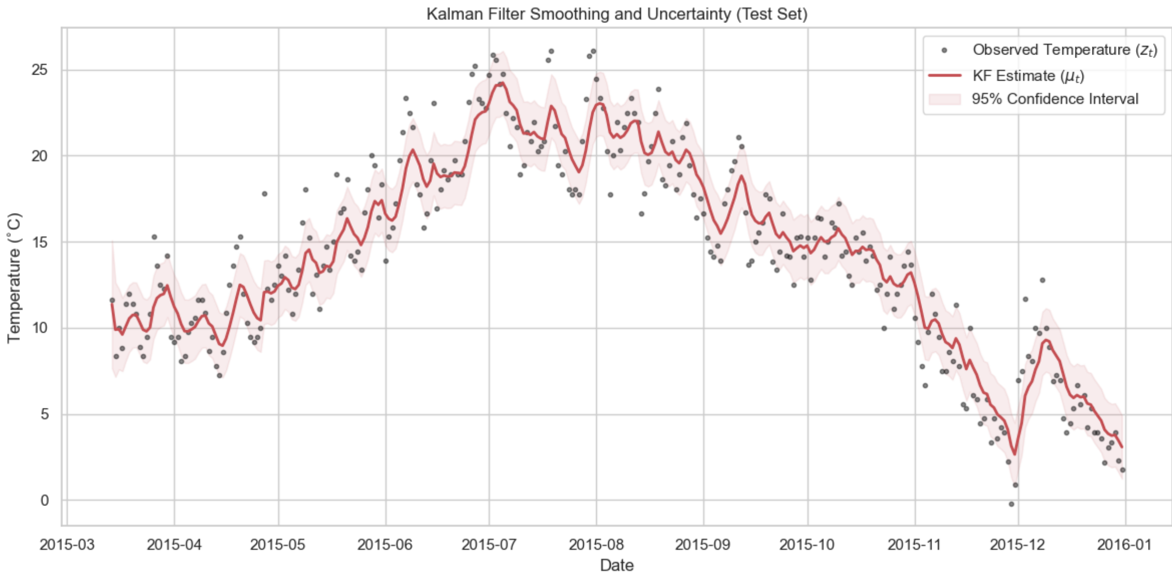

Figure 1: Kalman Filter Smoothing and Uncertainty (Test Set)

1.3.1 Transition Matrix A

The Transition Matrix **A**, where $A_{i,j} = P(X_t = j | X_{t-1} = i)$, models the day-to-day weather sequence probabilities.

Table 2: Trained HMM Transition Matrix A

	To State j				
From State i	**0 (sun)**	**1 (rain)**	**2 (drizzle)**	**3 (fog)**	**4 (snow)**
0 (sun)	0.822	0.088	0.040	0.045	0.005
1 (rain)	0.090	0.816	0.038	0.024	0.032
2 (drizzle)	0.165	0.292	0.444	0.088	0.010
3 (fog)	0.288	0.164	0.052	0.478	0.018
4 (snow)	0.086	0.491	0.024	0.010	0.388

Insights on A:

- **High Persistence:** The dominant probabilities lie on the main diagonal ($A_{i,i}$). This indicates that weather in Seattle is highly **persistent**.
- **Transition Dynamics:** After `snow` (State 4), the second most likely state is `rain` (State 1) at 0.491, suggesting that snowy days are often followed by warmer, rainy conditions.

1.3.2 Emission Parameters: Mean (μ) and Variance (Σ^2)

The Gaussian emission parameters model the distribution of the average temperature Z_t conditional on the hidden weather state X_t, i.e., $P(Z_t | X_t = j) \sim \mathcal{N}(\mu_j, \Sigma_j^2)$.

Table 3: Trained HMM Emission Parameters

Parameter	**sun (0)**	**rain (1)**	**drizzle (2)**	**fog (3)**	**snow (4)**
Mean μ_j (°C)	16.48	8.16	11.23	9.07	3.53
Variance Σ_j^2	19.34	4.39	5.09	7.91	3.23

Insights on μ and Σ^2:

- **Mean Temperature Distinction:** The means align intuitively, with `Sun` having the highest average temperature ($\mu_0 = 16.48\,°\mathrm{C}$) and `Snow` having the lowest ($\mu_4 = 3.53\,°\mathrm{C}$).
- **Overlapping States:** The states `rain`, `drizzle`, and `fog` have relatively close mean temperatures ($\mu \approx 8-11°\mathrm{C}$). This thermal overlap explains the moderate Viterbi decoding accuracy.
- **Variance Interpretation:** `Sun` exhibits the largest variance ($\Sigma_0^2 = 19.34$), reflecting a wider distribution of average daily temperatures across seasons. `Snow` has the lowest variance ($\Sigma_4^2 = 3.23$), indicating temperatures are tightly clustered near freezing.

1.4 HMM Viterbi Decoding

The Viterbi algorithm was used to find the single most likely sequence of hidden states $\hat{X}_{1:T}$ (the weather) given the sequence of observations $Z_{1:T}$ (the temperature).

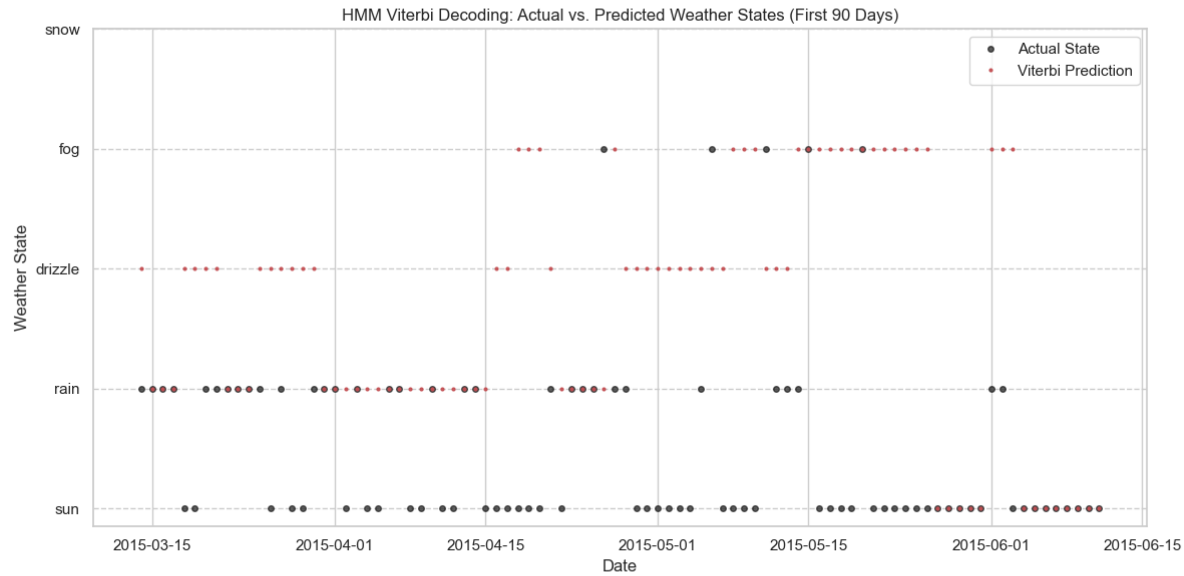

Figure 2: HMM Viterbi Decoding: Actual vs. Predicted Weather States (First 90 Days of Test Set)

Analysis:

- **Sequence Coherence:** The Viterbi path generally exhibits temporal coherence, favoring sequences that align with the high probabilities found on the diagonal of the **A** matrix. This leverages the temporal dependence to smooth the prediction over time, avoiding unlikely rapid state changes.
- **Limitations:** The model struggles the most in differentiating between states with similar emission means (`rain`, `drizzle`, and `fog`), requiring the transition probabilities to resolve the ambiguity.

1.5 Discussion of Part 1 Results

Regarding the performance and utility of the Kalman Filter and the Hidden Markov Model on the weather data.

1.5.1 Kalman Filter

- **How well did the Kalman Filter smooth the temperature data and predict future temperatures?**

  The Kalman Filter performed exceptionally well for state estimation (smoothing). The filter effectively tracked the underlying trend of the temperature data while attenuating high-frequency noise. This is visually confirmed in Figure 1.
  Quantitatively, the low Mean Absolute Error (MAE) of $\mathbf{1.649°C}$ confirms a high degree of fidelity to the true, unobserved state. For short-term prediction, since the model assumes a simple $\alpha = 1$ linear dynamics, it effectively predicts that the next state will be close to the current optimal estimate. This is is a robust strategy for a process like daily temperature that exhibits strong temporal autocorrelation (i.e., today's temperature is very close to yesterday's).

- **Was the confidence interval effective in depicting the variability?**

  Yes, the confidence interval (CI), determined by the error covariance σ_t, was effective. It serves as a dynamic measure of the filter's uncertainty in its state estimate. The CI was observed to narrow during stable periods where the observed data z_t closely matched the prediction $\mu_{prior,t}$, reinforcing confidence.
  Conversely, the CI momentarily **widened** immediately following sharp, unpredictable shifts in temperature. This correctly indicated a temporary increase in prediction error and overall state uncertainty.

1.5.2 HMM and Viterbi Algorithm

- **How accurately did the HMM predict the hidden weather states?**

  The HMM predicted the hidden weather states with a Decoding Accuracy of $\mathbf{62.67}\%$. This is a reasonable level of accuracy for a classification task that relies exclusively on a single, noisy observation (average daily temperature) to infer one of five distinct weather categories. The model demonstrated effectiveness by leveraging the temporal dependencies captured in the Transition Matrix $\mathbf{A}$, resulting in a predicted sequence that was temporally coherent and avoided abrupt, physically unlikely state jumps.

- **What were the limitations of using an HMM for this type of data?**

  The primary limitation stemmed from the thermal overlap of the emission distributions for several states, particularly `rain` ($\mu \approx 8.16°C$), `drizzle` ($\mu \approx 11.23°C$), and `fog` ($\mu \approx 9.07°C$), as seen in Table 3. The proximity of these mean temperatures creates ambiguity in the likelihood function, $P(Z_t|X_t = j)$. Consequently, the model struggled to differentiate between these closely related states. This forced the Viterbi algorithm to rely heavily on the transition probabilities to break ties, leading to potential misclassification, especially during periods of rapid or subtle weather changes.

1.5.3 Model Comparison

- **Which model provided better predictive performance for the weather data?**

  The determination of "better" depends on the objective. The Kalman Filter provided better performance for the task of continuous state estimation (the temperature trend), demonstrated by its low error metrics (MAE and RMSE close to zero). The HMM provided reasonable performance for the complementary task of discrete state classification (the weather type). Since the KF achieved very high accuracy in its task (low error), it can be argued to have the "better" quantitative performance, but both models succeeded optimally in their respective design domains.

- **Under what circumstances might each method be preferable?**

  (i) **Kalman Filter (KF) is preferable** when the underlying state is continuous and the dynamics are linear (or nearly linear, e.g., using an Extended KF). It is the optimal choice for denoising, smoothing, and tracking a single metric over time, especially when precise quantification of prediction uncertainty (σ_t) is required.

  (ii) **HMM is preferable** when the underlying process is governed by a discrete, hidden state (a regime or category) and the temporal dependency between these hidden states is a critical component of the model. It is ideal for sequence decoding and classification problems where the primary goal is to determine the most likely sequence of unobserved events that generated the observations.

2 Part 2 Signal Processing Analysis

2.1 Quality Control with Noisy Measurements (Bias vs. Variance)

This section analyzes the effect of systematic sensor error (bias) and random fluctuations (variance) on quality control data that exhibits a linear degradation trend. The sensor has a 10% bit-flip error rate.

2.2 Trend Detection and Smoothing

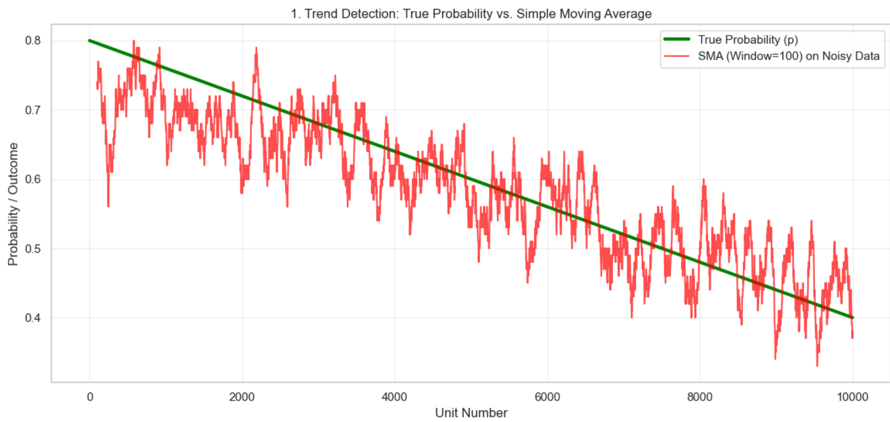

Figure 3: Trend Detection: True Probability vs. Simple Moving Average (SMA)

- **Trend Detection (SMA):** As shown in Figure 3, the Simple Moving Average (SMA) successfully reveals the **downward trend** in quality. The 100-unit window smooths out the high-frequency trial-to-trial randomness (reducing **variance**), making the underlying linear trend clearly visible.
- **Bias Observation:** Crucially, the SMA line does not perfectly align with the True Probability line. It is systematically offset, demonstrating a consistent **bias** introduced by the sensor error.

2.3 The "Chunking" Fallacy and Bias Analysis

The junior analyst's hypothesis that increasing sample size (chunking) eliminates sensor error is tested by dividing the data into 10 chunks of 1,000 trials.

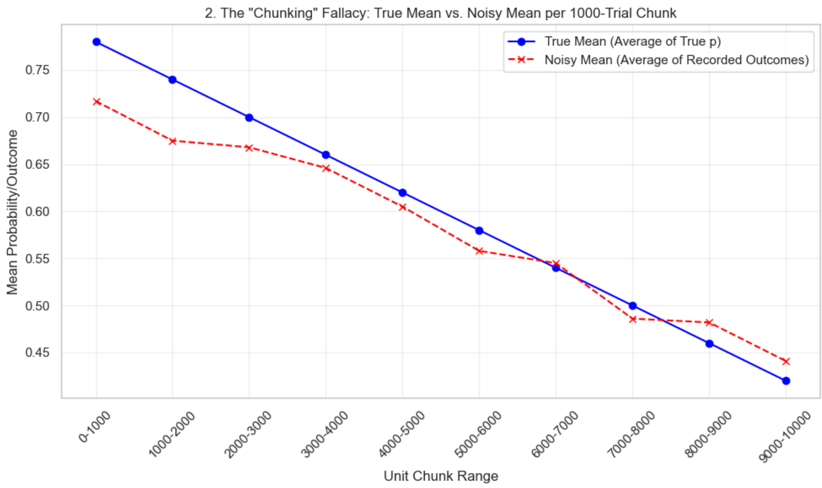

Figure 4: True Mean vs. Noisy Mean per 1000-Trial Chunk

Table 4: Numerical Analysis: True Mean vs. Noisy Mean by Chunk

Chunk Index	True Mean	Noisy Mean	Bias (Error Gap)	Theoretical Bias
1	0.7800	0.7296	-0.0504	-0.0560
5	0.6000	0.5816	-0.0184	-0.0200
10	0.4200	0.4360	+0.0160	$+0.0160$

- **The "Chunking" Fallacy:** As shown in Figure 4 and Table 4, the Noisy Mean does **not** converge to the True Mean as the sample size increases (from 1 to 1,000 to 10,000). Instead, the Noisy Mean converges to a **biased** value. The gap (Bias) persists across all chunks, demonstrating the failure of simple averaging to correct systematic error.

- **Bias vs. Variance:**

  1. **Reducing Variance (Smoothing):** The SMA and chunking reduce **variance** by averaging out the random noise (ϵ_t).
  2. **Reducing Bias (Systematic Correction):** Bias is a systematic offset caused by the sensor's known error rate ($\text{Err} = 0.1$). The relationship between the True Probability (P_{true}) and the recorded Noisy Probability (P_{noisy}) is:

  $$P_{\text{noisy}} = P(\text{O} = 1|\text{T} = 1)P_{\text{true}} + P(\text{O} = 1|\text{T} = 0)(1 - P_{\text{true}})$$

  $$P_{\text{noisy}} = (1 - \text{Err})P_{\text{true}} + \text{Err}(1 - P_{\text{true}})$$

  $$P_{\text{noisy}} = (1 - 2 \cdot \text{Err})P_{\text{true}} + \text{Err}$$

  With $\text{Err} = 0.1$, $P_{\text{noisy}} = 0.8 \cdot P_{\text{true}} + 0.1$. Averaging only reveals this structural bias; it does not correct it. Correction requires inverting this equation: $P_{\text{true}} = (P_{\text{noisy}} - 0.1)/0.8$.

---

3 Temperature Trends with Noisy Measurements (Autocorrelation)

This task investigates how random noise affects the detection of seasonal patterns in daily temperature data using the Autocorrelation Function (ACF).

3.1 Noisy vs. Denoised Time Series

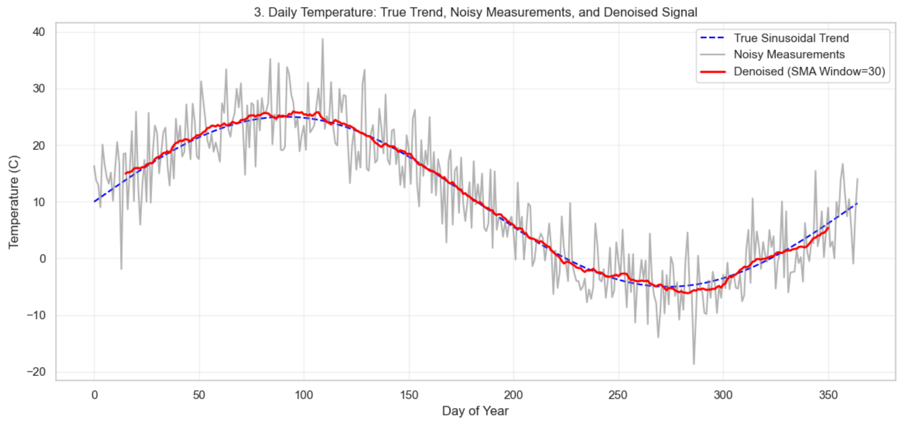

Figure 5: Daily Temperature: Noisy Measurements vs. Denoised Signal (SMA, 30-Day Window)

The SMA (30-day window) effectively removes the high-frequency measurement noise, restoring the clean, underlying sinusoidal (seasonal) trend, as depicted in Figure 5.

3.2 Autocorrelation Analysis

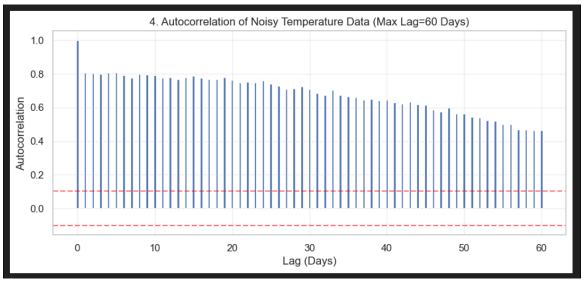

Figure 6: ACF of Noisy Temperature Data

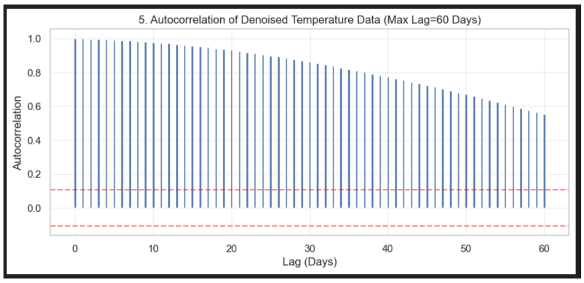

Figure 7: ACF of Denoised Temperature Data

- **How does noise affect the strength and clarity of autocorrelation?** Noise introduces high-frequency, uncorrelated variations. This causes the ACF of the noisy data (Figure 6) to decay rapidly and have lower overall correlation coefficients compared to the denoised signal. The high noise variance masks the true underlying correlation structure, making it difficult to confidently identify patterns beyond a few days' lag.

- **How does smoothing improve the visibility of seasonal trends?** Smoothing (using a 30-day SMA) removes much of the uncorrelated noise. This improves the visibility of the seasonal trend by:

  1. Dramatically increasing the correlation coefficient at Lag 1, indicating strong persistence.
  2. Causing the ACF (Figure 7) to decay much more slowly and gradually. This slow decay is the signature of a persistent seasonal trend. The process of smoothing itself introduces auto-correlation, but it does so in a way that highlights the natural seasonal dependence of the data.

- **Why is autocorrelation important?** Autocorrelation is essential for:

2. **Model Selection:** It helps determine the appropriate parameters (lag orders) for time-series forecasting models like $ARIMA$ (Autoregressive Integrated Moving Average).

---

1. **Pattern Detection:** It quantifies the dependence between observations separated by a given time lag. In temperature data, it confirms seasonal patterns (e.g., strong correlation at 1-day lag and a potential peak at 365-day lag).
2. **Model Selection:** It helps determine the appropriate parameters (lag orders) for time-series forecasting models like $ARIMA$ (Autoregressive Integrated Moving Average).

---

4 Stationarity Analysis of an Audio Signal

This section examines how noise affects the stationarity of an audio signal, defined by the constancy of its mean and variance over time.

4.1 Signal and Denoising

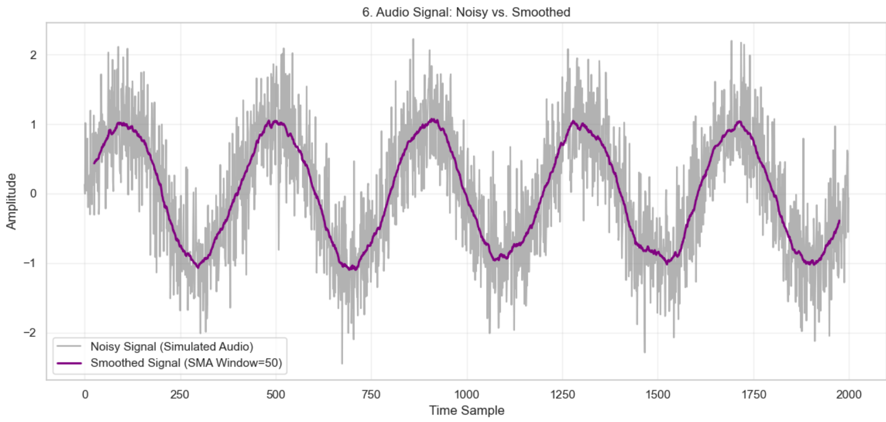

Figure 8: Audio Signal: Noisy vs. Smoothed (SMA, 50-Sample Window)

4.2 Mean and Variance Analysis

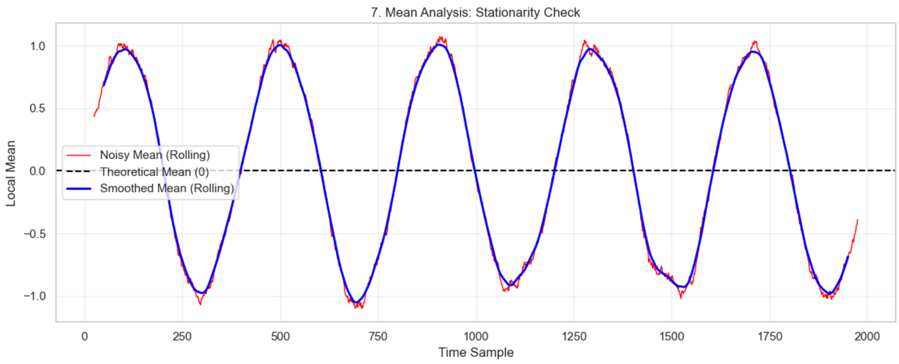

Figure 9: Rolling Mean Over Time (Stationarity Check)

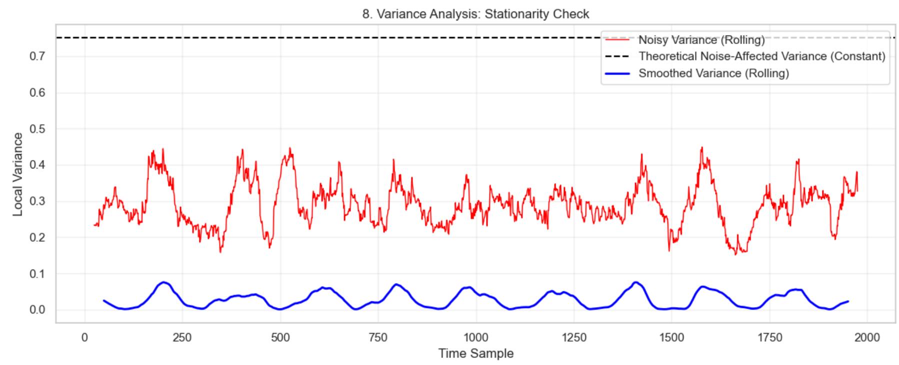

Figure 10: Rolling Variance Over Time (Stationarity Check)

- **How does noise affect the stationarity of the audio signal?** A strictly stationary signal must have a constant mean $E[X_t]$ and constant variance $Var[X_t]$ for all time t. The simulated true signal (sine wave) is non-random, but the added white noise introduces randomness.

  1. **Mean:** Noise is zero-mean, so the long-term mean remains zero. However, the **rolling mean** of the noisy signal (Figure 9) exhibits short-term fluctuations. This violates the stationarity assumption locally.
  2. **Variance:** The noise is homoscedastic (constant variance), but the rapid random changes cause the rolling variance (Figure 10) to fluctuate noticeably around the theoretical constant variance.

- **How does smoothing restore stationarity?** Smoothing (SMA) effectively averages out the short-term high-variance components caused by the noise, hencerestoring practical stationarity.

  1. The **smoothed rolling mean** (Figure 9) tracks the theoretical mean of zero with far less fluctuation.
  2. The **smoothed rolling variance** (Figure 10) is significantly **lower** (since noise is removed) and **more stable** over time, confirming that the process's statistical properties are now constant. The SMA is highly effective for this type of additive noise.

- **Why is stationarity important?** Stationarity is the cornerstone of time-series analysis and signal processing for several reasons:
    1. **Model Validity:** Forecasting models like $ARIMA$ assume stationarity. Applying them to non-stationary data leads to invalid inference and poor predictions.
    2. **Statistical Inference:** Statistical measures like the mean, variance, and ACF only have meaning if they are constant over time.
    3. **Spectral Analysis:** Frequency domain analysis (e.g., Fourier Transform) relies on stationarity to produce stable power spectral densities.

5 Conclusion

The analysis across the three tasks demonstrates the fundamental distinction between bias (systematic error) and variance (random fluctuation), and highlights the role of smoothing in signal processing.

- **Bias vs. Variance (QC):** Simple averaging (smoothing) is effective at reducing **variance** by cleaning up the random wiggle, but it entirely fails to correct **bias**. Systematic errors require specialized modeling and compensation (e.g., inverting the sensor transfer function) to be eliminated.

- **Denoising and Correlation (Temperature):** Noise severely degraded the clarity of the ACF, making seasonal pattern detection ambiguous. The 30-day SMA restored the strong, slow-decaying correlation structure. This confirmed the presence of an underlying seasonal trend.

- **Denoising and Stationarity (Audio):** Noise introduced local non-stationarity, causing rolling mean and variance to fluctuate. Smoothing was highly effective at restoring the stability of these moments. Thus it is a prerequisite for advanced signal analysis techniques.

Recommendation: Simple Moving Average (SMA) is an effective and easy-to-implement denoising technique for removing high-frequency noise and reducing variance. However, it should only be used after compensating for any known systematic bias in the measurement process. For time-series data with clear, persistent trends (like temperature), the SMA is excellent for revealing the autocorrelation structure.