Data Analysis Final Assignment Report

Team: Fantastic Three
Md Ehtashamul Huque & $MdShamsurRahmanShishir$ & $SheikhMd.Nayeem$

1 Contributions

Clearly state each team member's specific contributions. Be concrete.

- Md Ehtashamul Huque:
  - Dataset selection and acquisition
  - Data quality analysis and preprocessing pipeline
  - Missing-value handling and outlier analysis
- Md Shamsur Rahman Shishir:
  - Visualizations and exploratory data analysis (EDA)
  - Probability analysis tasks
  - Law of Large Numbers (LLN) and Central Limit Theorem (CLT) demonstrations
- Sheikh Md. Nayeem:
  - Regression modeling and interpretation
  - Model evaluation and comparison
  - Report writing and figure polishing

2 Dataset Description

- Dataset name and source (Kaggle, Hugging Face, Westermo tests, etc.): IoT Telemetry Sensor Dataset (public IoT sensor dataset)
- Why it is suitable for time-series analysis: The dataset consists of timestamped sensor readings collected continuously from multiple IoT devices, making it suitable for temporal pattern analysis and forecasting.
- Time period covered and sampling frequency: From 2020-07-12 00:01 to 2020-07-20 00:03 (approximately 8 days) with high-frequency asynchronous sampling.
- Key variables analyzed (signals, sensors, physical quantities): Temperature, Humidity, CO, Smoke, LPG, Motion (binary), Light (binary), Device ID
- Size and structure:
  - Number of observations (rows): 405,184
  - Number of features (columns): 9
  - Target variable(s) if any: Temperature

- Missing data summary: Sensor channels contain intermittent missing values; event variables have sparse missing entries.
- Any known limitations or caveats: Short observation window (8 days) and potential sensor noise due to real-world deployment.

3 Task 1. Data Preprocessing and Basic Analysis

3.1 Basic statistical analysis using pandas

- Descriptive stats (mean, std, min, max, quantiles) were computed for all numeric sensor variables.
- Grouped summaries by device and by day revealed differences in temperature and gas sensor distributions.

3.2 Original data quality analysis including visualization

- Missingness patterns (counts, heatmap, timeline gaps) were analyzed using a missingness heatmap.
- Outliers and suspicious values were identified using boxplots and percentile-based thresholds.
- Consistency checks confirmed correct timestamp ordering and absence of duplicates.

3.3 Data preprocessing

- Cleaning steps performed: Timestamp conversion and sorting.
- Missing-value treatment: Sensor variables were imputed using forward fill followed by backward fill; event variables were imputed with zeros.
- Outlier handling: Extreme values beyond reasonable physical limits were removed using percentile thresholds.
- Feature engineering: Standardization applied for PCA and regression models.
- Final dataset shape after preprocessing: 405,184 rows and 9 columns with no missing values.

3.4 Preprocessed vs original data visual analysis

- Before vs after comparison plots showed reduced skewness and improved distribution stability.
- What improved and what trade-offs exist: Minor smoothing occurred, but overall signal structure was preserved.

4 Task 2. Visualization and Exploratory Analysis

4.1 Time series visualizations

- Plot of main variable(s) over time: Temperature and humidity plots revealed clear diurnal cycles.
- Annotations for notable events or pattern shifts (if applicable): Daily cycles were consistent across days.

4.2 Distribution analysis with histograms

- Histograms for key numeric variables showed mild skewness in temperature and heavy tails in gas sensors.
- Notes on skewness, heavy tails, multi-modality: Gas sensors exhibited occasional extreme values.

4.3 Correlation analysis and heatmaps

- Correlation type used (Pearson or Spearman) and why: Pearson correlation was used due to approximately linear relationships.
- Heatmap and top correlated pairs with short interpretation: Strong correlations were observed among CO, Smoke, and LPG sensors.

4.4 Daily pattern analysis

- Aggregation method (hourly means, day-of-week, rolling averages): Hourly mean aggregation.
- Plots showing daily cycles or weekday-weekend differences: Stable daily temperature cycles were observed.
- What patterns are stable vs noisy: Temperature patterns were stable; gas sensors were more variable.

4.5 Summary of observed patterns, similar to True/False questions

- Statement 1 (True): **Temperature follows a daily cycle**. Evidence: Hourly mean plots.
- Statement 2 (True): **Gas sensors are strongly correlated**. Evidence: Pearson correlation heatmap.
- Statement 3 (False): **Motion events occur uniformly over time**. Evidence: Event density varies by hour.

5 Task 3. Probability Analysis

5.1 Threshold-based probability estimation

- Define threshold(s) and justify choice: Upper quantiles of temperature distribution.
- Estimate probabilities of exceeding thresholds: Empirical probabilities computed from data.
- Visual support: Empirical CDF plots.

5.2 Cross tabulation analysis

- Define two categorical variables: Motion and Light events.
- Present contingency table and interpret key cells: Co-occurrence was higher than expected under independence.

5.3 Conditional probability analysis

- Define events A and B: A = Motion detected, B = Light on.
- Compute and interpret $P(A)$, $P(B)$, $P(A \mid B)$, $P(B \mid A)$.
- Include at least one meaningful comparison and conclusion: $P(A \mid B) > P(A)$, indicating dependence; Bayes' rule verified empirically.

5.4 Summary of observations from each probability task

- Key takeaway from threshold probability: High-temperature events occur infrequently.
- Key takeaway from crosstab: Motion and light frequently co-occur.
- Key takeaway from conditional probability: Event variables are dependent.

6 Task 4. Statistical Theory Applications

6.1 Law of Large Numbers (LLN) demonstration

- Variable chosen and why it makes sense: Temperature is stable and continuously measured.
- Experiment: Sample mean plotted as n increases.
- Plot and short interpretation: Sample mean converges to a stable value.

6.2 Central Limit Theorem (CLT) application

- Sampling procedure: Repeated random sampling with increasing sample sizes.
- Show distribution of sample means for increasing n.
- Plot(s): Histograms of sample means approaching normality.

6.3 Result interpretation

- What LLN showed in your data context: Large samples yield reliable averages.
- What CLT showed, and any deviations and why: Sample means approximate a normal distribution with minor deviations due to noise.

7 Task 5. Regression Analysis

7.1 Linear or Polynomial model selection

- Define target y and predictors X: y = Temperature; X = sensor readings, events, device ID.
- Motivation for linear vs polynomial: Linear baseline compared with non-linear models.
- Any train-test split rationale: Time-aware split used to avoid data leakage.

7.2 Model fitting and validation

- Fit procedure and preprocessing: Scaling and feature preparation.
- Validation method: Chronological holdout split.
- Metrics reported (RMSE, MAE, R^2) and why: Standard regression performance metrics.
- Residual analysis: Residual plots examined for systematic error.

7.3 Result interpretation and analysis

- Main effects and practical meaning: Gas sensors and humidity were important predictors.
- Failure cases or where model performs poorly: Linear models struggled with non-linear relationships.

8 Bonus Tasks

- Q-Q plot with explanation: Regression residuals showed mild deviations from normality.

9 Key Findings and Conclusions

- Main findings from preprocessing and EDA: Clear temporal patterns and sensor correlations.
- Main findings from probability tasks: Motion and light events are dependent.
- Main findings from LLN and CLT: Empirical confirmation of statistical theory.
- Main findings from regression: Random Forest outperformed linear models.
- Limitations: Short time span and sensor noise.
- What you would do next if you had more time: Longer datasets and sequence models.

10 Reproducibility Notes

- Exact dataset source link and version or download date: Public IoT telemetry dataset (downloaded 2020).
- Key libraries used and versions: pandas, NumPy, matplotlib, scikit-learn, SciPy.
- How to run the notebook end-to-end: Execute all cells in the provided Jupyter Notebook.