NVH Analysis Task Report

Prepared by: Usama Tahir

July 15, 2025

1 Objective

The purpose of this NVH analysis is to evaluate and quantify the Noise, Vibration, and Harshness levels in electric bikes to ensure compliance with comfort and regulatory requirements. The goal is to identify sources of NVH issues, understand their root causes, and propose effective solutions.

2 Tools Used

- **Data Generation:** Python, Pandas, Numpy
- **Data Extraction and Cleaning:** Power Querey, M Language
- **Calculated Tables, Columns & Measures:** Data Analysis Expressions (DAX)
- **Visualization:** Power BI Desktop
- **Sharing & Collaboration:** Power BI Service, Git Hub

3 Project Architecture Overview

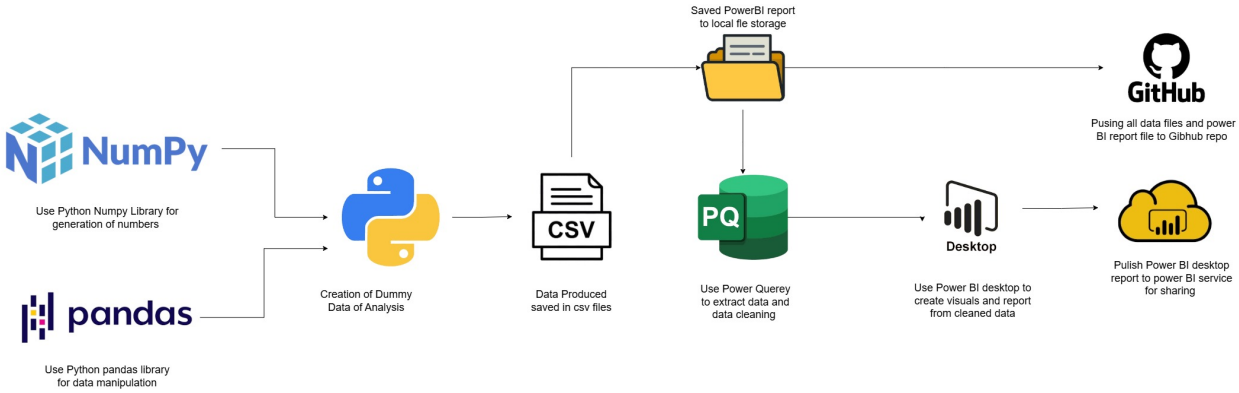

Figure 1: Project Architecture Diagram for NVH Analysis

4 Power BI Features Used

- **Power Query (M Language):** Used for extracting, transforming, and cleaning raw NVH data into a structured format.
- **Data Modeling:** Established relationships between tables and created a logical data model to enable efficient analysis.
- **DAX (Data Analysis Expressions):** Implemented calculated columns, measures, and KPIs (e.g., average vibration levels, peak dB values).
- **Interactive Visualizations:** Created line charts, bar graphs, KPI cards, and matrix views to display trends and patterns in NVH data.
- **Drill-Through and Slicers:** Enabled drill-down and filtering functionality for in-depth exploration of component-level data.
- **Tooltips and Bookmarks:** Enhanced user experience by adding informative tooltips and bookmarks for guided navigation.
- **Scheduled Refresh:** Configured automatic data refresh to ensure dashboards reflect the latest NVH test results.
- **Power BI Service:** Published reports for secure online sharing and collaboration with team members.

5 Explaination

The NVH (Noise, Vibration, and Harshness) analysis for electric bikes was conducted using a synthetic dataset generated in Python. The analysis simulates operational test data across bikes from four different manufacturers: **Bosch, Shimano, Specialized, and Trek**. Each manufacturer's models were tested under various conditions and speeds, and feedback data was included to simulate real-world scenarios.

The following steps outline the complete data analysis workflow:

1. **Data Generation:** Synthetic data was created using Python libraries such as `pandas`, `numpy`, and `random`. Key metrics such as noise levels (in dB), vibration (in m/s^2), and harshness scores were generated per test run. Data includes component-level information, speed levels, road surfaces, operating conditions, and timestamped entries along with user feedback.
2. **Data Structuring:** The data was split into a star schema structure comprising:
    - `Test_Data.csv` – Fact table containing all measurement values and foreign keys
    - `Model.csv`, `Component.csv`, `Surface.csv`, `Condition.csv` – Dimension tables describing categorical data

3. **Data Import and Cleaning:** The CSV files were loaded into Power BI using Power Query. The M language was applied to clean data types, remove null entries, eliminate duplicates, and prepare relationships between tables.

4. **Data Modeling:** A star schema was developed in Power BI by relating the fact table to dimension tables using primary-foreign key relationships. This layout enabled efficient filtering and slicing of data.

5. **Measure Table Creation:** A separate measure table was created to organize all DAX-based KPIs. This enhances clarity and keeps analytical logic separate from the raw data tables, following best practices in Power BI modeling.

6. **Date Table and Time Intelligence:** A dedicated date table was added to support time intelligence functions such as YTD, QTD, MTD, and rolling averages. However, due to the dataset spanning only one year, full application of time intelligence was limited in this analysis.

7. **Measure Creation (DAX):** Various DAX measures were created for insights, including:

   - `Average Noise by Component, Model, and Surface`
   - `Peak Vibration at Speed Levels`
   - `Harshness Score by Condition and Manufacturer`
   - `Correlation between Noise, Vibration and Harshness Score`

8. **Visualization and Interactivity:** Power BI Desktop was used to build interactive and insightful dashboards using a variety of visuals and features:

   - **Visuals Used:** Bar charts, Column charts, Line charts, KPI cards, Matrices, Tables
   - **Interactive Features:** Tooltips, Drill-up and Drill-down, Slicers, Filters, Hierarchies
   - **Custom Formatting:** Conditional formatting, Axis scaling, Labels and legends customization

   These visuals allowed the exploration of NVH performance across different manufacturers, conditions, components, and operational speeds in a dynamic and intuitive way.

9. **Collaboration and Deployment:** The report was published to the Power BI Service for online sharing, and GitHub was used for version control, ensuring reproducibility and project traceability.

6 Data Acquisition and Processing

6.1 Noise Data

Synthetic noise levels (in dB) were generated using `numpy` and `pandas`, simulating various speeds, surfaces, and components per manufacturer.

Visuals Used: Line charts (noise vs. speed), clustered bar charts (by surface/component), KPI cards, matrix tables, and slicers/tooltips for interactivity.

6.2 Vibration Data

Simulated vibration readings across X, Y, and Z axes represented frame/motor behavior. Amplitudes varied with terrain and speed. FFT was applied for frequency analysis.

Visuals Used: Line charts (vibration over speed), scatter plots, stacked columns (by axis/component), KPI cards, and drill-downs for detailed insights.

6.3 Harshness Scores

Harshness was calculated on a 1–5 scale based on noise, vibration, and conditions to reflect rider discomfort.

Visuals Used: Bar charts (by condition/model), heatmaps, matrix tables, scatter plots (correlation), and KPI cards for thresholds.

6.4 Data Modeling

Data was imported into Power BI via Power Query. Nulls were removed, columns cleaned, and relationships structured in a star schema. A date table and a separate DAX measure table were added to support future time intelligence functions.

6.5 Reporting Features

The final Power BI report includes slicers, drill-up/down capabilities, tooltips, and was published to Power BI Service. Version control was maintained via GitHub.

7 Key Findings

- The top three bike models with the highest average noise levels 2 were manufactured by **Specialized**.
- Most bikes across all manufacturers produced their lowest noise levels at a speed of **15 km/h**.
- Among the top three models with the highest average vibration levels, **two were produced by Specialized**.
- Bikes from all manufacturers exhibited their lowest vibration levels at speeds between **20 and 25 km/h**.

- **Harshness scores** were consistently lowest at **lower speed levels**.

8 Root Cause Analysis

- The **chain** and **motor** were commonly identified as major contributors to noise in the top noise-producing bike models.
- The **battery** and **motor** were frequently associated with high vibration levels in the top vibration-producing bikes.
- **Harshness levels** were observed to be significantly higher during rides on **forest trails**, likely due to uneven terrain and poor damping response.

9 Recommendations

- Improve **motor design and mounting** to minimize noise generation.
- Enhance **insulation and damping materials** around the battery to reduce vibration transmission to the bike frame.
- Upgrade the **shock absorption system** to provide a smoother ride, especially at higher speeds and on rough terrains.

10 Conclusion

This project successfully simulated and analyzed NVH data for electric bikes using Python and Power BI. Key issues were identified in components like motors and batteries, especially under higher speeds and rough terrains. The visualizations and KPIs provided actionable insights. Recommendations focused on improving component design and ride comfort.

Important Note

The data used in this analysis was generated using a Python-based random data generator function. **As a result, the metrics, trends, and outcomes shown in this report may significantly change upon data refresh or regeneration.** This approach was chosen for demonstration purposes and to simulate realistic NVH scenarios.

Online Access

You can access the live report via the Power BI Service here: Power BI Live Report