Sustainable Supply Chain Decision Tool — Technical Document

1. Project Overview

The **Sustainable Supply Chain Decision Tool** is a lightweight, beginner-friendly software solution designed to assist companies in **evaluating and selecting suppliers** based on multiple performance criteria. The tool uses a **decision matrix**, **Monte Carlo simulation**, and **clustering techniques** to provide a probabilistic and data-driven ranking of suppliers.

Key goals:

- Identify top-performing suppliers for sustainable procurement.
- Account for multiple metrics such as cost, delivery time, carbon footprint, quality, and compliance.
- Provide robust results under uncertainty using Monte Carlo simulation.
- Enable beginner-level coders to understand, modify, and extend the tool.

---

2. Technical Design

2.1 Architecture

notebooks/

│ ├── 01_data_generation.ipynb <- Generates synthetic supplier data

│ └── 01_decision_matrix_tool.ipynb <- Decision tool

src/

│ └── decision_matrix.py <- Helper functions (weighted scores, Monte Carlo, plotting)

data/

│ ├── suppliers.csv <- Supplier metrics

│ └── weights.csv <- Optional manual weights

2.2 Technology Stack

- Python 3.x
- Libraries: numpy, pandas, matplotlib, scikit-learn

2.3 Modules and Functions

- **Data Generation** (01_data_generation.ipynb):
    - Generates synthetic supplier data with realistic ranges.
    - Saves data to CSV for repeatability.
- **Decision Matrix Tool** (01_decision_matrix_tool.ipynb):
    - **Weighted Score Calculation:** Normalizes metrics and applies user-defined or optimized weights.
    - **Monte Carlo Simulation:** Simulates ±5% variation in metrics to estimate uncertainty in scores.
    - **Weight Optimization:** Searches grid of weight combinations to maximize expected top supplier score.

- **Clustering (KMeans):** Groups suppliers into clusters based on normalized metrics for pattern discovery.
- **Visualization:** Boxplots and histograms for supplier score distributions.
- **Top Supplier Recommendation:** Returns the highest-scoring supplier(s) under Monte Carlo simulation.

- **Helper Functions** (decision_matrix.py):
  - normalize_min_max()
  - calculate_weighted_score()
  - monte_carlo_simulation()
  - recommend_suppliers()
  - plot_mc_boxplot()
  - plot_mc_histogram()

---

3. Methodology

1. **Data Preparation:**
   - Supplier metrics include cost, delivery time, carbon footprint, quality rating, and compliance score.
   - All metrics are normalized using min-max scaling.
2. **Weighting & Decision Matrix:**
   - Users can supply manual weights (weights.csv) or use suggested optimized weights.
   - Weighted scores are calculated as the sum of normalized metrics multiplied by respective weights.
3. **Monte Carlo Simulation:**
   - For each supplier, metrics are perturbed ±5% over 1000 iterations.
   - Mean scores and ranks are computed to quantify uncertainty.
4. **Weight Optimization:**
   - Grid search over candidate weights summing to 1.0.
   - The weight combination that maximizes the top supplier mean score is suggested.
5. **Supplier Clustering:**
   - KMeans clusters suppliers into 3 groups based on normalized metrics.
   - Helps identify high-performing clusters.
6. **Visualization & Reporting:**
   - Boxplots for score distributions across suppliers.
   - Histograms for individual suppliers.

---

4. Features

Feature	Description
Weighted Scoring	Combines multiple metrics into a single score per supplier.
Monte Carlo Simulation	Estimates robustness of supplier ranking under metric uncertainty.
Weight Optimization	Suggests optimal weight allocation to maximize top supplier score.
Clustering	Groups suppliers to identify patterns and performance similarity.
Manual Weight Option	Users can input custom weights for decision flexibility.
Visualizations	Boxplots and histograms for intuitive understanding of scores.

---

5. Use Cases

1. **Sustainable Procurement:** Evaluate suppliers considering both cost and environmental impact.
2. **Risk-Aware Supplier Selection:** Incorporate uncertainty in metrics to reduce reliance on static scores.
3. **Supplier Benchmarking:** Compare suppliers in a cluster-based framework for strategic planning.
4. **Educational Tool:** Beginner-friendly ML + decision matrix example for learning purposes.

While the tool can theoretically be applied to areas like **trading or finance**, our focus on **supplier evaluation for sustainability** is unique and underexplored.

---

6. Code Implementation Highlights

- **Beginner-Friendly:** Uses only numpy, pandas, matplotlib, scikit-learn. No deep learning or complex libraries.
- **Modular:** All reusable code is in decision_matrix.py.
- **Interactive:** User chooses between manual or optimized weights.
- **Reproducible:** Synthetic supplier data ensures repeatable experiments.

Sample code snippet for weighted score calculation:

```
df_weighted = calculate_weighted_score(df, weights_dynamic, normalize_cols)
```

Sample code snippet for Monte Carlo:

```
df_mc, mc_results = monte_carlo_simulation(df_weighted, weights_dynamic, normalize_cols)
```

---

7. Performance Estimates

- **Data Size:** Tested with 10 suppliers and 5 metrics.
- **Monte Carlo Iterations:** 1000
- **Run Time Estimates (on standard laptop):**
    - Weighted score calculation: <0.01 sec

    - Monte Carlo simulation (1000 iterations): ~2–3 sec
    - Weight optimization (grid 0.1 increments): ~5–10 sec
    - KMeans clustering: <0.01 sec
- **Memory Usage:** <50 MB for this scale

These numbers are **estimated for small datasets**. Larger datasets may require longer simulation times, but the code remains beginner-friendly and modular.

---

8. Limitations and Extensions

- **Current Limitations:**
    - Grid search for weight optimization is computationally simple but not scalable to very large metric sets.
    - Monte Carlo uses ±5% variation — may need tuning for real-world data.
- **Possible Extensions:**
    - Allow dynamic selection of metrics.
    - Incorporate more sophisticated optimization algorithms (e.g., genetic algorithms) for weight tuning.
    - Include real supplier datasets for production use.
    - Add interactive dashboards (e.g., using Streamlit) for visualization.

---

9. Conclusion

The **Sustainable Supply Chain Decision Tool** provides a **robust, easy-to-understand framework** for supplier evaluation. By combining weighted scoring, Monte Carlo simulation, and clustering, it allows organizations to make **data-driven and uncertainty-aware decisions**.