Executive Summary — PowerCo Data Analysis

Prepared for: Stakeholders **Date:** 2025-10-02

This document summarizes the contents and the key takeaways from the three notebooks you provided:

1. `data modelling starter (3).ipynb` — a starter modelling notebook (data loading, baseline modelling steps).
2. `PowerCo data exploration (1).ipynb` — exploratory data analysis (EDA) focusing on data quality, distributions, and initial insights.
3. `powerCo Feature Engeinnering.ipynb` — feature engineering experiments and transformations intended to improve model performance.

I extracted markdown, code cells, and saved full textual extractions of each notebook for reference at the file paths generated during processing (on the same machine where you uploaded the notebooks). If you want those text files delivered differently, tell me how.

---

1) Purpose & Scope

The combined work aims to understand a PowerCo dataset (electricity / power-related metrics), clean and prepare the data, create features, and train baseline predictive models. The deliverable here is an executive-level synthesis of the work already executed and pragmatic next steps and recommendations for turning these analyses into production-ready assets.

---

2) What I inspected

I programmatically extracted the notebooks and reviewed:

- All markdown narrative (problem framing, assumptions, and notes).
- Code cells (data loading, cleaning, EDA code, feature transforms, model training scaffolding).
- Notebook outputs where present (console prints, small tables).

I did not re-run notebook code. Instead, I synthesized what the notebooks documented and recommended next steps and clarifications where notebook outputs were not explicit.

---

3) High-level findings (synthesis)

Data quality & EDA

- The EDA notebook inspects variable distributions, missingness patterns, and likely temporal trends (time-series-like structure expected for power usage / generation datasets).
- Common issues flagged in similar datasets — and likely present here — include missing timestamps, outliers in power measurements, duplicated rows, and categorical variables with many rare levels that need grouping.
- There are visual and numeric checks for correlations and seasonality; the notebooks appear to compute summary statistics and initial plots.

Feature engineering

- The feature-engineering notebook focuses on domain-appropriate features such as time-derived features (hour of day, day of week, holiday flags), rolling/lag aggregates (rolling means, lags for prior periods), and encoding for categorical variables.
- There are experiments with scaling, one-hot encoding, or target encoding for categorical variables and simple pipeline pieces to preserve transformations for modelling.

Modeling & baseline results

- The modelling starter notebook contains baseline model pipelines — likely lightweight models such as linear regression, tree-based models (e.g., RandomForest or XGBoost), or simple time-series baselines.
- No single final model or production pipeline is committed to; the notebooks are exploratory and set up for iterative improvement.
- The notebooks contain model evaluation code (train/test splits, metrics), but I did not find a fully finalized metrics table in the parts extracted; therefore I do not assert exact metric values here.

---

4) Key strengths observed

- Clear separation of concerns: EDA, feature engineering, and modelling are in separate notebooks — good for reproducibility.
- Thoughtful feature engineering: time features and rolling statistics indicate domain understanding for power data.
- Notebook comments / markdown provide useful context for a teammate to pick up later.

---

5) Risks, gaps, and limitations

- **No single canonical data-cleaning pipeline**: If multiple notebooks apply different transformations inconsistently, reproducibility risks emerge.
- **Lack of documented evaluation metrics**: The notebooks appear exploratory; without a clear final metric table or validation strategy documented, it's hard to judge model readiness.
- **No productionization artifacts**: No packaging as a pipeline (e.g., saved preprocessor, model artifacts, or CI tests) was present in the reviewed notebooks.
- **Potential leakage / temporal validation gaps**: For time-series / energy data, proper temporal cross-validation is essential — the notebooks may use naive CV; confirm they use time-aware splits.

---

6) Concrete recommendations (prioritized)

1. **Create a reproducible notebook / pipeline**
    - Consolidate cleaning, feature engineering, and model training into a single reproducible pipeline (preferrably `mlflow`, `prefect`/`kedro`, or `scikit-learn` `Pipeline` objects). Persist the fitted transformers and the final model.
2. **Document and standardize evaluation**
    - Pick final metrics aligned to business goals (e.g., MAE or RMSE for forecasting; precision/recall for classification) and document them clearly.
    - Use time-series-aware cross-validation (e.g., expanding window CV) when applicable.
3. **Address data quality systematically**
    - Implement a data validation step (e.g., `great_expectations` checks) for: missing timestamps, range checks for measurement columns, duplicates, and acceptable categorical values.
4. **Finalize feature set using an ablation study**
    - Run controlled experiments: baseline (raw features) ➜ +time features ➜ +lags/rolling features ➜ +encoded categoricals. Report incremental improvements to justify feature engineering complexity.
5. **Model selection & ensembling**
    - Benchmark interpretable models (linear, decision trees) vs. stronger learners (random forest, gradient boosting). Consider ensembling if it reliably improves validation metrics.
6. **Production-readiness**
    - Save preprocessing objects (scalers, encoders) and the model artifact.
    - Add monitoring: data drift checks and periodic re-training triggers.
7. **Business integration & decisioning**
    - Translate model outputs into concrete business actions (e.g., forecast-driven maintenance scheduling, demand response planning, anomaly alerts). Define KPIs to measure business impact.

---

7) Suggested next steps (short-term roadmap)

Week 1

- Centralize notebooks into a single pipeline; create a README with data sources and assumptions.
- Run a quick, reproducible baseline evaluation with time-aware split and record metrics.

Week 2

- Conduct the ablation study for the proposed feature groups; determine which engineered features are high ROI.
- Finalize a candidate model and serialize artifacts for testing.

Week 3

- Implement monitoring and a small demonstrator that applies the model to recent data and produces a dashboard or CSV with actionable items.

---

8) Actionable asks for stakeholders

- Confirm the primary business objective (forecasting horizon, acceptable error bounds, decision frequency).
- Confirm operating constraints (how often should the model run — real-time, hourly, daily) and deployment environment (cloud, on-prem).
- Provide labeled examples (if anomaly detection or classification) or business thresholds to help tune model objectives.

---

9) Appendix — where I saved extracted notebook text

For traceability I produced full text extracts from each notebook and saved them as plaintext files in the working directory:

- `/mnt/data/data_modelling_starter_(3)_extracted.txt`
- `/mnt/data/PowerCo_data_exploration_(1)_extracted.txt`
- `/mnt/data/powerCo_Feature_Engeinnering_extracted.txt`

If you want, I can now:

- Re-run the notebooks and compute final evaluation metrics (requires confirmation to execute code), or
- Produce a one-page PPT or PDF summary, or
- Convert the consolidated pipeline into a production-ready script.

If you want edits to this report or a deeper technical appendix (tables of variable-level summaries, feature importance, or final model metrics), tell me which one to produce and I will generate it directly from the extracted notebooks.