**Team A-Work Churn Modelling Jupyter Notebook Overview**

The Team A-Work Churn Modelling Jupyter Notebook applies a systematic, automated workflow to predicting an energy company’s customer turnover, or “churn.” This dynamic approach leverages reusable helper functions and patterns to generate dynamic model groups and pipelines, implementing a wide array of algorithms and balancing strategies while minimizing development toil. The notebook enables comprehensive analysis and comparison of models, culminating in the automated selection of single “Champion” model based on business priorities.

Because churn events represent less than ten percent of the records in the dataset, the notebook implements multiple sampling techniques in its pipelines, including SMOTE, BorderlineSMOTE, ADASYN, SMOTE-Tomek, SMOTE-ENN, Random Combination, and cost-sensitive weighting. The notebook implements several key machine learning algorithms, including ensembles based on method categories as top performers across various groupings. The model variations include :

* **Logistic Regression** (standard and cost-sensitive)
* **k-Nearest Neighbors (kNN)** with multiple balancing methods
* **Decision Trees** (unbalanced, balanced, and cost-sensitive)
* **Random Forest** (balanced, unbalanced, and cost-sensitive)
* **Gradient Boosting** (balanced and unbalanced)
* **XGBoost** (balanced, unbalanced, and cost-sensitive)
* **Diverse and Category-specific Ensembles**
* **Meta-Ensembles** combining top-performing models

By creating a dictionary of the trained models, the notebook is able to iterate over all models to apply uniform training and evaluation using a standardized set of functions. Where custom training is required to support unique model requirements (i.e., segment analysis), the notebook implements custom logic to train an test the model, but register the results in the model registry to support downstream analysis and execution against business questions.

Overall this approach supports rapid iteration under CRISP-DM, minimizing rework when new machine learning training algorithms, sampling techniques, or feature modifications.

**Use of Generative AI in the Team A-Work Churn Modelling Jupyter Notebook**

The Team A-Work Churn Modelling Jupyter Notebook leveraged GitLab Co-Pilot throughout the development process to provide code hints and generate blocks of autogenerated code code based on developer prompts (GitHub, 2025). The models used include GPT 4o, GTP 4.1, and Claud Sonnet 4. Team A-Work data scientists also leveraged OpenAI’s ChatGTP models (GPT 4o, GPT 4.1, and GPT 5) to perform troubleshooting and analysis of notebook code (OpenAI, 2025).

Applying responsible and ethical development practices, Team A-Work reviewed every line of code created by Generative AI tools, eliminating hallucinations (like hard-coded values or non-existent features) and verifying that code was not plagiarized from other sources. Because this notebook appears contains standard code libraries (like scikit-learn), pipeline patterns, sampling methods, and reporting code cells may resemble examples in documentation, tutorials, and “boilerplate” code. Where code is replicated from another source, the author is attributed in the notebook.

References

GitHub. (2025). GitHub Copilot (June - August). <https://github.com/features/copilot>

OpenAI. (2025). ChatGPT (June - August). <https://chat.openai.com/>