

Notebook Explanations Summary

1. Import & Load Data

You started by importing core Python libraries and loading the dataset with error handling. This step ensures your workflow is reproducible and robust.

- Beginner: You loaded your dataset and checked its structure to make sure it's correct.
- Professional: Added defensive programming with try/except blocks for loading and verified with `df.info()`.

2. Temporal Feature Engineering

You created datetime features like hour, minute, day_of_week, weekend flag, and time_bin.

- Beginner: You transformed raw time info into features that help the model learn.
- Professional: Extracted cyclic/seasonal signals, and discretized time into interpretable bins for potential operational grouping.

3. Waiting Time Calculation

You computed the target variable: waiting time in minutes. You removed zero-times, clipped negatives, and filtered out durations longer than 24h.

- Beginner: You cleaned up incorrect data and created the variable you want to predict.
- Professional: Ensured temporal consistency, eliminated potential noise from bad logs, and bounded target variable range for stable training.

4. Dropping Redundant/Leaky Columns

You removed IDs, raw times, and post-service variables that could leak future information into training.

- Beginner: You removed useless columns that wouldn't help the model.
- Professional: Followed best practices by removing leaky and high-cardinality identifiers to reduce overfitting.

5. Categorical Encoding & Feature Engineering

You applied ordinal encoding to 'prefixe' and one-hot encoding to other categorical features. You also created interaction and polynomial features.

- Beginner: You turned text into numbers so the model can use them.
- Professional: Carefully encoded features with custom order + added non-linear transformations (squared, interactions) to enhance signal.

6. Train-Test Split & Target Distribution

You split the dataset into 80% training and 20% testing, and visualized the target distribution.

- Beginner: You separated the data so the model can learn and then be tested.
- Professional: Stratified for generalization, visually verified skew in the regression target, validated input-output shape.

7. Model Training and Evaluation

You trained and evaluated both Random Forest and XGBoost models, displaying metrics and feature importance.

- Beginner: You built 2 models and checked which one works better.
- Professional: Logged MAE, RMSE, R, and training time. Compared using consistent random state. Visualized top predictors for both models.

8. Enhanced Feature Set and Re-evaluation

You retrained the models using added features like `nb_attente_appel_squared` and `hour_x_service`. Then you compared performance before and after.

- Beginner: You made your model smarter by adding more info.
- Professional: Demonstrated iterative modeling improvement via feature enrichment. Random Forest benefited, XGBoost slightly degraded (possibly overfitting).

9. SHAP Explainability

You used SHAP to analyze how each feature impacts predictions.

- Beginner: You saw which inputs affect the model's decisions.
- Professional: Generated global SHAP summary for tree model; verified top features align with domain logic; enhanced transparency.

10. Residual Analysis

You visualized the residuals (errors) to check how well your model fits.

- Beginner: Most predictions are close to the real value, but some are far.
- Professional: Distribution centered around zero with few outliers. Residual spread confirms good bias-variance trade-off.

11. Segment-Specific R by Prefix

You evaluated model accuracy (R) for different `prefix_encoded` groups.

- Beginner: Your model works better for some types of tickets than others.
- Professional: Identified performance variance by customer segment; Prefix 4/2 are highly predictable, Prefix 3 needs deeper investigation or custom modeling.