Modeling Process Documentation

1. Objective

The goal of the modeling stage was to predict the **enhanced risk score** for each driver using trip-level and driver-level features. This score serves as the foundation for determining insurance premiums, so model performance and reliability are critical.

---

2. Initial Approach

I started with tree-based models that are commonly strong performers on structured tabular data:

- **XGBoost Regressor**
  - Reason: It's a well-established gradient boosting model known for its ability to handle non-linear relationships and complex feature interactions.
  - Why first: It often serves as a strong baseline in structured datasets and has extensive community support for tuning and optimization.
- **CatBoost Regressor**
  - Reason: Specifically designed for tabular data with categorical variables, which made it attractive since `vehicle_type` is categorical.
  - Why next: It requires less manual preprocessing of categorical features and tends to be more robust with minimal tuning.

---

3. Expanding the Model Set

After experimenting with boosting algorithms, I broadened the search to include other strong, interpretable ensemble methods:

- **RandomForest Regressor**

    - Reason: Provides robustness by averaging across many decision trees.
    - Benefit: Less prone to overfitting than a single decision tree and provides feature importance measures.
    - Why included: A benchmark against boosting methods to see whether bagging-based models performed better.
- **GradientBoosting Regressor (sklearn)**
    - Reason: Another boosting method, simpler and more interpretable compared to XGBoost.
    - Why included: For comparison, to see how a "vanilla" gradient boosting approach stacked up against XGBoost and CatBoost.

---

4. Stacking Ensemble

Finally, I combined multiple base learners into a **stacking ensemble**:

- **Base estimators:** RandomForest, XGBoost, GradientBoosting.
- **Final estimator:** Ridge regression, chosen for stability and simplicity.
- **Passthrough:** Enabled so the meta-learner could use both the raw features and the base models' predictions.

Reasoning:
Stacking leverages the strengths of different algorithms. For example:

- RandomForest → handles variance well.
- XGBoost → captures complex interactions.
- GradientBoosting → interpretable baseline.
  The Ridge layer then balances their predictions, reducing error.

---

5. Preprocessing

- Used a **ColumnTransformer** to one-hot encode categorical variables (`vehicle_type`) and pass through numeric variables.
- This ensured consistent handling across pipelines, especially for models that don't natively support categorical inputs (RandomForest, XGBoost, GradientBoosting).
- CatBoost was run directly without preprocessing since it handles categoricals internally.

---

6. Evaluation Strategy

- **Cross-Validation (5-fold)**: Used for fair performance assessment across all models, reducing dependence on a single train-test split.
- **Hold-Out Test Set (20%)**: Used for final evaluation to simulate real-world generalization.
- **Metrics:**
    - **MAE (Mean Absolute Error):** Measures average prediction error in interpretable units.
    - **RMSE (Root Mean Squared Error):** Penalizes larger errors more heavily.
    - **R^2 (Coefficient of Determination):** Explains proportion of variance captured by the model.

---

7. Results

Hold-Out Test Results (selected):

- CatBoost → MAE: 5.99, RMSE: 6.53, R^2: 0.625
- RandomForest → MAE: 6.26, RMSE: 7.15, R^2: 0.552
- XGBoost → MAE: 9.06, RMSE: 9.95, R^2: 0.130
- GradientBoosting → MAE: 9.17, RMSE: 9.72, R^2: 0.171
- **Stacking Ensemble → MAE: 1.85, RMSE: 2.28, R^2: 0.955**

Interpretation:

- CatBoost outperformed other single models due to its ability to handle categorical features efficiently.
- RandomForest was a close second, providing a good benchmark.
- XGBoost and GradientBoosting underperformed in this setup, possibly due to limited tuning and higher sensitivity to hyperparameters.
- The Stacking Ensemble significantly outperformed all individual models, suggesting that the combination of different learners captured complementary patterns in the data.

---

8. Why This Progression Made Sense

1. Start with strong boosting baselines (XGB, CatBoost).
2. Compare against bagging (RandomForest) and simpler boosting (GradientBoosting).
3. Use ensemble stacking to combine the best elements of each.
4. Evaluate systematically with cross-validation and a hold-out set.

This progression allowed me to test both **individual model performance** and the potential of **hybrid approaches**, ultimately confirming that an ensemble was the most powerful option for this dataset.