

quantile_cv.py

Summary

This code defines a `QuantileCV` class for automated quantile regression tree modeling with Bayesian optimization and pluggable boosting backends.

Dependencies

Standard Library

- `typing`
- `functools`

Other

- `numpy`
- `pandas`
- `pydantic`
- `sklearn`
- `xgboost`
- `catboost`

Description

The `quantile_cv.py` file implements a `QuantileCV` class, which provides automated quantile regression using tree-based models and Bayesian optimization. This class is designed for flexible and robust quantile regression tasks, leveraging advanced machine learning techniques.

The implementation includes:

1. `QuantileCVConfig`: A Pydantic dataclass that defines configuration options for quantile regression, including:

2. `monotone_constraints` : Dictionary specifying monotonicity direction for variables (0 for none, 1 for increasing, -1 for decreasing)
3. `interactions` : List of lists containing permitted feature interactions
4. `n_jobs` : Number of parallel jobs to use when fitting the model
5. `parameters` : Hyperparameter search space definition (OptimizerParams instance)
6. `return_train_score` : Whether to include training scores during optimization
7. `quantile_alpha` : Parameter specifying the quantile to predict (e.g., 0.5 for median regression)
8. Pre-configured settings:
 9. `default_quantile` : A standard configuration using all hyperparameters
 10. `balanced_quantile` : A configuration focused on regularization parameters
 11. `QuantileCV` : The main class that inherits from `BaseAutoCV` and provides:
 12. Automated hyperparameter tuning via Bayesian optimization
 13. Backend selection via `backend` ("xgboost" or "catboost")
 14. Support for custom quantile levels via the `alpha` parameter
 15. Integration with scikit-learn's scoring system for quantile regression metrics
 16. Parallel processing support

The class uses cross-validation during the optimization process to ensure model robustness and prevent overfitting. It supports various regression metrics, with a focus on quantile loss, and leverages Pydantic for configuration validation.

Overall, this module offers a comprehensive solution for quantile regression tasks, combining automated model selection, advanced regression techniques, and robust validation mechanisms.