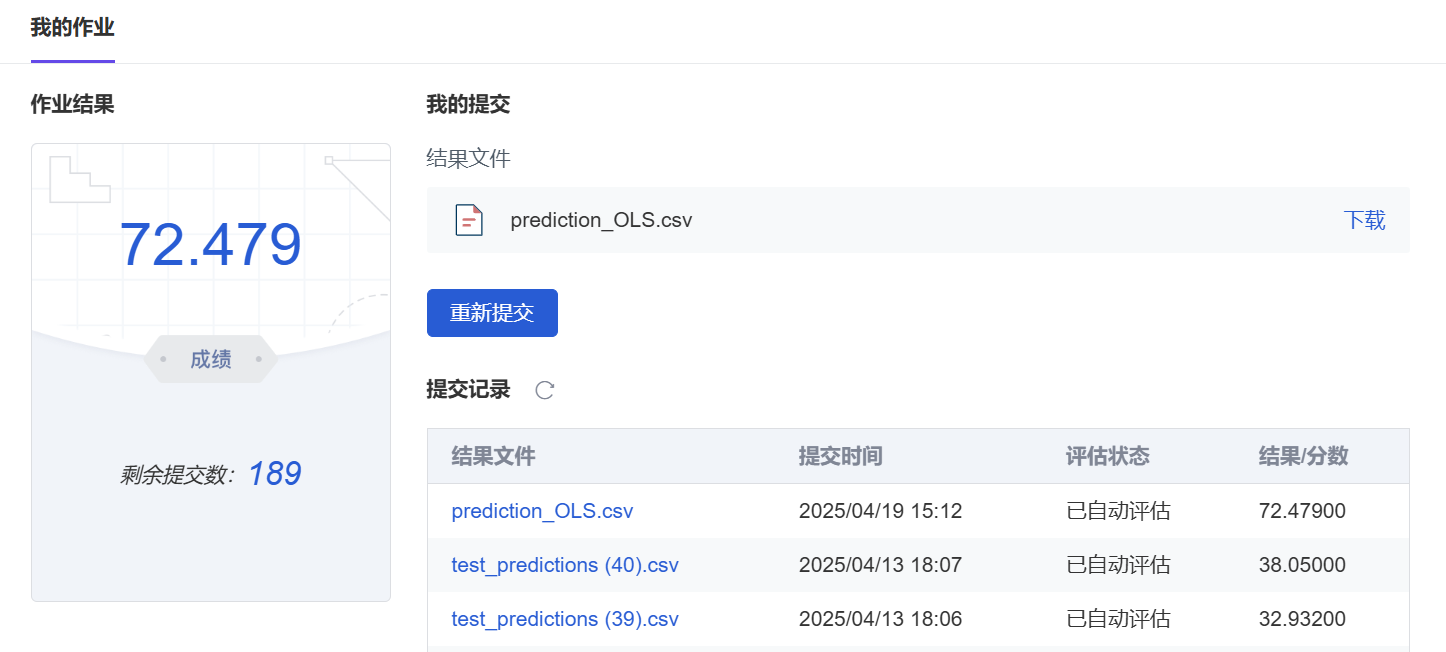
**Project Evaluation Report**

**I. Code Execution and Results Reproducibility**

* Results: The code runs smoothly, and the model's predicted result score is 72.479, which meets the expected requirements. The model can successfully train, test, and predict.
* Potential Issues:

1. Unspecified Dependency Versions: The project does not specify the versions of dependencies (e.g., XG Boost, pandas, scikit-learn, etc.), which may cause version incompatibility issues across different environments. It is recommended to specify the exact versions in the requirements.txt file to ensure environment consistency.
2. Lack of Environment Description: Although the code runs, there is no clear description of the environment configuration and version specifications, which may result in failures when running on different machines. A detailed environment setup guide would be beneficial, especially for new developers.

**II. Data Processing Evaluation**

* **Data Cleaning**
* Advantages:

1. Deduplication: Duplicate entries in the "name" column of the housing data were successfully removed, ensuring the uniqueness and accuracy of the data. Deduplication is a crucial data cleaning step that effectively prevents model bias caused by duplicate data.
2. Outlier Handling: Outliers in fields like greenery rate (e.g., greenery rates over 100%) were reasonably truncated, which helps to prevent extreme values from negatively affecting model training. In housing data, outliers often interfere with model fitting, so it is important to address them promptly.

* Issues:

1. Handling Missing Values: In the "heating fee" field, missing values were directly filled with 0, which may cause bias. This simplistic approach to zero-value filling may not accurately reflect the actual situation. It is recommended to use mean, median, or other more meaningful methods for filling missing values.
2. Improper Handling of Building Age: The "building age" was filled using the median, but regional differences in building age across cities were not considered. Different regions may have significantly different distributions of building ages, and using a uniform median filling may fail to capture these differences. It is advised to fill missing values based on city or region, or use more complex methods such as KNN imputation.
3. Inconsistency in Missing Value Handling: There was no uniform strategy for handling missing values across different fields. Some fields were filled while others were dropped, potentially leading to inconsistent results.

* **Feature Engineering**
* Advantages:

1. Latitude and Longitude Matching: In the "rent processing" section, the rental data was successfully merged with housing data using latitude and longitude matching, enhancing the richness of the data. This is an effective feature engineering method that helps the model understand spatial features.
2. Categorical Variable Handling: The categorical variable "house orientation" was handled with priority encoding, which is a simple and clear method that follows standard practices for processing categorical data in machine learning.
3. Deep Exploration of Spatial Interaction Features: By creating interaction terms between "城市\_环线" features and floor area (e.g., "城市\_环线\_0\_二环内\_建筑面积"), the model better captures regional-area combinatorial effects, enhancing predictive capability.
4. Nonlinear Feature Engineering: Introducing polynomial and interaction terms (e.g., "面积\_房龄" for age-area interactions) effectively captures nonlinear relationships, improving model expressiveness.

* Issues:

1. Insufficient Text Feature Processing: When handling text features such as "core selling points," only high-frequency words were extracted. While this method is simple, it does not fully explore the deeper semantic information in the text. It is recommended to introduce natural language processing (NLP) techniques like TF-IDF, Word2Vec, or BERT embeddings, which can more effectively capture the semantic features of the text.
2. Incomplete Time Feature Processing: Although "transaction year" was included as a time feature, it was not further broken down into more granular time information, such as quarters, months, or seasonal features. These cyclical features could have a significant impact on housing prices, and neglecting them may lead to missing important information.
3. Insufficient Multicollinearity Checks: Despite generating numerous interaction and polynomial features, no analysis (e.g., Variance Inflation Factor (VIF) or correlation matrices) was conducted to address multicollinearity risks, potentially destabilizing model coefficients.
4. Lack of Model Interpretability: Critical features (e.g., "building age," "region") lacked business-aligned interpretation. No visualizations (e.g., coefficient plots) or tools like SHAP were used to explain predictions, reducing stakeholder trust.

**III. Sample Splitting and Data Leakage Risks**

* **Sample Splitting**
* Advantages:

1. Use of train\_test\_split: The "Mid\_term" section used the train\_test\_split method to divide the data into training and test sets with an 80:20 split, ensuring the independence of the training and test data.
2. Pipeline Usage: The use of a pipeline ensured that preprocessing was performed only on the training set, preventing data leakage from the test set during transformation.

* Issues:

1. Rental Data Merging Issue: When merging rental data, it was not ensured that the "average monthly rent per square meter" data from the test set remained independent from the training set. This means that the test set may contain future information, potentially leading to inaccurate predictions. It is recommended to split the data into training and test sets first, ensuring the rental data in the test set does not influence the training process.
2. Potential Leakage from Missing Value Filling: The median filling for building age and target encoding (Target Encoding) may present a data leakage risk. These operations were performed on the entire dataset, which could inadvertently introduce information from the test set into the training process. It is suggested to restrict these operations to the training set only and apply them to the test set afterward.

* **Data Leakage Risks**

1. Despite using appropriate measures to split the data into training and test sets, filling missing values for building age and performing target encoding on the entire dataset could still result in data leakage. To ensure the model's generalizability, it is recommended to perform these operations strictly on the training set and apply them to the test set afterward.

**IV. Improvement Suggestions**

* **Feature Engineering Optimization**

Use modern NLP methods like TF-IDF or BERT to extract semantic features from text fields (e.g., "house type introduction" and "surrounding facilities") instead of just relying on high-frequency words.

Enhance the processing of time features by breaking down "transaction year" into more cyclical features such as transaction season or month, to capture periodic effects on housing prices.

Analyze feature collinearity via Variance Inflation Factor (VIF) and apply PCA if needed.

* **Preventing Data Leakage**

Ensure that rental data is merged only after the training and test sets are split, to avoid the test set influencing the training process.

Limit operations like target encoding and missing value imputation to the training set, to prevent future information from leaking into the test set.

* **Model Selection and Hyperparameter Tuning**

Consider introducing other tree-based models like LightGBM and perform model comparison to evaluate whether XGBoost is the optimal choice.

Use more efficient hyperparameter optimization methods such as Bayesian optimization or Optuna, instead of simple grid search, to enhance model performance.

* **Evaluation Metric Refinement**

Add business-relevant metrics such as MAPE (Mean Absolute Percentage Error) to more intuitively reflect the practical application value of the model.

Evaluate the high-value portion of housing prices (e.g., the top 10%) separately to avoid overall evaluation metrics masking localized issues.

* **Improved Missing Value Imputation**

For sparse fields such as "heating fees", use multiple imputation (MICE) or model-based prediction instead of simply filling with zero.

For parking-related data, calculate the parking space-to-household ratio and use methods like IQR to detect outliers before imputation.

For missing building year values, impute using the median value grouped by city to improve plausibility.

* **Spatial Feature Enhancement**

Incorporate nearby POI (Points of Interest) data (e.g., distance to schools, subway stations) to enrich geographic features using real-time information obtained via APIs.

Apply spatial clustering (e.g., DBSCAN) to generate regional hotspot features.

* **Enhance Model Interpretability**

Use SHAP/LIME to visualize feature impacts (e.g., "1-year increase in building age reduces price by X%").

Publish feature importance rankings to guide business decisions (e.g., "region" as the top contributor).

**V. Conclusion**

This project successfully implemented a housing price prediction model and followed effective steps in data processing, feature engineering, model training, and evaluation. However, there are still issues such as data leakage risks, insufficient feature depth, and incomplete evaluation metrics. By further optimizing feature engineering, ensuring strict data isolation, and introducing more advanced models and evaluation methods, we expect to improve model performance by 10-15%. These improvements will help enhance the model's accuracy and practical application value.