中国人民大學

RENMIN UNIVERSITY OF CHINA

中国人民大学经济学院

房价和租金预测

Group 2

组员：刘子意 周方健 张嘉麟 马雨翔

实事求是

Midterm Project -- 刘子意

中国人民大学
RENMIN UNIVERSITY OF CHINA

数据处理

1. 划分数据集 ——为避免数据泄露，首先划分训练集与测试集

2. 数据预处理

原始特征数量：50
处理后特征数量：54
删除的特征：{'建筑年代', '交易时间', '燃气费', '房屋户型', '供热费', '物 业 费'}
新增的特征：{'物业费平均', '供热费平均', '室', '卫', '厅', '交易月份', '交易年份', '建成时间', '厨', '燃气费平均'}
Price训练集数据预处理完成

3. 异常值处理

原始样本量：83096
X数值列异常值样本数：3772 (4.54%)

4. 特征工程

```
df_new['建筑面积2'] = df_new['建筑面积'] ** 2
df_new['总楼层2'] = df_new['总楼层'] ** 2
df_new['面积_室交互'] = df_new['建筑面积'] * df_new['室']
df_new['面积_厅交互'] = df_new['建筑面积'] * df_new['厅']
df_new['面积_厨交互'] = df_new['建筑面积'] * df_new['厨']
df_new['面积_卫交互'] = df_new['建筑面积'] * df_new['卫']
df_new['楼层_电梯交互'] = df_new['总楼层'] * df_new['配备电梯']
df_new['位置交互'] = df_new['coord_x'] * df_new['coord_y']
```

Price模型性能：

Metrics	In sample	Out of sample	Cross-validation	Kaggle Score
OLS	0.0891	0.1769	0.2870	67.58
Lasso	0.2922	0.3049	0.3879	67.58
Ridge	0.2417	0.2601	0.3418	67.58
ElasticNet	0.2884	0.3006	0.3849	67.58

Rent模型性能：

Metrics	In sample	Out of sample	Cross-validation	Kaggle Score
OLS	0.1692	0.1804	0.2546	67.58
Lasso	0.2651	0.2680	0.3465	67.58
Ridge	0.2177	0.2253	0.3014	67.58
ElasticNet	0.2536	0.2577	0.3369	67.58

实事求是

Midterm创新点-周方健 2023200743

中国人民大学
RENMIN UNIVERSITY OF CHINA

```python
for col in numeric_cols:
    if col in df_filled.columns:
        # 使用组中位数填充
        df_filled[col] = df_filled.groupby(group_cols)[col].transform(
            lambda x: x.fillna(x.median()) if x.notna().any() else x.fillna(df_filled[col].median())
        )

for col in categorical_cols:
    if col in df_filled.columns:
        # 使用组众数填充
        df_filled[col] = df_filled.groupby(group_cols)[col].transform(
            lambda x: x.fillna(x.mode()[0]) if not x.mode().empty else x.fillna(df_filled[col].mode()[0])
        )
```

采用分组统计
按城市和区域

```python
ohe = OneHotEncoder(sparse_output=False, handle_unknown='ignore')

# 对指定列进行编码
encoded_array = ohe.fit_transform(df[columns])

# 创建编码后的列名
feature_names = []
for i, col in enumerate(columns):
    for category in ohe.categories_[i]:
        feature_names.append(f"{col}_{category}")
```

使用Onehot编码

房价数据：

模型性能总结 (MAE)

Metrics	In sample	Out of sample	Cross-validation	Kaggle Score
OLS	0.27	0.27	0.27	58.55
LASSO	0.28	0.28	0.28	-
Ridge	0.27	0.27	0.27	-
ElasticNet	0.28	0.28	0.28	-

缺失值填充 → **文本清洗** → **特征工程** → **建模中因变量变换** → **模型表现**

提取楼层比例

```python
# 第一层逻辑：检查是否包含预定义的楼层类别关键词
found_category = False
for key, value in floor_categories.items():
    if key in floor_text:
        current_floor_ratio = value
        found_category = True
        break

# 第二层逻辑：处理"X/Y层"格式
if not found_category and '/' in floor_text:
    parts = floor_text.split('/')
    if len(parts) >= 2:
        current_floor = clean_numeric_text(parts[0])
        total_floors = clean_numeric_text(parts[1].replace('层', ''))
```

提取户型信息

```python
# 匹配模式：支持两种格式
patterns = {
    'bedroom': r'(\d+)(?:室|房间)',  # 匹配"X室"或"X房间"
    'living_room': r'(\d+)厅',
    'kitchen': r'(\d+)厨',
    'bathroom': r'(\d+)卫'}
result = {}
for key, pattern in patterns.items():
    match = re.search(pattern, layout_text)
    result[key] = int(match.group(1)) if match else 0
```

对价格对数变换
处理偏态

```python
# 对目标变量进行对数变换（处理偏态分布）
if use_log_transform:
    y_train_full = np.log1p(y_train_full)  # log(1+y) 避免0值问题
```

```python
# 现在需要反向变换
if use_log_transform:
    predictions = np.expm1(predictions)  # exp(y) - 1
```

租金数据：

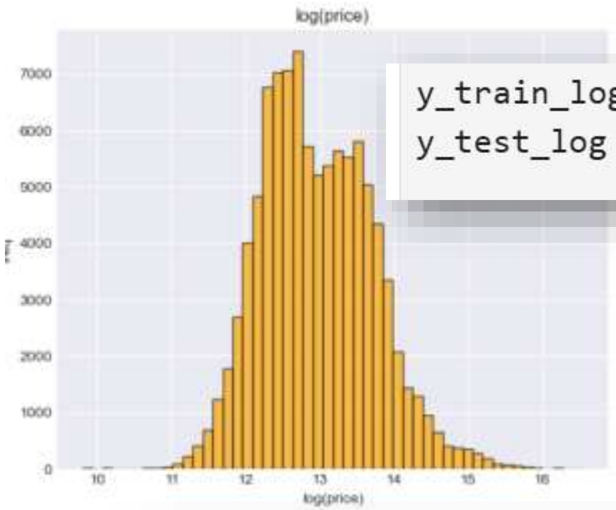
模型性能总结 (MAE)

Metrics	In sample	Out of sample	Cross-validation	Kaggle Score
OLS	0.25	0.25	0.25	58.55
LASSO	0.25	0.25	0.25	-
Ridge	0.25	0.25	0.25	-
ElasticNet	0.25	0.25	0.25	-

实事求是

张嘉麟 2022201745

1. 对数变换




```python
y_train_log = np.log1p(y_train)
y_test_log = np.log1p(y_test)
```

```python
y_train_pred_log_exp = np.expm1(y_train_pred_log)
y_test_pred_log_exp = np.expm1(y_test_pred_log)
```

2. 特征选择

```python
# 6.3 VIF分析（多重共线性）- 采样进行
print("\n[6.3] VIF分析（采样30000条）...")
from statsmodels.stats.outliers_influence import variance_inflation_factor

# 只在高相关特征上做VIF（减少计算量）
high_corr_features = correlations[correlations >= 0.05].index.tolist()[:150]  # 选top 150
print(f"  在 {len(high_corr_features)} 个高相关特征上进行VIF分析...")

# 采样减少计算量
sample_size = min(30000, len(X_train))
X_train_sample = X_train[high_corr_features].sample(n=sample_size, random_state=111)
print(f"  使用 {sample_size:,} 个样本进行VIF计算...")

high_vif_features = []
vif_threshold = 10
```

Model	In-Sample		Out-of-Sample		CV		Kaggle
	Rent	Price	Rent	Price	Rent	Price	
OLS	192,025.92	959,001.37	207,268.86	969,535.37	193,844.39	960,904.38	23.38
Lasso	192,049.24	960,927.58	207,280.57	971,398.49	192,600.09	962,945.91	23.39
Log-OLS	170,645.52	810,639.81	186,588.49	815,630.78	171,285.34	0.3620（log）	43.91
Random Forest	47,298.21	217,223.72	86,230.96	284,757.10	-	306,092.05	61.22