中國人民大學 财政金融学院
RENMIN UNIVERSITY OF CHINA | SCHOOL OF FINANCE

RENMIN UNIVERSITY OF CHINA

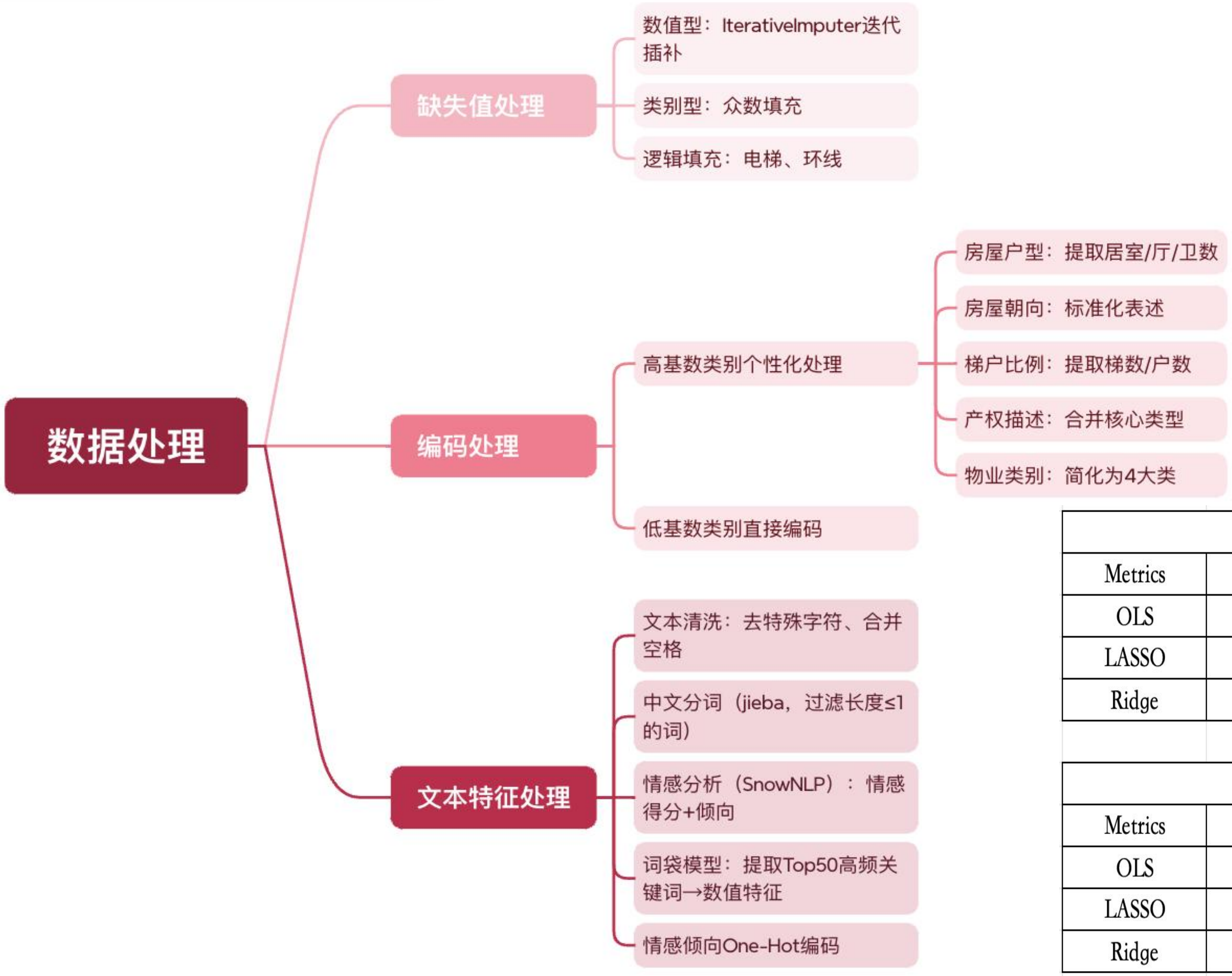
第三组展示

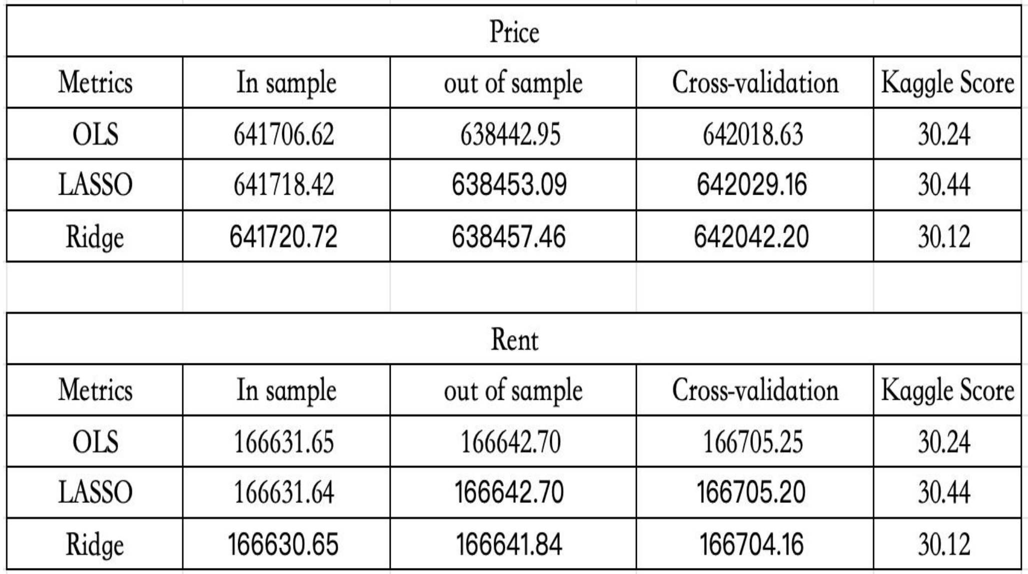
韩方媛

- **特征创造:**
  - 非线性变换+多项式特征：对数处理面积、价格等偏态数据，适配线性模型假设；新增面积平方项，捕捉价格与面积的非线性增长关系；
  - 交互特征构建：设计房间数 / 面积、梯户比、每栋户数等交互项，挖掘特征协同效应；
  - 类别型变量处理：对城市、楼层、供暖方式等分类变量进行独热编码；对环线、朝向、装修等级等可量化类别变量等人工设置得分；
  - 空缺值填空：数值型变量使用中位数填充，类别型变量使用众数填充；
- **异常值处理:**
  - IQR方法去除price异常值；Z-score 标准化去除数值型特征异常值；
- **数据结果**
  - 最终price清理后数据形状: (76408, 85)，rent清理后数据形状: (71622, 93)。

Metrics	Data	In sample	Out of sample	Cross-validation	Kaggle Score
OLS	price	0.2620	0.2633	0.2624	56.76
	rent	0.2207	0.2221	0.2210	
LASSO	price	0.2633	0.2640	0.2635	56.78
	rent	0.2235	0.2245	0.2238	
Ridge	price	0.2620	0.2633	0.2624	56.76
	rent	0.2207	0.2221	0.2210	
ElasticNet	price	0.2623	0.2632	0.2626	56.84
	rent	0.2215	0.2228	0.2219	

郭德昊





Price				
Metrics	In sample	out of sample	Cross-validation	Kaggle Score
OLS	641706.62	638442.95	642018.63	30.24
LASSO	641718.42	638453.09	642029.16	30.44
Ridge	641720.72	638457.46	642042.20	30.12

Rent				
Metrics	In sample	out of sample	Cross-validation	Kaggle Score
OLS	166631.65	166642.70	166705.25	30.24
LASSO	166631.64	166642.70	166705.20	30.44
Ridge	166630.65	166641.84	166704.16	30.12

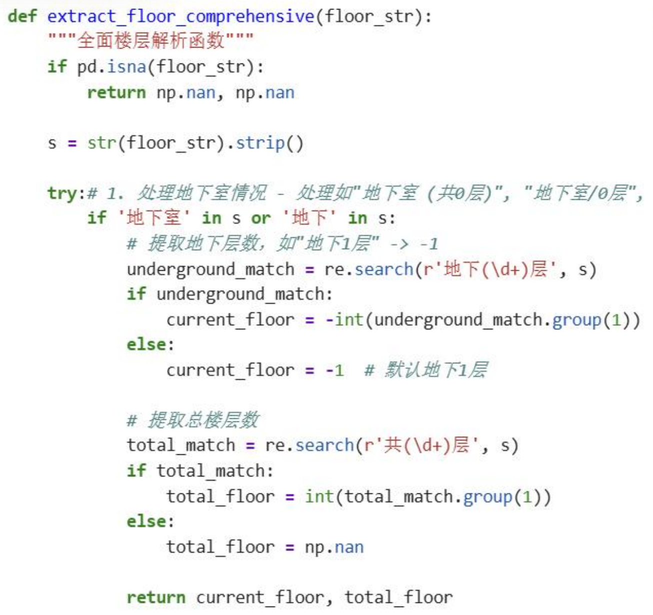
张海源

楼层解析

地下室（共0层)
地下1层

高楼层（共20层)
中楼层（共10层)

9/12层
高楼层/25层



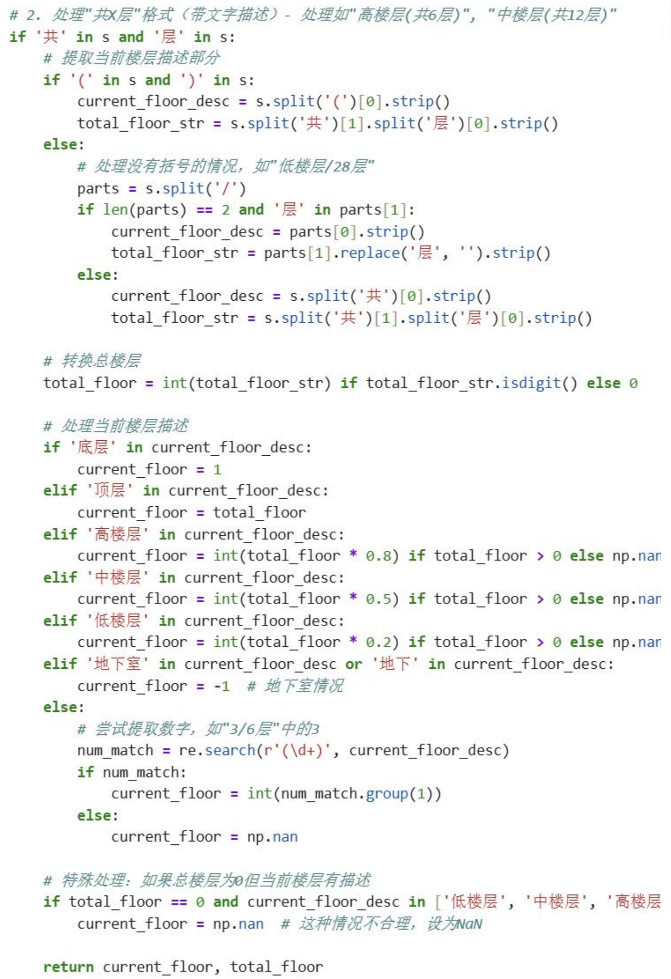
```python
def extract_floor_comprehensive(floor_str):
    """全面楼层解析函数"""
    if pd.isna(floor_str):
        return np.nan, np.nan

    s = str(floor_str).strip()

    try:# 1. 处理地下室情况 - 处理如"地下室 (共0层)", "地下室/0层",
        if '地下室' in s or '地下' in s:
            # 提取地下层数，如"地下1层" -> -1
            underground_match = re.search(r'地下(\d+)层', s)
            if underground_match:
                current_floor = -int(underground_match.group(1))
            else:
                current_floor = -1  # 默认地下1层

            # 提取总楼层数
            total_match = re.search(r'共(\d+)层', s)
            if total_match:
                total_floor = int(total_match.group(1))
            else:
                total_floor = np.nan

            return current_floor, total_floor
```



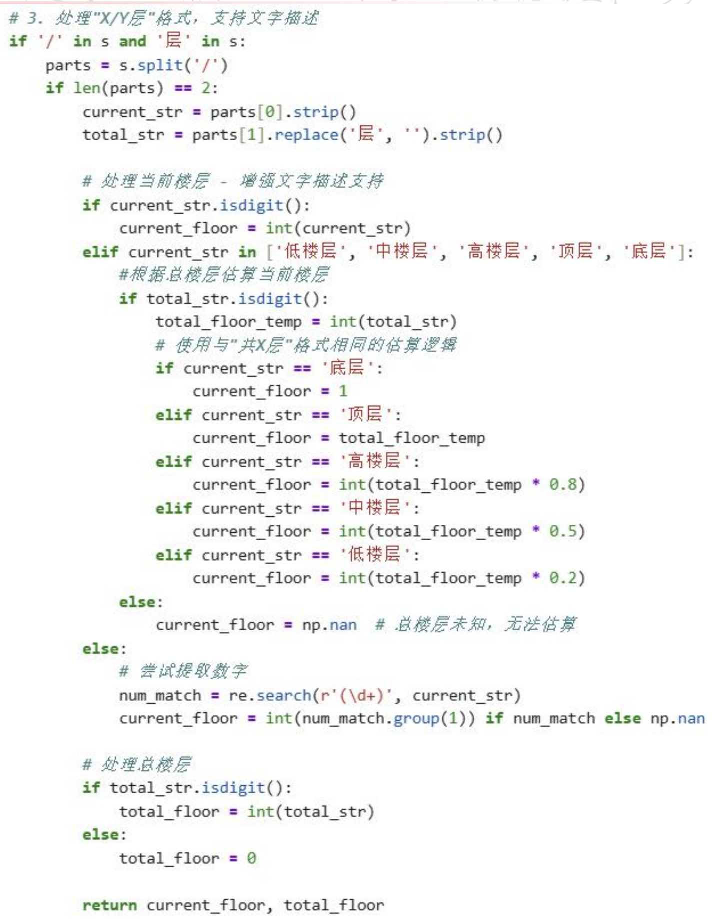
```python
# 2. 处理"共X层"格式（带文字描述）- 处理如"高楼层(共6层)", "中楼层(共12层)"
if '共' in s and '层' in s:
    # 提取当前楼层描述部分
    if '(' in s and ')' in s:
        current_floor_desc = s.split('(')[0].strip()
        total_floor_str = s.split('共')[1].split('层')[0].strip()
    else:
        # 处理没有括号的情况，如"低楼层/28层"
        parts = s.split('/')
        if len(parts) == 2 and '层' in parts[1]:
            current_floor_desc = parts[0].strip()
            total_floor_str = parts[1].replace('层', '').strip()
        else:
            current_floor_desc = s.split('共')[0].strip()
            total_floor_str = s.split('共')[1].split('层')[0].strip()

    # 转换总楼层
    total_floor = int(total_floor_str) if total_floor_str.isdigit() else 0

    # 处理当前楼层描述
    if '底层' in current_floor_desc:
        current_floor = 1
    elif '顶层' in current_floor_desc:
        current_floor = total_floor
    elif '高楼层' in current_floor_desc:
        current_floor = int(total_floor * 0.8) if total_floor > 0 else np.nan
    elif '中楼层' in current_floor_desc:
        current_floor = int(total_floor * 0.5) if total_floor > 0 else np.nan
    elif '低楼层' in current_floor_desc:
        current_floor = int(total_floor * 0.2) if total_floor > 0 else np.nan
    elif '地下室' in current_floor_desc or '地下' in current_floor_desc:
        current_floor = -1  # 地下室情况
    else:
        # 尝试提取数字，如"3/6层"中的3
        num_match = re.search(r'(\d+)', current_floor_desc)
        if num_match:
            current_floor = int(num_match.group(1))
        else:
            current_floor = np.nan

    # 特殊处理：如果总楼层为0但当前楼层有描述
    if total_floor == 0 and current_floor_desc in ['低楼层', '中楼层', '高楼层
        current_floor = np.nan  # 这种情况不合理，设为NaN

    return current_floor, total_floor
```



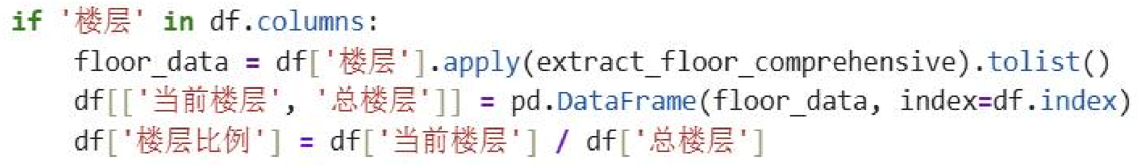
```python
# 3. 处理"X/Y层"格式，支持文字描述
if '/' in s and '层' in s:
    parts = s.split('/')
    if len(parts) == 2:
        current_str = parts[0].strip()
        total_str = parts[1].replace('层', '').strip()

        # 处理当前楼层 - 增强文字描述支持
        if current_str.isdigit():
            current_floor = int(current_str)
        elif current_str in ['低楼层', '中楼层', '高楼层', '顶层', '底层']:
            #根据总楼层估算当前楼层
            if total_str.isdigit():
                total_floor_temp = int(total_str)
                # 使用与"共X层"格式相同的估算逻辑
                if current_str == '底层':
                    current_floor = 1
                elif current_str == '顶层':
                    current_floor = total_floor_temp
                elif current_str == '高楼层':
                    current_floor = int(total_floor_temp * 0.8)
                elif current_str == '中楼层':
                    current_floor = int(total_floor_temp * 0.5)
                elif current_str == '低楼层':
                    current_floor = int(total_floor_temp * 0.2)
            else:
                current_floor = np.nan  # 总楼层未知，无法估算
        else:
            # 尝试提取数字
            num_match = re.search(r'(\d+)', current_str)
            current_floor = int(num_match.group(1)) if num_match else np.nan

        # 处理总楼层
        if total_str.isdigit():
            total_floor = int(total_str)
        else:
            total_floor = 0

        return current_floor, total_floor
```

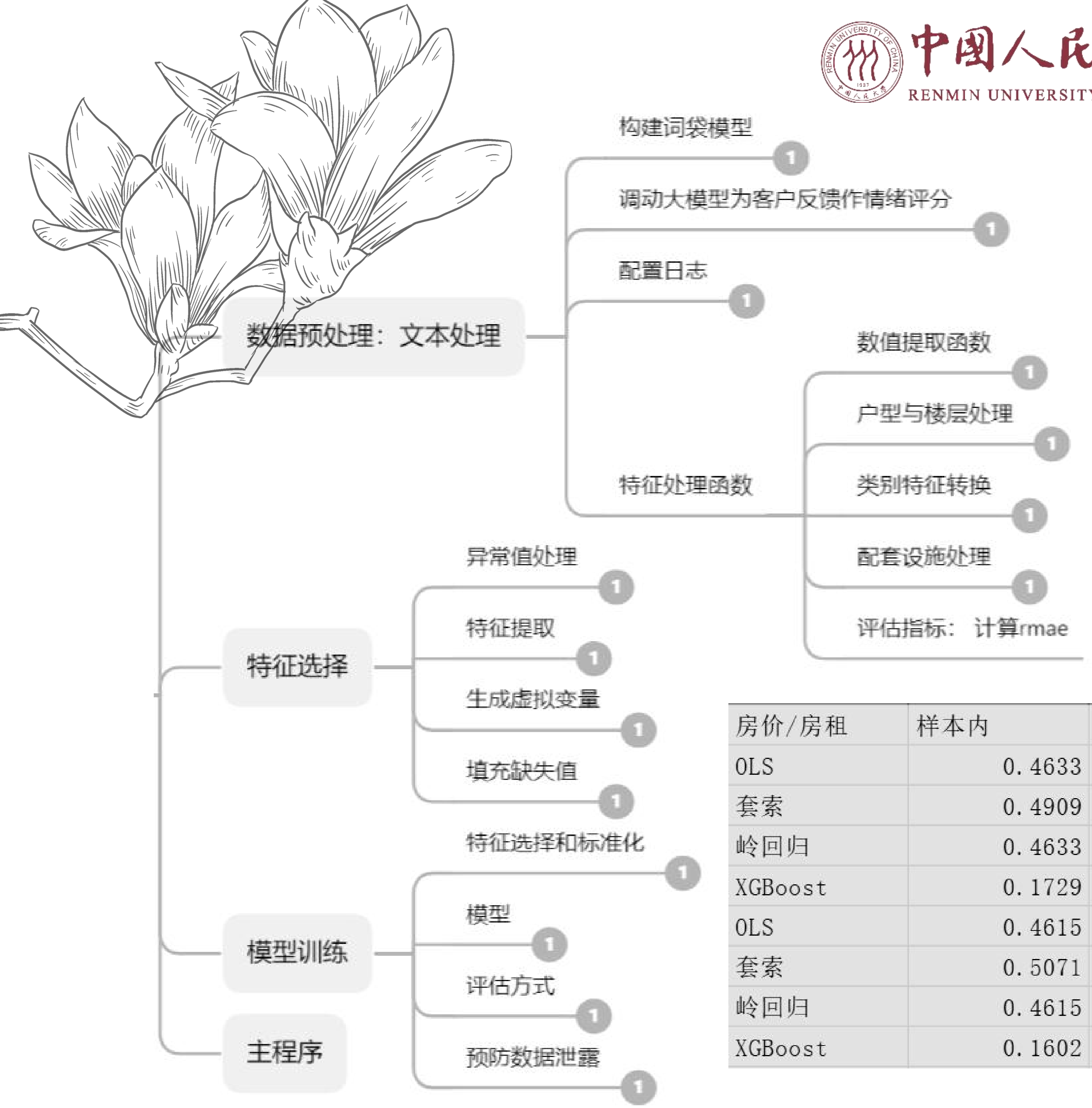
Model	In-sample	Out-of-sample	Cross-validation	Kaggle score
Linear Regression	0.4047	0.3989	0.4037	28.90
	0.4017	0.4036	0.4021	
Lasso	0.4074	0.4011	0.4062	28.22
	0.4065	0.4082	0.4068	
Ridge	0.4047	0.3989	0.4036	28.90
	0.4017	0.4036	0.4021	
Elastic Net	0.4053	0.3993	0.4043	28.56
	0.4032	0.4049	0.4035	
Linear (MAE)	0.4055	0.3993	0.4072	32.14
	0.4007	0.4023	0.3999	
XGBoost	0.0832	0.0978	0.0977	66.29
	0.1135	0.1307	0.1296	

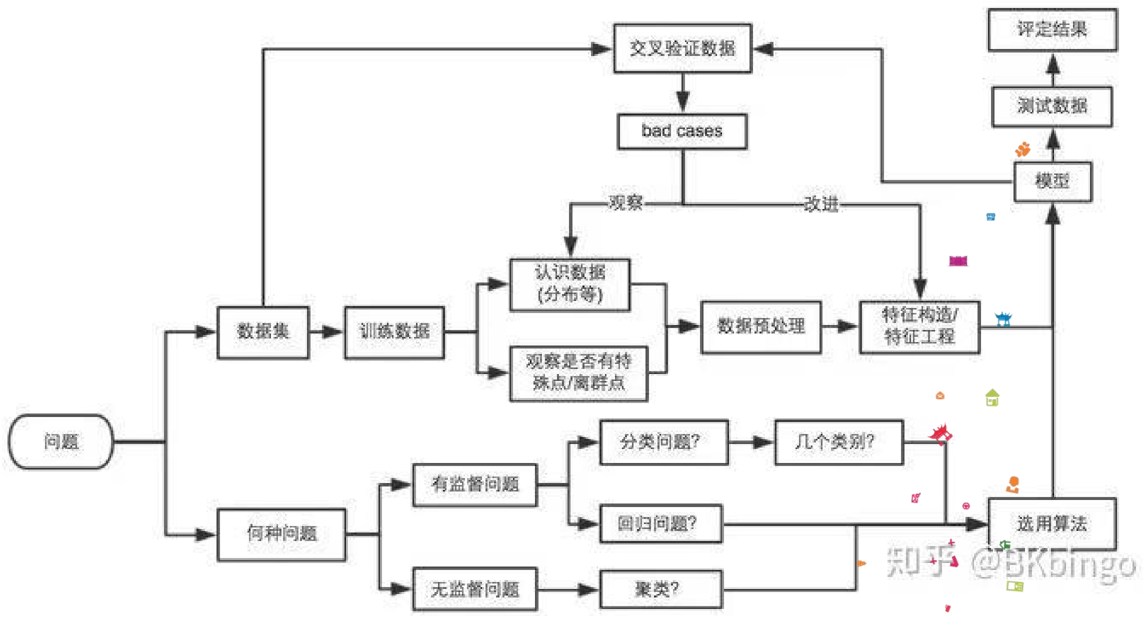


```python
if '楼层' in df.columns:
    floor_data = df['楼层'].apply(extract_floor_comprehensive).tolist()
    df[['当前楼层', '总楼层']] = pd.DataFrame(floor_data, index=df.index)
    df['楼层比例'] = df['当前楼层'] / df['总楼层']
```

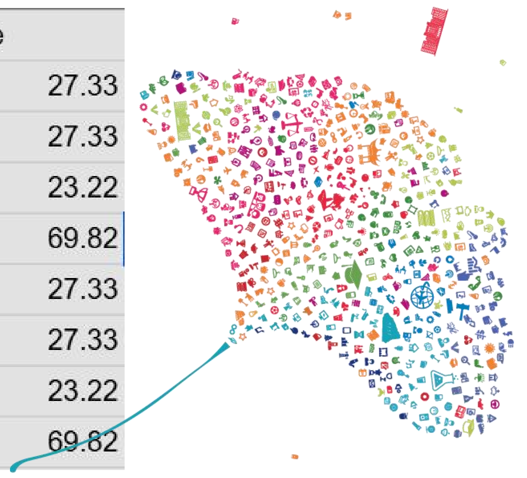
中国人民大學
RENMIN UNIVERSITY OF CHINA

RENMIN UNI





房价/房租	样本内	样本外	6倍交叉验证	kaggle
OLS	0.4633	0.4657	0.4631	27.33
套索	0.4909	0.4895	0.4907	27.33
岭回归	0.4633	0.4657	0.4631	23.22
XGBoost	0.1729	0.1759	0.1767	69.82
OLS	0.4615	0.454	0.4599	27.33
套索	0.5071	0.5063	0.5067	27.33
岭回归	0.4615	0.454	0.4599	23.22
XGBoost	0.1602	0.1629	0.1644	69.82



王泽疆 2023200910