Shanghai Housing Rental Price Prediction

CS 6220 Final Project

Team Members: Liangliang Sun, Weiyu Qiu, Lang Min, Zhi Wang, Zhaoshan Duan

Abstract

This project analyzes a comprehensive dataset of rental housing listings from Lianjia, a major Chinese real estate platform, focusing on properties in Shanghai. Using data mining techniques, we explore the factors that influence rental prices and develop a predictive model that can estimate rental prices based on property features. Our analysis reveals that location (district), property size, number of bedrooms, and the presence of amenities like elevators significantly impact rental prices. Through neural network modeling, we achieve a predictive accuracy of approximately 65.73% with a mean absolute error of ¥2,098.65, demonstrating the feasibility of using machine learning to estimate rental prices in the Shanghai housing market.

Table of Contents

Introduction

1. Statement of the Problem

The Shanghai housing rental market is one of the most dynamic and expensive in China, presenting challenges for renters and property managers in determining fair market prices. This project aims to analyze the key factors that influence rental prices and develop a predictive model that can accurately estimate rental prices based on property characteristics.

2. Importance of the Problem

Understanding rental pricing in Shanghai is crucial for several reasons:

1. For tenants, accurate price estimations help in budgeting and identifying potentially overpriced or underpriced properties
2. For property owners and managers, data-driven pricing strategies maximize occupancy and revenue
3. For real estate platforms, accurate price predictions enhance user experience and facilitate faster transactions
4. For urban planners and policymakers, insights into rental pricing patterns inform housing policy decisions

3. Background Information & Literature Survey

Shanghai, as China's financial and commercial hub, has experienced remarkable growth and urbanization over the past few decades. This rapid development has led to a dynamic and highly stratified housing rental market, where rental prices vary significantly across different districts, property types, and levels of amenities. Central districts such as Huangpu, Jing'an, and Xuhui typically command premium prices due to their proximity to business centers, historical value, and access to public infrastructure, while peripheral areas offer more affordable options but often at the cost of convenience.

Understanding the pricing mechanisms in Shanghai's rental market is complex. Traditional real estate valuation models have relied heavily on linear regression and hedonic pricing models. While these approaches have successfully identified key determinants such as square footage, number of bedrooms, and proximity to transport hubs, they often fall short in capturing nonlinear relationships and interactions among features.

Several academic and empirical studies have investigated these factors:

Li et al. conducted a spatial analysis of rental prices across Shanghai and concluded that intra-urban spatial patterns, especially proximity to central business districts and availability of public services, play a significant role in rental price variation[1].

Qin used a dataset of 20,000 listings and identified that housing size, renovation status, and interior layout had a stronger influence on rent than subway proximity or floor level, challenging the assumptions of earlier regression-based models[2].

Despite these efforts, relatively few studies have fully leveraged modern deep learning architectures, such as neural networks, to analyze and predict rental prices in the Shanghai market. Given the high dimensionality and complexity of urban housing data, machine learning and neural networks provide a powerful alternative to conventional models. By integrating spatial, structural, and semantic features, such approaches can potentially improve the accuracy and interpretability of rental pricing systems in megacities like Shanghai.

Methodology

Our approach involves several sequential steps:

1. Data Collection and Preprocessing

We utilized a dataset containing 29,980 rental listings from Lianjia, a major Chinese real estate platform. The preprocessing steps included:

- Extracting numeric values from text fields (area, floor, price)
- Parsing housing configurations to obtain bedroom, living room, and bathroom counts
- Extracting district information from location data
- Converting facility information into binary feature flags
- Handling missing values and removing outliers

2. Exploratory Data Analysis

We performed a comprehensive EDA to understand:

- Distribution of property sizes, prices, and floor levels
- Relationship between district and price
- Impact of bedrooms, living rooms, and bathrooms on price
- Relationship between amenities (elevator, heating, etc.) and price
- Correlation analysis between numeric features

3. Feature Engineering

We transformed the raw data into model-ready features:

- Numeric features: area, floor, number of bedrooms/living rooms/bathrooms
- Categorical features: district, elevator availability, utilities, orientation, lease type
- Text features: extracted tags with TF-IDF vectorization

4. Model Development

We developed a neural network model with:

- Standardized numerical inputs
- One-hot encoded categorical features
- TF-IDF vectorized text features
- Architecture: two hidden layers with 120 neurons each and dropout regularization
- Mean Absolute Error (MAE) loss function
- RMSprop optimizer with learning rate of 0.0001

5. Model Evaluation

We evaluated the model using:

- Mean Absolute Error (MAE) in RMB
- Percentage accuracy (percentage difference between predicted and actual values)
- Visual comparison of predicted vs. actual prices

Code

This project was implemented in Python using a wide range of libraries for data processing (pandas, numpy), visualization (matplotlib, seaborn), and machine learning (scikit-learn, TensorFlow/Keras). The code can be divided into five major parts:

1. Data Loading and Preprocessing

The dataset containing 29,980 rental listings from Lianjia was loaded using:

```
data = pd.read_csv('lianjia_en.csv')
```

Next, numerical values such as area, floor level, and price were extracted from text fields using regular expressions in the function extract_numeric_values():

```
def extract_numeric_values(row):
    area = re.findall(r'\d+', str(row['Area']))
    row['Area'] = float(area[0]) if area else np.nan
```

Similarly, the number of bedrooms, living rooms, and bathrooms were extracted from the HouseType field using extract_house_components().

The district name was derived from the Location string using string splitting, and a binary encoding (0/1) was applied to facility columns (e.g., washer, air conditioning) using process_facilities().

2. Exploratory Data Analysis

EDA was conducted to understand the distribution and relationships between features. Various plots were created, such as:

- **Histograms** for Area and Price
- **Heatmap** showing average prices by district and bedroom count
- **Scatterplots and Regression lines** for Area vs. Price
- **Boxplots** of rental price distributions across different districts and bedroom counts
- **Bar charts** for facility availability and floor distribution

Additionally, correlation coefficients were computed:

```
correlation_with_price =
numeric_data.corr()["Price"].sort_values(ascending=False)
```

This revealed the degree to which features like area, floor, and number of rooms are correlated with price.

3. Feature Engineering

To prepare the data for modeling:

- **Numeric features** (Area, Floor, etc.) were scaled using StandardScaler:
  `scaler = StandardScaler()`
  `scaled_numeric = scaler.fit_transform(numeric_features)`
- **Categorical features** (e.g., District, Elevator) were one-hot encoded:
  `enc = OneHotEncoder(sparse_output=False)`
  `encoded_categorical = enc.fit_transform(categorical_features)`
- **Textual features** (property tags) were vectorized using TF-IDF:
  `vectorizer = TfidfVectorizer()`
  `encoded_text = vectorizer.fit_transform(processed_text)`

All features were then concatenated into a single input matrix:

```
X_data = np.concatenate((scaled_numeric, encoded_categorical,
encoded_text.toarray()), axis=1)
```

The target variable Price was normalized (divided by 10,000) to improve training stability.

4. Model Design and Training

A neural network regression model was built using the Keras Sequential API. The architecture includes:

```
model = Sequential()
model.add(Dense(120, activation='relu',
kernel_initializer='he_normal', input_shape=(X_train.shape[1],)))
model.add(Dropout(0.2))
model.add(Dense(120, activation='relu',
kernel_initializer='he_normal'))
model.add(Dropout(0.2))
model.add(Dense(1))  # Output: predicted price
```

- **Two hidden layers** with 120 neurons each use the ReLU activation function and he_normal weight initialization.
- **Dropout layers** with 20% dropout rate are applied after each hidden layer to reduce overfitting.
- **Output layer** consists of a single neuron for price prediction.

The model was compiled using the RMSprop optimizer with a low learning rate:

```
optimizer = keras.optimizers.RMSprop(learning_rate=0.0001)
model.compile(loss='mae', optimizer=optimizer, metrics=['mae'])
```

It was trained over 50 epochs with a batch size of 64, using an 80/20 train-test split achieved through stratified sampling:

```
history = model.fit(X_train, y_train,
                    batch_size=64,
                    epochs=50,
                    validation_data=(X_test, y_test),
                    shuffle=True,
```

```
                        verbose=2)
```
The model training process was monitored using the Mean Absolute Error (MAE) metric on both training and validation sets. The learning curves were plotted and saved.

5. Model Evaluation and Prediction

After training, the model was evaluated on the test set:

```
y_pred = model.predict(X_test)
absolute_error = np.mean(np.abs(y_pred - y_test)) * 10000
```

- The final **Mean Absolute Error (MAE)** was printed in RMB.
- **Percentage accuracy** was calculated by comparing prediction error relative to actual price.

Several plots were generated to visualize model performance:

- Actual vs. Predicted Price (line plot)
- Regression plot with scatter distribution
- Bar charts of price distributions across various attributes

Additionally, a few random samples were selected to print their actual vs. predicted prices:

```
print(f"Actual Price: {y_test[i]*10000:.2f} RMB, Predicted Price: {y_pred[i][0]*10000:.2f} RMB")
```

Results

Our analysis uncovered several key insights about the Shanghai rental market:

District Analysis

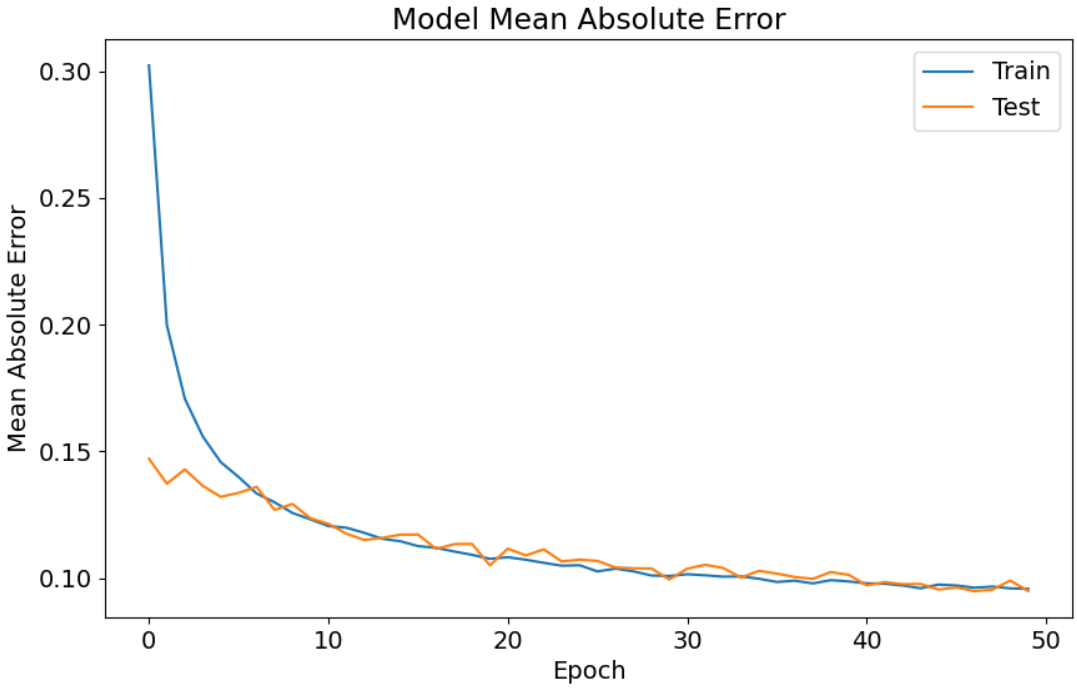
- Huangpu district commands the highest average rent (¥15,670/month), followed by Changning (¥10,228) and Xuhui (¥10,059)
- Outlying districts like Jinshan and Fengxian have the lowest rents (below ¥3,200/month)
- The three most represented districts in the dataset were Minhang, Pudong, and Baoshan, each comprising approximately 10% of listings

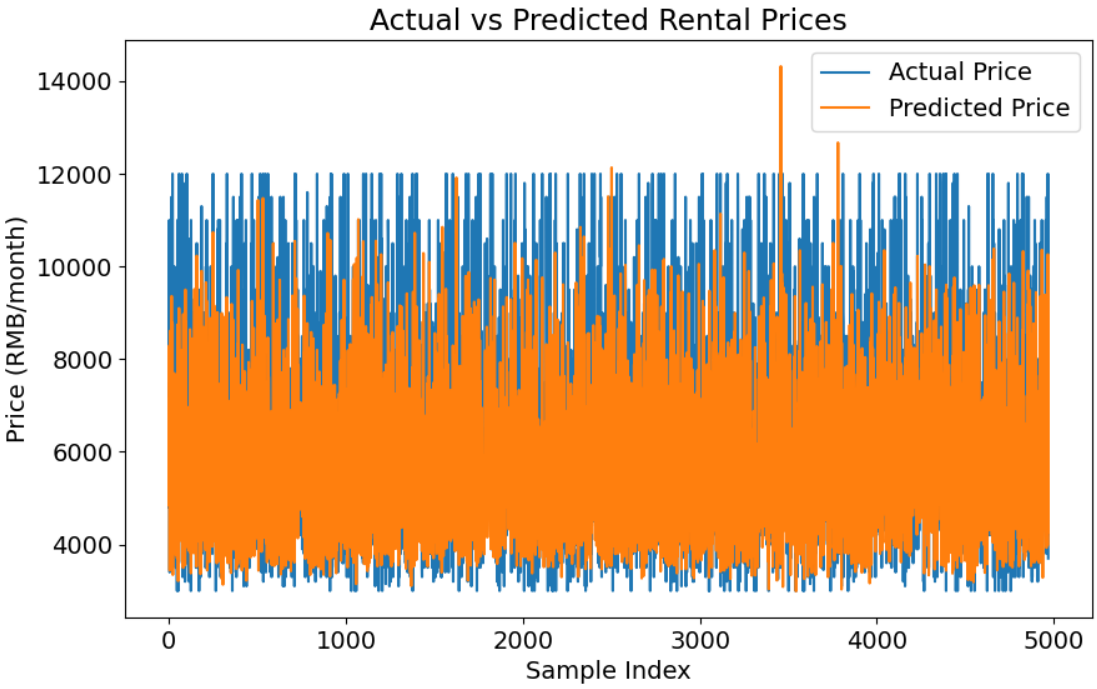
Property Features Impact

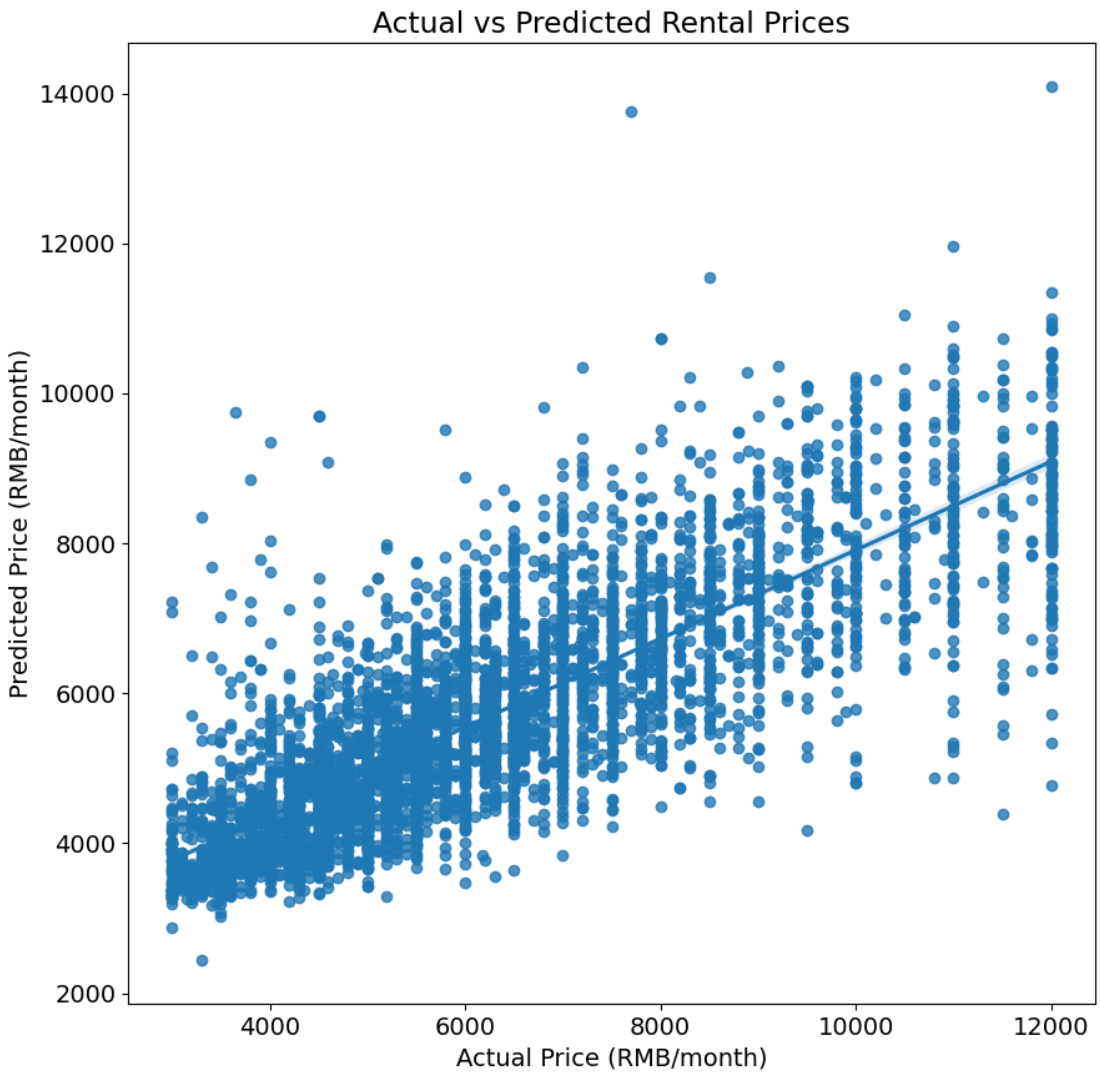
- A strong correlation exists between the number of bedrooms and price (0.438)
- Properties with elevators command an average price premium of ¥3,468/month
- South-facing properties are highly preferred, making up 94.6% of all listings
- Most common configuration is 2-bedroom units (43.4% of listings), followed by 1-bedroom units (29.5%)

Model Performance

- Mean Absolute Error: ¥2,098.65/month
- Model Accuracy: 65.73% (calculated as 100% minus the mean percentage error)
- The model performs better on mid-range properties than on high-end luxury rentals
- Prediction accuracy varies by district, with central districts showing higher prediction errors


Model Mean Absolute Error
Train
Test
Mean Absolute Error
0.30
0.25
0.20
0.15
0.10
0
10
20
30
40
50
Epoch


Actual vs Predicted Rental Prices
Actual Price
Predicted Price
Price (RMB/month)
14000
12000
10000
8000
6000
4000
0
1000
2000
3000
4000
5000
Sample Index



Key Price Determinants

Based on correlation analysis, the most influential features for price prediction are:

1. Number of bathrooms (0.57 correlation)
2. Number of bedrooms (0.44 correlation)
3. Number of living rooms (0.37 correlation)
4. Floor level (0.27 correlation)
5. Area (0.17 correlation)

Discussion

The results reveal several important patterns in Shanghai's rental market:

Spatial Variation

The substantial price differences between districts reflect Shanghai's concentric development pattern, with central districts commanding premium prices due to proximity to business districts, infrastructure, and amenities. This reinforces the real estate adage that "location is the primary determinant of value."

Property Configuration Impact

The strong correlation between bedrooms/bathrooms and price indicates that price scaling is not purely based on area but is significantly influenced by the functional division of space. This suggests that developers may maximize returns by optimizing room configurations rather than just maximizing floor area.

Amenity Premium

The significant price difference between properties with and without elevators (¥3,468/month) highlights the importance of amenities in Shanghai's rental market. This premium is likely influenced by both convenience factors and the correlation between elevator presence and building age/quality.

Model Limitations

While achieving 65.73% accuracy, our model faces challenges in capturing the full complexity of rental pricing. Possible improvements include:

- Incorporating temporal data to account for seasonal and yearly price trends
- Adding neighborhood-level amenity data (schools, metro stations, etc.)
- Implementing more sophisticated text analysis of property descriptions
- Developing separate models for different market segments (luxury vs. affordable)

Future Work

Based on our findings, several directions for future research emerge:

Enhanced Feature Engineering

- Incorporate proximity to key facilities (subway stations, schools, etc.)
- Add building age and construction quality metrics
- Include neighborhood development metrics
- Develop more sophisticated text analysis of property descriptions

Advanced Modeling Approaches

- Test ensemble methods combining multiple model types
- Experiment with separate models for different districts or property types
- Implement time series analysis to track price trends
- Explore transfer learning to adapt models to new districts or cities

Practical Applications

- Develop a rental price recommendation tool for property owners
- Create a fair price calculator for prospective tenants
- Design district investment potential indicators for investors
- Build an anomaly detection system to identify mispriced properties

Conclusion

This project demonstrates the feasibility of using machine learning techniques to predict rental prices in the Shanghai housing market with reasonable accuracy. Our findings confirm the significant impact of location, property size, and amenities on rental prices, while also revealing subtler influences of factors like orientation and floor level.

The neural network model achieves an accuracy of 65.73%, providing a solid foundation for rental price estimation. The mean absolute error of ¥2,098.65 indicates that the model can serve as a useful guide for market participants, though there remains room for improvement through more sophisticated feature engineering and modeling techniques.

The insights generated from this analysis can benefit multiple stakeholders in the rental market ecosystem, from prospective tenants seeking fair prices to property owners optimizing rental strategies and policymakers designing interventions to address housing affordability challenges.

References

[1]Li, H., Wei, Y. D., & Wu, Y. (2019). *Analyzing the private rental housing market in Shanghai with open data*. https://doi.org/10.1016/j.landusepol.2019.04.004
[2]Qin, Z. (2024). *Research on the influencing factors of housing rental prices: Take Shanghai housing rents as an example*. https://doi.org/10.54254/2753-8818/38/20240543

Annex-A

Complete Implementation and Used Data:
https://drive.google.com/file/d/1hLgR53sdjBILbPKbjhGzNNTEzxL6_fyJ/view?usp=drive_link