Rentalytics: Anlayzing Trends in Apartment Rent Pricing

Ankit Singh Chauhan, Hari Shivani Gudi, Kael Ecord,
Sai Aravind Donga

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Introduction

Accurate prediction of apartment rental prices is essential for landlords and tenants alike.

Landlords can optimize investment returns and occupancy rates.

Tenants benefit from informed decisions, cost savings, and suitable housing options.

Efficient prediction enhances decision-making, resource allocation, and quality of life for stakeholders.

Introduction

Previous Studies

- Utilized machine learning, statistical, and NLP techniques
- Considered socioeconomic factors, housing market conditions, and neighborhood amenities

Gap

- many comprehensive approaches
- lack of focus on specific predictive models
- Targeting apartment rental prices

Our Focus

- Investigating regression and classification models
- Incorporating a variety of pertinent characteristics
- Enhance prediction accuracy in apartment rental costs

Dataset

[Apartment for Rent Classified - UCI Machine Learning Repository]

99,492 rows of 22 columns

Final Columns:

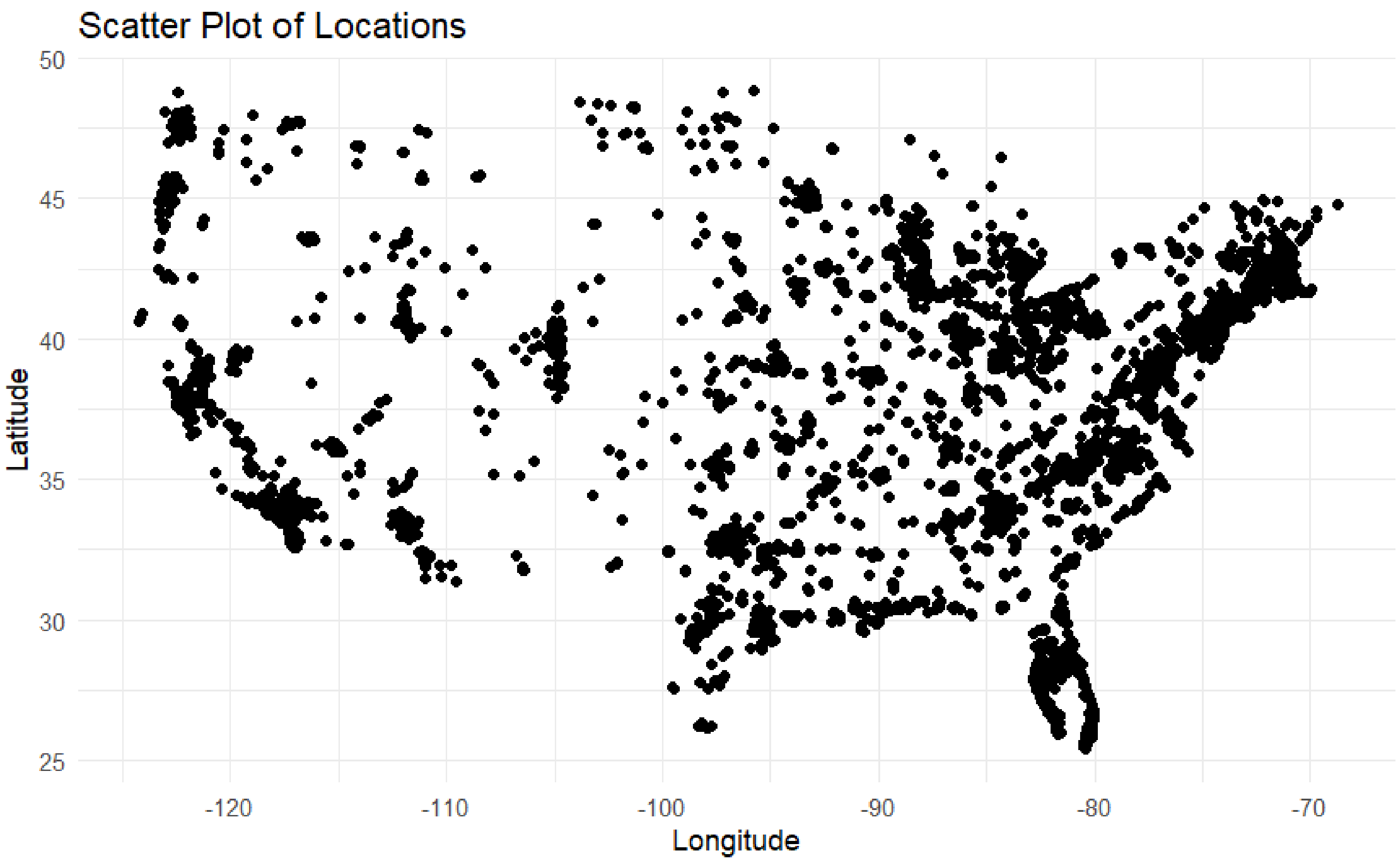
- bathrooms
- bedrooms
- fee
- has_photo
- price
- square_feet
- state
- latitude
- longitude
- studio
- dogs_allowed
- cats_allowed
- us_region
- us_division

Exploratory Data Analysis

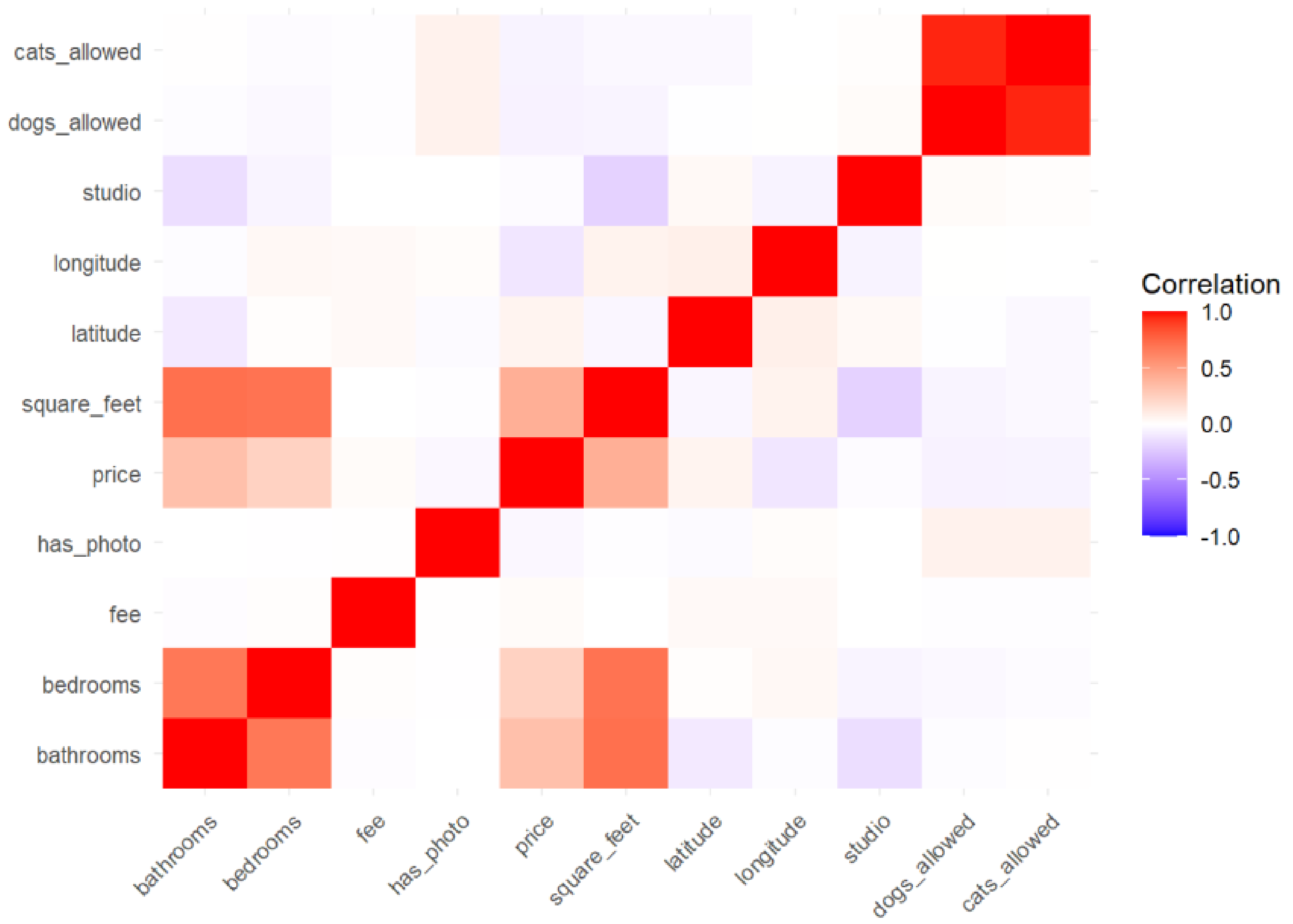
- After pre-processing the dataset:
  - 99,125 observations
  - 14 variables.
- Rental prices show significant regional variation
  - The Northeast has the highest average rent at $1,988
  - followed by the West at $1,851
  - South at $1,336
  - Midwest at $1,109
- Presence of photos slightly impacts the rental price; apartments with photos have an average price of $1,516, compared to $1,618 for those without.
- Properties allowing both dogs and cats tend to have a slightly lower average rent ($1,465) compared to those that allow cats only ($2,057)

Exploratory Data Analysis

- Distribution of properties across different states or regions:



Exploratory Data Analysis



Data Exploration, Cleaning and Preprocessing

- Cleaning
  - Removed all non-apartments
  - Removed apartments that didn't include monthly rent
  - Removed apartments will null values for state
- Preprocessing
  - Created new studio column from deleted title column
  - Broke down pets_allowed to cats_allowed and dogs_allowed columns
  - Created us_region and us_division columns based on state
  - Created new categorical price column
  - After all cleaning and preprocessing the final dataset that will be used for exploration and model building contains 99,125 observations and 14 columns for each observation. That leaves 13 columns for input and the price (monthly rent) as the response variable.
- Exploration
  - 15 columns, 98,944
  - 13 input variables
  - 2 response (price and price_cateogy)

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Simple Linear Regression

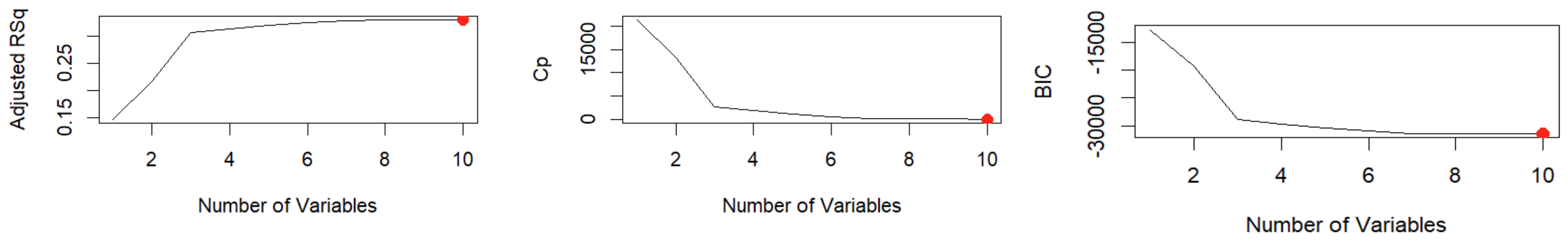
- Final Model:

$$log(price) = \beta_0 + \beta_1(squarefeet) + \epsilon$$

  - Test MSE: 504,417.37

Multiple Linear Regression

- Built model without state, us_division, longitude, and latitude
  - Check VIF values
  - Remove cats_allowed from variables used due to high VIF (>3)
- Apply best subset selection



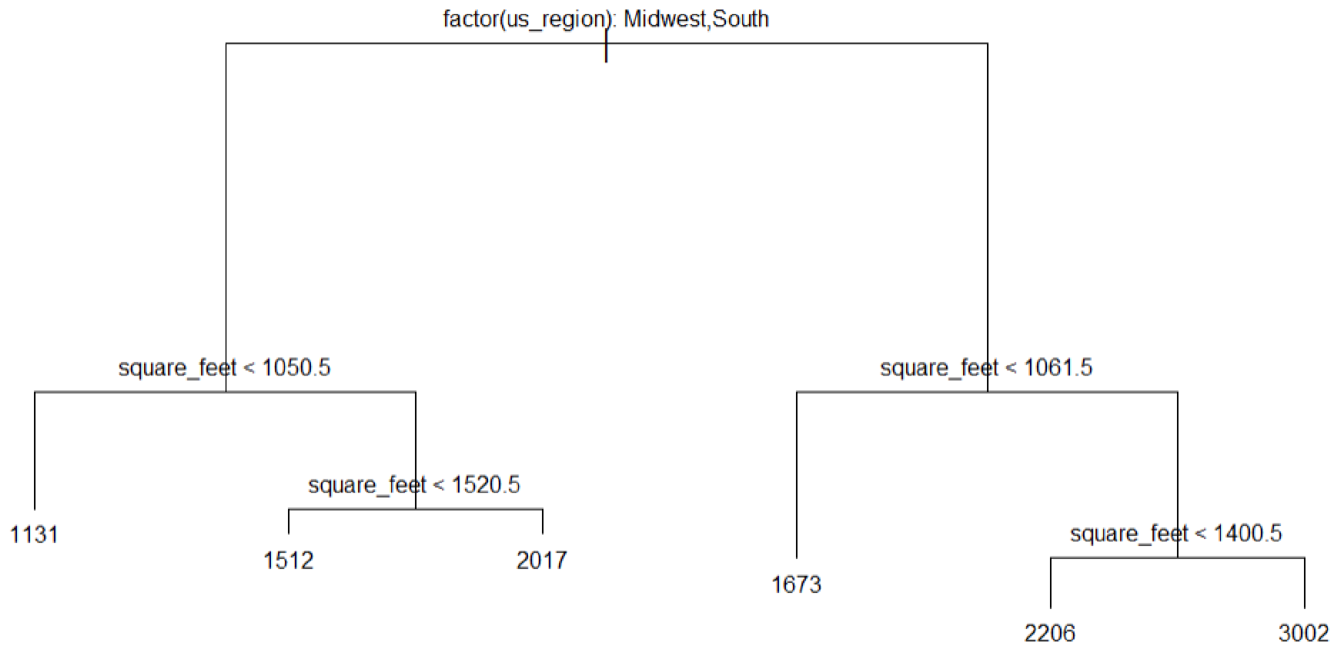
- Choose 7 variable Model
  - Bathrooms, bedrooms, square_feet, studio, all levels of us_region

- Most significant variables
  - Northeast region, West region
    - Being in this region significantly increases price
  - Square_feet

- Test MSE: 376,273.014

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Regression Tree



- Removed variables us_division, state, longitude, and latitude due to singularity issues
- Key variables:
  - Region: Midwest and South have lower priced apartments
  - Square_feet: increasing square footage increases price
- Test MSE: 402919.52
  - Higher than MLR model

Ridge Regression

- All variables included in model
- Trained using the base glmnet function
- Used cv.glmnet to select best lambda value
  - Best lambda value = 29.857
- Retrained used best value for lamda
- Test MSE:
  - 279,035.36
  - Best value for regression models

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Classification

Target Variable: price_category (low, medium, high)

Multinomial Logistic Regression
- Used the multinom() function from the nnet package
- Regressed all features on price_category

QDA (Quadratic Discriminant Analysis)
- Used the qda() function from the MASS package
- Trained on price_category ~ all features

LDA (Linear Discriminant Analysis)
- Used the MASS package, similar to QDA
- Trained on price_category ~ all features

KNN (K-Nearest Neighbors)
- Looped from k = 1 to 10
- For each k, trained the knn() model on the train set
- Identified the best k value with maximum accuracy

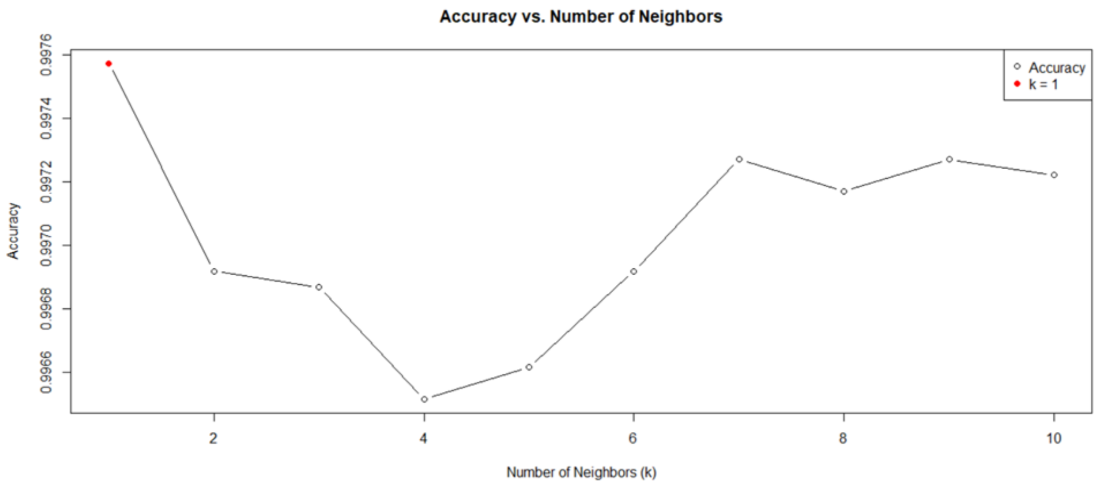
- Generated predictions on the test set using predict()
- Compared predictions to actuals to get accuracies and displayed confusion matrices

Classification- Test Accuracy Results

Model	Test Accuracy
Multinomial Logistic Regression	**0.9998484**
QDA	0.9147001
LDA	0.7682551
KNN(K=1)	**0.9975744**

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Classification- *Test accuracy for k = 1 through k =10*



The model's accuracy decreases with fewer neighbors, indicating insufficient information for accurate predictions. As the number of neighbors increases from 4 to 7, the accuracy improves, indicating better data capture and predictions. After k=7, adding more neighbors doesn't significantly improve accuracy, and may even lead to a slight decrease.

K value	Test Accuracy
1	**0.9975744**
2	0.9969175
3	0.9968669
4	0.9965132
5	0.9966143
6	0.9969175
7	0.9972712
8	0.9971701
9	0.9972712
10	0.9972207

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Classification-Confusion Matrices

```
> print(conf_matrix)
              multi_logistic_pred
actual_values   Low Medium High
       Low     6409      0    0
       Medium     3   6812    0
       High       0      0 6565
> print(conf_matrix_knn)
              knn_pred
actual_values   Low Medium High
       Low     6397     12    0
       Medium    23   6786    6
       High       1      6 6558
> print(confusion_matrix_lda)

actual_values   Low Medium High
       Low     5101   1308    0
       Medium  1362   5391   62
       High       8   1846 4711
> print(confusion_matrix_qda)

actual_values   Low Medium High
       Low     5884    367  158
       Medium    95   6577  143
       High       0    925 5640
```

Discussion

- Limitations
  - No use of NLP techniques
  - Sparse data for certain values of response variables and other key input values

- Final Results
  - Best regression model:
    - Ridge Regression
      - Test MSE = 279,035.36
  - Best classification model:
    - Multinomial Logistic Regression
      - Test Accuracy = 99.98%

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING