ChatGPT

Apartments Rental Analysis

Data Preparation

- Loaded the dataset (~99,492 listings) using a semicolon separator and **cp1252** encoding to avoid character decode errors [1].
- Inspected data schema (22 columns including price, area, amenities). Noted many missing entries (e.g. most *address* and *pets_allowed* fields are null) [2].
- Created unified price fields: e.g. **monthly_price** (convert weekly rents to monthly) and **price_per_sqrt_foot** for fair comparison [3].
- Handled missing/outlier data: dropped records missing key numeric fields for modeling (e.g. NA in latitude/longitude) [4] and filtered geographic outliers (limited to reasonable latitude/longitude range) [5].

Script:
"First we ingested the raw listings data. We needed to fix an encoding issue, so we read the CSV with cp1252 encoding [1]. This gave us about 99,500 entries. We examined the schema: there were 22 fields (IDs, title, price, square feet, etc.), but many fields like *address* or *pets_allowed* were largely empty [2]. In cleaning, we focused on the relevant columns. For consistency, we converted all prices to a **monthly price** (multiplying weekly rents by 4) and computed **price per square foot** [3]. We then filtered out incomplete records where needed – for example, clustering would only use listings with non-null latitude/longitude [4]. We also removed geographic outliers (we only kept listings within realistic US lat-long bounds) [5]. These steps ensured our data was clean and ready for analysis."

Exploratory Analysis (3 Key Questions)

- **Q1:** *Which cities have the highest average rent per sq.ft?* Method: grouped listings by city, averaged `price_per_sqrt_foot`, and plotted the top 10 cities (bar chart) [6]. Key finding: a few cities (e.g. New York, San Francisco, etc.) stand out with the highest rent per square foot.
- **Q2:** *How are pet policies distributed?* Method: created boolean flags for allowing dogs or cats from the text field, then counted categories (cats only, dogs only, both, none) and plotted a pie chart. Key finding: The largest segment of listings **do not allow pets** at all, with smaller slices allowing only cats, only dogs, or both.
- **Q3:** *Does apartment size affect rent?* Method: binned apartments by size (square feet) and computed the average monthly rent per size bin (line chart) [7] [8]. Key finding: There is a clear upward trend – larger apartments tend to have higher average rents.

Script:
"For our exploratory phase, we focused on three business questions. First, we asked **which cities command the highest rent per square foot**. We grouped the data by city and computed the mean *price per square foot*, then visualized the top 10 cities in a bar chart [6]. The result showed that major urban centers (for example New York, San Francisco, etc.) rank at the top. This bar chart (see slide) clearly highlights the most expensive markets.

Next, we looked at **pet allowance** policies. We derived two features: `allows_dogs` and `allows_cats` from the text description. We then counted how many listings fall into each category: allows only cats, only dogs, both, or neither. The pie chart of these counts (not shown here) indicated that **the majority of apartments do not allow pets** at all, while smaller percentages allow only one type or both. This insight helps understand tenant preferences for pet-friendly units.

Finally, we examined **rent vs. apartment size**. To do this, we created size bins (e.g. 0–1000 sq.ft, 1000–2000 sq.ft, etc.) and calculated the average rent for each bin [7]. We plotted these averages on a line chart [8]. The chart shows a steady increase: larger apartments (in square feet) generally have higher average rent. This confirms the expected positive relationship between apartment size and rental price."

Feature Engineering & Clustering

- Constructed numeric feature set for clustering: selected **monthly_price**, **square_feet**, **latitude**, and **longitude**, then scaled them using Standard Scaler [4].
- Determined cluster count with K-Means: evaluated **silhouette scores** and **elbow plots** for K=2..10 [9] [10]. Both methods suggested an "elbow" or peak around 6 clusters.
- Fitted final model: K=6 clusters (random_state fixed for reproducibility [5]).
- Key insight: The resulting clusters largely reflect **geographic groupings** of apartments. (See map plot of listings colored by cluster; for example, one cluster covers the Northeast, another the West Coast, etc.)

Script:
"We also engineered features to segment listings spatially and by price. We picked the numeric predictors: apartment size, rent, and location coordinates. We scaled all features to standardize units [4]. Then we used the K-Means clustering algorithm to group similar listings. To choose the number of clusters, we ran two diagnostics: a silhouette score plot and an inertia ("elbow") plot [9] [10]. Both analyses pointed to **6 clusters** as a good balance. We finally fit K-Means with K=6 [5].

The cluster map (slide) shows our results: each point is a listing, colored by its cluster. We observe that clusters correspond to regions – for example, one cluster covers apartments in the Northeast, another covers West Coast listings, etc. This geographic clustering confirms that location is a major factor in the data. These clusters can help us understand market segments (e.g., a cluster of high-price apartments in major cities vs. another cluster of more affordable units in smaller markets).

Supervised Learning Models

- **Target:** Predict rent price **category** (Low/Medium/High) from listing features.
- **Features:** Used `square_feet`, `bedrooms`, `bathrooms` as inputs (dropped records with missing values) [11]. Formed `price_category` by dividing monthly rent into 3 quantiles (Low, Medium, High).
- **Train/Test split:** 80/20 split with `random_state=55` for reproducibility [12].
- **Models:** Chosen models were **K-Nearest Neighbors** and a **Neural Network (MLP)**, as examples of distance-based and nonlinear classifiers [13].
- **Hyperparameter tuning:** For KNN, performed 10-fold cross-validation over odd K from 1 to 49. The best K found was 45 [14]. The ANN was configured with two hidden layers (100+100 neurons) and

logistic activation. No extensive grid search was shown, but random_state=55 fixed its randomness [15].

- **Evaluation metrics:** Used accuracy and classification reports (precision/recall/F1 for each category).

Script:
"With the data prepared, we built two supervised classifiers to predict the rent **category** (Low/Med/High). The features used were simple listing attributes: apartment size, bedroom count, and bathroom count [11]. We generated the target by splitting rent into three quantiles. We split the data into train/test sets (80/20, fixed random seed for consistency) [12].

We implemented two models. First, **K-Nearest Neighbors (KNN)**. We used a cross-validation loop to find the best neighbor count: testing k=1,3,...,49, and found **K=45** gave the highest average accuracy [14]. We retrained KNN with k=45 on the training set. Second, an **Artificial Neural Network (MLP)** classifier: we set up an MLP with two hidden layers of 100 neurons each, logistic activation, and a fixed random seed [15]. This model has more flexibility but also requires careful tuning.

We evaluated both models on the test set. The key metric was **accuracy**, supplemented by precision/recall in the classification report. For KNN, the test accuracy was about 0.49 [16]. For the ANN, accuracy was about 0.45 [17]. The detailed reports (precision/recall by class) showed that KNN had more balanced recall across categories, whereas the ANN achieved higher recall on the "Low" category but at the expense of the others [16] [17]. These metrics guided our model comparison."

Model Comparison & Recommendation

- **Results:** KNN achieved **~49% accuracy**, ANN about **45%** on the test set [18].
- KNN outperformed ANN in overall accuracy and more balanced class metrics [18] [16].
- **Recommendation:** The KNN model is the better choice given these results. It provides the highest accuracy and handles all categories reasonably well.
- **Conclusion:** Our analysis provided business insights (pricing differences by city, pet policy breakdown, rent-size trends) and built a predictive model (KNN) to classify rental price bands. Future work could explore other models or additional features to improve accuracy.

Script:
"In summary, we compared the two models' performance. The KNN classifier achieved about **0.49 accuracy**, whereas the ANN reached around **0.45** [18]. Thus, KNN was the better performer. In the classification reports, KNN showed more consistent F1-scores across Low/Medium/High categories [16], while the ANN's performance was lower overall [17]. Based on this, we recommend the **KNN model** for predicting rent categories, since it yields the highest accuracy and balanced performance.

In conclusion, our data preparation and exploratory analysis have yielded several actionable insights (e.g., city-level pricing, pet allowance stats, size–rent relationships). The clustering identified geographic segments of listings. On the predictive side, although the models are not extremely high in accuracy (around 50%), the KNN model provides a reasonable classification of rental price bands. Further improvements might come from richer features or more advanced models, but given time constraints we selected KNN as our current best model. That completes our report. Thank you."

Sources: Our findings and figures are drawn from the analysis notebook code and outputs (cited above) [6] [18].

---

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] DS_Project.ipynb
file://file-ELz39FeBKW6a8bPDu3ks5E