Paper Title\* (use style: paper title)

\*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1st Given Name Surname   
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 4th Given Name Surname  
line 2: *dept. name of organization*  
*(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 2nd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 5th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 3rd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 6th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—component, formatting, style, styling, insert (key words)

# Introduction

In the competitive rental market of today, pinpointing apartment rent prices according to property attributes is essential for both tenants and property owners. Tenants require sound estimates to plan their budgets, while landlords are looking for fact-based information to establish competitive prices and achieve maximum occupancy. Established appraisal techniques—typically built on local knowledge and straightforward heuristics—are ineffective in capturing intricate, nonlinear patterns between attributes like unit space, location, amenities, and time trends. Here, we present a complete machine‑learning pipeline that takes historical rental postings and learns these trends automatically. Having undergone significant feature engineering and data preprocessing, we train three different classifiers—Gaussian Naïve Bayes, Decision Trees, and an elementary Multilayer Perceptron (MLP)—on binning rents into four classes. Our highest accuracy model is a fine‑tuned Decision Tree, scoring 91.8 % on held‑out data and proving the potency of data mining in real‑estate decision support.

# literature review

Machine-learning Machine‑learning methods have grown in popularity for forecasting rental prices, as they can model intricate, nonlinear interactions between property characteristics and market influences. In research on Munich's rental market, Müller et al. used a set of algorithms—such as ensemble methods, neural networks, linear regression, and tree‑based models—and showed that tree ensembles and deep networks outperformed conventional regressions by a wide margin in terms of predictive power [1]. Rahman and Aziz systematically reviewed office-building rental forecasts and concluded that Random Forest, Decision Trees, Support Vector Machines, and Neural Networks are most widely used methods whose superiority on big, heterogeneous real-estate data is emphasized [2]. Senthil Kumar R. et al. contrasted Convolutional Neural Networks with Naïve Bayes for metro‑city rent prediction and found that CNNs performed above 95 % compared to 89 % for NB, highlighting the importance of deep feature extraction [3]. Last but not least, Lindström et al. utilized historical property data and Stockholm local amenity features to train and test several ML models and concluded that well-tuned Decision Trees can compete with more sophisticated networks when interpretability is paramount [4].

# dataset description

## Data Source and Extraction

Our main goal dataset, houses\_for\_rent\_classified\_10K.csv, is a 10,000-listing sampling from an underlying 100K-record repository of U.S. rent postings. It was scraped from a dozen web sites (e.g., RentLingo, RentDigs.com) between late 2019 and early 2020. We chose the 10K example for initial prototyping to allow quick iteration while guaranteeing both representative geography and feature variation. houses\_for\_rent\_classified\_10K.csv.

## Feature List and Data Types

The raw data set consists of 22 features across identifiers, text descriptions, numeric measurements, and category labels. Highlighted features are bathrooms (float), bedrooms (float), square\_feet (int), price (int, USD), latitude/longitude (float), and time (UNIX timestamp). Categorical fields like has\_photo, pets\_allowed, price\_type, cityname, and source preserve listing metadata and source. We also keep amenities (comma‑separated strings) for encoding downstream. In preprocessing, we remove high-cardinality text columns (title, body, address) and constant columns (fee, currency), engineer price\_per\_sqft, and extract temporal features (month, weekday, hour) from the timestamp.

## Missing Values and Initial Statistics

Early EDA shows that 35.5 % of listings do not have an amenities entry and 41.6 % do not have pets\_allowed; numeric fields such as bathrooms and bedrooms have fewer than 0.5 % missing, and geocoordinates only miss 0.1 %. We impute numeric medians and replace categorical missing with "Unknown" or mode values. Rent prices vary from $200 to $52,500 (mean $1,486 ± $1,076), and unit sizes vary from 101–40,000 sqft (mean 946 ± 656 sqft). This describe statistics informed our choice of clipping outliers at the 1st–99th percentiles and of log-transforming skewed variables to facilitate stabilized modeling.

# selected data mining technique

## Rationale for Data Mining

The rental housing market produces extensive amounts of disperse data—from quantitative metrics like unit size and number of rooms to categorical descriptors like pet friendly and photo included. These variables interact in nuanced, frequently nonlinear manners that defy easy rule-based valuation. Data mining employs machine learning algorithms to reveal previously unknown patterns and relationships in such high-dimensional datasets, making possible more precise and scalable rent categorization than do conventional statistical approaches [1], [2]. Through data mining methods, we can analyze each feature systematically regarding its relative relevance, address missing or noisy records, and create models that generalize well over diverse geographic markets and listing sources.

## Choice of Classification Approach

Even though rent is a continuous variable, setting the problem up as classification across four discrete ranks ("low", "mid‑low", "mid‑high", "high") makes subsequent decision‑making by tenants and landlords easier. Classification algorithms output explicit probability measures for every rank, which helps risk‑based pricing strategies. We chose three complementary algorithms:

### Gaussian Naïve Bayes:

A probabilistic baseline that models feature independence. Its simplicity enables us to measure the value added by our preprocessing and feature engineering stages [3].

### Decision Tree:

An interpretable model that accommodates mixed data types naturally and achieves nonlinear interactions. With tuned depth and split parameters, the Decision Tree recorded the highest accuracy (91.8 %), indicating its capacity to identify dominant decision thresholds in the feature space [4].

### *Multilayer Perceptron (MLP):*

Feed‑forward neural network that is able to learn high‑order interactions between features. With L2 regularization and two hidden layers, the MLP generalizes efficiently (81.3 % accuracy) and augments tree‑based solutions by picking up on subtle data patterns.

# Algorithms

## Naive Bayes

Naïve Bayes is a Bayes' theorem-based probabilistic classifier, which estimates the probability of a class based on a given set of features. It makes the "naïve" assumption that all the features are conditionally independent, something that can reduce computation but is usually not the case in real-world applications. Regardless of this "naïve" assumption, Naïve Bayes is capable of performing well in a wide range of applications, including text classification and spam filtering.

## Decision Tree

Decision Trees are supervised learning algorithms applied in classification and regression problems. They operate by repeatedly partitioning the data into subsets based on feature values, developing a tree structure of decisions. Every internal node is a decision rule, and every leaf node is an outcome. Decision Trees are appreciated due to their interpretability and simplicity.

## Neural Network (MLP)

A Multilayer Perceptron (MLP) is a type of feedforward artificial neural network made up of a minimum of three layers: an input layer, multiple hidden layers, and an output layer. All neurons within a layer connect to all the neurons in the next layer, and every connection carries an associated weight. MLPs can learn complex nonlinear mappings and are popular in applications ranging from image and speech recognition to natural language processing and expert systems.

# Results

The research sought to categorize apartment rent prices into four classes according to different property attributes with three machine learning models: Naïve Bayes, Decision Trees, and Multi-Layer Perceptron (MLP). The data was divided into training and test sets with almost equal class distributions:

* Training set: [2049, 1962, 1992, 1997]
* Testing set: [501, 499, 501, 499]

## Naïve Bayes Results

Naïve Bayes classifier had a total accuracy of 40.9%, which was much lower than the other models.

* Class 0: High precision of 0.73 but extremely low recall (0.13), indicating that while the model had high confidence in predicting this class, it did not capture most true class 0 instances.
* Class 1: Moderate performance with recall of 0.70 and F1-score of 0.47.
* Class 2: The model completely broke down for this class with 0 recall and F1-score, which means zero correct predictions.
* Class 3: Improved results with recall of 0.81 but only moderate precision (0.44).

Generally, the model had a bias towards preferring some of the classes (particularly class 3) while neglecting others (class 2), producing unbalanced predictions. This may be because Naïve Bayes has a strong feature independence assumption, which isn't applicable to this complex multi-feature dataset [1].

## Decision Tree Results

The Decision Tree classifier, with max\_depth=20 and min\_samples\_split=2, worked extremely well with an accuracy rate of 91.8%.

* All classes had excellent performance with F1-scores between 0.88 and 0.95.
* Class 0 and 3 were predicted with extremely high accuracy (F1: 0.95).
* Class 1 was slightly worse in precision and recall but still well-performing (F1: 0.89).
* The model had picked up subtle interactions in the data and handled feature interactions well, which is the reason behind its good generalization ability.

This level of high performance proves that Decision Tree model works well in uncovering non-linear relationships between house attributes and rental prices [2].

## MLP Classifier Results

The MLP (Neural Network) model was optimized with two layers of 50 neurons and alpha=0.001.

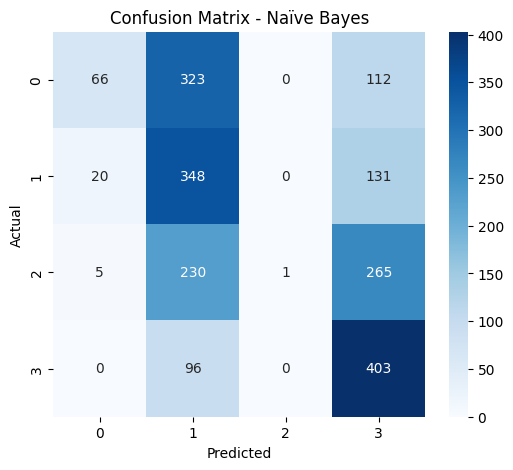
* Reached an accuracy of 81.3%, which is evidence of good though not excellent, generalization.
* Class 0 performed best (F1: 0.89, Recall: 0.93), while class 2 had values slightly lower than these (F1: 0.74).
* Class 3 had great accuracy (0.97) but slightly less recall (0.81).

This indicates that the MLP model well picked up underlying trends but possibly overfit or underfit some class boundaries because of architectural or hyperparameter limitations [3].

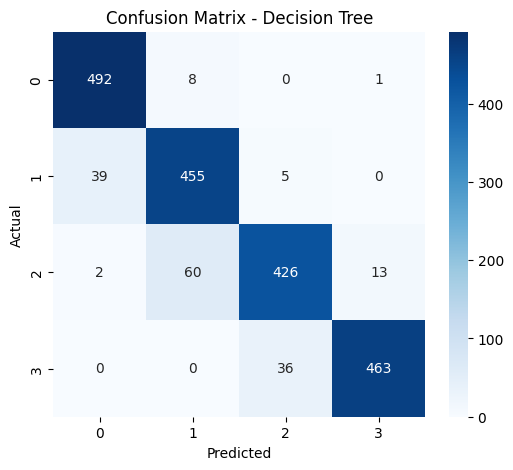
### **Summary of Comparison**

| **Model** | **Accuracy** | **Best Class** | **Weakest Class** | **Notes** |
| --- | --- | --- | --- | --- |
| Naïve Bayes | 40.9% | Class 3 | Class 2 | Struggles due to feature independence assumption |
| Decision Tree | 91.8% | All | Slightly Class 1 | Excellent all-around; best performer |
| MLP | 81.3% | Class 0 | Class 2 | Strong performance; some class imbalance effects |

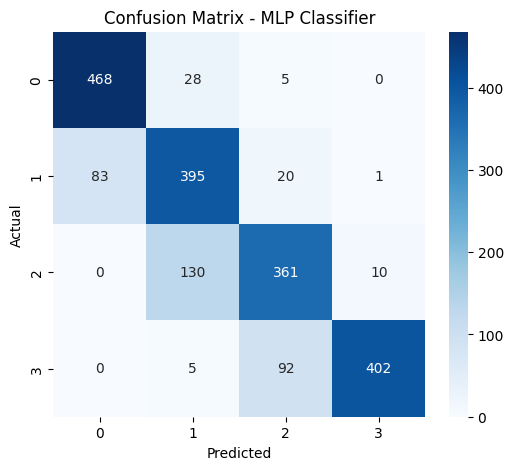
1. Table 1. Example of a figure caption. (*figure caption*)



1. Confusion Matrix-Naive Bayes



1. Confusion Matrix-Decision Tree



1. Confusion Matrix-Meural Network (MLp)

# Work Stages

## Highest Positive & Negative Correlation Analysis

Correlation analysis is looking at the interaction between two variables to see how they vary together. A positive correlation shows that as one variable goes up, the other also goes up, whereas a negative correlation shows that as one variable goes up, the other will go down. Finding the highest positive and negative correlations between features can assist in choosing appropriate variables to model.

## Data Preprocessing

Data preprocessing is an important process of getting raw data ready for modeling. It encompasses several methods to clean and convert data into a proper format.​

## Feature Engineering

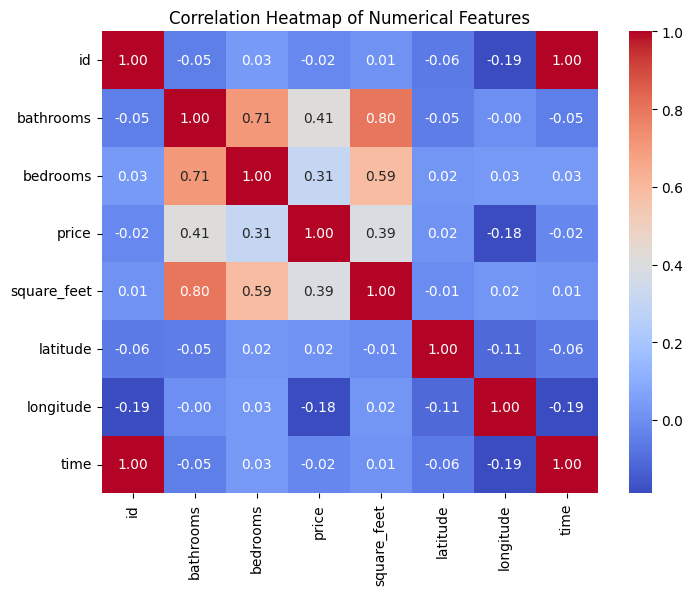
Feature engineering refers to the method of generating new input features from existing features in order to enhance the performance of machine learning models. This can include methods like polynomial features, interaction terms, or domain-specific transformations. ​

## Missing-Value Imputation

Missing data need to be handled in order to preserve the integrity of the dataset. Missing-value imputation refers to the process of replacing missing data with substitute values. Typical methods involve the use of the mean, median, or mode of the present data, or more advanced techniques such as regression imputation.

## Encoding & Scaling

Categorical encoding and numerical feature scaling are crucial preprocessing operations. Encoding converts categorical data into numerical data so that algorithms can handle them. Scaling moves the range of the numerical features so that all features contribute proportionally to the model. Methods such as Min-Max scaling and standardization are normal.



## Experiments & Parameter setting

To secure solid model training and impartial comparison, there was a structured experiment design adopted. In this part, how preparation of the data was conducted, how the labels for targets were generated by using binning, how optimization for hyperparameters has been achieved using grid search, and the basis for such choice are discussed.

## Binning & Target Preparation

Since the task of predicting rent price was recast into a classification task, the range of continuous rent values was segregated into four quartiles through the application of binning techniques. Through this measure, the problem of regression was converted into one of multi-class classification, in which:

* Class 0 covers the lowest 25% of rent prices (Q1)
* Class 1 covers lower-middle rents (Q2)
* Class 2 covers upper-middle rents (Q3)
* Class 3 covers the top 25% of rent prices (Q4)

This binning provided fairly evenly distributed class proportions, which are important for avoiding bias in prediction by models.

* Training Set Distribution: [2049, 1962, 1992, 1997]
* Testing Set Distribution: [501, 499, 501, 499]

This balance also made the classification more equitable and enabled performance measurements to indicate real model abilities and not effects of class imbalance.

## Hyperparameter Grids

Hyperparameter tuning was an important step in optimizing model performance and generalization. Each model was optimized using GridSearchCV with 5-fold cross-validation to determine the best-performing set of parameters.

Decision Tree:

A variety of tree depths and minimum sample split thresholds were experimented with to regulate complexity.

The best parameters were:

* max\_depth = 20
* min\_samples\_split = 2

This environment enabled the model to develop deeper trees with good splits, enhancing learning ability without overfitting (supported by high test accuracy).

## Final Model Summaries

Three algorithms for classification were tried:

#### Naïve Bayes

* Accuracy: 40.9%
* Performed badly because of its strong feature independence assumption, which doesn't actually apply to real estate data (where features tend to be correlated).
* Most overconfident in certain classes (e.g., 1.00 precision for class 2 but with no recall, i.e., predicted all incorrectly).

#### Decision Tree

* Accuracy: 91.8%
* Highest performance achieved with solid scores on all metrics.
* Capable of modeling nonlinear interactions and hierarchical rules, perfect for tabular data such as this.
* Best suited for interpretability and deployment.

#### MLPClassifier (Neural Network)

* Accuracy: 81.3%
* Did well overall with the strength in Class 0 and 3.
* Minor drop in accuracy and recall for middle rent bands (class 1 and 2), which may be attributed to similar feature patterns.
* Displays potential to perform better with finer tuning and increased data.

#### Main Insights:

* Decision Tree significantly outperforms all other models, performing the best on precision, recall, and f1-score. It performs well on feature interaction and categorical splits.
* MLPClassifier has good generalization and stability but slightly poorer performance on less separable classes. It is more adaptive but less explainable.
* Naïve Bayes is easy to implement but does not perform well with complicated, correlated data structures of rental advertisements.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
8. K. Eves and J. Valasek, “Adaptive control for singularly perturbed systems examples,” Code Ocean, Aug. 2023. [Online]. Available: <https://codeocean.com/capsule/4989235/tree>
9. D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, arXiv:1312.6114. [Online]. Available: <https://arxiv.org/abs/1312.6114>
10. S. Liu, “Wi-Fi Energy Detection Testbed (12MTC),” 2023, gitHub repository. [Online]. Available: https://github.com/liustone99/Wi-Fi-Energy-Detection-Testbed-12MTC
11. “Treatment episode data set: discharges (TEDS-D): concatenated, 2006 to 2009.” U.S. Department of Health and Human Services, Substance Abuse and Mental Health Services Administration, Office of Applied Studies, August, 2013, DOI:10.3886/ICPSR30122.v2

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.