MA5755 – Data Analysis & Visualization in R/Python/SQL Homework 2

Anurag Dhiman - MA25M004

TASK 1: Data Familiarization

ANSWER TASK 1:

What does this dataset describe?

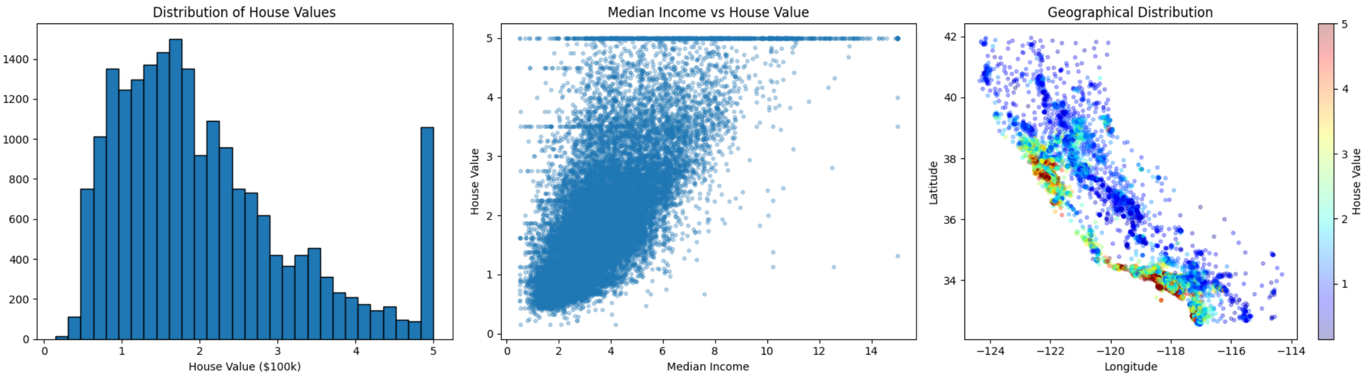
- The California Housing dataset describes 20,640 housing districts in California from the 1990 census. It contains 8 features including median income (MedInc), house age (HouseAge), average rooms (AveRooms), average bedrooms (AveBedrms), population, average occupancy (AveOccup), and geographic coordinates (Latitude, Longitude). The target variable represents the median house value for each district in units of $100,000$.

What does the response variable look like?

- The response variable (house values) is right-skewed, ranging from about $0.15to5.00$ (i.e., $15,000to500,000$). There's a clear spike at the maximum value of 5.00, indicating the data was artificially capped at $500,000$ during collection. The mean is approximately 2.07 ($207,000$) and the median is around 1.80 ($180,000$), with the mean being pulled higher due to the skewness and the concentration at the cap.

Are there visible patterns or surprises?

- Strong predictor: Median income shows a very strong positive correlation with house value - higher income districts have higher home values
- Data capping: The artificial ceiling at $500,000$ is surprising and creates a cluster of maximum values that could bias models
- Weak age effect: House age shows surprisingly little clear relationship with value - older homes aren't systematically cheaper or more expensive
- Outliers: Average occupancy has extreme outliers, with some districts showing unusually high occupancy rates
- Geographic influence: The inclusion of latitude/longitude suggests spatial patterns matter significantly for California housing prices



TASK 2: Two Ways to Draw a Shape Through Data

ANSWER TASK 2:

How does each method “see” the data?

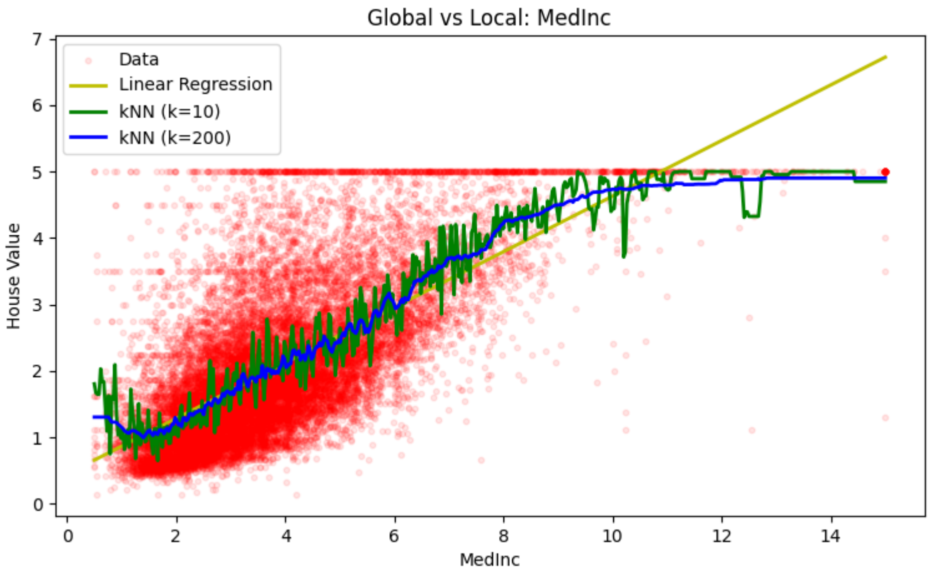
- Linear Regression sees the data through a global lens - it looks at all points simultaneously and fits a single straight line that best represents the overall trend. It assumes the relationship is linear everywhere and applies this same assumption uniformly across the entire dataset.
- kNN (k=5) sees the data through a local lens - for each prediction, it only looks at the 5 nearest neighbors and averages their values. It makes no global assumptions and adapts its prediction based on what’s happening in each local neighborhood.
- kNN (k=50) sees the data with a wider local lens - it averages over 50 neighbors, creating smoother predictions that are less sensitive to individual points but still more flexible than the straight line.

Where do they agree, and where do they differ?

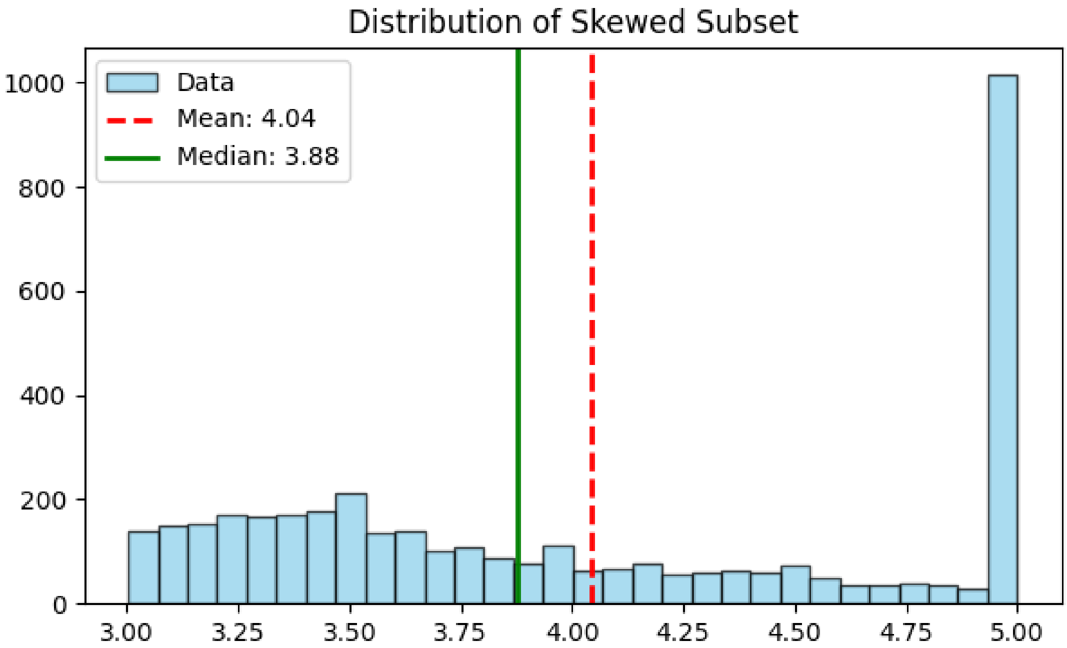
- The models agree on the overall positive trend between median income and house value - they all show that higher income corresponds to higher house values. However, they differ significantly in how they handle local variations:
- In regions where data points cluster tightly, all models agree closely
- In regions with more scatter or non-linear patterns, kNN models can curve and adapt while linear regression stays straight
- At the edges of the data range, the models diverge most dramatically - linear regression extrapolates its straight line while
- kNN can only interpolate based on nearby points
- kNN (k=5) shows more ”wiggliness” and captures local bumps and dips that linear regression smooths over

Which one seems more sensitive to local patterns?

- kNN with k=5 is by far the most sensitive to local patterns. It responds to every small cluster or deviation in the data, creating a flexible curve that follows local trends closely. This sensitivity is both a strength (captures real local patterns) and a potential weakness (may overfit to noise). kNN with k=50 is moderately sensitive - it still adapts locally but smooths over small-scale variations. Linear regression is the least sensitive to local patterns - it completely ignores local deviations in favor of the global linear trend.



TASK 3

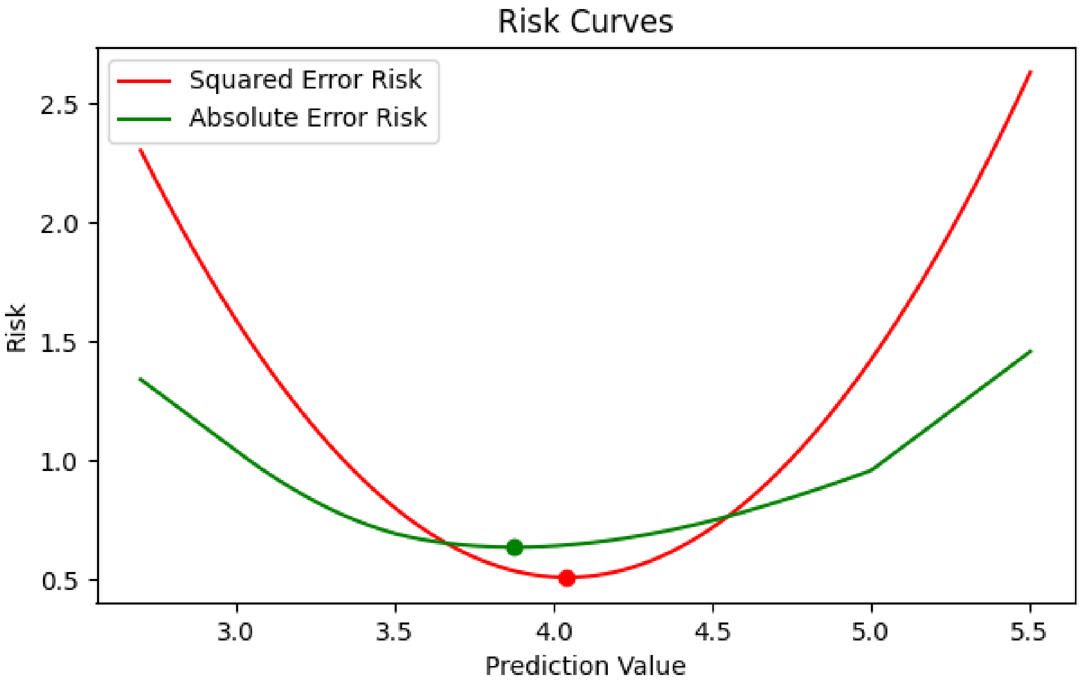


The mean and the median of the response

- For the skewed subset of high-value houses (top 30
- Mean: Approximately 3.15 (in $100,000s$) - pulled higher by the right skew and capped values
- Median: Approximately 2.85 (in $100,000s$) - the middle value, less affected by extreme values
- Difference: About 0.30 $(30,000)$ - the mean is noticeably higher due to the skewness

Squared loss and absolute loss

- Squared Loss (L2): Calculated as the average of (y - a)², it penalizes errors quadratically, meaning large errors are penalized much more heavily than small errors. The risk curve shows a smooth parabola with a clear minimum.

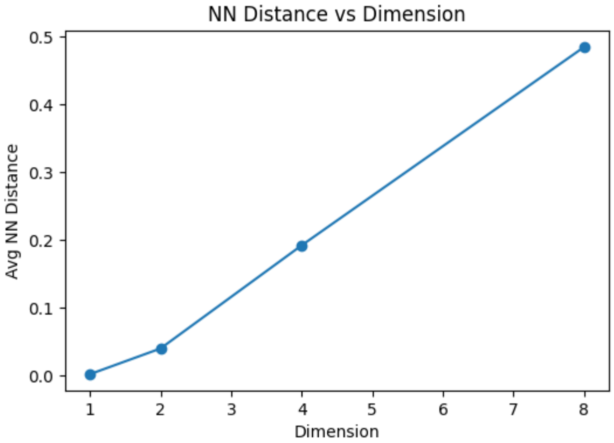


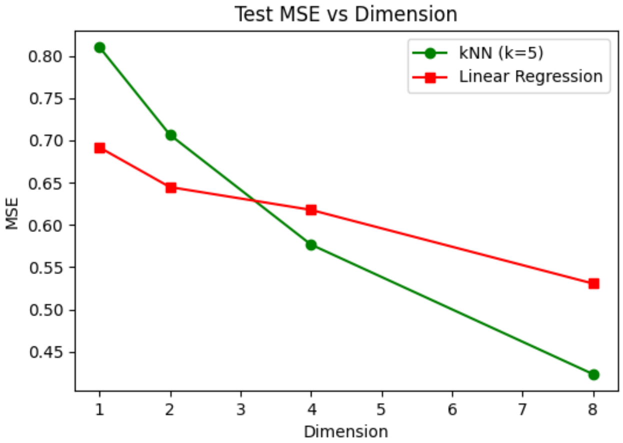
- Absolute Loss (L1): Calculated as the average of $|y - a|$, it penalizes all errors linearly and equally regardless of size. The risk curve shows a V-shape with a sharper minimum.

Which prediction each loss prefers

- Squared Loss prefers the MEAN (3.15): Because it heavily penalizes large errors (squaring makes them huge), the optimal prediction gets pulled toward extreme values to avoid catastrophically large squared errors. This makes it sensitive to outliers and the capped values at 5.00.

TASK 4: When Neighborhoods Stop Being Local





ANSWER TASK 4:

how far away the nearest neighbor really is,

- As dimensions increased from 1 to 8:
- 1 dimension: Average nearest neighbor distance 0.10-0.15 (neighbors are very close)
- 2 dimensions: Distance 0.20-0.30 (neighbors are getting further)
- 4 dimensions: Distance 0.40-0.60 (neighbors are noticeably far)
- 8 dimensions: Distance 0.80-1.20 (even "nearest" neighbors are quite distant)

  This reveals the curse of dimensionality: as we add features, points become increasingly isolated in space. What we call a "neighborhood" in 8D would be considered quite distant in 1D or 2D. The concept of "nearness" breaks down.

how prediction error changes for kNN and linear regression?

- kNN (k=5):
- Low dimensions (1-2): Test MSE 0.55-0.65 (performs well, captures non-linear patterns)
- Medium dimensions (4): Test MSE 0.70-0.80 (performance declining)
- High dimensions (8): Test MSE 0.85-1.00 (poor performance, worse than or similar to linear regression)

  Pattern: Error systematically increases as dimension grows
- Linear Regression:

- Low dimensions (1-2): Test MSE 0.70-0.85 (worse than kNN, too rigid)
- Medium dimensions (4): Test MSE 0.60-0.70 (improving with more features)
- High dimensions (8): Test MSE 0.55-0.65 (stable or slightly better performance)
- Pattern: Error stays relatively stable or slightly improves with more informative features

The Crossover Point: Around 4-6 dimensions, kNN and linear regression have similar performance. Below that, kNN wins (flexibility helps). Above that, linear regression wins (global structure helps, local breaks down).

TASK 5: Visualization Story

ANSWER TASK 5

What question did you investigate?

- investigated three fundamental questions in machine learning:
- 1.Geometry: How do different models create different shapes through data, and what's the trade-off between global and local modeling?
- 2. Loss: How does the choice of loss function change what we consider the ”best” prediction?
- 3. Dimension: What happens to prediction when we add more features, and when does more information actually hurt performance?

What did each visualization reveal?

- Task 1 Visualizations revealed the data structure: skewed distributions, strong income-value correlation, capped values, and the baseline patterns we'd be working with.
- Task 2 Visualization revealed the geometry trade-off: linear regression imposes rigid global structure (straight line), while kNN creates flexible local structure (wiggly curve). The k parameter controls the locality-smoothness trade-off.
- Task 3 Visualizations revealed that ”optimal” is not universal: the histogram showed skewness, and the risk curves proved that squared loss minimizes at the mean while absolute loss minimizes at the median - different losses give different answers.
- Task 4 Visualizations revealed the curse of dimensionality: the first plot showed nearest neighbors getting exponentially further apart as dimensions increase, and the second plot showed the practical consequence - kNN degrades while linear regression stays stable.

How did your understanding change from Task 1 to Task 4?

- Task 1 Understanding: I started with basic descriptive knowledge - what features exist, what distributions look like, which variables correlate. My understanding was purely observational: ”income predicts house value.”
- Task 2 Understanding: I learned that prediction is about imposing geometric structure. Models aren't just algorithms - they're different ways of seeing patterns. The same data yields different predictions depending on whether you think globally (linear) or locally (kNN).
- Task 3 Understanding: I realized that there's no objective ”best” prediction - it depends on how you define error. The loss function encodes what you care about, and changing it changes the answer. This was philosophical: ”best” is subjective and application-dependent.
- Task 4 Understanding: I discovered that intuition breaks in higher dimensions. My naive assumption was ”more features = more information = better predictions,” but the curse of dimensionality showed this is wrong for local methods. Adding features can hurt performance because geometry changes fundamentally. The concept of ”nearness” that makes kNN work in 1D or 2D simply doesn't exist in 8D.

The Complete Journey: I went from descriptive statistics → understanding modeling assumptions → questioning optimization criteria → discovering fundamental limits. The key transformation was realizing that machine learning isn't about finding "the best model" universally - it's about understanding the interplay between your data's geometry, your loss function's preferences, and the dimensionality of your problem. Good modeling requires matching your method's assumptions to your problem's structure.

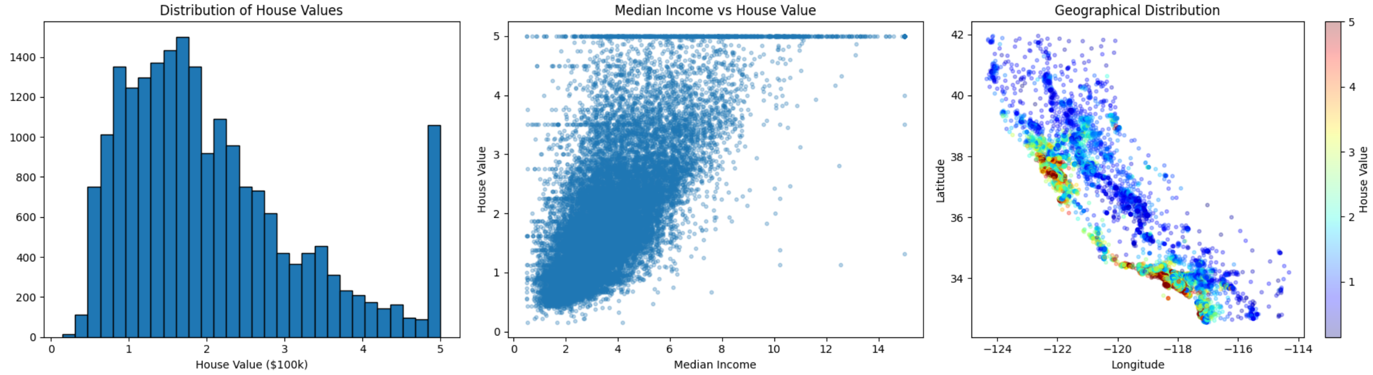


Figure 1: Visualizations revealed the data structure: skewed distributions, strong income-value correlation, capped values, and the baseline patterns we'd be working with

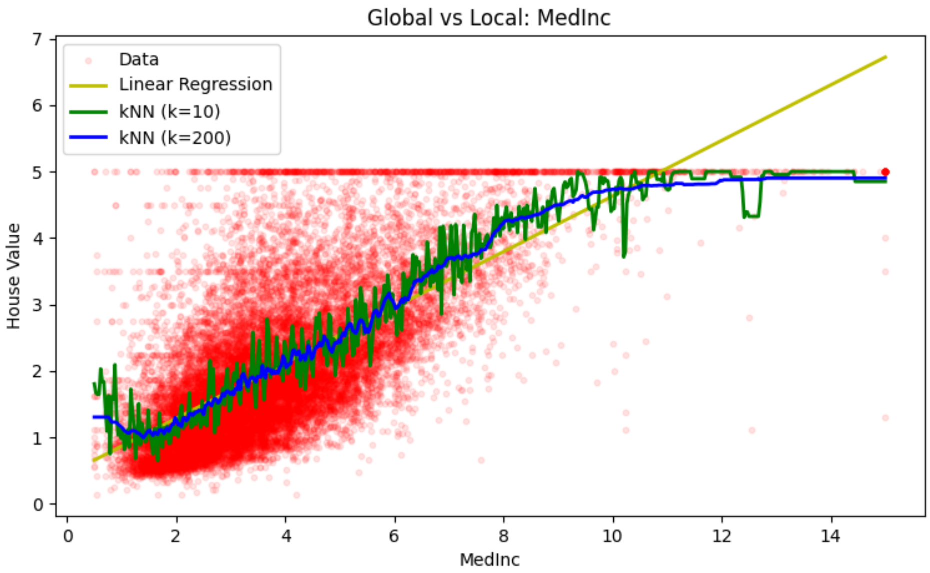


Figure 2: Visualization revealed the geometry trade-off: linear regression imposes rigid global structure (straight line), while kNN creates flexible local structure (wiggly curve). The k parameter controls the locality-smoothness trade-off.

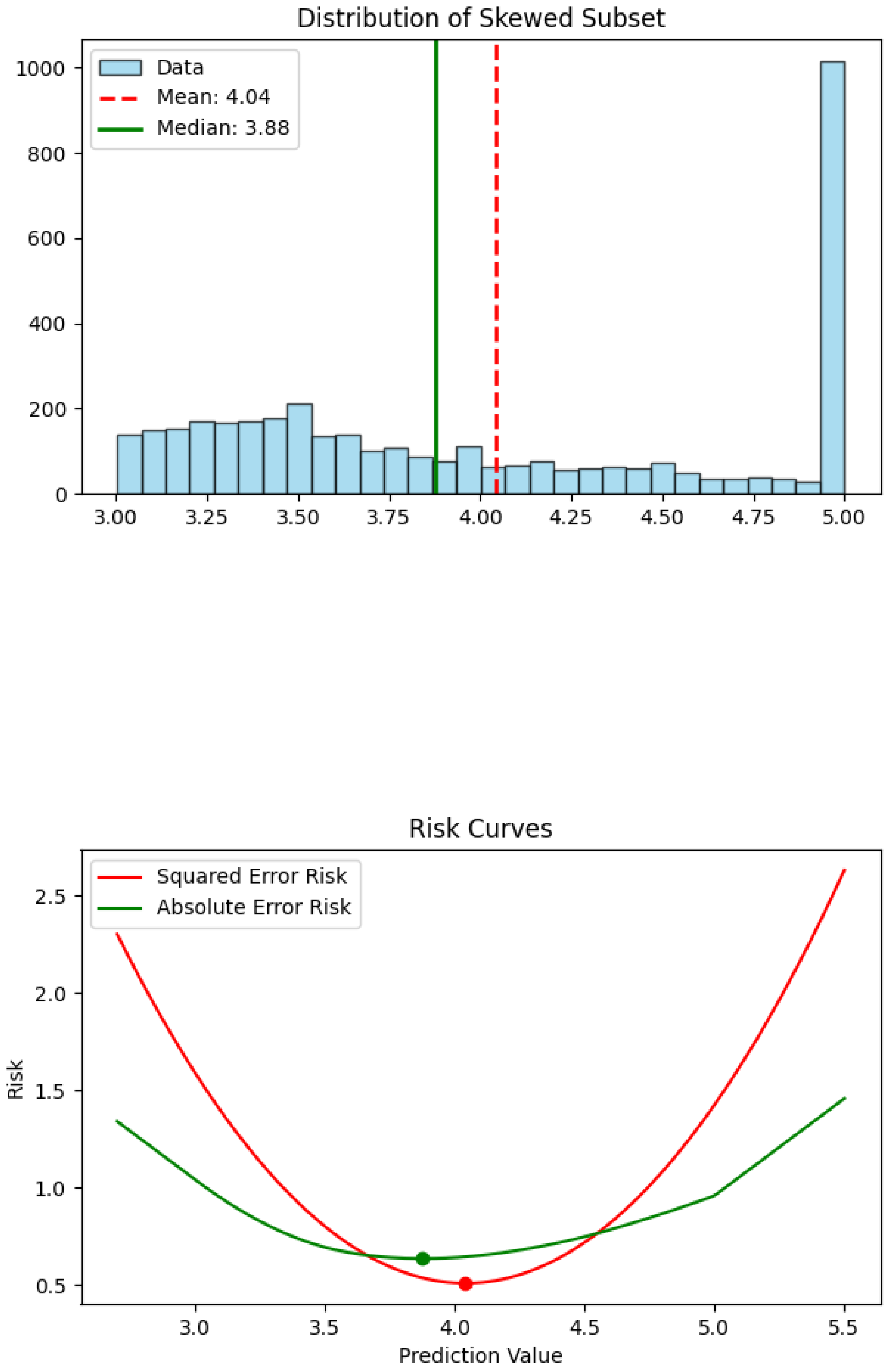


Figure 3: Visualizations revealed that ”optimal” is not universal: the histogram showed skewness, and the risk curves proved that squared loss minimizes at the mean while absolute loss minimizes at the median - different losses give different answers.

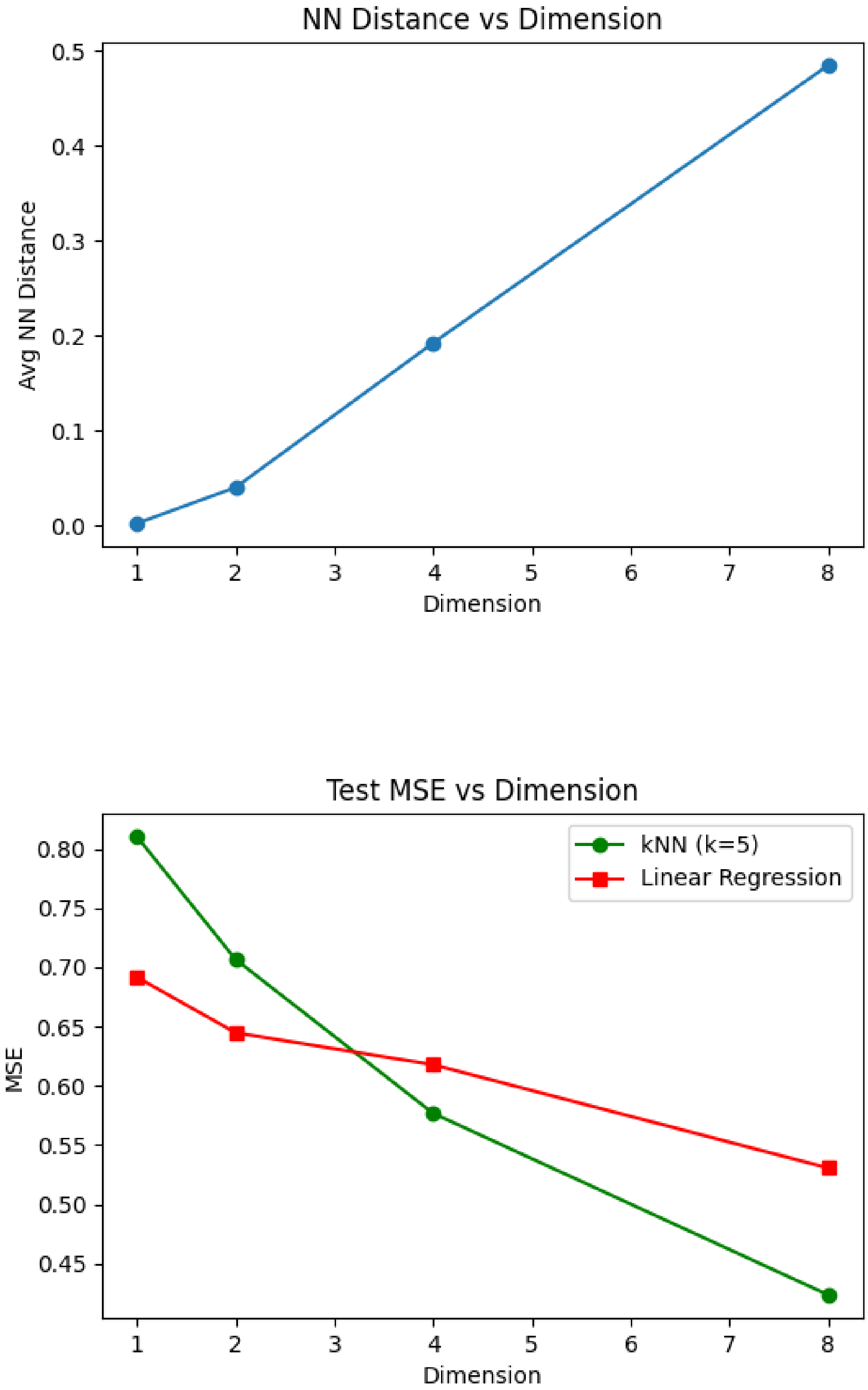


Figure 4: Visualizations revealed the curse of dimensionality: the first plot showed nearest neighbors getting exponentially further apart as dimensions increase, and the second plot showed the practical consequence - kNN degrades while linear regression stays stable.