Practical Part – Ex1 IML

Submitted by: Ori Sass

ID: 206789182

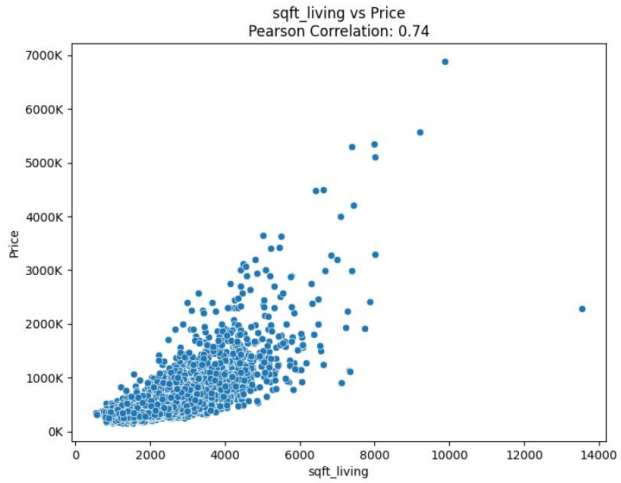
2.1 Fitting A Linear Regression Model:

3) Pre-processing:

- I checked that if a house was renovated, the renovation year is at the same year or later of the year the property was built.
- The lot is bigger than the house.
- Removed Rows:
  Numeric rows that after coercing to number format where nan
- Removed columns:
  - sqft_living15, sqft_lot15, sqft_above, sqft_basement –
    too similar to sqft_living, sqft_lot that is taken in consideration
  - id, date, lat, long, condition, view– small pearson correlation relatively (sub section 4)

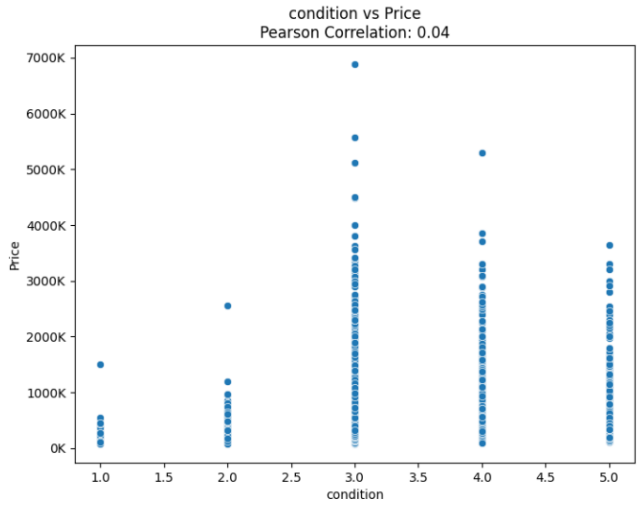
4) Feature evaluation:

A feature beneficial to the model:



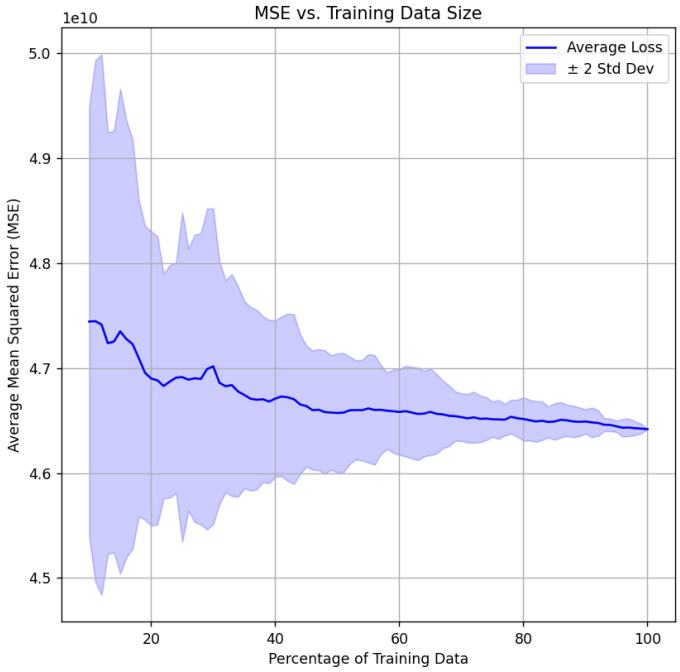
It seems with a high correlation (compared to other features), so i think it will help improving the model's predictions (the higher the sqft, the higher the price).

A feature non-beneficial to the model: condition



We can't rely on this plot to help get better predictions because of the low correlation; it may be irrelevant for price prediction.

6) Fitting a linear regression model:



1. Trend in Loss (Average MSE)
Overall decrease: As the percentage of training data increases (left to right), the average MSE (blue line) gradually decreases.

Why: More training data helps the model learn better general patterns, reducing error on unseen (validation/test) data.

Diminishing returns: The rate of improvement slows down—there's a noticeable drop early on, but by ~60–100%, the curve starts to flatten, meaning adding more data improves performance less dramatically.

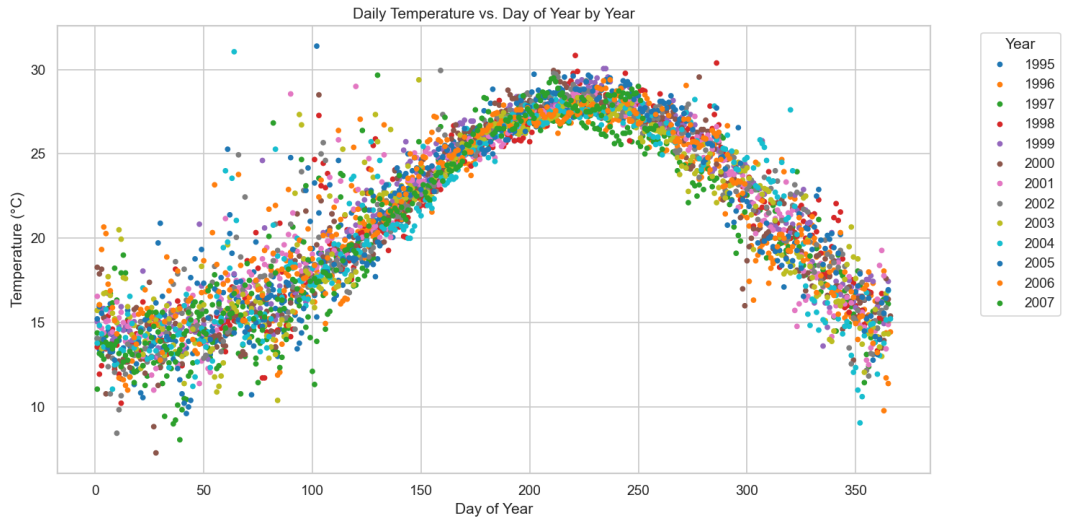
2. Trend in Confidence Interval (±2 Std Dev)
Wide at small training sizes: The shaded region is very large on the left side (small training sizes), indicating high variance in model performance across different training samples.

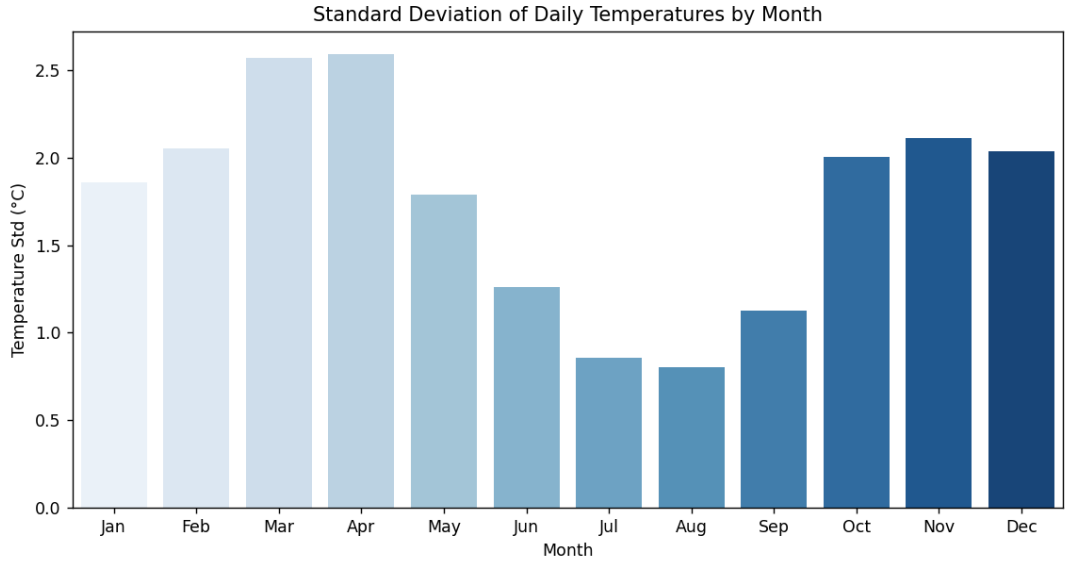
Shrinks with more data: As training size increases, the confidence interval narrows, meaning the model's performance becomes more stable and consistent.

Interpretation: With more data, the model is less sensitive to fluctuations in the specific training sample, which reduces uncertainty.

2.2 Polynomial Fitting

3) Israel daily avg temperatures a function of DayOfYear:

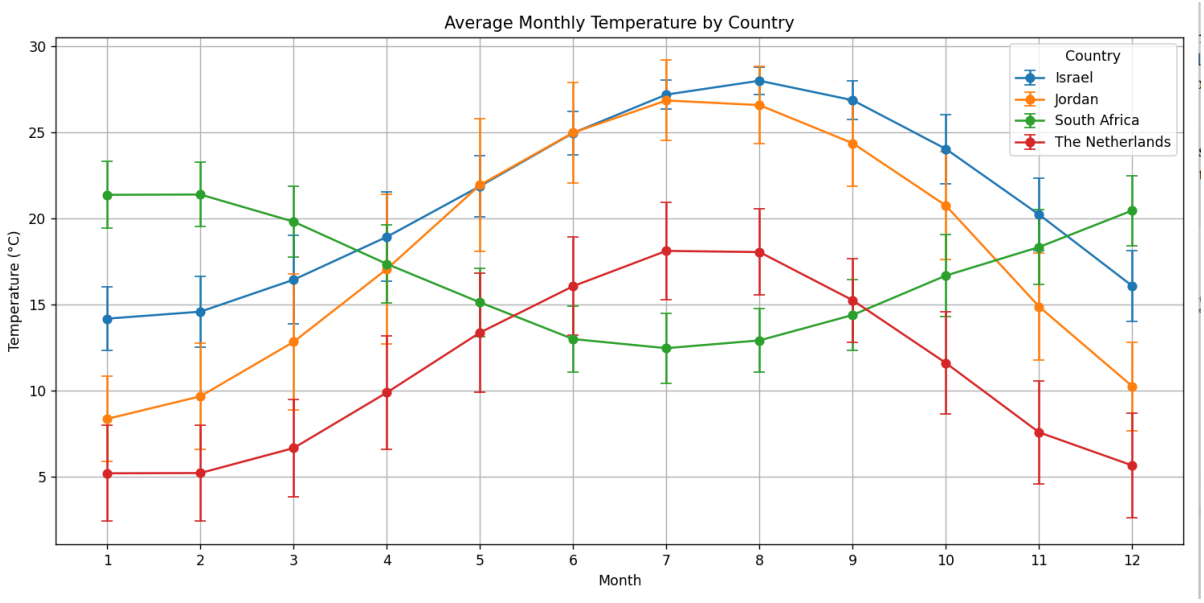




Observation- The best fit is day 200of the year, through all the years in the dataset. Regarding the first plot, a polynomial degree of 3 or more might be suitable for the data, because it looks like a smooth sinus wave. Of course we should try a couple of degrees and see who has the minimum MSE while considering overfitting to say for sure.

As for the second plot, and based on it, i expect a model to predict the temperature better at June-September than the rest of the year.
We can expect that because of the low standard deviation in those months over the years.

4) Monthly temperatures by Country

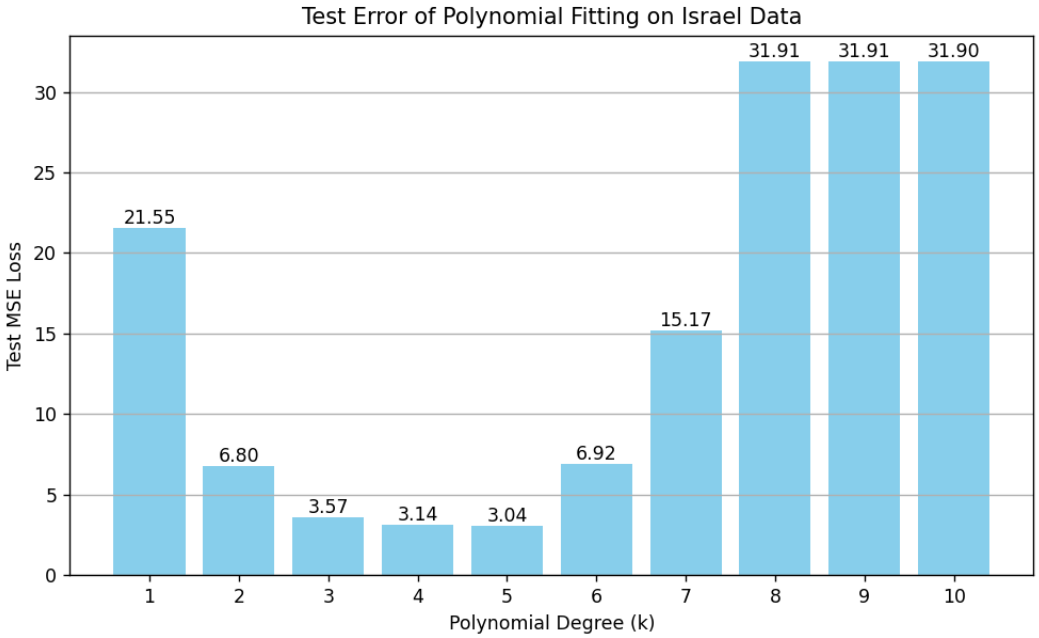


The countries don't share a similar pattern.
Most of the countries share a similar shape of distribution, except South Africa.

The model for Israel is likely to work for: Jordan over The Netherlands and South Africa, because Jordan's temperature values are the closest, even though The Netherlands have the most similar distribution.

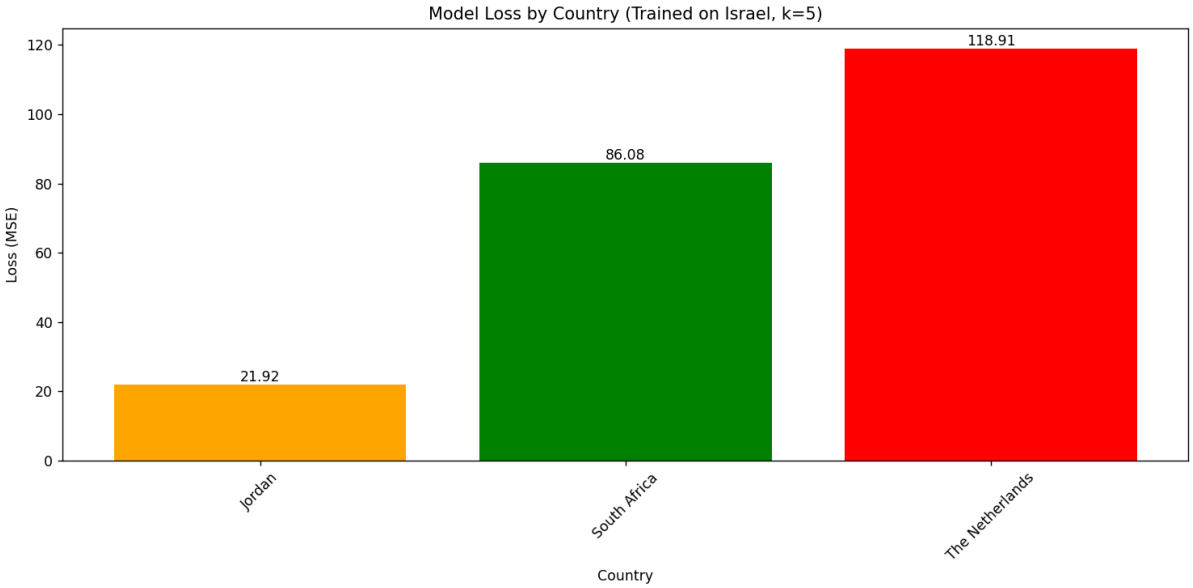
South Africa – doesn't have a similar distribution shape (it's even opposite).

5) Finding optimal rank of the Polynomial Fitting:



The lowest loss is 3.04 at k=5. Based on that, k=5 best fits the data.
K=3,4 are also close and maybe with different sampling they would achieve a lower loss and be chosen.

6) Testing the Israel fitted model on different countries:



The closest country (min loss) to Israel is Jordan. This makes sense as we saw in sub section 4 of Polynomial Fitting that Jordan has the most similar temperatures to Israel.

What's interesting, is that although the temperature distribution in the Netherlands is similar to the one in Israel to the point of constant shifting, the model has the biggest lost there.

*used ChatGPT to analyze plots