Bob Stutchbury
Math 5750/6880: Mathematics of Data Science
Project #1 Final Project
September 9, 2025

1. LaTeX

Experience. I know a bit of the equation syntax, though not much of the document formatting stuff. As a computer nerd with opinions, I am forced to mention Typst, which is built on being a good markup language, as opposed to LaTeX, which seems to largely survive on path dependence and spite.

If not obvious, It was a bit of a struggle to get LaTeX working in my preferred text editor, but I got it figured out now. Did you know it recompiles repeatedly until nothing changes, and much of the work of writing an efficent frontend is guessing when that will happen? INSANE scheme

2. GitHub

My GitHub Project1 repository is located here (see projects/Project1 for this wd):

https://github.com/kerbs23/F25_math_for_data_science

Experience. Used Git for keeping track of some personal projects, but I'm certainly no wizzard and no collaborative experience. Basic forking stuff on gh, but never studied it really.

3. Python and Google Colab

I solved the following ProjectEuler problem: https://projecteuler.net/problem=69

Solution. Used observations about how the score was connected to underlying prime factors to guess 510510, which a brute force run through showed to be correct.

Experience. Self-taught data science stuff by way of getting hired to do it before I was really qualified (Research econometrecs; so data cleaning, static analysis, fixed-effect regression and event study (staggered and no) causal type stuff), so my CS fundamentals arn't great but trying to improve. A bit more reliant on AI than I am proud to admit, but also working on that.

Also, when you look at the .ipynb the markdown/output sections might seem a little wonky, this is because I prefer to edit as one big file, instead of cells. It looks fine on gh, but I am not sure how google collab will deal with it.

4. Regression Analysis

4.1. **Dataset Description.** The California Housing dataset contains median house values for California districts (with a cap) along with eight predictor variables:

- `MedInc`: Median income in block group
- `HouseAge`: Median house age in block group
- `AveRooms`: Average number of rooms per household
- `AveBedrms`: Average number of bedrooms per household
- `Population`: Block group population
- `AveOccup`: Average number of household members
- `Latitude`: Block group latitude
- `Longitude`: Block group longitude

4.2. **Methodology.** I used both standard OLS and ridge regression techniques. The target variable (median house value) was log-transformed to interpret coefficients as approximate percentage changes. A logarithmic transformation was also applied to the `MedInc` variable based on economic priors suggesting diminishing marginal returns to income.

The `Population` variable was excluded from the final specification due to statistical insignificance in my preliminary univariate analyses of each variable. Geographic coordinates (`Latitude` and `Longitude`) were retained as continuous variables despite potential misspecification concerns (ie they only measure northerlyness and westrlyness, which I see as being at best proxies for distance to locations and at worst just a quirk of the shape of the city, but handling them inteligently in this context is difficult and this is all non-causal anyway).

4.3. **Results.**

4.3.1. *Linear Regression.* The final specification included seven predictors: `HouseAge`, `AveRooms`, `AveBedrms`, `AveOccup`, `Latitude`, `Longitude`, and `log_medinc`.

Table 1 reports the various errors for the model. Note the relatively small loss in R^2, suggisting we are not having issues with overfitting. Figures 1 and 2 report the actual vs predicted data, and the histogram of model errors, respectively.

Metric	Training Data	Test Data
R^2	0.629	0.615
MSE	0.120	0.123
RMSE	0.347	0.350
MAE	0.264	0.265

TABLE 1. Performance metrics for linear regression

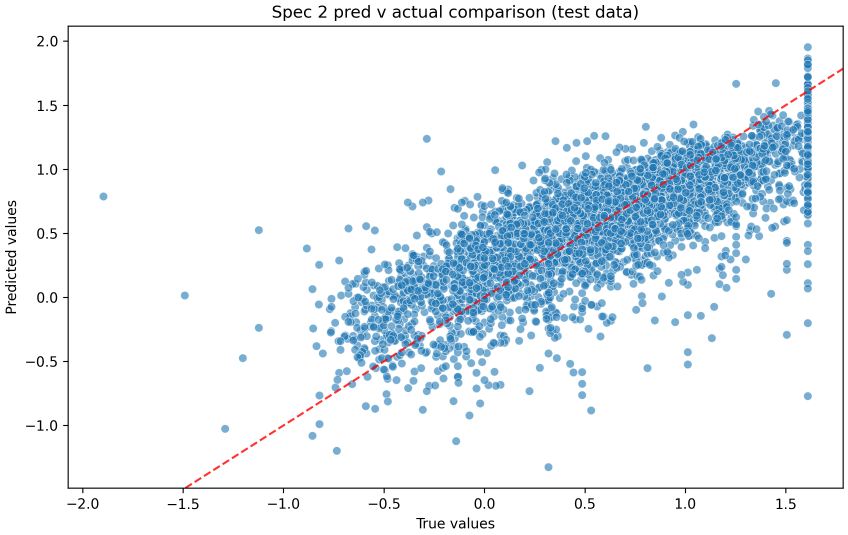

FIGURE 1. Predicted vs. True Values (Test Data)

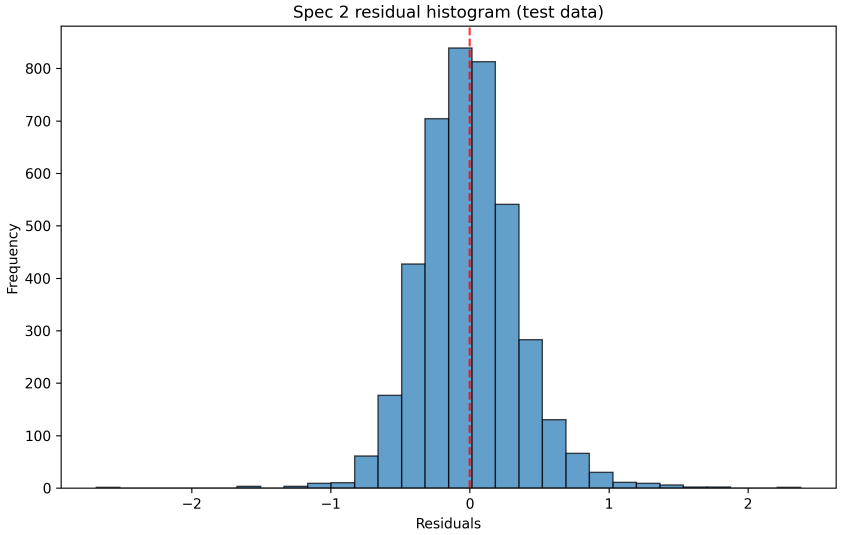

FIGURE 2. Residual Distribution (Test Data)

Metric	Training Data	Test Data
R^2	0.629	0.615
MSE	0.120	0.122
RMSE	0.347	0.350
MAE	0.264	0.265

TABLE 2. Performance metrics for ridge regression

4.3.2. *Ridge Regression.* Ridge regression was applied with ($\alpha = 1.0$). The results were virtually identical to standard linear regression, suggesting minimal multicollinearity concerns in the dataset. Plots not included for brevity, you can see them in the repo if desired.

4.4. **Variable Importance.** The most significant predictor was, by a wide margin, `log_medinc` (log median income), which I find consistent with my experience, confirming something is not horribly wromg. Geographic coordinates (`Latitude` and `Longitude`) showed moderate predictive power, while housing characteristics (`AveRooms`, `HouseAge`) contributed modestly to the model, with (`AveBedrms`) having the most impact from that family.

4.5. **Interpretation.** The residual distribution appears approximately normal, suggesting no bias issues. The similarity between training and test performance indicates reasonable generalization, though the moderate R^2 values suggest unobserved geographic or structural factors may influence housing prices beyond the available predictors. There is no mechanism to make causal claims with this analysis.

5. Classification Analysis

5.1. **Dataset Description.** The Breast Cancer Wisconsin Dataset contains 30 features computed from digitized images of fine needle aspirates of breast masses, including characteristics like radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. The target variable classifies tumors as malignant (0) or benign (1).

5.2. **Interpretation.** Two classification approaches were implemented: Support Vector Machine (SVM) with linear kernel and k-Nearest Neighbors (KNN). Features were standardized prior to modeling. My plotting function had the confusion matrixes as part of the chart before I read the assignment, but I think it is fine.

5.3. **Support Vector Machine Results.** The linear SVM did allright, correctly classifying 97.5% of the data:

Training Set Performance	**Value**
Accuracy	0.989
ROC AUC	0.998
Average Precision	0.998

Test Set Performance	**Value**
Accuracy	0.983
ROC AUC	0.996
Average Precision	0.998

Top 3 Predictive Features	**Coefficient**
mean concave points	0.967
radius error	0.944
worst fractal dimension	0.917

	Predicted Malignant	**Predicted Benign**
Actual Malignant	40	2
Actual Benign	2	70

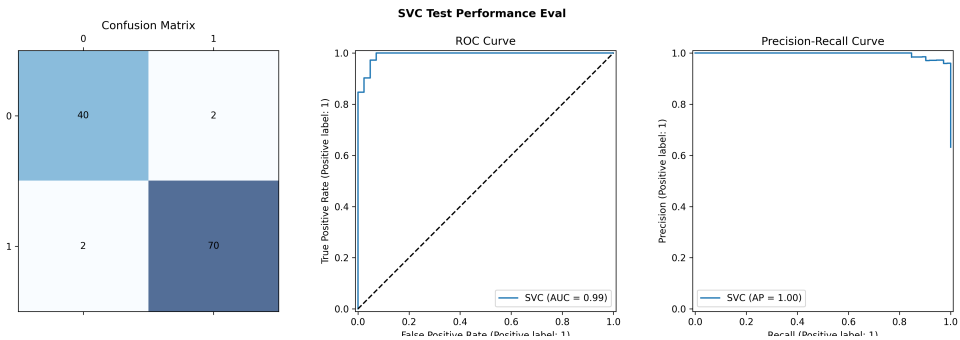

FIGURE 3. SVM Test Performance

Then onto the knn method; less a model and more just comparison but still does classification. With this model, the "most important parameter" is not a relevant question, and running the training data would just match 1-to-1 and get 100% on all metrics, so both are excluded. I put it in a loop, and found that 8 nearest neighbors was optimal, the following are the results for that method:

	Predicted Malignant	**Predicted Benign**
Actual Malignant	39	3
Actual Benign	1	71

Test Set Performance	Value
Accuracy	0.965
ROC AUC	0.977
Average Precision	0.976

5.4. **Interpretation.** Both models demonstrate solid discriminatory power, with both achieving the same overall error rate on the testing data. The models seem to have slight difference in false positive/negative distribution, but there is too few observed errors to realy tell. The feature importance analysis reveals that morphological characteristics related to tumor boundary irregularity (concave points, radius error) and structural complexity (fractal dimension) are most predictive of malignancy.

The minimal performance drop from training to test sets indicates good generalization without overfitting. The high AUC and average precision scores confirm both models effectively distinguish malignant from benign cases across probability thresholds.

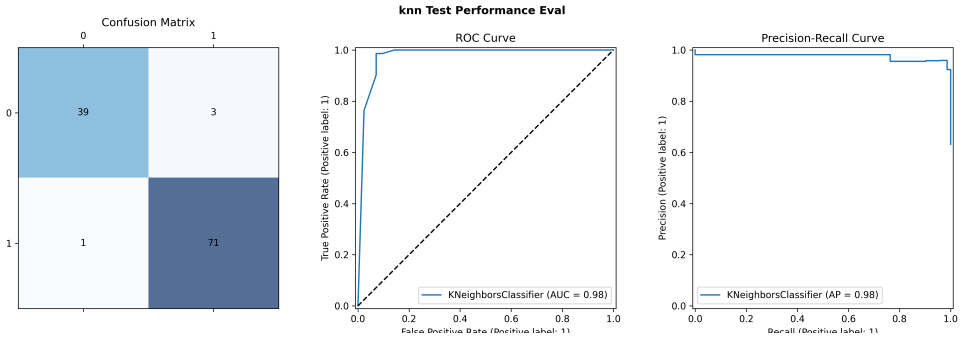

FIGURE 4. KNN Training Performance