Seattle data + Class provided by paper + Custom regression explainer

Experiments ran on a random sample of 500 datapoints from the Seattle housing prices dataset. Each datapoint has 10 features, and a random forest regressor was trained on these features to predict housing price.

- Create baselines with KernelSHAP and GeoXCP
    - Explained a single row of data and looked at distribution of explanations for all data
    - **Results:** The KernelSHAP values and GeoXCP means are exactly equal
- Write the BayesSHAP regression function to go with the 'BayesLocalExplanations' class provided by the authors of the original paper
    - Run a test on the same row of data as above
    - **Results:** BayesSHAP means are very different from KernelSHAP/GeoXCP results, including a few cases where the signs for a feature aren't even the same

Toy model + Class provided by paper + Custom regression explainer

Experiments ran on a toy dataset with the relationship $y = 1 + 1x_1 + 2x_2 + \epsilon$, where ϵ is an error term. A Linear Regressor was trained to predict the value of y from x_1 and x_2. The same single row of data was chosen for explanations in all experiments.

- Baselines with KernelSHAP and LIME
    - Look at feature importances (single value, without uncertainty) for KernelSHAP and LIME
    - **Results:** KernelSHAP and LIME values were not close at all. KernelSHAP produced values of `[0.45511187 0.04836711]` while LIME produced values of `[0.39 0.88]`
- Ran the same data instance through the custom BayesSHAP regression function (same as used in the Seattle housing data experiment)
    - **Results:** BayesSHAP means were very close to the LIME values, but not the KernelSHAP values (BayesSHAP posterior means were `[0.321199    0.87886171]`)

Toy model + Class provided by paper + Custom **classification** explainer

Experiments ran on a toy classification dataset created straight out of scikit-learn. The data contained 5 features (x-values) and were put into either the 1 or 0 class. A Logistic Regressor classifier was trained to predict the probability that the data instance was part of class 0.

Single instance of data

- Baselines with KernelSHAP and LIME
    - Same as above, where the pointwise values for KernelSHAP and LIME are compared

    - **Results:** KernelSHAP and LIME values (ie. Feature contributions towards the predicted probability) are generally similar to each other—see table below
- BayesSHAP results calculated using the classification function provided by the paper's authors, with no modifications at all. A 90% confidence interval was found.
    - **Results:** None of the KernelSHAP means were contained in the Bayes confidence intervals
- Bootstrapping was performed with KernelSHAP for 1500 iterations. A 90% confidence interval was found.
    - **Results:** 2/5 of the bootstrapped confidence intervals have any overlap with the BayesSHAP intervals.
    - **Results:** For any feature, no more than 18% of the KernelSHAP means found during boostrapping were contained in the corresponding BayesSHAP CI.

The exact values are recorded below:

Feature	KernelSHAP	LIME	Bayes mean	Bayes CI
X1	0.101129	0.0863666	-0.023382	(-0.036807, -0.009957)
X2	-0.015381	-0.037105	0.066955	(0.053297, 0.080613)
X3	0.055899	0.0757448	0.015089	(0.001495, 0.028682)
X4	0.086289	0.047751	0.092378	(0.078749, 0.106008)
X5	0.018583	0.0165798	0.048906	(0.035209, 0.062602)

Feature	Bootstrap mean	Bootstrap CI	Bootstrap/Bayes overlap?
X1	0.108743	(0.040700, 0.173411)	False
X2	-0.014261	(-0.036303, 0.008363)	False
X3	0.049845	(0.006528, 0.093296)	True
X4	0.059641	(0.045148, 0.074876)	False
X5	0.019040	(0.000908, 0.038328)	True

All instances of data

- KernelSHAP, LIME, and BayesSHAP were then calculated for all the toy data. KernelSHAP and LIME were both pointwise values while BayesSHAP produced a mean and confidence interval (90%).
    - **Results:** 2.26% of KernelSHAP values are contained in the BayesSHAP CI. 1.76% of LIME values are contained in the BayesSHAP CI.
    - **Results:** The mean relative error between LIME and KernelSHAP is 1.5841.
- A pearson correlation coefficient was also found for these different pairs

  - **Results:** Between LIME and BayesSHAP, as well as KernelSHAP and BayesSHAP, both correlations were very close to 0 with a statistically insignificant p-value. LIME and KernelSHAP had a 0.94 correlation and was statistically significant.

Qualitatively observing the data as well, BayesSHAP continues to struggle with signs and often predicts the feature to contribute in a different direction as compared to KernelSHAP and LIME.

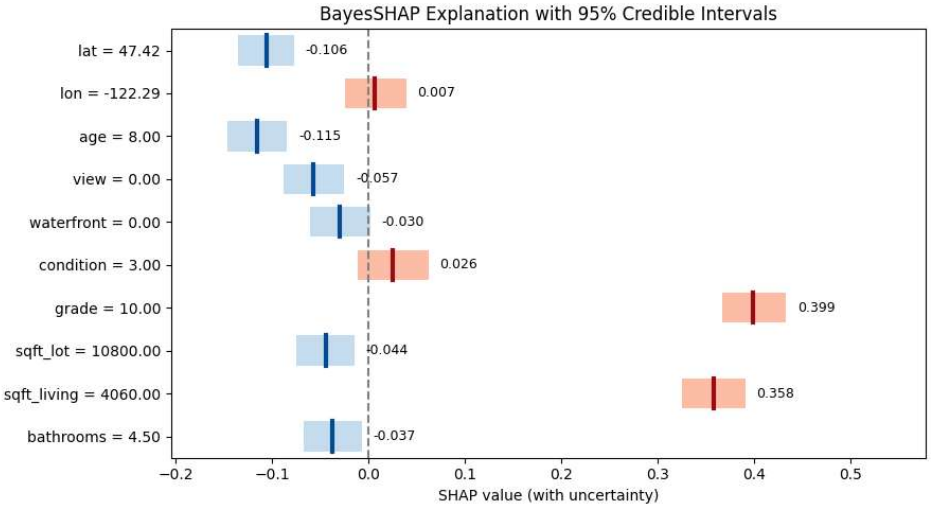
Seattle data + Custom class + Custom classification explainer

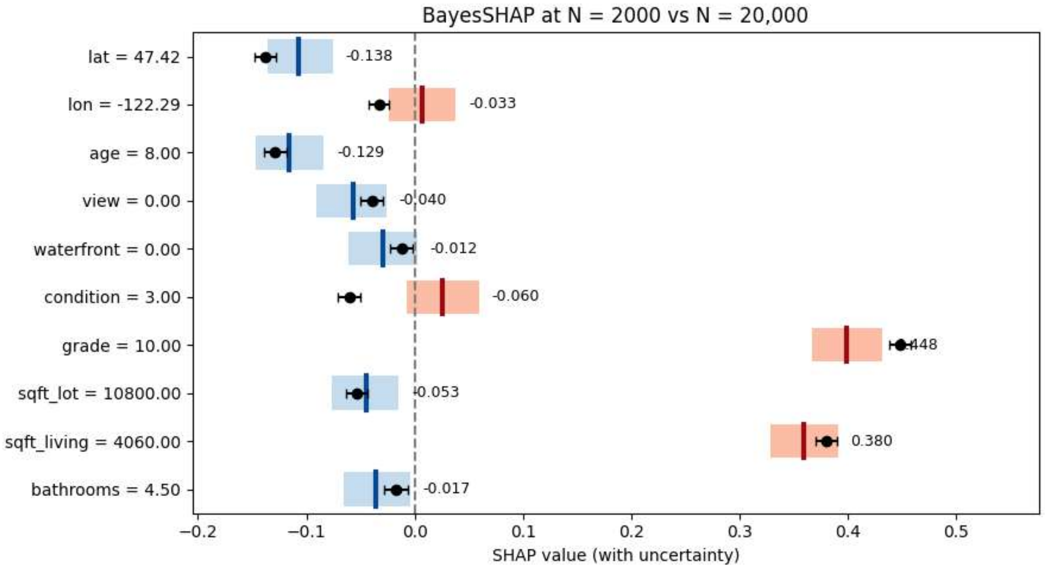
A custom BayesLinearRegression and BayesSHAP class were rewritten based on just the math in the paper, this way everything was written in raw numpy rather than pulling functions from the LIME library as the original code did.

I also went back to the 500-datapoint Seattle dataset so I could do an intuitive check on the feature importances, the data was labeled into ‘high’ or ‘low’ price based on whether or not the house was priced above $500k. A logistic regressor was trained to predict the probability that a house was in that ‘high’ price class.

Single instance of data

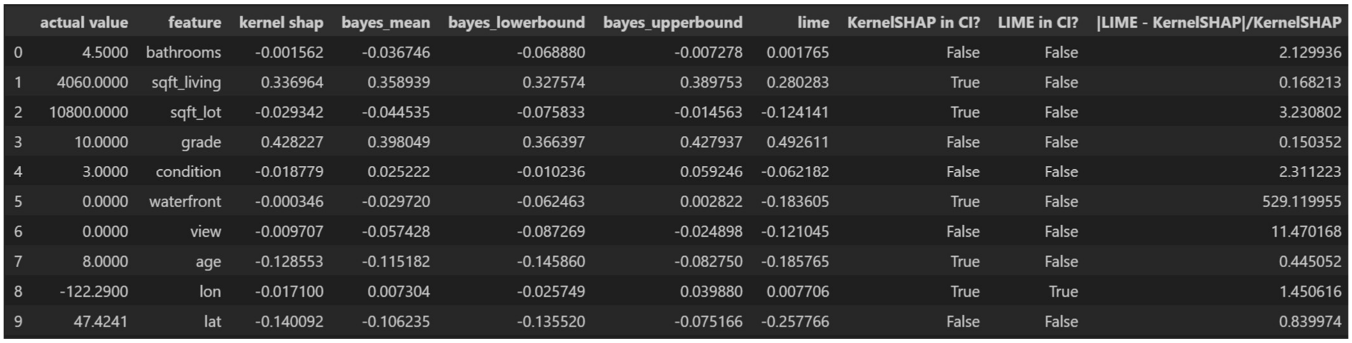
- Baselines with KernelSHAP and LIME
  - **Results:** KernelSHAP and LIME values vary slightly (see table below)
- The BayesSHAP classifier with a lower number of perturbations (N) was implemented, and compared to BayesSHAP at a high number of perturbations—this is the validation method the paper uses.
  - **Results:** In the figures below, the colored bars indicate BayesSHAP at a low N, while the black bars indicate BayesSHAP at high N. Evidently, the CIs indeed get tighter at high N, and the BayesSHAP CI at low N does a fairly good job of capturing the BayesSHAP value at high N.





  - Qualitatively checking the signs of these predictions, the BayesSHAP values with highest magnitude of contribution make sense—the grade and size are comparatively high and both are common factors that drive prices up.

- All three methods (KernelSHAP, LIME, and BayesSHAP) were compared and a check for containment was performed.
  - **Results:** While the CI doesn't perfectly contain everything, a notable improvement is that the signs of the feature contributions are generally correct.



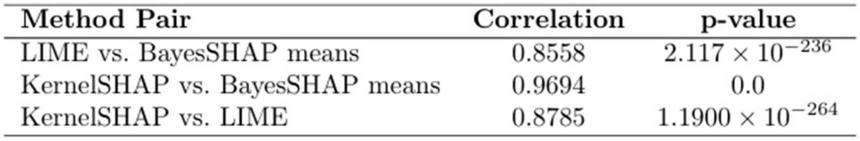
	actual value	feature	kernel shap	bayes_mean	bayes_lowerbound	bayes_upperbound	lime	KernelSHAP in CI?	LIME in CI?	|LIME - KernelSHAP|/KernelSHAP
0	4.5000	bathrooms	-0.001562	-0.036746	-0.068880	-0.007278	0.001765	False	False	2.129936
1	4060.0000	sqft_living	0.336964	0.358939	0.327574	0.389753	0.280283	True	False	0.168213
2	10800.0000	sqft_lot	-0.029342	-0.044535	-0.075833	-0.014563	-0.124141	True	False	3.230802
3	10.0000	grade	0.428227	0.398049	0.366397	0.427937	0.492611	False	False	0.150352
4	3.0000	condition	-0.018779	0.025222	-0.010236	0.059246	-0.062182	False	False	2.311223
5	0.0000	waterfront	-0.000346	-0.029720	-0.062463	0.002822	-0.183605	True	False	529.119955
6	0.0000	view	-0.009707	-0.057428	-0.087269	-0.024898	-0.121045	False	False	11.470168
7	8.0000	age	-0.128553	-0.115182	-0.145860	-0.082750	-0.185765	True	False	0.445052
8	-122.2900	lon	-0.017100	0.007304	-0.025749	0.039880	0.007706	True	True	1.450616
9	47.4241	lat	-0.140092	-0.106235	-0.135520	-0.075166	-0.257766	False	False	0.839974

All data

The full test dataset was filtered to only include instances that were correctly classified by the logistic regressor. Of the 100 test datapoints, 16 of those were incorrectly classified and thus removed from further analysis.

- KernelSHAP and LIME values, as well as a BayesSHAP CI were found for all correctly classified data.

    - **Results:** 72.32% of KernelSHAP estimates fell in the respective BayesSHAP CI while 32.68% of LIME estimates fell in the respective BayesSHAP CI.
        - Mean absolute error between LIME and KernelSHAP was 30.9969
- The correlation and p-values were also found for each pair of the three total methods used, listed below.
    - **Results:** The baseline values all have quite high, statistically significant correlations with BayesSHAP means, denoted in the table below.



Method Pair	**Correlation**	**p-value**
LIME vs. BayesSHAP means	0.8558	2.117×10^{-236}
KernelSHAP vs. BayesSHAP means	0.9694	0.0
KernelSHAP vs. LIME	0.8785	1.1900×10^{-264}

While the BayesSHAP confidence interval is technically supposed to be 95%, there are structural differences between each of the methods. Overall, this analysis provides decent validation for this custom BayesSHAP class.