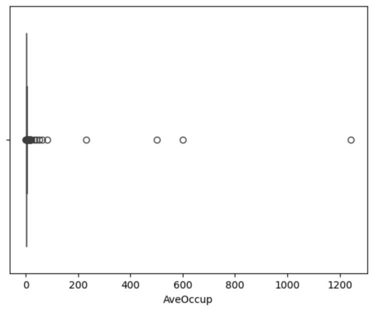
Saood Usmani’s Report
Project 2

Which techniques did you use to train the models?

- I used a pipeline that used the robust scaler and a model I specified to train all the models. I also used grid search to find all the optimal hyperparameters for my model. I used the robust scaler because I noticed that the data had a lot of outliers. For example, medium income, average rooms, average bedrooms, average occupancy, and population all had an extreme amount of outliers. In fact, the boxplot for average occupancy is so slim because of the massive outliers that exist; most houses have an average occupancy of 3(3.07), with 75% having 3.28, but the max is a staggering 1200!

  Because of all these outliers, I didn’t want the data to be negatively affected by standardization, which led me to use the robust scaler since it is better equipped to deal with data that is significantly away from the median by using percentage techniques. I used grid search to find all optimal hyperparameters for the models based on precision, which I will elaborate more on in question two and five. I opted to use pipelining because of the clear developmental efficiency it offers. I remember in the last project I would have to run the same line of code over and over again, just with a different model, and I was thinking that surely there must be some way to optimize this. Pipelining is just that, and made my code not only super easy to develop, but also very readable as well. For example, just looking at the param_grid object you can clearly see what models I am training, what hyperparameters I am training for those models, how exactly I’m looking for those hyperparameters, and more. This makes my code much easier to understand and may also make it easier for me in the future when I am referring back to this project and trying to see what I did, vs going through repetitive code spanning twenty different lines.

Explain any techniques used to optimize model performance?

- I used grid search to optimize model performance by attempting to find optimal hyperparameters for all my models. Some key optimizations that I made were attempting to expand the search space to find optimal hyperparameters I may have otherwise missed, and adding additional hyperparameters to my models so that I could fine tune them to the data as much as possible. For example, I expanded my RandomForestClassifier’s n estimators stop number from 100 to 300, which resulted in me finding the best Random Forest at n=179 at one point, which I would have missed if I didn’t expand my search space. I wrote attempting to expand the search space because I was unfortunately constrained by resources when running the Grid Search, and found that some code even took about an hour or more to run which limited my ability to test these hyperparameters. However, I still ended up with a precision of about 90% on my best model, which I am content with. If I were to continue researching into this problem further, I would definitely try to expand this space with a more computationally intensive machine to see if I was potentially missing better hyperparameters. I would also try to train more models aside from the four that I had, which could potentially have higher precision.

Compare the performance of all models to predict the dependent variable?

- The RandomForestClassifier was the best model to predict the dependent variable in terms of precision, with a precision of 0.90. Following it was AdaBoostClassifier with .87, then KNeighborsClassifier with .85 and DecisionTreeClassifier with another .85. It makes sense that the DecisionTreeClassifier was less than the RandomForestClassifier since we learned in class that the RandomForestClassifier typically does better since it aggregates a bunch of Decision Trees. I wish I could have ran the RandomForestClassifier with more hyperparameters, but maybe that is something I could do in the future when I have more computer power.

Which model would you recommend to be used for this dataset

- I would recommend the RandomForestClassifier to be used for this dataset. This is because it has the highest precision among the four models that I tested, which I believe is the most important metric for this dataset(elaborated more below).

For this dataset, which metric is more important, why?

- For this dataset, I believe the most important metric is precision. This is because I believe we want to minimize the number of false positives. In the context of this data set, this is minimizing the number of households within a block that are marked as price above median, when in reality they are underneath. This is very important because this metric may be used to disperse aid or other resources/grants to a community in need. If a community is falsely marked above this median, when in reality they are below it, they might not get vital initiatives that they need to get to the median; they might actually get less help than normal, since they are falsely marked above the median. Normally, we are worried about false negatives which is very important in the healthcare scene, as someone who gets a false negative could potentially miss out on life saving treatment. However, here, a community that gets a false negative means that the community was not marked for being above the median price, when in reality they were. This doesn't have as much impact as a false positive because this community might receive extra aid that they don't need. More likely, what would happen in a real world scenario is that further testing would be done by individual programs/groups to determine if the community really needed aid, which would reveal that these communities do not need aid. However, in the case of the false positive, once the community is deemed to not need aid, no one will go back to check if they really need aid, resulting in that community lacking the funding/resources they need to get back to the median line. In conclusion, I believe that precision is the most important metric because I believe we want to minimize the number of false positives so that communities don't get falsely marked as above the median line when in reality they are below it, stripping them of valuable resources/aid.

GPT usage:

- I used chatGPT to help me generate the code to print out the four best models from the grid search cv_results_. I used it to help me parse through all the results from the grid search, and then group those results by the four models, and then sort each of those lists for the best model, and then finally aggregate them into one list of best models which I printed out. Here is the thread if you want to see exactly what I prompted: https://chatgpt.com/share/67d3135c-55f8-8013-bee8-18f46b104491