**California house pricing report**

**Intro:** This data is from the 1990 census and contains 10 columns and 20640 rows with the response column being ‘median\_house\_value’. This is a house pricing regression problem which is important as people want an accurate house valuing to maximize their profits or minimize their mortgage and there is evidence of estate agents having subconscious bias in valuing houses.

**Date pre-processing:** I checked for non-values and removed those rows containing them. I used the longitude and latitude columns and the haversine formula to calculate distances between houses and the major cities; Los Angeles, San Diego, San Jose, San Francesco, Fresno, Sacramento and to the nearest city, creating columns for each and removing the longitude and latitude columns. New columns: ‘rooms per person’, ‘family size’, ‘bedrooms per person’ and ‘rooms per bedroom’ were created by combining original columns. I plotted via seaborn a distribution plot and a scatter plot against ‘median\_house\_value’ for each column. This shows that each variable has different ranges of values, most are non-normal and most don’t have a linear relationship with the response variable. This suggests standardizing each variable and that non-linear model such as random forest will work better than linear models. The ‘ocean\_proximity’ variable is nominal so is split into dummy variables by the ‘get\_dummies’ function. Each column is standardized with ‘StandardScaler’ and an intercept is added. The data is then split by explanatory and response variables and train and test datasets with the ‘train\_test\_split’ function.

**Modelling:** Each model will be evaluated with RMSE and r2 score. The 1st model was an average model with the mean of the house prices. The 2nd model was a linear regression model with the original variables and the 3rd model included all variables. The 4th model was linear regression including significant variables calculated by a p value test from the statsmodels.api module. The 5th and 6th modules were ridge regression first with all the variables and then with the parameters optimized via a grid search. The next 3 models are random forest the 1st with all the variables. Using this model and the ‘features\_importance’ function I found the importance of each variable. Using this I ranked the variables and added them cumulatively to reach 95% importance, using these as our strong variables which the next random forest model used. The next model uses a grid search to optimize the random forest parameters. The final 2 models are gradient boosting, 1 with the strong variables and then one with optimized parameters via a grid search.

**Evaluation:** Of the linear models the model using all variables is strongest with a RMSE of 0.58 and r2 of 0.53. This far outperforms the given variable model justifying adding new columns and is very similar to the significant variable model which is the best model of the 3 as it is a much simpler model but has very similar RMSE and r2. Both Ridge models have very similar results to the strong linear models suggesting the added complexity of these models was unnecessary. The random forest models performed better than the linear models with a RMSE of 0.41 and a r2 of 0.8. As predicted the non-linear models outperformed linear. For both Ridge and random forest, the grid search optimization did not improve the model suggesting it is not necessary. However, for gradient boosting the grid search improved the model making the strongest model with a RMSE of 0.39 and r2 of 0.83. Explaining 0.83 of the variance is impressive with such a small number of variables. Variables such as home size, interest rates, qualities of facilities and other variables were not included in the data which is why we cannot explain all the variability.

**Conclusion:** In this report we have pre-processed data, creating new variables to maximise the information used from the data and predicted that non-linear models would be effective via plot visualisation. These models were optimised first threw variable selection and then by a grid search on parameters. The models were then evaluated by RMSE and r2 score and the gradient boosting model was the best performer. To improve I could have used a different method to optimise parameters such as Optuna which is faster than grid search and should pick stronger parameters. XGboost could have also been used instead of the sklearn model as it normally performs better. I could have used a similar dataset but with more columns which would improve accuracy. The improvements will be included in my next project when I try a regression Kaggle challenge.