Grenoble INP
ENSE3

eit InnoEnergy
Knowledge Innovation Community

Machine Learning - Regression Project

Owen Palmer
Jorge Ballesteros

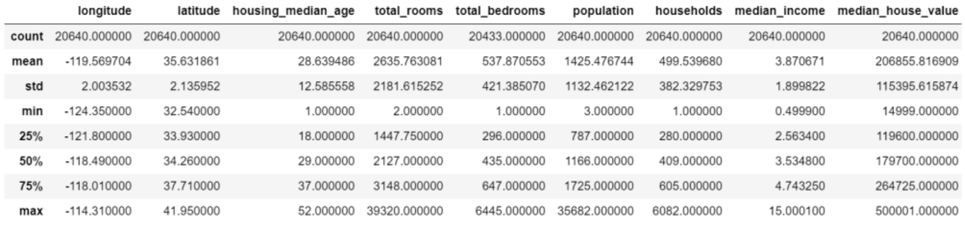
January 2021

Introduction

The aim of our regression project is to create a regression model which uses publicly available data regarding districts in California in order to accurately predict the median house price in the district. This is currently done by a team of experts, who have an error rate of approximately 15%. Our goal is therefore to build a ML model which predicts with equal or better prediction accuracy than 15%.

Data Exploration

We were given access to a dataset representing information about 20,640 districts in California with 10 features including locational data (latitude, longitude, and a categorical feature specifying distance from the ocean), as well as information about housing in the district (number of bedrooms, number of households), as well as information about its constituents (population, median age, median income).

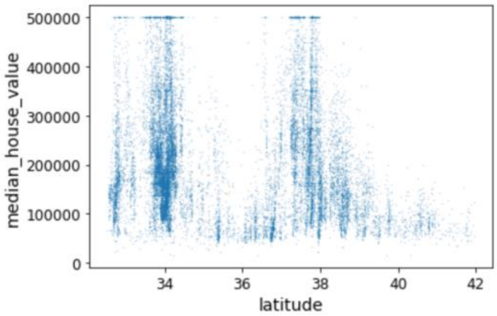


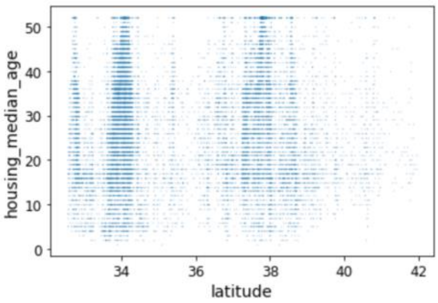
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

Using several plots including histograms and scatter_matrix (not shown here due to space restrictions), we observed some clear patterns between some features that we tried to exploit later in our feature engineering.

Data Cleaning

We were informed that the larger values for ‘median_house_values’ and ‘housing_median_age’ were capped at 500 000 and 50 000 respectively. This explains the unnatural clusters around these values :





Leaving these values as is will confuse our prediction models. In order to avoid this, we decided to remove these values, which represented about 10% of the dataset. This is a large loss, and may bias our models to favour lower price estimates. Other options could be investigated to improve the accuracy of the model, such as creating categorical data from the median ages.

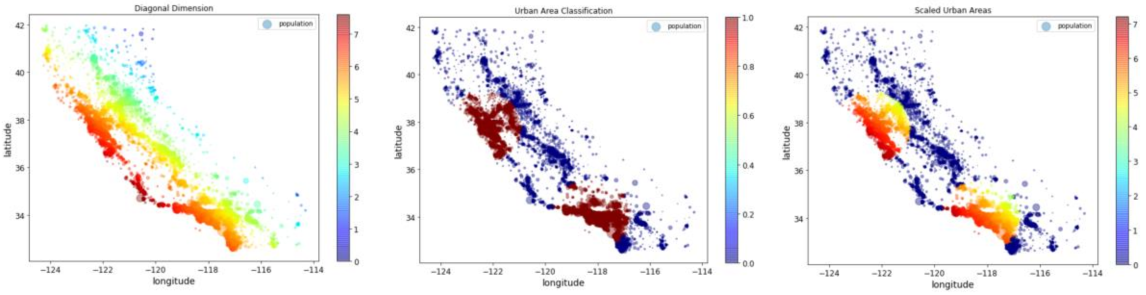
A second, smaller problem was a number of missing data values in the ‘total_bedrooms’ column, making up about 1% of the dataset. We can either interpolate from the mean total_bedrooms, however it was considered to be more reliable to remove these rows. It should however be considered that there is a single reason as to why this value was not recorded for these districts in particular, and as such it is possible that their removal may bias our sample in some way.

Feature Engineering and Importance Analysis

It was recognised in the data exploration stage that there was a clear relationship between the median housing price of a district, and that district's proximity to one of the two major cities in California - San Francisco and Los Angeles. This relationship was so far uncaptured in the features provided, but would be the first thing considered by a real estate expert in assessing pricing.

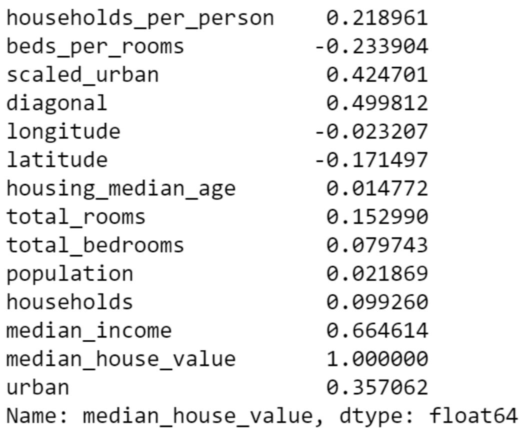
In order to better capture the effect of district’s location on its median house pricing, we introduced x3 new features :

1. Diagonal dimension, calculated as abs(‘latitude’ - ’longitude’)
2. Urban and non urban classification, calculated as a categorical feature based on the district being inside or outside a set ratio from the centre of either LA or SF)
3. Scaled urban, calculated as a multiple of 1 and 2 (Diagonal x Urban)

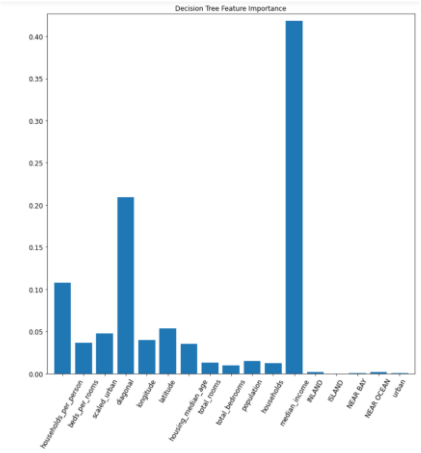


We also added two new features with the aim of taking better advantage of the remaining features that appear to be linked (intuitively, and as they appear to correlate as seen in the scatter plot during data exploration):

1. Bedrooms per number of rooms, calculated as 'total_bedrooms' / 'total_rooms'
2. Households per person ‘households’ / ’population’



```
households_per_person    0.218961
beds_per_rooms          -0.233904
scaled_urban             0.424701
diagonal                 0.499812
longitude               -0.023207
latitude                -0.171497
housing_median_age       0.014772
total_rooms              0.152990
total_bedrooms           0.079743
population               0.021869
households               0.099260
median_income            0.664614
median_house_value       1.000000
urban                    0.357062
Name: median_house_value, dtype: float64
```



The features were then all checked for correlation with the label variable - median housing price (printout to the left, above). A Decision Tree model was (later) trained and analysed to assess the likely feature importance used in the Random Forest model (plot, to the above):

It confirmed that the median income was the most useful for creating splits in the data. The new features ‘diagonal’ and ‘households per person’ were also deemed to be very useful descriptors.

Pre-Processing

StratifiedShuffleSplit was used to set aside a 20% test set that is stratified along the classes of ocean_proximity.
The non-categorical data was then scaled using StandardScaler, which is the necessary as linear models often require this information to be centred with a mean of zero and variance of 1.
Finally, the pandas function ‘get_dummies’ was used to create dummy variables to represent the categorical ocean_proximity data in a series of 4 columns, with each column representing a particular class (one column was left out as recommended - four zeros thus represents ‘1hr from the ocean’ in our data array.

Initial Model Training

The goal of our project is to make estimations that are generally at least 85% accurate. As such, we will assess our models finally on the percentage absolute percentage error (MAPE). For model assessment, we decided to use R2 however, as this will give us a more reliable indication of whether our models are fitting the data well. A fast initial check of a number of different models was carried out :
Lasso - MAPE : 24%   R2 : 0.68
Ridge - MAPE : 24%   R2 : 0.67
LinearRegressor - MAPE : 24%   R2 : 0.68
DecisionTreeRegressor - MAPE : 22%   R2 : 0.61
RandomForestRegressor - MAPE : 16%   R2 : 0.81

It was decided to continue further using the Ridge and RandomForest regressor as the main subjects for further tuning. Cross validation was used to confirm and obtain reliable starting R2 values from the training data before hyperparameter tuning.

Cross Validation and Hyperparameter Tuning

It was seen for s, and by cross validation using 10 fold k-fold, that the random forest regressor had a tendency to overfit the training data (excessively high training R2), and so could benefit from hyperparameter tuning. Particularly via complexity reduction using tree pruning (ccp_alpha), reducing max_depth, or max_number of features, all of which could be useful for reducing overfitting on the training data, and so improve the cross validation scores and eventual test scores.
The ridge regressor appeared to have consistent performance with the original alpha value used, however it was investigated if its performance could be ameliorated further.

Grid Search CV was used to tune the regularization parameter alpha for Ridge (approximately 1 was found to give the best performance, however only a slight increase in accuracy of less than 1 percent resulted). The four following parameters for Random Forest :

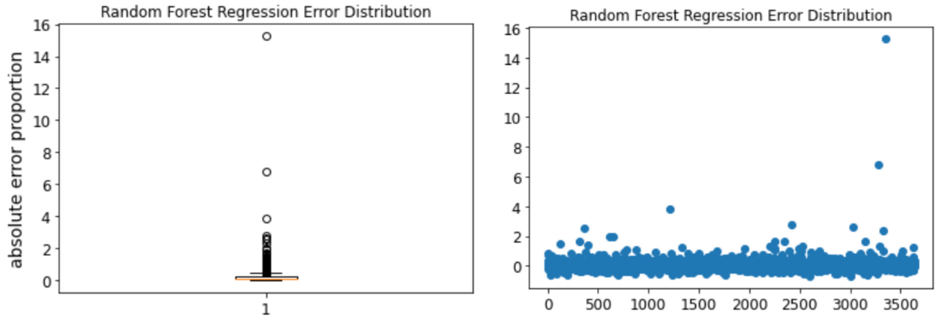
{'ccp_alpha': 1, 'max_depth': 70, 'max_features': 8, 'n_estimators': 20}

Note that these values would sometimes change depending on the initialization state, and that some parameters like n_estimators were kept at lower than ideal values to keep computation times reasonable. This did not significantly change the accuracy in any case.

It should also be noted that the introduction of the custom features, and the types of data cleaning done (ie. dropping all non-reliable columns, rather than interpolation or extrapolation), can all be considered hyperparameters, however these were not tuned for simplicity.

Results

The best result we were able to achieve with our random forest model was approximately 16% MAPE. However, it must be noted that there were also many predictions that were dramatically worse.



The following plots show that while there is a high density of prediction errors that are very tightly packed around the mean, there remain many prediction errors which are 2, 3, even 16 times the actual median housing price of the district in the test set.

While this is not in any way desired, it does suggest that there is good scope to improve our model by the addition of a booster model in order to correct these large outliers. It is also possible that these large outliers are an artifact of our decision to ‘drop’ many of the higher median house price districts in the data cleaning stage, thus biasing our data towards lower value districts.

Conclusion

Unfortunately we were not able to achieve the required 15% mean accuracy rate, however it could be argued that our software model can provide these results at a fraction of the cost of a team of experts, and can be run in a fraction of the time.

As an extension, the information obtained in the project in the assessment of feature importance could also be useful for experts. The famous mantra of real estate is ‘Location Location Location’, however many experts may be interested to know that in fact median income is a far more reliable indicator of higher prices than geographical information alone. This information could be valuable for improving the accuracy of their valuations to potentially below 15%.