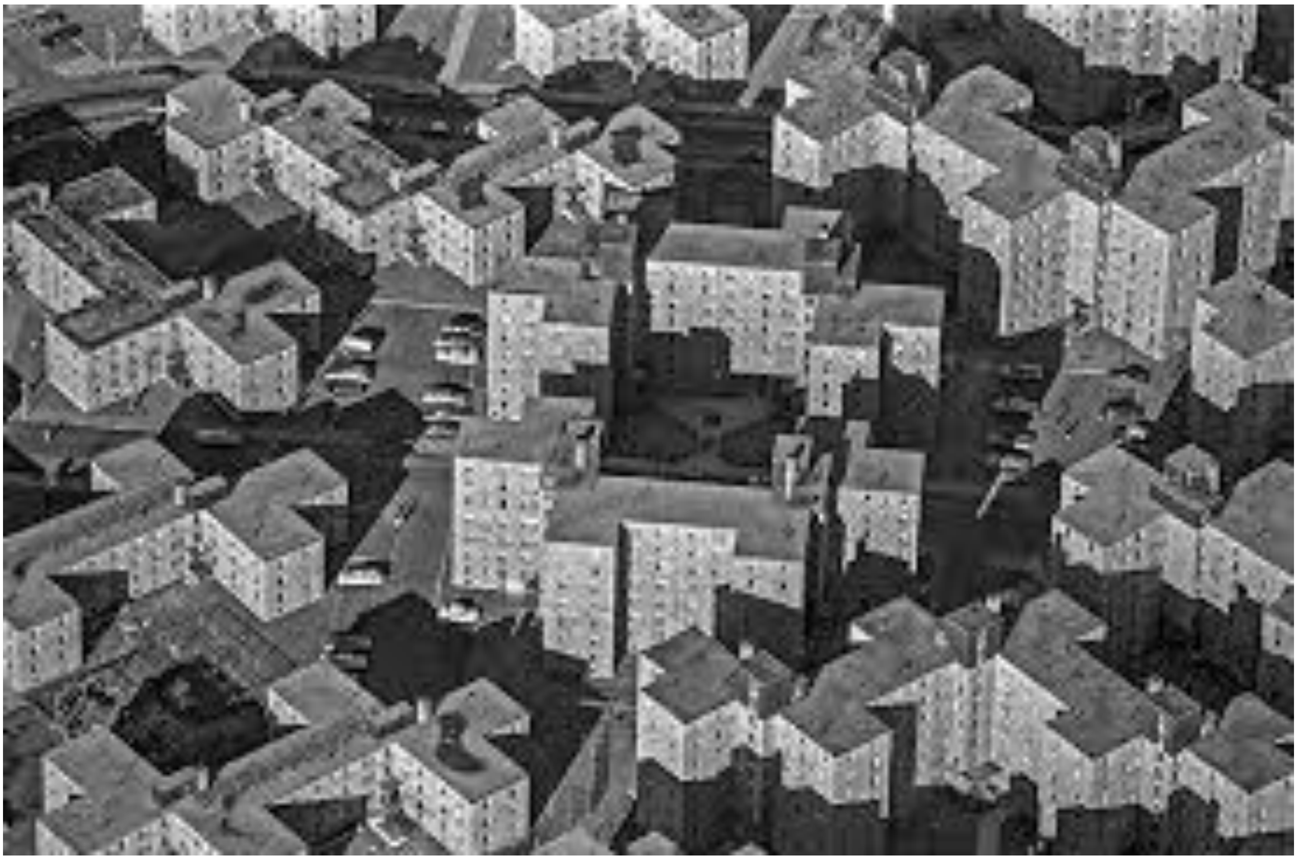
Machine Learning - Project 2



Responsibility table

	Regression a	Regression b	Classification	Discussion	Exam
Gabriel	30%	30%	30%	40%	40%
Jonathan	30%	30%	40%	30%	30%
Lucas	40%	40%	30%	30%	30%

Regression, part a

Continuing from our last report, we wish to model the median housing value of Boston towns based on the other features of the dataset. We don't have any other models to compare to, and thus our goal is simply to get the best model possible.

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT

See report 1 for detailed descriptions of features. These features are normalized and standardized, and we also added a new feature, RM^2, as we noticed a parabolic relationship in the residuals over RM in our preliminary analysis. Introducing RM^2 decreased our validation error by about 5, see figure 1 for reference. This feature is only introduced for linear regression, as it was the only model we had grounds for adding it to. Besides, the neural network can introduce nonlinear effects itself and might be overfit with the extra feature. As for k-nearest neighbors, the argument for introducing the extra feature is even weaker and it might also introduce overfitting.

With these features, we will first make a linear regression model with regularization. Using ten-fold cross validation, we have modeled the effect of λ, on the validation error in figure 1. We let λ range from 0 to 2 based on a preliminary analysis.

Results

The minimum validation error is achieved at about $\lambda = 0.33$ depending on exactly how the data was split into folds. Regularization has a very small effect on a linear regression of this dataset, and the training error isn't much lower than the testing error, meaning overfitting isn't a big issue yet. We could therefore probably benefit from employing a more complex model. Training a linear regression model with this regularization error on the entire dataset results in the model:

$$MEDV = 22.5 - 1.09 \cdot CRIM + 0.82 \cdot ZN + 0.44 \cdot INDUS + 0.61 \cdot CHAS - 2.03 \cdot NOX - 17.81RM + 20.70RM^2 - 0.11 \cdot AGE - 2.47 \cdot DIS + 2.27 \cdot RAD - 1.91 \cdot TAX - 1.67 \cdot PTRATIO + 0.69 \cdot B - 3.84 \cdot LSTAT$$

Where all the features are first standardized with our training set (entire dataset) means and standard deviations[1]. The standardizations allow us to interpret the contribution of each feature directly from its coefficient. We won't go through all the features, but in general, the signs of the coefficients make intuitive sense. We would expect crime rates and lower social status of population to relate to lower house prices etc.

[1] For brevity, these are not included in the report, but can be found in our last report

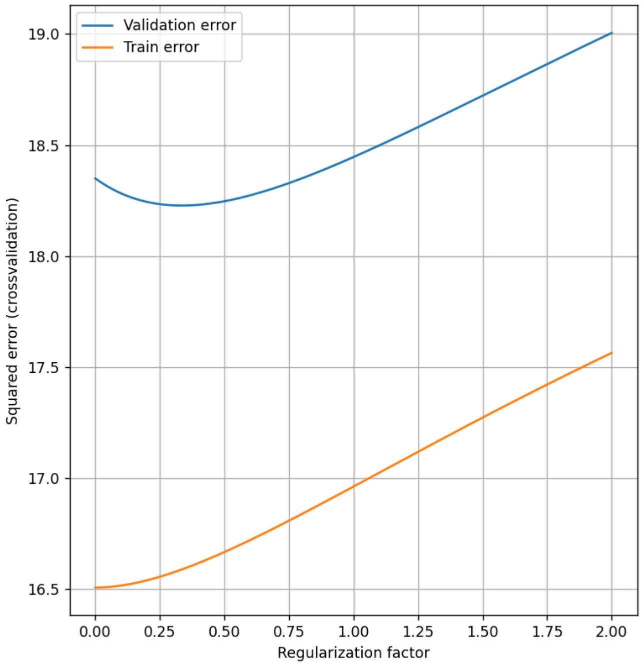


Figure 1: The validation and train error plotted as functions of the regularization factor used.

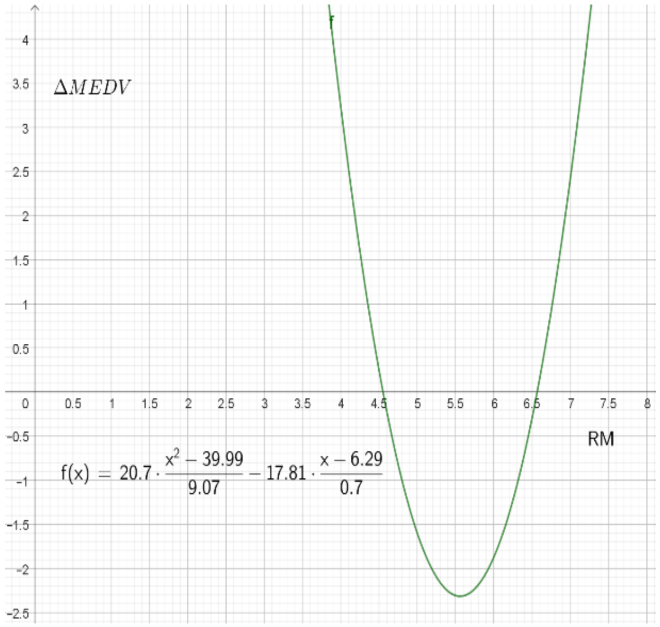


Figure 2: The total effect of RM on the linear regression model. The x-axis represents the raw value of RM and the y-axis represents the sum of the two terms based on RM in the model.

Interpretation

In our last report we noted that LSTAT and RM were highly correlated with MEDV, so we would expect them to contribute highly to the result of our model. LSTAT has a high negative effect in our model as expected, however, for RM it is harder as the square term was introduced.

To examine RM, we plotted the sum of the RM and RM^2 terms in figure 2, including the standardization. Interestingly, the parable is centered around 5.5, meaning that towns with fewer than 5.5 rooms per dwelling start to have higher median prices, according to the model. This might just be a spurious relationship, or one caused by confounders (e.g. dense cities with high housing prices), as there does not seem to be any logical reasoning behind this and relatively few datapoints ($\sim 8\%$) lie in this range. The parabolic relationship makes sense for the rest of the data, as one could expect prices of houses with more rooms to increase more than linearly, as the type of housing changes to something the likes of luxury villas.

Other things of note are the large contributions of NOX, DIS and RAD. We noted the weirdness of RAD jumping from 8 to 24 in the data in the last report, but it seems like this did represent something useful, as the feature still contributes highly. The scatterplot did not show the positive relationship at all, but it makes sense logically that better access to highways would increase house prices. It also fits nicely into our explanation about these datapoints representing dense cities. This could also explain the relatively low effect of crime, as houses in cities may both experience high amounts of crime and high house prices, despite crime being undesirable.

DIS was the average distance from employment centers. It is unexpected that it has such a high effect on housing prices, but here the most likely explanation is confounders, namely that employment centers are most likely close to dense cities, which have higher housing prices for other reasons.

The original article presenting the data was researching the effect of NOX on housing prices, and interestingly we found the same result, namely that it lowers the median housing value of towns. However, though the original article linked this to the buyers' interest in clean air, one should note that NOX and INDUS were highly correlated. NOX might simply better represent industry that people don't want to live near, like heavy-duty factories that pollute more, and not necessarily their awareness of or interest in clean air.

Finally, note that INDUS has a slight positive effect on MEDV in our model. This was against our intuition that industry lowered house prices. The value is low, so it may very well be spurious, however, it could also indicate that industry also has a positive effect. Possibly because people get to live close to where they work. Note also that since we have NOX as a different feature, the effect of heavy-duty polluting industry might already be accounted for by the NOX term.

The explanations are purely to show that the model makes sense based on our understanding of the problem. We cannot conclude anything certain about cause and effect.

Regression, part b

We wish to compare the linear regression model with regularization from last section with an artificial neural network (ANN) as well as a baseline. We do this with 2-level cross-validation with 10 folds for both the inner and outer level, making sure to use the exact same folds for all models. In the inner level, we find the optimal regularization parameter for linear regression and structure for the ANN and do nothing for the baseline. In the outer level, we “train” and evaluate the baseline, as well as new models based on the optimal structure/regularization parameter found by averaging the validation error over all the inner folds for each hyperparameter. The range of regularization values used for linear regression is 0 to 2, as in the last section.

Procedure

One can train a neural network in a multitude of ways, so we decided to make our process clear. Firstly, we decided to test the different structures:

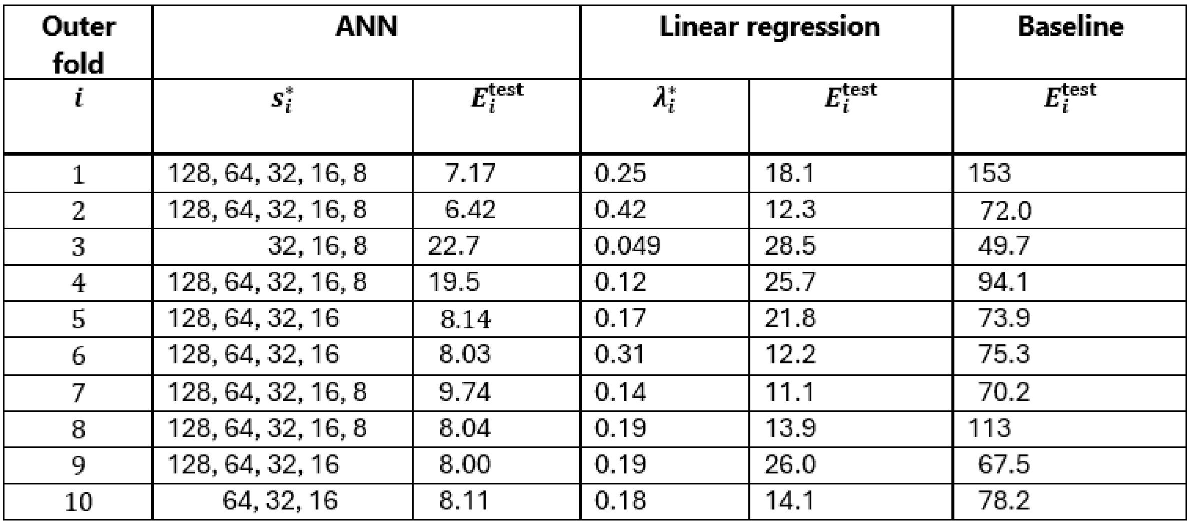
$[1], [8], [16,8], [32,16,8], [64,32,16,8], [128,64,32,16,8], [16], [32,16], [64,32,16], [128,64,32,16]$

Each set of square brackets represents one structure, where each number in the square brackets, *n*, represents a hidden layer with *n* units. These structures were chosen based on intuition gained from preliminary tests, where it seemed like having a decreasing number of neurons per hidden layer gave the best models. Ideally, we would have tested a lot more different structures, but we are limited by computation time. To train the neural network we decided to use early stopping. If the neural networks *validation* error hadn’t improved within the last 100 epochs (training rounds where each epoch corresponds to using the entire training set once), we stopped the training. This meant that we took out some of the training data (10%) of the outer loop to get a validation set for training on the found optimal structure. Thus, the final test set is completely unused for training.

We decided to train each neural net 3 times in the inner loop and choose the best validation error to represent that structure, to eliminate some of the randomness in initialization. Ideally, we would have done the same in our outer loop. However, we didn’t make the implementation, and only doing it once just means our generalization error is a bit higher than it could be.

Results

The results of the two-level cross-validation are noted in table 1. Note that the optimal regularization values vary between the folds but are all in the same order of magnitude as in the first section of this report, except for fold no. 3, with a much smaller regularization parameter. However, fold no. 3 seems to have held a lot of hard to predict



Outer fold	ANN		Linear regression		Baseline
i	s_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	128, 64, 32, 16, 8	7.17	0.25	18.1	153
2	128, 64, 32, 16, 8	6.42	0.42	12.3	72.0
3	32, 16, 8	22.7	0.049	28.5	49.7
4	128, 64, 32, 16, 8	19.5	0.12	25.7	94.1
5	128, 64, 32, 16	8.14	0.17	21.8	73.9
6	128, 64, 32, 16	8.03	0.31	12.2	75.3
7	128, 64, 32, 16, 8	9.74	0.14	11.1	70.2
8	128, 64, 32, 16, 8	8.04	0.19	13.9	113
9	128, 64, 32, 16	8.00	0.19	26.0	67.5
10	64, 32, 16	8.11	0.18	14.1	78.2

Table 1: Results from each fold of the two-level cross-validation

datapoints in the test-set (and thus not in the training set), as both non-trivial models had high test error in this fold, which may explain why it sticks out. Interestingly, a simpler neural net structure was chosen for this fold than others, which one would think is contradictory to a low amount regularization in the linear model. However, if we imagine that the training and validation sets consist of points that lie relatively close to a linear 13-dimensionel plane, the neural network could still be prone to overfitting, whereas a linear regression model would benefit most from little regularization. It is therefore important to note that comparing complexities between different model types includes much uncertainty. That being said, in accordance with our conclusion that we could benefit from more complex models in the first section of the report, the most complex neural network structures were most often chosen, with a first hidden layer of 128 in 8 of the 10 folds. As we didn’t try more complex models than this, it seems there is still something to gain from making even more complex models.

Statistic evaluation

To conclude upon which models are best we decided to employ a paired t-test of the individual test errors, resulting in the pairwise 95% confidence intervals and p-values shown in table 2. For column A/B, the difference is the error of A minus the error of B.

As seen, there is really no question that both the models are better than the baseline, and the ANN is better than the linear model with high significance. Furthermore, the difference is at least 3.32 with 95% confidence, which we judge is a sizeable amount considering the size of the errors. Therefore, we suggest training a neural net for this task.

	Baseline/Linear	Baseline/ANN	Linear/ANN
Mean difference	66.27	74.06	7.79
Confidence interval	[53.33; 79.21]	[60.25; 87.87]	[3.32; 12.26]
p-value	7.88e-22	1.35e-23	6.69e-04

Table 2: Pairwise comparisons between the three models

Classification

The medians house prices can be classified into the 3 categories “Low”, “Middle”, and “High” value, split at the 33 and 67 percentiles. We now have a multi-class classification problem. As before, we compare 3 models with 2-level cross-validation with 10 folds for both inner and outer level. These models are:

1. A K-nearest-neighbors with K ranging 0 to 10 as complexity controlling parameter
2. A multinomial logistic regression model with the regularization strength λ ranging 0 to 2 as complexity controlling parameter.
3. A baseline model which always guesses the most common classification in the training data

For each outer fold, the optimal complexity controlling parameter as well as the test error was logged. The results are shown in the table below.

Outer fold	**KNN**		**Logistic Regression**		**Baseline**
$\boldsymbol{i}$	$\boldsymbol{k_i^*}$	$\boldsymbol{E_i^{\text{test}}}$	$\boldsymbol{\lambda_i^*}$	$\boldsymbol{E_i^{\text{test}}}$	$\boldsymbol{E_i^{\text{test}}}$
1	1	0.29	1.90	0.24	0.76
2	1	0.12	0.74	0.16	0.71
3	1	0.24	1.48	0.16	0.73
4	1	0.22	2.00 **(cap)**	0.25	0.67
5	1	0.33	0.64	0.22	0.80
6	1	0.25	0.43	0.20	0.69
7	1	0.20	1.90	0.24	0.72
8	5	0.20	0.43	0.16	0.78
9	1	0.22	2.00 (**cap**)	0.18	0.76
10	1	0.38	1.48	0.30	0.74
Mean		0.25		0.21	0.74

Table 2: Results from the classification cross-validation. λ_4^* and λ_9^* are equal to 2.00. As we only tested values between 0 and 2, it is possible that more optimal values of λ exist but were not found.

KNN uses just $k = 1$ in 9 out of 10 folds, which suggests that most data points are close enough to another point that the two represent the same category. The regularization strength parameter λ for Logistic Regression are varied throughout the range from 0 to 2, but values closer to 2 are more common. From fold to fold, it varies whether KNN or Logistic Regression performs the best with lowest error.

Statistics

With McNemar's test, we can compute the p-values for the null hypotheses that the models perform the same on average in setup I.
H_0: $\mu_1 = \mu_2$, where μ_1 and μ_2 are the average error rate of two models.

Baseline & Logistic Regression: $p = 3.40 \cdot 10^{-51}$

Baseline & KNN: $p = 7.58 \cdot 10^{-46}$

Logistic Regression & KNN: $p = 0.108$

At 5% significance level, we conclude that Logistic Regression and KNN both perform significantly better than the Baseline. We cannot conclude that Linear Regression performs better on average than the KNN model.

The 95% confidence intervals of the error of the models are as follows:

Baseline: $[0.70,\ 0.77\ ]$

KNN: $[0.21,\ 0.28]$

Logistic Regression: $[0.18,\ 0.25]$

Despite KKN and Logistic Regression not performing significantly different on average, the lower mean error of Logistic Regression anyway means that it is more likely for that to be the better performing model than KNN. In addition, the Logistic Regression is easier to interpret, as it can be expressed by 3 equations computing the likelihood of the datapoint laying in each category. There are thus small advantages in using Linear Regression over KKN for this problem. A multinomial logistic regression model is our recommendation for this task.

Logistic regression model

With ten-fold cross-validation, we test 100 values of λ between 0 and 2 just like we did previously for a linear regression model. We find that $\lambda = 0.15$ makes for a good logistic regression model. For each category 'Low', 'Middle', and 'High', we find the model parameters:

Low

$$P('Low') = SOFTMAX(-0.34 + 0.29 \cdot CRIM + 0.04 \cdot ZN - 0.05 \cdot INDUS - 0.09 \cdot CHAS + 0.59 \cdot NOX - 0.39 \cdot RM + 0.57 \cdot AGE + 0.56 \cdot DIS - 0.43 \cdot RAD + 0.19 \cdot TAX + 0.52 \cdot PTRATIO - 0.31 \cdot B + 1.08 \cdot LSTAT)$$

Middle

$$P('Middle') = SOFTMAX(0.49 - 0.38 \cdot CRIM - 0.04 \cdot ZN + 0.16 \cdot INDUS + 0.02 \cdot CHAS - 0.19 \cdot NOX - 0.47 \cdot RM - 0.27 \cdot AGE - 0.11 \cdot DIS + 0.09 \cdot RAD + 0.06 \cdot TAX - 0.02 \cdot PTRATIO + 0.35 \cdot B - 0.08 \cdot LSTAT)$$

High

$$P('High') = SOFTMAX(-0.14 + 0.08 \cdot CRIM - 0.00 \cdot ZN - 0.11 \cdot INDUS + 0.07 \cdot CHAS - 0.41 \cdot NOX + 0.87 \cdot RM - 0.30 \cdot AGE - 0.46 \cdot DIS + 0.34 \cdot RAD - 0.25 \cdot TAX - 0.50 \cdot PTRATIO - 0.05 \cdot B - 0.10 \cdot LSTAT)$$

Interpretation

A multinomial logistic regression model computes the probability for a datapoint belonging to each classification with a softmax operation. We see that the proportion of the population of low social status $LSAT$ significantly increases the likelihood of a 'Low' classification while it slightly decreases the likelihood of a 'Middle' or 'High' classification. This corresponds well to the large negative weight in the linear regression from earlier. Similar relations between the two models are seen for $PTRATIO$, DIS, RAD, TAX, and NOX.
Features given insignificant weight in the linear regression also has little effect on the logistic regression classification. This is seen for AGE and $INDUS$ which are given low weight by the linear regression model and do not contribute significantly to any class being chosen by the logistic regression model.

Discussion

In our linear regression analysis of the Boston housing dataset, we noticed no significant changes with regularization, suggesting our model wasn't overfitting and could potentially benefit from more complexity. Our analysis also revealed logical connections between the median housing prices and their features. Notably, the average number of rooms per dwelling (RM) showed an interesting pattern where both larger and smaller homes resulted in higher prices, possibly due to the appeal of spaciousness and desirable locations, respectively. Additionally, the % lower status population (LSTAT) had a predictable negative effect on house prices. Factors like distance to employment centers (DIS) and accessibility to radial highways (RAD) also had great weight suggesting the importance of strategic placement of housing.

Expanding our analysis to compare linear regression with a baseline model and an ANN, we got pairwise 95% confidence intervals and p-values confirming that with statistical significance, both linear regression and ANN outperformed the baseline model, and that our ANN had at least 3.32 lower mean generalization error than our linear regression with 95% confidence.

For classification our analysis showed that both the KNN and logistic regression models significantly outperform the baseline model again. However, there is no obvious preference for KNN or logistic regression. Statistical analysis using the t-test showed no significant differences between the two models at a 95% confidence level.

Logistic regression showed a marginally better mean generalization error in comparison to the baseline model, which could suggest a slight improvement on average over the KNN model, although the significance between the two models isn't proven sufficiently.

Comparison to other literature

We will compare our analysis to *FACTORS DRIVING RESIDENTIAL PRICES IN BOSTON IN THE 1980'S* by Sebastian M. S. and Martin H. They carried out a linear regression, however, they did not do any feature transformations besides the normal standardization, contrary to us. Additionally, they omit the features LSTAT and RAD, which is interesting as they are large contributors to our model. A large part of their paper focuses on the high effect of RM on their model prediction, something we also discovered; however, they did not notice any nonlinear effects of the RM and neither the explanation that dense cities may give more positive value to otherwise "bad" features such as crime and industry. Their findings, however, do not contradict ours. However, they noted that there may have been a confounder to RM, namely dwelling size, which is not something we considered, but isn't contradictory to our findings.

They further noted that the highest contributors after room size were tax rate, pupil-teacher ratio and nitric oxide concentration. We also found NOX to be one of the most important features for our model, and if we ignore LSTAT and RAD which their paper didn't include, PTRATIO and TAX were also among the most contributing factors in our model. However, DIS contributed more to our model than the two others mentioned. One could argue that possibly since DIS is conceptually correlated to LSTAT, as there may be more employment centers in areas with low lower status of population, it makes sense that the positive effect of low LSTAT may partly cancel out the negative effect of DIS, thus making it not significant in their analysis.

Summing up, however, their paper's findings mostly agree with ours, only differing in the ordering of importance of variables, and focus of interpretations.

Problems:

Q1:

To find the correct predictions for the ROC curve it is useful to place the threshold θ on the subplot and calculate the TPR and FPR using following:

$$TPR = \frac{TP}{TP+FN}, FPR = \frac{FP}{FP+TN}$$

For Prediction A we'll place the threshold at 0.6 and count the amounts of TP, FP, TN and FN. We get the following matrix:

	Predicted Positive	Predicted negative
Actually positive	TP: 2	FN: 2
Actually negative	FP: 3	TN: 1

The rates are thus: $TPR = \frac{2}{2+2} = \frac{1}{2}, FPR = \frac{3}{3+1} = \frac{3}{4}$. Looking at Figure 1 the point $(\frac{3}{4}, \frac{1}{2})$ is not on the ROC curve and because of that can't be Prediction A.

For Prediction B we can place θ at 0.8 and do the same procedure as last time:

	Predicted Positive	Predicted negative
Actually positive	TP: 2	FN: 2
Actually negative	FP: 0	TN: 4

The rates are thus: $TPR = \frac{2}{2+2} = \frac{1}{2}, FPR = \frac{0}{0+4} = 0$. Looking at Figure 1 the point $(0, \frac{1}{2})$ is not on the ROC curve and because of that can't be Prediction A.

For Prediction D we can place θ between the first and the second observation at around 0.5 and do the same procedure as last time:

	Predicted Positive	Predicted negative
Actually positive	TP: 4	FN: 0
Actually negative	FP: 3	TN: 1

The rates are thus: $TPR = \frac{4}{4+0} = 1, FPR = \frac{3}{3+1} = \frac{3}{4}$. Looking at Figure 1 the point $(\frac{3}{4}, 1)$ is not on the ROC curve and because of that can't be Prediction A.

Since the only other option is C, prediction C must be the correct answer.

Q2:

To find the purity gain by using ClassError we find out how many observations N there is at the root, how many observations there is when $x_7 = 2$ and when $x_7 \neq 2$

$$N(r) = 33 + 28 + 30 + 29 + 4 + 2 + 3 + 5 + 1 = 135$$

$N(x_7 \neq 2) = 134$

$N(x_7 = 2) = 1$

Impurities can now be calculated for the root, $x_7 \neq 2$ and when $x_7 = 2$

$I(r) = 1 - \frac{33+4}{33+4+28+2+1+30+3+29+5} \approx 0.7259259$

$I(x_7 \neq 2) = 1 - \frac{37}{33+28+30+29+4+2+3+5} = 0.7238806$

$I(x_7 = 2) = 1 - \frac{1}{1} = 0$

And with these values calculated we can find the impurity gain by the following equation:

$$\Delta = I(r) - \sum_{k=1}^{K} \frac{N(x_k)}{N(r)} \cdot I(v_k)$$

$$\Delta = 0.7259259 - \frac{134}{135} \cdot 0.7238806 - \frac{1}{135} \cdot 0 = 0.007407379$$

From the calculations the correct answer is C

Q3:

With 7 inputs being x_1 to x_7 and a hidden layer containing 10 units there is in total $10 \cdot 7 = 70$ weights. In the hidden layer each unit has a bias term and therefor 10 biases in total. From the hidden layer to the output layer there is $10 \cdot 4 = 40$ weights. In the output layer there is also 1 bias term for each output neuron for a total of 4. In total there is $70 + 10 + 40 + 4 = 124$ parameters.

The correct answer is A.

Q4:

Let's test the rule assignments with the point $b_1 = -0.5$ $b_2 = 0.5$. This point is situated in the dark blue area corresponding to Congestion level 1.

Rule assignment A: A: $-0.5 \geq -0.16$ - False. We go to B: $0.5 \geq 0.01$ - True, we get the answer congestion level 2 which is not correct and we move on to the next.

Rule assignment B: A: $-0.5 \geq -0.76$ - True. We go to C: $0.5 \geq 0.03$ - True, we get the answer congestion level 4 which is not correct.

Rule assignment C: A: $0.5 \geq 0.03$ - True. We go to C: $0.5 \geq 0.01$ - True, we get the answer congestion level 4 which is incorrect.

Rule assignment D: A: $-0.5 \geq -0.76$ - True. We go to C: $-0.5 \geq -0.16$ - False. We go to D: $0.5 \geq 0.01$ - True, we get the answer congestion level 1 which is the only correct answer to this specific point and it is shown that D is the correct answer.