Danmarks Tekniske Universitet

DTU

Project 2: Supervised Learning on California Housing Prices

02450 Introduction to Machine Learning and Data Mining

Authors

Jordan Shopp - s226743
Natalie Valett - s226441
Ka Chun So (Andy) - s226545

July 13, 2023

Contents

1 Regression

1.1 Regularized Regression Model

The goal of our regression models is to predict the household median income of a housing block group. With our linear regression models, we hope to be able to predict median income accurately with low bias. Our model uses the remaining attributes in the data set, which consists of longitude, latitude, median age of housing, total rooms, total bedrooms, population, households, median house value, and ocean proximity. Because ocean proximity was reported as a nominal categorical variable, we transformed it using one-of-k encoding.

We also chose to add additional attributes by transforming certain pairs. The attributes total rooms and total bedrooms, are arbitrarily measured because they depend on the size of the arbitrary Housing Blocks the US Census studies, which vary largely in number of households and population. To compensate for this, we added 6 new attributes to our model: Rooms per Person, Bedrooms per Person, Households per Person, Rooms per Household, Bedrooms per Household, and People per Household.

We then standardized our data set $\mathbf{X}$ such that each column had an average 0 and standard deviation 1.

We began our investigation by trying to find an optimal value for regularization parameter λ that would minimize generalization error. We tested lambda values $10^{-5}, 10^{-4}, 10^{-3}$, $...10^{7}, 10^{8}, 10^{9}$ by testing each with K=10 fold cross-validation.

As seen in Figure 1, the optimal λ value is 1.0, which out of the K=10 graphs produced during our investigation, 6 found optimal λ=. As the figure shows, though, there does not seem to exist a U-shaped curve that is often found when examining the effect λ has on error. The U-shape curve typically is the result of over fitting-caused bias on the left of the λ axis, and under fitting-caused bias on the right. Because there are n=20433 observations in our data set, we believe that the relative flatness on the left-side of the S-curve could be the result of the seeming non-existence of over fitting bias. Though, we do see the effect high λ has on error in both graphs. As λ increases, most variables' weights decrease in magnitude due to the increased error penalty for larger magnitude weights. As this effect becomes more extreme, we experience under fitting bias, which explains the sharp slope in the S-curve's rise.

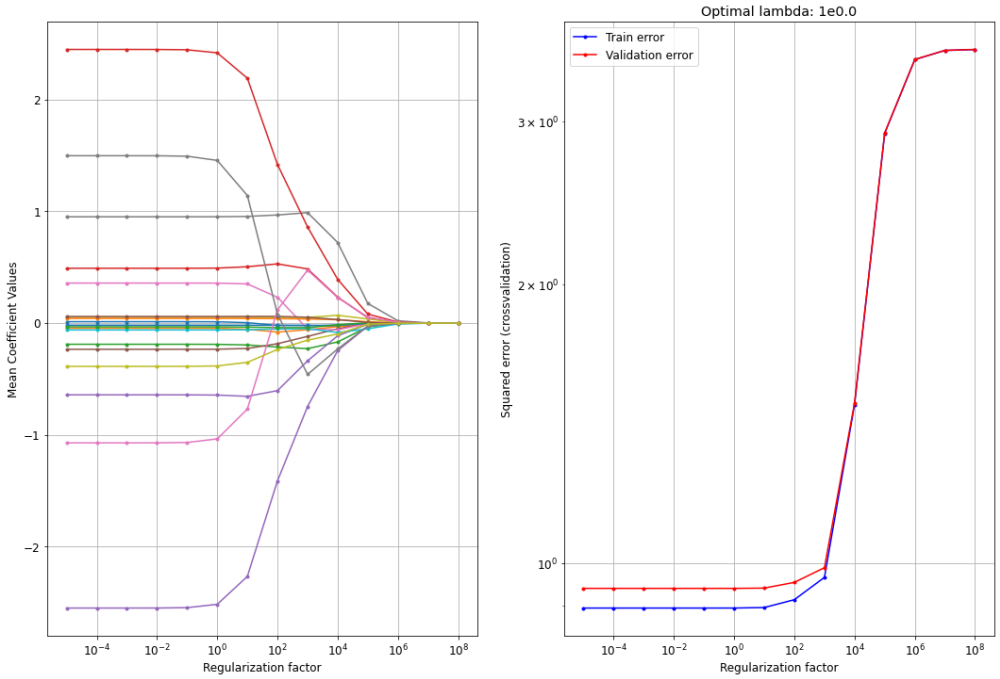


Figure 1: On the left: the effect of Regularization Factor on Mean Coefficient Values. On the right: the effect of Regularization Factor on Error.

Table 1 lists the Attribute Weights for the linear regression model found with optimal λ. The offset is merely the average median income over the data set. The latitude, longitude, and ocean proximity seem to have little influence on the median income attribute. The largest positive weights are Median House Value, Bedrooms per Person, and Rooms per Household, which intuitively makes sense because people who earn more can buy more expensive, larger houses with spare bedrooms. The largest negative weights are Bedrooms per Household and Rooms per Person. These findings initially surprised us, though it could be explained by the idea that people tend to make more money in cities where there is limited space, but one would need to study the relationship between these attributes to a depth greater than the scope of our investigation to better understand a possible reasoning behind these weights.

Table 1: Attribute Weights of the K=10 Fold With Optimal λ=1.

Attribute	Weight	Attribute	Weight
Offset	3.87	Ocean Proximity: 1 Hour of Ocean	0.05
Longitude	0.01	Ocean Proximity: Living Inland	-0.06
Latitude	-.05	Ocean Proximity: Living on an Island	-0.02
Median Housing Age	-0.19	Ocean Proximity: Living Near the Bay	0.04
Total Rooms	0.5	Ocean Proximity: Living Near the Ocean	-0.03
Total Bedrooms	-0.66	Bedrooms per Household	-2.52
Population	-0.23	People per Household	0.06
Households	0.36	Rooms per Person	-1.05
Median House Value	0.95	Bedrooms per Person	1.46
Rooms per Household	2.43	Households per Person	-0.38

1.2 Comparison with ANN and Baseline

Next, we will compare 3 different models: our linear regression model described above, an artificial neural network, and a trivial baseline. To do this, weâll use 10-fold cross-validation to analyze which of our models best achieves our goal of predicting the median income of a housing block group. We're also interested in finding the optimal complexity-controlling parameters for our models, so within each fold of the cross validation, we'll train and test each model on a set of parameters to find their optimal values. For the regression model, the parameter of interest is the regularization strength (λ), which is used in the loss function to penalize overfitting and reduce bias. As for the artificial neural network, our parameter of interest is the number of hidden units (h). The baseline model simply estimates the mean value from the training data, thus it has no parameters to optimize.

From the average error rates of the 3 models visualized in Figure 2, we can see intuitively that both the artificial neural net far outperforming the trivial baseline. With 99% confidence, the ANN model yields an error from 0.646 to 0.68, and the linear regression model from 0.908 to 0.974. The trivial baseline, however yields an error from 3.46 to 3.76, again with 99% confidence. Thus the most accurate model is the ANN, with the linear regression model performing only slightly worse, but both performing significantly better than the baseline. The inferiority of the baseline to the other 2 models is further confirmed by the p-values.

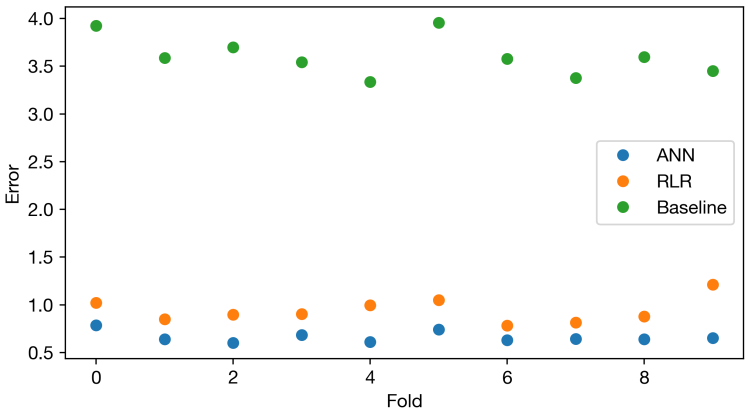


Figure 2: Errors for each model on every fold

Outer Fold	Baseline	ANN		Linear Regression	
i	E_i^{test}	h_i^*	E_i^{test}	λ_i^*	E_i^{test}
1	3.9239	3	0.8393	0.01	1.0238
		4	0.7846	0.1	1.0237
		5	0.7734	1	1.023
		6	0.7661	10	1.0188
		7	0.7602	100	1.0235
2	3.5857	3	0.6462	0.01	0.8458
		4	0.6199	0.1	0.8458
		5	0.6608	1	0.8459
		6	0.6398	10	0.8475
		7	0.6278	100	0.8699
3	3.6992	3	0.6365	0.01	0.8977
		4	0.6134	0.1	0.8977
		5	0.6164	1	0.8977
		6	0.5833	10	0.8975
		7	0.5635	100	0.8984
4	3.5412	3	0.7051	0.01	0.8954
		4	0.6937	0.1	0.8954
		5	0.6663	1	0.896
		6	0.6951	10	0.902
		7	0.6561	100	0.9362
5		3	0.6303	0.01	0.9875
	3.3372	4	0.6178	0.1	0.9875
		5	0.604	1	0.9881
		6	0.6156	10	0.9937
		7	0.5896	100	1.0151

Outer Fold	Baseline	ANN		Linear Regression	
i	E_i^{test}	h_i^*	E_i^{test}	λ_i^*	E_i^{test}
6	3.9542	3	0.7895	0.01	1.0436
		4	0.7295	0.1	1.0436
		5	0.7292	1	1.0439
		6	0.7236	10	1.0465
		7	0.7287	100	1.0645
7	3.5781	3	0.6253	0.01	0.7801
		4	0.6106	0.1	0.7801
		5	0.6207	1	0.7801
		6	0.6652	10	0.7812
		7	0.6353	100	0.7981
8	3.3768	3	0.6484	0.01	0.81
		4	0.6548	0.1	0.81
		5	0.632	1	0.8099
		6	0.6701	10	0.81
		7	0.6164	100	0.8257
9	3.5962	3	0.6734	0.01	0.8758
		4	0.6572	0.1	0.8758
		5	0.6494	1	0.8756
		6	0.6072	10	0.8754
		7	0.6132	100	0.8884
10	3.4501	3	0.6616	0.01	1.2295
		4	0.6406	0.1	1.2293
		5	0.6209	1	1.2275
		6	0.6778	10	1.2128
		7	0.656	100	1.1636

Table 2: Model parameters and corresponding errors of 3 models in 10-fold cross validation: a simple baseline, an ANN with the number of hidden units h, and a linear regression model, with the regularization strength parameter

The p-value between the baseline and ANN is 1.05e-12, and between the baseline and linear regressor is 3.39e-11. These very low p-values prove there is a significant difference in performance between our baseline and the other models. On the other hand, the p-value between the ANN and linear regressor is 4.62e-05. This measure is still non-zero, indicating there's still a difference in performance between the models, but a less drastic difference than either model had with the baseline, to be sure.

Based on these results, the artificial neural network is the best performer, and the recommended model for our use case. However, the linear regression is not far behind, and is a suitable alternative. Whichever model one chooses, there is still some variability in error introduced by the complexity-controlling parameter chosen. Table 2 presents the results of the 2-layer cross validation, including the error rates for each of the parameters chosen in the ANN and linear regression models. By taking the average error for each parameter tested, we can find the optimal parameters which yield the lowest errors. For the artificial neural network, we find 7 hidden units to be optimal, yielding an error rate of 0.6447 on average. For our linear regression model, a regularization strength of 10 is optimal, averaging an error of 0.9385.

2 Classification

2.1 A Multiclass Classification Problem

We have chosen to solve the classification problem of predicting the ocean proximity category for California housing data. This is a multiclass classification problem, as there are five distinct categories: "NEAR BAY", "NEAR OCEAN", "ISLAND", "INLAND", and "<1H OCEAN". We discovered that the total amount of category "ISLAND" is less than 10 therefore it is dropped before starting any types of learning. Four categories remain. The goal is to build a model that accurately distinguishes between these categories based on features such as longitude, latitude, housing median age, total rooms, total bedrooms, population, households, median income, and median house value. By solving this classification problem, we can gain insights into the factors that influence the ocean proximity of residential properties and potentially use the information to inform housing market analysis and urban planning decisions.

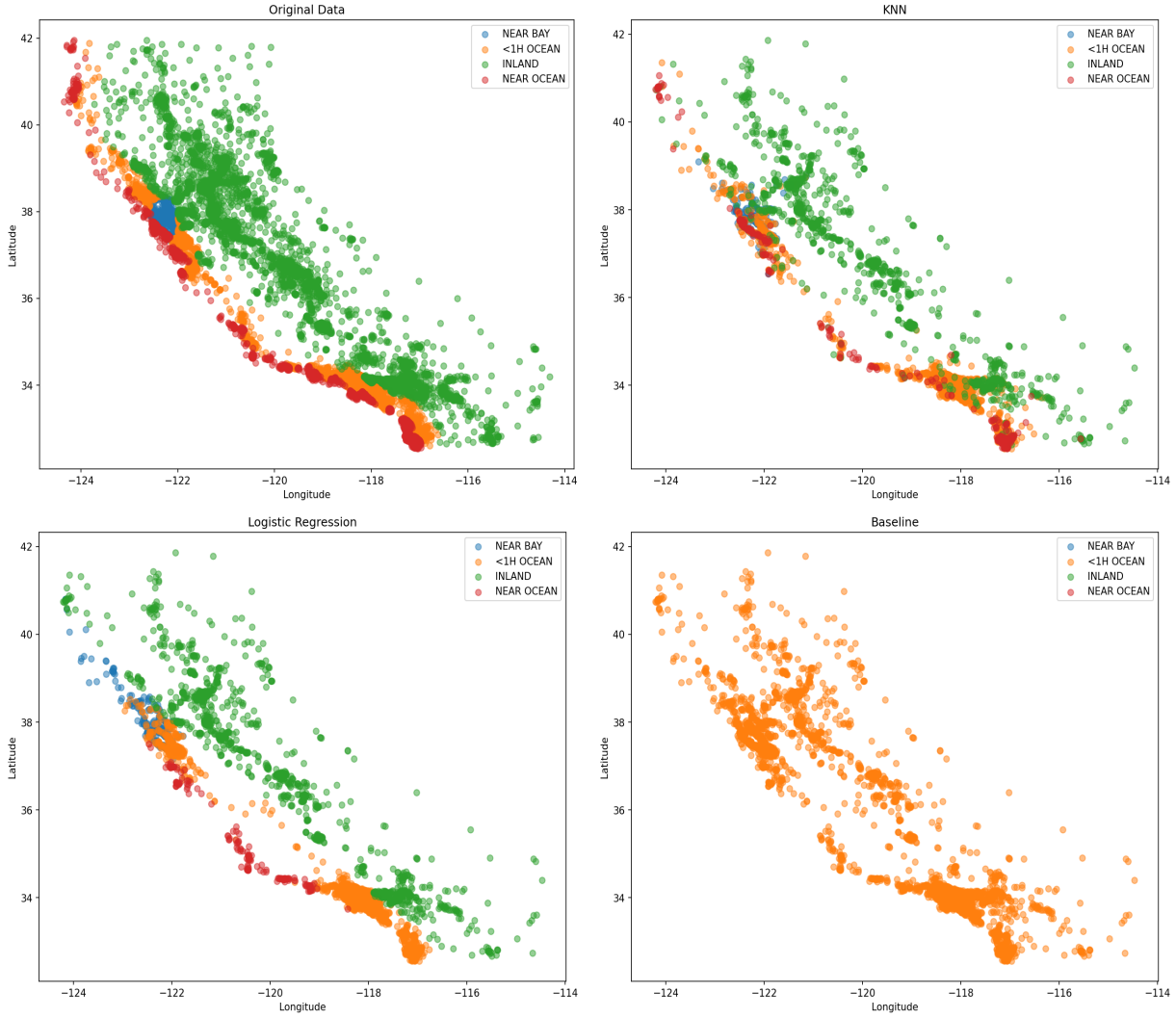
2.2 Performance of 3 Models

The baseline model is a simple classifier that predicts the most frequent class in the training data. This model helps us to understand the minimum performance we should expect from a more advanced classifier. To implement the baseline model, we have used the DummyClassifier with the most_frequent strategy.

KNN is a non-parametric, lazy learning algorithm that predicts the class of a new observation based on the majority class of its nearest neighbors. It requires you to choose the number of neighbors (k) as a hyperparameter. We have performed a nested cross-validation to find the optimal k value and error rate for each fold.

Logistic Regression is a linear classifier that models the probability of an observation belonging to each class using the logistic function. It requires regularization to prevent overfitting, and we have used the L2 regularization with the hyperparameter lambda. We have performed a nested cross-validation to find the optimal lambda value and error rate for each fold.

The below are the prediction made by KNN classifier, baseline classifier and logistic regression classifier, comparing with the original data.



Outer Fold	Baseline	KNN		Logistic Regression	
i	E_i^{test}	K_i^*	E_i^{test}	λ_i^*	E_i^{test}
1	55.7513	11	17.2785	0.00001	19.3343
2	55.7513	3	18.4043	0.1	19.2854
3	55.7513	5	18.6490	0.1	19.2854
4	55.7513	7	17.0827	0.1	19.2854
5	55.8003	5	17.7680	0.00001	19.3343
6	55.8003	4	18.1106	0.1	19.2854
7	55.8003	5	17.8169	0.01	19.3343
8	55.8003	5	19.0406	0.1	19.2854
9	55.7786	5	16.9931	0.1	19.2854
10	55.7786	4	18.2664	0.1	19.2854

This table presents a comparison of the error rates for three different classifiers: Baseline, KNN (K-Nearest Neighbors), and Logistic Regression. The comparison is based on a 10-

fold cross-validation process, with each row corresponding to a different outer fold of the cross-validation.

The baseline classifier has a consistently high error rate, around 55.75% to 55.80%, which indicates that it is not very effective at making accurate predictions.

The KNN classifier, on the other hand, shows a significant improvement in performance compared to the baseline classifier. The error rates for KNN range from 16.99% to 19.04% across the outer folds. It is important to note that the optimal number of nearest neighbors varies between the outer folds, suggesting that the performance of the KNN classifier is sensitive to the choice of neighbors.

The Logistic Regression classifier demonstrates a consistently better performance than the baseline classifier and a slightly better performance than the KNN classifier. The error rates for Logistic Regression range from 19.28% to 19.33%. The optimal regularization parameter shows less variation than the KNN's neighbors parameter, with 0.1 being the most common optimal value across the outer folds.

In summary, both the KNN and Logistic Regression classifiers show significantly improved performance compared to the baseline classifier.

2.3 McNemar's test

Baseline/KNN	**Correct**	**Incorrect**	**Total**
Correct	1645	143	1788
Incorrect	1662	636	2298
Total	3307	779	4086

The p value is 0. It is significant that KNN performs better than Baseline.

Baseline/Logistic Regression	**Correct**	**Incorrect**	**Total**
Correct	1633	155	1788
Incorrect	1661	637	2298
Total	3294	792	4086

The p value is 0. It is significant that Logistic Regression performs better than Baseline.

KNN/Logistic Regression	**Correct**	**Incorrect**	**Total**
Correct	2979	328	3307
Incorrect	315	464	779
Total	3294	792	4086

The p value is 0.6081822. There is not significantly difference between the performance of KNN and Logistic Regression.

KNN and Logistic Regression models are both better than the Baseline model. This can be seen from the contingency tables and the fact that the p-values for the McNemar's tests between Baseline/KNN and Baseline/Logistic Regression are both 0 (or very close to 0),

indicating a significant difference in performance between these pairs of models. The KNN and Logistic Regression models have lower error rates compared to the Baseline model. The p-value for the McNemar's test between KNN and Logistic Regression is 0.6081822, which is not significant (assuming a significance level of 0.05). This indicates that there is no significant difference in the performance of the KNN and Logistic Regression models. In other words, these models are not identical, but their performance is not significantly different in this specific dataset.

Recommendations based on the analysis: Both KNN and Logistic Regression models should be considered over the Baseline model since they have significantly better performance. It might be worthwhile to explore other models or optimization techniques (e.g., hyperparameter tuning) to improve the performance of the KNN and Logistic Regression models further.

3 Discussion

3.1 Reflection on our Results

In this paper we've presented our findings from performing various machine learning techniques on the California Housing dataset. The 2 goals we set out were:

1. Predict median income of a housing block group
2. Predict the ocean proximity category of a housing block group

With respect to our first goal, we trained a regularized linear regression model to predict median income, and used 10-fold cross validation to tune its regularization strength parameter, and to compare it against 2 alternative models. We learned that 10 is the optimal regularization strength for our regression model. Further, we learned that even with the optimal parameter, that a trained artificial neural network slightly outperforms our linear regression model at predicting median income. However, our linear regression model vastly outperforms a simple baseline.

As for the second goal, our analysis demonstrates that both KNN and Logistic Regression models are more effective than the Baseline model for this classification task. There is no significant difference between the KNN and Logistic Regression models, suggesting that either model can be chosen based on other factors, such as interpretability or computational requirements. Moreover, it is recommended to explore additional models, optimization techniques, or validation strategies to further improve the performance and robustness of the chosen classification model.

3.2 Comparison to Other Studies

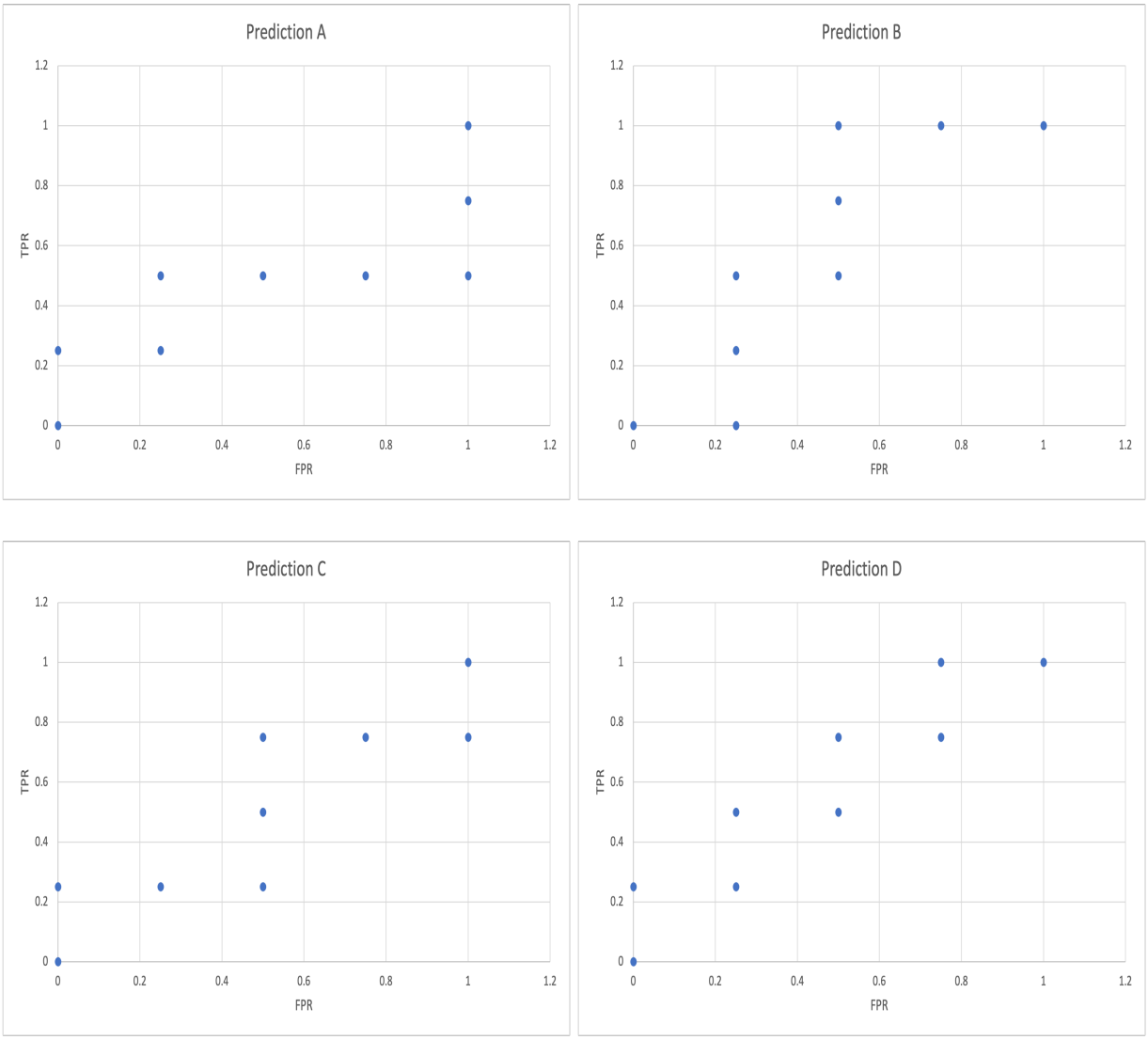
While the California Housing Prices data set is very popular for machine learning instruction and practice, most of the previous studies we have found focus on predicting the median house price rather than the median income. Additionally, we found that many of

the previous studies focused on different linear regression methods, and thus it was difficult for us to compare with others the relative performance of regularized linear regression and artificial neural networks. None the less, we did find that other researchers used similar techniques to ours to engineer new attributes using the existing ones. In **Predicting California Housing prices by using Machine Learning**, Abhishek Shah adds Rooms per Household, Bedrooms per Household, and People per Household attributes to improve the performance of their model. The paper states that these engineered attributes consist of two of the five most important attributes in their model. We found similar results, with all four of our largest attribute weights by magnitude being four of our engineered attributes. This illustrates the importance of engineered attributes.

4 Exam Problems

Problem 1
The ROC curve is influenced by the layout of the data points. Below are the corresponding ROC curves to each candidate prediction. The one that matches the provided ROC curve is correct.

Correct Answer: **C**



Problem 2 The Impurity Gain, $\Delta = I(r) - \sum_{k=1}^{2} \frac{N(v_k)}{N(r)} I(v_k)$, where I is the impurity measure, in this case, classification error. $I(v) = 1 - max(p(c|v))$. $I(r) = 1 - max(\frac{37}{135}, \frac{31}{135}, \frac{33}{135}, \frac{34}{135}) = .726$, where r represents the root. $I(v_1) = 1 - max(\frac{1}{1}) = 0$, and $I(v_2) = 1 - max(\frac{37}{134}, \frac{30}{134}, \frac{33}{134}, \frac{34}{134}) = .724$. The Impurity, $\Delta = .726 - (\frac{1}{135} * 0 + \frac{134}{135} * .724) = .0074$. Thus, the correct answer is option **C**.

Problem 3

The NN will have 88 trainable parameters to fit the model: 7 attributes each go to 11 nodes (10 hidden units and one bias node), then the 11 nodes go into 1. Thus the correct answer is 7 * 11 + 11 = 88 option **D**.

Problem 4

The answer is **D**. We can deduce this by observing that congestion level 4 spans b2, so no branch leading to congestion level 4 should discriminate based on attribute b2. D is the only option in which A and C (the branches leading to congestion level 4) aren't based on b2, so this is the only possible answer.

Problem 5

The answer is option **C** Let's calculate the time taken for each model in the inner cross-validation loop: Neural network: (20 ms train + 5 ms test) * 20 iterations = 500 ms Logistic regression: (8 ms train + 1 ms test) * 20 iterations = 180 ms

Now, let's calculate the time taken for each model in the outer cross-validation loop: Neural network: 5 outer folds * (500 ms inner loop + 20 ms train + 5 ms test) = 2625 ms Logistic regression: 5 outer folds * (180 ms inner loop + 8 ms train + 1 ms test) = 945 ms Total time taken to compose the table: 2625 ms (neural network) + 945 ms (logistic regression) = 3570 ms

So, the correct answer is: Option C 3570.0 ms

5 References

Data Set:
California Housing Prices
kaggle.com/datasets/camnugent/california-housing-prices

Previous Study:
Predicting California Housing prices by using Machine Learning
Abhishek Shah
medium.com/@jwbtmf/predicting-california-housing-prices-by-using-machine-learning-43ca0dcde392

6 Student Responsibilities

Section	Jordan	Andy	Natalie
Regularized Regression Model	100%	0%	0%
Comparison Between Regression, ANN, and Baseline	0%	0%	100%
Classification	0%	100%	0%
Discussion	40%	30%	30%
Exam Problems	40%	20%	40%