Luke Ganalon
3/16/23
CPSC 393 - 02

HW 2 Technical Report

Introduction

The data I chose to use is the Boston housing dataset. This dataset reflects the housing market of Boston in the year 1978. The dataset is small in size with only 506 cases. From this dataset, I am attempting to predict the median value of owner-occupied homes based on the variables listed below:

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940
8. DIS - weighted distances to five Boston employment centres
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per $10,000
11. PTRATIO - pupil-teacher ratio by town
12. B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT - % lower status of the population

It is important to keep in mind that the dataset is small and that it includes data from 1978. The year of 1978 is significant, for the purchasing power of the dollar was very much different than today.

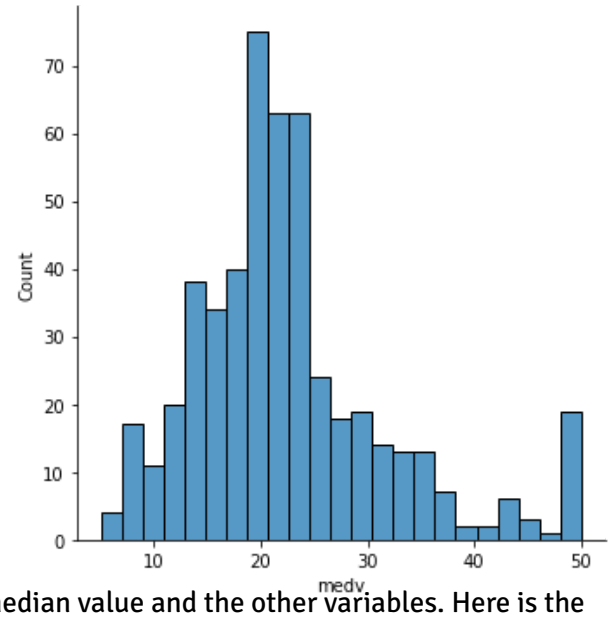
Analysis

Since my goal is to predict the median value of the homes, I first wanted to understand the data representing the median value of the homes. I discovered the following:

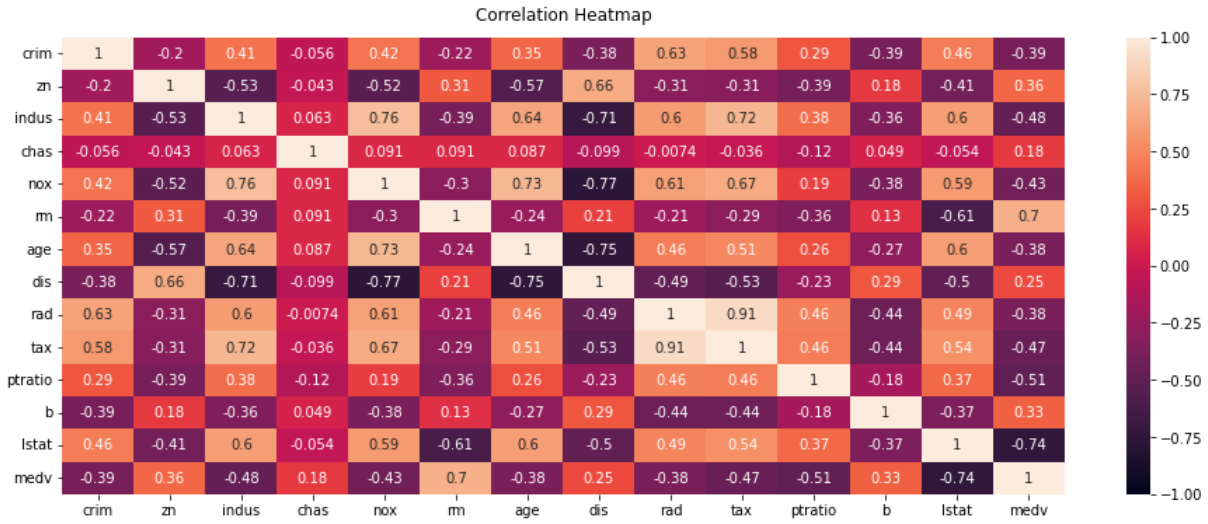
- The average median home value in Boston is 22,532.81 dollars.
- The range of median values of homes is 5,000 to 50,000 dollars

The distribution of the median home values can be seen below:

A majority of the houses' median values range between 14,000 and 24,000.



I then wanted to analyze the correlation between median value and the other variables. Here is the heatmap reflecting these relationships:



The three most impactful relational variables with median home value are:

- LSTAT - % lower status of the population (-0.74)
- RM - average number of rooms per dwelling (0.7)
- PTRATIO - pupil-teacher ratio by town (-0.51)

Methods

In terms of output units, or neurons, I began with 1000 and ended up with 1 output unit. The output shape is consistently decreasing as it approaches the final output of 1. There is no recurrence or convolutional layers. Due to 13 different variables, the input shape is 13. I tested both L2 regularization and L1 regularization. L1 regularization outperformed that of L2. The L1 kernel regularizer is performed at each hidden layer. Between these, batch normalization is used to normalize the activation vector for "each neuron's output follows a standard normal distribution across the batch"[1]. Due to the computational efficiency, I utilized the 'relu' activation function rather than the sigmoid activation. When selecting the optimizer, I tested both SGD and Adam. The SGD optimizer resulted in a much stronger MAE than Adam. Additionally, I set the learning rate to 0.001 rather than 0.01. By decreasing the learning rate, the model was able to learn more precisely despite the additional time added. The loss and metrics chosen were MSE and MAE, for the median value of homes are a continuous variable. Finally, 200 epochs were used instead of 100 to better improve the learning of the model.

```
model = kb.Sequential([
    kb.layers.Dense(1000, input_shape =[13]), #input
    kb.layers.Dense(800, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(650, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(500, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(450, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(350, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(225, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(100, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(50, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(30, kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(10,kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(8,kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(6,kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(4,kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(3,kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(2,kernel_regularizer = "l1"),
    kb.layers.BatchNormalization(),
    kb.layers.Dense(1, activation = "relu") #output
])

# compile model
model.compile(loss="mean_squared_error", optimizer=kb.optimizers.SGD(0.001),
  metrics= ["mean_absolute_error"])

#fit the model (same as SKlearn)
model.fit(X_train,y_train, epochs = 200, validation_data=(X_test, y_test))
```

[1] https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338

Results

MSE Train	21.99863175973955	MAE Train	3.2140399738703618
MSE Test	23.788568009691932	MAE Test	3.5219779809316

As shown, the errors increase on the test set. This means that the model is not overfit, but the values are still within reasonable limits and therefore not underfitting. Although I used MSE, RMSE is better. This is because the MSE is causing the value of US dollars to be squared as well in terms of interpretation. The RMSE would be the following:

RMSE Train	4.69	RMSE Test	4.88

To be approximately $4,700 off rather than $22,000 makes much more sense.

According to the data given, the average median value of a home is $22,533. Using the MAE metric on the test set, $3,522 is 15.6% of the average median value. This is a significant error because it is nearly 20% of the home's value. We would prefer a mean absolute error that is within 10% or less of the home's value. Additionally, the lowest median value of a home is $5,000. An error of $3,522 is more than half of that of the low-valued median home.

But consider purchasing power...

The Zestimate error rate today in Boston is 3.24%[2]. The median price of single-family homes in Boston increased from the previous year to $607,000.[3]

3.24% of $607,000 = $19,666.80 error of Zestimate in Boston

My results were a mean absolute error of $3,522 on the test set. Well, $3,500 in 1978 is equivalent in purchasing power to about $16,149.39 today[4]. Based on this information, $3,522 in 1978 is worth $16,250.90 today. This is $3,415.9 more accurate than Zillow in today's money. In 1978, that would be $740.32

So despite my previous criticism of the performance of the neural network model, the predictions are actually competitive to that of Zillow's Zestimate. It is also important to note that there were only 506 cases and 13 variables. Today, there is much more information and variables to consider.

---

2 https://www.upnest.com/1/post/zillow-estimate/#:~:text=If%20they%20base%20their%20estimated,error%20rate%20nationwide%20is%207.49%25.

3 https://www.noradarealestate.com/blog/massachusetts-housing-market/

4 https://www.in2013dollars.com/us/inflation/1978?amount=3500#:~:text=%243%2C500%20in%201978%20is%20equivalent,cumulative%20price%20increase%20of%20361.41%25.

Reflection

In this assignment, I was able to obtain a better grasp on neural networks and most importantly how to interpret them. I am now much more familiar with the various regularizations and other factors of a neural network that can affect the performance of the model such as the optimizer. Having a better conceptual understanding, I am much more comfortable with coding the actual model.

In the future, I would allow myself more time to experiment with different variations of the model. Although I did experiment with different variations, more variations would provide more insight as to how the model is performing. More importantly than the variations, I would do a better job of recording the variations in the model and their performance in order to better express the impacts of different variations of the model.