**K-fold cross validation**

Cross validation allows us to test our model on data it hasn't seen, to see how well the model performs when making predictions. Using cross validation can help identify if the model is overfitting the training data. When splitting data into training and test datasets there is a concern that important patterns found in the data may be left out of the training dataset. If these patterns are not present in the training dataset, the model won't be able to predict them in the test dataset. K-fold cross validation takes this into account by dividing the data into k folds. K-1 folds are used to train the model and the remaining fold is used for testing. This is repeated for k-1 models and every fold is used for testing once. This reduces bias and variance since we are essentially using the entire dataset for training and testing. We have chosen 5-folds for this project, a typical value chosen.

**Results for knn classification**

The textbook included mac address 00:0f:a3:39:e1:c0 and excluded 00:0f:a3:39:dd:cd. The reasoning found in the text indicates that these 2 have similar heat maps, indicating they are located close to one another. The first one (00:0f:a3:39:e1:c0) is chosen and 00:0f:a3:39:dd:cd is discarded. Our task was to determine if discarding 00:0f:a3:39:e1:c0 was the correct choice to make.

The table below shows results from running knn with 5-fold cross validation for different numbers of nearest neighbors. We selected to only run odd values of K to ensure there are no issues with a tie. We also chose to not run a value of one for K. This is to minimize the possibility of overfitting.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **mac Address** | **Accuracy for K Nearest Neighbors** | | | | | **Selected K** |  |
| K=3 | K=5 | K=7 | K=9 | K=11 | Test |
| Keep 00:0f:a3:39:e1:c0 | 71.99% | 71.81% | 71.02% | 70.03% | 69.08% | 3 | 0.30% |
| Keep 00:0f:a3:39:dd:cd | 71.39% | 71.67% | 71.05% | 70.10% | 69.38% | 5 | 0.69% |
| Keep all 7 | 75.58% | 75.57% | 74.69% | 73.78% | 72.80% | 3 | 0.43% |

Keeping 00:0f:a3:39:e1:c0 turns out to be the better choice, when attempting to decide between 00:0f:a3:39:e1:c0 and 00:0f:a3:39:dd:cd. With K=5 an accuracy of 72.13% is achieved. 6 mac addresses were chosen because of information provided about the dataset. However, using all mac addresses and 3 neighbors achieves a higher accuracy of 75.58%. It would be advisable to discuss this finding with the data provider. If there are truly only 6 mac addresses it would not be appropriate to use all 7.

**Multi-Target Regression**

A drawback for using a classification modeling technique for this dataset is that combinations of X and Y that are found in the test dataset, but not the training dataset, do not have the change of being classified correctly. A classification model can only classify observations based on the classes available in the training dataset. The training dataset provided had all X and Y values rounded to the nearest integer. The test dataset had many unrounded values. This issue and the fact that we really have 2 target variables (X and Y), lead us to research multi-target regression models.

The pls package in R provides a multi-target Partial Least Squares Regression model.[[1]](#footnote-1) Partial Least Squares decomposes a matrix of independent variables X and a matrix of target variables Y. While we could have evaluated the number of optimal components needed for the problem here, we choose to force the model to include all components (or variables). The ability to use a matrix Y is what we focus on.

Review evaluation metric here, but we need to discuss this.

|  |  |  |
| --- | --- | --- |
| **mac Address** |  | |
| **Training** | **Test** |
| Keep 00:0f:a3:39:e1:c0 | 1.68 | 3.42 |
| Keep 00:0f:a3:39:dd:cd | 1.86 | 3.22 |
| Keep all 7 | 2.61 | 3.20 |

1. https://www.jstatsoft.org/article/view/v018i02 [↑](#footnote-ref-1)