CS 372M Programming Assignment #3

Name: Carter Young

Due: 11 March 2024

1. My dataset is a repository of home and apartment prices with information including area, # bathrooms, # bedrooms, amenities, and whether pets are allowed. played since the founding of the league. Link: https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified. I will be using the version with 10k rows.

2. My goal in analyzing this dataset is to be able to predict two things whether a dwelling's price is materially and reliably impacted by its amenities and features, or whether it is dictated by location or some other feature.

3. A lot of data pre-processing is necessary to make this data appropriate and fit to analyze. Namely and primarily, there are many fields which are not essential or fit for this task.

   Our first task is removing numerous columns. Things like ID, category, title, or currency will not have a material impact on this analysis (see full list of removed categories in proj3_data_preprocess.ipnyb).

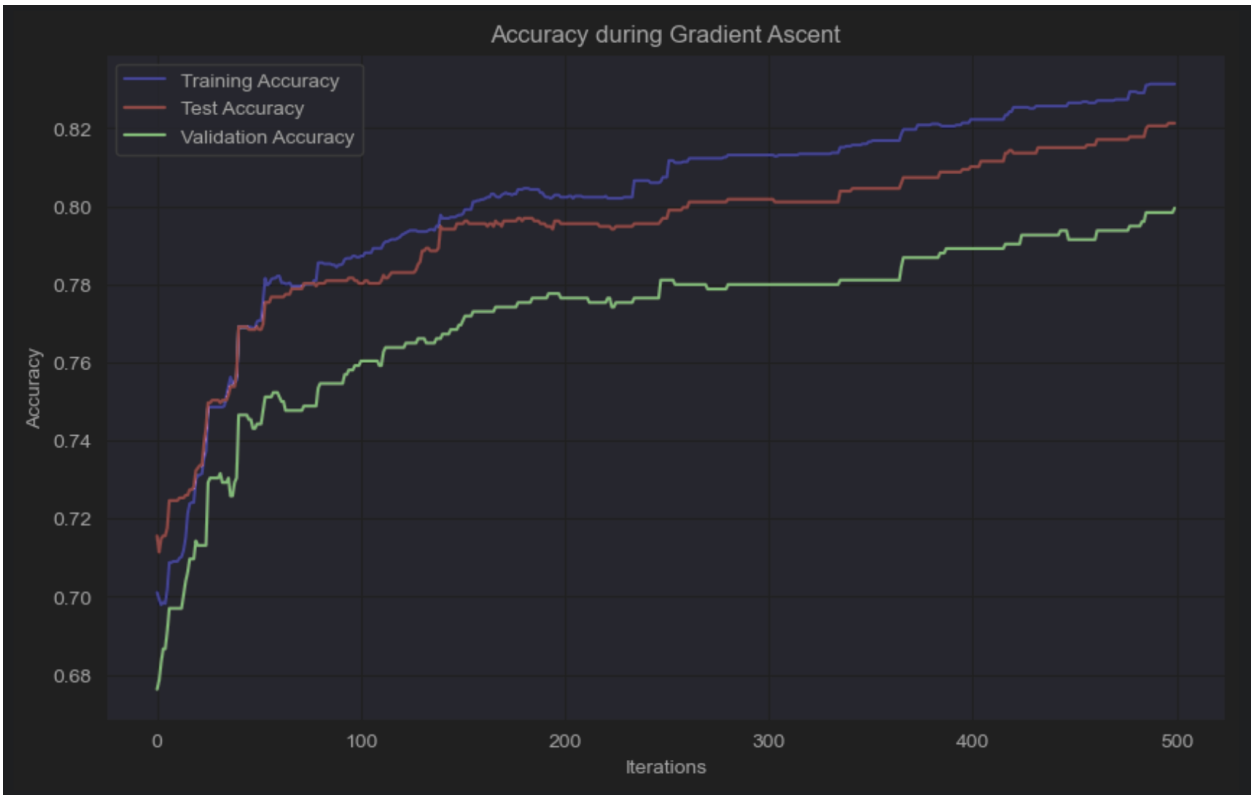
   Other, more essential columns had to be checked for null values. Any row containing 'null' in the columns 'price', 'pets_allowed', 'bedrooms', 'bathrooms', or 'square_feet' were removed. Although 'amenities' is being analyzed, I determined that the lack of a description of amenities can in and of itself signal no amenities.

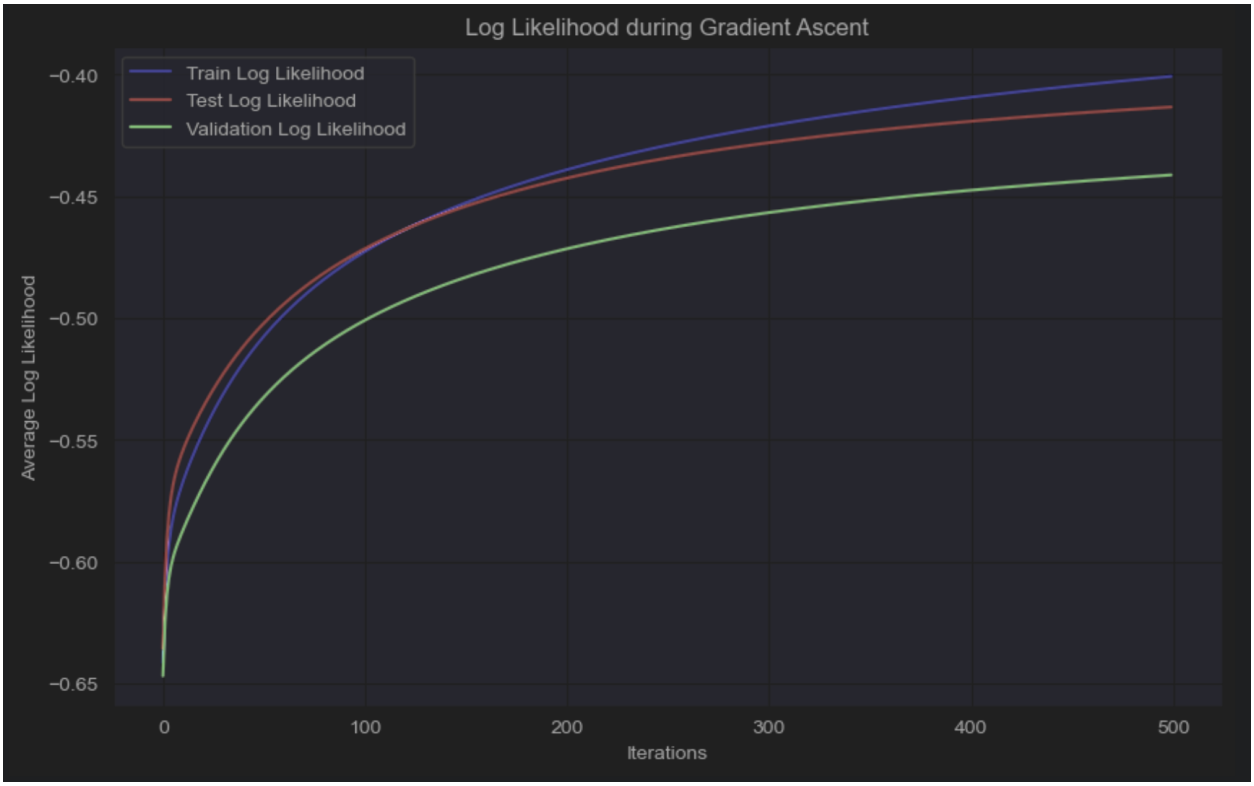
From this point on, I will focus on specific columns and detail the cleaning needed for them:

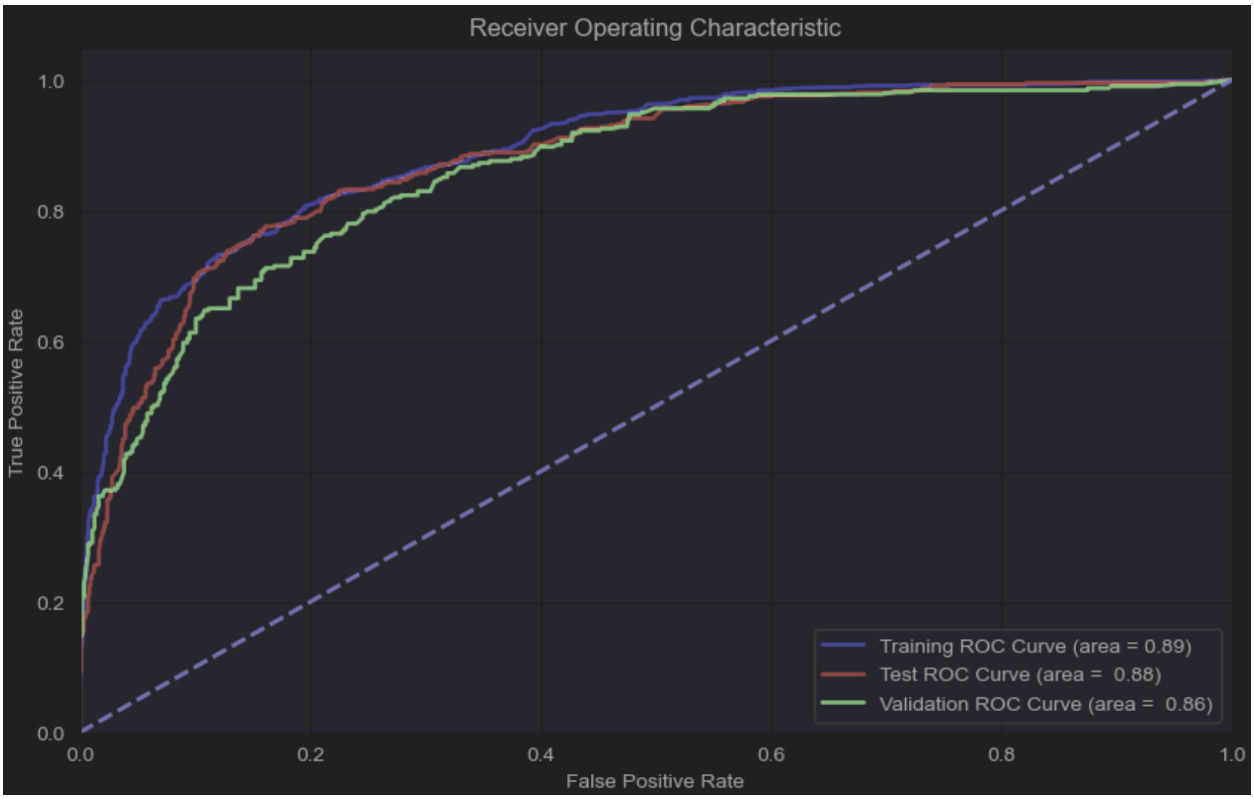
- Price:
  - Non-monthly price types were removed
  - The mean price was found
    - Dwellings below the mean were assigned a ‘0’
    - Dwellings above the mean were assigned a ‘1’
- Pets_Allowed:
  - Cleaning was done to resolve the naming ‘Cats,Dogs’
  - If no pets are allowed, assigned ‘0’. Otherwise, assigned ‘1’.
- Bedrooms:
  - If ≤3, assigned ‘0’. Otherwise, assigned ‘1’.
- Bathrooms:
  - Halves removed for ease of categorizing (i.e., if = 2.5, changed to 2)
  - If ≤3, assigned ‘0’. Otherwise, assigned ‘1’.
- Square_Feet:
  - The mean square_feet was found
    - Dwellings below the mean were assigned a ‘0’
    - Dwellings above the mean were assigned a ‘1’
- Amenities:
  - If ‘null’, assigned ‘0’
  - Otherwise, assigned ‘1’

4. Accuracies

Linear Regression:


Accuracy during Gradient Ascent
Training Accuracy
Test Accuracy
Validation Accuracy
0.82
0.80
0.78
0.76
0.74
0.72
0.70
0.68
Accuracy
0
100
200
300
400
500
Iterations


Log Likelihood during Gradient Ascent
Train Log Likelihood
Test Log Likelihood
Validation Log Likelihood
−0.40
−0.45
−0.50
−0.55
−0.60
−0.65
Average Log Likelihood
0
100
200
300
400
500
Iterations



Decision Tree:

```
Training Accuracy of the Decision Tree: 0.9144
Testing Accuracy of the Decision Tree: 0.8352
Validation Accuracy of the Decision Tree: 0.8157
```

Random Forests:

```
Training Accuracy of the Random Forest: 0.9144
Testing Accuracy of the Random Forest: 0.8345
Validation Accuracy of the Random Forest: 0.8203
```

Support Vector Machine:

Linear Kernel:

```
Linear Kernel SVM Classifier Report for Training Set:
              precision    recall  f1-score   support

           0       0.90      0.95      0.92      2227
           1       0.90      0.82      0.85      1289

    accuracy                           0.90      3516
   macro avg       0.90      0.88      0.89      3516
weighted avg       0.90      0.90      0.90      3516

Linear Kernel SVM Classifier Report for Testing Set:
              precision    recall  f1-score   support

           0       0.87      0.86      0.86       960
           1       0.72      0.75      0.74       478

    accuracy                           0.82      1438
   macro avg       0.80      0.80      0.80      1438
weighted avg       0.82      0.82      0.82      1438

Linear Kernel SVM Classifier Report for Validation Set:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85       545
           1       0.74      0.75      0.75       323

    accuracy                           0.81       868
   macro avg       0.80      0.80      0.80       868
weighted avg       0.81      0.81      0.81       868
```

RBF Kernel:

```
RBF Kernel SVM Classifier Report for Training Set:
              precision    recall  f1-score   support

           0       0.84      0.97      0.90      2227
           1       0.92      0.68      0.78      1289

    accuracy                           0.86      3516
   macro avg       0.88      0.82      0.84      3516
weighted avg       0.87      0.86      0.86      3516

RBF Kernel SVM Classifier Report for Testing Set:
              precision    recall  f1-score   support

           0       0.81      0.91      0.86       960
           1       0.76      0.58      0.66       478

    accuracy                           0.80      1438
   macro avg       0.78      0.74      0.76      1438
weighted avg       0.79      0.80      0.79      1438

RBF Kernel SVM Classifier Report for Validation Set:
              precision    recall  f1-score   support

           0       0.78      0.92      0.84       545
           1       0.81      0.55      0.66       323

    accuracy                           0.79       868
   macro avg       0.79      0.74      0.75       868
weighted avg       0.79      0.79      0.77       868
```

Neural Networks:

The accuracies above are the average for each dataset across 10 seeds.

5. Under-fitting/over-fitting/convergence

For all models, the training accuracy is significantly higher than the testing and validation accuracies. This discrepancy suggests all models are overfitting to the training data. The drop in performance in the testing and validation sets insinuates the model is not as good at learning or dealing with data it has not seen before. For the NN and Regression models, there is extremely rapid learning at the beginning, and convergence occurs very quickly. While there are gaps between the training/validation/test sets, they are not very large. This suggests only mild overfitting. Stopping training earlier (which I have already implemented), or utilizing dropout or regularization would help in improving accuracy and reducing computational overhead.