Technical Report: Final Project DS 5220: Supervised Machine Learning and Learning Theory

Real Estate Sales 2001-2020

Team Members: Ikonkar Kaur Khalsa and Matthew Gregorio

Khoury College of Computer Sciences
Data Science Program

khalsa.i@northeastern.edu
gregorio.m@northeastern.edu

Github Link

July 25, 2024

1 Abstract

This report presents a comprehensive analysis aimed at developing accurate and interpretable machine learning models for predicting house sale amounts and classifying property affordability. Utilizing an extensive dataset from Kaggle, encompassing over a million real estate transactions across two decades, the study evaluates both regression and classification models to identify the most effective predictive approach. The methodology involves preprocessing, exploratory data analysis, and training various algorithms, including Linear, Lasso, and Ridge for regression, and Logistic, Decision Trees, Gaussian Naive Bayes, and Neural Networks for classification. The comparative analysis reveals that the Decision Tree model, specifically the second variant, outperformed others in terms of model evaluation scores and interpretability, providing significant insights into the impact of features such as Longitude on sale amounts. Regression models demonstrated inferior performance, indicating that classification approaches were more suitable for this dataset. The study suggests that employing a meta-data regression analysis approach, as proposed in relevant literature, could streamline model selection and evaluation. Comparisons with findings from smaller datasets highlight the efficiency of the Decision Tree model over Random Forest, underscoring its practicality for large-scale datasets. Overall, the report concludes that the Decision Tree model offers the most reliable predictions and insights for the given dataset, marking a valuable contribution to real estate predictive analytics.

Contents

1 Abstract 2

2 List of Figures and Tables 4

3 Introduction 5

4 Literature Review 5

4.1 First Article: Selecting Machine Learning Algorithms Using Regression Models . . . 5

4.2 Second Article: Comparison of Machine Learning Algorithms in Data classification . . . 6

5 Methodology 6

5.1 Data Collection . . . 6

5.2 Data Preprocessing . . . 6

5.3 Exploratory Data Analysis . . . 7

5.4 Regression Analysis . . . 9

5.5 Classification Analysis . . . 9

6 Results 9

6.1 Linear Regression . . . 10

6.2 Ridge Regression . . . 10

6.3 Lasso Regression . . . 11

6.4 Logistic Regression . . . 11

6.5 Decision Tree #1 . . . 11

6.6 Decision Tree #2 . . . 13

6.7 Simple Neural Network for Regression . . . 15

6.8 Simple Neural Network for Classification . . . 15

6.9 Gaussian Naive Bayes . . . 15

6.10 Support Vector Classifier, K-Nearest Neighbor, Random Forest Classifier 16

7 Discussion 16

8 Conclusion 16

8.1 Statement of Contribution . . . 17

9 References 17

2 List of Figures and Tables

List of Figures

1 Missing Values After Label Encoding 7

2 Visualization of features that did not have much direct initial significance to the Sale Amount. 8

3 Two features that had significance on Sale Amount in the initial run. . . 9

4 Linear Regression: Actual vs Predicted Values 10

5 Ridge Regression: Actual vs. Predicted Values 10

6 Lasso Regression: Actual vs. Predicted Values 11

7 Model Evaluation 11

8 Decision Tree #1 Model Evaluation 12

9 The cross-validation scores has a mean of about 0.887, with some variability across the folds. This suggests that the model performs well and consistently across different subsets of the data. However, the initial evaluation metrics indicate that despite the high precision and overall accuracy, the model has a low recall, implying it misses a significant number of actual positive instances. 12

10 Decision Tree Model #1 12

11 The feature importance for the Decision Tree Model #1 brought out a striking evaluation. Before in the EDA, Longitude really had no significant on Sale Amount, however in this visualization it seems that it does! . . . 12

12 Confusion Matrix Decision Tree Model #1: The model is good at identifying the negative instances but misses a substantial number of positive instances. Overall accuracy is 80%. 13

13 ROC AUC Decision Tree Model #1 13

14 Lastly for the initial Decision Tree run, we did a Grid Search Cross Validation. It provided us with the optimal model configuration and the model achieved an accuracy score of about 81%. 13

15 Decision Tree Model #2 14

16 The scores increased in every aspect. There is a higher percentage of accuracy precision, recall and f1 compared to the first Decision Tree model. 14

17 The Cross Validation Scores also improved. The grid search has slightly improved the model's performance with a higher mean score and more consistent results across different folds. This indicates that the selected hyper parameters positively impacted the model's stability and accuracy. 14

18 Decision Model Tree #2: Feature Importance, Confusion Matrix, and ROC AUC. 14

19 The results here indicate a lot of errors in the predictions. The model here also fit poorly. 15

20 The model has a decent overall accuracy. However, it still does poorly on detective positive instances which then led to a low recall and f1 score. . 15

21 Gaussian NB Confusion Matrix 15

3 Introduction

The primary objective of this project is to create machine learning models that give accurate and interpretable predictions for unseen houses on the market. It attempts to do this for a continuous value of sale amounts and a classification of affordability and non-affordability. The aim is to select the best predictive model for both major predictive categories. The dataset chosen was taken from Kaggle and it was used for its extensive amount of mostly clean and complete data. This dataset was chosen due to it containing over a million transactions that explored two decades of real estate data. Link to the Kaggle dataset is here. This project includes several machine learning algorithms which were trained for both regression and classification. Comparisons were made between Linear, Lasso, and Ridge for regression. Classification, on the other hand, had models such as Logistic, Decision Trees, Gaussian Naive Bayes, and Neural Networks for comparison. The report is structured to go step by step of the project which includes: preprocessing, exploratory data analysis, methodology of different machine learning algorithms used, results, discussion and conclusion.

4 Literature Review

4.1 First Article: Selecting Machine Learning Algorithms Using Regression Models

Here is the link to the first article. The research of this study is focused primarily on finding the best model to find and ultimately optimize given a variety of datasets. Its aim is to bypass the intense and exhaustive process of trying every model and looking at its metrics on performance. Meta-learning is its proposed solution. Past performance of models is used to assess which model will perform. The meta data are statistical features with a fixed number of features such as mean, median, standard deviation, and skewness.This proposed method keeps the underlying importance of the features and seeks to bypass the curse of dimensionality. Applying regression trees leads to a ranked list of the best models dropping the least efficient ones by the SAR metric which is a combination of accuracy, AUC, and RMSE yielding a more robust solution. Once narrowed down hyperparameter tuning can be introduced to optimize the lead models. Thus, each and every model need not be trained and evaluated, saving computational capital.

There are a variety of methodologies employed in this study. The combination of statistical and non-statistical features as meta knowledge about a dataset is a highlight. Applying decision tree models to classify which regression model has the most predictive power is another point in its methodology. Both feature selection, the process of reducing dimensionality, and feature extraction, the technique to combine features into a singular new feature thus preserving valuable information, are utilized to this end. Hyperparameter tuning is the final process by which an optimal model is found from the best ranked candidates.

The findings of this article is that there is a save in computational complexity and the model selection process overall is improved with a more robust 'accuracy' metric. This approach works equally well for actual and artificial datasets both of which were tested by this proposed methodology.

The gap identified in the conclusion of the study is that there is a minimum of 4 features necessary to each dataset to construct the meta knowledge. Thus datasets with less than this number of features would ultimately be discarded in the process. Discarded data may limit the true predictive power of the optimal model found during model search. Also, statistical summaries may lose some of the important characteristics of the data which is an obvious drawback.

4.2 Second Article: Comparison of Machine Learning Algorithms in Data classification

The link to the second article is here.This study seeks to find the best classifier algorithms for predicting a binary outcome utilizing 6 models: Logistic Regression, Decision Tree, Naive Bayes, k-Nearest Neighbors, Support Vector Machine, and Random Forest. These models are evaluated, in the context of this research, for heart disease and hepatitis disease detection but have applications to many other datasets. The datasets collected in this instance are from UCI which is known for its ‘extensiveness in data, its completeness, and accuracy.’ The metrics used in this study are accuracy, precision, recall, and f-measure which are all defined in the article.

The methodology breaks down into 4 distinct phases. Data collection through UCI is the first step. Preprocessing the data is next and split into training and testing sets. All 6 classifier models are then trained. Lastly each is evaluated with the aforementioned metrics to compare and contrast the models for predictive fidelity.

A comprehensive review of all these models trained on these datasets shows that Random Forest yields the most effective model in terms of accuracy. Other metrics support this finding indicating a robust and accurate model.

A gap in this research is the lack of an extensive dataset. The dataset utilized is only a few hundred entries with 14-20 features. In the medical world there are often many more features collected on patients and quite more numerous is the number of patients that data is collected on. This could dramatically affect the results of which model works best. There is also no mention of optimizing, via hyperparameter tuning, the most desirable model given the metrics.

5 Methodology

5.1 Data Collection

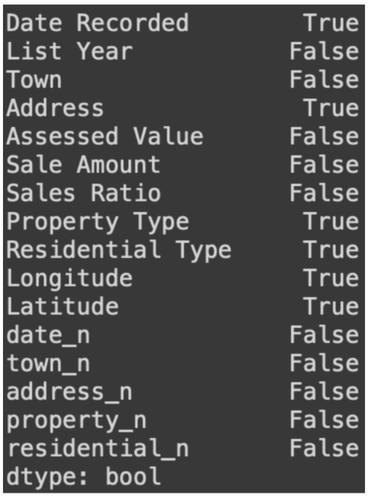
The dataset was taken from Kaggle and it was used for its extensive amount of mostly clean and complete data. This dataset was chosen due to it containing over a million transactions that explored two decades of real estate data and over 1 million transactions recorded. Link to the Kaggle dataset is here.

5.2 Data Preprocessing

The original dataset structure includes 11 features: Date Recorded, List Year, Town, Address, Sale Amount, Sales Ration, Property Type, Residential Type, Longitude, and Latitude. Before doing anything to the dataset, we first dropped any duplicate data that may be in the dataset. Doing this enabled us to drop 8,705 duplicate entries. The original dataset had 1,054,159 rows, and after dropping the duplicates the dataset had

1,045,454 rows. We then label-encoded the categorical values into numerical using one-hot encoding. For this step, 4 features (Date Recorded, Town, Address, Property Type, Residential Type) from the original dataset were label-encoded to “date_n”, “town_n”, “address_n”, “property_n”, and “residential_n”. After label-encoding, we checked to see if there were any missing values.

In the output, we ignored the original categorical values since we only want to keep the new label-encoded features. Out of the list, the only missing values that had to change were “Longitude” and “Latitude’. These missing values in these two features were filled with the mean. We then rechecked to see if there still were missing values in these features and there were none. After the exploratory data analysis, which will be explained in the next section, we also decided to use normalization for every numerical feature and numerically converted features.



```
Date Recorded        True
List Year           False
Town                False
Address              True
Assessed Value      False
Sale Amount         False
Sales Ratio         False
Property Type        True
Residential Type     True
Longitude            True
Latitude             True
date_n              False
town_n              False
address_n           False
property_n          False
residential_n       False
dtype: bool
```

Figure 1: Missing Values After Label Encoding

For the rest of the project these are the features we went with as well as the target and label:

Features: Date Recorded, List Year, Town, Address, Assessed Value, Sales Ratio, Property Type, Residential Type, Longitude, Latitude

Target variable: Sale Amount

Label: Affordable

5.3 Exploratory Data Analysis

After preprocessing the data we defined x and y and also split the data into training and testing data. In this case, the x dataframe included “List Year”, “Assessed Value”, “Longitude”, “Latitude”, “date_n”, “town_n”, “address_n”, “property_n”, and “residential_n”. The y dataframe only included the “Sale Amount” feature. We kept the test size as 0.2. During this moment in the project, we wanted to see how the features related to the Sale Amount feature. So we did this by visualizing every feature and its relation to Sale Amount.

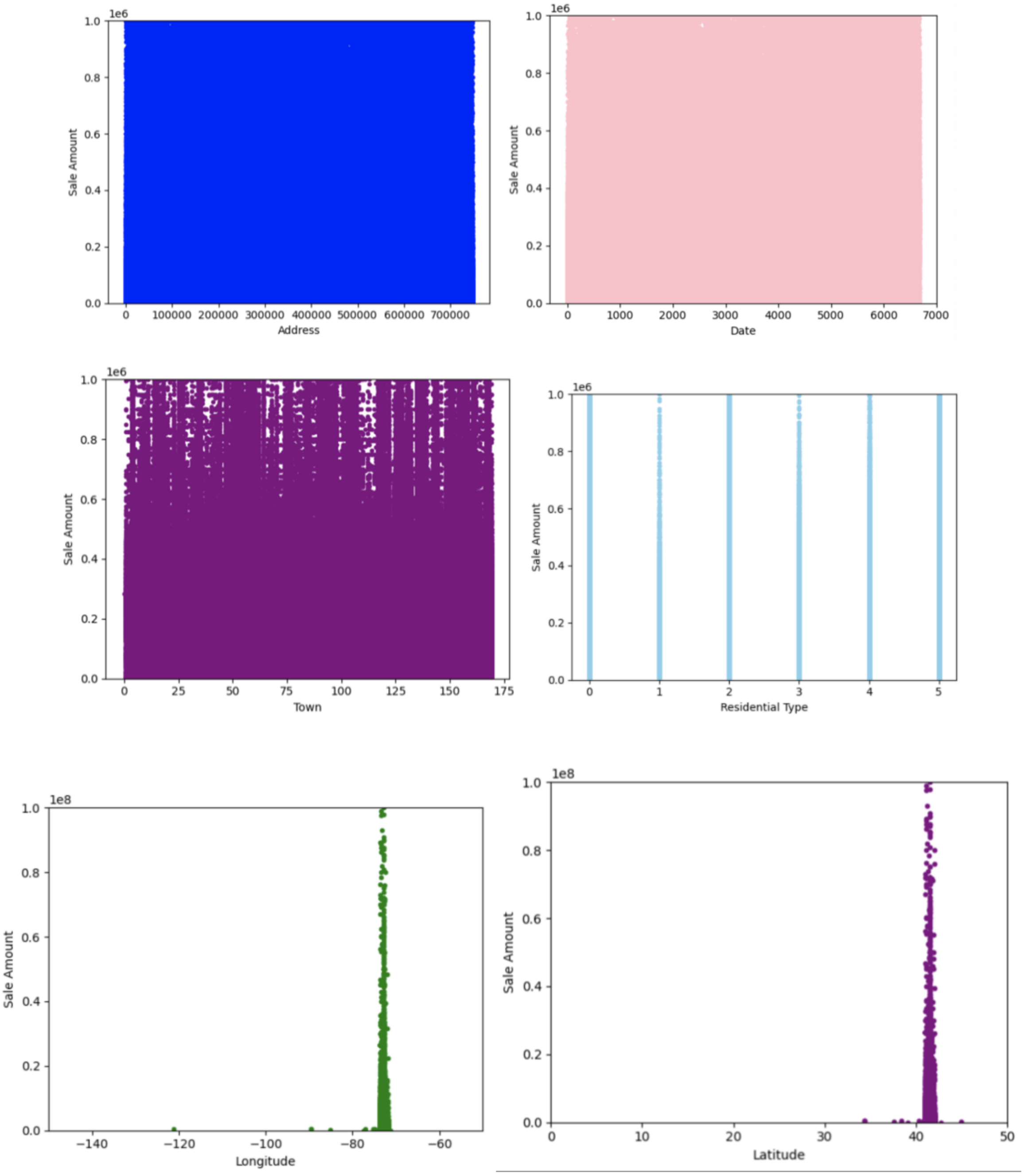


Figure 2: Visualization of features that did not have much direct initial significance to the Sale Amount.

These features that were visualized showed that there was no relationship with Sale Amount initially. However, two features during the EDA were shown to have great significance to Sale Amount. The first feature was “Property Type”. The visual indicated that property type number 5 had a cheaper sale amount. Since this category was label encoded we mapped property type and the output indicated that number 5 was a ‘Residential’ property type. The second feature was Assessed Value, which also showed some relationship to Sale Amount. The visuals for this in down below.

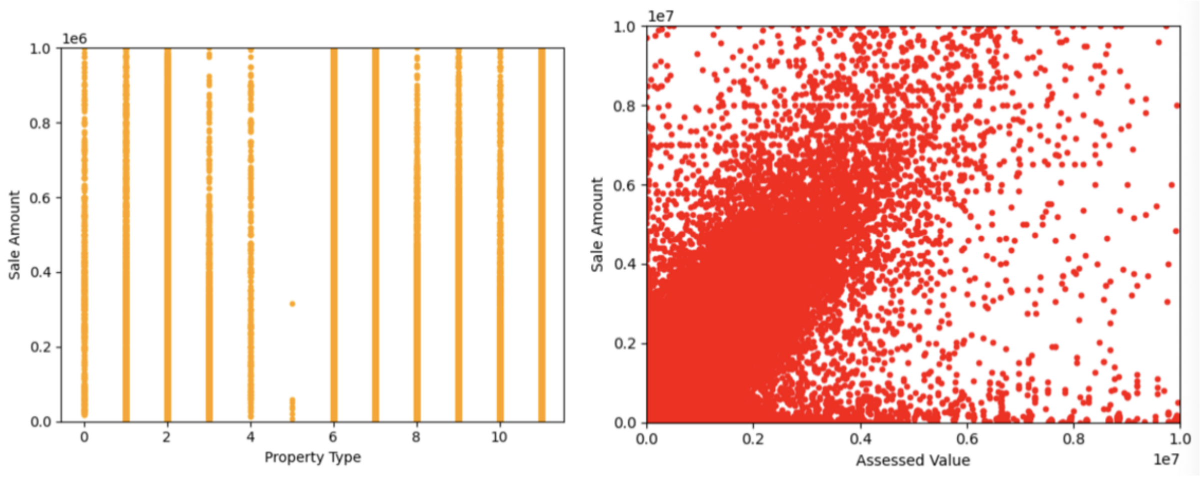


Figure 3: Two features that had significance on Sale Amount in the initial run.

5.4 Regression Analysis

We conducted three regression analysis: Linear, Ridge and Lasso. For Linear we fit the model, identified the intercept and coefficients and finally did a prediction which was then followed by evaluating the model by using mean squared error, mean absolute error, and r-squared. Lastly, we plotted the linear regression of the actual vs predicted values. Next, we performed Ridge by importing Ridge through sklearn.linear_model. We tested the prediction, conducted the evaluation of the model by doing mse, mae, and r-squared. We then also plotted Ridge. Lastly, we followed the same steps for Lasso.

5.5 Classification Analysis

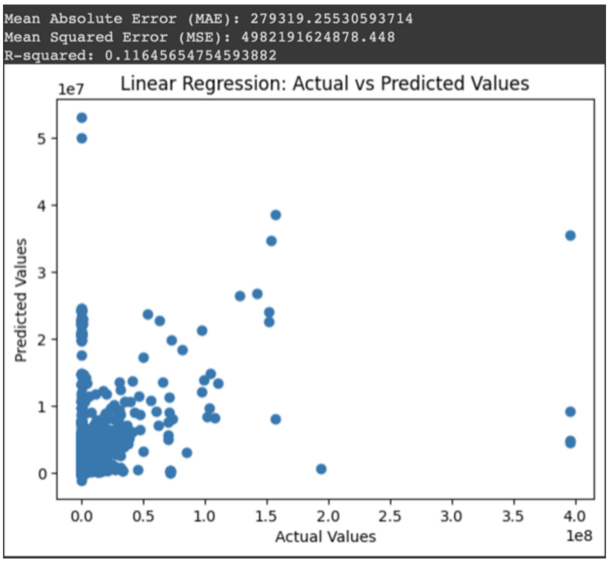
For classification analysis, we conducted 9 machine learning algorithms. Out of 9, there were 3 that we tried running (Random Forest Classifier, SVM, and KNN) that were too computationally expensive and was not able to run. So in total we were successfully able to conduct 6 fully functioning models for classification analysis. These models and their results will be discussed in the next section. We tested out all models, created visualizations, and evaluated each models' performance and accuracy.

6 Results

11 Machine Learning Algorithms were used for this dataset, 3 of them which we were not able to complete due to size of the data, as the dataset did not agree with the methodologies in theory and in practice. Linear, Lasso, Ridge, and Simple Neural Network (regression) were used for Regression, while Logistic, Decision Tree, Simple Neural Network (classification), and Gaussian Naive Bayes were used for classification. The metrics that were calculated and compared for regression and classification differed. For Regression we used mean absolute error, mean squared error, and r squared value. For Classification we used accuracy, precision, recall, and f1 score.

6.1 Linear Regression

According to the calculations below in the figure, it indicates that the linear model is in need of improvement. This could be improved by either doing some feature engineering, checking if there are any data issues, or adjusting the model's complexity. The MAE is showing that on average the model predictions are off by 279,319.25 units. The MSE has a lot more errors than MAE. The calculations for MSE indicate that there are significant errors in the model's predictions. Lastly, the r-squared is indicating that this model does not fit with the dataset.

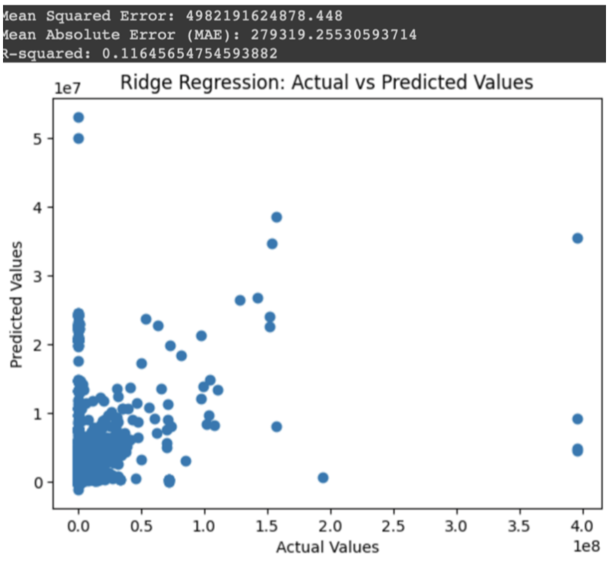


```
Mean Absolute Error (MAE): 279319.25530593714
Mean Squared Error (MSE): 4982191624878.448
R-squared: 0.11645654754593882
```

Figure 4: Linear Regression: Actual vs Predicted Values

6.2 Ridge Regression

The metrics here are identical to Linear Regression which means that Ridge also did not improve the model performance.

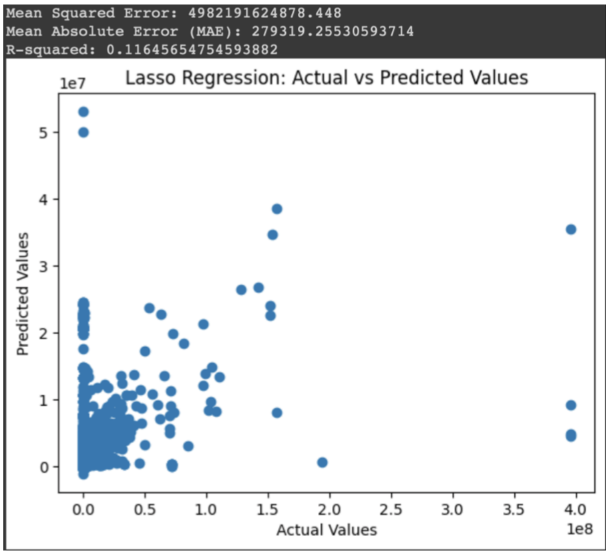


```
ean Squared Error: 4982191624878.448
ean Absolute Error (MAE): 279319.25530593714
R-squared: 0.11645654754593882
```

Figure 5: Ridge Regression: Actual vs. Predicted Values

6.3 Lasso Regression

The metrics for Lasso was also identical to Linear regression which also indicates that Lasso did not improve the model performance.

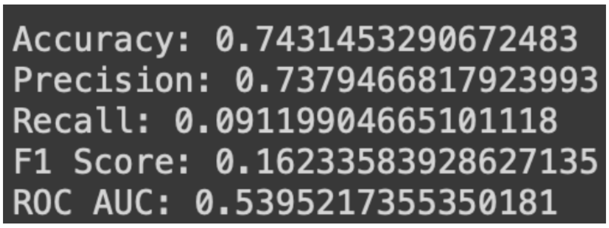


```
Mean Squared Error: 4982191624878.448
Mean Absolute Error (MAE): 279319.25530593714
R-squared: 0.11645654754593882
```

Figure 6: Lasso Regression: Actual vs. Predicted Values

6.4 Logistic Regression

The following image shows the model evaluation calculations for Logistic Regression. The accuracy for the Logistic Regression model is 74.31%, indicating that the model was correct about 74.31% of the time. The precision of 73.79% indicates that, of all the instances predicted as positive, 73.79% were true positives, suggesting a good positive predictive value. However, the recall is only 9.12%, meaning the model captures only 9.12% of the actual positive instances, missing many true positives. The F1 Score is 16.23%, reflecting a poor balance between recall and precision, likely due to dataset imbalance. Lastly, the ROC AUC is 53.95%, indicating that the model's ability to differentiate between classes is just slightly better than random guessing.

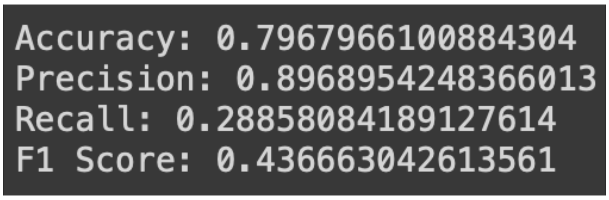


```
Accuracy: 0.7431453290672483
Precision: 0.7379466817923993
Recall: 0.09119904665101118
F1 Score: 0.16233583928627135
ROC AUC: 0.5395217355350181
```

Figure 7: Model Evaluation

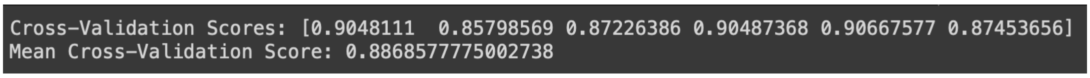
6.5 Decision Tree #1

The first decision tree model had high accuracy and precision but a really low recall score as well as F1. This indicated that while the model is good at predicting the positive classes when it makes a positive prediction, it misses several actual positive cases. The F1 score shows the imbalance between precision and recall.



```
Accuracy: 0.7967966100884304
Precision: 0.8968954248366013
Recall: 0.28858084189127614
F1 Score: 0.436663042613561
```

Figure 8: Decision Tree #1 Model Evaluation



```
Cross-Validation Scores: [0.9048111  0.85798569 0.87226386 0.90487368 0.90667577 0.87453656]
Mean Cross-Validation Score: 0.8868577775002738
```

Figure 9: The cross-validation scores has a mean of about 0.887, with some variability across the folds. This suggests that the model performs well and consistently across different subsets of the data. However, the initial evaluation metrics indicate that despite the high precision and overall accuracy, the model has a low recall, implying it misses a significant number of actual positive instances.

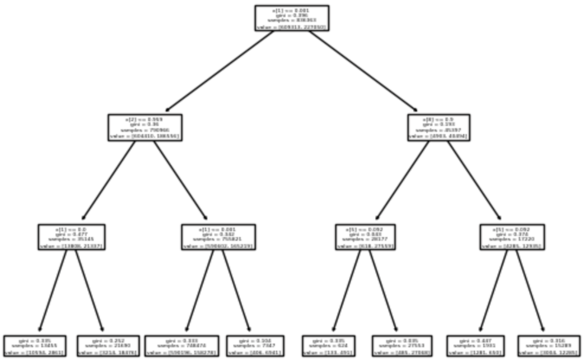


Figure 10: Decision Tree Model #1

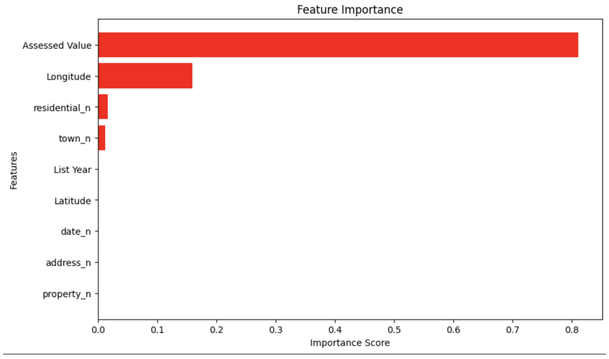


Figure 11: The feature importance for the Decision Tree Model #1 brought out a striking evaluation. Before in the EDA, Longitude really had no significant on Sale Amount, however in this visualization it seems that it does!

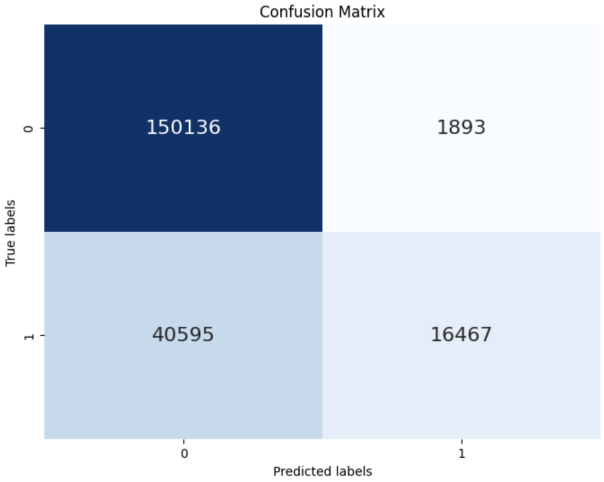


Figure 12: Confusion Matrix Decision Tree Model #1: The model is good at identifying the negative instances but misses a substantial number of positive instances. Overall accuracy is 80%.

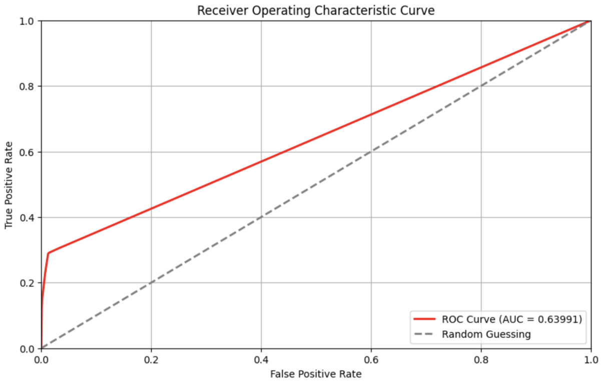
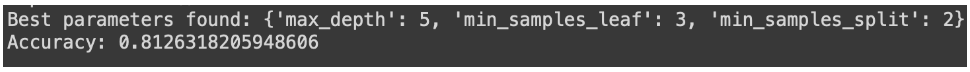


Figure 13: ROC AUC Decision Tree Model #1



```
Best parameters found: {'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 2}
Accuracy: 0.8126318205948606
```

Figure 14: Lastly for the initial Decision Tree run, we did a Grid Search Cross Validation. It provided us with the optimal model configuration and the model achieved an accuracy score of about 81%.

6.6 Decision Tree #2

After conducting the first Decision Tree Model, we decided to do another Decision Tree based off the best parameters that were found from the Grid Search Cross Validation from the first Decision Tree Model.

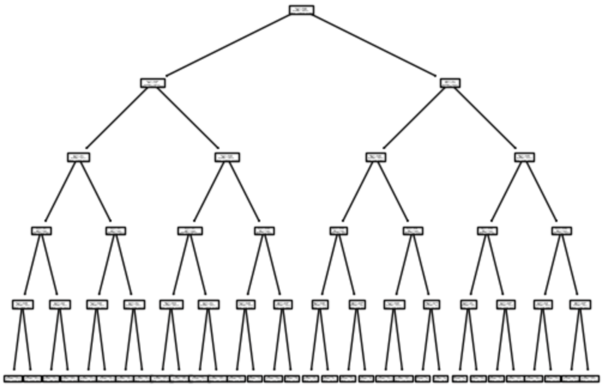
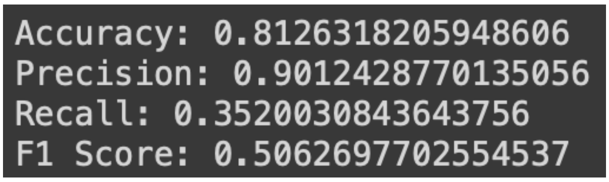


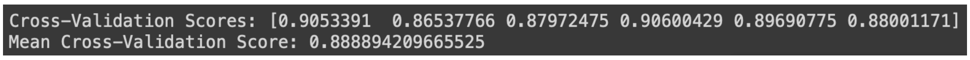
Figure 15: Decision Tree Model #2

The accuracy scores improved drastically which can be seen in the figure below.



```
Accuracy: 0.8126318205948606
Precision: 0.9012428770135056
Recall: 0.3520030843643756
F1 Score: 0.5062697702554537
```

Figure 16: The scores increased in every aspect. There is a higher percentage of accuracy precision, recall and f1 compared to the first Decision Tree model.



```
Cross-Validation Scores: [0.9053391  0.86537766 0.87972475 0.90600429 0.89690775 0.88001171]
Mean Cross-Validation Score: 0.888894209665525
```

Figure 17: The Cross Validation Scores also improved. The grid search has slightly improved the model's performance with a higher mean score and more consistent results across different folds. This indicates that the selected hyper parameters positively impacted the model's stability and accuracy.

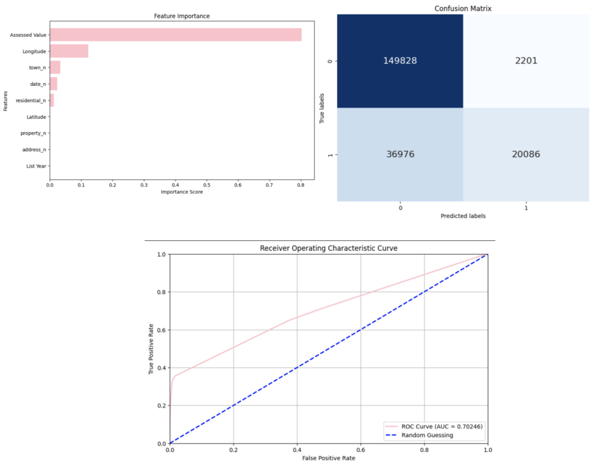
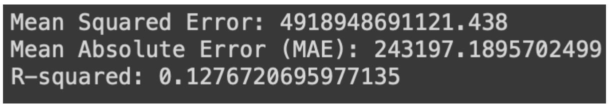


Figure 18: Decision Model Tree #2: Feature Importance, Confusion Matrix, and ROC AUC.

6.7 Simple Neural Network for Regression

The loss and MSE on the training data had a value of 4918952853504.0000, which is really high. This indicates that this model makes predictions that are far from the actual target value in the training data.

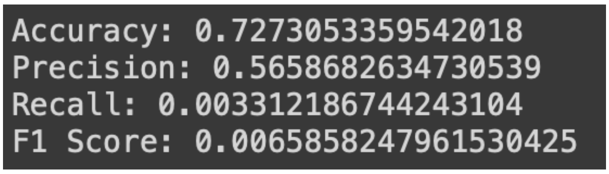


```
Mean Squared Error: 4918948691121.438
Mean Absolute Error (MAE): 243197.1895702499
R-squared: 0.1276720695977135
```

Figure 19: The results here indicate a lot of errors in the predictions. The model here also fit poorly.

6.8 Simple Neural Network for Classification

The evaluation for Simple Neural Network for Classification was a lot better than the SNN for regression.



```
Accuracy: 0.7273053359542018
Precision: 0.5658682634730539
Recall: 0.003312186744243104
F1 Score: 0.0065858247961530425
```

Figure 20: The model has a decent overall accuracy. However, it still does poorly on detective positive instances which then led to a low recall and f1 score.

6.9 Gaussian Naive Bayes

The evaluation for GausianNB performed well for negative classes but struggled with the positive classes. This led to a low recall and f1 score for negative classes. Overall the accuracy level is okay but the ability to detect the positive classes is limited for the model.

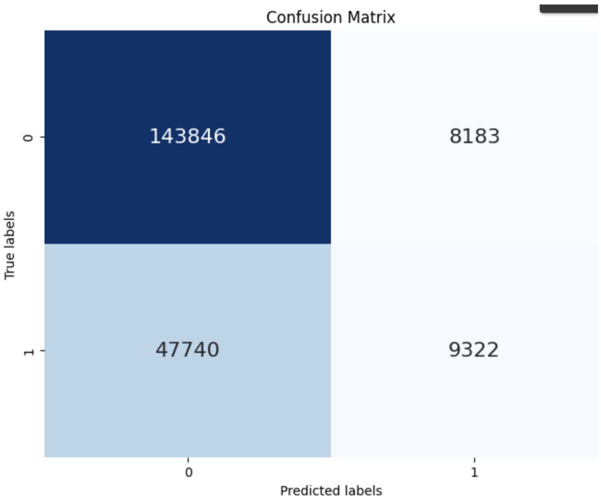


Figure 21: Gaussian NB Confusion Matrix

6.10 Support Vector Classifier, K-Nearest Neighbor, Random Forest Classifier

We attempted to do an SVC for our project. The dataset is too large to do SVC and the computation time was too long and never ran. We also attempted to do KNN, however it was too computationally expensive and utilized too much memory. This also did not run. The same goes for Random Forest Classifier.

7 Discussion

Out of all the machine learning algorithms used for this dataset, the second Decision Tree Model turned out to be the best. It had the highest model evaluation scores and gave significant insights of the dataset. It even gave us a striking insight of one feature, the Longitude, having a significance on Sale Amount. Initially, this was not the case when conducting exploratory data analysis prior to this model. The regression models all had the worst scores for the dataset. Classification models was the best approach for this dataset compared to regression.

Our methodology was exhaustive but tedious. The proposed method in the first literature article would have served this venture well considering all the models had to be trained and evaluated individually. Had the proposed method been employed from the study a list of top contenders could have been obtained utilizing meta-data regression analysis on the models themselves. It even utilizes a more robust metric known as SAR and works for a variety of real and fictitious datasets. From this a select few could have been trained and evaluated given sufficient confidence that these were the best models to choose from.

The second article delves into finding the best classifiers given relatively clean complete datasets. The dataset is significantly smaller, on the order of a hundred, than the one used in this report, on the order of a million. Also there were many instances where data did need to be cleaned (i.e. NaN values, duplicates, label encoding, etc). However a similar model seemed to provide the most accurate predictions. In the case of the article and its analysis of different models it found that Random Forest was the most accurate. In our study it proved impractical to continue to train the Random Forest model but Decision Tree worked the best. Random Forest is an amalgam of many Decision Trees and given our many orders of magnitude larger dataset it is plausible this is why a simplified version of Random Forest was so effective.

8 Conclusion

In this project, we sought to develop machine learning models capable of accurately predicting house sale amounts and classifying properties as affordable or non-affordable. Through a comprehensive analysis, we identified the Decision Tree model as the most effective for both tasks. It not only delivered superior performance compared to other models but also provided valuable insights, such as the notable impact of Longitude on Sale Amount. The regression models, in contrast, showed lower efficacy, highlighting that classification approaches were more appropriate for our dataset.

A significant challenge faced was managing the data for training various models. We needed to maintain separate datasets for regression and classification tasks, which complicated the process. Additionally, the absence of temporal separation in the data was a limitation. Inflation and other temporal factors could influence sale amounts and affordability, suggesting that segmenting the data into meaningful time periods might have provided more nuanced insights.

For future research, we recommend exploring the impact of temporal factors by segmenting data into shorter time periods and analyzing trends over time. This approach could enhance the accuracy and relevance of predictions. Furthermore, investigating other advanced model evaluation techniques, such as meta-data regression analysis, could streamline model selection and improve overall performance. Finally, expanding the dataset to include additional features and varying geographic regions might provide a more comprehensive understanding of the factors influencing real estate markets.

8.1 Statement of Contribution

- Ikonkar: Code, Technical Report (Overleaf, Contents, Abstract, List of Figures Contents, Introduction, Methodology, Results, Conclusion, References), Design Diagrams, Presentation, Iterations
- Matthew: Code, Technical Report (Literature Review, Methodology, and Discussion), Presentation, Iterations

9 References

References

[1] T. Doan and J. Kalita, ”Selecting Machine Learning Algorithms Using Regression Models,” 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA, 2015, pp. 1498-1505, doi: 10.1109/ICDMW.2015.43.

[2] C. A. Ul Hassan, M. S. Khan and M. A. Shah, ”Comparison of Machine Learning Algorithms in Data Classification,” 2018 24th International Conference on Automation and Computing (ICAC), Newcastle Upon Tyne, UK, 2018, pp. 1-6, doi: 10.23919/IConAC.2018.8748995.