Introduction:

The process of becoming a homeowner today has vastly changed compared to the process that was in place merely ten years ago. With the ever-increasing prices, unpredictable markets, stagnant wages, and erratic economy, understanding how environmental attributes affect house sales can help decide how to navigate property purchases. To facilitate this understanding, data from the quality of life website for Charlotte, North Carolina, was used to create regression and classification models. These models were created using various variables such as ‘Home Sales Prices’ - average sale price of units per specific neighborhood, ‘Residential New Construction’ - new residential construction projects permitted within a year, ‘Housing Density’ - housing units per acre of land, ‘Housing Age’ - age of residential units in years, ‘Housing Size’ - size of housing units per square ft, and ‘Home Ownership Rates’ - percentage of homes which are owned vs rented. All of these features were compiled into a single CSV file, with each row representing a specific Neighborhood Profile Area(NPA) for 2022. Throughout the report, the model explanations will provide answers to two distinct questions:

Regression: How do ‘Housing Age’, ‘Housing Density’, ‘Housing Size’, ‘Home Ownership Rates’, and ‘Residential New Construction’ impact ‘Home Sales Prices’?
Classification: How do ‘Housing Age’, ‘Housing Density’, ‘Housing Size’, ‘Home Sales Prices’, and ‘Residential New Construction’ impact ‘Home Ownership Status’?

By using subsetting and feature engineering, decisions were made on which predictors would be used for the final models, as well as interpretations of the story that is told by the results of the models.

Regression Model:

For the regression model, several methods were considered to create the best model for the prediction. The response variable was ‘Sale Price’, and the explanatory variables for the first multiple linear regression were ‘Housing Density’, ‘Residential New Construction’, ‘Home Ownership’, ‘Housing Age’, and ‘Housing Size’. These independent variables created a model with an R-squared of .16, meaning 16% of the variability was explained by the model. There were problems associated with the first attempt, one of them being that three of the five predictors were not statistically significant to the model. ‘Housing Density’, ‘Residential New Construction’, and ‘Home Ownership’ had p-values that were higher than 0.05, so for the next regression model, these three variables were left out. After limiting the X variables to ‘Housing Age’ and ‘Housing Size’, the model summary showed that these predictors had an appropriate probability value of less than 0.05, while the R-squared decreased by only one percent to 15, showing that the previous regression model had redundant features(Model A). To explore other modeling methods, a regression tree with ‘Housing Age’ and ‘Housing Size’ as the features was created to show its impact on the ‘Sale Price’ of homes in Charlotte. In the tree, the primary split was through size, showing that if a house was equal to 4050 sq ft, the value tended to be around half a million dollars, and if it was less, the value would be around 400 thousand dollars. If the house was greater than 4050 sq ft and less than 33 years old, the value tended to go up to a million dollars. Although the regression tree provided clear insights into the sales patterns, due to the sample size being relatively low, the shallow tree could be considered an unreliable model(Model C). When it came to choosing the final model, the generalized additive model presented comparatively better results. With the GAM, the R-squared was 0.324 or 32.4%, which meant that compared to the other models, it had a higher percentage of variance that was explained by the model. In the model, s(0) was ‘Housing Age’ and s(1) was ‘Housing Size’, and while the p-value for age was quite high, size appeared to have a strong effect on the prices of homes in Charlotte(Model B).

Classification Model:

Our classification model aims to determine how the features ‘Housing Age’, ‘Housing Density’, ‘Housing Size’, ‘Home Sales Prices’, and ‘Residential New Construction ’ affect the Home Ownership status is split between Good, which is above 50% and poor which is 50% and below. The classification models we implemented were KNN, QDA, LDA, Decision Tree, Linear SVM, and RBF SVM. To determine the best model, we measure the accuracy score, which is how accurate the model is at making correct predictions with our dataset, and the cross-validation score, which determines how effective the model is in predicting unseen data. After testing several classification methods to find the classification model that had the highest accuracy, the best model was LDA(Model D), yielding a cross-validation score of 76.05% and an accuracy of 77.11%. For comparison, our second-highest method, Linear SVM, yielded a cross-validation score of 75.2% and an accuracy score of 73.68%. These results indicate that our model is 77.11% accurate when determining whether the Home Ownership status is good or poor using the data set and 75.2% when using unseen data.

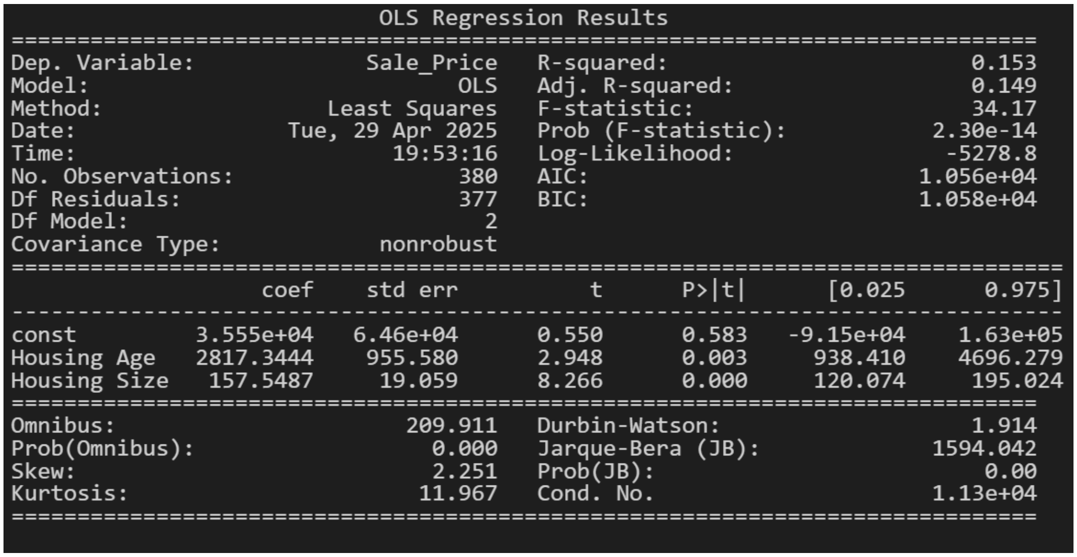
Conclusion:

Our findings discovered that the predictors for the regression were not as impactful as expected, as most of the predictor features besides ‘Housing Age’ and ‘Housing Size’ impacted the Sales Price. While these two predictors significantly impacted the Sales price, the model only explained 32.4% of the variability in the dataset. This implies that variables that we associated with the price of housing were far less impactful than we had believed. Our findings for the classification model indicate that using the features to determine the status of home Ownership in an area as good or poor was only 77.11% accurate. While our models' ability to predict that data was not as high as we had wanted or expected, the results of our model are still helpful in understanding the current status of the housing market, as it showed us that the relationship between our features and target variables was not as strong as we had expected.

Appendix:

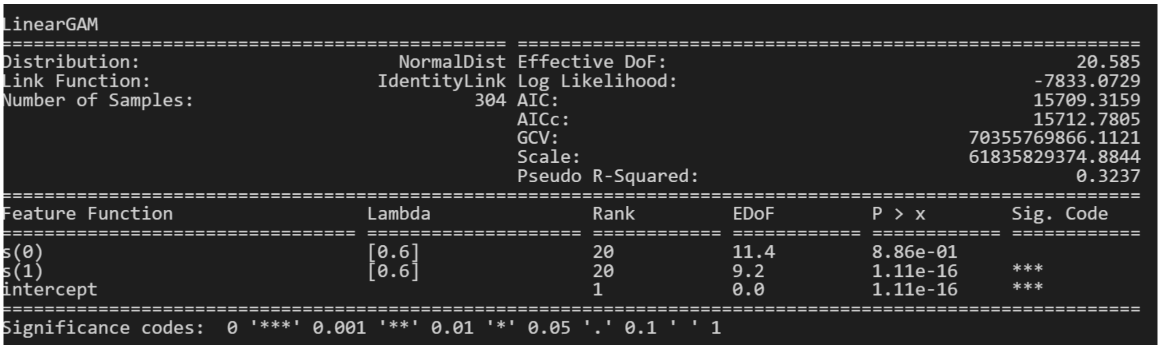
Regression:

Model A



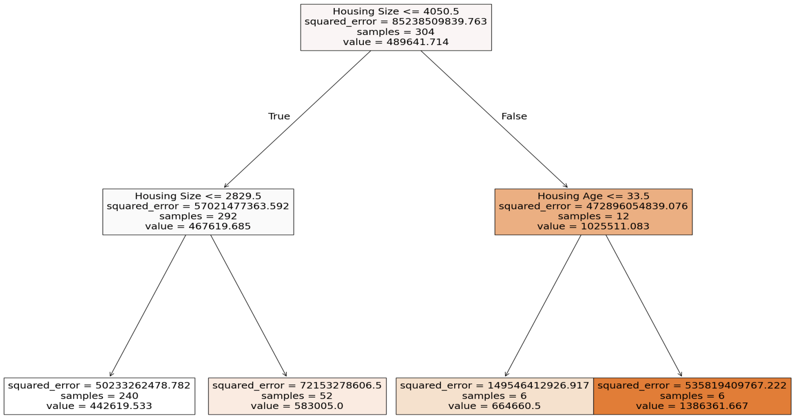
```
                            OLS Regression Results
==============================================================================
Dep. Variable:             Sale_Price   R-squared:                       0.153
Model:                            OLS   Adj. R-squared:                  0.149
Method:                 Least Squares   F-statistic:                     34.17
Date:                Tue, 29 Apr 2025   Prob (F-statistic):           2.30e-14
Time:                        19:53:16   Log-Likelihood:                -5278.8
No. Observations:                 380   AIC:                         1.056e+04
Df Residuals:                     377   BIC:                         1.058e+04
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const         3.555e+04   6.46e+04      0.550      0.583   -9.15e+04    1.63e+05
Housing Age   2817.3444    955.580      2.948      0.003     938.410    4696.279
Housing Size   157.5487     19.059      8.266      0.000     120.074     195.024
==============================================================================
Omnibus:                      209.911   Durbin-Watson:                   1.914
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1594.042
Skew:                           2.251   Prob(JB):                         0.00
Kurtosis:                      11.967   Cond. No.                     1.13e+04
==============================================================================
```

Model B



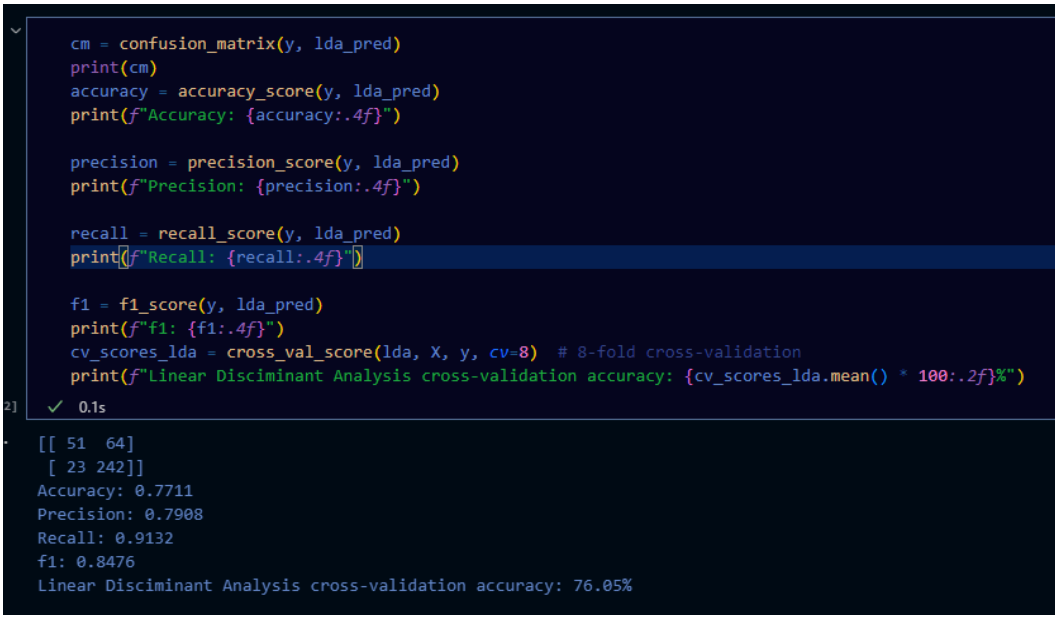
```
LinearGAM
=============================================== ==========================================================
Distribution:                        NormalDist Effective DoF:                                     20.585
Link Function:                     IdentityLink Log Likelihood:                                -7833.0729
Number of Samples:                          304 AIC:                                           15709.3159
                                                AICc:                                          15712.7805
                                                GCV:                                      70355769866.1121
                                                Scale:                                    61835829374.8844
                                                Pseudo R-Squared:                                  0.3237
==========================================================================================================
Feature Function                  Lambda               Rank         EDoF         P > x        Sig. Code
================================= ==================== ============ ============ ============ ============
s(0)                              [0.6]                20           11.4         8.86e-01
s(1)                              [0.6]                20           9.2          1.11e-16     ***
intercept                                              1            0.0          1.11e-16     ***
==========================================================================================================
Significance codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model C



Classification:

Model D

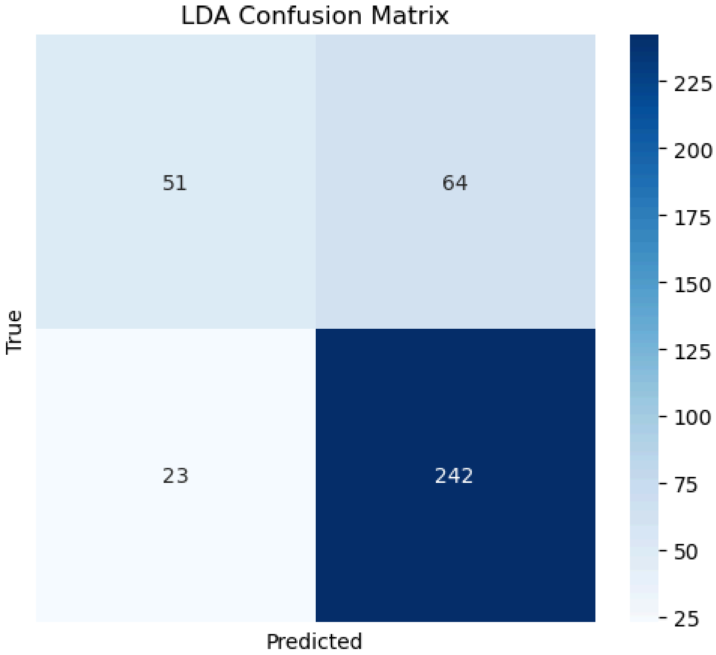


```
cm = confusion_matrix(y, lda_pred)
print(cm)
accuracy = accuracy_score(y, lda_pred)
print(f"Accuracy: {accuracy:.4f}")

precision = precision_score(y, lda_pred)
print(f"Precision: {precision:.4f}")

recall = recall_score(y, lda_pred)
print(f"Recall: {recall:.4f}")

f1 = f1_score(y, lda_pred)
print(f"f1: {f1:.4f}")
cv_scores_lda = cross_val_score(lda, X, y, cv=8)  # 8-fold cross-validation
print(f"Linear Discriminant Analysis cross-validation accuracy: {cv_scores_lda.mean() * 100:.2f}%")
```

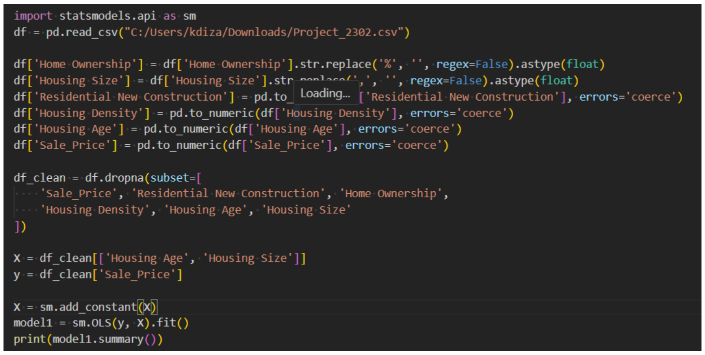
✓ 0.1s

```
[[ 51  64]
 [ 23 242]]
Accuracy: 0.7711
Precision: 0.7908
Recall: 0.9132
f1: 0.8476
Linear Discriminant Analysis cross-validation accuracy: 76.05%
```

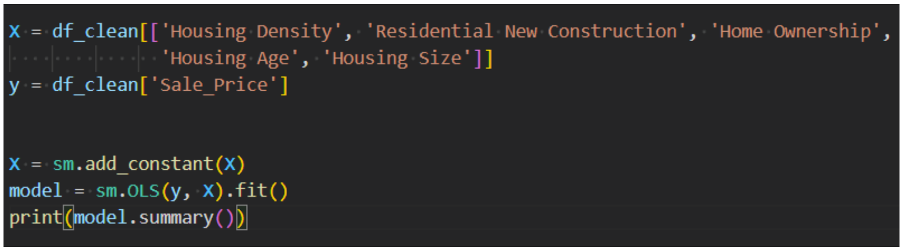
Figure E



Linear Regression:

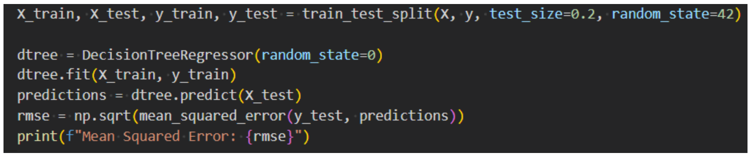


```
import statsmodels.api as sm
df = pd.read_csv("C:/Users/kdiza/Downloads/Project_2302.csv")

df['Home Ownership'] = df['Home Ownership'].str.replace('%', '', regex=False).astype(float)
df['Housing Size'] = df['Housing Size'].str.replace(',', '', regex=False).astype(float)
df['Residential New Construction'] = pd.to_ Loading... ['Residential New Construction'], errors='coerce')
df['Housing Density'] = pd.to_numeric(df['Housing Density'], errors='coerce')
df['Housing Age'] = pd.to_numeric(df['Housing Age'], errors='coerce')
df['Sale_Price'] = pd.to_numeric(df['Sale_Price'], errors='coerce')

df_clean = df.dropna(subset=[
    'Sale_Price', 'Residential New Construction', 'Home Ownership',
    'Housing Density', 'Housing Age', 'Housing Size'
])

X = df_clean[['Housing Age', 'Housing Size']]
y = df_clean['Sale_Price']

X = sm.add_constant(X)
model1 = sm.OLS(y, X).fit()
print(model1.summary())
```

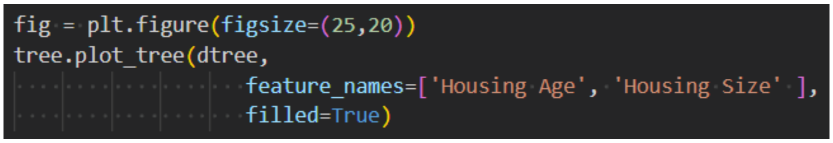


```
X = df_clean[['Housing Density', 'Residential New Construction', 'Home Ownership',
              'Housing Age', 'Housing Size']]
y = df_clean['Sale_Price']


X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())
```

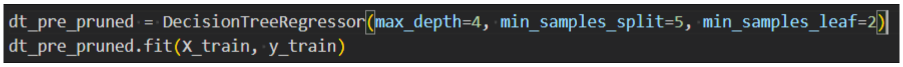
Regression Tree:



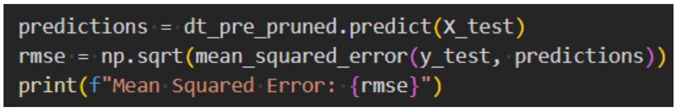
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

dtree = DecisionTreeRegressor(random_state=0)
dtree.fit(X_train, y_train)
predictions = dtree.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, predictions))
print(f"Mean Squared Error: {rmse}")
```



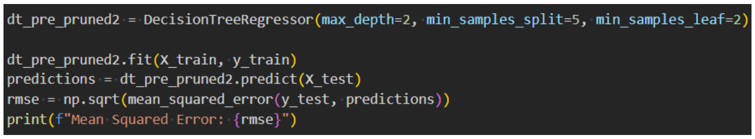
```
fig = plt.figure(figsize=(25,20))
tree.plot_tree(dtree,
               feature_names=['Housing Age', 'Housing Size' ],
               filled=True)
```



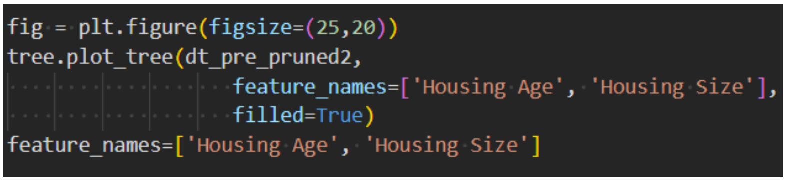
```
dt_pre_pruned = DecisionTreeRegressor(max_depth=4, min_samples_split=5, min_samples_leaf=2)
dt_pre_pruned.fit(X_train, y_train)
```



```
predictions = dt_pre_pruned.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, predictions))
print(f"Mean Squared Error: {rmse}")
```

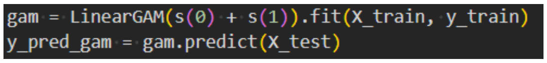


```
dt_pre_pruned2 = DecisionTreeRegressor(max_depth=2, min_samples_split=5, min_samples_leaf=2)

dt_pre_pruned2.fit(X_train, y_train)
predictions = dt_pre_pruned2.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, predictions))
print(f"Mean Squared Error: {rmse}")
```

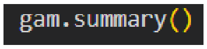


```
fig = plt.figure(figsize=(25,20))
tree.plot_tree(dt_pre_pruned2,
               feature_names=['Housing Age', 'Housing Size'],
               filled=True)
feature_names=['Housing Age', 'Housing Size']
```

Linear GAM:



```
gam = LinearGAM(s(0) + s(1)).fit(X_train, y_train)
y_pred_gam = gam.predict(X_test)
```



```
gam.summary()
```