Team_ID: CS_1

Project name: Apartment Rent Prediction

Member ID	Member name
2021170261	شهد حسن اكرام صالح فريد
2021170507	مريم صلاح سالم شعبان
2021170268	ضحى عادل حسن مصطفى
2021170183	دينا طارق عبد الحميد عبد الحميد
2021170504	مريم رضا عبد القادر محمود
2021170672	مريم أحمد طه أحمد الجوهري

MS.1

1. Handling Missing Values :

Bathrooms and Bedrooms : The missing values in the 'bathrooms' and 'bedrooms' columns were filled with mode.

Amenities : Mode values for 'amenities' were calculated separately for each category ('rent', 'apartment', 'housing'). Missing values in 'amenities' were then filled based on the mode of the respective category.

Pets Allowed : Missing values were filled with 'None'.

City Name : Missing values were filled with the mode (most frequent value) of the column.

State : Missing values were filled with the mode of the column.

Address : Addresses were filled based on the mode addresses for each city. If there's only one address for a city, that address is used; otherwise, the mode address for the city is used. If mode address isn't available, a default value ('unknown') is used.

Longitude and Latitude : Missing values in 'longitude' and 'latitude' columns were filled with the mean of the respective column. Additionally, negative values in the 'longitude' column were made positive using the absolute function (`abs()`), ensuring all longitude values are positive.

2. Handling Categorical Data :

Label Encoding : Categorical columns were identified, and label encoding was applied using `LabelEncoder()` from `sklearn.preprocessing`. This transforms categorical values into numerical labels, which is necessary for machine learning algorithms to operate on them.

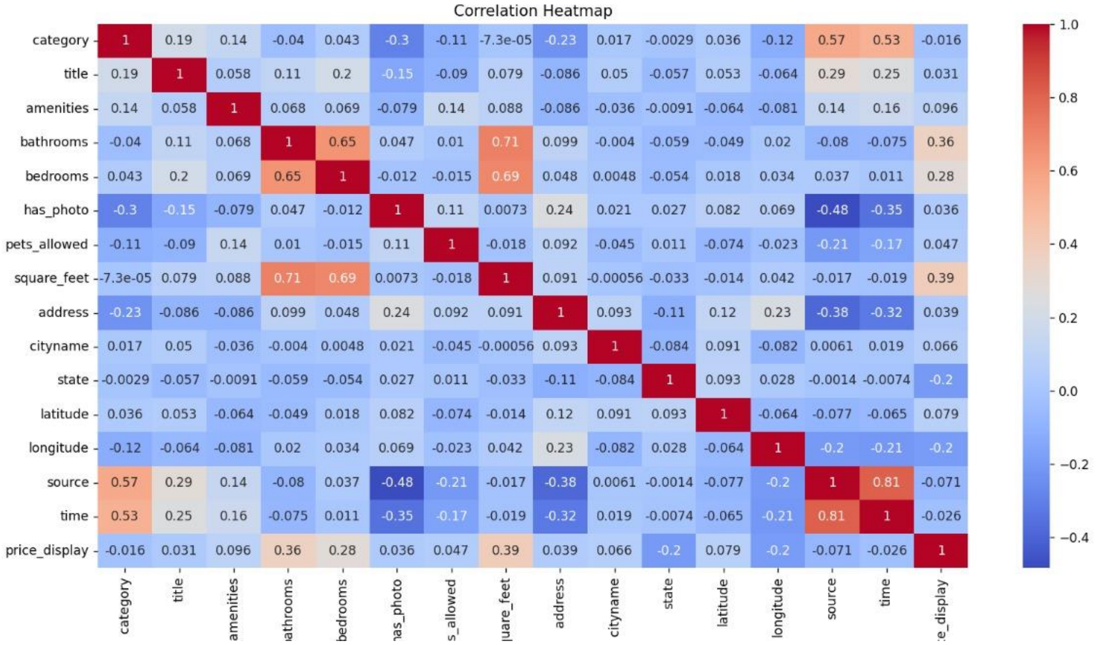
3. Feature Scaling :

To handle highly varying magnitudes or values or units, normalization was performed features which have a relatively small range of values ('category', 'amenities', 'bathrooms, bedrooms","has_photo", pets_allowed",'price"square_feet","address","cityname","st ate","latitude","longitude", 'source') and standardization on the features with a wide range of values ('title', 'body') to avoid weighing greater values, higher and consider smaller values as the lower values, regardless of the unit of the values

4. Feature Selection :

To select the most relevant features for the predictive model, feature selection was performed based on the correlation with the target column ”price display” . Features with high correlation coefficients with the target column were identified as top features and were selected for further analysis. This approach helped focus on the most influential features, potentially improving the performance of the predictive model.

We dropped certain features from the dataset as they were deemed unsuitable for prediction purposes.:
Some of them are unique features as: id
Some are almost unique and we take data from them for another features as: body
Some are with the same value for all data as: fee, currency, price_type.



But the correlation is not the best technique for the feature selection, so we make select features manually to get the best prediction.

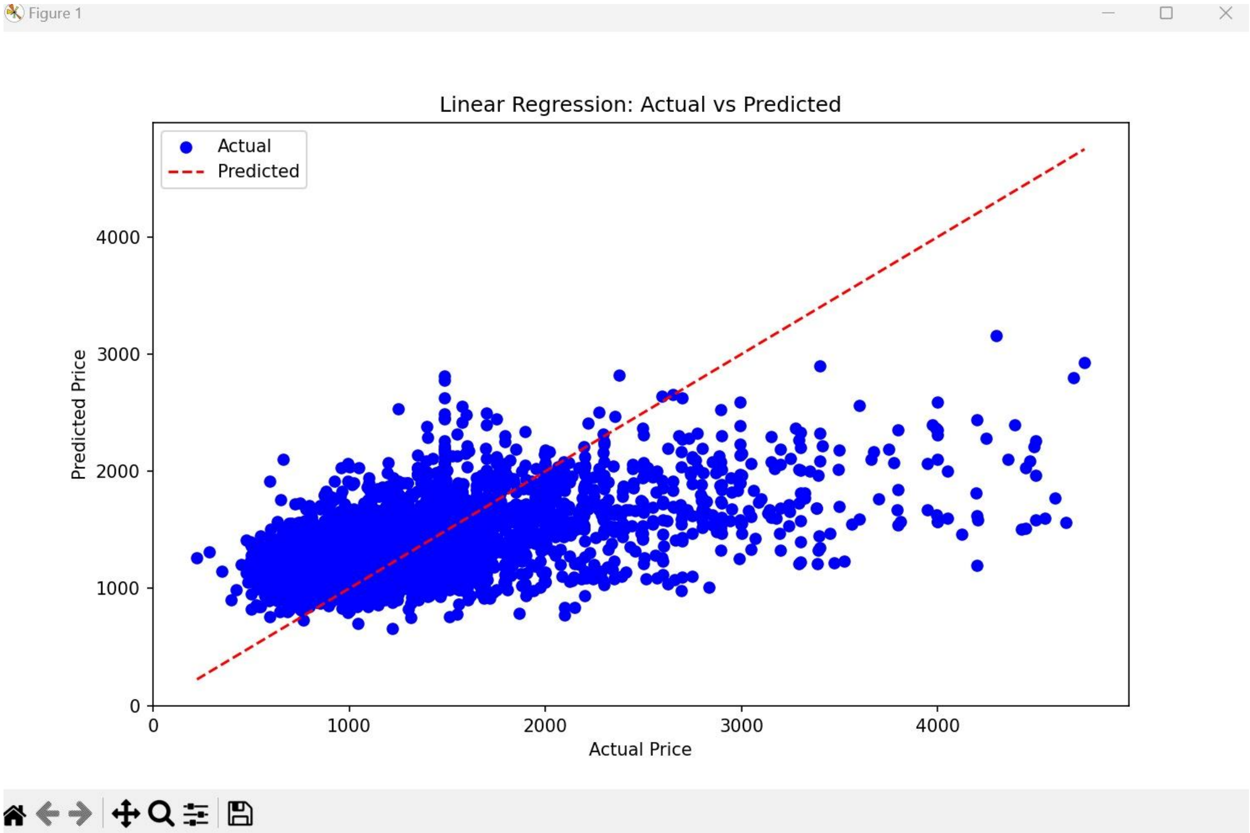
These preprocessing techniques ensure that the dataset is clean, filled with meaningful values, and properly formatted for training machine learning models. Additionally, the techniques applied maintain the integrity and relevance of the data for accurate modeling and predictions.

Data Splitting:

By using “train_test_split” function from scikit-learn library to split the data into training and testing sets with test size = 30%.

Model Selection:

1-Linear Regression: Trained the linear regression model on the training data X_train, Y_train using the **fit()** method and generated predictions on the test data X_test using the trained model's **predict()** method. The mean square error obtained from the model evaluation after setting the **random state** by **50** was **551916.432467397**.



2-Polynomial Regression

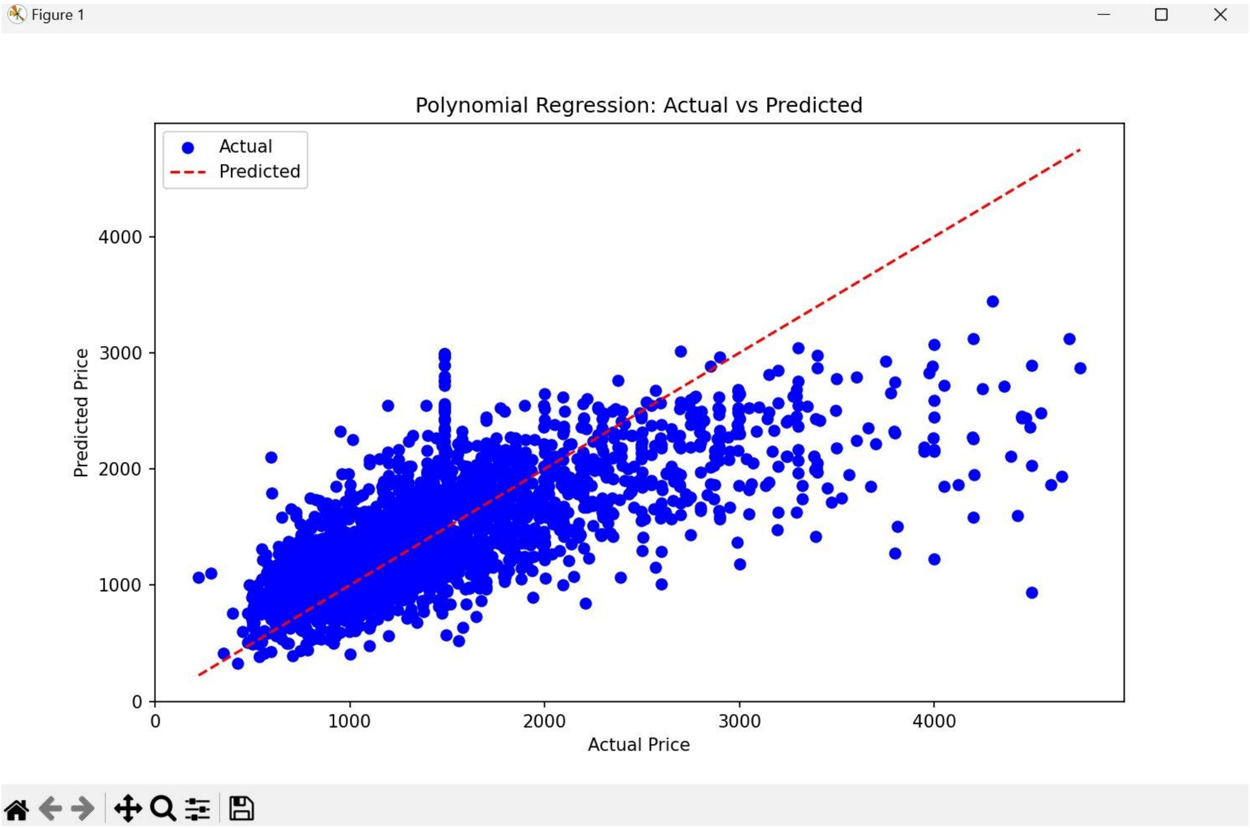
This code demonstrates polynomial regression, a type of regression analysis in which the relationship between the independent variable X and the dependent variable Y is modeled as an nth degree polynomial.

Sizes of your training & testing sets:

Test size: 20%.
Train size:80%

Techniques that were used to improve the results:

-Set different degrees for the model to get the best accuracy and the lowest error, the best degree was 2.
-Set different values to the parameter random state the best one was 10.



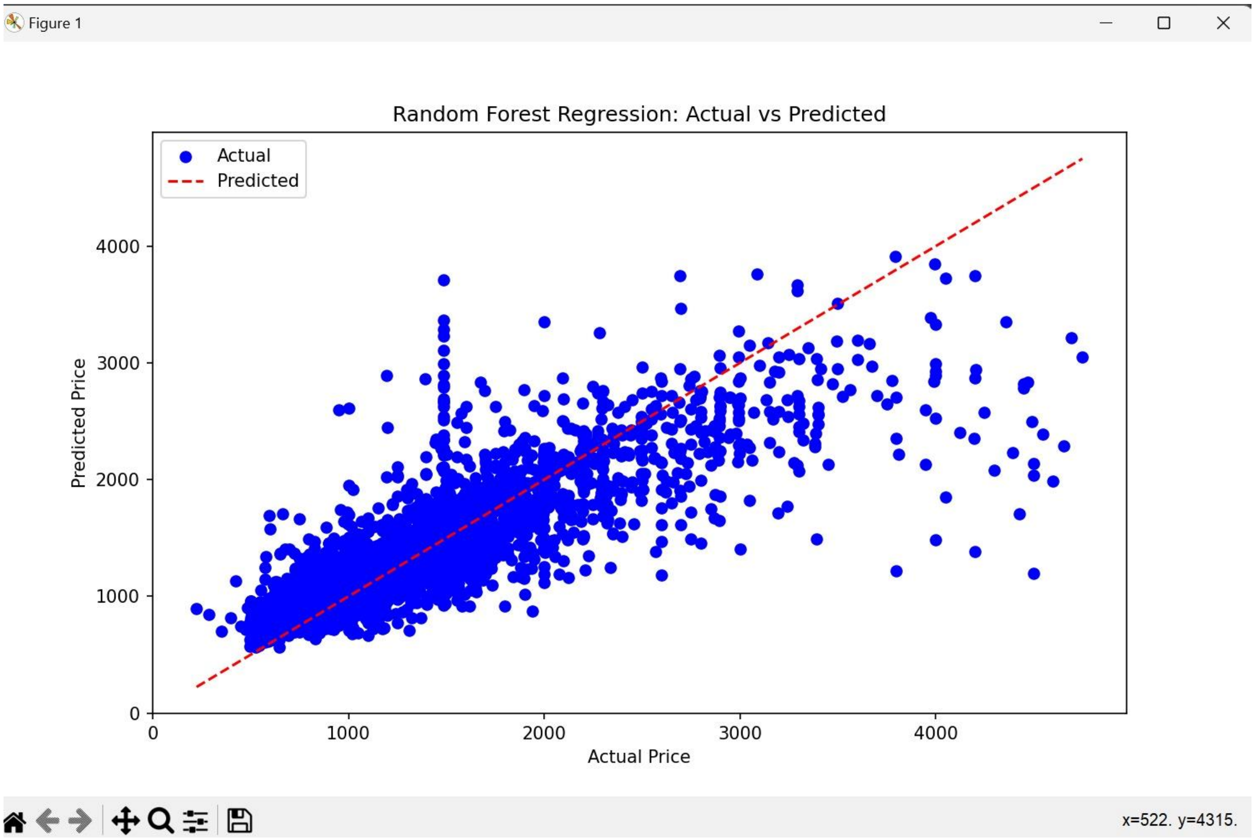
3-support vector regression

Linear SVR tries to find the best fit line similar to linear regression but the difference is it has region that is insensitive to error EPSILON the boundary of this region is decided upon support vectors. In multidimensional SVR, instead of fitting a line (as in the case of linear SVR in one dimension), the algorithm fits a hyperplane to the data.

if the data is highly non-linear or lacks clear separation SVR struggle to find decision boundary. The best prediction is with hyperparameters poly kernel function and gamma=1.

4-Random Forest

It is based on the concept of decision trees but improves upon them by introducing randomness during the tree-building process and combining the predictions of multiple trees to create a more robust model. provides a

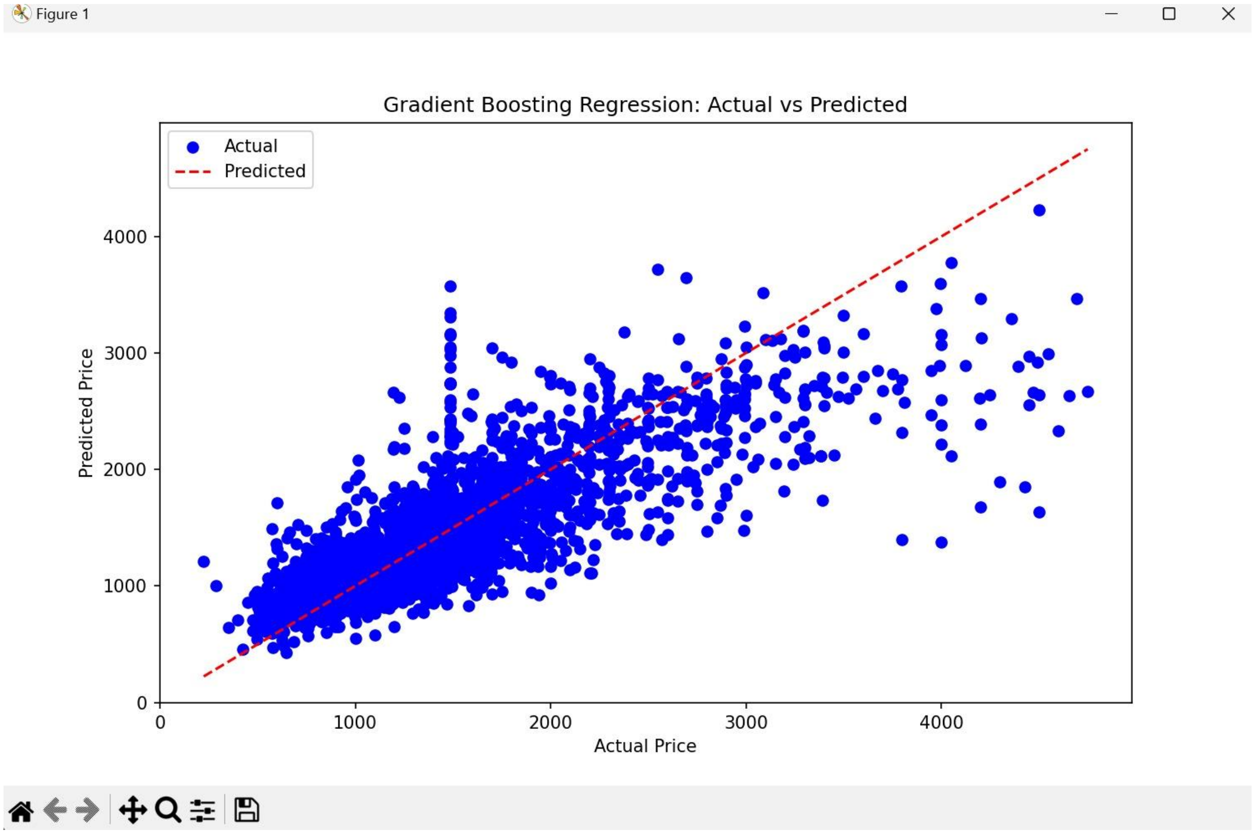


measure of feature importance, indicating which features are most influential in making predictions.

5-Gradient Boost

sequentially adds weak learners and focusing on the errors made by the ensemble, gradient boosting gradually improves the model's predictive performance. It effectively combines the strengths of multiple weak learners to create a strong and robust predictive model.

This sequential training process allows gradient boosting to iteratively improve the model's performance by gradually reducing prediction errors. In contrast, traditional regression techniques usually train a single model on the entire dataset in one step.



Comparison:

	MSE	R2 Score
Linear Regression	367105.38930956984	28.87961786246146 %
Polynomial Regression	237661.90430050532	53.95707623585566 %
SVR	247815.1268265442	51.99006325536677 %
Random Forest	152394.6580418947	70.47614491292752 %
Gradient boost	159936.3489086324	69.01507146637765 %

Conclusion:

The polynomial regression model has a lower MSE and a higher R2 score compared to linear regression, indicating that it provides a better fit to the data and explains a larger proportion of the variance in the target variable. this shows that the data is nonlinear.

SVR with the RBF kernel may not be able to capture the specific non-linear patterns present in the data, leading to inferior performance compared to polynomial regression.
the dataset appears to have characteristics that are well-suited for tree-based models like random forests. This model can effectively capture the underlying patterns and relationships in the data.

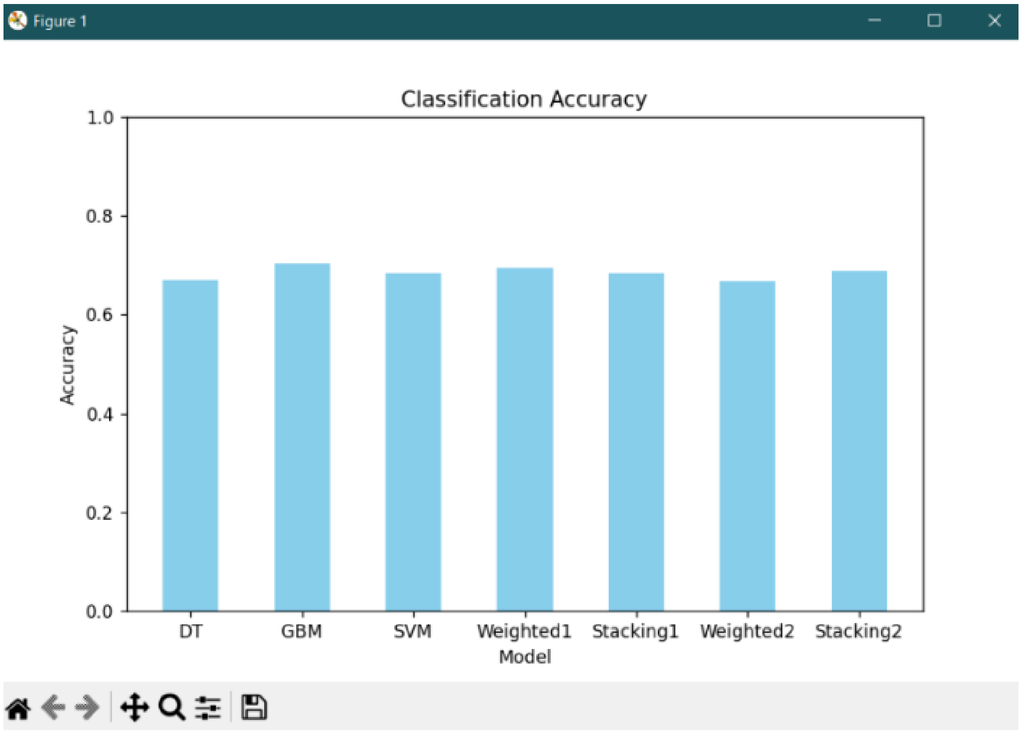
MS.2

Changes made in Preprocessing:

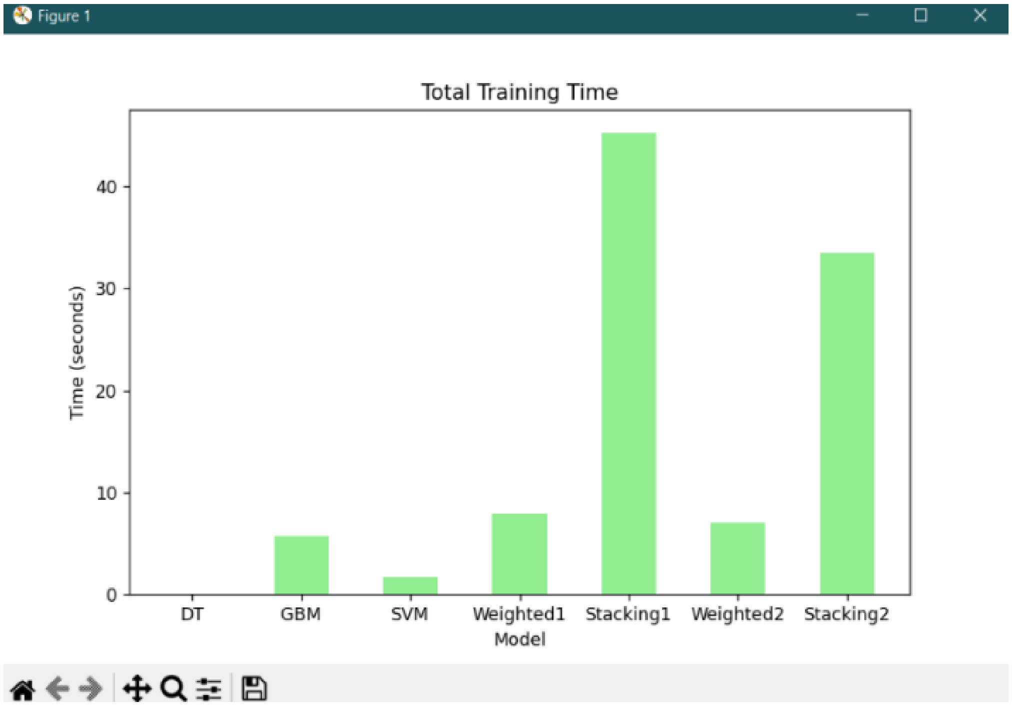
1. Removed standardization, which makes features have a mean of zero and a standard deviation of one, is usually done to keep feature scales consistent. But when using ensemble learning, this can be tricky, especially if some features have negative values. So, instead of standardization, we use different methods to scale the data for ensemble learning.

2. To deal with missing values:
   a. Imputation with Mean: We replace missing values in the dataset with the average value of that feature. This helps fill in the gaps while keeping the overall pattern of the data. “Done to latitude, longitude.”
   b. Imputation with Mode: For categories or groups, we replace missing values with the most common value in that group. This helps maintain the main characteristics of the categorical data. “Done to all features except latitude, longitude.”

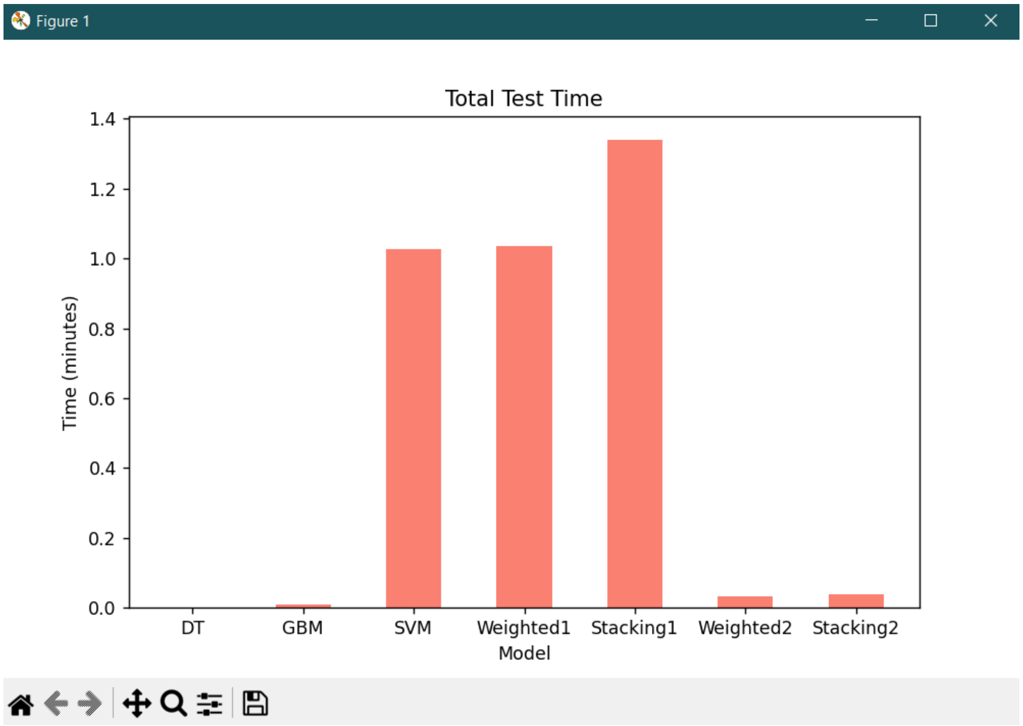
Classification Accuracies:



Training Time:



Testing Time:



Feature selection process:

ANOVA:

- ANOVA is used to assess the significance of the relationship between each numerical feature and the target variable (rent category).
- It calculates an F-statistic and associated p-value for each feature, where a lower p-value indicates higher significance.
- Features with p-values below a predefined threshold (e.g., 0.05) are considered significant and selected for further analysis.

Kendall's Tau Correlation Coefficient:

- Kendall's Tau correlation coefficient is a non-parametric measure of association between two numerical variables.
- It assesses the ordinal association between each numerical feature and the target variable.
- Features with p-values below a predefined threshold (e.g., 0.05) are considered significant and selected for further analysis.

Chi-squared Test:

This test helps us understand if there's a significant relationship between two categorical variables. In our case, it helps us see if certain categories of apartment features are related to rent prices.

Mutual Information:

This tells us how much one feature can tell us about another. It helps us understand if knowing one thing about an apartment can give us useful information about its rent.

By using these methods, we can select the features that are most likely to help us predict apartment rents accurately.

Hyperparameter Tunning:

Decision Tree Classifier:

max_depth	criterion	Accuracy
3	gini	63.2222222
5	entropy	65.77777777
7	gini	66.888888888

Gradient Boosting Classifier:

n_estimators	learning_rate	Accuracy
150	0.1	70.166666666
250	0.02	68.611111111
300	0.03	69.166666666

SVM Classifier:

Kernel	parameter	C	Accuracy
linear	-	1	55.16666666
rbf	gamma=0.8	0.5	68.33333333
poly	degree=4	1	66.0

Conclusion:

Data Preprocessing Importance: The preprocessing steps, including handling missing values, encoding categorical variables, and scaling numerical features, proved crucial in preparing the data for modeling. By ensuring data quality and consistency, we laid a solid foundation for accurate predictions.

Feature Selection Strategies: Employing various feature selection methods such as ANOVA, Kendall's tau correlation, Chi-squared test, and Mutual Information allowed us to identify the most informative features for rent prediction. This approach helped streamline the model input and potentially improve its interpretability and efficiency.

Model Training and Evaluation: Training multiple classifiers and evaluating their performance provided valuable insights into the predictive power of different algorithms. Decision Tree, Gradient Boosting, and Support Vector Machine classifiers exhibited varying degrees of accuracy, highlighting the importance of selecting the right model for the task at hand.

Ensemble Techniques: Leveraging ensemble methods like Weighted Voting and Stacking enabled us to harness the collective wisdom of multiple models, potentially enhancing prediction accuracy and robustness. These techniques demonstrated the benefits of combining diverse models to mitigate individual weaknesses and improve overall performance.

Continuous Improvement: While the models achieved promising accuracy levels, there is always room for improvement. Continuous refinement through hyperparameter tuning, feature engineering, and model optimization can further enhance predictive performance and address any remaining challenges.