Apartment Rent Prediction Project Milestone 2

TeamID: CS_13

Team Members

Name	ID	section	Department
مصطفي حسين محمود	2021170519	6	CS
وليد محمد مصلح	2021170616	7	
كيرلس سمعان رزق	2021170411	5	
منة الله عبد العاطي محمد	2021170548	7	
منة الله ابراهيم حسن	2021170550	7	
هاجر عصام محمود محمود	2021170600	7	

1) Preprocessing

Step 1: Apply Train Test Split

- **Split data into train set (80%) test set(20%) and Random State (42)**

Step 2: Dealing with missing values

- getting the number of null values in y (target) using:

```
y.isnull().sum()
```

-there are no null values in the RentCategory Column (Y)

- getting the number of null values in x (features) using this code:

```
x.isnull().sum()
```

- Then replace it by using **replace_nulls(df)** that:
  - Filling missing values with the mean for numeric columns in X_train.
  - Filling missing values with the mode for string columns in X_train.

Step 3: Replace and Handling the outliers

There are two functions are used there:

- **replace_outliers(df)**: This function replaces outliers with the third quartile value (Q3) for each numeric column in X_train.
- **Note:** We didn’t apply to replace outliers because of the enhancement of the accuracy of the model when the data with outliers

Step 4: Data Encoding

Encoding the categorical variables into a numerical format allows the algorithms to process them effectively.

Encoding is done using Label Encoder:

- **Label Encoding for columns**
  - In all categorical columns in X_train, we transformed the label encoder fit.

- We saved the fit transform in X_train using pickle for each categorical column
- In X_test we transformed it with the saved pickle file
- The values in X_test were not In X_train we replaced it with the max value plus one in each column then updated the max value (+=1)

Step 5: Feature Scaling

Doing so ensures that all features have the same scale, preventing features with larger scales from dominating those with smaller ones during model training. It equalizes the importance of features and helps the algorithm converge faster. Using **MinMaxScaler().**

Note: We applied the MinMax Scaler based on the values of X_train

Step 6: Feature Selection and Removal

We Used Anova in Feature Selection by making K=15.

2) Applying Model

We evaluate the performance of various regression models using the provided dataset and features.

The models considered include:

- Logistic Regression (we applied the cross-validation and the accuracy didn't change)
- RBF (we applied the cross-validation and the accuracy didn't change)
- Random Forest
- Decision Tree
- Voting Classifier
- Stacking
- KNN

Model Performance:

Case: Using Anova (K=15)

Model Name	Train R2 Score	Test R2 Score
Logistic Regression	0.60	0.58
RBF	0.94	0.58
Random Forest	1.0	0.76
Decision Tree	1.0	0.63
Voting Classifier		0.71
Stacking		0.76
KNN (K=7)	.74	.61
KNN (K=3)	.81	.60
KNN (K=5)	.77	.62

The Random Forest model achieved the highest R2 Score on the test set, indicating **strong predictive performance**.

Case: Using Kendall’s (K=15)

Model Name	**Train R2 Score**	**Test R2 Score**
Logistic Regression	0.59	0.53
RBF	0.90	0.54
Random Forest	1.0	0.52
Decision Tree	1.0	0.45
Voting Classifier		0.54
Stacking		0.54

Case: Without Feature Selection and with outliers

Model Name	Train R2 Score	Test R2 Score
Logistic Regression	0.60	0.58
RBF	0.94	0.55
Random Forest	1.0	0.76
Decision Tree	1.0	0.62
Voting Classifier		0.72
Stacking		0.76

Case: Without Feature Selection and with replacing outliers

Model Name	**Train R2 Score**	**Test R2 Score**
Logistic Regression	0.59	0.56
RBF	0.95	0.55
Random Forest	1.0	0.75
Decision Tree	1.0	0.59
Voting Classifier		0.72
Stacking		0.76

Appling Script

- **Fit the training data for each model**
- **Save the model with pickle**
- **Load model**
- **Transform**
- **Predict**