Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011

Machine Learning Analysis Report

A. Problem analysis
   a. Task objective

      The dataset of property in Surabaya contains features that might or might not affect pricing of the property in the dataset. Through it we can use it to predict the price itself and the pricing category. The usage of the prediction is to predict fair prices especially during the recession we are going to go through. The fair prices are especially important for investors that have a lot of properties.

   b. Target variable
      i. Price for regression
      ii. Pricing category for classification

         The classes for pricing category are underpriced, normal priced and overpriced.

B. Machine learning algorithm
   a. Data understanding

      The dataset we are using is in .csv format and consists of 490 data with 18 columns which are: cluster_name, surface_area, building_area, bedrooms, bathrooms, storey, community_price, price, ownership_status, facing, house_position, road_width, urget, building_age, ready_to_use, furnished, category, pricing_category. Below are the explanation of the columns:

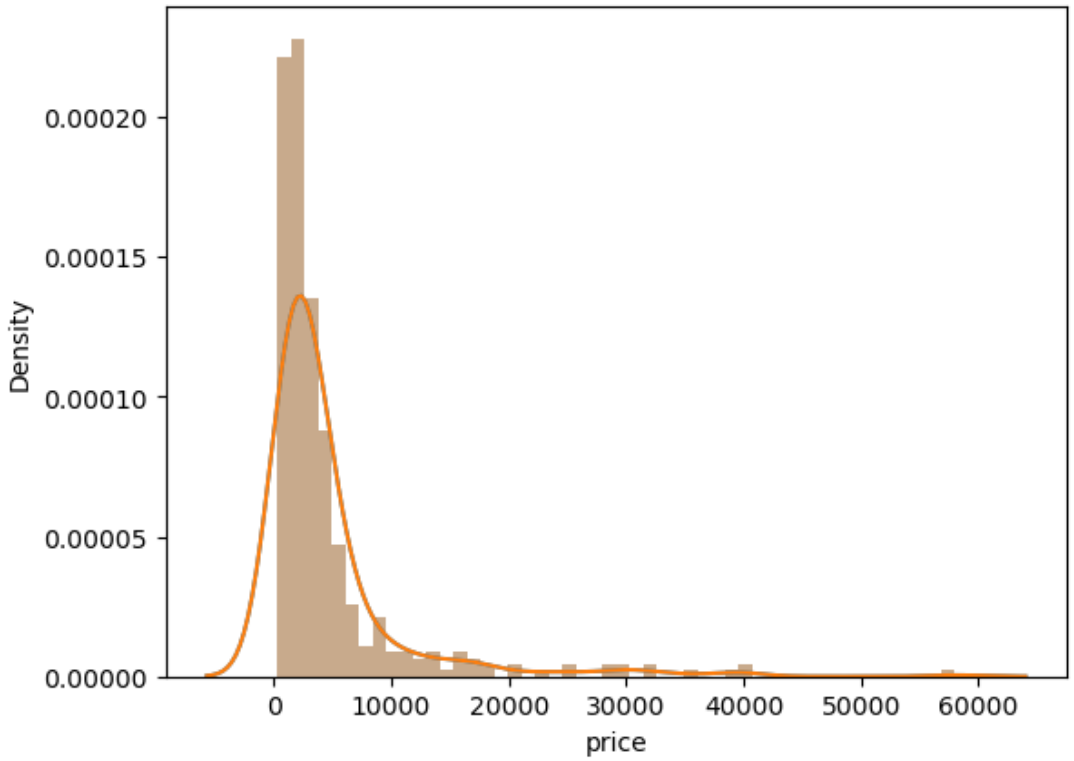
- cluster_name **:** Cluster of the property
- surface_area **:** Surface area of the property
- building_area **:** Building area of the property
- bedrooms **:** Number of bedrooms of the property
- bathrooms **:** Number of bathrooms of the property
- storey **:** Number of floors of the property
- community_price **:** Price determined by the community
- price **:** Listed price of the property
- ownership_status **:** Ownership of the property
- facing **:** The direction of the property
- house_position **:** Position of the property
- road_width **:** Road width near the property
- urgent **:** Urgency of the property owner

Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011

- building_age : Building age of the property
- ready_to_use : Usability of the property
- furnished : Furnishing condition of the property
- category : Category of the property
- pricing_category : Pricing category of the property

For better understanding of the data, we need to do an exploratory data analysis (EDA). Through EDA, we can get an idea of what's going on in the dataset:

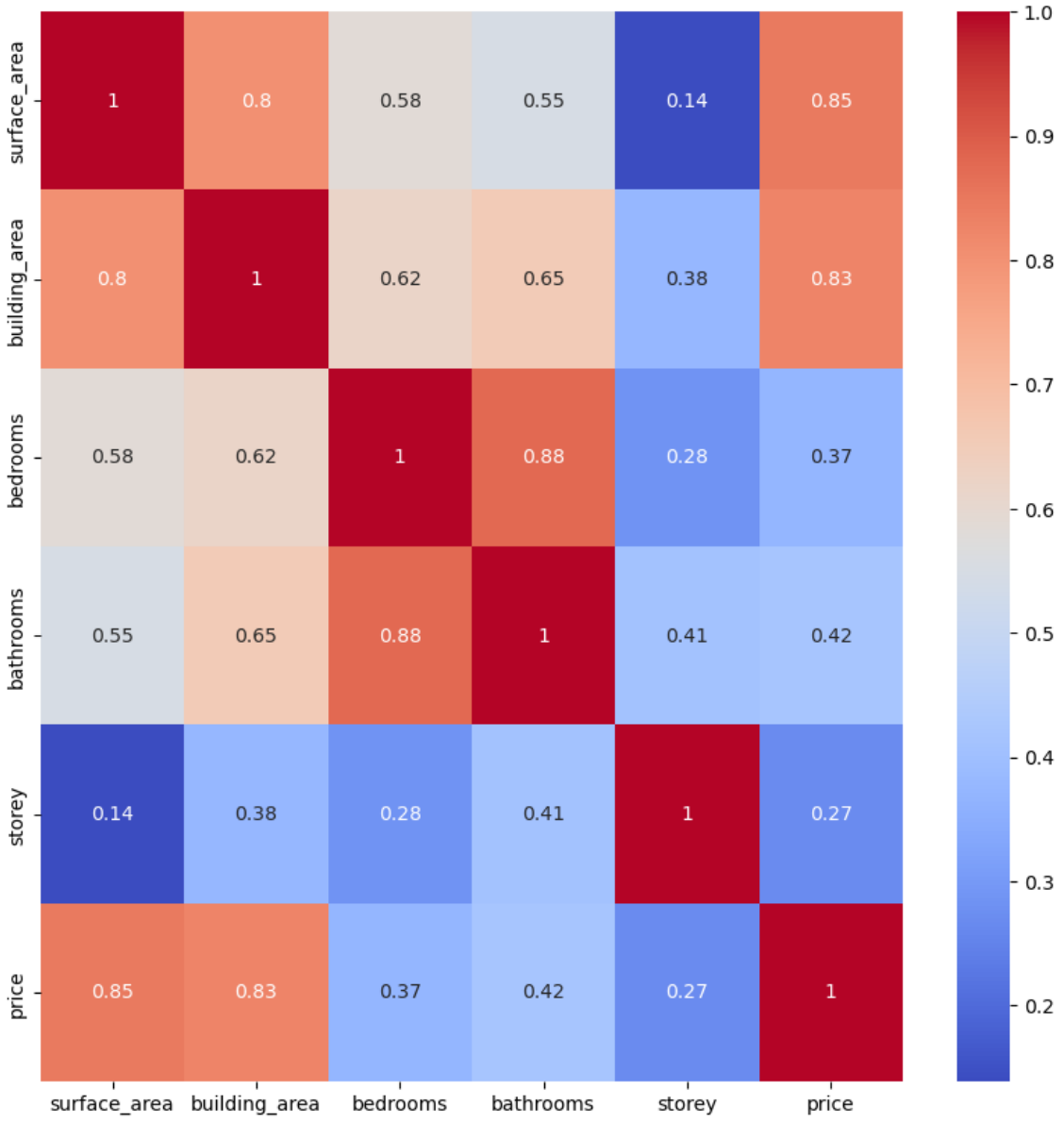
i. Price distribution



The price distribution is left-skewed.

ii. Heatmap

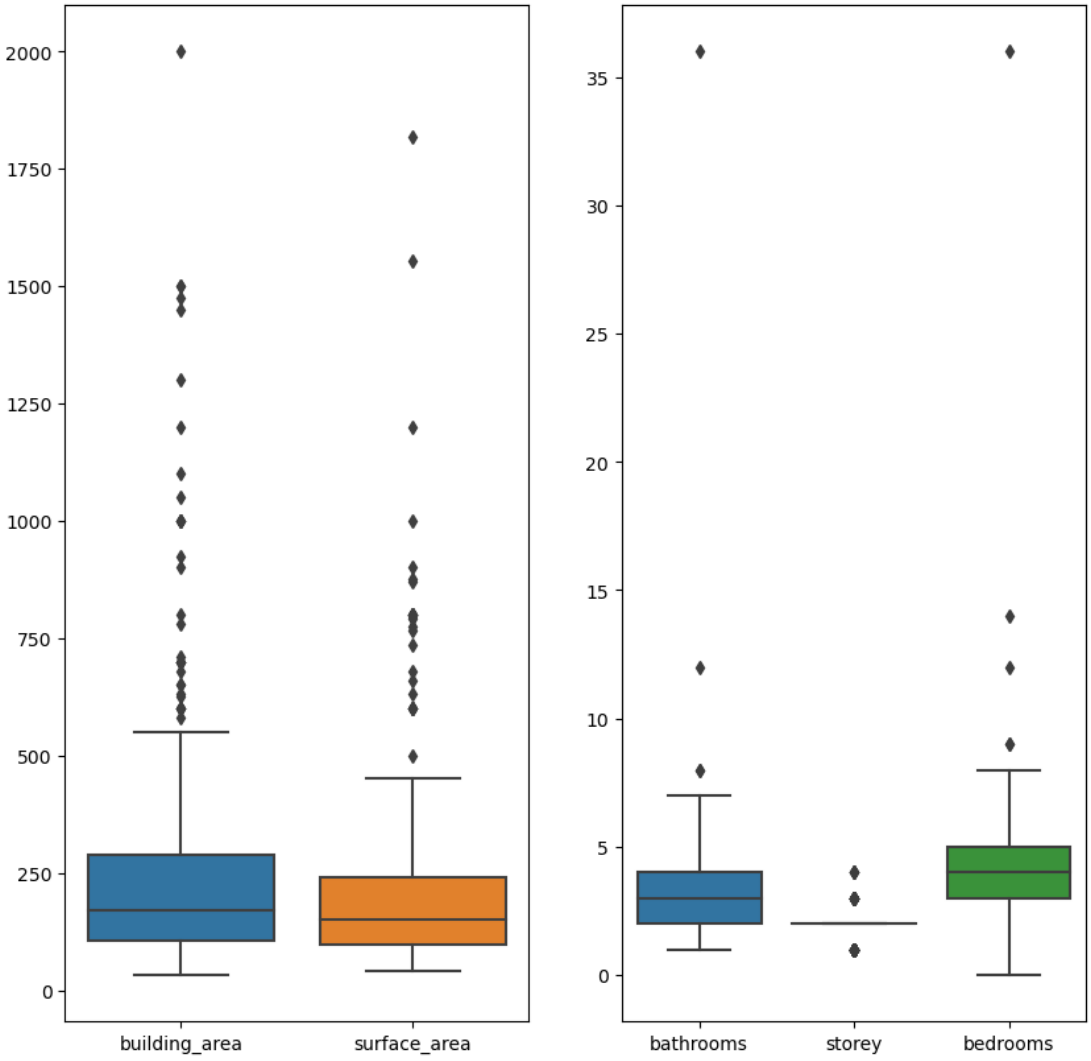
Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011



The heatmap shows that surface area and building area have a strong positive correlation with price. While bedrooms, bathrooms and storey have a weak positive correlation with price.

iii. Boxplot

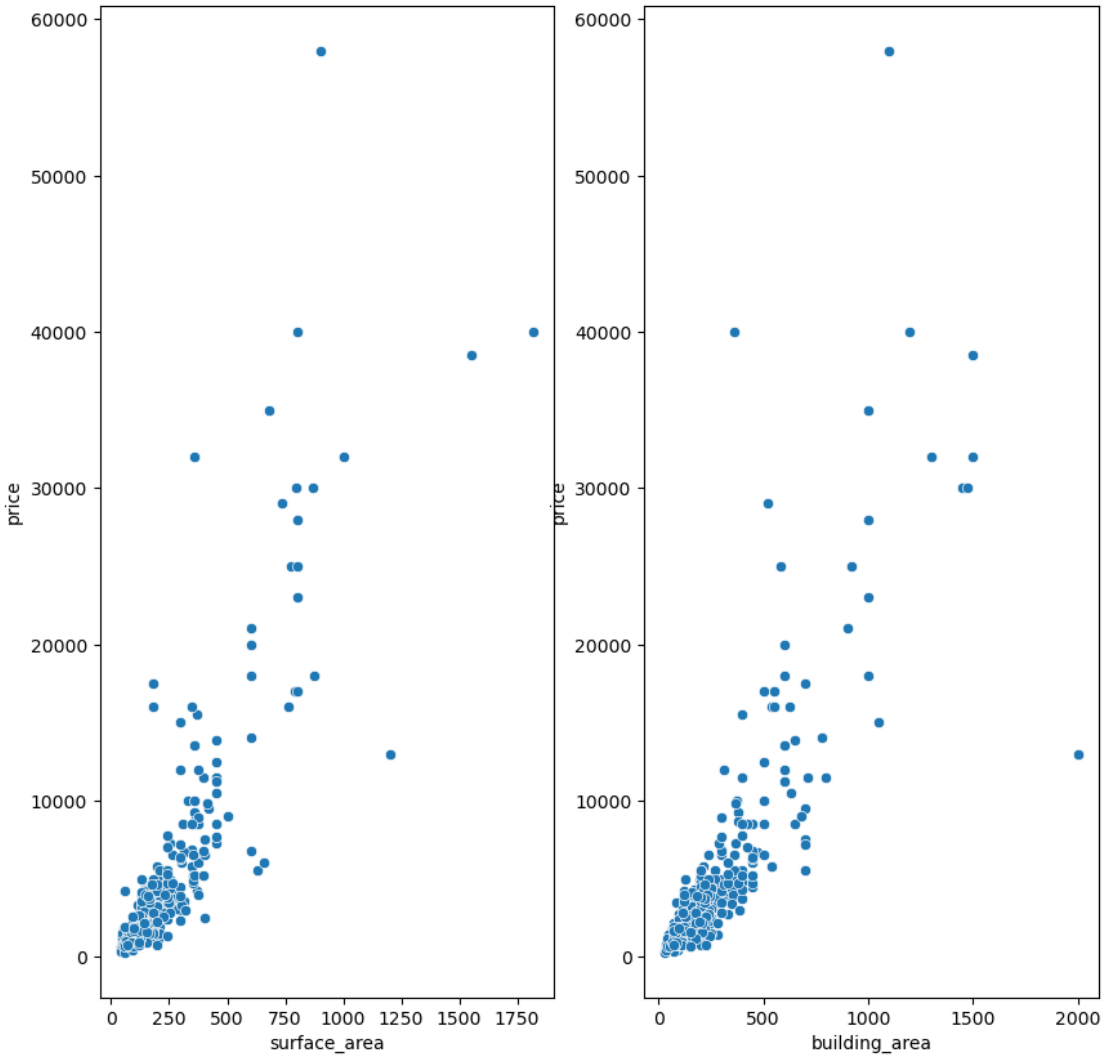
Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011



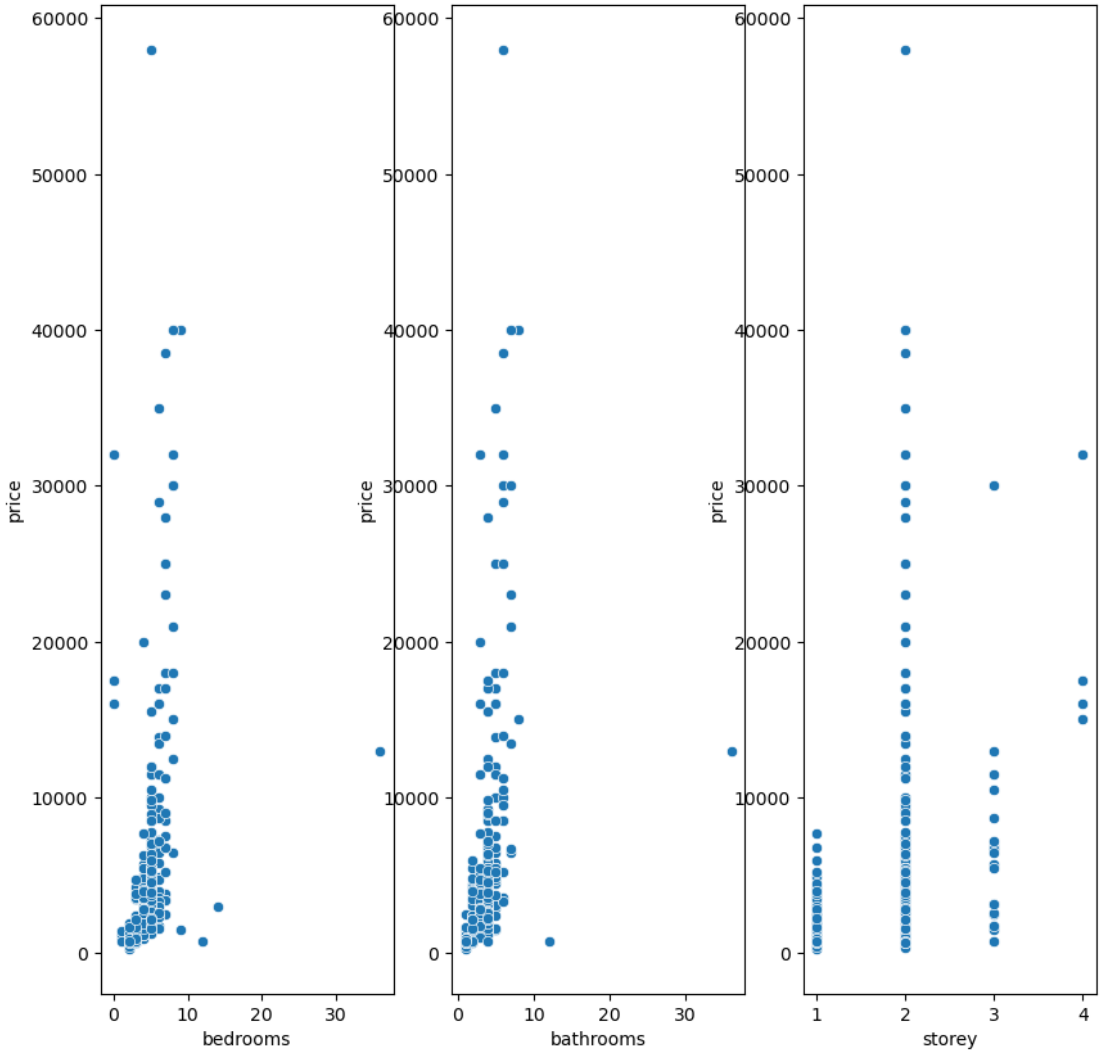
The boxplot shows the summary of data and points out existing outliers.

iv. Scatterplot

Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011

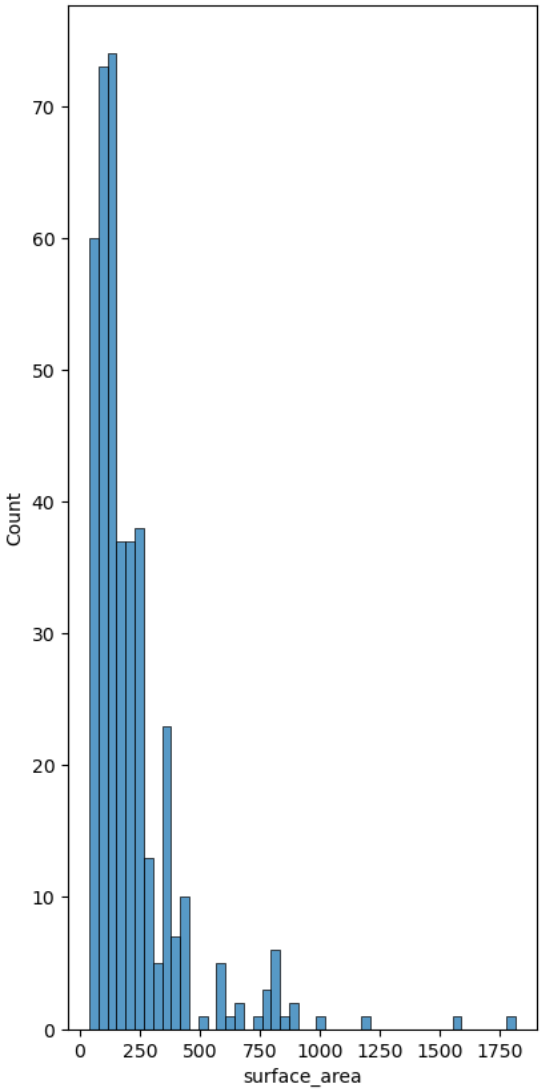


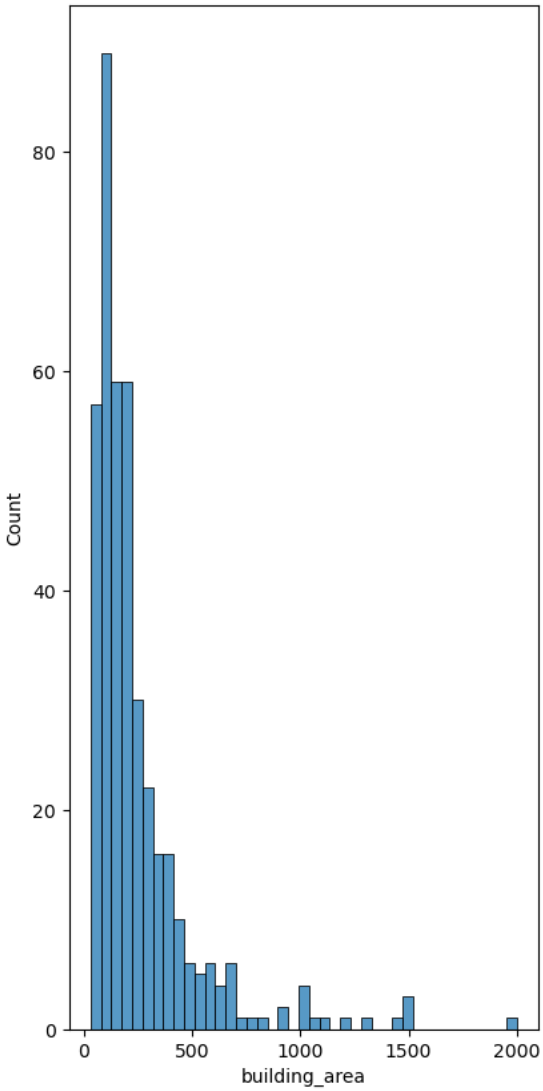
Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011



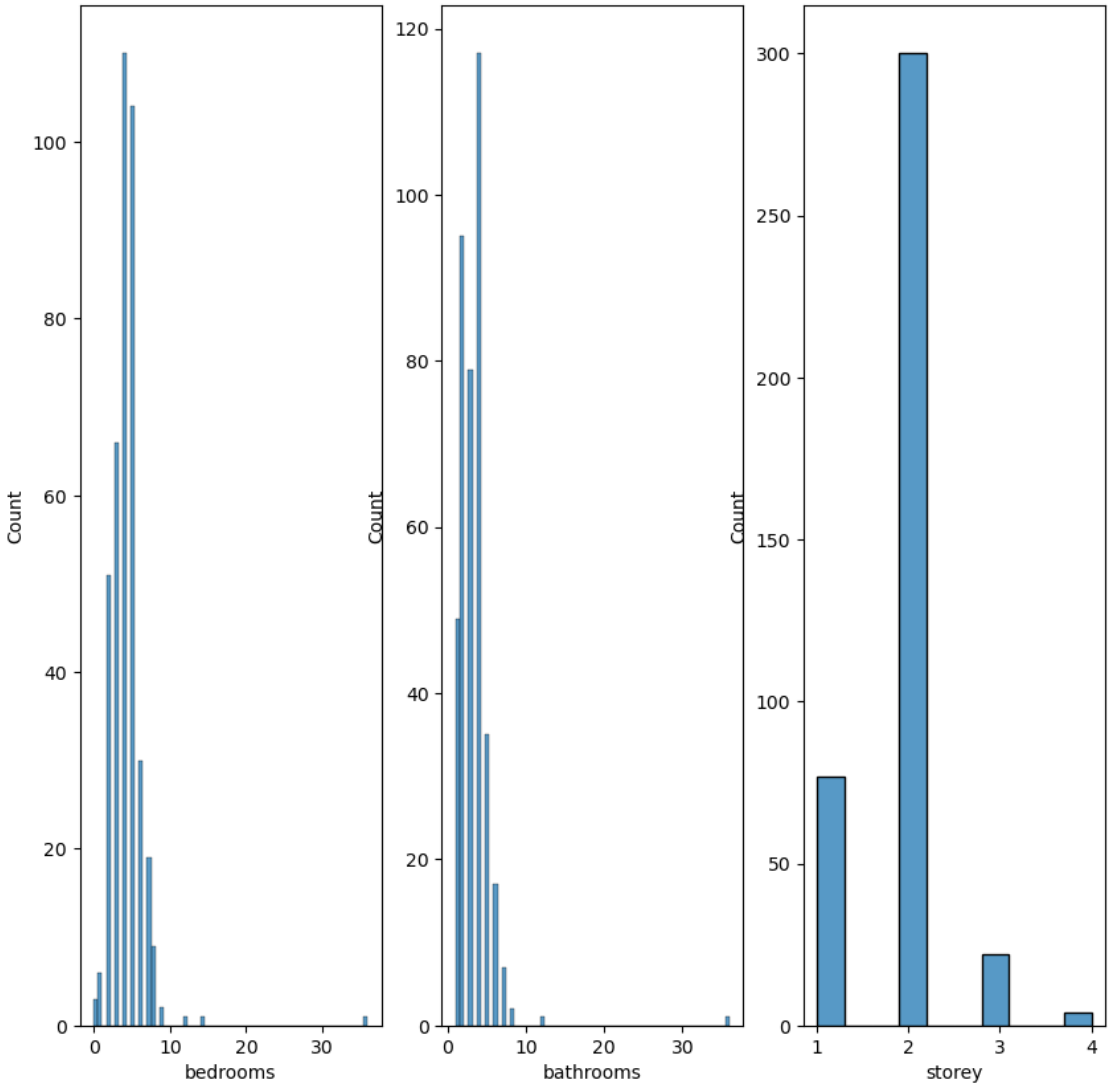
v. Histogram

Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011



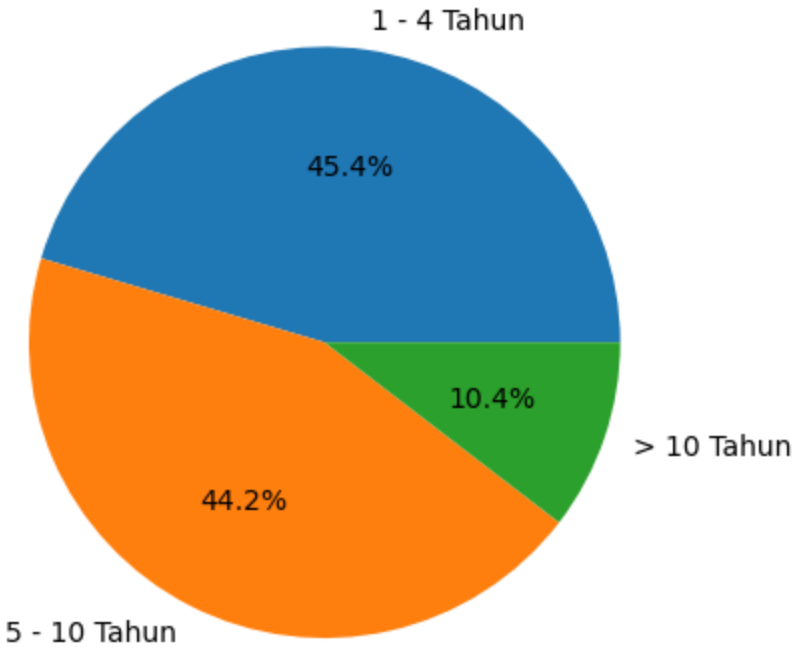


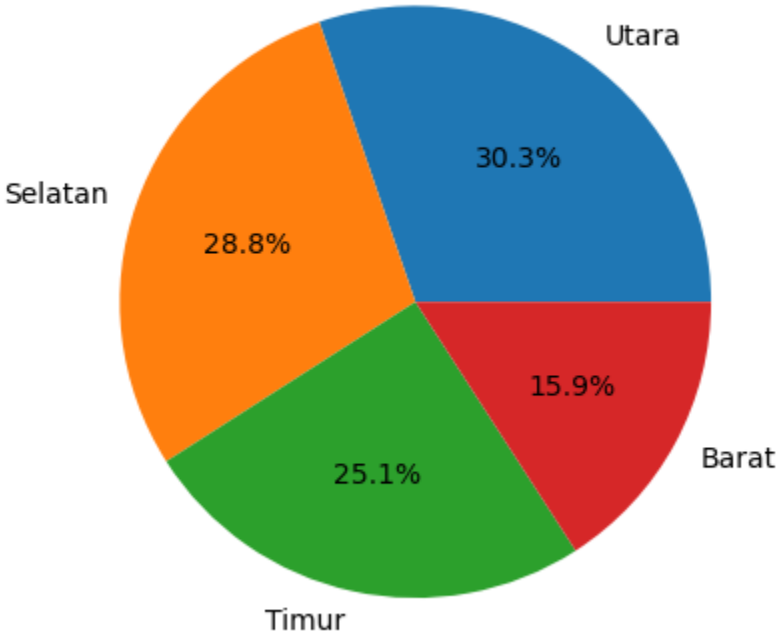
Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011



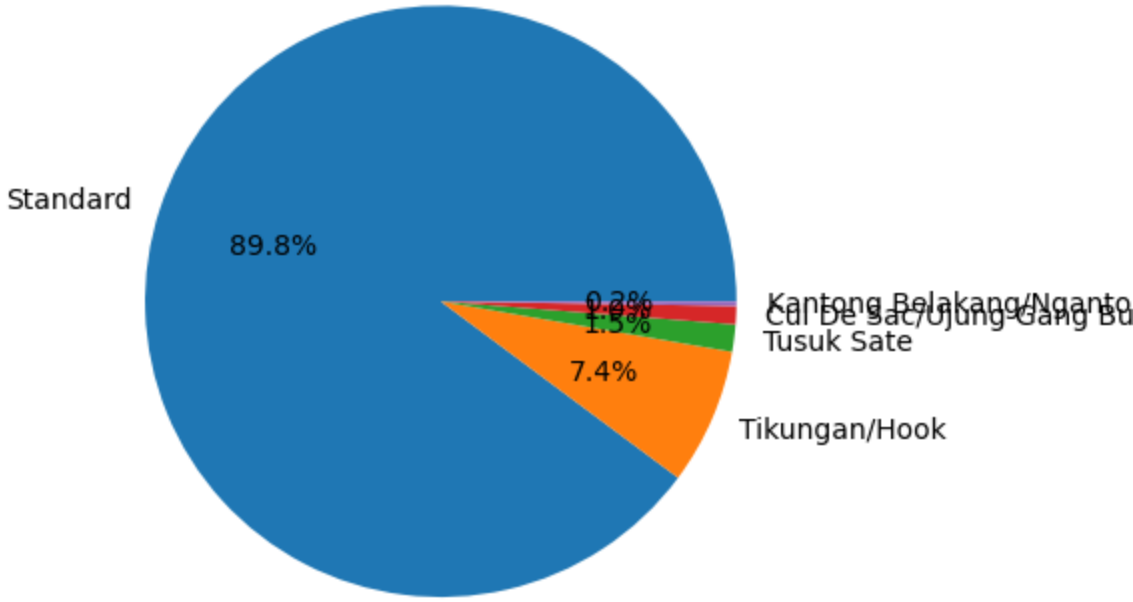
vi. Pie chart

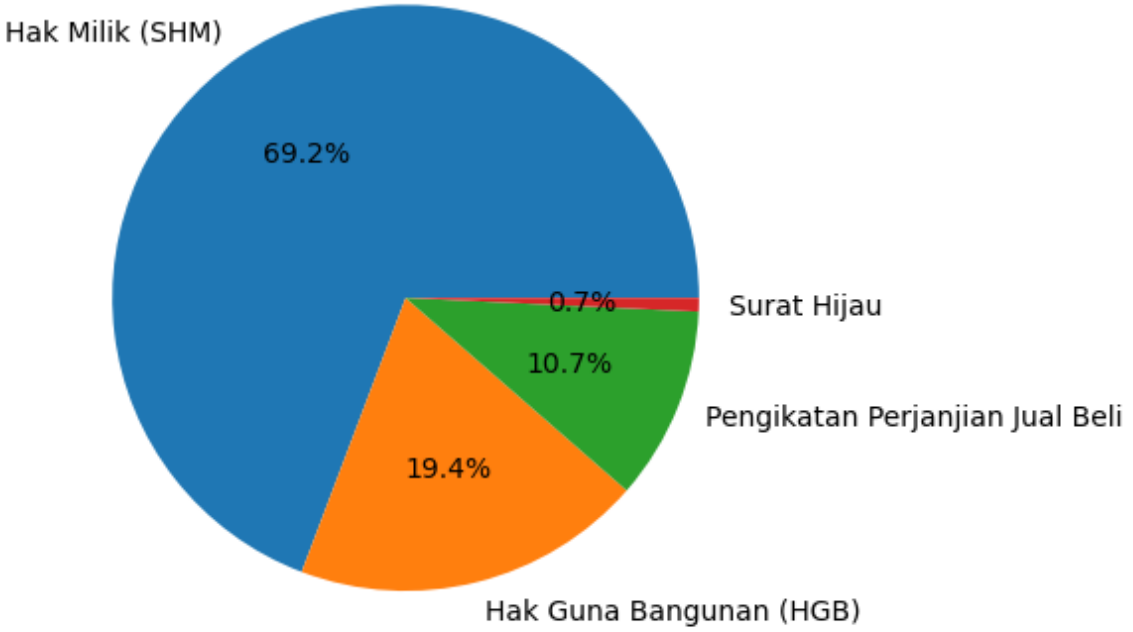
Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011





Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011





b. Data preparation

The dataset has a feature named “cluster_name” that has a high cardinality that would impact the model performance, so I decided to drop it from the start.

Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011

After that, I encoded the categorical features using sklearn LabelEncoder to allow categorical features to be included in the training and validation set. Afterwards, I splitted the whole dataset into X as features and y as target. Reminder that the X and y for regression and classification would be set differently. Hence the dataset will be created independently for regression and classification.

Regression would use price as the target variable and classification would use pricing category as the target variable. Then, I use sklearn train_tree_split to split the X features and y target to training and validation sets with 0.2 test size and 42 as the random state.

C. Solution concept

For regression I used: LinearRegression, KNN, SVR, DecisionTree, and RandomForest.

```
lin_reg = LinearRegression()
lin_reg.fit(X_reg_train, y_reg_train)

knn_reg = KNeighborsRegressor(n_neighbors=5)
knn_reg.fit(X_reg_train, y_reg_train)

svr = SVR(kernel='rbf')
svr.fit(X_reg_train, y_reg_train)

d_tree_reg = DecisionTreeRegressor()
d_tree_reg.fit(X_reg_train, y_reg_train)

r_forest_reg = RandomForestRegressor(n_estimators=100, random_state=42)
r_forest_reg.fit(X_reg_train, y_reg_train)
```

I created the models by importing it from the sklearn library and creating it and storing it in each respective model name as a variable then fitting the exact same regression training set.

For classification I used: SVC, KNN, DecisionTree, RandomForest, and Gaussian Naive Bayes.

Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011

```
svc = SVC()
svc.fit(X_clf_train, y_clf_train)

knn_clf = KNeighborsClassifier(n_neighbors=5)
knn_clf.fit(X_clf_train, y_clf_train)

d_tree_clf = DecisionTreeClassifier()
d_tree_clf.fit(X_clf_train, y_clf_train)

r_forest_clf = RandomForestClassifier(n_estimators=100, random_state=42)
r_forest_clf.fit(X_clf_train, y_clf_train)

gnb = GaussianNB()
gnb.fit(X_clf_train, y_clf_train)
```

I also created the models by importing it from the sklearn library and creating it and storing it in each respective model name as a variable then fitting the exact same classification training set.

For regression I choose to use LinearRegression, because of one simple reason. By including community price as one of the features, there is a higher probability of the prediction to be not far off from the validation sets. Because the community price is a price determined by the community and usually the listing price is not far from the community price. Furthermore, the price of the properties usually increases linearly along with larger surface area, building area and features of the properties.

For classification I choose to use KNN, because regardless of the cluster area properties that have certain features can be identified as the same category. We are talking

about clustering the properties with similar features as one pricing category. For example surface area and building area of similar properties would be considered in one pricing category.

D. Evaluation

For regression evaluation, the metrics I'm using are R^2, mean absolute error, and root mean square error. R^2 is used as a coefficient that determines the proportion of variance in the dependent variable that can be explained by the independent variable. R^2 shows how well the data fit the regression model. Mean absolute error is used to measure the average of errors in a set of predictions without considering the direction. Root mean square error directly shows how far the error goes. As its value increases along as the number of errors increases in the model.

Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011

	Model	R^2	MAE	RMSE
0	LinearRegression	0.956648	634.768353	1353.660548
1	KNeighborsRegressor	0.644650	1843.675333	3875.530363
2	SVR	-0.076844	3119.850211	6746.514343
3	DecisionTreeRegressor	0.889180	816.481481	2164.273513
4	RandomForestRegressor	0.938616	600.556560	1610.765057

For classification evaluation, the metrics I'm using are accuracy, precision, recall, and F1-score. Accuracy is calculated by dividing total correct prediction with total number of samples, accuracy is only good when the number of data is large enough. Precision is the ratio of true positives to the total number of data predicted as positive. Recall is the ratio of true positives to the total number of data that is true positives. F1-score can be simplified by seeing it as the harmonic mean between precision and recall. The higher the F1-score is, the better the model performance usually is.

	Model	Accuracy	Precision	Recall	F1
0	SVC	0.654321	0.654321	0.654321	0.654321
1	KNeighborsClassifier	0.703704	0.703704	0.703704	0.703704
2	DecisionTreeClassifier	0.567901	0.567901	0.567901	0.567901
3	RandomForestClassifier	0.629630	0.629630	0.629630	0.629630
4	GaussianNB	0.641975	0.641975	0.641975	0.641975

From the evaluation of both regression and classification, I can conclude that LinearRegression has the highest R^2 of 0.95, MAE of 634, and RMSE of 1353 which translates to 1.353.000.000 because I divided the initial price with 1.000.000 for better data visualization. Which means LinearRegression is the best model for the dataset. KNN as the best performing model has an accuracy of 0.70, precision of 0.70, recall of 0.70, and F1-score of 0.70.

Course : Machine Learning
Name : Reynaldi Kindarto **NIM** : 0706012010011

References

Agrawal, S. K. (2021, July 20). *Evaluation metrics for classification model: Classification model metrics*. Analytics Vidhya. Retrieved October 31, 2022, from https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

K-Nearest Neighbors (knn) — how to make quality predictions with ... (n.d.). Retrieved October 31, 2022, from https://towardsdatascience.com/k-nearest-neighbors-knn-how-to-make-quality-predictions-with-supervised-learning-d5d2f326c3c2

Mean absolute error. Mean Absolute Error - an overview | ScienceDirect Topics. (n.d.). Retrieved October 31, 2022, from https://www.sciencedirect.com/topics/engineering/mean-absolute-error

Predicting house prices with linear regression | machine learning from ... (n.d.). Retrieved October 31, 2022, from https://towardsdatascience.com/predicting-house-prices-with-linear-regression-machine-learning-from-scratch-part-ii-47a0238aeac1

R-squared. Corporate Finance Institute. (2022, May 8). Retrieved October 31, 2022, from https://corporatefinanceinstitute.com/resources/knowledge/other/r-squared/

RMSE: Root mean square error. Statistics How To. (2022, October 30). Retrieved October 31, 2022, from https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/