Real Estate Evaluation

GROUP 6 MEMBERS

1. Smart Ronald
2. Lubowa Lamech
3. Druscilla Nakanwagi
4. Fred Okorio
5. Cristian Timothy Waniaye
6. Kirabo Calvin

Real Estate Evaluation

Goal:

To develop a machine learning model for predicting house prices.

Dataset: link

Dataset description:

The real estate valuation dataset was collected from Sindian Dist., New Taipei City, Taiwan.

Dataset Overview

X1=the transaction date (for example, 2013.250=2013 March, 2013.500=2013 June, etc.)

X2=the house age (unit: year)

X3=the distance to the nearest MRT(Mass Rapid Transit) station (unit: meter)

X4=the number of convenience stores in the living circle on foot (integer)

X5=the geographic coordinate, latitude. (unit: degree)

X6=the geographic coordinate, longitude. (unit: degree)

The output is as follow

Y= house price of unit area (10000 New Taiwan Dollar/Ping, where Ping is a local unit, 1 Ping = 3.3 meter squared)

Data Exploration

The dataset had no missing values.

The date column whose values are in the year-month format was generated from the transaction date column whose values were in the decimal format.

We dropped the outlier from the price column.

Preparing Data for Machine Learning

We separated the data into the target and feature matrix.

- Target = “price”
- Feature = [“lat”,”lon”,”distance”,”stores”,”age”]

We used train test split to split our data into train and test sets

```
X_train,X_test,y_train,y_test =
train_test_split(X,y,random_state=42,test_size=0.05)
```

Linear regression model

Instantiating the model

```
model = LinearRegression()
```

Training the model

```
model.fit(X_train,y_train)
```

Model Evaluation:

- Prediction on test data
  - `y_pred_test = model.predict(X_test)`
- R-squared score
  - `r2 = r2_score(y_test,y_pred_test) = 0.6694871167578977`
- Root Mean Squared Error
  - `rmse = np.sqrt(mean_squared_error(y_test,y_pred_test)) = 6.884128885343115`

KNN model

Instantiating the model

```
model = KNeighborsRegressor()
```

Training the model

```
model.fit(X_train,y_train)
```

Model Evaluation:

- Prediction on test data
  - `y_pred_test = model.predict(X_test)`
- R-squared score
  - `r2 = r2_score(y_test,y_pred_test) = 0.7728824151195471`
- Root Mean Squared Error
  - `rmse = np.sqrt(mean_squared_error(y_test,y_pred_test)) = 5.706636070233954`

Decision Tree model

Instantiating the model

```
model = DecisionTreeRegressor()
```

Training the model

```
model.fit(X_train,y_train)
```

Model Evaluation:

- Prediction on test data
  - `y_pred_test = model.predict(X_test)`
- R-squared score
  - `r2 = r2_score(y_test,y_pred_test) = 0.8288181733646838`
- Root Mean Squared Error
  - `rmse = np.sqrt(mean_squared_error(y_test,y_pred_test)) = 4.954315098501315`

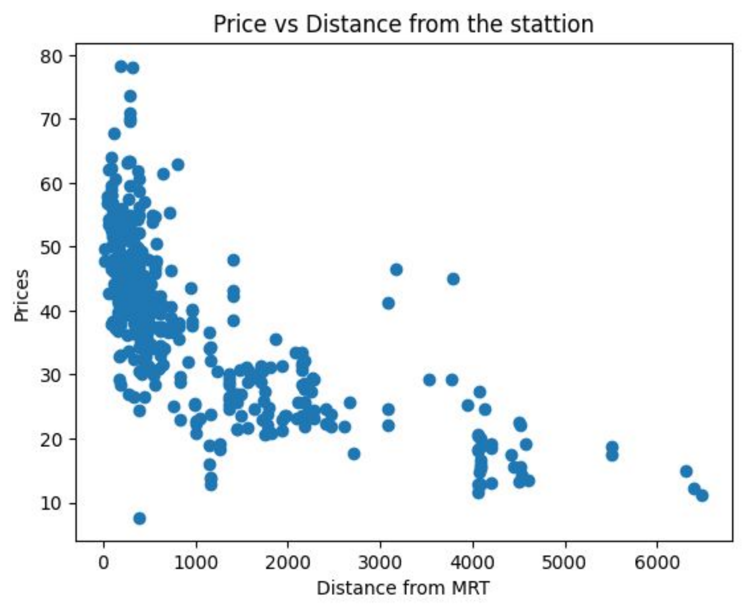
Model Selection

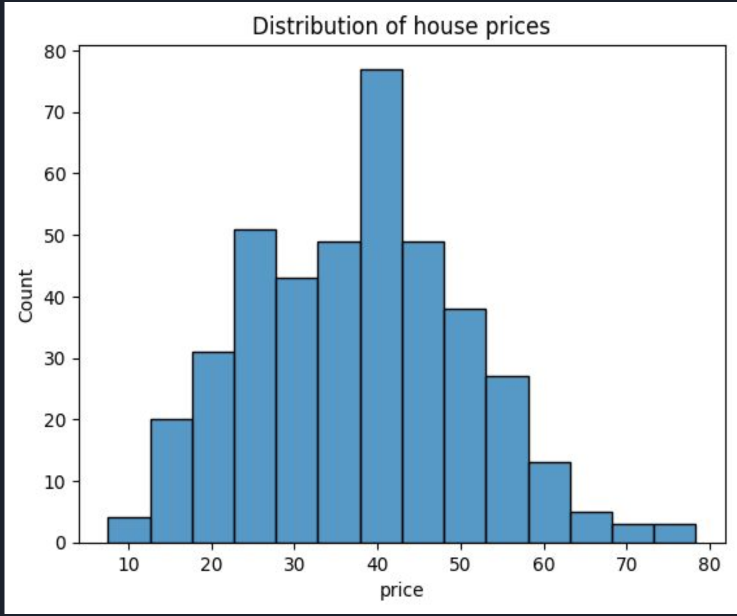
Decision Tree Regressor outperforms both Linear Regression and KNeighbours in terms of both R-squared and Root Mean Squared Error (RMSE).

R-squared represents the proportion of the variance in the target (price) that is explained by the features. While RMSE explains the magnitude of the error between the actual and the predicted values.

Decision Tree Regressor has a superior performance in explaining the variance and predicting house prices accurately.

Insights from the data





Houses which are closer to the MRT station have higher prices

The most houses have a price of 400000 New Taiwan Dollar/Ping