

Predicting HDB Resale Price

**Elang Setiawan
DSIF 2**





Agenda

Project

- **Problem Statement**
- **Result and conclusion**

Lessons Learned

- **Geoprocessing**
- **Mapping**
- **ML Automation**

Predicting HDB Resale Price

- **Property market is heating up**
- **Location, location, location**
- **Test the hypothesis with regression model**



Cooling Measures

Predicting HDB Resale Price

- **HDB property information**

Contains information for all HDB including under development (12k)

- **HDB transacted prices from 1990 to yesterday**

Outlier prices are actual transacted prices

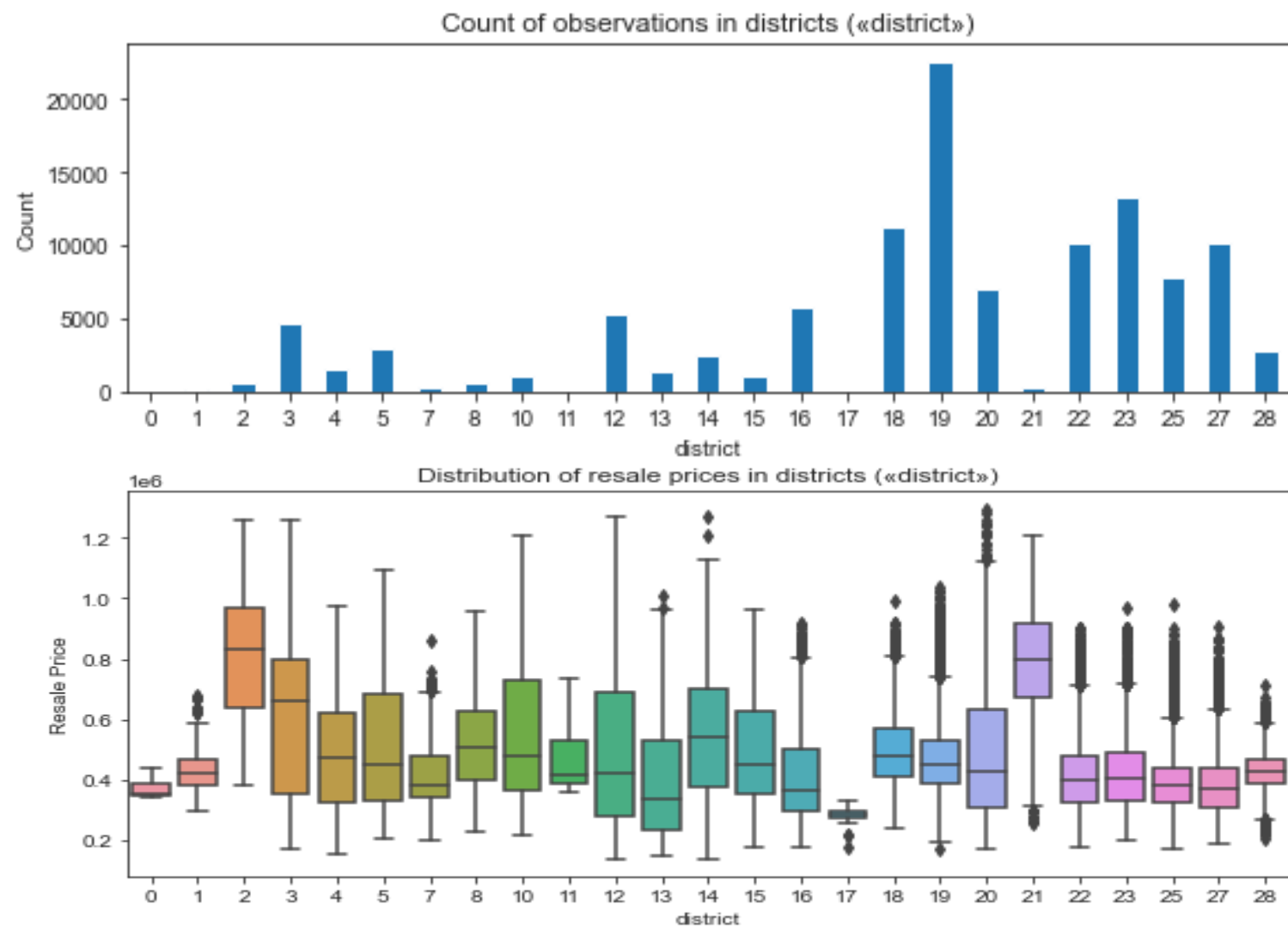
Only data from 2017 is used for modelling (112k)

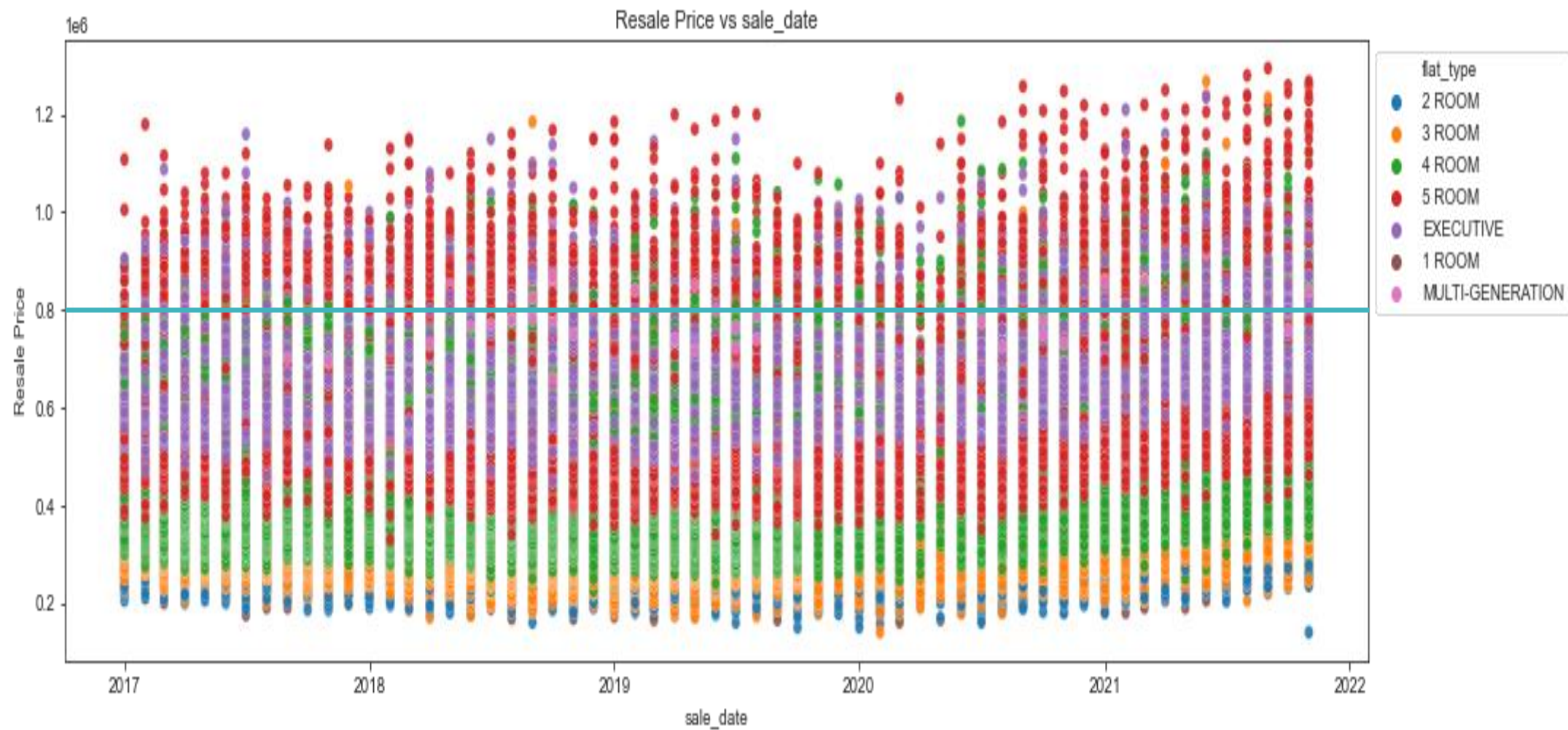
- + **Geographical data from onemap.gov.sg**

- **Query the API using address**

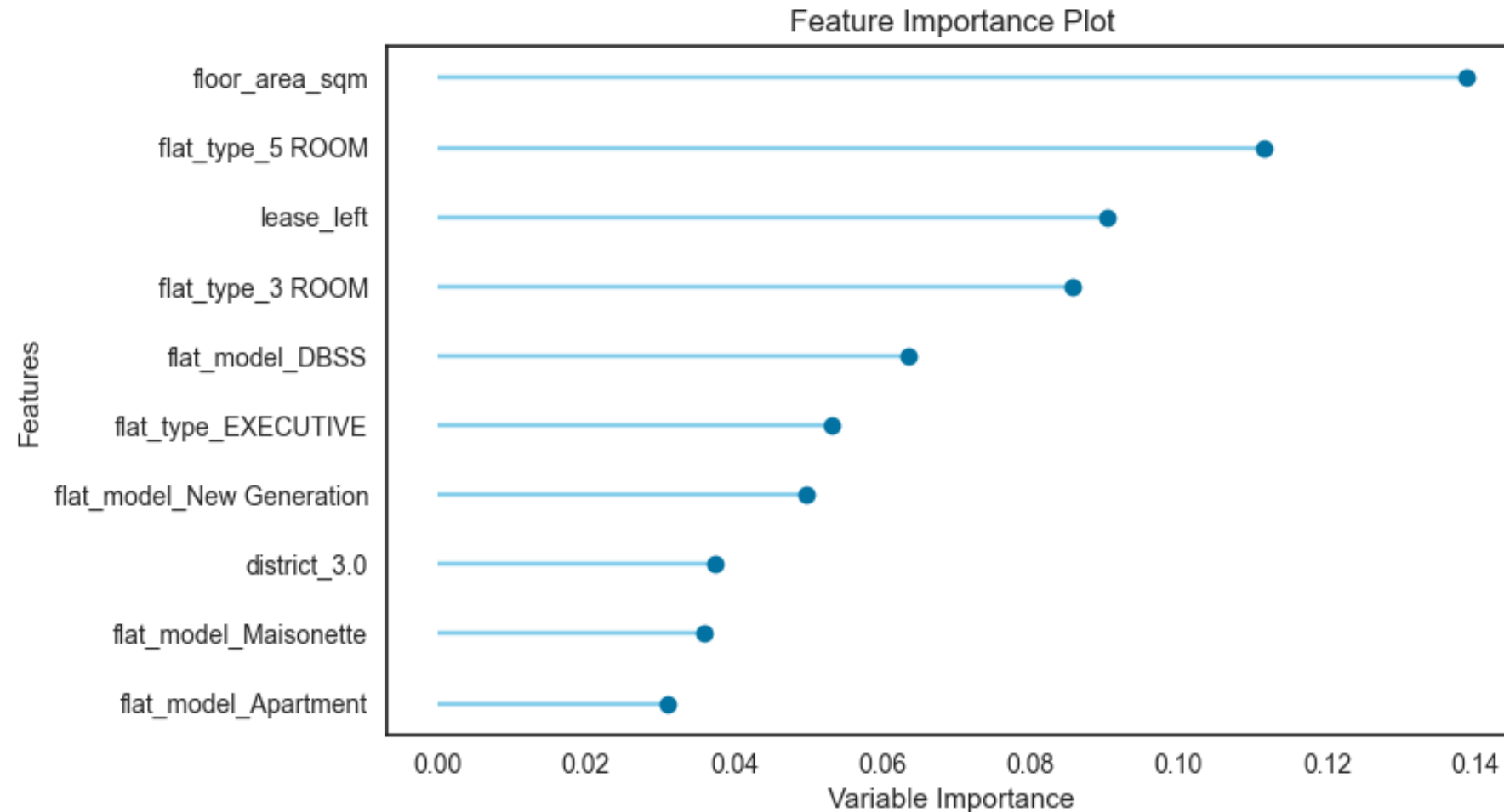
- **Returned data: Latitude, Longitude, Postcode**

- **Mapping**





- + **Features added: district code, number of MRT stations, primary school, and shopping centres within 1km.**
- + **Random Forest Regressor model is best with RMSE difference of 4.37%**



- + In conclusion, the location features do not affect HDB price too much.**
- + Size, type, and how many years to the end of lease are more important.**

Geoprocessing

+ The earth is not flat

World Geodetic System 1984 (WGS 84) used by onemap.gov.sg

Calculate distance:

```
from pyproj import Geod
```

```
from shapely.geometry import Point, LineString
```

```
a = POINT (103.95337, 1.34319) # Simei MRT Station
```

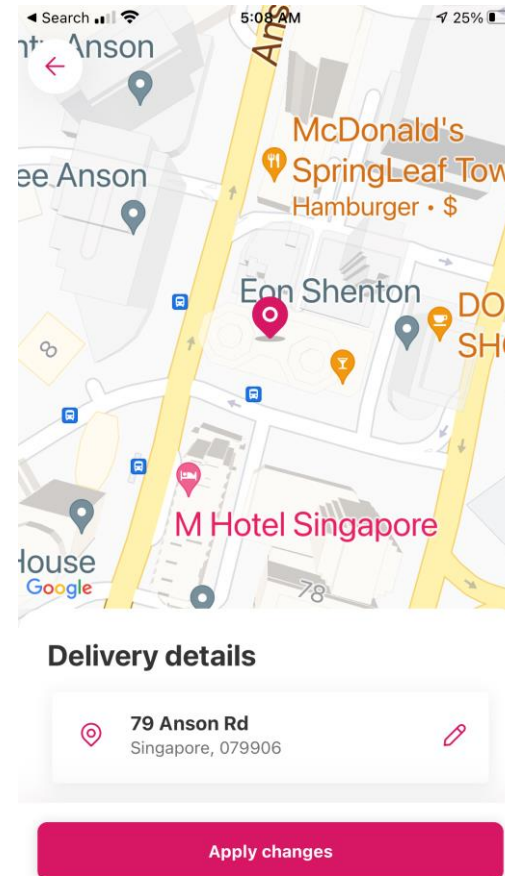
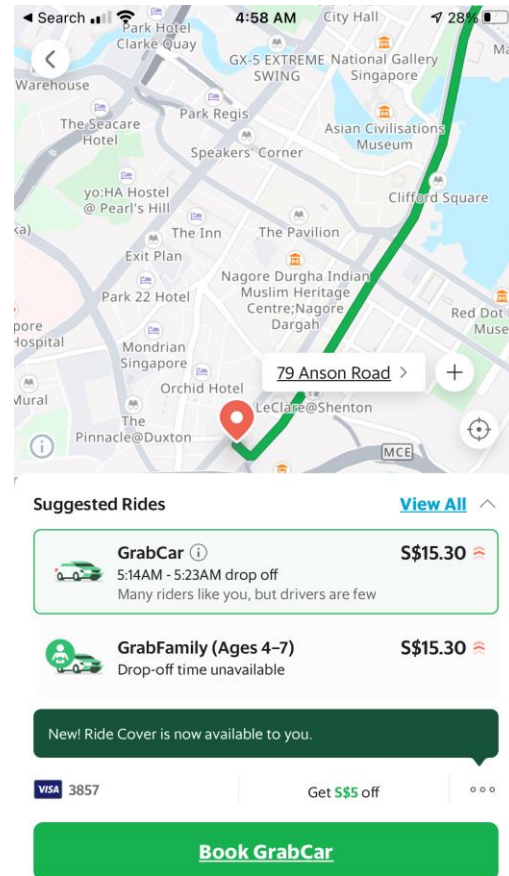
```
b = POINT (103.98809, 1.38861) # 1 Changi Village Road
```

```
geod = Geod(ellps="WGS84")
```

```
distance = geod.geometry_length(LineString([a, b]))/1000 # km
```

Singapore is located at latitude 1.28967 and longitude 103.85007

Mapping



Mapping with Folium

```
import folium
```

```
map = folium.Map(location=[1.28967, 103.85007], zoom_start=14)
```

```
map
```



ML Automation with pycaret

Sample and Split

- Train Test Split
- Sampling

Data Preparation

- Impute Missing Values
- One-Hot Encoding
- Ordinal Encoding
- Cardinal Encoding

Scale and Transform

- Normalization
- Transformation

Feature Engineering

- Feature Interaction
- Polynomial Features
- Group Features
- Bin Numeric Features

Feature Selection

- Feature Importance
- Remove Multicollinearity
- PCA
- Ignore Low Variance

Unsupervised

- Create Clusters
- Remove Outliers

ML Automation with pycaret

Modules

- Classification
- Regression
- Clustering
- Anomaly Detection
- Natural Language Processing
- Association Rule Mining

Automation with pycaret

```
from pycaret.regression import *  
data = df_price.sample(frac=0.9, random_state=random_seed)  
data_unseen = df_price.drop(data.index)  
setup(data = data, target = 'resale_price', session_id=42)  
best = compare_models(exclude = ['ransac'])
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
rf	Random Forest Regressor	23140.7051	1037548773.3689	32206.8675	0.9587	0.0693	0.0521	37.0500
catboost	CatBoost Regressor	24146.2746	1079360778.8293	32848.0380	0.9570	0.0706	0.0542	5.9330
et	Extra Trees Regressor	24662.5136	1171186720.0435	34217.3733	0.9534	0.0739	0.0555	48.6270
xgboost	Extreme Gradient Boosting	26177.1420	1264129702.4000	35548.0387	0.9497	0.0760	0.0587	11.3230
lightgbm	Light Gradient Boosting Machine	29976.8666	1642227201.0760	40520.2103	0.9346	0.0857	0.0668	0.5060
dt	Decision Tree Regressor	30790.2685	1902196054.1036	43610.0498	0.9243	0.0944	0.0693	0.8280

Automation with pycaret

```
rf_reg101 = create_model('rf')
rf_params = {'bootstrap':[True,False],
            'max_depth':[5,6,7],
            'max_features':['log2'],
            'max_leaf_nodes':[None],
            'min_impurity_decrease':[0.0, 0.0001],
            'min_samples_leaf':[5],
            'min_samples_split':[7],
            'n_estimators':[100]
            }
tuned_rf_reg101 = tune_model(rf_reg101, optimize='RMSE', custom_grid = rf_params)
predict_model(tuned_rf_reg101)
```



Summary

The hypothesis that location and other geodata is a strong price predictor is not true.

Use automation to reduce effort.

<https://python-visualization.github.io/folium/>

<https://pycaret.org/>