



## DataHeroes\_JC\_DS\_AH\_6\_FinalProject

by

Bayu Andrianto Wirawan - ([bay.psil@gmail.com](mailto:bay.psil@gmail.com))

Shafwan Hanif - ([shafwanh17@gmail.com](mailto:shafwanh17@gmail.com))



The background of the slide features a low-angle, upward-looking perspective of several tall skyscrapers. On the left, a prominent building has a reddish-pink facade. In the center, a dark, grey building with many windows is visible. On the right, parts of other buildings with brown and green facades are seen. The sky is a pale, clear blue.

# A LITTLE BIT OF BACKGROUND...

Kami adalah pebisnis Start Up bernama AYH Inc (*Acquisition your House*) yang bergerak di bidang jual beli rumah. AYH Inc memberikan alternatif solusi bagi pemilik rumah untuk menjual rumahnya dengan cepat dan mudah. Pemilik rumah dapat menjual rumahnya kepada kami untuk mendapatkan dana liquid yang cepat dibanding mengiklankannya di agency rumah ataupun online media.

# BUSINESS PROBLEMS

## Problem:

- Penjualan yang memakan waktu lama
- Harus mengurus beberapa administrative yang ribet
- Harus memikirkan untuk promosi dan perbaikan rumah

## Solusi:

- Penjualan dapat dilakukan dengan cepat jika menggunakan AYH, karena kita mempunyai dana yang liquid untuk mengakuisi rumah tersebut.

## CUSTOMER SELLING HOUSE



## AYH Inc

### Problem:

- Mengetahui harga pasar
- Memetakan resiko dari setiap rumah
- Memprediksi harga jual

### Solusi:

- Machine Learning

## Customer

### Problem:

- Sulit mencari rumah yang cocok
- Harus mempunyai harga pembeding

### Solusi:

- Pembelian dapat dilakukan dengan cepat dengan menggunakan AYH
- Harga Kompetitive

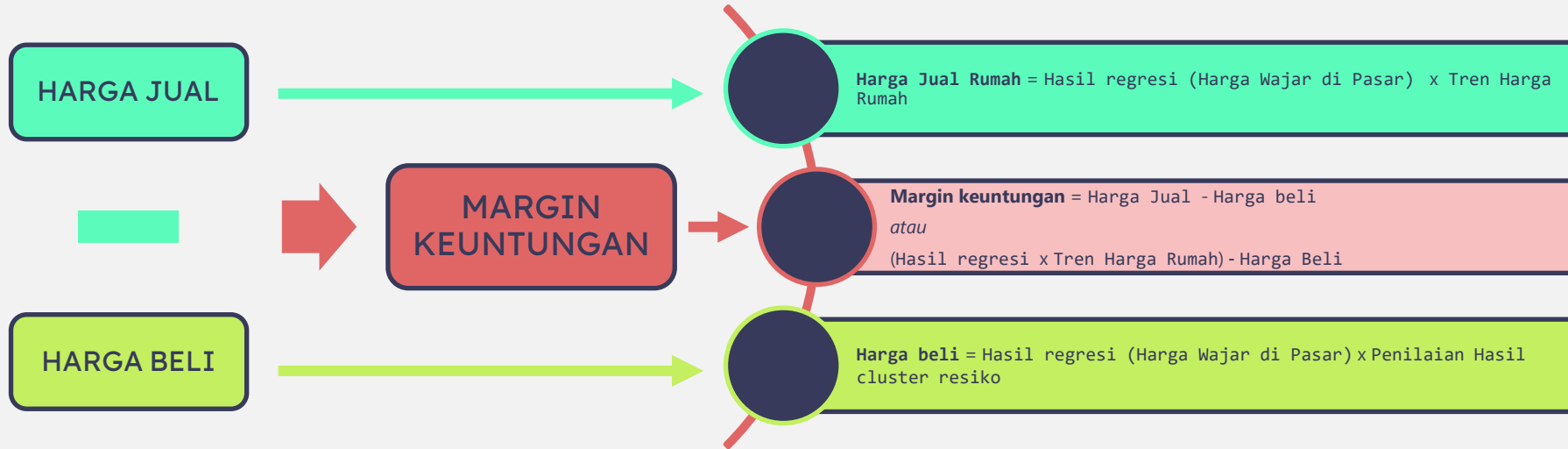
## CUSTOMER BUYING HOUSE

# GOAL SETTING

Untuk dapat menentukan harga beli dan harga jual yang tepat kami setidaknya membutuhkan beberapa hal, yaitu:

- **Model regresi** yang dapat memperkirakan harga wajar rumah di pasar
- **Model clustering** yang dapat memberikan informasi cluster resiko rumah (Rumah resiko tinggi kami akan tawar 50% harga pasar, Rumah resiko sedang 70% dan rumah resiko rendah 80 % dari harga pasar)
- **Model time-series** untuk memberikan informasi tren kenaikan harga

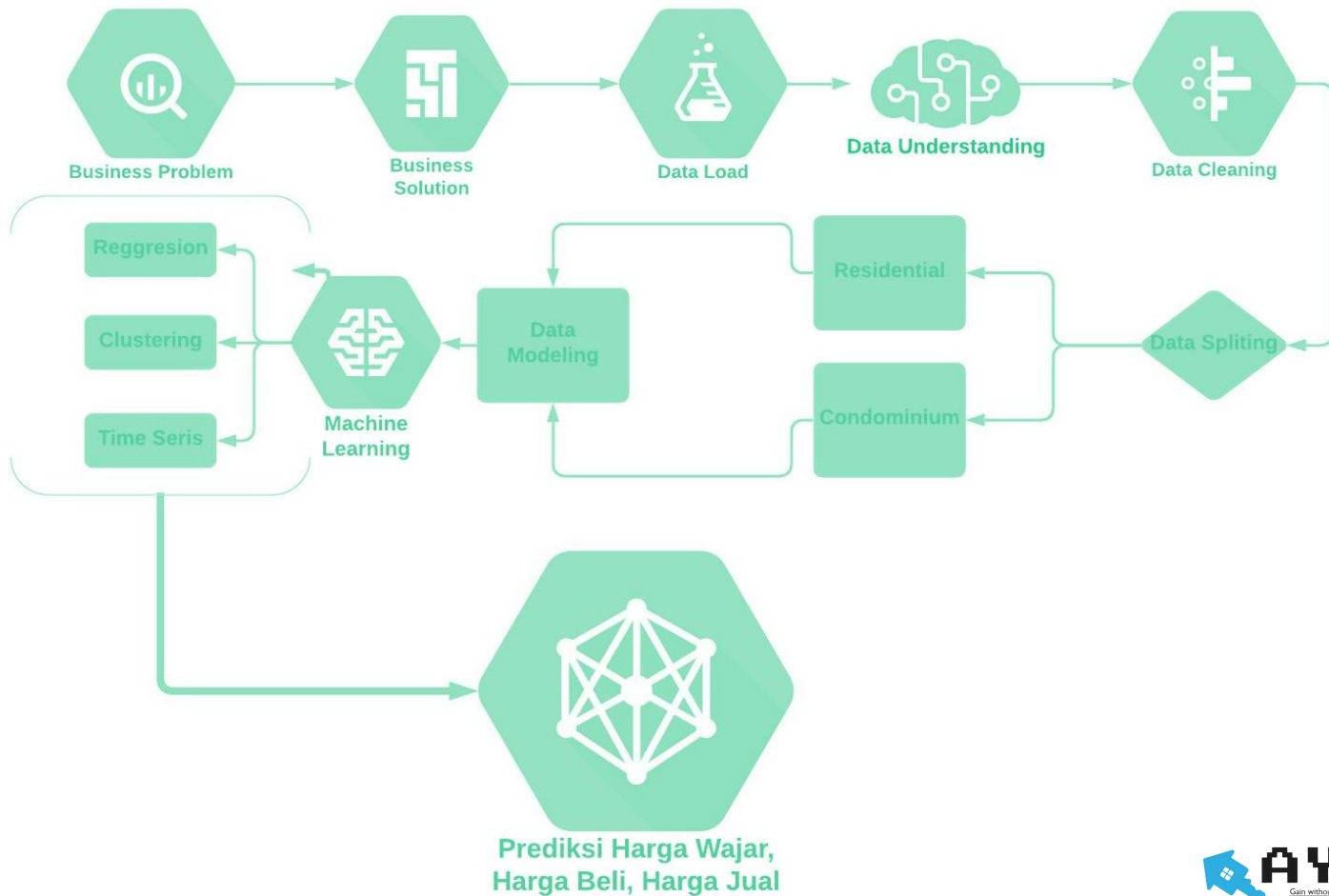
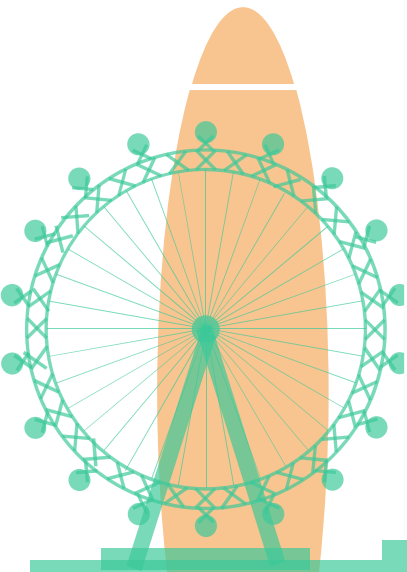
# BUSINESS FORMULATION



**Note:**

**\*(Kami memiliki target setidaknya rumah harus laku 1 tahun sejak kami beli)**

# WORKFLOW PROCESS





# DATA UNDERSTANDING & DATA CLEANING



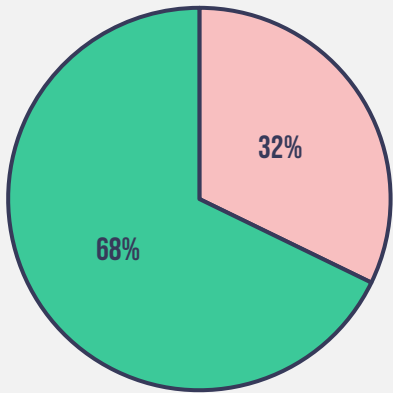
# DATA\_DICT

Kolom	Deskripsi
BATHRM	"Number of Full Bathrooms"
HF_BATHRM	Number of Half Bathrooms (no bathtub or shower)
HEAT	Heating
AC	Cooling
NUM_UNITS	Number of Units
ROOMS	Number of Rooms
BEDRM	Number of Bedrooms
AYB	The earliest time the main portion of the building was built
YR_RMDL	Year structure was remodeled
EYB	The year an improvement was built more recent than actual year built
STORIES	Number of stories in primary dwelling
SALEDATE	Date of most recent sale
PRICE	Price of most recent sale
QUALIFIED	Qualified
SALE_NUM	Sale Number
GBA	Gross building area in square feet
BLDG_NUM	Building Number on Property
STYLE	Style
STRUCT	Structure
GRADE	Grade
CNDTN	Condition
EXTWALL	Exterior wall
ROOF	Roof type
X	Latitude
Y	Longitude
QUADRANT	City quadrant (NE,SE,SW,NW)
INTWALL	Interior wall
KITCHENS	Number of kitchens

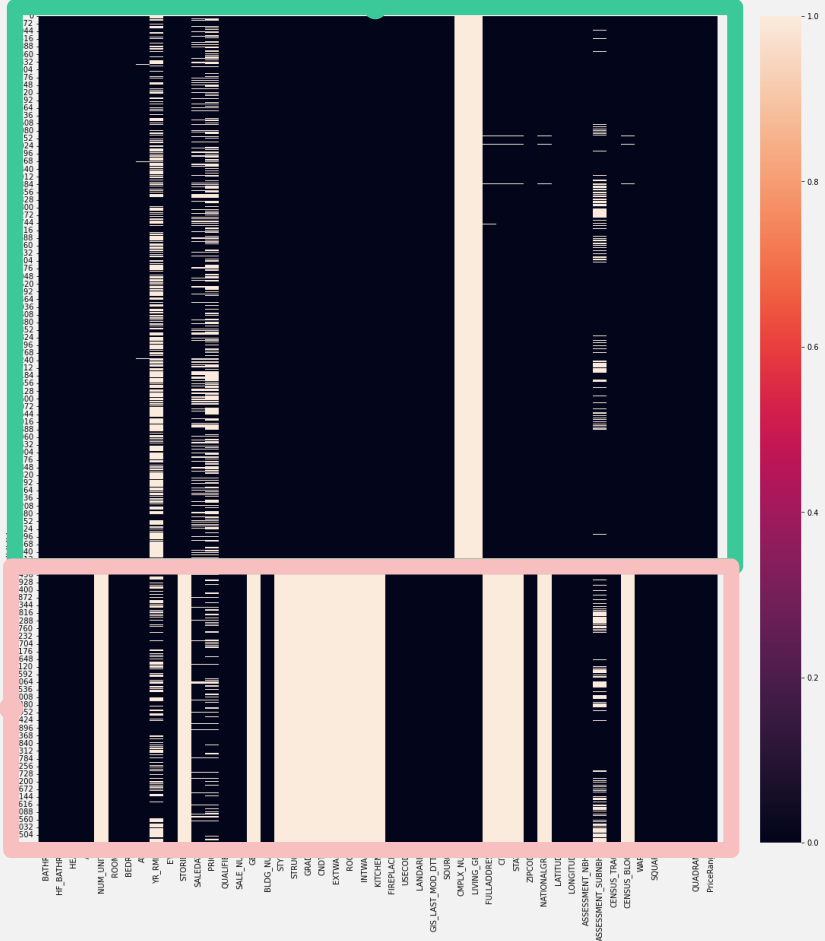
Kolom	Deskripsi
FIREPLACES	Number of fireplaces
USECODE	Property use code
LANDAREA	Land area of property in square feet
GIS_LAST_MOD_DTTM	Last Modified Date
SOURCE	Raw Data Source
CMPLX_NUM	Complex number
LIVING_GBA	Gross building area in square feet
FULLADDRESS	Full Street Address
CITY	City
STATE	State
ZIPCODE	Zip Code
NATIONALGRID	Address location national grid coordinate spatial address
LATITUDE	Latitude
LONGITUDE	Longitude
ASSESSMENT_NBHD	Neighborhood ID
ASSESSMENT_SUBNBHD	Subneighborhood ID
CENSUS_TRACT	Census tract
CENSUS_BLOCK	Census block
WARD	"Ward (District is divided into eight wards, each with approximately 75,000 residents) "
SQUARE	Square (from SSL - (Square, Suffix, Lot))

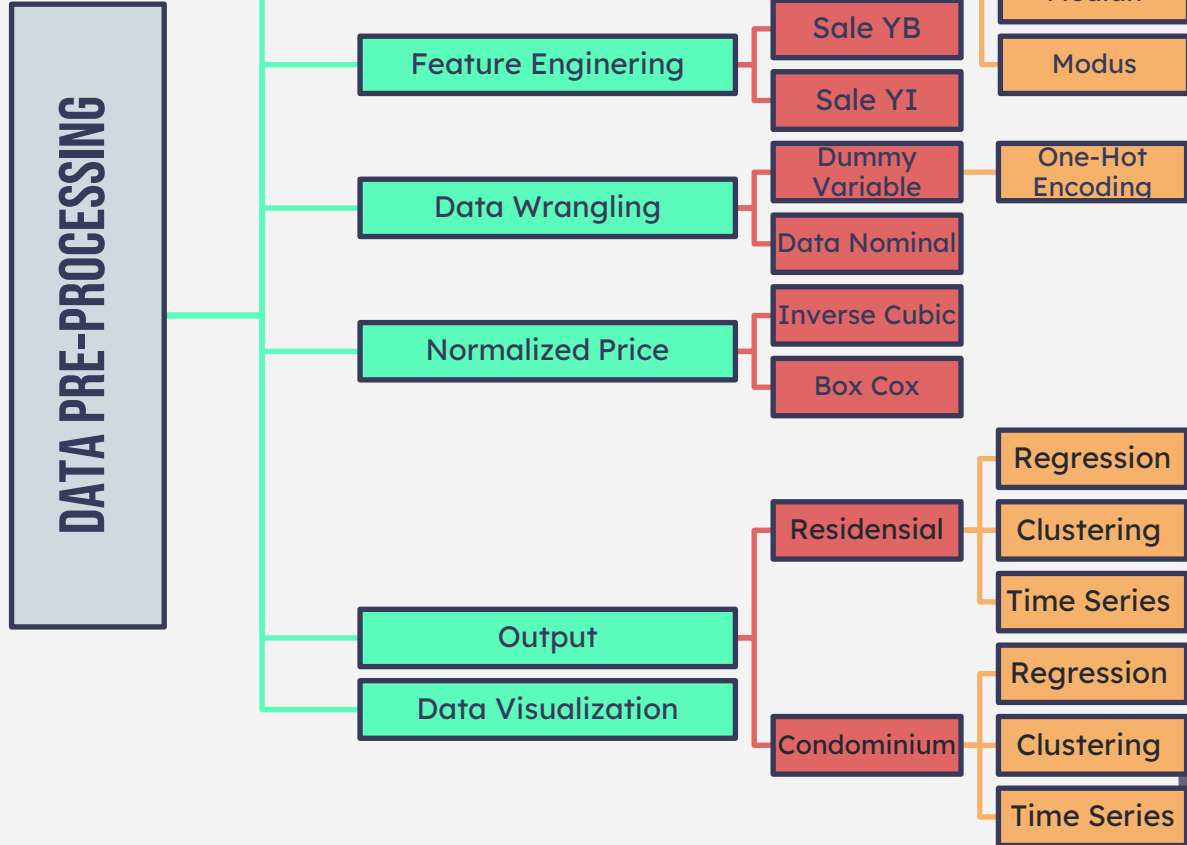


## SOURCE

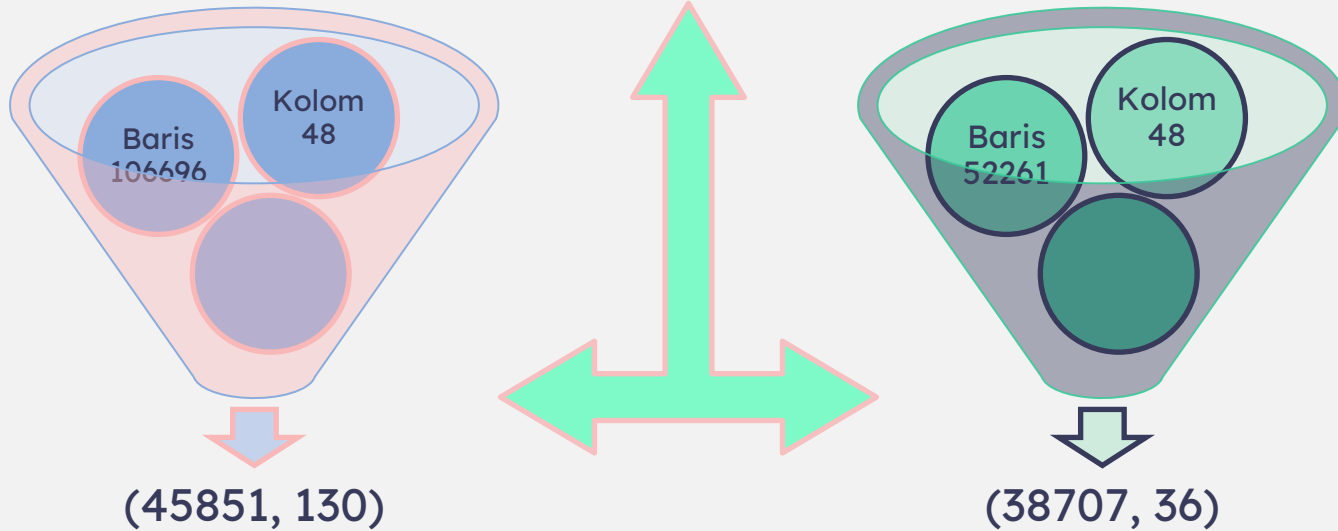


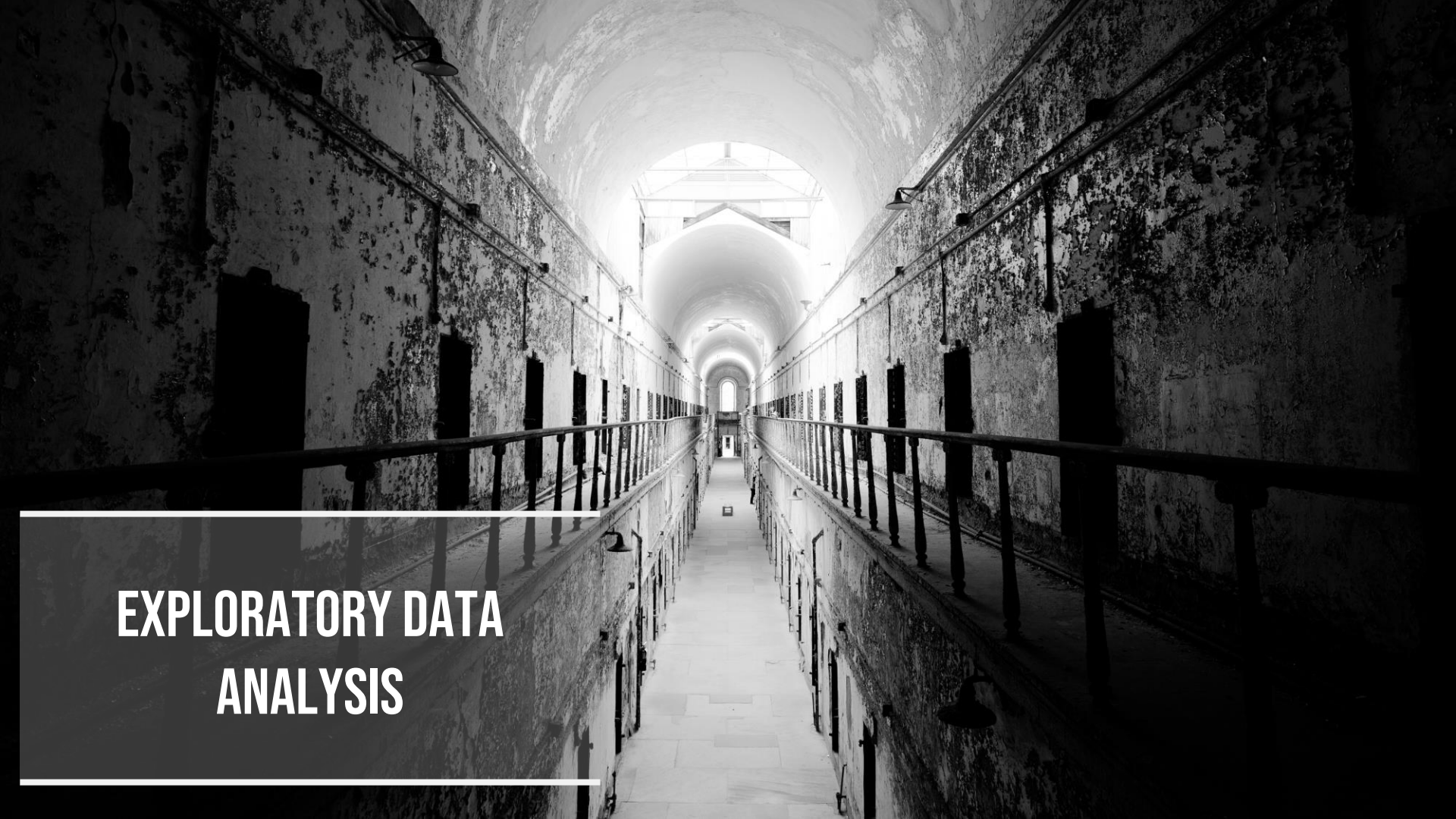
## CONDOMINIUM





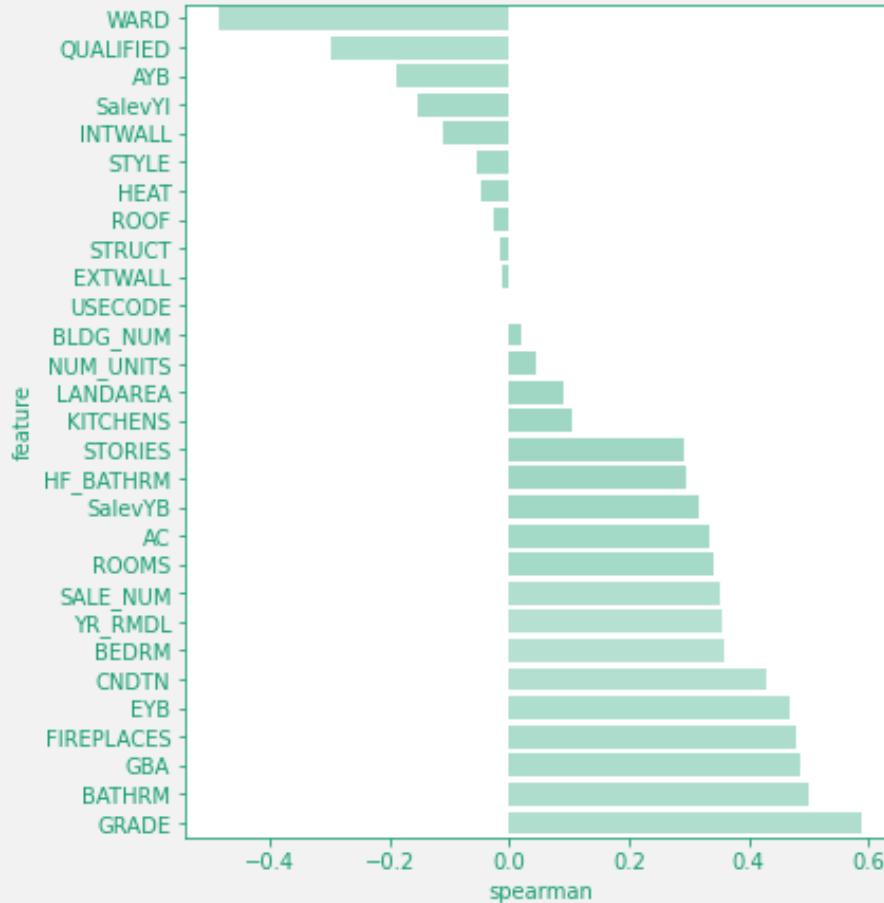
# PRE-PROCESSING RESULT RESIDENTIAL VS CONDOMINIUM



A black and white photograph of a long, arched prison corridor. The corridor features two levels of cell blocks on either side, with a central walkway. The walls are made of rough, textured stone or concrete, and the ceiling is a series of repeating arches. The lighting is dramatic, with bright light coming from the far end of the corridor, creating a strong sense of perspective and depth. The overall atmosphere is somber and institutional.

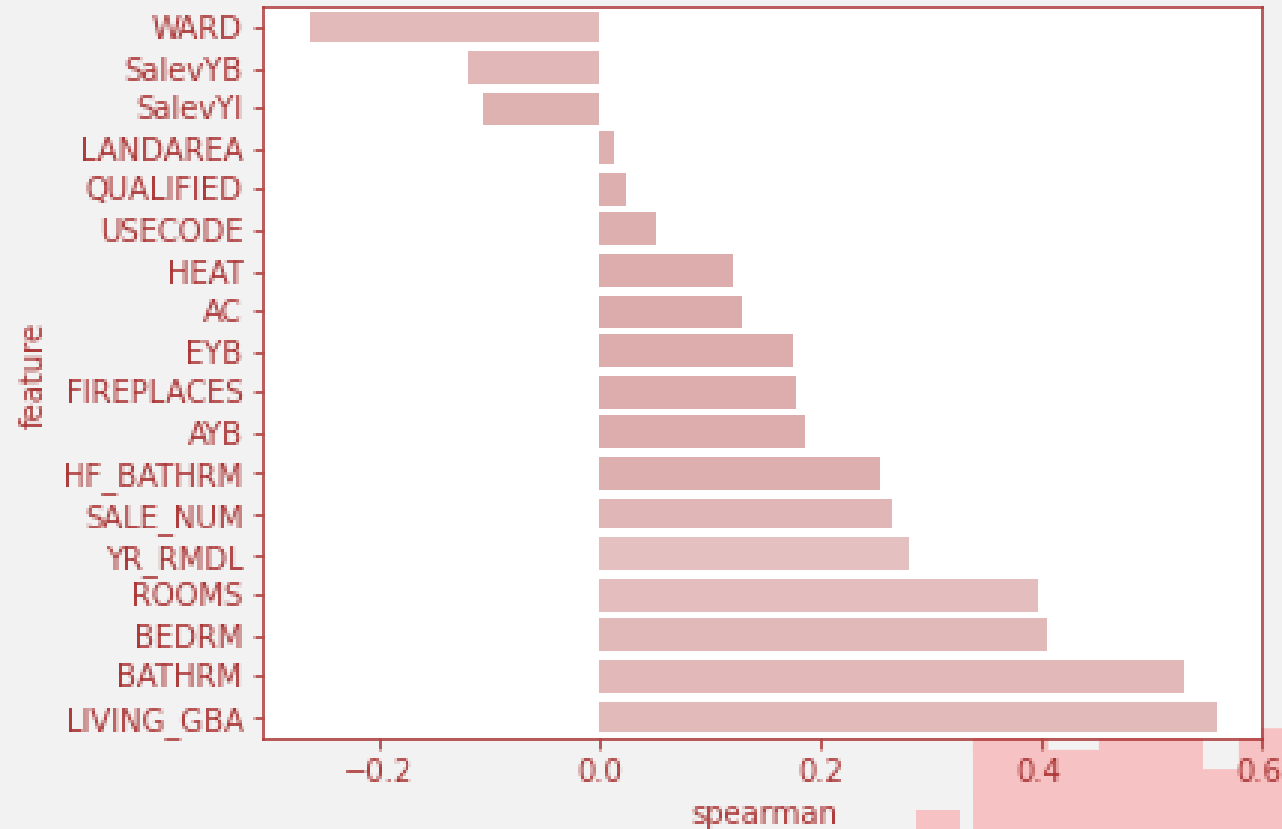
# **EXPLORATORY DATA ANALYSIS**

# RESIDENTIAL\_CORR

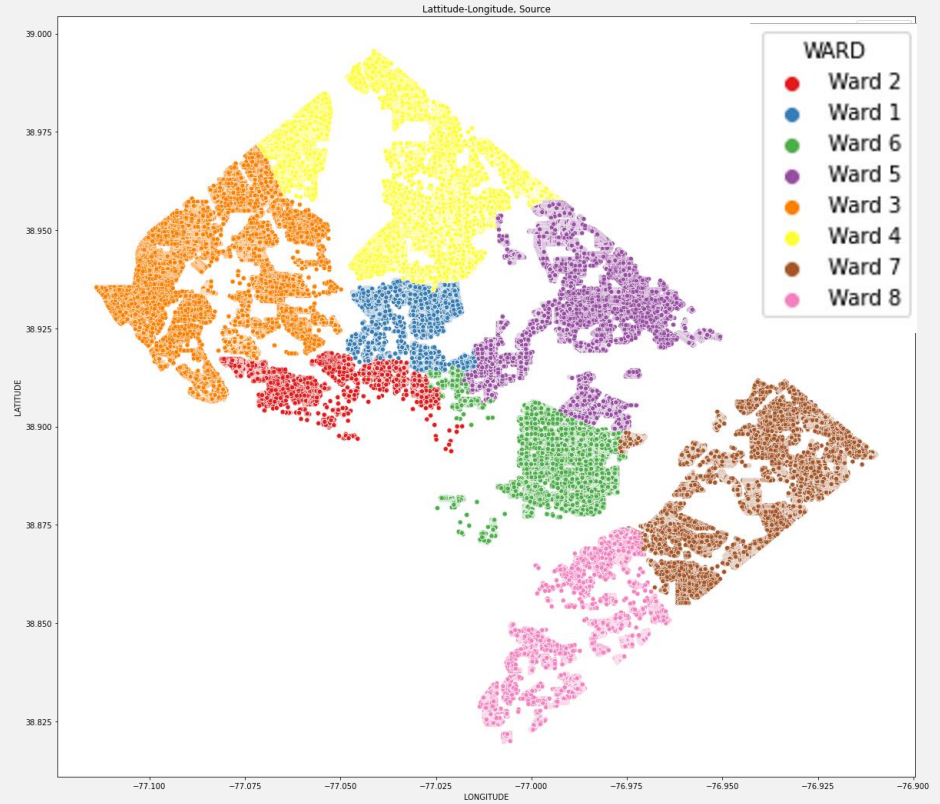
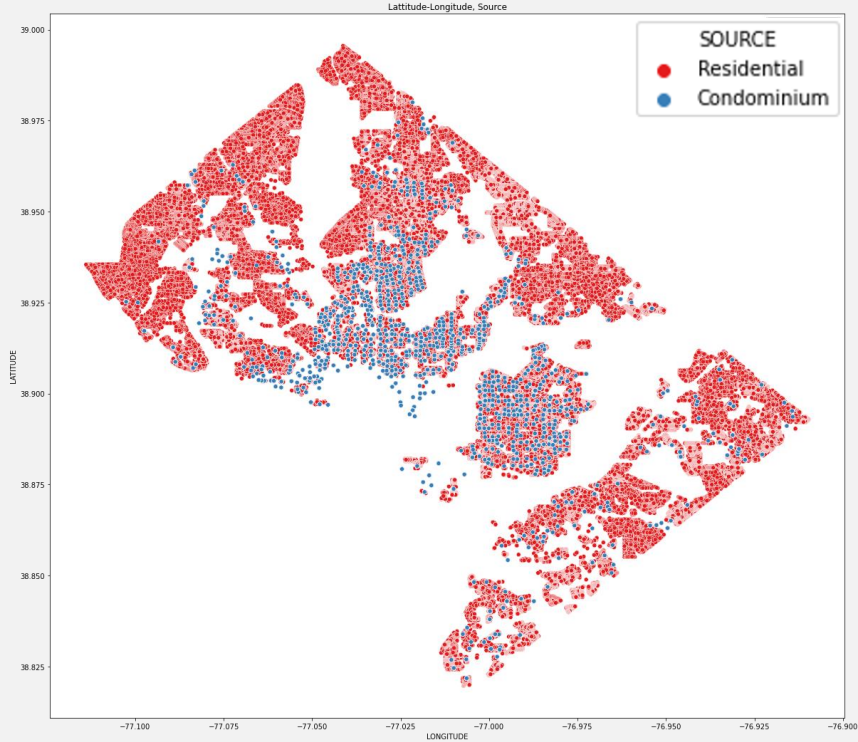




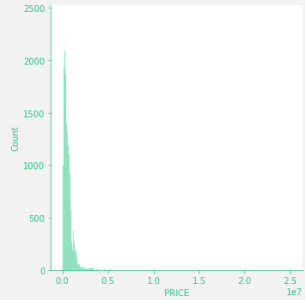
# CONDOMINIUM\_CORR



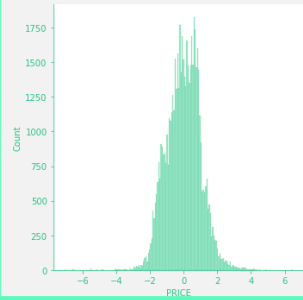
# MAP



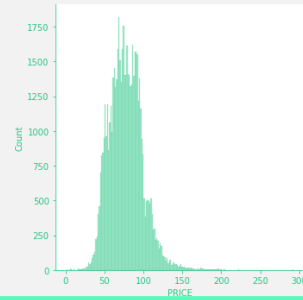
# NORMALISASI DATA



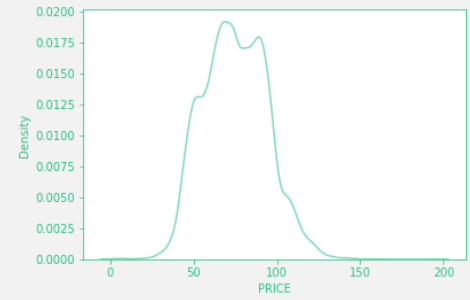
Residential



Box Cox



Inverse Cubic



Final Result\_Res

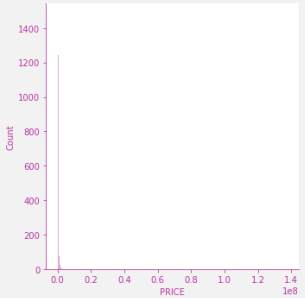
Base Data\_Price



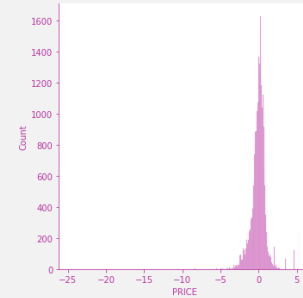
Box-cox & Inverse Cubic  
*Kami memilih metode Inverse Cubic*



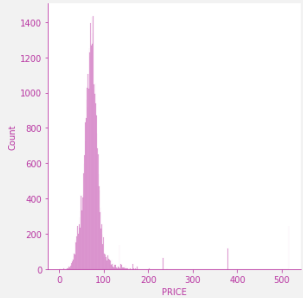
Remove Outlier



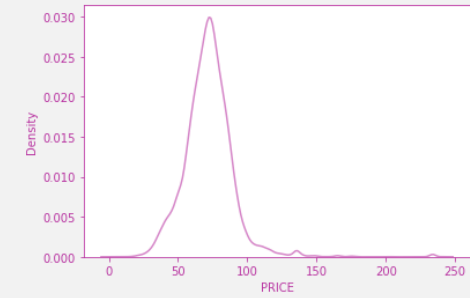
Condominium



Box Cox



Inverse Cubic



Final Result\_Con

# ML MODELING

# ML Modeling

OLS / Linear  
Regression

Ridge

Lasso

DecisionTreeRegressor

KNN

SVM

Random Forest

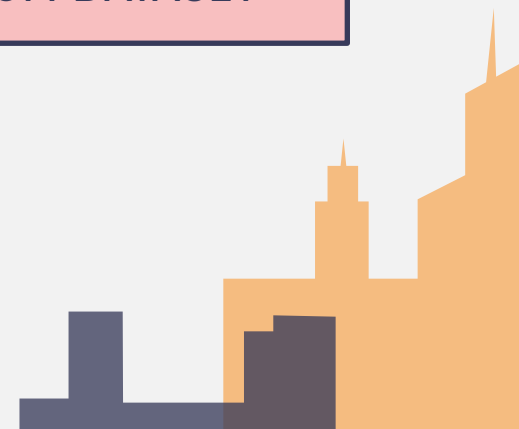
Adaboost

XGB Regressor



RESIDENTIAL DATASET

CONDOMINIUM DATASET





# RESIDENTIAL\_REG

\*without\_handling

Model	R2	MAE	MSE	RMSE	Best Estimator Hyperparameter Tuning
<b>OLS / Linear Regression</b>	0.414	7.834	188.608	13.733	-
<b>Ridge</b>	0.832	5.745	61.766	7.859	Ridge(alpha=0.9942074384219376)
<b>Lasso</b>	0.831	5.781	62.402	7.900	Lasso(alpha=0.036608530436010756)
<b>DecisionTreeRegressor</b>	0.767	4.666	75.175	8.670	DecisionTreeRegressor(max_depth=20)
<b>KNN</b>	0.695	5.623	98.162	9.908	KNeighborsRegressor(n_neighbors=3, weights='distance')
<b>SVM</b>	0.266	13.191	270.663	16.452	SVR(kernel='rbf', deegree = 3, epsilon=0.1)
<b>Random Forest</b>	0.877	3.647	39.611	6.294	RandomForestRegressor(max_depth=20, max_features='sqrt', n_estimators=500)
<b>Adaboost</b>	0.859	3.830	45.417	6.739	AdaBoostRegressor(base_estimator=DecisionTreeRegressor, n_estimators=100)
<b>XGB Regressor</b>	0.869	4.751	48.164	6.940	xgb(eta=0.3, max_depth=6, min_child_weight=1)

# RESIDENTIAL\_REG

\*with\_handling

model	R2	MAE	MSE	RMSE	Best Estimator Hyperparameter Tuning
<b>OLS / Linear Regression</b>	0.683	8.261	116.757	10.805	
<b>Ridge</b>	0.683	8.261	116.757	10.805	Ridge(alpha=0.036608530436010756)
<b>Lasso</b>	0.682	8.284	117.078	10.820	Lasso(alpha=0.036608530436010756)
<b>DecisionTreeRegressor</b>	0.534	9.064	171.865	13.110	DecisionTreeRegressor(max_depth=20)
<b>KNN</b>	0.395	11.374	223.201	14.940	KNeighborsRegressor(n_neighbors=7, weights='distance')
<b>SVM</b>	0.259	13.258	273.340	16.533	SVR(kernel='rbf', degree = 3, epsilon=0.1)
<b>Random Forest</b>	0.746	6.983	93.821	9.686	RandomForestRegressor(max_depth=20, max_features='sqrt', n_estimators=277)
<b>Adaboost</b>	0.733	7.050	98.457	9.923	AdaBoostRegressor(base_estimator=DecisionTreeRegressor, n_estimators=266)
<b>XGB Regressor</b>	0.741	7.114	95.642	9.780	xgb(eta=0.3, max_depth=6, min_child_weight=1)

# CONDOMINIUM\_REG

\*without\_handling

Model	R2	MAE	MSE	RMSE	Best Estimator Hyperparameter Tuning
<b>OLS / Linear Regression</b>	0.414	7.841	1.889	13.743	
<b>Ridge</b>	0.508	6.742	158.550	12.592	Ridge(alpha=0.9942074384219376)
<b>Lasso</b>	0.461	7.385	173.623	13.177	Lasso(alpha=0.4903677932571596)
<b>DecisionTreeRegressor</b>	0.748	4.612	81.063	9.003	DecisionTreeRegressor(max_depth=20)
<b>KNN</b>	0.709	5.698	93.595	9.674	KNeighborsRegressor(n_neighbors=7, weights='distance')
<b>SVM</b>	0.316	8.939	220.159	14.838	SVR(kernel='rbf', degree = 3, epsilon=0,1)
<b>Random Forest</b>	0.873	3.597	40.933	6.398	RandomForestRegressor(max_depth=20, n_estimators=366)
<b>Adaboost</b>	0.857	3.871	46.013	6.783	AdaBoostRegressor(base_estimator=DecisionTreeRegressor, n_estimators=100)
<b>XGB Regressor</b>	0.876	3.856	39.824	6.311	xgb(eta=0.3, max_depth=6, min_child_weight=1)

# CONDOMINIUM\_REG

\*with\_handling

model	R2	MAE	MSE	RMSE	Best Estimator Hyperparameter Tuning
<b>OLS / Linear Regression</b>	0.414	7.834	188.608	13.733	
<b>Ridge</b>	0.507	6.767	158.792	12.601	Ridge(alpha=0.9942074384219376)
<b>Lasso</b>	0.505	6.757	159.306	12.622	Lasso(alpha=0.036608530436010756)
<b>DecisionTreeRegressor</b>	0.767	4.666	75.175	8.670	DecisionTreeRegressor(max_depth=20)
<b>KNN</b>	0.695	5.623	98.162	9.908	KNeighborsRegressor(n_neighbors=3, weights='distance')
<b>SVM</b>	0.317	8.936	220.098	14.836	SVR(kernel='rbf', deegree = 3, epsilon=0.1)
<b>Random Forest</b>	0.877	3.647	39.611	6.294	RandomForestRegressor(max_depth=20, max_features='sqrt', n_estimators=500)
<b>Adaboost</b>	0.859	3.830	45.417	6.739	AdaBoostRegressor(base_estimator=DecisionTreeRegressor, n_estimators=100)
<b>XGB Regressor</b>	0.874	3.973	40.720	6.381	xgb(eta=0.3, max_depth=6, min_child_weight=1)

# CLUSTERING RESULT (K=3)

## RESIDENTIAL\_

model	Silhouette Score	Calinski Harabasz Score	Davies Bouldin Score
Kmeans	0.6057	3158.6073	0.4459
Hierarchical Clustering	0.5261	3509.4729	0.4610
GMM	-0.5940	171.3862	2.9834

## CONDOMINIUM\_

model	Silhouette Score	Calinski Harabasz Score	Davies Bouldin Score
KMeans	0.6677	4611.1489	0.4521
Hierarchical Clustering	0.6258	4592.2298	0.3785
GMM	-0.5499	374.4068	3.7613

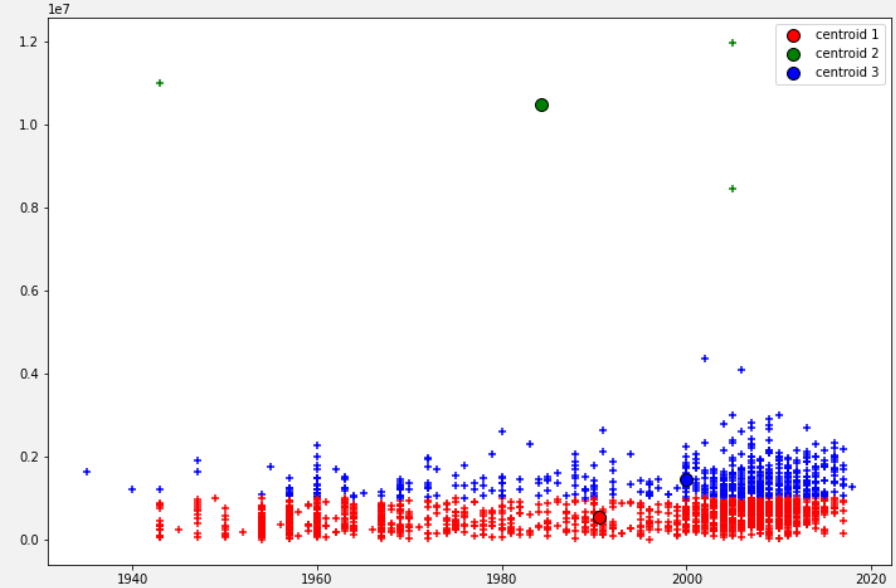


# CLUSTERING\_RES

1 Low Risk - 80% Market Value

3 Medium Risk - 70% Market Value

2 High Risk - 50% Market Value



cluster	PRICE	BATHRM	HF_BATHRM	HEAT	AC	NUM_UNITS	ROOMS	BEDRM	AYB	LANDAREA	WARD	SalevYB	SalevYI
1	1.46E+06	3.147826	0.770435	0.963478	0.902609	1.867826	9.325217	4.215652	1901.837	1607.358	1.071304	111.5791	40.67826
2	5.50E+05	2.126493	0.626845	1.132818	0.789178	1.419536	7.366128	3.248067	1912.365	1317.814	1.955025	94.05833	37.63739
3	1.05E+07	5	1.333333	0.333333	0.666667	1	14.33333	4.666667	1920.333	3530	2.333333	92.33333	30.33333

# CLUSTERING\_CON

1

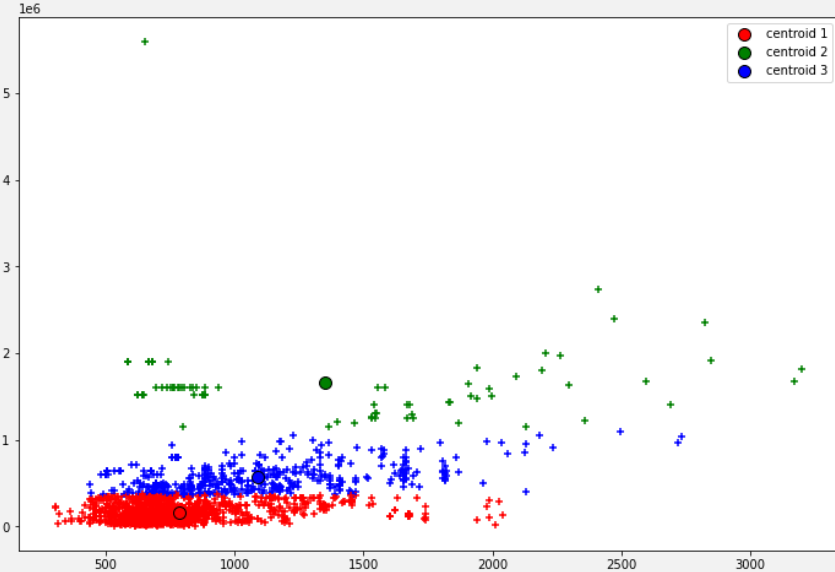
Low Risk - 80% Market Value

2

High Risk - 50% Market Value

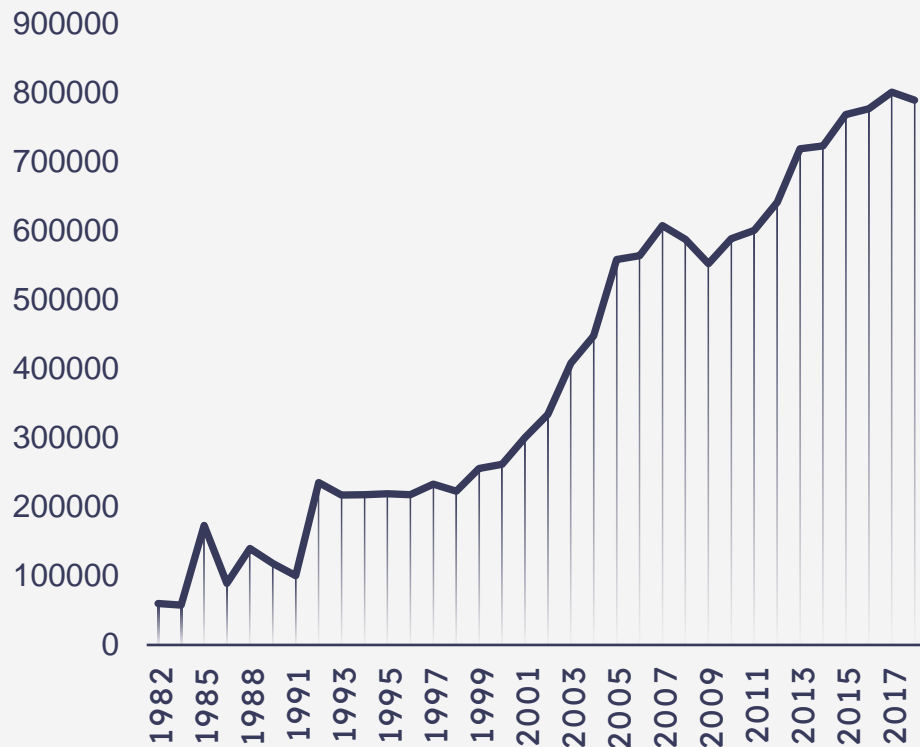
3

Medium Risk - 70% Market Value



cluster	BATHRM	LANDAREA	YR_RMDL	USECODE	EYB	LIVING_GBA	SaleYB	AYB	PRICE
1	1.143826	917.135007	1995.893487	16.186567	1965.117368	784.931479	46.249661	1963.005427	1.638123e+05
2	1.707317	879.853659	2000.097561	16.085366	1968.073171	1352.280488	46.548780	1967.256098	1.658994e+06
3	1.597753	624.912360	1994.662921	16.474157	1967.188764	1087.959551	46.195506	1966.768539	5.674701e+05

## PRICE\_AVG

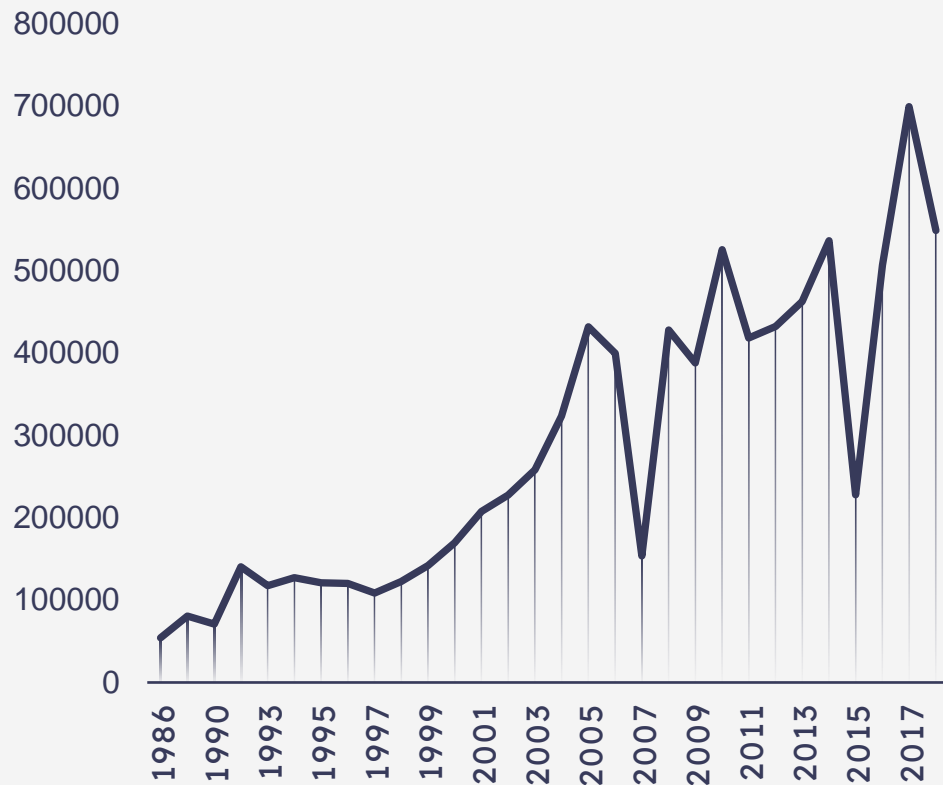


## TIMESERIES\_RES

Average % Increase  
Residential Price Per Year:

6.81%

## PRICE\_AVG



## TIMESERIES\_CON

Average % Increase  
Condominium Price Per  
Year:

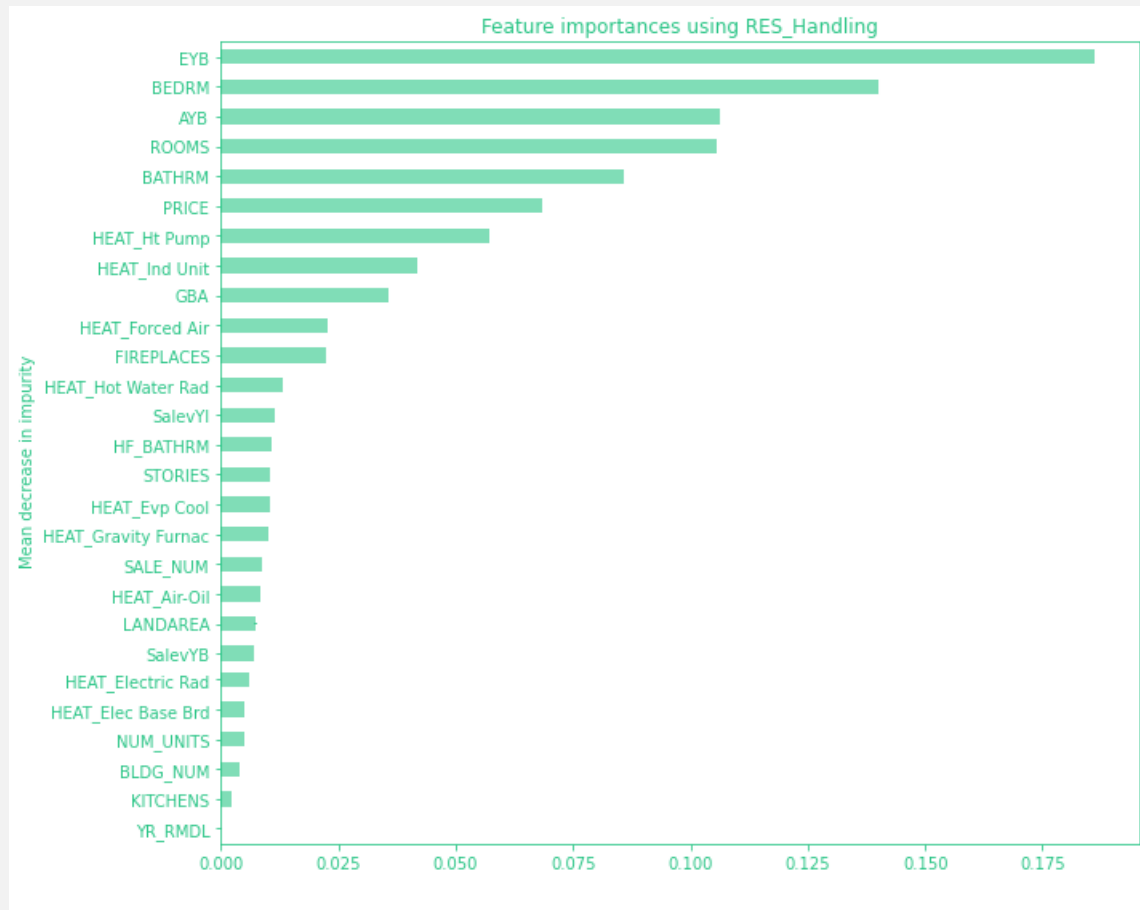
16.43%



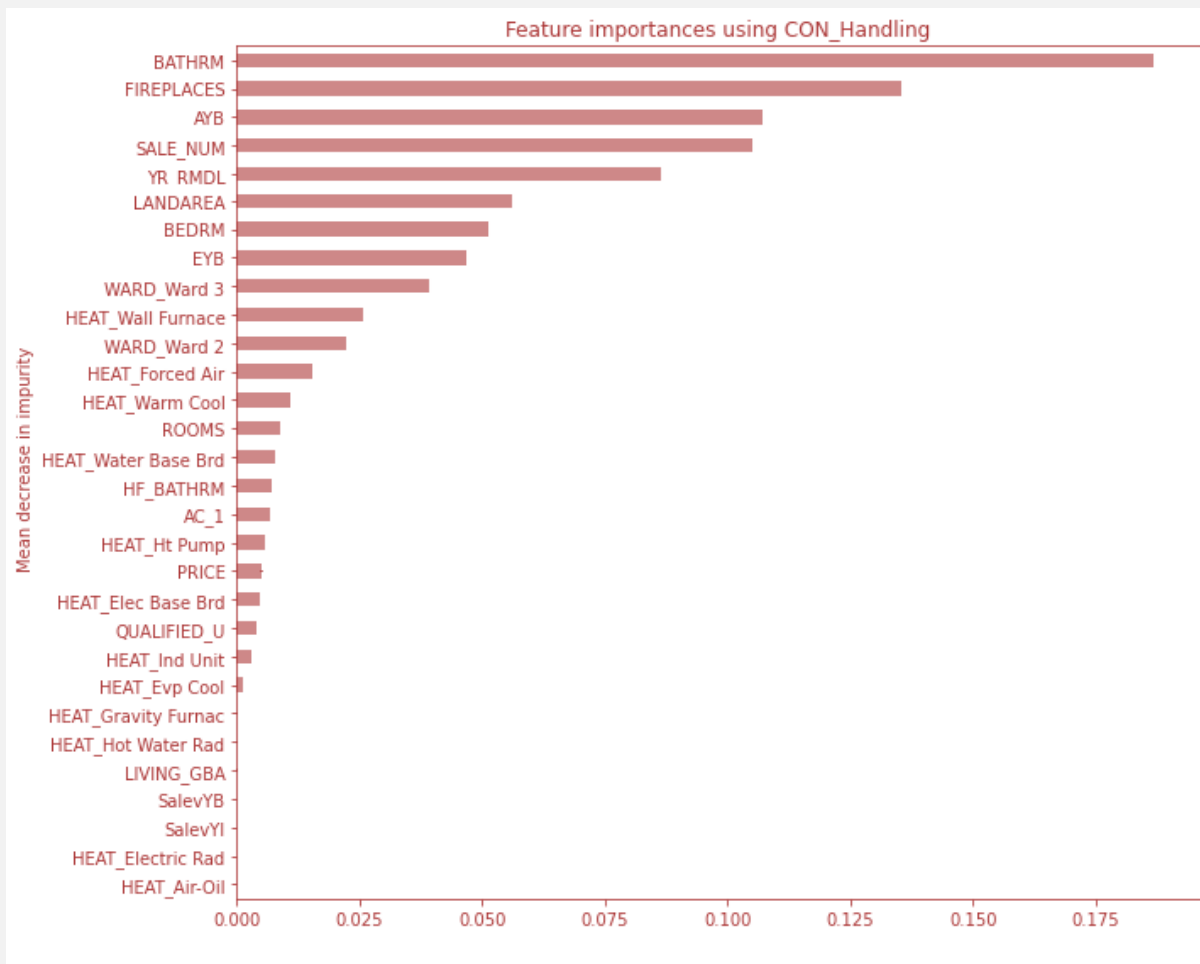
# FEATURE IMPORTANCE



# RESIDENTIAL



# CODOMINIUM





# CONCLUSION & RECOMMENDATION

## OUTPUT REGRESI:

- Kita dapat menjawab berapa harga pasar dari rumah yang ditawarkan penjual kepada kita, dengan mengetahui atribut dari rumah / condominium yang akan dijual.
- Fitur terpenting penentu harga Residential adalah EYB (Tahun Rumah dibangun) dan Untuk Condominium adalah BATHROOM
- Model terbaik yang di pakai adalah Random Forest dengan  $R^2 = \pm 87\%$

## OUTPUT CLUSTERING:

- Model clustering dapat memberikan informasi cluster resiko / condominium,
- Cluster resiko dapat meberikan suggestion kepada kita sebagai pembeli untuk basis harga tawar-menawar
- Model terbaik yang di pakai adalah Kmeans

## OUTPUT TIME SERIES:

- Model Time-Series dapat memberikan informasi tren kenaikan harga dari tren harga rumah maupun condominium
- Dari hasil perhitungan ML, Kenaikan harga dalam presentase pertahun adalah
  - Residential >>> 6,81%
  - Condominium >>> 16,43%

## KOMBINASI 3 OUTPUT

- Kombinasi 3 OUTPUT tsb dapat memberikan perhitungan margin keuntungan kami, dimana Margin keuntungan kami adalah:

Harga Jual Rumah - Harga beli

*atau*

(Hasil regresi (Harga Wajar di Pasar) x Tren Harga Rumah) - Harga beli

# RECOMMENDATION

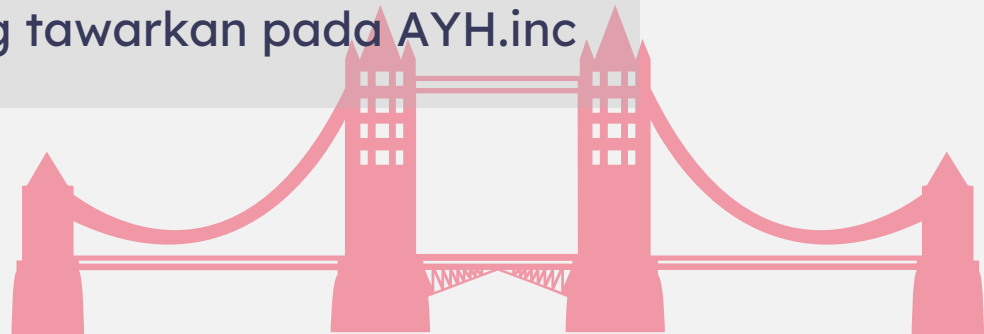
## Rekomendasi dan Future Works:

- Outlier dan data handling perlu untuk diperhatikan sebelum model dijalankan, karena data input sangat mempengaruhi output model (Garbage In Garbage Out)
- Melakukan prediksi harga rumah dengan mempertimbangkan aspek spasial dan melakukan peramalan harga masa depan dengan metode time-series terkini seperti prophet.
- Membuat Function untuk mengotomatisasi dan mendapatkan semua urutan analisis dalam 1x run.
- Membuat API Services Endpoint dari Function otomatisasi yang telah di buat dengan teknologi seperti Flask, Django, ataupun Fast API
- Mempublikasikan API Services Endpoint yang telah di buat dalam micro service environment seperti Docker

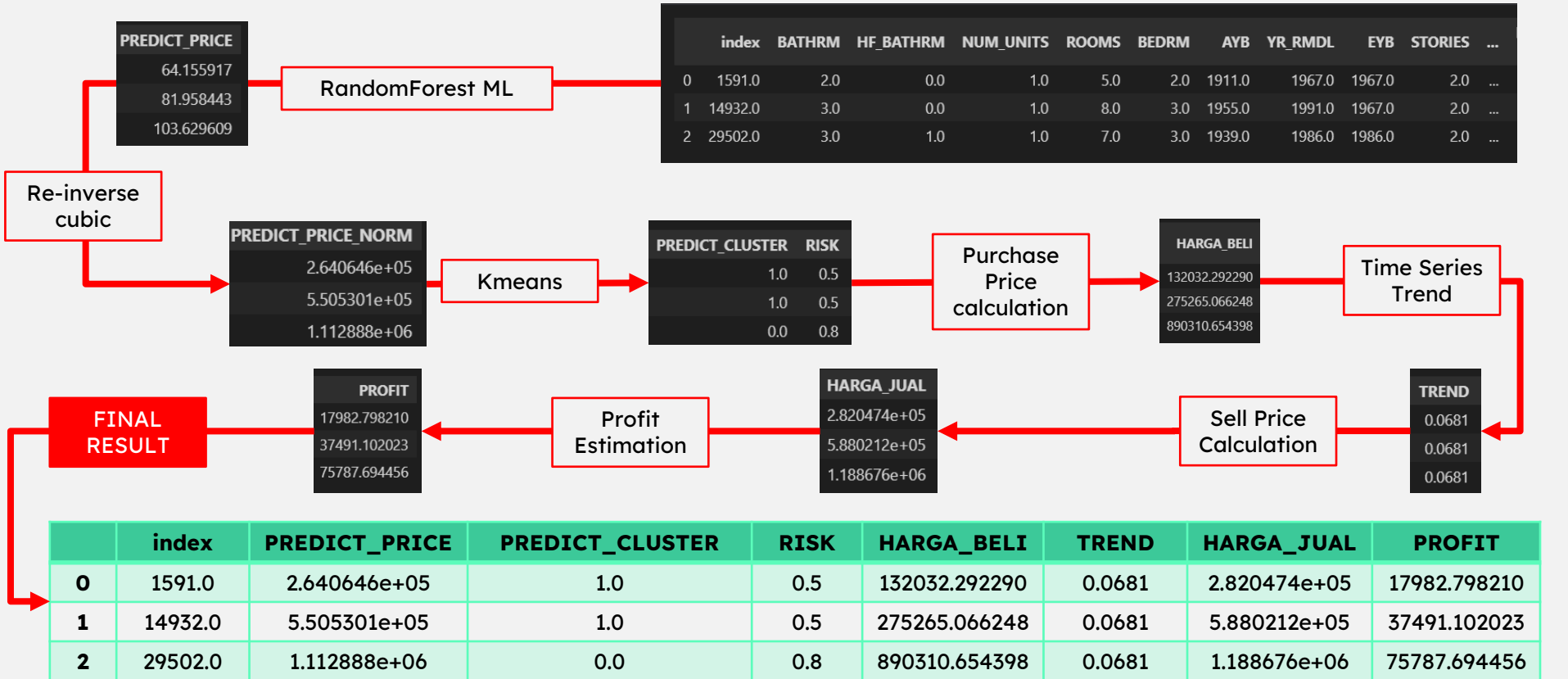


# DEMO

Memprediksi Harga Wajar, Profil Resiko, Harga Beli, Harga Jual, Tren Harga dan Perkiraan Profit dari 3 Residensial yang tawarkan pada AYH.inc



# DEMO





# TABLEAU

