

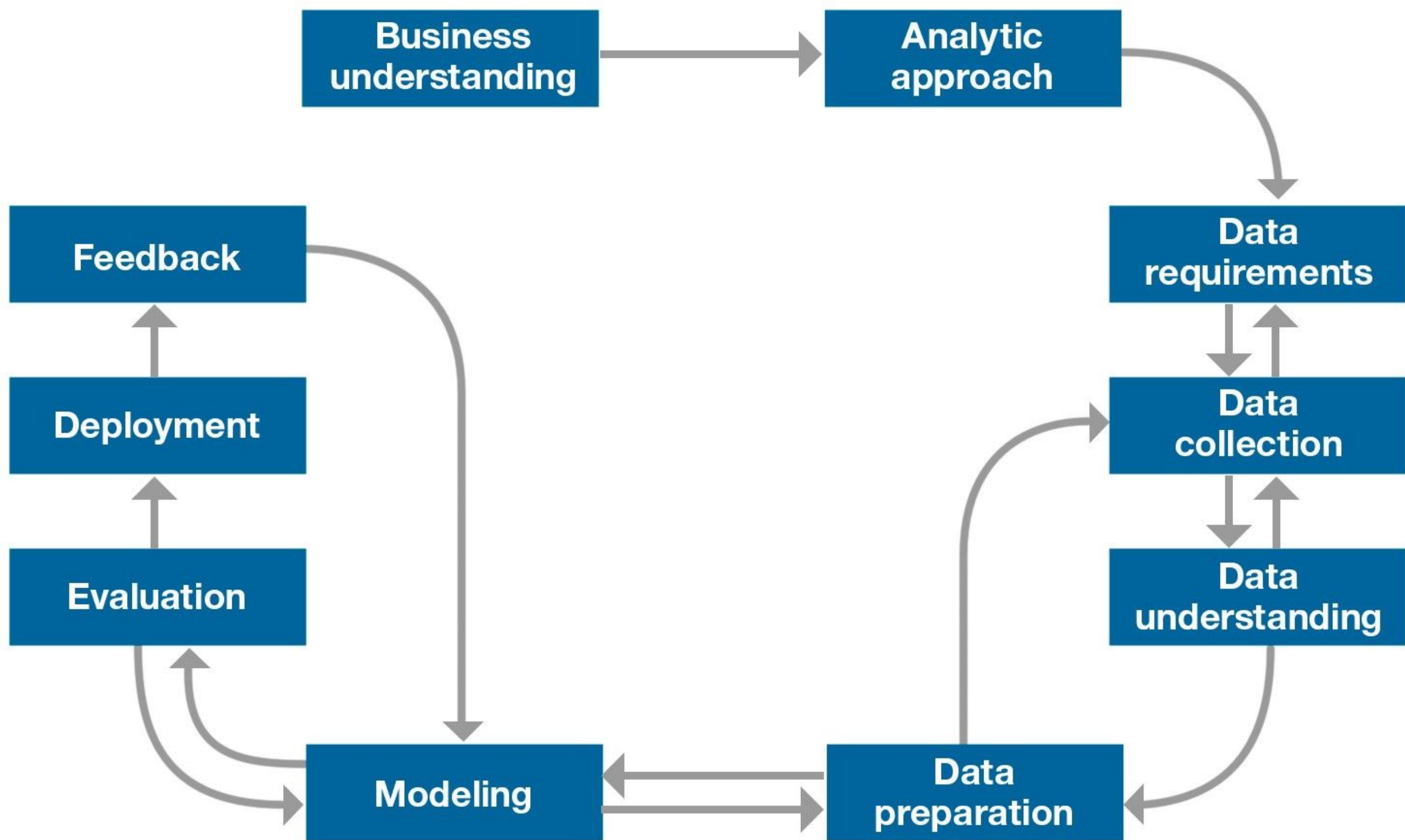
New York City Airbnb



GROUP Q 3B

Member :

1. Nida Khairunnisa Kusumawardhani
2. Satrio Wicaksono (not participate in the final project)
3. Khairul Hendra (not participate in the final project)
4. Ana Wahyuni (not active)
5. Sazali Rahman (not active)





BUSINESS UNDERSTANDING

•Problem

Airbnb is an online marketplace that connects people who want to rent out their homes with people who are looking for accommodations in that locale. It currently covers more than 81,000 cities and 191 countries worldwide. The company's name comes from "air mattress B&B."

Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present more unique, personalized way of experiencing the world. This dataset describes the listing activity and metrics in NYC, NY for 2019. This public dataset is part of Airbnb, and the original source can be found on this website.

•Clear question

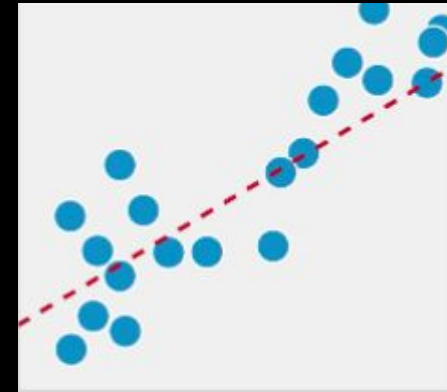
In this case, we'll be focus on ***Price Predictions*** in order to dive into possible future prices to open new Airbnb based on neighbourhood?

•Success criteria

Using Machine Learning techniques and with the power of Regression Models to ***find a positive result*** coming up with the generalized increase in prices in New York City



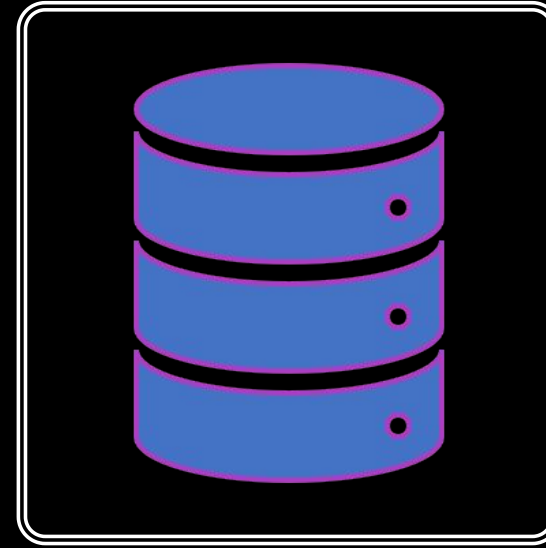
ANALYTIC APPROACH



- Regression :**
Machine Learning
1. **Linear Regression**
 2. **Decision Tree**
 3. **Logistic Regression**
 4. **Polynomial Regression**



DATA REQUIREMENT & DATA COLLECTION



New York City Airbnb (kaggle)

[https://www.kaggle.com/dgomonov/
new-york-city-airbnb-open-data/tasks](https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data/tasks)





Data Understanding

- Dataset contains New York City Airbnb

```
In [3]: df.shape  
Out[3]: (48895, 16)
```

- It has **16 Attributes** [id, name, host_id, host_name, neighbourhood_group, neighbourhood, latitude, longitude, room_type, price, minimum_nights, number_of_reviews, last_review, reviews_per_month, calculated_host_listings_count, availability_365] and **48.895 Instances**



Data Understanding

Python
Library

Scientific
Computing

Pandas

Numpy

Visualization

Matplotlib

Algorithmic

Scikit
Learn





Data Preparation

→ Cleaning Data

1. Count Missing Values

2. Drop unnecessary column

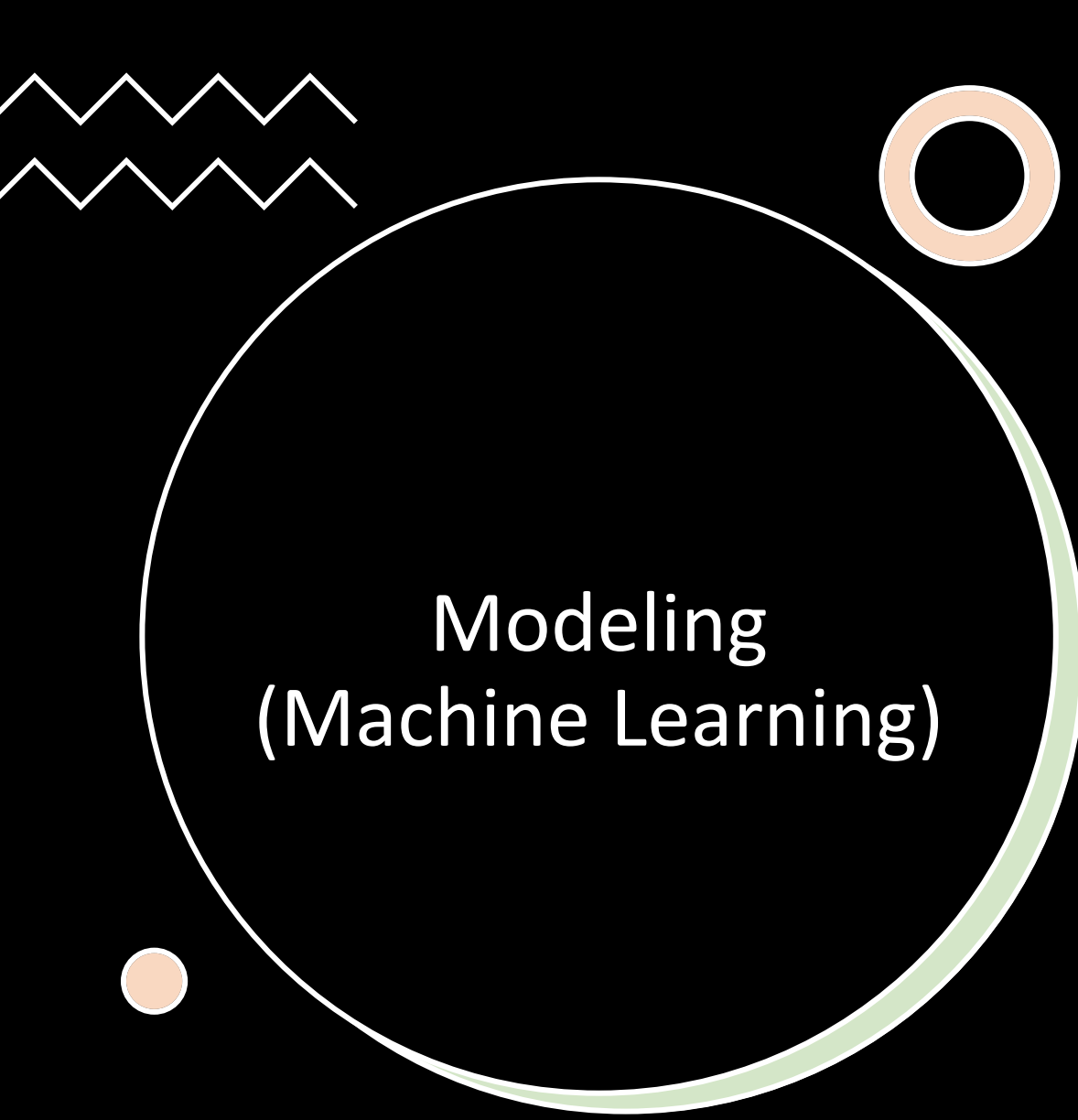
(id, name, host_id, host_name,
neighbourhood_group, latitude, longitude)

3. Fill missing value in column
'reviews_per_month' with 0

→ Transforming Data

Encoded 'neighbourhood' and 'room_type'





Modeling (Machine Learning)

```
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
```

- MSE shows how accurately the model predicts the response.
- r2_score will be calculated to find the goodness of fit measure.



Modeling

Linear Regression

```
decision_tree=DecisionTreeRegressor(min_samples_leaf=.0001)
decision_tree.fit(x_train,y_train)
predictions1=decision_tree.predict(x_test)

print('r2_score : ',r2_score(y_test,predictions1))
print('MSE : ', mean_squared_error(y_test, predictions1))
```

r2_score : 0.01918179108976259
MSE : 33056.27865683693

Decision Tree

```
reg=LinearRegression()
reg.fit(x_train,y_train)
predictions=reg.predict(x_test)

print('r2_score : ',r2_score(y_test,predictions))
print('MSE : ', mean_squared_error(y_test, predictions))
```

r2_score : 0.1072782030691396
MSE : 30087.186610418456

Logistic Regression

```
logit=LogisticRegression(solver="lbfgs", multi_class="auto")
logit.fit(x_train,y_train)
predictions2=decision_tree.predict(x_test)

print('r2_score : ',r2_score(y_test,predictions2))
print('MSE : ', mean_squared_error(y_test, predictions2))
```

r2_score : 0.01918179108976259
MSE : 33056.27865683693

Polynomial Regression

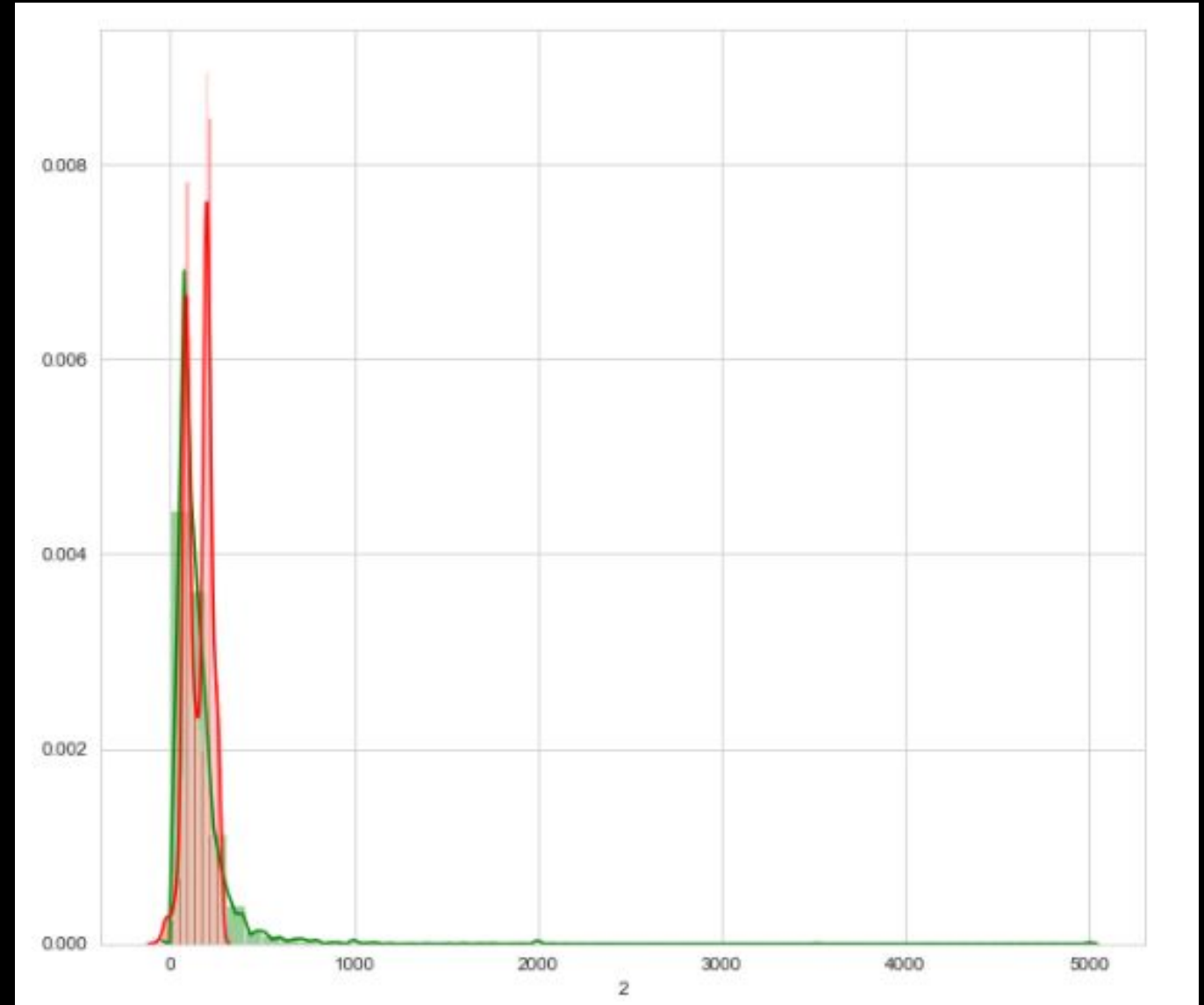
```
poly_reg = PolynomialFeatures(degree = 6)
X_poly = poly_reg.fit_transform(x)
lin_reg = LinearRegression()
lin_reg.fit(X_poly, y)y_pred = lin_reg.predict(X_poly)
print('r2_score : ',r2_score(y,y_pred))
print('MSE : ', mean_squared_error(y, y_pred))
```

r2_score : 0.1799995231385747
MSE : 47291.76097467095

Result Linear Regression

	Actual Prices	Predicted Prices
0	99.0	47.757057
1	75.0	208.159682
2	260.0	236.597683
3	200.0	201.947100
4	135.0	150.018265

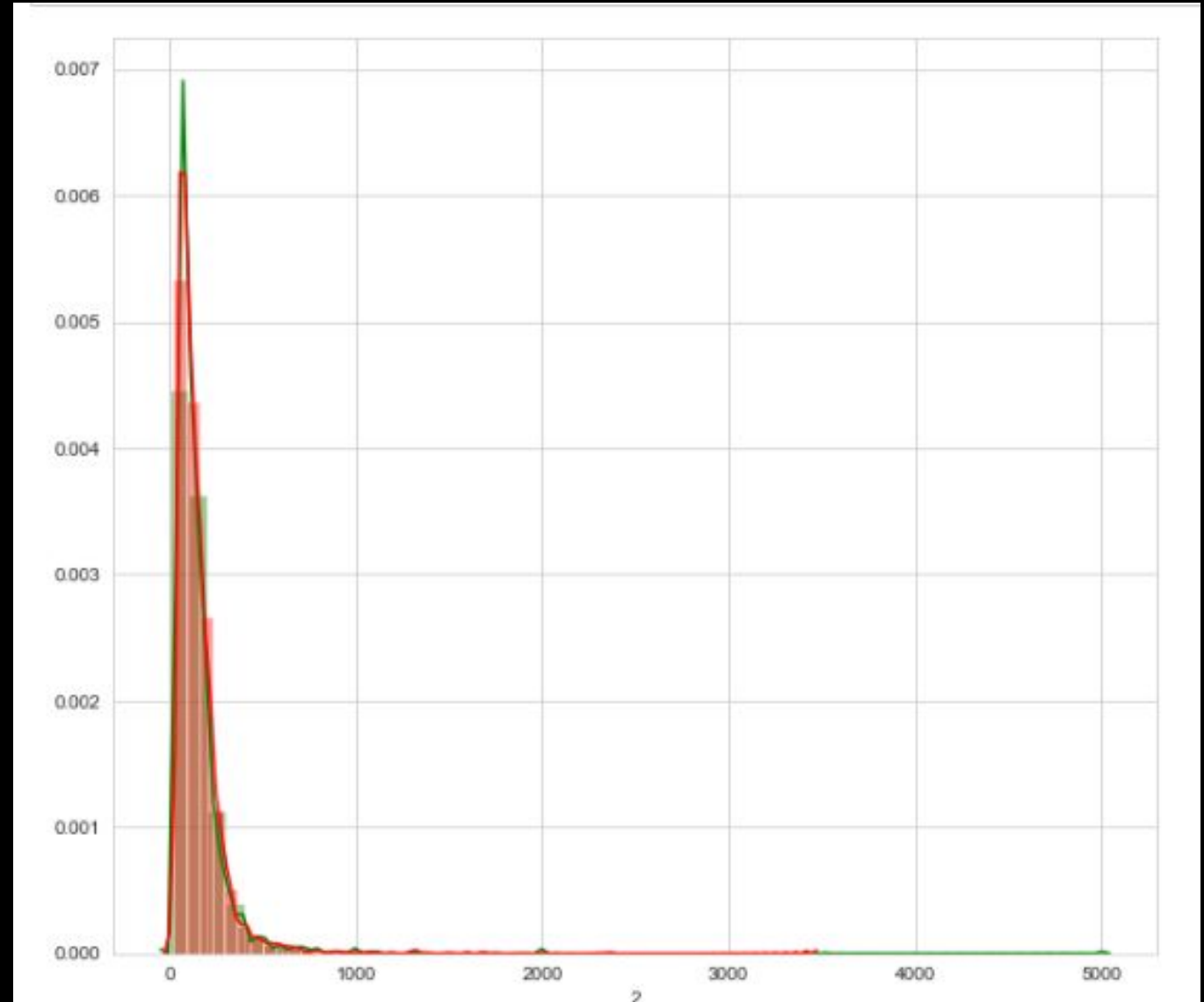
r2_score : 0.1072782030691396
MSE : 30087.186610418456



Result Decision Tree Regression

	Actual Prices	Predicted Prices
0	99.0	107.200000
1	75.0	302.000000
2	260.0	209.166667
3	200.0	347.888889
4	135.0	84.375000

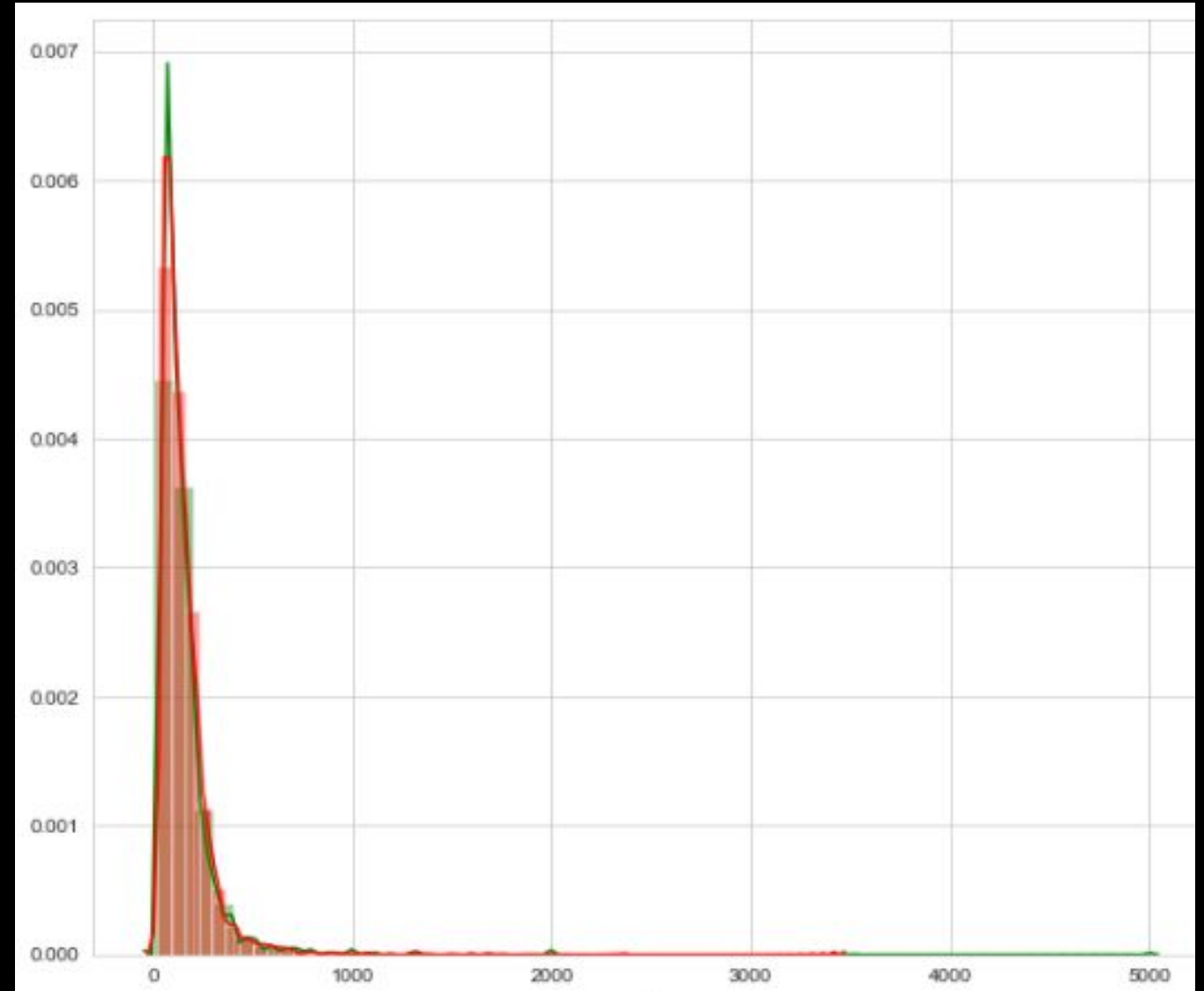
r2_score : 0.019593405460577862
MSE : 33042.40611734161



Result Logistic Regression

	Actual Prices	Predicted Prices
0	99.0	47.757057
1	75.0	208.159682
2	260.0	236.597683
3	200.0	201.947100
4	135.0	150.018265

r2_score : 0.019193815730424002
MSE : 33055.873393283655

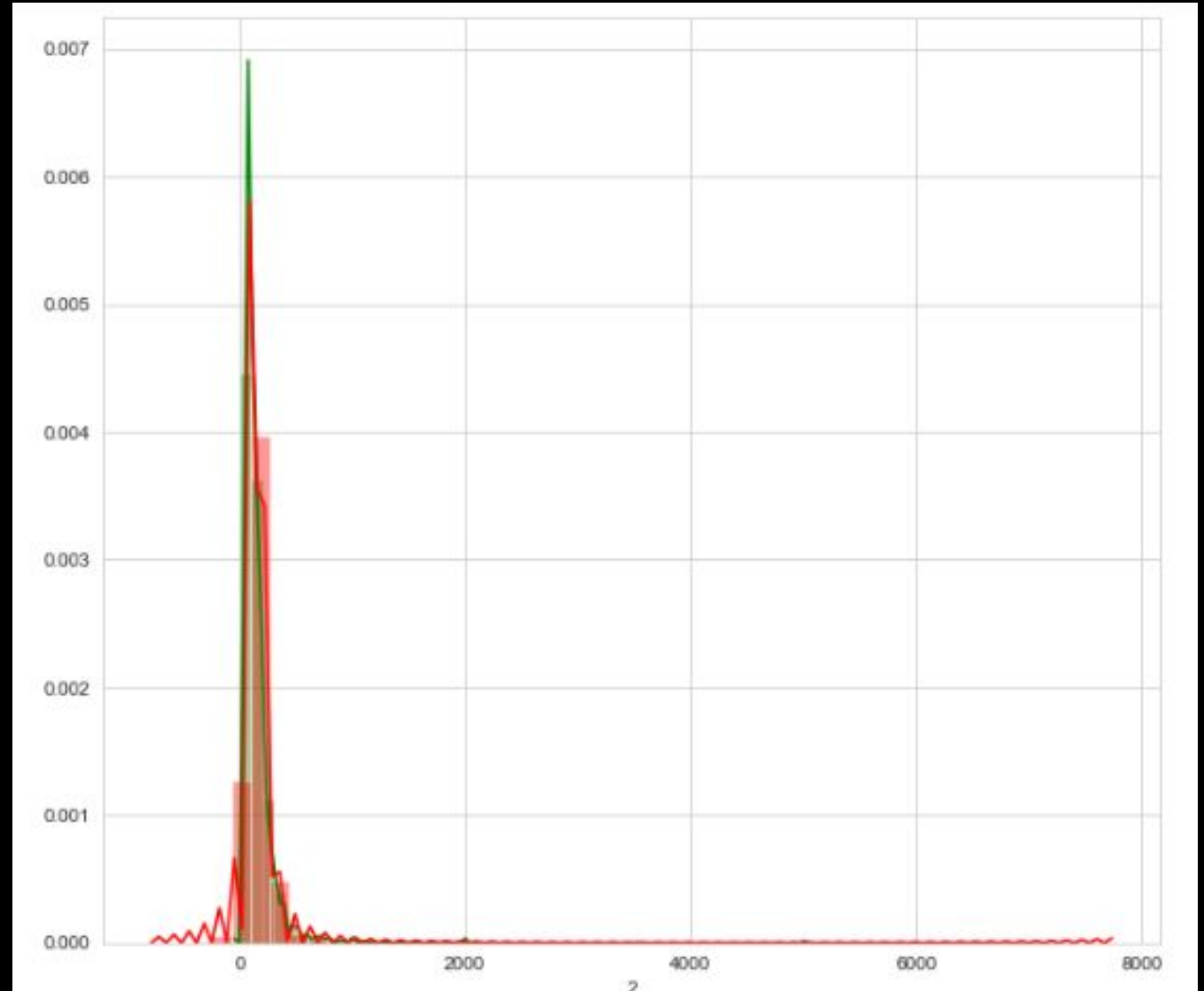


Result Polynomial Regression

	Actual Prices	Predicted Prices
0	149.0	230.319660
1	225.0	326.927900
2	150.0	156.304988
3	89.0	42.170426
4	80.0	144.982648

r2_score : 0.0803610529295068

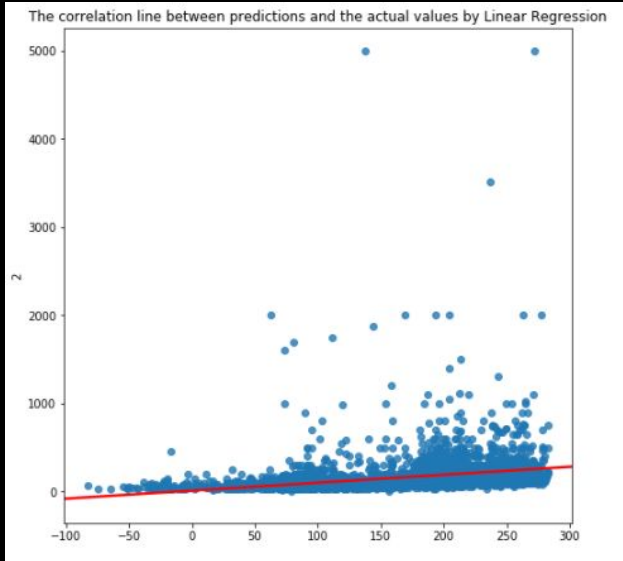
MSE : 53038.195092666494



Conclusion

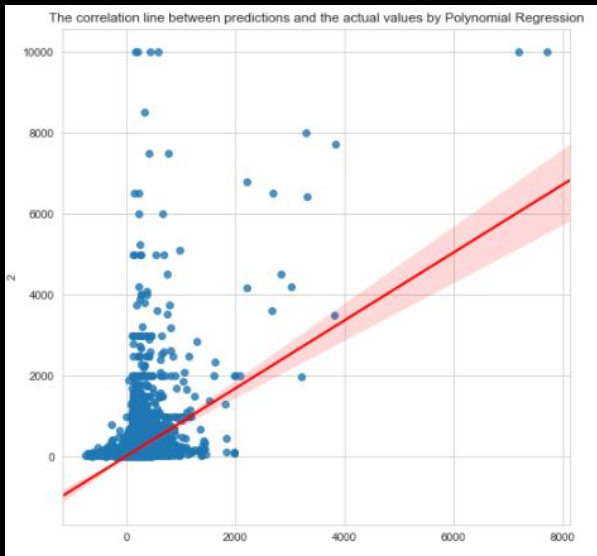
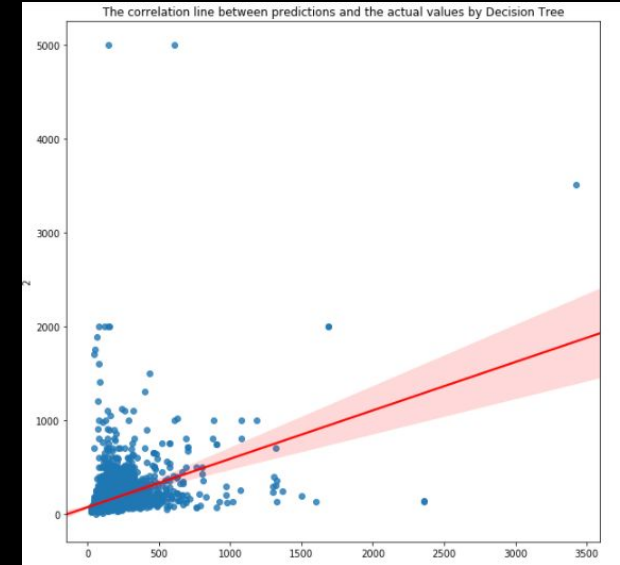
With Machine Learning, from r2 score adn MSE we can see Linear Regression has a better result than decision tree, logistic regression, and polynomial regression.

	Linear Regression	Logistic Regression	Descision Tree	Polynomial Regression
r2_score	0.1072782030691396	0.019193815730424002	0.01934813477914634	-0.03518334085086039
MSE	30087.186610418452	33055.873393283655	33050.672415742374	34888.533537832314



Linear Regression

Decision Tree Regression



Polynomial Regression

Logistic Regression

