# Airbnb Pricing Prediction

Colab Notebook Link: (please copy & paste if clicking  the link doesn't work)

https://colab.research.google.com/drive/1yUWyUfARPHWqQJAokob9Ds_uCjJicnx_#scrollTo=DlZ83skz8MTz&uniqifier=1

# Table of Contents

- Problem Statement
- Dataset Description
- Data Exploration, Visualization & Preprocessing
- Modeling and Results
- Challenges
- Conclusion & Learning

# Team 8

**Rohan Chaudhary**

https://www.linkedin.com/in/rohan-chaudhary-msba/

**Peiqi (Alma) Chen**

https://www.linkedin.com/in/peiqi-chen-alma/

**Ashley Mercado**

https://www.linkedin.com/in/ashmercado/

**Sifan Zhu**

https://www.linkedin.com/in/sifanzhu/

# Problem Statement

- The project aims to predict the price of Airbnb listings from the given number of features in the dataset
- The steps involved exploratory data analysis, data cleaning and preprocessing, fitting regression models, model comparison, model tuning and testing the outcomes.
- Airbnb company cares about the problem; the regression model predicts the price and helps to design better pricing strategies in advanced which is helful for both hosts and users

# Datset Description

- Source: Airbnb Price Prediction Dataset from Kaggle

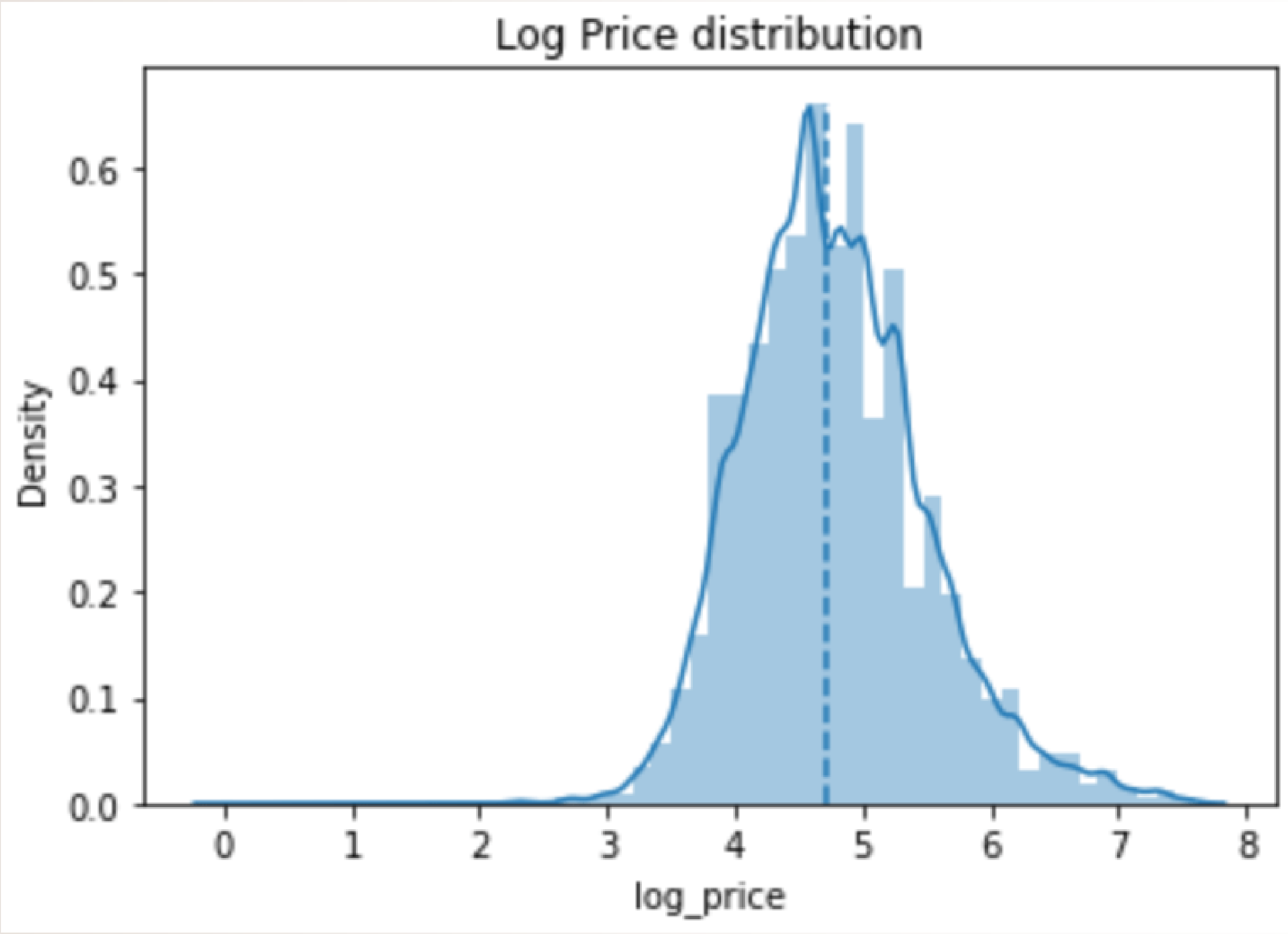
- The dataset has 74111 rows and 29 columns with

- The dataset has mixed data types: Numeric, Category, Date, String

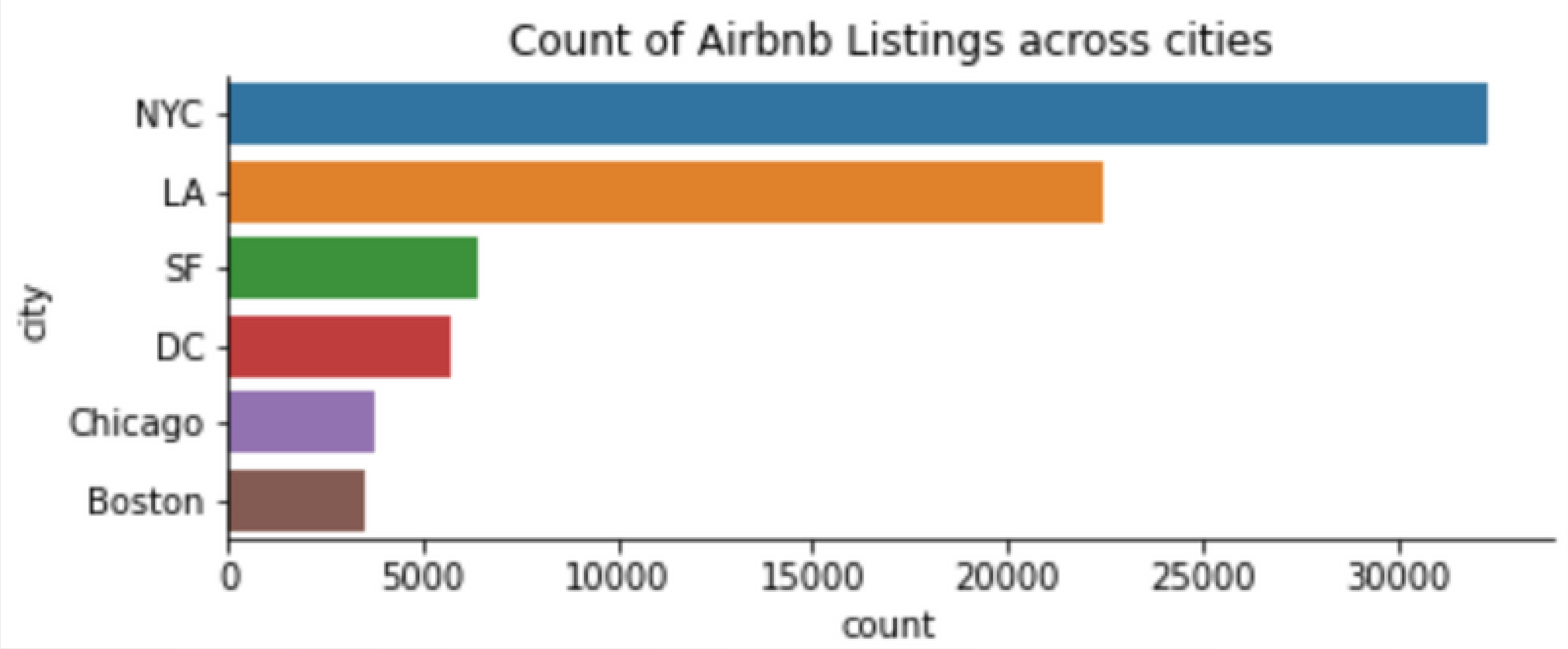
- Price is our target variable
    - Other variables: city, bathrooms, property type, latitude, longitude

# Data Exploration & Visualization

- Plotted the distribution plot for the target variable to check the price distribution
    - **The log_price variable** follows close to a normal distribution

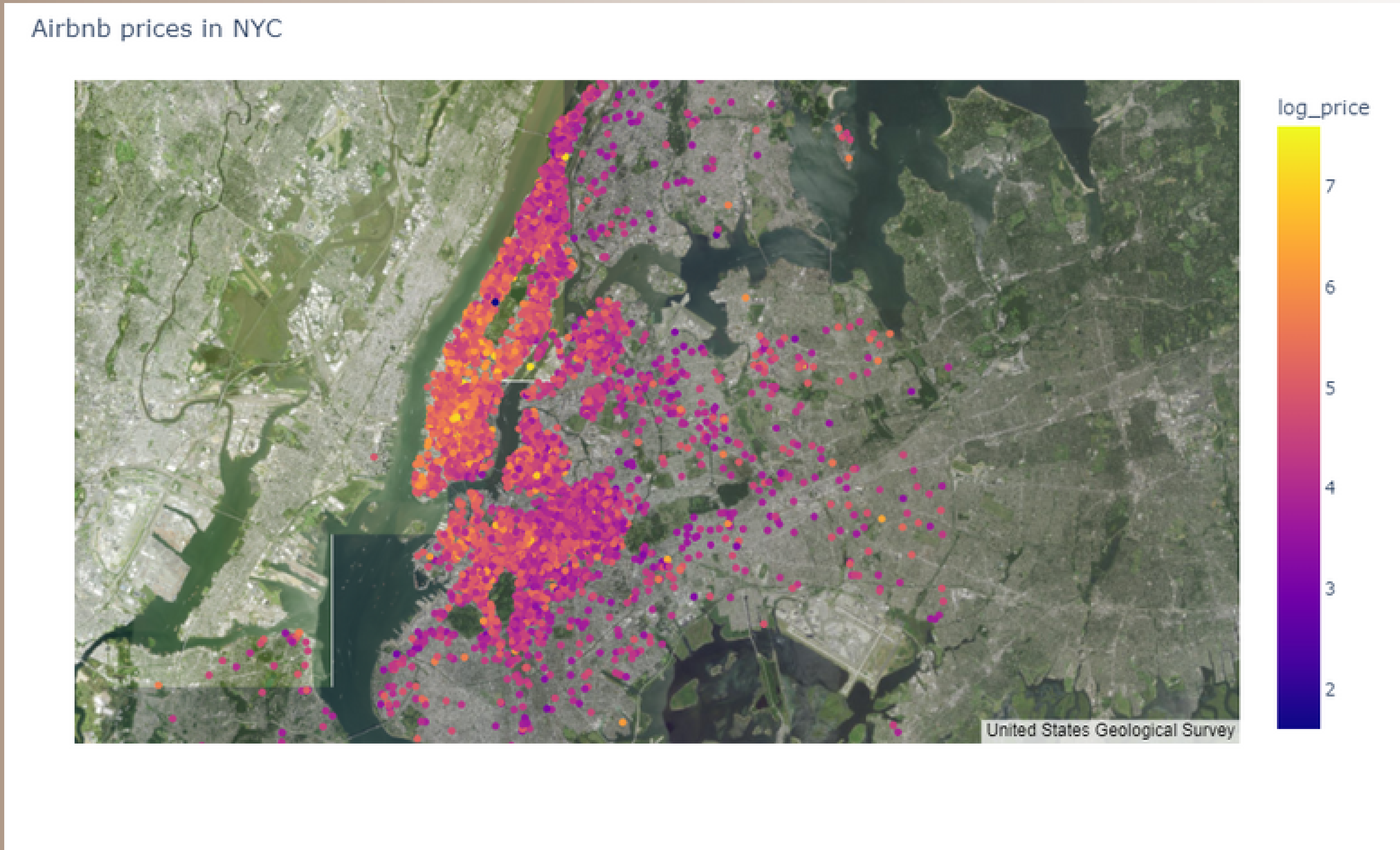


- Created a bar chart showing the count of Airbnb listings in each city
    - New York has the highest number of listings, and Boston has the least



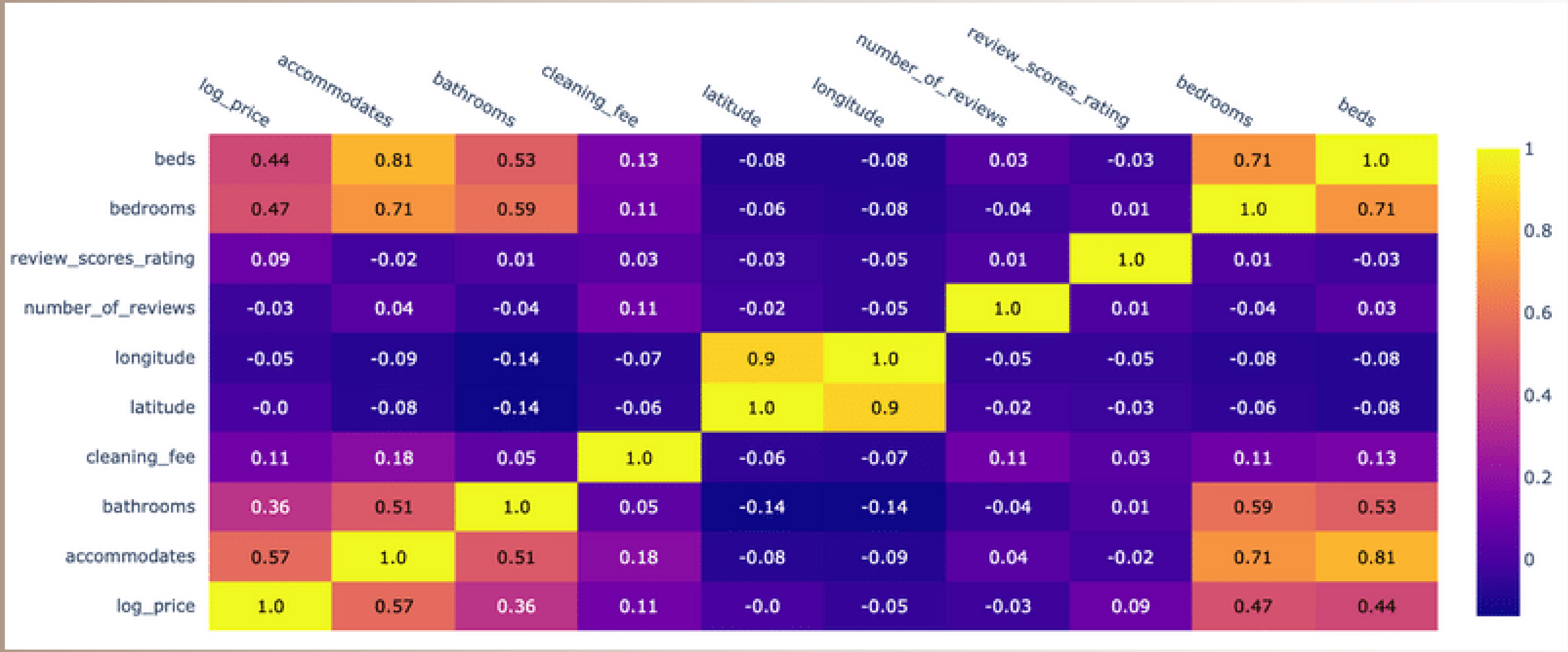
# Data Exploration & Visualization

The geographic plot of Airbnb locations as per their log_prices; the brighter colour indicates higher-priced properties; the darker colour  indicates lower-priced properties.



# Data Exploration & Visualization

- The heat map shows the relationship of numeric variables with each other, along with the target variable
- The variable that has the lowest correlation with log price is longitude with a value of -0.05.
- The variable that had the highest correlation with log price was accommodates with a value of 0.57.



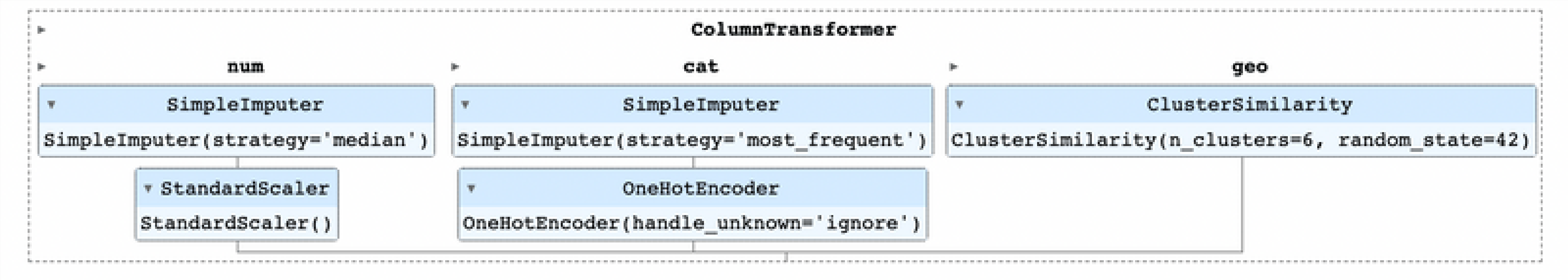
| | log_price | accommodates | bathrooms | cleaning_fee | latitude | longitude | number_of_reviews | review_scores_rating | bedrooms | beds |
|---|---|---|---|---|---|---|---|---|---|---|
| beds | 0.44 | 0.81 | 0.53 | 0.13 | -0.08 | -0.08 | 0.03 | -0.03 | 0.71 | 1.0 |
| bedrooms | 0.47 | 0.71 | 0.59 | 0.11 | -0.06 | -0.08 | -0.04 | 0.01 | 1.0 | 0.71 |
| review_scores_rating | 0.09 | -0.02 | 0.01 | 0.03 | -0.03 | -0.05 | 0.01 | 1.0 | 0.01 | -0.03 |
| number_of_reviews | -0.03 | 0.04 | -0.04 | 0.11 | -0.02 | -0.05 | 1.0 | 0.01 | -0.04 | 0.03 |
| longitude | -0.05 | -0.09 | -0.14 | -0.07 | 0.9 | 1.0 | -0.05 | -0.05 | -0.08 | -0.08 |
| latitude | -0.0 | -0.08 | -0.14 | -0.06 | 1.0 | 0.9 | -0.02 | -0.03 | -0.06 | -0.08 |
| cleaning_fee | 0.11 | 0.18 | 0.05 | 1.0 | -0.06 | -0.07 | 0.11 | 0.03 | 0.11 | 0.13 |
| bathrooms | 0.36 | 0.51 | 1.0 | 0.05 | -0.14 | -0.14 | -0.04 | 0.01 | 0.59 | 0.53 |
| accommodates | 0.57 | 1.0 | 0.51 | 0.18 | -0.08 | -0.09 | 0.04 | -0.02 | 0.71 | 0.81 |
| log_price | 1.0 | 0.57 | 0.36 | 0.11 | -0.0 | -0.05 | -0.03 | 0.09 | 0.47 | 0.44 |

# Pipeline

- Numeric Value :
  - SimpleImputer - median to filling null value;
  - StandardScaler -  standardize the data

- Categorical :
  - SimpleImputer - filling null value with mode Categorical
  - OneHotEncoder -  transform categorical features into numerical dummy features

- Geography :
  - grouping similar location



ColumnTransformer

num

SimpleImputer

SimpleImputer(strategy='median')

StandardScaler

StandardScaler()

cat

SimpleImputer

SimpleImputer(strategy='most_frequent')

OneHotEncoder

OneHotEncoder(handle_unknown='ignore')

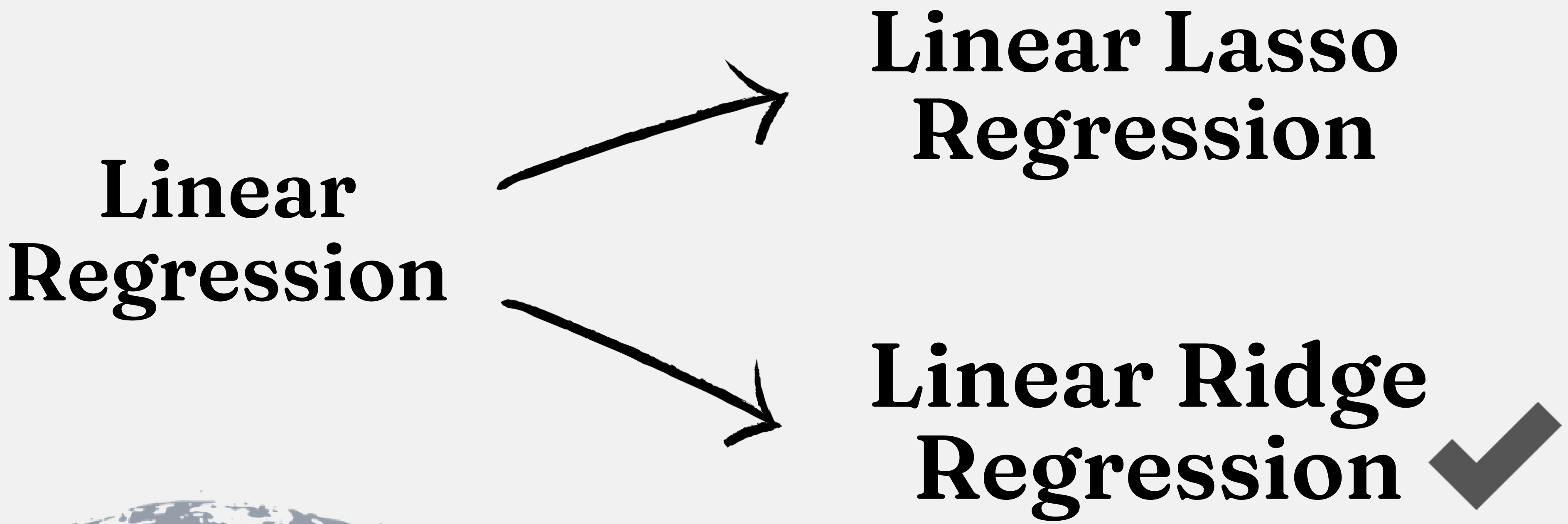
geo

ClusterSimilarity

ClusterSimilarity(n_clusters=6, random_state=42)

# Modelling &
# Result Comparison

# Modeling Techniques



Linear Regression

Linear Lasso Regression

Linear Ridge Regression ✓

Bayesian Linear Regression

Bayesian Ridge Regression

Decision Tree Regression

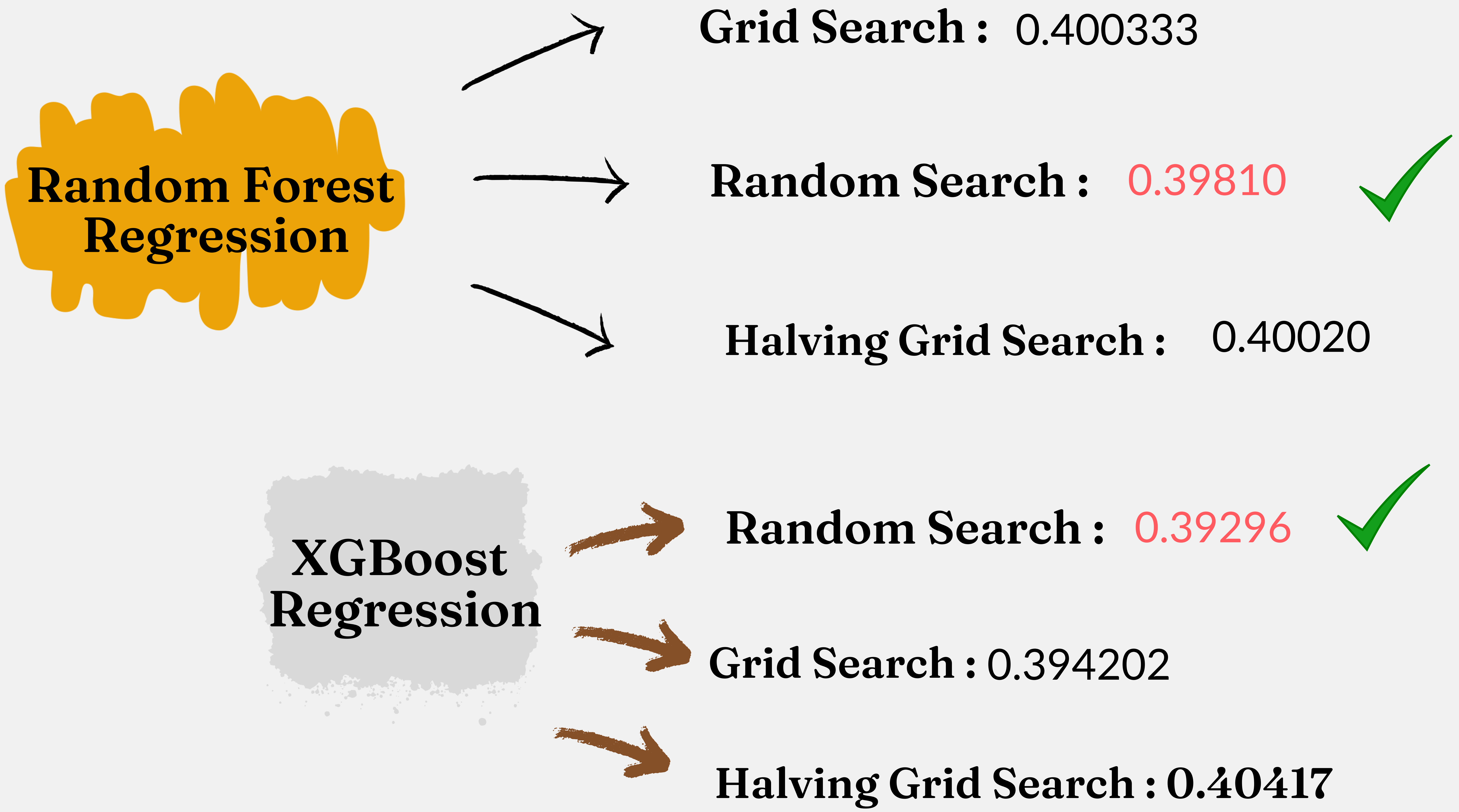
Random Forest Regression

XGBoost Regression

# Model Results

| | RMSE |
|---|---|
| Linear Regression | 25,918,537.99104 |
| Linear Lasso Regression | 0.55557 |
| Linear Ridge Regression | 0.46851 |
| Decision Tree Regression | 0.60239 |
| **Random Forest Regression** | 0.43514 |
| Bayeisan Linear Regression | 0.46684 |
| Bayeisian Ridge Regression | 0.46699 |
| **XGboost Regresssion** | 0.43247 |

# Tuning



Random Forest Regression

Grid Search : 0.400333

Random Search : 0.39810 ✓

Halving Grid Search : 0.40020

XGBoost Regression

Random Search : 0.39296 ✓

Grid Search : 0.394202

Halving Grid Search : 0.40417

# Result

- Final Best Model: XGBoost Regression Model with Random Search

- Final Test Error : 0.38731

# Challenges

- Used pipeline to convert data type and then used converted columns to make two new columns by doing a difference. We finally did the column transformation outside of the pipeline.

- We got a high test error from linear regression, but the problem was solved by lasso and ridge regression.

- Close test error in the different regression models, and we decided to tune on two models

# Conclusion & Learning

- What is the difference between regression models. Used lasso and ridge regression instead to fix the high test error from normal linear regression
- Building general pipelines and customizing a complex pipeline like using k-mean to cluster longitude and latitude in the pipeline.
- Different methods of tuning, and how to tune the parameters. We need the know the convention range of our tuning model like the learning rate

# Thank You