Statistical Learning

Rome Airbnb Open Data

Lorenzo Bonanni VR495629
Chiara Zandomeneghi VR513170

Summary

01

Introduction

Dataset presentation

02

Data Cleaning

Remove useless columns, drop the outliers

03

EDA

Visualized the data, plot correlation and selected features

04

Models

Transformed categorical variables and fitted models

05

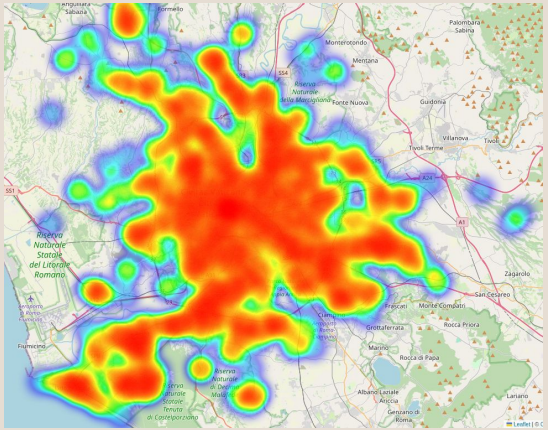
Results

Present metrics and show the results

01

Introduction

01 - Introduction



The initial dataset has 30318 samples and 75 features.

The goal of this analysis is predict the price.

The features could summarized in the follows macro categories:
Listing Information: Essential details about the property and its availability.
Host Information: Details about the person or entity offering the listing.
Review Information: Data related to guest reviews and ratings.
Legal and Booking Information: Information relevant to booking and legal aspects of the listing.

Source: https://insideairbnb.com/get-the-data/

02

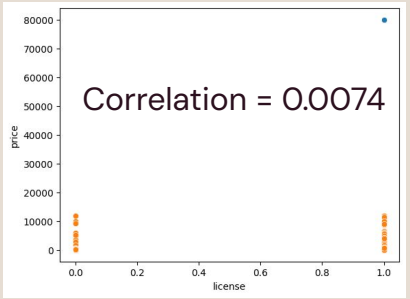
Data Cleaning

02 - Data Cleaning

The initial dataset has 30318 samples and 75 features.
After this procedure we ended with 22817 samples and 23 features.

We drop the features that we are not interested for our analysis, such as
Identification: This category includes attributes related to the identification and metadata of the listing.
Some Listing Information: This category covers details about the listing itself, including its description, availability, and other relevant features.
Host Information: This category contains attributes related to the host of the listing.

- Assumption: License influence Price
  Result: No Influence
- Dropped Nans and Outliers



03

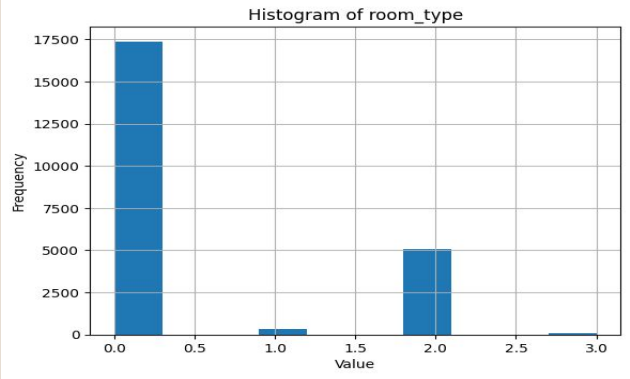
EDA

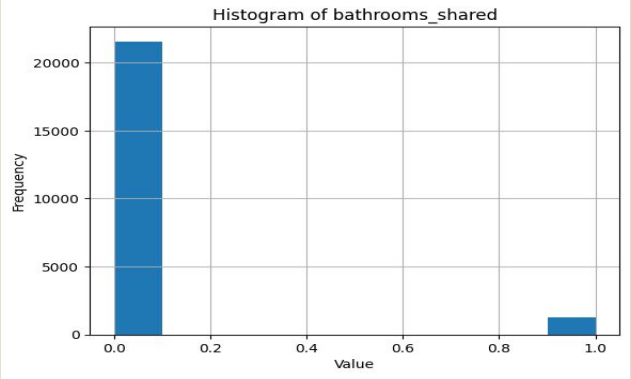
03 - EDA

The initial dataset has 22817 samples and 23 features.
After this procedure we ended with 22817 samples and 10 features.

- We plotted data distribution and *features vs price*.
- Computed *logprice* and plotted data distribution and *features vs logprice*.
- Correlation matrix and dropped unrelated features (~0 corr).

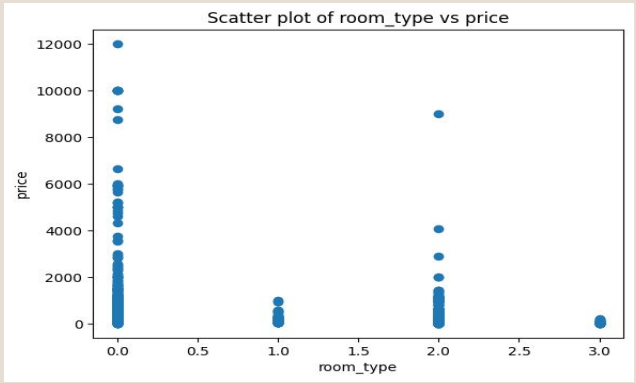
03 - EDA

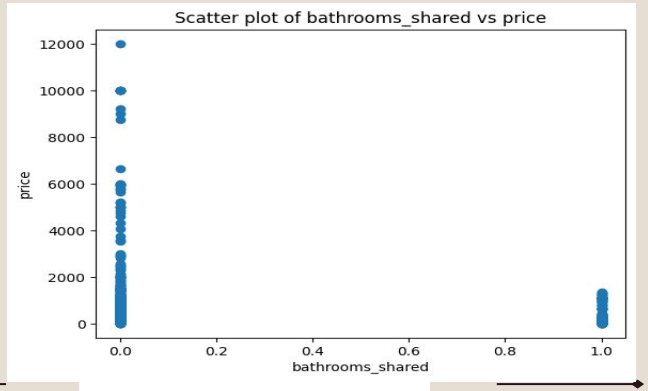




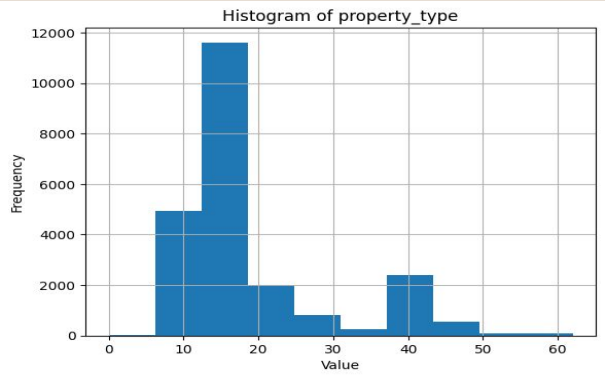
Distributions

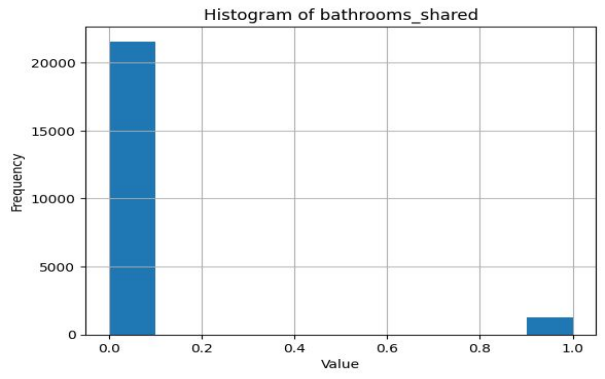
Features vs Price





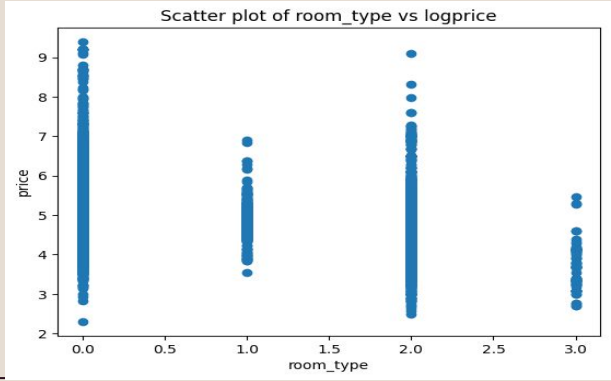
03 - EDA

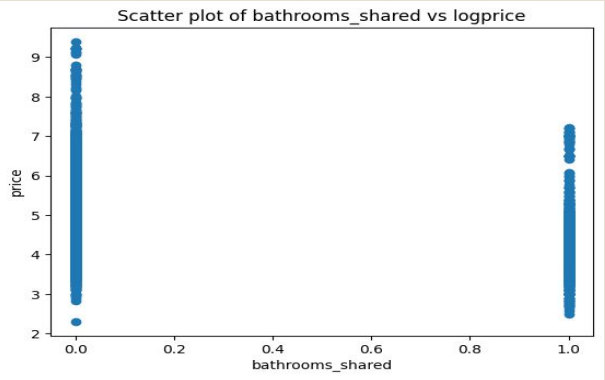




Distributions

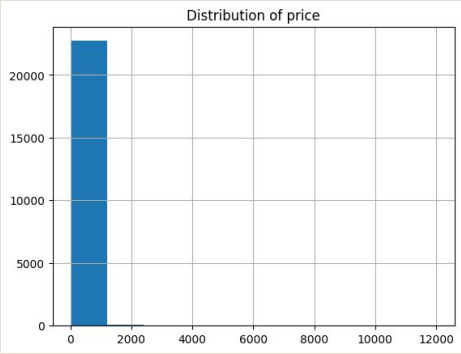
Features vs logprice

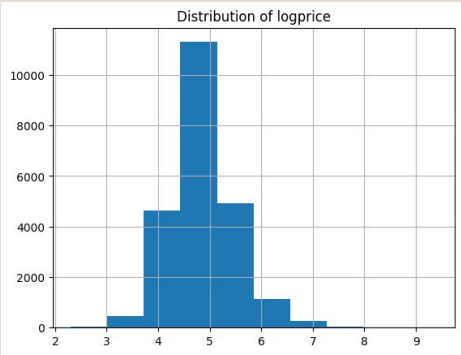


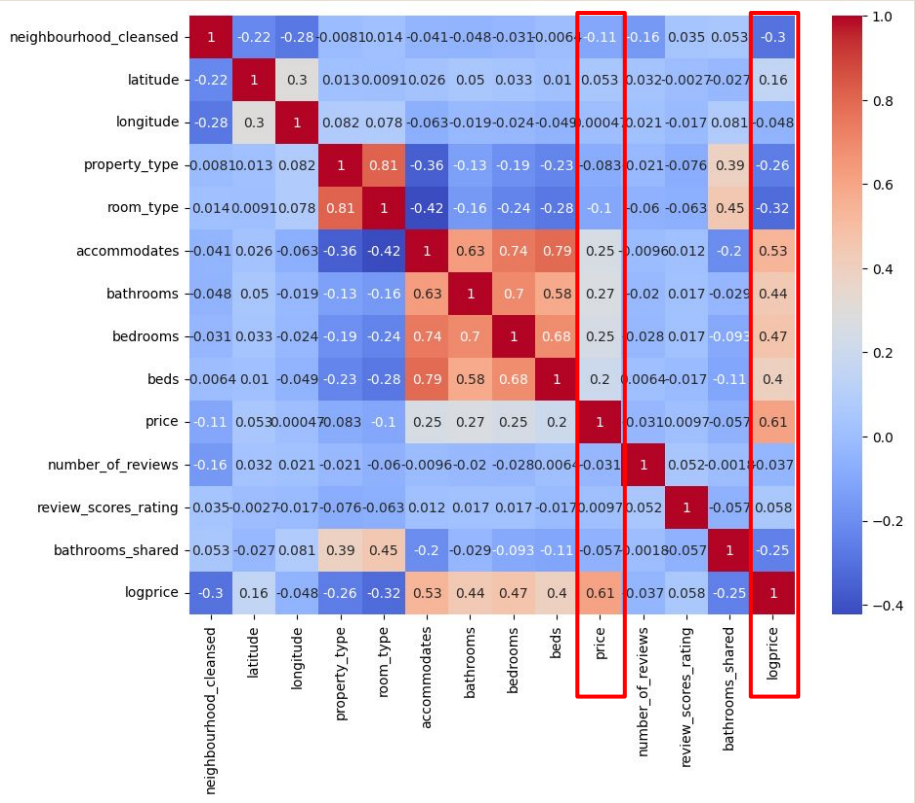


03 - EDA

Distributions







Correlation Matrix heatmap

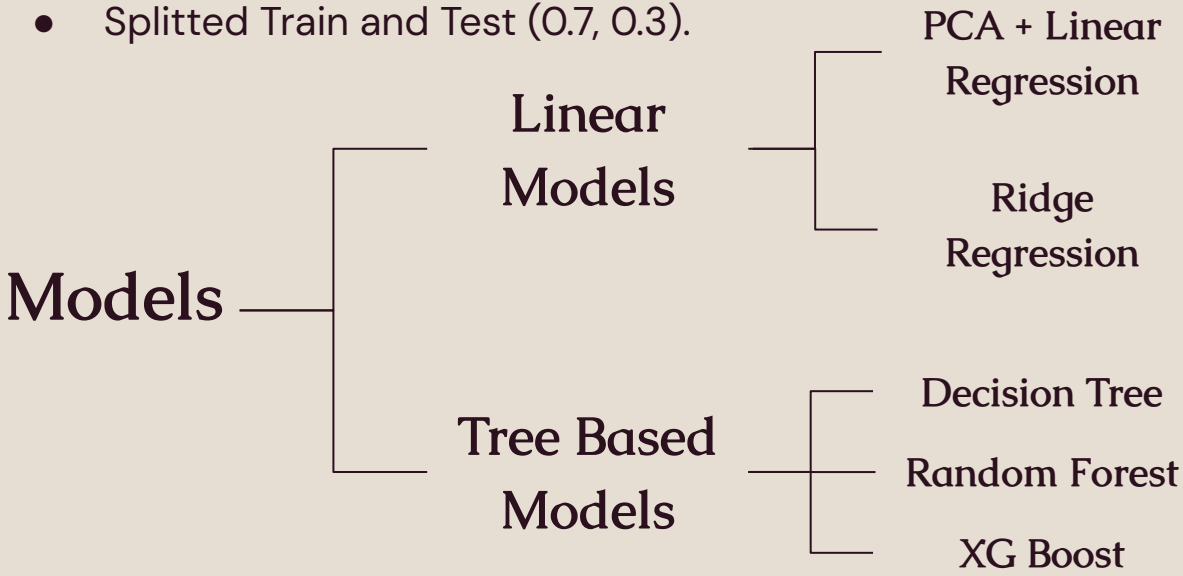
04

Models

04 - Models

The initial dataset has 22817 samples and 10 features.
After this procedure we ended with 22817 samples and 31 features.

- Convert the categorical variables.
- Standardized features.
- Splitted Train and Test (0.7, 0.3).



04 - Models

- One-Hot encoding: *room_type*

```
df["room_type"].unique()
✓ 0.0s
array(['Private room', 'Entire home/apt', 'Hotel room', 'Shared room'],
      dtype=object)
```

room_type_Entire home/apt	room_type_Hotel room	room_type_Private room	room_type_Shared room
False	False	True	False
False	False	True	False

- One-Hot encoding: *neighbourhood_cleansed*

```
df["neighbourhood_cleansed"].unique()
✓ 0.0s
array(['VIII Appia Antica', 'V Prenestino/Centocelle', 'I Centro Storico',
       'II Parioli/Nomentano', 'XII Monte Verde', 'XIV Monte Mario',
       'VII San Giovanni/Cinecittà', 'IV Tiburtina',
       'VI Roma delle Torri', 'XV Cassia/Flaminia', 'XIII Aurelia',
       'III Monte Sacro', 'IX Eur', 'X Ostia/Acilia',
       'XI Arvalia/Portuense'], dtype=object)
```

neighbourhood_I Centro Storico	neighbourhood_II Parioli/Nomentano	neighbourhood_III Monte Sacro	neighbourhood_IV Tiburtina	neighbourhood_IX Eur	neighbourhood_V Prenestino/Centocelle	neighbourhood_VI Roma delle Torri	neighbourhood_VII San Giovanni/Cinecittà	neighbourhood_VIII Appia Antica	neighbourhood_X Ostia/Acilia	neighbourhood_XI Arvalia/Portuense	neighbourhood_XII Monte Verde	neighbourhood_XIII Aurelia	neighbourhood_XIV Monte Mario	neighbourhood_XV Cassia/Flaminia
False	False	False	False	False	False	False	False	True	False	False	False	False	False	False
False	False	False	False	False	True	False	False	False	False	False	False	False	False	False

04 - Models

- Binary encoding: property_type

```
df['property_type'].unique()
```

0.0s

```
array(['Private room', 'Private room in bed and breakfast',
       'Entire rental unit', 'Entire vacation home',
       'Private room in rental unit', 'Private room in guesthouse',
       'Entire loft', 'Private room in home', 'Private room in condo',
       'Entire condo', 'Room in bed and breakfast',
       'Private room in loft', 'Farm stay', 'Entire home',
       'Room in serviced apartment', 'Entire guest suite',
       'Room in boutique hotel', 'Private room in casa particular',
       'Entire cottage', 'Tiny home', 'Entire villa',
       'Shared room in rental unit', 'Private room in villa',
       'Private room in guest suite', 'Entire cabin',
       'Private room in vacation home', 'Shared room in hostel',
       'Private room in farm stay', 'Entire place',
       'Entire serviced apartment', 'Entire bungalow', 'Room in hotel',
       'Entire bed and breakfast', 'Private room in townhouse',
       'Private room in serviced apartment', 'Entire guesthouse',
       'Entire townhouse', 'Room in aparthotel', 'Private room in tower',
       'Dome', 'Private room in tiny home', 'Shared room in loft',
       'Shared room in condo', 'Private room in boat',
       'Private room in pension', 'Private room in cottage',
       'Entire chalet', 'Private room in hostel',
       'Private room in castle', 'Shared room in guesthouse',
       'Private room in nature lodge', 'Room in hostel',
       'Shared room in home', 'Shared room in bed and breakfast',
       'Casa particular', 'Shared room in townhouse', 'Entire home/apt',
       'Camper/RV', 'Holiday park', 'Windmill', 'Tower',
       'Shared room in hotel', 'Cave'], dtype=object)
```

	property_type_0	property_type_1	property_type_2	property_type_3	property_type_4	property_type_5
0	0	0	0	0	0	1
1	0	0	0	0	1	0
2	0	0	0	0	1	1

04 - Models: Linear Models

Linear Regression

Linear regression is a statistical method for modeling the relationship between a dependent variable and one or more independent variables using a linear equation.

PCA

PCA is a dimensionality reduction technique that transforms high-dimensional data into a smaller set of uncorrelated variables called principal components.

Ridge Regression

Ridge regression is a type of linear regression that includes a regularization term (L2) to prevent overfitting by penalizing large coefficients.

04 - Models: Tree Based

Decision Tree

A decision tree is a flowchart-like structure where each internal node represents a decision based on input features, leading to leaf nodes representing the outcome.

Random Forest

Random forest is an ensemble learning method that aggregates the predictions of multiple decision trees to enhance accuracy and generalization.

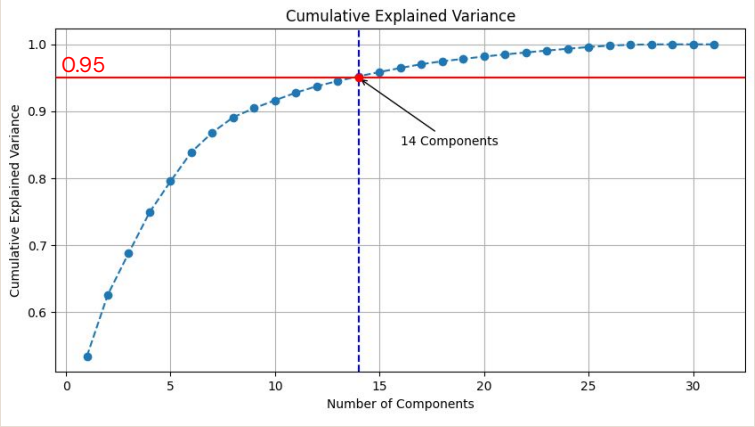
XG Boost

XGBoost, or Extreme Gradient Boosting, is a powerful machine learning algorithm that boosts decision trees sequentially to optimize predictive performance.

04 - Models: Hyper Parameter Tuning

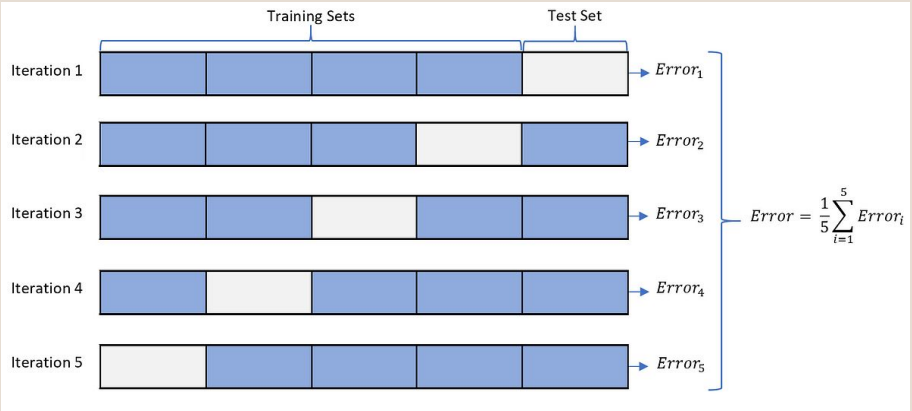
PCA

Explained Variance Plot



Others

Grid Search + 5 Fold Cross Validation



05

Results

05 - Results: Metrics

MSE

A metric that measures the average of the squares of the errors between predicted and actual values.

R^2

A statistical measure that indicates the proportion of the variance in the dependent variable explained by the independent variables.

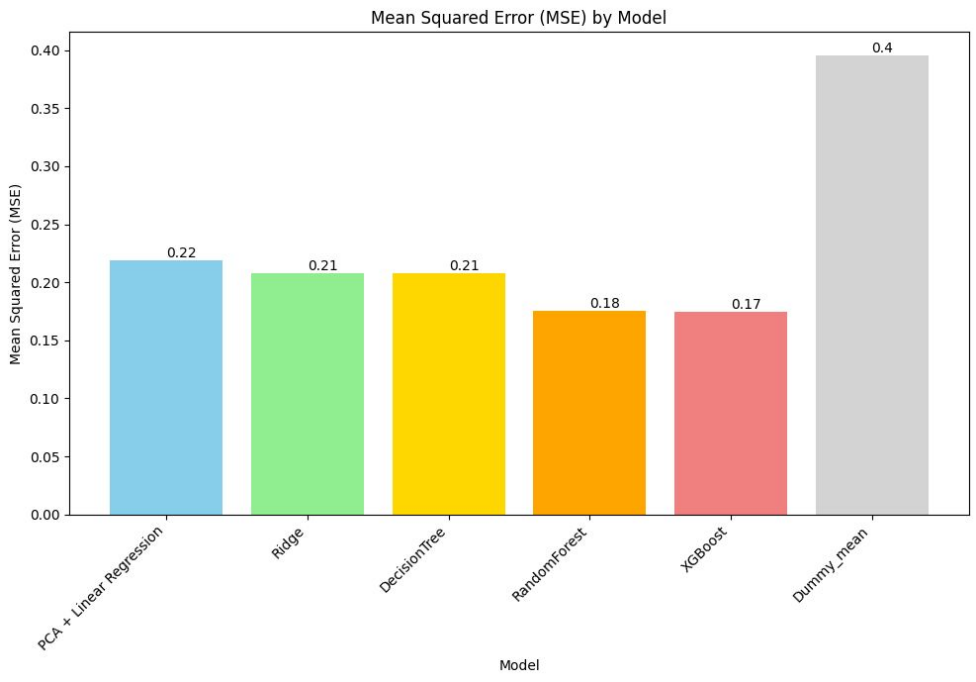
05 - Results: Metrics of LogPrice

Present MSE and R^2 metrics specifically for models predicting log price, highlighting their performance and reliability.

	MSE	R^2
PCA + Linear Regression	0.21848696685185198	0.44805200480424157
Ridge	0.2080314630492423	0.4744649961408144
DecisionTree	0.20760043375539028	0.4755538746127902
RandomForest	0.17566849730092127	0.5562212414709389
XGBoost	0.17456825549319976	0.5590007036455144
Dummy_mean	0.39585233573262424	-1.3439013642591036e-05

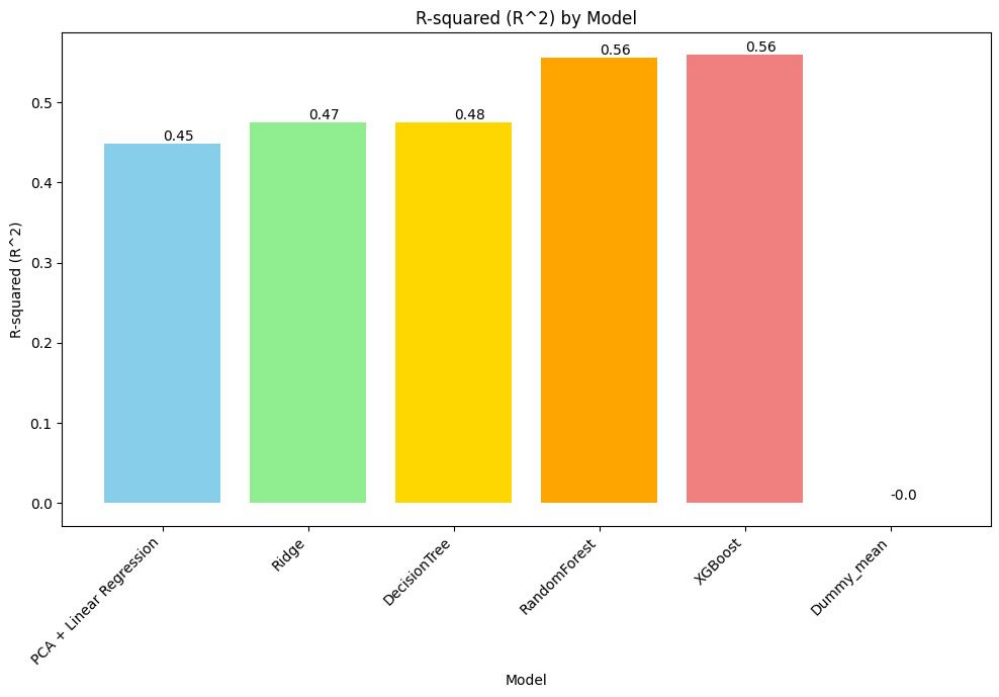
05 - Results: MSE by Model

Display a comparative analysis of MSE values across different models, illustrating their predictive efficacy for log prices.



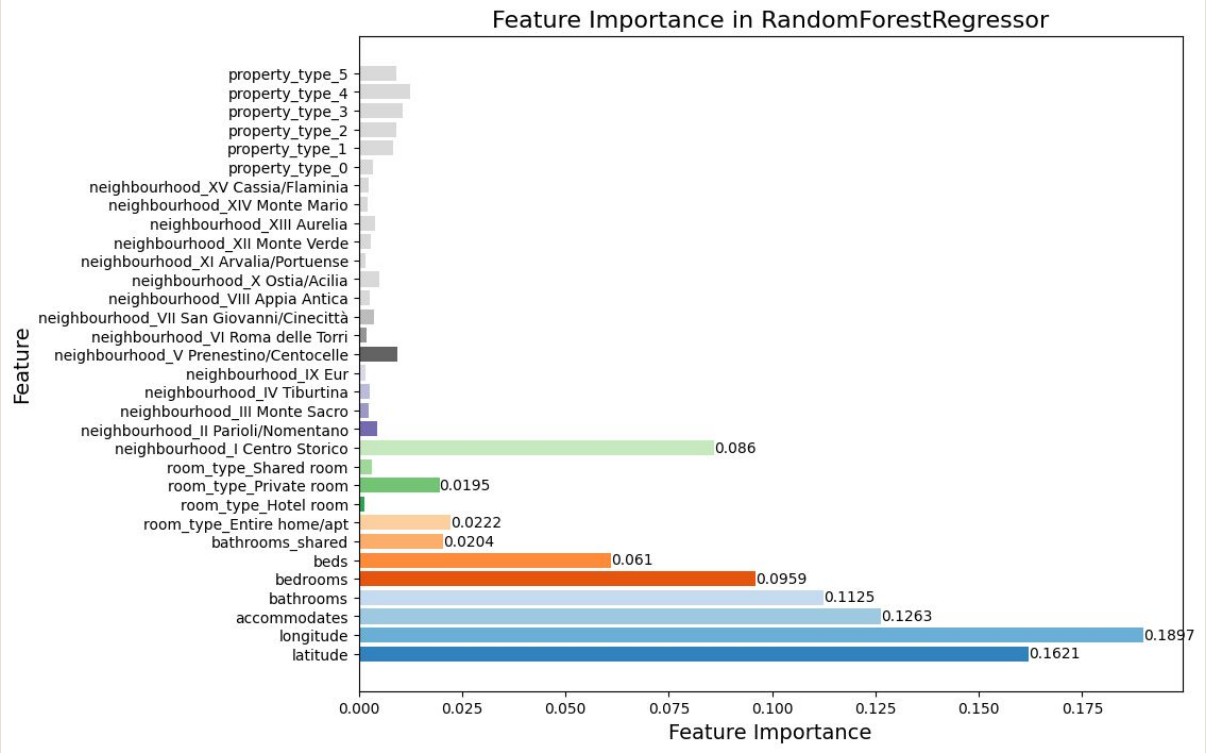
05 - Results: R^2 by Model

Present R² scores for each model, showing the variance explained and identifying the best-performing model in predicting log price.



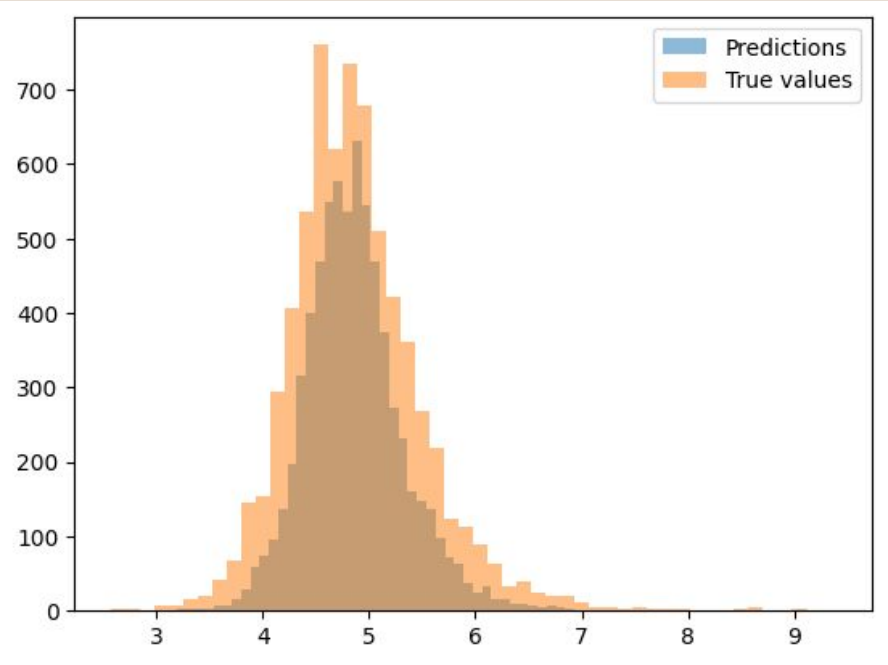
05 - Results:

Feature Importance in Random Forest Regressor



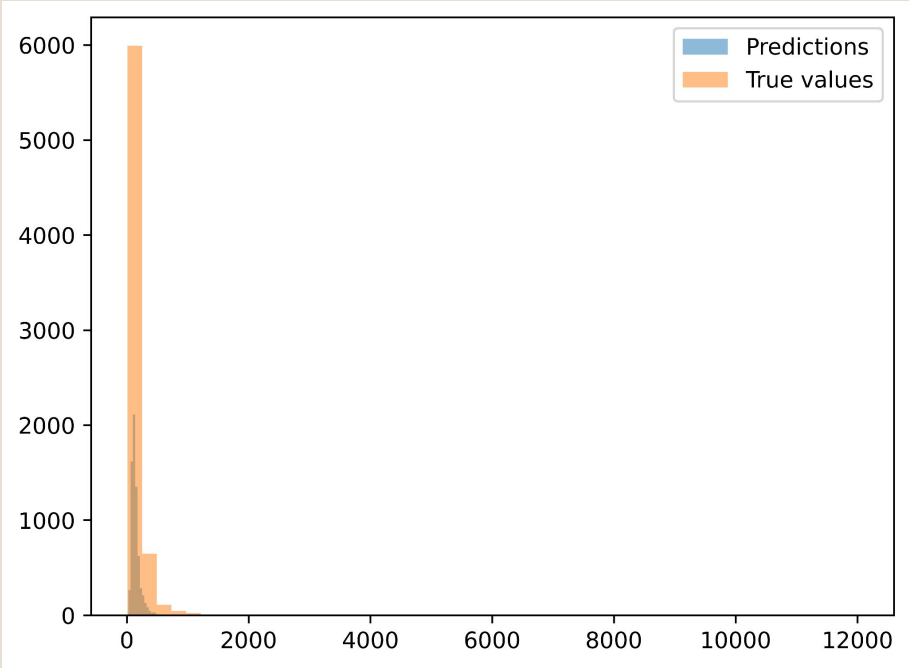
05 - Results:
True vs. Predicted Distribution (Log Price)

Visual representation comparing the distribution of true log prices against predicted log prices, assessing model accuracy.



05 – Results:
True vs. Predicted Distribution (Price)

Visualize how the predicted price distribution aligns with the actual price distribution, noting discrepancies and accuracy.



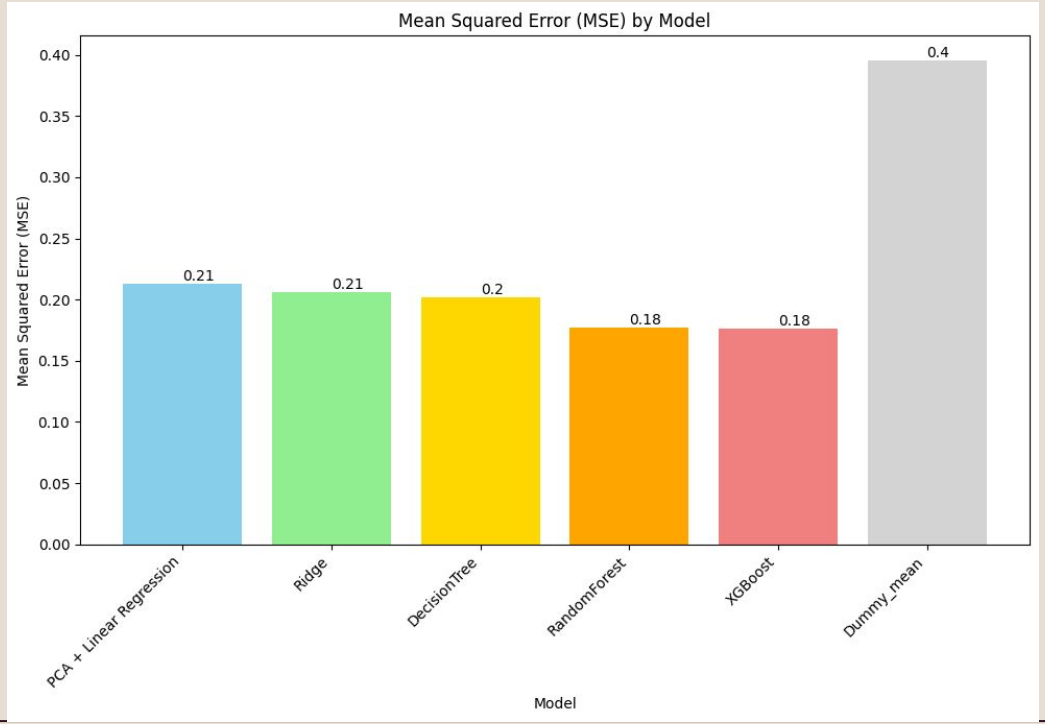
05 - Results: Metrics of LogPrice

Presenting the MSE values for each model, reflecting their accuracy in predicting log price without exponential transformation.

	MSE	R^2
PCA + Linear Regression	0.21331098310620425	0.46112772228405463
Ridge	0.20629578082517336	0.47884972598385966
DecisionTree	0.20142974152346727	0.4911424529863888
RandomForest	0.17680824218069371	0.5533419855112386
XGBoost	0.17650707675125435	0.55410279819218
Dummy_mean	0.39585233573262424	-1.3439013642591036e-05

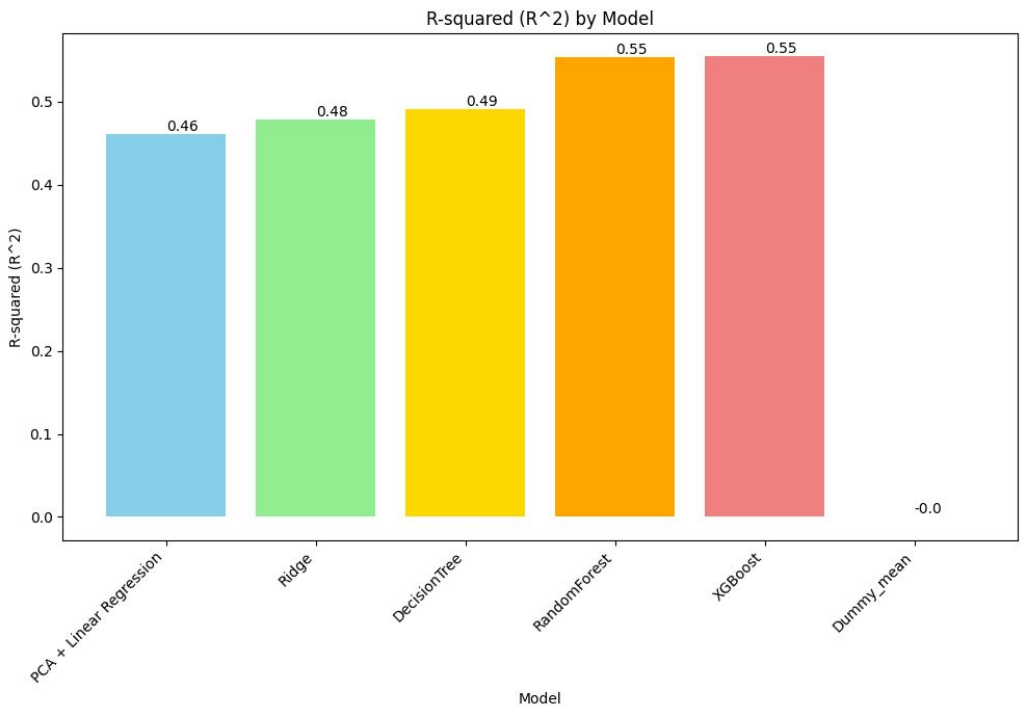
05 - Results: MSE by Model

Display a comparative analysis of MSE values across different models, illustrating their predictive efficacy for log prices.



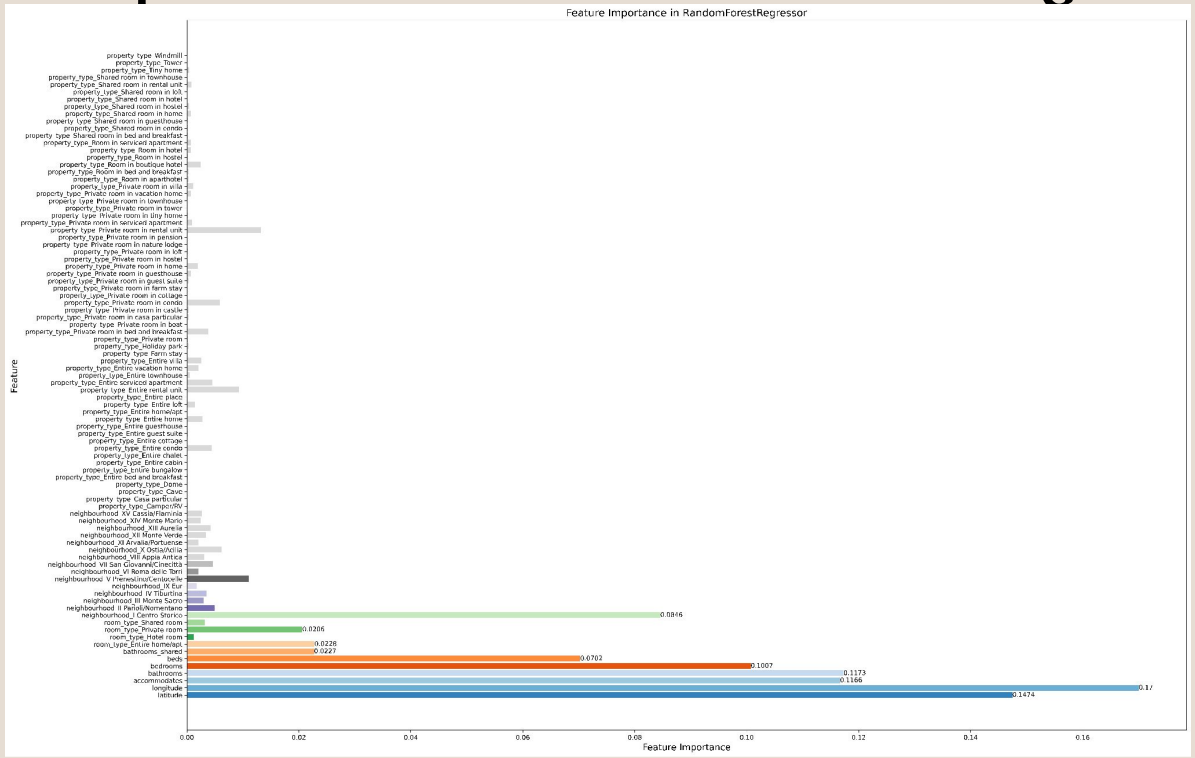
05 - Results: R^2 by Model

Present R² scores for each model, showing the variance explained and identifying the best-performing model in predicting log price.



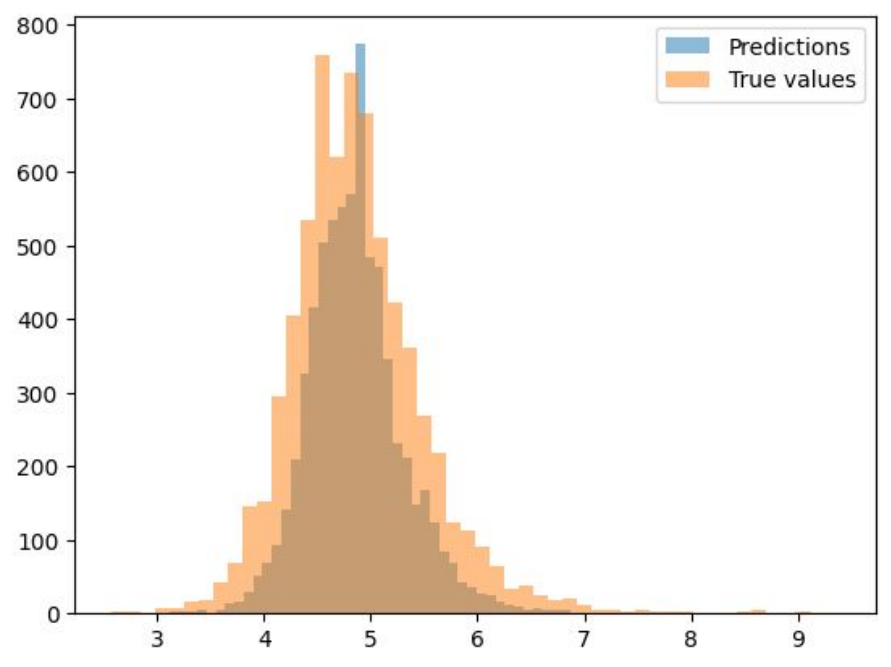
05 - Results:

Feature Importance in Random Forest Regressor



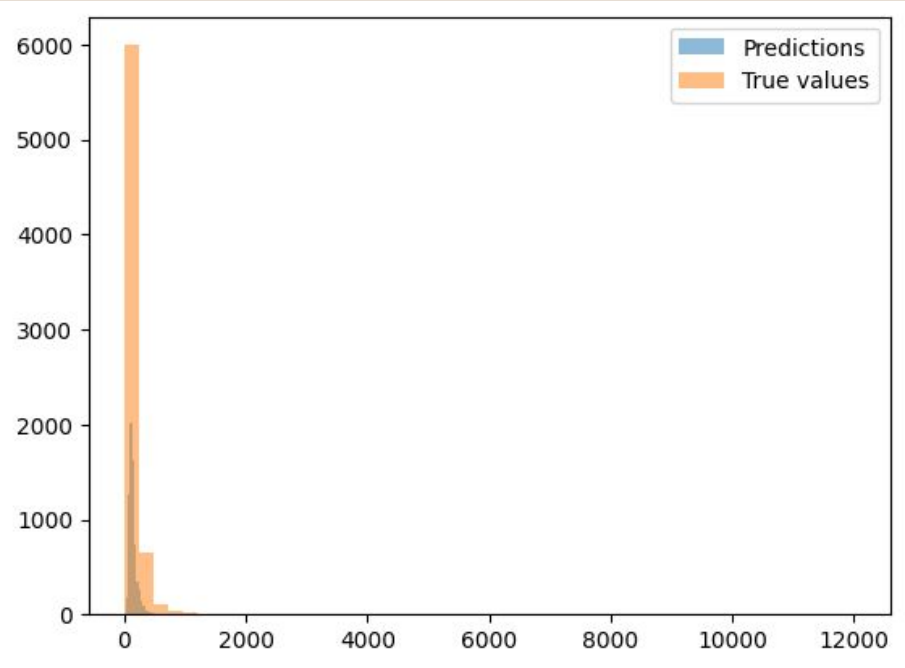
05 - Results:
True vs. Predicted Distribution (Log Price)

Visual representation comparing the distribution of true log prices against predicted log prices, assessing model accuracy.



05 - Results:
True vs. Predicted Distribution (Price)

Visualize how the predicted price distribution aligns with the actual price distribution, noting discrepancies and accuracy.



Thanks for your attention

Lorenzo e Chiara