CALIFORNIA HOUSE MARKET PREDICTION

Author: Cayke Felipe dos Anjos

TABLE OF CONTENTS

01
SUMMARY

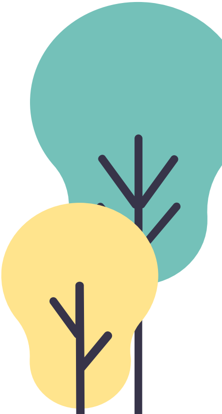
02
BUSINESS PROBLEM

03
DATA

04
METRICS USED

05
MODELS AND RESULTS

06
NEXT STEPS & QUESTIONS

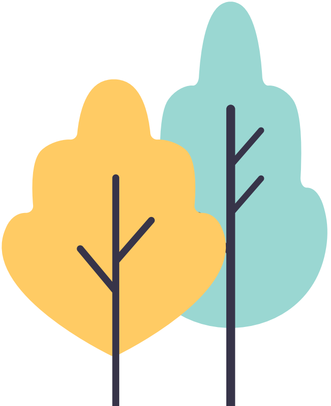


SUMMARY

There are many reasons why knowing the price of a house is important:

1. Buying a house;
2. Selling your own house;
3. Charging rent if you are the landlord;
4. Using as a collateral for a bank loan;
5. Invest;

These are all reasons why the US house market is valued in $50 trillion. The real state is an important industry because all people need a place to live.



BUSINESS PROBLEM

The data is available at https://www.kaggle.com/competitions/california-house-prices/overview competition.

Our client is a real state company in California and is always looking for the best deals for their clients who are trying to buy a house, meaning good houses being sold for less than what they should, or for a higher price for those are willing to sell.

This project aims to analyze data from the California House Market in order to be able to accurately predict the price of a house based on its features. Besides trying to answer which model is the best and how well does the algorithm do when compared to others we also try to answer:

* What are features that increase the house price?
* What are the most important features used to predict the price?

MYHOME
REAL ESTATE

DATA

Datasets

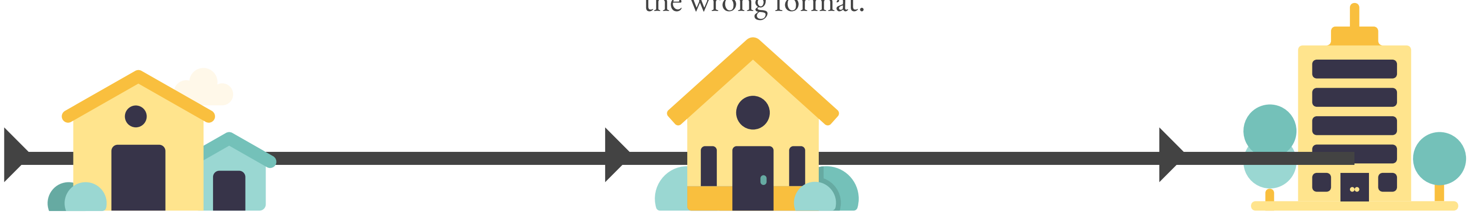
We are given a training and test data set, each with about 40k entries.

Cleaning

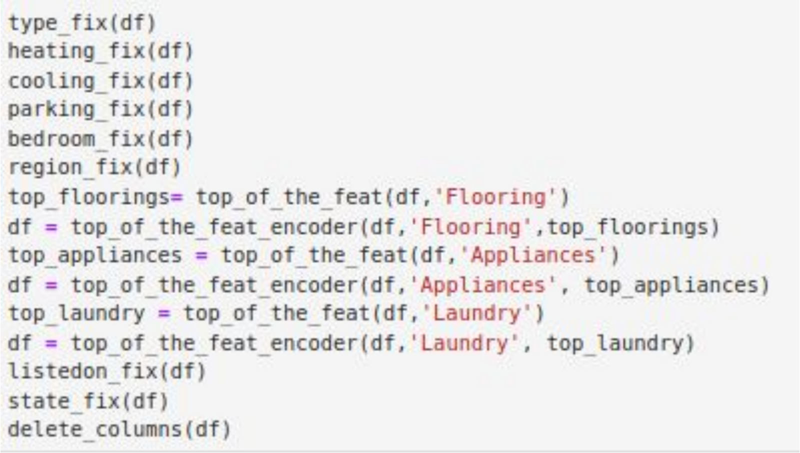
Each dataset contains redundant columns, missing values, outliers and data in the wrong format.

EDA

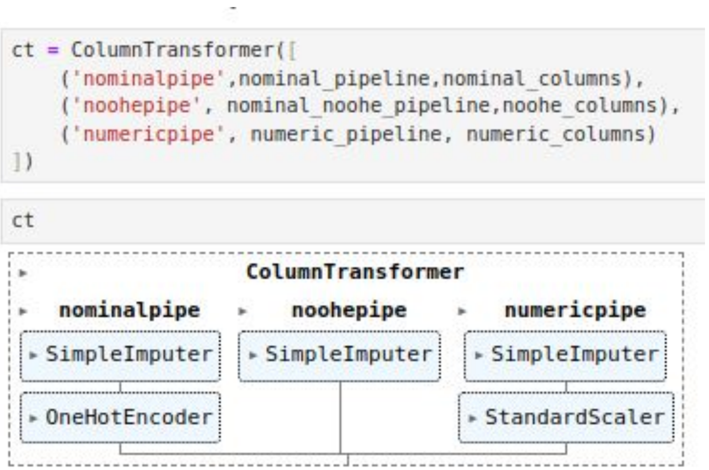
Following the data cleaning, we analyze some of its important features.



DATA



```
type_fix(df)
heating_fix(df)
cooling_fix(df)
parking_fix(df)
bedroom_fix(df)
region_fix(df)
top_floorings= top_of_the_feat(df,'Flooring')
df = top_of_the_feat_encoder(df,'Flooring',top_floorings)
top_appliances = top_of_the_feat(df,'Appliances')
df = top_of_the_feat_encoder(df,'Appliances', top_appliances)
top_laundry = top_of_the_feat(df,'Laundry')
df = top_of_the_feat_encoder(df,'Laundry', top_laundry)
listedon_fix(df)
state_fix(df)
delete_columns(df)
```



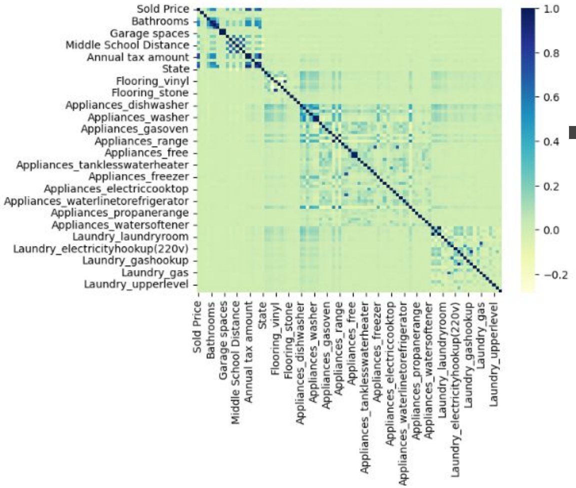
```
ct = ColumnTransformer([
    ('nominalpipe',nominal_pipeline,nominal_columns),
    ('noohepipe', nominal_noohe_pipeline,noohe_columns),
    ('numericpipe', numeric_pipeline, numeric_columns)
])
```

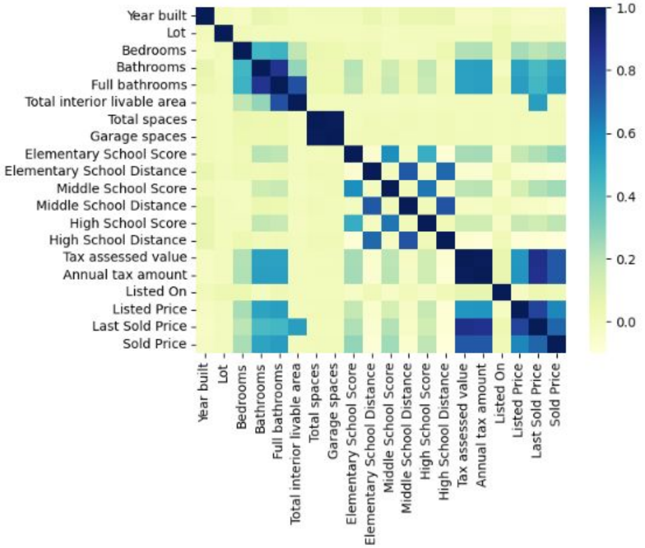
```
ct
```

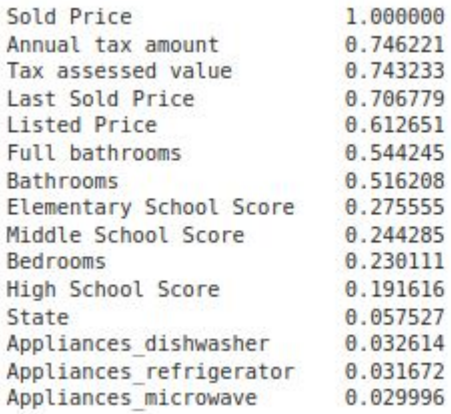
On the left we have different functions created to clean and obtain most relevant information about different features (fix) and to the right we have our preprocessing part of our pipeline. Only after doing both parts is our dataset able to be used

Cleaning

DATA





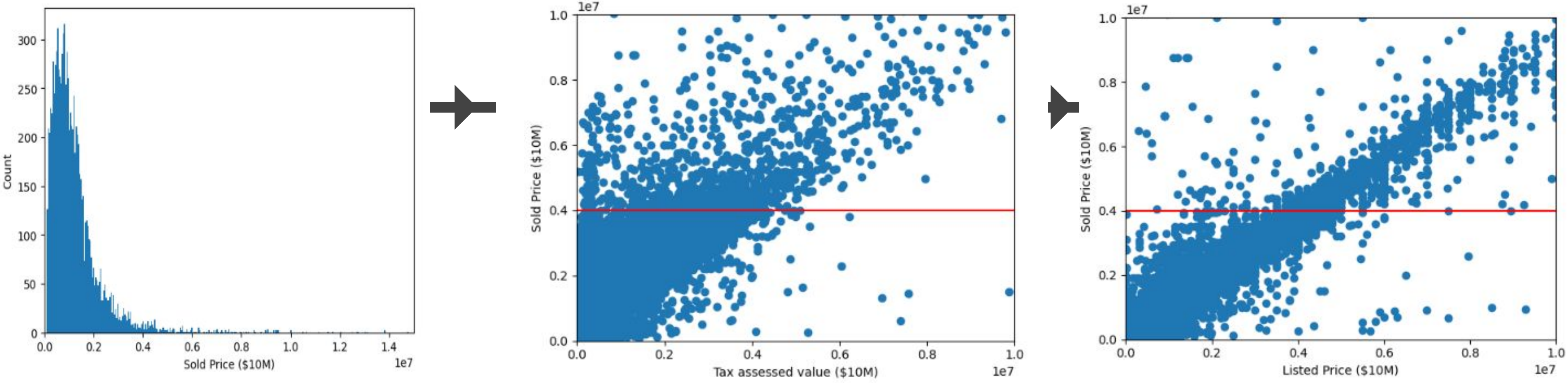


```
Sold Price                   1.000000
Annual tax amount            0.746221
Tax assessed value           0.743233
Last Sold Price              0.706779
Listed Price                 0.612651
Full bathrooms               0.544245
Bathrooms                    0.516208
Elementary School Score      0.275555
Middle School Score          0.244285
Bedrooms                     0.230111
High School Score            0.191616
State                        0.057527
Appliances_dishwasher        0.032614
Appliances_refrigerator      0.031672
Appliances_microwave         0.029996
```

EDA

The correlation map allows us to find features that correlate with Sold Price target. Not only this is important for feature engineering but also it helps us to divide the dataset between "normal prices" and outliers.

DATA



We use both Tax Assessed Value and Listed Price to define an outlier. Because of missing data in both columns, we first define an outlier of having a Tax Assessed Value of over $5M. If that data is not present, we check if Listed Price is over $5M.

EDA

METRICS USED

The most important metric for this problem is called Log RMSE

$$\text{logRMSE}(y_{\text{true}}, y_{\text{pred}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log y_{\text{true}} - \log y_{\text{pred}})^2}$$

In this metric, the error is calculated between the different orders of magnitude between what is true and what is predicted.

However, y_{pred} cannot be negative and some of the models do not deal well with that restriction. Therefore, we use RMSE for hyperparameter tuning but choose the best model based on logRMSE.

If y_{pred} = 1,000,000 but y_{true} = 100,000:

Loss = 900,000

logLoss = 1

If y_{pred} = 10,000,000 but y_{true} = 1,000,000:

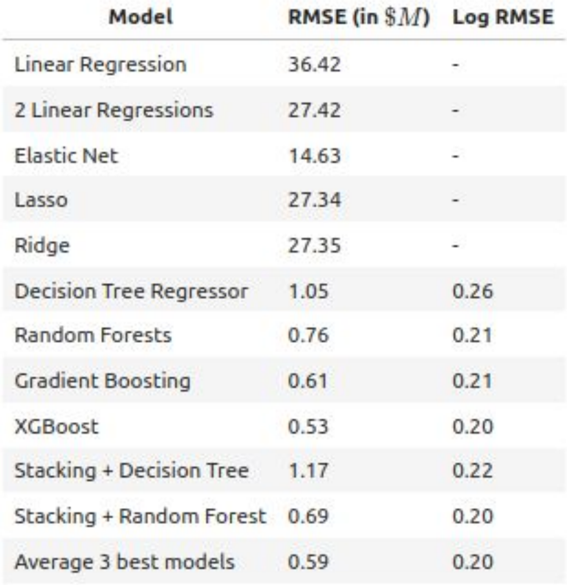
Loss = 9,000,000

logLoss = 1

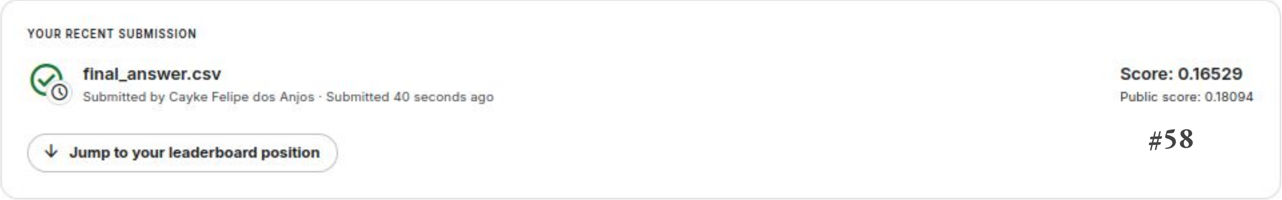
MODELS AND RESULTS

Results are as follow:

1. Linear Models did not do very well, failing to predict that all the prices should all be positive;
2. Creating a dataset for outliers lowered RMSE by 25% and regularization (Elastic Net) lowered RMSE by 50%;
3. Ensemble models were overall the best models;
4. Although Decision Trees was not the best model, it ran faster than any others;
5. We attempted to use other models but they took too long (AdaBoost, StackingRegressor), restarted the kernel due to computation being too intense (Polynomial Features + LASSO) or did not couple well with sklearn Pipeline (NN) ;



Model	RMSE (in $\$M$)	Log RMSE
Linear Regression	36.42	-
2 Linear Regressions	27.42	-
Elastic Net	14.63	-
Lasso	27.34	-
Ridge	27.35	-
Decision Tree Regressor	1.05	0.26
Random Forests	0.76	0.21
Gradient Boosting	0.61	0.21
XGBoost	0.53	0.20
Stacking + Decision Tree	1.17	0.22
Stacking + Random Forest	0.69	0.20
Average 3 best models	0.59	0.20



YOUR RECENT SUBMISSION

final_answer.csv

Submitted by Cayke Felipe dos Anjos · Submitted 40 seconds ago

Score: 0.16529

Public score: 0.18094

#58

NEXT STEPS

SMART REALTORS
We need a standardized way to input data;

MORE MODELS
We were not able to run some models for lack of time;

Use logRMSE
We can now use logRMSE for models that will not predict a negative y value;

LOCATION LOCATION LOCATION
While we did consider the region the house was, we did not consider the city

Thanks for watching

Cayke Felipe dos Anjos

✉ caykefelipe01@gmail.com
@cayke-fda
in cayke-fda