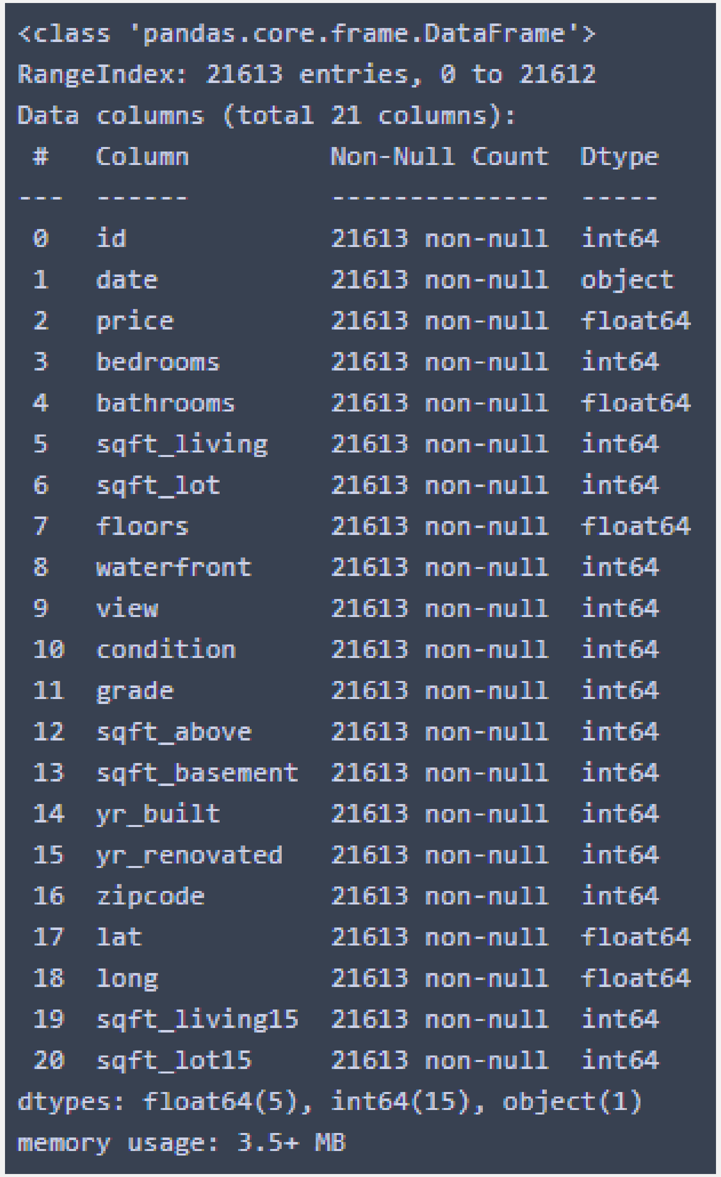
KING COUNTY HOUSE SALES PREDICTION

Made by Mirolim Saidakhmatov

THE DATASET

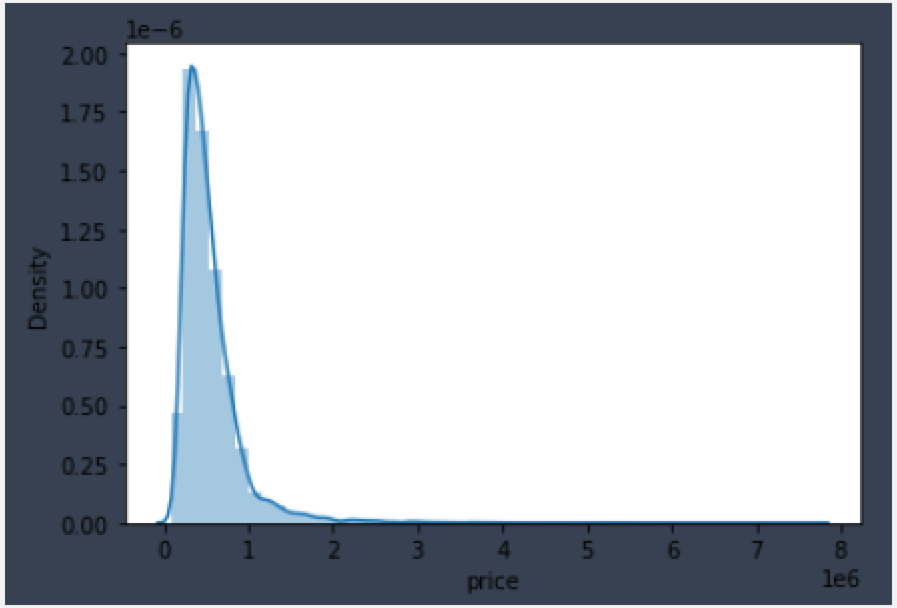
- The dataset was taken from: https://www.kaggle.com/harlfoxem/housesalesprediction
- It consists of data about house sales in King County.
- The dataset is consisted of 21613 rows with 21 columns.
- It has no null values.
- It has right-skewed distribution for target column named 'price'.



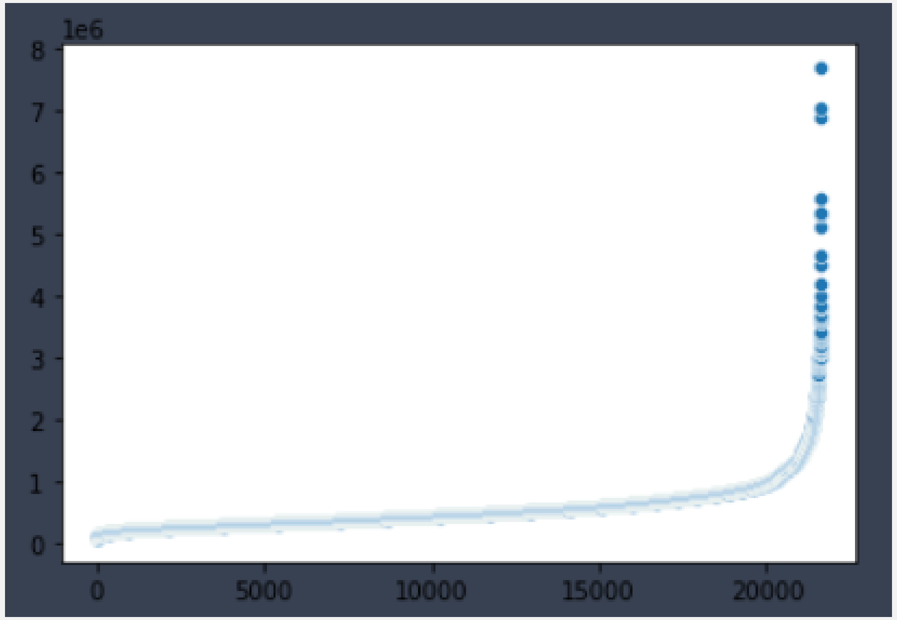
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             21613 non-null  int64  
 1   date           21613 non-null  object 
 2   price          21613 non-null  float64
 3   bedrooms       21613 non-null  int64  
 4   bathrooms      21613 non-null  float64
 5   sqft_living    21613 non-null  int64  
 6   sqft_lot       21613 non-null  int64  
 7   floors         21613 non-null  float64
 8   waterfront     21613 non-null  int64  
 9   view           21613 non-null  int64  
 10  condition      21613 non-null  int64  
 11  grade          21613 non-null  int64  
 12  sqft_above     21613 non-null  int64  
 13  sqft_basement  21613 non-null  int64  
 14  yr_built       21613 non-null  int64  
 15  yr_renovated   21613 non-null  int64  
 16  zipcode        21613 non-null  int64  
 17  lat            21613 non-null  float64
 18  long           21613 non-null  float64
 19  sqft_living15  21613 non-null  int64  
 20  sqft_lot15     21613 non-null  int64  
dtypes: float64(5), int64(15), object(1)
memory usage: 3.5+ MB
```

EXPLANATORY DATA ANALYSIS

As we said, the distribution of price column



And the scatter plot of the target

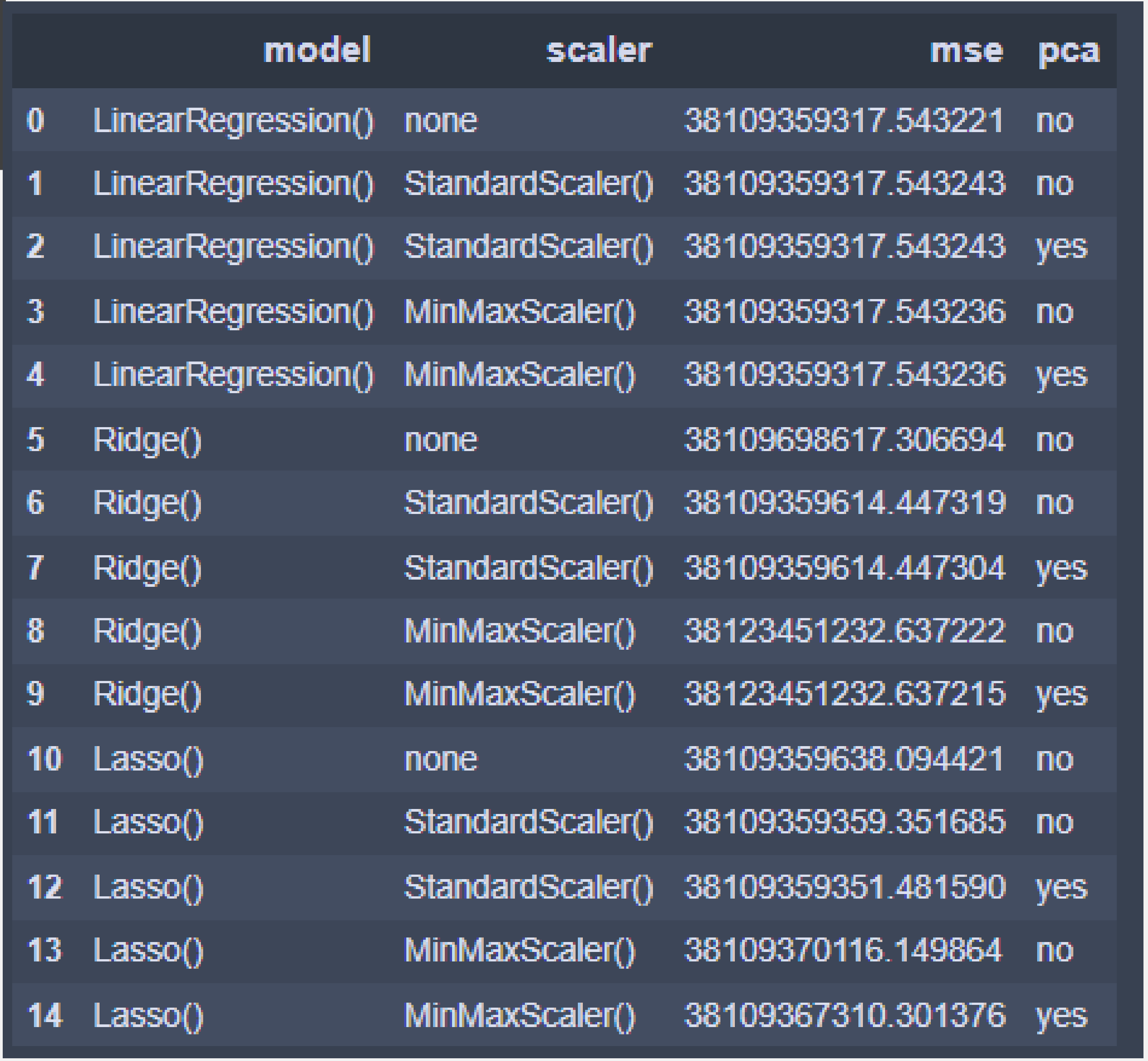


PREPROCESSING

- We realized that "id" and "date" columns we will not use as they do not contain any specific information and so dropped them.
- After, we splitted our dataset into train and test set.
- Then by running double loop, we train and tested our models by choosing scalers, models, PCA.
- We tested two scalers: Standard Scaler and Min Max Scaler.
- We had three models to compare: Linear Regression, Ridge, Lasso.
- And tried PCA to select best features.
- We had not done any feature engineering techniques as the dataset was ready to work with, i. e. all columns were converted to numeric values.

COMPARING RESULTS

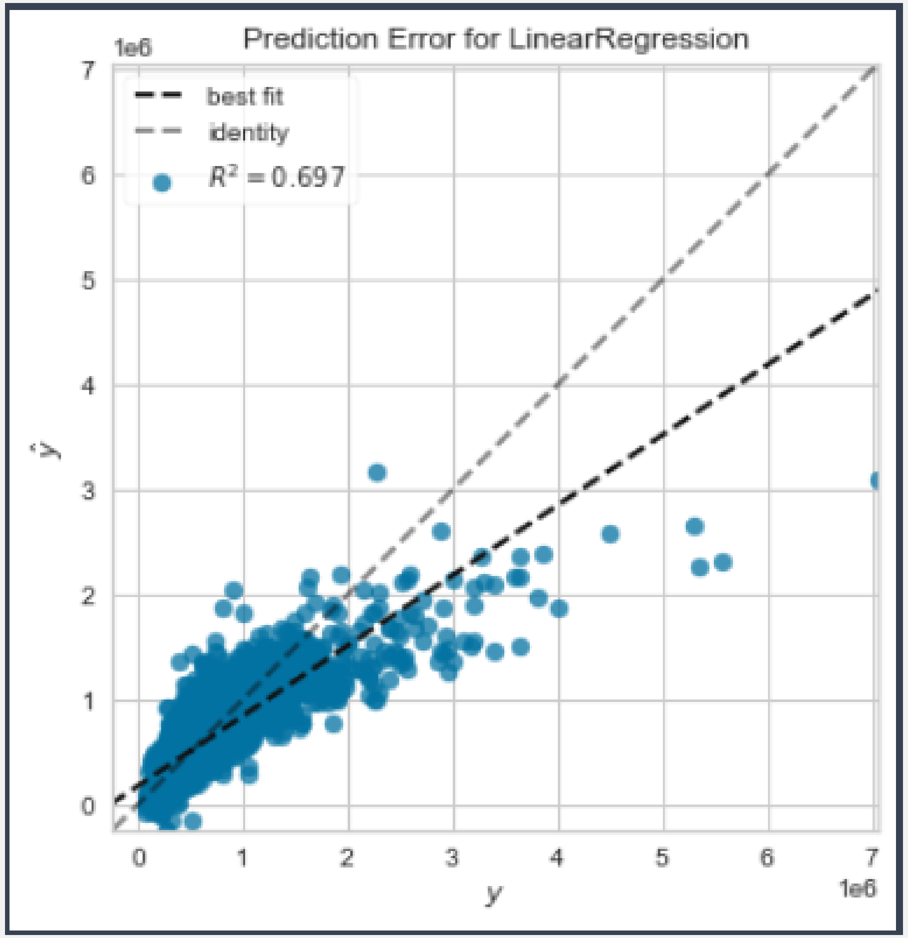
- After all, we stores results in one data frame and sorted for convenience.
- As we see, the best performance was showed by simple Linear Regression with no scaler and no PCA, which gave us the least mean squared error.



	model	scaler	mse	pca
0	LinearRegression()	none	38109359317.543221	no
1	LinearRegression()	StandardScaler()	38109359317.543243	no
2	LinearRegression()	StandardScaler()	38109359317.543243	yes
3	LinearRegression()	MinMaxScaler()	38109359317.543236	no
4	LinearRegression()	MinMaxScaler()	38109359317.543236	yes
5	Ridge()	none	38109698617.306694	no
6	Ridge()	StandardScaler()	38109359614.447319	no
7	Ridge()	StandardScaler()	38109359614.447304	yes
8	Ridge()	MinMaxScaler()	38123451232.637222	no
9	Ridge()	MinMaxScaler()	38123451232.637215	yes
10	Lasso()	none	38109359638.094421	no
11	Lasso()	StandardScaler()	38109359359.351685	no
12	Lasso()	StandardScaler()	38109359351.481590	yes
13	Lasso()	MinMaxScaler()	38109370116.149864	no
14	Lasso()	MinMaxScaler()	38109367310.301376	yes

RESULTS VISUALIZATION

- In the final part, we have plot showing error, i. e. R^2 difference between pure data for testing and predicted data for that testing set.
- The "identity"'s broken line shows the original data, whereas best fit is showing the performance of the predicted values.



CONCLUSION

- In the end, we have built a model to predict house prices.
- We have tried 3 models, with Principal Components Analysis technique for feature selection.
- The best performance was obtained by pure Linear Regression without any scalings and PCA.
- For more details you can run IPython notebook named "test_task.ipynb".
- Thank you for your attention.