HW3

In this homework, you will build an Jupyter notebook for predicting house prices, using data from King County, USA, the region where the city of Seattle is located.

1. Polynomial Regression

Split the original data into train-validate set (90%) and test set (10%) using random state = 0.

Build a polynomial regression model with the degree = 15 using just ‘sqft_living’ on the train-validation set. Remember to run standard scaler to normalize the features before building your model.

Report the model’s performance on the train-validate set and the test set.

2. Ridge Regression

Split the original data into train-validate set (90%) and test set (10%) using random state = 0.

For each L2_penalty λ in [10^3, 10^3.5, 10^4, 10^4.5, ..., 10^9], use GridSearchCV and 10-fold cross validation to compare the performance of the ridge regression with polynomial degree = 15 using just ‘sqft_living’ on the train-validation set. Remember to run standard scaler to normalize the features before building your model.

Report which L2 penalty λ produced the lowest average validation error. Report the best model’s performance on the test set.

3. Lasso Regression

Part 1

Create new features by performing following transformation on inputs: (assume you have named your data frame “sales”)

```
from math import log, sqrt

sales['sqft_living_sqrt'] = sales['sqft_living'].apply(sqrt)

sales['sqft_lot_sqrt'] = sales['sqft_lot'].apply(sqrt)

sales['bedrooms_square'] = sales['bedrooms']*sales['bedrooms']
```

```
sales['floors_square'] = sales['floors']*sales['floors']
```

- Squaring bedrooms will increase the separation between not many bedrooms (e.g. 1) and lots of bedrooms (e.g. 4) since 1^2 = 1 but 4^2 = 16. Consequently this variable will mostly affect houses with many bedrooms.
- On the other hand, taking square root of sqft_living will decrease the separation between big house and small house. The owner may not be exactly twice as happy for getting a house that is twice as big.

Split the data into train-validate set (90%) and test set (10%) using random state = 0.

Run Lasso regression with $\lambda = 100$ using the following features on the train-validate set. Remember to run standard scaler to normalize the features before building your model.

['bedrooms', 'bedrooms_square', 'bathrooms', 'sqft_living', 'sqft_living_sqrt', 'sqft_lot', 'sqft_lot_sqrt', 'floors', 'floors_square', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated']

Which features have been chosen by LASSO, i.e. which features were assigned nonzero weights? Report the model’s performance on the test set.

Part 2 (continue from Part 1)

For each L1_penalty λ in [10^1, 10^1.5, 10^2, 10^2.5, ..., 10^7], use GridSearchCV and 10-fold cross validation to compare the performance of the lasso regression using all the features used in Part 1 on the train-validation set. Remember to run standard scaler to normalize the features.

Report which L1 penalty $\boldsymbol{\lambda}$ produced the lowest average validation error. Which features have been chosen by the best model, i.e. which features were assigned nonzero weights? Report the best model’s performance on the test set.