SML2

October 9, 2019

ASSIGNMENT 2
Mounica Subramani

0.0.1 Problem 1 [Linear regression]

[63]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

import seaborn as sns

from sklearn.metrics import mean_squared_error
from sklearn import preprocessing

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

from numpy.linalg import inv

from statsmodels.regression.linear_model import OLS

from sklearn import warnings

from scipy import stats
```

[64]:
```
# read in data
KChouse_train = pd.read_csv('train.csv')
KChouse_test = pd.read_csv('test.csv')

# ignore the columns id, date, unnamed column as well as the categorical column␣
↪zipcode.
KChouse_train = KChouse_train.drop(columns = ['zipcode','Unnamed: 0'])
KChouse_test = KChouse_test.drop(columns = ['id','zipcode','date','Unnamed: 0'])
```

[65]:
```
KChouse_train.head()
```

[65]:
```
      price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  waterfront  \
0  221900.0         3       1.00         1180      5650     1.0           0
1  538000.0         3       2.25         2570      7242     2.0           0
2  180000.0         2       1.00          770     10000     1.0           0
3  604000.0         4       3.00         1960      5000     1.0           0
4  510000.0         3       2.00         1680      8080     1.0           0

   view  condition  grade  sqft_above  sqft_basement  yr_built  yr_renovated  \
0     0          3      7        1180              0      1955             0
1     0          3      7        2170            400      1951          1991
2     0          3      6         770              0      1933             0
3     0          5      7        1050            910      1965             0
4     0          3      8        1680              0      1987             0

       lat     long  sqft_living15  sqft_lot15
0  47.5112 -122.257           1340        5650
1  47.7210 -122.319           1690        7639
2  47.7379 -122.233           2720        8062
3  47.5208 -122.393           1360        5000
4  47.6168 -122.045           1800        7503
```

[66]:
```
# correlation coefficients of features/variables
KChouse_train.corr()['price'].sort_values(ascending=False)
```

[66]:
```
price            1.000000
sqft_living      0.704776
grade            0.647349
sqft_living15    0.645106
sqft_above       0.582407
bathrooms        0.487157
view             0.445316
sqft_basement    0.367365
lat              0.365770
waterfront       0.317143
bedrooms         0.307058
floors           0.239935
sqft_lot15       0.161746
sqft_lot         0.146645
yr_renovated     0.146348
condition        0.073961
long             0.032846
yr_built         0.016055
Name: price, dtype: float64
```

(a) Use an existing package to train a multiple linear regression model on the training set using all the features (except the ones excluded above). Report the coefficients of the linear regression models and the MSE metric on the training data

[67]:
```
# Linear regression model
X_train = KChouse_train.drop(['price'], axis=1)
y_train = KChouse_train[['price']]

model = LinearRegression()

scores = cross_val_score(model, X_train, y_train, cv=10)
print("Cross-validation scores: {}".format(scores))
print("Average cross-validation score: {:.2f}".format(scores.mean()))
# print("X_train.shape:",X_train.shape)
```

```
Cross-validation scores: [0.67837004 0.70416297 0.6125577  0.72692348 0.55953864
0.79355793
 0.71355718 0.7597062  0.67484746 0.72691946]
Average cross-validation score: 0.70
```

[68]:
```
model_fitted = model.fit(X_train, y_train)
```

[69]:
```
predictions = model_fitted.predict(X_train)
# predictions
```

[70]:
```
print("Mean squared error for train (unscaled) data:
↪",mean_squared_error(y_train,predictions))
```

```
Mean squared error for train data: 31486167775.794888
```

(b) Perform feature standardization so that each feature (including the response) has mean 0 and variance of 1. Train again a linear regression model on the training data and report the MSE on the training data.

Scaling data so that each feature has mean 0 and standard deviation 1.

[71]:
```
scaler = StandardScaler()
data_scaled = scaler.fit_transform(KChouse_train)
data_scaled_test = scaler.fit_transform(KChouse_test)
```

[72]:
```
data_scaled
```

[72]:
```
array([[-0.87974769, -0.40982347, -1.44988843, ..., -0.35519332,
        -0.96563661, -0.3128578 ],
       [ 0.05182493, -0.40982347,  0.28318404, ..., -0.7998304 ,
        -0.44332966, -0.23355563],
       [-1.00323042, -1.58410276, -1.44988843, ..., -0.18307573,
         1.09374507, -0.21669046],
       ...,
       [ 0.0975047 , -1.58410276, -1.44988843, ..., -0.86437449,
        -1.02532883, -0.4185143 ],
       [-0.97390696, -1.58410276, -1.44988843, ...,  1.07911986,
        -0.80148299, -0.40352304],
       [-0.68199849, -0.40982347, -0.06343045, ..., -0.46993837,
```

```
         0.39236146, -0.15736334]])
```

[73]:
```
df = pd.DataFrame(data=data_scaled[:,0:],
                  index=data_scaled[:,0],
                  columns=['price', 'bedrooms', 'bathrooms', 'sqft_living',
        'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
        'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'lat',
        'long', 'sqft_living15', 'sqft_lot15'])
```

[74]:
```
df = df.reset_index(drop=True)
df.shape
```

[74]:
```
(1000, 18)
```

[75]:
```
X_train_scaled = df.drop(['price'], axis=1)
y_train_scaled = df[['price']]

model1 = LinearRegression()

scores = cross_val_score(model1, X_train_scaled, y_train_scaled, cv=10)
print("Cross-validation scores: {}".format(scores))
print("Average cross-validation score: {:.2f}".format(scores.mean()))
# X_train_scaled.columns
```

```
Cross-validation scores: [0.67837004 0.70416297 0.6125577  0.72692348 0.55953864
0.79355793
 0.71355718 0.7597062  0.67484746 0.72691946]
Average cross-validation score: 0.70
```

[76]:
```
model_fitted1 = model1.fit(X_train_scaled, y_train_scaled)
```

[77]:
```
predictions_scaled = model_fitted1.predict(X_train_scaled)
# predictions_scaled
```

[78]:
```
model_fitted1.coef_
```

[78]:
```
array([[-0.03690325,  0.05460245,  0.16724347,  0.03206976,  0.0237055 ,
         0.18785555,  0.14204967,  0.03820676,  0.27181372,  0.14231485,
         0.07997506, -0.19934981,  0.05090017,  0.23097972, -0.00305083,
         0.13432109, -0.03810604]])
```

[79]:
```
print("Mean squared error for scaled train data:
 ↪",mean_squared_error(y_train_scaled,predictions_scaled))
```

```
Mean squared error for scaled train data: 0.2734665681293983
```

(c) Evaluate both models on the testing set. Report the MSE on the testing set

Linear regression model on unscaled test data

[80]:
```
X_test = KChouse_test.drop(['price'], axis=1)
y_test = KChouse_test[['price']]

model2 = LinearRegression()

scores = cross_val_score(model2, X_test, y_test, cv=10)
print("Cross-validation scores: {}".format(scores))
print("Average cross-validation score: {:.2f}".format(scores.mean()))
```

```
Cross-validation scores: [0.572243   0.7302379  0.4195207  0.69421544 0.58056117
0.64406129
 0.58310599 0.67480098 0.69470867 0.5749148 ]
Average cross-validation score: 0.62
```

[81]:
```
model_fitted2 = model2.fit(X_test, y_test)
```

[82]:
```
predictions2 = model_fitted2.predict(X_test)
# predictions
```

[83]:
```
model_fitted2.coef_
```

[83]:
```
array([[-5.09706680e+04,  4.53522145e+04,  1.29557364e+02,
         4.60107448e-01, -4.08290867e+03,  6.67916885e+05,
         6.41825495e+04,  3.74395531e+04,  8.69134817e+04,
         7.58952045e+01,  5.36621579e+01, -2.59297980e+03,
        -5.13466279e+00,  5.67370348e+05, -7.67745545e+04,
         3.41095820e+01, -8.48109171e-01]])
```

[84]:
```
print("Mean squared error for non scaled test data:
 ↪",mean_squared_error(y_test,predictions2))
```

```
Mean squared error for non scaled test data: 54185036855.79511
```

Linear regression on scaled test data

[85]:
```
df_scaled = pd.DataFrame(data=data_scaled_test[1:,0:],
                  index=data_scaled_test[1:,0],
                  columns=['price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'lat',
       'long', 'sqft_living15', 'sqft_lot15'])

df_scaled = df_scaled.reset_index(drop=True)
```

[86]:
```
X_test_scaled = df_scaled.drop(['price'], axis=1)
y_test_scaled = df_scaled[['price']]

model3 = LinearRegression()

scores = cross_val_score(model3, X_test_scaled, y_test_scaled, cv=10)
```

```
print("Cross-validation scores: {}".format(scores))
print("Average cross-validation score: {:.2f}".format(scores.mean()))
```

```
Cross-validation scores: [0.61750399 0.72177335 0.41520807 0.69590594 0.58111104
0.63899562
 0.58430498 0.67685041 0.6919166  0.57445912]
Average cross-validation score: 0.62
```

[87]:
```
model_fitted3 = model3.fit(X_test_scaled, y_test_scaled)
```

[88]:
```
predictions_scaled1 = model_fitted3.predict(X_test_scaled)
# predictions_scaled1
```

[89]:
```
model_fitted3.coef_
```

[89]:
```
array([[-0.11656073,  0.08800974,  0.23817802,  0.0661224 , -0.00525078,
         0.1852539 ,  0.12709521,  0.06128675,  0.25616884,  0.21405969,
         0.08813423, -0.17705287, -0.00531226,  0.19009029, -0.02545619,
         0.05815344, -0.0584464 ]])
```

[90]:
```
print("Mean squared error for scaled test data:
↪",mean_squared_error(y_test_scaled,predictions_scaled1))
```

```
Mean squared errorfor scaled test data: 0.3253062062957616
```

(d) Interpret the results in your own words. Which features contribute mostly to the linear regression model? Is the model fitting the data well? How large is the model error?

- Grade feature contributes mostly to the linear regression model. It has the highest coefficient value of 0.25616884.
- We din generalize the model well by taking all the features of the data into modelling.
- The model is fitting the data well and it is evident from the r-sqaured value calculated below as the difference between r squared calculated for training and testing has no much difference.
- The error is not large and it is appropriate for the data used and the model as well.

[91]:
```
from sklearn.metrics import r2_score
r2_test = r2_score(y_test_scaled, predictions_scaled1)
print("Rsquared error on testing data:",r2_test)

r2_train = r2_score(y_train_scaled,predictions_scaled)
print("Rsquared error on training data:",r2_train)
```

```
Rsquared error on testing data: 0.6749701848743102
Rsquared error on training data: 0.7265334318706018
```

0.0.2 Problem 2 [Closed-form solution for linear regression]

(a) Implement simple linear regression using the closed form and train a model for one feature (sqft living) using the training set. Write code to predict a response for a new single-dimensional data point in the testing set.

Simple linear regression with one feature(sqft living)

[92]:
```
# obtain the feature matrix
KChouse_train1 = df.drop(columns = ['price'])
KChouse_test1 = df_scaled.drop(columns = ['price'])

X = KChouse_train1[['sqft_living15']]
X['ones'] = 1
X_test = KChouse_test1[['sqft_living15']]
X_test['ones'] = 1

# obtain the target variable
y = df[['price']]
y_test = df_scaled[['price']]

# calculate coefficients using closed-form solution
coeffs_CF = inv(X.transpose().dot(X)).dot(X.transpose()).dot(y)

y_prediction = X.dot(coeffs_CF)
y_test_prediction = X_test.dot(coeffs_CF)
```

```
C:\Users\mouni\Anaconda3\lib\site-packages\ipykernel_launcher.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy

C:\Users\mouni\Anaconda3\lib\site-packages\ipykernel_launcher.py:8:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
```

[93]:
```
# extract the feature names into list

feature_names = list(X.columns)
feature_names

# convert list to array

features = np.asarray(feature_names)
features
```

```
# convert both the array vectors into dataframe
res = pd.DataFrame(coeffs_CF)

res1 = pd.DataFrame(features)

# merge dataframes
results = pd.merge(res1, res, left_index=True, right_index=True)
results = results.rename(columns={'0_x':'Features','0_y':'coeffs_CF'})

results
```

[93]:

```
        Features      coeffs_CF
0  sqft_living15  6.451060e-01
1           ones  6.288373e-17
```

[94]:

```
y_prediction.head()
```

[94]:

```
          0
0 -0.622938
1 -0.285995
2  0.705582
3 -0.603684
4 -0.180098
```

[95]:

```
y_test_prediction.head()
```

[95]:

```
          0
0 -0.355080
1 -0.141438
2 -0.057839
3 -0.292845
4 -0.596589
```

[96]:

```
print("Mean squared error for training data (simple Linear regression):
 ↪",mean_squared_error(y,y_prediction))
print("Mean squared error for testing data (simple Linear regression):
 ↪",mean_squared_error(y_test,y_test_prediction))
```

```
Mean squared error for training data: 0.5838382382385958
Mean squared error for testing data: 0.6764263056279269
```

(b) Implement the closed-from solution for multiple linear regression using matrix operations and train a model on the training set. Write code to predict a response for a new multi-dimensional data point in the testing set.

Multiple linear regression with all features

[97]:

```
X1 = KChouse_train1
X1['ones'] = 1

X_test1 = KChouse_test1
```

```
X_test1['ones'] = 1

# obtain the target variable
y1 = df[['price']]
y_test1 = df_scaled[['price']]

# calculate coefficients using closed-form solution
coeffs_CF_1 = inv(X1.transpose().dot(X1)).dot(X1.transpose()).dot(y1)

y_prediction1 = X1.dot(coeffs_CF_1)
y_test_prediction1 = X_test1.dot(coeffs_CF_1)
```

[98]:
```
# extract the feature names into list

feature_names1 = list(X1.columns)
# print(feature_names1)

# convert list to array

features1 = np.asarray(feature_names1)
# print(features1)

# convert both the array vectors into dataframe
res2 = pd.DataFrame(coeffs_CF_1)

res3 = pd.DataFrame(features1)

# merge dataframes
results1 = pd.merge(res2, res3, left_index=True, right_index=True)
results1 = results1.rename(columns={'0_x':'Features','0_y':'coeffs_CF'})

results1
```

[98]:
```
        Features      coeffs_CF
0   1.819630e-02       bedrooms
1   2.082002e-01      bathrooms
2  -5.465047e-01    sqft_living
3   1.959751e-02       sqft_lot
4   1.108606e-02         floors
5   1.888729e-01     waterfront
6   1.634773e-01           view
7   4.165114e-02      condition
8   2.387074e-01          grade
9   8.340260e-01     sqft_above
10  4.157826e-01  sqft_basement
11 -1.993498e-01       yr_built
12  5.090017e-02   yr_renovated
13  2.309797e-01            lat
```

```
14 -3.050828e-03           long
15  1.343211e-01  sqft_living15
16 -3.810604e-02     sqft_lot15
17  3.165437e-15           ones
```

[99]:
```python
y_prediction1.head()
```

[99]:
```
          0
0 -0.979190
1  0.658167
2 -0.586131
3 -0.085220
4 -0.232246
```

[100]:
```python
y_test_prediction1.head()
```

[100]:
```
          0
0 -0.766049
1 -0.971737
2  0.011118
3 -1.221909
4 -0.026984
```

(c) Compare the models given by your implementation with those trained in Problem 1 by the R or Python packages. Report the MSE metrics for the models you implemented on both training and testing sets

- The package generated linear regression model has appropriate mean squared error value just right for the data used.
- The implemented model's mean squared error values a bit high than package model, but the difference is quite neglegible.

MSE for Python package used models

[101]:
```python
print("Mean squared error for scaled train data:
 ↪",mean_squared_error(y_train_scaled,predictions_scaled))
print("Mean squared errorfor scaled test data:
 ↪",mean_squared_error(y_test_scaled,predictions_scaled1))
```

```
Mean squared error for scaled train data: 0.2734665681293983
Mean squared errorfor scaled test data: 0.3253062062957616
```

MSE for implemented model

[102]:
```python
print("Mean squared error for training data:
 ↪",mean_squared_error(y1,y_prediction1))
print("Mean squared error for testing data:
 ↪",mean_squared_error(y_test1,y_test_prediction1))
```

```
Mean squared error for training data: 0.31194738550839385
Mean squared error for testing data: 0.37329326533579754
```

0.0.3 Problem 3 [Gradient descent]

[103]:
```
X2 = KChouse_train1
X2['ones'] = 1

X_test2 = KChouse_test1
X_test2['ones'] = 1

# obtain the target variable
y2 = df[['price']]
y_test2 = df_scaled[['price']]
```

(a) Write code for gradient descent for training linear regression using the algorithm from class

[104]:
```
def gradient_descent1(X,y,theta,alpha,n):

    m = len(y)

    for i in range(n):
        prediction = np.dot(X,theta)

        theta = theta - (1/m) * alpha * (X.T.dot(prediction - y))

    return theta
```

(b) Vary the value of the learning rate (3 different values) and report the value of after different number of iterations (10, 50, and 100). Include the MSE metric on the training and testing set for all values of and number of iterations.

[105]:
```
# np.random.seed(2)
alpha = 0.01
iter = [10,50,100]
cols = KChouse_train1.columns
theta = np.random.randn(len(cols),1)
theta_df = theta
for i in iter:
    theta_j = gradient_descent1(X2,y2,theta_df,alpha,i)
    y_test_pred_gd = X_test2.dot(theta_j)
    print("mean squared error on scaled test data (GD):
 ↪",mean_squared_error(y_test2,y_test_pred_gd))
```

```
mean squared error on scaled test data: 17.464163451221626
mean squared error on scaled test data: 6.97958920199715
mean squared error on scaled test data: 3.445182931877422
```

[106]:
```
alpha = 0.1
iter = [10,50,100]
```

```
cols = KChouse_train1.columns
theta = np.random.randn(len(cols),1)
theta_df = theta
for i in iter:
    theta_j = gradient_descent1(X2,y2,theta_df,alpha,i)
    y_test_pred_gd = X_test2.dot(theta_j)
    print("mean squared error on scaled test data (GD):
 ↪",mean_squared_error(y_test2,y_test_pred_gd))
```

```
mean squared error on scaled test data: 1.413135824269324
mean squared error on scaled test data: 0.361498220452008
mean squared error on scaled test data: 0.34150739669043473
```

[107]:
```
alpha = 0.2
iter = [10,50,100]
cols = KChouse_train1.columns
theta = np.random.randn(len(cols),1)
theta_df = theta
for i in iter:
    theta_j = gradient_descent1(X2,y2,theta_df,alpha,i)
    y_test_pred_gd = X_test2.dot(theta_j)
    print("mean squared error on scaled test data (GD):
 ↪",mean_squared_error(y_test2,y_test_pred_gd))
```

```
mean squared error on scaled test data: 1.5245356316084024
mean squared error on scaled test data: 0.3824370082195767
mean squared error on scaled test data: 0.3449699649933864
```

[115]:
```
alpha = 0.1
iter = [10,50,100,200,300,400,500,600,700,800,900,1000]
cols = KChouse_train1.columns
theta = np.random.randn(len(cols),1)
theta_df = theta
for i in iter:
    theta_j = gradient_descent1(X2,y2,theta_df,alpha,i)
    y_test_pred_gd = X_test2.dot(theta_j)
    print("mean squared error on scaled test data (GD):
 ↪",mean_squared_error(y_test2,y_test_pred_gd))
```

```
mean squared error on scaled test data: 1.8160006493541812
mean squared error on scaled test data: 0.40963304455213345
mean squared error on scaled test data: 0.3573510297650286
mean squared error on scaled test data: 0.34246820956709867
mean squared error on scaled test data: 0.3409119071387248
mean squared error on scaled test data: 0.3406447292674675
mean squared error on scaled test data: 0.3405909841654776
```

```
mean squared error on scaled test data: 0.34057982865194064
mean squared error on scaled test data: 0.34057750551290006
mean squared error on scaled test data: 0.34057702248141386
mean squared error on scaled test data: 0.34057692224133757
mean squared error on scaled test data: 0.3405769014708617
```

0.0.4 Problem 4

(a) Write the derivation of the closed form solution for parameter that minimizes the loss function J() in ridge regression.

- In paper format at the end of the pdf

(b) Modify your linear regression implementation from Problem 2 to handle ridge regression. Take several values of the regularization parameter and output the MSE metric. Plot the value of MSE as a function of . What is the best value of that you found? Compare the results of linear regression and ridge regression on the dataset.

[116]:
```
I = np.identity(18)
lamda = []
MSE = []

# derivation of the closed form solution for parameter  that minimizes the loss␣
↪function J() in ridge regression

for lamda_i in range(0,10000,20):
    lamda.append(lamda_i)
    coeffs_CF_2 = inv(X1.transpose().dot(X1) + lamda_i * I).dot(X1.transpose()).
↪dot(y1)
    # Y prediction values for training data
    y_prediction1_rg = X1.dot(coeffs_CF_2)
    # y prediction values for testing data
    y_test_prediction1_rg = X_test1.dot(coeffs_CF_2)

    mean_error = mean_squared_error(y1,y_prediction1_rg)
    mean_error_test = mean_squared_error(y_test1,y_test_prediction1_rg)

    print("Mean squared error for training data (ridge):
↪",mean_squared_error(y1,y_prediction1_rg))
    print("Mean squared error for testing data (ridge):
↪",mean_squared_error(y_test1,y_test_prediction1_rg))
    print("\n")
    MSE.append(mean_error_test)
```

```
Mean squared error for training data (ridge): 0.31194738550839385
Mean squared error for testing data (ridge): 0.37329326533579754
```

```
Mean squared error for training data (ridge): 0.27357928858440955
Mean squared error for testing data (ridge): 0.3402268155372297


Mean squared error for training data (ridge): 0.27387872101234956
Mean squared error for testing data (ridge): 0.34027999083813976


Mean squared error for training data (ridge): 0.2743231393764978
Mean squared error for testing data (ridge): 0.3404495886233099


Mean squared error for training data (ridge): 0.2748848144849361
Mean squared error for testing data (ridge): 0.34071532960019146


Mean squared error for training data (ridge): 0.27554425161640084
Mean squared error for testing data (ridge): 0.3410631869726186


Mean squared error for training data (ridge): 0.2762870722453272
Mean squared error for testing data (ridge): 0.34148287414599043


Mean squared error for training data (ridge): 0.277102234053876
Mean squared error for testing data (ridge): 0.341966462191841


Mean squared error for training data (ridge): 0.27798096659100985
Mean squared error for testing data (ridge): 0.3425075830816734


Mean squared error for training data (ridge): 0.27891610801660377
Mean squared error for testing data (ridge): 0.34310095097301613


Mean squared error for training data (ridge): 0.27990167604354055
Mean squared error for testing data (ridge): 0.3437420632398855


Mean squared error for training data (ridge): 0.2809325807156815
Mean squared error for testing data (ridge): 0.3444270067699917


Mean squared error for training data (ridge): 0.28200442596952
Mean squared error for testing data (ridge): 0.3451523279478502
```

```
Mean squared error for training data (ridge): 0.2831133684788264
Mean squared error for testing data (ridge): 0.3459149423530667


Mean squared error for training data (ridge): 0.2842560145083424
Mean squared error for testing data (ridge): 0.3467120699491553


Mean squared error for training data (ridge): 0.28542934265521624
Mean squared error for testing data (ridge): 0.34754118709422116


Mean squared error for training data (ridge): 0.2866306446591221
Mean squared error for testing data (ridge): 0.34839998995780264


Mean squared error for training data (ridge): 0.28785747911648085
Mean squared error for testing data (ridge): 0.34928636587973166


Mean squared error for training data (ridge): 0.2891076346109302
Mean squared error for testing data (ridge): 0.35019837040432145


Mean squared error for training data (ridge): 0.2903790998547345
Mean squared error for testing data (ridge): 0.3511342084735976


Mean squared error for training data (ridge): 0.2916700391492076
Mean squared error for testing data (ridge): 0.35209221874302743


Mean squared error for training data (ridge): 0.29297877195148697
Mean squared error for testing data (ridge): 0.3530708602958413


Mean squared error for training data (ridge): 0.29430375566291617
Mean squared error for testing data (ridge): 0.3540687012395934


Mean squared error for training data (ridge): 0.2956435709825937
Mean squared error for testing data (ridge): 0.35508440880894215


Mean squared error for training data (ridge): 0.29699690933122275
Mean squared error for testing data (ridge): 0.3561167406952055
```

```
Mean squared error for training data (ridge): 0.2983625619665918
Mean squared error for testing data (ridge): 0.35716453739092224


Mean squared error for training data (ridge): 0.2997394104968199
Mean squared error for testing data (ridge): 0.35822671538589906


Mean squared error for training data (ridge): 0.3011264185603035
Mean squared error for testing data (ridge): 0.3593022610862243


Mean squared error for training data (ridge): 0.30252262448844575
Mean squared error for testing data (ridge): 0.3603902253535811


Mean squared error for training data (ridge): 0.30392713480310213
Mean squared error for testing data (ridge): 0.36148971858157997


Mean squared error for training data (ridge): 0.30533911842828126
Mean squared error for testing data (ridge): 0.36259990624063665


Mean squared error for training data (ridge): 0.3067578015171429
Mean squared error for testing data (ridge): 0.3637200048343915


Mean squared error for training data (ridge): 0.3081824628122753
Mean squared error for testing data (ridge): 0.3648492782196938


Mean squared error for training data (ridge): 0.30961242947070716
Mean squared error for testing data (ridge): 0.36598703424937484


Mean squared error for training data (ridge): 0.3110470732959466
Mean squared error for testing data (ridge): 0.36713262170285915


Mean squared error for training data (ridge): 0.31248580732811687
Mean squared error for testing data (ridge): 0.3682854274744279


Mean squared error for training data (ridge): 0.31392808275044726
Mean squared error for testing data (ridge): 0.3694448739928876
```

```
Mean squared error for training data (ridge): 0.31537338607629656
Mean squared error for testing data (ridge): 0.37061041684969775


Mean squared error for training data (ridge): 0.31682123658580325
Mean squared error for testing data (ridge): 0.3717815426153839


Mean squared error for training data (ridge): 0.3182711839853701
Mean squared error for testing data (ridge): 0.37295776682642806


Mean squared error for training data (ridge): 0.3197228062666523
Mean squared error for testing data (ridge): 0.3741386321268488


Mean squared error for training data (ridge): 0.32117570774464177
Mean squared error for testing data (ridge): 0.3753237065504282


Mean squared error for training data (ridge): 0.32262951725693917
Mean squared error for testing data (ridge): 0.3765125819310488


Mean squared error for training data (ridge): 0.32408388650843273
Mean squared error for testing data (ridge): 0.37770487242992823


Mean squared error for training data (ridge): 0.3255384885474401
Mean squared error for testing data (ridge): 0.3789002131696804


Mean squared error for training data (ridge): 0.3269930163609463
Mean squared error for testing data (ridge): 0.380098258966154


Mean squared error for training data (ridge): 0.32844718157794717
Mean squared error for testing data (ridge): 0.3812986831498834


Mean squared error for training data (ridge): 0.32990071327109766
Mean squared error for testing data (ridge): 0.3825011764697844


Mean squared error for training data (ridge): 0.33135335684790984
Mean squared error for testing data (ridge): 0.3837054460724232
```

```
Mean squared error for training data (ridge): 0.33280487302365264
Mean squared error for testing data (ridge): 0.3849112145508136


Mean squared error for training data (ridge): 0.3342550368689135
Mean squared error for testing data (ridge): 0.38611821905725147


Mean squared error for training data (ridge): 0.3357036369254865
Mean squared error for testing data (ridge): 0.38732621047519983


Mean squared error for training data (ridge): 0.33715047438487444
Mean squared error for testing data (ridge): 0.38853495264567445


Mean squared error for training data (ridge): 0.3385953623242524
Mean squared error for testing data (ridge): 0.38974422164399114


Mean squared error for training data (ridge): 0.3400381249952225
Mean squared error for testing data (ridge): 0.39095380510308714


Mean squared error for training data (ridge): 0.34147859716113804
Mean squared error for testing data (ridge): 0.39216350157996116


Mean squared error for training data (ridge): 0.3429166234791569
Mean squared error for testing data (ridge): 0.39337311996206414


Mean squared error for training data (ridge): 0.34435205792353923
Mean squared error for testing data (ridge): 0.3945824789107431


Mean squared error for training data (ridge): 0.34578476324701435
Mean squared error for testing data (ridge): 0.395791406339075


Mean squared error for training data (ridge): 0.3472146104773235
Mean squared error for testing data (ridge): 0.3969997389216509


Mean squared error for training data (ridge): 0.34864147844629656
Mean squared error for testing data (ridge): 0.3982073216340617
```

```
Mean squared error for training data (ridge): 0.3500652533490513
Mean squared error for testing data (ridge): 0.3994140073200243


Mean squared error for training data (ridge): 0.35148582833110287
Mean squared error for testing data (ridge): 0.4006196562842403


Mean squared error for training data (ridge): 0.35290310310136336
Mean squared error for testing data (ridge): 0.4018241359092386


Mean squared error for training data (ridge): 0.3543169835691722
Mean squared error for testing data (ridge): 0.40302732029458005


Mean squared error for training data (ridge): 0.35572738150365785
Mean squared error for testing data (ridge): 0.40422908991693385


Mean squared error for training data (ridge): 0.35713421421385816
Mean squared error for testing data (ridge): 0.4054293313096412


Mean squared error for training data (ridge): 0.3585374042481641
Mean squared error for testing data (ridge): 0.40662793676049136


Mean squared error for training data (ridge): 0.35993687911175537
Mean squared error for testing data (ridge): 0.40782480402652815


Mean squared error for training data (ridge): 0.36133257100080635
Mean squared error for testing data (ridge): 0.4090198360647893


Mean squared error for training data (ridge): 0.3627244165523323
Mean squared error for testing data (ridge): 0.4102129407779651


Mean squared error for training data (ridge): 0.36411235660863245
Mean squared error for testing data (ridge): 0.4114040307740346


Mean squared error for training data (ridge): 0.3654963359953668
Mean squared error for testing data (ridge): 0.41259302313900115
```

```
Mean squared error for training data (ridge): 0.3668763033123738
Mean squared error for testing data (ridge): 0.41377983922191847


Mean squared error for training data (ridge): 0.36825221073640374
Mean squared error for testing data (ridge): 0.4149644044314494


Mean squared error for training data (ridge): 0.3696240138350016
Mean squared error for testing data (ridge): 0.4161466480432525


Mean squared error for training data (ridge): 0.37099167139083034
Mean squared error for testing data (ridge): 0.4173265030175461


Mean squared error for training data (ridge): 0.3723551452357748
Mean squared error for testing data (ridge): 0.41850390582623337


Mean squared error for training data (ridge): 0.3737144000942155
Mean squared error for testing data (ridge): 0.41967879628902605


Mean squared error for training data (ridge): 0.3750694034349027
Mean squared error for testing data (ridge): 0.42085111741803216


Mean squared error for training data (ridge): 0.37642012533090213
Mean squared error for testing data (ridge): 0.42202081527031304


Mean squared error for training data (ridge): 0.3777665383271196
Mean squared error for testing data (ridge): 0.4231878388079483


Mean squared error for training data (ridge): 0.37910861731494616
Mean squared error for testing data (ridge): 0.424352139765175


Mean squared error for training data (ridge): 0.3804463394135967
Mean squared error for testing data (ridge): 0.42551367252219857


Mean squared error for training data (ridge): 0.38177968385774286
Mean squared error for testing data (ridge): 0.42667239398529394
```

```
Mean squared error for training data (ridge): 0.383108631891068
Mean squared error for testing data (ridge): 0.427828263472847


Mean squared error for training data (ridge): 0.38443316666539884
Mean squared error for testing data (ridge): 0.4289812426070019


Mean squared error for training data (ridge): 0.3857532731450862
Mean squared error for testing data (ridge): 0.43013129521060356


Mean squared error for training data (ridge): 0.38706893801633374
Mean squared error for testing data (ridge): 0.431278387209145


Mean squared error for training data (ridge): 0.38838014960119205
Mean squared error for testing data (ridge): 0.4324224865374459


Mean squared error for training data (ridge): 0.3896868977759479
Mean squared error for testing data (ridge): 0.4335635630508049


Mean squared error for training data (ridge): 0.390989173893666
Mean squared error for testing data (ridge): 0.4347015884403856


Mean squared error for training data (ridge): 0.39228697071064605
Mean squared error for testing data (ridge): 0.4358365361526098


Mean squared error for training data (ridge): 0.39358028231657805
Mean squared error for testing data (ridge): 0.4369683813123436


Mean squared error for training data (ridge): 0.39486910406819187
Mean squared error for testing data (ridge): 0.4380971006496799


Mean squared error for training data (ridge): 0.39615343252620777
Mean squared error for testing data (ridge): 0.4392226724301242


Mean squared error for training data (ridge): 0.39743326539540885
Mean squared error for testing data (ridge): 0.4403450763880112
```

```
Mean squared error for training data (ridge): 0.3987086014676641
Mean squared error for testing data (ridge): 0.4414642936629812


Mean squared error for training data (ridge): 0.39997944056774487
Mean squared error for testing data (ridge): 0.4425803067393616


Mean squared error for training data (ridge): 0.40124578350178386
Mean squared error for testing data (ridge): 0.44369309938830587


Mean squared error for training data (ridge): 0.4025076320082378
Mean squared error for testing data (ridge): 0.44480265661254803


Mean squared error for training data (ridge): 0.40376498871121663
Mean squared error for testing data (ridge): 0.445908964593641


Mean squared error for training data (ridge): 0.4050178570760623
Mean squared error for testing data (ridge): 0.447012010641558


Mean squared error for training data (ridge): 0.40626624136705175
Mean squared error for testing data (ridge): 0.4481117831465345


Mean squared error for training data (ridge): 0.40751014660711915
Mean squared error for testing data (ridge): 0.44920827153304355


Mean squared error for training data (ridge): 0.4087495785394909
Mean squared error for testing data (ridge): 0.450301466215799


Mean squared error for training data (ridge): 0.40998454359113484
Mean squared error for testing data (ridge): 0.4513913585576854


Mean squared error for training data (ridge): 0.4112150488379322
Mean squared error for testing data (ridge): 0.4524779408295252


Mean squared error for training data (ridge): 0.41244110197148465
Mean squared error for testing data (ridge): 0.4535612061715897
```

```
Mean squared error for training data (ridge): 0.41366271126747206
Mean squared error for testing data (ridge): 0.4546411485567745


Mean squared error for training data (ridge): 0.41487988555548594
Mean squared error for testing data (ridge): 0.45571776275535686


Mean squared error for training data (ridge): 0.41609263419026277
Mean squared error for testing data (ridge): 0.456791044301262


Mean squared error for training data (ridge): 0.4173009670242483
Mean squared error for testing data (ridge): 0.45786098945976594


Mean squared error for training data (ridge): 0.4185048943814275
Mean squared error for testing data (ridge): 0.45892759519656856


Mean squared error for training data (ridge): 0.4197044270323569
Mean squared error for testing data (ridge): 0.4599908591481719


Mean squared error for training data (ridge): 0.4208995761703411
Mean squared error for testing data (ridge): 0.46105077959350327


Mean squared error for training data (ridge): 0.4220903533886992
Mean squared error for testing data (ridge): 0.4621073554267281


Mean squared error for training data (ridge): 0.4232767706590644
Mean squared error for testing data (ridge): 0.4631605861311934


Mean squared error for training data (ridge): 0.4244588403106717
Mean squared error for testing data (ridge): 0.46421047175445534


Mean squared error for training data (ridge): 0.42563657501058305
Mean squared error for testing data (ridge): 0.4652570128843375


Mean squared error for training data (ridge): 0.426809987744807
Mean squared error for testing data (ridge): 0.46630021062597604
```

```
Mean squared error for training data (ridge): 0.4279790918002689
Mean squared error for testing data (ridge): 0.46734006657980637


Mean squared error for training data (ridge): 0.4291439007475927
Mean squared error for testing data (ridge): 0.4683765828204489


Mean squared error for training data (ridge): 0.4303044284246553
Mean squared error for testing data (ridge): 0.4694097618764551


Mean squared error for training data (ridge): 0.4314606889208783
Mean squared error for testing data (ridge): 0.47043960671087404


Mean squared error for training data (ridge): 0.4326126965622211
Mean squared error for testing data (ridge): 0.4714661207026056


Mean squared error for training data (ridge): 0.433760465896844
Mean squared error for testing data (ridge): 0.47248930762850333


Mean squared error for training data (ridge): 0.43490401168141124
Mean squared error for testing data (ridge): 0.47350917164619666


Mean squared error for training data (ridge): 0.43604334886800133
Mean squared error for testing data (ridge): 0.4745257172775997


Mean squared error for training data (ridge): 0.43717849259160013
Mean squared error for testing data (ridge): 0.47553894939307695


Mean squared error for training data (ridge): 0.4383094581581486
Mean squared error for testing data (ridge): 0.47654887319623934


Mean squared error for training data (ridge): 0.43943626103311995
Mean squared error for testing data (ridge): 0.4775554942093418


Mean squared error for training data (ridge): 0.44055891683060255
Mean squared error for testing data (ridge): 0.47855881825925684
```

```
Mean squared error for training data (ridge): 0.44167744130286535
Mean squared error for testing data (ridge): 0.4795588514640021


Mean squared error for training data (ridge): 0.44279185033038493
Mean squared error for testing data (ridge): 0.4805556002197948


Mean squared error for training data (ridge): 0.44390215991231263
Mean squared error for testing data (ridge): 0.4815490711886148


Mean squared error for training data (ridge): 0.44500838615736255
Mean squared error for testing data (ridge): 0.48253927128625096


Mean squared error for training data (ridge): 0.44611054527510174
Mean squared error for testing data (ridge): 0.48352620767081483


Mean squared error for training data (ridge): 0.4472086535676248
Mean squared error for testing data (ridge): 0.48450988773169945


Mean squared error for training data (ridge): 0.4483027274215954
Mean squared error for testing data (ridge): 0.4854903190789653


Mean squared error for training data (ridge): 0.44939278330063986
Mean squared error for testing data (ridge): 0.4864675095331373


Mean squared error for training data (ridge): 0.4504788377380756
Mean squared error for testing data (ridge): 0.4874414671153946


Mean squared error for training data (ridge): 0.4515609073299609
Mean squared error for testing data (ridge): 0.48841220003813685


Mean squared error for training data (ridge): 0.45263900872845286
Mean squared error for testing data (ridge): 0.48937971669591385


Mean squared error for training data (ridge): 0.4537131586354573
Mean squared error for testing data (ridge): 0.4903440256567011
```

```
Mean squared error for training data (ridge): 0.45478337379656214
Mean squared error for testing data (ridge): 0.4913051356535101


Mean squared error for training data (ridge): 0.4558496709952377
Mean squared error for testing data (ridge): 0.49226305557631667


Mean squared error for training data (ridge): 0.4569120670472951
Mean squared error for testing data (ridge): 0.49321779446429803


Mean squared error for training data (ridge): 0.457970578795592
Mean squared error for testing data (ridge): 0.4941693614983651


Mean squared error for training data (ridge): 0.4590252231049729
Mean squared error for testing data (ridge): 0.49511776599397544


Mean squared error for training data (ridge): 0.4600760168574361
Mean squared error for testing data (ridge): 0.496063017394222


Mean squared error for training data (ridge): 0.4611229769475166
Mean squared error for testing data (ridge): 0.4970051252631801


Mean squared error for training data (ridge): 0.4621661202778759
Mean squared error for testing data (ridge): 0.4979440992795071


Mean squared error for training data (ridge): 0.4632054637550905
Mean squared error for testing data (ridge): 0.49887994923028317


Mean squared error for training data (ridge): 0.46424102428563035
Mean squared error for testing data (ridge): 0.4998126850050841


Mean squared error for training data (ridge): 0.46527281877201887
Mean squared error for testing data (ridge): 0.5007423165902761


Mean squared error for training data (ridge): 0.46630086410916716
Mean squared error for testing data (ridge): 0.501668854063526
```

```
Mean squared error for training data (ridge): 0.4673251771808763
Mean squared error for testing data (ridge): 0.5025923075885175


Mean squared error for training data (ridge): 0.468345774856499
Mean squared error for testing data (ridge): 0.5035126874098653


Mean squared error for training data (ridge): 0.4693626739877539
Mean squared error for testing data (ridge): 0.5044300038482191


Mean squared error for training data (ridge): 0.47037589140568853
Mean squared error for testing data (ridge): 0.5053442672955522


Mean squared error for training data (ridge): 0.4713854439177819
Mean squared error for testing data (ridge): 0.5062554882106258


Mean squared error for training data (ridge): 0.4723913483051827
Mean squared error for testing data (ridge): 0.5071636771146234


Mean squared error for training data (ridge): 0.4733936213200768
Mean squared error for testing data (ridge): 0.5080688445869475


Mean squared error for training data (ridge): 0.4743922796831792
Mean squared error for testing data (ridge): 0.508971001261175


Mean squared error for training data (ridge): 0.47538734008134487
Mean squared error for testing data (ridge): 0.5098701578211624


Mean squared error for training data (ridge): 0.47637881916529357
Mean squared error for testing data (ridge): 0.5107663249972978


Mean squared error for training data (ridge): 0.4773667335474443
Mean squared error for testing data (ridge): 0.5116595135628917


Mean squared error for training data (ridge): 0.47835109979985474
Mean squared error for testing data (ridge): 0.5125497343307043
```

```
Mean squared error for training data (ridge): 0.47933193445226185
Mean squared error for testing data (ridge): 0.5134369981496021


Mean squared error for training data (ridge): 0.480309253990218
Mean squared error for testing data (ridge): 0.514321315901339


Mean squared error for training data (ridge): 0.4812830748533204
Mean squared error for testing data (ridge): 0.5152026984974593


Mean squared error for training data (ridge): 0.4822534134335298
Mean squared error for testing data (ridge): 0.5160811568763155


Mean squared error for training data (ridge): 0.4832202860735734
Mean squared error for testing data (ridge): 0.516956702000199


Mean squared error for training data (ridge): 0.48418370906543046
Mean squared error for testing data (ridge): 0.5178293448525787


Mean squared error for training data (ridge): 0.48514369864889634
Mean squared error for testing data (ridge): 0.5186990964354418


Mean squared error for training data (ridge): 0.48610027101022035
Mean squared error for testing data (ridge): 0.5195659677667366


Mean squared error for training data (ridge): 0.4870534422808184
Mean squared error for testing data (ridge): 0.5204299698779113


Mean squared error for training data (ridge): 0.48800322853605294
Mean squared error for testing data (ridge): 0.5212911138115454


Mean squared error for training data (ridge): 0.4889496457940795
Mean squared error for testing data (ridge): 0.5221494106190699


Mean squared error for training data (ridge): 0.4898927100147583
Mean squared error for testing data (ridge): 0.5230048713585765
```

```
Mean squared error for training data (ridge): 0.4908324370986258
Mean squared error for testing data (ridge): 0.5238575070927073


Mean squared error for training data (ridge): 0.49176884288592587
Mean squared error for testing data (ridge): 0.5247073288866263


Mean squared error for training data (ridge): 0.4927019431556968
Mean squared error for testing data (ridge): 0.525554347806068


Mean squared error for training data (ridge): 0.49363175362491224
Mean squared error for testing data (ridge): 0.5263985749154613


Mean squared error for training data (ridge): 0.49455828994767514
Mean squared error for testing data (ridge): 0.5272400212761242


Mean squared error for training data (ridge): 0.49548156771445967
Mean squared error for testing data (ridge): 0.5280786979445291


Mean squared error for training data (ridge): 0.49640160245140297
Mean squared error for testing data (ridge): 0.528914615970634


Mean squared error for training data (ridge): 0.4973184096196413
Mean squared error for testing data (ridge): 0.5297477863962806


Mean squared error for training data (ridge): 0.4982320046146906
Mean squared error for testing data (ridge): 0.5305782202536515


Mean squared error for training data (ridge): 0.4991424027658693
Mean squared error for testing data (ridge): 0.5314059285637903


Mean squared error for training data (ridge): 0.5000496193357623
Mean squared error for testing data (ridge): 0.5322309223351785


Mean squared error for training data (ridge): 0.5009536695197223
Mean squared error for testing data (ridge): 0.5330532125623691
```

```
Mean squared error for training data (ridge): 0.5018545684454099
Mean squared error for testing data (ridge): 0.5338728102246733


Mean squared error for training data (ridge): 0.5027523311723683
Mean squared error for testing data (ridge): 0.5346897262848999


Mean squared error for training data (ridge): 0.5036469726916338
Mean squared error for testing data (ridge): 0.5355039716881463


Mean squared error for training data (ridge): 0.5045385079253774
Mean squared error for testing data (ridge): 0.5363155573606356


Mean squared error for training data (ridge): 0.5054269517265793
Mean squared error for testing data (ridge): 0.5371244942086024


Mean squared error for training data (ridge): 0.5063123188787334
Mean squared error for testing data (ridge): 0.537930793117224


Mean squared error for training data (ridge): 0.5071946240955804
Mean squared error for testing data (ridge): 0.5387344649495935


Mean squared error for training data (ridge): 0.5080738820208691
Mean squared error for testing data (ridge): 0.5395355205457384


Mean squared error for training data (ridge): 0.5089501072281446
Mean squared error for testing data (ridge): 0.5403339707216755


Mean squared error for training data (ridge): 0.5098233142205617
Mean squared error for testing data (ridge): 0.5411298262685093


Mean squared error for training data (ridge): 0.510693517430723
Mean squared error for testing data (ridge): 0.5419230979515659


Mean squared error for training data (ridge): 0.5115607312205411
Mean squared error for testing data (ridge): 0.5427137965095649
```

```
Mean squared error for training data (ridge): 0.5124249698811213
Mean squared error for testing data (ridge): 0.5435019326538252


Mean squared error for training data (ridge): 0.51328624763267
Mean squared error for testing data (ridge): 0.544287517067507


Mean squared error for training data (ridge): 0.5141445786244179
Mean squared error for testing data (ridge): 0.5450705604048847


Mean squared error for training data (ridge): 0.5149999769345697
Mean squared error for testing data (ridge): 0.5458510732906541


Mean squared error for training data (ridge): 0.5158524565702673
Mean squared error for testing data (ridge): 0.5466290663192668


Mean squared error for training data (ridge): 0.5167020314675743
Mean squared error for testing data (ridge): 0.5474045500542994


Mean squared error for training data (ridge): 0.5175487154914766
Mean squared error for testing data (ridge): 0.5481775350278467


Mean squared error for training data (ridge): 0.5183925224359007
Mean squared error for testing data (ridge): 0.5489480317399444


Mean squared error for training data (ridge): 0.5192334660237464
Mean squared error for testing data (ridge): 0.5497160506580201


Mean squared error for training data (ridge): 0.5200715599069374
Mean squared error for testing data (ridge): 0.5504816022163679


Mean squared error for training data (ridge): 0.5209068176664827
Mean squared error for testing data (ridge): 0.5512446968156488


Mean squared error for training data (ridge): 0.5217392528125566
Mean squared error for testing data (ridge): 0.5520053448224158
```

```
Mean squared error for training data (ridge): 0.5225688787845879
Mean squared error for testing data (ridge): 0.5527635565686617


Mean squared error for training data (ridge): 0.5233957089513644
Mean squared error for testing data (ridge): 0.5535193423513894


Mean squared error for training data (ridge): 0.5242197566111488
Mean squared error for testing data (ridge): 0.5542727124322044


Mean squared error for training data (ridge): 0.5250410349918065
Mean squared error for testing data (ridge): 0.555023677036928


Mean squared error for training data (ridge): 0.5258595572509442
Mean squared error for testing data (ridge): 0.5557722463552301


Mean squared error for training data (ridge): 0.5266753364760597
Mean squared error for testing data (ridge): 0.5565184305402837


Mean squared error for training data (ridge): 0.5274883856847015
Mean squared error for testing data (ridge): 0.557262239708436


Mean squared error for training data (ridge): 0.5282987178246379
Mean squared error for testing data (ridge): 0.5580036839388992


Mean squared error for training data (ridge): 0.5291063457740358
Mean squared error for testing data (ridge): 0.5587427732734576


Mean squared error for training data (ridge): 0.5299112823416487
Mean squared error for testing data (ridge): 0.559479517716194


Mean squared error for training data (ridge): 0.5307135402670117
Mean squared error for testing data (ridge): 0.5602139272332303


Mean squared error for training data (ridge): 0.531513132220646
Mean squared error for testing data (ridge): 0.5609460117524854
```

```
Mean squared error for training data (ridge): 0.5323100708042698
Mean squared error for testing data (ridge): 0.5616757811634479


Mean squared error for training data (ridge): 0.5331043685510178
Mean squared error for testing data (ridge): 0.5624032453169634


Mean squared error for training data (ridge): 0.5338960379256669
Mean squared error for testing data (ridge): 0.5631284140250371


Mean squared error for training data (ridge): 0.534685091324867
Mean squared error for testing data (ridge): 0.563851297060649


Mean squared error for training data (ridge): 0.5354715410773814
Mean squared error for testing data (ridge): 0.5645719041575833


Mean squared error for training data (ridge): 0.5362553994443299
Mean squared error for testing data (ridge): 0.5652902450102713


Mean squared error for training data (ridge): 0.5370366786194395
Mean squared error for testing data (ridge): 0.5660063292736444


Mean squared error for training data (ridge): 0.5378153907292985
Mean squared error for testing data (ridge): 0.5667201665630017


Mean squared error for training data (ridge): 0.5385915478336185
Mean squared error for testing data (ridge): 0.567431766453888


Mean squared error for training data (ridge): 0.5393651619254981
Mean squared error for testing data (ridge): 0.5681411384819832


Mean squared error for training data (ridge): 0.540136244931693
Mean squared error for testing data (ridge): 0.5688482921430023


Mean squared error for training data (ridge): 0.5409048087128896
Mean squared error for testing data (ridge): 0.5695532368926062
```

```
Mean squared error for training data (ridge): 0.5416708650639831
Mean squared error for testing data (ridge): 0.5702559821463223


Mean squared error for training data (ridge): 0.5424344257143588
Mean squared error for testing data (ridge): 0.5709565372794755


Mean squared error for training data (ridge): 0.5431955023281778
Mean squared error for testing data (ridge): 0.5716549116271272


Mean squared error for training data (ridge): 0.5439541065046639
Mean squared error for testing data (ridge): 0.5723511144840249


Mean squared error for training data (ridge): 0.5447102497783979
Mean squared error for testing data (ridge): 0.5730451551045589


Mean squared error for training data (ridge): 0.5454639436196086
Mean squared error for testing data (ridge): 0.5737370427027301


Mean squared error for training data (ridge): 0.546215199434473
Mean squared error for testing data (ridge): 0.5744267864521229


Mean squared error for training data (ridge): 0.5469640285654149
Mean squared error for testing data (ridge): 0.5751143954858869


Mean squared error for training data (ridge): 0.5477104422914068
Mean squared error for testing data (ridge): 0.5757998788967278


Mean squared error for training data (ridge): 0.5484544518282741
Mean squared error for testing data (ridge): 0.5764832457369032


Mean squared error for training data (ridge): 0.549196068329003
Mean squared error for testing data (ridge): 0.5771645050182279


Mean squared error for training data (ridge): 0.5499353028840468
Mean squared error for testing data (ridge): 0.5778436657120826
```

```
Mean squared error for training data (ridge): 0.5506721665216369
Mean squared error for testing data (ridge): 0.578520736749433


Mean squared error for training data (ridge): 0.5514066702080939
Mean squared error for testing data (ridge): 0.579195727020853


Mean squared error for training data (ridge): 0.5521388248481417
Mean squared error for testing data (ridge): 0.5798686453765537


Mean squared error for training data (ridge): 0.5528686412852202
Mean squared error for testing data (ridge): 0.5805395006264191


Mean squared error for training data (ridge): 0.5535961303018018
Mean squared error for testing data (ridge): 0.5812083015400478


Mean squared error for training data (ridge): 0.5543213026197077
Mean squared error for testing data (ridge): 0.5818750568467989


Mean squared error for training data (ridge): 0.5550441689004251
Mean squared error for testing data (ridge): 0.5825397752358439


Mean squared error for training data (ridge): 0.5557647397454261
Mean squared error for testing data (ridge): 0.5832024653562238


Mean squared error for training data (ridge): 0.5564830256964858
Mean squared error for testing data (ridge): 0.5838631358169106


Mean squared error for training data (ridge): 0.5571990372360027
Mean squared error for testing data (ridge): 0.5845217951868737


Mean squared error for training data (ridge): 0.5579127847873183
Mean squared error for testing data (ridge): 0.5851784519951501


Mean squared error for training data (ridge): 0.558624278715038
Mean squared error for testing data (ridge): 0.5858331147309205
```

```
Mean squared error for training data (ridge): 0.5593335293253509
Mean squared error for testing data (ridge): 0.5864857918435875


Mean squared error for training data (ridge): 0.5600405468663519
Mean squared error for testing data (ridge): 0.5871364917428594


Mean squared error for training data (ridge): 0.5607453415283628
Mean squared error for testing data (ridge): 0.5877852227988378


Mean squared error for training data (ridge): 0.5614479234442524
Mean squared error for testing data (ridge): 0.5884319933421068


Mean squared error for training data (ridge): 0.5621483026897588
Mean squared error for testing data (ridge): 0.5890768116638285


Mean squared error for training data (ridge): 0.5628464892838099
Mean squared error for testing data (ridge): 0.5897196860158405


Mean squared error for training data (ridge): 0.5635424931888443
Mean squared error for testing data (ridge): 0.5903606246107564


Mean squared error for training data (ridge): 0.5642363243111322
Mean squared error for testing data (ridge): 0.5909996356220701


Mean squared error for training data (ridge): 0.5649279925010948
Mean squared error for testing data (ridge): 0.5916367271842626


Mean squared error for training data (ridge): 0.5656175075536257
Mean squared error for testing data (ridge): 0.5922719073929122


Mean squared error for training data (ridge): 0.5663048792084091
Mean squared error for testing data (ridge): 0.5929051843048077


Mean squared error for training data (ridge): 0.5669901171502391
Mean squared error for testing data (ridge): 0.5935365659380615
```

```
Mean squared error for training data (ridge): 0.5676732310093384
Mean squared error for testing data (ridge): 0.5941660602722302


Mean squared error for training data (ridge): 0.5683542303616758
Mean squared error for testing data (ridge): 0.5947936752484327


Mean squared error for training data (ridge): 0.5690331247292834
Mean squared error for testing data (ridge): 0.5954194187694742


Mean squared error for training data (ridge): 0.5697099235805726
Mean squared error for testing data (ridge): 0.5960432986999701


Mean squared error for training data (ridge): 0.5703846363306512
Mean squared error for testing data (ridge): 0.5966653228664727


Mean squared error for training data (ridge): 0.5710572723416361
Mean squared error for testing data (ridge): 0.5972854990576003


Mean squared error for training data (ridge): 0.5717278409229687
Mean squared error for testing data (ridge): 0.5979038350241687


Mean squared error for training data (ridge): 0.5723963513317281
Mean squared error for testing data (ridge): 0.598520338479323


Mean squared error for training data (ridge): 0.5730628127729424
Mean squared error for testing data (ridge): 0.5991350170986711


Mean squared error for training data (ridge): 0.5737272343999005
Mean squared error for testing data (ridge): 0.599747878520422


Mean squared error for training data (ridge): 0.5743896253144617
Mean squared error for testing data (ridge): 0.6003589303455216


Mean squared error for training data (ridge): 0.5750499945673659
Mean squared error for testing data (ridge): 0.6009681801377917
```

```
Mean squared error for training data (ridge): 0.575708351158541
Mean squared error for testing data (ridge): 0.6015756354240717


Mean squared error for training data (ridge): 0.5763647040374097
Mean squared error for testing data (ridge): 0.6021813036943595


Mean squared error for training data (ridge): 0.5770190621031951
Mean squared error for testing data (ridge): 0.6027851924019553


Mean squared error for training data (ridge): 0.5776714342052272
Mean squared error for testing data (ridge): 0.6033873089636047


Mean squared error for training data (ridge): 0.5783218291432437
Mean squared error for testing data (ridge): 0.603987660759646


Mean squared error for training data (ridge): 0.5789702556676946
Mean squared error for testing data (ridge): 0.6045862551341551


Mean squared error for training data (ridge): 0.579616722480042
Mean squared error for testing data (ridge): 0.6051830993950937


Mean squared error for training data (ridge): 0.5802612382330605
Mean squared error for testing data (ridge): 0.6057782008144583


Mean squared error for training data (ridge): 0.5809038115311355
Mean squared error for testing data (ridge): 0.606371566628429


Mean squared error for training data (ridge): 0.5815444509305597
Mean squared error for testing data (ridge): 0.6069632040375192


Mean squared error for training data (ridge): 0.5821831649398305
Mean squared error for testing data (ridge): 0.6075531202067284


Mean squared error for training data (ridge): 0.5828199620199439
Mean squared error for testing data (ridge): 0.6081413222656917
```

```
Mean squared error for training data (ridge): 0.5834548505846869
Mean squared error for testing data (ridge): 0.6087278173088343


Mean squared error for training data (ridge): 0.5840878390009309
Mean squared error for testing data (ridge): 0.6093126123955233


Mean squared error for training data (ridge): 0.5847189355889207
Mean squared error for testing data (ridge): 0.6098957145502215


Mean squared error for training data (ridge): 0.585348148622565
Mean squared error for testing data (ridge): 0.6104771307626419


Mean squared error for training data (ridge): 0.5859754863297225
Mean squared error for testing data (ridge): 0.6110568679879018


Mean squared error for training data (ridge): 0.5866009568924894
Mean squared error for testing data (ridge): 0.6116349331466793


Mean squared error for training data (ridge): 0.5872245684474839
Mean squared error for testing data (ridge): 0.6122113331253664


Mean squared error for training data (ridge): 0.5878463290861293
Mean squared error for testing data (ridge): 0.612786074776227


Mean squared error for training data (ridge): 0.5884662468549356
Mean squared error for testing data (ridge): 0.6133591649175524


Mean squared error for training data (ridge): 0.5890843297557814
Mean squared error for testing data (ridge): 0.6139306103338174


Mean squared error for training data (ridge): 0.589700585746191
Mean squared error for testing data (ridge): 0.6145004177758373


Mean squared error for training data (ridge): 0.5903150227396132
Mean squared error for testing data (ridge): 0.6150685939609244
```

```
Mean squared error for training data (ridge): 0.590927648605697
Mean squared error for testing data (ridge): 0.6156351455730463


Mean squared error for training data (ridge): 0.5915384711705657
Mean squared error for testing data (ridge): 0.6162000792629818


Mean squared error for training data (ridge): 0.5921474982170906
Mean squared error for testing data (ridge): 0.6167634016484784


Mean squared error for training data (ridge): 0.5927547374851616
Mean squared error for testing data (ridge): 0.6173251193144111


Mean squared error for training data (ridge): 0.5933601966719586
Mean squared error for testing data (ridge): 0.6178852388129376


Mean squared error for training data (ridge): 0.5939638834322187
Mean squared error for testing data (ridge): 0.6184437666636579


Mean squared error for training data (ridge): 0.5945658053785039
Mean squared error for testing data (ridge): 0.6190007093537707


Mean squared error for training data (ridge): 0.5951659700814662
Mean squared error for testing data (ridge): 0.6195560733382306


Mean squared error for training data (ridge): 0.5957643850701115
Mean squared error for testing data (ridge): 0.6201098650399067


Mean squared error for training data (ridge): 0.5963610578320627
Mean squared error for testing data (ridge): 0.6206620908497386


Mean squared error for training data (ridge): 0.59695599581382
Mean squared error for testing data (ridge): 0.6212127571268944


Mean squared error for training data (ridge): 0.59754920642102
Mean squared error for testing data (ridge): 0.6217618701989271
```

```
Mean squared error for training data (ridge): 0.5981406970186947
Mean squared error for testing data (ridge): 0.6223094363619325


Mean squared error for training data (ridge): 0.5987304749315265
Mean squared error for testing data (ridge): 0.6228554618807048


Mean squared error for training data (ridge): 0.5993185474441038
Mean squared error for testing data (ridge): 0.6233999529888938


Mean squared error for training data (ridge): 0.5999049218011739
Mean squared error for testing data (ridge): 0.6239429158891608


Mean squared error for training data (ridge): 0.6004896052078952
Mean squared error for testing data (ridge): 0.6244843567533354


Mean squared error for training data (ridge): 0.6010726048300872
Mean squared error for testing data (ridge): 0.62502428172257


Mean squared error for training data (ridge): 0.6016539277944785
Mean squared error for testing data (ridge): 0.6255626969074967


Mean squared error for training data (ridge): 0.6022335811889554
Mean squared error for testing data (ridge): 0.6260996083883817


Mean squared error for training data (ridge): 0.6028115720628066
Mean squared error for testing data (ridge): 0.6266350222152804


Mean squared error for training data (ridge): 0.6033879074269675
Mean squared error for testing data (ridge): 0.6271689444081923


Mean squared error for training data (ridge): 0.6039625942542622
Mean squared error for testing data (ridge): 0.6277013809572142


Mean squared error for training data (ridge): 0.604535639479647
Mean squared error for testing data (ridge): 0.6282323378226958
```

```
Mean squared error for training data (ridge): 0.6051070500004466
Mean squared error for testing data (ridge): 0.6287618209353916


Mean squared error for training data (ridge): 0.6056768326765948
Mean squared error for testing data (ridge): 0.6292898361966147


Mean squared error for training data (ridge): 0.6062449943308704
Mean squared error for testing data (ridge): 0.6298163894783904


Mean squared error for training data (ridge): 0.6068115417491318
Mean squared error for testing data (ridge): 0.630341486623606


Mean squared error for training data (ridge): 0.6073764816805514
Mean squared error for testing data (ridge): 0.6308651334461657


Mean squared error for training data (ridge): 0.6079398208378468
Mean squared error for testing data (ridge): 0.6313873357311401


Mean squared error for training data (ridge): 0.608501565897513
Mean squared error for testing data (ridge): 0.6319080992349179


Mean squared error for training data (ridge): 0.60906172350005
Mean squared error for testing data (ridge): 0.6324274296853558


Mean squared error for training data (ridge): 0.6096203002501911
Mean squared error for testing data (ridge): 0.6329453327819297


Mean squared error for training data (ridge): 0.6101773027171292
Mean squared error for testing data (ridge): 0.6334618141958835


Mean squared error for training data (ridge): 0.610732737434742
Mean squared error for testing data (ridge): 0.633976879570378


Mean squared error for training data (ridge): 0.6112866109018148
Mean squared error for testing data (ridge): 0.6344905345206404
```

```
Mean squared error for training data (ridge): 0.6118389295822619
Mean squared error for testing data (ridge): 0.6350027846341112


Mean squared error for training data (ridge): 0.6123896999053482
Mean squared error for testing data (ridge): 0.6355136354705926


Mean squared error for training data (ridge): 0.612938928265907
Mean squared error for testing data (ridge): 0.6360230925623959


Mean squared error for training data (ridge): 0.6134866210245591
Mean squared error for testing data (ridge): 0.6365311614144868


Mean squared error for training data (ridge): 0.6140327845079269
Mean squared error for testing data (ridge): 0.6370378475046325


Mean squared error for training data (ridge): 0.6145774250088509
Mean squared error for testing data (ridge): 0.6375431562835462


Mean squared error for training data (ridge): 0.6151205487866018
Mean squared error for testing data (ridge): 0.6380470931750332


Mean squared error for training data (ridge): 0.6156621620670929
Mean squared error for testing data (ridge): 0.6385496635761336


Mean squared error for training data (ridge): 0.6162022710430906
Mean squared error for testing data (ridge): 0.6390508728572678


Mean squared error for training data (ridge): 0.6167408818744233
Mean squared error for testing data (ridge): 0.6395507263623783


Mean squared error for training data (ridge): 0.6172780006881892
Mean squared error for testing data (ridge): 0.640049229409073


Mean squared error for training data (ridge): 0.6178136335789625
Mean squared error for testing data (ridge): 0.6405463872887673
```

```
Mean squared error for training data (ridge): 0.618347786608998
Mean squared error for testing data (ridge): 0.6410422052668246


Mean squared error for training data (ridge): 0.6188804658084359
Mean squared error for testing data (ridge): 0.6415366885826987


Mean squared error for training data (ridge): 0.6194116771755022
Mean squared error for testing data (ridge): 0.6420298424500726


Mean squared error for training data (ridge): 0.6199414266767108
Mean squared error for testing data (ridge): 0.6425216720569991


Mean squared error for training data (ridge): 0.6204697202470629
Mean squared error for testing data (ridge): 0.6430121825660391


Mean squared error for training data (ridge): 0.6209965637902446
Mean squared error for testing data (ridge): 0.6435013791144008


Mean squared error for training data (ridge): 0.6215219631788236
Mean squared error for testing data (ridge): 0.6439892668140772


Mean squared error for training data (ridge): 0.622045924254446
Mean squared error for testing data (ridge): 0.644475850751983


Mean squared error for training data (ridge): 0.6225684528280288
Mean squared error for testing data (ridge): 0.6449611359900916


Mean squared error for training data (ridge): 0.6230895546799536
Mean squared error for testing data (ridge): 0.645445127565571


Mean squared error for training data (ridge): 0.6236092355602582
Mean squared error for testing data (ridge): 0.6459278304909193


Mean squared error for training data (ridge): 0.6241275011888268
Mean squared error for testing data (ridge): 0.6464092497540986
```

```
Mean squared error for training data (ridge): 0.6246443572555783
Mean squared error for testing data (ridge): 0.6468893903186708


Mean squared error for training data (ridge): 0.625159809420655
Mean squared error for testing data (ridge): 0.6473682571239289


Mean squared error for training data (ridge): 0.6256738633146081
Mean squared error for testing data (ridge): 0.6478458550850319


Mean squared error for training data (ridge): 0.6261865245385837
Mean squared error for testing data (ridge): 0.6483221890931354


Mean squared error for training data (ridge): 0.6266977986645056
Mean squared error for testing data (ridge): 0.6487972640155242


Mean squared error for training data (ridge): 0.6272076912352592
Mean squared error for testing data (ridge): 0.6492710846957431


Mean squared error for training data (ridge): 0.6277162077648717
Mean squared error for testing data (ridge): 0.649743655953727


Mean squared error for training data (ridge): 0.6282233537386923
Mean squared error for testing data (ridge): 0.6502149825859299


Mean squared error for training data (ridge): 0.6287291346135726
Mean squared error for testing data (ridge): 0.6506850693654558


Mean squared error for training data (ridge): 0.629233555818042
Mean squared error for testing data (ridge): 0.6511539210421848


Mean squared error for training data (ridge): 0.629736622752486
Mean squared error for testing data (ridge): 0.6516215423429029


Mean squared error for training data (ridge): 0.6302383407893206
Mean squared error for testing data (ridge): 0.6520879379714273
```

```
Mean squared error for training data (ridge): 0.6307387152731666
Mean squared error for testing data (ridge): 0.6525531126087347


Mean squared error for training data (ridge): 0.631237751521022
Mean squared error for testing data (ridge): 0.6530170709130859


Mean squared error for training data (ridge): 0.6317354548224343
Mean squared error for testing data (ridge): 0.6534798175201509


Mean squared error for training data (ridge): 0.6322318304396709
Mean squared error for testing data (ridge): 0.6539413570431344


Mean squared error for training data (ridge): 0.6327268836078879
Mean squared error for testing data (ridge): 0.6544016940728988


Mean squared error for training data (ridge): 0.6332206195352985
Mean squared error for testing data (ridge): 0.6548608331780881


Mean squared error for training data (ridge): 0.6337130434033406
Mean squared error for testing data (ridge): 0.6553187789052501


Mean squared error for training data (ridge): 0.6342041603668415
Mean squared error for testing data (ridge): 0.6557755357789592


Mean squared error for training data (ridge): 0.6346939755541837
Mean squared error for testing data (ridge): 0.6562311083019363


Mean squared error for training data (ridge): 0.635182494067468
Mean squared error for testing data (ridge): 0.6566855009551706


Mean squared error for training data (ridge): 0.6356697209826757
Mean squared error for testing data (ridge): 0.6571387181980399


Mean squared error for training data (ridge): 0.6361556613498298
Mean squared error for testing data (ridge): 0.6575907644684288
```

```
Mean squared error for training data (ridge): 0.636640320193157
Mean squared error for testing data (ridge): 0.6580416441828494


Mean squared error for training data (ridge): 0.637123702511244
Mean squared error for testing data (ridge): 0.6584913617365574


Mean squared error for training data (ridge): 0.6376058132771976
Mean squared error for testing data (ridge): 0.658939921503671


Mean squared error for training data (ridge): 0.6380866574388006
Mean squared error for testing data (ridge): 0.6593873278372875


Mean squared error for training data (ridge): 0.6385662399186683
Mean squared error for testing data (ridge): 0.6598335850695991


Mean squared error for training data (ridge): 0.6390445656144031
Mean squared error for testing data (ridge): 0.6602786975120096


Mean squared error for training data (ridge): 0.6395216393987477
Mean squared error for testing data (ridge): 0.6607226694552484


Mean squared error for training data (ridge): 0.6399974661197384
Mean squared error for testing data (ridge): 0.6611655051694852


Mean squared error for training data (ridge): 0.6404720506008564
Mean squared error for testing data (ridge): 0.6616072089044442


Mean squared error for training data (ridge): 0.640945397641178
Mean squared error for testing data (ridge): 0.6620477848895164


Mean squared error for training data (ridge): 0.6414175120155248
Mean squared error for testing data (ridge): 0.6624872373338729


Mean squared error for training data (ridge): 0.6418883984746119
Mean squared error for testing data (ridge): 0.6629255704265765
```

```
Mean squared error for training data (ridge): 0.642358061745195
Mean squared error for testing data (ridge): 0.6633627883366927


Mean squared error for training data (ridge): 0.6428265065302173
Mean squared error for testing data (ridge): 0.6637988952134012


Mean squared error for training data (ridge): 0.6432937375089548
Mean squared error for testing data (ridge): 0.6642338951861045


Mean squared error for training data (ridge): 0.6437597593371607
Mean squared error for testing data (ridge): 0.6646677923645393


Mean squared error for training data (ridge): 0.644224576647208
Mean squared error for testing data (ridge): 0.6651005908388834


Mean squared error for training data (ridge): 0.644688194048234
Mean squared error for testing data (ridge): 0.6655322946798651


Mean squared error for training data (ridge): 0.6451506161262793
Mean squared error for testing data (ridge): 0.6659629079388706


Mean squared error for training data (ridge): 0.6456118474444303
Mean squared error for testing data (ridge): 0.6663924346480505


Mean squared error for training data (ridge): 0.6460718925429574
Mean squared error for testing data (ridge): 0.6668208788204268


Mean squared error for training data (ridge): 0.6465307559394539
Mean squared error for testing data (ridge): 0.6672482444499986


Mean squared error for training data (ridge): 0.6469884421289738
Mean squared error for testing data (ridge): 0.6676745355118465


Mean squared error for training data (ridge): 0.6474449555841684
Mean squared error for testing data (ridge): 0.6680997559622386
```

```
Mean squared error for training data (ridge): 0.6479003007554219
Mean squared error for testing data (ridge): 0.6685239097387333


Mean squared error for training data (ridge): 0.648354482070986
Mean squared error for testing data (ridge): 0.6689470007602831


Mean squared error for training data (ridge): 0.6488075039371141
Mean squared error for testing data (ridge): 0.6693690329273382


Mean squared error for training data (ridge): 0.6492593707381937
Mean squared error for testing data (ridge): 0.6697900101219466


Mean squared error for training data (ridge): 0.6497100868368794
Mean squared error for testing data (ridge): 0.6702099362078585


Mean squared error for training data (ridge): 0.6501596565742223
Mean squared error for testing data (ridge): 0.6706288150306254


Mean squared error for training data (ridge): 0.650608084269802
Mean squared error for testing data (ridge): 0.6710466504177011


Mean squared error for training data (ridge): 0.6510553742218553
Mean squared error for testing data (ridge): 0.6714634461785418


Mean squared error for training data (ridge): 0.6515015307074031
Mean squared error for testing data (ridge): 0.6718792061047053


Mean squared error for training data (ridge): 0.6519465579823807
Mean squared error for testing data (ridge): 0.6722939339699499


Mean squared error for training data (ridge): 0.6523904602817621
Mean squared error for testing data (ridge): 0.6727076335303325


Mean squared error for training data (ridge): 0.6528332418196868
Mean squared error for testing data (ridge): 0.6731203085243054
```

```
Mean squared error for training data (ridge): 0.6532749067895847
Mean squared error for testing data (ridge): 0.6735319626728149


Mean squared error for training data (ridge): 0.6537154593642996
Mean squared error for testing data (ridge): 0.673942599679397


Mean squared error for training data (ridge): 0.6541549036962133
Mean squared error for testing data (ridge): 0.6743522232302722


Mean squared error for training data (ridge): 0.6545932439173671
Mean squared error for testing data (ridge): 0.6747608369944434


Mean squared error for training data (ridge): 0.6550304841395838
Mean squared error for testing data (ridge): 0.6751684446237882


Mean squared error for training data (ridge): 0.6554666284545881
Mean squared error for testing data (ridge): 0.6755750497531551


Mean squared error for training data (ridge): 0.6559016809341269
Mean squared error for testing data (ridge): 0.6759806560004551


Mean squared error for training data (ridge): 0.6563356456300878
Mean squared error for testing data (ridge): 0.6763852669667566


Mean squared error for training data (ridge): 0.6567685265746175
Mean squared error for testing data (ridge): 0.6767888862363781


Mean squared error for training data (ridge): 0.6572003277802394
Mean squared error for testing data (ridge): 0.6771915173769785


Mean squared error for training data (ridge): 0.65763105323997
Mean squared error for testing data (ridge): 0.6775931639396507


Mean squared error for training data (ridge): 0.6580607069274348
Mean squared error for testing data (ridge): 0.677993829459011
```

```
Mean squared error for training data (ridge): 0.6584892927969834
Mean squared error for testing data (ridge): 0.6783935174532902


Mean squared error for training data (ridge): 0.6589168147838036
Mean squared error for testing data (ridge): 0.6787922314244237


Mean squared error for training data (ridge): 0.6593432768040347
Mean squared error for testing data (ridge): 0.6791899748581405


Mean squared error for training data (ridge): 0.6597686827548808
Mean squared error for testing data (ridge): 0.6795867512240521


Mean squared error for training data (ridge): 0.6601930365147215
Mean squared error for testing data (ridge): 0.6799825639757414


Mean squared error for training data (ridge): 0.6606163419432248
Mean squared error for testing data (ridge): 0.6803774165508497


Mean squared error for training data (ridge): 0.6610386028814558
Mean squared error for testing data (ridge): 0.6807713123711641


Mean squared error for training data (ridge): 0.6614598231519874
Mean squared error for testing data (ridge): 0.6811642548427053


Mean squared error for training data (ridge): 0.6618800065590085
Mean squared error for testing data (ridge): 0.6815562473558119


Mean squared error for training data (ridge): 0.6622991568884329
Mean squared error for testing data (ridge): 0.6819472932852286


Mean squared error for training data (ridge): 0.6627172779080063
Mean squared error for testing data (ridge): 0.6823373959901896


Mean squared error for training data (ridge): 0.6631343733674123
Mean squared error for testing data (ridge): 0.6827265588145034
```

```
Mean squared error for training data (ridge): 0.6635504469983803
Mean squared error for testing data (ridge): 0.6831147850866385


Mean squared error for training data (ridge): 0.6639655025147881
Mean squared error for testing data (ridge): 0.6835020781198056


Mean squared error for training data (ridge): 0.6643795436127689
Mean squared error for testing data (ridge): 0.6838884412120413


Mean squared error for training data (ridge): 0.6647925739708137
Mean squared error for testing data (ridge): 0.6842738776462912


Mean squared error for training data (ridge): 0.6652045972498742
Mean squared error for testing data (ridge): 0.684658390690492


Mean squared error for training data (ridge): 0.6656156170934668
Mean squared error for testing data (ridge): 0.6850419835976528


Mean squared error for training data (ridge): 0.6660256371277729
Mean squared error for testing data (ridge): 0.6854246596059369


Mean squared error for training data (ridge): 0.6664346609617396
Mean squared error for testing data (ridge): 0.6858064219387421


Mean squared error for training data (ridge): 0.6668426921871822
Mean squared error for testing data (ridge): 0.6861872738047823


Mean squared error for training data (ridge): 0.6672497343788807
Mean squared error for testing data (ridge): 0.6865672183981648


Mean squared error for training data (ridge): 0.6676557910946825
Mean squared error for testing data (ridge): 0.6869462588984725


Mean squared error for training data (ridge): 0.6680608658755972
Mean squared error for testing data (ridge): 0.6873243984708405
```

```
Mean squared error for training data (ridge): 0.6684649622458969
Mean squared error for testing data (ridge): 0.687701640266036


Mean squared error for training data (ridge): 0.6688680837132115
Mean squared error for testing data (ridge): 0.6880779874205355


Mean squared error for training data (ridge): 0.6692702337686263
Mean squared error for testing data (ridge): 0.6884534430566026


Mean squared error for training data (ridge): 0.6696714158867765
Mean squared error for testing data (ridge): 0.6888280102823645


Mean squared error for training data (ridge): 0.6700716335259437
Mean squared error for testing data (ridge): 0.6892016921918894


Mean squared error for training data (ridge): 0.670470890128149
Mean squared error for testing data (ridge): 0.6895744918652615


Mean squared error for training data (ridge): 0.6708691891192474
Mean squared error for testing data (ridge): 0.689946412368658


Mean squared error for training data (ridge): 0.6712665339090208
Mean squared error for testing data (ridge): 0.6903174567544222


Mean squared error for training data (ridge): 0.6716629278912715
Mean squared error for testing data (ridge): 0.6906876280611409


Mean squared error for training data (ridge): 0.672058374443912
Mean squared error for testing data (ridge): 0.691056929313717


Mean squared error for training data (ridge): 0.6724528769290583
Mean squared error for testing data (ridge): 0.6914253635234427


Mean squared error for training data (ridge): 0.6728464386931198
Mean squared error for testing data (ridge): 0.6917929336880753
```

```
Mean squared error for training data (ridge): 0.6732390630668893
Mean squared error for testing data (ridge): 0.6921596427919072


Mean squared error for training data (ridge): 0.6736307533656328
Mean squared error for testing data (ridge): 0.6925254938058414


Mean squared error for training data (ridge): 0.674021512889178
Mean squared error for testing data (ridge): 0.6928904896874606


Mean squared error for training data (ridge): 0.6744113449220028
Mean squared error for testing data (ridge): 0.6932546333811017


Mean squared error for training data (ridge): 0.6748002527333233
Mean squared error for testing data (ridge): 0.6936179278179246


Mean squared error for training data (ridge): 0.6751882395771804
Mean squared error for testing data (ridge): 0.6939803759159847


Mean squared error for training data (ridge): 0.6755753086925265
Mean squared error for testing data (ridge): 0.6943419805803028


Mean squared error for training data (ridge): 0.6759614633033115
Mean squared error for testing data (ridge): 0.6947027447029339


Mean squared error for training data (ridge): 0.6763467066185689
Mean squared error for testing data (ridge): 0.695062671163039


Mean squared error for training data (ridge): 0.6767310418324987
Mean squared error for testing data (ridge): 0.6954217628269516


Mean squared error for training data (ridge): 0.677114472124554
Mean squared error for testing data (ridge): 0.6957800225482491


Mean squared error for training data (ridge): 0.6774970006595229
Mean squared error for testing data (ridge): 0.6961374531678187
```

```
Mean squared error for training data (ridge): 0.6778786305876131
Mean squared error for testing data (ridge): 0.6964940575139268


Mean squared error for training data (ridge): 0.6782593650445327
Mean squared error for testing data (ridge): 0.6968498384022853


Mean squared error for training data (ridge): 0.6786392071515743
Mean squared error for testing data (ridge): 0.6972047986361205


Mean squared error for training data (ridge): 0.6790181600156947
Mean squared error for testing data (ridge): 0.6975589410062373


Mean squared error for training data (ridge): 0.6793962267295972
Mean squared error for testing data (ridge): 0.6979122682910877


Mean squared error for training data (ridge): 0.6797734103718113
Mean squared error for testing data (ridge): 0.6982647832568355


Mean squared error for training data (ridge): 0.6801497140067727
Mean squared error for testing data (ridge): 0.6986164886574214
```

[110]:
```python
# convert list to array

Lamda = np.asarray(lamda)
MSE_df = np.asarray(MSE)
# print(features1)

# convert both the array vectors into dataframe
res_1 = pd.DataFrame(Lamda)

res_2 = pd.DataFrame(MSE_df)

# merge dataframes
result_df = pd.merge(res_1, res_2, left_index=True, right_index=True)
result_df = result_df.rename(columns={'0_x':'Lamda','0_y':'MSE'})

result_df.head()
```

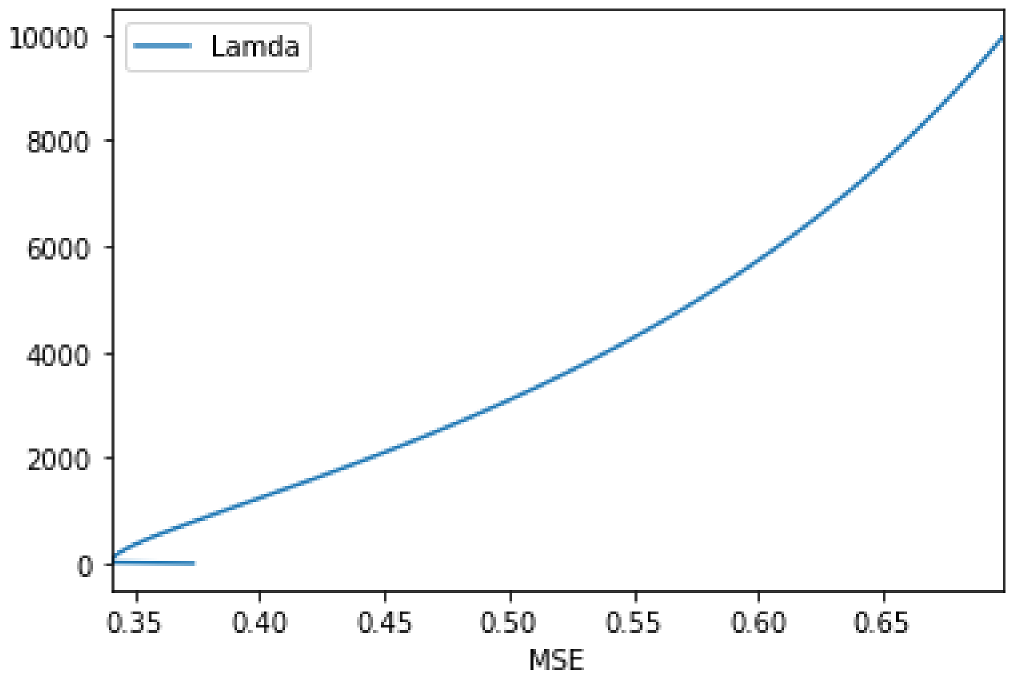
[110]:

	Lamda	MSE
0	0	0.373293

```
1      20   0.340227
2      40   0.340280
3      60   0.340450
4      80   0.340715
```

[111]:
```
result_df.plot.line(x='MSE', y='Lamda')
```

[111]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x291fc58b978>
```



[117]:
```
print("Mean squared error for scaled train data:
 ↪",mean_squared_error(y_train_scaled,predictions_scaled))
print("Mean squared errorfor scaled test data:
 ↪",mean_squared_error(y_test_scaled,predictions_scaled1))
```

```
Mean squared error for scaled train data: 0.2734665681293983
Mean squared errorfor scaled test data: 0.3253062062957616
```

- The best value for lambda is 1, below are the MSE values for lamba = 1.

Ridge regression MSE

- Mean squared error for training data (ridge): 0.27357928858440955
- Mean squared error for testing data (ridge): 0.3402268155372297

package based linear regression MSE

- Mean squared error for scaled train data: 0.2734665681293983
- Mean squared errorfor scaled test data: 0.3253062062957616

There is no much difference in MSE values of rigde and linear regression.

0.0.5 Problem 5 [Dependent features in linear regression]

- In paper format at the end of the pdf

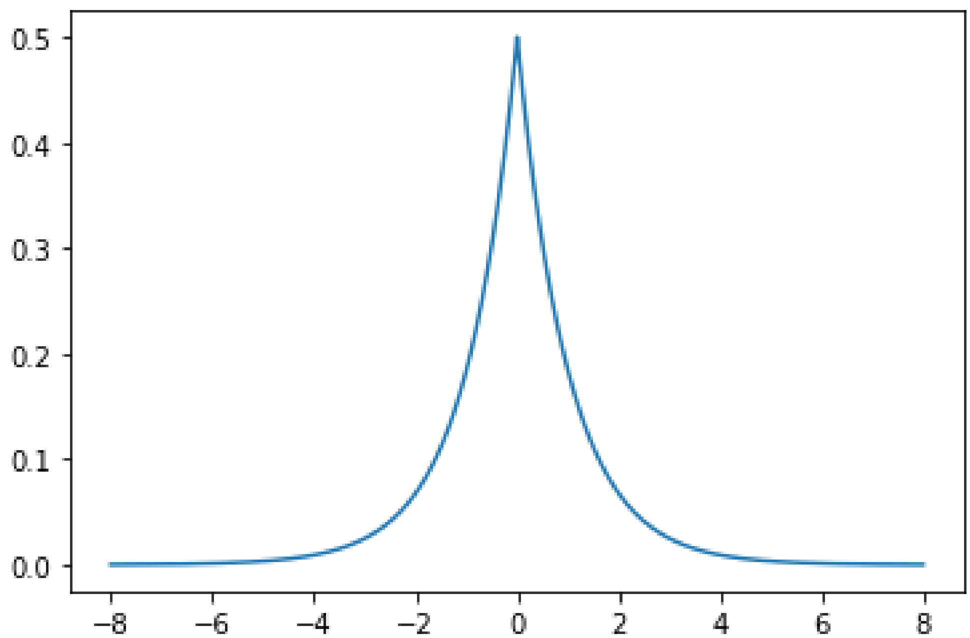
0.0.6 Problem 6 [Laplace noise in MLE estimate]

(a) Plot the Laplace pdf for b {1, 2, 4}. How is the Laplace distribution different from the normal distribution?

[112]:
```
loc, scale = 0., 1.
x_lap = np.random.laplace(loc, scale, 1000)

x = np.arange(-8., 8., .01)
pdf = np.exp(-abs(x-loc)/scale)/(2.*scale)
plt.plot(x, pdf)
```

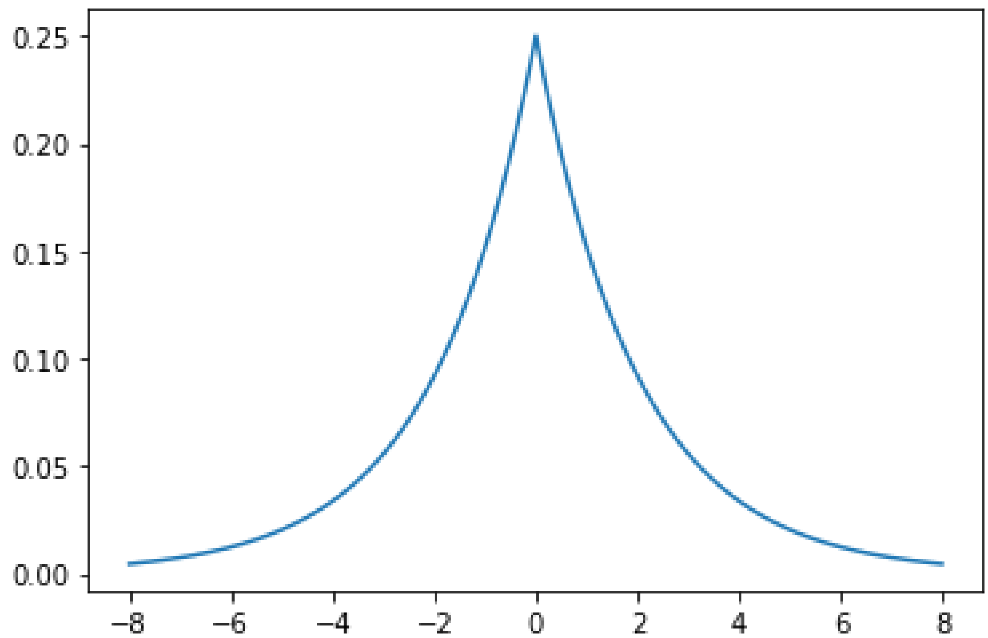
[112]:
```
[<matplotlib.lines.Line2D at 0x291fdad32e8>]
```



[113]:
```
loc, scale = 0., 2.
x_lap = np.random.laplace(loc, scale, 1000)

x = np.arange(-8., 8., .01)
pdf = np.exp(-abs(x-loc)/scale)/(2.*scale)
plt.plot(x, pdf)
```

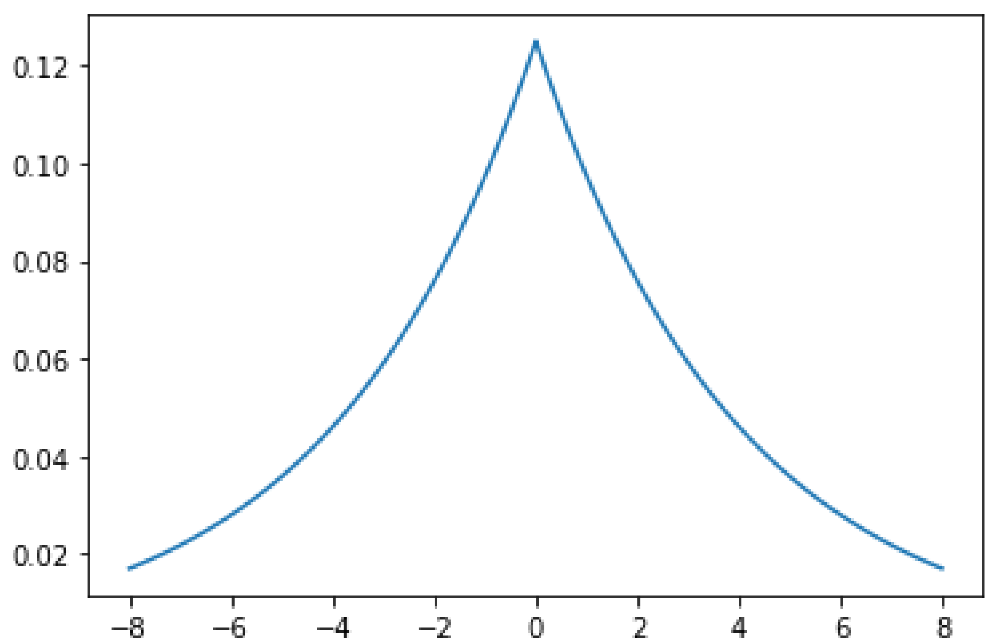
[113]:
```
[<matplotlib.lines.Line2D at 0x291fdb31588>]
```



[114]:
```
loc, scale = 0., 4.
x_lap = np.random.laplace(loc, scale, 1000)

x = np.arange(-8., 8., .01)
pdf = np.exp(-abs(x-loc)/scale)/(2.*scale)
plt.plot(x, pdf)
```

[114]: [<matplotlib.lines.Line2D at 0x291fdb908d0>]



(b) Derive the objective J() that maximizes the MLE for Laplace noise. The objective should depend on , the training dataset (xi, yi), for i {1, . . . , N}, and b.

- In paper format at the end of the pdf

(c) Consider an outlier in the training data, defined as a point of high residual. Which of the two objectives (derived for normal or Laplace noise) are more resilient to the effect of outliers?

- The pdf of laplace distribution is (1/2b)*eˆ(-x/b) and the MLE is 1.
- The pdf of normal distribution is (1/sqrt(2pi))*e$^{(x}$2/b) and the MLE is 2x.
- The normal distribution is more vulnerable to outliers.
- Therefore, laplace noise is more resilient to the effect of outliers.

4a) Derivation of the closed form solution for parameter θ that minimizes the loss function $J(\theta)$ in ridge regression.

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N}\left(h_\theta(x_i) - y_i\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

The term d in regularization part denotes amount of covariables used in the model.

we can rewrite $J(\theta)$ in matrix-notation and further break it down.

$$J(\theta) = \frac{1}{2}(X\theta - y)^T(X\theta - y) + \frac{\lambda}{2}(\theta^T\theta)$$

the term $\frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$ represents regularization we apply on coefficients.

$$\Rightarrow \frac{1}{2}\left[X^T\theta^T X\theta - X^T\theta^T y - X\theta y^T + y^T y + \lambda\theta^T\theta\right]$$

$\left[(X\theta)^T y = (X\theta)y^T\right]$ the a transposed scalar is same scalar.

$$\Rightarrow \frac{1}{2}\left[X^T\theta^T X\theta - 2X^T\theta^T y + y^T y + \theta^T\lambda I\,\theta\right]$$

$[I = \text{Identity matrix}]$

$$\Rightarrow \frac{1}{2}\left[y^T y - 2X^T\theta^T y + \theta^T(X^T X + \lambda I)\theta\right]$$

θ should minimize $J(\theta)$.

By matrix differentiation rule $\frac{\partial x^T A x}{\partial x} = (A + A^T)x = 2Ax$

we can apply it in here as

$$\frac{\partial J}{\partial\theta} \Rightarrow \frac{1}{2}\left[0 - 2X^T y + 2(X^T X + \lambda I)\theta\right] = 0.$$

$$\Rightarrow -X^T y + (X^T X + \lambda I)\theta = 0$$

$$\Rightarrow (X^T X + \lambda I)\theta = X^T y.$$

$\boxed{\theta = (X^T X + \lambda I)^{-1} + X^T y.}$ is a closed form eq.

5) $X = \begin{bmatrix} 1 & x_{i1} & x_{j1} \\ 1 & x_{i2} & x_{j2} \\ 1 & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1 & x_{in} & x_{jn} \end{bmatrix}_{n\times 3}$

$X^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{i1} & x_{i2} & \cdots & x_{in} \\ x_{j1} & x_{j2} & \cdots & x_{jn} \end{bmatrix}_{3\times n}.$

$x_{j1} = 2x_{i1}$

So $X = \begin{bmatrix} 1 & x_{i1} & 2x_{i1} \\ 1 & x_{i2} & 2x_{i2} \\ \vdots & \vdots & \vdots \\ 1 & x_{in} & 2x_{in} \end{bmatrix}_{n \times 3}$

$X^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{i1} & x_{i2} & \cdots & x_{in} \\ 2x_{i1} & 2x_{i2} & \cdots & 2x_{in} \end{bmatrix}_{3 \times n}$.

$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{i1} & x_{i2} & \cdots & x_{in} \\ 2x_{i1} & 2x_{i2} & \cdots & 2x_{in} \end{bmatrix} \begin{bmatrix} 1 & x_{i1} & 2x_{i1} \\ 1 & x_{i2} & 2x_{i2} \\ \vdots & \vdots & \vdots \\ 1 & x_{in} & 2x_{in} \end{bmatrix}$

$X^T X = \begin{bmatrix} 1 & \sum_{i=1}^{n} x_i & 2\sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & 2\sum_{i=1}^{n} x_i^2 \\ 2\sum_{i=1}^{n} x_i & 2\sum_{i=1}^{n} x_i^2 & 4\sum_{i=1}^{n} x_i^2 \end{bmatrix}$ Row 1 and Row 2 are linearly dependent

Rank $(X^T X) = 1$ (using row echelon form).

The columns of $X^T X$ are linearly dependent. col 1 $\times \sum_{i=1}^{n} x_i =$ col 2 and etc.

As per invertible matrix theorem linearly dependent matrix do not have inverses.

So, $X^T X$ is non-invertible.

6 b) Laplace PDF

$$P(x) = \frac{e^{-|x|/b}}{2b}$$

hypothesis be,

$$y_i = h_\theta(x_i) + \epsilon_i$$

ϵ_i - random noise generated from laplace distribution.

$$f(y_i | x_i ; \theta, b) = \frac{1}{2b} e^{\frac{-|y_i - (\theta_0 + \theta_1 x_i)|}{b}}$$

(func of y_i given x_i and parameters for laplace distribution)

For, Assuming that the points are independent.

$f(y_1 \ldots y_n | x_1 \ldots x_n, \theta, b)$. $P[y|x, \theta] = \max_\theta L(\theta)$

MLE $L(\theta) = \prod_{i=1}^{n} f(y_i | x_i, \theta)$

$$\log(L(\theta)) = \sum_{i=1}^{n} \log(f(y_i | x_i, \theta))$$

$$\Rightarrow \sum_{i=1}^{n} \log\left(\frac{1}{2b} \cdot e^{-|y_i - (\theta_0 + \theta_1 x_i)|/b}\right)$$

$$\Rightarrow \sum_{i=1}^{n} \left[\log\left(\frac{1}{2b}\right) - \frac{|y_i - (\theta_0 + \theta_1 x_i)|}{b} \right]$$

Minimize all negative terms to maximize the MLE function.

(ie) $$J(\theta) = \sum_{i=1}^{n} \frac{|y_i - (\theta_0 + \theta_1 x_i)|}{b}$$

$\longrightarrow$ MAE.

$$\therefore J(\theta) = \frac{1}{b} \sum_{i=1}^{n} |y_i - (\theta_0 + \theta_1 x_i)|$$