Assignment 1: Linear, Ridge, & Lasso Regression

09/29/2024

Submitted by:

Arun Rimal

Contents

List of Tables

List of Figures

1 Introduction

Regression is a method to determine the strength between the dependent and independent variables. It is used to determine the relation between the variables. Linear Regression is the fundamental type of regression analysis. It has an equation also called a function similar to the equation of the line which has one independent(predictor) and dependent(target label) variable with a constant(intercept). The objective of linear regression is to find the best-fit line in the case of one independent variable whereas the hyperplane in the case of multiple independent variables.
For a simple linear regression with one target lable:

$$\text{y = wx + b}$$

Where:

- y is the target or dependent variable,
- x is the feature or independent variable,
- b is the constant or intercept,
- w is the coefficient or weight

Multi variant linear regression looks like:

$$y = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

Ridge Regression is a type of regression that can regularize the prediction by penalizing the features having multicollinearity between themself. Multicollinearity adds up during prediction which leads to overfitting. So to prevent overfitting ridge regression adds a regilarization term to the cost function to prevent overfitting.
Cost function with regularization by adding penalty:

$$\text{Cost} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}w_j^2$$

where:

- λ is the regularization parameter,
- $\sum_{j=1}^{p} w_j^2$ is sum of square of coefficient.

Lasso Regression is similar to Ridge Regression in terms of applying regularization. Like Ridge, Lasso also penalizes the features by adding certain corrections. However, the lasso's regularisation is with the absolute value of coefficients, unlike the square value of the coefficient. It has a significant impact on feature

selection, as it can eliminate irrelevant features by setting their coefficient to zero.
Cost function with regularization by adding penalty:

$$\text{Cost} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |w_j|$$

Where:

- λ is the regularization parameter,
- $|w_j|$ is the absolute value of the coefficient.

2 Methods

2.1 Datasets

The datasets used for regression analysis are Boston Housing data sets and Fish data sets.
The details for the Boston Dataset are as follows:
Number of Instances: 506
Number of Attributes: 13 numeric/categorical predictive Median Value (attribute 14) is usually the target.
Attribute Information (in order):

CRIM: Per capita crime rate by town.

ZN: Proportion of residential land zoned for lots over 25,000 sq. ft.

INDUS: Proportion of non-retail business acres per town.

CHAS: Charles River dummy variable (1 if tract bounds river; 0 otherwise).

NOX: Nitric oxides concentration (parts per 10 million).

RM: Average number of rooms per dwelling.

AGE: Proportion of owner-occupied units built prior to 1940.

DIS: Weighted distances to five Boston employment centers.

RAD: Index of accessibility to radial highways.

TAX: Full-value property-tax rate per $10,000.

PTRATIO: Pupil-teacher ratio by town.

B: $1000(Bk - 0.63)^2$, where Bk is the proportion of black people by town.

LSTAT: Percentage of lower status of the population.

The details for the Fish dataset are as follows:
Number of Instances: 159
Number of Attributes: 7 numeric/categorical predictive Weight is the target variable.
Attribute Information (in order):

Species: Species of fish.

Weight: Species of fish.

Length1: Length of fish.

Length2: Length of fish.

Length3: Length of fish.

Height: Height of fish.

Width: Width of fish.

2.2 Methods for Exploratory Data Analysis

Exploratory Data Analysis(EDA) is used to understand data by summarizing and analyzing the important statistics of the dataset. It is done using both methods graphical and non-graphical techniques by visualizing the patterns. Some of the methods used are as follows

- **Statistics**: Mean, median, variance, percentiles. Using describe() method.

```
# Checking the statistics of dataset
df_boston.describe()
```

- **Visualizations**: Using histograms, scatter plots, box plots, heatmaps, and other plots.

```
fig, ax = plt.subplots(figsize = (18,10))
sns.heatmap(correlation_matrix, annot=True, annot_kws =
    {'size': 12} )

fig, axis = plt.subplots(2,2, figsize = (20,12))

# Creating a stacked histogram
axis[0,0].hist([df_boston['RM'], df_boston['Price']],
    bins=20, stacked=True, color=['cyan', 'Purple'],
    edgecolor='black')
axis[0,0].legend(['RM', 'Price'])
axis[0,0].set_xlabel('Values')
axis[0,0].set_ylabel('Frequency')
axis[0,0].set_title('Stacked Histogram')
```

```
# Creating box plot
axis[0,1].boxplot([df_boston['RM'], df_boston['Price']],
    patch_artist=True, labels=['RM', 'Price'])
axis[0,1].set_title('Box Plot')
axis[0,1].set_xlabel('Feature and Target')
axis[0,1].set_ylabel('Values')

# Creating Scatter plot
axis[1,0].scatter(x = df_boston['RM'], y = df_boston['
    Price'])
axis[1,0].set_title('Scatter Plot')
axis[1,0].set_xlabel('RM')
axis[1,0].set_ylabel('Price')


# Creating a 3D scatter plot
axis_3d = fig.add_subplot(2, 2, 4, projection='3d')

axis_3d.scatter(df_boston['RM'], df_boston['PTRATIO'],
    df_boston['Price'], c='blue', marker='o')
axis_3d.set_xlabel('RM')
axis_3d.set_ylabel('PTRATIO')
axis_3d.set_zlabel('Price (USD)')
axis_3d.set_title('3D Scatter Plot of House Prices')

# Display the plot
plt.tight_layout()
plt.show()
```

- **Missing Data**: Identifying and visualizing missing or null values. Using isnull(), sum() and other methods.

```
# Check if the data frame has any null values
df_boston.isnull().sum()
```

- **Outlier Detection**: Identifying and visualizing outliers. Using box plots, Z-scores, Interquartile Range (IQR) and other methods.

- **Correlation**: Analyzing the relation between features and target. Using correlation matrices, scatter plots and other methods.

```
# Creating correlation matrix
correlation_matrix = df_boston.corr()
correlation_matrix

# Selecting features which has strong corelation with
    target variable from correlation matrix
```

```
def getTargetFeaturepair(correlation_matrix ,
    targetVariable='Price'):
    unstack = correlation_matrix.unstack()
    # We know what our target variable is
    target_variable = targetVariable
    df_coef = pd.DataFrame(columns=['Target', 'Feature',
        'Corr_coef'])
    for multiIndex in unstack.index:
        if multiIndex[0]==target_variable:
            # Get correlation having strong positive and
                negative from threshold value +- 0.5 (
                Suppose)
            if correlation_matrix[multiIndex[0]][
                multiIndex[1]] > 0.5 or correlation_matrix
                [multiIndex[0]][multiIndex[1]] < -0.5:
                corr_coef = correlation_matrix[multiIndex
                    [0]][multiIndex[1]]
                df_coef = df_coef.append({
                    'Target': target_variable,
                    'Feature': multiIndex[1],
                    'Corr_coef': corr_coef
                }, ignore_index=True)
    return df_coef

df_coef = getTargetFeaturepair(correlation_matrix)
df_coef
```

- **Dimensionality Reduction**: Detecting similar or highly multi-collinear features and reducing them to a single feature. Using Techniques like Principal component analysis (PCA).

```
# Performing Principle Component Analysis(PCA)
from sklearn.decomposition import PCA
pca = PCA(3) # 3 refers to 3 components or feature
X_pca = pca.fit_transform(X)
X_pca.shape
```

Correlation Matrix for Boston Housing datasets:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	Price
CRIM	1.000000	-0.200469	0.406583	-0.055892	0.420972	-0.219247	0.352734	-0.379670	0.625505	0.582764	0.289946	-0.385064	0.455621	-0.388305
ZN	-0.200469	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.406583	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055892	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.420972	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219247	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.352734	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.379670	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.625505	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.582764	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
PTRATIO	0.289946	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.385064	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.455621	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
Price	-0.388305	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

Table 1: Correlation matrix of Boston Datasets

Target	**Feature**	**Corr_coef**
Price	RM	0.695360
Price	PTRATIO	-0.507787
Price	LSTAT	-0.737663
Price	Price	1.000000

Table 2: Feature and Target Correlation

Histogram, Box-plot, Scatter plot for feature and target of the housing data set is shown below:

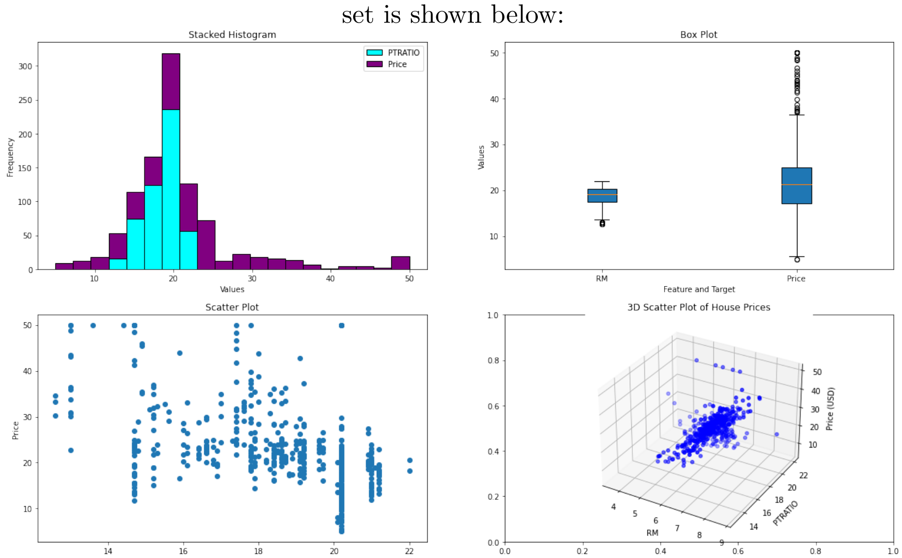

Figure 1: Feature vs Target Plot

Features with high correlation with Target in Fish Datasets:

	Weight	Length1	Length2	Length3	Height	Width
Weight	1.000000	0.915712	0.918618	0.923044	0.724345	0.886507
Length1	0.915712	1.000000	0.999517	0.992031	0.625378	0.867050
Length2	0.918618	0.999517	1.000000	0.994103	0.640441	0.873547
Length3	0.923044	0.992031	0.994103	1.000000	0.703409	0.878520
Height	0.724345	0.625378	0.640441	0.703409	1.000000	0.792881
Width	0.886507	0.867050	0.873547	0.878520	0.792881	1.000000

Heat map for Fish data set is shown below:

Figure 2: Fish Heat Plot

2.3 Methods for Multivariate Linear Regression

Multivariate Linear Regression is a type of linear regression where multiple independent variables are introduced to predict the target label.
Steps involved in regression analysis are:

- The Loaded data frame is divided into X and Y variable. X is a data frame or numpy ndarray of features, Y is a series with target value

```
# Loading datasets for Boston Housing and creating
    DataFrame for features only
df_feature = pd.DataFrame(data = bostn.data , columns=
    bostn.feature_names)
# Creating dataframe for target variable
sr_target = pd.Series(data=bostn.target, name='Price')
# Concatinating two data frames feature and target
df_boston = pd.concat([df_feature,sr_target], axis=1)

# Load CSV file of Fish Dataset
df = pd.read_csv("Fish.csv")
```

- X and Y are split into train and test data.

```
# Split into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y,
     test_size=0.2, random_state=42)
```

- **Feature Scaling** is performed on the X train and test data.

```
# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Why Feature Scaling ?

Importance of Feature Scaling in Regression Analysis

1. Feature Scaling is used to bring the column values in a range suitable for modelling to reduce the disparity among the feature range values. Some features might be normalized but others may not. So to eliminate such issues other features are brought into the range easier for an algorithm to perform. 2. The ranges are generally reduced to very small range like -1 to 1 or sth similar. 3. Facilitate easier and faster convergence of gradient descent algorithm to minimum error. 4. To reduce the impact of outliers and perform regularization by penalizing features fairly using L2 and L1 technique. Feature having large values may unfairly get penalized so feature scaling is important. 5. Common feature scaling are z-score normalization and min-max scaling.

- For *OLS* regression analysis, X train and test datasets are added with a constant value using the *add_constant* function.

```
# Add a constant to the feature set for OLS regression
X_train_ols = sm.add_constant(X_train)
X_test_ols = sm.add_constant(X_test)
```

- For Linear, Ridge and Lasso regression does not require constant.
- Ridge and Lasso uses λ (α in code) as a regularization values.
- X train data is fitted with the respective regression models.

```
from sklearn.linear_model import LinearRegression,
    RidgeCV, LassoCV
# Initialize and train models
ols_model = sm.OLS(y_train, X_train_ols).fit()
linear_model = LinearRegression().fit(X_train_scaled,
    y_train)
ridge_cv = RidgeCV(alphas=[0.1, 1.0, 10.0, 100.0]).fit(
    X_train_scaled, y_train)
lasso_cv = LassoCV(alphas=[0.001, 0.01, 0.1, 1.0, 10.0]).
    fit(X_train_scaled, y_train)
```

- After training the model with the train data, test data is fed to make the prediction.

```
# Predict with the test data
y_pred_ols = ols_model.predict(X_test_ols)
y_pred_linear = linear_model.predict(X_test_scaled)
y_pred_ridge = ridge_cv.predict(X_test_scaled)
y_pred_lasso = lasso_cv.predict(X_test_scaled)
```

- After the prediction is calculated, models R^2 and MSE are calculated.

```
from sklearn.metrics import mean_squared_error, r2_score
# Calculate and print R^2 score and MSE for each model
models = ['OLS', 'Linear', 'Ridge', 'Lasso']
predictions = [y_pred_ols, y_pred_linear, y_pred_ridge,
    y_pred_lasso]
for name, y_pred in zip(models, predictions):
print(f"{name} Regression R^2 score: {r2_score(y_test,
    y_pred):.4f}")
print(f"{name} Regression MSE: {mean_squared_error(y_test
    , y_pred):.4f}\n")
```

Methods used to implement multivariant linear regression are as follows:

- **Ordinary Least Square (OLS) Regression**: It is a common technique used to find the prediction value in multivariate linear regression. It focuses on calculating the minimum sum of squares of residual. Residual is a difference between actual and predicted value.
- **Linear Regression**: It is another common technique used to find the prediction value in multiple linear regression. It also consists of many independent variables and single dependent variables.
- **L2 Regularization**: L2 regularization is an approach used to penalize the features with multicollinearity. This method is used when there are many features in the datasets which can influence the prediction causing an overfitting.
- **L1 Regularization**: L1 regularization is also an approach used to penalize the features with multicollinearity. This method is used when there are many features in the datasets which can influence the prediction causing an overfitting. The penalizing value is different than the L2 regularization.

3 Result and Discussion

3.1 Performance Evaluation Using R^2 Score

The performance of the model is evaluated using R^2 score. It measures how well the independent features represent the variance of the target variable during prediction. It is also termed as the goodness of fit.

R^2 Score for different datasets are given below.

Datasets	$R^2 Scores$			
	OLS	**Linear Regression**	**Ridge Linear Regression**	**Lasso Linear Regression**
Boston Housing Dataset	0.6688	0.6688	0.6660	0.6687
Fish Dataset	0.8840	0.8821	0.8770	0.8773

Table 3: R^2 Score Table

3.2 Performance Evaluation Using MSE

The model's performance is evaluated using Mean Square Error(MSE). It measures the accuracy of the prediction. The lower the MSE the more accurate the prediction is.

MSE Score for different datasets are given below.

Datasets	MSE			
	OLS	**Linear Regression**	**Ridge Linear Regression**	**Lasso Linear Regression**
Boston Housing Dataset	27.1150	27.1150	27.0673	27.0806
Fish Dataset	16763.8872	17343.8112	17488.5753	17458.6532

Table 4: MSE Score Table

3.3 Analyzing MSE and R^2 for Multivariate Linear Regression

The model's performance is evaluated using R^2 and Mean Square Error(MSE). The combination of both values gives more insight into the model's accuracy and goodness of fit.

R^2 and MSE Score for different datasets are given below.

Datasets	Test	Regression Analysis			
		OLS	**Linear Regression**	**Ridge Regression**	**Lasso Regression**
Boston Dataset	R^2	0.6688	0.6688	0.6660	0.6687
	MSE	27.1150	27.1150	27.0673	27.0806
Fish Dataset	R^2	0.8840	0.8821	0.8770	0.8773
	MSE	16763.8872	16763.8872	17488.5753	17458.6532

Table 5: R^2 and MSE table

For the **Boston Housing Dataset**, all models((OLS, Linear, Ridge, and Lasso)) have R^2 around 0.6688, indicating that about 66.88% of the variance in the target variable is explained by the models. Ridge and Lassos show nominally below the other two models which might be due to the regulation factor applied by the models. In the case of MSE, all the model has a consistent accuracy of around 27. The lower the MSE the higher the accuracy is. The works on to reduce the disparity between actual target value and predicted value. The difference is the error, this error is represented by MSE.
For the **Fish Dataset**, R^2 is higher between 0.884 and 0.877, which shows that these models explain around 88.0% of the variance in the target variable. However, the R^2 score of ridge and lasso are below the OLS and linear model due to the regulatory factor applied to the features. This is done to reduce the overfitting. However, overfitting is not observed here. In the case of MSE, all the model has very high MSE values. The lower the MSE the higher the accuracy is. Although the R^2 values are in good range the MSE is very high. This might be due to the larger range of target values or more variability in the data.

All model has similar values of R^2 scores due to the following reasons:

1. The dataset used might be small in size.
2. The feature might be less.
3. Dataset may not need L1 and L2 regularization due to low-value range and no multicollinearity.
4. Maybe features were inadequate to require 'Ridge Regression' and 'Lasso Regression'.

3.4 Analyzing MSE and R^2 for Controlled(selected features) Linear Regression for Boston Housing Dataset

The model's performance is evaluated using only selected or strong features. R^2 and Mean Square Error(MSE) are than analyze to evaluate model performance. The combination of both values gives more insight into the model's accuracy and goodness of fit. Features such as RM, PTRATIO and LSTAT are selected. These features have strong correlation with the Price target variable.

R^2 and MSE Score for Boston datasets for strong correlated features are given below.

Controlled/Full LR	**Test**	**Regression Analysis for Boston Housing**			
		OLS	**Linear Regression**	**Ridge Regression**	**Lasso Regression**
Full LR(all features)	R^2	0.6688	0.6688	0.6660	0.6687
	MSE	27.1150	27.1150	27.0673	27.0806
Controlled LR(few features)	R^2	0.6303	0.6303	0.6309	0.6307
	MSE	27.1150	27.1150	27.0673	27.0806

Table 6: Full vs Controlled Linear Regression

The R^2 and MSE scores of the Boston Housing dataset, when selected all features, is around 0.6688 and around 27 for all models, which means the model is successful in explaining 66.88% of variance in the target variable with an error of MSE approximately 27. Whereas, with the selected feature only i.e. RM, PTRATIO and LSTAT, the model performs slightly lower and is only able to explain around 63% of the target variance with MSE of 27 for all models. This shows that selecting only strongly correlated features from this dataset is not applicable or may cause a loss of information from other weakly correlated features. In the case of the Boston dataset, controlled LR causes to drop some useful features that can provide some relevant information.

3.5 Analyzing MSE and R^2 for Controlled(selected features) Linear Regression using PCA for Fish Dataset

The model's performance is evaluated using only selected features by Principle Component Analysis(PCA). R^2 and Mean Square Error(MSE) are then calculated to evaluate the model. The combination of both values gives more insight into the model's accuracy and goodness of fit.

R^2 and MSE Score for Fish datasets are given below.

Dataset	**PCA**	**Linear Regression**	
		R^2	**MSE**
Fish Dataset	**w/o PCA**	0.8821	16763.8872
	with PCA	0.8781	17343.8112

Table 7: With and w/o PCA R^2 and MSE for Linear Regression

The R^2 and MSE scores without using Principle Component Analysis(PCA) is 0.8821 and around 16763, which means the model is successful in explaining 88.21% of variance in target variable with an error of MSE approximately 16763. Whereas with PCA, the model performs slightly lower and is only able to explain 87.81% of the target variance with MSE of 17343. This shows that PCA for this dataset is not applicable. In the case of the fish dataset, PCA has dropped some useful features that can provide some relevant information.

3.6 Analyzing MSE and R^2 for Univariate Linear Regression

Under univariate linear regression, only one feature is taken under consideration to predict the target variable.

Dataset	Test	Regression Analysis			
		OLS	**Linear Regression**	**Ridge Regression**	**Lasso Regression**
Boston Dataset	R^2	0.3708	0.3708	0.3711	0.3730
	MSE	46.1448	46.1448	46.1170	45.9829

Table 8: R^2 and MSE table for univariate LR

When only one feature is taken as an independent variable the R^2 score is significantly lower whereas the MSE value is significantly higher compared to the multivariate linear regression. R^2 score 0.3708 shows that the target variable variance is only 37% explained by the input variable. This R^2 score is considered lower in linear regression models. Close to 1 is considered the better model. Similarly, the MSE is high close to 47. It shows that the sum of the error between the actual and predicted values is high.

Actual vs Prediction for Boston housing dataset is shown below:

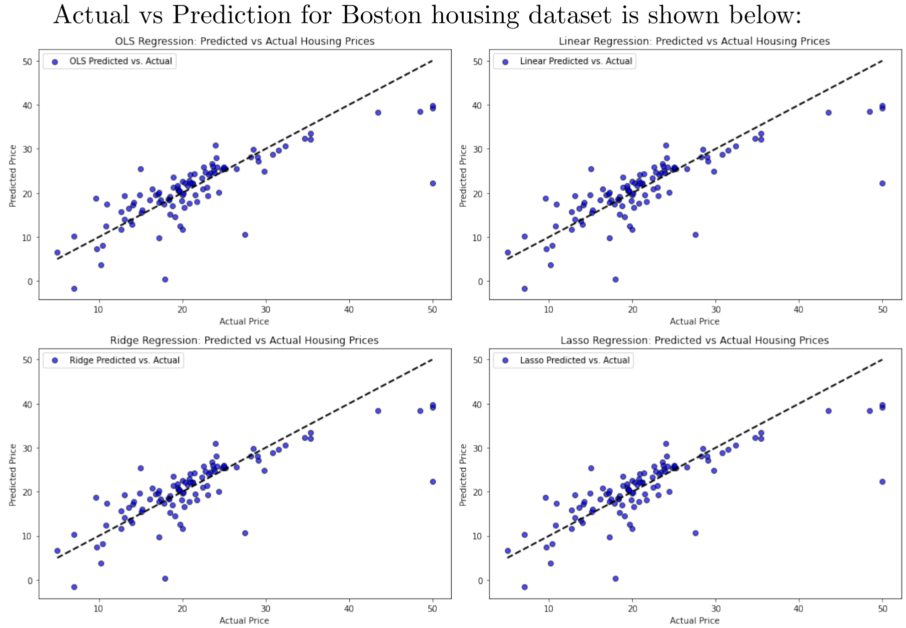

Figure 3: Actual vs Prediction Plot for Boston Housing

The above plot consists of the actual housing price vs. the predicted cost of Boston housing for four models i.e. OLS, Linear regression, Ridge and Lasso respectively. On the x-axis, it is the actual housing prices whereas on the y-axis, it is the predicted housing prices. The diagonal dotted line is a reference line which shows the ideal case of the best-fit line (or perfect prediction line). The points close to the line show that the prediction is more close to accurate.

Actual vs Prediction for Fish dataset is shown below:

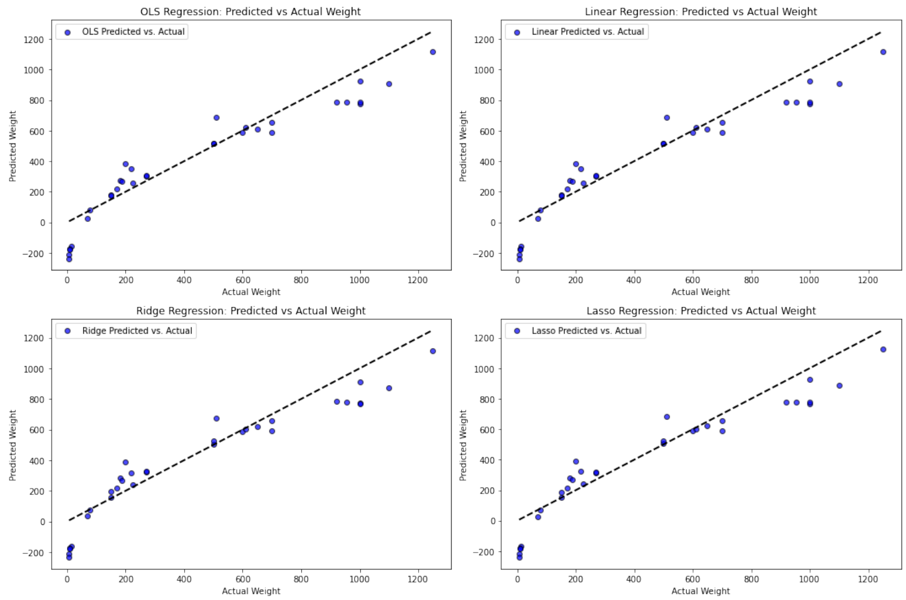

Figure 4: Actual vs Prediction Plot for Fish

The above plot consists of the actual weight vs. the predicted weight of the Fish dataset for four models i.e. OLS, Linear regression, Ridge and Lasso respectively. On the x-axis, it is the actual Weight whereas on the y-axis, it is the predicted Weight. The diagonal dotted line is a reference line which shows the ideal case of the best-fit line (or perfect prediction line). The points close to the line show that the prediction is more close to accurate.

4 Conclusion

The report consists of the implementation of linear regression on Boston housing and Fish dataset. Both datasets are applied with various models of linear regression. The datasets were fed into OLS, Linear regression, and Ridge and Lasso regression models for training the datasets. All models performed similarly for the Boston Housing dataset with R^2 scores around 0.6688 and MSE around 27. Ridge and Lasso showed slightly lower performance due to regularization. It limits overfitting by reducing the variance explained by the model for target variables. For the Fish dataset, R^2 scores remained high (around 0.88), and the MSE was significantly larger. This shows that the models explained a large proportion of the target variance. But made high errors in predictions. Likely due to the range of target values in the dataset. Fish dataset when applied with PCA, features were reduced to only three predictor features. However, the model performance shows slightly lower with 87.81% target variance explained with an error as MSE 17343. Univariate regression models showed weak performance, with lower R^2 and higher MSE, indicating that multiple features are necessary to predict the target variable accurately. The results obtained from the model were similar. There is no significant change in the data sets. However, there are some common patterns observed as ridge and lasso regression R^2 scores were found nominally lower than ols and linear regression. This is due to the penalty that has been applied as regularization on features. The result indicated that:

- All models performed similarly on both datasets but the ridge and lasso performed slightly lower with lower R^2 and higher MSE scores.
- Boston housing dataset showed consistent performance irrespective of the model with moderate R^2 and MSE values. Fish datasets show a high R^2 value but a very large MSE value which creates some doubt on its prediction accuracy.
- Linear regression on the Fish dataset showed better performance without PCA. Showing that the PCA may cause the loss of some important information.
- Controlled Linear Regression for Boston also caused low-performance model to lead to low accuracy and higher error.
- Univariate regression performs worse than multivariate, highlighting the consequences of information loss and the importance of multiple features to capture complex relationships in data.