Classification and Regression Problems on Four Datasets

Author: Likhitha Eda
University of Massachusetts Lowell

I. INTRODUCTION

In our course, we learned many Machine Learning techniques to analyze datasets. In this research assignment, I have explored four different classification and regression algorithms. Some of these algorithms were taught in our course. In particular, I used Linear Regression and Polynomial regression on the Boston House Prices dataset. I also used SVM classification from our and Decision Tree Classifier on Olivetti Faces dataset. The algorithms that I learned on my own are Ridge Regression and Bayesian Ridge Regression, which I applied on California Housing Dataset. Lastly, Naïve Bayes Gaussian Distribution and Logistic Regression were used on Breast Cancer Dataset.

II. REGRESSION DATASET: BOSTON HOUSE PRICE

a. The Dataset

The first dataset that I chose was the Boston House Price dataset taken from StatLib library at Carnegie Mellon University. This is a small dataset consisting of 506 instances and 13 attributes. The models I used to visualize the correlation between the variables in the dataset are heat map, scatter plot, distance plots, regression plots, and line plots. Using these methods I was able to implement linear regression and polynomial regression to best fit the dataset.

b. Visualization

To understand the dataset, I printed the target values, feature names, and data for the features. I created a data frame on for the dataset. I made sure that no data is missing in the data frame by using the method isnull(). Next using the heat map feature from seaborn, I created a correlation matrix(fig 1). The heat map allowed me to find that MEDV and LSTAT, RM and MEDV had high correlation. Using these variables, I applied the regression algorithms.

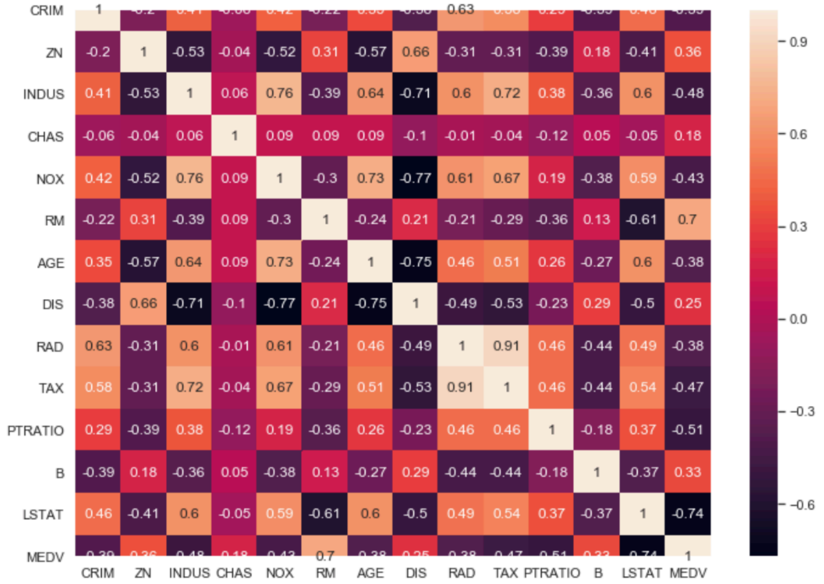
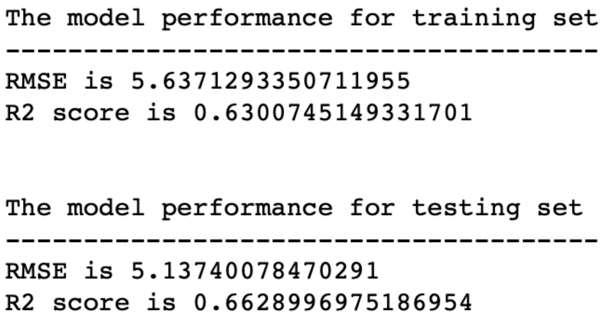


Figure 1: Boston House Prices HeatMap

c. Linear Regression

For this algorithm I created X and Y variables to use as input of the train_test_split function. The X variable is a data frame of data from two variables, LSTAT and RM which were highly correlated with MEDV. The Y variable is the list of data from MEDV column. I split this data using train_test_split then applied the linear regression function and fit the training data. The figure below is the performance output for this algorithm. As we can see the accuracy score is very low meaning that there were many outliers in the graph.



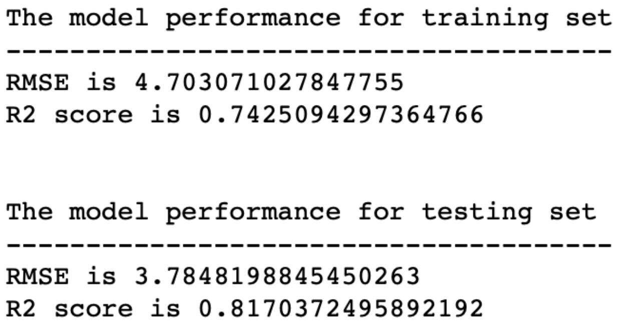
```
The model performance for training set
--------------------------------------
RMSE is 5.6371293350711955
R2 score is 0.6300745149331701


The model performance for testing set
--------------------------------------
RMSE is 5.13740078470291
R2 score is 0.6628996975186954
```

Figure 2: Linear regression on Boston Housing Dataset

d. Polynomial Regression

Since the Linear regression gave a low accuracy score, I decided to use Polynomial Regression to improve the accuracy score. As we can see in figure below, the score had improved by 10% for training set and 15% for the testing set.

The implementation of Polynomial Regression is similar to the Linear Regression. Using numppy and polynomial features, I was able to polyfit the train_test_split data. The methods I used to visualize the regression graphs are scatterplot, regplot, and lineplot.



```
The model performance for training set
--------------------------------------
RMSE is 4.703071027847755
R2 score is 0.7425094297364766


The model performance for testing set
--------------------------------------
RMSE is 3.7848198845450263
R2 score is 0.8170372495892192
```

Figure 3:Polynomial Regression on Boston Housing Dataset

III. REGRESSION DATASET: CALIFORNIA HOUSES

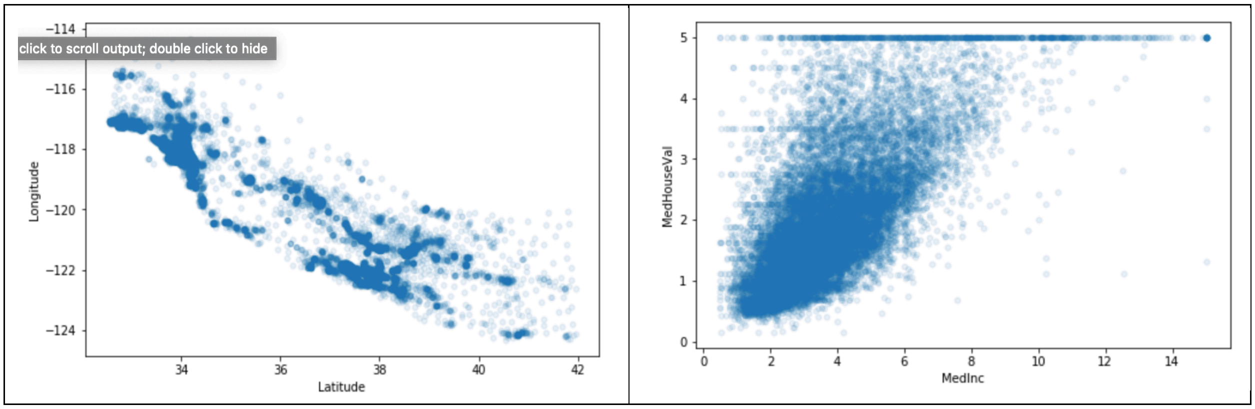
a. The Dataset

The California houses dataset is large regression dataset consisting of 20,640 instances and 8 attributes. The dataset is derived from 1990 U.S. census. To analyze this dataset, I had find the features that are most correlated with the target, Median House Value. I applied the Ridge Regression and Bayesian Ridge Regression to this dataset.

b. Visualization

To visualize the data, I used a scatter matrix. The matrix allowed me to see the variables that were most correlated are longitude and latitude as well as median house value and median Income. I found that the most correlation was between latitude vs longitude and

Median Income and Median House Value. The location of the house tells where most houses were located. But Locational information was not included in the dataset.



c. *Ridge Regression*

Ridge regression algorithm helps us to reduce the complexity of the model. It does this by adding a cost (lambda) to the optimization function whenever large values are read. This is important for this dataset because it has a lot of data points and can help with trying not to overfit. My results show that the train and test score with low alpha and high alpha do not vary at all. The train score with low and high alpha is 0.609. The test score for low alpha is 0.591 and high alpha is 0.596. The graph below shows that the blue and red dots align very well on top of each other. The only exception is at coefficient 3 where the higher alpha has a lower coefficient magnitude, means that the coefficient is shrinking.

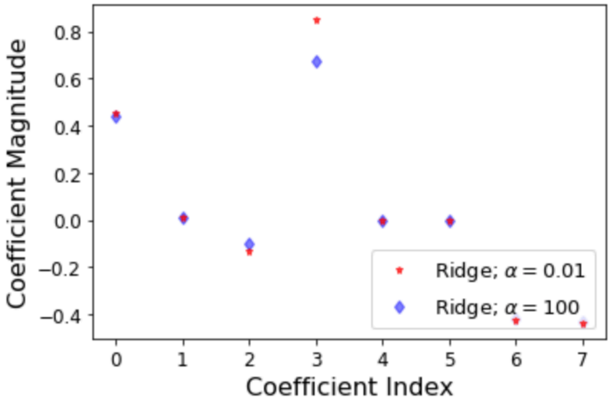


Figure 4

d. *Bayesian Ridge Regression*

After loading the data, I trained the data, then used Bayesian Ridge function and fit the model to obtain the graph below. This graph shows us the predicted value is fitting most of the original values.

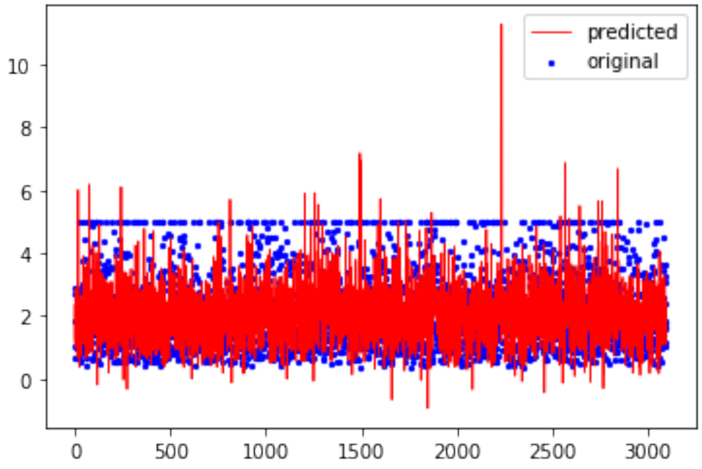


Figure 5: Bayesian ridge reg

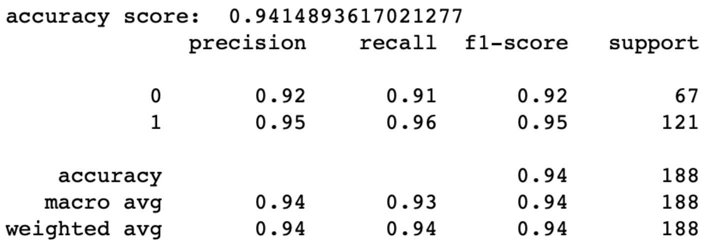
IV. CLASSIFICATION DATASET: BREAST CANCER

a. The Dataset

The Breast Cancer Dataset is second dataset from the small datasets. It is taken from the Breast Cancer Wisconsin Datasets. The dataset contains 569 instances with 30 attributes. The attributes include radius, texture, perimeter, area, and other variables that describe the tumor. I explored two different classification methods to predict if the tumor was Malignant or Benign and compared their accuracies.

b. Bayes Naïve Gaussian Distribution

The first classification algorithm I used was Gaussian Distribution. Before applying the method, I split the data into training and testing variables, then I used the fit the train variables to the GaussianNB function to train the model. Next I made predictions using the test variable. Using the accuracy score method, I saw that this algorithm had 94% accuracy.



```
accuracy score:  0.9414893617021277
              precision    recall  f1-score   support

           0       0.92      0.91      0.92        67
           1       0.95      0.96      0.95       121

    accuracy                           0.94       188
   macro avg       0.94      0.93      0.94       188
weighted avg       0.94      0.94      0.94       188
```

Figure 6:Classification Report for Gaussian Distribution in Breast Cancer Dataset (0:malignant 1:benign)

c. Logistic Regression

The next type of classification method I used was logistic regression. This was one of the techniques we learned in our class. As I did in the previous algorithm, I trained and split the breast cancer data. The important methods I used for this algorithm are fit, score for accuracy and heat map to visual the score. The input parameters for the fit function are x_train and y_train. For the score function, the inputs are x_test, y_test. For the heat map, I used a confusion matrix for an input, which had y_test and predictions as parameters.

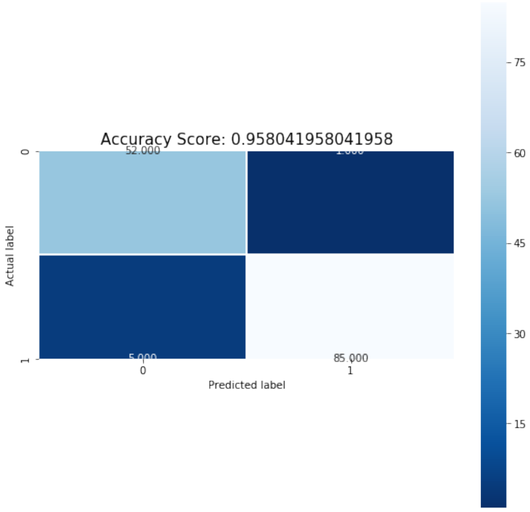


Figure 7: Heat Map to visualize accuracy score

V. CLASSIFICATION DATASET: OLIVETTI FACES

a. The Dataset

The Olivetti faces dataset is from the large datasets. The dataset consists of images of 40 different subjects at different times and facial expressions. The background and frontal position of the face was the same for all subjects. The target of the dataset contains the identity of the person.

b. Visualization

To visualize the dataset, I used subplot method inside of for loops.

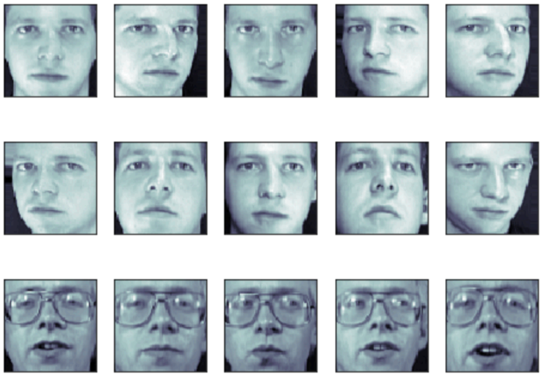
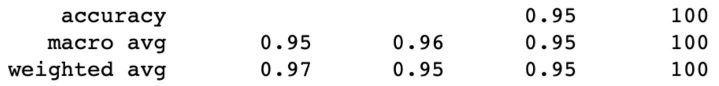


Figure 8: Displaying the types of data in the dataset

c. Support Vector Machines

This classification algorithm was a good to implement on this dataset since it was not very big. To Implement SVM, I used PCA eigen vectors to preprocess the data. PCA was helpful to lower the dimension of dataset given that 1850 dimensions were a lot of SVM algorithm. After training and fitting the model, I used predict and subplot functions to display the results of the SVM classification. The image results showed that they were very close and the classification report tells us that we had 95% accuracy on the dataset.



```
    accuracy                           0.95       100
   macro avg       0.95      0.96      0.95       100
weighted avg       0.97      0.95      0.95       100
```

Figure 9: SVM results for Olivetti Faces

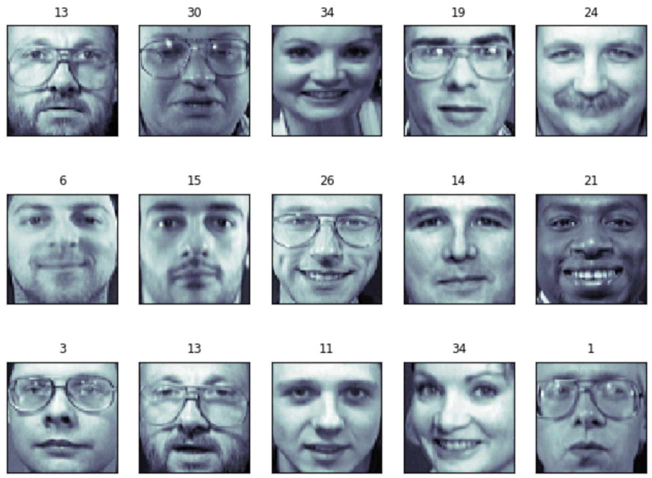
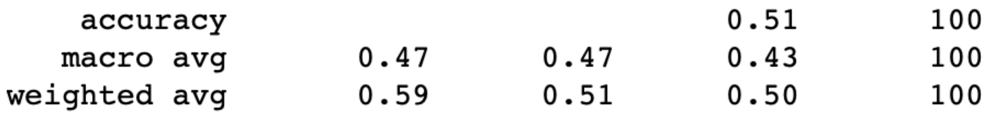


Figure 10: SVM predictions

d. Decision Tree Classifier

Decision Tree Classifier is a powerful algorithm. Using the trained data from previous classification model, I used decision tree algorithm from the sklearn module. It is surprising to see that this algorithm was not very accurate. This could also be because Olivetti dataset is the type of dataset that does not work with this algorithm. Figure 11 shows how the tree outputted after the running it. Figure 12 shows the classification report.

Figure 11: Decision Tree result



```
    accuracy                           0.51       100
   macro avg       0.47      0.47      0.43       100
weighted avg       0.59      0.51      0.50       100
```

Figure 12:Decision Tree Accuracy

VI. FUTURE WORKS

If I had more time with this project, I would learn more about the datasets to choose the best algorithm for the datasets. For example, Decision Tree Classifier wasn’t the best technique for the Olivetti Faces dataset. It had low accuracy score, 47%. Lastly, I would also like to see how the I could implement a neural network for regression and classification on these datasets.