ML Homework Assignment-1

Linear / Logistic Regression

Rajat Talukdar MT20343

Q1

1.1

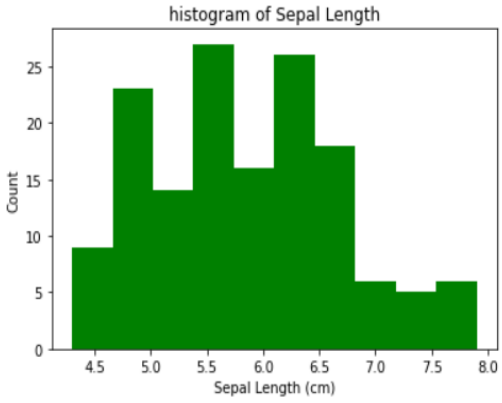
(a) Load the dataset using Pandas library in python.

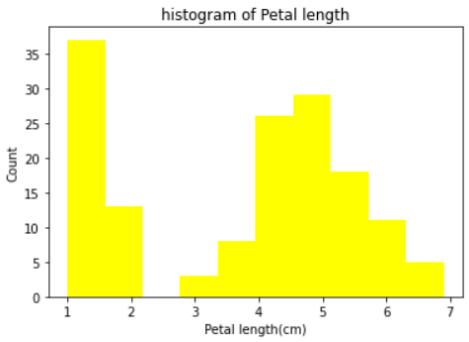
(b) Print column information (name, data types, value range or counts).

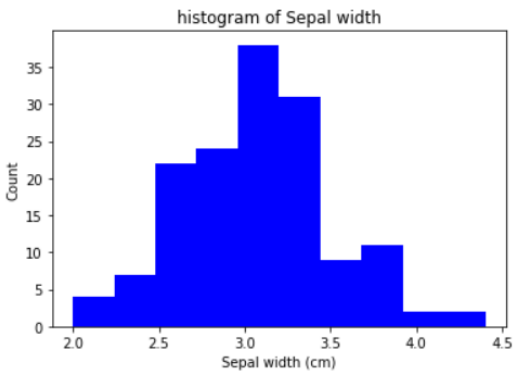
(c) Plot histograms for continuous valued attributes and bar graphs for the discrete valued attributes and the target class.

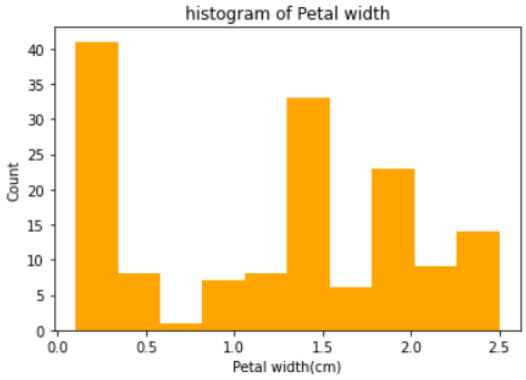
- The above problem is executed in the file Q1.ipynb

Histograms for continuous valued attributes and bar graphs for the discrete valued attributes and the target class.

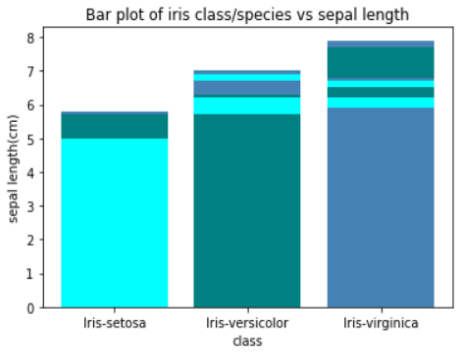


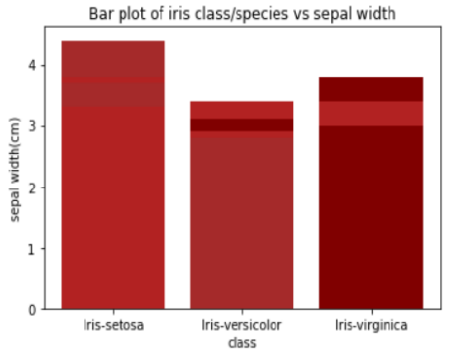


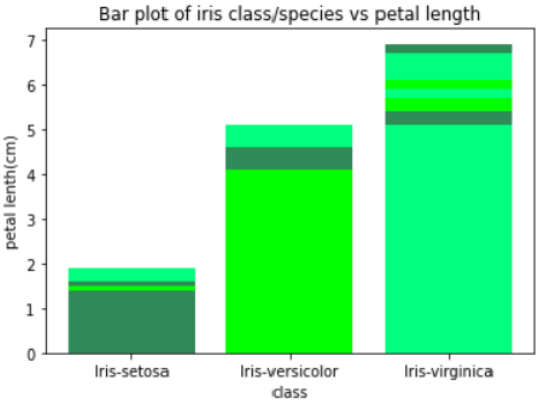


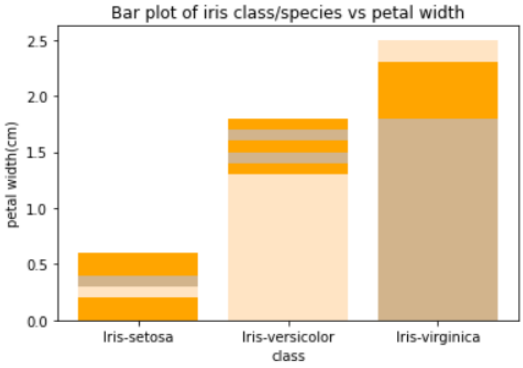


Bar graphs for the discrete valued attributes and the target class.









Q1

1.2

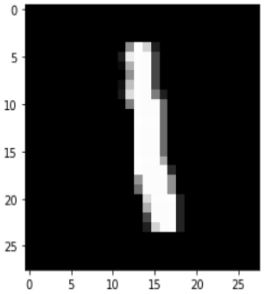
(a) Load the dataset using “idx numpy” package.

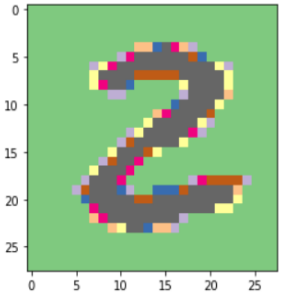
(b) Visualize 2 random images from the dataset.

(c) Use TSNE (t-distributed stochastic neighbour embedding) algorithm to reduce data dimensions to 2, and plot the resulting data as a scatter plot. Comment on the separability of the data.

- The above problem is executed in the file Q1.ipynb

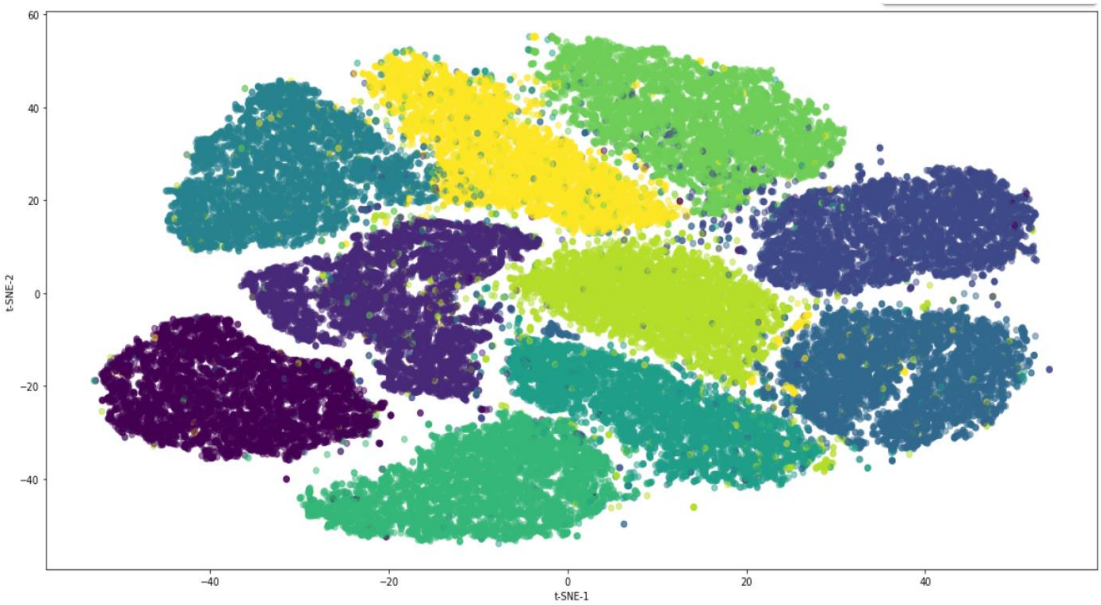
Visualizing 2 random images from the dataset.





Using TSNE (t-distributed stochastic neighbour embedding) algorithm to reduce data dimensions to 2

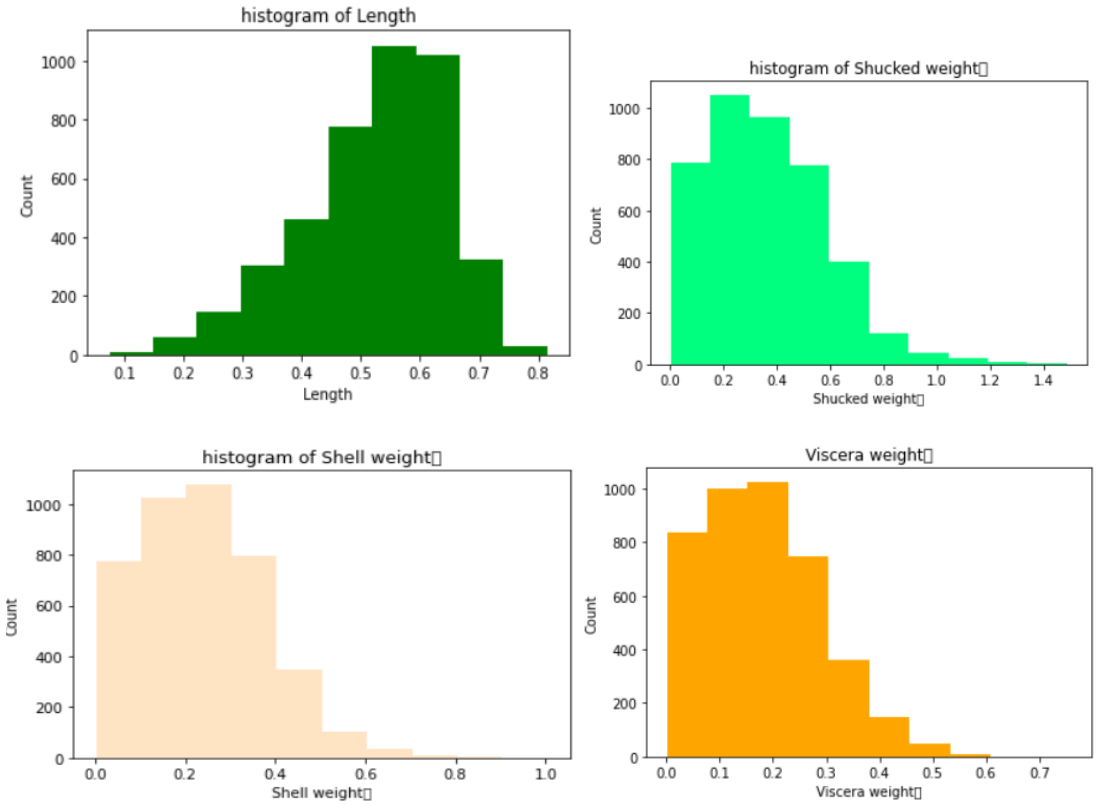
TSNE scatter plot



Q2. Implement Linear Regression for the Abalone Dataset.

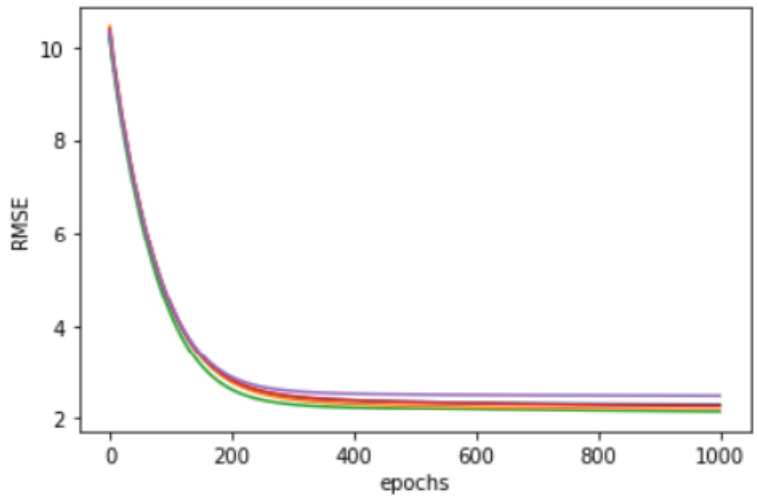
-The above problem is executed in the file Q1.ipynb

Visualizing some attributes via histograms.



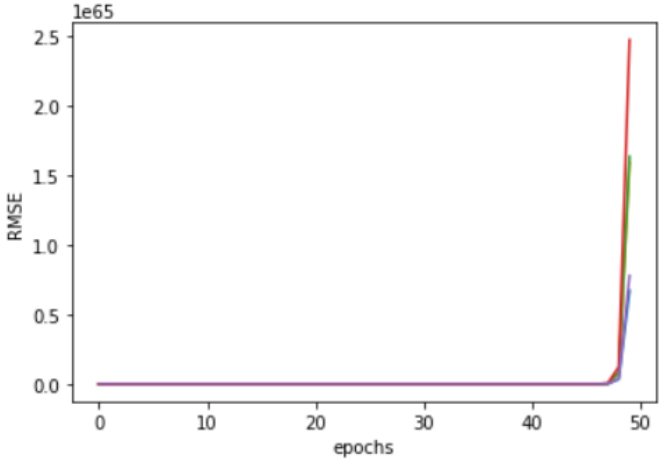
Linear regression from scratch

Iteration vs RMSE graph for all 5 models.

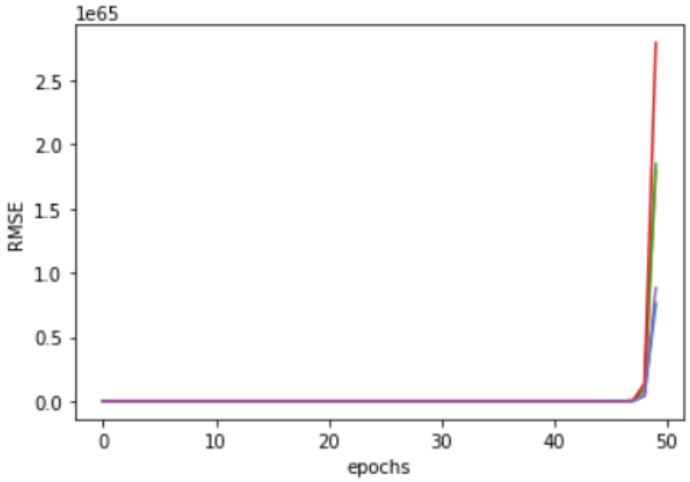


Implimenting L1 (LASSO)and L2 (Ridge Regression) regularization.

Iteration vs RMSE plot for L1 regularization



Iteration vs RMSE plot for L2 regularization



The best performing model was found to be that of only regression among Only regression, Regression+L1, Regression + L2. In the other two the line is not fitting well.

Scikit-learn's implementation of Linear Regression

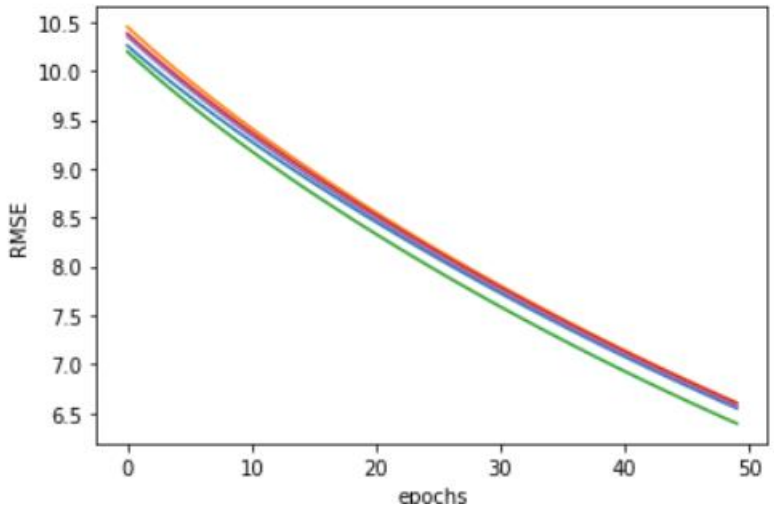
The accuracy on 10% test set for Scikit-learn's implementation was found to be 0.54 which is less in comparison to linear regression model from scratch

```
#comparing the analysis on test set
print(regression.score(testx, testy))
```

```
0.5473971328968752
```

Normal equation (closed form) for linear regression

Iteration vs RMSE plot

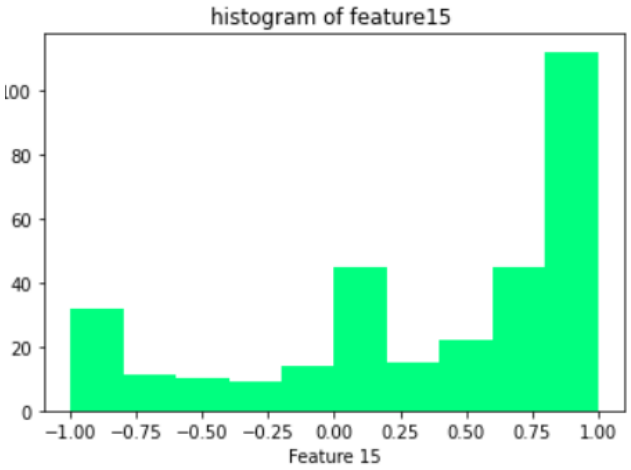


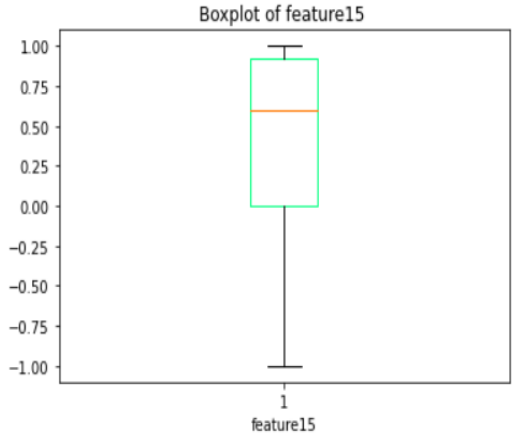
Q3.

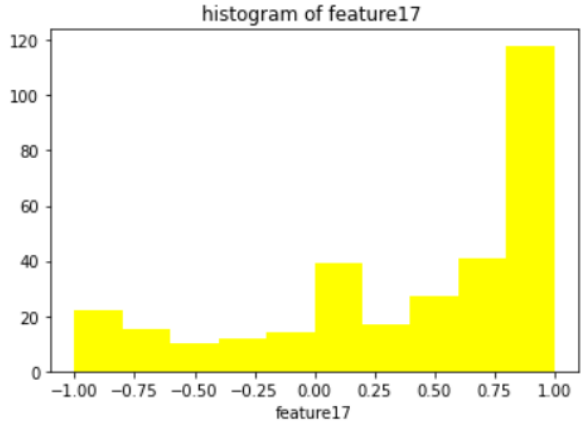
1. Implement Binary Logistic Regression for the UCI Ionosphere

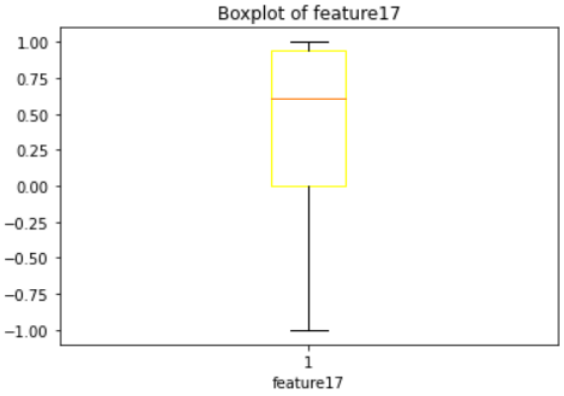
- The above problem is executed in the file Q3.1.ipynb

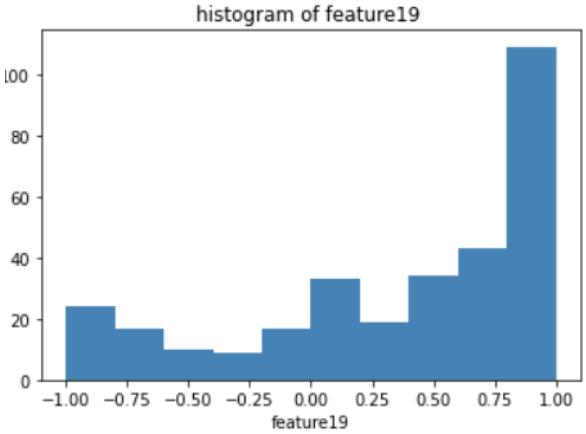
Ploting histograms and box plots for the 5 features with the highest variance.

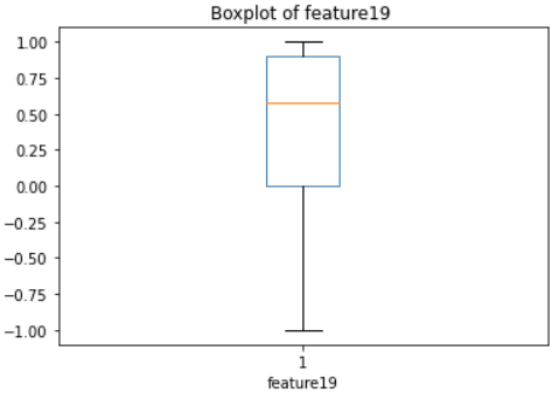


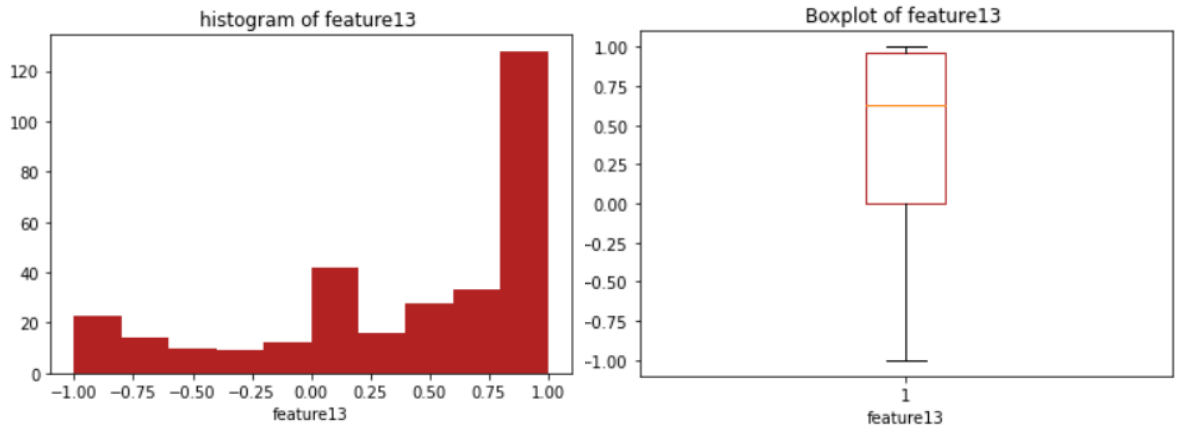


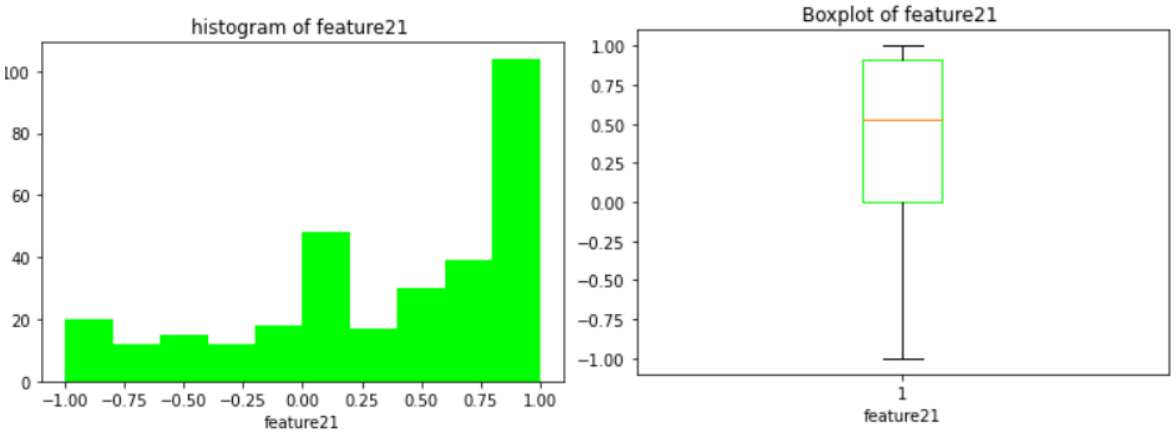












Implementing Binary Logistic Regression for the UCI Ionosphere dataset.

Logistic regression model- accuracy on 10% test data

```
The Accuracy for Test Set is 94.44444444444444
```

	precision	recall	f1-score	support
0	0.92	1.00	0.96	22
1	1.00	0.86	0.92	14
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

After using kfold cross validation with 5 folds.

Following are the classification report of each folds

	precision	recall	f1-score	support
0	0.92	1.00	0.96	22
1	1.00	0.86	0.92	14
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

	precision	recall	f1-score	support
0	0.92	1.00	0.96	22
1	1.00	0.86	0.92	14
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

	precision	recall	f1-score	support
0	0.92	1.00	0.96	22
1	1.00	0.86	0.92	14
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

	precision	recall	f1-score	support
0	0.92	1.00	0.96	22
1	1.00	0.86	0.92	14
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

	precision	recall	f1-score	support
0	0.92	1.00	0.96	22
1	1.00	0.86	0.92	14
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

```
[0.85714286 0.88888889 0.85714286 0.92063492 0.82539683]
```

Reducing the number of features via Principal Component Analysis (PCA)

The accuracy of the logistic regression model after reducing the number of features were found to be

```
The Accuracy for Test Set is 94.44444444444444
```

	precision	recall	f1-score	support
0	0.92	1.00	0.96	22
1	1.00	0.86	0.92	14
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

The model performance with and without PCA is found to be same in this case

Logistic Regression from scikit-learn with L1 and L2 regularization.

```
The Accuracy for Test Set with L1 regularization is 61.111111111111114
The Accuracy for Test Set with L2 regularization is 97.22222222222221
```

Kfold implementation

Classification report for L1 regularization after implementing kfold with 5 folds.

	precision	recall	f1-score	support
0	0.61	1.00	0.76	22
1	0.00	0.00	0.00	14
accuracy			0.61	36
macro avg	0.31	0.50	0.38	36
weighted avg	0.37	0.61	0.46	36

	precision	recall	f1-score	support
0	0.61	1.00	0.76	22
1	0.00	0.00	0.00	14
accuracy			0.61	36
macro avg	0.31	0.50	0.38	36
weighted avg	0.37	0.61	0.46	36

	precision	recall	f1-score	support
0	0.61	1.00	0.76	22
1	0.00	0.00	0.00	14
accuracy			0.61	36
macro avg	0.31	0.50	0.38	36
weighted avg	0.37	0.61	0.46	36

	precision	recall	f1-score	support
0	0.61	1.00	0.76	22
1	0.00	0.00	0.00	14
accuracy			0.61	36
macro avg	0.31	0.50	0.38	36
weighted avg	0.37	0.61	0.46	36

	precision	recall	f1-score	support
0	0.61	1.00	0.76	22
1	0.00	0.00	0.00	14
accuracy			0.61	36
macro avg	0.31	0.50	0.38	36
weighted avg	0.37	0.61	0.46	36

Classification report for L1 regularization after implementing kfold with 5 folds.

	precision	recall	f1-score	support
0	0.96	1.00	0.98	22
1	1.00	0.93	0.96	14
accuracy			0.97	36
macro avg	0.98	0.96	0.97	36
weighted avg	0.97	0.97	0.97	36

	precision	recall	f1-score	support
0	0.96	1.00	0.98	22
1	1.00	0.93	0.96	14
accuracy			0.97	36
macro avg	0.98	0.96	0.97	36
weighted avg	0.97	0.97	0.97	36

	precision	recall	f1-score	support
0	0.96	1.00	0.98	22
1	1.00	0.93	0.96	14
accuracy			0.97	36
macro avg	0.98	0.96	0.97	36
weighted avg	0.97	0.97	0.97	36

	precision	recall	f1-score	support
0	0.96	1.00	0.98	22
1	1.00	0.93	0.96	14
accuracy			0.97	36
macro avg	0.98	0.96	0.97	36
weighted avg	0.97	0.97	0.97	36

	precision	recall	f1-score	support
0	0.96	1.00	0.98	22
1	1.00	0.93	0.96	14
accuracy			0.97	36
macro avg	0.98	0.96	0.97	36
weighted avg	0.97	0.97	0.97	36

Comparing model performance on test set with and without L1, L2

Without regularization

	precision	recall	f1-score	support
0	0.92	1.00	0.96	22
1	1.00	0.86	0.92	14
accuracy			0.94	36
macro avg	0.96	0.93	0.94	36
weighted avg	0.95	0.94	0.94	36

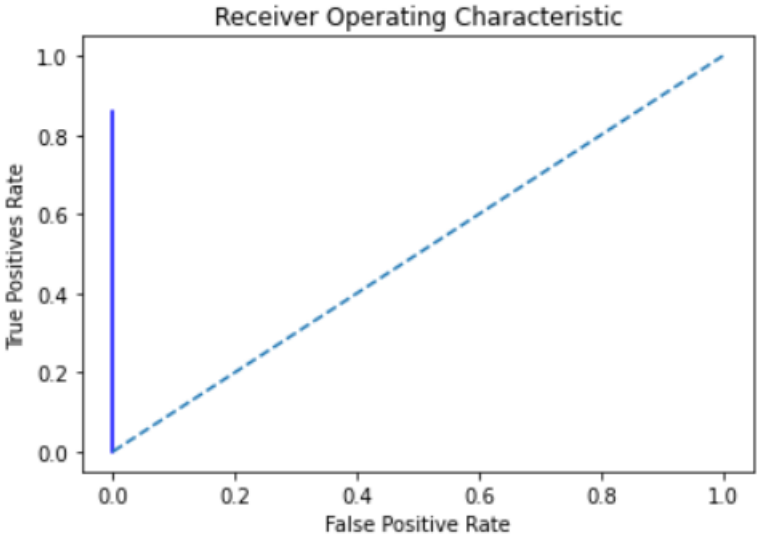
With L1 regularization

	precision	recall	f1-score	support
0	0.61	1.00	0.76	22
1	0.00	0.00	0.00	14
accuracy			0.61	36
macro avg	0.31	0.50	0.38	36
weighted avg	0.37	0.61	0.46	36

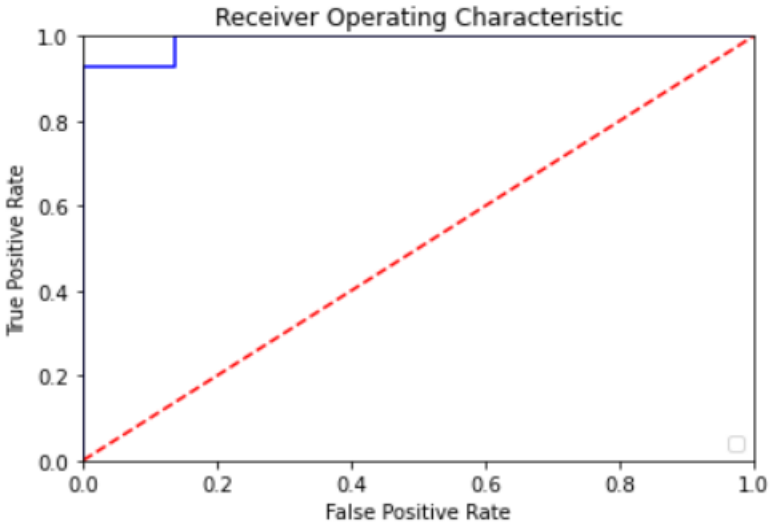
With L2 regularization

	precision	recall	f1-score	support
0	0.96	1.00	0.98	22
1	1.00	0.93	0.96	14
accuracy			0.97	36
macro avg	0.98	0.96	0.97	36
weighted avg	0.97	0.97	0.97	36

ROC-AUC curve



ROC-AUC curve using scikit-learn



The difference is between the values of threshold for ROC-AUC curve and ROC-AUC curve in scikit-learn

Q3.

2. Implementing Multiclass Logistic Regression for the MNIST dataset OVO and OVR

-The above problem is executed in the file Q3.2.ipynb

Classification report L2 Logistic regression OVO

	precision	recall	f1-score	support
0	0.97	0.96	0.97	980
1	0.98	0.97	0.98	1135
2	0.92	0.91	0.91	1032
3	0.90	0.92	0.91	1010
4	0.94	0.94	0.94	982
5	0.87	0.88	0.87	892
6	0.96	0.94	0.95	958
7	0.95	0.92	0.93	1028
8	0.86	0.90	0.88	974
9	0.90	0.91	0.91	1009
accuracy			0.93	10000
macro avg	0.93	0.92	0.92	10000
weighted avg	0.93	0.93	0.93	10000

```
The Accuracy for Test Set is 92.57
```

Classification report L2 Logistic regression OVR

	precision	recall	f1-score	support
0	0.97	0.96	0.97	980
1	0.98	0.97	0.98	1135
2	0.92	0.91	0.91	1032
3	0.90	0.92	0.91	1010
4	0.94	0.94	0.94	982
5	0.87	0.88	0.87	892
6	0.96	0.94	0.95	958
7	0.95	0.92	0.93	1028
8	0.86	0.90	0.88	974
9	0.90	0.91	0.91	1009
accuracy			0.93	10000
macro avg	0.93	0.92	0.92	10000
weighted avg	0.93	0.93	0.93	10000

```
The Accuracy for Test Set is 92.57
```

Classification report Simple Logistic regression OVO

	precision	recall	f1-score	support
0	0.97	0.96	0.97	980
1	0.98	0.97	0.98	1135
2	0.92	0.91	0.91	1032
3	0.90	0.92	0.91	1010
4	0.94	0.94	0.94	982
5	0.87	0.88	0.87	892
6	0.96	0.94	0.95	958
7	0.95	0.92	0.93	1028
8	0.86	0.90	0.88	974
9	0.90	0.91	0.91	1009
accuracy			0.93	10000
macro avg	0.93	0.92	0.92	10000
weighted avg	0.93	0.93	0.93	10000

```
The Accuracy for Test Set is 92.57
```

Classification report Simple Logistic regression OVR

	precision	recall	f1-score	support
0	0.97	0.96	0.97	980
1	0.98	0.97	0.98	1135
2	0.92	0.91	0.91	1032
3	0.90	0.92	0.91	1010
4	0.94	0.94	0.94	982
5	0.87	0.88	0.87	892
6	0.96	0.94	0.95	958
7	0.95	0.92	0.93	1028
8	0.86	0.90	0.88	974
9	0.90	0.91	0.91	1009
accuracy			0.93	10000
macro avg	0.93	0.92	0.92	10000
weighted avg	0.93	0.93	0.93	10000

```
The Accuracy for Test Set is 92.57
```

The accuracy of OVO , OVR with and without L2 regularization are found to be same

THEORY QUESTIONS

4.1 Derive the closed form solution to the linear regression problem for the dataset $D = \{(x_1, y_2), (x_2, y_2) \ldots, (x_N, y_N)\}$, where $x_i \in R^d$, and $y_i \in R$, for $i = 1, 2, \ldots, N$. Let $y = X\theta + \epsilon$ be the regression model, where y is an $N \times 1$ vector constructed by concatenating the target variables y_i, $i = 1, \ldots, N$, and the matrix $X_{N \times d} = [x_1, x_2, \ldots x_N]^T$ contains the input data vectors. The regression parameters θ needs to be estimated

→ Given, dataset $D = \{(x_1, y_1), (x_2, y_2) \ldots\ldots, (x_N, y_N)\}$

where $x_i \in R^d$, $y_i \in R$, for $i = 1, 2, \ldots N$

and linear regression model

$$y = X\theta + \epsilon$$

'X' is a design matrix of dimension $N \times d$

$$X_{N \times d} = [x_1, x_2, \ldots x_N]^T$$

$$X = \begin{bmatrix} -(x^{(1)})^T- \\ -(x^{(2)})^T- \\ \vdots \\ -(x^{(N)})^T- \end{bmatrix}$$

'y' vector contains the target variables

$$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} \quad y_i, i = 1, \ldots N.$$

Since $h_\theta(x^{(i)}) = (x^{(i)})^T\theta$

(predicted output)

And $y = X\theta + \epsilon$ (where 'θ' is the parameter and 'ϵ' is the error term)

we can verify

$$X\theta - \vec{y} = \begin{bmatrix} (x^{(1)})^T\theta \\ \vdots \\ x^{(N)})^T\theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

$$= \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \vdots \\ h_\theta(x^{(N)}) - y^{(N)} \end{bmatrix}$$

for any vector z, z^Tz can be written as $z^Tz = \sum_i z_i^2$

$$\frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) = \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= J(\theta)$$

Finding derivative of J w.r.t to θ, to minimize J

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

$$= \frac{1}{2}\nabla_\theta((X\theta)^TX\theta - (X\theta)^T\vec{y} - \vec{y}^T(X\theta) + \vec{y}^T\vec{y})$$

$$= \frac{1}{2} \nabla_\theta \left(\theta^T (X^T X) \theta - \vec{y}^T (X\theta) - \vec{y}^T (X\theta) \right)$$

$$= \frac{1}{2} \nabla_\theta \left(\theta^T (X^T X) \theta - 2 (X^T \vec{y})^T \theta \right)$$

$$= \frac{1}{2} \left(2 X^T X \theta - 2 X^T \vec{y} \right)$$

$$= X^T X \theta - X^T \vec{y}$$

In order to minimise J, its derivative is set to zero, to obtain the normal equation.

~~$X^T X \theta = X^T \vec{y}$~~

$$\nabla_\theta J(\theta) = X^T X \theta - X^T \vec{y}$$

$$0 = X^T X \theta - X^T \vec{y}$$

$$X^T X \theta = X^T \vec{y}$$

Therefore, the value of θ which minimizes $J(\theta)$ in closed form is given by the equation.

$$\boxed{\theta = (X^T X)^{-1} X^T \vec{y}}$$

2. Write the conditions under which the closed form solution to Linear Regression exists.

→ The conditions under which the closed form solution to linear regression exists are.

- Assuming X^TX to be invertible or positive definitive make sures the critical point to be minimum.
- Number of datapoint in dataset N is larger than the dimensionality K of the input space, the matrix X to be of full column rank.

3. If we have a closed form solution for Linear Regression, why do we use Gradient Descent? Give an example of a situation where Gradient Descent is a better option than closed form calculations.

→ Gradient descent is used for linear regression as it is more computationaly viable, as the complexity increases such as features of dataset, gradient descent is more likely to find the ~~co~~ solution ~~toriory~~ by minimizing the cost function iteritavely moving towards the direction of the steepest.

Gradient descent has linear time complexity so in a case where we have lot of features that is difficult to work on, Gradient descent is a better option in that case.

4. Prove that for simple linear regression, the least square fit line always passes through the point $(\vec{x}, \vec{y})$, where $\vec{x}$ and $\vec{y}$ represent the airithmetic mean of the independent variables and dependent variables respectively.

Let y_i be the predicted value

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{\nu}_i$$

where x_i is the independent value

for n number of values that is

$$i = 1, 2, 3 \cdots n$$

Summating y_i can be written as

$$y_1 + y_2 + \cdots + y_n = (\hat{\beta}_0 + \hat{\beta}_0 + \cdots + \hat{\beta}_n) + (\hat{\beta}_1 x + \hat{\beta}_1 x_2 + \cdots + \hat{\beta}_1 x_n) + (\hat{\nu}_1 + \hat{\nu}_2 + \cdots + \hat{\nu}_n)$$

$$\sum_{i=1}^{n} y_i = n \cdot \hat{\beta}_0 + \hat{\beta}_i \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} \hat{u}_i$$

Considering the condition of optimality

$$\sum_{i=1}^{n} \hat{u}_i = 0$$

substituting this in above equation

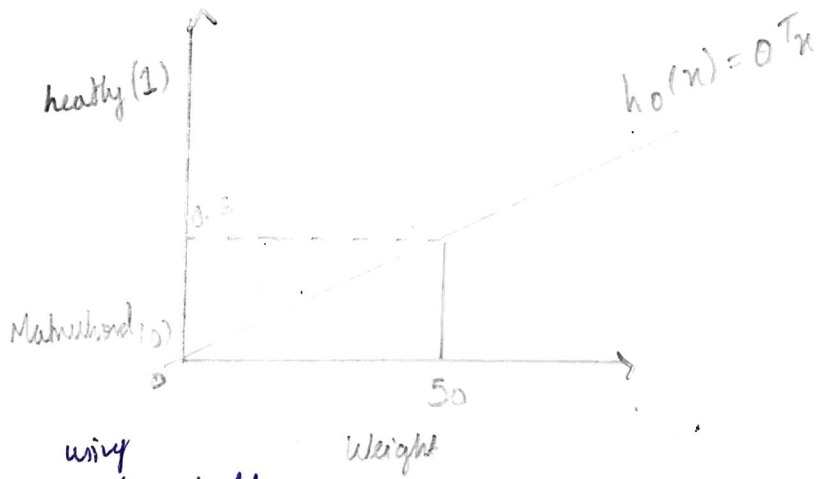
$$\sum_{i=1}^{n} y_i = n \cdot \hat{\beta}_0 + \hat{\beta}_i \sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} \frac{y_i}{n} = \frac{n \cdot \hat{\beta}_0}{n} + \frac{\hat{\beta}_i \sum_{i=1}^{n} x_i}{n}$$

$$\boxed{\bar{y} = \hat{\beta}_0 + \hat{\beta}_i \cdot \bar{x}}$$

5 Can we use Linear regression for classification? If yes, how?

→ Linear regression can be used for classification problem, If we consider persons weight below 50 kg to be malnutritioned (0) and above 50 to be healthy (1), Now if we ~~used~~ use Linear regression for this kind of ~~data~~ binary classification. by giving a threshold, and fitting a line based on the threshold for classification.



Classifying using Threshold

- $h_\theta(x) \geq 0.5$

When $h_\theta(x) \geq 0.5$ $y = 1$

$h_\theta(x) < 0.5$ $y = 0$

THEORY QUESTIONS (PG)

5. Which of the below expressions are linear regression models? Justify.

1. $\theta_0 x_0 + \theta_1 x_1 + \ldots\ldots + \theta_n x_n = 0$
2. $\theta_0 \sin(x_0) + \theta_1 \sin(x_1) + \ldots + \theta_n \sin(x_n) = 0$
3. $\sin(\theta_0 x_0) + \sin(\theta_1 x_1) + \ldots\ldots + \sin(\theta_n x_n) = 0$
4. $y_i = w_0 + \sum_{j=1}^{N} w_j \sinh(x_{ij})$

1. $\theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n = 0$

This expression is a linear regression model. as it shows linear relationship and forms a straight line

2. $\theta_0 \sin(x_0) + \theta_1 \sin(x_1) + \ldots\ldots + \theta_n \sin(x_n) = 0$

This expression is a linear regression model as it can fit in straight line and shows linear relationship between ~~da~~ along with the parameters.

3. $\sin(\theta_0 x_0) + \sin(\theta_1 x_1) + \cdots + \sin(\theta_n x_n) = 0$

This expression is not a linear regression model as it cannot be fitted in a line and the parameter 'θ' along with independent variable 'x' doesnot show linear relationship.

4. $y_i = \omega_0 + \sum_{j=1}^{N} \omega_j \sinh(x_{ij})$

This expression is a linear regression model as the regression line can be fitted, since the parameters along with the independent variable 'x' and dependent variable 'y' shows linear relationship.

6 Logistic Regression

1. What is the Loss function used to train a logistic Regression model. Write the expression for the probabilistic model used and the mathematical expression for the Loss function.

→ The Loss function of Logistic regression is Logistic Loss

$$\text{Loss}\{h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y) \log(h_\theta(x))$$

The expression for the probabilistic model

$$P(y|x;\theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

The mathematical expression for the Loss function

$$l(\theta) = \log L(\theta)$$

$$= \sum_{i=1}^{n} y^{(i)} \log\{h(x^{(i)}) + (1-y^{(i)}) \log(1-h(x^{(i)}))$$

2. Modify the above expression to include

(a) Gaussain

(b) Laplacian (doubly exponential) regularization

(a) $$P(y|x;\theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(\theta^T x - y)^2}{2}\right)$$

(b) for Laplacian regularization

$$P(y|x;\theta) = \frac{1}{2} \exp\left(-|\theta^T x - y|\right)$$