ML Assignment1 Report

Q1)

1) IRIS Dataset

	sepal length	sepal width	petal length	petal width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

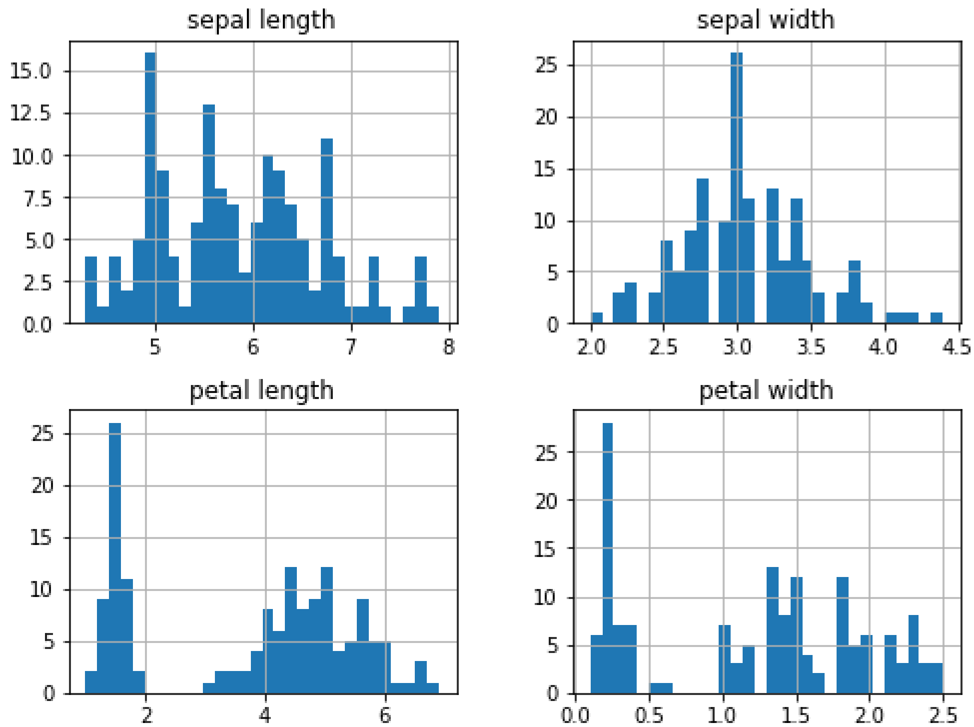
150 rows × 5 columns

Column info**-**

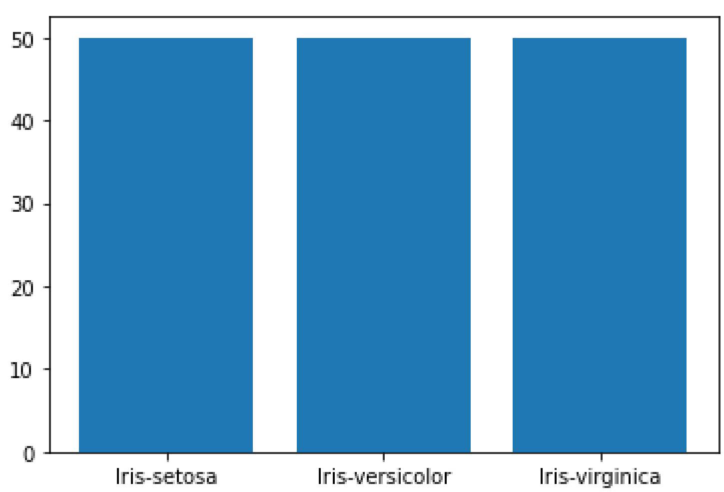
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal length  150 non-null    float64
 1   sepal width   150 non-null    float64
 2   petal length  150 non-null    float64
 3   petal width   150 non-null    float64
 4   class         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

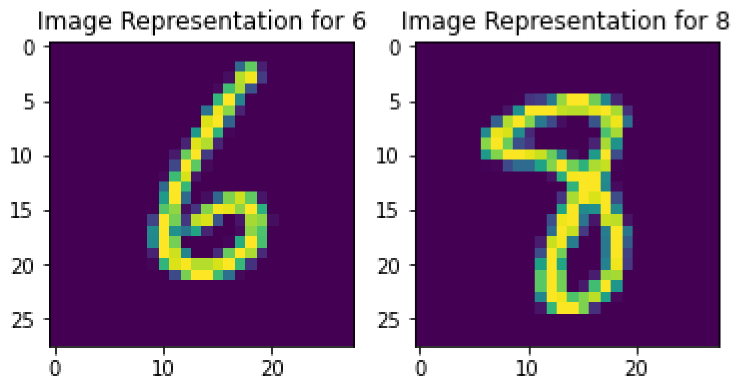
Histogram for attributes -



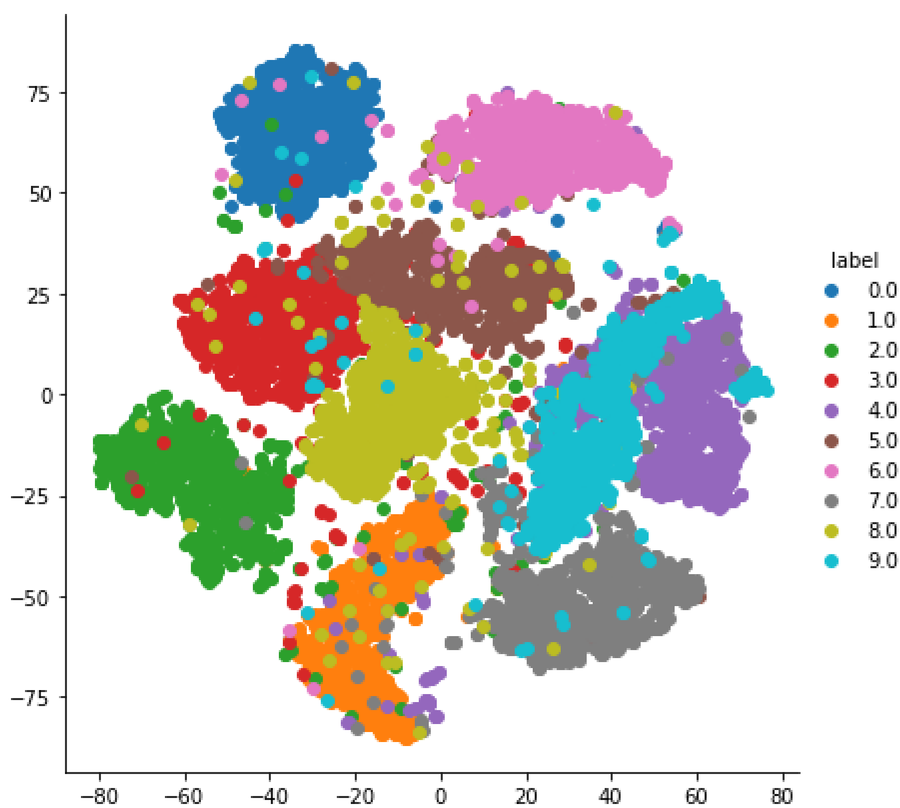
Bar Graph for class attribute -



2) Image visualization of MNIST dataset



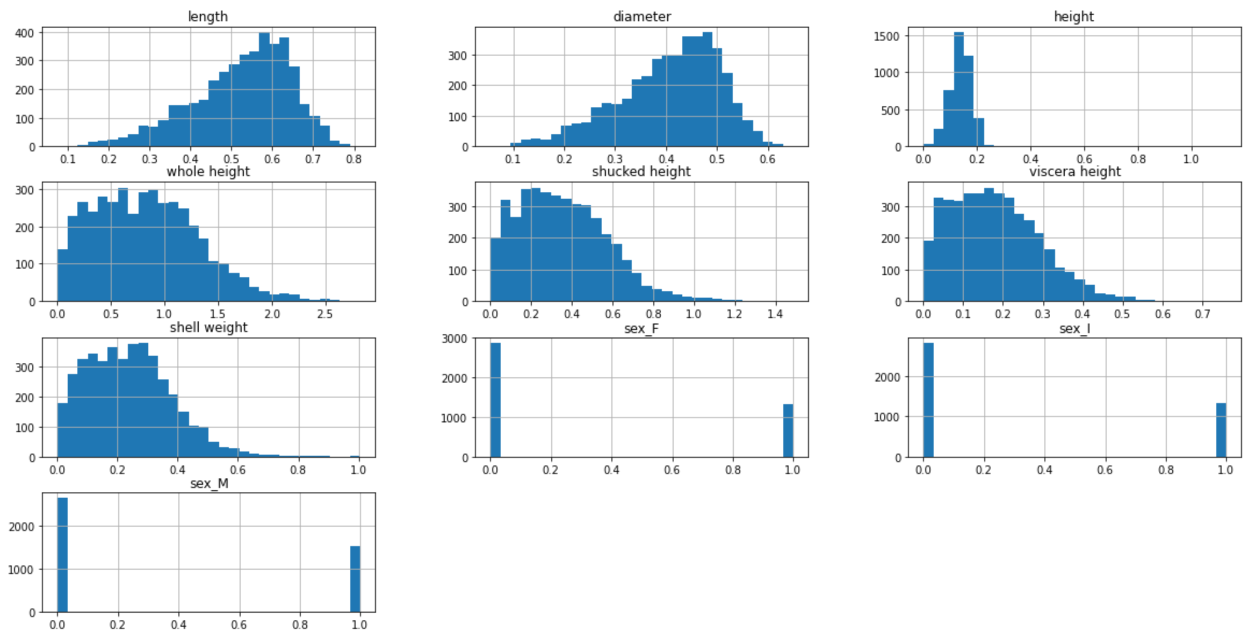
PLot after TSNE-



Here, the scatter plot shows different clusters of the data and we can we can see the labels(0-9 numbers) which look similar to each other have their clusters closer to each other in comparison to the labels that look distinct .

Q2)

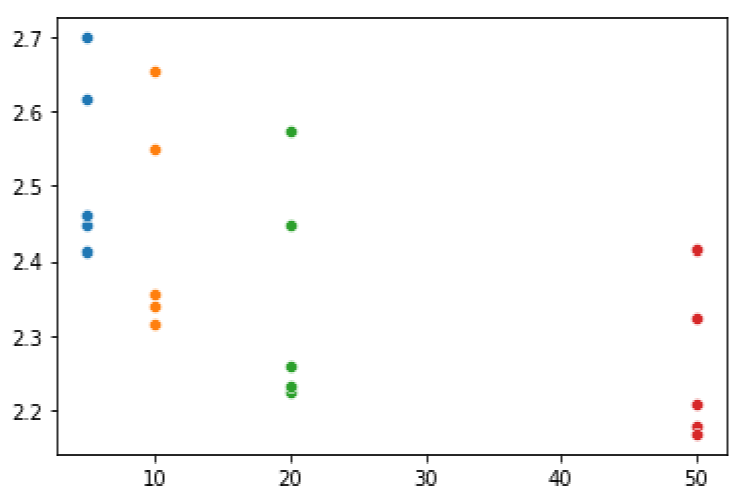
1) Feature visualization -



a)

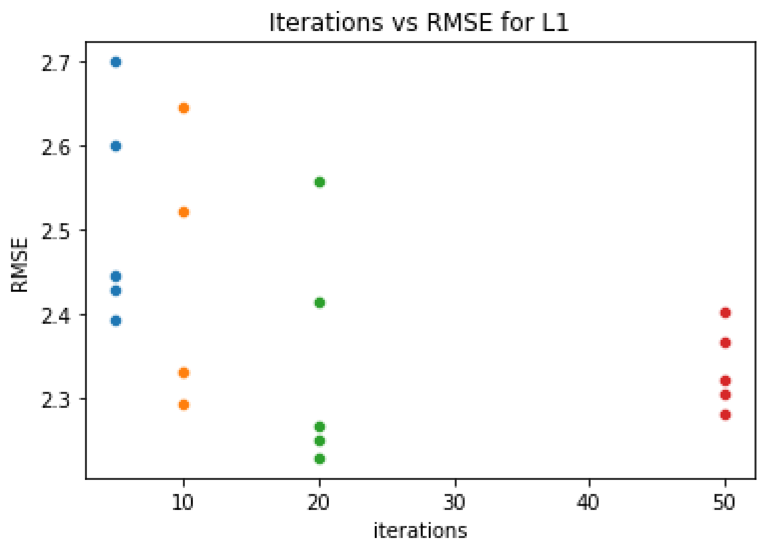
On checking for different learning rate value I have chosen it as 0.001

Iteration vs RMSE plot for Logistic Regression -

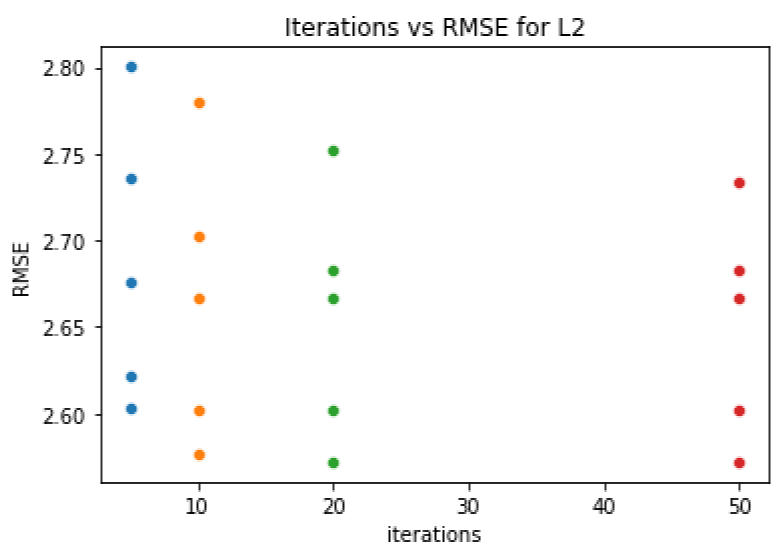


b)

Iteration vs RMSE plot for Logistic Regression with Lasso - lembda= 0.01

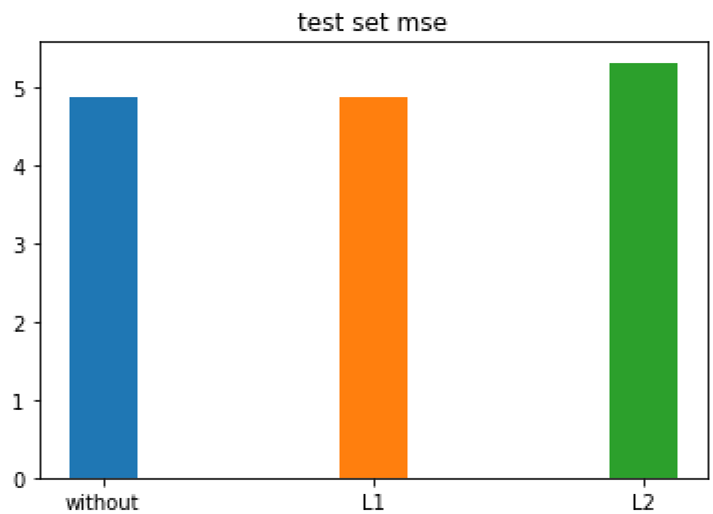


Iteration vs RMSE plot for Logistic Regression with L2 - lembda= 0.01

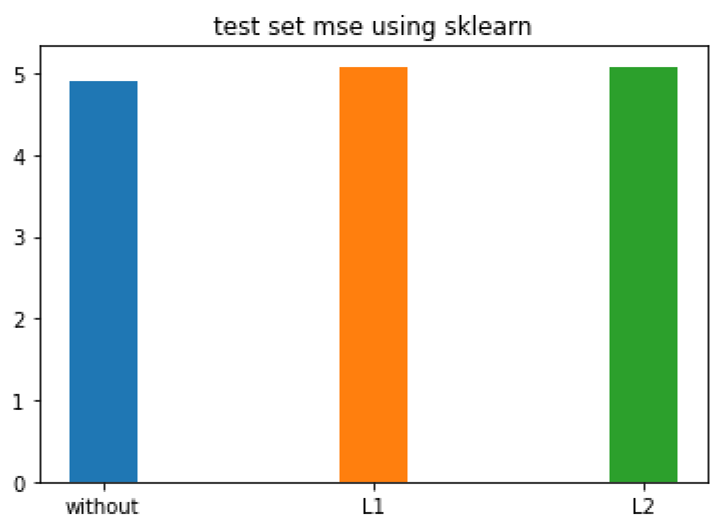


From all above graphs we can conclude that RMSE values decreases with increased iterations. And RMSE values on val set are greater with regularization

c) test set accuracies for all 3 above implemented models -



d) test set accuracies for all 3 models implemented with sklearn -



We can see here that sklearn implementation has given less test accuracies for all 3 models than my model

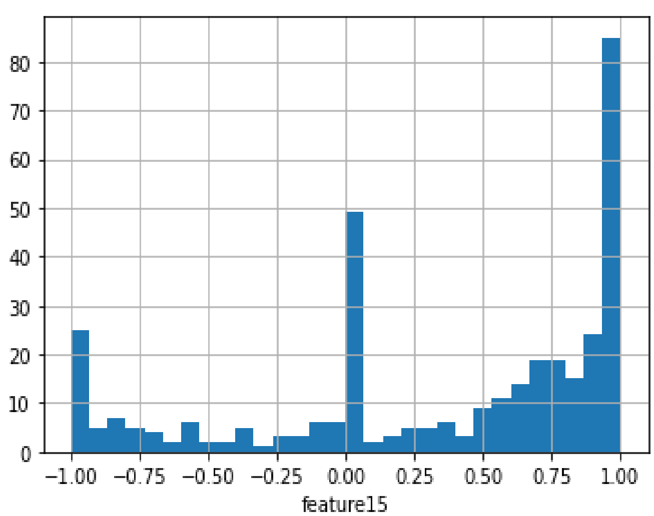
e) validation set accuracy for models with closed form of Linear Regression-

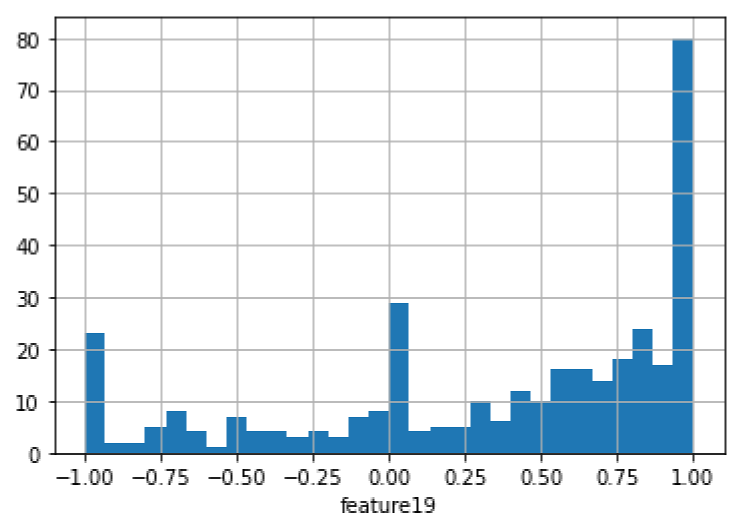
```
mse for model 1  =  5.0040500253146725
mse for model 2  =  5.018084909369004
mse for model 3  =  4.783313929136972
mse for model 4  =  5.252756282087939
mse for model 5  =  4.608845587491214
```

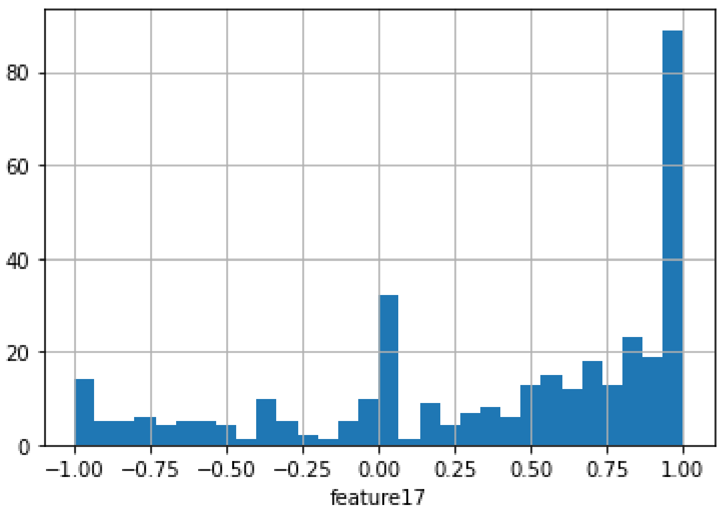
Here we can see that mean squared error values for model with closed form and gradient descent have similar values which can be due to small number of features and small training dataset size

Q3)

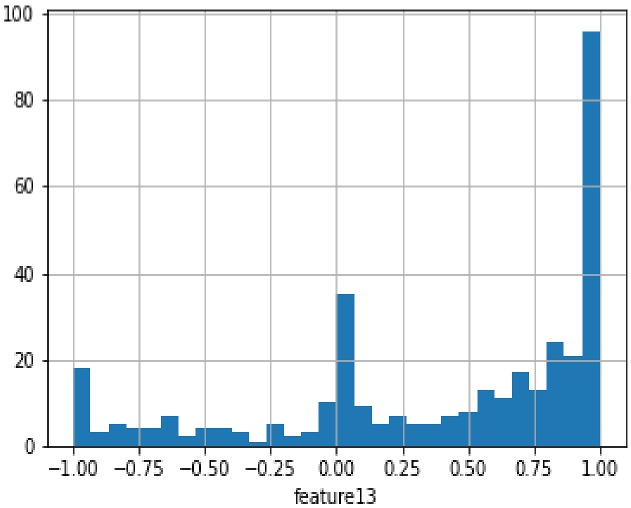
1) Histogram for top 5 features with highest variances -



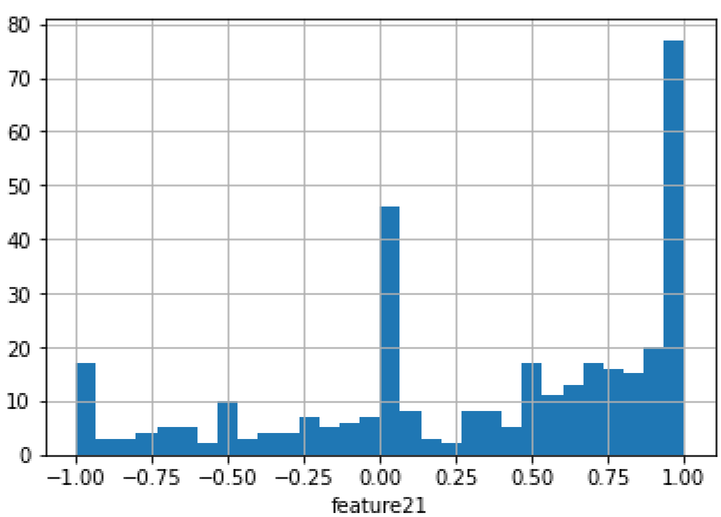




0
20
40
60
80
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
feature17

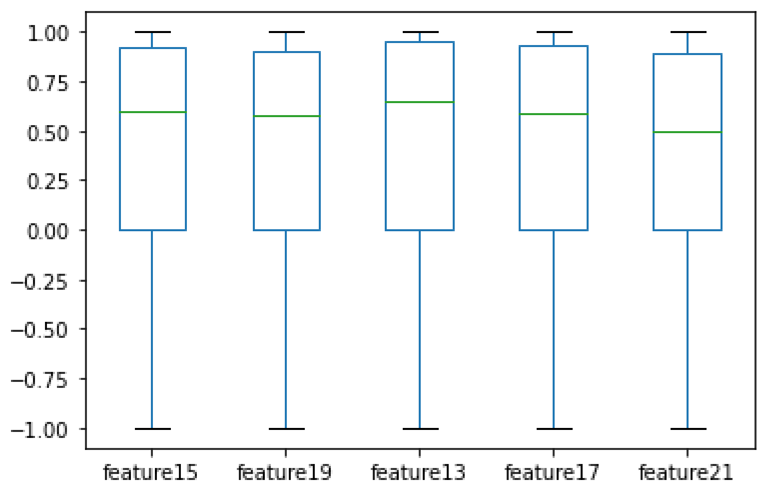


0
20
40
60
80
100
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
feature13

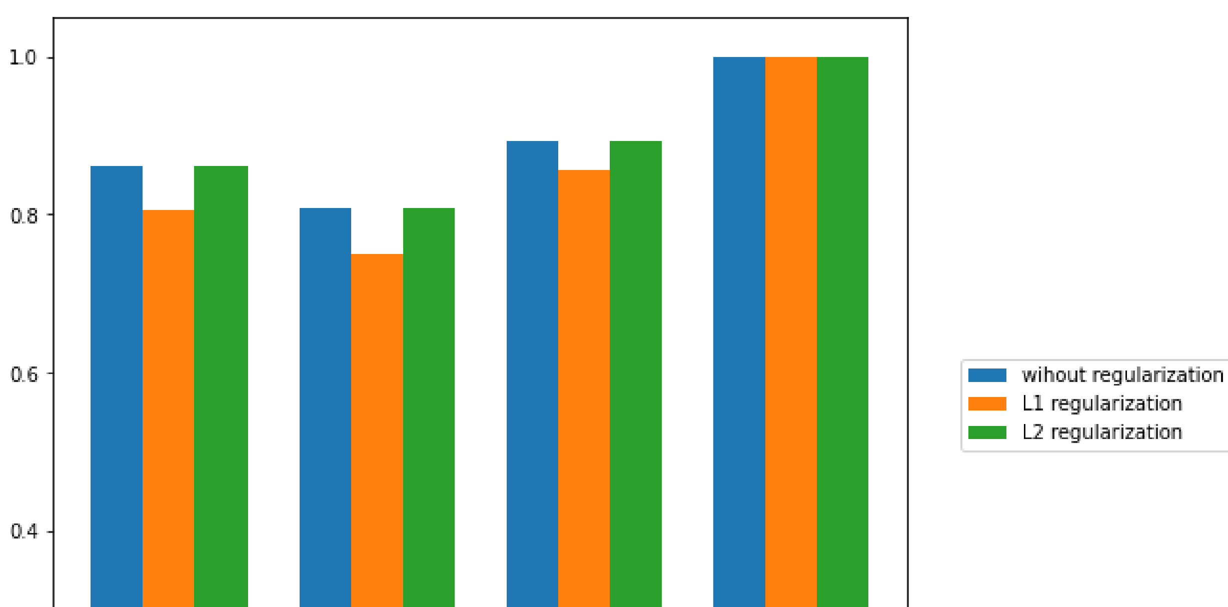


0
10
20
30
40
50
60
70
80
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
feature21

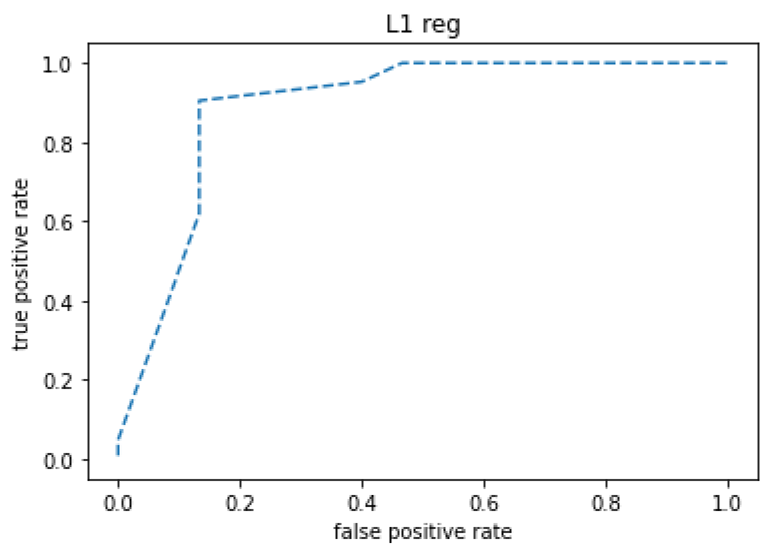
Boxplot for those same features -

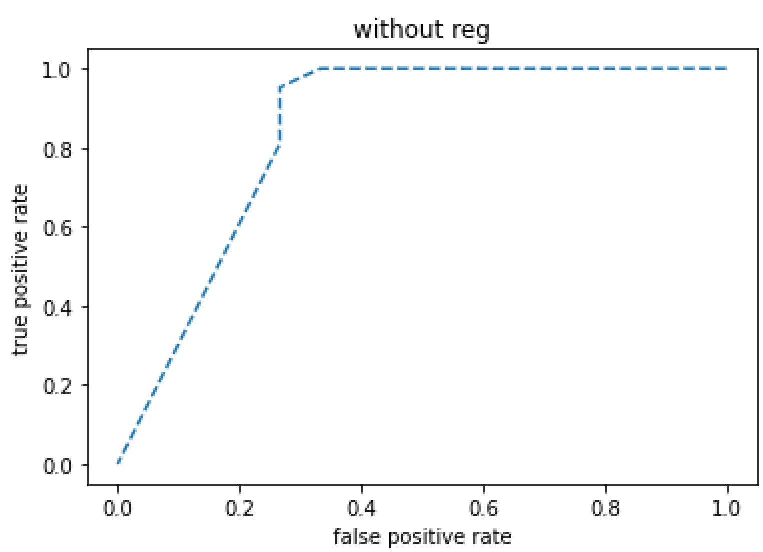


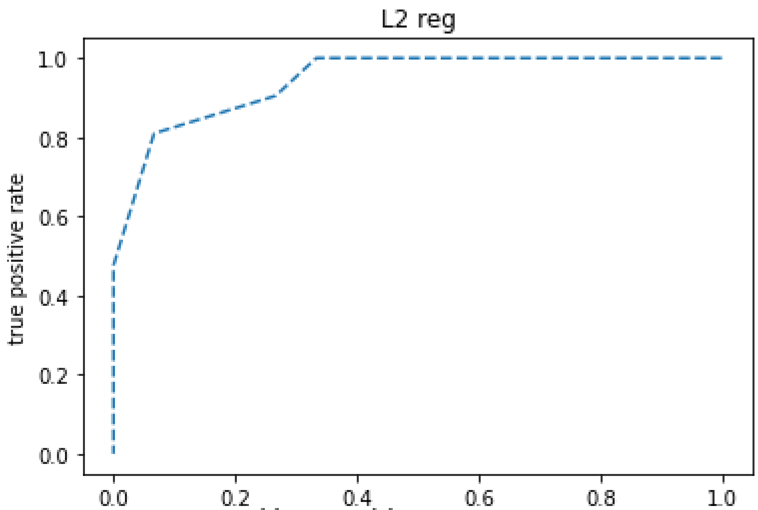
b)



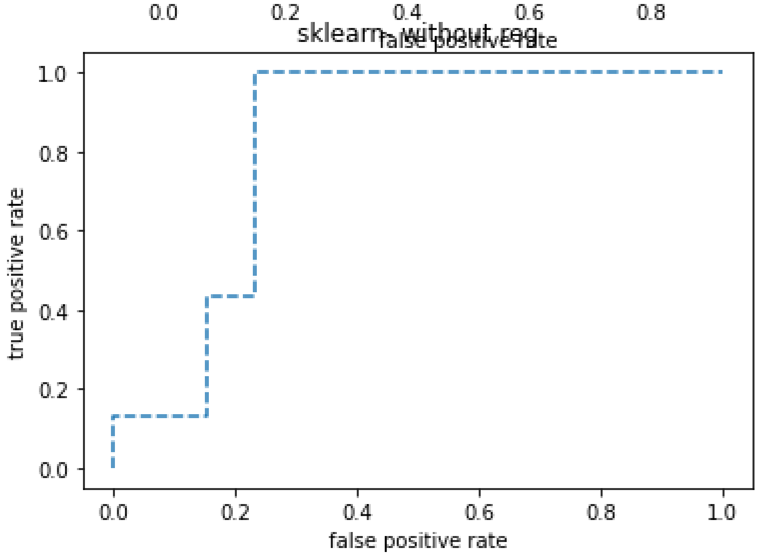
c)

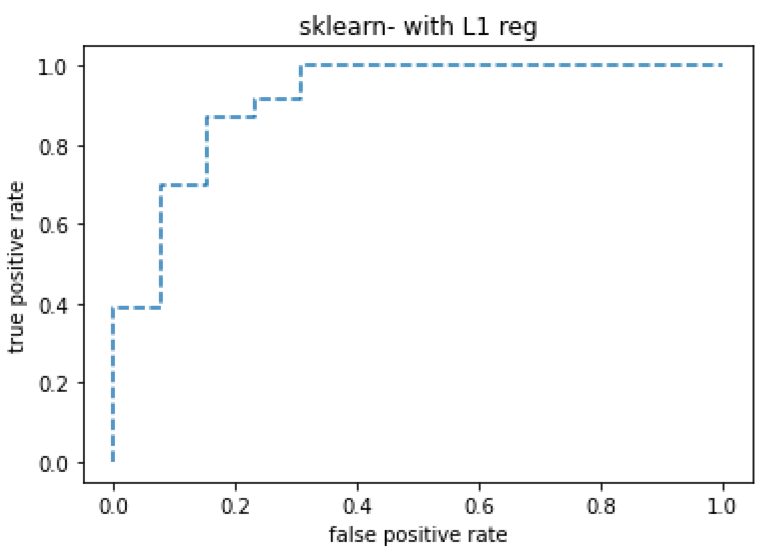

L1 reg
true positive rate
false positive rate
0.0
0.2
0.4
0.6
0.8
1.0

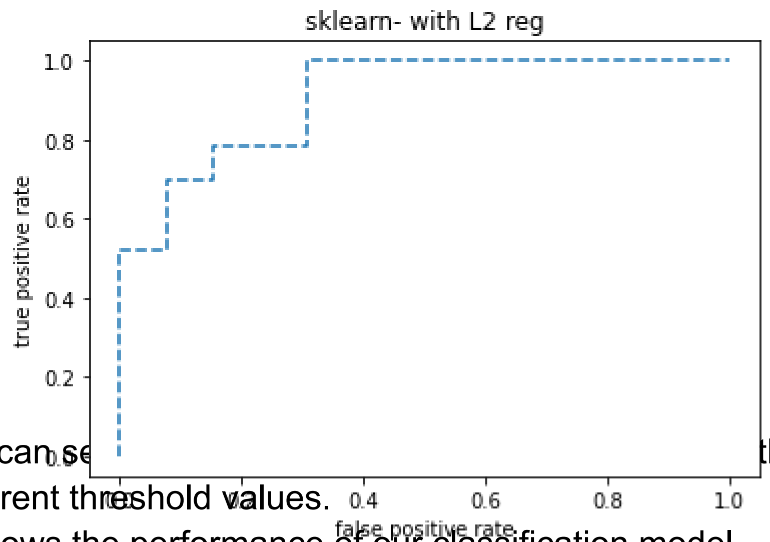

without reg
true positive rate
false positive rate
0.0
0.2
0.4
0.6
0.8
1.0


L2 reg
true positive rate
false positive rate

d)


sklearn- without reg
true positive rate
false positive rate


sklearn- with L1 reg
true positive rate
false positive rate



We can se ... th false positive rate for different threshold values.
It shows the performance of our classification model

2)

a)

Metrics for OVO Logistic Regression without Regularization-

OVO Logistic Regression without regularization

	precision	recall	f1-score	support
0	0.9596	0.9684	0.9639	980
1	0.9763	0.9797	0.9780	1135
2	0.9175	0.9157	0.9166	1032
3	0.9127	0.9208	0.9167	1010
4	0.9255	0.9491	0.9372	982
5	0.8950	0.8700	0.8823	892
6	0.9483	0.9374	0.9428	958
7	0.9390	0.9280	0.9335	1028
8	0.8761	0.9076	0.8916	974
9	0.9304	0.9009	0.9154	1009
accuracy			0.9289	10000
macro avg	0.9280	0.9278	0.9278	10000
weighted avg	0.9290	0.9289	0.9289	10000

Metrics for OVO Logistic Regression with L2 Regularization-

OVO Logistic Regression with L2 regularization

	precision	recall	f1-score	support
0	0.9602	0.9837	0.9718	980
1	0.9588	0.9833	0.9709	1135
2	0.9405	0.9041	0.9219	1032
3	0.9110	0.9218	0.9163	1010
4	0.9225	0.9450	0.9336	982
5	0.9097	0.8812	0.8952	892
6	0.9360	0.9614	0.9485	958
7	0.9393	0.9183	0.9287	1028
8	0.9073	0.8943	0.9007	974
9	0.9139	0.9049	0.9094	1009
accuracy			0.9307	10000
macro avg	0.9299	0.9298	0.9297	10000
weighted avg	0.9305	0.9307	0.9305	10000

b)

Metrics for OVR Logistic Regression without Regularization-

OVR Logistic Regression without regularization

	precision	recall	f1-score	support
0	0.9467	0.9786	0.9624	980
1	0.9545	0.9797	0.9670	1135
2	0.9316	0.8847	0.9076	1032
3	0.8923	0.9109	0.9015	1010
4	0.9190	0.9358	0.9273	982
5	0.8910	0.8610	0.8757	892
6	0.9380	0.9468	0.9423	958
7	0.9253	0.9163	0.9208	1028
8	0.8718	0.8727	0.8722	974
9	0.9039	0.8860	0.8949	1009
accuracy			0.9184	10000
macro avg	0.9174	0.9173	0.9172	10000
weighted avg	0.9182	0.9184	0.9181	10000

Metrics for OVR Logistic Regression with L2 Regularization-

OVR Logistic Regression with L2 regularization

	precision	recall	f1-score	support
0	0.9375	0.9796	0.9581	980
1	0.9470	0.9762	0.9614	1135
2	0.9302	0.8779	0.9033	1032
3	0.9021	0.9030	0.9025	1010
4	0.8996	0.9308	0.9149	982
5	0.8885	0.8487	0.8681	892
6	0.9228	0.9478	0.9351	958
7	0.9189	0.9144	0.9166	1028
8	0.8698	0.8573	0.8635	974
9	0.8966	0.8761	0.8862	1009
accuracy			0.9124	10000
macro avg	0.9113	0.9112	0.9110	10000
weighted avg	0.9121	0.9124	0.9120	10000

Theory Questions

Sol^n 4) 1) Let our hypothesis f^n : $h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$
where $\{\theta_1, \theta_2, \cdots \theta_j\} = \theta$ are parameters of our hypoth.
$\hookrightarrow j$ = no. of features

Cost f^n $\rightarrow J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x_i) - y_i)^2$
n ↓ size of dataset

Writing $J(\theta)$ in terms of X we get,

$$J(\theta) = \frac{1}{2n} (X\theta - y)^T (X\theta - y) \quad \left\langle \because A^T A = |A|^2 \right\rangle$$

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

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

$\left\langle \because A^T B = B^T A \right\rangle$

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

$\left\langle \because \frac{\partial (x^T x)}{\partial x} = 2x \right\rangle$ On Taking Partial derivative,

$$\hookrightarrow \frac{\partial J(\theta)}{\partial \theta} = \frac{1}{2n} \left[2X^T X\theta - 2X^T y \right] = 0$$

$$\Rightarrow X^T X \theta = X^T y$$

On ~~multiplying~~ multiplying $(X^T X)^{-1}$ on both sides we get

$$\cancel{\theta = (X^T X)^{-1} X^T}$$

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

→ this is our closed form of Linear Regression

Sol^n 4) 2) It exists when X matrix is invertible

Sol^n 4) 3) ~~Wel~~, Closed from sol^n of Linear Reg. just looks easy but it is computationally very expensive as it requires to invert $(X^T X^{-1})$ ~~and~~ matrix which is very expensive and matrix multiplication for large ~~dat~~ feature dataset is also expensive.

~~for~~ We can use ~~gd~~ gradient descent when no. of features are small and no. of training sets are also small (i.e., < 20000)

Sol^n 4) 4) for Simple linear regression, line eq^n,

$$y = m x + b$$

where, $m = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$ & $b = \bar{y} - m\bar{x}$ for ~~for~~ Least Squares estimates

So, when $y = \bar{x}$ => $b = \frac{\sum(x - \bar{x})(\bar{x} - \bar{y})}{\sum(x - \bar{x})^2}$

~~Sol 5)~~

~~Sol 4) 5)~~

Sol 4) 4) for simple linear regression, eqn of line is:

$y = mx + b$

for least square line, we have to minimize

$$J = \sum_{k=1}^{n} [y_k - (mx_k + b)]^2 \quad \langle n = \text{no. of dataset} \rangle$$

Here we have 2 parameters m & b

$\therefore$ for minm, $\frac{\partial J}{\partial b} = 0$ & $\frac{\partial J}{\partial m} = 0$

$$\frac{\partial J}{\partial b} = -2 \sum_{k=1}^{n} [y_k - (mx_k + b)] = 0$$

$$\Rightarrow \sum_{1}^{n} y_k = m\sum_{1}^{n} x_k + \sum_{1}^{n} b$$

$$\Rightarrow \sum y_k = m\sum x_k + nb$$

$$\Rightarrow \frac{1}{n}\sum_{k=1}^{n} y_k = \frac{m}{n}\sum_{k=1}^{n} x_k + b$$

Page :
Date :

$\Rightarrow \bar{y} = m\bar{x} + b$

$\therefore$ Since $(\bar{x}, \bar{y})$ satisfies the eqⁿ ~~of~~ for Linear Regression line

$\Rightarrow$ for Least square line always passes through $(\bar{x}, \bar{y})$

Sol 4) (i) Yes, We can use Linear Regression for classification, by assigning a threshold. All the $h_\theta(x)$ below that threshold should go to class 1 and above the threshold should go to class 2.
But ~~logist~~ for classification problems we should prefer for logistic regression or other classification specific algos.