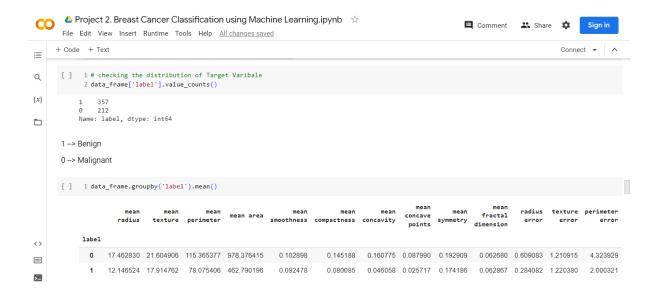


] 1 # checking for missing	y values		
2 data_frame.isnull().s	ım()		
mean radius	0		
mean texture	0		
mean perimeter	0		
mean area	0		
mean smoothness	0		
mean compactness	0		
mean concavity	0		
mean concave points	0		
mean symmetry	0		
mean fractal dimension	0		
radius error	0		
texture error	0		
perimeter error	0		
area error	0		
smoothness error	0		
compactness error	0		
concavity error	0		
concave points error	0		
symmetry error	0		
fractal dimension error	0		
	^		

[] 1 # statistical measures about the data 2 data_frame.describe()														
:}			mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	radius error	texture error
		count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
		mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798	0.405172	1.216853
		std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060	0.277313	0.551648
		min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960	0.111500	0.360200
		25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700	0.232400	0.833900
		50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540	0.324200	1.108000
		75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120	0.478900	1.474000
>		max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440	2.873000	4.885000
≕		4												→



```
Separating the features and target
Q
                            [ ] 1 X = data_frame.drop(columns='label', axis=1)
{x}
                                                    2 Y = data_frame['label']
[ ] 1 print(X)

        mean radius
        mean texture
        ...
        worst symmetry
        worst fractal dimension

        17.99
        10.38
        ...
        0.4601
        0.11890

        20.57
        17.77
        ...
        0.2750
        0.88902

        19.69
        21.25
        0.3613
        0.88758

        11.42
        20.38
        0.6638
        0.17300

        20.29
        14.34
        0.2364
        0.07678

        ...
        ...
        ...
        ...

        21.56
        22.39
        0.2060
        0.07118

        20.13
        28.25
        0.2572
        0.06637

                                               564
                                                                                                                                               28.25 ...
28.08 ...
29.33 ...
24.54 ...
                                               565
566
                                                                                                                                                                                                                                0.2572
0.2218
                                                                                                                                                                                                                                                                                                                              0.06637
0.07820
                                                                                          20.13
                                               567
                                                                                          20.60
                                                                                                                                                                                                                                 0.4087
                                                                                                                                                                                                                                                                                                                              0.12400
                                                                                                                                                                                                                                0.2871
                                                                                                                                                                                                                                                                                                                              0.07039
<>
                                               [569 rows x 30 columns]
\equiv
                             Splitting the data into training data & Testing data
Q
                           [ ] 1 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
{x}
                            [ ] 1 print(X.shape, X_train.shape, X_test.shape)
(569, 30) (455, 30) (114, 30)
                             Model Training
                            Logistic Regression
                           [ ] 1 model = LogisticRegression()
                            [\ ] 1 # training the Logistic Regression model using Training data
<>
==
                                                     3 model.fit(X_train, Y_train)
                                               /usr/local/lib/python 3.7/dist-packages/sklearn/linear\_model/\_logistic.py: 818: Convergence Warning: lbfgs failed to converge (status=1): lbfgs failed to con
>_
```