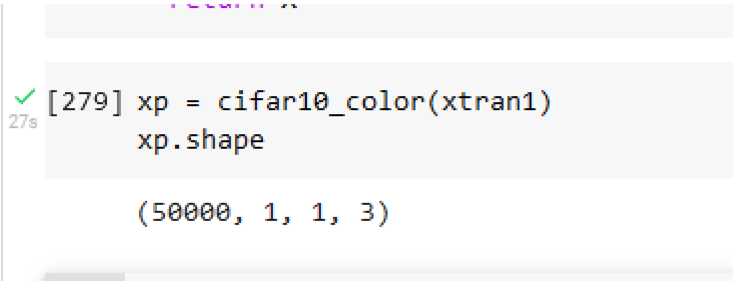
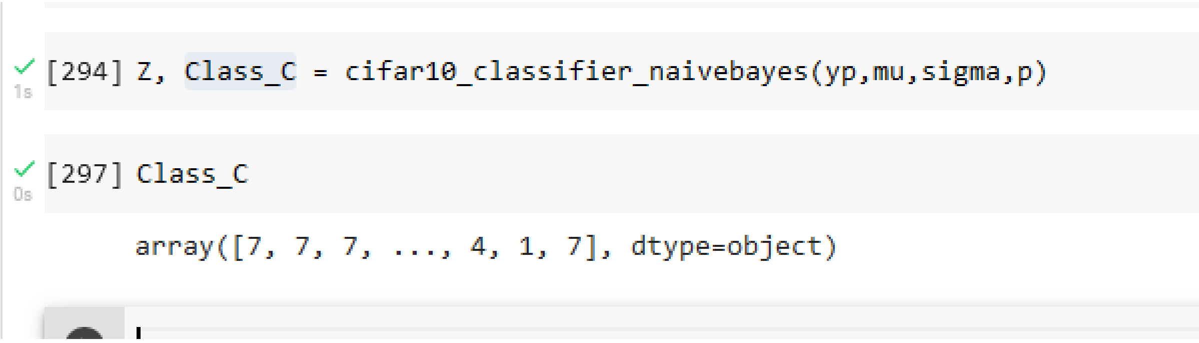
Question _ 1



```
        return x
```

```
[279] xp = cifar10_color(xtran1)
      xp.shape
```

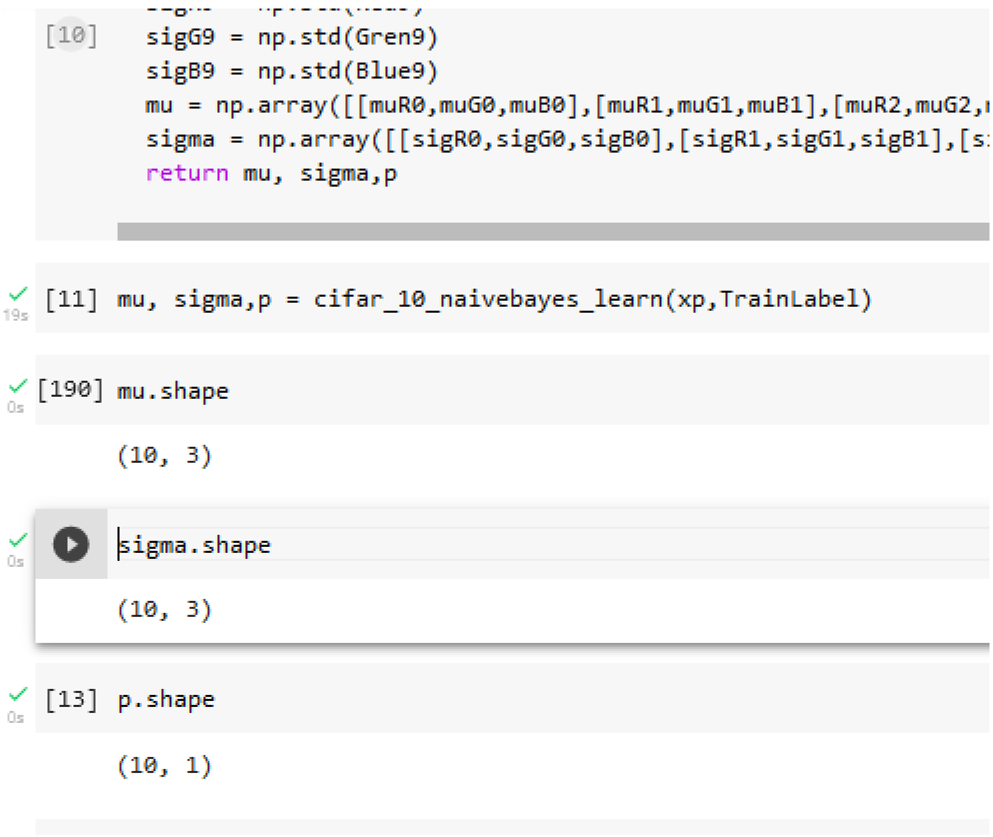
```
(50000, 1, 1, 3)
```



```
[294] Z, Class_C = cifar10_classifier_naivebayes(yp,mu,sigma,p)
```

```
[297] Class_C
```

```
array([7, 7, 7, ..., 4, 1, 7], dtype=object)
```



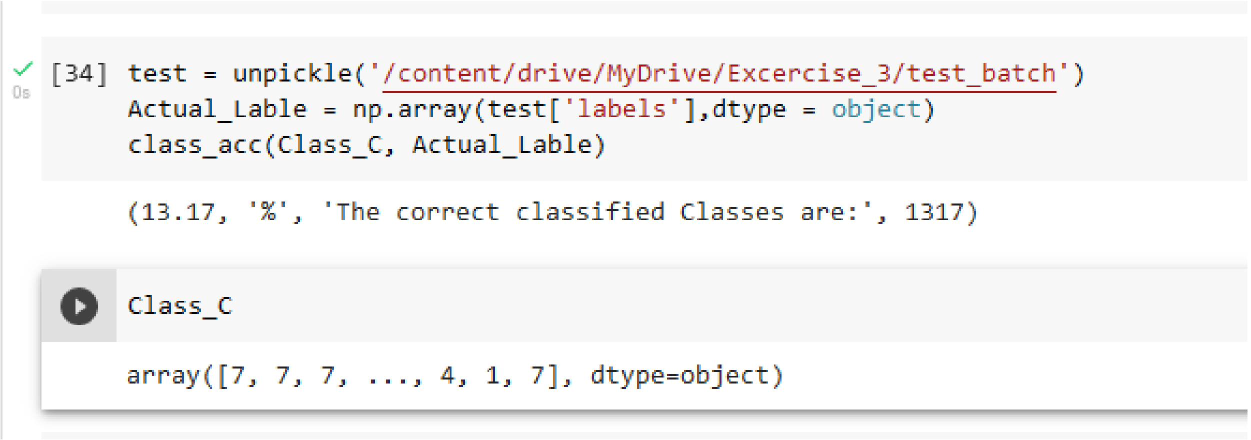
```
[10]  sigG9 = np.std(Gren9)
      sigB9 = np.std(Blue9)
      mu = np.array([[muR0,muG0,muB0],[muR1,muG1,muB1],[muR2,muG2,
      sigma = np.array([[sigR0,sigG0,sigB0],[sigR1,sigG1,sigB1],[s
      return mu, sigma,p
```

```
[11] mu, sigma,p = cifar_10_naivebayes_learn(xp,TrainLabel)
```

```
[190] mu.shape

(10, 3)
```

```
sigma.shape

(10, 3)
```

```
[13] p.shape

(10, 1)
```



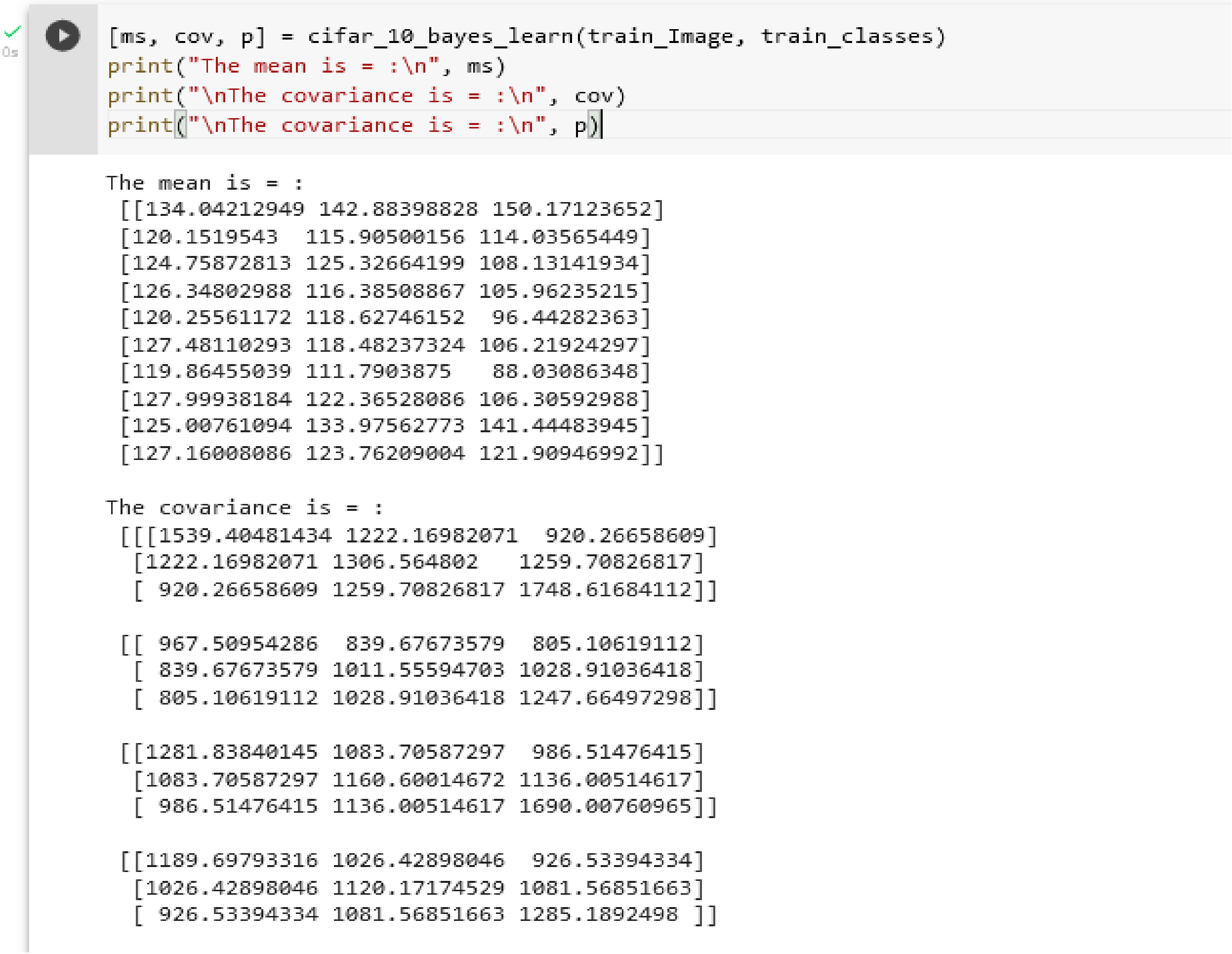
```
[34] test = unpickle('/content/drive/MyDrive/Excercise_3/test_batch')
     Actual_Lable = np.array(test['labels'],dtype = object)
     class_acc(Class_C, Actual_Lable)
```

```
(13.17, '%', 'The correct classified Classes are:', 1317)
```

```
Class_C
```

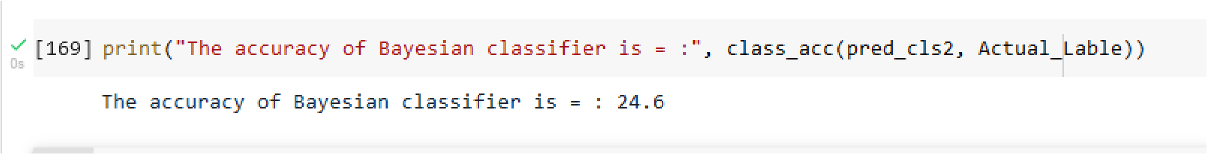
```
array([7, 7, 7, ..., 4, 1, 7], dtype=object)
```

Question # 2



```
[ms, cov, p] = cifar_10_bayes_learn(train_Image, train_classes)
print("The mean is = :\n", ms)
print("\nThe covariance is = :\n", cov)
print("\nThe covariance is = :\n", p)
```

```
The mean is = :
 [[134.04212949 142.88398828 150.17123652]
 [120.1519543  115.90500156 114.03565449]
 [124.75872813 125.32664199 108.13141934]
 [126.34802988 116.38508867 105.96235215]
 [120.25561172 118.62746152  96.44282363]
 [127.48110293 118.48237324 106.21924297]
 [119.86455039 111.7903875   88.03086348]
 [127.99938184 122.36528086 106.30592988]
 [125.00761094 133.97562773 141.44483945]
 [127.16008086 123.76209004 121.90946992]]

The covariance is = :
 [[[1539.40481434 1222.16982071  920.26658609]
  [1222.16982071 1306.564802   1259.70826817]
  [ 920.26658609 1259.70826817 1748.61684112]]

 [[ 967.50954286  839.67673579  805.10619112]
  [ 839.67673579 1011.55594703 1028.91036418]
  [ 805.10619112 1028.91036418 1247.66497298]]

 [[1281.83840145 1083.70587297  986.51476415]
  [1083.70587297 1160.60014672 1136.00514617]
  [ 986.51476415 1136.00514617 1690.00760965]]

 [[1189.69793316 1026.42898046  926.53394334]
  [1026.42898046 1120.17174529 1081.56851663]
  [ 926.53394334 1081.56851663 1285.1892498 ]]
```

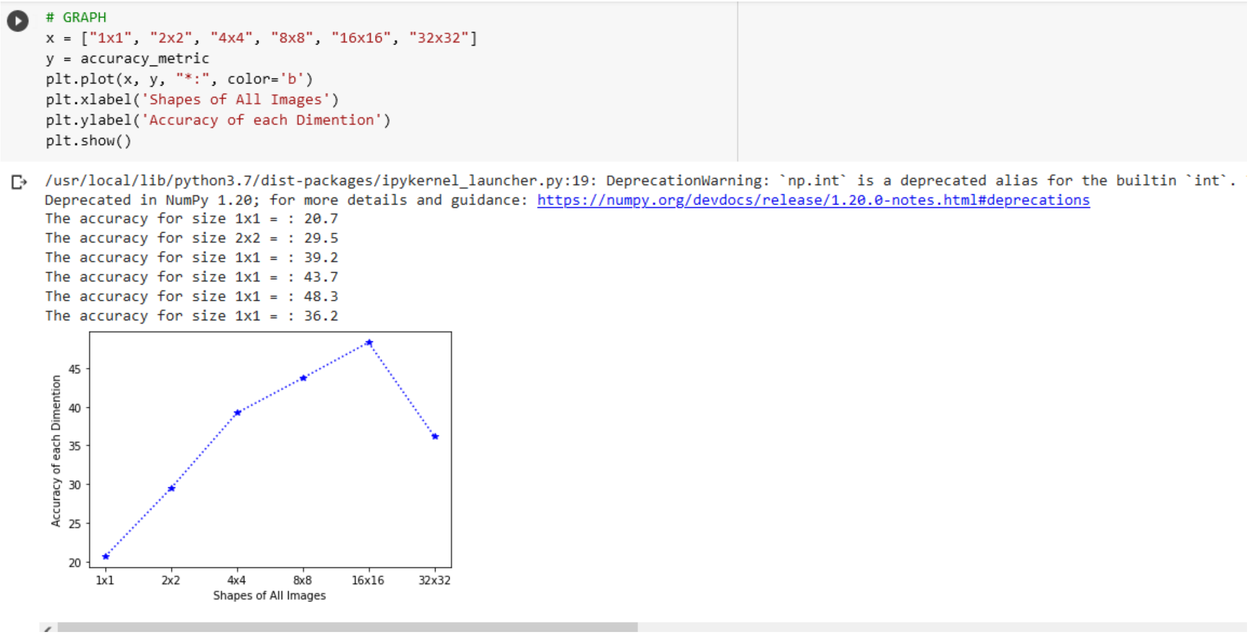


```
[169] print("The accuracy of Bayesian classifier is = :", class_acc(pred_cls2, Actual_Lable))
```

```
The accuracy of Bayesian classifier is = : 24.6
```

The accuracy for this is greater because here in this question we use the multivariate normal distribution. Furthermore, in the Naïve Bayes classifier, all the features are independent which is generally not the case.

Question # 3



```
# GRAPH
x = ["1x1", "2x2", "4x4", "8x8", "16x16", "32x32"]
y = accuracy_metric
plt.plot(x, y, "*:", color='b')
plt.xlabel('Shapes of All Images')
plt.ylabel('Accuracy of each Dimention')
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:19: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int`.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
The accuracy for size 1x1 = : 20.7
The accuracy for size 2x2 = : 29.5
The accuracy for size 1x1 = : 39.2
The accuracy for size 1x1 = : 43.7
The accuracy for size 1x1 = : 48.3
The accuracy for size 1x1 = : 36.2
```