```
# %%

# ==============================================================================
# LOGISTIC REGRESSION
# ==============================================================================


# %%

# importing the scikit-learn.dataset library

from sklearn import datasets


# %%

# generating the dataset for 1000 samples with linear relation from make_regression() fucntion of skearn

x, y = datasets.make_classification(n_samples=1000,  # number of samples
                                    n_features=4)  # number of features


# %%

# importing the matplotlib library

import matplotlib.pyplot as plt

# %%

# visualizing each feature

fig, axs = plt.subplots(2, 2)
axs[0, 0].scatter(x[:,0],y)
axs[0, 0].set_title('feature 1')
axs[0, 1].scatter(x[:,1], y)
axs[0, 1].set_title('feature 2')
axs[1, 0].scatter(x[:,2], y)
axs[1, 0].set_title('feature 3')
axs[1, 1].scatter(x[:,3], y)
axs[1, 1].set_title('feature 4')

# Hide x labels and tick labels for top plots and y ticks for right plots.
for ax in axs.flat:
    ax.label_outer()
# %%

# importing the train_test_split function of sklearn
```

```
from sklearn.model_selection import train_test_split

# %%

# creating training and test datasets with train_test_split func. of sklearn.model_selection

train_x, test_x, train_y, test_y = train_test_split(x, y, test_size=0.2)

print("train data size:", len(train_x))  # 800

print("test data size:", len(test_x))  # 200


# %%

# importing the LogisticRegression and metrics of sklearn

from sklearn.linear_model import LogisticRegression

# %%

# craeting training and testing data

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2)

# %%

# training the model
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_predicted = logreg.predict(X_test)

# %%

# accuracy of logistic regression model

print("accuracy: ", logreg.score(X_test, y_test)*100, "%", sep='')

# %%

# importing the confusion matrix library

from sklearn.metrics import confusion_matrix

#%%

# creating confusion matrix
```

```
cm = confusion_matrix(y, logreg.predict(x))

fig, ax = plt.subplots(figsize=(8, 8))
ax.imshow(cm)
ax.grid(False)
ax.xaxis.set(ticks=(0, 1), ticklabels=('Predicted 0s', 'Predicted 1s'))
ax.yaxis.set(ticks=(0, 1), ticklabels=('Actual 0s', 'Actual 1s'))
ax.set_ylim(1.5, -0.5)
for i in range(2):
    for j in range(2):
        ax.text(j, i, cm[i, j], ha='center', va='center', color='red')
plt.show()

# %%

# ==============================================================================
# THE END
# ==============================================================================
```