

```
In [163]: import pandas as p
import numpy as n
import seaborn as s
import matplotlib.pyplot as plt
df = p.read_csv("Admission_Predict_Ver1.1.csv")
df
```

Out[163]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65
...	...	...	...	...	...	...	...	...	...
495	496	332	108	5	4.5	4.0	9.02	1	0.87
496	497	337	117	5	5.0	5.0	9.87	1	0.96
497	498	330	120	5	4.5	5.0	9.56	1	0.93
498	499	312	103	4	4.0	5.0	8.43	0	0.73
499	500	327	113	4	4.5	4.5	9.04	0	0.84

500 rows × 9 columns

```
In [164]: df["Chance of Admit "] = [1 if each >0.75 else 0 for each in df["Chance of Admit "]]
df
```

Out[164]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	1
1	2	324	107	4	4.0	4.5	8.87	1	1
2	3	316	104	3	3.0	3.5	8.00	1	0
3	4	322	110	3	3.5	2.5	8.67	1	1
4	5	314	103	2	2.0	3.0	8.21	0	0
...	...	...	...	...	...	...	...	...	...
495	496	332	108	5	4.5	4.0	9.02	1	1
496	497	337	117	5	5.0	5.0	9.87	1	1
497	498	330	120	5	4.5	5.0	9.56	1	1
498	499	312	103	4	4.0	5.0	8.43	0	0
499	500	327	113	4	4.5	4.5	9.04	0	1

500 rows × 9 columns

```
In [165]: X = df.iloc[:,1:8]
Y = df[["Chance of Admit "]]
X
```

```
Out[165]:
```

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
0	337	118	4	4.5	4.5	9.65	1
1	324	107	4	4.0	4.5	8.87	1
2	316	104	3	3.0	3.5	8.00	1
3	322	110	3	3.5	2.5	8.67	1
4	314	103	2	2.0	3.0	8.21	0
...	...	...	...	...	...	...	...
495	332	108	5	4.5	4.0	9.02	1
496	337	117	5	5.0	5.0	9.87	1
497	330	120	5	4.5	5.0	9.56	1
498	312	103	4	4.0	5.0	8.43	0
499	327	113	4	4.5	4.5	9.04	0

500 rows × 7 columns

```
In [166]: Y
```

```
Out[166]:
```

	Chance of Admit
0	1
1	1
2	0
3	1
4	0
...	...
495	1
496	1
497	1
498	0
499	1

500 rows × 1 columns

```
In [167]: from sklearn.model_selection import train_test_split
```

```
In [168]: X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.25,random_s
print(X_train.shape,X_test.shape,Y_train.shape,Y_test.shape)
```

(375, 7) (125, 7) (375, 1) (125, 1)

```
In [169]: from sklearn.tree import DecisionTreeClassifier, plot_tree
```

```
In [170]: DT = DecisionTreeClassifier(random_state=1)
```

```
In [171]: from sklearn.metrics import ConfusionMatrixDisplay, classification_report
```

```
In [172]: DT.fit(X_train, Y_train)
```

```
Out[172]: DecisionTreeClassifier(random_state=1)
```

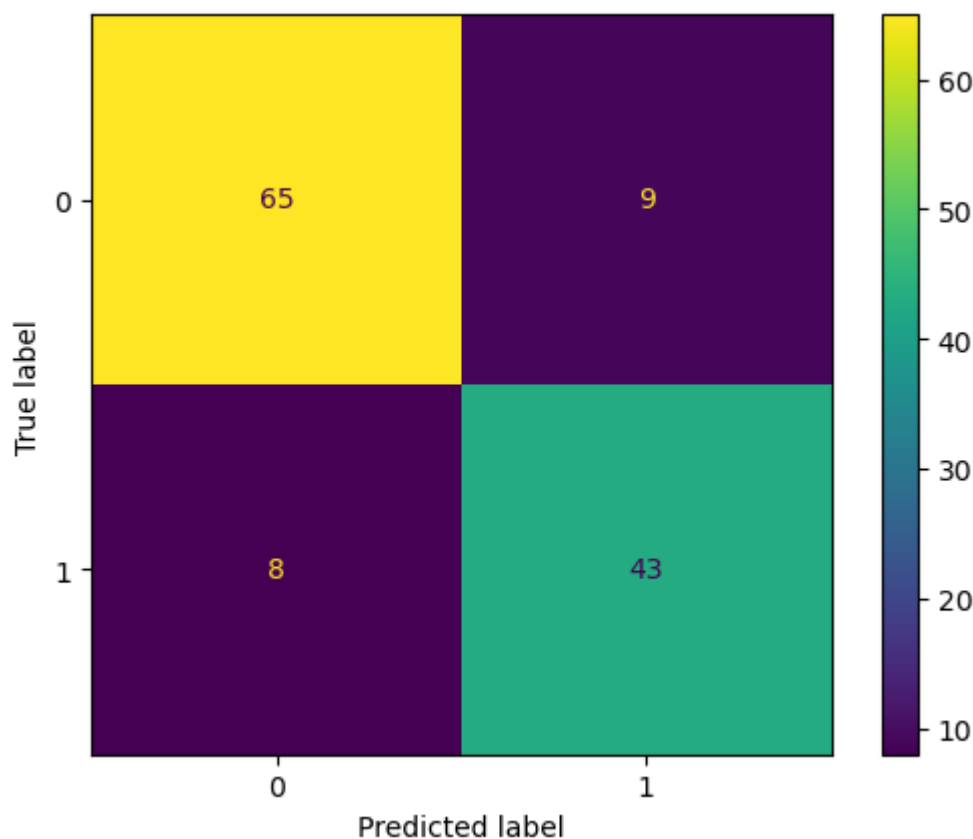
**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**

**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
In [173]: Y_predict = DT.predict(X_test)
```

```
In [174]: ConfusionMatrixDisplay.from_predictions(Y_test, Y_predict)
```

```
Out[174]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x20a11409210>
```



```
In [175]: print(classification_report(Y_test,Y_predict))
```

	precision	recall	f1-score	support
0	0.89	0.88	0.88	74
1	0.83	0.84	0.83	51
accuracy			0.86	125
macro avg	0.86	0.86	0.86	125
weighted avg	0.86	0.86	0.86	125

```
In [177]: new = [[312,8.43]]  
DT.predict(new)[0]
```

```
C:\Users\suraj fartale\AppData\Roaming\Python\Python311\site-packages\sklearn\base.py:465: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names  
  warnings.warn(
```

```

-----
-
ValueError                                Traceback (most recent call last)
Cell In[177], line 2
      1 new = [[312,8.43]]
----> 2 DT.predict(new)[0]

File ~\AppData\Roaming\Python\Python311\site-packages\sklearn\tree\_classes.py:500, in BaseDecisionTree.predict(self, X, check_input)
    477 """Predict class or regression value for X.
    478
    479 For a classification model, the predicted class for each sample in
X is
    (...)
    497 The predicted classes, or the predict values.
    498 """
    499 check_is_fitted(self)
--> 500 X = self._validate_X_predict(X, check_input)
    501 proba = self.tree_.predict(X)
    502 n_samples = X.shape[0]

File ~\AppData\Roaming\Python\Python311\site-packages\sklearn\tree\_classes.py:460, in BaseDecisionTree._validate_X_predict(self, X, check_input)
    458 else:
    459     force_all_finite = True
--> 460 X = self._validate_data(
    461     X,
    462     dtype=DTYPE,
    463     accept_sparse="csr",
    464     reset=False,
    465     force_all_finite=force_all_finite,
    466 )
    467 if issparse(X) and (
    468     X.indices.dtype != np.intc or X.indptr.dtype != np.intc
    469 ):
    470     raise ValueError("No support for np.int64 index based sparse matrices")

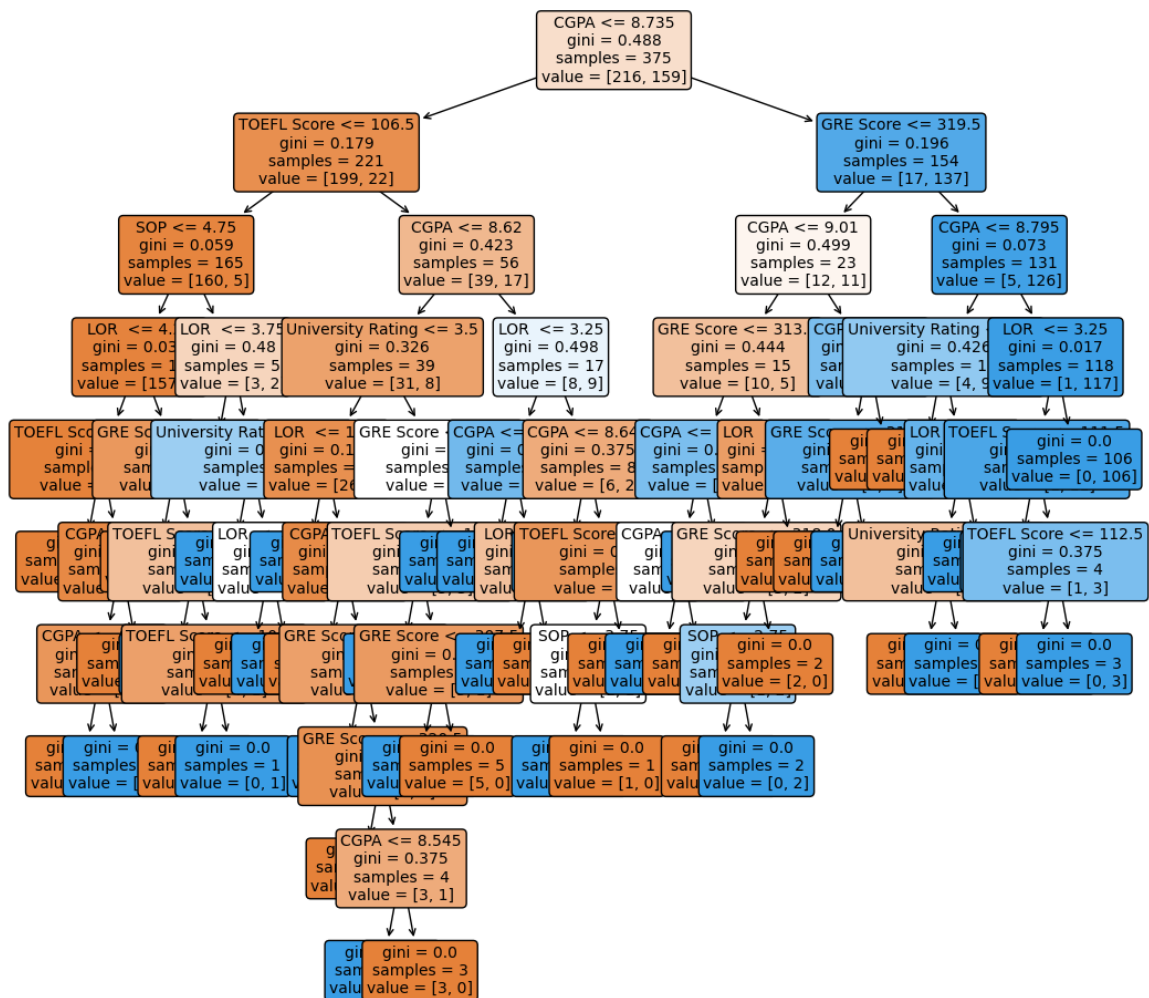
File ~\AppData\Roaming\Python\Python311\site-packages\sklearn\base.py:626, in BaseEstimator._validate_data(self, X, y, reset, validate_separately, cast_to_ndarray, **check_params)
    623 out = X, y
    625 if not no_val_X and check_params.get("ensure_2d", True):
--> 626     self._check_n_features(X, reset=reset)
    628 return out

File ~\AppData\Roaming\Python\Python311\site-packages\sklearn\base.py:415, in BaseEstimator._check_n_features(self, X, reset)
    412 return
    414 if n_features != self.n_features_in_:
--> 415     raise ValueError(
    416         f"X has {n_features} features, but {self.__class__.__name__} "
    417         f"is expecting {self.n_features_in_} features as input."
    418     )

ValueError: X has 2 features, but DecisionTreeClassifier is expecting 7 features as input.

```

```
In [178]: plt.figure(figsize=(12,12))
plot_tree(DT,fontsize=10,rounded=True,filled=True,feature_names=X.columns)
plt.show()
```



```
In [179]: from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LogisticRegression
```

```
In [180]: DTR = DecisionTreeRegressor(random_state=0)
RFR = RandomForestRegressor(random_state=0)
LR = LogisticRegression(random_state=1,solver='lbfgs',max_iter=1000)
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.25,random_s
print(X_train.shape,X_test.shape,Y_train.shape,Y_test.shape)
```

```
(375, 7) (125, 7) (375, 1) (125, 1)
```

```
In [181]: DTR.fit(X_train,Y_train)
```

```
Out[181]: DecisionTreeRegressor(random_state=0)
```

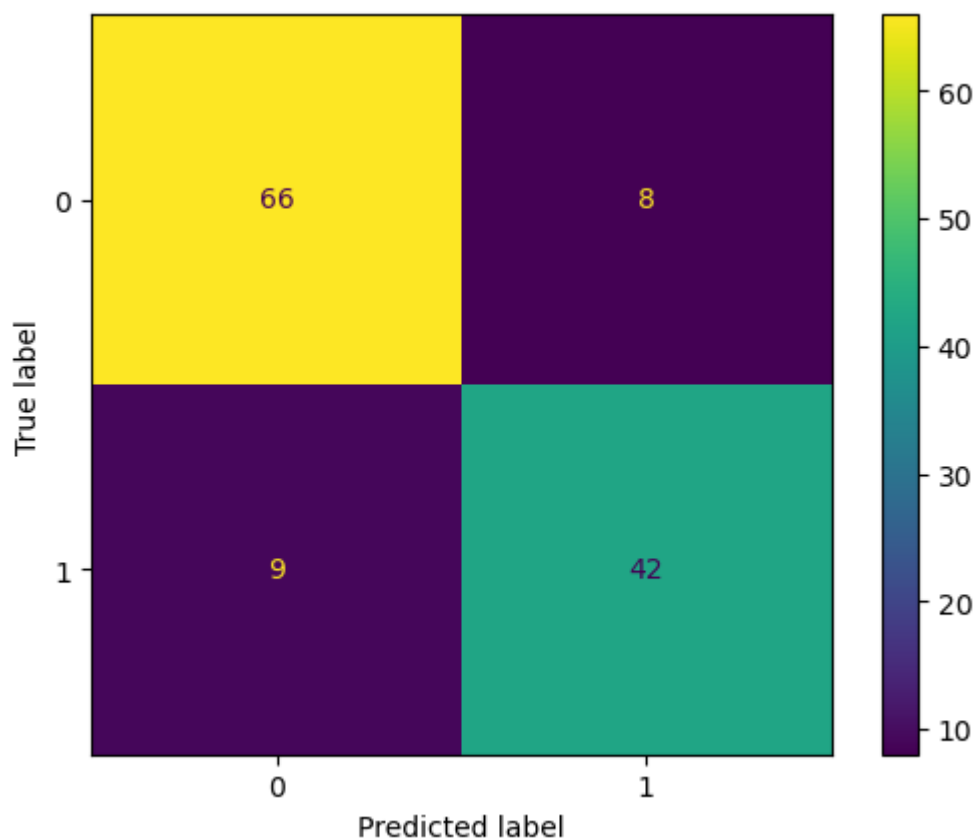
**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**  
**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

```
In [182]: Y_predict_DTR = DTR.predict(X_test)
Y_predict_DTR
```

```
Out[182]: array([0., 0., 1., 1., 1., 0., 1., 0., 1., 0., 0., 0., 1., 1., 1., 1., 0.,
        1., 1., 0., 0., 1., 1., 0., 1., 0., 1., 1., 1., 0., 0., 1., 0., 1.,
        0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0., 0., 1., 1.,
        1., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 0., 0., 0.,
        1., 0., 0., 0., 1., 0., 1., 0., 1., 1., 0., 0., 1., 0., 1., 1., 1.,
        0., 1., 1., 0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 1., 0., 0.,
        0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0.,
        1., 1., 0., 0., 1., 0.]
```

```
In [183]: ConfusionMatrixDisplay.from_predictions(Y_test,Y_predict_DTR)
```

```
Out[183]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x20a11e376d0>
```



```
In [184]: print(classification_report(Y_test,Y_predict_DTR))
```

	precision	recall	f1-score	support
0	0.88	0.89	0.89	74
1	0.84	0.82	0.83	51
accuracy			0.86	125
macro avg	0.86	0.86	0.86	125
weighted avg	0.86	0.86	0.86	125



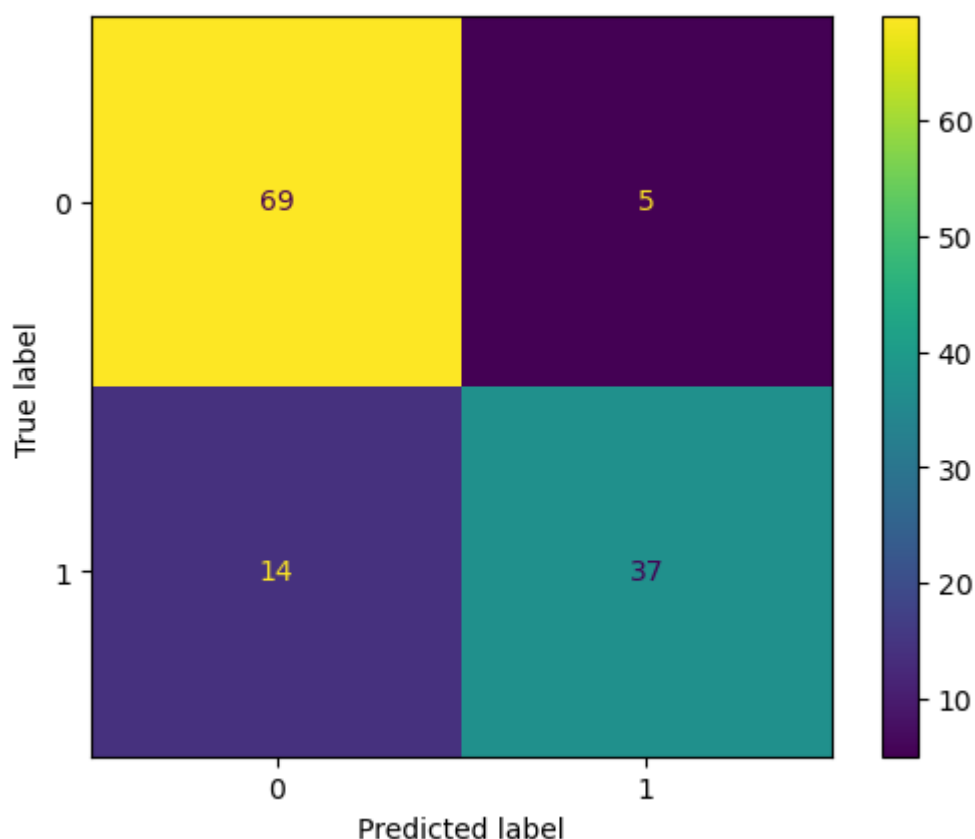
```
In [187]: RFR.fit(X_train,Y_train)
Y_predict_RFR = RFR.predict(X_test)
Y_predict_RFR = [1 if each >0.75 else 0 for each in Y_predict_RFR]
```

C:\Users\suraj fartale\AppData\Roaming\Python\Python311\site-packages\sklearn\base.py:1152: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

```
return fit_method(estimator, *args, **kwargs)
```

```
In [188]: ConfusionMatrixDisplay.from_predictions(Y_test,Y_predict_RFR)
```

```
Out[188]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x20a13053dd0>
```



```
In [190]: print(classification_report(Y_test,Y_predict_RFR))
```

	precision	recall	f1-score	support
0	0.83	0.93	0.88	74
1	0.88	0.73	0.80	51
accuracy			0.85	125
macro avg	0.86	0.83	0.84	125
weighted avg	0.85	0.85	0.85	125

```
In [191]: LR.fit(X_train,Y_train)
Y_predict_LR = LR.predict(X_test)
Y_predict
```

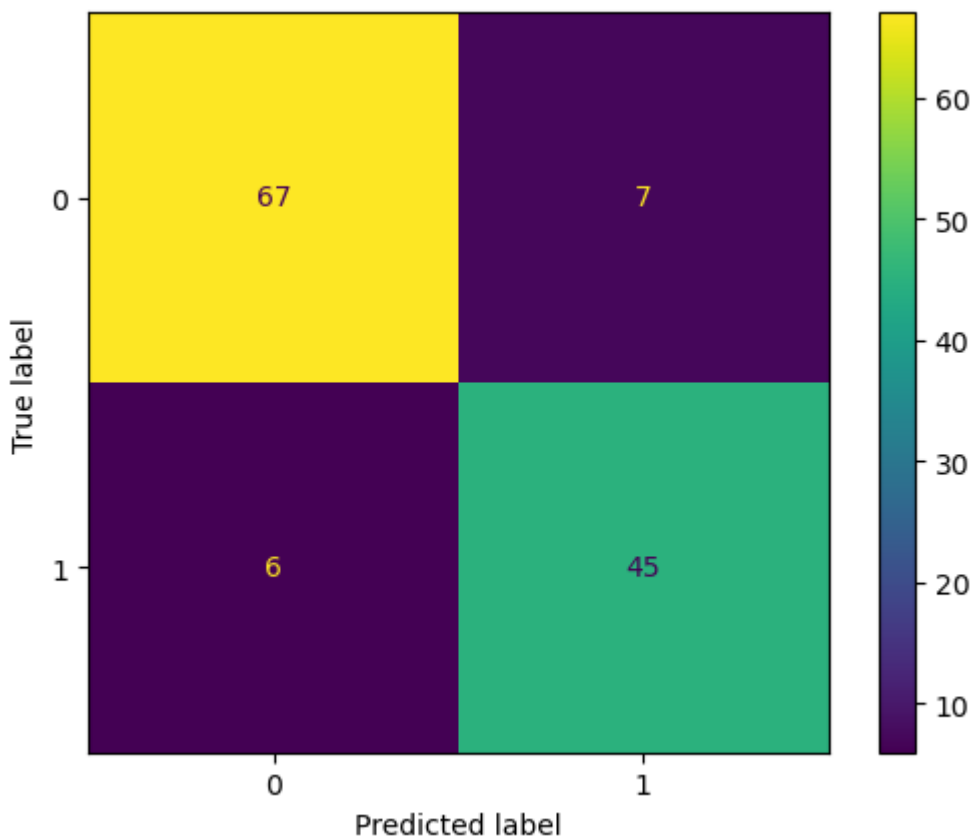
C:\Users\suraj fartale\AppData\Roaming\Python\Python311\site-packages\sklearn\utils\validation.py:1183: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

```
Out[191]: array([0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1,
        1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
        1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
        0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1,
        0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
        0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0], dtype=int64)
```

```
In [193]: ConfusionMatrixDisplay.from_predictions(Y_test,Y_predict_LR)
```

```
Out[193]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x20a130f0f90>
```



```
In [194]: print(classification_report(Y_test,Y_predict_LR))
```

	precision	recall	f1-score	support
0	0.92	0.91	0.91	74
1	0.87	0.88	0.87	51
accuracy			0.90	125
macro avg	0.89	0.89	0.89	125
weighted avg	0.90	0.90	0.90	125

In [ ]: