```
# Load the diabetes dataset

from sklearn import tree
from pandas import read_csv
import os
import numpy as np
df = read_csv("glass.csv")


df
```

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
0	1.51793	12.79	3.50	1.12	73.03	0.64	8.77	0.0	0.00	'build wind float'
1	1.51643	12.16	3.52	1.35	72.89	0.57	8.53	0.0	0.00	'vehic wind float'
2	1.51793	13.21	3.48	1.41	72.64	0.59	8.43	0.0	0.00	'build wind float'
3	1.51299	14.40	1.74	1.54	74.55	0.00	7.59	0.0	0.00	tableware
4	1.53393	12.30	0.00	1.00	70.16	0.12	16.19	0.0	0.24	'build wind non-float'
...	...	...	...	...	...	...	...	...	...	...
209	1.51610	13.42	3.40	1.22	72.69	0.59	8.32	0.0	0.00	'vehic wind float'
210	1.51592	12.86	3.52	2.12	72.66	0.69	7.97	0.0	0.00	'build wind non-float'
211	1.51613	13.92	3.52	1.25	72.88	0.37	7.94	0.0	0.14	'build wind non-float'
212	1.51689	12.67	2.88	1.71	73.21	0.73	8.54	0.0	0.00	'build wind non-float'
213	1.51852	14.09	2.19	1.66	72.67	0.00	9.32	0.0	0.00	tableware

214 rows × 10 columns

```
# Separate the target variable (Outcome) from the features
x = np.array(df.drop(["Type"], 1))
y = np.array(df["Type"])
```

```
<ipython-input-4-e54205976c66>:2: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument
  x = np.array(df.drop(["Type"], 1))
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 100)


# Create a Logistic Regression model
from sklearn.linear_model import LogisticRegression
logistic_regression = LogisticRegression()


# Fit the model to the training data
logistic_regression.fit(x_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

▾ LogisticRegression

LogisticRegression()

```
# Make predictions on the test data
y_pred = logistic_regression.predict(x_test)


# Evaluate the model
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
accuracy = accuracy_score(y_test, y_pred)
print(accuracy*100,"%")
```

```
65.11627906976744 %
```

```
# Create a logistic regression model
logistic_regression = LogisticRegression(multi_class='auto', max_iter=1000)

# # Optimization algorithmDefine hyperparameters for tuning
param_grid = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100],  # Regularization parameter
    'penalty': ['l1', 'l2'],             # Regularization type
    'solver': ['lbfgs', 'liblinear', 'saga'],
}


# Create a GridSearchCV object for hyperparameter tuning
from sklearn.model_selection import GridSearchCV
grid_search = GridSearchCV(logistic_regression, param_grid, cv=5, verbose=1)



# Fit the model to the training data
grid_search.fit(x_train, y_train)

# Get the best hyperparameters
best_params = grid_search.best_params_
best_params
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was reached which means th
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was reached which means th
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was reached which means th
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was reached which means th
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was reached which means th
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py:378: FitFailedWarning:
30 fits failed out of a total of 180.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
30 fits failed with the following error:
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py", line 686, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 1162, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 54, in _check_solver
    raise ValueError(
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:952: UserWarning: One or more of the test scores are non-f
        nan 0.35680672 0.35680672 0.41546218 0.39781513 0.41529412
        nan 0.46823529 0.49764706 0.50907563 0.46252101 0.47394958
        nan 0.56756303 0.5907563  0.60840336 0.54991597 0.58504202
        nan 0.60235294 0.58504202 0.63159664 0.61983193 0.57915966
        nan 0.59663866 0.57915966 0.60235294 0.61394958 0.57915966]
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
{'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
```

```
# Create a new logistic regression model with the best hyperparameters
best_logistic_regression = LogisticRegression(
    multi_class='auto',
    max_iter=1000,
    C=best_params['C'],
    penalty=best_params['penalty'],
    solver=best_params['solver']
)


# Fit the best model to the training data
best_logistic_regression.fit(x_train, y_train)

# Make predictions on the test data
y_pred = best_logistic_regression.predict(x_test)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
# Evaluate the model
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
accuracy = accuracy_score(y_test, y_pred)
print(accuracy*100,"%")
```

```
67.44186046511628 %
```