```python
#importing reqd. modules
from sklearn import datasets
from sklearn.linear_model import LogisticRegression
import numpy as np

#loading dataset
iris = datasets.load_iris()

#slicing dataset
X = iris["data"][:, 3:]
Y = (iris["target"] == 2).astype(np.int)

#Training a logistic regression classifier
clf = LogisticRegression()
clf.fit(X,Y)

#prdicting result
pred = clf.predict([[2.6]])
print(pred)
```

Output:

```
(learning_ml) PS C:\Users\prath\Desktop\Coding> & "c:/Users/prath/Desktop/Coding/Machine Learning/learning_ml/Scripts/python.exe" "c:/Users/prath/Desktop/Coding/Machine Learning/logistic_regression.py"
c:\Users\prath\Desktop\Coding\Machine Learning\logistic_regression.py:11: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If you wish to review your current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  Y = (iris["target"] == 2).astype(np.int)

[1]
```