Logistic Regression using sklearn

```
from sklearn import datasets
from sklearn.linear_model import LogisticRegression

cancer = datasets.load_breast_cancer()

clf = LogisticRegression()

clf.fit(cancer.data, cancer.target)

clf.predict(cancer.data)

clf.score(cancer.data, cancer.target)
```

↓

mean accuracy

```
clf.predict(cancer.data) - cancer.target
```

↓

wherever 0 : correct prediction

1, -1 : wrong prediction

↓

helps us recognise the points of distinction and can be used to refine the system.

```
clf.predict_proba(cancer.data)
```

↓

gives the value of the hypothesis function for the data point

↓

tells us as to how sure, we are about choosing a value.

↓

if the difference is not much, then the model is not very certain.

↓

these point are the main points of error

*) Tweaking the sklearn logistic regression classifier

1. for regularization,

$$\text{cost} = \text{error function} + \lambda \Sigma m_i$$

↓

c parameter in classifier in sklearn

$$\text{net cost} = c(\text{org\_cost}) + \Sigma (m_i)^2$$

↓

if very large, then regularization won't happen.

↓

Rather than applying weight to summation of parameters, it applied weight to the original cost and that is a hyperparameter for the sklearn classifier

2. Solver = "liblinear" by default → does not support multinomial.

↓

sag or saga are faster alternatives.

3. Tolerance value : when change in cost fn. value is less than this value, we don't proceed further.

4. multiclass = "ovr" , "multinomial"