

# Model Validation Methods

## 1. Evaluate using a train and a test set

In [1]:

```
# Evaluate using a train and a test set
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
filename = 'C:/Users/Ashraf/Documents/Datafiles/pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
test_size = 0.33
seed = 7
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=test_size,
random_state=seed)
model = LogisticRegression()
model.fit(X_train, Y_train)
result = model.score(X_test, Y_test)
```

C:\Users\Ashraf\anaconda3\lib\site-packages\sklearn\linear\_model\\_logistic.p  
y:763: 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> (<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](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression) ([https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression))  
n\_iter\_i = \_check\_optimize\_result(

In [2]:

```
result*100.0
```

Out[2]:

78.74015748031496

## 2. Evaluate using Cross Validation

In [3]:

```
# Evaluate using Cross Validation
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
filename = 'C:/Users/Ashraf/Documents/Datafiles/pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
num_folds = 10
seed = 7
kfold = KFold(n_splits=num_folds)
model = LogisticRegression(max_iter=400)
results = cross_val_score(model, X, Y, cv=kfold)
```

In [4]:

```
results.mean()*100.0
```

Out[4]:

77.60423786739577

In [5]:

```
results.std()*100.0
```

Out[5]:

5.157545262086822

### 3.Evaluate using Leave One Out Cross Validation

In [6]:

```
# Evaluate using Leave One Out Cross Validation
from pandas import read_csv
from sklearn.model_selection import LeaveOneOut
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
filename = 'C:/Users/Ashraf/Documents/Datafiles/pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
loocv = LeaveOneOut()
model = LogisticRegression(max_iter=300)
results = cross_val_score(model, X, Y, cv=loocv)
```

In [7]:

```
results.mean()*100.0
```

Out[7]:

77.60416666666666

In [8]:

```
results.std()*100.0
```

Out[8]:

41.68944689773287

In [ ]: