Classification

Logistic regression

In [1]:

```
# Load packages
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
```

In [2]:

```
# Load data
df = pd.read_csv('default.csv')
```

In [3]:

```
# Prepare data
predictors = ['student', 'balance', 'income']
X = pd.get_dummies(df[predictors], columns=['student'], drop_first=True)
y = df['default']
```

In [4]:

```
# Fit model to data
model = LogisticRegression(C=1e8, tol=1e-8)
model.fit(X.values, y.values)
```

Out[4]:

```
LogisticRegression(C=100000000.0, class_weight=None, dual=False,
          fit_intercept=True, intercept_scaling=1, max_iter=100,
          multi_class='ovr', n_jobs=1, penalty='l2', random_state=None,
          solver='liblinear', tol=1e-08, verbose=0, warm_start=False)
```

In [5]:

```
# Display intercept
model.intercept_
```

Out[5]:

```
array([-10.39076375])
```

In [6]:

```
# Display coefficients
pd.Series(model.coef_[0], index=X.columns)
```

Out[6]:

```
balance        0.008292
income        -0.000019
student_Yes    0.357429
dtype: float64
```

In [7]:

```
# Make predictions
predictions = model.predict(X)
predictions[:5]
```

Out[7]:

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

In [8]:

```
# Get probabilities
probability = model.predict_proba(X)
probability[:5, 1]
```

Out[8]:

```
array([0.99931655, 0.86022571, 0.22745248, 0.98866793, 0.99044681])
```

Inference statistics for logistic regression

In [9]:

```
# Optional: get OLS regression results
from statsmodels.api import Logit
from patsy import dmatrices
```

In [10]:

```
# Define logistic regression model
y, X = dmatrices('default ~ student + balance + income', df, return_type='dataframe')
```

In [11]:

```
Logit(y, X).fit().summary()
```

```
Optimization terminated successfully.
         Current function value: 0.197660
         Iterations 9
```

Out[11]:

Logit Regression Results

Dep. Variable:	default	**No. Observations:**	100
Model:	Logit	**Df Residuals:**	96
Method:	MLE	**Df Model:**	3
Date:	Sun, 21 Oct 2018	**Pseudo R-squ.:**	0.7148
Time:	16:36:43	**Log-Likelihood:**	-19.766
converged:	True	**LL-Null:**	-69.315
		LLR p-value:	2.430e-21

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-10.3929	3.031	-3.428	0.001	-16.334	-4.452
student[T.Yes]	0.3577	1.393	0.257	0.797	-2.372	3.087
balance	0.0083	0.002	4.357	0.000	0.005	0.012
income	-1.881e-05	5.04e-05	-0.373	0.709	-0.000	7.99e-05

In [12]:

```
# Visualize balance and income
df.loc[df.default==1, :].plot.scatter(x='balance', y='income',
                                      color='red', label='default');
df.loc[df.default==0, :].plot.scatter(ax=plt.gca(), x='balance', y='income',
                                      color='blue', label='no-default');
plt.legend();
```