

BUSA3020-Tutorial-Week4

February 7, 2023

Week 4 Tutorial

Unit Convenor & Lecturer

George Milunovich

george.milunovich@mq.edu.au

Objectives

- Q1: Go again through the code of **Training a Logistic Regression Model with scikit-learn** from Week 3 lecture
 - Q2: Compute the effect of `penalty` on estimated weights in Logistic Regression (code on p. 77 of Raschka & Mirjalili textbook).
 - Explain the code and the results
 - Q3: Complete Kaggle's [Booleans and Conditionals Tutorial](#)
 - Q4: Work on Assignment 1
-

Q1:

- Go through the code of Training a Logistic Regression Model with scikit-learn from Week 3 lecture

Q2:

Compute the effect of `penalty` on estimated weights in Logistic Regression (code on p. 77 of textbook). Explain the program and the result.

In order to execute that code we need to first: - import data - make X & y - split data into training and test sets - standardise data

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import numpy as np

iris = datasets.load_iris()
# print(iris.data)
```

```

# print(iris.target)

X = iris.data[:, [2, 3]]
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 1, s

# print(X_train.shape)
# print(y_train.shape)

sc = StandardScaler()

sc.fit(X_train)

X_train_scaled = sc.transform(X_train)
X_test_scaled = sc.transform(X_test)

# print(X_test_scaled)

```

[]:

We can then implement the code on p. 77

```

from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt

weights, params = [], []
# print(type(weights))
# print(weights)

# for c in np.arange(-5, 5):
#     C = 10.**c
#     print('c:', c, 'C:', C)

for c in np.arange(-5, 5):
    C_ = 10.**c
    print('c:', c, 'C_:', C_)

    lr = LogisticRegression(C=C_, random_state=1, solver='lbfgs', multi_class='ovr')
    lr.fit(X_train_scaled, y_train)
#     print(lr.coef_[1])
    weights.append(lr.coef_[1])
    params.append(10.**c)

# print('weights list', weights)
weights = np.array(weights)

```

```
# print('weights np array', weights)

plt.plot(params, weights[:, 0], label='petal length')
plt.plot(params, weights[:, 1], linestyle='--', label='petal width')
plt.ylabel('weight coefficient')
plt.xlabel('C')
plt.legend(loc='upper left')
plt.xscale('log')
plt.show()

print(np.around(weights, 2))
```

[]:

Q3:

- Complete Kaggle's [Booleans and Conditionals Tutorial](#)

Q4:

- Work on Assignment 1

[]: