



MACHINE LEARNING 3

Assignment 1

MACHINE LEARNING 3: Assignment 1

Table of Contents

1. Introduction

2. Problem Statement

3. Output

1. Introduction

This assignment will help you to consolidate the concepts learnt in the session.

2. Problem Statement

I decided to treat this as a classification problem by creating a new binary variable `affair` (did the woman have at least one affair?) and trying to predict the classification for each woman.

Dataset

The dataset I chose is the `affairs` dataset that comes with `Statsmodels`. It was derived from a survey of women in 1974 by *Redbook* magazine, in which married women were asked about their participation in extramarital affairs. More information about the study is available in a 1978 paper from the *Journal of Political Economy*.

Description of Variables

The dataset contains 6366 observations of 9 variables:

`rate_marriage`: woman's rating of her marriage (1 = very poor, 5 = very good)

`age`: woman's age

`yrs_married`: number of years married

`children`: number of children

`religious`: woman's rating of how religious she is (1 = not religious, 4 = strongly religious)

`educ`: level of education (9 = grade school, 12 = high school, 14 = some college, 16 = college graduate, 17 = some graduate school, 20 = advanced degree)

occupation: woman's occupation (1 = student, 2 = farming/semi-skilled/unskilled, 3 = "white collar", 4 = teacher/nurse/writer/technician/skilled, 5 = managerial/business, 6 = professional with advanced degree)

occupation_husb: husband's occupation (same coding as above)

affairs: time spent in extra-marital affairs

Code to loading data and modules:

```
import numpy as np

import pandas as pd

import statsmodels.api as sm

import matplotlib.pyplot as plt

from patsy import dmatrices

from sklearn.linear_model

import LogisticRegression from

sklearn.cross_validation

import train_test_split from sklearn

import metrics from

sklearn.cross_validation

import cross_val_score dta =

sm.datasets.fair.load_pandas().data
```

```

#add "affair" column: 1 represents having affairs, 0
represents not dta['affair'] = (dta.affairs >
0).astype(int)

y, X = dmatrices('affair ~ rate_marriage + age +
                yrs_married + children + \ religious + educ +
                C(occupation) + C(occupation_husb)',
                dta, return_type="dataframe")

X = X.rename(columns =
{'C(occupation)[T.2.0]':'occ_2',
 'C(occupation)[T.3.0]':'occ_3',
 'C(occupation)[T.4.0]':'occ_4',
 'C(occupation)[T.5.0]':'occ_5',
 'C(occupation)[T.6.0]':'occ_6',
 'C(occupation_husb)[T.2.0]':'occ_husb_2',
 'C(occupation_husb)[T.3.0]':'occ_husb_3',
 'C(occupation_husb)[T.4.0]':'occ_husb_4',
 'C(occupation_husb)[T.5.0]':'occ_husb_5',
 'C(occupation_husb)[T.6.0]':'occ_husb_6'})

y = np.ravel(y)

```

NOTE: The solution shared through Github should contain the source code used and the screenshot of the output.

3. Output

This assignment consists of 200 marks and has to be submitted in .ipynb/PDF format in the upcoming session for evaluation.