MACHINE LEARNING 1,2,3,4,5 & 6

Assignment 1

MACHINE LEARNING 1 to 6: Assignment 1

Table of Contents

1. Introduction

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

2.1. Problem Statement: Machine Learning 1

1. What are the three stages to build the hypotheses or model in machine learning?

2. What is the standard approach to supervised learning?

3. What is Training set and Test set?

4. What is the general principle of an ensemble method and what is bagging and boosting in ensemble method?

5. How can you avoid overfitting ?

2.2. Problem Statement: Machine Learning 2

Build the linear regression model using scikit learn in boston data to predict 'Price' based on other dependent variable.

Here is the code to load the data:

```
import numpy as np

import pandas as pd

import scipy.stats as stats

import matplotlib.pyplot as plt

import sklearn
```

```
from sklearn.datasets import load_boston

boston = load_boston()

bos = pd.DataFrame(boston.data)
```

2.3. Problem Statement: Machine Learning 3

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
```

2. 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)
```

2.4. Problem Statement: Machine Learning 4

Predicting Survival in the Titanic Data Set

We will be using a decision tree to make predictions about the Titanic data set from Kaggle. This data set provides information on the Titanic passengers and can be used to predict whether a passenger survived or not.

Loading Data and modules

```
import numpy as np

import pandas as pd

import seaborn as sb

import matplotlib.pyplot as plt

import sklearn

from pandas import Series, DataFrame

from pylab import rcParams

from sklearn import preprocessing

from sklearn.linear_model import LogisticRegression

from sklearn.cross_validation import train_test_split

from sklearn import metrics

from sklearn.metrics import classification_report
```

Url=
https://raw.githubusercontent.com/BigDataGal/Python-for-Data-Science/master/titanic-train.csv

```
titanic = pd.read_csv(url)

titanic.columns =
['PassengerId','Survived','Pclass','Name','Sex','Age','SibSp','Parch','
Ticket','Fare','Cabin','E mbarked']
```

You use only Pclass, Sex, Age, SibSp (Siblings aboard), Parch (Parents/children aboard), and Fare to predict whether a passenger survived.

2.5. Problem Statement: Machine Learning 5

In this assignment students will build the random forest model after normalizing the variable to house pricing from boston data set.

Following the code to get data into the environment:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import

train_test_split from

sklearn.preprocessing import

StandardScaler from sklearn import

datasets boston = datasets.load_boston()

features = pd.DataFrame(boston.data,
columns=boston.feature_names)

targets = boston.target
```

2.6. Problem Statement: Machine Learning 6

In this assignment students need to predict whether a person makes over 50K per year or not from classic adult dataset using XGBoost. The description of the dataset is as follows:

Data Set Information:

Extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0))

Attribute Information:

Listing of attributes: >50K, <=50K.

age: continuous.

workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

fnlwgt: continuous.

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: continuous.

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex: Female, Male.

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous.

native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Following is the code to load required libraries and data:

```
import numpy as np

import pandas as pd

train_set = pd.read_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.dat a', header = None)

test_set = pd.read_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test' , skiprows = 1, header = None)

col_labels = ['age', 'workclass', 'fnlwgt', 'education', 'education_num', 'marital_status', 'occupation','relationship', 'race', 'sex', capital_gain', 'capital_loss', 'hours_per_week', 'native_country', 'wage_class']

train_set.columns = col_labels

test_set.columns = col_labels
```

3. Output

This assignment consists of 1200 marks and and needs to be submitted in GitHub. You can follow GitHub submission guide provided to do the same.