

# 华东师范大学计算机科学技术系上机实践报告

|               |                |                 |
|---------------|----------------|-----------------|
| 课程名称：人工智能     | 年级：2018级       | 上机实践成绩：         |
| 指导教师：周爱民      | 姓名：汪子凡         | 创新实践成绩：         |
| 上机实践名称：机器学习入门 | 学号：10185102153 | 上机实践日期：2020/5/2 |
| 上机实践编号：No.3   | 组号：            | 上机实践时间：         |

## 一、问题介绍

- 1.安装python环境与jupyter notebook
- 2.上手sklearn，熟悉基本的数据载入，抽取和分析模型API
- 3.探索seaborn可视化，输出基本的图表

## 二、程序设计与算法分析

Python环境当前是3.8.2(64bit)，jupyter notebook使用的是anaconda自带的，通过pip命令可以安装sklearn, seaborn。

```
pip3 install seaborn
```

可以查找sklearn的官网手册来了解sklearn的基本用法：

“Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.”

可以通过import来使用sklearn和seaborn的模型算法与API：

```
from sklearn import datasets, linear_model
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.cross_validation import train_test_split
import seaborn as sns
```

由于本次实验没有具体的实验数据和要求，于是参考了一篇博客来练习sklearn和seaborn的基本用法(<https://www.cnblogs.com/pinard/p/6016029.html>)

在这个入门教程中使用了UCI大学公开的机器学习数据来练习线性回归(<http://archive.ics.uci.edu/ml/machine-learning-databases/00294/>)，里面是一个循环发电场的数据，共有9568个样本数据，每个数据有5列，分别是：AT（温度），V（压力），AP（湿度），RH（压强），PE（输出电力）。

可以先通过pandas包来载入数据，然后用成员变量shape和方法head () 来测试是否正确载入：

```
io = r'G:\课件\人工智能\作业\人工智能作业\CCPP\Folds5x2_pp.xlsx' #使用pandas读取数据
data = pd.read_excel(io, sheet_name = 0)

print("数据信息： ", data.shape)
data.head() #可以读取前5行数据
```

可以将前四项作为自变量X，最后的PE作为因变量Y，探测其中的线性关系，在其中可以通过sklearn调用train\_test\_split来划分训练集和测试集

```
X = data[['AT', 'V', 'AP', 'RH']]
Y = data[['PE']]
X.head()

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state=1)
#用sklearn的estimators train_test_split划分训练集与测试集
print("训练集信息： ", X_train.shape, Y_train.shape)
print("测试集信息： ", X_test.shape, Y_test.shape)
```

可以用scikit-learn的线性模型来拟合, 获取相应的参数,然后通过均方差 (Mean Squared Error, MSE) 或者均方根差(Root Mean Squared Error, RMSE)在测试集上的表现来评价这个模型的好坏。

```
linreg = LinearRegression()
#用sklearn的线性回归算法来拟合训练集
linreg.fit(X_train, Y_train)
print("训练集拟合的截距与系数： ", linreg.intercept_, linreg.coef_)

Y_pred = linreg.predict(X_test)
print("MSE:", metrics.mean_squared_error(Y_test, Y_pred))
print("RMSE:", np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
```

可以发现seaborn是一个是基于matplotlib的图形可视化python包，可以通过调用里面的API来完成作图，比如我简单地用图形来展示第一个参数AT与因变量PE的关系

```
sns.scatterplot(x = data['AT'], y = data['PE'])
```

### 三、实验结果

可以通过pip命令安装和查看本次实验需要的包：

```
C:\Users\admin>pip show seaborn
Name: seaborn
Version: 0.10.1
Summary: seaborn: statistical data visualization
Home-page: https://seaborn.pydata.org
Author: Michael Waskom
Author-email: mwaskom@nyu.edu
License: BSD (3-clause)
Location: f:\python\lib\site-packages
Requires: pandas, numpy, matplotlib, scipy
Required-by:
```

读取数据内容，展示数据信息：

```
io = r'G:\课件\人工智能\作业\人工智能作业\CCPP\Folds5x2_pp.xlsx' #使用pandas读取数据
data = pd.read_excel(io, sheet_name = 0)

print("数据信息: ", data.shape) #打印数据的维度，第一个参数表示行数，第二个参数表示列数
data.head() #可以读取前5行数据
```

数据信息: (9568, 5)

Out[47]:

|   | AT    | V     | AP      | RH    | PE     |
|---|-------|-------|---------|-------|--------|
| 0 | 14.96 | 41.76 | 1024.07 | 73.17 | 463.26 |
| 1 | 25.18 | 62.96 | 1020.04 | 59.08 | 444.37 |
| 2 | 5.11  | 39.40 | 1012.16 | 92.14 | 488.56 |
| 3 | 20.86 | 57.32 | 1010.24 | 76.64 | 446.48 |
| 4 | 10.82 | 37.50 | 1009.23 | 96.62 | 473.90 |

划分训练集和测试集，并使用线性模型通过最小二乘法进行拟合，可以发现训练集占了75%，测试集占了25%：

```
In [49]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state=1) #用sklearn的estimators train_test_split划分训练集与测试集
print("训练集信息: ", X_train.shape, Y_train.shape)
print("测试集信息: ", X_test.shape, Y_test.shape)
```

```
训练集信息: (7176, 4) (7176, 1)
测试集信息: (2392, 4) (2392, 1)
```

#### 运行线性模型

```
In [50]: linreg = LinearRegression() #用sklearn的线性回归算法来拟合训练集
linreg.fit(X_train, Y_train)
print("训练集拟合的截距与系数: ", linreg.intercept_, linreg.coef_)
```

```
训练集拟合的截距与系数: [460.05727267] [[-1.96865472 -0.2392946  0.0568309 -0.15861467]]
```

可以通过计算MSE, RMSE或者画图的方式来观察模型的好坏：

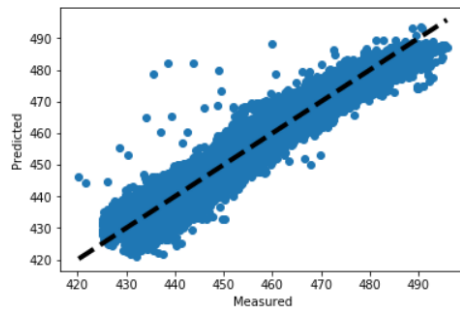
#### 模型分析评价

```
In [51]: Y_pred = linreg.predict(X_test)
print("MSE:", metrics.mean_squared_error(Y_test, Y_pred))
print("RMSE:", np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))

MSE: 20.837191547220353
RMSE: 4.564777272465805
```

#### 画图观察结果


```
In [52]: Y_pred = linreg.predict(X)
fig, ax = plt.subplots()
ax.scatter(Y, Y_pred)
ax.plot([Y.min(), Y.max()], [Y.min(), Y.max()], 'k--', lw=4)
ax.set_xlabel('Measured')
ax.set_ylabel('Predicted')
plt.show()
```

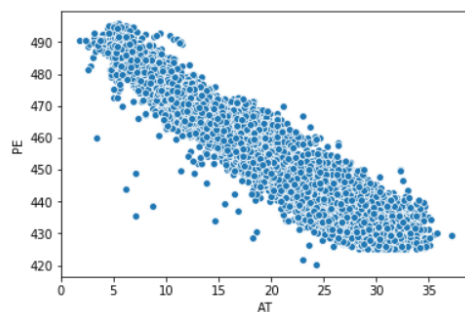


简单地使用seaborn,可以发现第一个参数AT与因变量PE之间存在负相关:

#### 使用seaborn画图

```
In [53]: sns.scatterplot(x = data['AT'], y = data['PE']) #通过绘图发现AT与PE呈负相关
```

Out[53]: <matplotlib.axes.\_subplots.AxesSubplot at 0x237f1d1f5600> 



## 四、附件