Lasso

1. 实验目的

了解 lasso 回归算法的原理，并且可以简单应用

2. 算法原理

LASSO 回归的特点是在拟合广义线性模型的同时进行变量筛选（variable selection）和复杂度调整（regularization）。因此，不论目标因变量（dependent/response varaible）是连续的（continuous），还是二元或者多元离散的（discrete），都可以用 LASSO 回归建模然后预测。这里的变量筛选是指不把所有的变量都放入模型中进行拟合，而是有选择的把变量放入模型从而得到更好的性能参数。复杂度调整是指通过一系列参数控制模型的复杂度，从而避免过度拟合（overfitting）。对于线性模型来说，复杂度与模型的变量数有直接关系，变量数越多，模型复杂度就越高。更多的变量在拟合时往往可以给出一个看似更好的模型，但是同时也面临过度拟合的危险。此时如果用全新的数据去验证模型（validation），通常效果很差。一般来说，变量数大于数据点数量很多，或者某一个离散变量有太多独特值时，都有可能过度拟合。

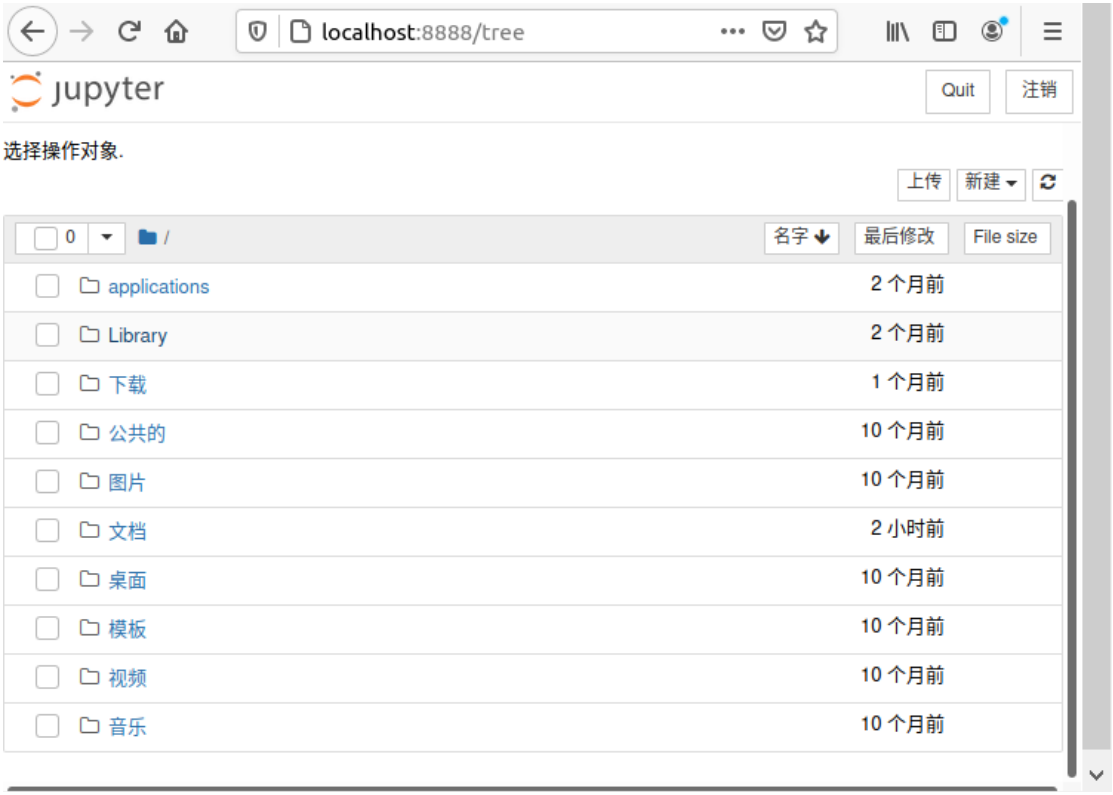
LASSO 回归复杂度调整的程度由参数 λ 来控制，λ 越大对变量较多的线性模型的惩罚力度就越大，从而最终获得一个变量较少的模型。 LASSO 回归与 Ridge 回归同属于一个被称为 Elastic Net 的广义线性模型家族。 这一家族的模型除了相同作用的参数 λ 之外，还有另一个参数 α 来控制应对高相关性（highly correlated）数据时模型的性状。 LASSO 回归 α=1，Ridge 回归 α=0，一般 Elastic Net 模型 0<α<1。

3. 实验环境

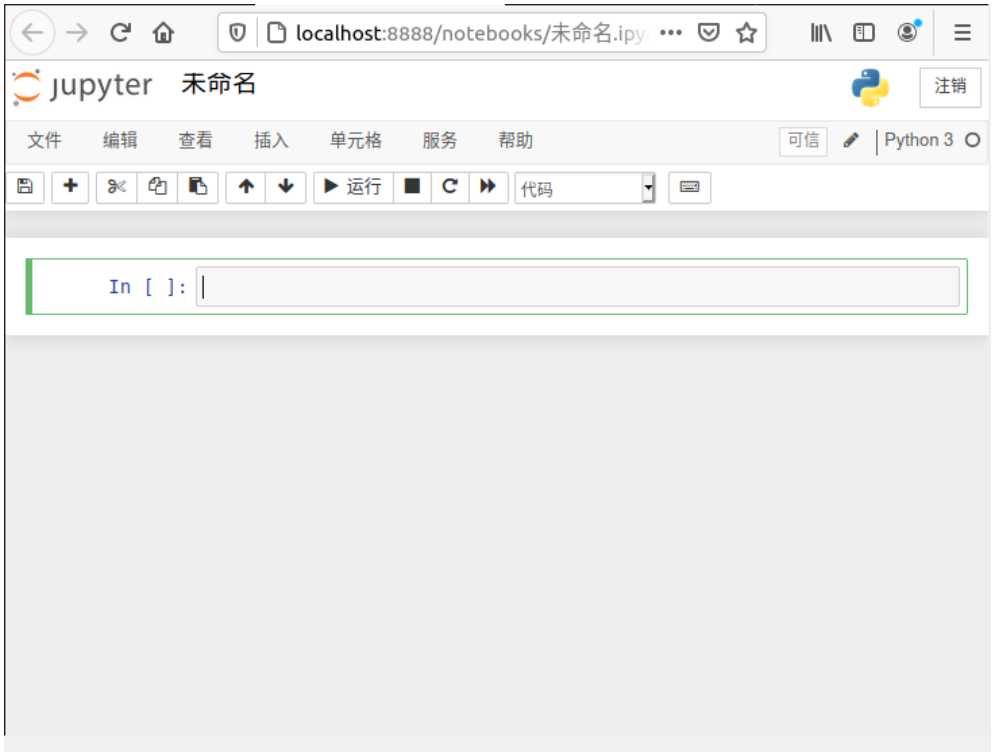
Ubuntu 20.04
Python 3.6
Jupyter notebook

4. 实验步骤

1)打开终端，然后输入 jupyter notebook，出现如下界面



2)选定特定文件夹，新建 ipynb 文件，在未命名出可重命名文件



5. 实操

Step 1:数据预处理

1. 导入库
2. 导入数据集
3. 展示数据集
4. 分割特征

```
#导入库
from sklearn.datasets import load_boston
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#导入数据集
df=load_boston()


# 展示数据集
dataset = pd.DataFrame(df.data)
print(dataset.head())
dataset.columns=df.feature_names
dataset.head()

dataset["price"]=df.target
dataset.head()

# 分割特征
x=dataset.iloc[:,:-1] ##independent features
y=dataset.iloc[:,-1]  ##dependent features
```

Step 2:lasso 模型

```
from sklearn.linear_model import Lasso
from sklearn.model_selection import GridSearchCV
lasso=Lasso()
parameters={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-
2,1,5,10,20,30,40,45,50,55,100]}
lasso_regressor=GridSearchCV(lasso,parameters,scoring='neg_mean_squared
_error',cv=5)

lasso_regressor.fit(x,y)
print(lasso_regressor.best_params_)
print(lasso_regressor.best_score_)

# 分割数据集
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,rand
om_state=0)
```

Step 3:预测并绘制

```
prediction_lasso=lasso_regressor.predict(x_test)
import seaborn as sns
sns.distplot(y_test-prediction_lasso)
```