机器学习 上机3 北京PM2.5预测

姓名： 张泽群 学号： 19049100002 班级：2班

1. 数据集分析与预处理

1.1 数据集分析

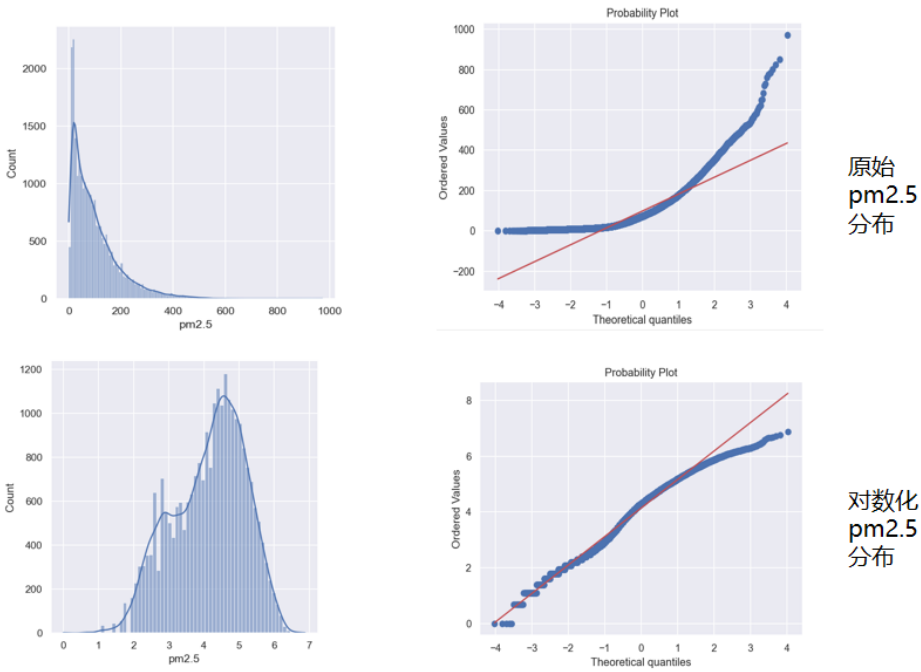
本次实验所用数据集是北京PM2.5数据集，数据具有year(年)，month（月）day（日），hour（小时），DEWP（露点），TEMP（温度），PRES（大气压）， cbwd（风向），lws（风速），ls（累积雪时），lr（累积雨时）11个维度的数据特征，以及需要回归预测的pm2.5值和数据记录标识符 NO。

其中，数据集划分为训练、验证、测试集，样本数量分别为26294、8765、8765。

在原始数据集中存在的数据缺失的问题，部分数据的某些特征值有所缺失，这也是数据预处理需要解决的问题。

此外，通过统计训练数据集中标签的分布情况，可以看到大量的样本pm2.5聚集在5-300的区间中，占据了整个训练数据集的95%以上。该数据集也存在着样本分布不均衡的问题。

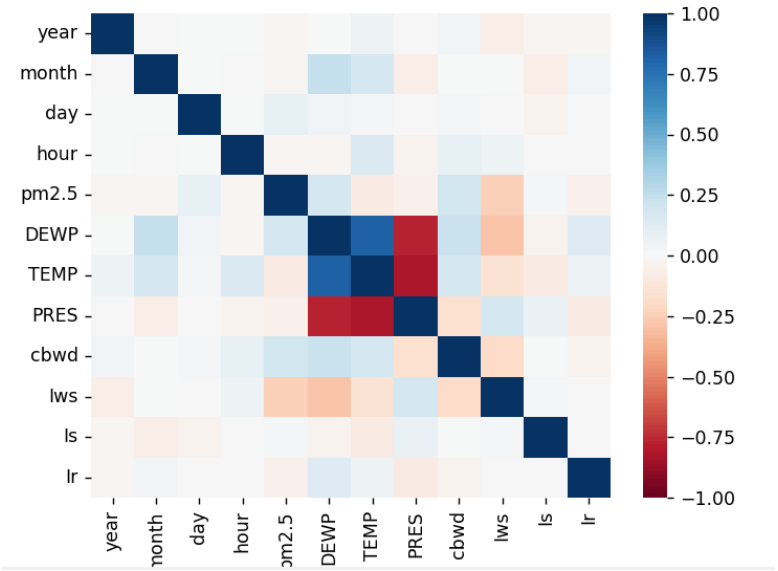
但是通过探索数据，我们发现将pm2.5值进行对数化后，其分布又近似正态分布，且其值的顺序排列近似于线性，这为我们寻找回归方法提供了某一种可能。



原始pm2.5分布

对数化pm2.5分布

再看11个数据维度以及pm2.5的相关性热力图，我们发现仅有DEWP,TEMP和PRES具有明显的正负相关性，而PM2.5并没有与其他数据特征有较为明显的相关性。



1.2 数据预处理

1.2.1 清洗数据

数据集中所有的缺失值都集中在pm2.5一列，并且数据的缺失都是按时间序列的一连串缺失，可能是因为气象站对该时间点没有测验而造成的。因此先将所有NAN值赋值为0，然后在后续结合时序处理的时候将有当前时间pm2.5，下一个时间步的pm2.5的缺失值的数据给去除，从而可以达到洗去缺失值的目的。

```
def delete_na(data):
    i = 0
    temp = len(data)
    while i < temp:
        if data[i, 0] == 0 or data[i, -1] == 0:
            index = [i]
            data = np.delete(data, index, 0)
            temp = temp - 1
            i = i - 1
        i = i + 1
    return data
```

1.2.2 对于非数值数据进行整数编码

其中，cbwd 风向特征是标签数据，我们可以将该数据进行整数编码，从而将其转化为数值型的数据。共有4种风向，因此将其编码为0，1，2，3

```
# integer encode direction
encoder = LabelEncoder()
values[:, 4] = encoder.fit_transform(values[:, 4])
```

1.2.3 添加数据时序序列维度

实际上，pm2.5浓度的提升或下降并不是一蹴而就的，而是与前一个时间步的污染测量和天气条件的情况紧密相关，因此预测pm2.5值，除了需要原始数据集的8个维度的代表天气情况，还需要利用4个时间维度的特征来构建时间序列，并且将前一个时间步的污染测量作为新一个维度的数据特征来进行回归预测。

2. 模型建立过程

(1). 首先针对处理完毕的数据集，我们尝试使用线性回归模型直接进行回归预测：

```
# 线性回归
 lr = LinearRegression()
 lr.fit(train_data, train_value)
```

所得到的**MSE = 6400**,显然这样直接的处理获得的均方差非常之大，由相关性热力图可以看出这几个特征维度显然与pm2.5并非线性相关，直接采用线性回归的预测方式显然不能得出较好的结果。

(2). 由于单一的线性回归模型无法得到较好的结果，我想到利用stacking方法，通过训练出多个异质弱学习器，生成一个新的数据集用于训练元学习器，从而达成将多个弱学习器结合为强学习器，获得更优的回归效果。

利用staking**模型结合lasso回归，岭回归以及随机森林回归算法**，同时利用网格搜索在验证集上调整超参数，最终获得最优超参数的结合模型。

```
RANDOM_SEED = 42
ridge = Ridge(random_state=RANDOM_SEED)
lasso = Lasso(random_state=RANDOM_SEED)
rf = RandomForestRegressor(random_state=RANDOM_SEED)
stack = StackingCVRegressor(regressors=(lasso, ridge),
                            meta_regressor=rf,
                            random_state=RANDOM_SEED,
                            use_features_in_secondary=True)
params = {'lasso__alpha': [0.1, 1.0, 10.0],
          'ridge__alpha': [0.1, 1.0, 10.0]}
grid = GridSearchCV(
    estimator=stack,
    param_grid={
        'lasso__alpha': [x / 5.0 for x in range(1, 10)],
        'ridge__alpha': [x / 20.0 for x in range(1, 10)],
        'meta_regressor__n_estimators': [10, 100]
    },
    cv=5,
    refit=True
)
grid.fit(validation_data, validation_value)
print(grid.best_params_)
model = grid.best_estimator_
```

模型运行结果： 可见得到的 MSE = 2200，混合模型确实相较于直接使用线性回归模型能获得更优的效果，但是现在的均方差仍然非常大，没有达到令人满意的效果。

```
D:\Python37\python.exe F:/python/ML3/test2.py
{'lasso__alpha': 0.6, 'meta_regressor__n_estimators': 100, 'ridge__alpha': 0.35}

[168.  82. 307. ...  78. 472. 195.]
[152.29 222.66 222.99 ...  72.84 355.45 105.15]
lr MSE =  2237.8374085304654
```

除了以上三个模型结合的stacking模型，还尝试利用其他的模型运用stacking方法，但得到的结果都在2000以上，因此最终决定结合时序特征进行回归预测。

(3). 结合时序特征进行回归预测

去除na值并且结合时序特征进行回归预测后**MSE = 525**，可见准确率上升了许多。

```
D:\Python37\python.exe F:/python/ML3/regression.py
[142.  87. 347. ...  90. 453. 220.]
[169.3490252   82.9393794  295.97522128 ...  76.56803269 457.19089116
 192.89215279]
MSE =  525.8175503484808
MAE =  12.605887909081924
R2 =  0.9392789252458504

Process finished with exit code 0
```

3. 实验结果

从最终结果上看，模型训练后在测试集上的**MSE=525** 左右浮动，MAE = 12.5, R2 =0.939。

```
D:\Python37\python.exe F:/python/ML3/regression.py
[142.  87. 347. ...  90. 453. 220.]
[169.3490252   82.9393794  295.97522128 ...  76.56803269 457.19089116
 192.89215279]
MSE =  525.8175503484808
MAE =  12.605887909081924
R2 =  0.9392789252458504

Process finished with exit code 0
```

```
[142.  87. 347. ...  90. 453. 220.]
[171.42886674  82.16213742 296.59014219 ...  75.50372058 456.74015532
 192.66327753]
MSE =  525.5976971315632
MAE =  12.560905838892694
R2 =  0.9393043137548694
```

同时利用joblib导出模型为model_3.model。

```
# 导出模型
joblib.dump(sclf, 'model_3.model')
```

4. 讨论与结论

4.1 评价指标的讨论

回归算法中常用的评价指标是：MAE、MSE、RMSE、R^2

(1). MSE (Mean Square Error: 均方误差)

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

(2). MAE (Mean Absolute Error: 平均绝对误差)

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|(y_i - \hat{y}_i)\right|$$

(3). RMSE (Root Mean Square Error: 均方根误差)

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

(4). R^2 (Mean Absolute Error: 平均绝对误差)

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=0}^{m}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=0}^{m}(y^{(i)} - \overline{y})^2} = \frac{\mathrm{SSR}}{\mathrm{SST}}$$

$$= 1 - \frac{\mathrm{SSE}}{\mathrm{SST}}$$

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=0}^{m}(y^{(i)} - \hat{y}^{(i)})^2/m}{\sum_{i=0}^{m}(y^{(i)} - \overline{y})^2/m}$$

$$= 1 - \frac{\mathrm{MSE}}{\mathrm{Var}}$$

用于度量因变量的变异中可由自变量解释部分所占的比例，以此来判断统计模型的解释力。它将已解释的方差除以总方差，**代表了总方差被预测变量所解释或决定的比率**。越接近于1，说明模型的效果越好。

对于上述指标，MAE**看重真实值和预测值的绝对误差,对极端值比较敏感**，而MSE和RMSE更**看重真实值和预测值的差的平方。**

4.2 stacking方法中L1、L2正则化的讨论

在Stacking方法中我们结合了L1、L2正则化，即Lasso回归和岭回归。

```
    ridge = Ridge()  # 岭回归(L2正则化)
    lasso = Lasso()  # Lasso回归(L1正则化)
```

通过控制变量法的实验，我们发现单使用lasso回归和ridge回归都无法提升回归的MSE，但结合两者后，回归的MSE具有明显的上升。

L1正则化是指在损失函数中加入权值向量w的绝对值之和，L1的功能是使权重稀疏。

L2 在损失函数中加入权值向量w的平方和，L_2的功能是使权重平滑，防止过拟合。