Datawhale 零基础入门数据挖掘 Task4 建模与预测

分享人：小雨姑娘

目录

contents

自我介绍

Datawhale

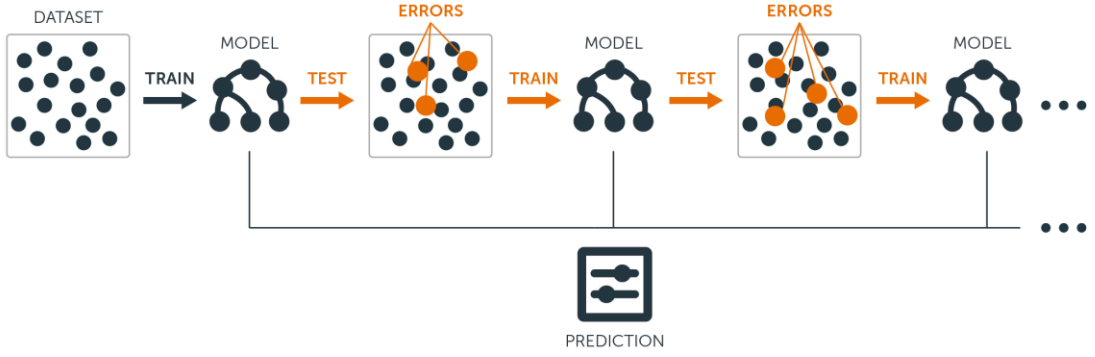
- **小雨姑娘**

  青岛大学大四本科生，已收到北美计算机博士全奖offer。
  知乎专栏：小雨姑娘的机器学习笔记

Part 1 基础知识

- **统计学习分类**
  1. **监督学习:**
     利用一组带标签的数据, 学习从输入到输出的映射, 然后将新数据用这种映射关系可以得到映射结果, 达到分类或者回归的目的。线性回归、决策树、SVD等
  2. **非监督学习**
     输入数据没有被标记，也没有确定的结果。K-means聚类、层次聚类等
  3. **半监督学习**
     在实际情况中，获取的数据大部分都是无标签的，人们企图加入一些人为标注的样本，使得无标签的数据通过训练自动获取标签，这相当于对无监督学习是一种改进。生成模型算法等
  4. **强化学习**
     用于描述和解决智能体在与环境的交互过程中通过学习策略以达成回报最大化或实现特定目标的问题。Q-Learning，隐马尔可夫模型

- 常见的监督学习模型

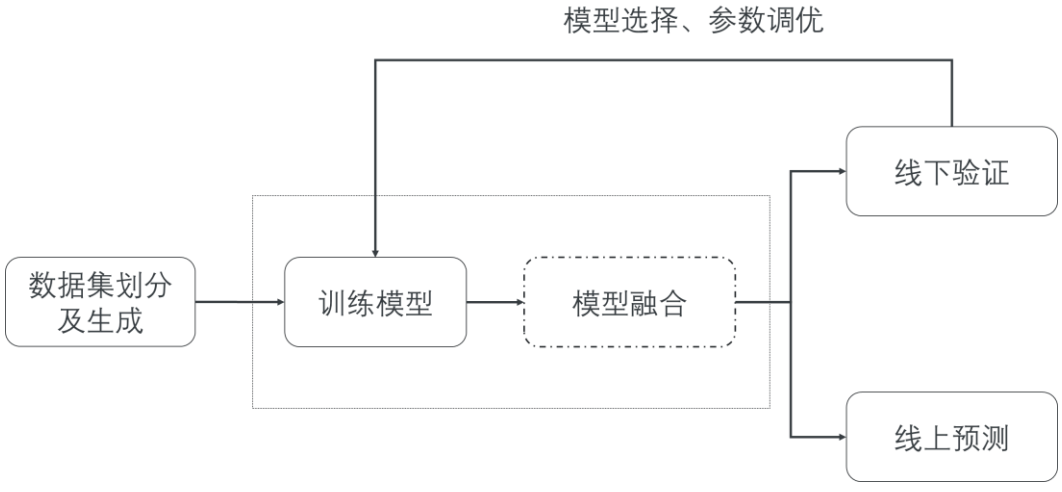


1. 线性模型
2. 决策树
3. 神经网络
4. 支持向量机
5. 贝叶斯分类
6. 集成学习模型（RandomForest, GBDT, XGBoost, LightGBM）

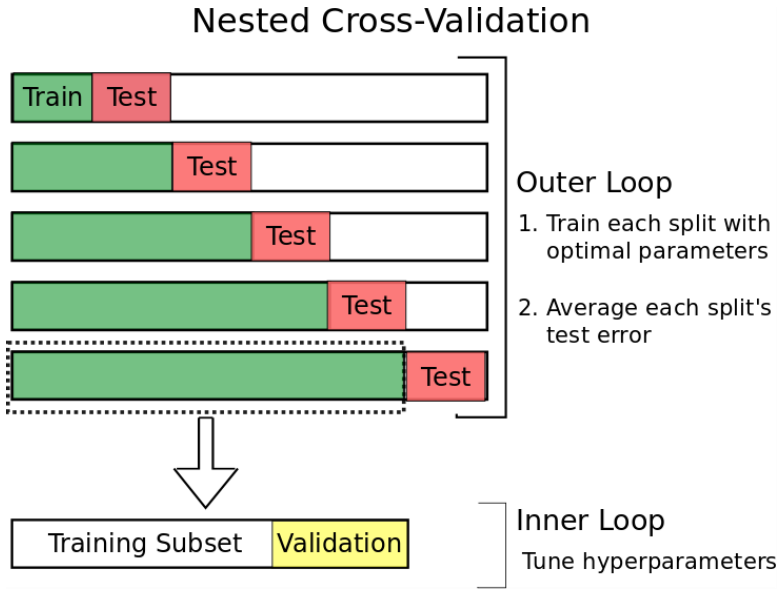
- **基本概念**
  1. **模型、策略与算法**
  2. **评价函数**
  3. **目标函数**
  4. **过拟合与欠拟合**
  5. **正则化**
  6. **交叉验证**
  7. **泛化能力**

- **推荐材料**
  1. 《机器学习（西瓜书）》周志华
  2. 《统计学习方法》李航
  3. 《深度学习（花书）》Ian Goodfellow、Yoshua Bengio、Aaron Courville
  4. 《Python大战机器学习》

- 训练及预测的一般流程



- **验证方法**
  1. 训练集、线下验证集、线下测试集、线上测试集
  2. 无时序的数据集：简单划分、交叉验证划分等
  3. 有时序的数据集：需考虑时序，nested交叉验证划分等

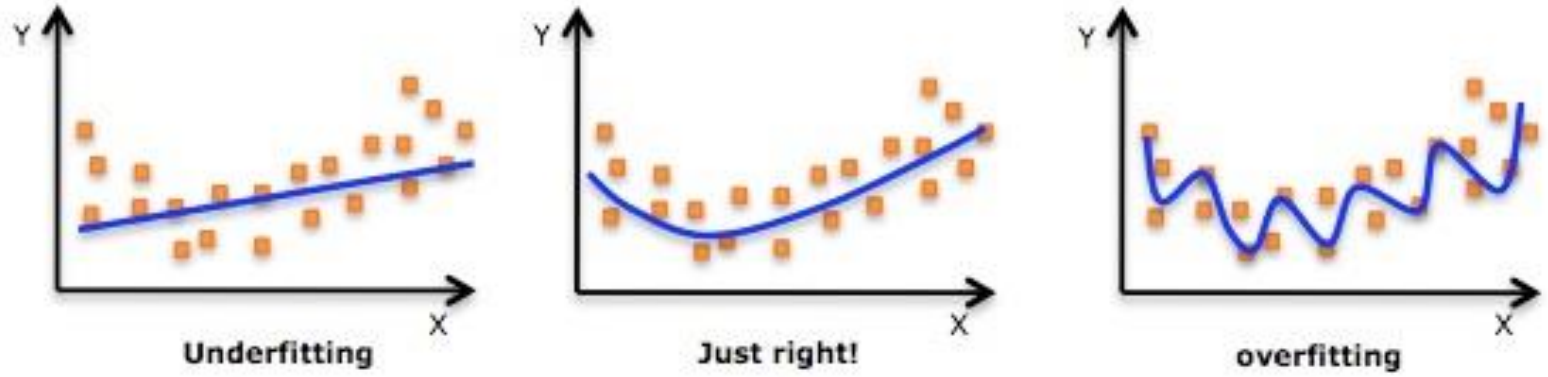


- 模型选择
  1. 依据在验证集上的效果选择
  2. 除了关注效果的均值，还要关注稳健性
  3. 还需考虑线上效果；可将线上效果视为一折数据

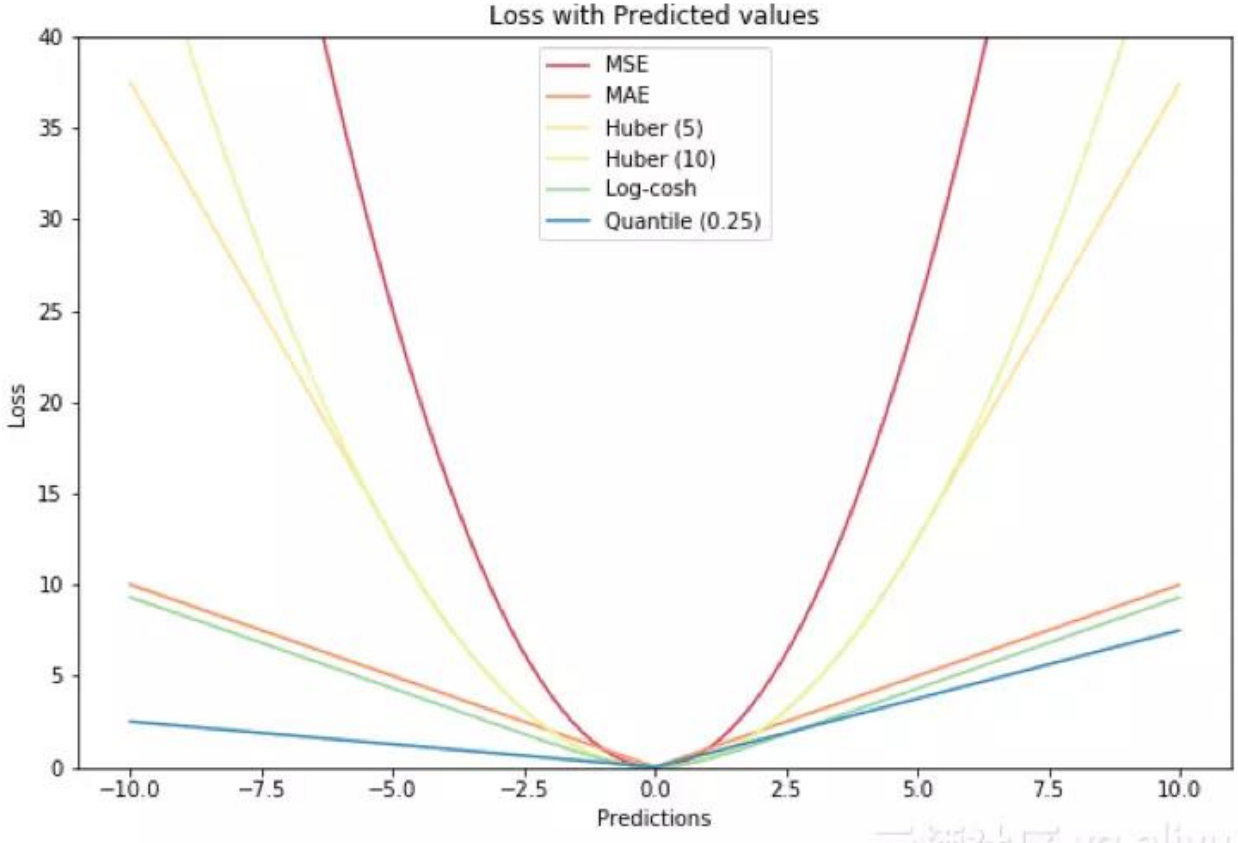
- 参数调优
  1. 不建议将精力放在参数调优上；容易过拟合
  2. 大体的设置参数即可
  3. 应将精力重点放在特征工程；其次是模型融合

- 过拟合与欠拟合

考雅思口语的时候，有些人背了很多口语话题答案，结果考试的时候没考到他背的题，只得了5.5分。因为他非常复杂的记住了刻板的对话，而没有真正抽象出答题的方法。



- 目标函数

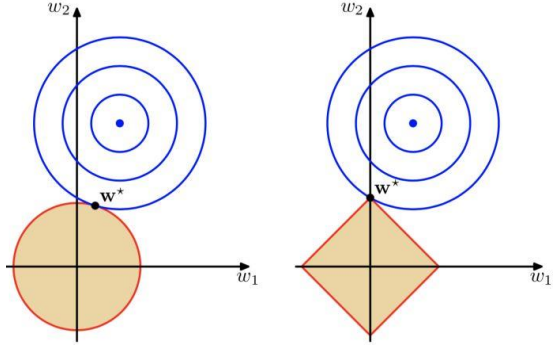


- 正则化
  - 奥卡姆剃刀原理

    Do not multiply entities beyond necessity, but also do not reduce them beyond necessity

  - 数学原理

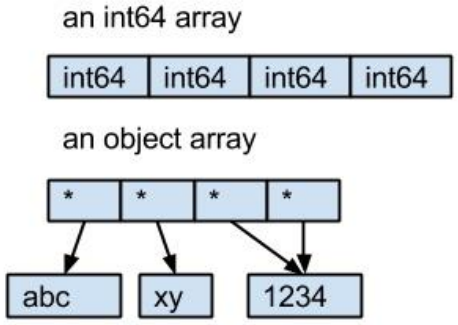
$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Part 2 Q&A

Q&A

Datawhale



- Q：关于category与object类型？

  A：

- Q：小雨大佬，请问怎么解决每次改特征工程，跑模型，搞到最后，代码好混乱，都不知道哪个模型效果好了。举个例子，我用了N种特征工程，M种模型，最后得出了N*M种解决方案。好复杂啊...

  A：

- Q：在EDA中，你提到johnsonsu的拟合分布的效果最好那怎么把y转换成johnsonsu分布呢？还有出结果后，怎么转变回来？

  A：

Q&A

Datawhale

Johnson 系统	Johnson 分布	正态变换	参数约束	X约束
S_B	$k_1 = ln\left(\frac{x-\varepsilon}{\lambda+\varepsilon-x}\right)$	$z = \gamma + \eta ln\left(\frac{x-\varepsilon}{\lambda+\varepsilon-x}\right)$	$\eta,\lambda>0$ $-\infty<\gamma<\infty$ $-\infty<\varepsilon<\infty$	$\varepsilon<x<\varepsilon+\gamma$
S_L	$k_2 = ln(x-\varepsilon)$	$z = \gamma + \eta ln(x-\varepsilon)$	$\eta>0$ $-\infty<\gamma<\infty$ $-\infty<\varepsilon<\infty$	$x>\varepsilon$
S_U	$k_3 = \sinh^{-1}\left(\frac{x-\varepsilon}{\lambda}\right)$	$z = \gamma + \eta \sinh^{-1}\left(\frac{x-\varepsilon}{\lambda}\right)$	$\eta,\lambda>0$ $-\infty<\gamma<\infty$ $-\infty<\varepsilon<\infty$	$-\infty<x<\infty$

对于Johnson变换，有两个问题需要解决，一是在三个变换中选择哪一个，二是如何计算变换的参数。

Ref: https://zhuanlan.zhihu.com/p/26869997

Q&A

- Q：呃～你是赛题设计者，可能了解相关信息比我们要多一些～如果作为一名参赛者，面对完全陌生的数据，仅仅告诉了地区信息，已脱敏，你是怎样知道前两位就是省份呢？会不会直接上来就直接切个片呢？
  A：

- Q：那这样在训练集上进行删除异常值 但是在测试集上没有做这个步骤 不是很容易会造成过拟合吗？
  A：

- Q：xgb.cv用交叉验证调好参后，怎么转到xgb.train呢，调好的参数用xgb.train还需要交叉验证训练吗？
  A：

Q&A

- Q：模型调参有什么技巧吗？比如每个参数设置范围，另外xgboost 网格搜索调参速度很慢，有什么方式可以提高速度吗？
  A：

- Q：小雨大佬，可以深入讲下特征工程那里的方差过滤、卡方过滤、嵌入法、包装法吗，这是不是需要统计学知识啊，需要学习统计学码
  A：

- Q：小雨大佬，xgboost源码好读吗，网上搜了一下说是C++的，有python版的吗，阅读这个源码，需要什么知识积累呢，需要先读陈天奇大佬的论文原文吗

  A：

Q&A

- Q：小雨大佬，针对一个城市的二手车数据训练好模型，可以直接应用于其他城市的二手车数据上吗（一样特征工程处理后）？若可以，可能会有什么结果呢？如不可以，那一个城市是不是需要训练一个模型、调一次参数啊？

  A:

- Q：小雨大佬，针对一批数据调参训练好模型后，若新数据的特征维数有新增的，应该怎么处理呢，还可以使用训练好的模型预测吗，还是说需要训练新的模型？

  A:

- Q：小雨大佬，直播时可以分享下你的参加竞赛的经历吗，通过这次竞赛发现，自己学的各个模块和理论知识，一到实际比赛（针对不同的数据）时不知道怎么用了，好多学过的还是需要现查，然后参加竞赛需要局限于某一类型的竞赛吗，多参加不同类型的竞赛（图像、nlp、预测）对提升实战能力帮助大呢，还是说尽量针对一种类型的竞赛来参加，进步较大？

  A:

Part 3 代码实战

代码实战

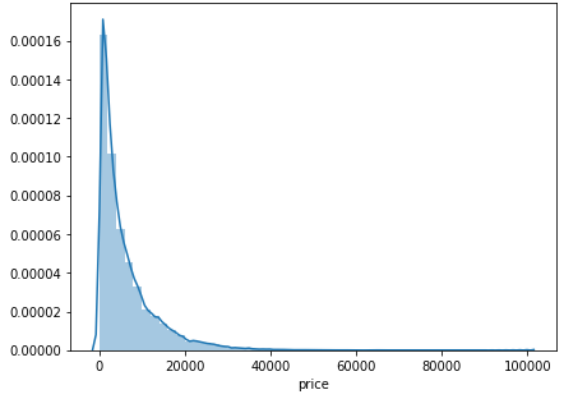
- **读取特征**

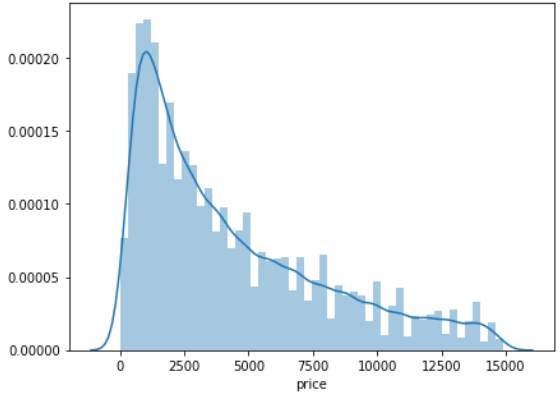
  祖传降低内存代码，通过精度转换降低内存损耗

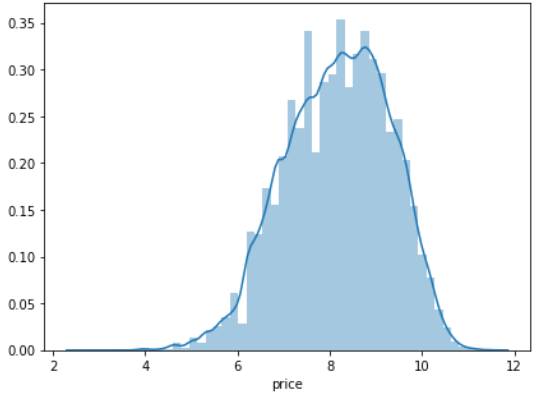
```python
def reduce_mem_usage(df):
    for col in df.columns:
        col_type = df[col].dtype

        if col_type != object:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max < np.finfo(np.float16).max:
                    df[col] = df[col].astype(np.float16)
                elif c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).max:
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)
        else:
            df[col] = df[col].astype('category')
    return df
```

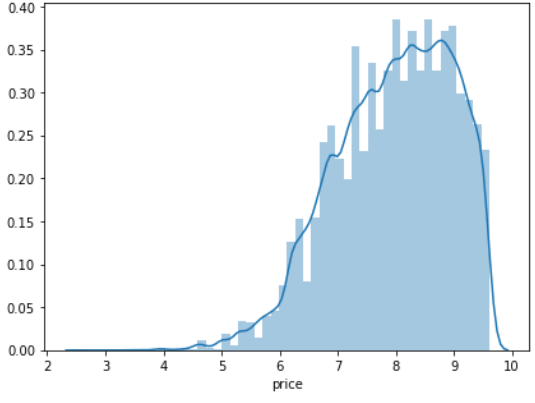
代码实战

- 对Price进行转换



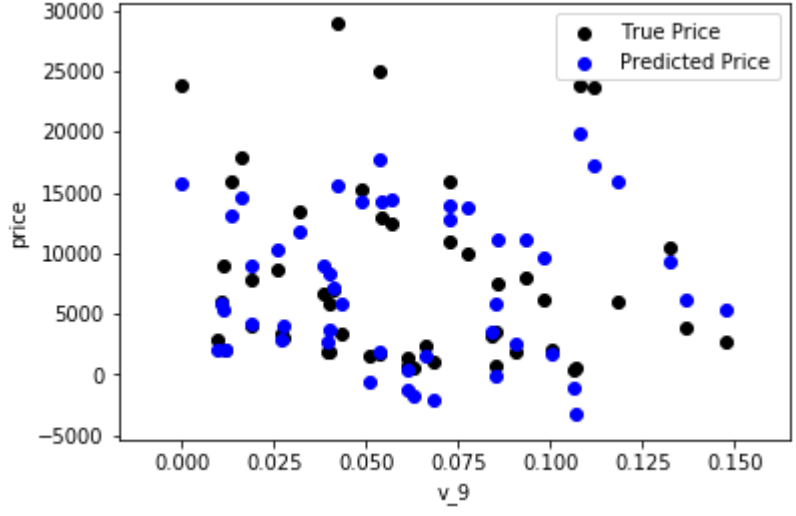


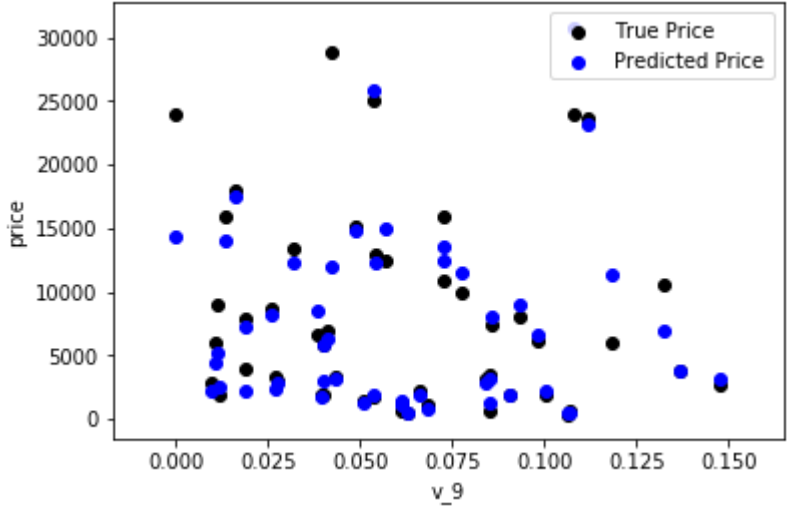




代码实战

- 对Price进行转换





代码实战

- 函数装饰器

```
def log_transfer(func):
    def wrapper(y, yhat):
        result = func(np.log(y), np.nan_to_num(np.log(yhat)))
        return result
    return wrapper
```

- Make_scorer

```
scores = cross_val_score(model,
                         X=train_X, y=train_y,
                         verbose=1, cv = 5,
                         scoring=make_scorer(log_transfer(mean_absolute_error)))
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    1.1s finished
```

代码实战

Datawhale

- 五折交叉验证

使用线性回归模型，对未处理标签的特征数据进行五折交叉验证（Error 1.36）

In [21]:
```
print('AVG:', np.mean(scores))
```

```
AVG: 1.3641908155886227
```

使用线性回归模型，对处理过标签的特征数据进行五折交叉验证（Error 0.19）

In [22]:
```
scores = cross_val_score(model,
                         X=train_X, y=train_y_ln,
                         verbose=1, cv = 5,
                         scoring=make_scorer(mean_absolute_error))
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    1.1s finished
```

In [23]:
```
print('AVG:', np.mean(scores))
```

```
AVG: 0.19382863663604424
```

In [24]:
```
scores = pd.DataFrame(scores.reshape(1,-1))
scores.columns = ['cv' + str(x) for x in range(1, 6)]
scores.index = ['MAE']
scores
```

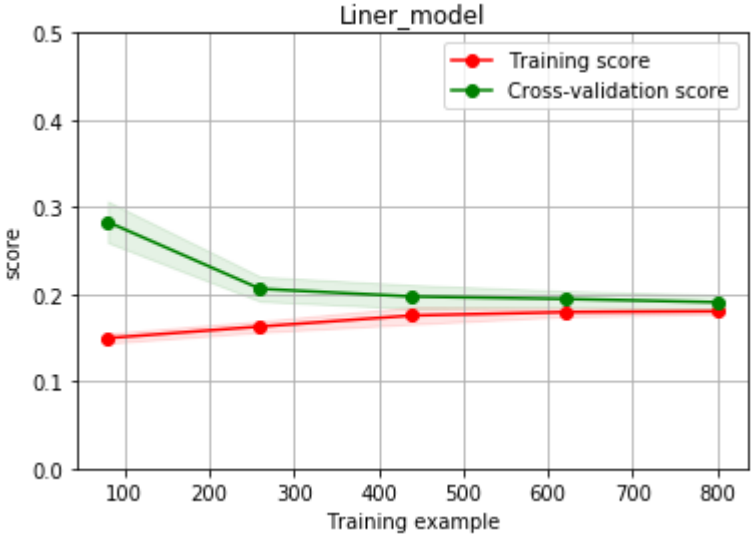
Out[24]:

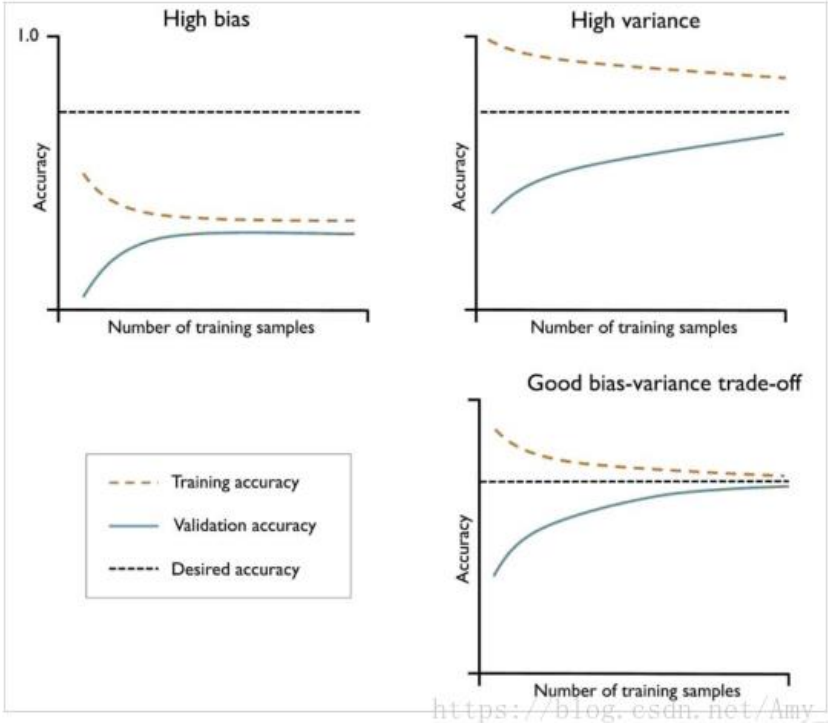
	cv1	cv2	cv3	cv4	cv5
MAE	0.191642	0.194986	0.192737	0.195329	0.19445

代码实战

Datawhale

- 绘制学习率曲线

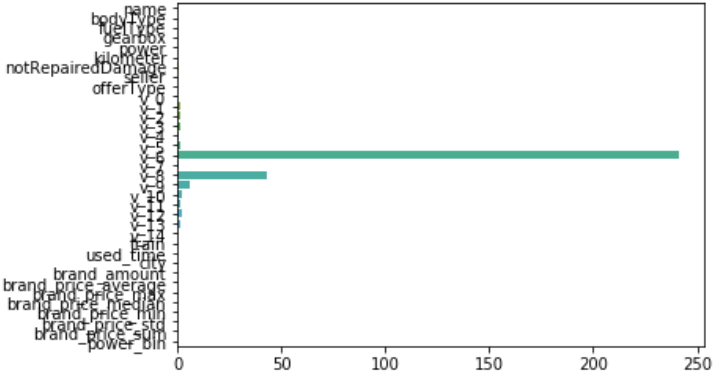


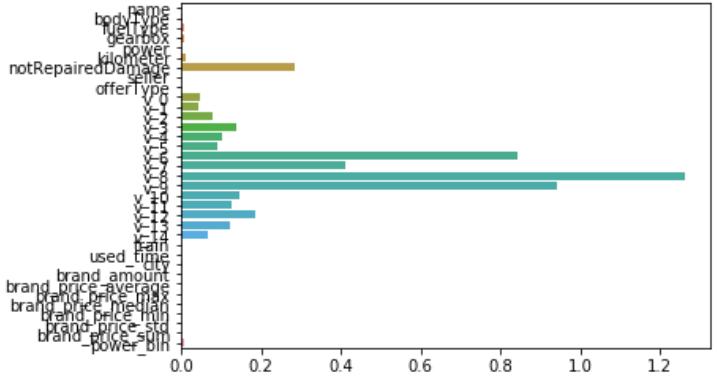


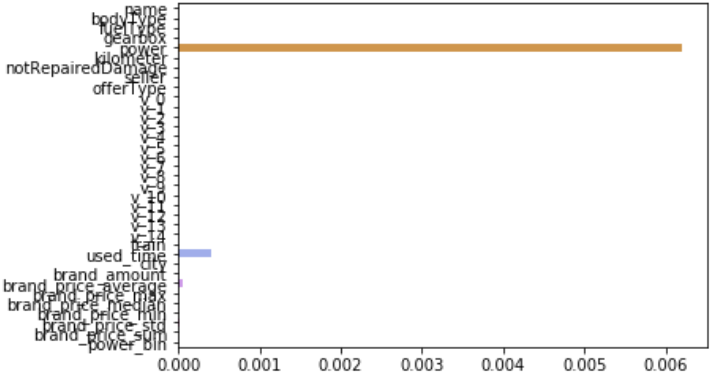
代码实战

Datawhale

	LinearRegression	Ridge	Lasso
cv1	0.191642	0.195665	0.382708
cv2	0.194986	0.198841	0.383916
cv3	0.192737	0.196629	0.380754
cv4	0.195329	0.199255	0.385683
cv5	0.194450	0.198173	0.383555







代码实战

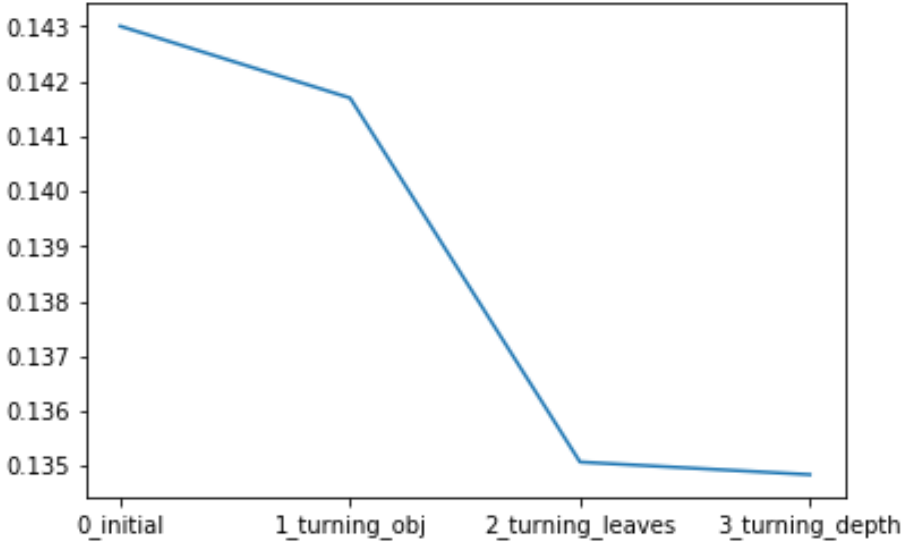
- 多种模型对比

```
models = [LinearRegression(),
          DecisionTreeRegressor(),
          RandomForestRegressor(),
          GradientBoostingRegressor(),
          MLPRegressor(solver='lbfgs', max_iter=100),
          XGBRegressor(n_estimators = 100, objective='reg:squarederror'),
          LGBMRegressor(n_estimators = 100)]
```

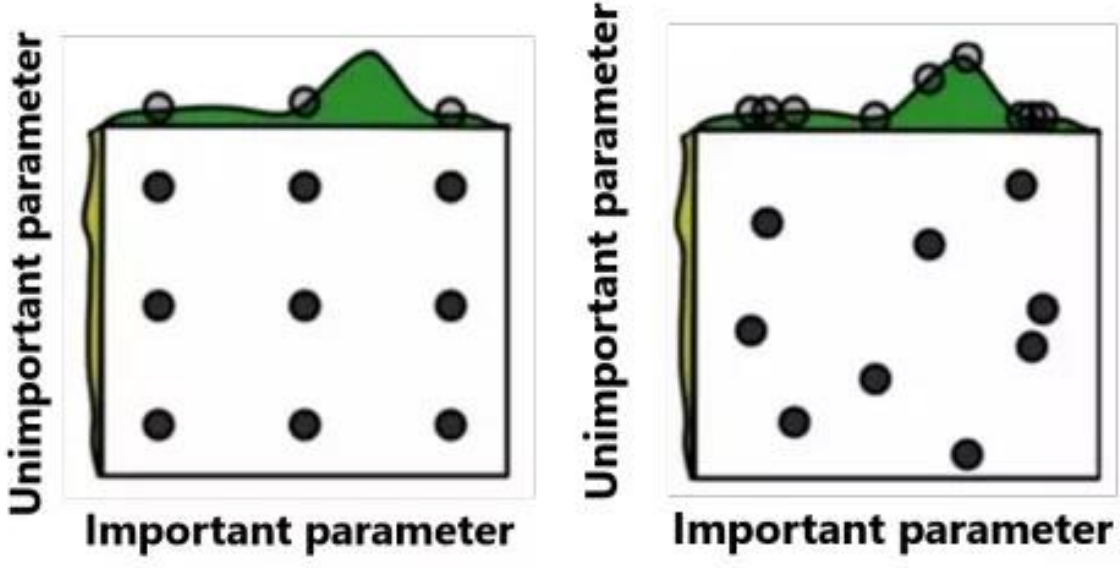
	LinearRegression	DecisionTreeRegressor	RandomForestRegressor	GradientBoostingRegressor	MLPRegressor	XGBRegressor	LGBMRegressor
cv1	0.191642	0.184566	0.136266	0.168626	124.299426	0.168698	0.141159
cv2	0.194986	0.187029	0.139693	0.171905	257.886236	0.172258	0.143363
cv3	0.192737	0.184839	0.136871	0.169553	236.829589	0.168604	0.142137
cv4	0.195329	0.182605	0.138689	0.172299	130.197264	0.172474	0.143461
cv5	0.194450	0.186626	0.137420	0.171206	268.090236	0.170898	0.141921

代码实战

- 贪心调参


0.143
0.142
0.141
0.140
0.139
0.138
0.137
0.136
0.135
0_initial
1_turning_obj
2_turning_leaves
3_turning_depth

- 网格搜索与随机搜索



(a) Grid Search (b) Random Search

代码实战

- 贝叶斯调参

iter	target	max_depth	min_ch...	num_le...	subsample
1	0.8649	89.57	47.3	55.13	0.1792
2	0.8477	99.86	60.91	15.35	0.4716
3	0.8698	81.74	83.32	92.59	0.9559
4	0.8627	90.2	8.754	43.34	0.7772
5	0.8115	10.07	86.15	4.109	0.3416
6	0.8701	99.15	9.158	99.47	0.494
7	0.806	2.166	2.416	97.7	0.224
8	0.8701	98.57	97.67	99.87	0.3703
9	0.8703	99.87	43.03	99.72	0.9749
10	0.869	10.31	99.63	99.34	0.2517
11	0.8703	52.27	99.56	98.97	0.9641
12	0.8669	99.89	8.846	66.49	0.1437
13	0.8702	68.13	75.28	98.71	0.153
14	0.8695	84.13	86.48	91.9	0.7949
15	0.8702	98.09	59.2	99.65	0.3275
16	0.87	68.97	98.62	98.93	0.2221
17	0.8702	99.85	63.74	99.63	0.4137
18	0.8703	45.87	99.05	99.89	0.3238
19	0.8702	79.65	46.91	98.61	0.8999
20	0.8702	99.25	36.73	99.05	0.1262
21	0.8702	85.51	85.34	99.77	0.8917
22	0.8696	99.99	38.51	89.13	0.9884
23	0.8701	63.29	97.93	99.94	0.9585
24	0.8702	93.04	71.42	99.94	0.9646
25	0.8701	99.73	16.21	99.38	0.9778
26	0.87	86.28	58.1	99.47	0.107
27	0.8703	47.28	99.83	99.65	0.4674
28	0.8703	68.29	99.51	99.4	0.2757
29	0.8701	76.49	73.41	99.86	0.9394
30	0.8695	37.27	99.87	89.87	0.7588

Datawhale

一个专注于AI领域的开源组织