Datawhale 零基础入门数据挖掘 Baseline & Task1 赛题理解

分享人：ML67&AI蜗牛车

目录

contents

Part 1 赛题理解

作者：AI蜗牛车

赛题理解
-磨刀不误砍柴工

	状态	举办方	赛季1	奖金	参赛队伍	
零基础入门数据挖掘 - 二手车交易价格预测	进行中	Datawhale TIANCHI天池	2020-04-11	¥0	1572	报名

理解赛题是不是把一道赛题的背景介绍读一遍就OK了呢?

Datawhale 零基础入门数据挖掘-Task1 赛题理解

一、 赛题理解

Tip:此部分为零基础入门数据挖掘的 Task1 赛题理解 部分，为大家入门数据挖掘比赛提供一个基本的赛题入门讲解，欢迎后续大家多多交流。

赛题：零基础入门数据挖掘 - 二手车交易价格预测

地址：https://tianchi.aliyun.com/competition/entrance/231784/introduction?spm=5176.12281957.1004.1.38b02448ausjSX

赛题理解

Datawhale

一、赛题背景

本次新人赛是Datawhale与天池联合发起的0基础入门系列赛事第一场 —— 零基础入门数据挖掘之二手车交易价格预测大赛。

赛题以二手车市场为背景，要求选手预测二手汽车的交易价格，这是一个典型的回归问题。通过这道赛题来引导大家走进AI数据竞赛的世界，主要针对于于竞赛新人进行自我练习、自我提高。

二、赛制说明

本次赛事分为两个阶段，分别为正式赛及长期赛。

正式赛（3月12日 - 4月11日）

1. 报名成功后，选手下载数据，在本地调试算法，通过赛题页左侧提交入口提交结果；
2. 提交后将进行实时评测；每天每支队伍可提交2次；排行榜每小时更新，按照评测指标得分从高到低排序；排行榜将选择历史最优成绩进行展示；
3. 最后一次排行榜更新时间为4月11日晚上20点，将以该榜单成绩作为依照，评选出正式赛期间的奖项名次，予以奖励。

长期赛（4月11日以后）

自4月1日开始，本场比赛将长期开放，报名和参赛无时间限制。

赛题理解

Datawhale

一、赛题数据

赛题以预测二手车的交易价格为任务，数据集报名后可见并可下载，该数据来自某交易平台的二手车交易记录，总数据量超过40w，包含31列变量信息，其中15列为匿名变量。为了保证比赛的公平性，将会从中抽取15万条作为训练集，5万条作为测试集A，5万条作为测试集B，同时会对name、model、brand和regionCode等信息进行脱敏。

字段表

Field	Description
SaleID	交易ID，唯一编码
name	汽车交易名称，已脱敏
regDate	汽车注册日期，例如20160101，2016年01月01日
model	车型编码，已脱敏
brand	汽车品牌，已脱敏
bodyType	车身类型：豪华轿车：0，微型车：1，厢型车：2，大巴车：3，敞篷车：4，双门汽车：5，商务车：6，搅拌车：7
fuelType	燃油类型：汽油：0，柴油：1，液化石油气：2，天然气：3，混合动力：4，其他：5，电动：6
gearbox	变速箱：手动：0，自动：1
power	发动机功率：范围 [0, 600]
kilometer	汽车已行驶公里，单位万km
notRepairedDamage	汽车有尚未修复的损坏：是：0，否：1
regionCode	地区编码，已脱敏
seller	销售方：个体：0，非个体：1
offerType	报价类型：提供：0，请求：1
creatDate	汽车上线时间，即开始售卖时间
price	二手车交易价格（预测目标）
v系列特征	匿名特征，包含v0-14在内15个匿名特征

二、评测标准

评价标准为MAE(Mean Absolute Error)。

若真实值为$y=(y_1,y_2,\cdots,y_n)$，模型的预测值为$\hat{y}=(\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_n)$，那么该模型的MAE计算公式为

$$MAE=\frac{\sum_{i=1}^{n}|y_i-\hat{y}_i|}{n}.$$

例如，真实值$y=(15,20,12)$，预测值$\hat{y}=(17,24,9)$，那么这个预测结果的MAE为

$$MAE=\frac{|15-17|+|20-24|+|12-9|}{3}=3.$$

MAE越小，说明模型预测得越准确。

三、结果提交

提交前请确保预测结果的格式与sample_submit.csv中的格式一致，以及提交文件后缀名为csv。

形式如下：

```
SaleID,price
150000,687
150001,1250
150002,2580
150003,1178
```

赛题理解

一般问题评价指标说明:

什么是评估指标:

> 评估指标即是我们对于一个模型效果的数值型量化。（有点类似与对于一个商品评价打分，而这是针对于模型效果和理想效果之间的一个打分）

一般来说分类和回归问题的评价指标有如下一些形式:

分类算法常见的评估指标如下:

- 对于二类分类器/分类算法，评价指标主要有accuracy， [Precision，Recall，F-score，Pr曲线]，ROC-AUC曲线。
- 对于多类分类器/分类算法，评价指标主要有accuracy， [宏平均和微平均，F-score]。

对于回归预测类常见的评估指标如下:

- 平均绝对误差（Mean Absolute Error，MAE），均方误差（Mean Squared Error，MSE），平均绝对百分误差（Mean Absolute Percentage Error，MAPE），均方根误差（Root Mean Squared Error）， R2（R-Square）

赛题理解

平均绝对误差 平均绝对误差（Mean Absolute Error，MAE）:平均绝对误差，其能更好地反映预测值与真实值误差的实际情况，其计算公式如下:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

均方误差 均方误差（Mean Squared Error，MSE）,均方误差,其计算公式为:

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

R2（R-Square）的公式为： 残差平方和:

$$SS_{res} = \sum(y_i - \hat{y}_i)^2$$

总平均值:

$$SS_{tot} = \sum(y_i - \bar{y}_i)^2$$

其中$\bar{y}$表示y的平均值 得到R^2表达式为:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

R^2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，取值范围是 0~1，R^2越接近1,表明回归平方和占总平方和的比例越大,回归线与各观测点越接近，用x的变化来解释y值变化的部分就越多,回归的拟合程度就越好。所以R^2也称为拟合优度（Goodness of Fit）的统计量。

y_i表示真实值，$\hat{y}_i$表示预测值，$\bar{y}_i$表示样本均值。得分越高拟合效果越好。

1.2.4. 分析赛题

1. 此题为传统的数据挖掘问题，通过数据科学以及机器学习深度学习的办法来进行建模得到结果。
2. 此题是一个典型的回归问题。
3. 主要应用xgb、lgb、catboost，以及pandas、numpy、matplotlib、seabon、sklearn、keras等等数据挖掘常用库或者框架来进行数据挖掘任务。
4. 通过EDA来挖掘数据的联系和自我熟悉数据。

1.3 代码示例

本部分为对于数据读取和指标评价的示例。

1.3.1 数据读取pandas

```
import pandas as pd
import numpy as np

path = './data/'
## 1) 载入训练集和测试集;
Train_data = pd.read_csv(path+'train.csv', sep=' ')
Test_data = pd.read_csv(path+'testA.csv', sep=' ')
print('Train data shape:',Train_data.shape)
print('TestA data shape:',Test_data.shape)
```

```
Train data shape: (150000, 31)
TestA data shape: (50000, 30)
```

1.3.2 分类指标评价计算示例

```
## accuracy
import numpy as np
from sklearn.metrics import accuracy_score
y_pred = [0, 1, 0, 1]
y_true = [0, 1, 1, 1]
print('ACC:',accuracy_score(y_true, y_pred))
```

```
ACC: 0.75
```

```
## Precision,Recall,F1-score
from sklearn import metrics
y_pred = [0, 1, 0, 0]
y_true = [0, 1, 0, 1]
print('Precision',metrics.precision_score(y_true, y_pred))
print('Recall',metrics.recall_score(y_true, y_pred))
print('F1-score:',metrics.f1_score(y_true, y_pred))
```

```
Precision 1.0
Recall 0.5
F1-score: 0.6666666666666666
```

```
## AUC
import numpy as np
from sklearn.metrics import roc_auc_score
y_true = np.array([0, 0, 1, 1])
y_scores = np.array([0.1, 0.4, 0.35, 0.8])
print('AUC socre:',roc_auc_score(y_true, y_scores))
```

```
AUC socre: 0.75
```

1.3.3 回归指标评价计算示例

```
# coding=utf-8
import numpy as np
from sklearn import metrics

# MAPE需要自己实现
def mape(y_true, y_pred):
    return np.mean(np.abs((y_pred - y_true) / y_true))

y_true = np.array([1.0, 5.0, 4.0, 3.0, 2.0, 5.0, -3.0])
y_pred = np.array([1.0, 4.5, 3.8, 3.2, 3.0, 4.8, -2.2])

# MSE
print('MSE:',metrics.mean_squared_error(y_true, y_pred))
# RMSE
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_true, y_pred)))
# MAE
print('MAE:',metrics.mean_absolute_error(y_true, y_pred))
# MAPE
print('MAPE:',mape(y_true, y_pred))
```

```
MSE: 0.2871428571428571
RMSE: 0.5358571238146014
MAE: 0.4142857142857143
MAPE: 0.1461904761904762
```

```
## R2-score
from sklearn.metrics import r2_score
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
print('R2-score:',r2_score(y_true, y_pred))
```

```
R2-score: 0.9486081370449679
```

总结

Datawhale

- 赛题理解究竟是理解什么
- 有了赛题理解后能做什么
- 赛题背景中可能潜在隐藏的条件

Part 2 Baseline

作者：ML67

Baseline讲解

Datawhale 零基础入门数据挖掘-Baseline

Baseline-v1.0 版

Tip:这是一个最初始baseline版本,抛砖引玉,为大家提供一个基本Baseline和一个竞赛流程的基本介绍，欢迎大家多多交流。

赛题：零基础入门数据挖掘 - 二手车交易价格预测

地址：https://tianchi.aliyun.com/competition/entrance/231784/introduction?spm=5176.12281957.1004.1.38b02448ausjSX

Baseline讲解

Step 1:导入函数工具箱

```
## 基础工具
import numpy as np
import pandas as pd
import warnings
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.special import jn
from IPython.display import display, clear_output
import time

warnings.filterwarnings('ignore')
%matplotlib inline

## 模型预测的
from sklearn import linear_model
from sklearn import preprocessing
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor

## 数据降维处理的
from sklearn.decomposition import PCA,FastICA,FactorAnalysis,SparsePCA

import lightgbm as lgb
import xgboost as xgb

## 参数搜索和评价的
from sklearn.model_selection import GridSearchCV,cross_val_score,StratifiedKFold,train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

Baseline讲解

Datawhale

Step 2:数据读取

```
## 通过Pandas对于数据进行读取 (pandas是一个很友好的数据读取函数库)
Train_data = pd.read_csv('datalab/231784/used_car_train_20200313.csv', sep=' ')
TestA_data = pd.read_csv('datalab/231784/used_car_testA_20200313.csv', sep=' ')

## 输出数据的大小信息
print('Train data shape:',Train_data.shape)
print('TestA data shape:',TestA_data.shape)
```

```
Train data shape: (150000, 31)
TestA data shape: (50000, 30)
```

1) 数据简要浏览

```
## 通过.head() 简要浏览读取数据的形式
Train_data.head()
```

	SaleID	name	regDate	model	brand	bodyType	fuelType	gearbox	power	kilometer	...	v_5	v_6	v_7	v_8	v_9	
0	0	736	20040402	30.0	6	1.0	0.0	0.0	60	12.5	...	0.235676	0.101988	0.129549	0.022816	0.097462	-2.88
1	1	2262	20030301	40.0	1	2.0	0.0	0.0	0	15.0	...	0.264777	0.121004	0.135731	0.026597	0.020582	-4.90
2	2	14874	20040403	115.0	15	1.0	0.0	0.0	163	12.5	...	0.251410	0.114912	0.165147	0.062173	0.027075	-4.84
3	3	71865	19960908	109.0	10	0.0	0.0	1.0	193	15.0	...	0.274293	0.110300	0.121964	0.033395	0.000000	-4.50
4	4	111080	20120103	110.0	5	1.0	0.0	0.0	68	5.0	...	0.228036	0.073205	0.091880	0.078819	0.121534	-1.89

5 rows × 31 columns

Baseline讲解

Datawhale

3) 数据统计信息浏览

```
## 通过 .describe() 可以查看数值特征列的一些统计信息
Train_data.describe()
```

	SaleID	name	regDate	model	brand	bodyType	fuelType	gearbox	power	kilometer
count	150000.000000	150000.000000	1.500000e+05	149999.000000	150000.000000	145494.000000	141320.000000	144019.000000	150000.000000	150000.000000
mean	74999.500000	68349.172873	2.003417e+07	47.129021	8.052733	1.792369	0.375842	0.224943	119.316547	12.597160
std	43301.414527	61103.875095	5.364988e+04	49.536040	7.864956	1.760640	0.548677	0.417546	177.168419	3.919576
min	0.000000	0.000000	1.991000e+07	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000
25%	37499.750000	11156.000000	1.999091e+07	10.000000	1.000000	0.000000	0.000000	0.000000	75.000000	12.500000
50%	74999.500000	51638.000000	2.003091e+07	30.000000	6.000000	1.000000	0.000000	0.000000	110.000000	15.000000
75%	112499.250000	118841.250000	2.007111e+07	66.000000	13.000000	3.000000	1.000000	0.000000	150.000000	15.000000
max	149999.000000	196812.000000	2.015121e+07	247.000000	39.000000	7.000000	6.000000	1.000000	19312.000000	15.000000

8 rows × 30 columns

```
TestA_data.describe()
```

	SaleID	name	regDate	model	brand	bodyType	fuelType	gearbox	power	kilometer	...	50
count	50000.000000	50000.000000	5.000000e+04	50000.000000	50000.000000	48587.000000	47107.000000	48090.000000	50000.000000	50000.000000	...	
mean	174999.500000	68542.223280	2.003393e+07	46.844520	8.056240	1.782185	0.373405	0.224350	119.883620	12.595580	...	
std	14433.901067	61052.808133	5.368870e+04	49.469548	7.819477	1.760736	0.546442	0.417158	185.097387	3.908979	...	
min	150000.000000	0.000000	1.991000e+07	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000	...	
25%	162499.750000	11203.500000	1.999091e+07	10.000000	1.000000	0.000000	0.000000	0.000000	75.000000	12.500000	...	
50%	174999.500000	52248.500000	2.003091e+07	29.000000	6.000000	1.000000	0.000000	0.000000	109.000000	15.000000	...	
75%	187499.250000	118856.500000	2.007110e+07	65.000000	13.000000	3.000000	1.000000	0.000000	150.000000	15.000000	...	
max	199999.000000	196805.000000	2.015121e+07	246.000000	39.000000	7.000000	6.000000	1.000000	20000.000000	15.000000	...	

Baseline讲解

Step 3:特征与标签构建

1) 提取数值类型特征列名

```
numerical_cols = Train_data.select_dtypes(exclude = 'object').columns
print(numerical_cols)
```

```
Index(['SaleID', 'name', 'regDate', 'model', 'brand', 'bodyType', 'fuelType',
       'gearbox', 'power', 'kilometer', 'regionCode', 'seller', 'offerType',
       'creatDate', 'price', 'v_0', 'v_1', 'v_2', 'v_3', 'v_4', 'v_5', 'v_6',
       'v_7', 'v_8', 'v_9', 'v_10', 'v_11', 'v_12', 'v_13', 'v_14'],
      dtype='object')
```

```
categorical_cols = Train_data.select_dtypes(include = 'object').columns
print(categorical_cols)
```

```
Index(['notRepairedDamage'], dtype='object')
```

2) 构建训练和测试样本

```
## 选择特征列
feature_cols = [col for col in numerical_cols if col not in ['SaleID','name','regDate','creatDate','price','model','brand','regionCode','
feature_cols = [col for col in feature_cols if 'Type' not in col]

## 提前特征列，标签列构造训练样本和测试样本
X_data = Train_data[feature_cols]
Y_data = Train_data['price']

X_test  = TestA_data[feature_cols]

print('X train shape:',X_data.shape)
print('X test shape:',X_test.shape)
```

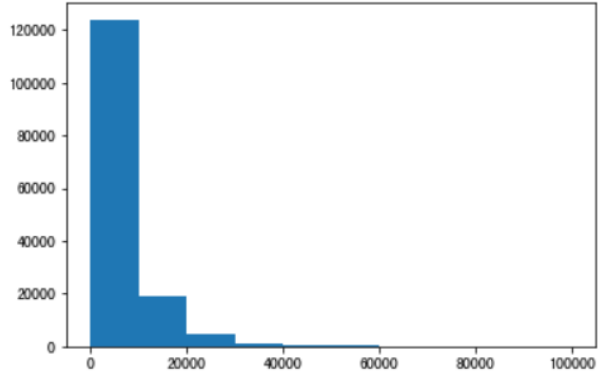
Baseline讲解

3) 统计标签的基本分布信息

```
print('Sta of label:')
Sta_inf(Y_data)
```

```
Sta of label:
_min 11
_max: 99999
_mean 5923.32733333
_ptp 99988
_std 7501.97346988
_var 56279605.9427
```

```
## 绘制标签的统计图，查看标签分布
plt.hist(Y_data)
plt.show()
plt.close()
```



4) 缺省值用-1填补

```
X_data = X_data.fillna(-1)
X_test = X_test.fillna(-1)
```

Baseline讲解

Step 4:模型训练与预测

1) 利用xgb进行五折交叉验证查看模型的参数效果

```
## xgb-Model
xgr = xgb.XGBRegressor(n_estimators=120, learning_rate=0.1, gamma=0, subsample=0.8,\
        colsample_bytree=0.9, max_depth=7) #,objective ='reg:squarederror'

scores_train = []
scores = []

## 5折交叉验证方式
sk=StratifiedKFold(n_splits=5,shuffle=True,random_state=0)
for train_ind,val_ind in sk.split(X_data,Y_data):

    train_x=X_data.iloc[train_ind].values
    train_y=Y_data.iloc[train_ind]
    val_x=X_data.iloc[val_ind].values
    val_y=Y_data.iloc[val_ind]

    xgr.fit(train_x,train_y)
    pred_train_xgb=xgr.predict(train_x)
    pred_xgb=xgr.predict(val_x)

    score_train = mean_absolute_error(train_y,pred_train_xgb)
    scores_train.append(score_train)
    score = mean_absolute_error(val_y,pred_xgb)
    scores.append(score)

print('Train mae:',np.mean(score_train))
print('Val mae',np.mean(scores))
```

```
Train mae: 628.086664863
Val mae 715.990013454
```

Baseline讲解

Datawhale

2） 定义xgb和lgb模型函数 ¶

```
def build_model_xgb(x_train,y_train):
    model = xgb.XGBRegressor(n_estimators=150, learning_rate=0.1, gamma=0, subsample=0.8,\
        colsample_bytree=0.9, max_depth=7) #, objective ='reg:squarederror'
    model.fit(x_train, y_train)
    return model

def build_model_lgb(x_train,y_train):
    estimator = lgb.LGBMRegressor(num_leaves=127,n_estimators = 150)
    param_grid = {
        'learning_rate': [0.01, 0.05, 0.1, 0.2],
    }
    gbm = GridSearchCV(estimator, param_grid)
    gbm.fit(x_train, y_train)
    return gbm
```

3）切分数据集（Train,Val）进行模型训练，评价和预测

```
## Split data with val
x_train,x_val,y_train,y_val = train_test_split(X_data,Y_data,test_size=0.3)
```

Baseline讲解


```
print('Train lgb...')
model_lgb = build_model_lgb(x_train,y_train)
val_lgb = model_lgb.predict(x_val)
MAE_lgb = mean_absolute_error(y_val,val_lgb)
print('MAE of val with lgb:',MAE_lgb)

print('Predict lgb...')
model_lgb_pre = build_model_lgb(X_data,Y_data)
subA_lgb = model_lgb_pre.predict(X_test)
print('Sta of Predict lgb:')
Sta_inf(subA_lgb)
```

```
Train lgb...
MAE of val with lgb: 689.084070621
Predict lgb...
Sta of Predict lgb:
_min -519.150259864
_max: 88575.1087721
_mean 5922.98242599
_ptp 89094.259032
_std 7377.29714126
_var 54424513.1104
```

```
print('Train xgb...')
model_xgb = build_model_xgb(x_train,y_train)
val_xgb = model_xgb.predict(x_val)
MAE_xgb = mean_absolute_error(y_val,val_xgb)
print('MAE of val with xgb:',MAE_xgb)

print('Predict xgb...')
model_xgb_pre = build_model_xgb(X_data,Y_data)
subA_xgb = model_xgb_pre.predict(X_test)
print('Sta of Predict xgb:')
Sta_inf(subA_xgb)
```

```
Train xgb...
MAE of val with xgb: 715.37757816
Predict xgb...
Sta of Predict xgb:
_min -165.479
_max: 90051.8
_mean 5922.9
_ptp 90217.3
_std 7361.13
_var 5.41862e+07
```

Baseline讲解

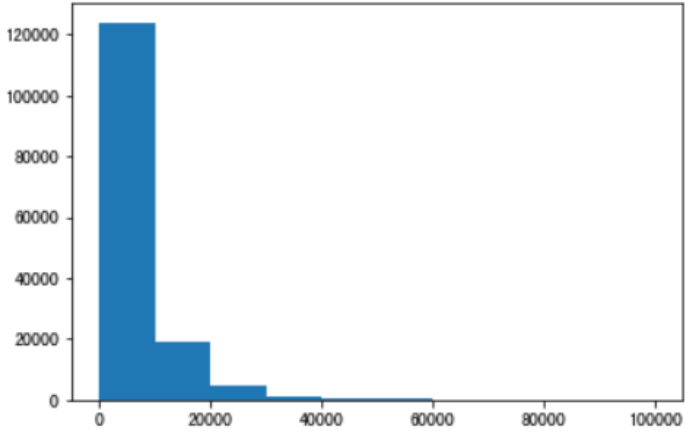
4）进行两模型的结果加权融合

```
## 这里我们采取了简单的加权融合的方式
val_Weighted = (1-MAE_lgb/(MAE_xgb+MAE_lgb))*val_lgb+(1-MAE_xgb/(MAE_xgb+MAE_lgb))*val_xgb
val_Weighted[val_Weighted<0]=10 # 由于我们发现预测的最小值有负数，而真实情况下，price为负是不存在的，由此我们进行对应的后修正
print('MAE of val with Weighted ensemble:',mean_absolute_error(y_val,val_Weighted))
```

```
MAE of val with Weighted ensemble: 687.275745703
```

```
sub_Weighted = (1-MAE_lgb/(MAE_xgb+MAE_lgb))*subA_lgb+(1-MAE_xgb/(MAE_xgb+MAE_lgb))*subA_xgb

## 查看预测值的统计进行
plt.hist(Y_data)
plt.show()
plt.close()
```



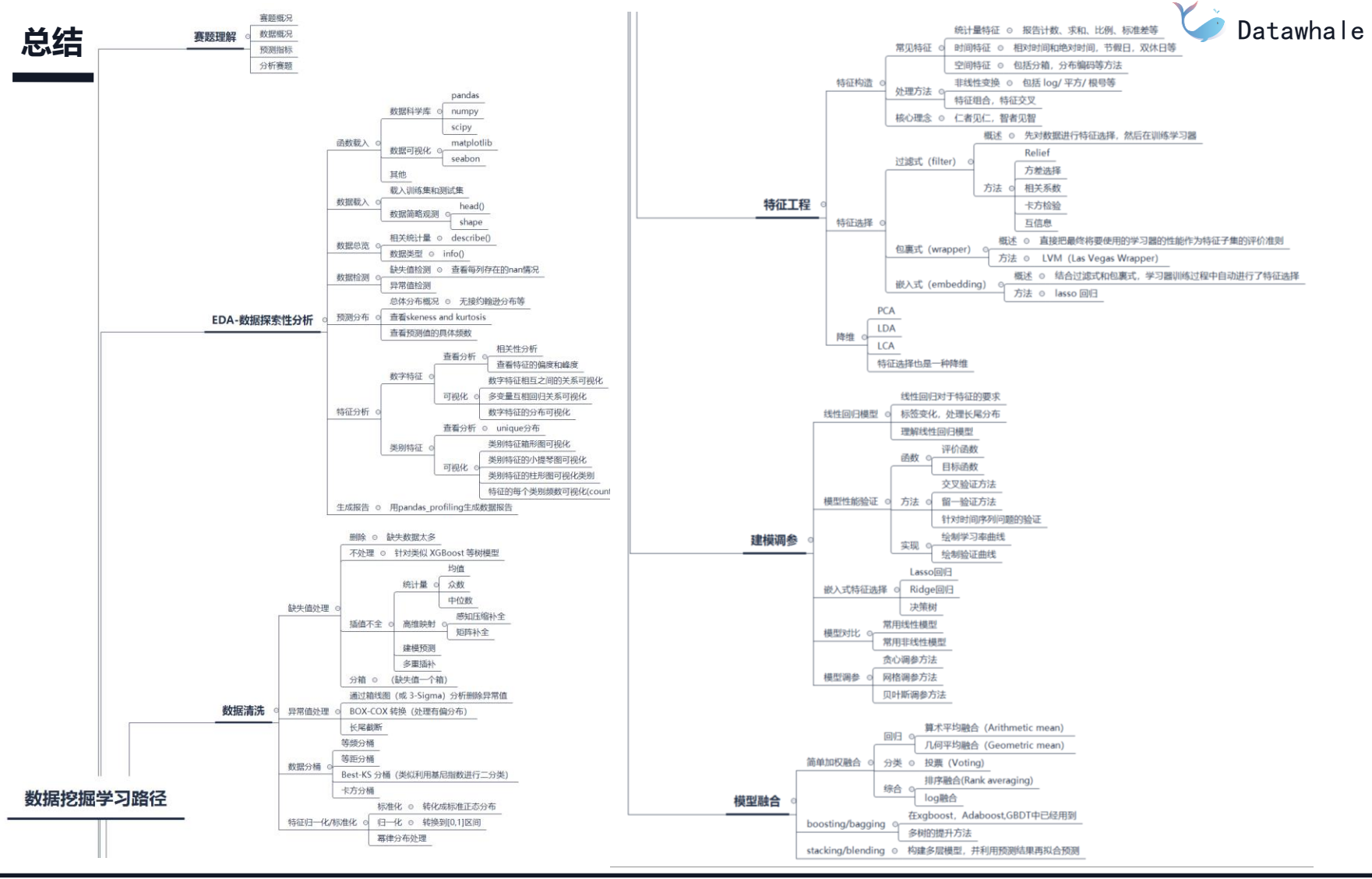
5）输出结果

```
sub = pd.DataFrame()
sub['SaleID'] = X_test.SaleID
sub['price'] = sub_Weighted
sub.to_csv('./sub_Weighted.csv',index=False)
```

```
sub.head()
```

	SaleID	price
0	0	39533.727414
1	1	386.081960
2	2	7791.974571
3	3	11835.211966
4	4	585.420407

Part 3 总结


总结
数据挖掘学习路径
赛题理解
赛题概况
数据概况
预测指标
分析赛题
EDA-数据探索性分析
函数载入
数据科学库
pandas
numpy
scipy
数据可视化
matplotlib
seabon
其他
数据载入
载入训练集和测试集
数据简略观测
head()
shape
数据总览
相关统计量
describe()
数据类型
info()
数据检测
缺失值检测
查看每列存在的nan情况
异常值检测
预测分布
总体分布概况
无接约翰逊分布等
查看skeness and kurtosis
查看预测值的具体频数
特征分析
数字特征
查看分析
相关性分析
查看特征的偏度和峰度
可视化
数字特征相互之间的关系可视化
多变量互相回归关系可视化
数字特征的分布可视化
类别特征
查看分析
unique分布
可视化
类别特征箱形图可视化
类别特征的小提琴图可视化
类别特征的柱形图可视化类别
特征的每个类别频数可视化(count
生成报告
用pandas_profiling生成数据报告
数据清洗
缺失值处理
删除
缺失数据太多
不处理
针对类似 XGBoost 等树模型
插值不全
统计量
均值
众数
中位数
高维映射
感知压缩补全
矩阵补全
建模预测
多重插补
分箱
(缺失值一个箱)
异常值处理
通过箱线图（或 3-Sigma）分析删除异常值
BOX-COX 转换（处理有偏分布）
长尾截断
数据分桶
等频分桶
等距分桶
Best-KS 分桶（类似利用基尼指数进行二分类）
卡方分桶
特征归一化/标准化
标准化
转化成标准正态分布
归一化
转换到[0,1]区间
幂律分布处理
特征工程
特征构造
常见特征
统计量特征
报告计数、求和、比例、标准差等
时间特征
相对时间和绝对时间，节假日，双休日等
空间特征
包括分箱，分布编码等方法
处理方法
非线性变换
包括 log/ 平方/ 根号等
特征组合，特征交叉
核心理念
仁者见仁，智者见智
特征选择
过滤式（filter）
概述
先对数据进行特征选择，然后在训练学习器
方法
Relief
方差选择
相关系数
卡方检验
互信息
包裹式（wrapper）
概述
直接把最终将要使用的学习器的性能作为特征子集的评价准则
方法
LVM（Las Vegas Wrapper）
嵌入式（embedding）
概述
结合过滤式和包裹式，学习器训练过程中自动进行了特征选择
方法
lasso 回归
降维
PCA
LDA
LCA
特征选择也是一种降维
建模调参
线性回归模型
线性回归对于特征的要求
标签变化，处理长尾分布
理解线性回归模型
模型性能验证
函数
评价函数
目标函数
方法
交叉验证方法
留一验证方法
针对时间序列问题的验证
实现
绘制学习率曲线
绘制验证曲线
嵌入式特征选择
Lasso回归
Ridge回归
决策树
模型对比
常用线性模型
常用非线性模型
模型调参
贪心调参方法
网格调参方法
贝叶斯调参方法
模型融合
简单加权融合
回归
算术平均融合（Arithmetic mean）
几何平均融合（Geometric mean）
分类
投票（Voting）
综合
排序融合(Rank averaging)
log融合
boosting/bagging
在xgboost，Adaboost,GBDT中已经用到
多树的提升方法
stacking/blending
构建多层模型，并利用预测结果再拟合预测
Datawhale

作者

--- By: ML67

Email: maolinw67@163.com
PS: 华中科技大学研究生，长期混迹Tianchi等，希望和大家多多交流。
github: https://github.com/mlw67 （近期会做一些书籍推导和代码的整理）

--- By: AI蜗牛车

PS：东南大学研究生，研究方向主要是时空序列预测和时间序列数据挖掘
公众号： AI蜗牛车
知乎： https://www.zhihu.com/people/seu-aigua-niu-che
github: https://github.com/chehongshu

--- By: 阿泽

PS：复旦大学计算机研究生
知乎：阿泽 https://www.zhihu.com/people/is-aze（主要面向初学者的知识整理）

--- By: 小雨姑娘

PS：数据挖掘爱好者，多次获得比赛TOP名次。
知乎：小雨姑娘的机器学习笔记：https://zhuanlan.zhihu.com/mlbasic

Datawhale

一个专注于AI领域的开源组织