数据分析语言选择

Thinking：如何选择数据分析语言?

- Python是首选的数据分析语言
- 在数据分析/数据科学领域中占有率70%
- 有强大的生态（社区+工具）

科学计算：Sklearn, Numpy, Pandas

人工智能：Tensorflow, PyTorch

网络爬虫：Scrapy, Request, BeautifulSoap

每篇顶级会议发表的最新技术，网络模型很多使用Python

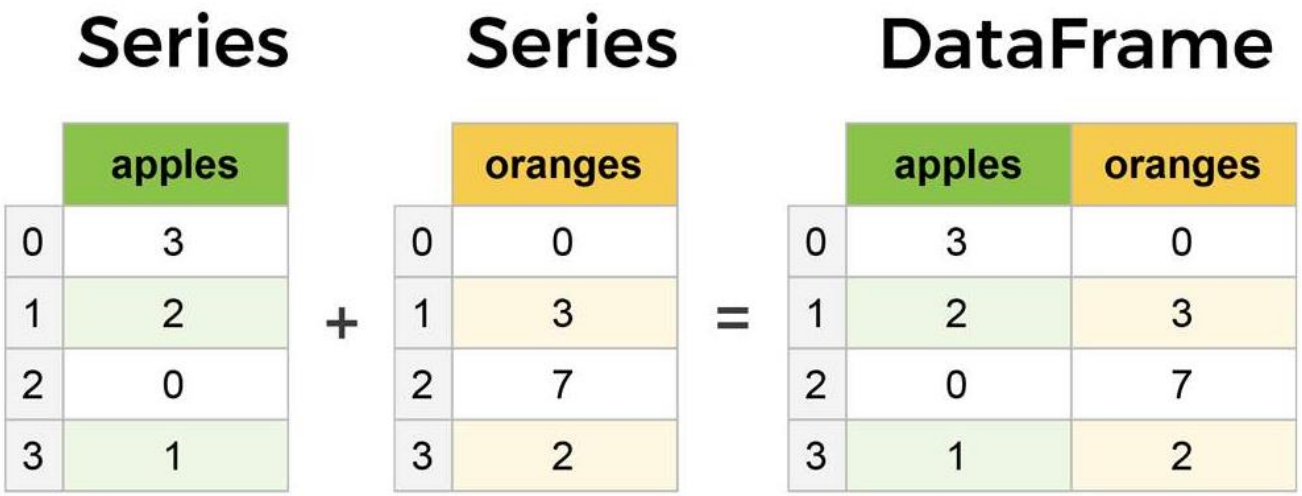
选语言就是选生态（Julia VS Python）

Worldwide, Jun 2020 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	31.6 %	+4.3 %
2		Java	17.67 %	-2.4 %
3		Javascript	8.02 %	-0.2 %
4		C#	6.87 %	-0.4 %
5		PHP	6.02 %	-0.9 %
6		C/C++	5.69 %	-0.2 %
7		R	3.86 %	-0.1 %
8		Objective-C	2.5 %	-0.3 %
9		Swift	2.24 %	-0.1 %
10	↑	TypeScript	1.86 %	+0.2 %

Pandas使用

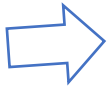
- 在数据分析中，Pandas的使用频率很高
- Pandas可以说是基于 NumPy 构建的含有更高级数据结构和分析能力的工具包
- Series和 DataFrame是两个核心数据结构，分别代表着一维的序列和二维的表结构
- 基于这两种数据结构，Pandas可以对数据进行导入、清洗、处理、统计和输出



Pandas使用

- Series是个定长的字典序列
- 在存储的时候，相当于两个ndarray，这也是和字典结构最大的不同。因为字典结构，元素个数是不固定的
- Series有两个基本属性：index 和 values

```
from pandas import Series, DataFrame
x1 = Series([1,2,3,4])
x2 = Series(data=[1,2,3,4], index=['a', 'b', 'c', 'd'])
# 使用字典来进行创建
d = {'a':1, 'b':2, 'c':3, 'd':4}
x3 = Series(d)
print(x1)
print(x2)
print(x3)
```

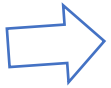


```
0    1
1    2
2    3
3    4
dtype: int64
a    1
b    2
c    3
d    4
dtype: int64
a    1
b    2
c    3
d    4
dtype: int64
```

Pandas使用

- DataFrame使用
- 类似数据库表，包括了行索引和列索引，可以将DataFrame 看成是由相同索引的Series组成的字典类型

```
from pandas import Series, DataFrame
data = {'Chinese': [66, 95, 93, 90,80],
'Math': [30, 98, 96, 77, 90], 'English': [65,
85, 92, 88, 90]}
df1 = DataFrame(data)
df2 = DataFrame(data, index=['ZhangFei',
'GuanYu', 'LiuBei', 'DianWei', 'XuChu'],
columns=['Chinese', 'Math', 'English'])
print(df1)
print(df2)
```



```
   Chinese  Math  English
0       66    30       65
1       95    98       85
2       93    96       92
3       90    77       88
4       80    90       90
          Chinese  Math  English
ZhangFei       66    30       65
GuanYu         95    98       85
LiuBei         93    96       92
DianWei        90    77       88
XuChu          80    90       90
```

Pandas使用

- 删除 DataFrame 中的不必要的列或行

```
df2 = df2.drop(columns=['Chinese']) #删除列

df2 = df2.drop(index=['ZhangFei']) #删除行
```

- 重命名列名columns，让列表名更容易识别

```
df2.rename(columns={'Chinese': '语文', 'English': 'Yingyu'}, inplace = True)
```

- 去掉重复的值

```
df = df.drop_duplicates()
```

- 更改数据格式

```
df2['Chinese'].astype('str')
df2['Chinese'].astype(np.int64)
```

- 去掉数据间的空格

```
#删除左右两边空格
df2['Chinese']=df2['Chinese'].map(str.strip)
```

	语文	数学	英语
ZhangFei	66	30	65
GuanYu	95	98	85
LiuBei	93	96	92
DianWei	90	77	88
XuChu	80	90	90

Pandas使用

- 大小写转换

#全部大写

df2.columns = df2.columns.str.upper()

#全部小写

df2.columns = df2.columns.str.lower()

#首字母大写

df2.columns = df2.columns.str.title()

- 查找空值

姓名	语文	英语	数学
张飞	66	65	
关羽	95	85	98
赵云	95	92	96
黄忠	90	88	77
典韦	80	90	90

使用df.isnull()，结果如下

	姓名	语文	英语	数学
0	False	False	False	True
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False

Pandas使用

- 使用apply函数进行数据清洗

apply函数是Pandas中自由度非常高的函数，使用频率高

比如对name列的数值都进行大写转化

```
df['name'] = df['name'].apply(str.upper)
```

也可以定义个函数，在apply中进行使用

```
def double_df(x):
        return 2*x
df1[u'语文'] = df1[u'语文'].apply(double_df)
```

Thinking: apply和map的区别是什么?

- apply 用在dataframe上，用于对row或者column进行计算
- applymap 用于dataframe上，是元素级别的操作
- map ， 是python自带的，用于series上，是元素级别的操作

Pandas使用

- Pandas中的统计函数

count() 统计个数，空值NaN不计算

describe() 一次性输出多个统计指标，包括：count, mean, std, min, max等

min()最小值

max()最大值

sum()总和

mean()平均值

median()中位数

var()方差

std()标准差

argmin()　　统计最小值的索引位置

argmax()统计最大值的索引位置

idxmin()　　统计最小值的索引值

idxmax()　　统计最大值的索引值

Pandas使用

- Pandas中的统计函数

print(df2.describe())

	语文	数学	英语
count	5.000000	5.000000	5.000000
mean	84.800000	78.200000	84.000000
std	11.987493	28.163807	10.931606
min	66.000000	30.000000	65.000000
25%	80.000000	77.000000	85.000000
50%	90.000000	90.000000	88.000000
75%	93.000000	96.000000	90.000000
max	95.000000	98.000000	92.000000

- 数据表合并

df1 = DataFrame({'name':['ZhangFei', 'GuanYu', 'a', 'b', 'c'], 'data1':range(1,6)})

df2 = DataFrame({'name':['ZhangFei', 'GuanYu', 'A', 'B', 'C'], 'data2':range(1,6)})

1）基于name这列进行连接

df3 = pd.merge(df1, df2, on='name')

	name	data1
0	ZhangFei	1
1	GuanYu	2
2	a	3
3	b	4
4	c	5

	name	data2
0	ZhangFei	1
1	GuanYu	2
2	A	3
3	B	4
4	C	5

	name	data1	data2
0	ZhangFei	1	1
1	GuanYu	2	2

Pandas使用

- inner内连接

df3 = pd.merge(df1, df2, how='inner')

	name	data1
0	ZhangFei	1
1	GuanYu	2
2	a	3
3	b	4
4	c	5

	name	data2
0	ZhangFei	1
1	GuanYu	2
2	A	3
3	B	4
4	C	5

	name	data1	data2
0	ZhangFei	1	1
1	GuanYu	2	2

- left左连接

df3 = pd.merge(df1, df2, how='left')

	name	data1
0	ZhangFei	1
1	GuanYu	2
2	a	3
3	b	4
4	c	5

	name	data2
0	ZhangFei	1
1	GuanYu	2
2	A	3
3	B	4
4	C	5

	name	data1	data2
0	ZhangFei	1	1.0
1	GuanYu	2	2.0
2	a	3	NaN
3	b	4	NaN
4	c	5	NaN

- right右连接

df3 = pd.merge(df1, df2, how='right')

	name	data1
0	ZhangFei	1
1	GuanYu	2
2	a	3
3	b	4
4	c	5

	name	data2
0	ZhangFei	1
1	GuanYu	2
2	A	3
3	B	4
4	C	5

	name	data1	data2
0	ZhangFei	1.0	1
1	GuanYu	2.0	2
2	A	NaN	3
3	B	NaN	4
4	C	NaN	5

- outer外连接

df3 = pd.merge(df1, df2, how='outer')

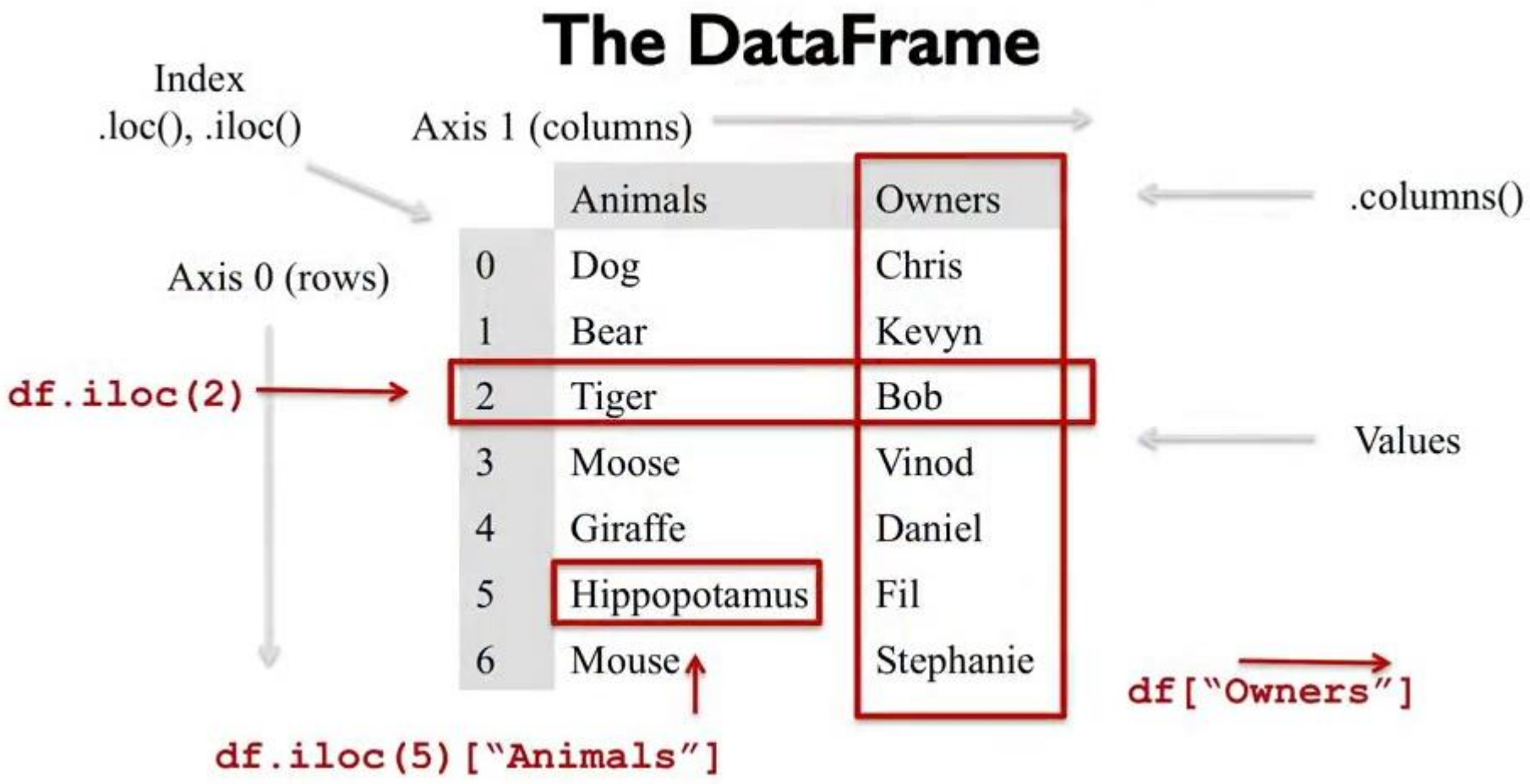
	name	data1
0	ZhangFei	1
1	GuanYu	2
2	a	3
3	b	4
4	c	5

	name	data2
0	ZhangFei	1
1	GuanYu	2
2	A	3
3	B	4
4	C	5

	name	data1	data2
0	ZhangFei	1.0	1.0
1	GuanYu	2.0	2.0
2	a	3.0	NaN
3	b	4.0	NaN
4	c	5.0	NaN
5	A	NaN	3.0
6	B	NaN	4.0
7	C	NaN	5.0

Pandas使用

- loc函数：通过行索引 "Index" 中的具体值来取行数据（如取"Index"为"A"的行）
- iloc函数：通过行号来取行数据（如取第二行的数据）



Pandas使用

```
from pandas import Series, DataFrame

data = {'Chinese': [66, 95, 93, 90,80],
'Math': [30, 98, 96, 77, 90], 'English': [65,
85, 92, 88, 90]}

df = DataFrame(data, index=['ZhangFei',
'GuanYu', 'LiuBei', 'DianWei', 'XuChu'],
columns=['Chinese', 'Math', 'English'])

# 提取Index为ZhangFei的行

print(df.loc['ZhangFei'])

# 提取第0行

print(df.iloc[0])
```

```
Chinese    66
Math       30
English    65
Name: ZhangFei, dtype: int64
Chinese    66
Math       30
English    65
Name: ZhangFei, dtype: int64
```

Pandas使用

```
# 提取列为English的所有行

print(df.loc[:,['English']])

# 提取第2列的所有行

print(df.iloc[:,2])
```

```
          English
ZhangFei       65
GuanYu         85
LiuBei         92
DianWei        88
XuChu          90
ZhangFei    65
GuanYu      85
LiuBei      92
DianWei     88
XuChu       90
Name: English, dtype: int64
```

```
# 查看ZhangFei, GuanYu的Chinese Math成绩

print(df.loc[['ZhangFei','GuanYu'],
['Chinese','Math']])

print(df.iloc[[0,1],[0,1]])
```

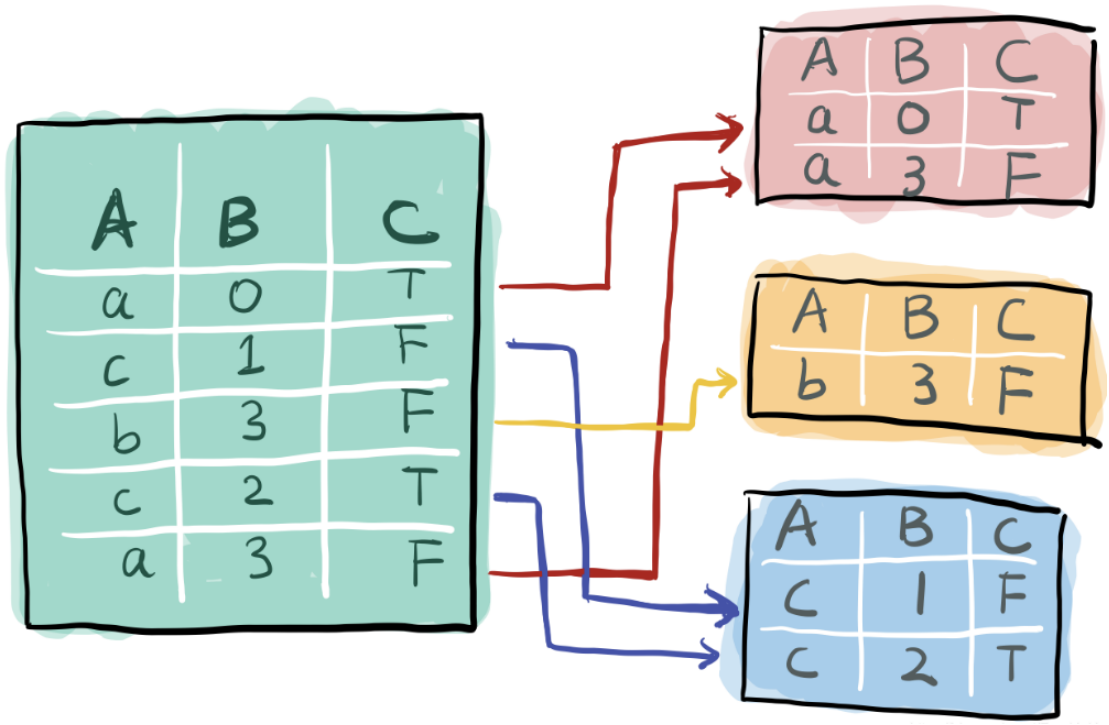
```
          Chinese  Math
ZhangFei       66    30
GuanYu         95    98
          Chinese  Math
ZhangFei       66    30
GuanYu         95    98
```

Pandas使用

- Pandas中的groupby使用

作用是进行数据的分组以及分组后地组内运算

```
import numpy as np

import pandas as pd

# 因为文件中有中文，所以采用gbk编码读取

data = pd.read_csv('heros2.csv',
encoding='gbk')

result = data.groupby('role').agg([np.sum,
np.mean])

print(result)
```



	Chinese		Maths		English	
	sum	mean	sum	mean	sum	mean
role						
坦克	146	73.000000	120	60.000000	155	77.500000
战士	278	92.666667	271	90.333333	265	88.333333

Pandas使用

- 排序函数sort_values()

sort_values()函数原理类似于SQL中的order by

df.sort_values('A11', ascending=False)

对df的A11列从大到小进行排序

- df.reset_index(inplace=True)

reset_index可以还原索引，重新变为默认的整型索引

inplace = True：不创建新的对象，直接对原始对象进行修改

Pandas使用

- 读取csv文件

pd.read_csv('car_complain.csv')

- 写csv文件

pd.to_csv('car_complain.csv', index=False)

index=False，表明不保存index

- 使用merge通过索引合并两个Dataframe

df2 = df.merge(df2, left_index=True, right_index=True,
how='left')

Project：使用Python对汽车质量数据进行统计

To DO: 对汽车质量数据进行统计

- 数据集：car_complain.csv

600条汽车质量投诉

Step1，数据加载

Step2，数据预处理

拆分problem类型 => 多个字段

Step3，数据统计

按照brand统计 投诉总数，不同problem类型的总数

按照投诉总数进行排序

按照指定的problem类型进行排序

id	brand	car_model	type	desc	problem	datetime	status
491238	上汽斯柯达	柯迪亚克	2018款 TSI330 两驱 豪华优享版	斯柯达柯迪亚克排气门密	A251,	2020/7/3	信息审核
491237	比亚迪	比亚迪F3	2014款 1.5L 手动 尊贵型	比亚迪F3中控大屏发白影	H134,	2020/7/3	信息审核
491236	广汽丰田	丰田C-HR	2018款 2.0L 领先版 国VI	广汽丰田C-HR刹车盘生锈	E42,	2020/7/3	信息审核
491235	吉利汽车	帝豪GL	2018款 1.8L 自动 精英智联型	吉利帝豪GL行驶中变速箱	B19,I297,	2020/7/3	信息审核
491233	领克汽车	领克01	2019款 2.0T 两驱型 Pro版 国VI	领克01发动机漏油维修后	A12,	2020/7/3	信息审核
491231	一汽-大众奥迪	奥迪A4L	2019款 40 TFSI 时尚型 国V	一汽大众奥迪A4L空调异	A9,H101,	2020/7/3	信息审核
491229	吉利汽车	博越	2020款 1.8TD DCT 智领PRO	吉利博越变速箱挡位灯与	M279,	2020/7/3	信息审核
491227	大众（进口）	蔚揽新能源	2019款 GTE	进口大众蔚揽新能源空调	H101,	2020/7/3	信息审核
491226	一汽-大众奥迪	奥迪A6L	2017款 TFSI 双离合 技术型	一汽大众奥迪A6L节温器	A37,	2020/7/3	信息审核
491225	北京现代	现代ix25	2017款 1.6L 自动 智能型	北京现代ix25车漆生锈起	H79,H81,H169,I295	2020/7/3	信息审核
491223	一汽-大众奥迪	奥迪A3	2017款 Sportback 2.0T 双离合	一汽大众奥迪A3保修起始	L310,	2020/7/3	信息审核
491221	上汽通用五菱	宝骏560	2016款 1.8L 手动 豪华型	宝骏560车内异味严重要	H185,	2020/7/3	信息审核
491220	东风标致	标致408	2016款 1.6T 自动 豪华版	东风标致408蓄电池与发	A34,A44,H146,	2020/7/3	厂家受理
491218	吉利汽车	帝豪GL	2017款 1.8L 自动 精英型	吉利帝豪GL右后刹车灯进	H88,	2020/7/3	信息审核
491217	东风风行	景逸X3	2016款 1.5L 豪华型	东风风行景逸X3行驶中右	G210,H168,	2020/7/3	厂家受理
491216	重汽王牌	7系	YCD4D4S-140	重汽王牌7系发动机缸体	A37,	2020/7/3	信息审核
491215	一汽-大众	探岳	2019款 330TSI 两驱 豪华型Plus	一汽大众探岳天窗漏水多	H90,J298,	2020/7/3	信息审核
491214	北京现代	名图	2017款 1.8L 自动 智能型 GLS 国	北京现代名图转向异响希	D,I,O,	2020/7/3	信息审核
491212	上汽大通	上汽MAXUS G20	2019款 首发款 2.0T 自动 豪华行	上汽大通G20天窗漏水严	H90,J298,	2020/7/3	信息审核
491211	东风标致	标致408	2015款 1.2T 自动 豪华版	东风标致408固特异轮胎	F100,	2020/7/3	厂家受理
491209	吉利汽车	星越	2019款 350T 耀星者	吉利星越车机系统升级后	H21,O348,	2020/7/3	厂家受理
491208	吉利汽车	博越	2016款 1.8TD 自动 两驱 智尚型	吉利博越大灯远光反光板	H82,I295,	2020/7/3	厂家受理
491206	吉利汽车	星越	2019款 改款 350T 耀星者	吉利星越车机系统升级后	H155,O348,	2020/7/3	厂家受理
491203	长安马自达	马自达3 昂克赛拉	2020款 2.0L 自动 质雅版	长安马自达昂克赛拉方向	D59,D337,	2020/7/3	信息审核
491200	华晨宝马	宝马X3	2019款 xDrive28i 豪华套装	华晨宝马X3经销商私自试	J306,O305,	2020/7/3	信息审核
491198	一汽丰田	RAV4荣放	2020款 2.5L CVT 四驱 精英版 双	一汽丰田RAV4荣放车顶纵	H209,I294,	2020/7/3	信息审核
491193	比亚迪	比亚迪S6	2014款 2.4L 自动 尊贵型 7座	比亚迪S6变速箱严重抖动	B19,B95,	2020/7/3	厂家受理
491192	上汽集团	名爵6	2019款 20T 自动 Trophy 竞技版	上汽名爵6变速箱异响倒	B33,	2020/7/3	厂家受理
491191	吉利汽车	博越	2016款 1.8TD 自动 四驱 智尊型	吉利博越变速箱空档没有	B246,I297,	2020/7/3	厂家受理