课　　程：2017 软件工程综合实训

项　　目：　数据挖掘比赛

院　　系：　数据科学与计算机学院

专　　业：　嵌入式系统

学生姓名：　14331107

学　　号：　黄裕全

授课教师：　郑子彬，曾海标

2017 年 7 月 6 日

目录

一、比赛描述

这次比赛目的是为了预测俄罗斯的房价，通过给出房子的相关信息和俄罗斯的宏观经济情况来判断一个房子的价格，是一个多特征的线性回归问题。下面是官方的题目描述。

Housing costs demand a significant investment from both consumers and developers. And when it comes to planning a budget—whether personal or corporate—the last thing anyone needs is uncertainty about one of their biggets expenses. Sberbank, Russia's oldest and largest bank, helps their customers by making predictions about realty prices so renters, developers, and lenders are more confident when they sign a lease or purchase a building.

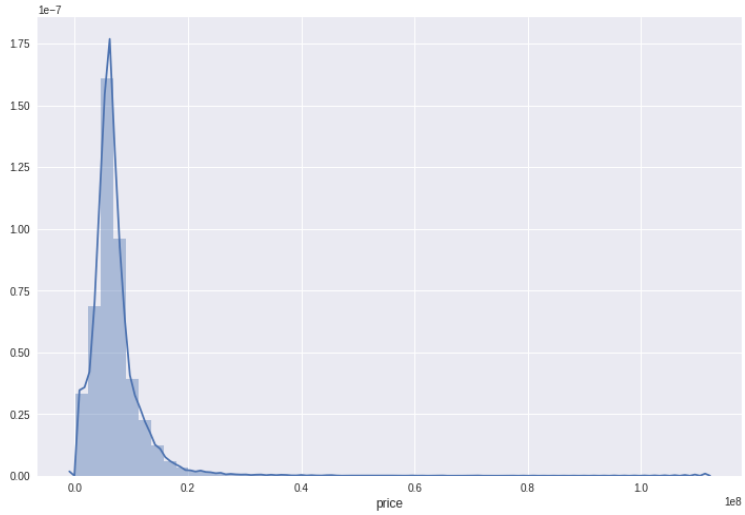
Although the housing market is relatively stable in Russia, the country's volatile economy makes forecasting prices as a function of apartment characteristics a unique challenge. Complex interactions between housing features such as number of bedrooms and location are enough to make pricing predictions complicated. Adding an unstable economy to the mix means Sberbank and their customers need more than simple regression models in their arsenal.

In this competition, Sberbank is challenging Kagglers to develop algorithms which use a broad spectrum of features to predict realty prices. Competitors will rely on a rich dataset that includes housing data and macroeconomic patterns. An accurate forecasting model will allow Sberbank to provide more certainty to their customers in an uncertain economy.

二、比赛分析

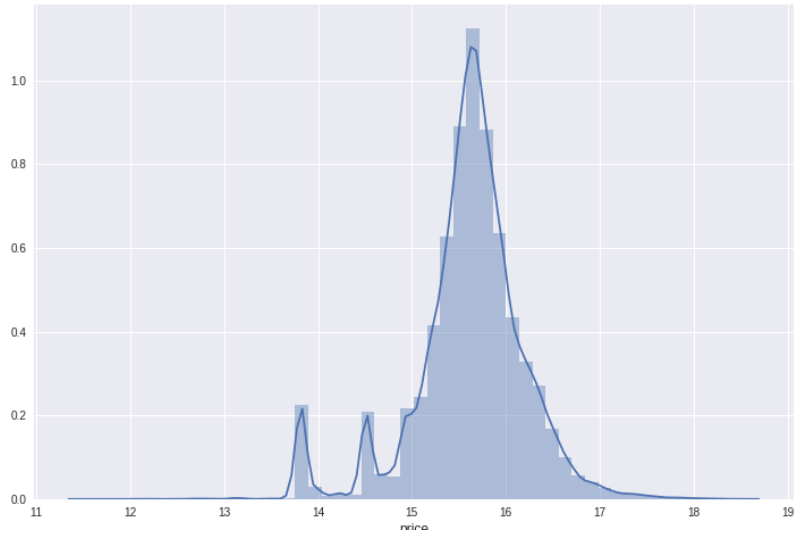
1.对结果[doc_price]的分析

从比赛网站的讨论上可以看到有许多人已经对相关特征做了一些可视化的处理。（引用【1】）



这是根据训练集数据画出来的价格分布曲线，可以看到价格的分布主要是集中在左边，同时我们了解到这次比赛是用 rmse（标准误差）来评判的。而通

过查询资料可以得知，通过 rmse 计算评判的比赛可以先对数据进行对数处理，之后再复原，这样的效果会比较好。（引用【2】）



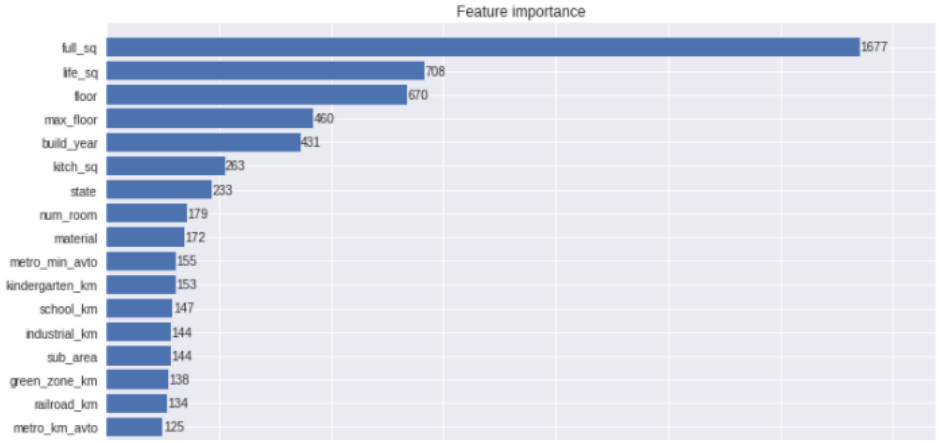
主要代码实现：

```
ylog_train_all = np.log1p(df_train['price_doc'].values)
```

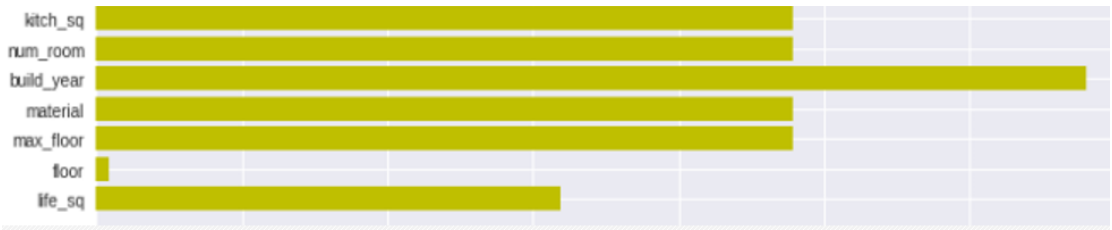
```
y_pred = np.exp(ylog_pred) - 1
```

2.对现有特征分析

由于比赛模型大多使用的是 xgboost，因此主要关心的是其分裂节点的重要程度。（引用【3】）



从另外一张图我们可以发现很多重要的数据存在着大量的缺失，因此首先需要对缺失的重要数据进行处理（引用【4】）



具体缺失数量如下：

Life_sq(NAN):

```
False    30574
True      7559
Name: life_sq, dtype: int64
```

Kitch_sq(NAN):

```
False    28561
True      9572
Name: kitch_sq, dtype: int64
```

对重要性排名第二的 life_sq 进行处理，利用总面积减去厨房面积的方法：

kitch_sq

```
df_all['kitch_sq'] = df_all['kitch_sq'].fillna(df_all['kitch_sq'].mean())
```

life_sq

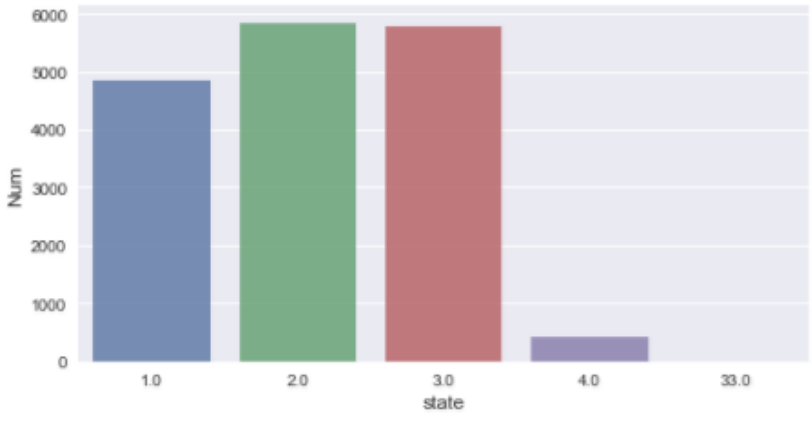
```
df_all['life_sq'] = df_all['life_sq'].fillna(df_all['full_sq']-df_all['kitch_sq'])
bad_index = df_all[df_all.life_sq < 0].index
df_all.loc[bad_index, "life_sq"] = df_all['life_sq'].fillna(df_all['life_sq'].mean())
```

对 material 进行众数填充：

```
train['material'] = train['material'].fillna(train['material'].mode().iloc[0])
```

其他缺失特征不用管，因为 xgboost 有很好地处理 NAN 的能力，自己盲目地填充会导致结果变得更差。

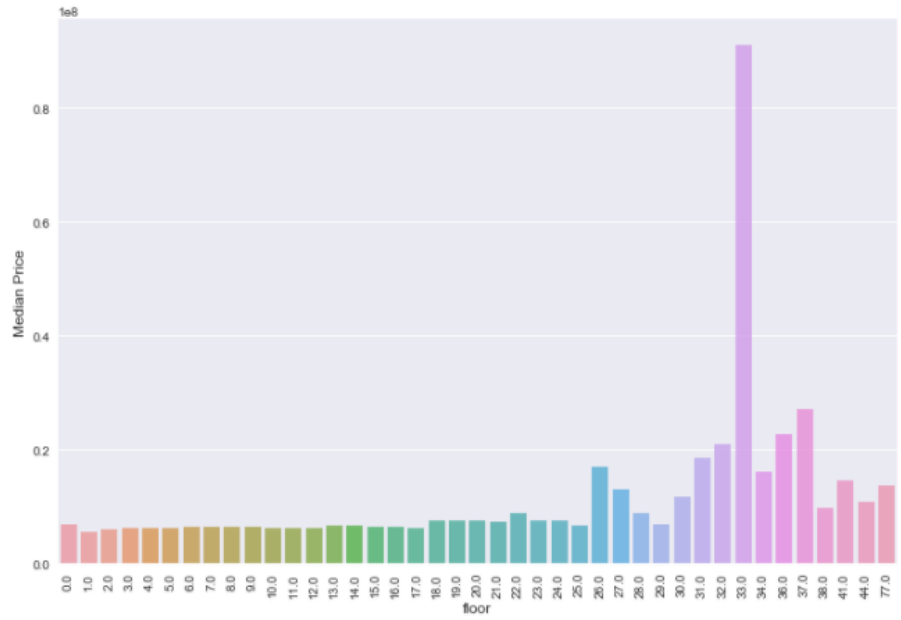
接下来是对错误数据的处理



State:

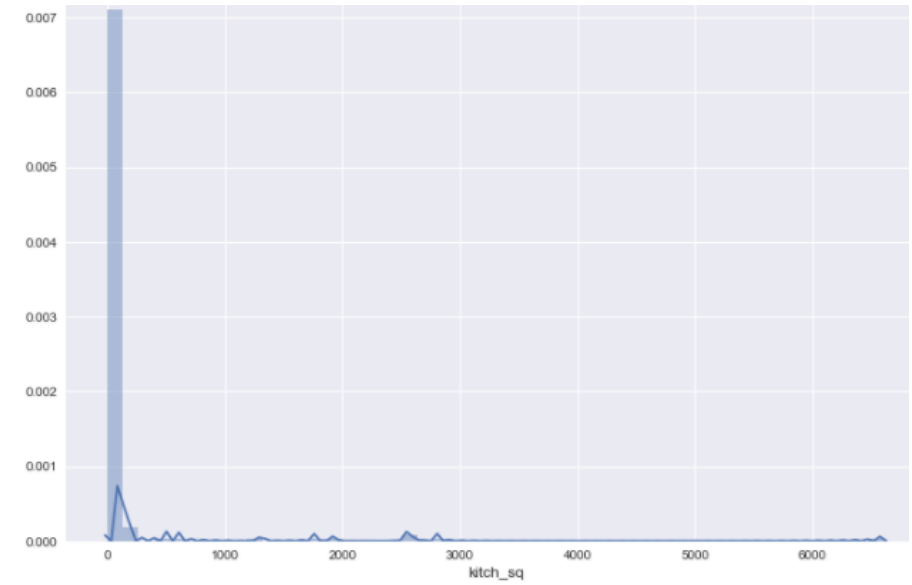
存在一个异常数据 State=33，将其改为 3

Floor：



数据存在 0，不符合实际情况。（num_room, max_floor）也存在相同情况，将所有错误数据置 NAN 处理。

Kitch_sq:



存在一些极大极小值，设置一个范围（5<kitch_sq<100）将不符合设定的错误数据置为 NAN.

另外还有一些错误不符合常理的如下：

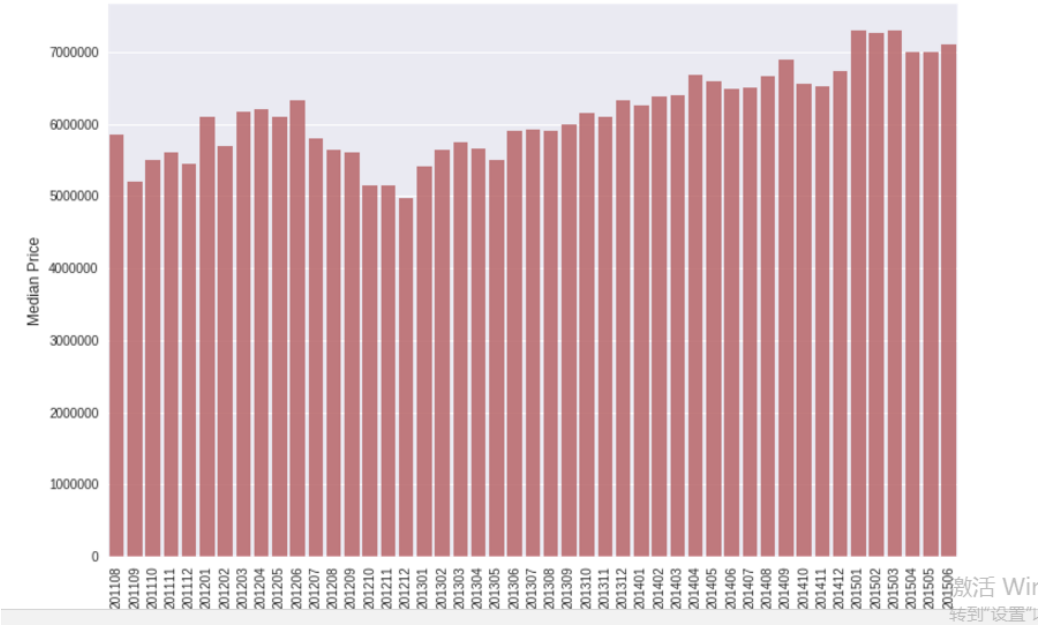
Full_life < life_sq

full_sq < kitch_sq

floor > max_floor

3.增加新特征

由于是有关于时间的，因此可对时间拆分做处理：



主要是增加了 yearmonth 这个特征，由于价格的高低走势和时间有相当大的关系，因此增加该特征。这个图在一定程度上也反应了俄罗斯的经济趋势。

其他新特征的增加主要参考了别人的代码（引用【5】）：

```
train['rel_floor'] = .05 + train['floor'] / train['max_floor'].astype(float)
train['rel_kitch_sq'] = .05 + train['kitch_sq'] / train['full_sq'].astype(float)

test['rel_floor'] = .05 + test['floor'] / test['max_floor'].astype(float)
test['rel_kitch_sq'] = .05 + test['kitch_sq'] / test['full_sq'].astype(float)

train.apartment_name=train.sub_area + train['metro_km_avto'].astype(str)
test.apartment_name=test.sub_area + train['metro_km_avto'].astype(str)

train['room_size'] = train['life_sq'] / train['num_room'].astype(float)
test['room_size'] = test['life_sq'] / test['num_room'].astype(float)
```

这里主要对每一个特征进行分析：

Rel_Floor：当前楼层在所在楼的高度百分比。从现实生活中可以看出，对于相同的一栋楼，楼层越高无疑价格越高。在这里使用百分比而不是具体楼层数主要还是考虑了不同地区的建筑分布不同，用百分比代表具体高度更具有普遍性。

Rel_kitch_sq：厨房面积占房子总面积的比例。其实就是区分厨房区和生活区的比例，一般厨房区占得比例越大，说明可居住人数少，价格也就不同。

Apartment_name：公寓名字。从含义上来看其实是指在同一地区的房子，如果房子在相同地区的话，价格的差距也不会特别明显。

Room_size：平均每个房间的大小。这个很好理解，就不多加解释了。

4.使用 MagicNumber（引用【6】）

House price change

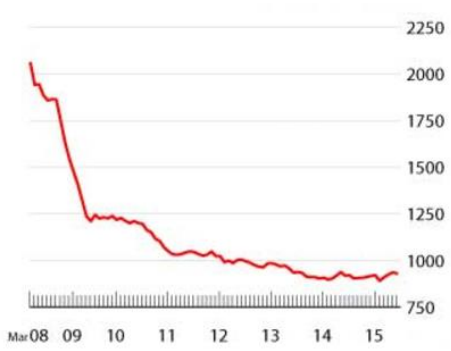
% change over a year earlier

	Q1	Q2	Q3	Q4
2016	6.52	0.69		
2015	5.05	-0.32	3.32	-1.35
2014	4.45	-3.62	0.88	-2.30
2013	-2.04	-2.51	-2.46	-1.43
2012	1.21	-1.60	-2.43	1.56
2011	0.97	-0.86	1.35	-5.44
2010	-1.38	-3.96	-5.07	-6.43
2009	-14.48	-5.90	-1.53	0.16
2008		-8.36	-1.22	-16.96

% change over a quarter (QoQ)

Source: imobiliare.ro

Average Selling Price of Apartments




```
rate_2015_q2 = 1
rate_2015_q1 = rate_2015_q2 / 0.9932
rate_2014_q4 = rate_2015_q1 / 1.0112
rate_2014_q3 = rate_2014_q4 / 1.0169
rate_2014_q2 = rate_2014_q3 / 1.0086
rate_2014_q1 = rate_2014_q2 / 1.0126
rate_2013_q4 = rate_2014_q1 / 0.9902
rate_2013_q3 = rate_2013_q4 / 1.0041
rate_2013_q2 = rate_2013_q3 / 1.0044
rate_2013_q1 = rate_2013_q2 / 1.0104
rate_2012_q4 = rate_2013_q1 / 0.9832
rate_2012_q3 = rate_2012_q4 / 1.0277
rate_2012_q2 = rate_2012_q3 / 1.0279
rate_2012_q1 = rate_2012_q2 / 1.0279
rate_2011_q4 = rate_2012_q1 / 1.076
rate_2011_q3 = rate_2011_q4 / 1.0236
rate_2011_q2 = rate_2011_q3 / 1
rate_2011_q1 = rate_2011_q2 / 1.011
```

这段代码其实是对俄罗斯的经济趋势进行了处理，用代码消除了宏观经济对价格的影响。应用之后确实能提高一些成绩。

5.简单多模型应用

```
result = output.merge(gunja_output, on="id", suffixes=['_follow','_gunja'])

result["price_doc"] = np.exp( .7*np.log(result.price_doc_follow) +
                              .3*np.log(result.price_doc_gunja) )

result["price_doc"] =result["price_doc"] *0.98
```

对多个模型进行加权平均，是在本次比赛中提升最大的一步。可能在运用多模型方面还需要多尝试。

三、比赛心得与反思

1.在本次比赛中学会了很多东西，主要还是学会了数据挖掘的几个重要步骤，数据可视化、数据清洗、特征工程还有模型应用等。
2.比赛最开始的时候还是比较摸不着头脑，因为存在大量的特征，无从下手，而且最主要的还是自己对 python 语言的不熟悉。虽然提供了参考文档及 python 学习书籍，但由于太长只看了前面的几十页，因此对一些数据处理一直不知道怎么办，只能进行简单的增删改。因此在开始时对数据处理变得很麻烦，而且一些简单的数据清洗反而导致了成绩的下降。
3.在本次跑的结果没有参考价值，cv 无从下手也是令人烦恼。后面索性放弃理解这些问题，只能通过提交后的成绩来判断修改的结果。
4.发现了自己的不足，还有耐心太少，有时一个下午辛辛苦苦查资料做处理往往得不到一个好结果，反而是参考别人发布的代码并在上面进行更改会得到一些好的结果，导致最后都是在网站上寻求别人发布且效果很好的代码，自己的思考就少了许多。

参考文献

https://www.kaggle.com/bguberfain/naive-xgb-lb-0-317

【1】【2】【3】【4】
https://www.kaggle.com/sudalairajkumar/simple-exploration-notebook-sberbank
【5】【6】
https://www.kaggle.com/cjansen/magic-numbers-private-lb-0-31647

github 链接:
https://github.com/nameiswhat/DataMiningCompetition/blob/nameiswhat-patch-1/RussiaHouse.py