学号：19122753

**《数据分析程序设计》**

**课程论文**

**题目：二手车交易价格预测**

|  |  |
| --- | --- |
| **姓 名** | 顾骏杰 |
| **学 号** | 19122753 |
| **论文评分** |  |

**2021年 10月23日**

题目描述：

赛题以预测二手车的交易价格为任务，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。

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 |

目 录

[1、引言 2](#_Toc86499180)

[2、数据预处理与数值分析 3](#_Toc86499181)

[2.1数据预处理 3](#_Toc86499182)

[2.2数据相关性分析 5](#_Toc86499183)

[2.3 非数值对象对价格的影响 7](#_Toc86499184)

[2.4 数据预处理 8](#_Toc86499185)

[3、回归分析 9](#_Toc86499186)

[3.1 线性回归 9](#_Toc86499187)

[3.2 使用集成模型进行回归 10](#_Toc86499188)

[3.2.1 随机森林 10](#_Toc86499189)

[3.2.2 极端随机森林 11](#_Toc86499190)

[3.2.3 梯度回升 11](#_Toc86499191)

[3.3 使用神经网络进行回归 12](#_Toc86499192)

[3.3.1 网络结构选择 13](#_Toc86499193)

[3.3.2 参数选择 13](#_Toc86499194)

[3.3.3 网络结构构建 13](#_Toc86499195)

[3.3.4 激活函数 14](#_Toc86499196)

[3.3.5 Loss函数 15](#_Toc86499197)

[3.3.6 神经网络回归小结 16](#_Toc86499198)

[4、结论与展望 17](#_Toc86499199)

二手车交易价格预测

[摘要]

二手车价格的评定一直是一个非常困难的问题，二手车的价格变化因素太多，即使一个二手车人对价格也很难把控，完全依据二手车市场而定。我们经常说一车一况，一车一价，没有任何两台车的价格是一模一样的，同年限，同配置，同款式，公里数相差不大，都没有发生交通事故的两台车价格可能会大不一样。本文通过大数据分析回归，希望找出对二手车价格影响较大的因素，给出不同因素对二手车价格的影响程度。

[关键字]

二手车价格，大数据，回归分析

**Used car transaction price forecast**

**Abstract:**

The evaluation of second-hand car price has always been a very difficult problem. There are too many factors affecting the price of second-hand cars. Even a second-hand car owner is difficult to predict the price, which is completely based on the second-hand car market. As we all know, no two cars have exactly the same price. The same age, the same configuration, the same style, the number of kilometers is quite similar, with no traffic accident, the price of the two cars may be quite different. Through big data analysis and regression, this paper hopes to find out the factors that have a great impact on the price of second-hand cars and give the influence degree of different factors on the price of second-hand cars.

**Key words:**

Used car price, big data, regression analysis

**第一章、引言**

车型分类是制定二手车价格评估方法的基础工作,现行国家颁布的机动车分类标准并不十分适合二手车价格评估的需要,因而研究合乎二手车价格评估自身特点的车型分类方法,可以为提高二手车价格评估准确性和客观性提供保证,也将为建立二手车价格评估计算机平台奠定基础。有必要根据现有交易数据，依托数据分析，明确不同参数对二手车价格的影响，为二手车价格的标准化定下基础，使得二手车价格有明确的标准，不再依照经验确定。有利于二手车价格持续稳定的发展。

**2、数据预处理与数值分析**

**2.1数据预处理**

在已经得到所需数据的情况下，需要对数据进行处理。数据预处理（data preprocessing）是指在主要的处理以前对数据进行的一些处理。现实世界中数据大体上都是不完整，不一致的脏数据，无法直接进行数据挖掘，或挖掘结果差强人意。为了提高数据挖掘的质量产生了数据预处理技术。 　数据预处理有多种方法：数据清理，数据集成，数据变换，数据归约等。这些数据处理技术在数据挖掘之前使用，大大提高了数据挖掘模式的质量，降低实际挖掘所需要的时间。

对于原始数据应主要从完整性和准确性两个方面去审核。完整性审核主要是检查应调查的单位或个体是否有遗漏，所有的调查项目或指标是否填写齐全。准确性审核主要是包括两个方面：一是检查数据资料是否真实地反映了客观实际情况，内容是否符合实际；二是检查数据是否有错误，计算是否正确等。审核数据准确性的方法主要有逻辑检查和计算检查。逻辑检查主要是审核数据是否符合逻辑，内容是否合理，各项目或数字之间有无相互矛盾的现象，此方法主要适合对定性（品质）数据的审核。计算检查是检查调查表中的各项数据在计算结果和计算方法上有无错误，主要用于对定量（数值型）数据的审核。

使用pandas读取csv文件，输入data.info()查看数据情况。

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 150000 entries, 0 to 149999

Data columns (total 31 columns):

# Column Non-Null Count Dtype

0 SaleID 150000 non-null int64

1 name 150000 non-null int64

2 regDate 150000 non-null int64

**3 model 149999 non-null float64**

4 brand 150000 non-null int64

**5 bodyType 145494 non-null float64**

**6 fuelType 141320 non-null float64**

**7 gearbox 144019 non-null float64**

8 power 150000 non-null int64

9 kilometer 150000 non-null float64

**10 notRepairedDamage 150000 non-null object**

11 regionCode 150000 non-null int64

12 seller 150000 non-null int64

13 offerType 150000 non-null int64

14 creatDate 150000 non-null int64

15 price 150000 non-null int64

16 v\_0 150000 non-null float64

17 v\_1 150000 non-null float64

18 v\_2 150000 non-null float64

19 v\_3 150000 non-null float64

20 v\_4 150000 non-null float64

21 v\_5 150000 non-null float64

22 v\_6 150000 non-null float64

23 v\_7 150000 non-null float64

24 v\_8 150000 non-null float64

25 v\_9 150000 non-null float64

26 v\_10 150000 non-null float64

27 v\_11 150000 non-null float64

28 v\_12 150000 non-null float64

29 v\_13 150000 non-null float64

30 v\_14 150000 non-null float64

dtypes: float64(20), int64(10), object(1)

可以看到部分值存在缺失。使用np.isnan(data).any()统计缺失数据。

|  |  |
| --- | --- |
| SaleID | False |
| name | False |
| regDate | False |
| **model** | **True** |
| brand | False |
| **bodyType** | **True** |
| **fuelType** | **True** |
| **gearbox** | **True** |
| power | False |
| kilometer | False |
| notRepairedDamage | False |
| regionCode | False |
| seller | False |
| offerType | False |
| creatDate | False |
| price | False |
| v\_0 | False |
| v\_1 | False |
| v\_2 | False |
| v\_3 | False |
| v\_4 | False |
| v\_5 | False |
| v\_6 | False |
| v\_7 | False |
| v\_8 | False |
| v\_9 | False |
| v\_10 | False |
| v\_11 | False |
| v\_12 | False |
| v\_13 | False |
| v\_14 | False |

发现model，bodyType，fuelType，gearbox存在缺失数据。

由于数据集的总体数据量非常大，含有nan数据的条目只占其中非常小一部分，认为可以直接抛弃。使用dropna将属性值有缺失的数据条目全部删去。

此外，注意到notRepairedDamage属性的dtype为object。值为’0’，‘1’， ‘-’。判断’-’是缺省值，可以使用抛弃数据的思想，也可以将其统一转化为某一个值。由于含有缺省值的项目占了总体数据集的不小一部分，不使用直接丢弃数据的方法，使用data["notRepairedDamage"] = data["notRepairedDamage"].apply(lambda x: x.replace(r'-', '0'))统一替换为0。

数据标准化（归一化）处理是数据挖掘的一项基础工作，不同评价指标往往具有不同的量纲和量纲单位，这样的情况会影响到数据分析的结果，为了消除指标之间的量纲影响，需要进行数据标准化处理，以解决数据指标之间的可比性。原始数据经过数据标准化处理后，各指标处于同一数量级，适合进行综合对比评价。数据的标准化（normalization）是将数据按比例缩放，使之落入一个小的特定区间。在某些比较和评价的指标处理中经常会用到，去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行比较和加权。其中最典型的就是数据的归一化处理，即将数据统一映射到[0,1]区间上。

为实现上述目标，我们将数数值类据值全部转化为float64数据类型，缩放到[-1,1]空间上。有利于后续的模型训练，降低收敛难度。

**2.2数据相关性分析**

Pearson相关系数（Pearson Correlation Coefficient）是用来衡量两个数据集合是否在一条线上面，它用来衡量定距变量间的线性关系。

Pearson相关系数衡量的是线性相关关系。若r=0，只能说x与y之间无线性相关关系，不能说无相关关系。相关系数的绝对值越大，相关性越强：相关系数越接近于1或-1，相关度越强，相关系数越接近于0，相关度越弱。

通常情况下通过以下取值范围判断变量的相关强度：

0.8-1.0 极强相关

0.6-0.8 强相关

0.4-0.6 中等程度相关

0.2-0.4 弱相关

0.0-0.2 极弱相关或无相关

Pearson相关系数可以由公式计算得到：

通过计算不同数值变量之间的相关性系数，可以得到下图：

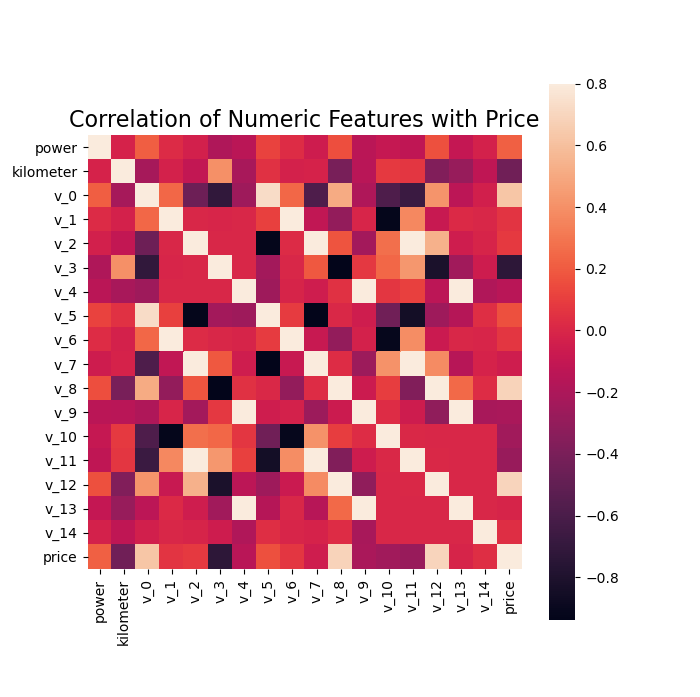


图2.2.1 数值项相关性系数

整理可得如下的数值相关性表：

表2.2.1 数值相关性

|  |  |
| --- | --- |
| price | 1.000000 |
| v\_3 | -0.730946 |
| v\_12 | 0.692823 |
| v\_8 | 0.685798 |
| v\_0 | 0.628397 |
| kilometer | -0.440519 |
| v\_11 | -0.275320 |
| v\_10 | -0.246175 |
| power | 0.219834 |
| v\_9 | -0.206205 |
| v\_5 | 0.164317 |
| v\_4 | -0.147085 |
| v\_2 | 0.085322 |
| v\_6 | 0.068970 |
| v\_1 | 0.060914 |
| v\_7 | -0.053024 |
| v\_14 | 0.035911 |
| v\_13 | -0.013993 |

可以看出，v3,v12,v8,v0,公里数与价值有着一定的关联，其余变量则与价值关联不大。

进一步探究变量之间的关系，可以通过绘制交叉表来实现。

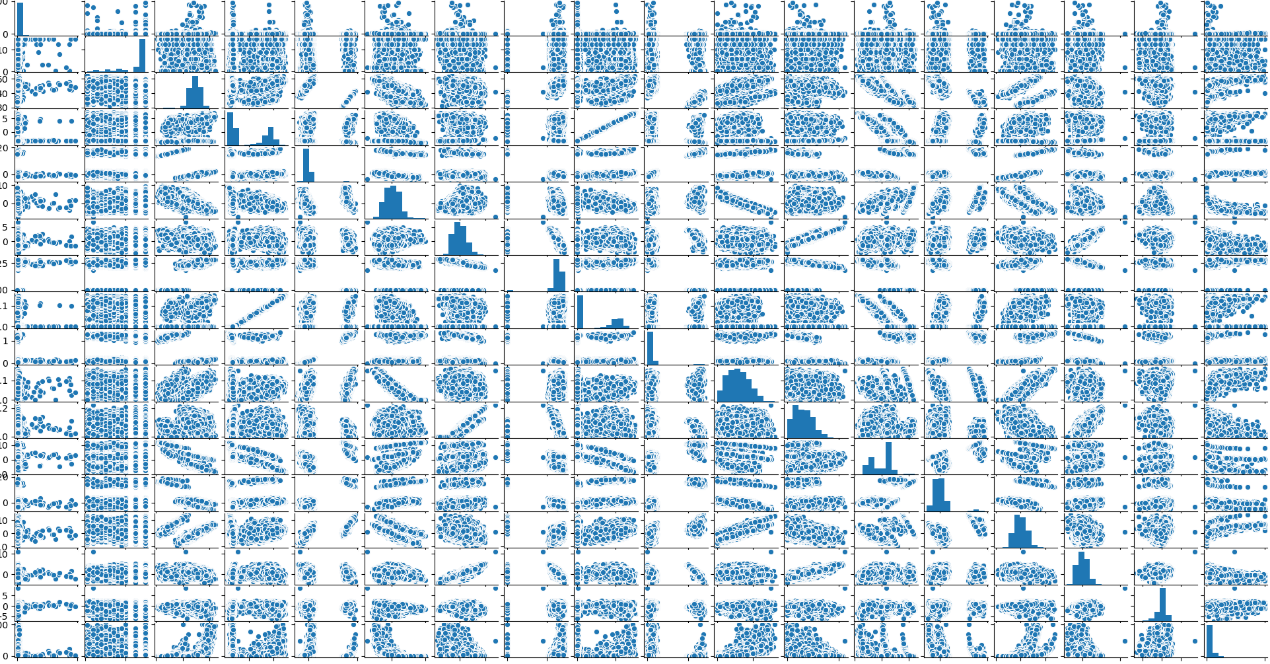


图2.2.2 数值交叉表

数值交叉表中的散点规整化程度与相关性系数图中的相对位置一一对应。相比相关性系数图，交叉表给出了更为具体、全面的诠释，能够更好的体现不同数值变量之间的关系。但缺乏定量的表示。由于后续引入其他模型的必要，数值分析以相关性系数值为主要考虑对象。

**2.3 非数值对象对价格的影响**

非数值对象对于最终结果的影响不直接体现在值的大小与最终结果的关系，而是每一类拥有自己的特性。数据集中的非数值对象主要包括'name', 'model', 'brand', 'bodyType', 'fuelType', 'gearbox', 'notRepairedDamage', 'regionCode'等一系列属性。将数据依照上述特征分类，分别绘制价格箱型图。

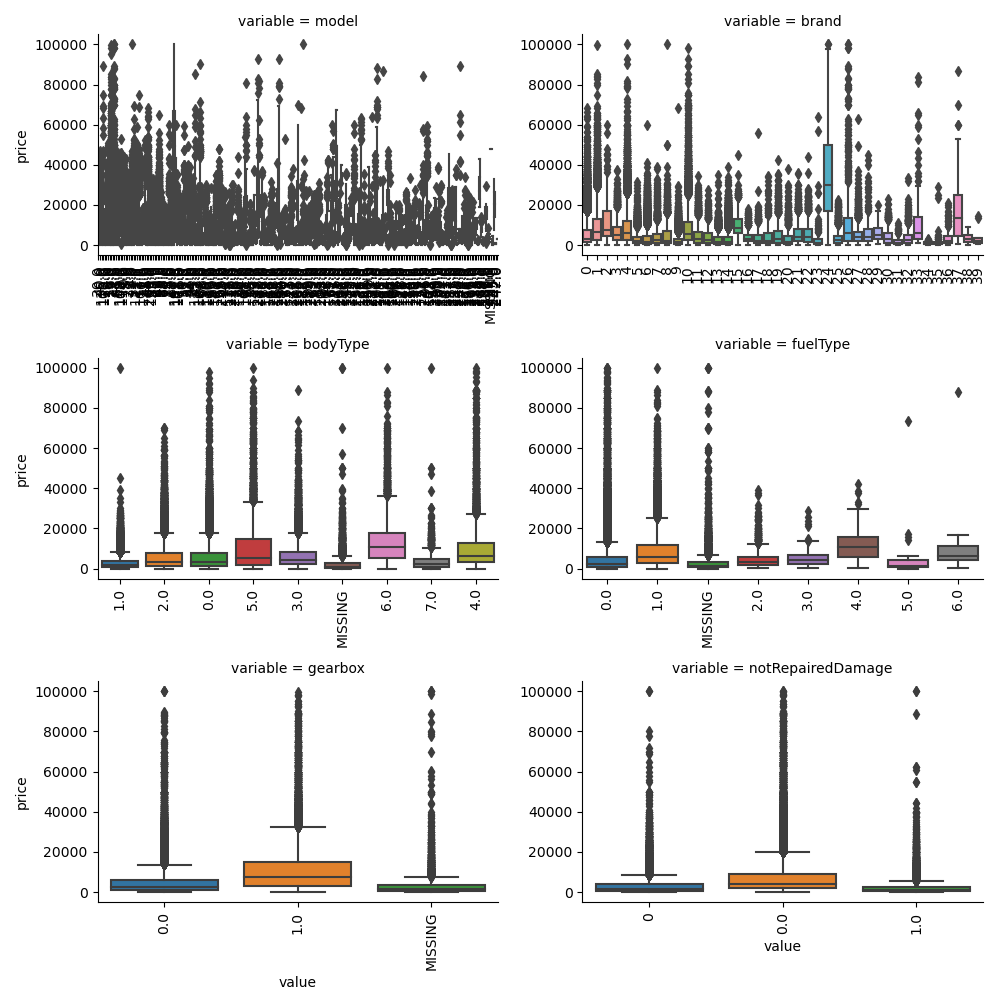


图2.3.1 分类价格箱型图

可以从图中看出，对于大部分多分类的参数，各参数对应的价格分布是离散的，难以直接进行数据的分析。需要将每一个值转化为状态参与后续的计算。对于二分类的参数，可以直接参与数值运算。

**3、回归分析**

由于本问题是一个回归问题，在这一章中，由于不同模型的量纲不是统一的，我们使用R方值误差来评定模型的好坏。

在python中，我们可以调用r2\_score(real, pred)来便捷的计算R方误差。计算公式如下：

R方误差可以由均方误差和方差快速计算的到：

其中，均方误差MSE为：

R2\_score = 1，达到最大值。即分子为 0 ，意味着样本中预测值和真实值完全相等，没有任何误差。也就是说我们建立的模型完美拟合了所有真实数据，是效果最好的模型;但通常模型不会这么完美，总会有误差存在，当误差很小的时候，分子小于分母，模型会趋近 1，仍然是好的模型。随着误差越来越大，R2\_score 也会离最大值 1 越来越远。

R2\_score = 0。此时分子等于分母，样本的每项预测值都等于均值。也就是说训练出来的模型和前面说的均值模型完全一样，还不如不训练，直接让模型的预测值全取均值。当误差越来越大的时候就出现了第三种情况。

R2\_score < 0 ：分子大于分母，训练模型产生的误差比使用均值产生的还要大，也就是训练模型反而不如直接取均值效果好。

**3.1 线性回归**

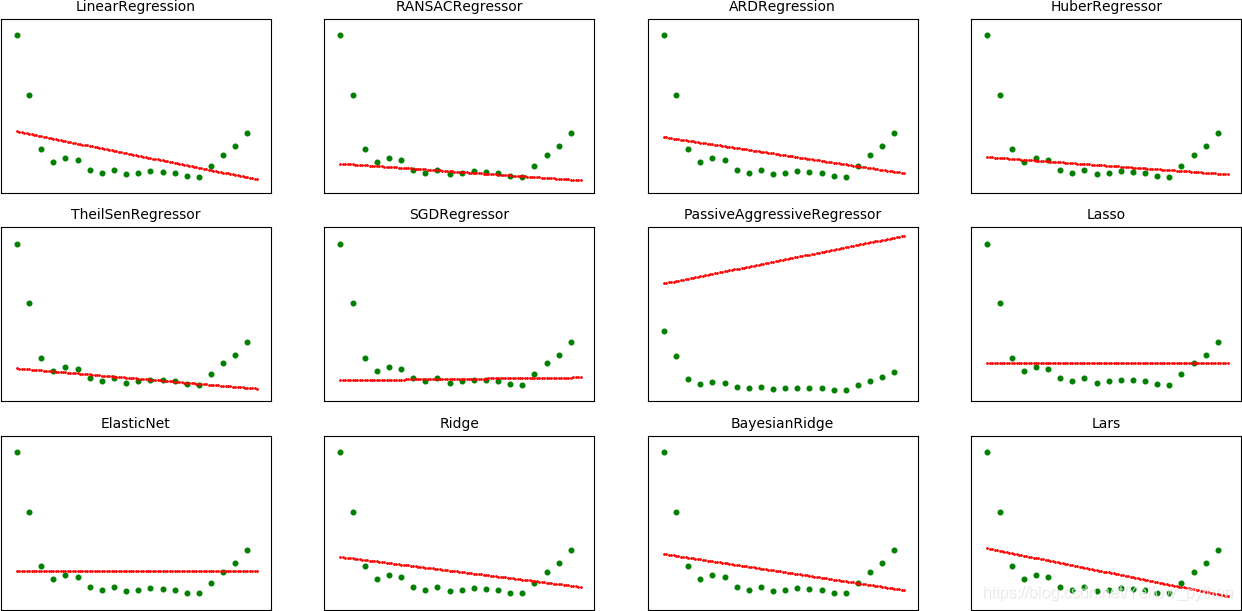


图3.1 sklearn提供的线性回归模型

首先，我们可以使用线性回归方式对数据集进行简单的回归分析。所谓线性回归，就是是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法，运用十分广泛。其表达形式为，其中为误差服从均值为0的正态分布。

可以使用sklearn库中的model = LinearRegression(normalize=True)进行简单线性回归分析。得出的R2Score为0.19。可知在本例中线性回归有一定的预测数值能力，但效果比较差。

**3.2 使用集成模型进行回归**

在python的sklearn库中提供了大量的回归模型可以用于拟合函数。我们可以借助已有的集成模型进行回归分析。

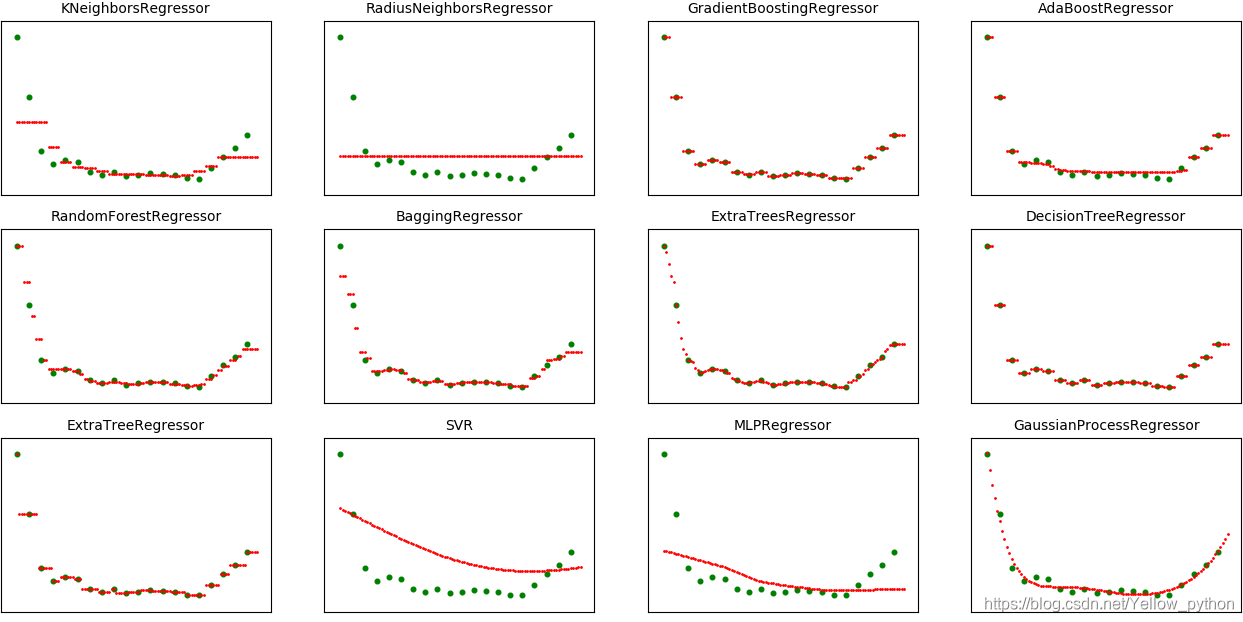


图3.2 sklearn提供的非线性回归模型

**3.2.1 随机森林**

随机森林是一种重要的基于Bagging的集成学习方法，可以用来做分类、回归等问题。

随机森林由LeoBreiman（2001）提出，从原始训练样本集N中有放回地重复随机抽取k个样本生成新的训练样本集合，然后根据自助样本集生成k个分类树组成随机森林，新数据的分类结果按分类树投票多少形成的分数而定。其实质是对决策树算法的一种改进，将多个决策树合并在一起，每棵树的建立依赖于一个独立抽取的样品，森林中的每棵树具有相同的分布，分类误差取决于每一棵树的分类能力和它们之间的相关性。特征选择采用随机的方法去分裂每一个节点，然后比较不同情况下产生的误差。能够检测到的内在估计误差、分类能力和相关性决定选择特征的数目。单棵树的分类能力可能很小，但在随机产生大量的决策树后，一个测试样品可以通过每一棵树的分类结果经统计后选择最可能的分类。

随机森林有许多优点：具有极高的准确率,随机性的引入，使得随机森林不容易过拟合,随机性的引入，使得随机森林有很好的抗噪声能力,能处理很高维度的数据，并且不用做特征选择,既能处理离散型数据，也能处理连续型数据，数据集无需规范化,训练速度快，可以得到变量重要性排序,容易实现并行化

随机森林的缺点：当随机森林中的决策树个数很多时，训练时需要的空间和时间会较大,随机森林模型还有许多不好解释的地方，是个黑盒模型。

经过训练，随机森林回归的R\_squared值为： 0.950541808299053。

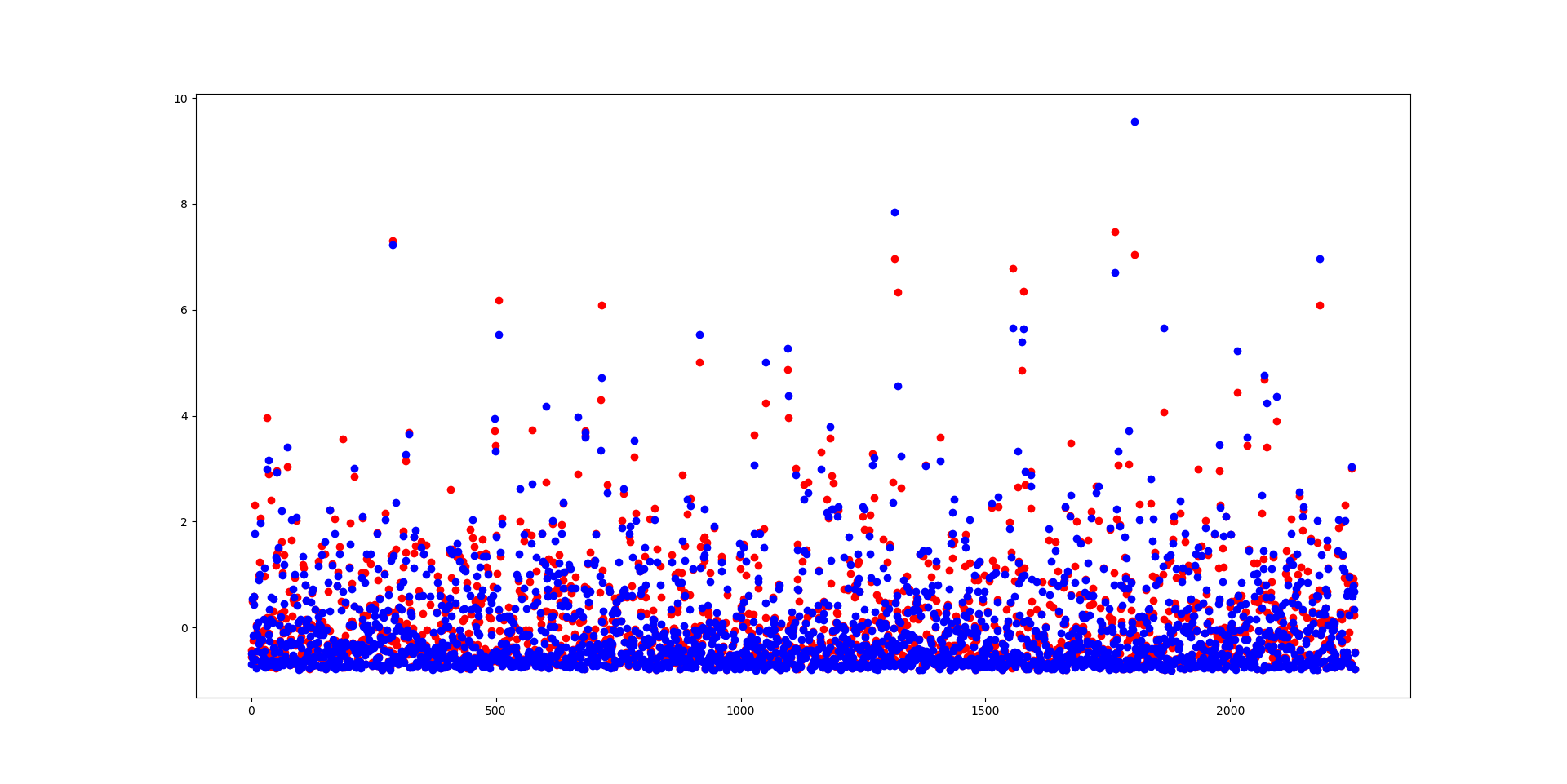


图3.2.1 使用随机森林的预测结果与原始结果对比

**3.2.2 极端随机森林**

极端随机森林同样是一种多棵决策树集成的分类器，与随机森林分类器比较，主要有两点不同：对于每个决策树的训练集，RF采用的是随机采样bootstrap来选择采样集作为每个决策树的训练集，而ET一般不采用随机采样，即每个决策树采用原始训练集。Random Forest应用的是Bagging模型，Extra Tree使用的所有的样本，只是特征是随机选取的，因为分裂是随机的，所以在某种程度上比随机森林得到的结果更加好。在选定了划分特征后，RF的决策树会基于信息增益，基尼系数，均方差之类的原则，选择一个最优的特征值划分点，这和传统的决策树相同。但是Extra tree比较的激进，会随机的选择一个特征值来划分决策树。从第二点可以看出，由于随机选择了特征值的划分点位，而不是最优点位，这样会导致生成的决策树的规模一般会大于RF所生成的决策树。也就是说，模型的方差相对于RF进一步减少，但是bias相对于RF进一步增大。在某些时候，Extra tree的泛化能力比RF更好。

极端随机森林回归的R\_squared值为： 0.9515190748002282。

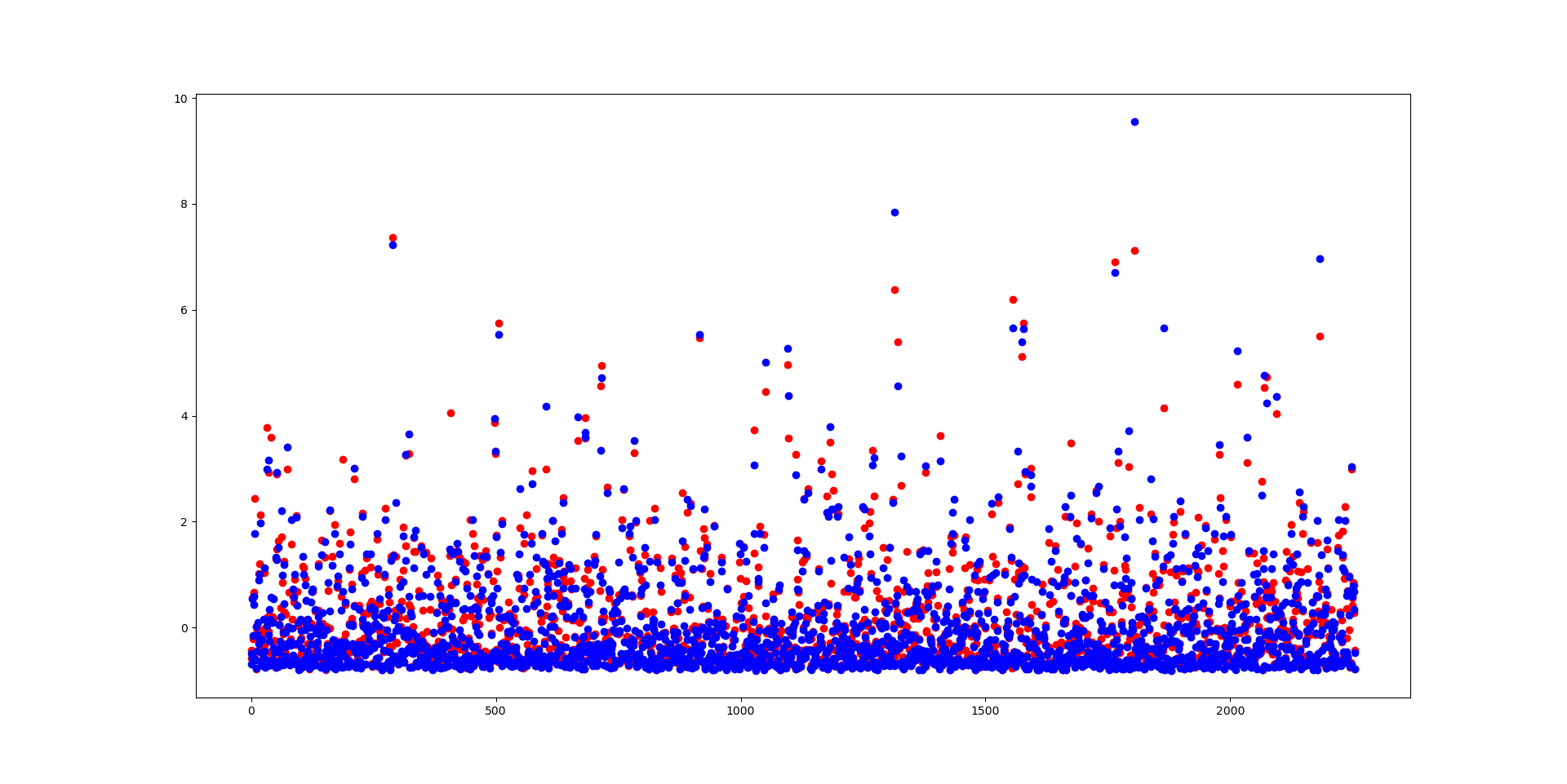


图3.2.2 使用极端森林的预测结果与原始结果对比

**3.2.3 梯度回升**

梯度提升回归树是一种从它的错误中进行学习的技术。它本质上就是集思广益，集成一堆较差的学习算法进行学习。

GBDT是基于Boosting思想的，Adaboosting是最著名的Boosting算法，其基本思想是使用多个弱分类器来构建一个强分类器。Adaboosting构造方法是一个迭代的过程，大致思路是：针对同一个训练集训练多层的弱分类器，每层使用训练集训练一个弱分类模型，我们从训练出的模型中得到预测结果。之后根据训练集中样本分类是否正确、总体分类的准确率来确定每个样本上应重新分配的权值，将修改过权重后的新数据集训练一个下层的分类器。这样不断进行训练直到有很少的错分样本，最后将每层的分类器有权重分配的融合在一起，这样下来就组成了最终的决策分类器。

每次迭代需要三次计算。

第一次计算分类误差率：

第二次计算分配给当前分类器的系数：

第三个是计算新的数据集的权值，其中为规划因子：

最终得到将所有层分类器的组合：

梯度提升回归的R\_squared值为： 0.9492847721692155。

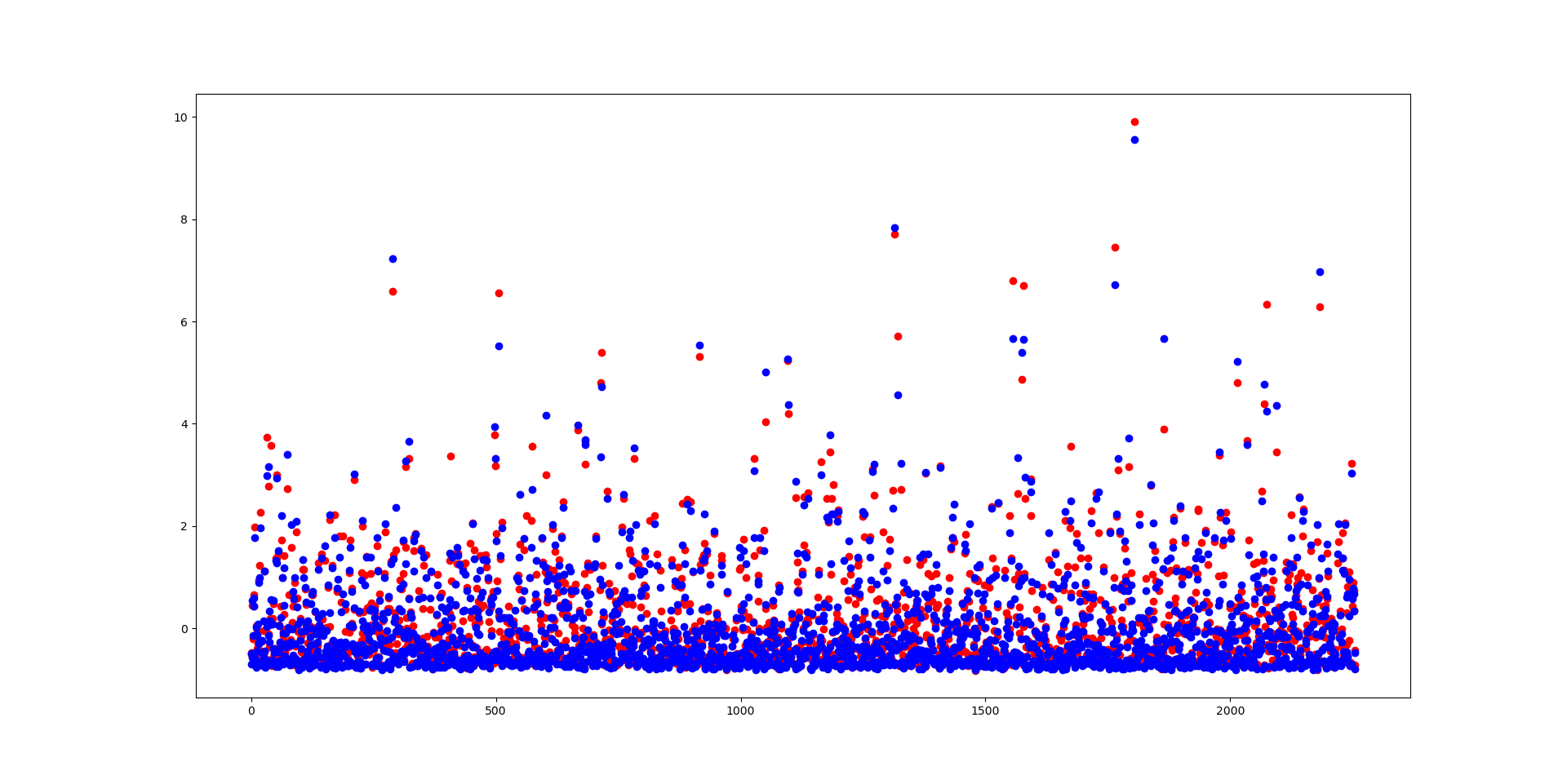


图3.2.3 使用梯度回升的预测结果与原始结果对比

**3.3 使用神经网络进行回归**

人工神经网络（Artificial Neural Networks，简写为ANNs）也简称为神经网络（NNs）或称作连接模型（Connection Model），它是一种模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。

**3.3.1 网络结构选择**

选择一个较好的神经网络模型是保证模型能够收敛并得到预期效果的最基本要求。对于本文讨论的多变量非线性回归问题中，直接使用全连接神经网络不一定可以得到好的结果，需要在其基础上增加激活函数，为模型增加非线性特性。

**3.3.2 参数选择**

为了在保证模型具有较为良好的拟合结果的基础上，尽可能降低模型的复杂度，不使用所有的特征参加训练，而选取与最终回归目标：价格因素中强相关的数值特征和二分类以及可被归纳为二分类的非数值特征进行计算。

参考2.2节数值相关性分析，最终选定的参与模型构建的参数为：

"kilometer", "v\_0", "v\_3", "v\_8", "v\_12", "v\_9", "v\_10", "v\_11", "power", "gearbox", "notRepairedDamage"。

**3.3.3 网络结构构建**

输入层参数为输入的特征值个数11个，输入11维的Tensor。输出层设定为回归目标price值，是单一值，设定为1。得到最基础的模型。

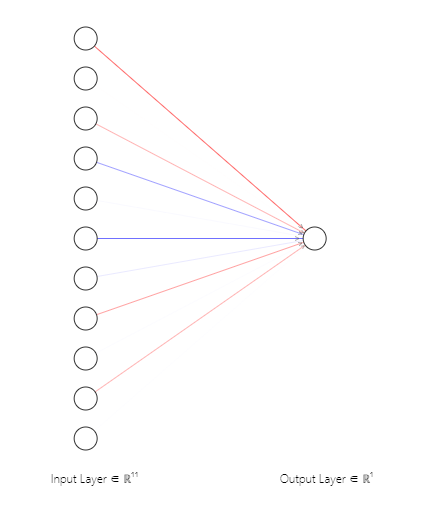


图3.3.3a 基础的神经网络

此时的神经网络近似于线性回归模型，各输入值直接通过一个线性变换作用于输出值。通过3.1节已经证明模型效果较差。必须添加隐藏层加大模型复杂度，以实现非线性映射。

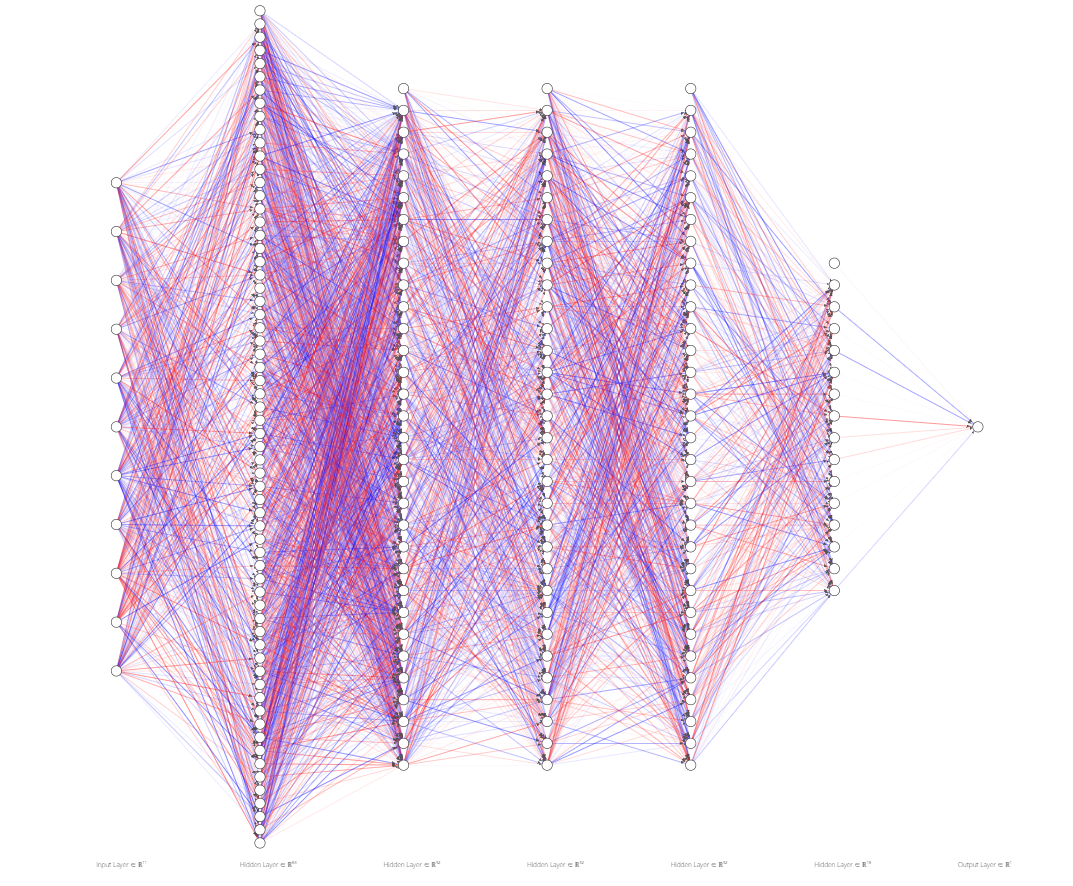


图3.3.3b 神经网络

隐藏层的意义，是把前一层的向量变成新的向量，进行一次维度重映射。这可以使得神经网络学习到更多的特征。

**3.3.4 激活函数**

全连接网络的激活函数必须使用非线性函数。线性网络的问题在于，无论如何增加网络的层数，总存在与之等效的无隐藏层的神经网络。如果使用线性函数做激活函数，无法发挥神经网络叠加层带来的优势。

常用的非线性激活函数包括Tanh, Sigmoid，SoftMax。SoftMax常用于多分类模型的输出层，将值映射到概率空间，而Sigmoid的输出值恒为正值，由于存在与输出值负相关的变量，我们采用Tanh作为隐藏层的激活函数。

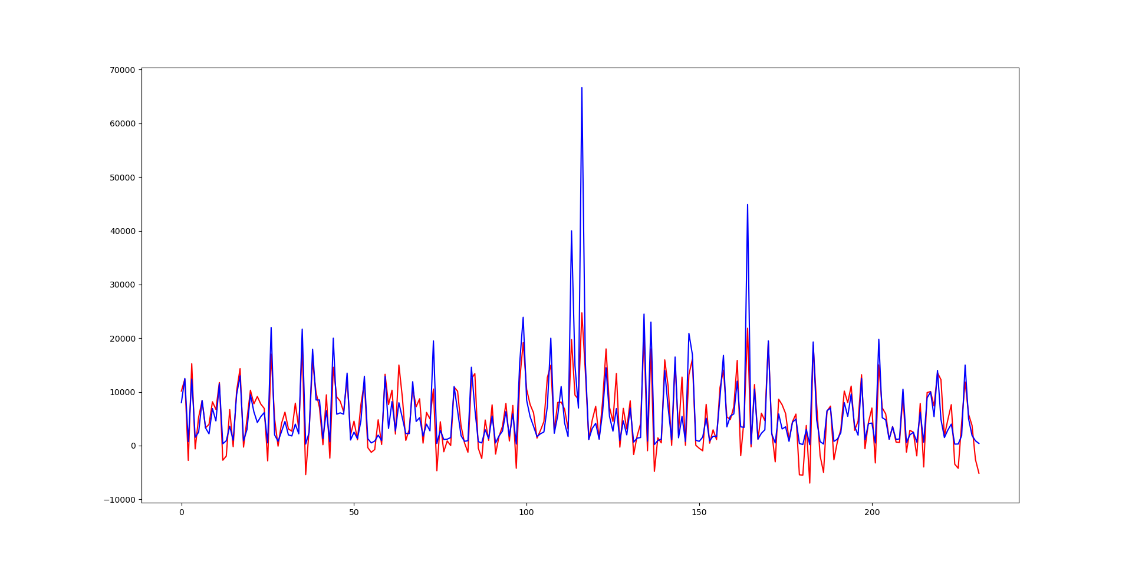


图3.3.4a 使用Tanh作为激活函数的预测值

可以发现一个问题，预测值出现了不应该出现的负值。我们在输出层增加激活函数ReLU将所有负值舍去。这有助于改善模型表现。

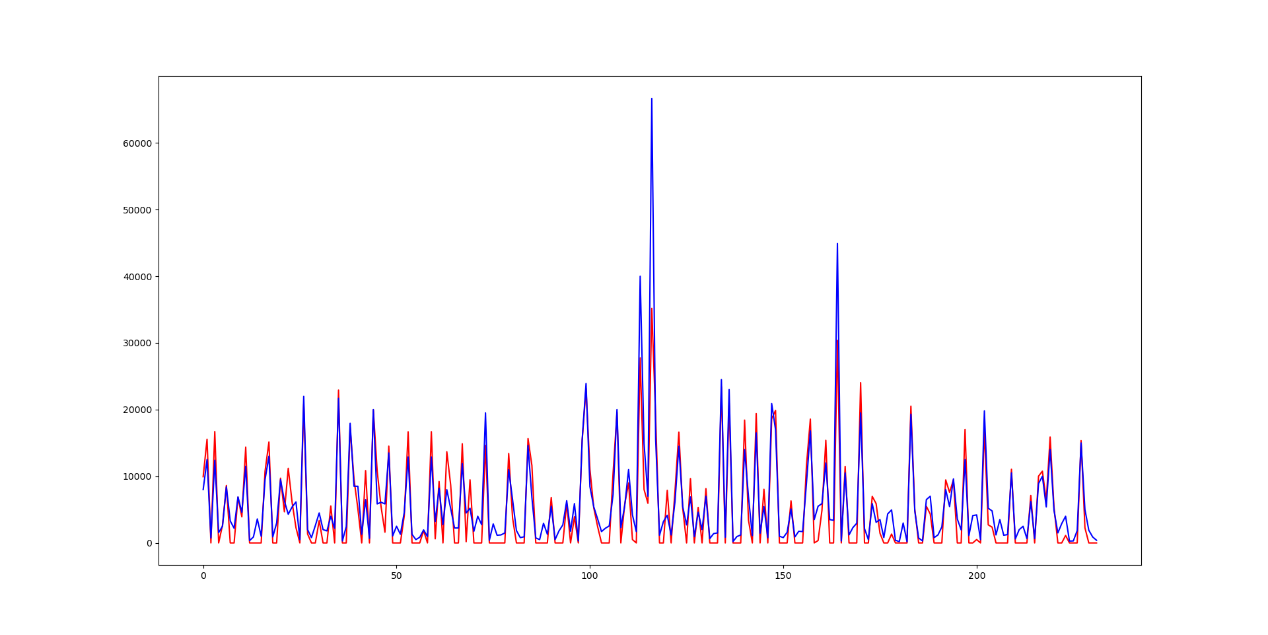


图3.3.4b 增加ReLU作为激活函数的预测值

观察到，在高价值部分拟合出的点较原先更接近真实点。但依旧有大量值为0的点，可以认为是负值被ReLU函数变化为0，仍没有解决出现负值的问题。

**3.3.5 Loss函数**

损失函数大致可以分成两类：回归（Regression）和分类（Classification）。常用的Loss函数包括均方误差MSE和交叉熵CEE。根据本文任务是回归的目标，应当选择MSE作为损失函数。

均方误差指的就是模型预测值 f(x) 与样本真实值 y 之间距离平方的平均值。MSE 曲线的特点是光滑连续、可导，便于使用梯度下降算法，是比较常用的一种损失函数。而且，MSE 随着误差的减小，梯度也在减小，这有利于函数的收敛，即使固定学习因子，函数也能较快取得最小值。

平方误差有个特性，就是当 yi 与 f(xi) 的差值大于 1 时，会增大其误差；当 yi 与 f(xi) 的差值小于 1 时，会减小其误差。这是由平方的特性决定的。也就是说， MSE 会对误差较大（>1）的情况给予更大的惩罚，对误差较小（<1）的情况给予更小的惩罚。从训练的角度来看，模型会更加偏向于惩罚较大的点，赋予其更大的权重。如果样本中存在离群点，MSE 会给离群点赋予更高的权重，但是却是以牺牲其他正常数据点的预测效果为代价，这最终会降低模型的整体性能。

为改善这一问题，可以使用平均绝对误差MAE。平均绝对误差是所有单个观测值与算术平均值的偏差的绝对值的平均。平均绝对误差可以避免误差相互抵消的问题，因而可以准确反映实际预测误差的大小。问题在于MAE 大部分情况下梯度都是相等的，这意味着即使对于小的损失值，其梯度也是大的。这不利于函数的收敛和模型的学习。

从计算机求解梯度的复杂度来说，MSE 要优于 MAE，而且梯度也是动态变化的，能较快准确达到收敛。但是从离群点角度来看，如果离群点是实际数据或重要数据，而且是应该被检测到的异常值，那么我们应该使用MSE。另一方面，离群点仅仅代表数据损坏或者错误采样，无须给予过多关注，那么我们应该选择MAE作为损失。综合MSE与MAE的优缺点，在这里使用Huber Loss作为损失函数。

Huber Loss的公式如下：

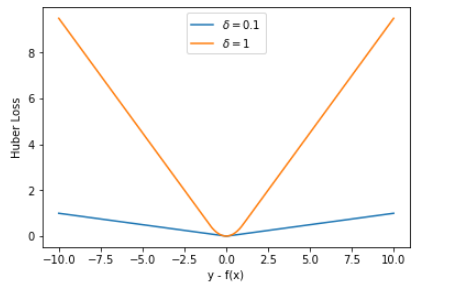


图3.3.5a Huber Loss的函数图像

Huber Loss包含了一个超参数。值的大小决定了 Huber Loss 对 MSE 和 MAE 的侧重性，当时，变为 MSE；当时，则类似于 MAE，因此 Huber Loss 同时具备了 MSE 和 MAE 的优点，减小了对离群点的敏感度问题，实现了处处可导的功能。

将Loss函数更改为Huber Loss后，可以得到较好的结果。

**3.3.6 神经网络回归小结**

采用与price值中强相关的变量作为输入，使用PyTorch构建神经网络，建立全连接神经网络，隐藏层采用Tanh，输出层使用ReLU，层之间增加DropOut层防止过拟合，使用Huber Loss作为损失函数，选用Adam优化器。

在小规模数据集上尝试训练2000个Epoch：

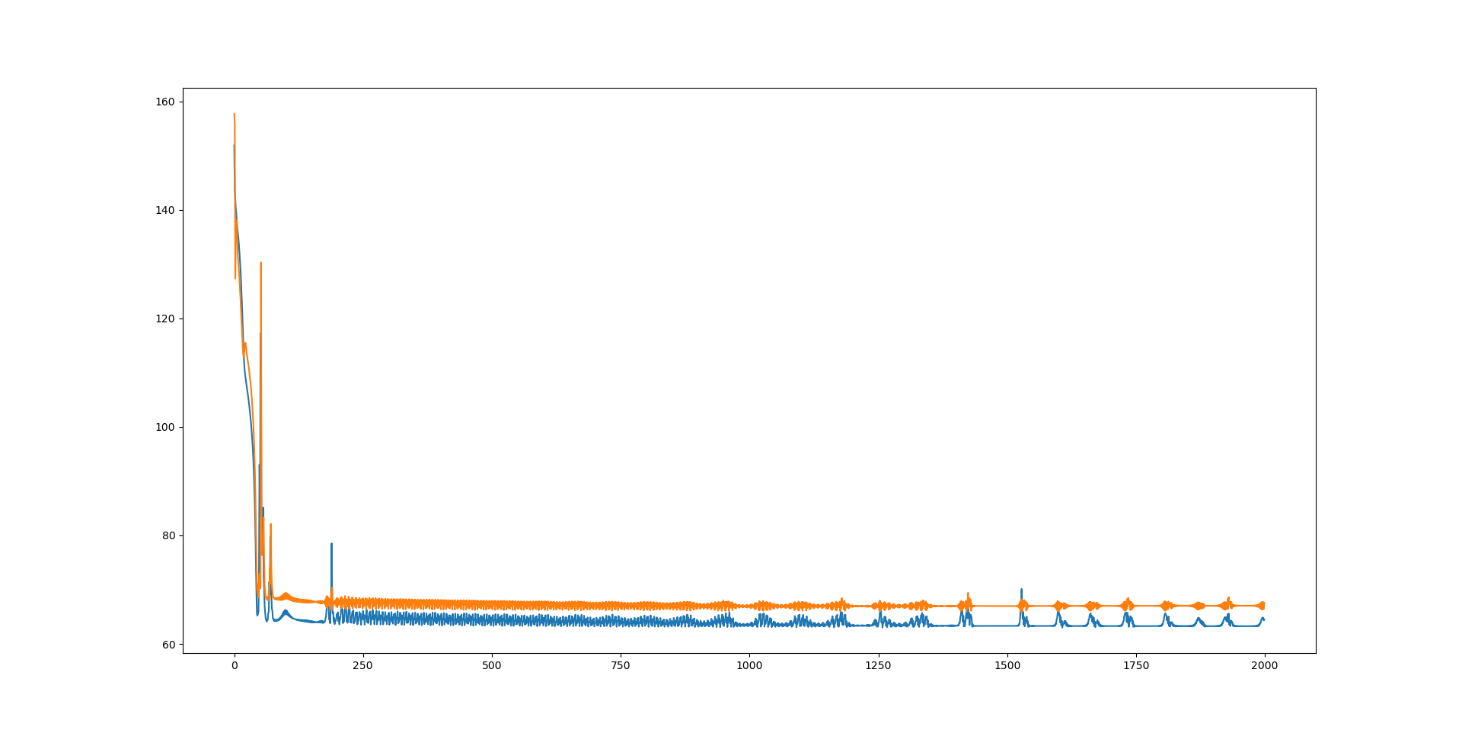


图3.3.6a 小规模数据集的Loss曲线

可看到在小规模数据集上模型能够收敛。采用全量数据集，Batch Size设为64，训练500个Epoch：

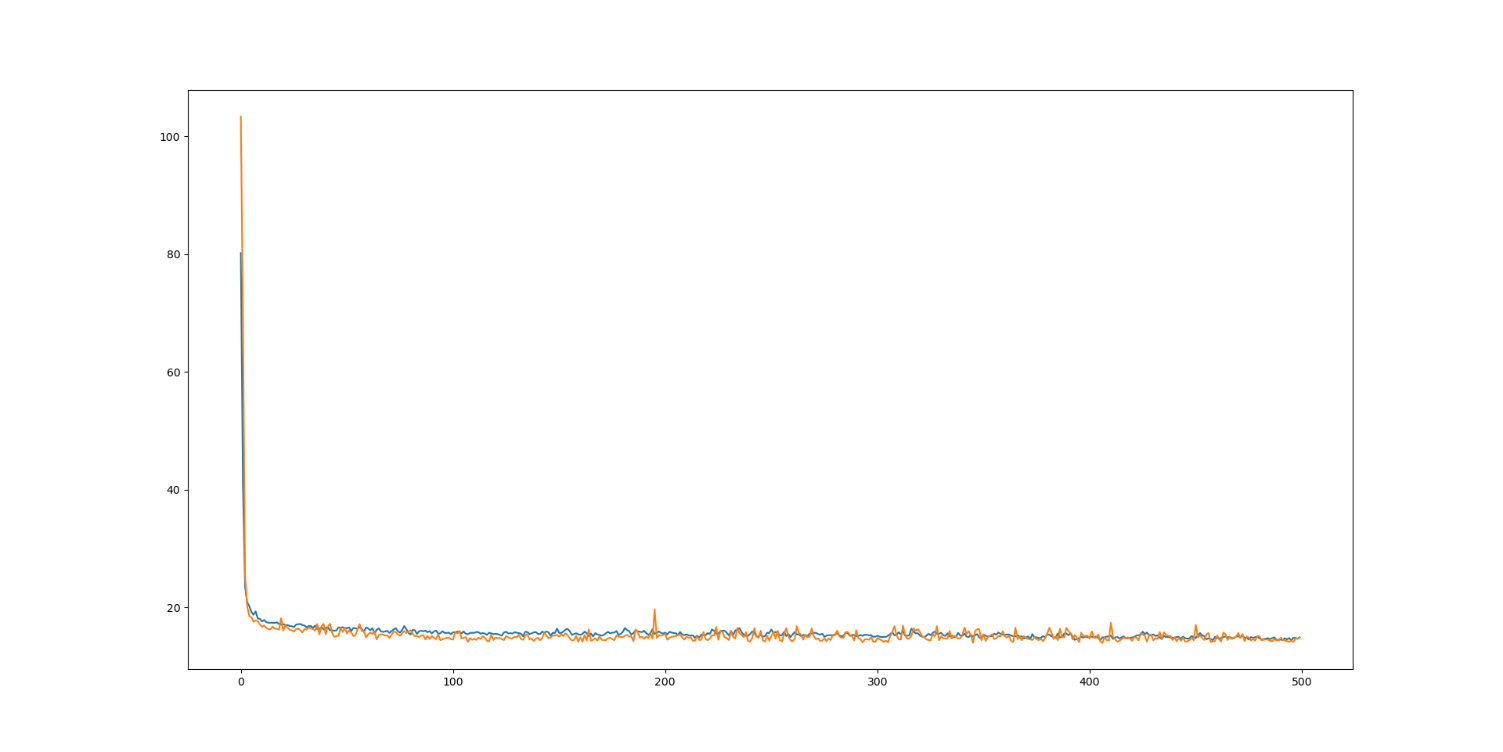


图3.3.6b 全量数据集的Loss曲线

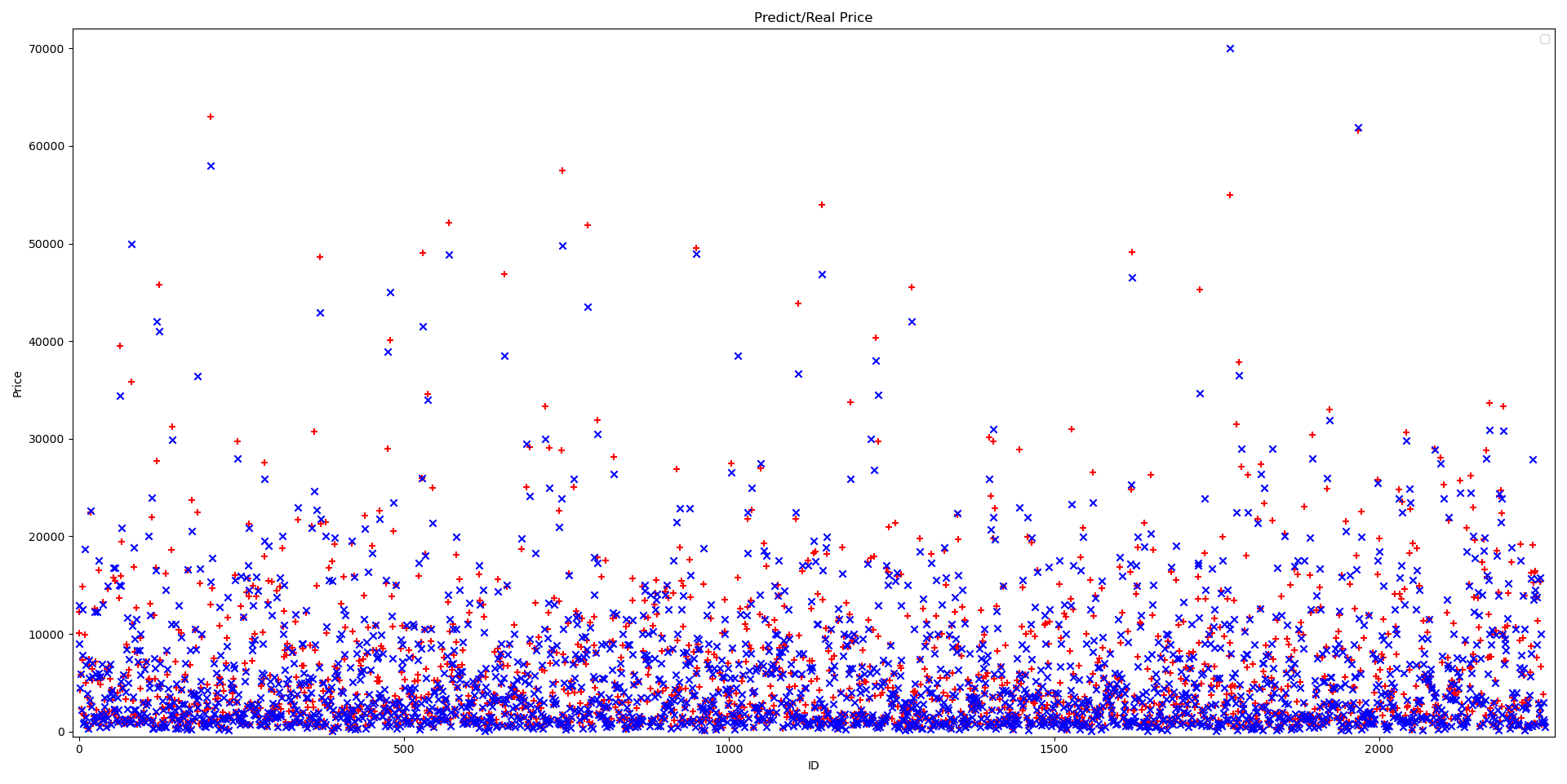


图3.3.6c 使用NN的预测结果与原始结果对比

神经网络回归的R\_squared值为： 0.9364836721592325。

**4、结论与展望**

本文通过对大量数据进行分析，通过不同模型的训练，对已有的大量数据进行拟合，归纳总结出了精度、正确率较好的一系列模型。通过项目，对数据挖掘、数据清洗、特征提取、模型训练有了更深刻的理解。

**参考文献**

[1] 何清, 李宁, 罗文娟,等. 大数据下的机器学习算法综述[J]. 模式识别与人工智能, 2014.

[2] 方匡南, 吴见彬, 朱建平,等. 随机森林方法研究综述[J]. 统计与信息论坛, 2011, 26(003):32-38.

[3] 李欣海. 随机森林模型在分类与回归分析中的应用[J]. 应用昆虫学报, 2013, 50(4):8.

[4] 王建荣, 娄超, 于健,等. 基于极端梯度推进和随机森林的热门微博预测方法:, CN107423339A[P]. 2017.

[5] 周志华, 陈世福. 神经网络集成[J]. 计算机学报, 2002, 25(01).

[6] 刘苏苏, 孙立民. 支持向量机与RBF神经网络回归性能比较研究[J]. 计算机工程与设计, 2011, 32(012):4202-4205.

**部分源代码**

1、神经网络模型：

import torch.nn as nn

class FCNNModel(nn.Module):

def \_\_init\_\_(self):

super(Model, self).\_\_init\_\_()

self.model = nn.Sequential(

nn.Linear(11, 128),

nn.Tanh(),

nn.Linear(128, 256),

nn.Tanh(),

nn.Linear(256, 128),

nn.Linear(128, 64),

nn.Linear(64, 64),

nn.Dropout(0.1),

nn.Linear(64, 1),

nn.ReLU(),

)

def forward(self, x):

x = self.model(x)

x = x.squeeze(-1)

return x

2、创建新dataset读取DF数据

from torch.utils.data import Dataset

class MyDataset(Dataset):

def \_\_init\_\_(self, data\_X, data\_y):

self.data\_X = data\_X

self.data\_y = data\_y

def \_\_getitem\_\_(self, idx):

return self.data\_X.iloc[idx, :].values, self.data\_y[idx]

def \_\_len\_\_(self):

return self.data\_X.shape[0]