

**课 程 实 验 报 告**

**课程名称： 人工智能导论**

**专业班级： 计卓2001**

**学 号： U202015290**

**姓 名： 朱子城**

**指导教师： 金 燕**

**报告日期： 2021.12.31**

**计算机科学与技术学院**

目 录

[知识介绍 3](#_Toc10318)

[线性回归理论基础 3](#_Toc6600)

[回归效果的显著性检验 3](#_Toc20873)

[基于机器学习的线性回归与梯度下降 5](#_Toc20518)

[算法原理 6](#_Toc17247)

[算法步骤 9](#_Toc14968)

[实验结果及分析 10](#_Toc18705)

[心得体会 11](#_Toc20232)

[附录A 源代码 12](#_Toc16270)

[附录B 参考文献 15](#_Toc24493)

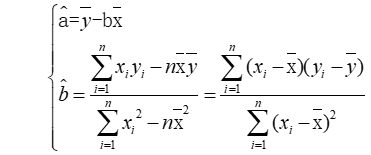
## 

# 知识介绍

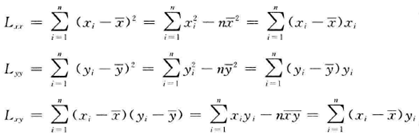
## 线性回归理论基础

一元线性回归是分析只有一个自变量（自变量x和因变量y）线性相关关系的方法。一元线性回归分析的数学模型为：y = a+bx+ε。

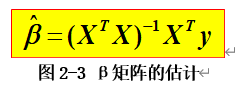
使用偏差平方和分别对参数a和参数b求偏导，可以得到线性模型的未知参数a、b的最小二乘估计值，其中，偏差平方和定义为∑(yi-a-bXi)2，a和b的唯一解如图所示。



为了方便回归效果显著性检验，根据b的估计，引入LXX、LYY、LXY三个数学符号，这三个数学符号定义如图所示。



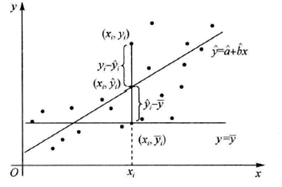
在现实问题研究中，因变量的变化往往受几个重要因素的影响，此时就需要用两个或两个以上的影响因素作为自变量来解释因变量的变化，这就是多元回归。也就是说，当多个自变量与因变量之间是线性关系时，所进行的回归分析就是多元性回归。多元线性回归的数学模型为：y=β0+β1X1+β2X2+…++βpXp+ε。使用残差平方和分别对参数βi（i=0,1,…,p）求偏导，可以得到线性模型的未知参数βi（i=0,1,…,p）的估计值，β矩阵的估计值如图所示。



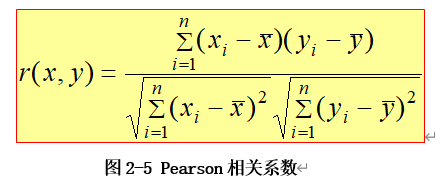
## 回归效果的显著性检验

对平面上杂乱无章的点，利用最小二乘法求解出的线性回归方程是毫无意义的，线性回归反映出的趋势描述是否合理，需要一个数量指标来度量。

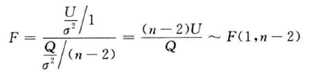
数据总的波动可以用离差平方和LYY来描述。它表示y的各次离差yi-y ̅的平方和。LYY数值越大，说明yi数值波动越大，也就是越分散。离差平方和LYY可以分解为回归直线上y的离差平方和U以及yi与回归直线上的y间的差的平方和Q。其中，U是由于x对y的线性相关关系引起的y的分散性，Q是由随机误差引起的分散性。yi-y ̅分解如图所示。在总和中，U所占比重越大，说明随机误差所占的比重越小，回归效果越显著。故此，可以使用决定系数R2来度量线性回归效果是否显著，R2作为拟合优度，表示用直线来拟合数据的好坏，R2等于U/Lyy。



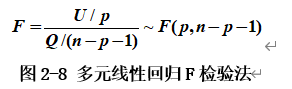
R2开方后的结果为皮尔逊相关系数，皮尔逊（Pearson）相关系数可以用来衡量两个数据集合是否在一条线上面，从而衡量定距变量间的线性关系。相关系数的绝对值越大，相关性越强；相关系数越接近于1或-1，相关度越强，相关系数越接近于0，相关度越弱。当|r|>=0.8时，x和y强相关，当|r|<0.3时，x和y弱相关。皮尔逊相关系数定义为如图所示。



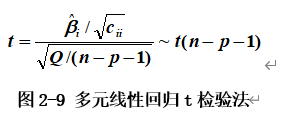
对于一元线性回归模型，线性回归模型效果的显著性可以通过假设检验问题H0：b=0；H1：b≠0进行判断，检验方法包括F检验法和t检验法。F检验属于回归方程显著性检验，是检验x与y是否相关的检验方法。t检验是回归系数显著性检验，是检验变量x是否有用的方法。H0成立时，两种检验方法定义如图所示。H0不成立时，对于给定的显著性水平α，当F>F1-α(1,n-2)时，回归效果显著。当|t|>t1-α/2(n-2)时，认为回归系数影响显著，否则回归系数的效果不显著。

f9kzosiv

对于多元线性回归模型，回归效果的显著性可以使用F检验法通过假设检验问题H0：β0=β1=β2=…=βp=0；H1：βi（i=0,1,…,p）不全为0进行判断，H0成立时，F检验方法定义如图所示。H0不成立时，对于给定的显著性水平α，当F>F1-α(p,n-p-1)时，回归效果显著。



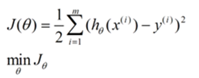
回归系数的显著性检验可以使用t检验法通过假设检验问题H0：βi=0；H1：βi≠0进行判断。H0成立时，t检验方法定义如图2-9所示。H0不成立时，对于给定的显著性水平α，当|t|>t1-α/2(n-p-1)时，认为回归系数影响显著，否则回归系数的效果不显著。



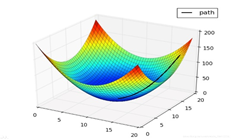
## 基于机器学习的线性回归与梯度下降

机器学习横跨计算机科学、工程技术和统计学等多个学科，渗透到了人们生产和生活中的各个领域，被广泛应用于各行各业之中，在当今世界激烈的竞争中，掌握和理解机器学习的基础模型和基本方法是非常有必要的。

机器学习中的线性回归模型以数理统计的线性回归模型为基础，它用一条直线对数据点进行拟合，在机器学习中，回归问题的求解过程就是寻找最佳拟合参数集的过程，也就是寻找满足使得估计值与实际值的方差最小时的参数解，这个过程用到了损失函数，损失函数定义如图所示。利用损失函数，可以求解最佳拟合参数集。利用损失函数进行求解可以采用梯度下降法。

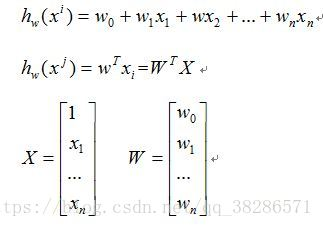


梯度下降法的计算过程就是沿梯度下降的方向求解极小值或沿梯度上升方向求解极大值。一般情况下，梯度向量为0的话说明是到了一个极值点，此时梯度的幅值也为0。采用梯度下降算法进行最优化求解时，算法迭代的终止条件是梯度向量的幅值接近0或接近一个非常小的常数阈值。梯度下降的过程如图所示。



# 算法原理

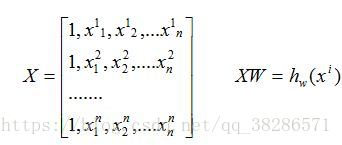
函数模型（model）：



假设有训练数据

4fxmp0uz

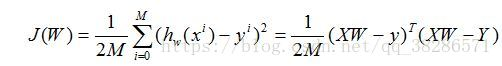
我们写成矩阵的形式为



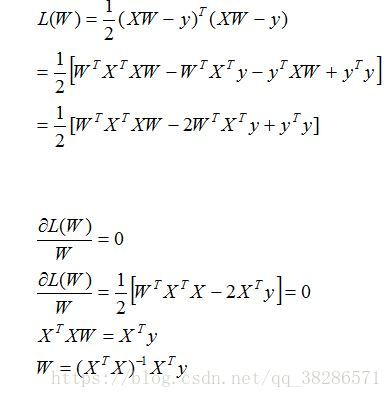
对于损失函数（cost）

现在我们需要根据给定的X求解W的值，这里采用最小二乘法， 何为最小二乘法，其实很简单。我们有很多的给定点，这时候我们需要找出一条线去拟合它，那么我先假设这个线的方程，然后把数据点代入假设的方程得到观测值，求使得实际值与观测值相减的平方和最小的参数。对变量求偏导联立便可求。

因此损失代价函数为



梯度下降算法求解：   
现在我们的目的就是求解出一个使得代价函数最小的W：   
a.矩阵满秩可求解时（求导等于0）



b.矩阵不满秩时（梯度下降）：

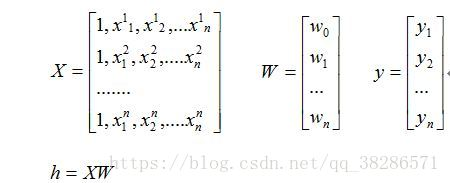
梯度下降算法是一种求局部最优解的方法，对于F(x)，在a点的梯度是F(x)增长最快的方向，那么它的相反方向则是该点下降最快的方向，具体参考wikipedia。 原理：将函数比作一座山，我们站在某个山坡上，往四周看，从哪个方向向下走一小步，能够下降的最快；注意：当变量之间大小相差很大时，应该先将他们做处理，使得他们的值在同一个范围，这样比较准确。

1）首先对θ赋值，这个值可以是随机的，也可以让θ是一个全零的向量。

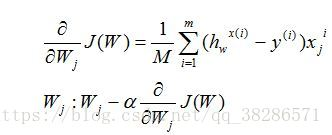
2）改变θ的值，使得J(θ)按梯度下降的方向进行减少。描述一下梯度减少的过程，对于我们的函数J(θ)求偏导J；

acgi4eao

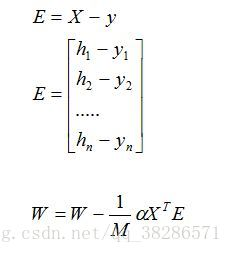
假设有数据集D时：



对损失函数求偏导如下：



使用矩阵表示



基本上每个模型都会有一个对应的目标函数，可以通过不同的最优化求解方法（梯度下降）对这些对应的目标函数进行求解。

# 算法步骤

##### 1.读取数据

##### 2.数据预处理

第一列数据是ID，对房价并无影响，先把它单独抽离出来。

然后进行缺失值处理

分析含缺失值特征的作用，没用的特征直接删除，有用的特征依据缺失量，少则删除样本，多则用mean,median或mod补全；

首先，如果缺失率达到15%以上，那这项特征应该予以删除并认为数据集中不存在这样的特征，既假定它是不存在的，因此，本数据集删除的特征有：’PoolQC’,‘MiscFeature’,‘Alley’，‘Fence’，‘FireplaceQu’和‘LotFrontage’这几列。

其次，在剩下的含缺失值变量中，以Garage开头的5个GarageX特征具有相同数量的缺失值，据此推测他们可能代表的是同一组观测值，而关于Garage的信息，’GarageCars’已经能够很好地表征了，因此删除这几个特征，对BsmtX也可以进行同样的操作。

之后，对于MasVnrArea和MasVnrType，它们与YearBuilt和OverallQual有较强的相关性。因此，删除这两个特征也不会丢失任何信息。

然后，除了Electrical，其它无意义的含缺失值的变量都已经删除了，Electrical这个变量下只有一个样本带有缺失值，因此不妨删除带有这个缺失值的那各样本。

最后将字符串型特征映射为数值型特征

factorize函数可以将Series中的标称型数据映射称为一组数字，相同的标称型映射为相同的数字，将所有的非数值型数据转换为数值型数据。

##### 建模

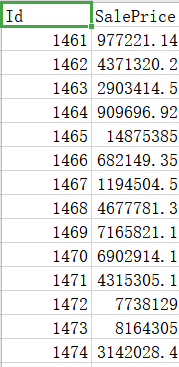
按上述算法原理进行建模。

##### 输出结果

将最终结果按格式输出到submission.csv。

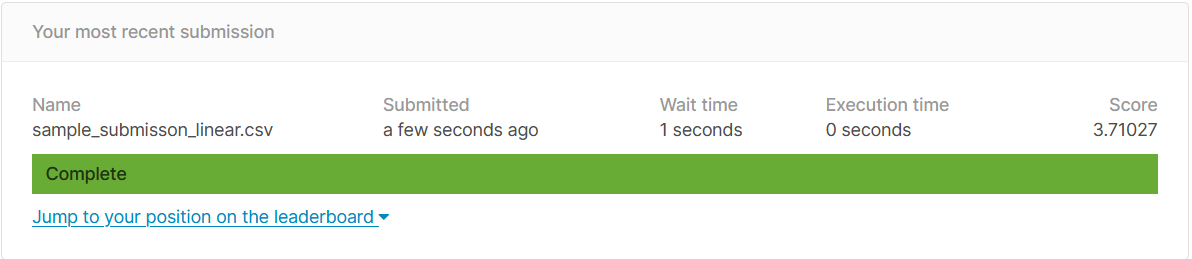
# 实验结果及分析

输出的部分结果如图所示：



上传至

https://www.kaggle.com/c/house-prices-advanced-regression-techniques验证如图所示：



# 心得体会

在学习人工智能导论这门课之前我对人工智能的认识更偏于理论和模糊，之前的上课也更多是偏向理论的学习而很少有实操的机会。这次作业让我亲身体会了人工智能技术如何应用并服务于社会，让我感受到了人工智能的魅力，同时，它对我的python编程能力与自我学习能力提出了很高的挑战，使我受益匪浅。

# 附录A 源代码

import numpy as np

import pandas as pd

class LinearRegression:

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

'''初始化模型'''

self.coef\_ = None

self.interception\_ = None

self.\_theta = None

def fit(self, X\_train, y\_train):

'''根据训练数据集X\_train,y\_train训练模型'''

X\_b = np.hstack([np.ones((len(X\_train), 1)), X\_train])

self.\_theta = np.linalg.inv(X\_b.T.dot(X\_b)).dot(X\_b.T).dot(y\_train)

self.interception\_ = self.\_theta[0]

self.coef\_ = self.\_theta[1:]

return self

def predict(self, X\_predict):

X\_b = np.hstack([np.ones((len(X\_predict), 1)), X\_predict])

return X\_b.dot(self.\_theta)

def \_\_repr\_\_(self):

return 'LinearRegression()'

#1、读取数据

train\_house= pd.read\_csv('train.csv')

#2、数据预处理

#第一列数据是ID，对房价并无影响，先把它单独抽离出来。

train\_house.drop("Id", axis=1, inplace=True)

"""

2.1 缺失值处理

分析含缺失值特征的作用，没用的特征直接删除，有用的特征依据缺失量，少则删除样本，多则用mean,median或mod补全；

首先，如果缺失率达到15%以上，那这项特征应该予以删除并认为数据集中不存在这样的特征，既假定它是不存在的，因此，本数据集删除的特征有：’PoolQC’,‘MiscFeature’,‘Alley’，‘Fence’，‘FireplaceQu’和‘LotFrontage’这几列。

其次，在剩下的含缺失值变量中，以Garage开头的5个GarageX特征具有相同数量的缺失值，据此推测他们可能代表的是同一组观测值，而关于Garage的信息，’GarageCars’已经能够很好地表征了，因此删除这几个特征，对BsmtX也可以进行同样的操作。

之后，对于MasVnrArea和MasVnrType，它们与YearBuilt和OverallQual有较强的相关性。因此，删除这两个特征也不会丢失任何信息。

然后，除了Electrical，其它无意义的含缺失值的变量都已经删除了，Electrical这个变量下只有一个样本带有缺失值，因此不妨删除带有这个缺失值的那各样本。

"""

na\_count = train\_house.isnull().sum().sort\_values(ascending=False)#得到各个特征的缺失量

na\_rate = na\_count / len(train\_house)#计算各个特征的缺失率

na\_data = pd.concat([na\_count,na\_rate],axis=1,keys=['count','ratio'])

train\_house.drop(na\_data[na\_data['count'] > 1].index, axis=1, inplace=True)

train\_house.drop(train\_house.loc[train\_house['Electrical'].isnull()].index, inplace=True)

"""

2.2 字符串型特征映射为数值型特征

factorize函数可以将Series中的标称型数据映射称为一组数字，相同的标称型映射为相同的数字，将所有的非数值型数据转换为数值型数据：

"""

for col in train\_house.columns:

if train\_house[col].dtypes == "object":

train\_house[col], uniques = pd.factorize(train\_house[col])

#3、建模

X = train\_house.drop('SalePrice', axis=1)#自变量

y = train\_house["SalePrice"]#因变量

model=LinearRegression()

model=model.fit(X,y)

test\_house = pd.read\_csv("test.csv")

#test数据集进行处理

test\_house\_ID = test\_house["Id"]

test\_house.drop("Id", axis=1, inplace=True)

test\_house.drop(na\_data[na\_data['count'] > 1].index, axis=1, inplace=True)

for col in test\_house.columns:

if test\_house[col].dtypes == "object":

test\_house[col], uniques = pd.factorize(test\_house[col])

test\_house[col].fillna(test\_house[col].mean(), inplace=True)

#预测房价

test\_predict= model.predict(test\_house)

save\_result=pd.concat([test\_house\_ID,pd.Series(abs(test\_predict))],axis=1,keys=["Id","SalePrice"])

save\_result.to\_csv("submisson.csv",index=False)

# 附录B 参考文献

[1]多元线性回归与BP神经网络预测模型对比与运用研究[J] 张景阳,潘光友.昆明理工大学学报(自然科学版) . 2013 (06)

[2]多元线性回归的数学模型[J] 刘严.沈阳工程学院学报(自然科学版) . 2005 (Z1)

[3]多元线性回归统计预测模型的应用[J] 冷建飞,高旭,朱嘉平.统计与决策 . 2016 (07)

[4]基于多变量自回归分析的北京房价预测研究[J]. 刘永泽. 现代商贸工业. 2019(06)