**第一部分概览**

机器学习是什么：A field of study on the development of *computer algorithms* for *transforming data* into *intelligent action*. goal: *teaching computers how to use data to solve a problem* s数据来源：the available data；statistical methods, and computing power

The basic learning process

四个基本机器学习步骤1.Data storage utilizes observation, memory, and recall to provide a factual basis for further reasoning 2.Abstraction involves the translation of stored data into broader representations and concepts 3.Generalization uses abstracted data to create knowledge and inferences that drive action in new contexts 4.Evaluation provides a feedback mechanism to measure the utility of learned knowledge and inform potential。机器学习算法应用步骤：数据搜集 数据探索准备 模型训练 模型评估 模型改进。 机器学习的类型**Supervised learning** **Unsupervised learning**

**reinforcement learning**]) **adversarial learning Meta-learners**

回归：dependent variable；independent variables Slope -intercept form (or coefficients, parameters) simple linear regression multiple linear regression

more advanced regression analysis logistic regression multinominal logistic regression Poisson regression generalized linear models (GLMs)广义线性

最小二乘法：残差平方和RSS 多元线性回归 the relationship between a dependent variable and two or more independent predictor variables

**第二部分线性回归**

基本的规则： （组成部分：dependent variable independent variables Slope -intercept form (or coefficients, parameters)）

关键步骤:搜集数据、整理准备、可视化数据关系和特点、训练模型、评估效果、改进、预测. 模型分类：logistic regression；multinominal logistic regression；Poisson regression；

generalized linear models (GLMs)：（两个关键点） the distribution of the target feature must be chosen from members of the [**exponential family** of distributions](https://en.wikipedia.org/wiki/Exponential_family)includes the normal distribution, and Poisson, Binomial, and Gamma.The **link function** transforms the relationship between the predictors and the target so that it can be modeled using a linear equation, despite the original relationship being nonlinear

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**R线性回归模型**

图形用户界面, 文本, 应用程序

描述已自动生成3.拟合数据# Number of iterations and sample size

n\_iter <- 1000

图形用户界面, 文本, 应用程序

描述已自动生成sample\_size <- 100

simulate\_iteration <- **function**(sample\_size) {

t\_data <- simulate\_data(sample\_size)

linear\_test\_predictions <- predict(linear\_fit, t\_data)

cubic\_test\_predictions <- predict(cubic\_fit, t\_data)

linear\_test\_mse <- mean((t\_data$y - linear\_test\_predictions)^2)

cubic\_test\_mse <- mean((t\_data$y - cubic\_test\_predictions)^2) return(data.frame( linear\_test\_mse = linear\_test\_mse,cubic\_test\_mse =cubic\_test\_mse))}results<-map\_dfr(1:n\_iter, ~simulate\_iteration(sample\_size))

**第三部分 knn &nearest neighbor classification**

近邻分类是什么： nearest neighbor classifiers are defined by their characteristic of classifying **unlabeled examples** by assigning them to the class of **similar labeled examples**. With nearest neighbor classification, computers apply a human-like ability. recall past experiences to make conclusions about current circumstances

Knn步骤：T

choose k**；t**he algorithm requires a training dataset made up of samples that have been classified into several categories

often labeled by a nominal variable for each “unlabeled” record in the test dataset, k-NN identifies the

records in the training data that are the “nearest” in similarity The “unlabeled” test instance is assigned the class representing the majority of the

nearest neighbors

常用数据处理方式：； knn的优劣： StrengthsSimple and effective

* 文本

  描述已自动生成Makes no assumptions about the underlying data distribution
* Fast training phase

 Weaknesses

* Does not produce a model, limiting the ability to understand how the features are related to the class
* Requires selection of an appropriate k
* Slow classification phase
  + Nominal features and missing data require additional processing

图示

低可信度描述已自动生成

逻辑回归模型：

**第四部分 朴素贝叶斯和LDA**

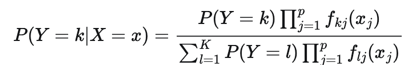
前提：Naive Bayes makes the naive assumption of independence among events assumes class-conditional independence events are independent so long as they are conditioned on the same class value（相互之间独立）

常用r包表格

描述已自动生成：e1071 klaR naivebayes h20

**Linear Discriminant Analysis (LDA)** is a classification method that is based on Bayes’ theorem

LDA is a **generative model** that estimates the probability of each class by modeling the distribution of the features for each class

图示, 示意图, 箱线图

描述已自动生成图示, 示意图

描述已自动生成LDA is a linear classifier. assumes that the featuresare **normally distributed** and that the variance of the features is the. **same** for all classes. LDA is a **parametric model**. it estimates the parameters of the normal distributions for each class

文本, 信件

描述已自动生成**第五部分 决策树**

The process of building a decision tree (in more detail)（构建决策树的步骤）

the root node represents the entire dataset

the decision tree algorithm must choose a feature to split upon

(ideally) it chooses the feature most predictive of the target class

the dataset is partitioned into groups according to the distinct values of this feature

the first set of tree branches is formed

Working down each branch, the algorithm continues to divide and conquer the data

choosing the best candidate feature each time to create another decision node

until a stopping criterion is reached

Divide and conquer might stop at a node if:

All (or nearly all) of the examples at the node have the same class

There are no remaining features to distinguish among the examples

The tree has grown to a predefined size limit

图形用户界面, 文本, 应用程序

描述已自动生成

**树状模型与线性模型**

图形用户界面, 文本, 应用程序, 信件, 电子邮件

描述已自动生成树状模型的缺点树状模型的预测准确度通常不及其他一些回归和分类方法容易出现数据拟合过度的情况尤其是在不对树进行修剪而任由其生长，或每片树叶的最小样本数设置过低的情况下树是局部最优的使用贪婪算法构建树不一定能得到最佳树回归设置中的预测面缺乏平滑性预测树可能非常不稳定数据的微小变化都可能导致最终估计树的巨大变化高方差为了解决高方差问题，套袋法、随机森林法和提升法。

**第六部分 正则化与重采样方法**

**Lasso regression**： adds a penalty term to the least squares loss function is the sum of the absolute values of the coefficients multiplied by a constant文本

描述已自动生成 is a tuning parameter that controls the strength of the penalty the larger the value of , the greater the penalty on the coefficients **Ridge regression**： adds a penalty term to the least squares loss function is the sum of the squares of the coefficients multiplied by a constantis a tuning parameter that controls the strength of the penaltythe larger the value of , the greater on the coefficientswhere is a hyperparameter controlling how much to shrink coefficients**Resampling methods：**Resampling is a method that involves repeatedly drawing samples from a training set and refitting a model of interest to each sample in order to obtain additional information about the fitted modelMain idea These methods refit a model of interest to samples formed from the training set to obtain additional information about the fitted modelThe bootstrap method involves repeatedly sampling observations from the training set with replacement refitting the model to each “new” sampleCross-validation is used to “estimate” the test error associated with a given model in order to evaluate its performance**Bootstrapping** Fundamental idea: pretend the observed data is the population The general procedures for bootstrapping is as follows Draw a sample of size n with replacement from the observed data Fit a model to the bootstrap sample Repeat the above two steps B times to produce bootstrap datasets**Cross-validation：** Training set: A subsample used to fit a model (e.g. estimate model parameters) Validation set: A subsample used for model selection (e.g. hyperparameter tuning) Test set: A subsample used only for (final) model assessment Validation-set approach: randomly divide the training set into two parts （第一次作业的3&4都是编程，所以没放上来了）

**文本, 信件

描述已自动生成**

**作业：**

a)

Answer: I expect the polynomial regression to have a lower training RSS than the linear regression because it could make a tighter fit against data that matched with a wider irreducible error.

b)

Answer: The answer is opposite to the previous question, I expect the polynomial regression to have a higher test RSS as the over fit from training would have more error than the linear regression.

c)

Answer: Polynomial regression has lower training RSS than the linear fit because of its higher flexibility, no matter what the underlying true relationship is the more flexible model will more closely follow points and reduce the training RSS.

d) Answer: There is not enough information to determine which type of RSS test will have lower regression, as we do not know how far it is from linear. If it is closer to linear than linear, then the linear regression test RSS may be lower than the cubic regression test RSS.

a )

salary (college)=50+20X1+0.07X2+35+0.01\* 𝑋1\* 𝑋2-10\* 𝑋1 salary(high school)=50+20X1+0.07X2+0.01\* 𝑋1\* 𝑋2 Therefore: Salary(college)= salary(high school)+35-10\* 𝑋1 In conclusion, for a fixed value of IQ and GPA, high school graduates earn more, on average, than college graduates provided that the GPA is more than equal to 3.5

文本

低可信度描述已自动生成b ) Salary = 50+20\*(4)+0.07\*(110)+35\*(1)+0.01\*(110x4)-10\*(4) = 137.1 (in thousands of dollars)

文本, 信件

描述已自动生成C ) FALSE, the size of the coefficient does not directly determine its significance. It should look at the internal relationship of the calculation.

When X = 0.6, the range is [0.55 , 0.65], therefore 0.05 is the key value. p=1 and assume X is distributed on [0 , 1], therefore the range is [0.05 , 0.95] Therefore it is necessary to segment x to discuss the interval of its values.

Because X in [0.05 , 0.95], the whole range is [X-0.05 , X+0.05] to make the range can reach 1.

When X in [0 , 0.05], the whole range is [0 , X+0.05] When X in [0.05 , 0.95], the whole range is 10%=0.1

When X in [0.95 , 1], the whole range is [X-0.05 , 1]

∫\_0^0.05▒〖(X+0.05-0〗)dx + ∫\_0.05^0.95▒(0.1)dx +∫\_0.95^1▒〖(1-X+0.05)〗dx

=0.00375+0.09+0.00375=0.0975=9.75%

Therefore, the available observation is 9.75%

(b)All of conditions in (b) are same with (a), and the X1 and X2 are uniformly distributed on [0 , 1]\* [0 , 1]——p = 2 features

Threrfore, the available observation is 9.75%\*9.75% 0.950625%

(c)All of conditions in (c) are same with (a), and the X1, X2, X3, X4, … ,X100, are uniformly distributed on [0 , 1]\* [0 , 1]\*…[0 , 1]——p = 100 features

Threrfore, the available observation is 9.75%\*9.75%\*…(100 times)… \*9.75% The conclusion is close to 0

Q2(a) β\_0=-6，β\_1=0.05，β\_2=1 Y=β\_0+β\_1\*X\_1+β\_2\*X\_2 Y=-6+0.05\*X\_1+X\_2 In question(a), X\_1=40, X\_2=3.5

Y=-6+2+3.5=-0.5 The resulting y is the value of k in the normalization formula, and therefore needs to be substituted into the normalization formula in order to calculate the probabilities p(k)= e^k/(1+e^k ) ,k=-0.5 p(k)= e^(-0.5)/(1+e^(-0.5) ) = 0.37754 Therefore, the probability is 37.75% .

(b) p(k)= e^k/(1+e^k ) =0.5 2e^k=1+e^k e^k = 1 k = 0 0 = -6+0.05\*X\_1+X\_2 and X\_2=3.5 Therefore, X\_1=50 In conclusion, the student needs 50 hours.

Q3 分别是区分训练集和测试机、逻辑回归、朴素贝叶斯、knn

set.seed(0) tn=0.3 sub<-sample(1:nrow(Auto),round(nrow(Auto)\*tn),rep=FALSE) data\_train<-Auto[-sub,] data\_test<-Auto[sub,] dim(data\_train) dim(data\_test)

modelA <- lm(mpg01~cylinders+weight+displacement+horsepower,data=data\_train,family = binomial) summary(modelA)

NB<-naiveBayes(mpg01~cylinders+weight+displacement+horsepower,data=data\_train) NB\_pred<-predict(NB,data\_test) mean(NB\_pred!=data\_test$mpg01)

文本

描述已自动生成install.packages("class") library(class) knn\_pred1 <-knn(data\_train[,-9], data\_test[,-9], data\_train$mpg01, k = 1) mean(knn\_pred1!=data\_test$mpg01)

#A

set.seed(1). sample1 <- sample(c(TRUE, FALSE), nrow(College), replace=TRUE, prob = c(.6,.4))

train <- College[sample1, ]. test <- College[!sample1, ]. dim(train) dim(test)

#B

lm.fit.college <- lm(Apps ~ ., data = train) summary(lm.fit.college)

lm.pred <- predict(lm.fit.college, test) lm.MSE <- mean((lm.pred - test$Apps)^2). lm.MSE. #The MSE is calculated as 1622925.

#C

install.packages('glmnet')library(glmnet) train.rm <- model.matrix(Apps~., data=train). test.rm <- model.matrix(Apps ~., data=test). grid = 10^seq(10, -2, length=100) par(mfrow = c(1, 2))

ridgemodel <- glmnet(train.rm, train$Apps, alpha = 0, lambda = grid, thresh = 1e-12) plot(ridgemodel)

cv.ridge <- cv.glmnet(train.rm, train[, "Apps"], alpha=0). plot(cv.ridge)

lambest <- cv.ridge$lambda.min. lambest pred <- predict(ridgemodel,test.rm,s=lambest). rss <- sum((pred-test$Apps)^2)

tss <- sum((test$Apps-mean(test$Apps))^2). test.rsq <- 1-(rss/tss). test.rsq

#D. lassomod <- cv.glmnet(train.rm, train$Apps, alpha = 1, lambda = grid, thresh = 1e-12). lassbest <- lassomod$lambda.min plot(lassomod). pred <- predict(lassomod,test.rm,s=lassbest). rss <- sum((pred-test$Apps)^2). tss <- sum((test$Apps-mean(test$Apps))^2). test.rsq <- 1-(rss/tss). test.rs sum(coef(lassomod)[,1]==0) names(coef(lassomod)[, 1][coef(lassomod)[, 1] == 0])