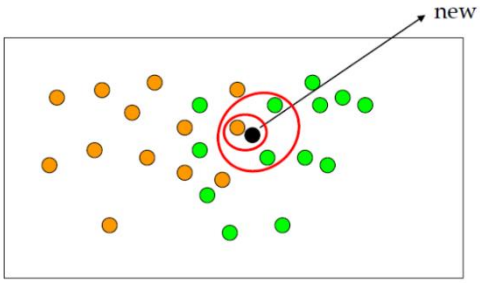
1. Description

▶ quantitative 데이터의 경우 회귀를 통해서 문제를 접근하고, qualitative 데이터는 분류 문제로 Class를 할당하는 것이 주요 관심사이다. category 또는 class로 할당하는 문제를 ***Classification***이라고 한다. 분류 방법으로는 *logistic regression, linear discriminant analysis, K-nearest neighbors* 등이 있다.

▶ ***binary logistic regression***은 범주 정보를 모르는 입력벡터 x에 대해 범주 0에 속할 확률, 범주 1에 속할 확률을 구한 후 확률 값이 큰 class로 할당되는 모델이다. K개 범주를 분류하는 다항 로지스틱 회귀의 경우 이항 로지스틱 회귀 모델 K-1개를 써서, K번째 범주에 속할 확률을 1에서 K-1개 각각의 범주에 속할 확률의 합을 뺀 나머지로 구한다.

▶ 판별분석(Discriminant Analysis)은 두개 이상의 모집단에서 추출된 표본들의 정보를 이용하여 이 표본들이 어느 모집단에서 추출된 것인지를 결정해주는 기준을 찾는 분석 방법이다. 판변분석에는 선형판별분석(***Linear Discriminant Analysis***) 그리고 이차판별분석(***Quadratic Discriminant Analysis***)가 있다. 설명변수들의 벡터가 각각 다변량 정규분포를 따르고, 변수들 간의 분산-공분산행렬이 동일하면 LDA, 동일하지 않으면 QDA라고 한다.

▶ K-최근접이웃(K-Nearest Neighbor, ***KNN***) 알고리즘은 새로운 데이터가 주어졌을 때 기존 데이터 가운데 가장 가까운 k개 이웃의 정보로 새로운 데이터를 예측하는 방법론이다. 그래서 KNN을 모델을 별도로 구축하지 않는다는 뜻으로 게으른 모델(Lazy model)이라고도 한다. KNN의 hyper parameter는 탐색할 이웃 수(k), 거리 측정 방법 두 가지이다. k가 작을 경우 데이터의 지역적 특성을 지나치게 반영하게 되고(overfitting), 반대로 매우 클 경우 모델이 과하게 정규화되는 경향이 있다(underfitting).



2. Implement

Lab : Linear Regression

- ISLR library 안에 내장 데이터인 *Smarket* data 를 이용하여 여러 방법으로 classification 을 해보았다. *Smarket* 은 2001 년 초부터 2005 년 말까지 1, 250 일 동안의 S&P 500 주가 지수의 수익률 데이터이고 1250 개의 관측치와 9 개의 변수를 가지고 있다.

- *Logistic Regression* 은 glm(formula, family=family(link=function), data)을 이용하여 모델을 fitting 시킨다. family 는 종속변수의 분포가 정규분포인 경우 gaussian, 이항분포인 경우 binomial, 포아송분포인 경우 poisson, 역정규분포인 경우 inverse.gaussian, 감마분포인 경우 gamma 를 사용할 수 있다. 생성된 모델로 예측된 확률을 0.5 를 기준으로 class 를 결정한다.

- *LDA* 는 MASS library 에 있는 lda()함수를 사용한다. 사용법은 glm()과 비슷하지만 familiy 를 설정하지 않는다. lda() 는 label 이 output 으로 나오기 때문에 만약 확률을 알고 싶다면, predict()$posterior 을 쓰면 된다. *QDA* 는 qda() 함수를 이용하고 lda()와 사용법이 같다.

- KNN 는 class library 의 knn(train 데이터, test 데이터, cl = train set 의 범주, k 의 수)로 모델을 만든다.

- 전반적으로 fitting 된 모델을 summary()에 넣어서 잔차와 추정값 등을 얻을 수 있다. coef() 함수를 사용하여 모형 인수들의 절편과 기울기 등을 얻을 수 있으며, predict() 함수로 새로운 data 에 대한 예측치를 추정할 수 있다. 모델의 예측력을 확인하기 위해서 classification 은 confusion matrix 를 확인한다. table(예측치,실제값)을 통해 confusion matrix 를 생성하고 예측 결과와 실제 결과가 일치하는 경우와 그렇지 않은 경우를 쉽게 알 수 있다. Accuracy 는 예측값 중 올바른 값의 비율로 sum(predicted == actual) / NROW(actual)과 같이 계산한다.

4. When the number of features *p* is large, there tends to be a deterioration in the performance of KNN and other *local* approaches that perform prediction using only observations that are *near* the test observation for which a prediction must be made. This phenomenon is known as the *curse of dimensionality*, and it ties into the fact that parametric approaches often perform poorly when *p* is large. We will now investigate this curse.

(a) Suppose that we have a set of observations, each with measurements on *p* = 1 feature, *X*. We assume that *X* is uniformly (evenly) distributed on [0, 1]. Associated with each observation is a response value. Suppose that we wish to predict a test observation's response using only observations that are within 10% of the range of *X* closest to that test observation. For instance, in order to predict the response for a test observation with *X* = 0.6, we will use observations in the range [0.55, 0.65]. On average, what fraction of the available observations will we use to make the prediction?

X는 uniformly 분포를 따르므로, 평균적으로 예측을 위해 사용 가능한 관측치는 (0.65-0.55)/(1-0) = 10%이다.

(b) Now suppose that we have a set of observations, each with measurements on *p* = 2 features, *X*1 and *X*2. We assume that (*X*1,*X*2) are uniformly distributed on [0, 1] × [0, 1]. We wish to predict a test observation's response using only observations that are within 10% of the range of *X*1 *and* within 10% of the range of *X*2 closest to that test observation. For instance, in order to predict the response for a test observation with *X*1 = 0.6 and *X*2 = 0.35, we will use observations in the range [0.55, 0.65] for *X*1 and in the range [0.3, 0.4] for *X*2. On average, what fraction of the available observations will we use to make the prediction?

X1과 X2는 독립이고 각각이 10%이기 때문에, 10%×10%=1%이다.

(c) Now suppose that we have a set of observations on *p* = 100 features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations within the 10% of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?

같은 방법으로, 관측치가 100개이므로 평균적으로 예측을 위해 사용 가능한 관측치는 $10\%^{100}$로 엄청 작은 값이다.

(d) Using your answers to parts (a)–(c), argue that a drawback of KNN when *p* is large is that there are very few training observations "near" any given test observation.

데이터 수는 동일하더라도 차원이 증가하면, 즉 변수가 늘어나면 데이터 공간을 채우는 비율(%)이 줄어들기 때문에 변수가 많아질 수록 분석에 요구되는 데이터 건수도 증가한다. 변수의 수가 많아지면 데이터에 대한 정보가 많아지는 것 같지만, 변수의 수가 많아지면 차원이 커지므로 분석을 위한 최소한의 필요 데이터 건수도 많아진다.

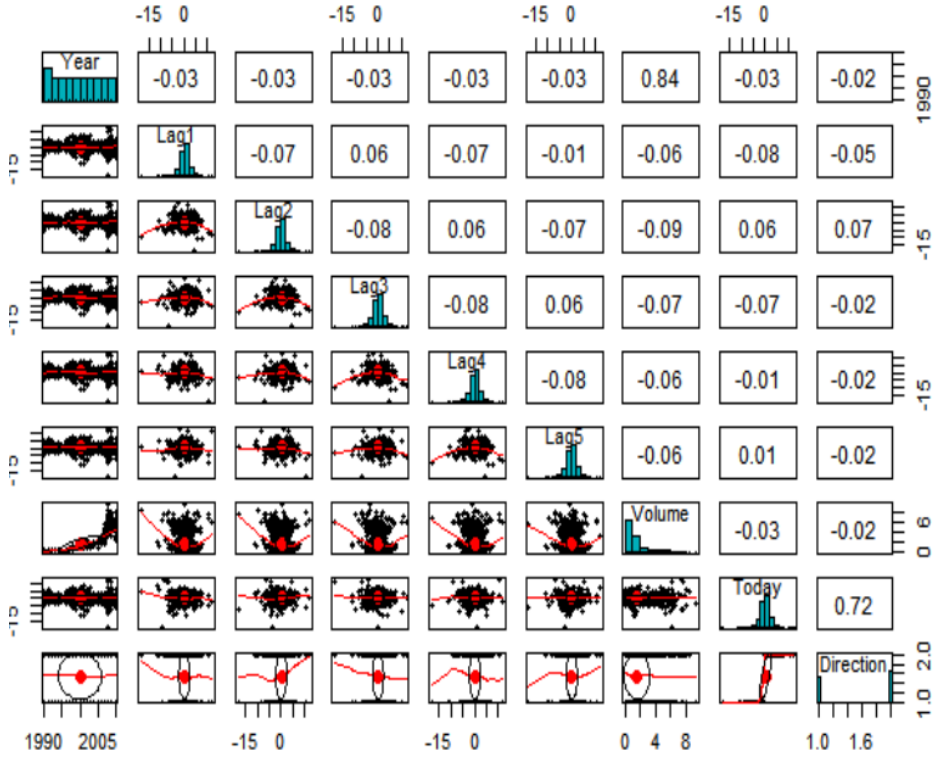
충분히 공간을 표현할 만큼 큰 데이터 수집 없이, 적은 데이터로만 이 공간을 표현하는 경우 과적합(Overfitting)이 발생할 수 있다.

(e) Now suppose that we wish to make a prediction for a test observation by creating a *p*-dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For *p* = 1, 2, and 100, what is the length of each side of the hypercube? Comment on your answer.

p=1일 때, 한 면의 길이는 0.1, p=2일 때 $0.1^{1/2}$=0.3162, p=100일 때 $0.1^{1/00}$=0.9772이다. 이는 피쳐 수가 많은 경우 (즉, p = 100), test 관측치의 평균 10 %를 사용하면 각 피쳐의 거의 모든 범위를 포함해야 한다는 것을 의미한다.

10. This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?



대부분에서 변수들간의 관계가 보이지 않는데, Year과 Volume은 양의 관계가 있음을 볼 수 있다.

(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

p-value를 보면 Lag2가 유의수준 0.01에 대해 유의하다.

lm(Direction ~ Lag1 + ⋯ + Lag5 + Volume, data = Weekly)					
Coefficients :					
	Estimate	Std.Error	t-value	Pr(>| t |)	
(intercept)	0.2668	0.0859	3.106	0.0019	**
Lag1	-0.0412	0.0264	-1.563	0.1181	
Lag2	0.0584	0.0268	2.175	0.0896	*
Lag3	-0.016	0.0266	-0.602	0.5649	
Lag4	-0.0277	0.0264	-1.05	0.2937	
Lag5	-0.0144	0.0263	-0.549	0.5833	
Volume	-0.0227	0.0369	-0.616	0.5377	
AIC : 1500.4					

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

	Down	Up
Down	54	48
Up	430	557

전체 정확도는 56.1%로 Down 으로 예측했을 때는 54/(54+48) = 52.9%, Up 으로 예측했을 때는 557/(430+557)=56.4% 의 정확도를 갖는다.

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

예측된 값과 실제 값의 정확도는 62.5%으로 전체 변수에 대한 logistic model 보다 예측력이 높다.

	Down	Up
Down	9	5
Up	34	56

(e-h) Repeat (d) using LDA, QDA, KNN with *K* = 1. Which of these methods appears to provide the best results on this data?

LDA를 이용하였을 때는 예측력이 62.5%, QDA는 58.6%, KNN은 50%로 LDA에서 예측력이 제일 높았다.

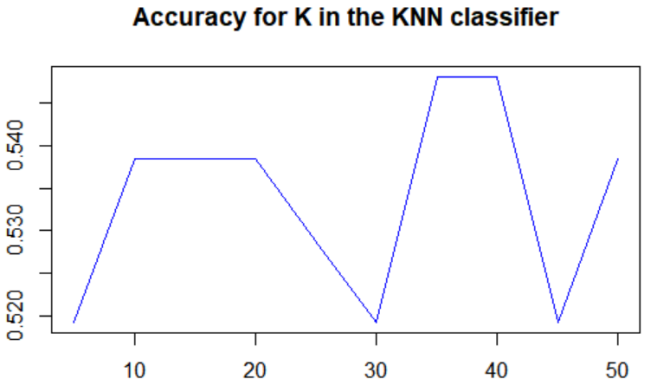
LDA		
	Down	Up
Down	9	5
Up	34	56
Accuracy : 62.5%		

QDA		
	Down	Up
Down	0	0
Up	43	61
Accuracy : 58.6%		

KNN		
	Down	Up
Down	21	30
Up	22	31
Accuracy : 50%		

(i) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for *K* in the KNN classifier.

- knn에서 k값을 변화시키면서 예측력을 비교해보았다. 다음은 k에 따른 accuracy에 대한 그래프이다. K가 35-40일 때, 예측력이 제일 높고 같은 예측력에서는 k가 작을수록 좋으므로 k=35일 때, 최적의 모델을 갖는다. 이때의 정확도는 57.6%이다.



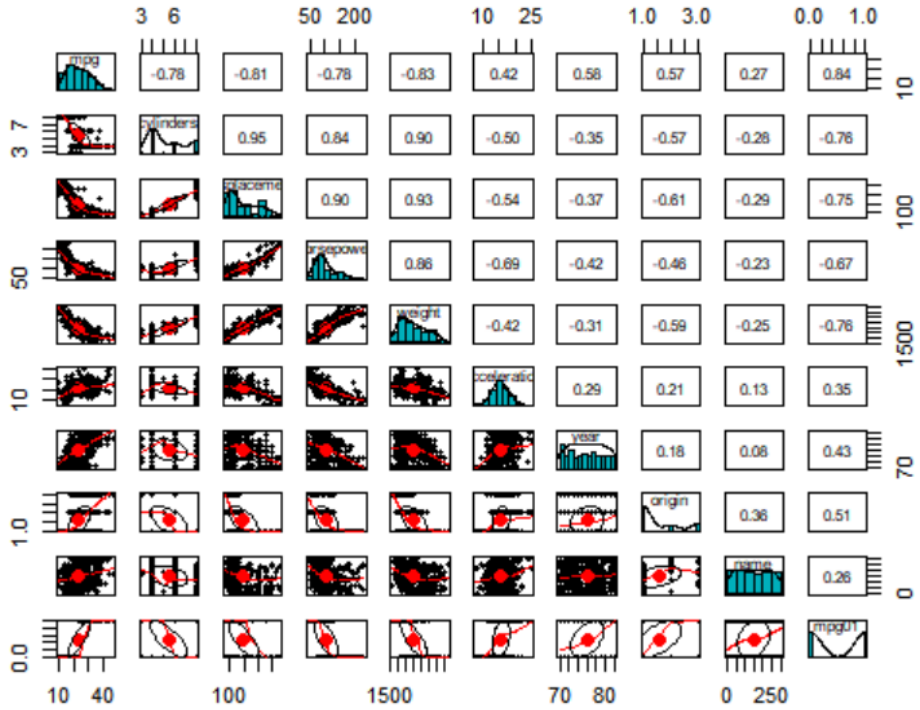
K=35		
	Down	Up
Down	22	23
Up	21	38
Accuracy : 57.6%		

11. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.

(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median() function. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.

Mpg의 median은 22.75로 전체 392개의 관측치를 0/1 각각 196개로 생성되었다.

(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.



`displacement`, `horsepower`, `weight`, `acceleration`이 강한 상관관계가 있는 것 처럼 보인다.

(c) Split the data into a training set and a test set.

train : test = 7:3으로 나누었고 train은 274개, test는 118개의 행으로 나누어졌다.,

(d-f) Perform LDA, QDA, logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

LDA, QDA, logistic regression의 confusion matrix와 오분류율은 다음과 같다. QDA는 두 방법에 비해 오분류율이 높고, LDA와 Logistic Regression 방법의 오분류율은 같다. Confusion matrix를 살펴보면 LDA는 0을 예측할 때 조금 더 좋은 성능을 보이는 반면, Logistic은 1을 예측할 때 조금 정확한 결과를 낸다.

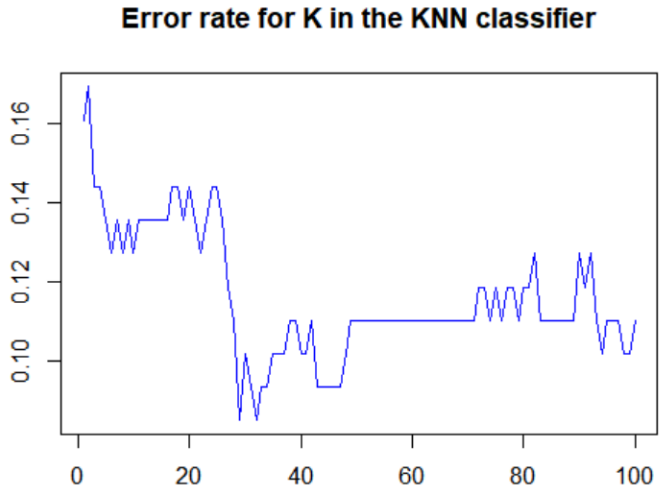
LDA		
	0	1
0	47	0
1	10	61
Error rate : 0.0847		

QDA		
	0	1
0	48	3
1	9	58
Error rate : 0.1016		

Logistic		
	0	1
0	50	3
1	7	58
Error rate : 0.0847		

(g) Perform KNN on the training data, with several values of *K*, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of *K* seems to perform the best on this data set?

k값을 변화시키면서 knn 모델을 학습하였고 그에 따른 error rate plot은 다음과 같다. 20정도까지는 오분류율이 감소하다가 그 이후는 조금 증가에서 거의 일정하게 값이 유지됨을 볼 수 있다. 오분류율이 최소일 때, k값은 29, error rate는 0.0847로 앞의 LDA와 Logistic Regression 오분류율과 같다.



K=29		
	0	1
0	49	2
1	8	59
Error rate : 0.0847		

3. Discussion

▶ 4.4 에서는 k-NN 알고리즘에서 '차원의 저주'에 대한 내용을 다루고 있다. 데이터 수가 동일하더라도 차원이 증가하면 데이터 공간을 채우는 비율이 줄어들기 때문에 변수가 많아질수록 분석에 요구되는 데이터 건수도 증가한다. 충분히 공간을 표현할 만큼 데이터가 크지 않다면 과적합이 발생할 수 있다. 따라서, 고차원의 데이터(예를 들어 10 차원을 넘는 경우)에 대해선 차원의 저주를 피하기 위해 보통 k-NN 알고리즘을 사용하기 전에 차원 축소를 수행한다.

▶ 4.10 에서는 Weekly 데이터를 이용해 Classification 을 해보았다. 전체 변수를 사용하여 logistic regression 을 돌려 유의한 변수로 나온 Lag2 로만 모델을 만들면 예측력이 10%정도 증가하였다. LDA, QDA 에서는 LDA 의 예측력이 좋은 것으로 보아 이 데이터는 변수간의 분산-공분산 행렬이 동일하다는 가정에 더 적합했다고 생각했다. k 값을 변화시켜가며 KNN 알고리즘도 사용하였다. 일반적으로 k 값이 커질수록 분류에서 잡음의 영향이 줄어들지만 항목 간 경계가 불분명해지기 때문에 최적의 k 값을 찾는 것이 관건이다.

▶ 4.11 에서는 Auto 데이터를 이용하여 Classification 을 하였다. 먼저, scatterplots 을 통해 종속변수와 관계가 있어 보이는 피쳐들을 추출하였다. 추출한 변수로 모델에 적용하였고 QDA 의 성능은 떨어지지만 LDA 와 Logistic 의 성능은 동일하였다. confusion matrix 를 통해 LDA 는 0 을 예측을 더 잘하고, logsitic 은 1 에 대해 예측을 더 잘했다. classification 은 전체 accuracy 도 중요하지만 confusion matrix 를 통해 어떤 class 를 잘 분류하는지 파악하는 것이 중요하다는 것을 배웠다.