```
## The following objects are masked from Auto (pos = 3):
##
##     acceleration, cylinders, displacement, horsepower, mpg, name,
##     origin, weight, year
## The following objects are masked from Auto (pos = 4):
##
##     acceleration, cylinders, displacement, horsepower, mpg, name,
##     origin, weight, year
## The following objects are masked from Auto (pos = 5):
##
##     acceleration, cylinders, displacement, horsepower, mpg, name,
##     origin, weight, year
## The following objects are masked from Auto (pos = 6):
##
##     acceleration, cylinders, displacement, horsepower, mpg, name,
##     origin, weight, year
## The following objects are masked from Auto (pos = 7):
##
##     acceleration, cylinders, displacement, horsepower, mpg, name,
##     origin, weight, year
## The following object is masked from package:ggplot2:
##
##     mpg
## The following objects are masked from Auto (pos = 13):
##
##     acceleration, cylinders, displacement, horsepower, mpg, name,
##     origin, weight, year
```

DataMining Assignment

Haoyu Wang (116220323)
Abeer Mohammed H Almabdi (116221065)

April 2, 2017

Contents

1 Introduction

From my point of view it is impossible to give you elevation, Aspect, Slope and etc. to let you predict CoverType in several decades years ago. But in this project we will rise to the challenge even though this technic is normal in the real world. In our project we will focus in "Predicting forest cover type". We all know every kind of plants grow in their suitable environments. That is, if we can collect mass environmental data, such as land types and slopes, we can predict the types of plants by some specific algorithm. By doing this, we can not only manage forests better, but efficiently adopt a different disease prevention method according to different types of vegetation region. Moreover we can generally predict the distribution of phytophagous animals according to these information. In the report we aim to predict forest CoverType using this data and several algorithms and PCA to find a model which can have best performance on prediction which means lowest error rate.

1.1 Data and Features

This data set is the Colorado vegetation type data, the only one that concerns about the real data of forest. Each record contains many indicators to describe each piece of land. For example: height, slope, the distance to the water, the area of the shade, soil type, and so on. Forest type of coverage is needed according to the features of other 54 features to predict. This is an interesting data set, which contains the classification and the numerical characteristics. A total of 581012 records. Each record has 55 column, one of the column is the type of soil, and other 54 column is the input features. Although the data set cannot calculate on the really big data, it can explain a lot of problems. Because this total data-set its too big, for reducing processing time we use two sample size 50000,100000 to test algorithms. And we use sample(c(1:nrow(ds)),50000) to select data to ensure data-set is highly reliable.

2 Cleaning DATA

[6]
In this part we are going to clean the data-set, the following paragraph will show how we do this work:
We get basic idea from Sebastain about how to clean data. Loading data-set as .csv file and then naming columns which can let us easily manage and operate data-set in future works. And then we use R function below to remove any empty or NAs row in this data-set in that the NAs will generate errors when we scale data-set.

```
ds = ds[!apply(is.na(ds) | ds == "", 1, all), ]
```

Reducing number of variables is ours second step, in original data-set we have 40 variables to represent soiltype using binary code. It is an unreadable information to human, so we use SoilType variable instead of all binary variables. We recode soiltype variable based on official (ELU) codes (2072,2073,2074..). And for the same reason we change wilderness variables from 4 to 1. Using integer 1,2,3,4 correspond to four types wilderness area.
Then we use setdiff() to get new set of variables and use ds=ds("New Set of Variables") to get new variables reduced data-set. Finally, use factor() function to covert CoverType and then we rename seven levels of CoverType use tree name (Such as Willow,Pinepr,PineLp). As introduced previously we have already reduce 4 wildnessarea columns to one and 40 soiltype binary variables to 1, but we manually did those.
And in the rest part we will continue using normalized data-set to apply different algorithms.

3 Models & Methods

3.1 Dimensionality reduction Methods

Dimensionality Reduction literally is to transform data of high dimensionality into data of significantly lower dimensionality so that much more information can be conveyed in each of the lower dimensions. This helps to get better features for a classification or regression task when dealing with machine learning problems.
In this part we will apply PCA to data-set. And this technic is necessary in Data Mining project, because there are a lot troubles will occurs when number of independent variable dimensions increasing, such as decreasing the process speed of modelling algorithm, increasing the probability of over fit and difficulty to get a representative sample that causes sparse distribution of samples.

3.1.1 Supervised Learning—LDA

LDA is a supervised Learning Dimensionality reduction method but we will not spend too much time on this feature, because this algorithm is also a classification method and we will use it in the classification section. And there is an interesting problem we want to discuss when we apply PCA before LDA: What will happen? Its effect classification result? Please see Part 6 (Performance Evaluating).

3.1.2 Unsupervised Learning—PCA

With the application of an orthogonal transformation, PCA transfers a collection of observations of probably correlated variables into a collection of principal

components which are linearly uncorrelated variables.
If we can take the set of vectors in that space and then reverse the coordinate system in which they are represented, the new X-axis can capture as much variation as possible. And its convenience in R we can use prcomp() function to do principle component analysis.

```
dsnew1 = scale(dsnew[, c(-11, -12, -13, -14)])
pr.out = prcomp(dsnew1)
```

If applying the prcomp() function, it is unnecessary to multiply the data explicitly by the principal component loading vectors to get the principal component score vectors. Rather the 50000 10 matrix x has as its columns the principal component score vectors. That is, the kth column is the kth principal component score vector.

To compute the proportion of variance explained by each principal component, we simply divide the variance explained by each principal component by the total variance explained by all ten principal components:

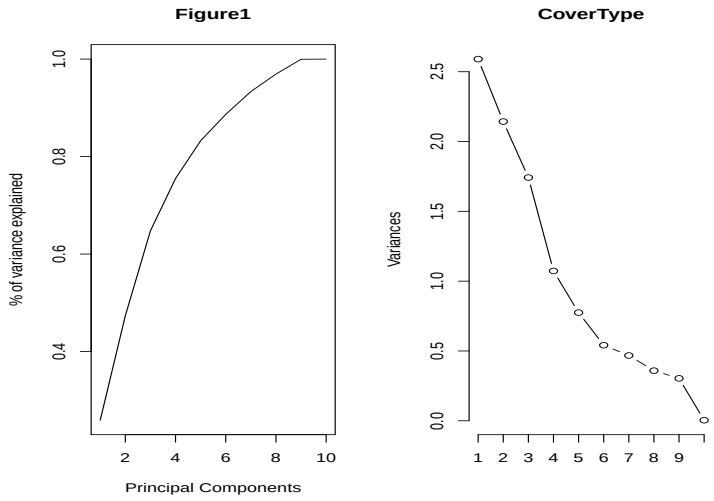
```
## Importance of components:
##                            PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.6095 1.4640 1.3200 1.0360 0.8803 0.73574 0.68380
## Proportion of Variance 0.2591 0.2143 0.1742 0.1073 0.0775 0.05413 0.04676
## Cumulative Proportion  0.2591 0.4734 0.6476 0.7549 0.8324 0.88655 0.93331
##                            PC8     PC9    PC10
## Standard deviation     0.59926 0.55129 0.06259
## Proportion of Variance 0.03591 0.03039 0.00039
## Cumulative Proportion  0.96922 0.99961 1.00000
```

We can see PC1 explaining most variance than rest principle components.Please see Figure 1.

We compared 4 major principal components and then we can see PC1 and PC2 have best performance of this data. Please see Figure 2.

We can clearly see first plot explain most variations then other rest plots.
Under this data-set, the first Five principle components explained over 80% variances, we could pretty much ignore all expect the first five components. In other word the data would be essentially 10-D data, spreading out in 5-D space. Then we will only use first five components as our Dimensionality reduced data-set (we will call it "After PCA") in the later section.

Figure 1: Cumulative proportion plot and Screeplot



3.2 Clustering

As a machine learning task, unsupervised machine learning is able to infer a function to describe hidden structure from unlabeled data whose clustering is not included in the observations.

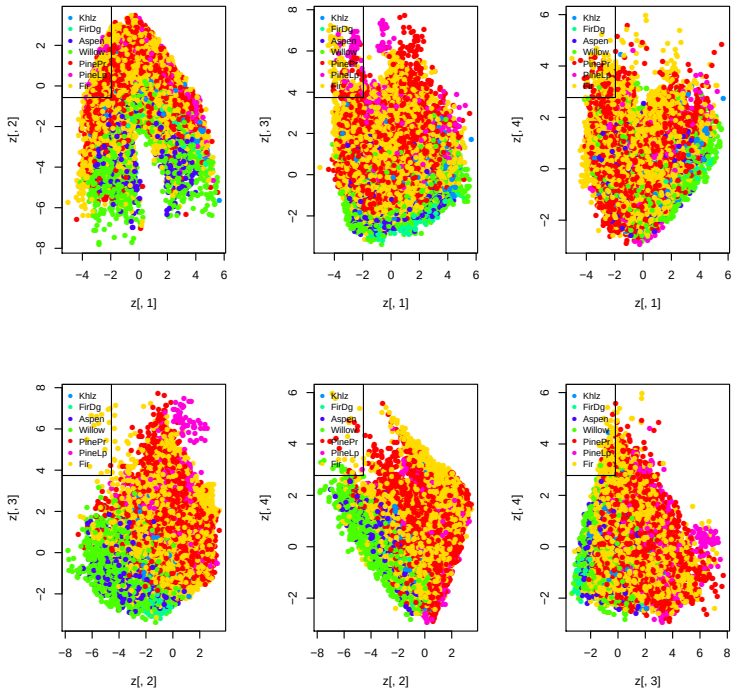
3.2.1 KMEANS

K-means clustering focuses on partitioning n observations into k clusters where each observation is owned by the cluster with the nearest mean, as a prototype of the cluster.
Simply to say K-means algorithm will random generate k points and cluster observation through distance, we applied a k means function in R to achieve it. In this part we will apply k means function on two dataset which one is original and other is reduced variables dataset. And then see any improvement of performance in results.
we can see the correct rate of clustering below:

```
table(y, o$cluster)

##
## y      1    2    3    4    5    6    7
##   1 3783 4967 1258 3737 2013 1589  851
##   2 3030 4948 3036 7031 2203 2226 1859
##   3    0  214    0  930   70  854 1054
##   4    0   15    0   99    2   84   20
```

Figure 2: Pairwise scatterplots of principle components 1-4



```
##   5   16  142   31  316   49  253   30
##   6    0  132    0  377   10  381  633
##   7  502  384   29  148  467  193   34
```

```
## [1] 0.18588
```

When Sample size incresed to 100000, the correct rate decreased to **0.10339**. After PCA among which we chose first 5 components, correct rate decreased from **0.18588** to **0.11598** when sample size was 50000. And when we changed sample size to 100000, correct rate only slightly decreased from **0.10339** to **0.09929**.

From the R output we know the rate of success cluster is very low even increase sample size from 50000 to 100000. And when we processing data after PCA, rate will lower then previous. We can know when we apply PCA to data will decrease variables and it will affect the clustering result.

3.3 Classification

Classification is a data mining technique used to predict group membership for data instances.
As a machine learning task, supervised machine learning is able to infer a function to describe hidden structure from labeled data whose classification is not included in the observations.
In this part we will use K-NearestNeighbor(KNN), Linear discriminant analysis(LDA) and Quadratic discriminant analysis(QDA) to classify the data and compare the performance of those algorithm.

3.3.1 LDA And QDA

We already know some simple examples in An Introduction to Statistical Learning with R [5], "Fisher's Discriminant Analysis" [6] is simply LDA in a 2 classes situation. LDA need requires an assumption of equal variance-covariance matrices of classes. And invention of QDA just in order to overcome this problem. QDA is mutation of LDA, allowing the heterogeneous class of covariance matrix Now we will apply it on 10 variables (except factor variables) original and After PCA data-set. We use lda() function which option is CV=TRUE(CrossValidation) to process data in R and see the difference between QDA and LDA.

```
out1 = lda(y ~ ., X[, c(-11, -12, -13, -14)], CV = TRUE)
```

```
##
## y          Fir PineLp PinePr Willow Aspen FirDg  Khlz
##   Fir    13089   4934      0      0     1     1   186
##   PineLp  5089  18646    373      1    27   212     3
##   PinePr     0    614   2001     53     0   426     0
##   Willow     0      2    201     20     0     6     0
##   Aspen      1    842      1      0     4     0     0
##   FirDg      0    455    712      0     0   372     0
##   Khlz    1506     16      0      0     0     0   206
## [1] 0.68676
##
## y          Fir PineLp PinePr Willow Aspen FirDg  Khlz
##   Fir    14195   3429     22      0    60    44   461
##   PineLp  7613  14607    771      3   373   921    60
##   PinePr     0    405   1766     78     1   843     1
##   Willow     0      2     48    142     0    37     0
##   Aspen     46    560     23      0   193    22     4
##   FirDg      0    237    399     30     3   870     0
##   Khlz    1188     35      0      0     0     0   505
## [1] 0.6455987
```

Sample size increased to 100000, we can see correct rate of LDA only have small increment from **0.68676** to **0.68115**.And around 2% increment in QDA.

After PCA:
We used first five components, and LDA performed not well, decreasing around 8% to 0.5968 and QDA decreasing only 3% to 0.650653. When sample size increase to 100000, correct rate of LDA decrease from **0.68847** to **0.6** and QDA keep stable around 0.6596.

3.3.2 KNN

In KNN classification we need to find a value for k. We should choose it to minimize predication error, and to measure prediction error we need a loss function. A function which takes as input the truth and the prediction and returns a value that is large when the prediction is far from the truth and 0 when it matches the truth.[4] We used KNN classification in three ways which use two data-set, one of which is original and the other one is processed by PCA:

3.3.3 KNN Manually

Choose training-set Manually which means we choose first 35000 rows data without shuffling be training data:

```
##         te
## pre       Fir PineLp PinePr Willow Aspen FirDg Khlz
##   Fir    4586    754      1      0    10     1  109
##   PineLp  852   6352     33      0   102    57   15
##   PinePr    0     65    868     12    10    67    0
##   Willow    0      0     12     40     0     4    0
##   Aspen    10     44      1      0   129     1    0
##   FirDg     1     32     57     13     2   311    0
##   Khlz     48      7      0      0     0     0  394
## [1] 0.8453333
```

When sample size increased to 100000,we can see correct increased form **0.845333** to **0.88757**.

After PCA:

Similairly, we chose five components of data-set, the correct rate have huge decrement around 15% when sample size is 50000. when we use 100000 rows data, it decreased from **0.88757** to **0.7212**.

3.3.4 KNN Random select 70% Training data

We choose 70% data from dataset randomly become Training data rather then manually selected. The reason why we do not Manual Choosing is this method will occur many problems when data is sparse or dense. In that situation sampling would not reliable anymore.

```
##         te
## pre       Fir PineLp PinePr Willow Aspen FirDg Khlz
##   Fir    4566    659      1      0    20     3   85
##   PineLp  894   6401     48      1   104    65   26
##   PinePr    1     73    820     26     4    95    0
##   Willow    0      0      6     34     0     4    0
##   Aspen     5     51      1      0   119     0    0
##   FirDg     4     34     59      8     2   300    0
##   Khlz     56      8      0      0     0     0  417
## [1] 0.8438
```

When sample size increased to 100000: correct rate increase to **0.8885**.

After PCA:

The correct rate of sample size 50000 and 100000 all have large decrement which decreased to 0.6902 and 0.72193 respectively.

3.3.5 KNN CrossValidation

We already applied two different training data selection methods, the third method is K-folder CrossValidation. The reason why we use K-folder rather

than LOOCV is slowly processing speed when we applied this algorithm in our dataset.
c K-folder CV will partitioning Data set into complementary subsets, we will perform the analysis on one subset and validate the analysis on the other set. Using K-folder CV has benefits to reduce changeability, for example many rounds of cross-validation are predicted by using different partitions, and the final results are averaged over the rounds.
We find when folders equal to 10 have best performance from [7], so set v(folders number)=10,and random set k=7, the success rate will show below:

```
##
## wh         Fir PineLp PinePr Willow Aspen FirDg  Khlz
##   Fir    15352   2206      4      0    51     4   306
##   PineLp  2641  21645    170      1   300   192    41
##   PinePr     1    194   2761     69    28   280     0
##   Willow     0      0     27    108     0    14     0
##   Aspen     20    130      3      0   453     4     0
##   FirDg      9    137    157     42     5  1039     0
##   Khlz     175     21      0      0     0     0  1410
## [1] 0.85536
```

When sample size increased to 100000:we can see same trend here, correct rate increased from **0.85536** to **0.89769**. After PCA:
When we only apply 5 principle components, correct rate all have great decrements in both sample, first one decreased to **0.69636** and second one decreased to **0.72463**.

4 Result And Discussion

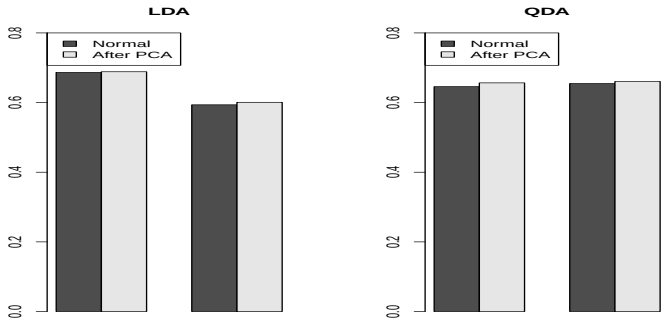
4.1 The result when using LDA Classify processed Data-set by PCA

From Figure 3, there have an interesting situation that we can see only slight impact on LDA and nearly no impace on QDA and it does not happened in other models. We suggest the main reason is that those algorithms are using techniques to reduce dimensions to process the data so it result in similar correct rate in both data-set.

4.2 Relationship between K, Sample size and success rate

Until now, we can see sample size have strong relationship with success rate in classification algorithms. Please see Figure 4. And we also make an assumption

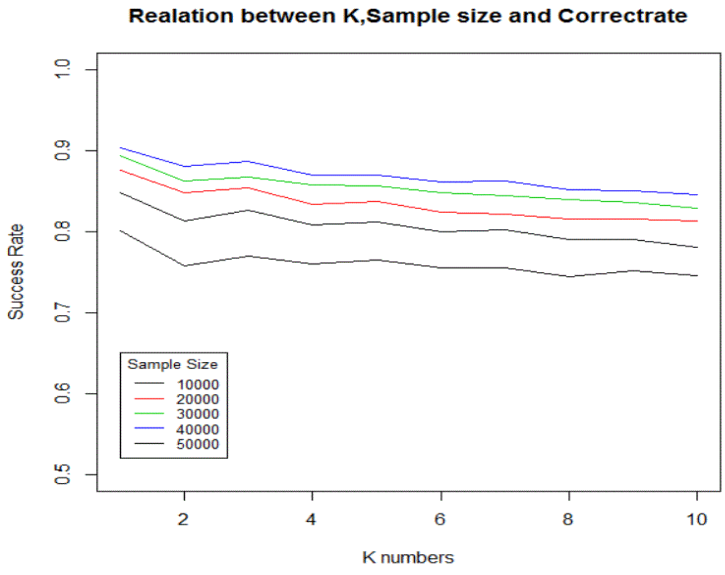
Figure 3: LDA & QDA performance within and without PCA



of K numbers that the success rate will be decreasing with increasing of K.
Then we are assume to choose the best performance model (KNN-CV) using 5 different sample size and 1:10 k numbers to test the relationship between K ,success rate and sample size.
Please see plot below:

We can see when sample size is increasing the success rate also increasing.

Figure 4: Relationship between K,sample size and correct rate



It means the if we want high success rate, one possible way is using huge data set. From this plot, we conclude classification correct rate is negative linear

relationship with K numbers.

4.3 Performance between different methods

Compare the performances of different algorithms and show below:

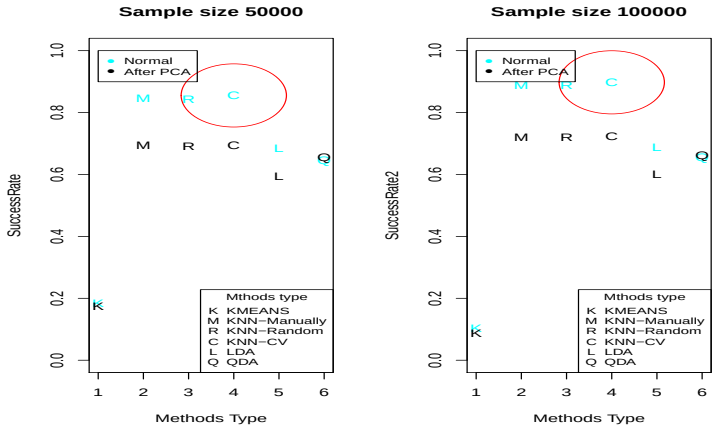
	Kmeans	KNN-Manually	KNN-Random	KNN-CV	LDA	QDA
Normal	0.185	0.84	0.845	0.86	0.6867	0.6456
After PCA	0.17528	0.696	0.78	0.696	0.5968	0.6503

Table 1: Correct Rate (sample size 50000)

	Kmeans	KNN-Manually	KNN-Random	KNN-CV	LDA	QDA
Normal	0.1034	0.888	0.6902	0.897	0.6867	0.68115
After PCA	0.09929	0.7212	0.7219	0.724	0.6	0.6596

Table 2: Correct Rate(sample size 100000)

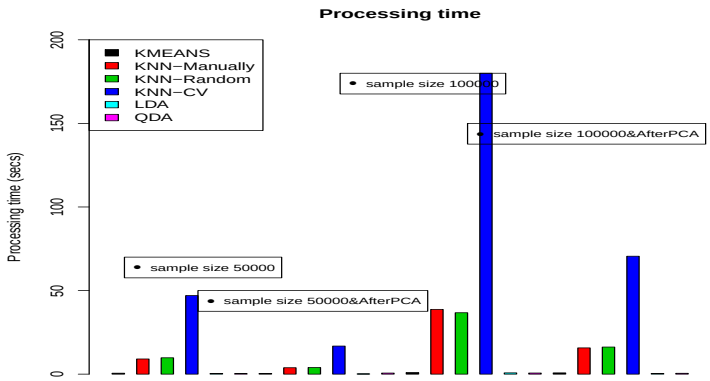
Figure 5: Success Rate between 6 Models



The aim of this project is to predict covertype of forest based on cartographical data. We can clearly see from Figure 5, Table 1 and Table 2 that our KNN-CV classifier and KNN-Random algorithms performed quite well and correct rate is 90% and 87% respectively on both training and LDA have a good performance around 69%. While clustering algorithm K-means is the lowest around 10And now we want to see is KNN-CV also the fastest processing speed algorithm. Then we use a function in R (Sys.time()) to get the running time. Please see

below:

Figure 6: Processing time comparsion



The Figure 6 gives information about algorithms processing time in different sample size. We can see the major reason that affects running time is sample size, processing time increasing with sample size increasing. And also we can see a majority of algorithms which process data-set after PCA will be faster than the normal data-set does, however it only has little effect on LDA and QDA and the time of process after PCA Data-set is much more than normal data-set when the sample size is 50000.

5 Conculsion

Our three ways KNN performed very well and correct rates are over 80% and increase with samplesize,LDA & QDA performed ordinarily in the behind aroud 70%, but accurate rate of k means is unexpectedly lower, with only 20%.
Finally, through comparison between all methods we conclude the best model is KNN-CV(CrossValidation),and there are strong relationship between accurate rate and sample size, and we can see correct rate of three KNN methods are very close, we conculde random choose and cross-validation have similar result if sample size is enough big. We also learn the importance of preprocess data-set by Dimensionality reduction method (PCA) in reducing processing time. In this pratical also has a abnormality, there are slight impact on LDA and no impact on QDA before and after PCA. We conclude KNN (CrossValidation) is the best model in this project.

References

[1] Phyu, T. N. (2009, March). Survey of classification techniques in data mining. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1, pp. 18-20).¡8f¿

[2] https://archive.ics.uci.edu/ml/datasets/Covertype

[3] R Core Team, R:¡9d¿ A Language and Environment for Statistical Computingd¿, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/ . Accessed May 5th, 2014.

[4] http://www.stat.berkeley.edu/ nolan/stat133/Fall04/lectures/CV.pdf

[5] James, G.,Written,D.,Hastie,T.,&Tibshirani,R.(2013) An Introduction to Statistical Learning with R

[6] Sebastian Labs idea

[7] https://www.quora.com/For-K-fold-cross-validation-what-k-should-be-selected