Final Project

Jerry Chan

Introduction

In this project, we are going to inspect the Swiss Banknote Dataset[1]. The dataset contains genuine and counterfeit old Swiss bank notes. We want to classify those notes by some measurements of the notes. I perform LDA and Logistic regression to accomplish this task and use PCA and factor analysis to improve my model's performance In this report, I discuss about the assumptions of each models, implement the model, and explain the result.

Data Description

The dataset contains 100 genuine and 100 counterfeit old Swiss 1000-franc bank notes. The first 100 samples are genuine and the other 100 are counterfeit. Each observation has six features as described below:

Metadata [2]

1. **Length**: Double, Length of the note
2. **Left** Double, Width of the Left-Hand side of the note
3. **Right**: Double, Width of the Right-Hand side of the note
4. **Bottom**: Double, Width of the Bottom Margin
5. **Top**: Double, Width of the Top Margin
6. **Diagonal**: Double, Diagonal Length of Printed Area

Import Package

```
library(GGally)
library(caret)
library(Rfast)
library(e1071)
library(factoextra)
```

Load Data

```
df <- read.csv('SBN.txt', header = TRUE, sep="")
head(df)
```

	Length <dbl>	Left <dbl>	Right <dbl>	Bottom <dbl>	Top <dbl>	Diagonal <dbl>
BN1	214.8	131.0	131.1	9.0	9.7	141.0
BN2	214.6	129.7	129.7	8.1	9.5	141.7
BN3	214.8	129.7	129.7	8.7	9.6	142.2
BN4	214.8	129.7	129.6	7.5	10.4	142.0
BN5	215.0	129.6	129.7	10.4	7.7	141.8
BN6	215.7	130.8	130.5	9.0	10.1	141.4

6 rows

Body

1. Explore and Visualize the data

First, I calculate each column's mean and variance.

```
colMeans(df)
```

```
##   Length     Left    Right   Bottom      Top Diagonal
## 214.8960 130.1215 129.9565   9.4175  10.6505 140.4835
```

```
colVars(as.matrix(df))
```

```
## [1] 0.1417930 0.1303394 0.1632741 2.0868781 0.6447234 1.3277163
```

Next, I append a label column to the dataframe. The label is TRUE if the note is genuine. It's FALSE otherwise.

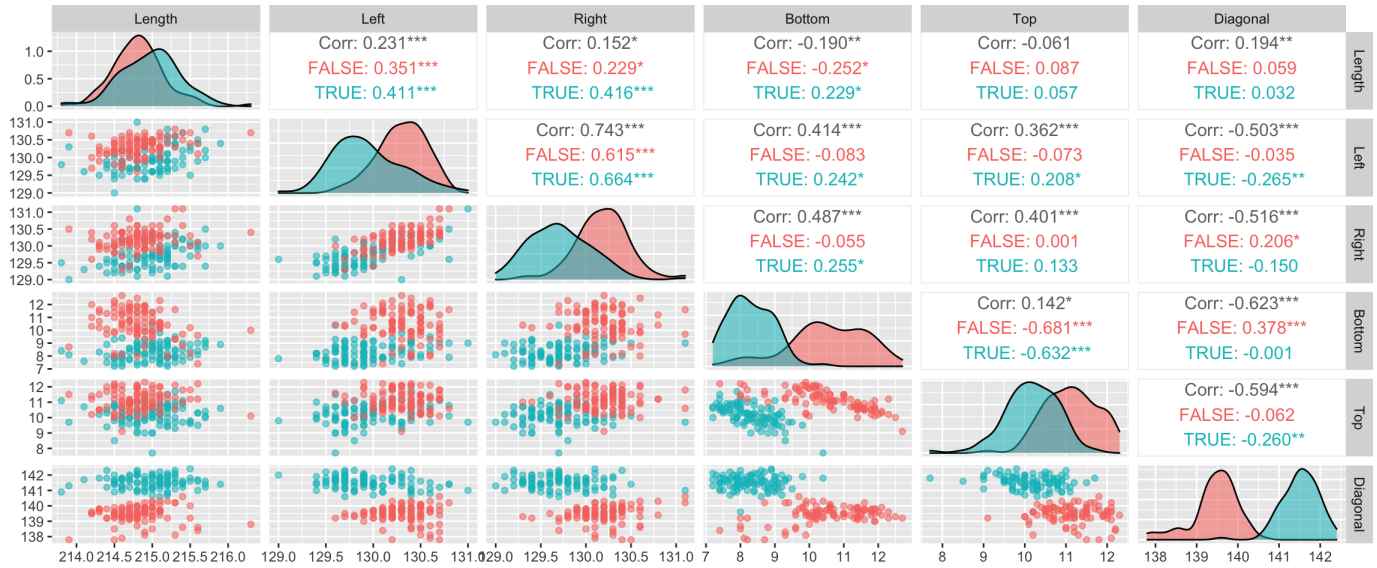
```
df$label <- c(1:200) <= 100
head(df)
```

	Length <dbl>	Left <dbl>	Right <dbl>	Bottom <dbl>	Top <dbl>	Diagonal <dbl>	label <lgl>
BN1	214.8	131.0	131.1	9.0	9.7	141.0	TRUE
BN2	214.6	129.7	129.7	8.1	9.5	141.7	TRUE
BN3	214.8	129.7	129.7	8.7	9.6	142.2	TRUE
BN4	214.8	129.7	129.6	7.5	10.4	142.0	TRUE
BN5	215.0	129.6	129.7	10.4	7.7	141.8	TRUE
BN6	215.7	130.8	130.5	9.0	10.1	141.4	TRUE

6 rows

With the label appended to each row, I can now plot the pairwise scatter plots to exam each variables relation with other variables as well as the distribution of each variable.

```
ggpairs(df, aes(color = label, alpha = 0.5), columns = 1:6, )
```



Here is my observation from the graph above:

1. It's pretty easy to classify genuine and counterfeit notes since the distribution of two class is very different on some features. For example, genuine notes have significant larger diagonal length. We can almost make a clean separation with the diagonal length alone.
2. Some variables are highly related with others. For example, "Left" and "Right" seems to linearly dependent to each other and has a high corrilation of 0.74. Pairs like ("Diagonal", "Top"), ("Top", "Right"), and ("Bottom", "Right") all have very high correlations.

2. K-fold Cross Validation

In this section, I'll implement some functions for k-fold cross validation to test the performance of my models for in the later sections. I choose to split the data into five fold. I set the random seed to 42 before each run to ensure the data is shuffled the same way each time. During cross validation, the dataset will be shuffled and split in to 5 non-overlapping subsets. The model will be trained on four of them and tested on the holdout set. Five model will be trained in total, each with a different set as the holdout set. The goal of cross validation is to prevent a easier or harder test set effecting the test result, making the validation step more robust.

Functions for k fold cross validation:

```
cv <- trainControl(method = "cv", number = 5)
```

```
validate <- function(df, mdl){
  set.seed(42)
  return (train(label~.,data = df, trControl = cv, method = mdl, maxit = 100, family
 = binomial()))
}
```

(The maxit and family arguments are for the linear regression model)

A table to record the performance of the models:

```
result <- data.frame(fold = 1:5)
```

2. LDA and Logistic Regression

In this section, I'll first discuss the assumption of both models. Then, I'll implement those models and fit it with our dataset. Finally, I'll compare the result of both model.

1. Assuptions:

Assuptions for LDA[3]:

1. Multivariate normality: Independent variables are normal for each level of the grouping variable.

   From the previos section we can see that features such as “Bottom” and “Diagonal” don’t follows a normal distribution.

2. Homoscedasticity: Variances among group variables are the same across levels of predictors.

   The variances among variables are not the same.

3. Multicollinearity: Predictive power can decrease with an increased correlation between predictor variables.

   There is high correlation between some variable, such as “Right” and “Left”.

4. Independence: Participants are assumed to be randomly sampled, and a participant’s score on one variable is assumed to be independent of scores on that variable for all other participants.

   Likely to be true as each sample are separate banknotes.

LDA has very strong assumptions that are not satisfied by our dataset

Assuptions for Logistic Regression:

1. The observations are independent of each other

   Likely to be true as each sample are separate banknotes.

2. Little or no multicollinearity among the independent variables.

   Not fully satisfied as discussed in the previous part.

Logistic regression’s assumptions are a lot weaker than the assumptions for LDA.

2. LDA

Following is the implementation of LDA model and the test result:

```
df$label <- as.factor(df$label)
lda <- validate(df, "lda")
result$lda <- lda$resample$Accuracy
lda$resample$Accuracy
```

```
## [1] 1.000 1.000 0.975 1.000 1.000
```

3. Logistic Regression

Following is the implementation of Logistic regression model and the test result:

```
lr <- validate(df, "glm")
result$lr <- lr$resample$Accuracy
lr$resample$Accuracy
```

```
## [1] 1.000 1.000 0.975 0.975 0.950
```

4. Discussion

```
result
```

fold <int>	lda <dbl>	lr <dbl>
1	1.000	1.000
2	1.000	1.000
3	0.975	0.975
4	1.000	0.975
5	1.000	0.950

5 rows

```
colMeans(result)
```

```
##  fold   lda    lr
## 3.000 0.995 0.980
```

We can see that the LDA perform better than Linear Regression. LDA has a average accuracy of 99.5% and logistic regression has a average accuracy of 98%. There are less than 5 misclassification for both models. Consider that we have a really small dataset, the difference in their performance is not very significant.

3. Dimension Reduction

Now I am going apply dimension reduction methods to the dataset and see if this effects the models performance. Since there are some dependency among variables as I addressed in the visualization section, the dimension reduction method should decorrelate those variables while maintaining the accuracy.

1. Assumptions

Assumptions for PCA[4]:

1. Variables are continuous.

   True for our dataset.

2. Linear relationship between all variables.

   Not satisfied. However there are linear relationship between some variables, such as “Right” and “Left”, as presented in the pair-wise scatterplot.

3. Sampling adaquacy, which means that we should have a relatively large sample size.

   Not satisfied. The dataset is relatively small.

4. Suitable for dimension reduction.

   As discussed earlier, some variables are highly correlated to other variables. Our dataset is suitable for dimension reduction.

5. No significant outliers.

   From the pair-wise scatterplot, there is no significant outlier.

Assumptions for MLE[5]:

The data are independently sampled from a multivariate normal distribution with mean vector $\underline{\mu}$ and variance-covariance matrix $\mathbf{\Sigma}$.

This is a very strong assumption. Our dataset fails to satisfy it. From the histogram in the visualization section, we can see that “Diagonal” and “Bottom” don’t follow a bell-shaped curve. They are unlikely to be samples from any normal distribution.

2. PCA

I scale the dataset and calculate the principal components.

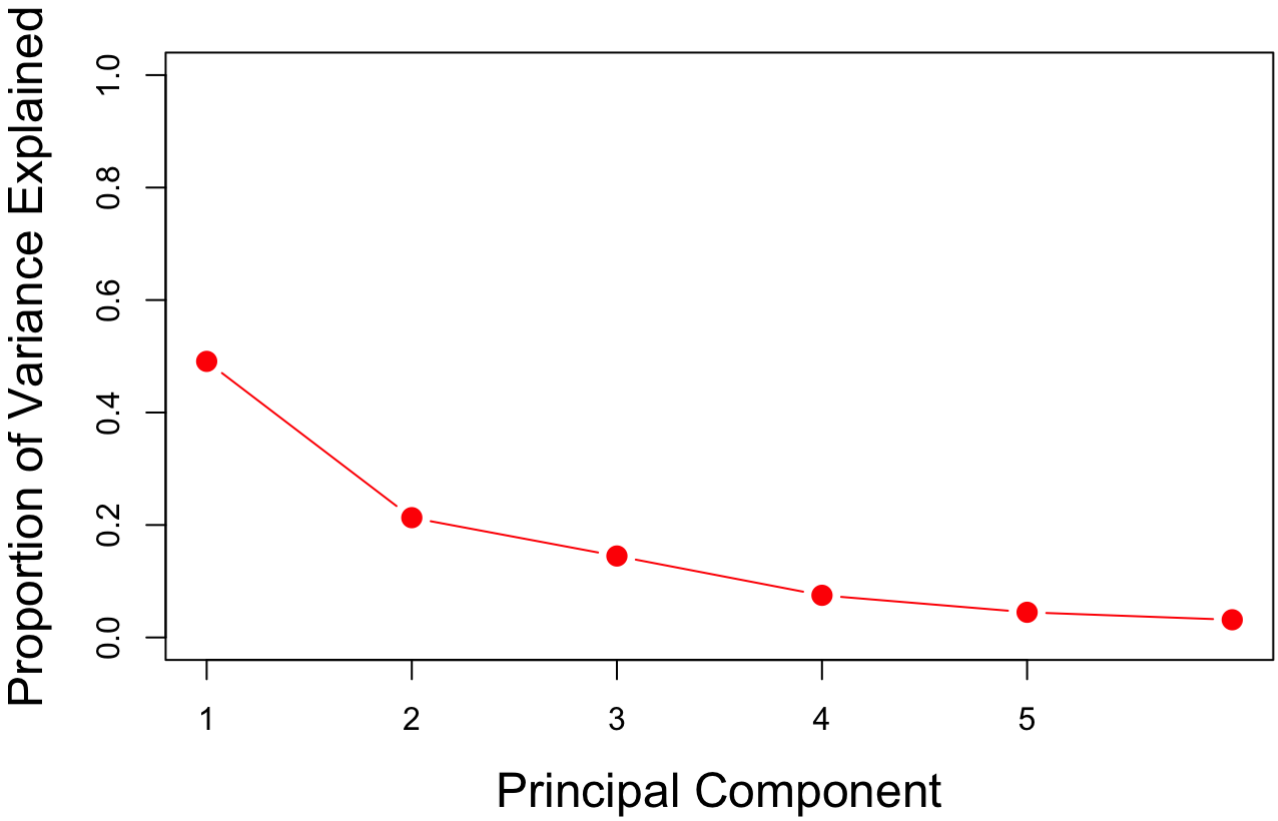
```
scaled <- scale(df[,1:6])
pca <- prcomp(scaled)
pca
```

```
## Standard deviations (1, .., p=6):
## [1] 1.7162629 1.1305237 0.9322192 0.6706480 0.5183405 0.4346031
##
## Rotation (n x k) = (6 x 6):
##                   PC1         PC2          PC3        PC4         PC5          PC6
## Length    0.006987029 -0.81549497  0.01768066  0.5746173 -0.0587961  0.03105698
## Left     -0.467758161 -0.34196711 -0.10338286 -0.3949225  0.6394961 -0.29774768
## Right    -0.486678705 -0.25245860 -0.12347472 -0.4302783 -0.6140972  0.34915294
## Bottom   -0.406758327  0.26622878 -0.58353831  0.4036735 -0.2154756 -0.46235361
## Top      -0.367891118  0.09148667  0.78757147  0.1102267 -0.2198494 -0.41896754
## Diagonal  0.493458317 -0.27394074 -0.11387536 -0.3919305 -0.3401601 -0.63179849
```

Then, I plot out the scree plot to exam how each principal component explained the variance of our dataset.

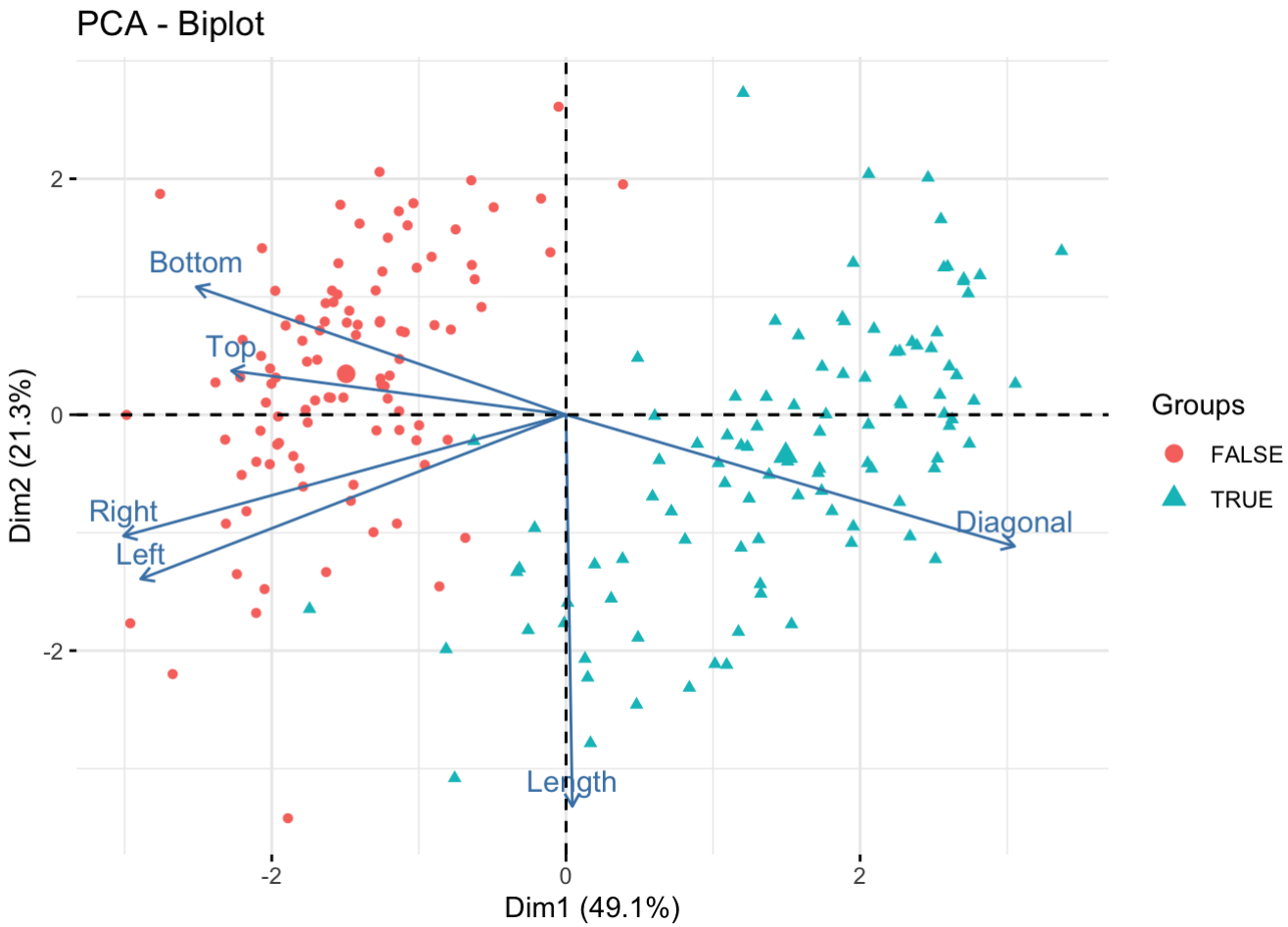
```
pca_var <- pca$sdev^2
pve <- pca_var/sum(pca_var)

plot(pve, xlab=" Principal Component ",
  ylab=" Proportion of Variance Explained ",
  ylim=c(0,1), xaxt="n" ,type='b', col="red",
  cex=2,pch=20, cex.lab=1.5)
axis(1, at=c(1,2,3,4,5),labels=c(1,2,3,4,5))
```



The first two principal components are able to explain 70% of the variance. Using the first two component alone gives us a lot of information about the data. I plot out the biplot to see what the first two principal present and exam if the data is separable after being projected to the latent space.

```
fviz_pca_biplot(pca,habillage = df$label, label = "var")
```



The first principal component (Dim1) mostly represent “Length”. The second principal component(Dim2) focuses on “Diagonal”, “Bottom”, “Top”, “Right”, and “Left”. Since the correlations between “Diagonal” and the other four variables are all negative, the arrow of “Diagonal” is pointing to the positive side of Dim2 while the rest points at the other direction. The biplot shows that most of the data is separable on 2 dimensional latent space, so I decided to use only 2 principal components.

Now, I’ll train both LDA model and linear regression model. Discussion of the result will be on the Result section.

```
pca_x <- as.data.frame(pca$x[,1:2])
pca_x$label <- df[,7]
pca_lda <- validate(pca_x,"lda")
result$pca_lda <- pca_lda$resample$Accuracy
pca_lda$resample$Accuracy
```

```
## [1] 1.000 1.000 0.950 0.975 1.000
```

```
pca_x <- as.data.frame(pca$x[,1:2])
pca_x$label <- df[,7]
pca_lr <- validate(pca_x,"glm")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
result$pca_lr <- pca_lr$resample$Accuracy
pca_lr$resample$Accuracy
```

```
## [1] 1.000 1.000 0.950 0.975 1.000
```

3. MLE

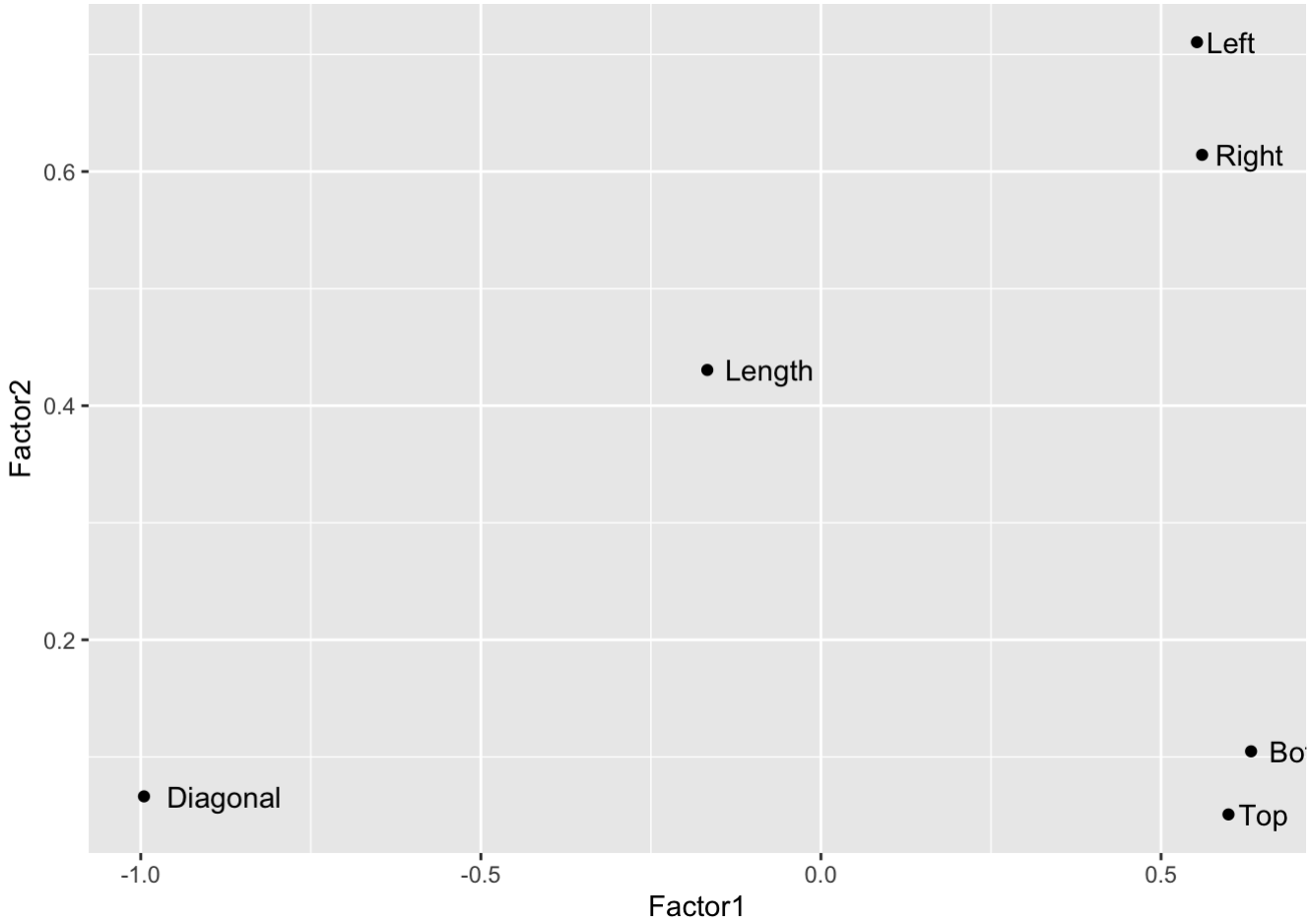
Here, I'll train a MLE model with 2 factors. I choose to use only 2 factors to match the number of principal component I used in the PCA part, so I can compare the performance of the two model on same extent of dimension reduction.

```
MLE <- factanal(factors = 2, x = scaled, score = "regression")
MLE
```

```
##
## Call:
## factanal(x = scaled, factors = 2, scores = "regression")
##
## Uniquenesses:
##   Length     Left    Right   Bottom      Top Diagonal
##    0.787    0.190    0.309    0.589    0.638    0.005
##
## Loadings:
##          Factor1 Factor2
## Length   -0.167   0.431
## Left      0.553   0.711
## Right     0.560   0.614
## Bottom    0.632   0.105
## Top       0.599
## Diagonal -0.995
##
##                Factor1 Factor2
## SS loadings      2.397   1.085
## Proportion Var   0.399   0.181
## Cumulative Var   0.399   0.580
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 51.41 on 4 degrees of freedom.
## The p-value is 1.83e-10
```

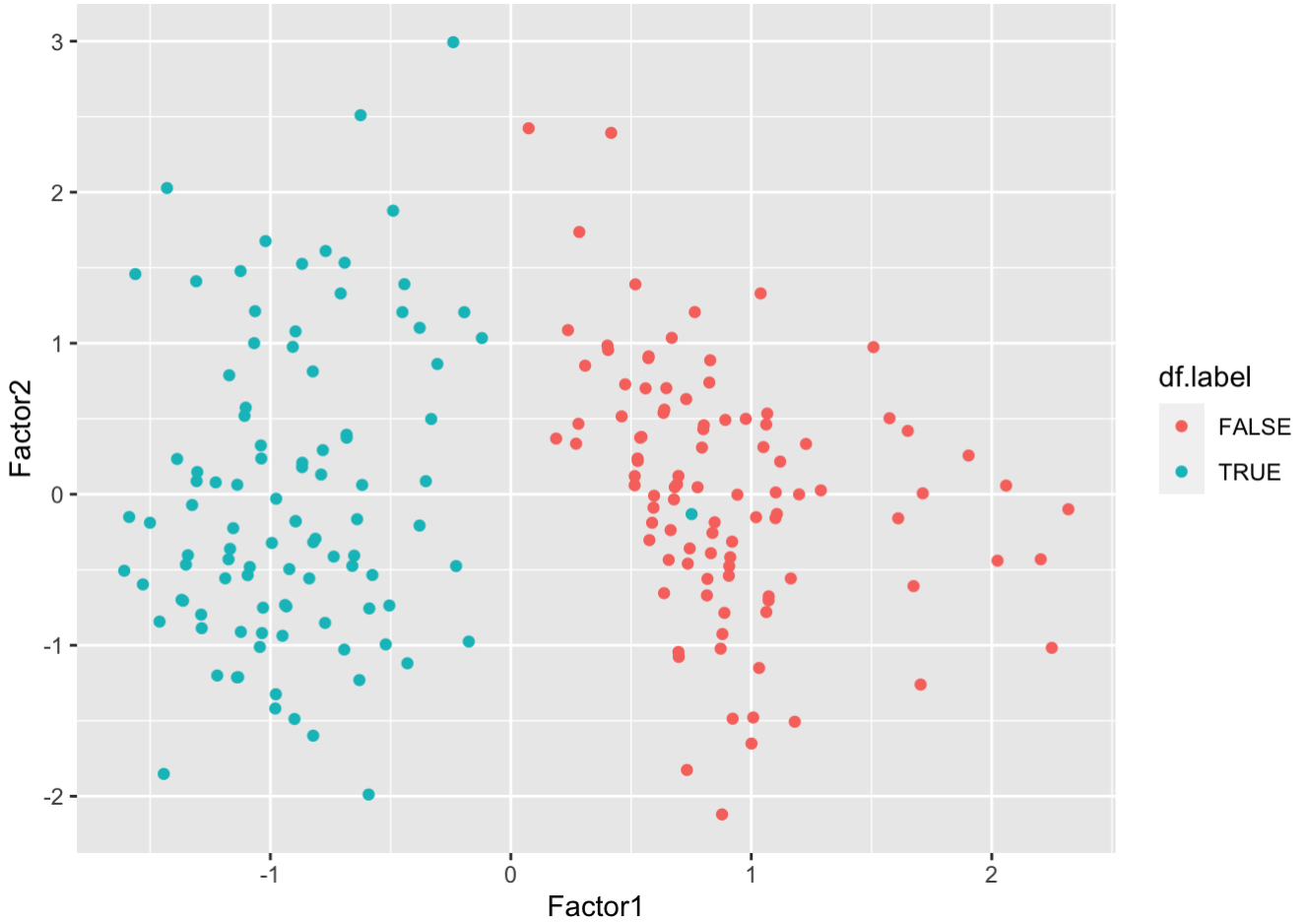
The two factors are able to remain 58% of the variance. I'll plot out the biplot to exam the quality of the reduced data and explain what each factor represents.

```
load <- MLE$loadings
ggplot(data.frame(load[,1:2]),aes(x=Factor1, y = Factor2)) + geom_point() + geom_text
(label = names(df)[1:6], hjust = -0.2)
```



Factor1 focuses on “Diagonal” and variables that is negatively correlated with it. Factor2 describe “Left”, “Right”, and “Length”.

```
ggplot(data.frame(MLE$scores, df$label), aes(x = Factor1, y=Factor2, color=df.label))
+ geom_point()
```



From the biplot, we can see that the two class are clearly separable. Moreover, they seems to be separable by only using Factor1 with only one misclassification. However, to compare it with PCA, I'll fit scores from both factor to LDA and Logistic Regression model. The result will be discussed in the next section.

```
mle_x <- as.data.frame(MLE$scores)
mle_x$label <- df[,7]
mle_lda <- validate(mle_x,"lda")
result$mle_lda <- mle_lda$resample$Accuracy
mle_lda$resample$Accuracy
```

```
## [1] 1.000 1.000 0.975 1.000 1.000
```

```
mle_x <- as.data.frame(MLE$scores)
mle_x$label <- df[,7]
mle_lr <- validate(mle_x,"glm")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
result$mle_lr <- mle_lr$resample$Accuracy
mle_lr$resample$Accuracy
```

```
## [1] 1.000 1.000 0.975 1.000 1.000
```

4. Result

Accuracy for each models:

```
result[6,] = colMeans(result)
result[6,1] = "mean"
result
```

	fold	lda	lr	pca_lda	pca_lr	mle_lda	mle_lr
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	1.000	1.000	1.000	1.000	1.000	1.000
2	2	1.000	1.000	1.000	1.000	1.000	1.000
3	3	0.975	0.975	0.950	0.950	0.975	0.975
4	4	1.000	0.975	0.975	0.975	1.000	1.000
5	5	1.000	0.950	1.000	1.000	1.000	1.000
6	mean	0.995	0.980	0.985	0.985	0.995	0.995

6 rows

1. LDA vs Logistic Regression:

Both model achieve a really high accuracy (0.995 and 0.980). The LDA model performs slightly better than the Logistic Regression model. It make only 1 misclassification out of 200 samples.

2. PCA vs MLE:

LDA model's accuracy drops from 0.995 to 0.985 after we reduce the input dimension with PCA. However, Logistic Regression model's accuracy increases by 0.005 with PCA. The result matches our observation of the biplot. The data is linear separable on the latent space with less than 5 misclassifications. Therefore, accuracy for LDA and Logistic Regression model remains about the same after PCA. From the biplot of MLE, we can clearly see the data is linear separable with only one mistake. The validation result verifies this observation. Both model only make one misclassificiation and achieve a accuracy of 0.995. Using the same number of factors, MLE performs better than PCA.

3. Effectiveness of Dimension Reduction

In terms of accuracy and the complexity reduced, the dimension reduction I performed is really successful. Both method reduce the feature dimension from six to two. MLE maintains the accuracy of LDA model and increase the accuracy of Logistic Regression model by 0.015.

Conclusion

In this work, I tried different models and dimension reduction technique to classify genuine and counterfeit Swiss banknotes from the Swiss Banknote Dataset. I measured the models' performances with k fold cross validation with k = 5. Without any preprocessing, LDA model has the highest accuracy. I reduced the feature dimension from 6 to 2 with PCA and MLE and found that the models with MLE processed data preforms better. The dimension reduction successfully decreases the complexity of the model and even increases the accuracy. In conclusion, the best model is LDA with MLE preprocessing which achieves an accuracy of 0.995.

---

1. : Flury, B. and Riedwyl, H. (1988). Multivariate Statistics: A practical approach. London: Chapman & Hall, Tables 1.1 and 1.2, pp. 5-8.↩
2. : T. McElroy, "Ma189Fianl" [Online]. Available: https://canvas.ucsd.edu/courses/24041/assignments/375756 (https://canvas.ucsd.edu/courses/24041/assignments/375756) [Accessed: 14-Feb-2021]↩

3. : En.wikipedia.org. 2021. Linear discriminant analysis. [online] Available at: https://en.wikipedia.org/wiki/Linear_discriminant_analysis (https://en.wikipedia.org/wiki/Linear_discriminant_analysis) [Accessed 15 March 2021].↩

4. “Principal Components Analysis (PCA) using SPSS Statistics,” How to perform a principal components analysis (PCA) in SPSS Statistics | Laerd Statistics. [Online]. Available: https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php (https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php). [Accessed: 06-Mar-2021].↩

5. “Principal Components Analysis (PCA) using SPSS Statistics,” How to perform a principal components analysis (PCA) in SPSS Statistics | Laerd Statistics. [Online]. Available: https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php (https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php). [Accessed: 06-Mar-2021].↩