A Discussion about PCA Limitations on Some Prediction Models

Member 1, Member 2, Member 3, Durvish Patel

Abstract

Principal Component Analysis(PCA) is used for coordinate descending by projecting each data onto the first few main components to obtain data with lower dimensions and lower data noise, so using PCA commonly helps to improve accuracy of prediction models which need to rule out the adverse impact of data noise or massive features, such as Decision Trees(DTs). However, in our study of repeating the study process made by the team of M Z F Nausion, we find an abnormal result that using the PCA, in fact, decreases the accuracy of the decision tree which contradicts the conclusion as above. Thus, this paper focuses on figuring out reasons of accuracy loss in using the PCA as the dimension reduction tool, where the analyses are obtained based on testing the accuracy of two Decision trees constructed using original data (Model 1) and new data formed by the first few significant components of PCA (Model 2).

Keywords

Principal Component Analysis; Decision Tree; uncorrelated variables; small feature size; overfitting; double-descent risk curve.

Introduction

The Principal Component Analysis (PCA) is a kind of statistical algorithm that mainly implements dimensionality reduction, which rules out some data noise from a dataset with massive features. The decision tree, whose precise is influenced by noisy datasets, requires a method of coordinate descending to cope with interruptions from multiple variables.

According to recent study from M Z F Nasution et al. (2018) [1], using the decision tree as a prediction model would cause a circumstance of overfitting which would decrease the prediction accuracy. Thus, using the Principal Component Analysis is helpful for the decision tree to avoid the situation of overfitting by reducing the dimension of data.

This paper, we also apply the prediction model --- DTs with PCA --- to analyze and determine which features will make a contribution to the change of Salary. However, we find that the accuracy of Decision Tree with data processed by PCA is lower than the accuracy of Decision Tree constructed using the original dataset. The result contradicts the conclusion in M Z F Nasution' article that the accuracy of Decision Tree with PCA is supposed to be higher than the accuracy of Decision Tree without PCA. Thus, this paper is going to demonstrate reasons for the use of PCA decreasing the model accuracy of the decision tree.

By comparing several papers, we find the decision tree with PCA is effective to make predictions, but there are some limitations for using the Principal Component Analysis method in some dataset. There are several possible problems shown as below: first, the PCA will bring about overfitting in small sample size(SSS); second, it’s possible that the features in Salary data we use are uncorrelated with each other or the correlation is weak; third, the number of data features, p, is far less than the data size, which is insufficient to describe the Salary data precisely.

Materials and Methods

The Decision Tree model and dimensional reduction method are conducted on Salary dataset named “adult”. There are 15 variables in the datasets, as listed in Table 1, and a total of 32561 observations.

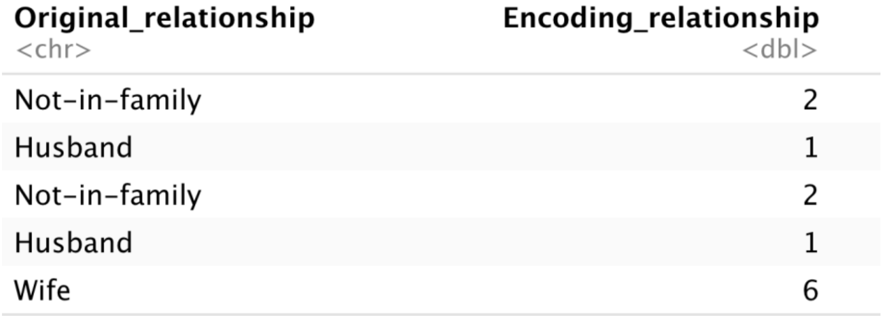
#	Variable	Type
0	age	numeric
1	workclass	character
2	fnlwgt	character
3	education	character
4	education_num	numeric
5	marital_status	character
6	occupation	character
7	relationship	character
8	race	character
9	sex	character
10	capital_gain	numeric
11	capital_loss	numeric
12	hours_per_week	numeric
13	native_country	character
14	(label)salary	character

Table 1. shows 14 variables with corresponding data types

While among all 32561 observations with 14 features, there are 4262 entries missing information. For rows with insufficient information, we delete them to get a subset of data of all participants. However, we have limited information on the reasons that data are missing, so it is hard to determine whether data are missing completely at random (MCAR), at random (MAR), or not missing at random (NMAR). Therefore, if the data is not MCAR, there is a lack of comparability over time points, which would lead to highly biased results [5]. This will be illustrated later in the discussion part.

As we have shown in Table 1, among all variables, there are two data types, numeric and character. In order to perform Principal Component Analysis on these variables, all datasets should be numeric.
The main method we use to transform character to numeric is Label Encoding.

- **Label Encoding:**
  Label Encoding is simply assigning an integer value to every feature included in a variable [4].
  For example, the dataset used in this project has a categorical variable “relationship”. The values drawn from this set are { “Husband”, “Not-in-family”, “Own-child”, “Other-relative”, “Unmarried”, “Wife”}. Then the label encoding will assign the mapped values from the set {1, 2, 3, 4, 5, 6}. The first five rows of results on the relationship variable are shown in Table 2.



Original_relationship <chr>	**Encoding_relationship** <dbl>
Not-in-family	2
Husband	1
Not-in-family	2
Husband	1
Wife	6

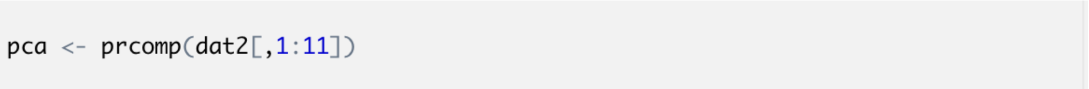
Table 2. shows the first five rows of data of relationship variable after doing label encoding

After transforming all variables into quantitative type, PCA is available to do dimensional reduction.

- **Principal Component Analysis :**
  PCA gives good results when applied to correlated features. In this research, PCA is applieds in both training and testing groups of the Salary data set. The PCA will identify patterns in the data set, and find their similarities and differences between each feature. The covariance matrix of the Salary data is computed whereby the result is used in calculating the eigenvectors and eigenvalues [1] and the eigenvector with the highest eigenvalue is chosen as the principle component of the Salary data set as it exhibits the most significant relationship between the data set features. The Eigenvalues are sorted in ascending order to choose the most significant data and discard the least significant one. By this means data with higher dimensions is reduced to lower dimensions [1]. Thus, the main purpose of using PCA is that : First, to reduce the dimensionality of this data. Second, to find and determine the main

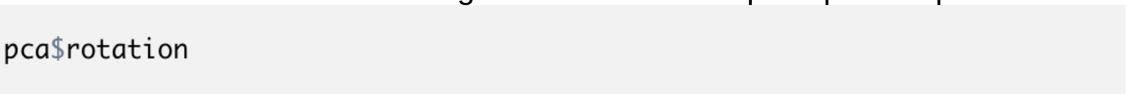
features from many latent features. The remaining non-chosen features in PCA are recognized as the data noise in our research.

The PCA code is shown as below:



```
pca <- prcomp(dat2[,1:11])
```

and this code have shown the eigenvectors for each principal component :



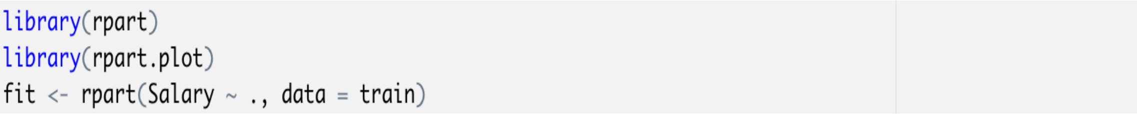
```
pca$rotation
```

- **Decision Tree:**

Decision Tree reserves important features of data for future study due to its advantage of interpretability. The important components of the decision tree are nodes and branches and we construct the decision tree by splitting, puring, and stopping[2].

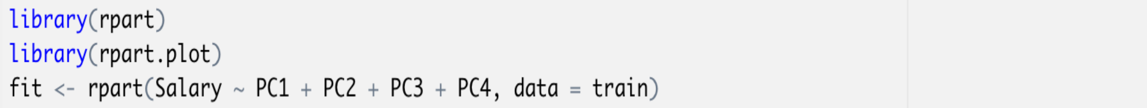
In this project, we mainly use two datasets, original datasets (D1) and data formed by the first four PCs (D2). The way we test for accuracy of decision trees is by randomly selecting 80% of data as training and the rest 20% as testing datasets. This process repeats for both D1 and D2, so we get two pairs of training and testing datasets, where training sets are used for the construction of decision trees and accuracies are measured by processing testing sets to the model.

Decision trees constructed using the two training sets are constructed by selecting most representative features as classifiers to split data into two groups and stops while enough proportion of records are classified[2]. This is done automatically by an r function rpart() which is in the library, “rpart”, as shown in Figure 1 and 2. Models are structured using Salary as the response variable and training datasets as basis.



```
library(rpart)
library(rpart.plot)
fit <- rpart(Salary ~ ., data = train)
```

Figure 1. r code for constructing Decision Tree using original datasets (Model1)



```
library(rpart)
library(rpart.plot)
fit <- rpart(Salary ~ PC1 + PC2 + PC3 + PC4, data = train)
```

Figure 2. r code for constructing Decision Tree using data formed by the first four PCs (Model2)

- **Double-descent risk curve :**

The “double descent” risk curve was proposed by Belkin et al. [3] as a general way to qualitatively describe the out-of-sample prediction performance of variably parameterized prediction models. This risk curve reconciles the classical bias-variance trade-off with the behavior of predictive models that interpolate training data, as observed for several model families in a wide variety of applications. In these studies, a predictive model with p parameters is fit to a training sample of size n, and the test risk (i.e., out-of-sample error) is examined as a function of p. When p is below the sample size n (for regression or binary classification), the test risk is governed by the usual bias-variance decomposition. As p is increased toward n, the training risk (i.e., in-sample error) is driven to zero, but the test risk shoots up, sometimes toward infinity. The classical bias-variance analysis identifies a “sweet spot" value of p \in [0, n] at which the bias and variance are balanced to achieve low test risk. However, in the “modern regime," as p grows beyond n, the training risk remains zero, but the test risk decreases again, even when fitting noisy data, provided that the model is fit using a suitable inductive bias (e.g., least norm solution).

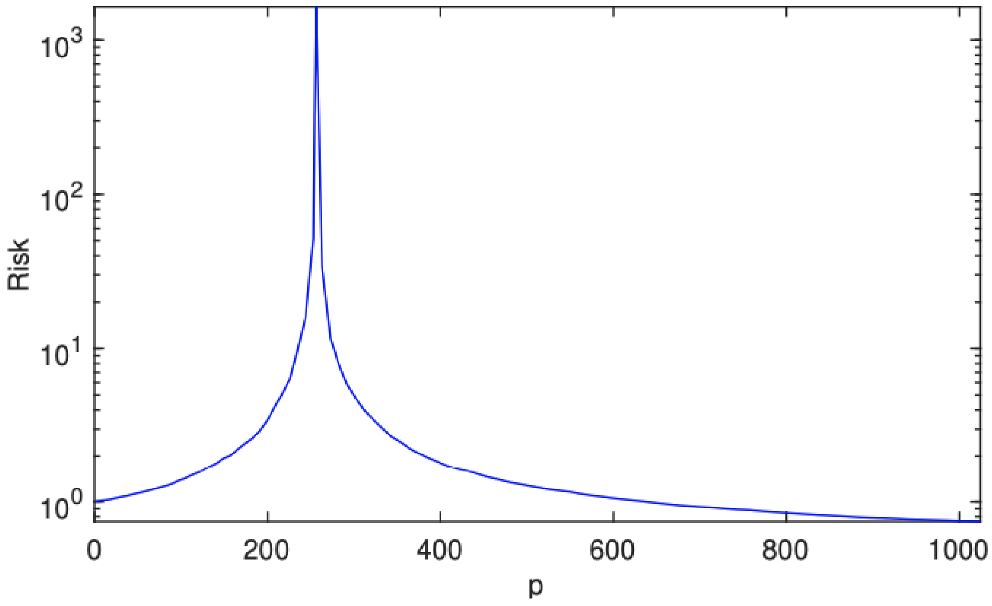


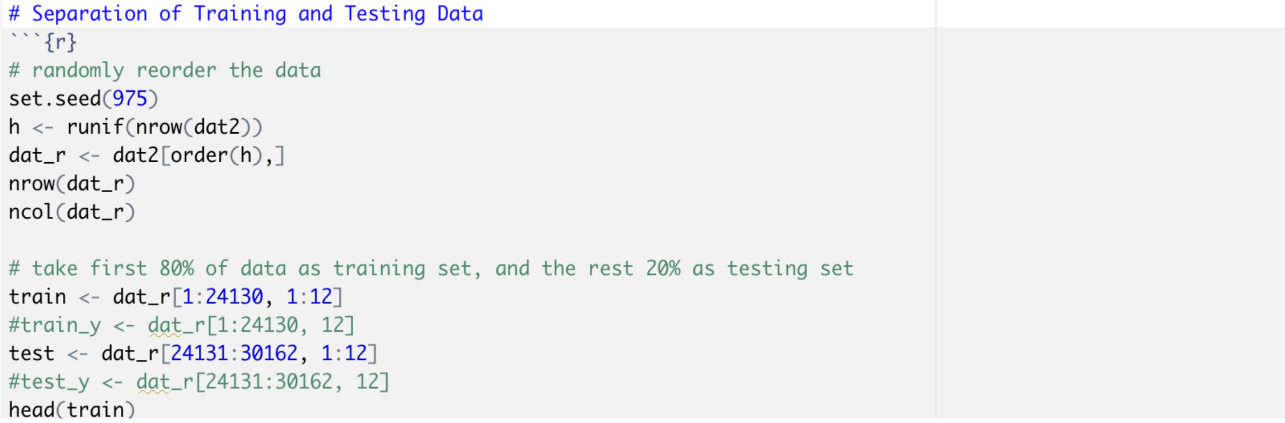
Table 3. shows the risk tendency with the increasing number of p.

Result

Just as what we have mentioned in the introduction, we are trying to get a fitted prediction model for the Salary data (adult.cvs). In our dataset, there are some undefined entries or some categorical entries which can not be detected by our algorithm. Thus, we first delete those empty entries and use the **“Label Encoding” Method** to transform the categorical entries to numeric entries.

1. Decision Tree without PCA

Then, we begin to follow the steps provided by the article of M Z F Nasution,
We need to divide the dataset into two groups randomly --- training group including 80% data and testing group including 20% data.

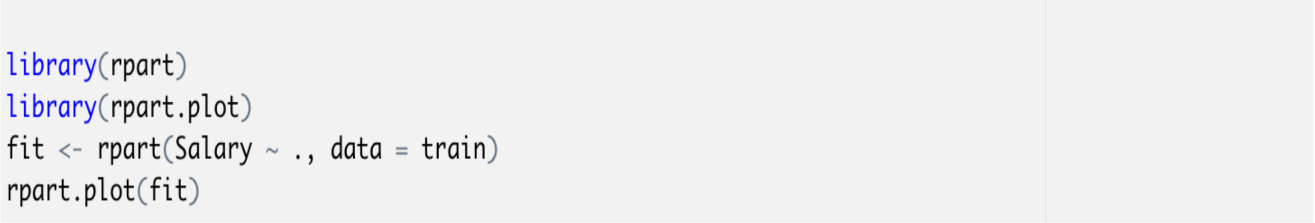


~~~~
# Separation of Training and Testing Data
```{r}
# randomly reorder the data
set.seed(975)
h <- runif(nrow(dat2))
dat_r <- dat2[order(h),]
nrow(dat_r)
ncol(dat_r)

# take first 80% of data as training set, and the rest 20% as testing set
train <- dat_r[1:24130, 1:12]
#train_y <- dat_r[1:24130, 12]
test <- dat_r[24131:30162, 1:12]
#test_y <- dat_r[24131:30162, 12]
head(train)
~~~~

Figure 3. r code for separating the original data into training and testing.

Then, we build up the Decision Tree without PCA.



```
library(rpart)
library(rpart.plot)
fit <- rpart(Salary ~ ., data = train)
rpart.plot(fit)
```

Figure 4. r code for constructing the Decision Tree without PCA

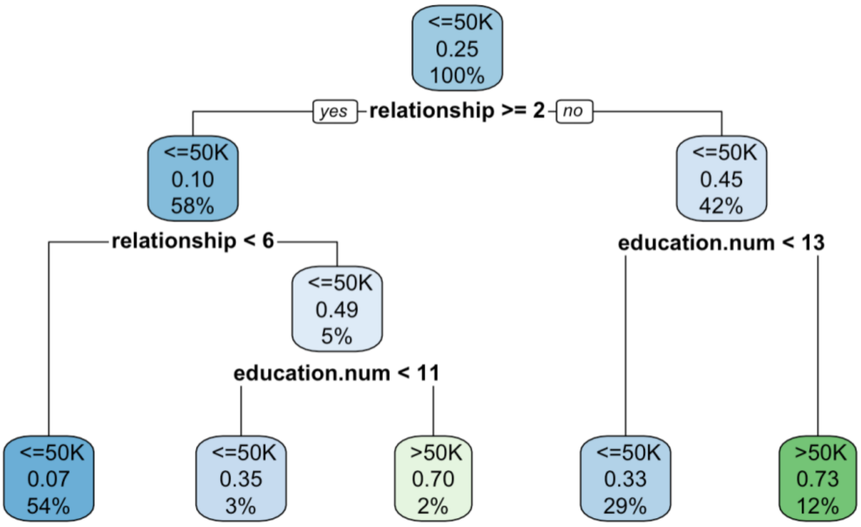
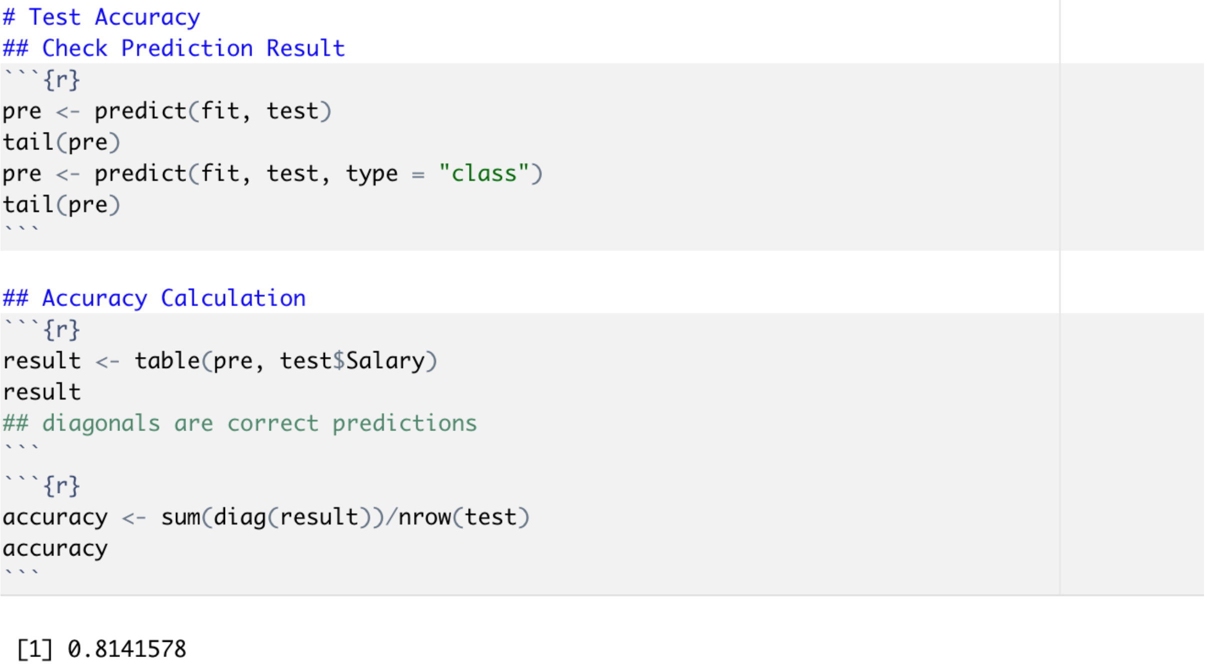


Table 4. shows the decision tree without PCA

The graph above represents how we set the level node of the decision tree without the PCA. Then, we test the accuracy of the Decision Tree prediction model without PCA.



~~~
# Test Accuracy
## Check Prediction Result
```{r}
pre <- predict(fit, test)
tail(pre)
pre <- predict(fit, test, type = "class")
tail(pre)
~~~

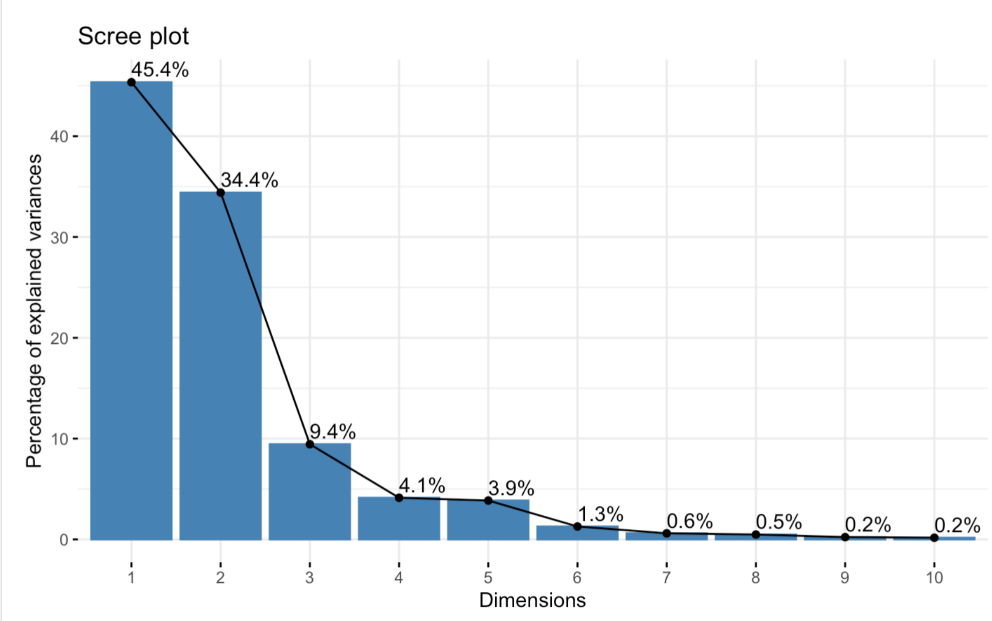
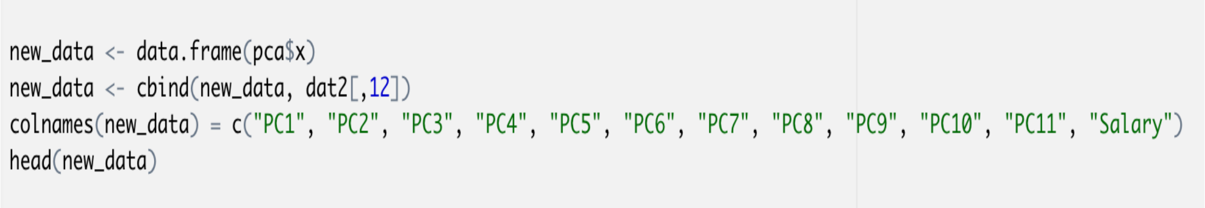
Accuracy Calculation
```{r}
result <- table(pre, test$Salary)
result
## diagonals are correct predictions
```
```{r}
accuracy <- sum(diag(result))/nrow(test)
accuracy
```
```

```
 [1] 0.8141578
```

Figure 5. r code for testing the accuracy of Decision Tree without PCA

Here, we get the accuracy of **0.8141578.**

2. Decision Tree with PCA

Here, we will incorporate PCA into the prediction model, Decision Tree .

We firstly use the PCA method to deal with the Data, and we determine the first 4 PCs are the most principal factors for Salary data as Table5. shows


```

Table 5. shows the Scree plot for proportion of explained Variances in PCA

The first 4 PCs have accounted for 93.3% of the explained variances, so the first 4 PCs are determinant to describe the whole Salary data.

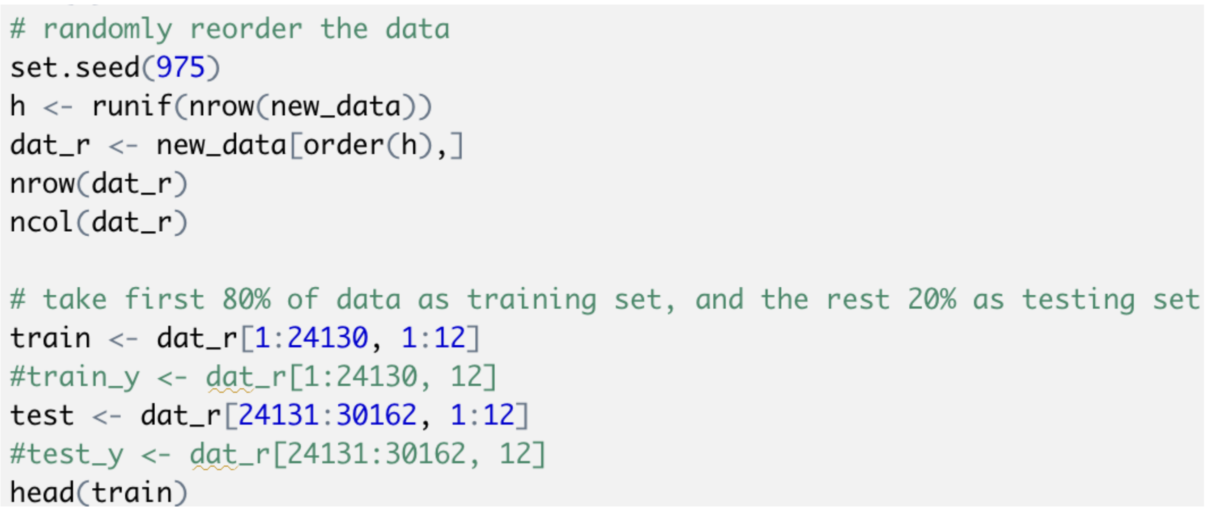
Next, we build up a new_data by PCA, and then choose the first 4 PCs to represent the whole Salary data.



```
new_data <- data.frame(pca$x)
new_data <- cbind(new_data, dat2[,12])
colnames(new_data) = c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9", "PC10", "PC11", "Salary")
head(new_data)
```

Figure 6. r code for forming a new_data

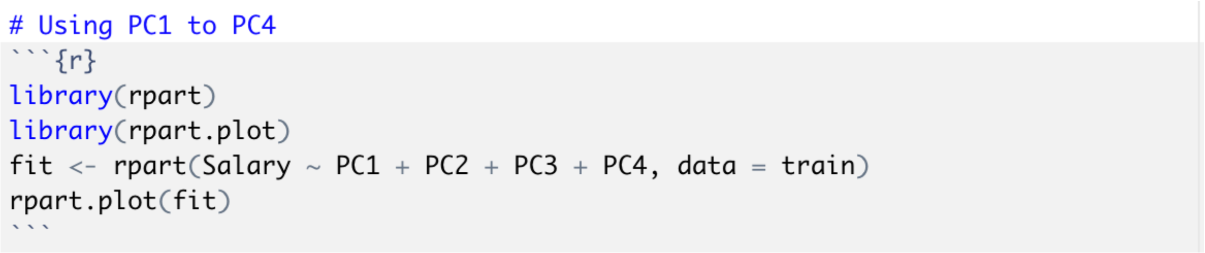
providing a new training group with 80% random data of new_data and a new testing group with 20% random data of new_data.



```
# randomly reorder the data
set.seed(975)
h <- runif(nrow(new_data))
dat_r <- new_data[order(h),]
nrow(dat_r)
ncol(dat_r)

# take first 80% of data as training set, and the rest 20% as testing set
train <- dat_r[1:24130, 1:12]
#train_y <- dat_r[1:24130, 12]
test <- dat_r[24131:30162, 1:12]
#test_y <- dat_r[24131:30162, 12]
head(train)
```

Figure 7. r code for building up a new training set and testing set.

Then, constructing the Decision Tree with PCA



~~~
# Using PC1 to PC4
```{r}
library(rpart)
library(rpart.plot)
fit <- rpart(Salary ~ PC1 + PC2 + PC3 + PC4, data = train)
rpart.plot(fit)
```
~~~

Figure 8. r code for constructing a new Decision Tree with PCA

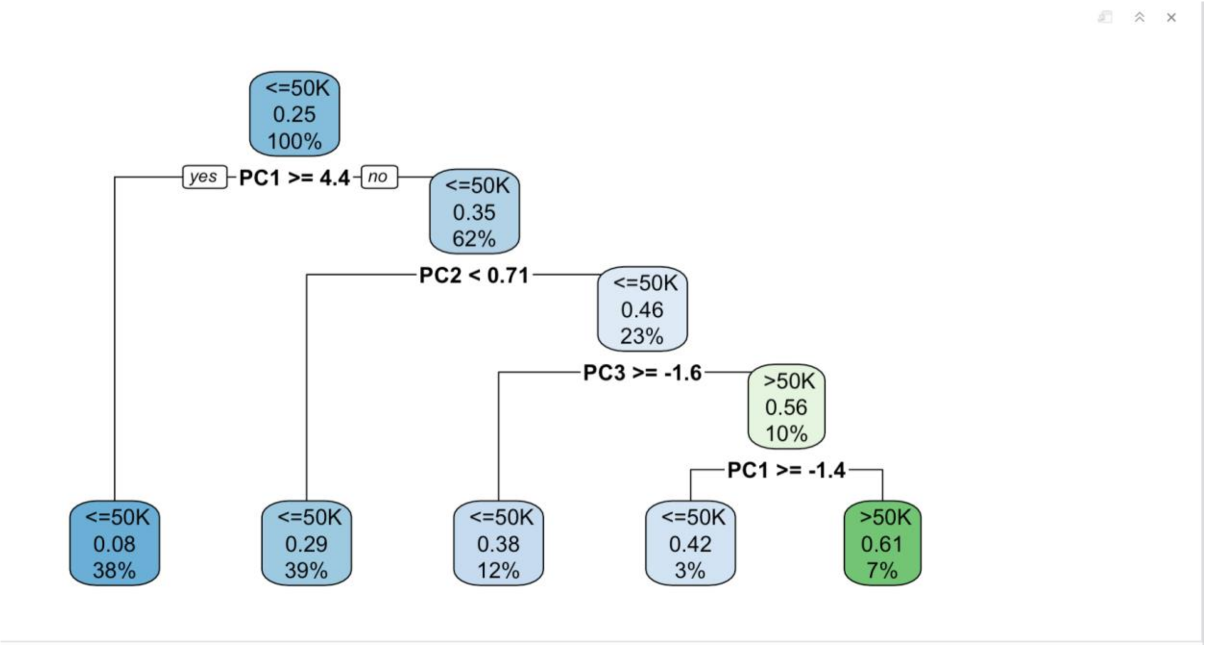
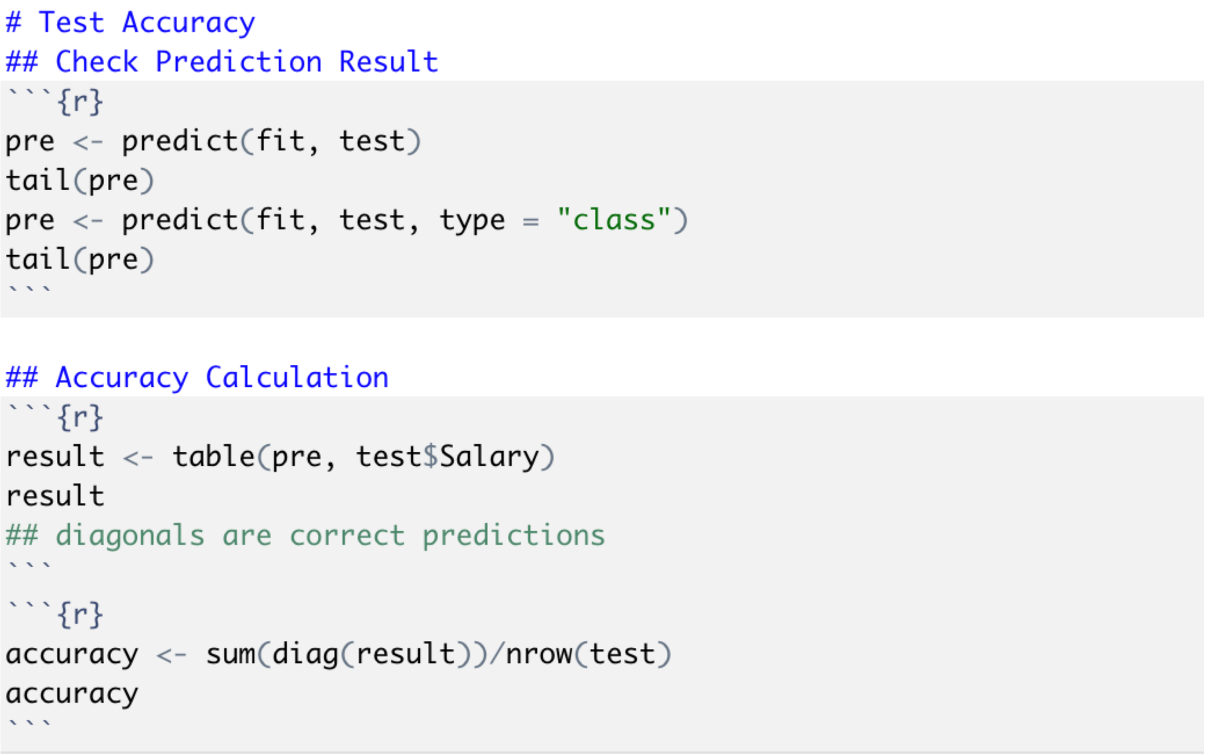
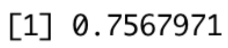


Table 6. shows the Decision Tree with PCA

and then we test the accuracy of the prediction model, Decision Tree with PCA,



~~~
# Test Accuracy
## Check Prediction Result
```{r}
pre <- predict(fit, test)
tail(pre)
pre <- predict(fit, test, type = "class")
tail(pre)
~~~

Accuracy Calculation
```{r}
result <- table(pre, test$Salary)
result
## diagonals are correct predictions
```
```{r}
accuracy <- sum(diag(result))/nrow(test)
accuracy
```
```



```
[1] 0.7567971
```

Figure 9. r code for testing the accuracy of Decision Tree with PCA
```

We find that the accuracy of Decision Tree with PCA is **0.7567971**, which is lower about 0.06 than the accuracy of Decision Tree without PCA, **0.8141578.** Thus, it contradicts the conclusion we get in the article of M Z F Nasution[1].

### 3. More studies in the contradiction

Because of this new contradiction rising, we are now attempting to find out which factors lead to this situation. After comparing with several relevant academic papers, we find there exists 3 limitations for the utilization of PCA 1) PCA is prone to overfit the data in Small Sample Size(SSS) problems. 2) PCA is not effective to reduce the dimensionality of Data when the data itself includes some uncorrelated features or the correlation is weak. 3) it's possible that the number of features to describe the dataset is too small to represent all the information of Salary data according to the "Double-Descent" risk curve.

## Discussion

As we have mentioned in the Result section, part 3, More studies in the contradiction, there are 3 limitations for using the PCA method. Here, we will make more elaborated explanations for these 3 limitations.

### 1. The Overfitting Problem of Classical PCA

When applied to image recognition, classical PCA is prone to be over-fitted to the training set due to the SSS problem. To verify this perspective, we carried out a series of experiments using the ORL database.

The ORL database contains 400 facial images, with 10 images per individual. The images vary in age, light conditions, facial expressions, facial details (glasses/no glasses), scale and tilt. The size of these images is 112×92.

Now we use the normalized mean-square error (MSE) to evaluate the overfitting problem. One statistical characteristic of PCA is that the MSE between random vector x and its subspace projection is minimal. Thus the difference of MSE on the training set and the testing set can be used to investigate the over-fitting problem.

Given the first L principal components, we can obtain the corresponding projector Wl. Then a vector x can be transformed into the PCA subspace by

$$\mathbf{y} = \mathbf{W}_L^T(\mathbf{x} - \overline{\mathbf{x}}), \tag{1}$$

and the reconstructed vector x~ can be represented as

$$\tilde{\mathbf{x}} = \overline{\mathbf{x}} + \mathbf{W}_L\mathbf{y} = \overline{\mathbf{x}} + \mathbf{W}_L\mathbf{W}_L^T(\mathbf{x} - \overline{\mathbf{x}}), \tag{2}$$

where $\overline{\mathbf{x}}$ is the mean vector. The MSE on the training set $MSE_L^{train}$ is defined as

$$MSE_L^{train} = \sum_{i=1}^{N_1} \left\| \mathbf{x}_i^{train} - \tilde{\mathbf{x}}_i^{train} \right\|^2 \Big/ \sum_{i=1}^{N_1} \left\| \mathbf{x}_i^{train} - \overline{\mathbf{x}_i^{train}}^2 \right. , \qquad (3)$$

where $N_1$ is the size of training set, $\mathbf{x}_i^{train}$ is the $i$th training samples, $\tilde{\mathbf{x}}_i^{train}$ is the reconstructed vector of $\mathbf{x}_i^{train}$, and $\mathbf{x}^{train}$ is the mean vector of all training samples. Similarly, we can calculate the MSE on the testing set $MSE_L^{test}$.

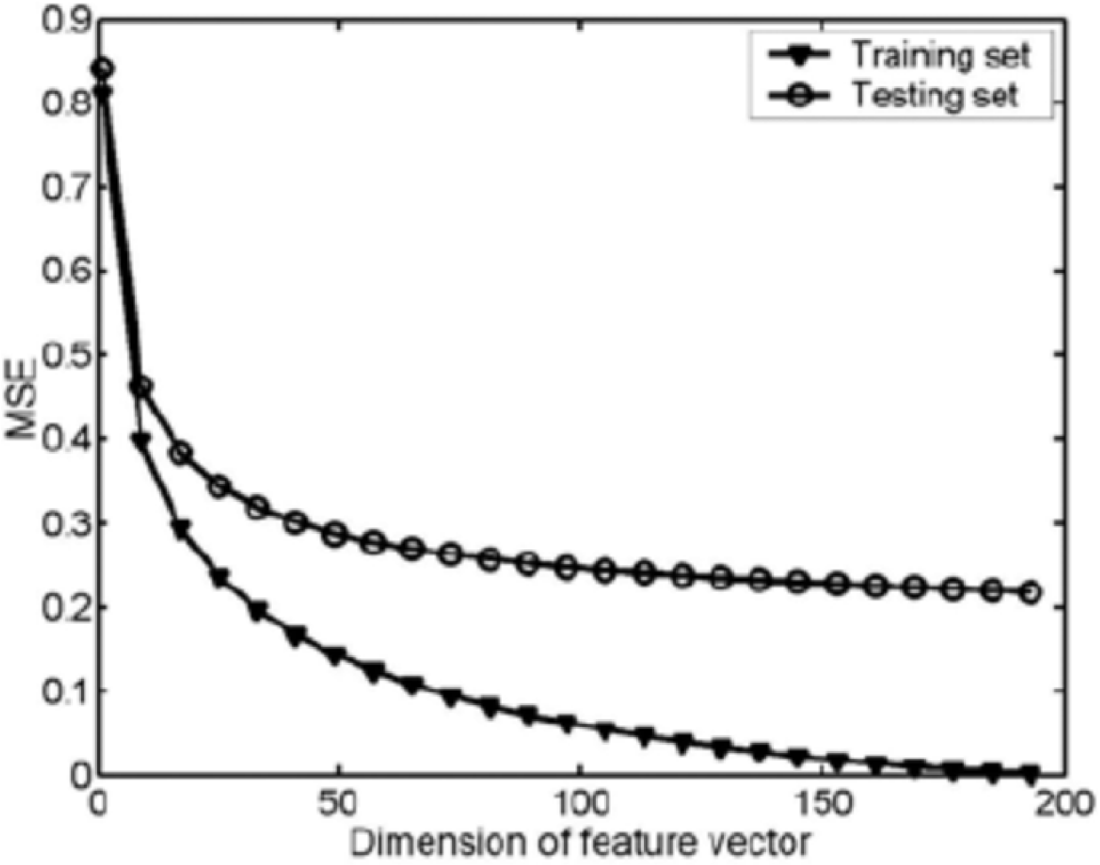


Table 7. shows the PCA's MSE on the training set and the testing set as the function of feature dimension

## 2. PCA is not effective to reduce dimension in orthogonal features.

The principle of PCA to reduce the dimensionality is through projecting non-orthogonal features on a null space into fewer dimensions with maximum variability of a data. And then, the center of a new coordinate system built by principal dimensions will locate on the mean vector ***u.***

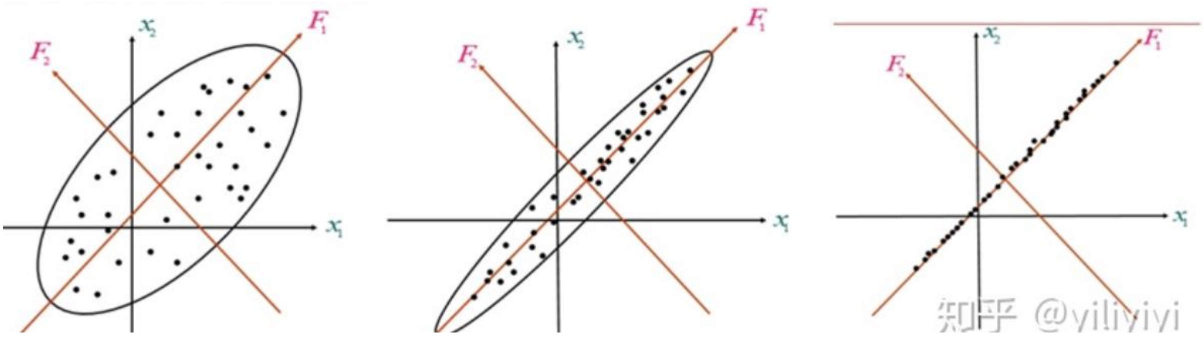


Table 8. shows how to keep the maximum variability of a data by rotating

After projecting those non-orthogonal features into some dimensions, the PCA algorithm will rotate the coordinate system until getting the maximum variability of this data. Then, we have built up the principal component of this data.

Thus, under this case, we can find that the correlation --- "correlated" is equivalent to "non-orthogonal" --- between each feature in this Salary Data is weak. Just as the table shown below:

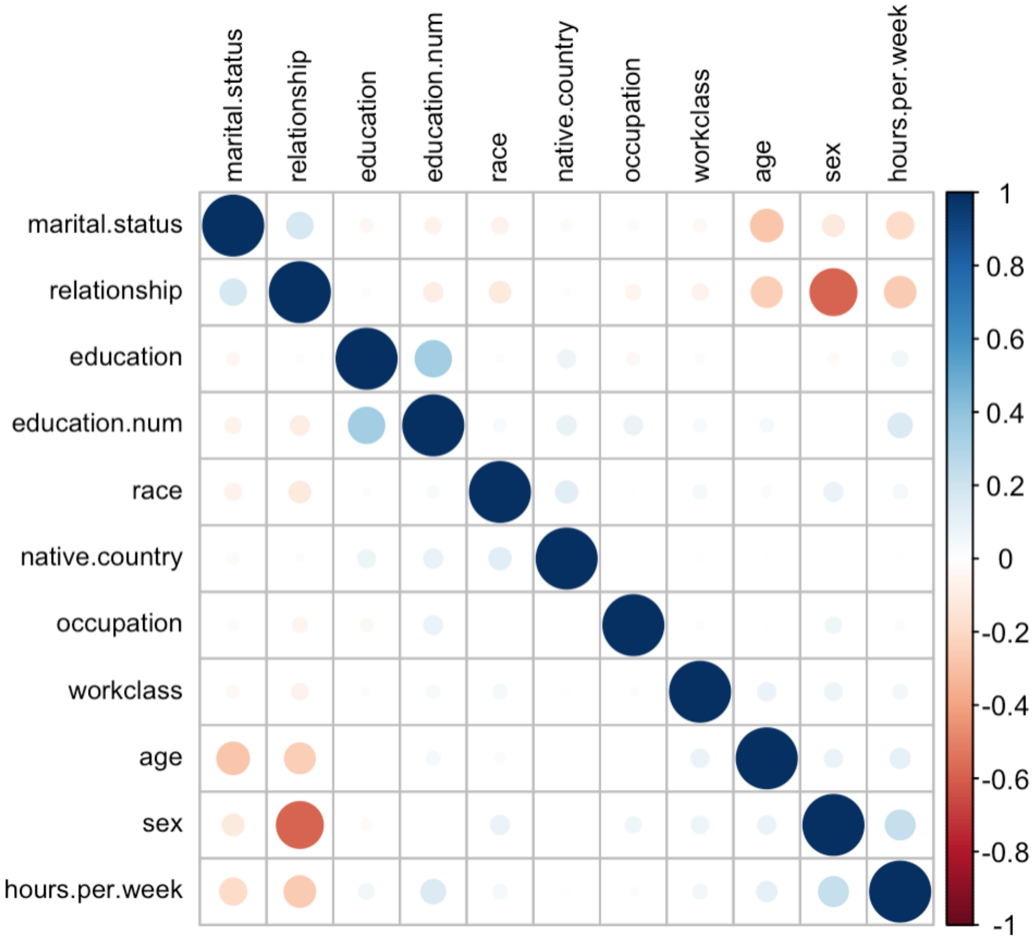


Table 9. shows the correlation between each feature.

Thus, the PCA method is not appropriate for the Salary data.

### 3. Double-Descent risk curve

In the modern Regime of machine learning, PCA also plays an important role like in image recognition. However, we always will encounter a situation that the error in the test group will first decrease and then increase in the classic Regime; however, when the number of features is big enough, the curve of test error will decrease again, which does better in fitting to the reality. Just as the tendency shown in the table below,

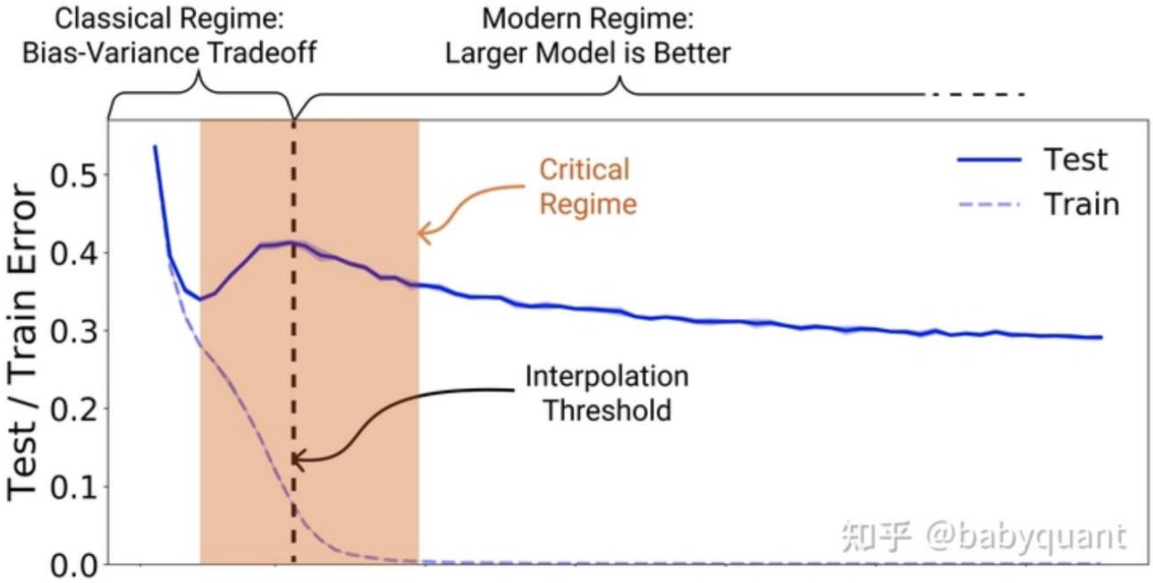


Table 10. shows the tendency of error in Train set and Test set.

Thus, this theory provides us with so many realistic meanings --- if we are supported by a supercomputer with very strong computing power to collect as many features as possible in our model, then we are able to fit as many complicated models with too many features as possible. Thus, in some way, we will not be afraid of overfitting problems in high dimensional statistics if we are supported by supercomputing power. In other words, if we want to let the prediction model be as fitted as possible, then it’s better that we use as many features as possible to describe the data.

Then, Coming back to our research, we analyze that the number of features, p = 14, is far less than the number of data size, n = 30162, so we infer that the p features is not sufficient to describe all the information in salary data.

## Conclusion

After the above analysis, we conclude that there are three limitations for us to use the PCA method for the Salary data in this study: 1) Small sample sizes cause PCA to over fit the data; 2) significant relationships between variables is missing while using PCA to deal with data containing uncorrelated variables; 3) limited number of features cannot describe all information of data, which causes unavailability of using PCA to improve model precise. To use PCA, we should think more about the requirements and appropriate situations. For the data of Salary, insufficient amount of data size, low correlation between variables, and small number of features are inevitable. For our goal of achieving a high accuracy prediction model, building the Decision Tree without PCA is much more appropriate.

## Works Cited

**[1]M Z F Nasution, O S Sitompul, and M Ramli "PCA based feature reduction to improve the accuracy of decision tree c4.5 classification" al 2018 J. Phys.: Conf. Ser. 978 012058**

**[2] Song, Yan-Yan, and L. U. Ying. "Decision tree methods: applications for classification and prediction."** ***Shanghai archives of psychiatry*** **27.2 (2015): 130.**

**[3]Mikhail Belkin, Daniel Hsu, and Ji Xu "Two Models of Double Descent for Weak Features".** ***SIAM Journal on Mathematics of Data Science*****, 2(4), 1167–1180.**

**[4] Hancock, John T., and Taghi M. Khoshgoftaar. "Survey on categorical data for neural networks." Journal of Big Data 7 (2020): 1-41.**

**[5] Bennett, Derrick A. "How can I deal with missing data in my study?."** ***Australian and New Zealand journal of public health*** **25.5 (2001): 464-469.**

**[6]Wangmeng Zuo, Kuanquan Wang and D. Zhang, "Bi-directional PCA with assembled matrix distance metric,"** ***IEEE International Conference on Image Processing 2005*****, Genova, Italy, 2005, pp. II-958, doi: 10.1109/ICIP.2005.1530216.**