Лабораторная работа №4 «Применение методов машинного обучения к данным из хранилища больших данных»

Выполнили студенты группы ИТ-50916: Кураженкова О.С., Ставских А.Д., Тюленева Е.М.

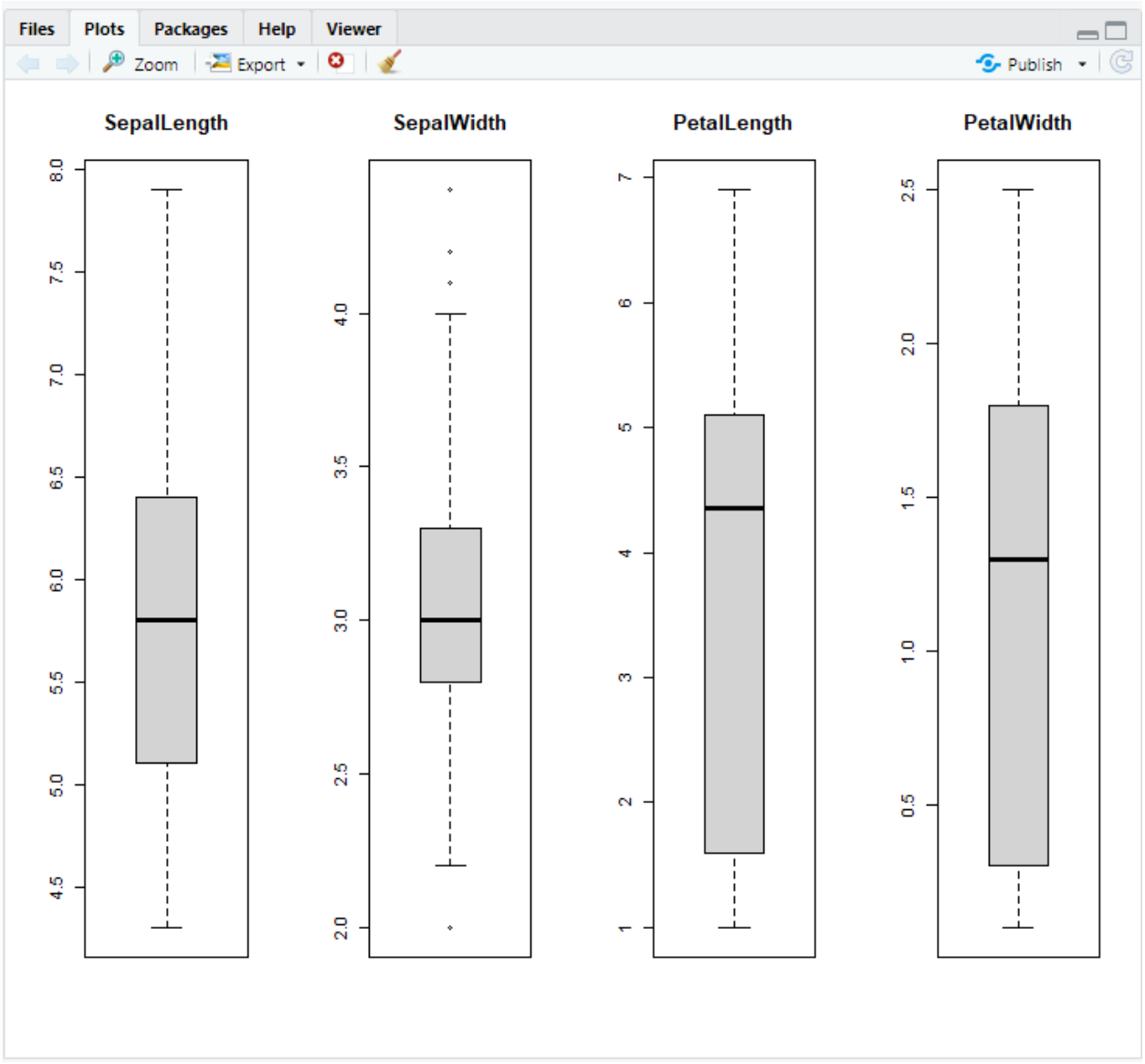
Изучение структуры данных

```
Console   Terminal ×   Jobs ×
~/
> #Изучениие структуры полученных данных
> summary(res)
  SepalLength      SepalWidth     PetalLength      PetalWidth                Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Iris-setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Iris-versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Iris-virginica :50
 Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> dim(res)
[1] 150   5
> str(res)
'data.frame':   150 obs. of  5 variables:
 $ SepalLength: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ SepalWidth : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ PetalLength: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ PetalWidth : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species    : Factor w/ 3 levels "Iris-setosa",..: 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "data.type")= chr [1:5] "Float64" "Float64" "Float64" "Float64" ...
> sapply(res, class)
SepalLength  SepalWidth PetalLength  PetalWidth     Species
  "numeric"   "numeric"   "numeric"   "numeric"    "factor"
> head(res)
  SepalLength SepalWidth PetalLength PetalWidth     Species
1         5.1        3.5         1.4        0.2 Iris-setosa
2         4.9        3.0         1.4        0.2 Iris-setosa
3         4.7        3.2         1.3        0.2 Iris-setosa
4         4.6        3.1         1.5        0.2 Iris-setosa
5         5.0        3.6         1.4        0.2 Iris-setosa
6         5.4        3.9         1.7        0.4 Iris-setosa
> levels(res$Species)
[1] "Iris-setosa"     "Iris-versicolor" "Iris-virginica"
>
```

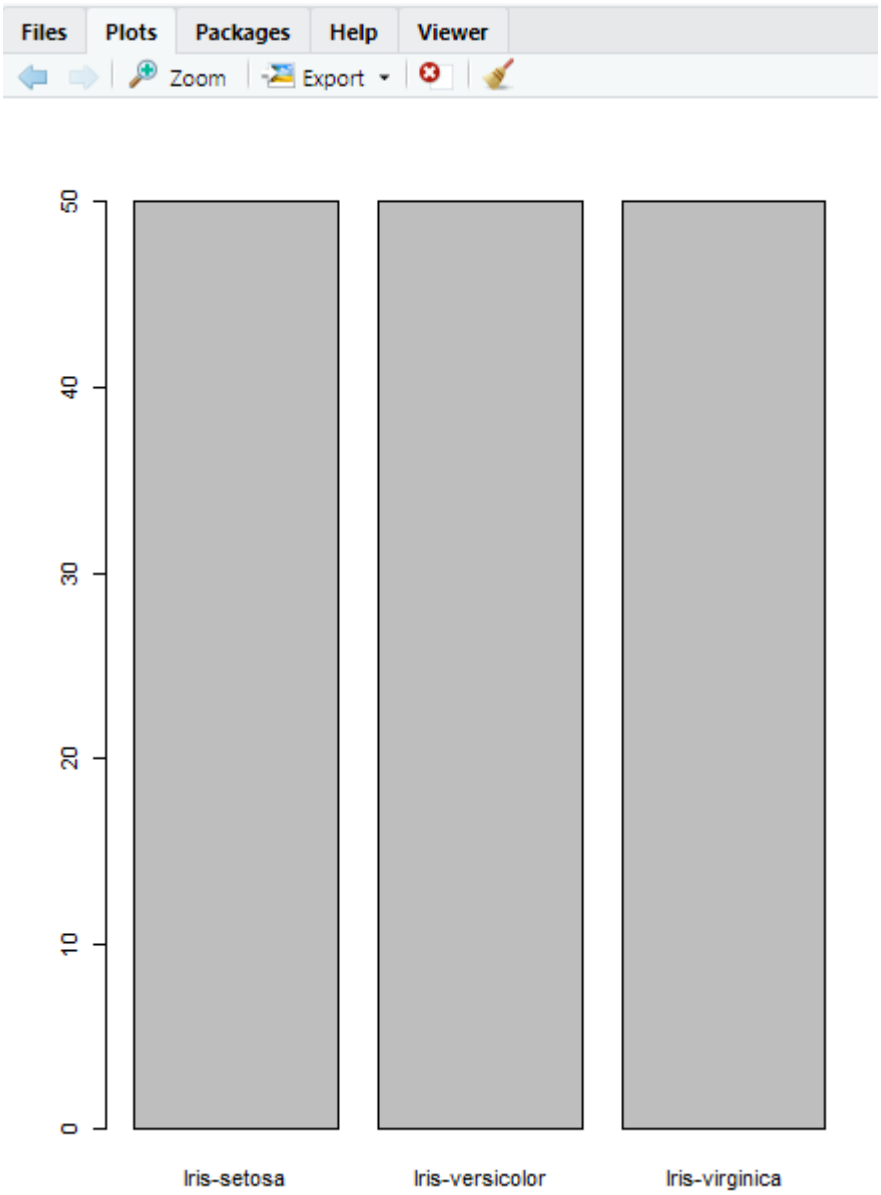
Исследование и визуализирование данных

```
Console  Terminal  Jobs
~/
> #Исследование и визуализирование данных
> #Распределение видов ирисов в данном наборе
> prct<-prop.table(table(res$Species))*100
> cbind(frequency=table(res$Species), percentage=prct)
                frequency percentage
Iris-setosa            50   33.33333
Iris-versicolor        50   33.33333
Iris-virginica         50   33.33333
> #Разобьём данные на переменные (x) и отклик (y)
> x<-res[,1:4]
> y<-res[,5]
> #Визуализируем выборки диаграммой размаха
> par(mfrow=c(1,4))
> for(i in 1:4)
+ {
+   boxplot(x[,i], main=names(res)[i])
+ }
>
```

Диаграммы размаха для каждой из переменных



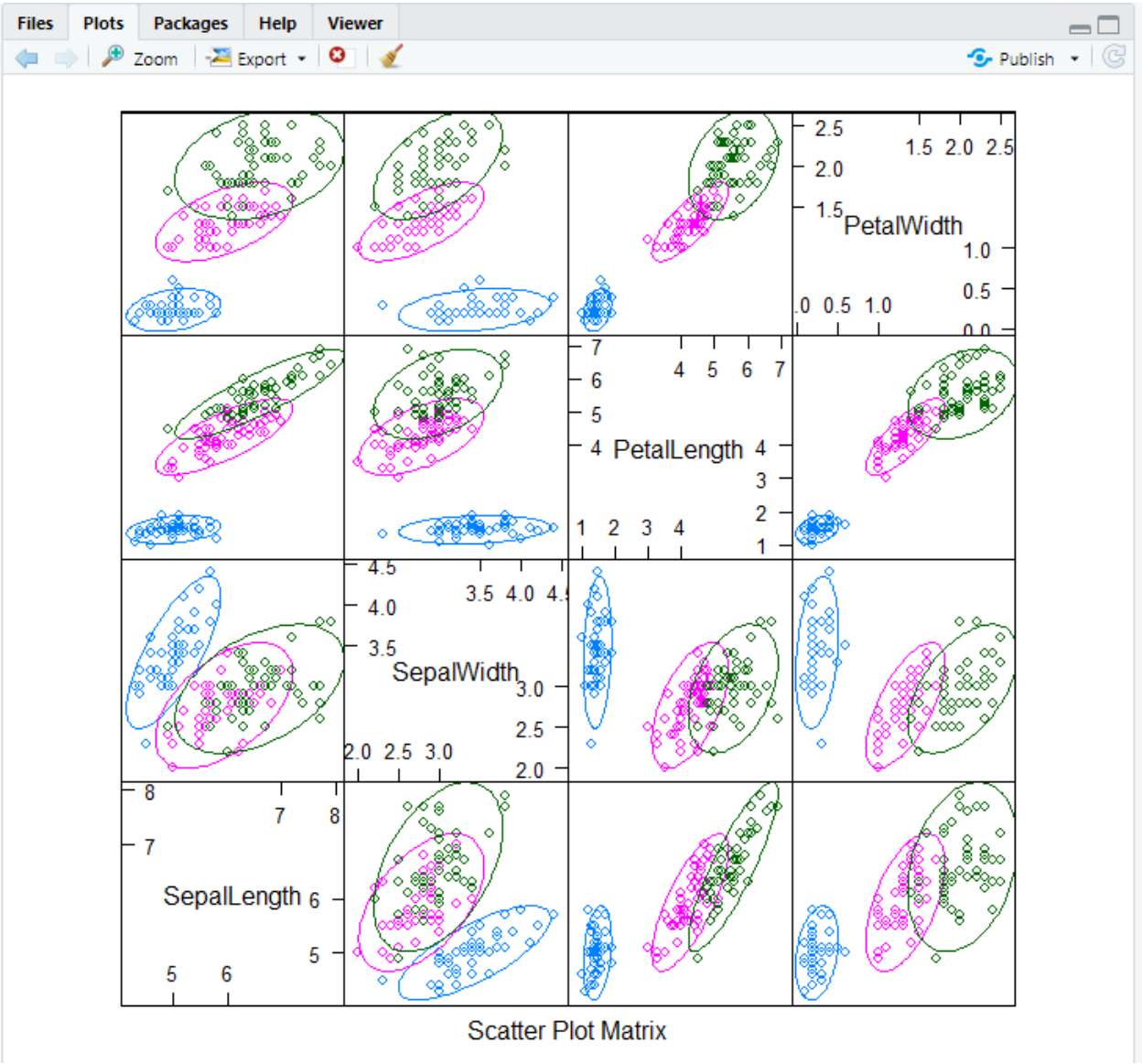
Графическое распределение ирисов в данном наборе



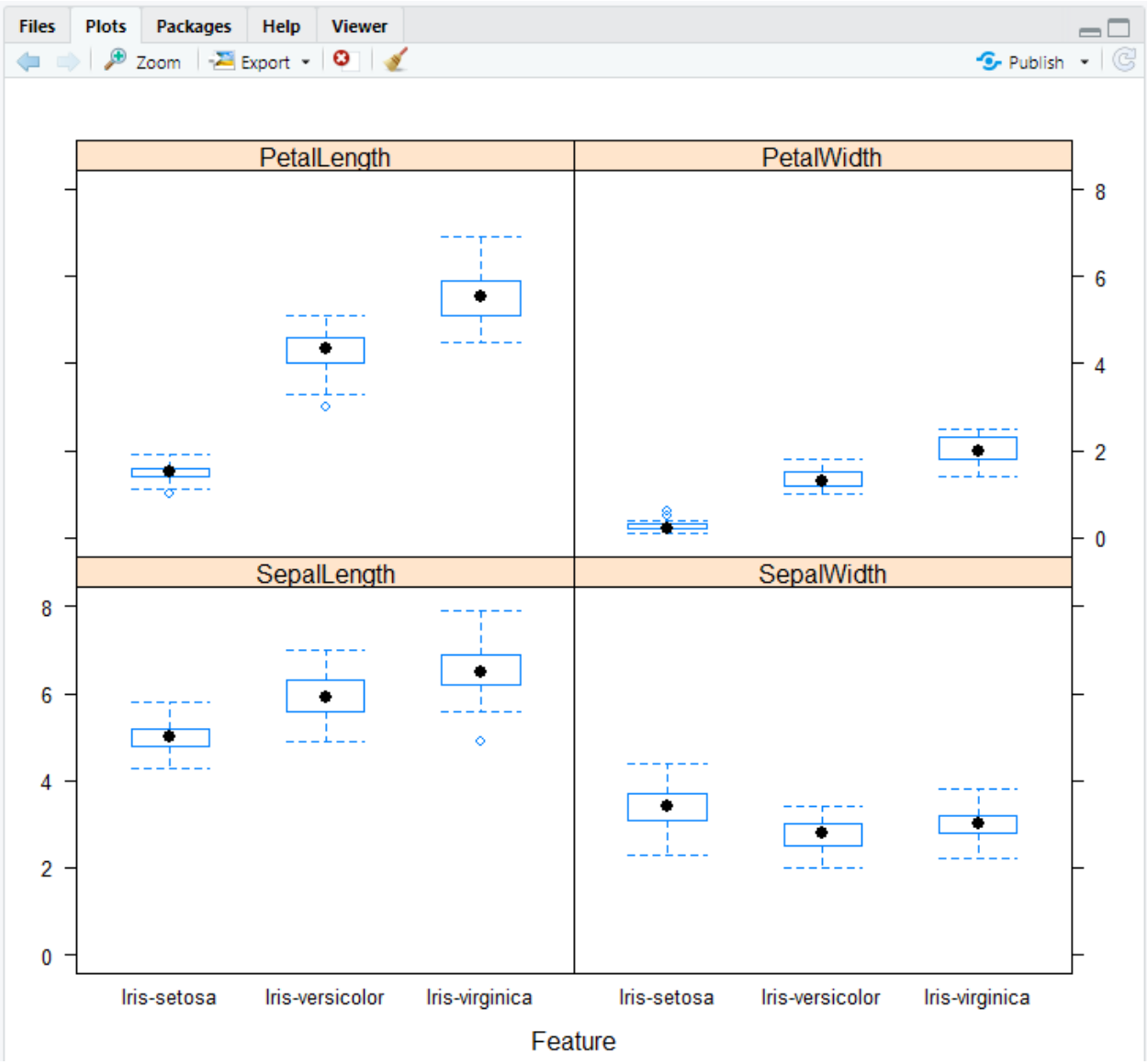
Исследуем взаимосвязи между переменными

```
> library(caret)
Загрузка требуемого пакета: lattice
Загрузка требуемого пакета: ggplot2
> #Исследуем взаимодействие внутри данных
> featurePlot(x=x, y=y, plot="ellipse")
> featurePlot(x=x, y=y, plot="box")
>
```

Диаграмма рассеяния для всех пар атрибутов с объединением в эллипсы



Диаграммы размаха каждой переменной с разбиением на классы



Перед началом машинного обучения разобьём данные на два набора: данные для обучения, данные для проверки качества обучения.

```
Console  Terminal  Jobs
~/
> #Машинное обучение
>
> #Заранее "отрежем" набор проверочных данных, для проверки качества обученной модели
> #возьмём 80% данных
> validIndex<-createDataPartition(res$Species, p=0.8, list=FALSE)
> #20% данных для проверки качества обученной модели
> validation<-res[-validIndex,]
> #80% данных для обучения моделей
> res<-res[validIndex,]
>
```

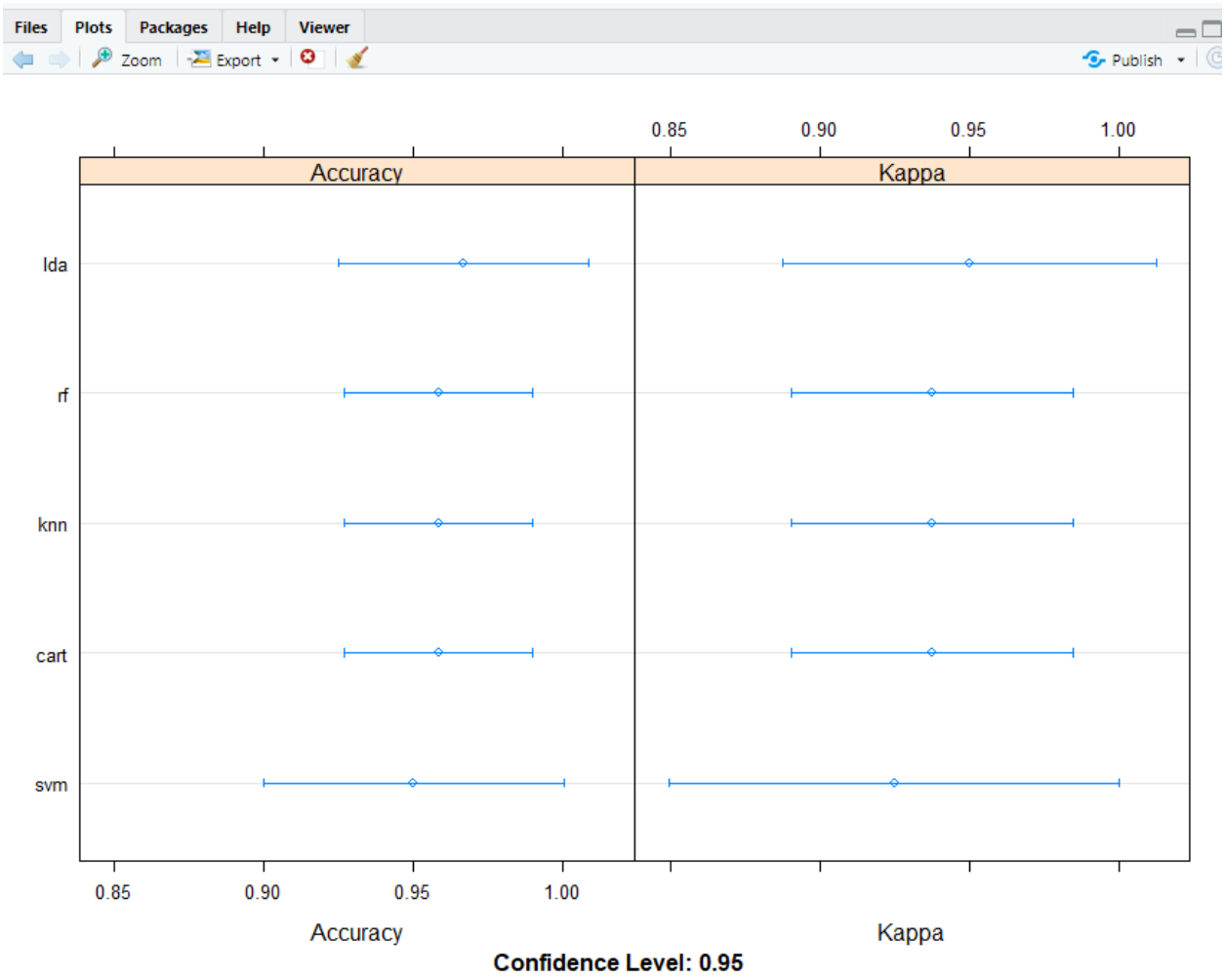
Проведём обучение разных 5-ти моделей

```
Console  Terminal  Jobs
~/
> #Настроим перекрёстную проверку (кроссвалидацию) по 10 блокам
> control<-trainControl(method="cv",number=10)
> #Проверяемая метрика - точность
> metric<-"Accuracy"
> #Построение моделей обучения
> #LDA (Линейные алгоритмы)
> set.seed(13)
> fit.lda<-train(Species~., data=res, method="lda", metric=metric, trControl=control)
> #CART (Нелинейные алгоритмы)
> set.seed(13)
> fit.cart<-train(Species~., data=res, method="rpart", metric=metric, trControl=control)
> #KNN (Нелинейные алгоритмы)
> set.seed(13)
> fit.knn<-train(Species~., data=res, method="knn", metric=metric, trControl=control)
> #SVM (Сложные алгоритмы)
> set.seed(13)
> fit.svm<-train(Species~., data=res, method="svmRadial", metric=metric, trControl=control)
> #Random Forest (Сложные алгоритмы)
> set.seed(13)
> fit.rf<-train(Species~., data=res, method="rf", metric=metric, trControl=control)
```

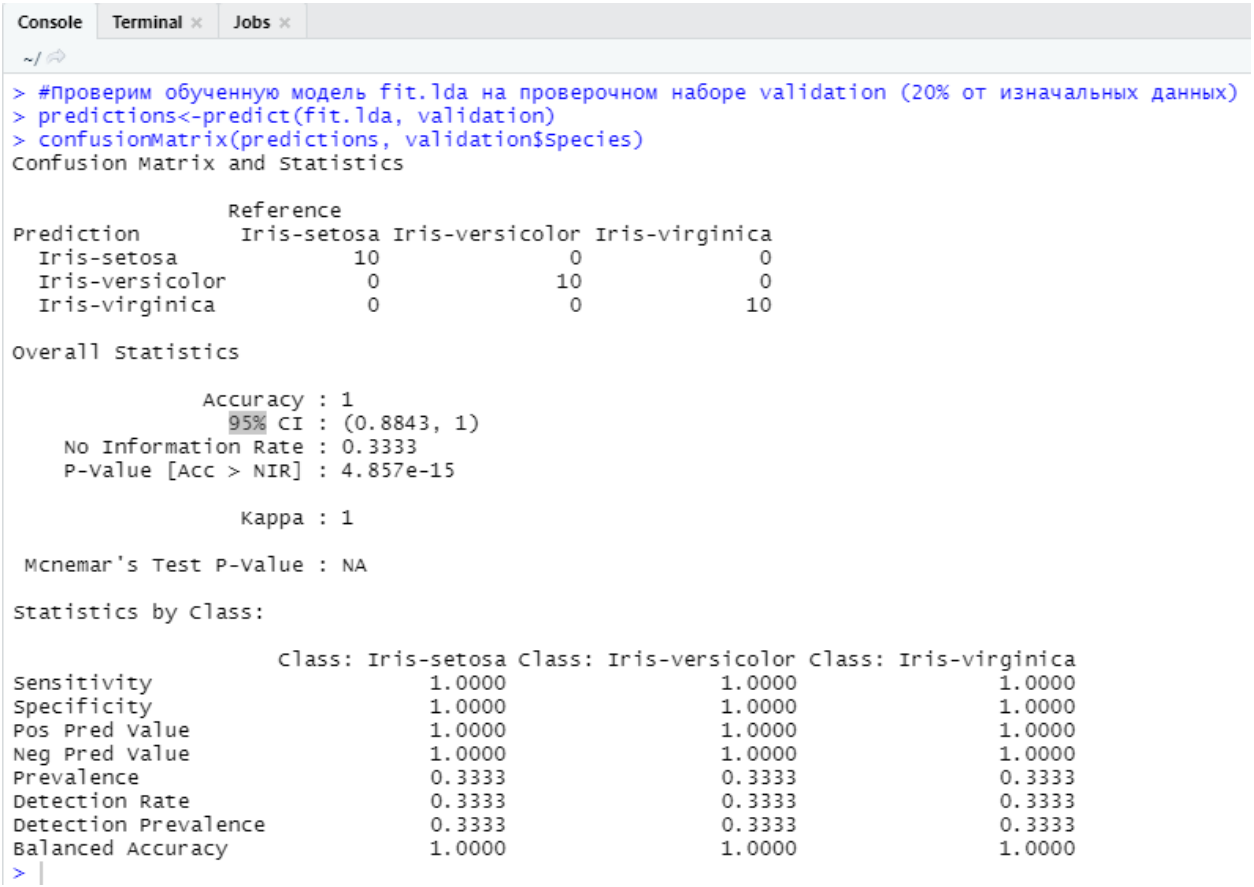
Получим оценки точности для каждого алгоритма

```
Console  Terminal  Jobs
~/
>
> #Получим оценки контролируемой метрики (точности) для каждого алгоритма
> results<-resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))
> summary(results)

Call:
summary.resamples(object = results)

Models: lda, cart, knn, svm, rf
Number of resamples: 10

Accuracy
          Min.   1st Qu.    Median      Mean 3rd Qu. Max. NA's
lda  0.8333333 0.9375000 1.0000000 0.9666667       1    1    0
cart 0.9166667 0.9166667 0.9583333 0.9583333       1    1    0
knn  0.9166667 0.9166667 0.9583333 0.9583333       1    1    0
svm  0.8333333 0.9166667 1.0000000 0.9500000       1    1    0
rf   0.9166667 0.9166667 0.9583333 0.9583333       1    1    0

Kappa
      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
lda  0.750 0.90625 1.0000 0.9500       1    1    0
cart 0.875 0.87500 0.9375 0.9375       1    1    0
knn  0.875 0.87500 0.9375 0.9375       1    1    0
svm  0.750 0.87500 1.0000 0.9250       1    1    0
rf   0.875 0.87500 0.9375 0.9375       1    1    0

> #Визуализируем полученные оценки
> dotplot(results)
```

Визуализация полученных оценок точности (самая высокая у LDA)



Проверим точность определения вида ирисов LDA-модели, с помощью валидационных данных, не вошедших в данные для обучения модели



```
Console   Terminal ×   Jobs ×
~/
> #Проверим обученную модель fit.lda на проверочном наборе validation (20% от изначальных данных)
> predictions<-predict(fit.lda, validation)
> confusionMatrix(predictions, validation$Species)
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              10              0              0
  Iris-versicolor           0             10              0
  Iris-virginica            0              0             10

Overall Statistics

               Accuracy : 1
                 95% CI : (0.8843, 1)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 4.857e-15

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 1.0000                1.0000
Specificity                      1.0000                 1.0000                1.0000
Pos Pred Value                   1.0000                 1.0000                1.0000
Neg Pred Value                   1.0000                 1.0000                1.0000
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3333                0.3333
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 1.0000                1.0000
> |
```

Точность определения сорта ириса у выбранного алгоритма – 95%.

Полный код программы в файле «4_MachineLearning.R»