kNN

K Nearest Neighbours

K Nearest Neighbours

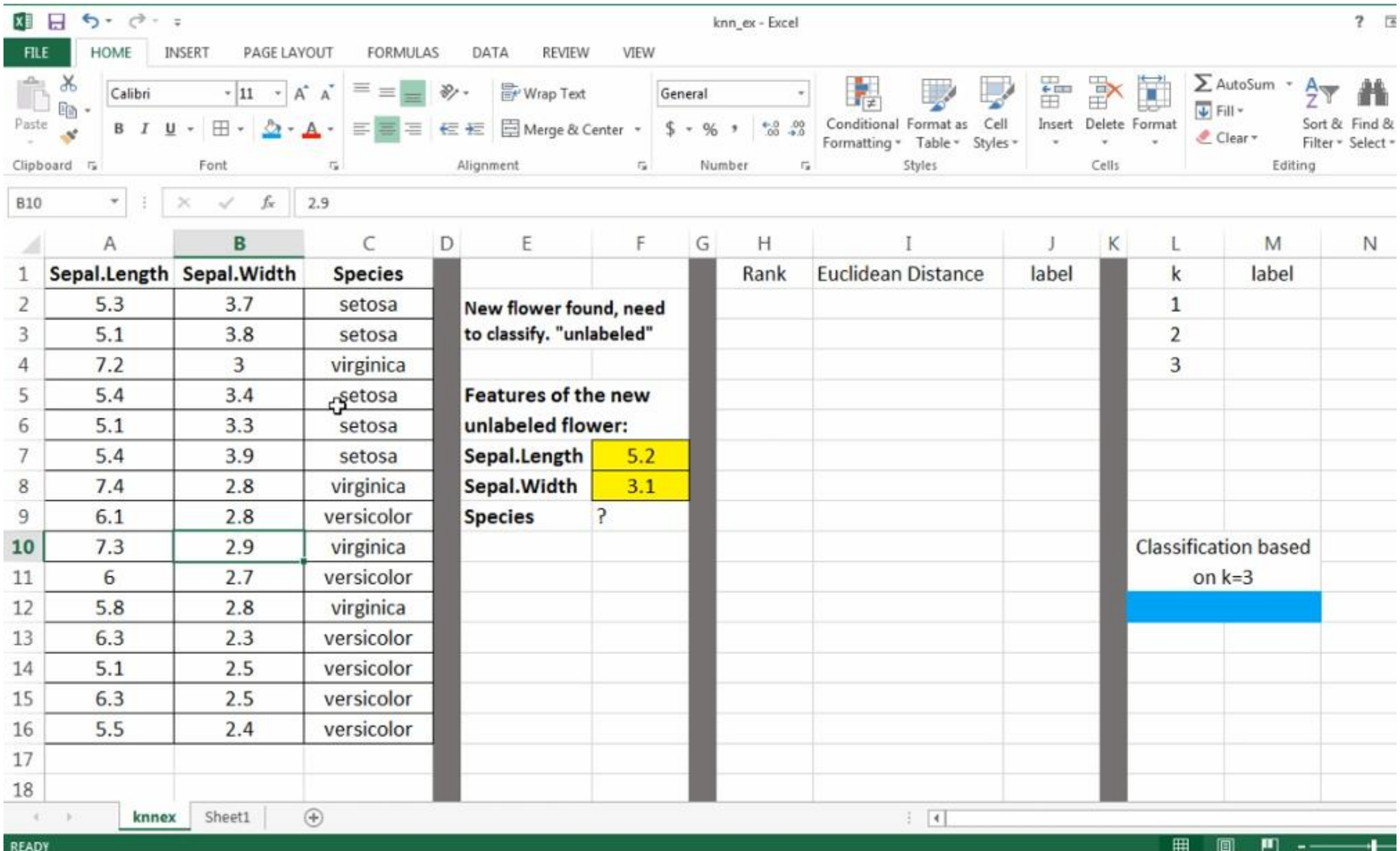
- Machine Learning Algorithm for classification and prediction
- Only three features are used from iris data set just for simplicity

- Sepal length, sepal width and species
- Which one out of three is a class variable?

Training data set

Sepal.Length	Sepal.Width	Species
5.3	3.7	setosa
5.1	3.8	setosa
7.2	3	virginica
5.4	3.4	setosa
5.1	3.3	setosa
5.4	3.9	setosa
7.4	2.8	virginica
6.1	2.8	versicolor
7.3	2.9	virginica
6	2.7	versicolor
5.8	2.8	virginica
6.3	2.3	versicolor
5.1	2.5	versicolor
6.3	2.5	versicolor
5.5	2.4	versicolor

Setup your Excel screen



Sepal.Length	Sepal.Width	Species
5.3	3.7	setosa
5.1	3.8	setosa
7.2	3	virginica
5.4	3.4	setosa
5.1	3.3	setosa
5.4	3.9	setosa
7.4	2.8	virginica
6.1	2.8	versicolor
7.3	2.9	virginica
6	2.7	versicolor
5.8	2.8	virginica
6.3	2.3	versicolor
5.1	2.5	versicolor
6.3	2.5	versicolor
5.5	2.4	versicolor

New flower found, need to classify. "unlabeled"

Features of the new unlabeled flower:

Sepal.Length	5.2
Sepal.Width	3.1
Species	?

Rank	Euclidean Distance	label

k	label
1	
2	
3	

Classification based on k=3

What to find?

- A new flower is found and its sepal length and sepal width is found

- And we want to know the which type of species is this?

Proceed

- As the data is already sorted on the bases of species therefore we have make it random
- Insert a new column and place 150 uniformaly distributed random numbers in it
- As these random # will be change each time when we change any of the cell in excel therefore copy these random numbers and place them as values in a new column

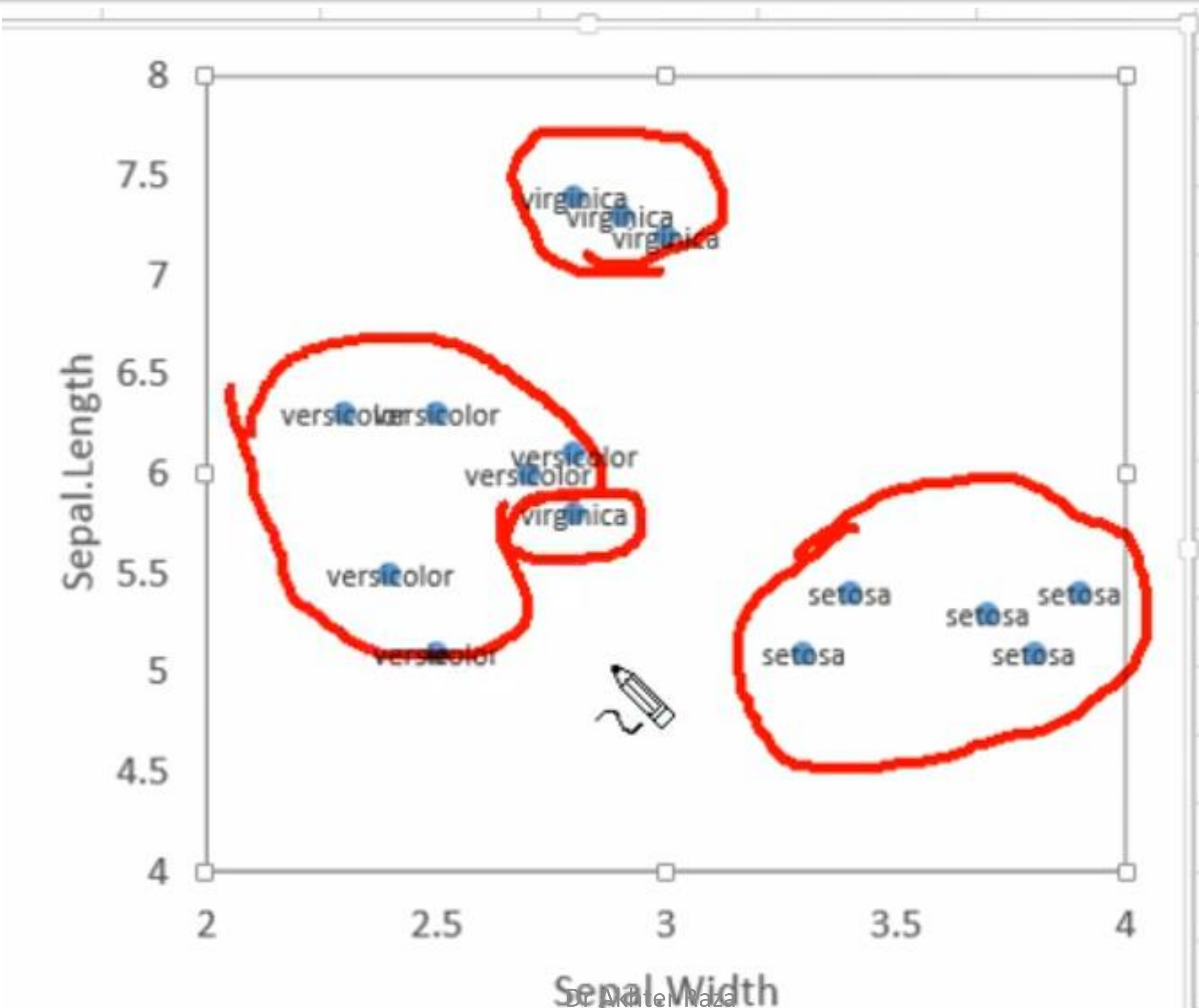
Procedure

- Sort the data using key as random numbers
- This makes the data jumbled w.r.t. species
- This is required for predictions

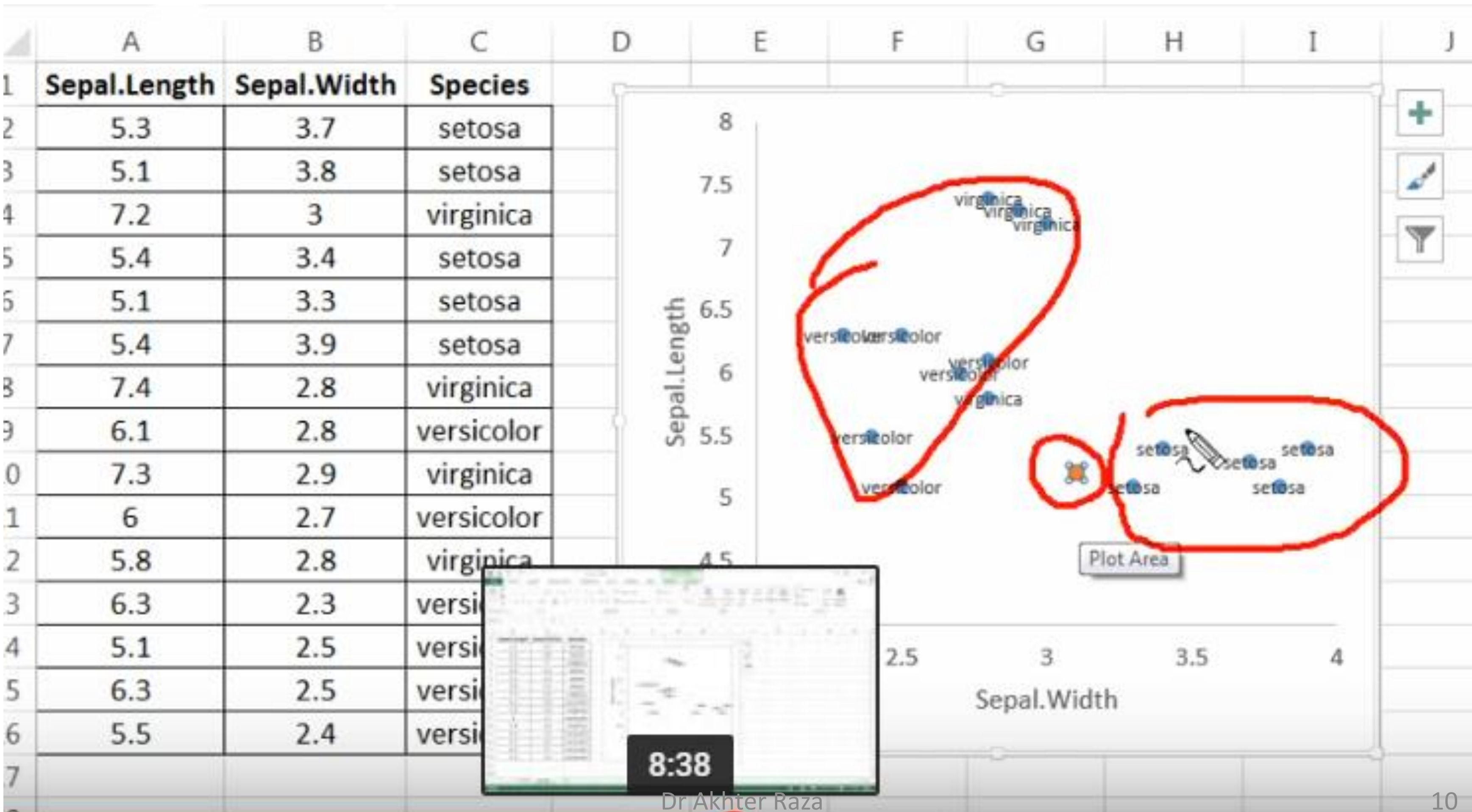
Scatterplot

- Insert→Scatterplot
- Change axis titles
- Change axis min and max ranges
- Change data labels to species name

Scatterplot of a sample of 15 obs



New unknown data point



Sepal.Length	Sepal.Width	Species
5.3	3.7	setosa
5.1	3.8	setosa
7.2	3	virginica
5.4	3.4	setosa
5.1	3.3	setosa
5.4	3.9	setosa
7.4	2.8	virginica
6.1	2.8	versicolor
7.3	2.9	virginica
6	2.7	versicolor
5.8	2.8	virginica
6.3	2.3	versi
5.1	2.5	versi
6.3	2.5	versi
5.5	2.4	versi

Finding distances

- K should be square root of number of data points in the data set
- K=sqrt(15)=3 (approximately)
- Now computing Euclidian distance from the unknown data point to the most three nearest point
- $D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Finding distances

- Now computing all 15 distances from unknown point to each of data point given in sample
- If we have three features then euclidean distance formula will be

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

- Now rank these distances using the excel formula   =rank(which, range, ascending=1)

vlookup in Excel

- Use vlookup function of excel to find the labels of these three closeset data points
- The unknown label will be filled with the majority vote
- It is setosa as it has majority votes

kNN in R

KNN classification algorithm in R

Iris Data Set is used for processing

data(iris)

```
iris_test_target setosa versicolor virginica
      setosa          6          0         0
      versicolor      0          8         2
      virginica       0          0         5
> data(iris)
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : 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 "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
>
```

str(iris)

table(iris$Species)

head(iris)

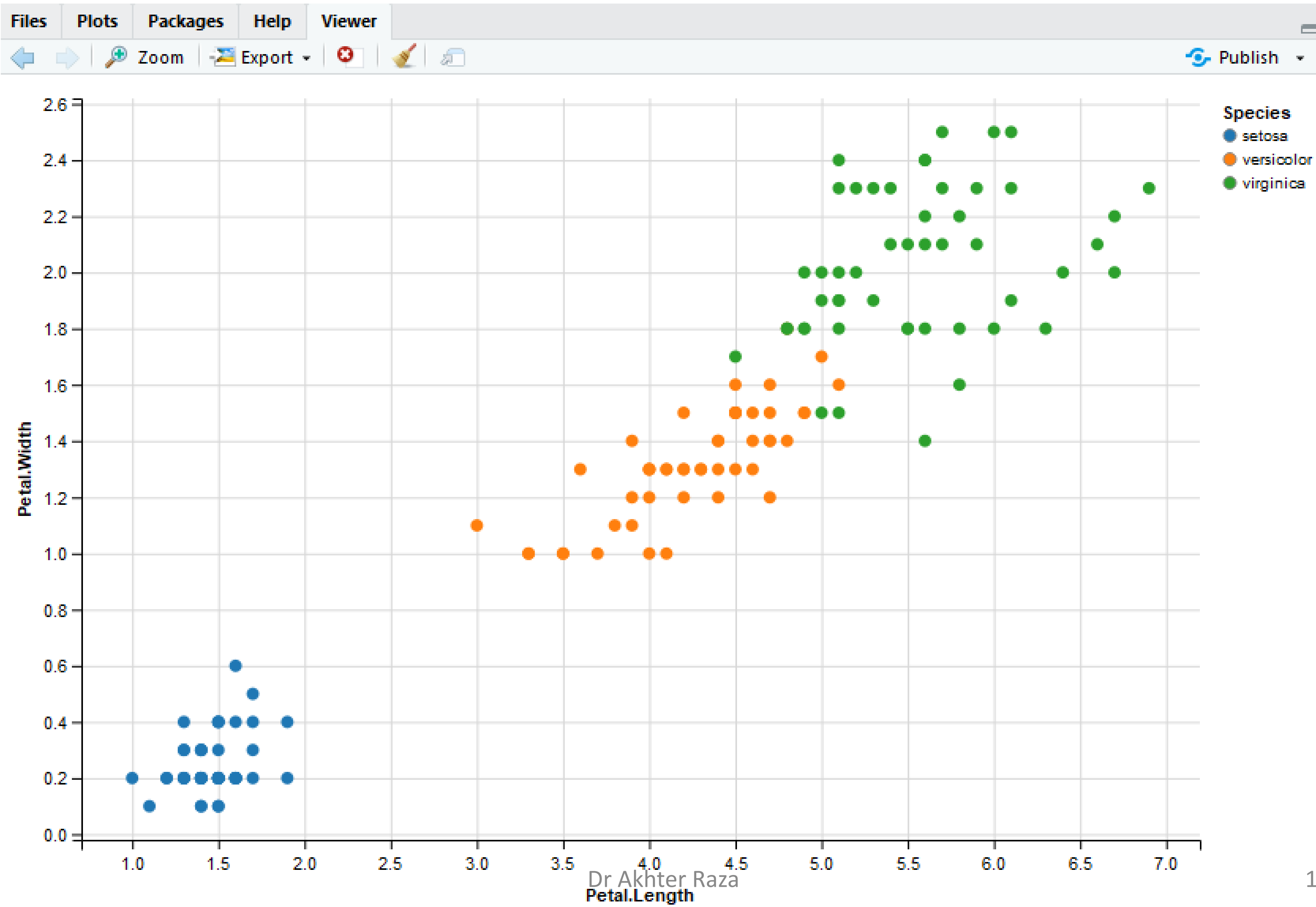
kNN in R

```
#ploting scatterplot of iris data    Load in `ggvis`
library(ggvis)


iris %>% ggvis(~Sepal.Length, ~Sepal.Width, fill =
~Species) %>% layer_points()
iris %>% ggvis(~Petal.Length, ~Petal.Width, fill =
~Species) %>% layer_points()


# Overall correlation `Petal.Length` and `Petal.Width`
cor(iris$Petal.Length, iris$Petal.Width)
```

Scatterplot



Reshuffling the data set

#the data set is organized with respect to the species type we have to mix for better classification

seed(9850)

runif(5)

runif(nrow(iris))

gp<-runif(nrow(iris))

Reshuffling the data set

#the data set is organized with respect to the species type we have to mix for better classification

```
iris2<- iris[order(gp),]

head(iris2)

#Rescaling numerical features

summary(iris2[,c(1,2,3,4)])
```

Normalize the data set

```
#normalize (X-min value)/range convert the data
rescaled from 0 to 1

normalize<-function(x){return((x-min(x))/(max(x)-
min(x)))}

normalize(c(1,2,3,4,5,6))

normalize(c(50,60,70,40,50))
```

Normalize the data set

```
iris_n <-
as.data.frame(lapply(iris2[,c(1,2,3,4)],normalize))

summary(iris_n)
```

Spliting in train and test set

```
# Splitting the train and test data sets

iris_train <- iris_n[1:129,]
iris_test<- iris_n[130:150,]
```

Splitting the train and test set

```
iris_train_target <- iris2[1:129,5]
iris_test_target  <- iris2[130:150,5]
```

Library required for kNN

```
# KNN algorithm is found in class package
# K nearest neighbours
# k=sqrt(datasize)=13 in iris case


require(class)
```

Learning and testing kNN

```
knn_model_1 <- knn(train=iris_train, test=iris_test, cl =
iris_train_target, k=13)
knn_model_1

#checking model accuracy
merge <- data.frame(knn_model_1, iris_test_target)

#confusion matrix of the model accuracy
table(iris_test_target,knn_model_1)
```

Actual v/s predicted data

```
> merge <- data.frame(knn_model_1, iris_test_target)
> merge
   knn_model_1 iris_test_target
1    virginica       versicolor
2       setosa           setosa
3   versicolor       versicolor
4    virginica        virginica
5   versicolor       versicolor
6    virginica        virginica
7    virginica        virginica
8   versicolor       versicolor
9       setosa           setosa
10   virginica        virginica
11   virginica       versicolor
12   virginica        virginica
13  versicolor       versicolor
14  versicolor       versicolor
15  versicolor       versicolor
16      setosa           setosa
17      setosa           setosa
18  versicolor       versicolor
19  versicolor       versicolor
20      setosa           setosa
21      setosa           setosa
```

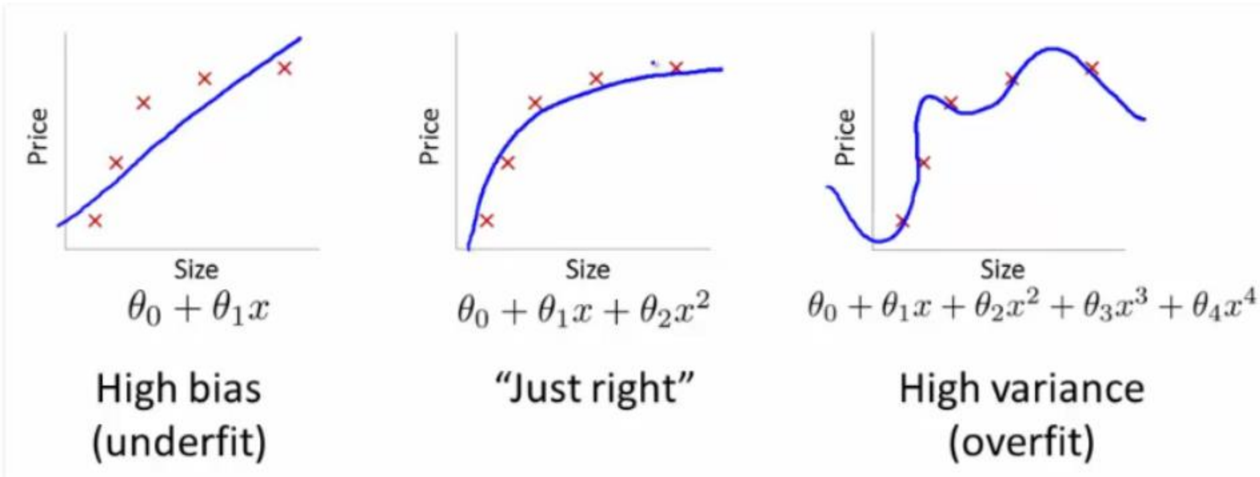
Confusion matrix

	knn_model_1		
iris_test_target	setosa	versicolor	virginica
setosa	6	0	0
versicolor	0	8	2
virginica	0	0	5

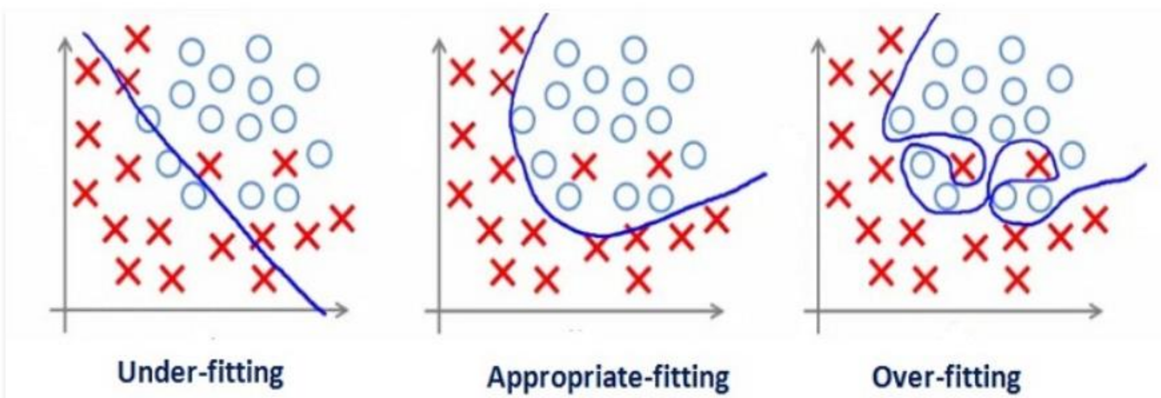
Under fitting

A machine learning algorithm is said to have under fitting when it cannot capture the underlying trend of the data. Under fitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model does not fit the data well enough. It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data.

Under fitting, right fitting and over fitting



Image_source: i.stack.imgur.com/t0zit.png



Over fitting

A model is said to be over fitted, when we train it with a lot of data. When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set. Then the model does not categorize the data correctly, because of too much of details and noise.