Data Science: Capstone - CYO project: Exoplanet Hunting in Deep Space

Introduction

This report describes the analysis of the "Kepler labelled time series data" set as published on Kaggle. According to its description it is largely derived from campaign 3 of the NASA Kepler Mission enriched with data from exoplanets confirmed earlier. As already stated in the title, it is labelled time-series data, describing the observed light intensity (flux) of stars at regular time intervals t (columns 2 - 3198). The data is prepared for the search of star systems with exoplanets by the so-called transit method. Periodic fluctuations in the emitted light indicate the presence of one or more exoplanets orbiting the star. The included label (column 1) is binary with "1" indicating "no exoplanets detected" and "2" "at least one exoplanet". The dataset is subdivided into a training data set of 5087 stars (one line per star) and a validation data set comprising 570 stars. Distinctive frequency regions in the periodograms of the individual time series indicating potential transits where defined and models predicting stars with exoplanets from the data investigated. Only a random forest model yielded true positive results with a sensitivity of 0.4 and specificity of 0.88 on the validation data.

Methods

Data is downloaded from the Kaggle web site (https://www.kaggle.com/keplersmachines/kepler-labelled-time-series-data?select=exoTrain.csv) and imported into R. Data exploration and model development is solely performed on the training data set.

There are 37 stars with exoplanets in the training data resulting in a probability of approx. 0.7 % of picking one at random from the data set. This means the data is highly imbalanced. There are no apparent gaps in the data.

```r
# number of stars with exoplanets in the training data set
sum(train_data$LABEL == 2)
```

```
## [1] 37
```

```r
# probability of picking a star with exoplanets at random from training data
sum(train_data$LABEL == 2)/nrow(train_data)
```

```
## [1] 0.007273442
```

```r
# check if any "NA"s or "-Inf"s are in the data
na_inf_check <- apply(train_data, 2, function(x) any(is.na(x) | is.infinite(x)))
sum(na_inf_check == TRUE)
```

```
## [1] 0
```

Data preparation

The data is presented in a wide format which needs to be converted to a tidy format to be compatible with *ggplot2*. Additionally, the parameter *flux* is standardized, a unique ID for every observed star is introduced and a time interval t is extracted from the column names holding the flux values.

```
# add ID column and time interval column and convert wide to tidy

train_tidy <- train_data %>%
  cbind(id=as.numeric(rownames(.)),.) %>% # add ID
  gather(time,flux, 'FLUX.1':'FLUX.3197') %>% # convert to tidy
  mutate(t=as.numeric(str_extract(time,"\\d{1,4}"))) %>% # extract numeric time interval t
  select(id,LABEL,t,flux) # throw away column names

train_tidy_norm <- train_tidy %>%
  mutate_at(c("flux"), ~(scale(.) %>% as.vector))


# do the same to validation_data

validation_tidy <- validation_data %>%
  cbind(id=rownames(.),.) %>% # add ID
  gather(time,flux, 'FLUX.1':'FLUX.3197') %>% # convert to tidy
  mutate(t=as.numeric(str_extract(time,"\\d{1,4}"))) %>% # extract numeric time interval t
  select(id,LABEL,t,flux) # throw away column names

validation_tidy_norm <- validation_tidy %>%
  mutate_at(c("flux"), ~(scale(.) %>% as.vector))
```

Each three examples for flux time series of stars with and without exoplanets are randomly selected from the training data. Time series before and after centering and rescaling are compared to ensure no artefacts are introduced.

```
# select three stars w/ exoplanets and three w/o exoplanets from train data as examples

# select ids of stars w/ exoplanets
ep <- train_data %>%
  cbind(id=rownames(.),.) %>%
  filter(LABEL==2) %>%
  select(id)

# random select 3 of them

ep_3 <- sample(1:length(ep[,1]), 3)

# select ids of stars w/o exoplanets
no_ep <- train_data %>%
  cbind(id=rownames(.),.) %>%
  filter(LABEL==1) %>%
  select(id)

# random select 3 of them

no_ep_3 <- sample(1:length(no_ep[,1]), 3)

# create example plots

# w/ exoplanets
fig_1 <- train_tidy_norm %>%
```

```
  filter(id %in% ep_3) %>%
  ggplot(.,aes(t,flux)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  theme_bw()

fig_2 <- train_tidy %>%
  filter(id %in% ep_3) %>%
  ggplot(.,aes(t,flux)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  theme_bw()

#w/o exoplanet
fig_3 <- train_tidy_norm %>%
  filter(id %in% no_ep_3) %>%
  ggplot(.,aes(t,flux)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  theme_bw()

fig_4 <- train_tidy %>%
  filter(id %in% no_ep_3) %>%
  ggplot(.,aes(t,flux)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  theme_bw()
```

Feature selection

Raw flux examples plotted against time t are shown in Fig. 1 (stars with exoplanets) and Fig. 2 (stars without exoplanets). It can be observed, that the flux recordings from stars w/ exoplanets demonstrate periodical flux reductions due to the revolutions of the exoplanets around the stars,

```
fig_1
```

```
fig_2
```

```
fig_3
```

```
fig_4
```

As a first try, to separate the two classes of time series, the raw fluxes are analyzed by a principal component analysis. The analysis shows that 99 % of the variance in the populations can be attributed to the first 21 principal components. The first 10 principal components are analyzed in scatter plots (limited to the first 1000 time series).

```
# perform pca (normalized and center data in function, first 50 principal components)
pca_raw_flux <- prcomp(train_data[,2:3198],scale=TRUE,center=TRUE,rank.=50)
summary(pca_raw_flux)
```

```
## Importance of first k=50 (out of 3197) components:
##                            PC1     PC2     PC3      PC4     PC5      PC6
## Standard deviation     29.5826 26.8418 21.8468 17.70289 13.6405 11.41092
## Proportion of Variance  0.2737  0.2254  0.1493  0.09803  0.0582  0.04073
```

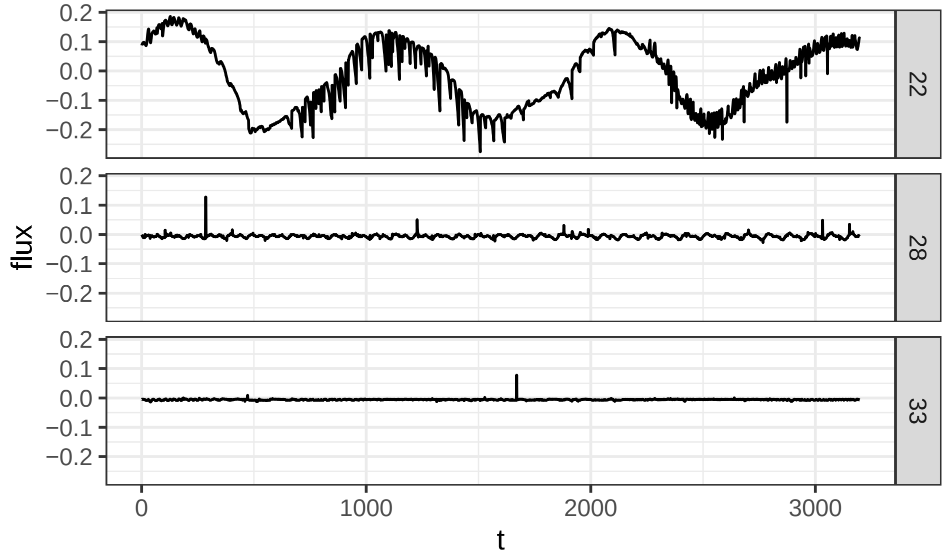

Figure 1: pre-processed flux time series of three stars with exoplanets

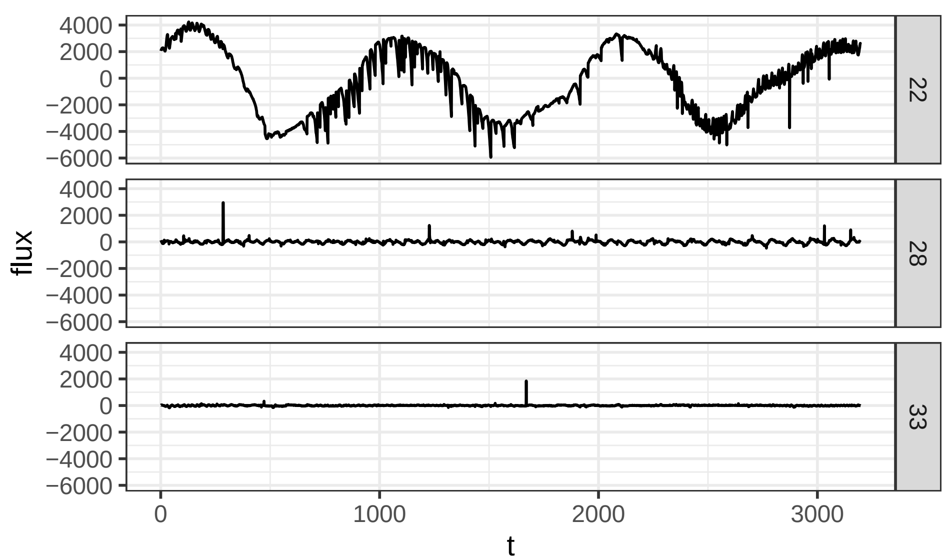

Figure 2: raw flux time series of three stars with exoplanets

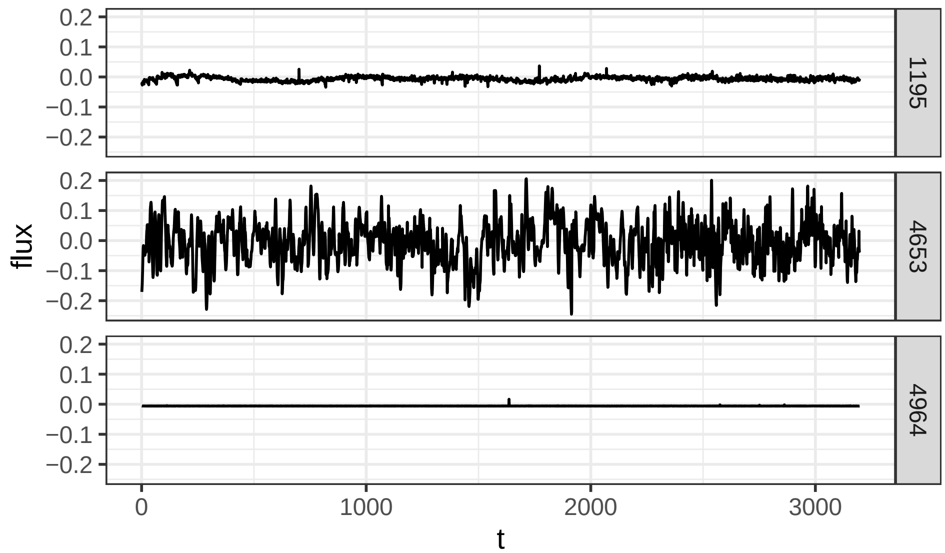

Figure 3: pre-processed flux time series of three stars without exoplanets

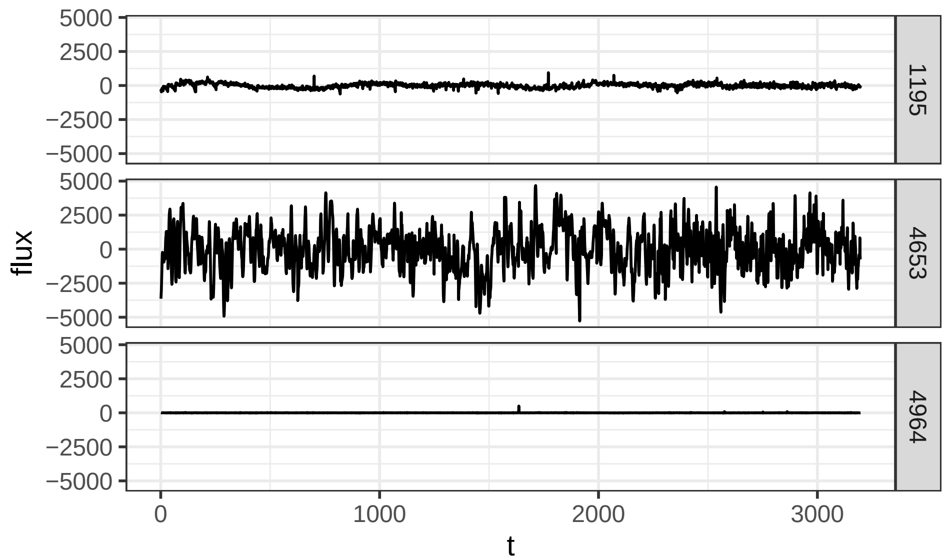

Figure 4: raw flux time series of three stars without exoplanets

```
## Cumulative Proportion   0.2737  0.4991  0.6484  0.74642  0.8046  0.84534
##                            PC7      PC8      PC9    PC10    PC11    PC12   PC13
## Standard deviation     10.29523 9.58261 8.18447 7.6694 6.05913 5.09641 4.7966
## Proportion of Variance  0.03315 0.02872 0.02095 0.0184 0.01148 0.00812 0.0072
## Cumulative Proportion   0.87850 0.90722 0.92817 0.9466 0.95805 0.96618 0.9734
##                           PC14    PC15    PC16   PC17    PC18    PC19    PC20
## Standard deviation     4.53229 3.20156 2.53216 2.3334 2.19964 1.71444 1.62107
## Proportion of Variance 0.00643 0.00321 0.00201 0.0017 0.00151 0.00092 0.00082
## Cumulative Proportion  0.97980 0.98301 0.98501 0.9867 0.98823 0.98915 0.98997
##                           PC21    PC22    PC23    PC24   PC25    PC26    PC27
## Standard deviation     1.57317 1.51420 1.48531 1.45724 1.3874 1.30818 1.28431
## Proportion of Variance 0.00077 0.00072 0.00069 0.00066 0.0006 0.00054 0.00052
## Cumulative Proportion  0.99074 0.99146 0.99215 0.99282 0.9934 0.99395 0.99447
##                           PC28   PC29    PC30   PC31    PC32    PC33    PC34
## Standard deviation     1.15086 1.1239 1.06502 0.9741 0.95249 0.87731 0.86490
## Proportion of Variance 0.00041 0.0004 0.00035 0.0003 0.00028 0.00024 0.00023
## Cumulative Proportion  0.99488 0.9953 0.99563 0.9959 0.99621 0.99645 0.99669
##                          PC35   PC36    PC37    PC38    PC39    PC40    PC41
## Standard deviation     0.8050 0.7926 0.74707 0.70965 0.66296 0.65542 0.62304
## Proportion of Variance 0.0002 0.0002 0.00017 0.00016 0.00014 0.00013 0.00012
## Cumulative Proportion  0.9969 0.9971 0.99726 0.99742 0.99756 0.99769 0.99781
##                           PC42   PC43   PC44    PC45    PC46    PC47    PC48
## Standard deviation     0.59191 0.5688 0.5623 0.54578 0.52839 0.51951 0.50968
## Proportion of Variance 0.00011 0.0001 0.0001 0.00009 0.00009 0.00008 0.00008
## Cumulative Proportion  0.99792 0.9980 0.9981 0.99822 0.99830 0.99839 0.99847
##                           PC49    PC50
## Standard deviation     0.48568 0.47558
## Proportion of Variance 0.00007 0.00007
## Cumulative Proportion  0.99854 0.99861
```

```r
# create scatter plots of first 10 principal components,
#limit data to first 1000 rows (planet w/ exoplanets are in the first 37 rows)
fig_5 <- data.frame(PC1 = pca_raw_flux$x[1:1000,1],
                    PC2 = pca_raw_flux$x[1:1000,2],
                    label=train_data[1:1000,1]) %>%
  ggplot(aes(PC1, PC2, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_6 <- data.frame(PC3 = pca_raw_flux$x[1:1000,3],
                    PC4 = pca_raw_flux$x[1:1000,4],
                    label=train_data[1:1000,1]) %>%
  ggplot(aes(PC3, PC4, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_7 <- data.frame(PC5 = pca_raw_flux$x[1:1000,5],
                    PC6 = pca_raw_flux$x[1:1000,6],
                    label=train_data[1:1000,1]) %>%
  ggplot(aes(PC5, PC6, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_8 <- data.frame(PC7 = pca_raw_flux$x[1:1000,7],
                    PC8 = pca_raw_flux$x[1:1000,8],
                    label=train_data[1:1000,1]) %>%
  ggplot(aes(PC7, PC8, fill=as.factor(LABEL)))+
```

```
  geom_point(cex=3, pch=21)

fig_9 <- data.frame(PC9 = pca_raw_flux$x[1:1000,9],
                    PC10 = pca_raw_flux$x[1:1000,10],
                    label=train_data[1:1000,1]) %>%
  ggplot(aes(PC9, PC10, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)
```

The scatter plots (Fig. 5 - Fig. 9) show that stars w/ exoplanets cannot be distinguished from the principal components of the raw fluxes of the first 1000 time series (visual assessment).

```
fig_5
```

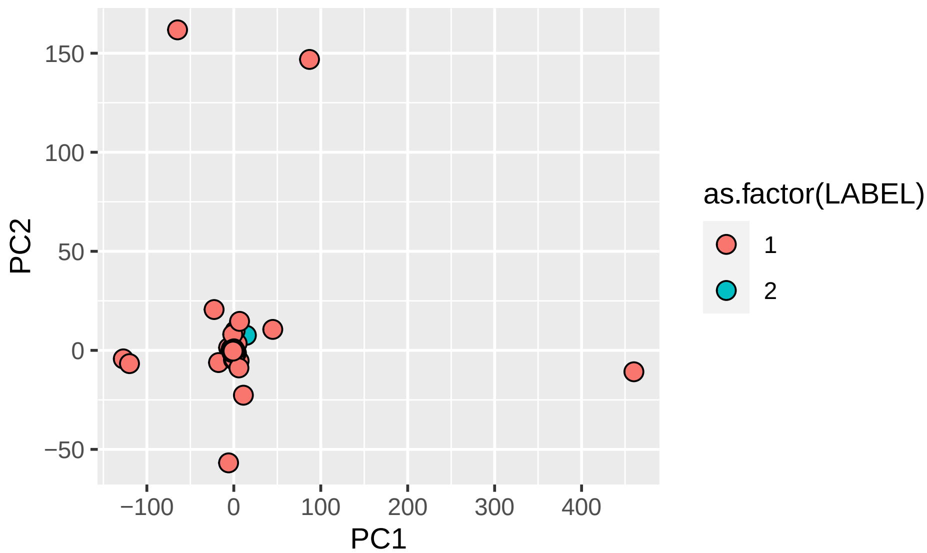

Figure 5: PC1 vs. PC2

```
fig_6
```

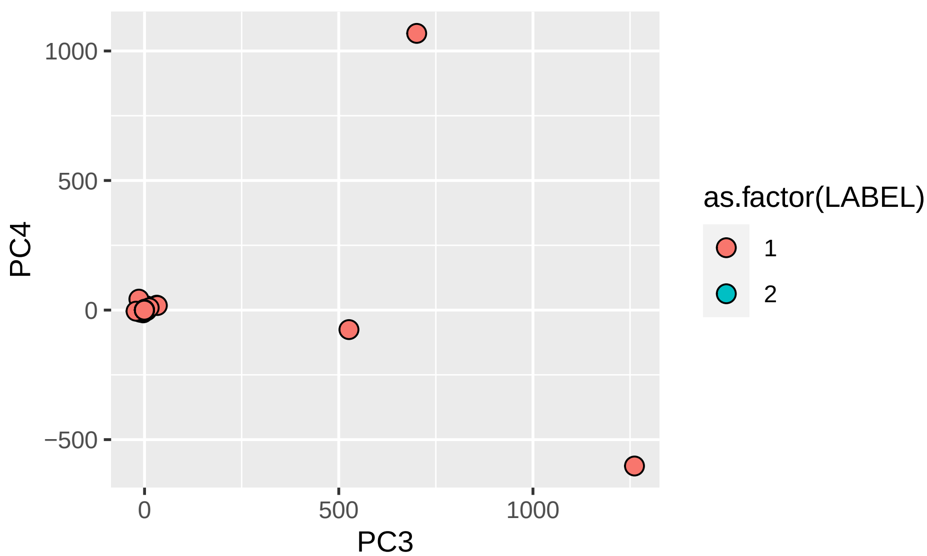

Figure 6: PC3 vs. PC4

```
fig_7
```

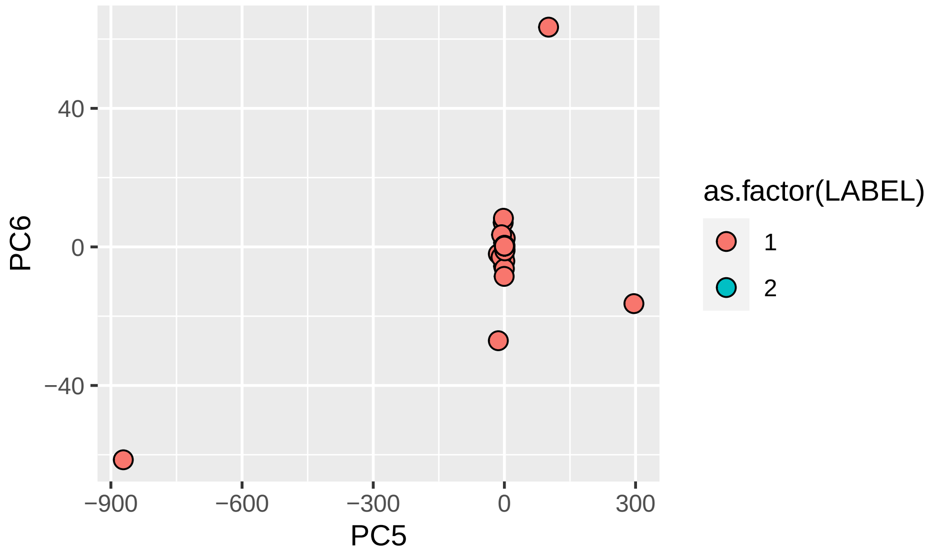

Figure 7: PC5 vs. PC6

```
fig_8
```

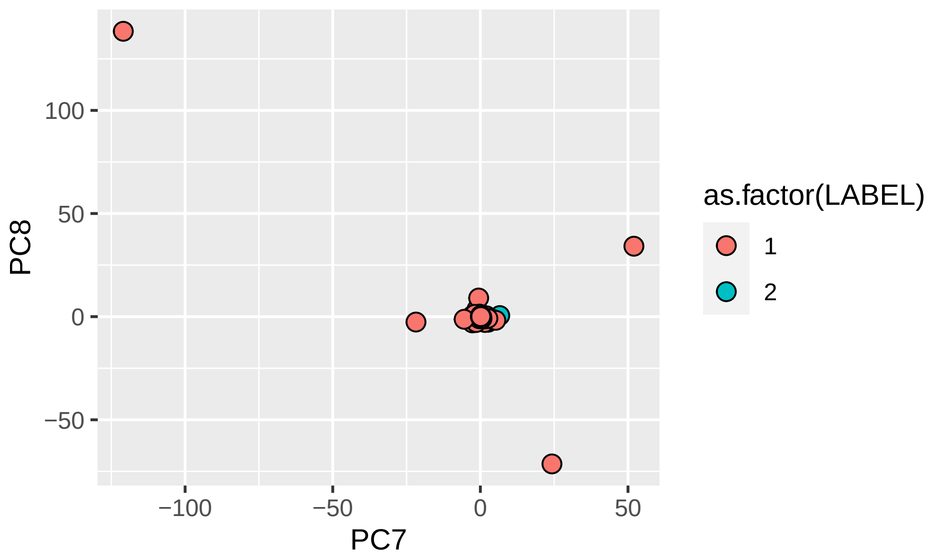

Figure 8: PC7 vs. PC8

```
fig_9
```

As a next step, the raw flux time series are converted to periodograms to investigate the periodic changes in light fluxes which are caused by the periodic movement of the planets around their stars. For this, the time series are fourier-transformed (fast fourier transformation methodology) and the intensities calculated from the square amplitudes.

```
# calculation of periodograms
train_ps <- matrix(ncol = 1599, nrow = 0) #create empty ps matrix
```

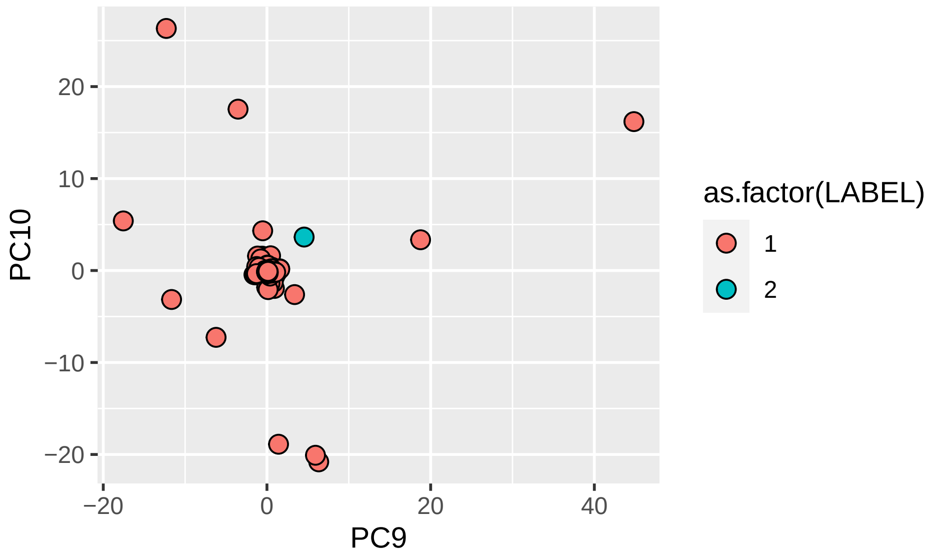

Figure 9: PC9 vs. PC 10

```
for (i in 1:5087) { # extract time series row by row
  temp<-as.matrix(train_data[i,2:3197])
  ints2 <- abs(fft(temp))^2/3196 # calculate intensities from squared amplitudes
  scaled_ints <- (4/3196)*ints2[1:1599] # re-scale
  train_ps <- rbind(train_ps,scaled_ints) # create ps matrix
}

# rename rows and columns
rownames(train_ps)<- seq(1,5087,1)
colnames(train_ps)<- seq(1,1599,1)

# min-max scaling of each individual periodogram
train_ps_centered <- sweep(train_ps, 1, apply(train_ps,1,min))
train_ps_standardized <- sweep(train_ps_centered, 1, apply(train_ps,1,max), FUN = "/")

# add ID column, labels, and frequency column and convert wide to tidy

train_ps_tidy <- train_ps_standardized %>%
  as_tibble() %>%
  cbind(select(train_data,LABEL),.) %>% # add labels
  cbind(id=as.numeric(rownames(.)),.) %>% # add ID
  gather(key,ints, "1":"1599") %>% # convert to tidy
  mutate(f=(as.numeric(key)-1)/3196) %>% # add frequency
  select(id,LABEL,f,ints) # throw away column names

# extract periodograms for examples in fig. 1 and fig. 3
# w/ exoplanets
fig_10 <- train_ps_tidy %>%
  filter(id %in% ep_3) %>%
  ggplot(.,aes(f,ints)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
```

```
  labs(x="frequency",y="intensity") +
  theme_bw()

# scale x to log10 to emphasize low-frequencies in visualization

fig_11 <- train_ps_tidy %>%
  filter(id %in% ep_3) %>%
  ggplot(.,aes(f,ints)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  labs(x="frequency",y="intensity") +
  scale_x_log10() +
  theme_bw()

#w/o exoplanet
fig_12 <- train_ps_tidy %>%
  filter(id %in% no_ep_3) %>%
  ggplot(.,aes(f,ints)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  labs(x="frequency",y="intensity") +
  theme_bw()

# scale x to log10 to emphasize low-frequencies in visualization
fig_13 <- train_ps_tidy %>%
  filter(id %in% no_ep_3) %>%
  ggplot(.,aes(f,ints)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  labs(x="frequency",y="intensity") +
  scale_x_log10() +
  theme_bw()
```

In the periodograms the periodic fluctuations become more clear. Especially, in Fig. 11 and Fig. 13 the logarithmic frequency represention shows, that the periodograms of stars with exoplanets demonstrate differences in the low frequencies lower than approx. 0.1 when compared to those of stars without exoplanets.

```
fig_10
```

```
fig_11
```

```
fig_12
```

```
fig_13
```

To enhance the differences between periodograms a filter mask is created based on the difference between the average periodogram of stars with exoplanets and the average periodogram of stars without exoplanets. The difference is additionally smoothed using *ksmooth* and a bandwith of $f = 63/3196$ (empirically determined).

```
# make empty matrix to hold filters
filters <-matrix(ncol = 1599, nrow = 0)

f <- 0:1598/3196

# calculate the average periodogram for a star w/ exoplanets
```

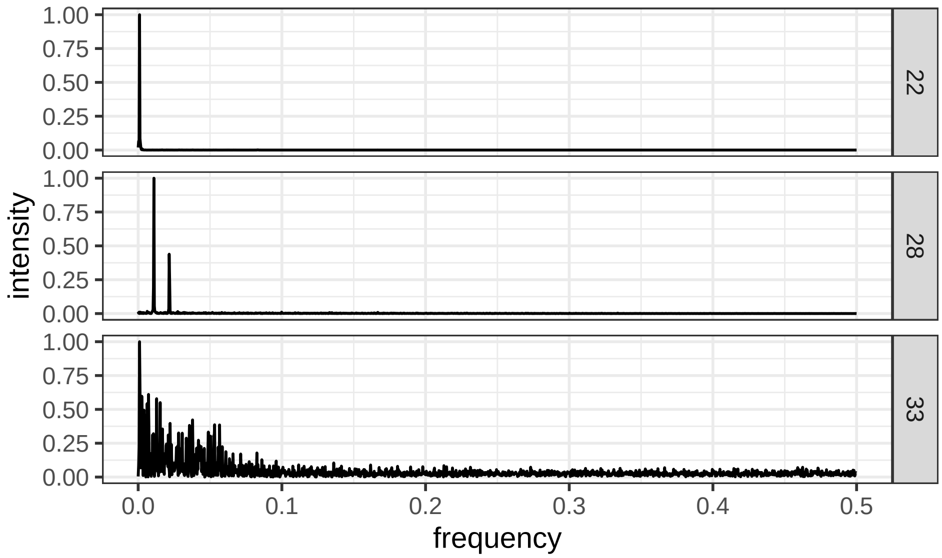

Figure 10: scaled periograms of three stars with exoplanets

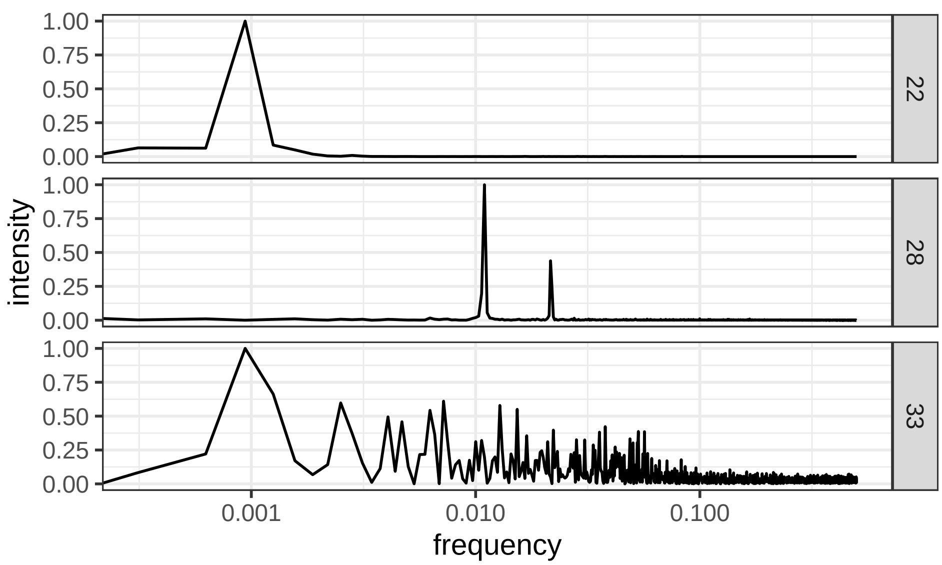

Figure 11: scaled periograms of three stars with exoplanets (frequencies on log10 scale)

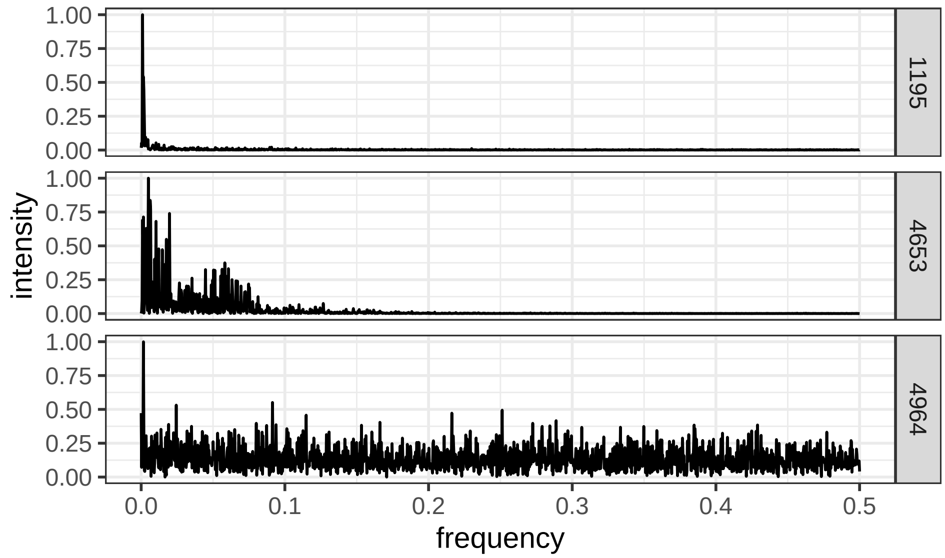

Figure 12: scaled power spectra of three stars without exoplanets

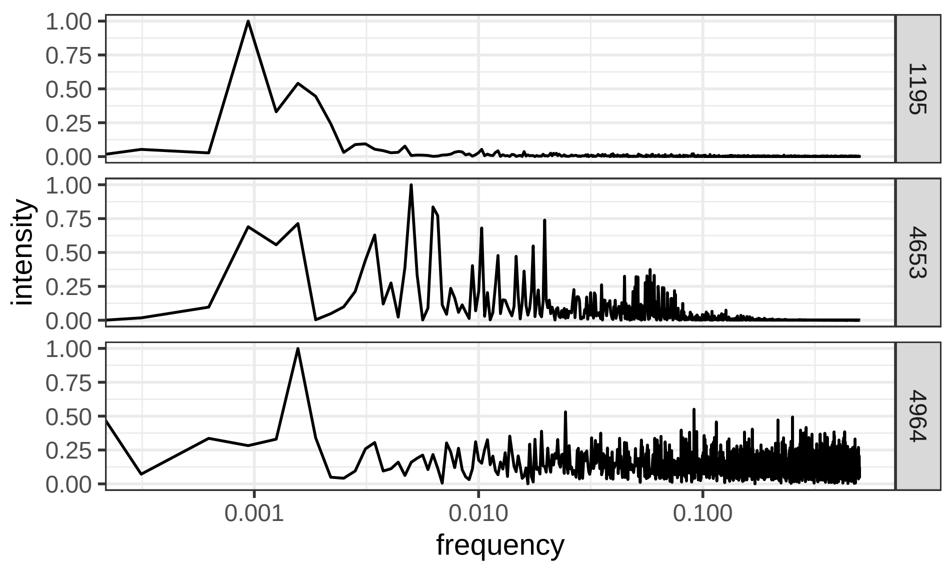

Figure 13: scaled power spectra of three stars without exoplanets (frequencies on log10 scale)

```r
avg_ep <- train_ps_standardized[1:37,] %>%
  colMeans()

auc1 <- trapz(f,avg_ep)

avg_ep_norm <- avg_ep/auc1

filters <- rbind(filters,avg_ep_norm)

# calculate the average periodogram for a star w/ exoplanets

avg_no_ep <-  train_ps_standardized[38:5087,] %>%
  colMeans()

auc2 <- trapz(f,avg_no_ep)

avg_no_ep_norm <- avg_no_ep/auc2

filters <- rbind(filters,avg_no_ep_norm)

# calculate the difference between the two average and take the square amplitudes

diff <- (avg_no_ep-avg_ep)^2

auc3 <- trapz(f,diff)

diff_norm <- diff/auc3

filters <- rbind(filters,diff_norm)

# convert to tidy for visualization

filters_tidy <- filters %>%
  as_tibble() %>%
  cbind(name=c("avg_ep","avg_no_ep","diff"),.) %>%
  gather(key,ints,"1":"1599") %>% # convert to tidy
  mutate(f=(as.numeric(key)-1)/3196) %>% # add frequency
  select(-key)

# make figures for averages and diff periodograms

fig_14 <- filters_tidy %>%
  ggplot(aes(x=f,y=ints)) +
  geom_line()+
  facet_grid(rows=vars(name)) +
  labs(x="frequency",y="intensity") +
  theme_bw()

# create filter mask
sm <- with(filter(filters_tidy,name=="diff"),ksmooth(f,ints,kernel="normal",bandwidth = 63/3196))

filter_norm <- sm$y/trapz(f,sm$y)
```

```r
# make overlay of diff and filter

fig_15 <-filter(filters_tidy,name=="diff") %>% mutate(sm_ints=filter_norm) %>%
  ggplot(aes(x=f,y=ints)) +
  geom_line()+
  geom_line(aes(f,sm_ints),color="red") +
  labs(x="frequency",y="intensity") +
  annotate(geom ="text",x=0.04,y=50,label="region I") +
  annotate(geom ="text",x=0.1,y=50,label="region II") +
  annotate(geom ="text",x=0.18,y=50,label="region III") +
  annotate(geom ="text",x=0.27,y=50,label="region IV") +
  scale_y_log10() +
  theme_bw()
```

Fig. 14 shows the respective average periodograms of the two classes and the difference periograms. Inspecting the difference periogram shows the biggest differences between the classes are at frequencies below 0.1. Overlay of the difference and filter mask (Fig. 15) follows the main peak regions I- IV of the difference periodograms very well while attenuating the other regions.

```r
fig_14
```

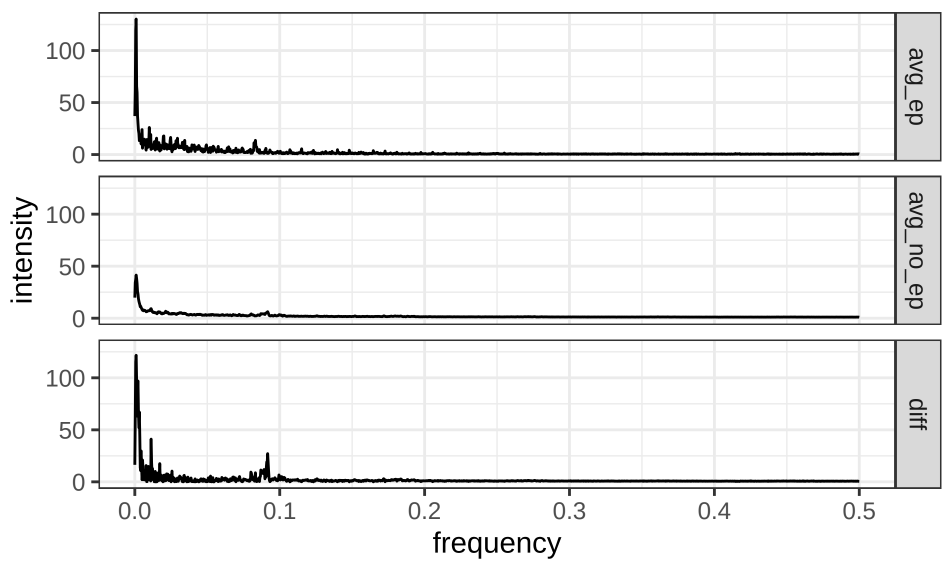

Figure 14: average periodograms of stars with and without exoplanets and difference periodogram

```r
fig_15
```

The filter mask is applied to the individual periodograms by multiplication.

```r
# filter

train_ps_filt <- sweep(train_ps_standardized, 1, filter_norm, FUN = "*")

# convert to tidy

train_ps_filt_tidy <- train_ps_filt%>%
  as_tibble() %>%
  cbind(select(train_data,LABEL),.) %>% # add labels
  cbind(id=as.numeric(rownames(.)),.) %>% # add ID
```

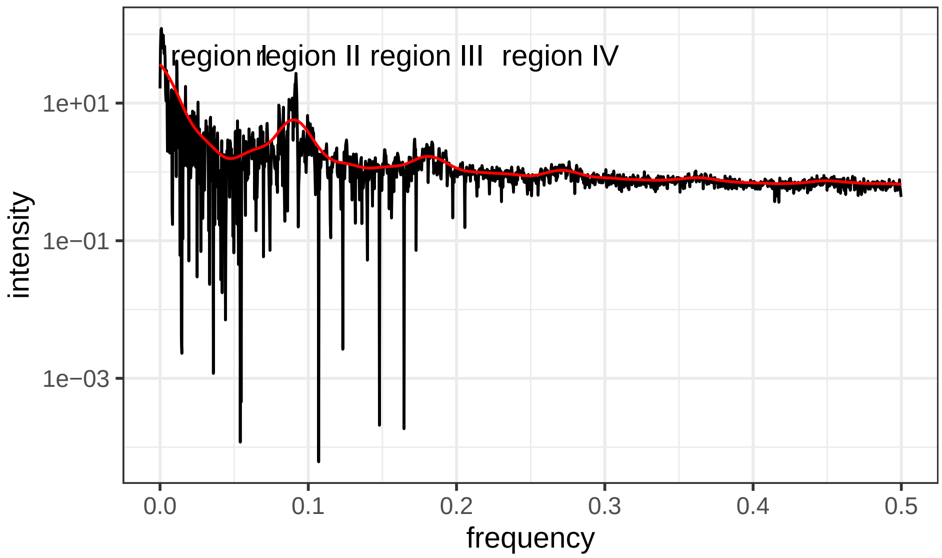

Figure 15: overlay of difference periodogram and filter mask (red)

```
  gather(key,ints, "1":"1599") %>% # convert to tidy
  mutate(f=(as.numeric(key)-1)/3196) %>% # add frequency
  select(id,LABEL,f,ints) # throw away column names

# extract periodograms for examples in fig. 1 and fig. 3
# w/ exoplanets
fig_16 <- train_ps_filt_tidy %>%
  filter(id %in% ep_3) %>%
  ggplot(.,aes(f,ints)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  labs(x="frequency",y="intensity") +
  theme_bw()

#w/o exoplanet
fig_17 <- train_ps_filt_tidy %>%
  filter(id %in% no_ep_3) %>%
  ggplot(.,aes(f,ints)) +
  geom_line() +
  facet_grid(rows=vars(id)) +
  labs(x="frequency",y="intensity") +
  theme_bw()
```

Fig. 16 and Fig. 17 show the filtered periodograms of the examples of Fig. 1 and Fig. 3.

```
fig_16
```

```
fig_17
```

The possibility to distinguish the two classes based on their frequency composition is further explored by PCA. Filtered and un-filtered periodograms are compared head to head, to see whether the filter really enhances class separation. Frequencies are limited to a maximum frequency of *0.3 ~ 959/3196* as variations of interest are within this range (see Fig. 15).

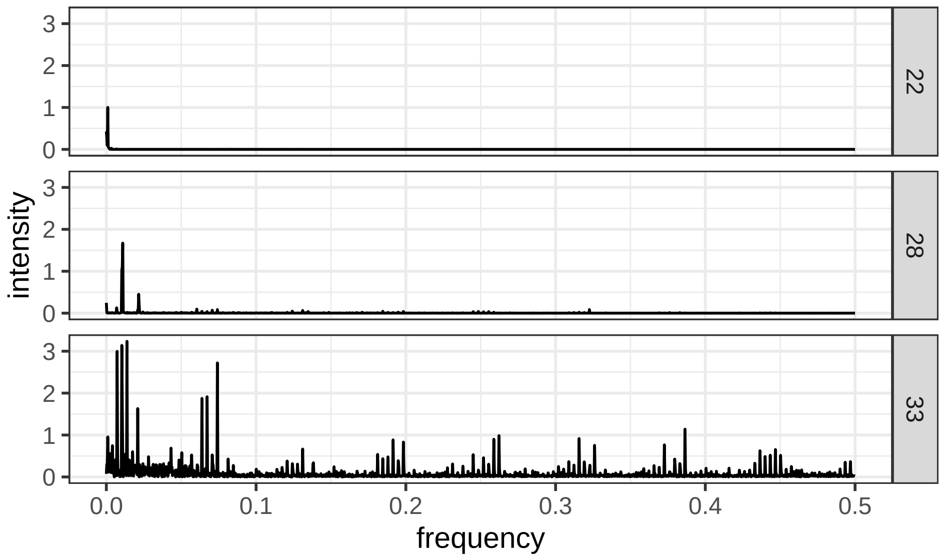

Figure 16: filtered periodograms of three stars with exoplanets

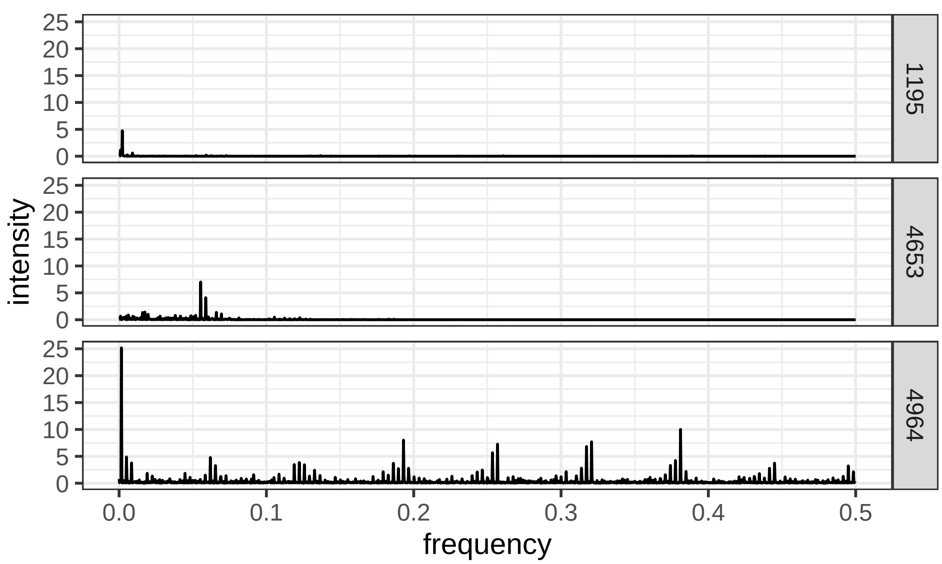

Figure 17: filtered periodograms of three stars without exoplanets

```r
# perform pca (normalized and center data in function,)
pca_ps <- prcomp(train_ps_standardized[,1:959],scale=FALSE,center=FALSE)
pca_ps_filt <- prcomp(train_ps_filt[,1:959],scale=FALSE,center=FALSE)

# extract cumulative proportion of variance from summarys

untreated <- summary(pca_ps)$importance[3,]

filtered <- summary(pca_ps_filt)$importance[3,]

fig_18 <- cbind(untreated,filtered) %>%
  as_tibble() %>%
  ggplot(aes(x=1:959,y=untreated)) +
  geom_line()+
  geom_line(aes(x=1:959,y=filtered,color="red"))+
  labs(x="# principal components",y="cumulative proportion of explained variability") +
  theme_bw() +
  theme(legend.position = "none")

# create scatter plots of first 10 principal components,
#limit data to first 1000 rows (planet w/ exoplanets are in the first 37 rows)
fig_19 <- data.frame(PC1 = pca_ps_filt$x[1:1000,1],
                     PC2 = pca_ps_filt$x[1:1000,2],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC1, PC2, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_20 <- data.frame(PC1 = pca_ps$x[1:1000,1],
                     PC2 = pca_ps$x[1:100,2],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC1, PC2, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_21 <- data.frame(PC3 = pca_ps_filt$x[1:1000,3],
                     PC4 = pca_ps_filt$x[1:1000,4],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC3, PC4, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_22 <- data.frame(PC3 = pca_ps$x[1:1000,3],
                     PC4 = pca_ps$x[1:1000,4],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC3, PC4, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_23 <- data.frame(PC5 = pca_ps_filt$x[1:1000,5],
                     PC6 = pca_ps_filt$x[1:1000,6],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC5, PC6, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_24 <- data.frame(PC5 = pca_ps$x[1:1000,5],
                     PC6 = pca_ps$x[1:1000,6],
```

```r
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC5, PC6, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_25 <- data.frame(PC7 = pca_ps_filt$x[1:1000,7],
                     PC8 = pca_ps_filt$x[1:1000,8],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC7, PC8, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_26 <- data.frame(PC7 = pca_ps$x[1:1000,7],
                     PC8 = pca_ps$x[1:1000,8],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC7, PC8, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_27 <- data.frame(PC9 = pca_ps_filt$x[1:1000,9],
                     PC10 = pca_ps_filt$x[1:1000,10],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC9, PC10, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)

fig_28 <- data.frame(PC9 = pca_ps$x[1:1000,9],
                     PC10 = pca_ps$x[1:1000,10],
                     label=train_data[1:1000,1]) %>%
  ggplot(aes(PC9, PC10, fill=as.factor(LABEL)))+
  geom_point(cex=3, pch=21)
```

Comparing the contribution of the principal components in the untreated and in the filtered dataset shows that the variability per principal component is more evenly distributed in the filtered dataset among the first 500 principal components (Fig. 18). Comparing the scatter plots of the first 10 principal components in the filtered dataset (Fig. 19, 21, 23, 25 and 27) and the untreated dataset (Fig. 20, 22, 24, 26 and 28) further demonstrates that separation of classes is better in the filtered dataset. Therefore the values of the principal components of the filtered periodograms are used as input for machine learning.

```r
fig_18
```

```r
fig_19
```

```r
fig_20
```

```r
fig_21
```

```r
fig_22
```

```r
fig_23
```

```r
fig_24
```

```r
fig_25
```

```r
fig_26
```

```r
fig_27
```

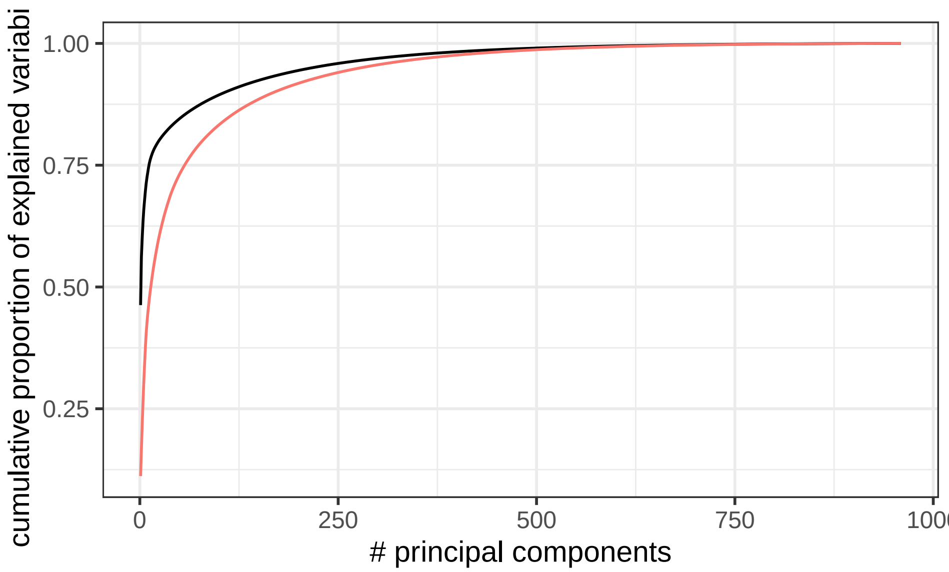

Figure 18: Comparison of explained variability per principal component in the untreated (black) and in the untreated (red) dataset

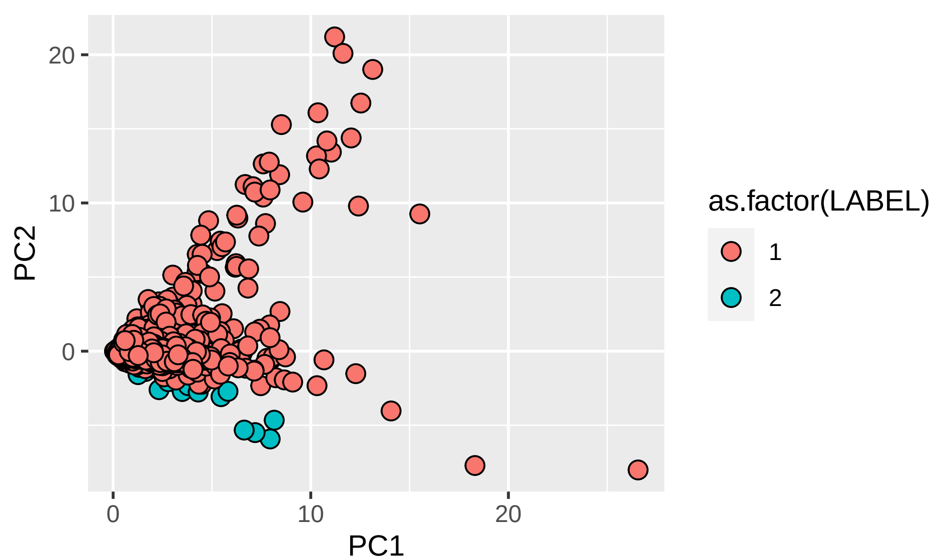

Figure 19: PC1 vs. PC2

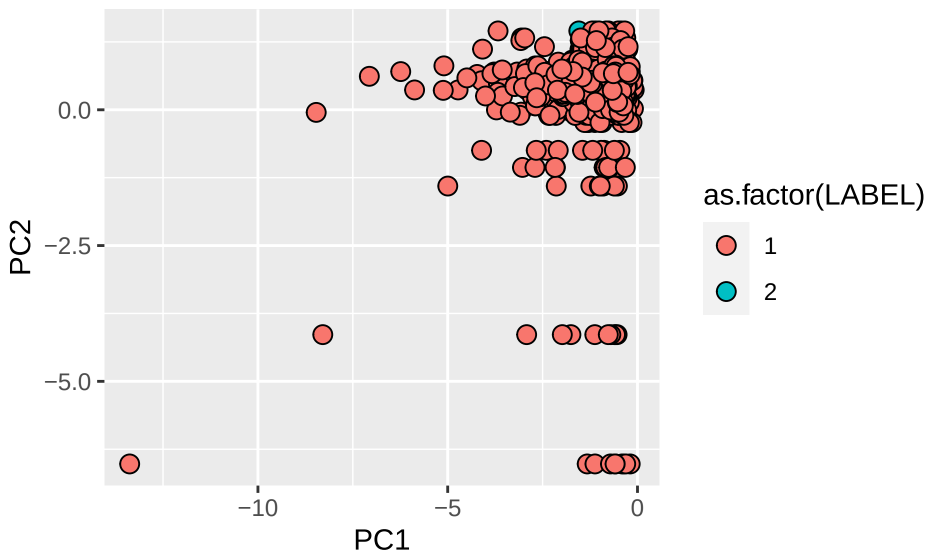

Figure 20: PC1 vs. PC2 (untreated)

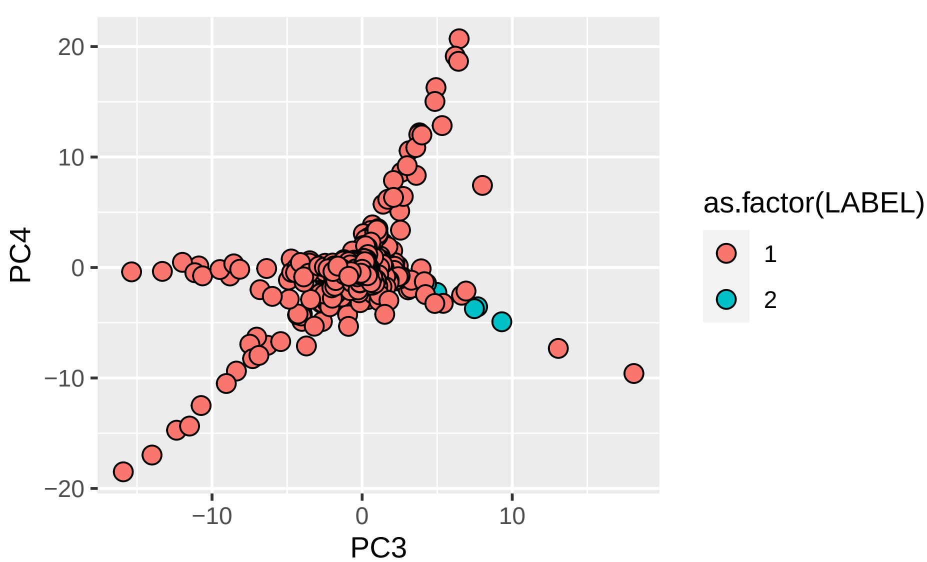

Figure 21: PC3 vs. PC4

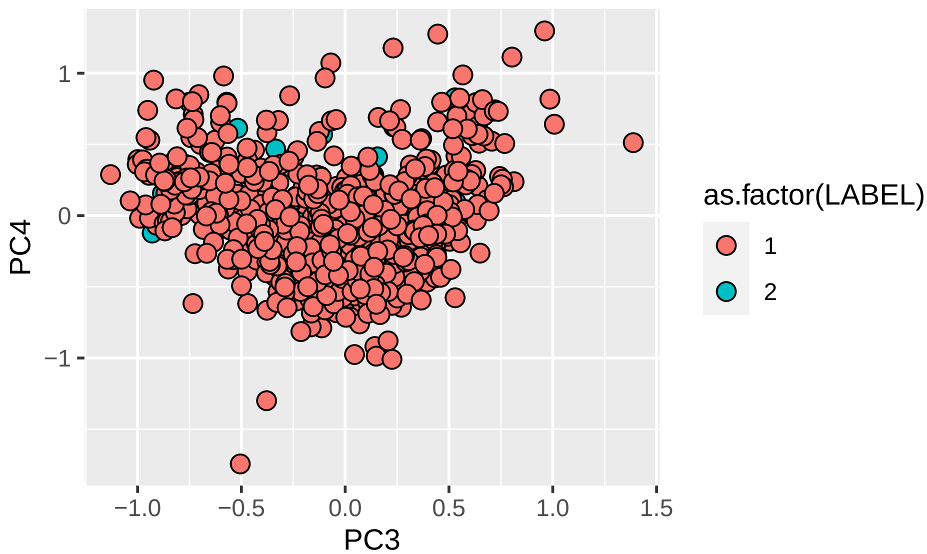

Figure 22: PC3 vs. PC4 (untreated)

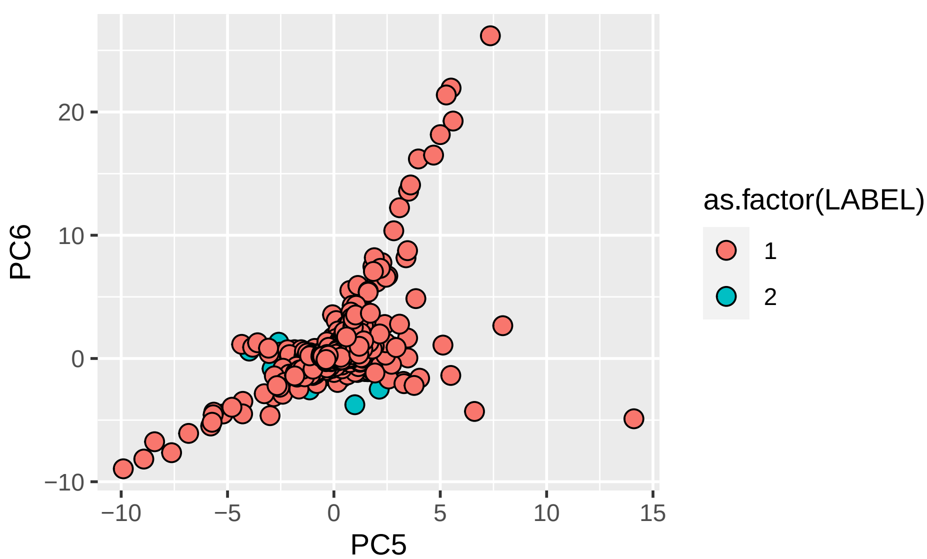

Figure 23: PC5 vs. PC6

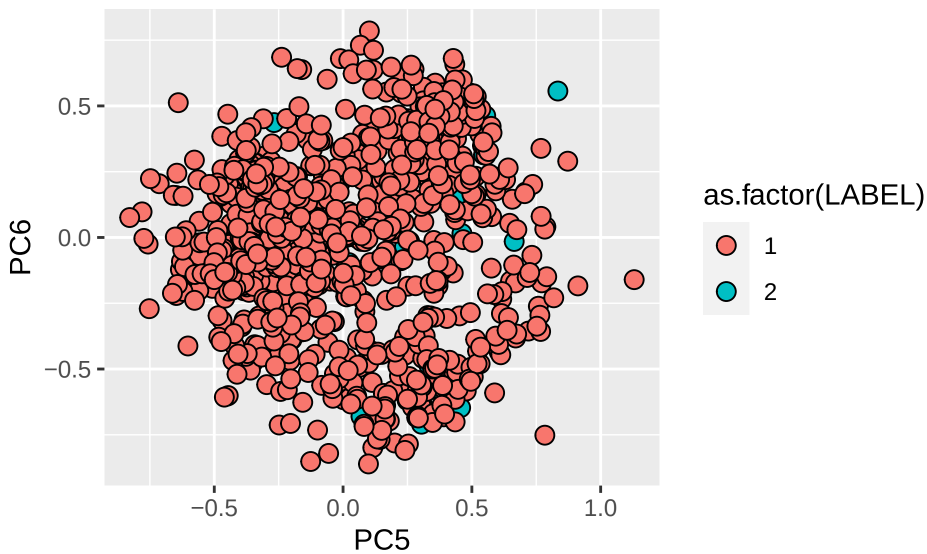

Figure 24: PC5 vs. PC6 (untreated)

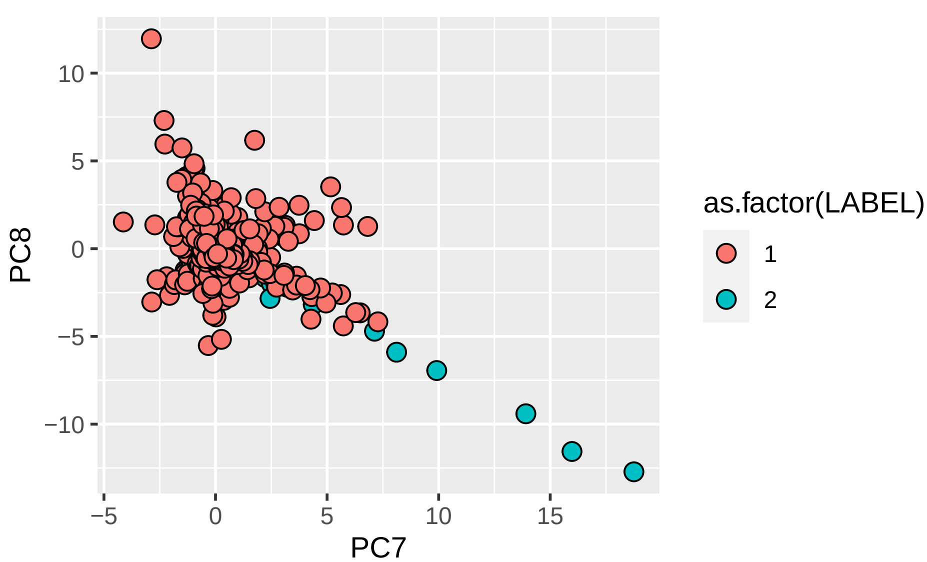

Figure 25: PC7 vs. PC8

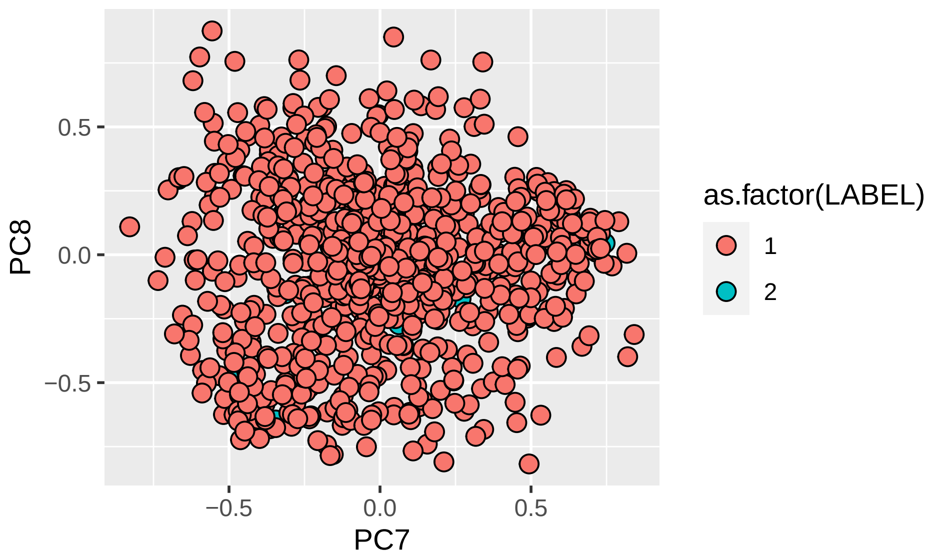

Figure 26: PC7 vs. PC8 (untreated)

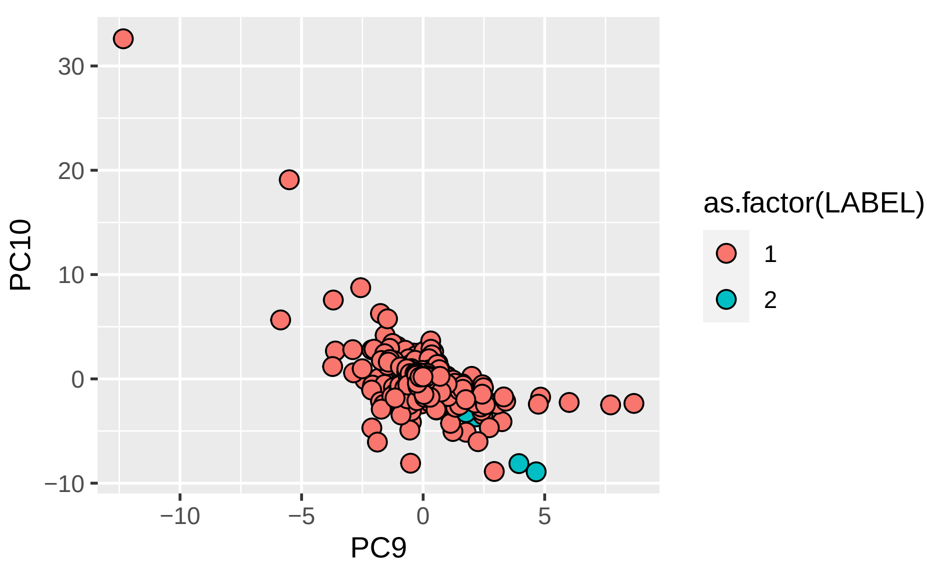

Figure 27: PC9 vs. PC 10

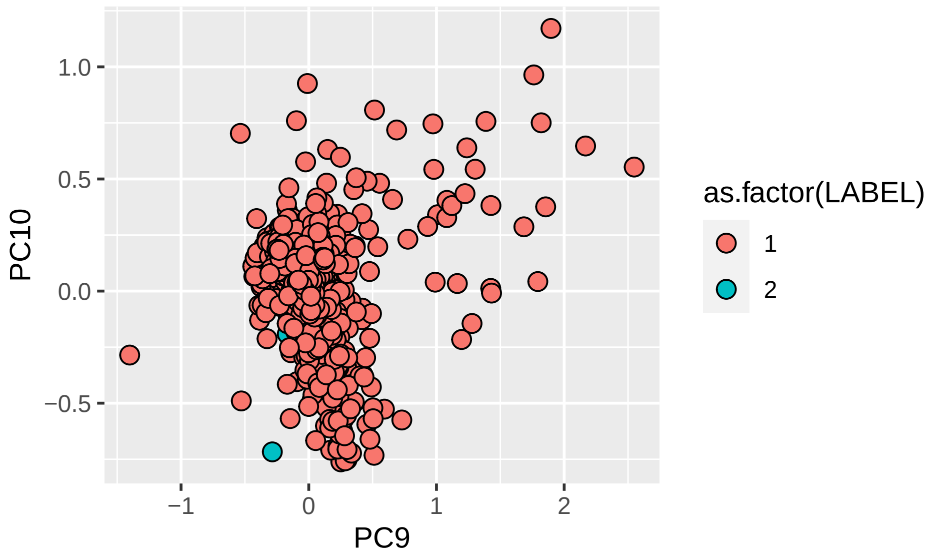

Figure 28: PC9 vs. PC 10 (untreated)

```
fig_28
```

Classification methods

The main issue in this classification task is the very pronounced class imbalance. Therefore random forest, a support vector machine, and a k-nearest neighbor algorithm are combined with randomized under-sampling (number of samples from majority class is adjusted to minority class) ([1], [2]). Over-sampling and class-weighting is omitted due to high computational costs . Additionally, a one-class support vector machine predictor for stars with exoplanets is investigated as the few representatives of stars with exoplanets can be basically considered as an anomaly in the dataset. The support vector machine is employed with a radial kernel function. While some of the scatter plots shown above point towards linear relationships, it is not assumed for the complete data set. Since classification accuracy is misleading when used to evaluate this task (the classification would 99% accurate when assigning all time series to stars without exoplanets), the area under the receiver operator curve is used as metric for tuning. Here sensitivity and specificity of the classification task are evaluated. Class labels are adjusted to conform with the requirements of the caret package: the positive class "2" of stars with exoplanets becomes "X1", while the negative class "1" is defined as "X2"

Down-sampling

The code to train the models using down-sampling is shown below. Tuning hyper parameters is performed using 10-fold cross-validation in 5 repeats because when the dataset is undersampled, not all examples of stars without exoplanets will be included in the actual training set.

```
## define the settings for cross-validation.
ctrl_pars <- trainControl(method="repeatedcv",
                          classProbs = TRUE ,
                          number=10, repeats = 5,
                          summaryFunction = twoClassSummary,
                          sampling = "down")
## define training inputs
```

```
TrainX <- pca_ps_filt$x
## define labels
# change labels to make original class "2" of stars w/ exoplanets the positive class "X1"
temp <- mutate(train_data,LABEL = ifelse(LABEL=="2","1","2"))
# change labels to make original class "2" of stars w/ exoplanets the positive class "X1"
TrainY <- make.names(temp$LABEL)
# set up random forest model
set.seed(1, sample.kind="Rounding")
rf_model_down <- train(x=TrainX, y=TrainY,
                   method = "rf",
                   trControl = ctrl_pars,
                   metric = "ROC",
                  tuneGrid = data.frame(mtry =seq(3,99,3)))

## set up knn model
set.seed(1, sample.kind="Rounding")
knn_model_down <- train(x=TrainX, y=TrainY,
                   method = "knn",
                   trControl = ctrl_pars,
                   metric = "ROC",
                   tuneGrid = data.frame(k =seq(1,100,2)))

## set up svm model
set.seed(1, sample.kind="Rounding")
svm_model_down <- train(x=TrainX, y=TrainY,
                   method = "svmRadial",
                   trControl = ctrl_pars,
                   metric = "ROC",
                   tuneLength = 10)
```

Anomaly detection

The support vector machine for anomaly detection is tuned in 10 repeats with 10-fold cross-validation. As the model will only trained on the 37 examples of stars with exoplanets in the dataset, issues in the generalizability of model is expected.

```
## define the settings for cross-validation.
## define training inputs (stars w/ exoplanets only)
TrainX <- pca_ps_filt$x[1:37,]
## define labels
# change labels to make original class "2" of stars w/ exoplanets the positive class "X1"
temp <- mutate(train_data,LABEL = ifelse(LABEL=="2",TRUE,FALSE))
TrainY <- temp$LABEL[1:37]

# set up anomaly model
set.seed(1, sample.kind="Rounding")
svm_model_anomaly <- tune(svm, train.x = TrainX, train.y = as.factor(TrainY),
                         kernel="radial",
                         type="one-classification",
                         ranges = list(gamma=c(0.1,0.5,1,2,4),
                                       cost = c(0.1,1,10,100,1000)
                                       ),
                         tunecontrol = tune.control(nrepeat = 10,
```

```
                                                     sampling = "cross",
                                                     cross = 10,
                                                     performances = TRUE)
                  )
```

Results

Comparison of all models

As first step, the models are compared in confusion matrices based on their respective predictions on the training dataset.

```
#make predictions on the training data for all models
pred_rf <- predict(rf_model_down, pca_ps_filt$x)

pred_knn <- predict(knn_model_down, pca_ps_filt$x)

pred_svm <- predict(svm_model_down, pca_ps_filt$x)

pred_anomaly<- predict(svm_model_anomaly$best.model, pca_ps_filt$x)

# prepare confusion matrices
# change labels to make original class "2" of stars w/ exoplanets the positive class "X1"
ref <- mutate(train_data,LABEL = ifelse(LABEL=="2","X1","X2"))
# change labels to make original class "2" of stars w/ exoplanets the positive class TRUE
ref_anom <- mutate(train_data,LABEL = ifelse(LABEL=="2",TRUE,FALSE))

cm_rf <- table(Predicted_rf =pred_rf, Reference =ref$LABEL)

cm_knn <- table(Predicted_knn =pred_knn, Reference =ref$LABEL)

cm_svm <- table(Predicted_svm =pred_svm, Reference =ref$LABEL)

cm_anom <- table(Predicted_anom =pred_anomaly, Reference =ref_anom$LABEL)

# print matrices

cm_rf
```

```
##             Reference
## Predicted_rf   X1   X2
##           X1   37  888
##           X2    0 4162
```

```
cm_knn
```

```
##              Reference
## Predicted_knn   X1   X2
##            X1   37 2812
##            X2    0 2238
```

```
cm_svm
```

```
##              Reference
```

```
## Predicted_svm   X1   X2
##            X1   37 1302
##            X2    0 3748
```

```
cm_anom
```

```
##                Reference
## Predicted_anom FALSE TRUE
##          FALSE  4928   13
##          TRUE    122   24
```

The confusion matrices of the random forest, the k-nearest neighbor, and the support vector machine show that all stars with exoplanets are detected in the training data (high specificity), however there is also a significant amount of false positives (low sensitivity). In fact with respect to sensitivity, the anomaly detector performs best at the cost of lower specificity.

Three models are selected to be tested on the validation data: the random forest model (best specificity of the highly sensitive models), the anomaly detector and a two tier predictor consisting of the random forest model (tier 1) and the anomaly detector (tier 2).

Model validation

As first the the validation data is prepared to match the expected input of the models. For this periodograms are calculated, the intensities are standardized, the filter mask is applied, and lastly the values with respect to the principal components of the training data are calculated.

```
# process validation data

# calculation of power spectra
validation_ps <- matrix(ncol = 1599, nrow = 0) #create empty ps matrix

# create periograms
for (i in 1:570) { # extract time series row by row
  temp<-as.matrix(validation_data[i,2:3197])
  ints2 <- abs(fft(temp))^2/3196 # calculate intensities from squared amplitudes
  scaled_ints <- (4/3196)*ints2[1:1599] # re-scale
  validation_ps <- rbind(validation_ps,scaled_ints) # create ps matrix
}

# filter, min-max scaling
validation_ps_centered <- sweep(validation_ps, 1, apply(validation_ps,1,min))
validation_ps_standardized <- sweep(validation_ps_centered, 1,
                                    apply(validation_ps,1,max), FUN = "/")

validation_ps_filt <- sweep(validation_ps_standardized, 1,
                            filter_norm, FUN = "*")

# transform the validation dataset to obtain eigenvalues
pca_val <- validation_ps_filt[,1:959] %*% pca_ps_filt$rotation

#model 1: random forest predictor only

pred_val_1 <- predict(rf_model_down, pca_val)

#model 2: random forest predictor + anomaly detector
```

```r
data_tier1 <- cbind(pca_val,
                    pred_val_1,
                    ind=1:length(validation_data$LABEL)) %>%
  .[.[,"pred_val_1"] =="1",] # select "X1" from first prediction and add index ind

data_tier2 <- cbind(data_tier1,
                    pred_anom=predict(svm_model_anomaly$best.model,
                                      data_tier1[,1:959])) # combine with anomaly prediction

pred_val_2 <- ifelse(1:length(validation_data$LABEL) %in% data_tier2[data_tier2[,"ind"]],"TRUE","FALSE"

#model 3: anomaly detector only

pred_val_3 <- predict(svm_model_anomaly$best.model, pca_val)

# evaluation of predictive performance
# change labels to make original class "2" of stars w/ exoplanets the positive class "X1"
ref_val <- mutate(validation_data,LABEL = ifelse(LABEL=="2","X1","X2"))
# change labels to make original class "2" of stars w/ exoplanets the positive class TRUE
ref_val_anom <- mutate(validation_data,LABEL = ifelse(LABEL=="2",TRUE,FALSE))

cm_model_1 <- table(Predicted =pred_val_1, Reference =ref_val$LABEL)

cm_model_2 <- table(Predicted =pred_val_2, Reference =ref_val_anom$LABEL)

cm_model_3 <- table(Predicted =pred_val_3, Reference =ref_val_anom$LABEL)


#print rf matrix
print("Confusion matrix random forest model only")
```

```
## [1] "Confusion matrix random forest model only"
```

```r
cm_model_1
```

```
##          Reference
## Predicted  X1  X2
##        X1   2  66
##        X2   3 499
```

```r
#print 2-tier matrix
print("Confusion matrix two tier model")
```

```
## [1] "Confusion matrix two tier model"
```

```r
cm_model_2
```

```
##          Reference
## Predicted FALSE TRUE
##     FALSE   565    5
```

```r
#print anomaly detector matrix
print("Confusion matrix anomaly detector")
```

```
## [1] "Confusion matrix anomaly detector"
```

```
cm_model_3
```

```
##          Reference
## Predicted FALSE TRUE
##     FALSE   562    5
##     TRUE      3    0
```

Confusion matrices show, that only the random forest model yields true positive results while the other models are not able to detect any of the five stars with exoplanets in the validation dataset. The sensitivity of the prediction is *2/(2+3) = 0.4* while the specificity is *499/(499+66) = 0.88*. This means with respect to the task "predict stars w/o exoplanets" the model performs worse in terms of accuracy than assigning the this label to the complete dataset.

Conclusion

In this work, the Kepler labelled time series data" data set as provided on the Kaggle platform has been investigated and models have been developed to predict stars with exoplanets based on differences in frequency composition of observed intensity fluctuations in their emitted light fluxes. A random forest model combined with randomized downsampling, a support vector machine based one-class predictor and a combination of those two were tested on a validation dataset. Only the random forest model yielded true positive results with a sensitivity of 0.4 and specificity of 0.88. Since the observations of stars with exoplanets are very rare, especially the sensitivity should be optimized.

The dataset posed two main challenges: feature selection in a time series and class-imbalance. The first issue was addressed by representing the time series as periodograms. However, selection of distinctive frequency patterns could be improved. The convolution of the periodograms with a filter mask which enhanced periodogram components at the most distinctive frequency regions artificially enhanced also frequency peaks in the examples of stars without exoplanets which potentially introduced artefacts.Therefore, as a potential follow-up mask tresholds should be refined or regions of interest properly extracted from the frequency matrices.

Secondly, options to address the class imbalance were limited by the available computational performance. Only anomaly detection and down-sampling were tried out. Randomized over-sampling was deemed to costly as the training data would be greatly inflated due to the oversampling. However, the option of oversampling by creating "synthetic" minory test cases with techniques such as SMOTE [3] should be further investigated.

References

[1] Kumar, P., Bhatnagar, R., Gaur, K., and Bhatnagar, A. (2021) *Classification of Imbalanced Data:Review of Methods and Applications.* IOP Conf. Series: Materials Science and Engineering. doi:10.1088/1757-899X/1099/1/012077

[2] Brownlee, J. (2020) *A Gentle Introduction to Imbalanced Classification.* Available at: https://machinelearningmastery.com/what-is-imbalanced-classification/ (Accessed: 06/03/2022)

[3] Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W. (2002) *SMOTE: Synthetic Minority Over-sampling Technique.* Journal Of Artificial Intelligence Research. https://doi.org/10.1613/jair.953