Comparing different models for inferring sequency type from metadata

Table of contents

Introduction

Here we compare different models (and preprocessing strategies for working with mostly text-based metadata). Particularly, we compare the two (three) following strategies:

1. Textual (counting words in different columns) and numeric (characterising character-separated numeric columns as their size, minimum, maximum, sum and mean) preprocessing. We then use these values with a random forest classifier or with an XGBoost classifier
2. Plugging the raw data into a CatBoost model.

In terms of development, strategy 1. is characterized by a faster training, but a greater involvement of textual processing. Strategy 2., on the other hand, is characterised by lengthier training but a much easier development as it requires no preprocessing or hyperparameter optimization.

We also test how removing certain features - such as series description (SR), percent phase field of view (% phase FOV) and SAR - affects performance. The reason why we do this is tied with the facts that oftentimes the SR is highly specific of the centre conducting each scan, and that the % phase FOV and SAR are frequently missing from ADC sequences.

In any case we compare results hailing from cross-validation and from a hold-out test set (corresponding to 20% of the full dataset).

```r
library(tidyverse)
library(knitr)

dir.create("figures",showWarnings=F)

exclusion_match <- c(
    `standard` = "All features",
    `series_description` = "No SR",
    `percent_phase_field_of_view:sar` = "No % phase FOV or SAR",
    `percent_phase_field_of_view:sar:series_description` = "No % phase FOV, SAR or SR"
)
all_metrics <- read_csv("../data_output/metrics.csv") %>%
    mutate(exclusion = factor(exclusion,
                              levels=names(exclusion_match),
                              labels=exclusion_match)) %>%
    mutate(model = factor(
        model,
        levels = c("rf","extra_trees","xgb","catboost"),
        labels = c("Random forest","Extra trees","XGBoost","CatBoost"))) %>%
    group_by(model,exclusion,metric,set) %>%
    mutate(best_fold = ifelse(
        split == "test",
        fold == fold[which.max(value[split == "cv"])],
        NA)) %>%
    subset(metric != "support") %>%
    mutate(metric = factor(
        metric,
        levels = c("auc","cm",
                   "precision","recall","f1-score"),
        labels = c("auc","cm",
                   "Precision","Recall","F1-score"))) %>%
    filter(set != "weighted avg") %>%
    mutate(
        set = factor(set,
                     levels = c("adc","dce","dwi",
                                "t2","others",
                                "macro avg"),
                     labels = c("ADC","DCE","DWI",
                                "T2W","Others",
                                "Average"))
    )
all_metrics_cv <- all_metrics %>%
    subset(split == "cv")
all_metrics_test <- all_metrics %>%
    subset(split == "test")
all_metrics_test_consensus <- all_metrics %>%
    subset(split == "test_consensus")
all_metrics_test_ensemble <- all_metrics %>%
    subset(split == "test_ensemble")
```

Results

Predictive performance

Confusion matrices

CV

As continuous variables

```
all_metrics_cv %>%
    subset(metric == "cm") %>%
    group_by(model,exclusion,true,fold) %>%
    mutate(p = value / sum(value)) %>%
    group_by(model,exclusion,true,pred) %>%
    summarise(p = mean(p),
              value = mean(value)) %>%
    ggplot(aes(x = true,y = pred,
           label = sprintf("%.1f%%\n(%.0f)",p*100,value),
           fill = p)) +
    geom_tile() +
    geom_text(size = 2) +
    facet_grid(exclusion ~ model) +
    theme_minimal(base_size = 8) +
    scale_y_discrete(limits=rev) +
    scale_fill_gradient2(low="lightskyblue1",
                         mid="white",
                         high="goldenrod1",
                         midpoint=0.5,
                         limits = c(0,1),
                         name = "") +
    xlab("True") +
    ylab("Predicted") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"))
ggsave(filename="figures/cm_cv.png",
       height=7,width=7)
```

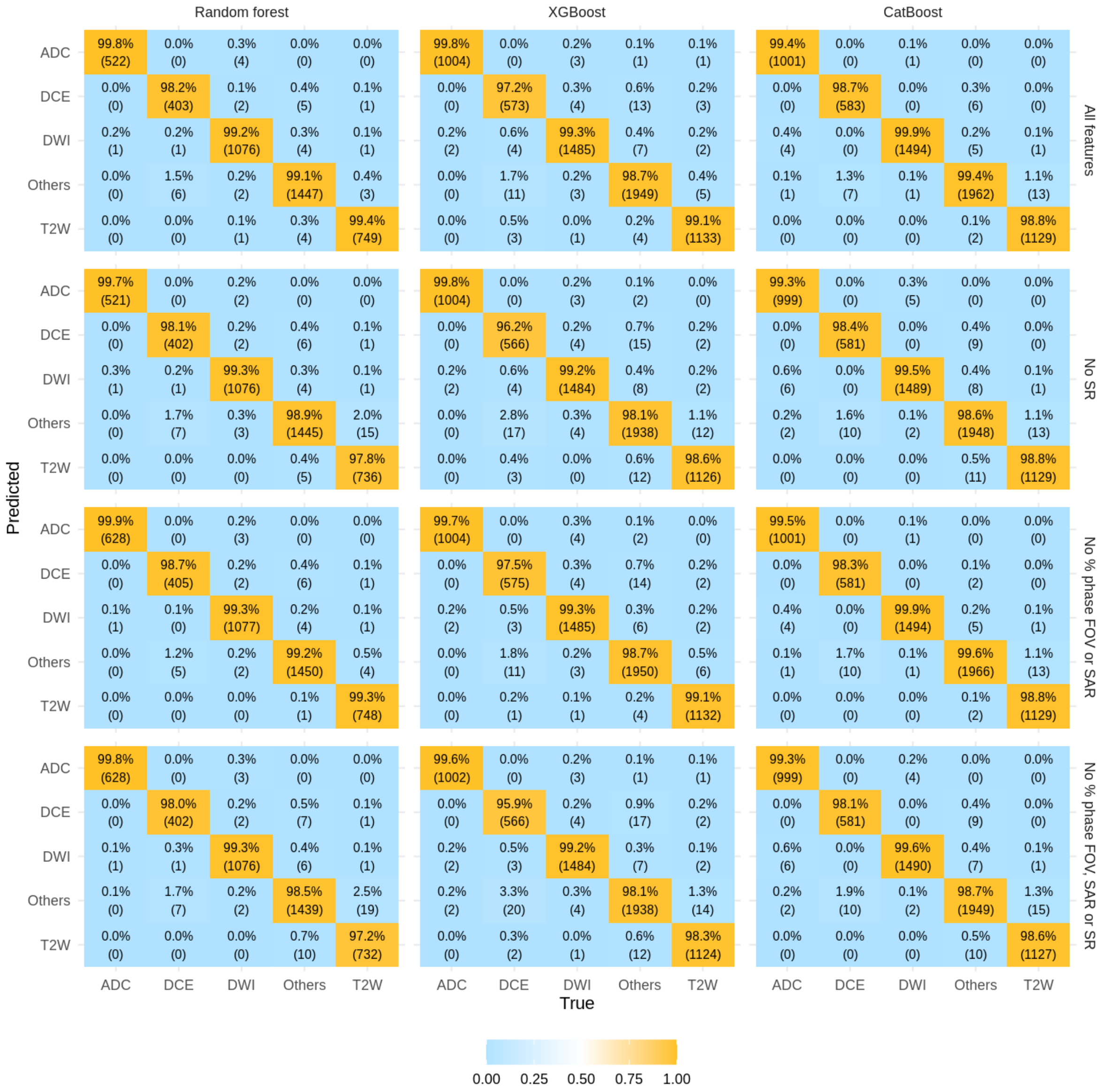


Figure 1: CV confusion matrices.

As binned variables (0.02 intervals)

```
all_metrics_cv %>%
    subset(metric == "cm") %>%
    group_by(model,exclusion,true,fold) %>%
    mutate(p = value / sum(value)) %>%
    group_by(model,exclusion,true,pred) %>%
    summarise(p = mean(p),
              value = mean(value)) %>%
    mutate(p_fill = cut(
        p,seq(0,1,by=0.02),
        include.lowest=T)) %>%
    ggplot(aes(x = true,y = pred,
           label = sprintf("%.1f%%\n(%.0f)",p*100,value),
           fill = p_fill)) +
    geom_tile(alpha = 0.6) +
    geom_text(size = 2) +
    facet_grid(exclusion ~ model) +
    theme_minimal(base_size = 8) +
    scale_y_discrete(limits=rev) +
    scale_fill_brewer(palette = "PiYG",name = "") +
    xlab("True") +
    ylab("Predicted") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"),
          panel.grid = element_blank())
ggsave(filename="figures/cm_cat_cv.png",
       height=7,width=7)
```

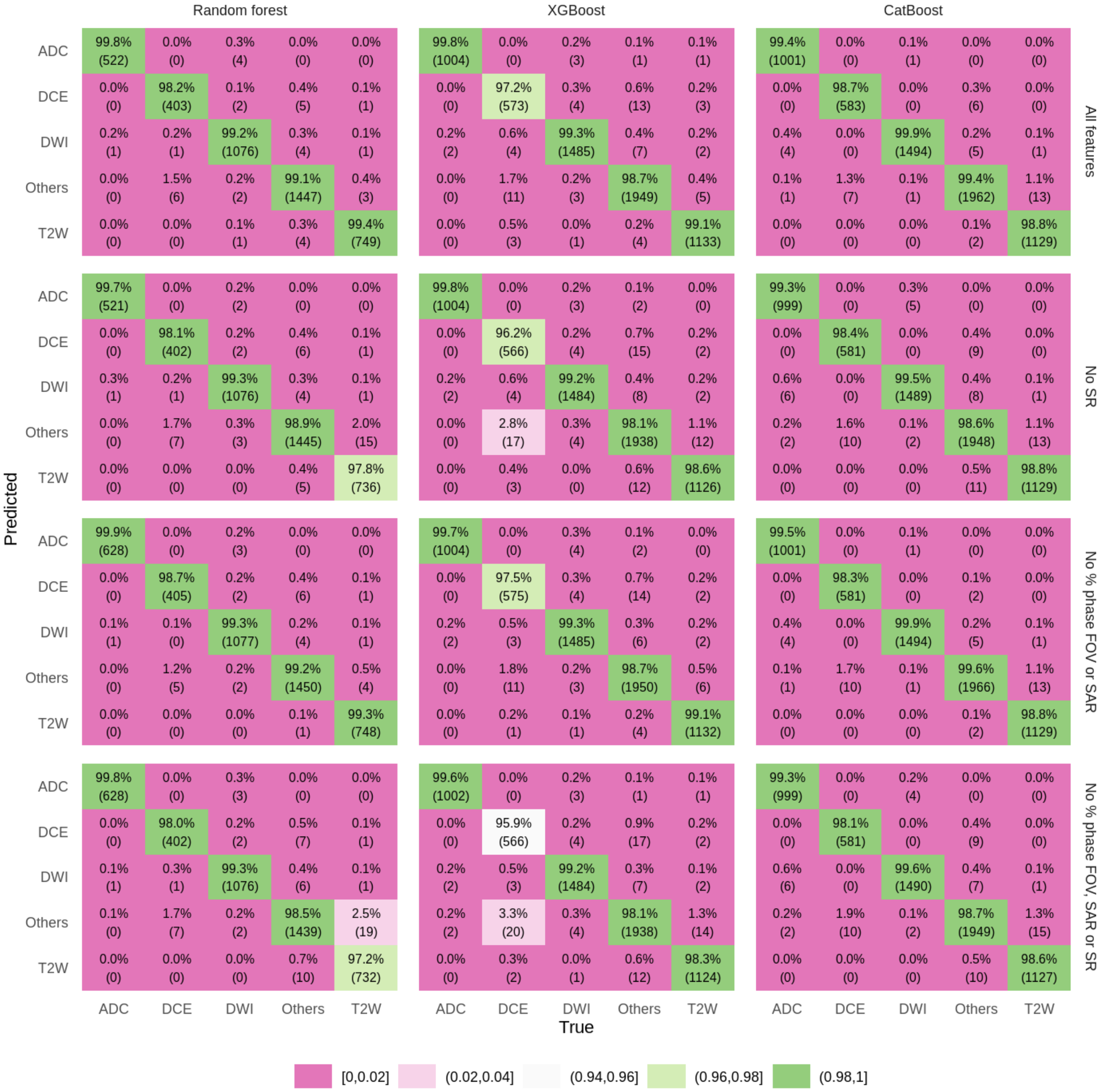
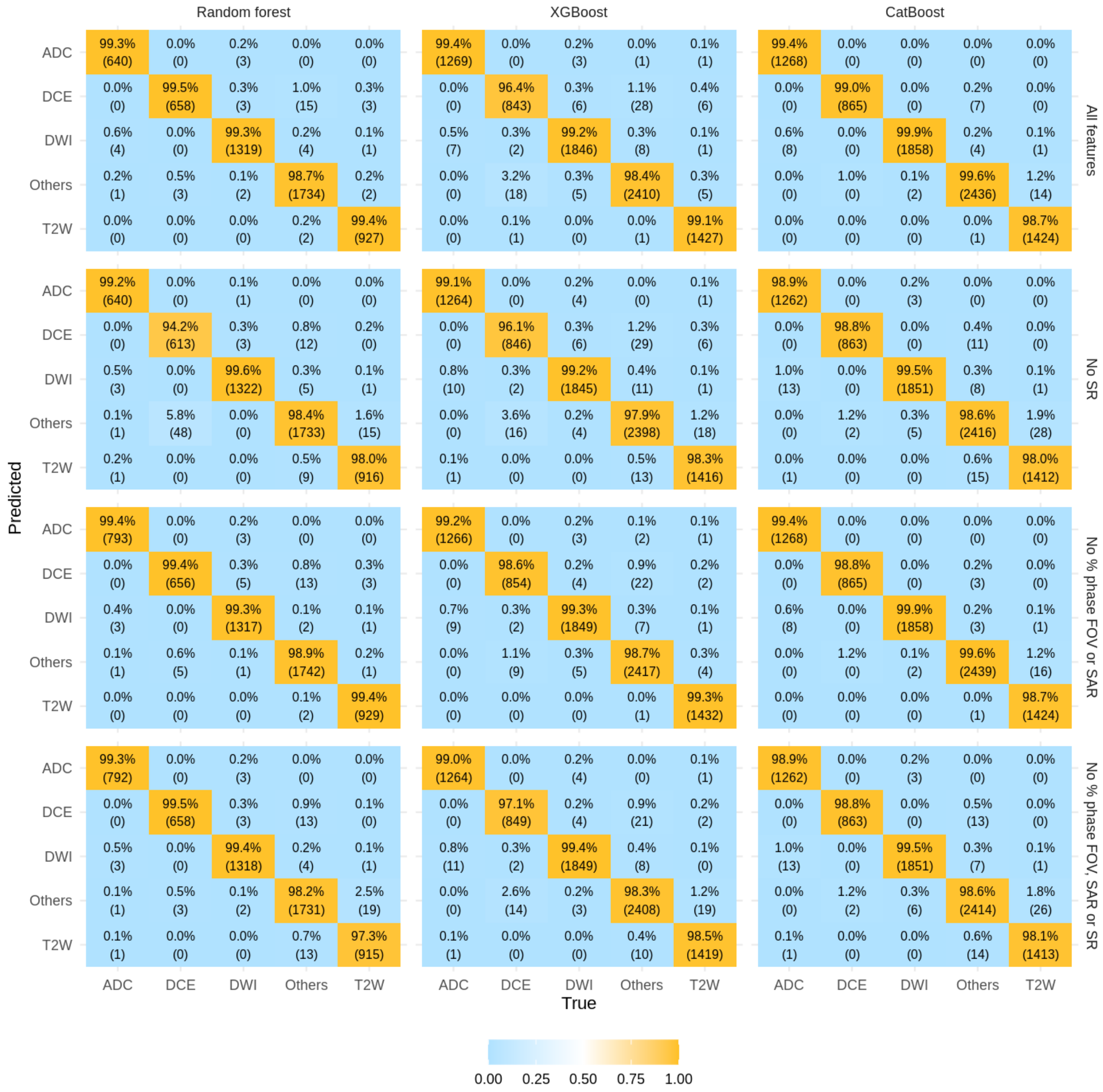


Figure 2: CV confusion matrices (binned).

Hold-out test-set

As continuous variables

```
all_metrics_test %>%
    subset(metric == "cm") %>%
    group_by(model,exclusion,true,fold) %>%
    mutate(p = value / sum(value)) %>%
    group_by(model,exclusion,true,pred) %>%
    summarise(p = mean(p),
              value = median(value)) %>%
    ggplot(aes(x = true,y = pred,
           label = sprintf("%.1f%%\n(%s)",p*100,value),
           fill = p)) +
    geom_tile() +
    geom_text(size = 2) +
    facet_grid(exclusion ~ model) +
    theme_minimal(base_size = 8) +
    scale_y_discrete(limits=rev) +
    scale_fill_gradient2(low="lightskyblue1",
                         mid="white",
                         high="goldenrod1",
                         midpoint=0.5,
                         limits = c(0,1),
                         name = "") +
    xlab("True") +
    ylab("Predicted") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"))
ggsave(filename="figures/cm_test.png",
       height=7,width=7)
```



	Random forest					XGBoost					CatBoost					
	ADC	DCE	DWI	Others	T2W	ADC	DCE	DWI	Others	T2W	ADC	DCE	DWI	Others	T2W	
ADC	99.3% (640)	0.0% (0)	0.2% (3)	0.0% (0)	0.0% (0)	99.4% (1269)	0.0% (0)	0.2% (3)	0.0% (1)	0.1% (1)	99.4% (1268)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	All features
DCE	0.0% (0)	99.5% (658)	0.3% (3)	1.0% (15)	0.3% (3)	0.0% (0)	96.4% (843)	0.3% (6)	1.1% (28)	0.4% (6)	0.0% (0)	99.0% (865)	0.0% (0)	0.2% (7)	0.0% (0)	
DWI	0.6% (4)	0.0% (0)	99.3% (1319)	0.2% (4)	0.1% (1)	0.5% (7)	0.3% (2)	99.2% (1846)	0.3% (8)	0.1% (1)	0.6% (8)	0.0% (0)	99.9% (1858)	0.2% (4)	0.1% (1)	
Others	0.2% (1)	0.5% (3)	0.1% (2)	98.7% (1734)	0.2% (2)	0.0% (0)	3.2% (18)	0.3% (5)	98.4% (2410)	0.3% (5)	0.0% (0)	1.0% (0)	0.1% (2)	99.6% (2436)	1.2% (14)	
T2W	0.0% (0)	0.0% (0)	0.0% (0)	0.2% (2)	99.4% (927)	0.0% (0)	0.1% (1)	0.0% (0)	0.0% (1)	99.1% (1427)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (1)	98.7% (1424)	
ADC	99.2% (640)	0.0% (0)	0.1% (1)	0.0% (0)	0.0% (0)	99.1% (1264)	0.0% (0)	0.2% (4)	0.0% (0)	0.1% (1)	98.9% (1262)	0.0% (0)	0.2% (3)	0.0% (0)	0.0% (0)	No SR
DCE	0.0% (0)	94.2% (613)	0.3% (3)	0.8% (12)	0.2% (0)	0.0% (0)	96.1% (846)	0.3% (6)	1.2% (29)	0.3% (6)	0.0% (0)	98.8% (863)	0.0% (0)	0.4% (11)	0.0% (0)	
DWI	0.5% (3)	0.0% (0)	99.6% (1322)	0.3% (5)	0.1% (1)	0.8% (10)	0.3% (2)	99.2% (1845)	0.4% (11)	0.1% (1)	1.0% (13)	0.0% (0)	99.5% (1851)	0.3% (8)	0.1% (1)	
Others	0.1% (1)	5.8% (48)	0.0% (0)	98.4% (1733)	1.6% (15)	0.0% (0)	3.6% (16)	0.2% (4)	97.9% (2398)	1.2% (18)	0.0% (0)	1.2% (2)	0.3% (5)	98.6% (2416)	1.9% (28)	
T2W	0.2% (1)	0.0% (0)	0.0% (0)	0.5% (9)	98.0% (916)	0.1% (1)	0.0% (0)	0.0% (0)	0.5% (13)	98.3% (1416)	0.0% (1)	0.0% (0)	0.0% (0)	0.6% (15)	98.0% (1412)	
ADC	99.4% (793)	0.0% (0)	0.2% (3)	0.0% (0)	0.0% (0)	99.2% (1266)	0.0% (0)	0.2% (3)	0.1% (2)	0.1% (1)	99.4% (1268)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	No % phase FOV or SAR
DCE	0.0% (0)	99.4% (656)	0.3% (5)	0.8% (13)	0.3% (3)	0.0% (0)	98.6% (854)	0.2% (4)	0.9% (22)	0.2% (2)	0.0% (0)	98.8% (865)	0.0% (0)	0.2% (3)	0.0% (0)	
DWI	0.4% (3)	0.0% (0)	99.3% (1317)	0.1% (2)	0.1% (1)	0.7% (9)	0.3% (2)	99.3% (1849)	0.3% (7)	0.1% (1)	0.6% (8)	0.0% (0)	99.9% (1858)	0.2% (3)	0.1% (1)	
Others	0.1% (1)	0.6% (5)	0.1% (1)	98.9% (1742)	0.2% (1)	0.0% (0)	1.1% (9)	0.3% (5)	98.7% (2417)	0.3% (4)	0.0% (0)	1.2% (0)	0.1% (2)	99.6% (2439)	1.2% (16)	
T2W	0.0% (0)	0.0% (0)	0.0% (0)	0.1% (2)	99.4% (929)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (1)	99.3% (1432)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (1)	98.7% (1424)	
ADC	99.3% (792)	0.0% (0)	0.2% (3)	0.0% (0)	0.0% (0)	99.0% (1264)	0.0% (0)	0.2% (4)	0.0% (0)	0.1% (1)	98.9% (1262)	0.0% (0)	0.2% (3)	0.0% (0)	0.0% (0)	No % phase FOV, SAR or SR
DCE	0.0% (0)	99.5% (658)	0.3% (3)	0.9% (13)	0.1% (0)	0.0% (0)	97.1% (849)	0.2% (4)	0.9% (21)	0.2% (2)	0.0% (0)	98.8% (863)	0.0% (0)	0.5% (13)	0.0% (0)	
DWI	0.5% (3)	0.0% (0)	99.4% (1318)	0.2% (4)	0.1% (1)	0.8% (11)	0.3% (2)	99.4% (1849)	0.4% (8)	0.1% (0)	1.0% (13)	0.0% (0)	99.5% (1851)	0.3% (7)	0.1% (1)	
Others	0.1% (1)	0.5% (3)	0.1% (2)	98.2% (1731)	2.5% (19)	0.0% (0)	2.6% (14)	0.2% (3)	98.3% (2408)	1.2% (19)	0.0% (0)	1.2% (2)	0.3% (6)	98.6% (2414)	1.8% (26)	
T2W	0.1% (1)	0.0% (0)	0.0% (0)	0.7% (13)	97.3% (915)	0.1% (1)	0.0% (0)	0.0% (0)	0.4% (10)	98.5% (1419)	0.1% (1)	0.0% (0)	0.0% (0)	0.6% (14)	98.1% (1413)	

Figure 3: Hold-out test-set confusion matrices.

As binned variables (0.02 intervals)

```
all_metrics_test %>%
    subset(metric == "cm") %>%
    group_by(model,exclusion,true,fold) %>%
    mutate(p = value / sum(value)) %>%
    group_by(model,exclusion,true,pred) %>%
    summarise(p = mean(p),
              value = median(value)) %>%
    mutate(p_fill = cut(p,seq(0,1,by=0.02),
                        include.lowest=T)) %>%
    ggplot(aes(x = true,y = pred,
           label = sprintf("%.1f%%\n(%.0f)",p*100,value),
           fill = p_fill)) +
    geom_tile(alpha = 0.6) +
    geom_text(size = 2) +
    facet_grid(exclusion ~ model) +
    theme_minimal(base_size = 8) +
    scale_y_discrete(limits=rev) +
    scale_fill_brewer(palette = "PiYG",name = "") +
    xlab("True") +
    ylab("Predicted") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"),
          panel.grid = element_blank())
ggsave(filename="figures/cm_cat_test.png",
       height=7,width=7)
```

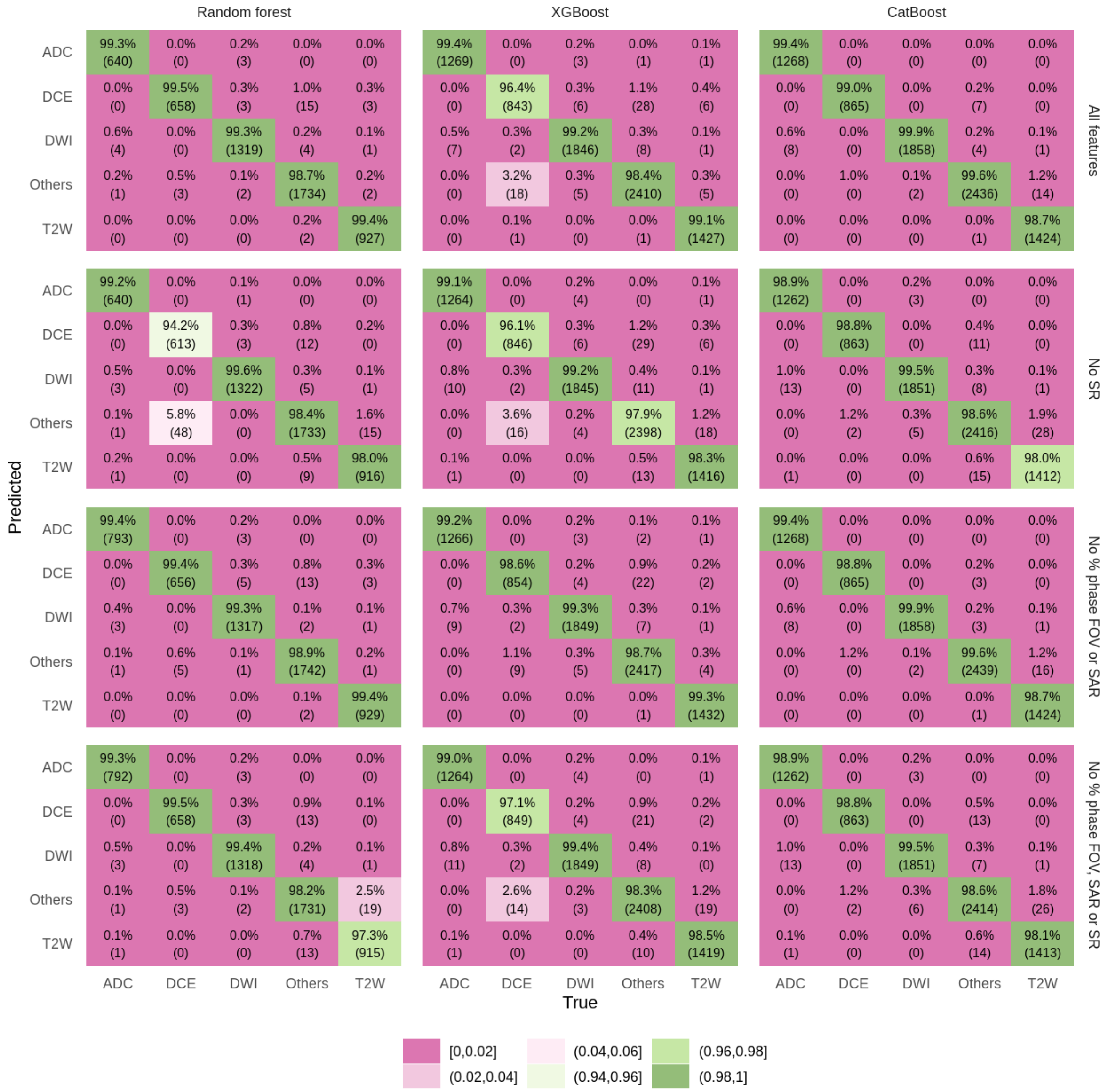


Figure 4: Hold-out test-set confusion matrices (binned).

Hold-out test-set (consensus)

As continuous variables

```
all_metrics_test_consensus %>%
    subset(metric == "cm") %>%
    group_by(model,exclusion,true,fold) %>%
    mutate(p = value / sum(value)) %>%
    ggplot(aes(x = true,y = pred,
           label = sprintf("%.1f%%\n(%s)",p*100,value),
           fill = p)) +
    geom_tile() +
    geom_text(size = 2) +
    facet_grid(exclusion ~ model) +
    theme_minimal(base_size = 8) +
    scale_y_discrete(limits=rev) +
    scale_fill_gradient2(low="lightskyblue1",
                         mid="white",
                         high="goldenrod1",
                         midpoint=0.5,
                         limits = c(0,1),
                         name = "") +
    xlab("True") +
    ylab("Predicted") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"))
ggsave(filename="figures/cm_test_consensus.png",
       height=7,width=7)
```

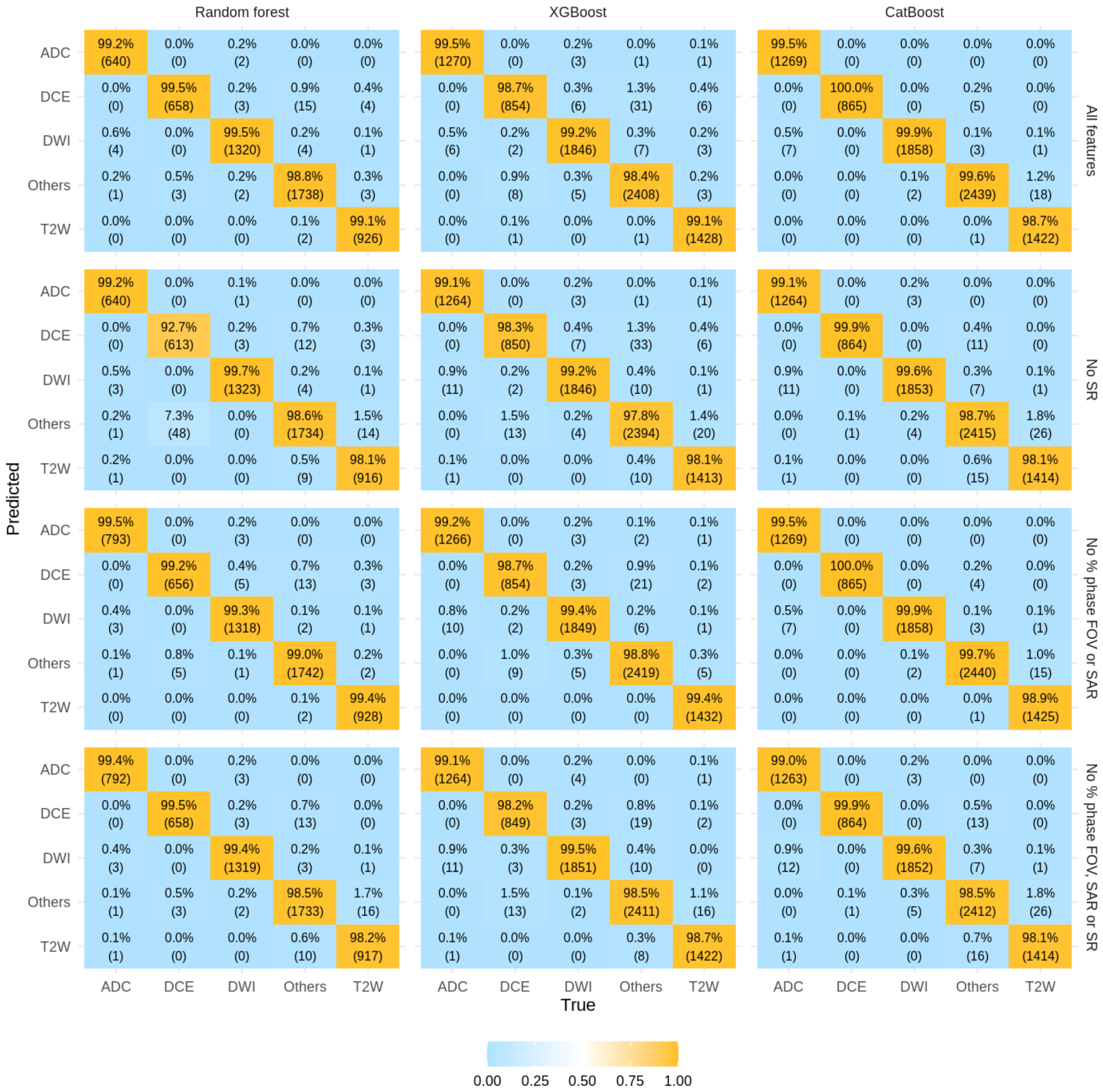


Figure 5: Hold-out test-set consensus confusion matrices.

As binned variables (0.02 intervals)

```
all_metrics_test_consensus %>%
    subset(metric == "cm") %>%
    group_by(model,exclusion,true,fold) %>%
    mutate(p = value / sum(value)) %>%
    mutate(p_fill = cut(p,seq(0,1,by=0.02),
                        include.lowest=T)) %>%
    ggplot(aes(x = true,y = pred,
           label = sprintf("%.1f%%\n(%.0f)",p*100,value),
           fill = p_fill)) +
    geom_tile(alpha = 0.6) +
    geom_text(size = 2) +
    facet_grid(exclusion ~ model) +
    theme_minimal(base_size = 8) +
    scale_y_discrete(limits=rev) +
    scale_fill_brewer(palette = "PiYG",name = "") +
    xlab("True") +
    ylab("Predicted") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"),
          panel.grid = element_blank())
ggsave(filename="figures/cm_cat_test_consensus.png",
       height=7,width=7)
```

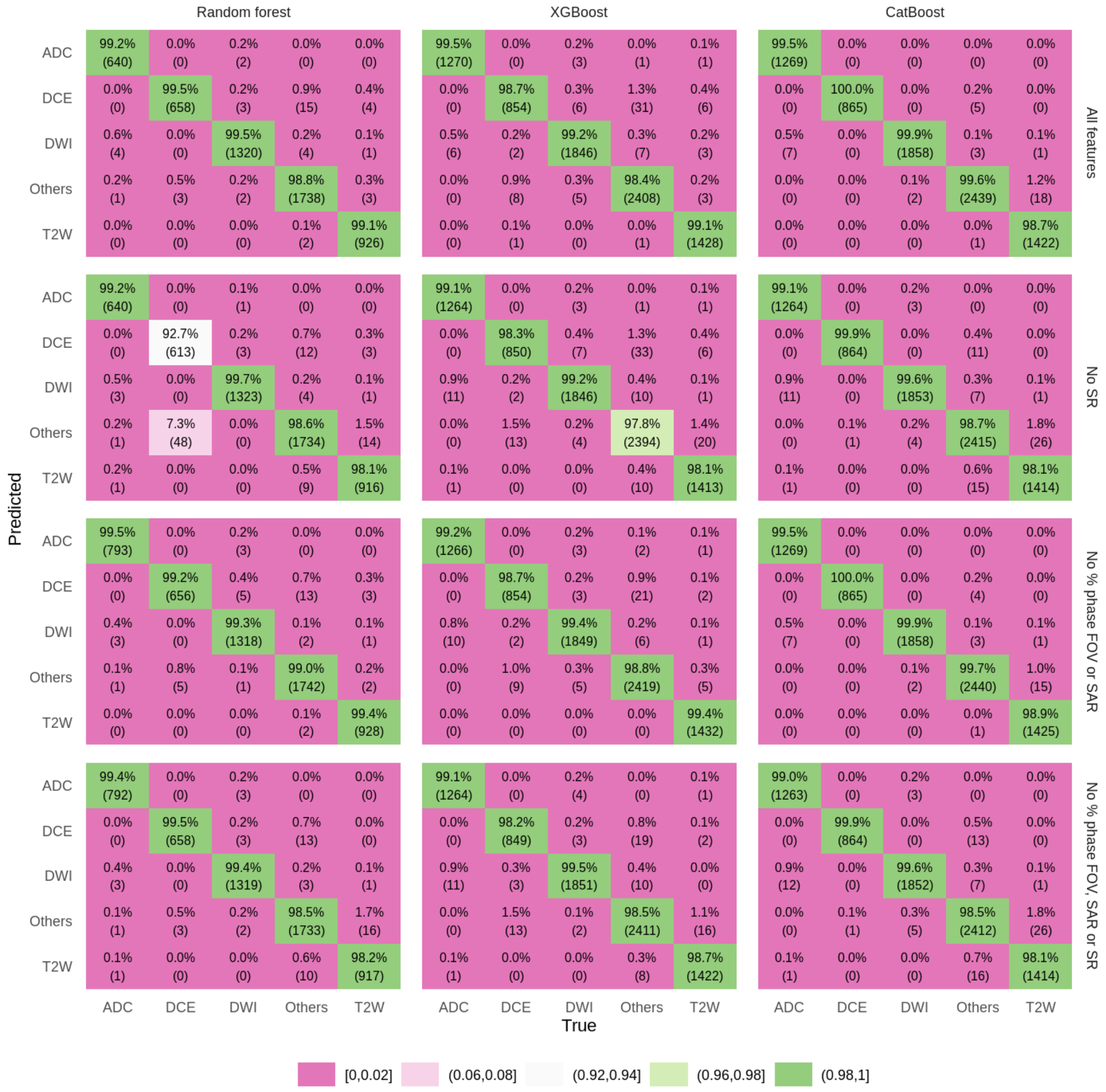


Figure 6: Hold-out test-set consensus confusion matrices (binned).

Hold-out test-set (ensemble)

As continuous variables

```r
all_metrics_test_ensemble %>%
    subset(metric == "cm") %>%
    group_by(model,exclusion,true,fold) %>%
    mutate(p = value / sum(value)) %>%
    ggplot(aes(x = true,y = pred,
           label = sprintf("%.1f%%\n(%s)",p*100,value),
           fill = p)) +
    geom_tile() +
    geom_text(size = 2) +
    facet_grid(exclusion ~ model) +
    theme_minimal(base_size = 8) +
    scale_y_discrete(limits=rev) +
    scale_fill_gradient2(low="lightskyblue1",
                         mid="white",
                         high="goldenrod1",
                         midpoint=0.5,
                         limits = c(0,1),
                         name = "") +
    xlab("True") +
    ylab("Predicted") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"))
ggsave(filename="figures/cm_test_ensemble.png",
       height=7,width=2.5)
```

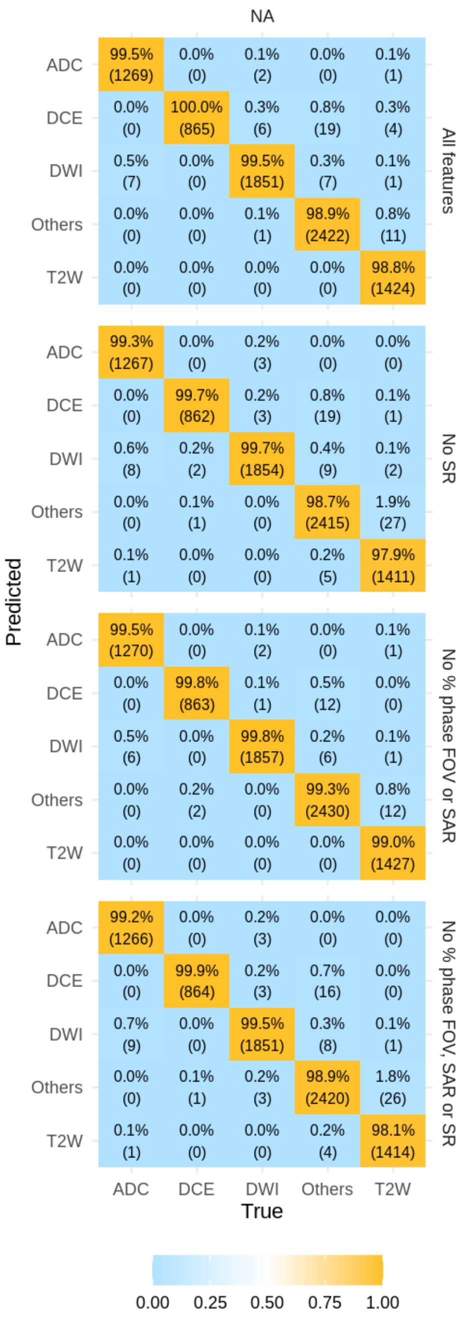
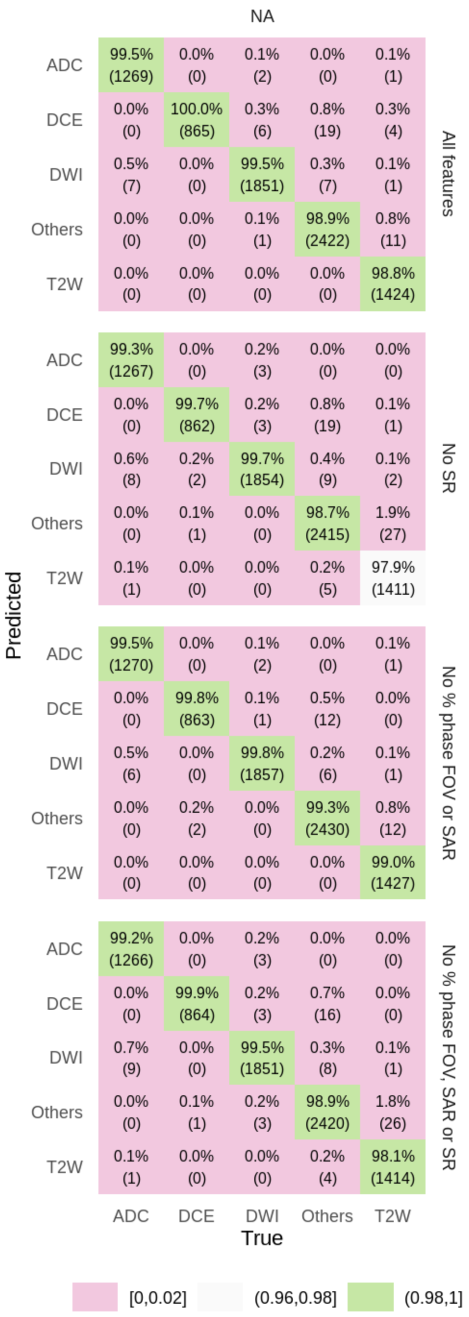


Figure 7: Hold-out test-set ensemble confusion matrices.

As binned variables (0.02 intervals)

```
all_metrics_test_ensemble %>%
    subset(metric == "cm") %>%
    group_by(model,exclusion,true,fold) %>%
    mutate(p = value / sum(value)) %>%
    mutate(p_fill = cut(p,seq(0,1,by=0.02),
                        include.lowest=T)) %>%
    ggplot(aes(x = true,y = pred,
           label = sprintf("%.1f%%\n(%.0f)",p*100,value),
           fill = p_fill)) +
    geom_tile(alpha = 0.6) +
    geom_text(size = 2) +
    facet_grid(exclusion ~ model) +
    theme_minimal(base_size = 8) +
    scale_y_discrete(limits=rev) +
    scale_fill_brewer(palette = "PiYG",name = "") +
    xlab("True") +
    ylab("Predicted") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"),
          panel.grid = element_blank())
ggsave(filename="figures/cm_test_ensemble.png",
       height=7,width=2.5)
```



All features

Predicted \ True	ADC	DCE	DWI	Others	T2W
ADC	99.5% (1269)	0.0% (0)	0.1% (2)	0.0% (0)	0.1% (1)
DCE	0.0% (0)	100.0% (865)	0.3% (6)	0.8% (19)	0.3% (4)
DWI	0.5% (7)	0.0% (0)	99.5% (1851)	0.3% (7)	0.1% (1)
Others	0.0% (0)	0.0% (0)	0.1% (1)	98.9% (2422)	0.8% (11)
T2W	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	98.8% (1424)

No SR

Predicted \ True	ADC	DCE	DWI	Others	T2W
ADC	99.3% (1267)	0.0% (0)	0.2% (3)	0.0% (0)	0.0% (0)
DCE	0.0% (0)	99.7% (862)	0.2% (3)	0.8% (19)	0.1% (1)
DWI	0.6% (8)	0.2% (2)	99.7% (1854)	0.4% (9)	0.1% (2)
Others	0.0% (0)	0.1% (1)	0.0% (0)	98.7% (2415)	1.9% (27)
T2W	0.1% (1)	0.0% (0)	0.0% (0)	0.2% (5)	97.9% (1411)

No % phase FOV or SAR

Predicted \ True	ADC	DCE	DWI	Others	T2W
ADC	99.5% (1270)	0.0% (0)	0.1% (2)	0.0% (0)	0.1% (1)
DCE	0.0% (0)	99.8% (863)	0.1% (1)	0.5% (12)	0.0% (0)
DWI	0.5% (6)	0.0% (0)	99.8% (1857)	0.2% (6)	0.1% (1)
Others	0.0% (0)	0.2% (2)	0.0% (0)	99.3% (2430)	0.8% (12)
T2W	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	99.0% (1427)

No % phase FOV, SAR or SR

Predicted \ True	ADC	DCE	DWI	Others	T2W
ADC	99.2% (1266)	0.0% (0)	0.2% (3)	0.0% (0)	0.0% (0)
DCE	0.0% (0)	99.9% (864)	0.2% (3)	0.7% (16)	0.0% (0)
DWI	0.7% (9)	0.0% (0)	99.5% (1851)	0.3% (8)	0.1% (1)
Others	0.0% (0)	0.1% (1)	0.2% (3)	98.9% (2420)	1.8% (26)
T2W	0.1% (1)	0.0% (0)	0.0% (0)	0.2% (4)	98.1% (1414)

Figure 8: Hold-out test-set ensemble confusion matrices (binned).

AUC

CV

```
all_metrics_cv %>%
    subset(metric == "auc") %>%
    group_by(model,exclusion) %>%
    summarise(vmin = min(value),
              vmax = max(value),
              value = mean(value)) %>%
    ggplot(aes(y = value,x = model,
               ymin = vmin,ymax=vmax,
               label = sprintf("%.2f%%",value*100),
               colour = exclusion)) +
    geom_point(position = position_dodge(0.8)) +
    geom_linerange(position = position_dodge(0.8)) +
    theme_minimal(base_size = 8) +
    scale_colour_brewer(type = "qual",palette = 2,
                        name = "Feature subset") +
    xlab("Models") +
    ylab("AUC") +
    scale_x_discrete(limits = rev) +
    scale_y_continuous(
        labels = function(x) sprintf("%.2f%%",x*100)) +
    theme(legend.key.height = unit(0.2,"cm"),
          legend.key.width = unit(0.2,"cm")) +
    coord_flip() +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm")) +
    guides(colour = guide_legend(nrow = 2))
ggsave(filename="figures/auc_cv.png",
       height=2,width=4)
```

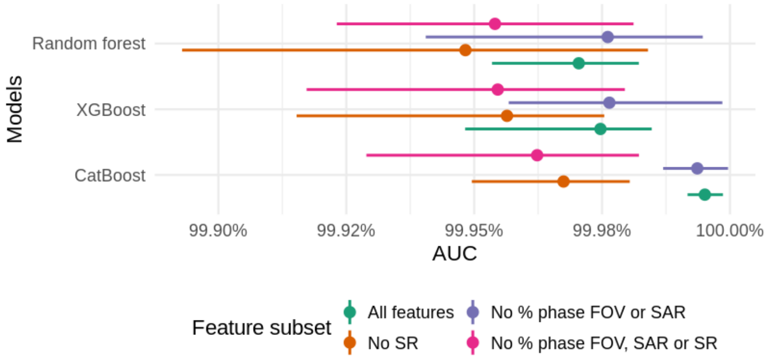


Figure 9: CV AUC.

Hold-out test set

```
all_metrics_test %>%
    subset(metric == "auc") %>%
    group_by(model,exclusion) %>%
    summarise(vmin = min(value),
              vmax = max(value),
              value = mean(value)) %>%
    ggplot(aes(y = value,x = model,
               ymin = vmin,ymax=vmax,
               label = sprintf("%.2f%%",value*100),
               colour = exclusion)) +
    geom_point(position = position_dodge(0.8)) +
    geom_linerange(position = position_dodge(0.8)) +
    geom_point(
        data=subset(all_metrics_test,
                    metric == "auc" & best_fold == T),
        aes(x = model,y = value,
            group = exclusion,
            shape = ifelse(best_fold,"Best CV model")),
        inherit.aes=F,
        position = position_dodge(0.8),
        size = 4) +
    scale_shape_manual(values = c("+"),
                       guide = F) +
    theme_minimal(base_size = 8) +
    xlab("Models") +
    ylab("AUC") +
    scale_colour_brewer(type = "qual",palette = 2,
                        name = "Feature subset") +
    scale_x_discrete(limits = rev) +
    scale_y_continuous(
        labels = function(x) sprintf("%.2f%%",x*100)) +
    theme(legend.key.height = unit(0.2,"cm"),
          legend.key.width = unit(0.2,"cm")) +
    coord_flip() +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm")) +
    guides(colour = guide_legend(nrow = 2))
ggsave(filename="figures/auc_test.png",
       height=2,width=4)
```

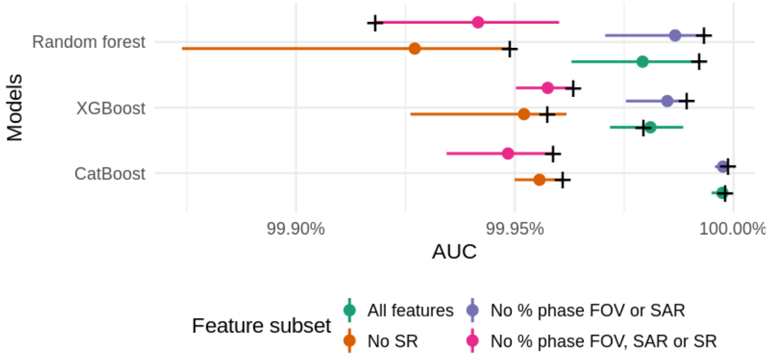


Figure 10: Hold-out test-set AUC.

Other metrics (precision, recall, F1-score)

CV

```
M <- c("Precision","Recall","F1-score")

all_metrics_cv %>%
    subset(metric %in% M) %>%
    group_by(model,exclusion,metric,set) %>%
    summarise(vmin = min(value),
              vmax = max(value),
              value = mean(value)) %>%
    ggplot(aes(y = value,x = model,
               ymin = vmin,ymax=vmax,
               label = sprintf("%.2f%%",value*100),
               colour = exclusion)) +
    geom_point(position = position_dodge(0.8)) +
    geom_linerange(position = position_dodge(0.8)) +
    theme_minimal(base_size = 8) +
    xlab("Models") +
    ylab("Value") +
    facet_grid(set ~ metric) +
    scale_colour_brewer(type = "qual",palette = 2,
                        name = "Feature subset") +
    scale_y_continuous(
        labels = function(x) sprintf("%.2f%%",x*100)) +
    theme(legend.key.height = unit(0.2,"cm"),
          legend.key.width = unit(0.2,"cm"),
          panel.background = element_rect(
            fill = NA,colour = "black")) +
    coord_flip() +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm")) +
    guides(colour = guide_legend(nrow = 2))
ggsave(filename="figures/metrics_cv.png",
       height=7,width=7)
```

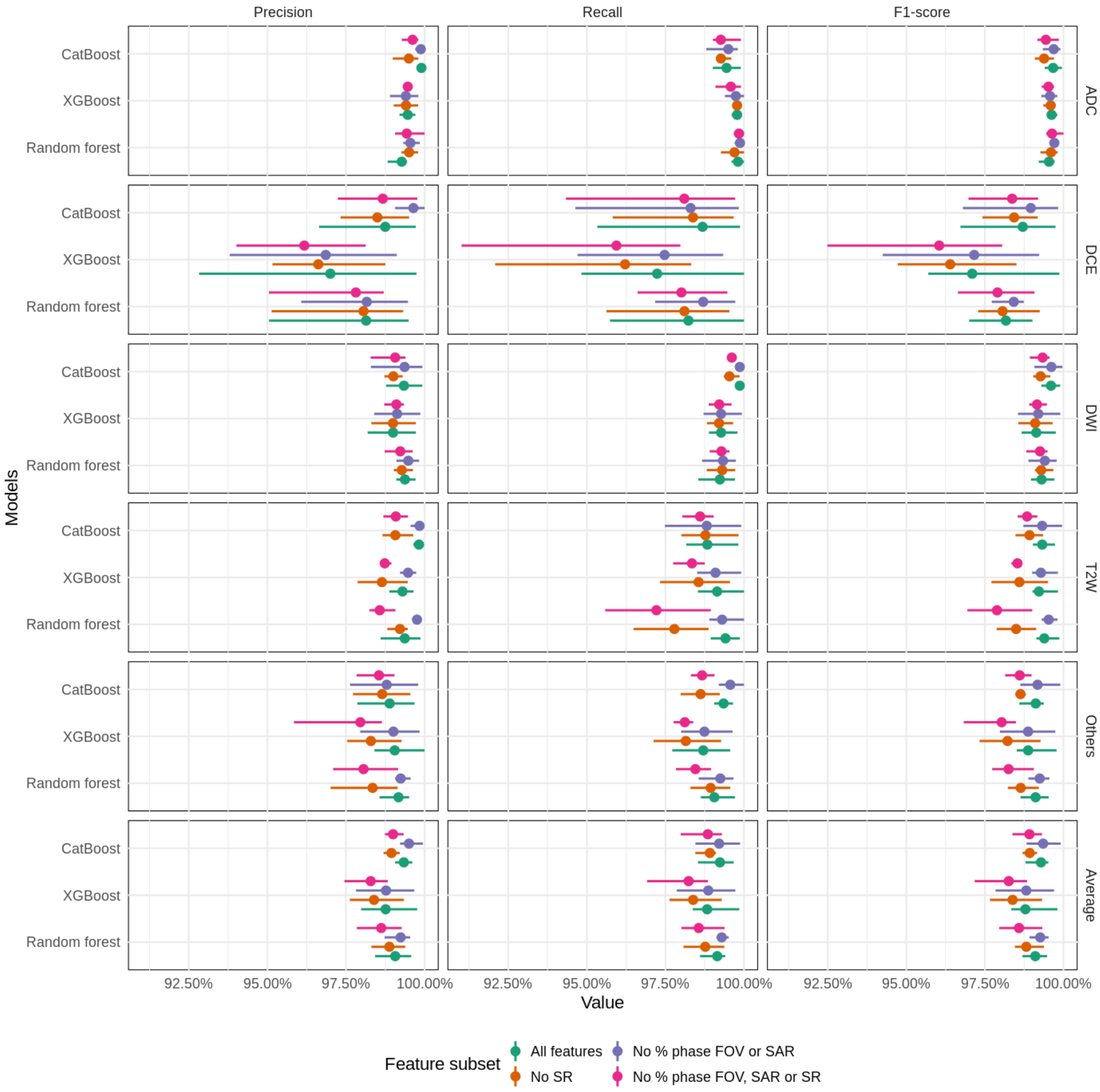


Figure 11: CV precision, recall and F1-score.

Hold-out test-set

```
all_metrics_test %>%
    subset(metric %in% M) %>%
    group_by(model,exclusion,metric,set) %>%
    summarise(vmin = min(value),
              vmax = max(value),
              value = mean(value)) %>%
    ggplot(aes(y = value,x = model,
               ymin = vmin,ymax=vmax,
               label = sprintf("%.2f%%",value*100),
               colour = exclusion)) +
    geom_point(position = position_dodge(0.8)) +
    geom_linerange(position = position_dodge(0.8)) +
    geom_point(
        data=subset(all_metrics_test,
                    metric %in% M & best_fold == T),
        aes(x = model,y = value,
            group = exclusion,
            shape = ifelse(best_fold,"Best CV model")),
        inherit.aes=F,
        position = position_dodge(0.8),
        size = 3) +
    scale_shape_manual(values = c("+"),
                       guide = F) +
    theme_minimal(base_size = 8) +
    xlab("Models") +
    ylab("Value") +
    facet_grid(set ~ metric) +
    scale_colour_brewer(type = "qual",palette = 2,
                        name = "Feature subset") +
    scale_y_continuous(
        labels = function(x) sprintf("%.2f%%",x*100)) +
    theme(legend.key.height = unit(0.2,"cm"),
          legend.key.width = unit(0.2,"cm"),
          panel.background = element_rect(
            fill = NA,colour = "black")) +
    coord_flip() +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm")) +
    guides(colour = guide_legend(nrow = 2))
ggsave(filename="figures/metrics_test.png",
       height=7,width=7)
```

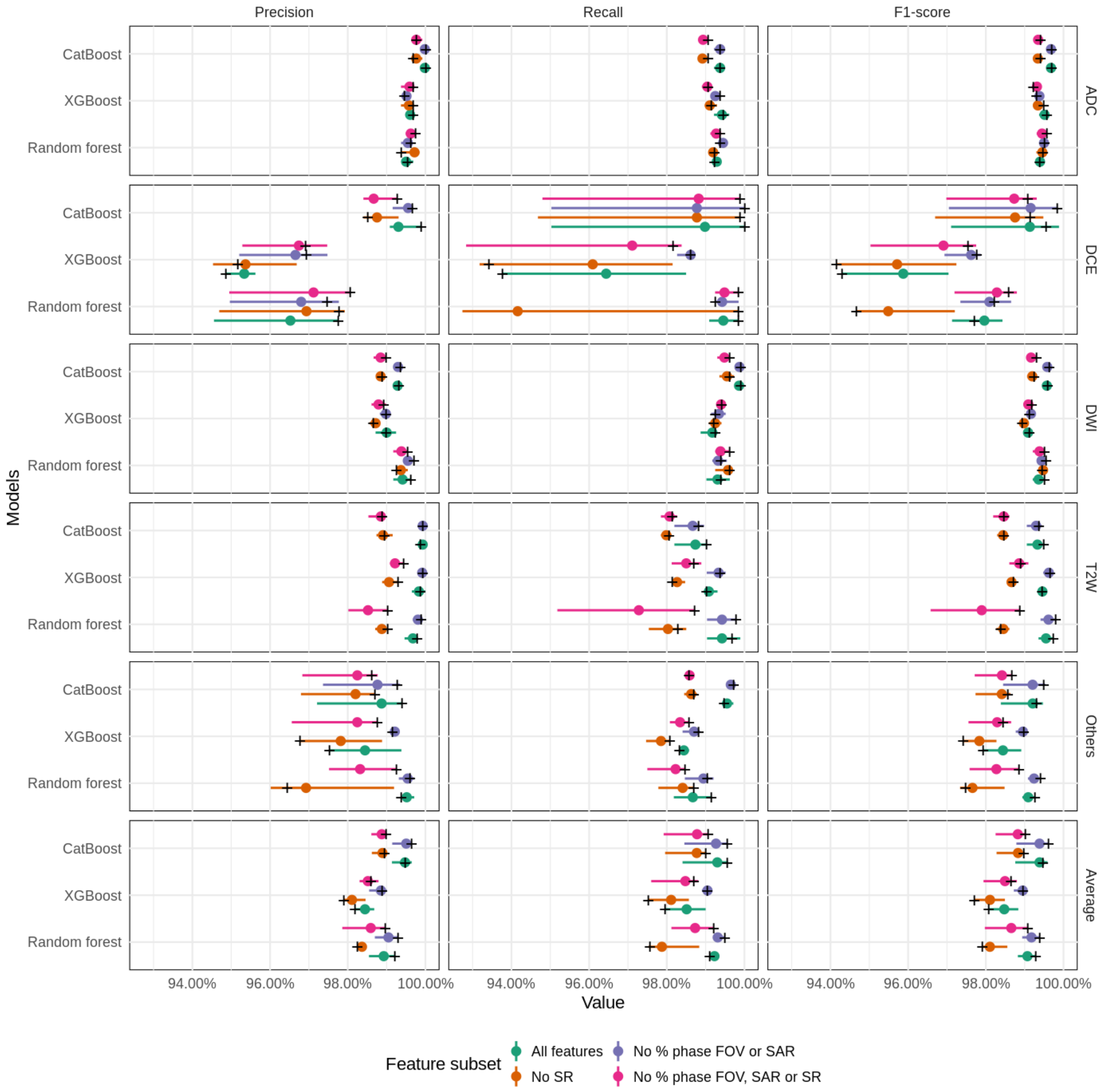
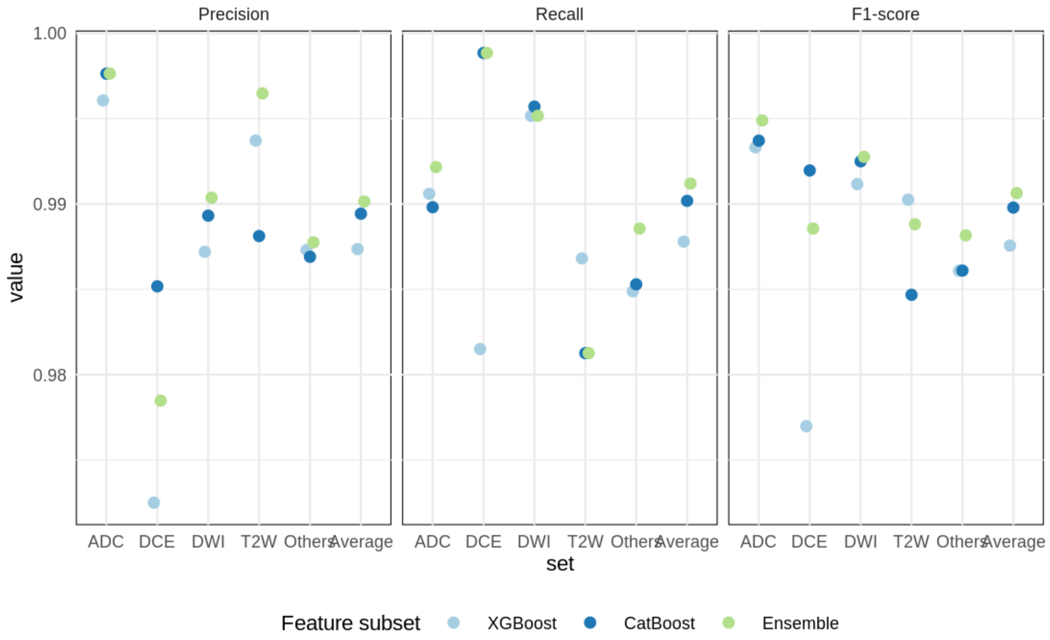


Figure 12: Hold-out test-set precision, recall and F1-score.

```
tmp_df <- all_metrics_test_ensemble %>%
    subset(metric %in% M) %>%
    subset(exclusion == "No % phase FOV, SAR or SR") %>%
    mutate(model = "Ensemble")
```

```
tmp_df <- all_metrics_test_consensus %>%
    subset(metric %in% M) %>%
    subset(model %in% c("XGBoost","CatBoost") &
            exclusion == "No % phase FOV, SAR or SR") %>%
    rbind(tmp_df) %>%
    mutate(model = factor(
        model,
        levels = c("XGBoost","CatBoost","Ensemble")))

tmp_df %>%
    ggplot(aes(x = set,y = value,colour = model)) +
    geom_point(position=position_dodge(width=0.2)) +
    facet_grid(~ metric) +
    theme_minimal(base_size = 8) +
    scale_colour_brewer(type = "qual",palette = 3,
                        name = "Feature subset") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"),
          panel.background = element_rect(
            fill=NA,colour="black"))
```



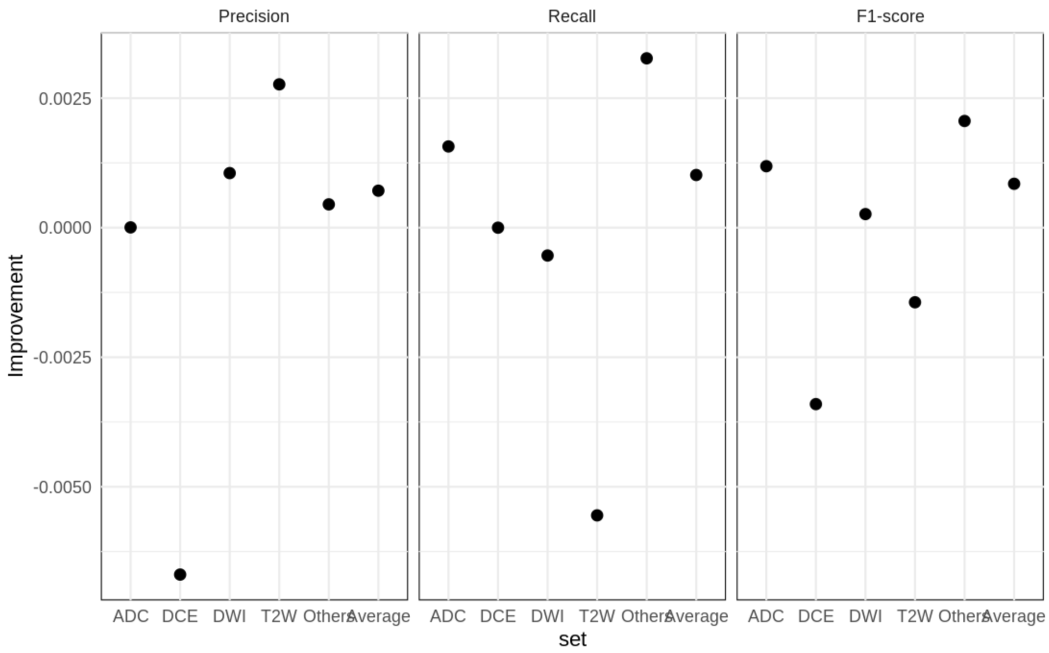
```
tmp_df %>%
    group_by(metric,set) %>%
    summarise(Improvement = value[model == "Ensemble"] - max(value[model != "Ensemble"]))
    group_by(metric) %>%
    subset(set != "Average") %>%
    summarise(`Average Improvement` = mean(Improvement))
```

```
`summarise()` has grouped output by 'metric'. You can override using the
`.groups` argument.
```

```
# A tibble: 3 x 2
  metric    `Average Improvement`
  <fct>                     <dbl>
1 Precision             -0.000484
2 Recall                -0.000251
3 F1-score              -0.000267
```

```
tmp_df %>%
    group_by(metric,set) %>%
    summarise(Improvement = value[model == "Ensemble"] - max(value[model != "Ensemble"]))
    ggplot(aes(x = set,y = Improvement)) +
    geom_point(position=position_dodge(width=0.2)) +
    facet_grid(~ metric) +
    theme_minimal(base_size = 8) +
    scale_colour_brewer(type = "qual",palette = 3,
                        name = "Feature subset") +
    theme(legend.position = "bottom",
          legend.key.height = unit(0.4,"cm"),
          panel.background = element_rect(
            fill=NA,colour="black"))
```

```
`summarise()` has grouped output by 'metric'. You can override using the
`.groups` argument.
```



Feature importance

```
feature_importance <- read_csv("../data_output/feature_importances.csv") %>%
    mutate(sub_feature = str_match(feature,":.*"),
           feature = ifelse(
            grepl(":",feature),
            gsub(":","",str_match(feature,".*:")),
            feature)) %>%
    group_by(model,feature,class,fold,exclusion) %>%
    summarise(value = value[which.max(abs(value))]) %>%
    mutate(model = factor(
        model,
        levels = c("rf","extra_trees","xgb","catboost"),
        labels = c("Random forest","Extra trees","XGBoost","CatBoost"))) %>%
    mutate(exclusion = factor(exclusion,
                              levels=names(exclusion_match),
                              labels=exclusion_match))
```

```
Rows: 34870 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
```

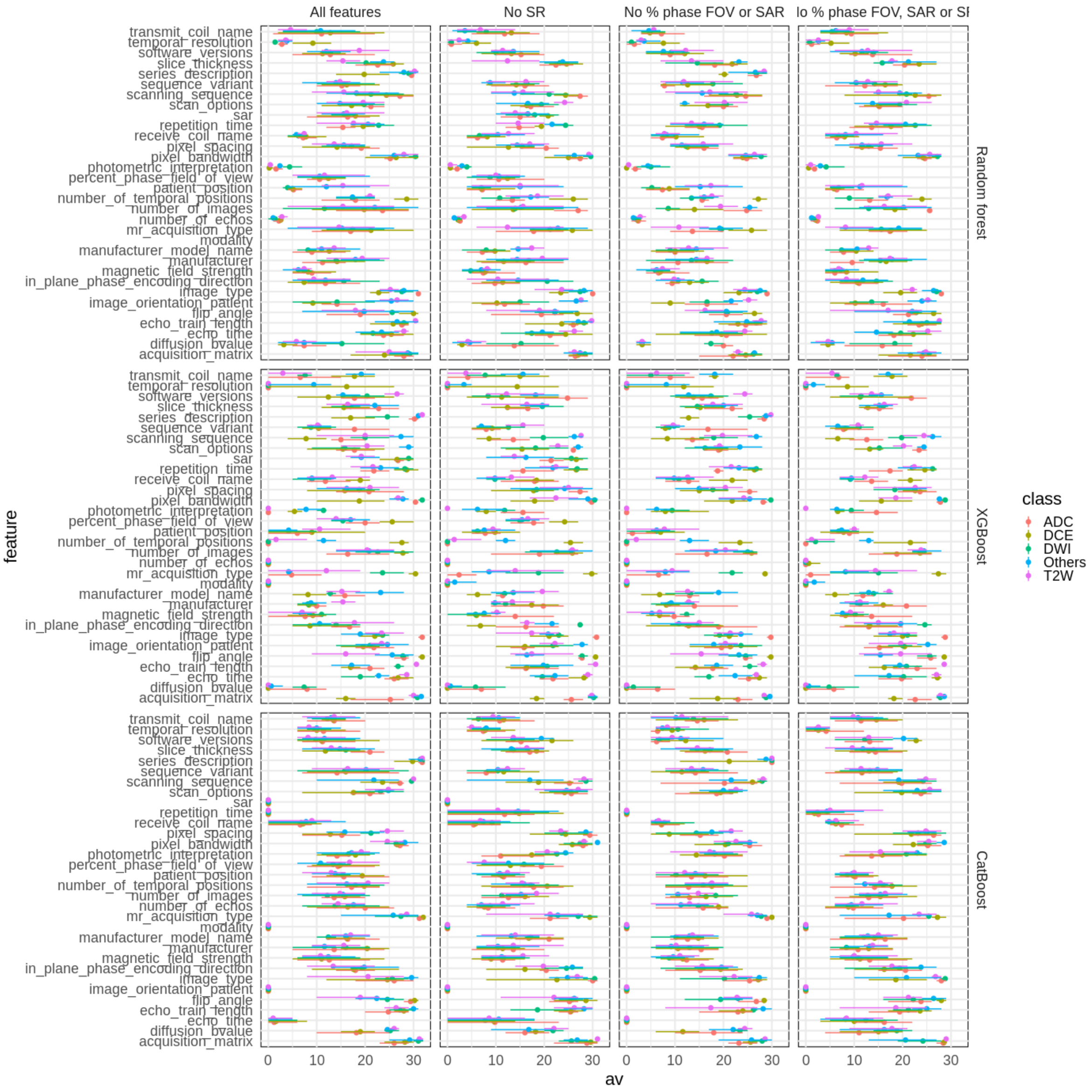
```
chr (4): model, exclusion, class, feature
dbl (2): value, fold

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
`summarise()` has grouped output by 'model', 'feature', 'class', 'fold'. You can override usi
```

Within fold consistency

```r
feature_importance %>%
    group_by(model,fold,exclusion,class) %>%
    mutate(R = sign(value) * rank(abs(value))) %>%
    group_by(model,feature,exclusion,class) %>%
    summarise(
        av = mean(abs(R)),m = min(abs(R)),M = max(abs(R))) %>%
    ggplot(aes(feature,y = av,ymin = m,ymax = M,colour = class)) +
    geom_point(size = 0.5,position = position_dodge(0.5)) +
    geom_linerange(size = 0.25,position = position_dodge(0.5)) +
    facet_grid(model ~ exclusion) +
    coord_flip() +
    theme_minimal(base_size = 8) +
    theme(legend.key.height = unit(0.2,"cm"),
          legend.key.width = unit(0.2,"cm"),
          panel.background = element_rect(
            fill = NA,colour = "black"))
```

```
`summarise()` has grouped output by 'model', 'feature', 'exclusion'. You can
override using the `.groups` argument.
```



```
ggsave(filename="figures/feature_importance.png",
       height=7,width=7)
```

```
library(cowplot)

best_folds <- all_metrics %>%
    filter(metric == "auc" & split == "cv") %>%
    subset(best_fold) %>%
    ungroup %>%
    select(model,exclusion,fold) %>%
    distinct

plot_df <- merge(
    feature_importance,best_folds,
    by = c("model","exclusion","fold"),all=F) %>%
    group_by(model,exclusion,class) %>%
    mutate(R = sign(value) * rank(abs(value))) %>%
    group_by(model,feature,exclusion,class) %>%
    summarise(value = mean(value),
              average_rank = mean(R)) %>%
    group_by(model,feature,exclusion) %>%
    mutate(order = sum(abs(value)))
```

```
`summarise()` has grouped output by 'model', 'feature', 'exclusion'. You can
override using the `.groups` argument.
```

```
all_plots <- list()

for (m in unique(plot_df$model)) {
    for (exc in unique(plot_df$exclusion)) {
        str_ <- sprintf("%s %s",m,exc)
        all_plots[[str_]] <- plot_df %>%
            subset(model == m & exclusion == exc) %>%
            ggplot(aes(reorder(feature,order),
                   y = value,
                   colour = class)) +
            geom_point(size = 0.5) +
            coord_flip() +
            theme_minimal(base_size = 8) +
            theme(legend.key.height = unit(0.2,"cm"),
                  legend.key.width = unit(0.2,"cm"),
                  panel.background = element_rect(
                    fill = NA,colour = "black"),
                  legend.position = "bottom") +
            ggtitle(m,subtitle=exc) +
            xlab("") +
            ylab("SHAP value") +
            scale_colour_brewer(
                type = "qual",palette = 3,
                name = NULL)
    }
}

plot_grid(plotlist=all_plots[1:4])
```

```r
ggsave(
    filename="figures/feature_importance_best_fold_1.png",
    height=6,width=8)

plot_grid(plotlist=all_plots[5:8])
```

```
ggsave(
    filename="figures/feature_importance_best_fold_2.png",
    height=6,width=8)

plot_grid(plotlist=all_plots[9:12])
```

```
ggsave(
    filename="figures/feature_importance_best_fold_3.png",
    height=6,width=8)
```

Discussion

The consensus is relatively easy to reach - CatBoost is, by far, the best performing model out of the two. The advantages are considerable - the optimization is simpler and no preprocessing is required. While it takes longer to train, one can be sure that the results will be hard to dispute.