

Supercapacitors - Analysis of Graphene Electrode Data

Marcin Szulc

2025-12-10

- Introduction
- Loading data
- Basis statistics
- Distribution analysis
- Coleration analysis
- Interactive plots
- Building train and test models
- Random forest
- Model evaluation
- Using Machine Learning

Introduction

Analizowany zbiór danych zawiera 925 rekordów oraz 21 cech opisujących elektrody grafenowe, stosowane w superkondensatorach. Po wykonaniu raportu można stwierdzić najważniejszy fakt, którym jest brak dużej ilości danych. Ten problem został rozwiązany poprzez wpisaniu 0 w brakujących miejscach. Najbardziej kompletne zbiory to zbiór "electrolyte concentration" oraz "electro conductivity". Analiza rozkładu kolumny o nazwie "capacitance" wykazała znaczną zmienność w danych. Macierz korelacji cech numerycznych nie ujawniła silnej zależności pomiędzy analizowanymi parametrami. Korzystając z Nested cross-Validation zbiór został podzielony na 30% danych testowych oraz 70% danych uczących. Dzięki temu model nie zapamiętuje wyników z przeprowadzonego eksperymentu i po wprowadzeniu biblioteki iml jest w stanie działać na "świeżych danych". Dane uzyskane po machine learningu pokazują, że predykcja modelu jest na poziomie około 68%. Taki wynik pokazuje, że model jest w stanie wychwycić dużą część zależności występujących w zbiorze danych, ale nie wszystkie.

Loading data

```
data <- tryCatch(read_data("data.csv"), error = function(e) {  
  message(e$message)  
  data[is.na(data)] <- 0  
  NULL  
})  
  
if(is.null(data)) stop()  
  
glimpse(data)
```

```

## Rows: 925
## Columns: 21
## $ ref
## $ limits_of_potential_window_v
## $ lower_limit_of_potential_window_v
## $ upper_limit_of_potential_window_v
## $ potential_window_v
## $ current_density_a_g
## $ capacitance_f_g
## $ specific_surface_area_m_2_g
## $ charge_transfer_resistance_rct_ohm
## $ equivalent_series_resistance_rs_ohm
## $ electrode_configuration
## $ pore_size_nm
## $ pore_volume_cm_3_g
## $ ratio_of_id_iq
## $ n_at_percent
## $ c_at_percent
## $ o_at_percent
## $ electrolyte_chemical_formula
## $ electrolyte_ionic_conductivity
## $ electrolyte_concentration_m
## $ cell_configuration_three_two_electrode_system <chr> "three-electrode system"
<chr> "DOI: 10.1039/c7ta03093b..."
<chr> "0 to 0.8", "0 to 1", "0...
<dbl> 0.0, 0.0, 0.0, 0.0, 0.0, ...
<dbl> 0.80, 1.00, 1.00, 1.00, ...
<dbl> 0.80, 1.00, 1.00, 1.00, ...
<dbl> 1.0, 1.0, 2.0, 5.0, 10.0...
<dbl> 680, 367, 338, 283, 246, ...
<dbl> 186.3, 537.0, 537.0, 537...
<dbl> NA, 6.1, 6.1, 6.1, 6.1, ...
<dbl> 7.70, 1.95, 1.95, 1.95, ...
<chr> "CNF/RGO/moOxNy", "sulfu...
<dbl> NA, NA, NA, NA, NA, NA, ...
<dbl> NA, NA, NA, NA, NA, NA, ...
<dbl> 1.450, 1.280, 1.280, 1.2...
<dbl> 2.1, 0.0, 0.0, 0.0, 0.0, ...
<dbl> NA, 85.6, 85.6, 85.6, 85...
<dbl> NA, 9.1, 9.1, 9.1, 9.1, ...
<chr> "H2SO4", "KOH", "KOH", ...
<int> 7, 6, 6, 6, 6, 6, NA, NA...
<dbl> 1.0, 6.0, 6.0, 6.0, 6.0, ...

```

Basis statistics

```

n_obs <- nrow(data); n_vars <- ncol(data)
cat(paste0("Liczba obserwacji: ", n_obs, "\nLiczba zmiennych: ", n_vars, "\n"))

```

```

## Liczba obserwacji: 925
## Liczba zmiennych: 21

```

```

# Using skimr
skimr::skim(data)

```

Data summary

Name	data
Number of rows	925
Number of columns	21

Column type frequency:

character	5
numeric	16

Group variables

None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ref	0	1.00	20	38	0	198	0
limits_of_potential_window_v	4	1.00	6	13	0	63	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
electrode_configuration	0	1.00	2	104	0	353	0
electrolyte_chemical_formula	22	0.98	3	54	0	23	0
cell_configuration_three_two_electrode_system	14	0.98	20	22	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
lower_limit_of_potential_window_v	4	1.00	-0.23	0.37	-1.10	-0.30	0.00	0.00	0.20	
upper_limit_of_potential_window_v	4	1.00	0.63	0.45	-0.20	0.40	0.60	0.80	3.50	
potential_window_v	5	0.99	0.86	0.35	0.40	0.60	0.82	1.00	3.50	
current_density_a_g	16	0.98	5.86	13.35	0.05	1.00	2.00	5.00	200.00	
capacitance_f_g	17	0.98	415.50	447.53	1.40	148.60	260.25	509.85	3344.08	
specific_surface_area_m_2_g	572	0.38	417.44	546.58	8.90	57.00	159.97	546.00	2400.00	
charge_transfer_resistance_rct_ohm	786	0.15	3.05	4.61	0.08	0.67	1.54	3.24	24.20	
equivalent_series_resistance_rs_ohm	772	0.17	1.60	2.43	0.20	0.35	0.58	2.00	17.50	
pore_size_nm	769	0.17	8.62	8.10	0.53	3.04	4.34	13.62	44.13	
pore_volume_cm_3_g	729	0.21	0.49	0.59	0.02	0.17	0.22	0.51	2.35	
ratio_of_id_ig	596	0.36	1.12	0.43	0.12	0.94	1.05	1.17	2.90	
n_at_percent	690	0.25	2.50	4.57	0.00	0.00	0.00	3.20	23.82	
c_at_percent	699	0.24	66.52	28.66	1.40	37.32	81.00	85.57	98.10	
o_at_percent	703	0.24	19.18	14.49	1.90	8.88	13.70	27.10	54.28	
electrolyte_ionic_conductivity	99	0.89	5.81	1.39	1.00	6.00	6.00	7.00	8.00	
electrolyte_concentration_m	62	0.93	2.58	2.19	0.10	1.00	1.00	6.00	6.00	

table(data\$cell_configuration)

< table of extent 0 >

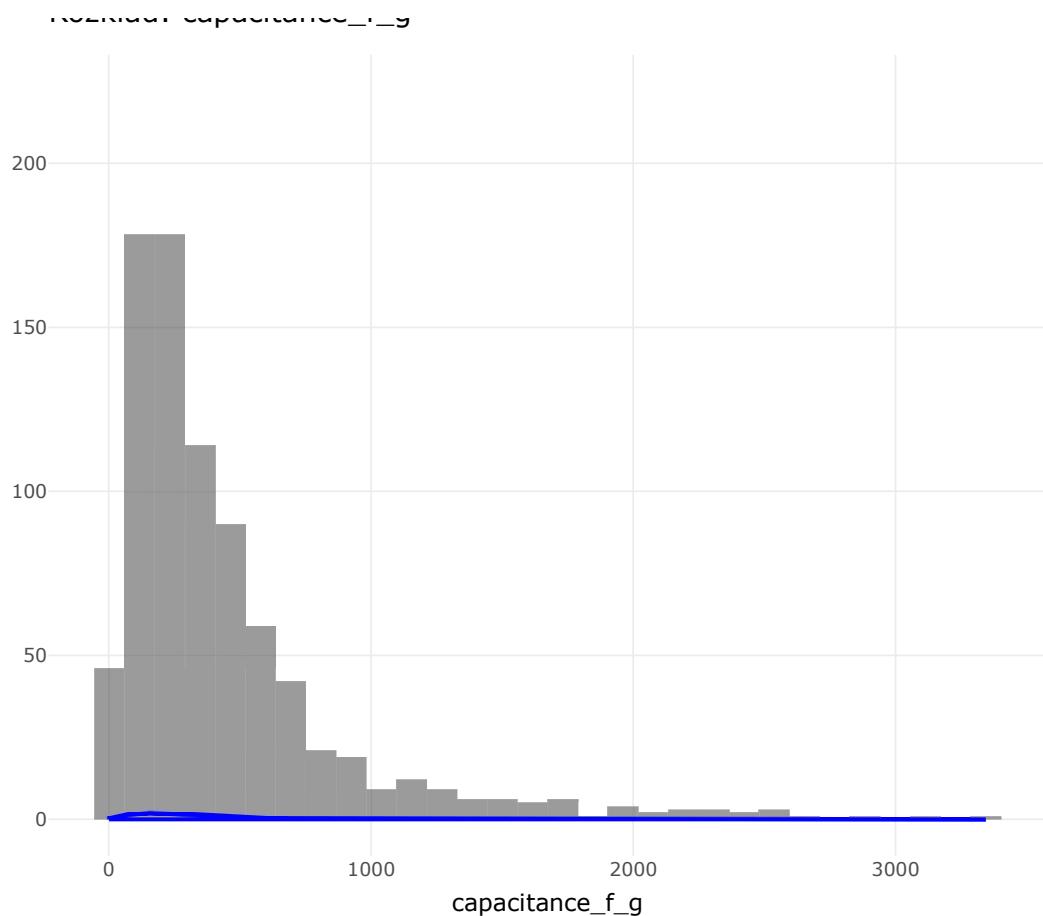
Distribution analysis

```
# Histograms
plot_vars <- c('capacitance_f_g','specific_surface_area_m2_g','pore_size_nm','pore_volume_cm3_g','charge_transfer_resistance_rct_ohm','equivalent_series_resistance_rs_ohm')
plot_vars <- intersect(plot_vars, names(data))

plots <- lapply(plot_vars, function(v){
  p <- ggplot(data, aes_string(x = v)) +
    geom_histogram(bins = 30, alpha = 0.6) +
    geom_density(aes(y = ..count..), color = "blue", size = 0.7) +
    ggtitle(paste("Rozkład:", v)) + theme_minimal()
  ggplotly(p)
})

# First plot as example
if(length(plots)>0) plots[[1]]
```

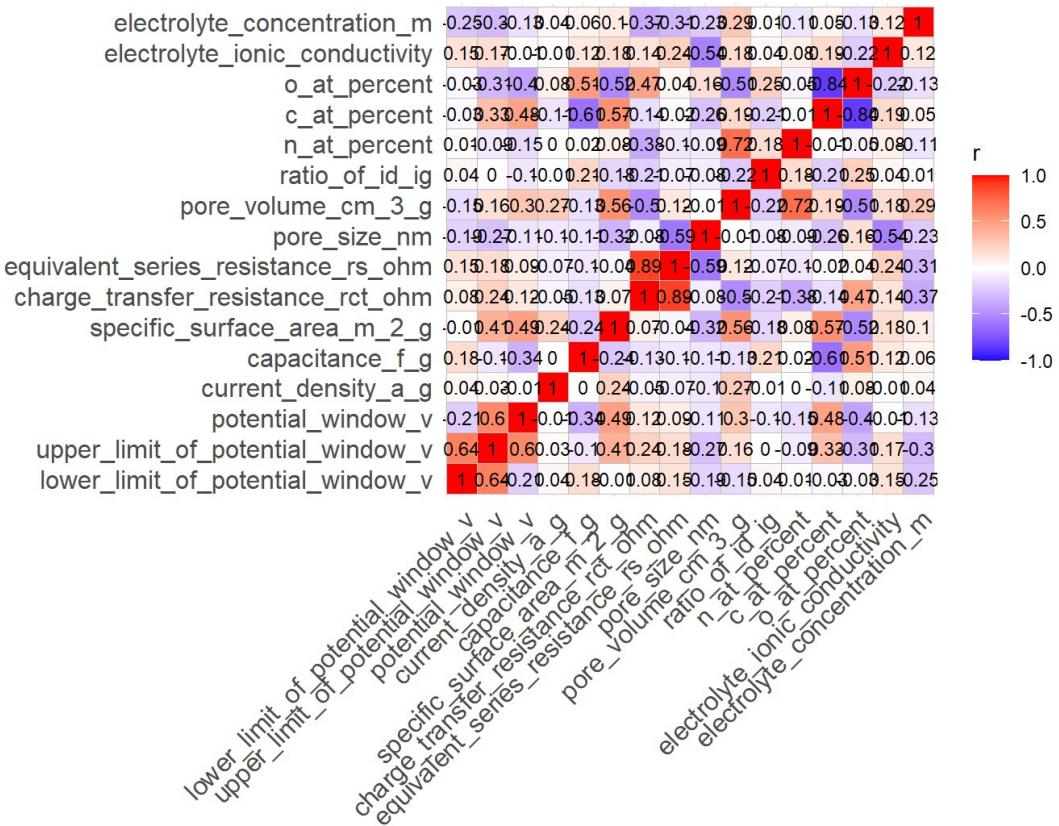
Rozkład·capacitance f g



Correlation analysis

```
num_data <- data %>% select(where(is.numeric))
cor_mat <- cor(num_data, use = 'pairwise.complete.obs')

# Heatmap
ggcorrplot::ggcorrplot(cor_mat, lab = TRUE, lab_size = 3, legend.title = "r")
```



```
# Interactive plot
if(ncol(num_data) <= 10){
  gg <- GGally::ggpairs(num_data)
  print(gg)
}
```

Interactive plots

```
if(all(c('specific_surface_area_m2_g', 'capacitance_f_g') %in% names(data))){
  p <- ggplot(data, aes(x = specific_surface_area_m2_g, y = capacitance_f_g, text = paste('Ref:', ref))) +
    geom_point() + theme_minimal() + ggtitle('SSA vs Capacitance')
  ggplotly(p)
}
```

Building train and test models

```

for (col in names(data)) {
  if (is.numeric(data[[col]])) {
    data[[col]][is.na(data[[col]])] <- 0
  } else if (is.character(data[[col]])) {
    data[[col]][is.na(data[[col]])] <- ""
  } else if (is.factor(data[[col]])) {
    levels(data[[col]]) <- c(levels(data[[col]]), "Unknown")
    data[[col]][is.na(data[[col]])] <- "Unknown"
  }
}

set.seed(123)

train_index <- sample(seq_len(nrow(data)), size = 0.7 * nrow(data))

train_data <- data[train_index, ]
test_data <- data[-train_index, ]

cat("Train size:", nrow(train_data), "\n")

```

```
## Train size: 647
```

```
cat("Test size:", nrow(test_data), "\n")
```

```
## Test size: 278
```

Random forest

```

rf_model <- randomForest(
  capacitance_f_g ~ .,
  data = train_data,
  ntree = 400,
  importance = TRUE
)

rf_model

## 
## Call:
##   randomForest(formula = capacitance_f_g ~ ., data = train_data,      ntree = 400, importance = TRUE)
##   Type of random forest: regression
##   Number of trees: 400
##   No. of variables tried at each split: 6
## 
##   Mean of squared residuals: 47809.28
##   % Var explained: 76.29

```

Model evaluation

```

pred <- predict(rf_model, newdata = test_data)

mae <- mean(abs(pred - test_data$capacitance_f_g))
rmse <- sqrt(mean((pred - test_data$capacitance_f_g)^2))
r2 <- cor(pred, test_data$capacitance_f_g)^2

cat("MAE:", mae, "\n")

```

```
## MAE: 162.8565

cat("RMSE:", rmse, "\n")

## RMSE: 251.2584

cat("R^2:", r2, "\n")

## R^2: 0.6836475
```

Using Machine Learning

```
X_train <- train_data %>% select(-capacitance_f_g)
y_train <- train_data$capacitance_f_g

predictor <- Predictor$new(
  model = rf_model,
  data = X_train,
  y = y_train
)

shap <- Shapley$new(predictor, x.interest = X_train[1, ])
shap$plot()
```

Actual prediction: 1104.48
 Average prediction: 405.77

