```
function boostedrirt(data) {
    checking_inputs(data)
    rmse_results               # Ergebnisobjekt initialisieren
    param_grid <- expand.grid(subsample, colsample)  # Tuning-Grid

    for (param in param_grid) {   # ueber Parameter iterieren
        cv_data <- cv_ziehen(datensatz)   # Crossvalidation-Folds

        for (i in 1:folds) {   # ueber CV-Folds iterieren
            test_data  <- cv_data[i]
            train_data <- cv_data[-i]

            f_train <- ...   # Schritt 1: Initialisierung
            f_test  <- ...
            improv_count <- 0
            iter <- 1

            while (iter <= itermax && improv_count < 5) # Schritt 2: adaptierter EM-Algorithmus
                train_data$neg_grad <- berechne_neg_grad(train_data)          # b)
                para_data <- ziehe_stichprobe(train_data, param)              # a)
                cart_modell <- schaetze_cart(para_data)                       # c)
                para_data$temp_res <- berechne_temp_res(para_data, cart_modell, f_train)  # d)
                lmm_modell <- schaetze_lmm(para_data)                         # e)

                train_data$f_iter_act <- berechne_aktuelle_vorhersage(
                    train_data, lmm_modell, f_train, cart_modell)             # f)
                test_data$f_iter_act <- berechne_aktuelle_vorhersage(
                    test_data, lmm_modell, f_test, cart_modell)

                f_train <- train_data$f_iter_act # Aktualisieren von f train
                rmse_train <- berechne_rmse(train_data)
                rmse_test  <- berechne_rmse(test_data)                        # g)

```

```
                if (iter > 1 && rmse_test >= min(rmse_
                    results[rmse_results$fold == i,]$rmse_
                    test)) {
                    improv_count <- improv_count + 1
                }
                if (improv_count == 5) { iter_final <- iter
                    - 5 }
                if (iter == itermax)  { iter_final <- iter }

                rmse_results  # speichern
                iter <- iter + 1
            }
        }
    }

    final_model <- berechne_finales_model(data, rmse_results
        )   # h)
}
```