use anyhow::Result;

use log::info;

use navi::cli\_args::{ARGS, MODEL\_SPECS};

use navi::onnx\_model::onnx::OnnxModel;

use navi::{bootstrap, metrics};

fn main() -> Result<()> {

env\_logger::init();

info!("global: {:?}", ARGS.onnx\_global\_thread\_pool\_options);

let assert\_session\_params = if ARGS.onnx\_global\_thread\_pool\_options.is\_empty() {

// std::env::set\_var("OMP\_NUM\_THREADS", "1");

info!("now we use per session thread pool");

MODEL\_SPECS.len()

}

else {

info!("now we use global thread pool");

0

};

assert\_eq!(assert\_session\_params, ARGS.inter\_op\_parallelism.len());

assert\_eq!(assert\_session\_params, ARGS.inter\_op\_parallelism.len());

metrics::register\_custom\_metrics();

bootstrap::bootstrap(OnnxModel::new)

}