Decision-Focused Carbon Forecasting

A Lightweight README

1 Project Overview

This repository contains a complete pipeline for training, evaluating, and analyzing **Temporal Fusion Transformer (TFT)** models together with **decision-focused** simulations for carbon-intensity forecasting across ERCOT, NYISO, and PJM.

The workflow has three stages:

1. Data loading and TFT model training.
2. Exporting multi-step forecasts.
3. Running decision simulations and summarizing results.

2 Folder Structure

```
code/
  configs/               -- JSON configs for each region
  src/
    data_loader.py
    build_dataset.py
    train_tft.py
    export_forecasts.py
    decision_simulate.py
  results/
    figures/
    decision/decision_summary/
data/
  ERCO/
  NYISO/
  PJM/
readme.tex
```

3 Data Placement

Place your data under:

```
data/ERCO/
data/NYISO/
data/PJM/
```

Each region must contain these files (CarbonCast format):

- `<REGION>_96hr_forecasts_DA.csv`
- `<REGION>_direct_emissions.csv`

- <REGION>_weather_forecast.csv
- (Optional) <REGION>_lifecycle_emissions.csv

4 Editing Config Files

The config files are located in code/configs/*.json.

Example fields that typically require modification:

```
"paths": {
    "forecast_csv": "data/ERCO/ERCO_96hr_forecasts_DA.csv",
    "emission_csv": "data/ERCO/ERCO_direct_emissions.csv",
    "weather_csv": "data/ERCO/ERCO_weather_forecast.csv"
}
```

You must ensure that all paths point to the files in your data/ directory.

5 Training All Regions

Run:

```
bash run_all.sh
```

This executes TFT training for ERCOT, NYISO, and PJM sequentially.

6 Exporting Forecasts

After each region is trained, export validation forecasts:

```
python -m src.export_forecasts \
   --config code/configs/ercot.json \
   --run_dir results_ercot/default/<timestamp>
```

This produces:

```
forecasts_val.parquet
```

7 Decision Simulation

Run a simulation for a region:

```
python -m src.decision_simulate \
    --run_dir results_ercot/default/<timestamp> \
    --region ERCOT
```

This outputs:

```
decision_sim.parquet
```

8 Summaries and Plots

Summaries can be generated via:

```
python -m plot_decision_results.py
```

Generated artifacts will appear in:

```
results/decision/decision_summary/
results/figures/
```

9 Dependencies

Requirements are listed in:

`requirements.txt`

Install with:

```
pip install -r requirements.txt
```

10 Notes

- All file paths in JSON configs must be updated if you move your data.
- The forecasting stage and decision stage are independent; you may run them separately.
- The repository contains no training data due to size limits.