Ghana Data Centre and Ghana R Users Conference

2025

Theme

Harnessing R for Sustainable Development: Innovations collaborations and Health Impacts

Ghana R Users Community & NHR – GSU
Ghana Hub Data Centre

GlobalSurg | NIHR Global Health Research Unit on Global Surgery

MACHINE LEARNING IN ENVIRONMENTAL MONITORING: An R Perspective

Presented By

George Kyei Agyen
gkagyen@ghana-rusers.org

Ghana R Users Community

Content Outline

Part 1:
Env. modelling & Machine Learning

- □ Overview of Environmental Monitoring
- □ Role of Machine Learning in Env. Monitoring
- □ Types of machine learning
- □ Why Use R
- □ Key Packages to Consider

Part 2:
ML Workflow Live Coding Demo

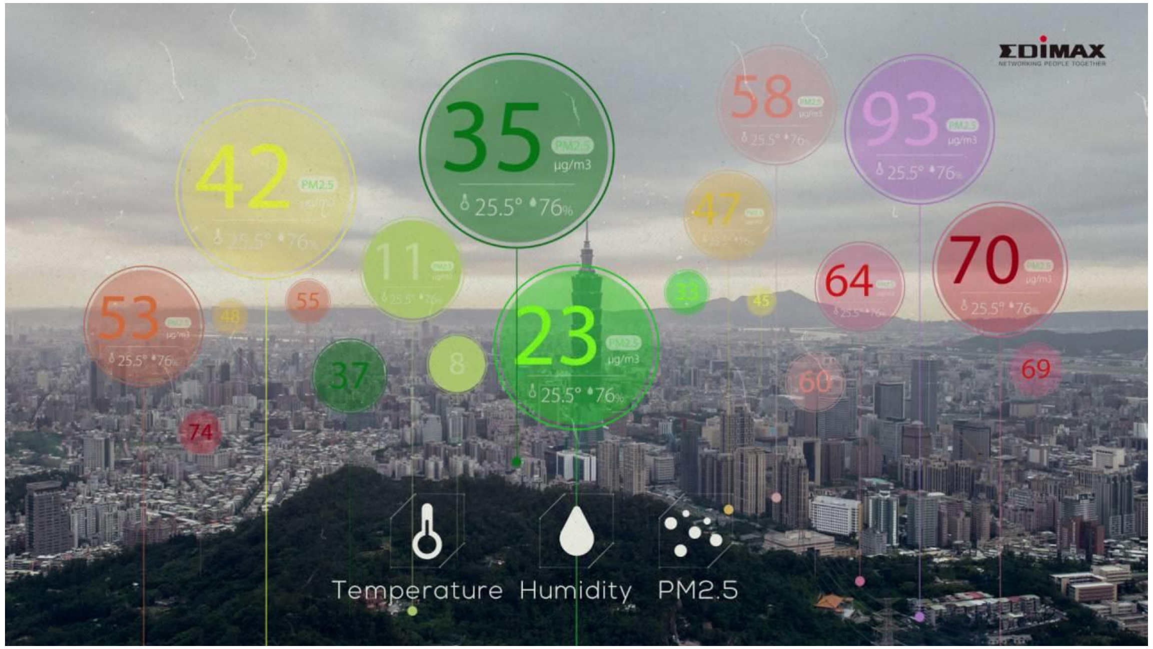
- □ Tidymodels workflow code snippet
- □ Live code demonstration
- □ Useful resources

PART 1:

Environmental Modelling and ML Presentation

Overview of Environmental Monitoring

Environmental monitoring is the collection and analysis of data to manage natural resources sustainably



It involves monitoring air quality, water resources, soil conditions, wildlife populations, and weather patterns

The Role of Machine Learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that focuses on enabling computers to learn from data without being explicitly programmed.

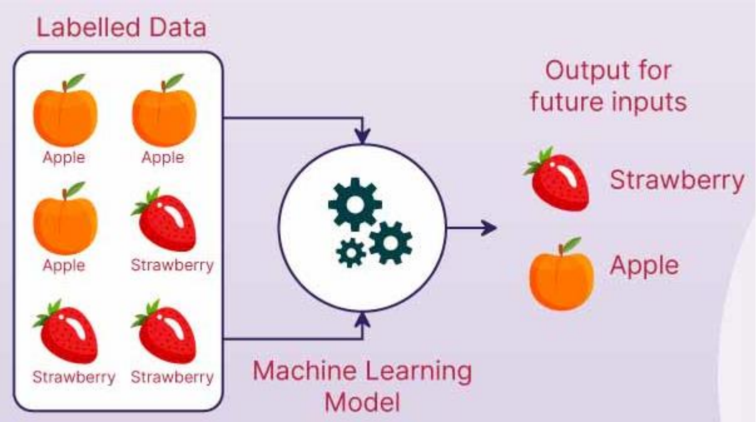
Basically, ML algorithms analyse datasets to identify patterns, learn from these patterns, and then make predictions on new unseen data

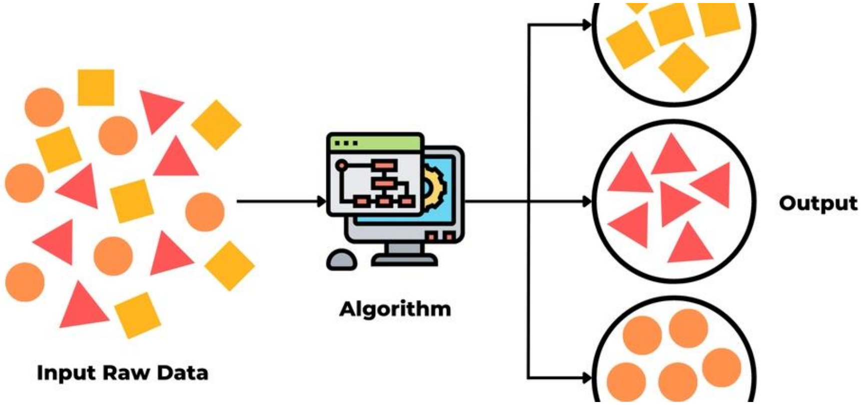
This "learning" process allows systems to improve their performance over time as they are exposed to more data

Types of Machine Learning

Supervised Learning

This method uses labelled data (input-output pairs) to predict outcomes (e.g., regression, classification).



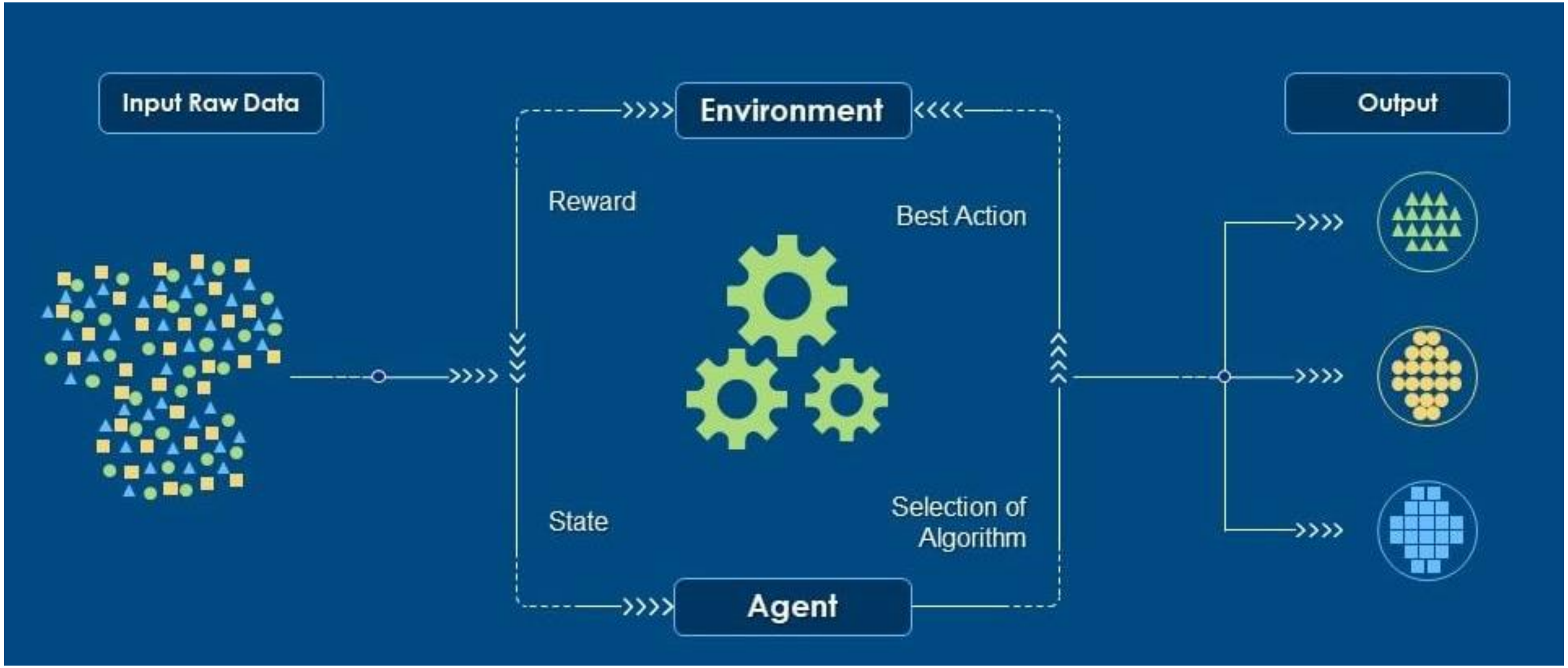


Unsupervised Learning

This approach finds patterns in unlabelled data

Reinforcement Learning

This ML method differs from supervised and unsupervised learning. Reinforcement learning (RL) focuses on learning by trial and error through interaction with an environment



The Machine Learning Workflow

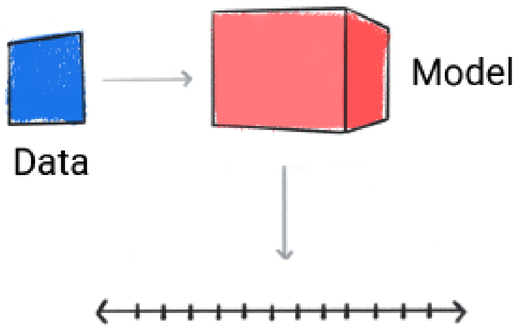
To successfully build and deploy ML models, it's essential to follow a structured process.

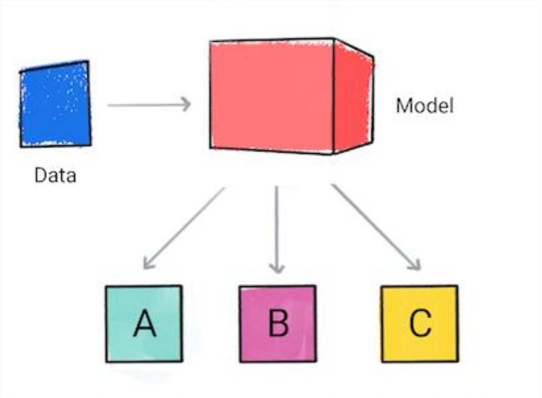
This structured process is called the **machine learning workflow**

Step 1: Problem Definition

- Clearly Define the main purpose of your ML model

- Understand the type of problem being tackled: **classification, regression, clustering**





Step 2: Data Collection

- Gather relevant data from various sources (databases, files, sensors, recording devices etc)

- Ensure the data is a representative of the real-world scenario

Step 3: Data Preprocessing

Cleaning: Handle missing values, remove duplicates and correct errors

Feature Engineering: Create new features, encode categorical data

Normalise data: standardise or normalise numeric data

Splitting: Divide the data into training and testing sets (and sometimes validation sets)

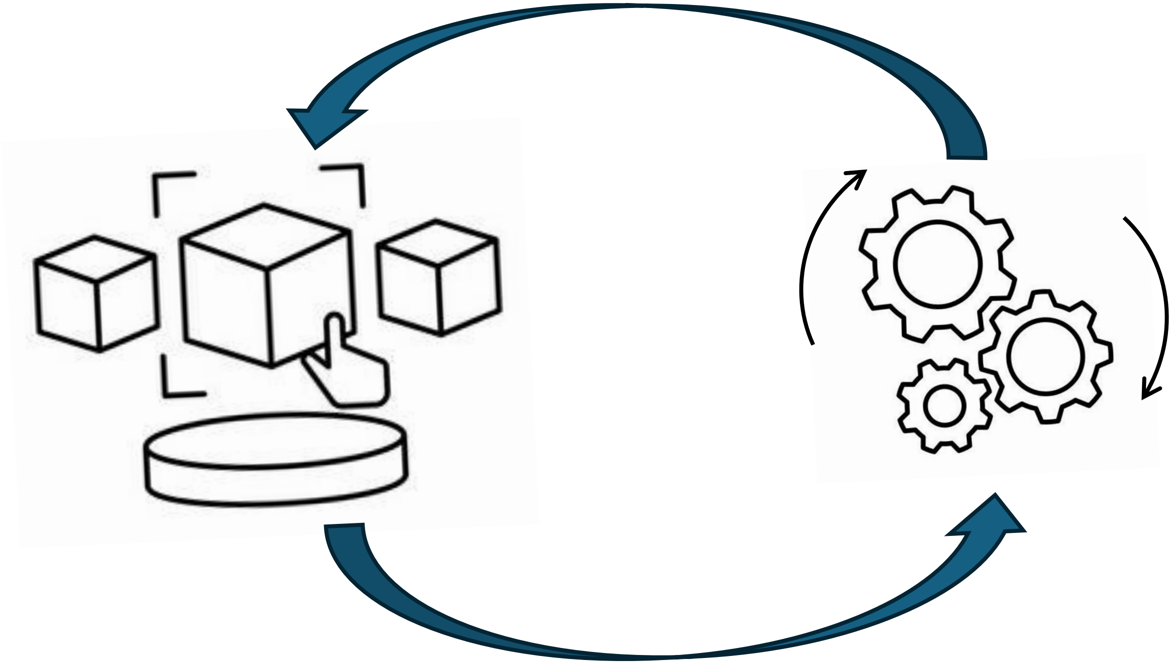
Step 4: Model Building

Model Selection

- Choose an appropriate algorithm
- Consider the complexity, interpretability, and performance of the model.

Model Training

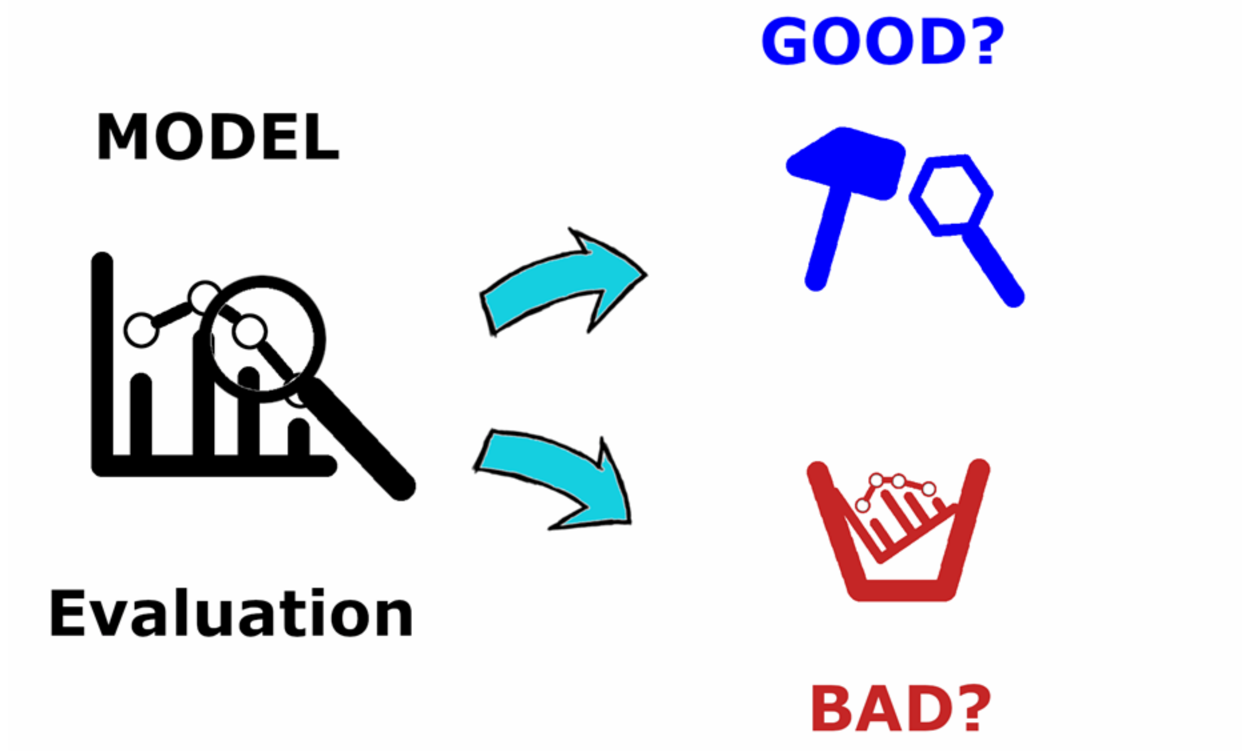
- Train model on the training dataset
- Adjust internal parameters to reduce errors or loss



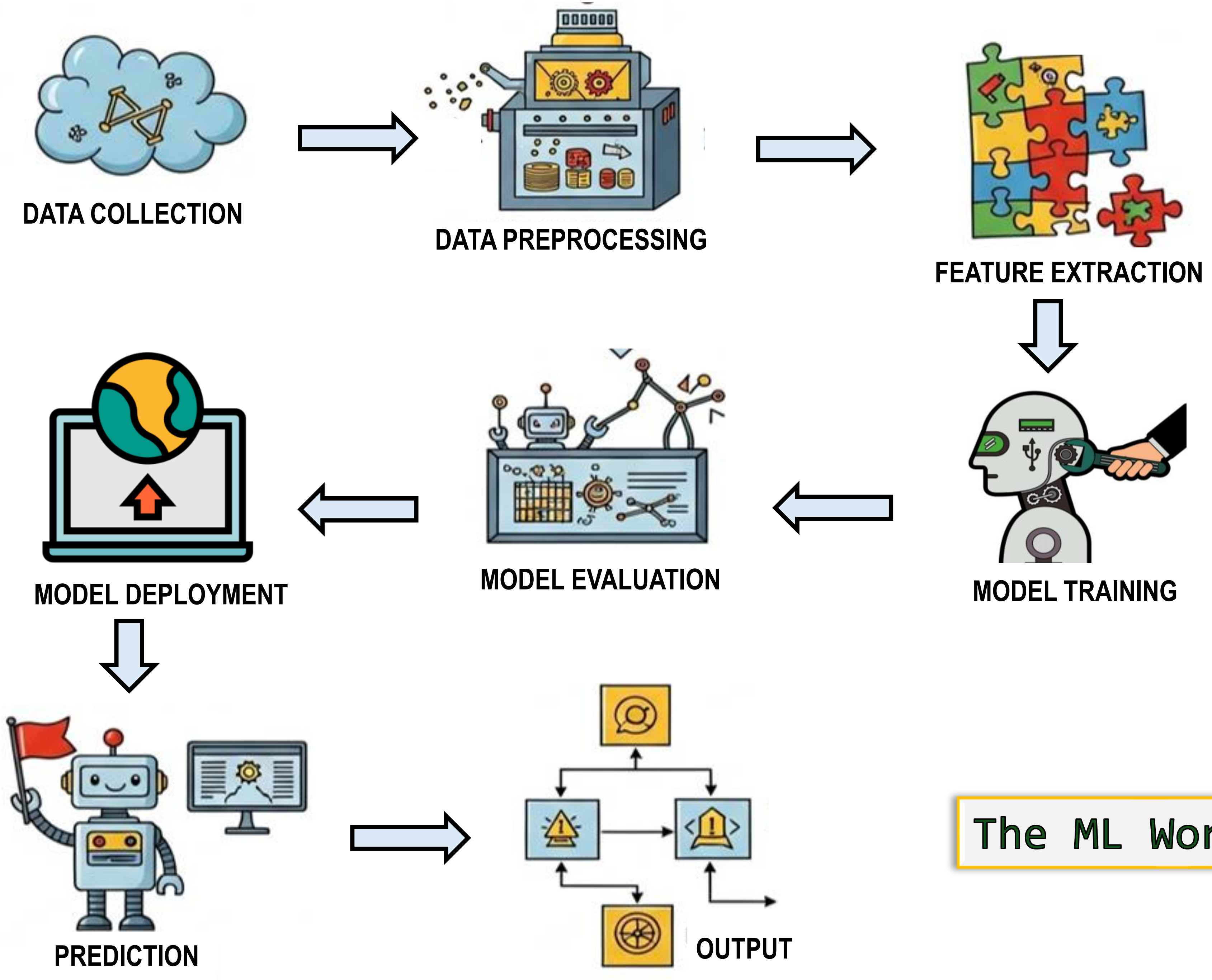
Step 5: Model Evaluation

- Use metrics like accuracy, precision, recall, F1-score, or RMSE depending on the problem.

- Evaluate on the testing set to assess generalization



The ML Workflow



DATA COLLECTION

DATA PREPROCESSING

FEATURE EXTRACTION

MODEL TRAINING

MODEL EVALUATION

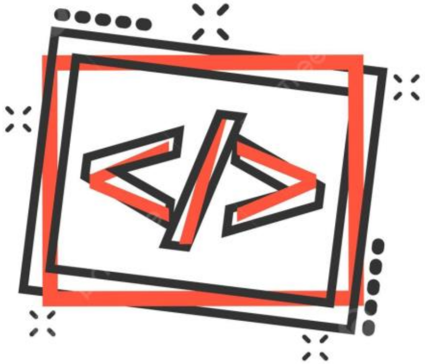
MODEL DEPLOYMENT

PREDICTION

OUTPUT

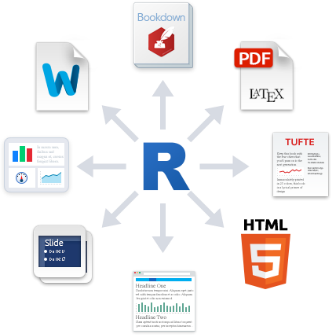
Why Use R

Free and Open Source



No licensing costs, making it accessible to individuals, researchers, and organizations of all sizes

Reproducible Research



Tools like R Markdown and its successor, Quarto, allow users to combine code, output (tables, plots), and narrative text into a single, dynamic document

Rich Packages for Analysis

R's strength lies in its extensive collection of "packages" that extend its capabilities;

```
caret, tidymodels,
randomForest, ggplot2,
tidyr, dplyr
```

Active Community Worldwide

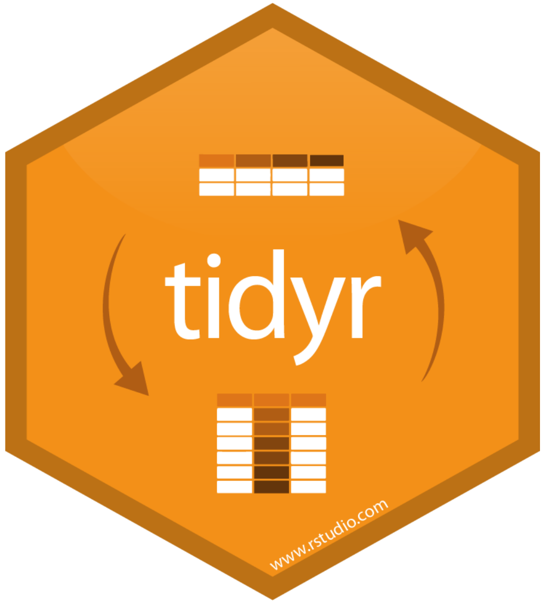
A vast and supportive global community means readily available help, tutorials, and solutions for common problems

Some Applications of ML in Environmental Monitoring

- Predictive modelling for climate forecasting, species migration, and disaster prediction
- Image classification for satellite data analysis and species identification
- Automated anomaly detection for pollution spikes or ecosystem disruptions
- Real-time processing of sensor data streams for immediate alerts
- Time series analysis for trend detection in long-term environmental data

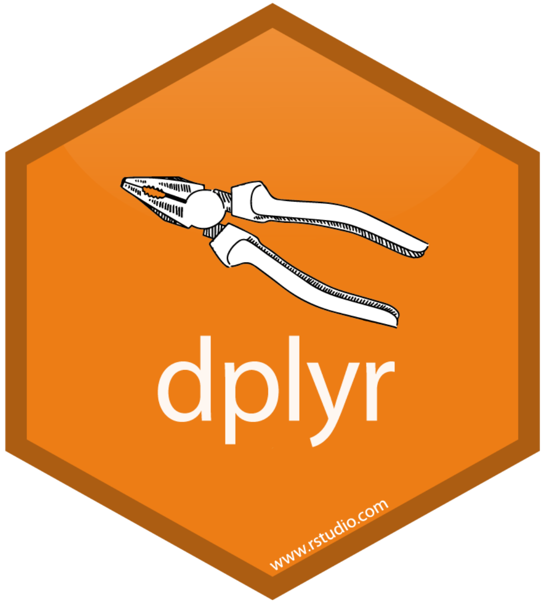
Key R Packages to Consider

Data Cleaning



The **dplyr** and **tidyr** packages are fundamental tools in the R tidyverse for efficient and expressive data manipulation and cleaning.

They promote a consistent syntax and work seamlessly together to transform messy data into a clean, tidy format suitable for analysis



Visualisation

For your visualisations in R, **`ggplot2`**, **`tmap`**, and **`leaflet`** are your go-to packages, especially for environmental monitoring.

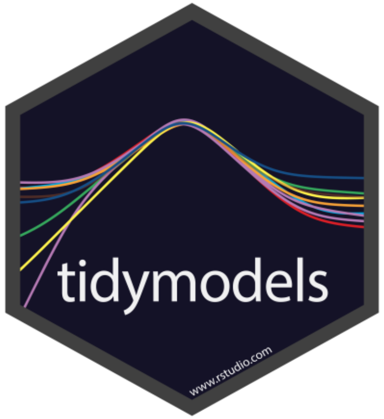
`ggplot2``, for static customisable plots`

`tmap`, for thematic static and interactive maps

`Leaflet`, for full interactive web maps

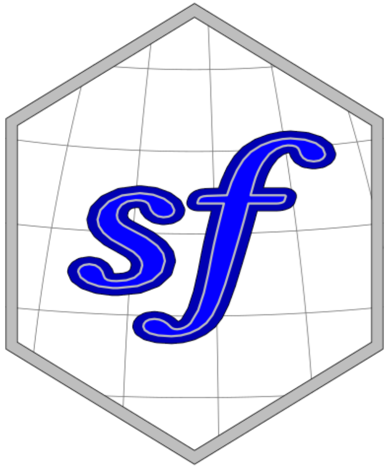
Machine Learning

caret(classification and regression training) is a well-established and comprehensive package that provides a **unified interface** to train and evaluate a vast number of machine learning models in R.



tidymodels is a newer, modular framework that brings the principles of the tidyverse to machine learning. It's a collection of interoperable packages rather than a single monolithic one.

Spatial Data Analysis



The **raster** package was the previous standard for **raster spatial data** in R. While still functional, it has largely been superseded by **terra**

terra is the successor to the **raster** package, designed to be faster, more memory-efficient, and capable of handling both **raster and vector data**

sf is a modern and highly recommended package for working with **vector spatial data** (points, lines, polygons) in R.

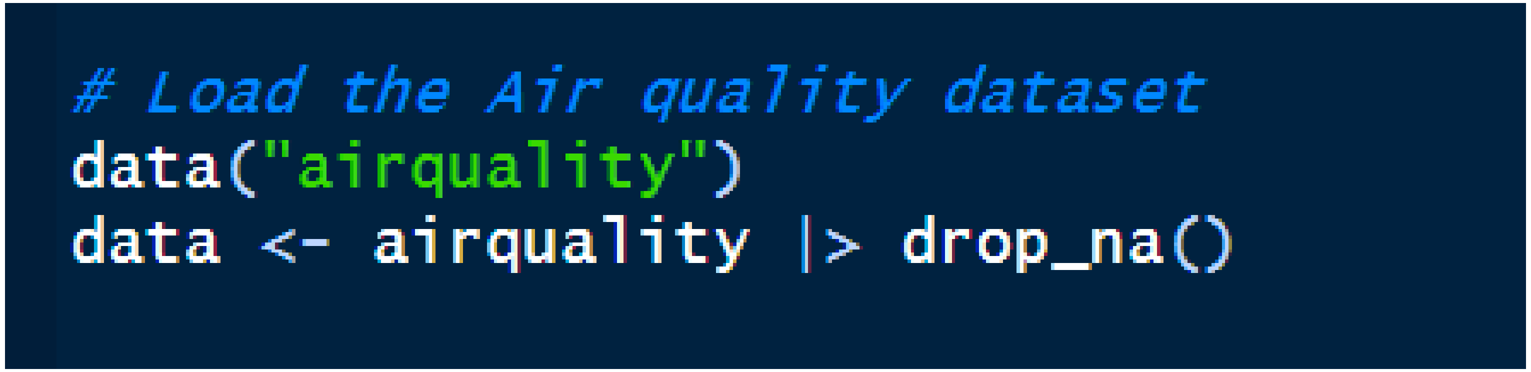
PART 2:

ML Workflow Live Coding Demo

Tidymodels Workflow Code Snippet

tidymodels provides a consistent, modular, and human-readable framework for building and evaluating models, making

1.Loading and preparing data



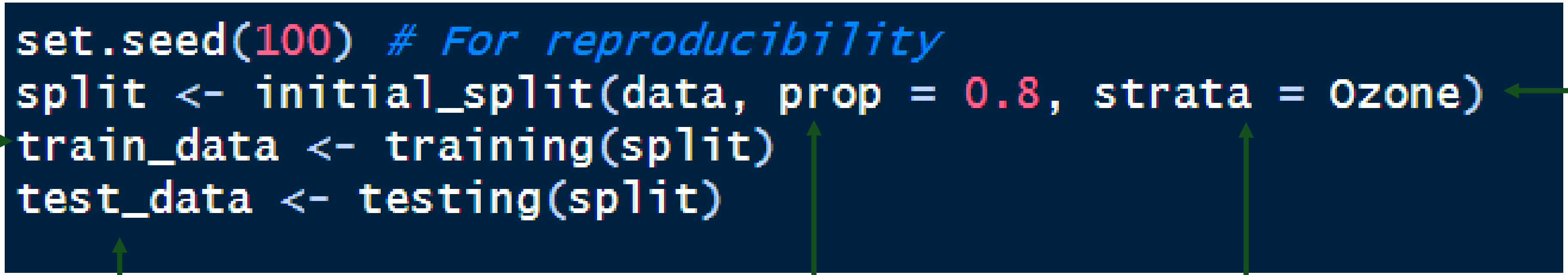
```r
# Load the Air quality dataset
data("airquality")
data <- airquality |> drop_na()
```

start by importing your data from various sources (csv, excel etc)

tidymodels integrate seamlessly with tidyverse for cleaning and manipulating data

2. Data splitting

Before any modelling, the data is split into at least two subsets: **training** – to teach the model, **testing** – to estimate model performance on unseen data



```
set.seed(100) # For reproducibility
split <- initial_split(data, prop = 0.8, strata = Ozone)
train_data <- training(split)
test_data <- testing(split)
```

Create testing dataset

Proportion to split for training

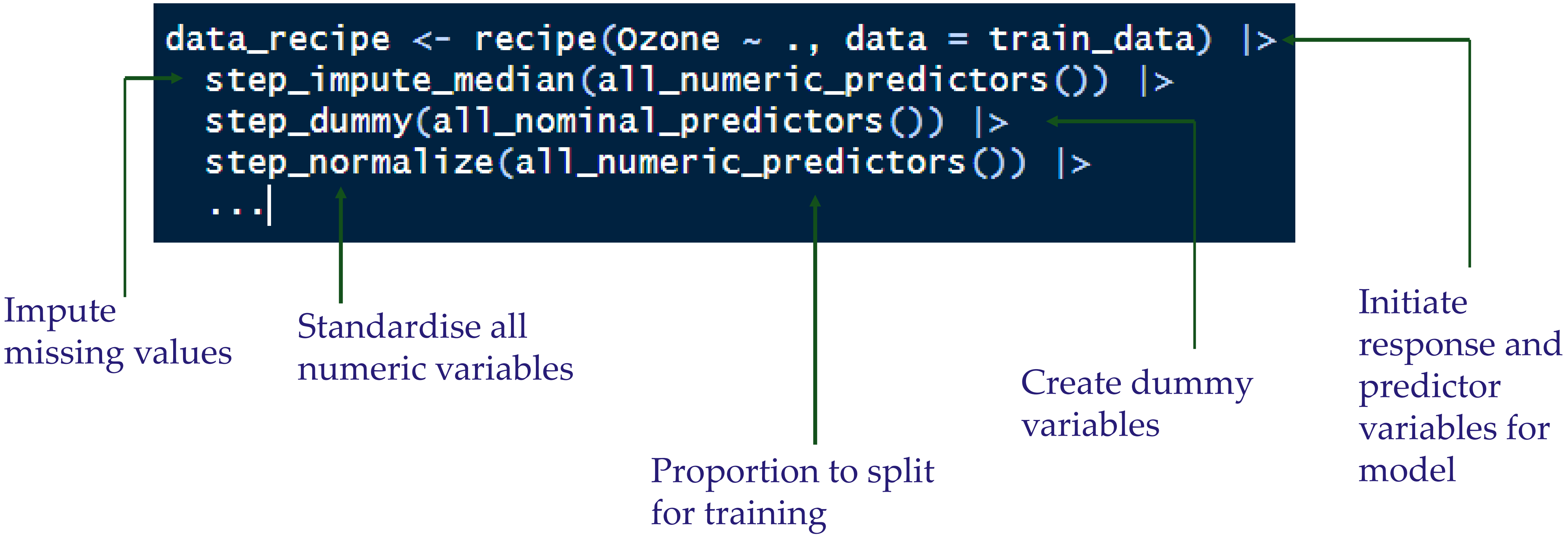
Variable to conduct stratified sampling

Create training dataset

Split data into training and testing set

3. Feature engineering and preprocessing (recipes)

This is crucial for preparing your data for modelling using the **recipes** package preprocessing steps



```
data_recipe <- recipe(Ozone ~ ., data = train_data) |>
  step_impute_median(all_numeric_predictors()) |>
  step_dummy(all_nominal_predictors()) |>
  step_normalize(all_numeric_predictors()) |>
  ...
```

Impute missing values

Standardise all numeric variables

Proportion to split for training

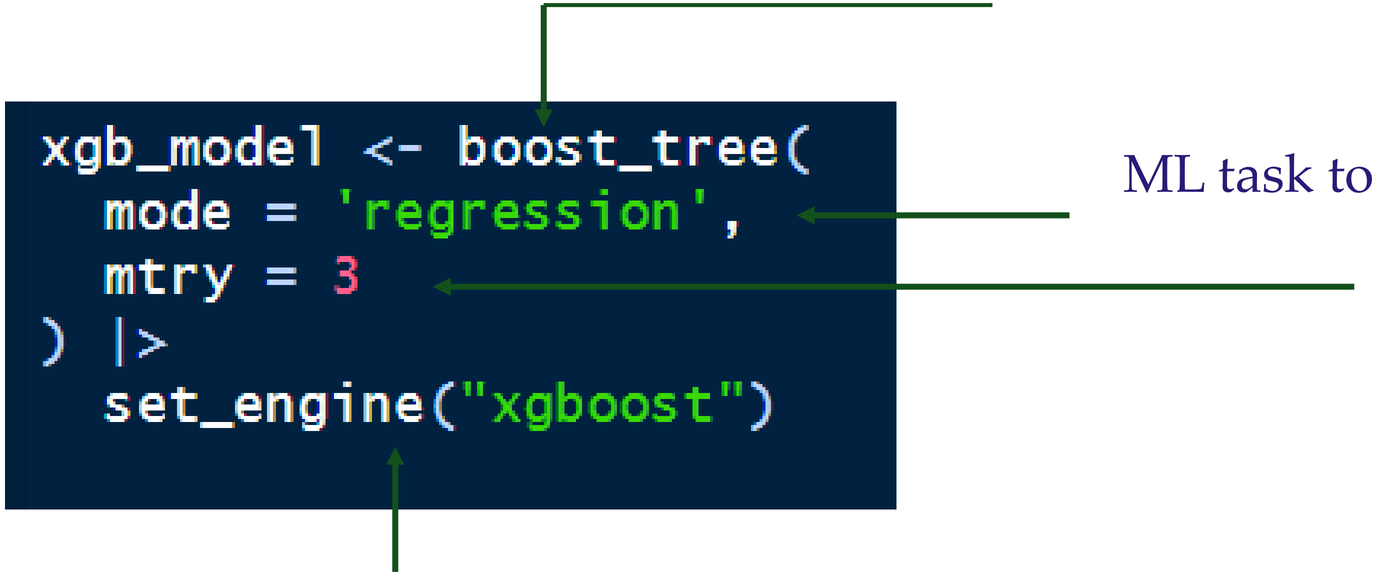
Create dummy variables

Initiate response and predictor variables for model

They define a sequence of preprocessing steps on the training data

4. Model Specification (parsnip)

Declare the type of model you want to use at this stage



```r
xgb_model <- boost_tree(
  mode = 'regression',
  mtry = 3
) |>
  set_engine("xgboost")
```

Specify ML model (extreme gradient boosting)

ML task to perform

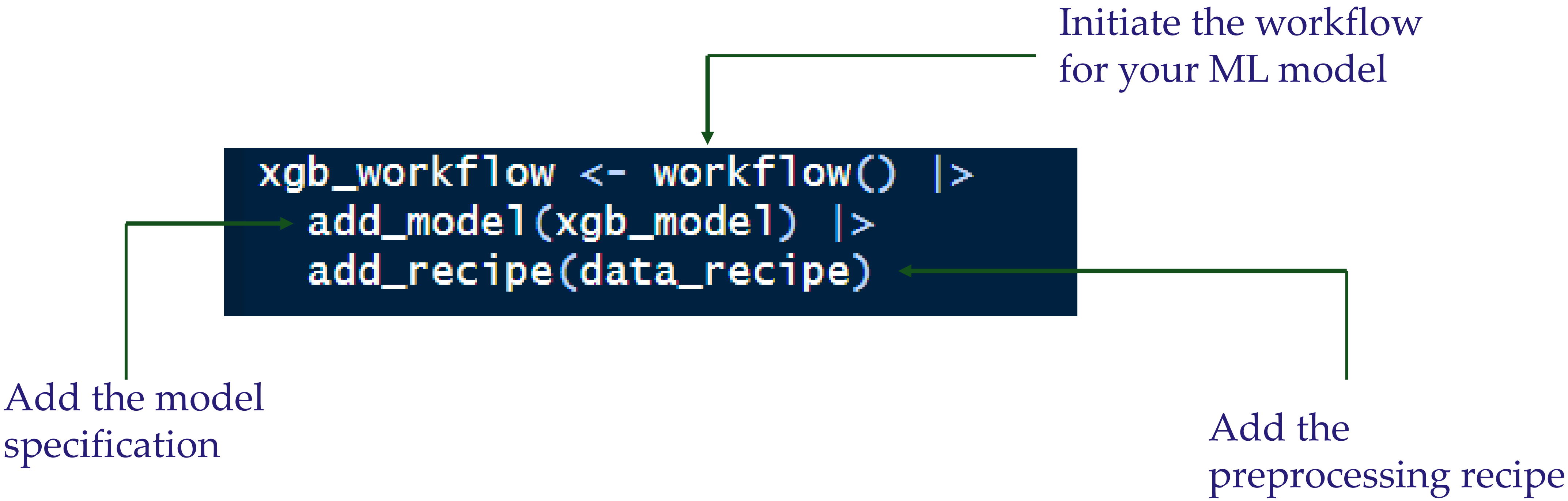
Predictors proportion to randomly sample at each split

Background engine (package) to use for this model

Several ML models exist within the **`parsnip`** package for use at this stage

5. Creating a workflow (workflows)

Bundel together your preprocessing recipe and your model specification to create a workflow



```
xgb_workflow <- workflow() |>
  add_model(xgb_model) |>
  add_recipe(data_recipe)
```

Initiate the workflow for your ML model

Add the model specification

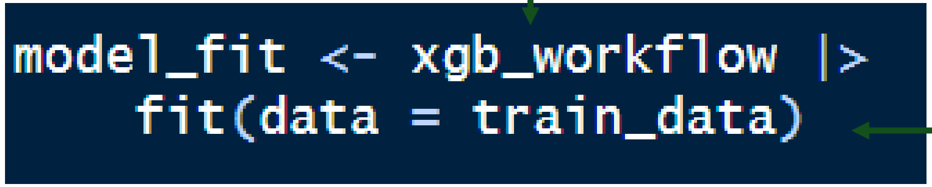
Add the preprocessing recipe

This ensures that the defined preprocessing steps are always applied before the model is fitted

6. Model training and fitting

With the workflow defined, you "fit" the workflow to your ***training data***

Defined workflow



```
model_fit <- xgb_workflow |>
    fit(data = train_data)
```

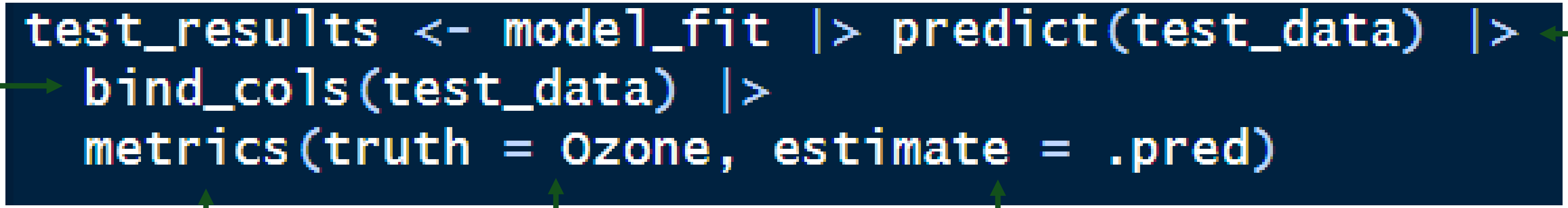
Fit model to the training data

The result is a fitted model object, ready for making predictions

Additionally, you can fine tune the model by varying various hyperparameter and performing cross-validations

7. Model training and fitting

Once the model is tuned it is fitted again and the final fit is used for making predictions on the test data



```
test_results <- model_fit |> predict(test_data) |>
  bind_cols(test_data) |>
  metrics(truth = Ozone, estimate = .pred)
```

Make predictions using the fitted model

Add the predicted data to the original dataset

Evaluate model performance

Outcome variable

Compare predicted to actual

Then, this final model makes predictions on the completely unseen **testing data**

7. Making Predictions

The ultimate goal!

The finalized, evaluated model can now be used to make predictions on truly new, never-before-seen data.

Live Code Demonstration

Goal:

Predict rainfall in Ghana using environmental data.

Steps:

1. Load synthetic environmental data (rainfall, temperature, humidity).

Scan this code to access dataset

2. Train a regression model using caret or tidymodels.

3. Visualize predictions with ggplot2.

RESOURCES

Nowosad, M. T. and J. (2025, June 17). *Elegant and informative maps with tmap*. https://tmap.geocompx.org/

Welcome | Geocomputation with R. (n.d.). Retrieved 17 June 2025, from https://r.geocompx.org/

Welcome! – Tidymodels. (n.d.). Retrieved 17 June 2025, from https://www.tidymodels.org/start/