Soybean Yield Forecasting Pipeline Documentation

Maryam Rehmatulla

December 2025

1 Introduction

The soybean forecasting module estimates weekly and annual yield using USDA crop condition reports, Earth Engine–derived Enhanced Difference Vegetation Index (EDVI), and a set of linear models linking these predictors to annual percent deviation from trend yield. The forecasting pipeline has three major components:

1. Cleaning and preparing annual, weekly, and remote-sensing data.
2. Computing trend yields and percent deviation models.
3. Feeding cleaned data into three forecast models:
    - Conditions-only forecast
    - EDVI-only forecast
    - Conditions + EDVI hybrid forecast

Each of these components is documented below.

2 Data Sources

2.1 Annual Data: `soy_annual`

The annual dataset contains one row per year (2014–2024) with the following columns:

- **Year** – integer year
- **Yield** – final USDA soybean yield (bu/acre)

- **Trend_Yield** – predicted yield based on a linear trend model
- **Percent_Deviation** – deviation from trend in percentage points
- **mean_EDVI** – annual mean EDVI value

These values are extracted from the Excel sheet and cleaned using:

```
soy_annual <- readxl::read_excel(data_file, sheet = "Annual_2014_2024") %>%
  mutate(
    Year = as.integer(Year),
    Yield = as.numeric(Yield),
    Trend_Yield = as.numeric(Trend_Yield),
    Percent_Deviation = as.numeric(Percent_Deviation),
    mean_EDVI = as.numeric(mean_EDVI)
  )
```

Explanation. This code reads the annual USDA data, converts all numeric fields to the correct types, and prepares the dataset for modeling. Each row corresponds to one crop year and includes the USDA-reported yield, the trend yield computed from a linear model, the percent deviation from that trend, and the annual mean EDVI. This dataset is the foundation for all three regression models.

2.2 Weekly Crop Conditions: `soy_weekly`

The weekly dataset contains weekly USDA crop condition values for each year. Key variables include:

- **Year**
- **Week** – calendar week (as a date)
- **Excellent, Good, Fair, Poor, VeryPoor**
- **GE** – Good + Excellent total

Loaded using:

```
soy_weekly <- readxl::read_excel(data_file, sheet = "Weekly_Raw") %>%
  mutate(
    Year = as.integer(Year),
    Week = as.Date(Week)
  )
```

Explanation. This code loads weekly USDA condition reports and ensures that the Year field is stored as an integer and the Week field is parsed as a calendar date. Each row represents one weekly observation and includes the percent of acres in each condition category (Excellent, Good, Fair, Poor, Very Poor). These weekly condition percentages feed directly into the conditions-only and hybrid forecasting models.

2.3 Weekly EDVI Time Series: `soy_edvi`

This dataset contains weekly EDVI band values (derived from MODIS and processed using Google Earth Engine):

- **Year**
- **Week**
- **mean_EDVI**

Loaded as:

```
soy_edvi <- readr::read_csv(
  "Soybeans_WeeklyBands_2013_2025_clean_EDVI.csv",
  show_col_types = FALSE
) %>% mutate(
    Year = as.integer(year),
    Week = as.Date(date)
)
```

Explanation. This code imports the weekly EDVI time series produced from Google Earth Engine. It converts the year and date fields into proper formats and keeps one EDVI value per week. These values are later used as predictors in both the EDVI-only and hybrid forecast models. Unlike crop conditions, EDVI reflects satellite-measured vegetation vigor, making it an independent biophysical signal of crop health.

3 Forecasting Models

3.1 Conditions-Only Model

$$\text{PercentDeviation} = \beta_0 + \beta_1(\text{Excellent}) + \beta_2(\text{Good}) + \beta_3(\text{Fair}) + \beta_4(\text{Poor})$$

Implemented:

```
reg_model_conditions <- lm(
  Percent_Deviation ~ Excellent + Good + Fair + Poor,
  data = soy_annual
)
```

Explanation. This regression model estimates how much each USDA crop condition category contributes to the annual percent deviation from trend yield. The coefficients represent the historical relationship between weekly crop conditions and final season yield. This model is later used to convert weekly condition percentages into predicted percent deviations for any selected year.

3.2 EDVI-Only Model

$$\text{PercentDeviation} = \beta_0 + \beta_1(\text{mean EDVI})$$

```
reg_model_edvi_only <- lm(
  Percent_Deviation ~ mean_EDVI,
  data = soy_annual
)
```

Explanation. The EDVI-only model uses annual mean EDVI as a single predictor of percent deviation from trend. EDVI is highly correlated with crop canopy health, so this model captures the relationship between satellite-derived vegetation vigor and yield outcomes. It provides a non-survey-based forecasting method.

3.3 Conditions + EDVI Hybrid Model

$$\text{PercentDeviation} = \beta_0 + \beta_1(E) + \beta_2(G) + \beta_3(F) + \beta_4(P) + \beta_5(\text{mean EDVI})$$

```
reg_model_cond_edvi <- lm(
  Percent_Deviation ~ Excellent + Good + Fair + Poor + mean_EDVI,
  data = soy_annual
)
```

Explanation. The hybrid model combines both USDA condition percentages and mean EDVI. This allows the model to use human-reported crop conditions and satellite-observed canopy health simultaneously, improving predictive accuracy. Historically, this model produces the lowest RMSE across years because it integrates both subjective and objective indicators.

4 Forecast Functions

Each forecast function returns a weekly dataset with predicted deviation, trend yield, and forecast yield.

4.1 Conditions-Only Forecast Function

```
make_forecasts_conditions <- function(year) {
  conds <- soy_weekly %>% filter(Year == year)
  if (nrow(conds) == 0) return(NULL)

  conds %>%
    mutate(
      Forecast_Dev = predict(reg_model_conditions, newdata = .),
      Trend_Yield = predict(
        lm(Yield ~ Year, data = soy_annual),
        newdata = data.frame(Year = year)
      ),
      Forecast_Yield = Trend_Yield * (1 + Forecast_Dev/100)
    ) %>%
    left_join(soy_annual %>% select(Year, Yield), by = "Year")
}
```

Explanation. This function filters weekly condition data for the selected year, generates a weekly predicted percent deviation using the conditions-only regression model, computes the trend yield for that same year, and converts the deviation into a predicted weekly yield. It also joins the true USDA-reported annual yield for comparison. The output powers the "Conditions-Only Forecast" plot in the dashboard.

4.2 EDVI-Only Forecast Function

```
make_forecasts_edvi <- function(year) {
  conds <- soy_edvi %>% filter(Year == year)
  if (nrow(conds) == 0) return(NULL)

  conds %>%
    mutate(
```

```
      Forecast_Dev = predict(reg_model_edvi_only,
                             newdata = data.frame(mean_EDVI = mean_EDVI)),
      Trend_Yield = predict(
        lm(Yield ~ Year, data = soy_annual),
        newdata = data.frame(Year = year)
      ),
      Forecast_Yield = Trend_Yield * (1 + Forecast_Dev/100)
    ) %>%
    left_join(soy_annual %>% select(Year, Yield), by = "Year")
}
```

Explanation. This function pulls weekly EDVI values for the selected year and uses the EDVI-only regression model to compute a predicted percent deviation for each week. It then multiplies the deviation by the annual trend yield to generate a weekly forecasted yield. This forecast does not use USDA condition reports at all—only satellite data.

4.3 Conditions + EDVI Hybrid Forecast Function

```
make_forecasts_cond_edvi <- function(year) {
  conds <- soy_weekly %>% filter(Year == year)
  if (nrow(conds) == 0) return(NULL)

  edvi_val <- soy_annual %>% filter(Year == year) %>% pull(mean_EDVI)
  newdata <- conds %>% mutate(mean_EDVI = edvi_val)

  newdata %>%
    mutate(
      Forecast_Dev = predict(reg_model_cond_edvi, newdata = newdata),
      Trend_Yield = predict(
        lm(Yield ~ Year, data = soy_annual),
        newdata = data.frame(Year = year)
      ),
      Forecast_Yield = Trend_Yield * (1 + Forecast_Dev/100)
    ) %>%
    left_join(soy_annual %>% select(Year, Yield), by = "Year")
}
```

Explanation. The hybrid function merges weekly crop conditions with the annual EDVI value for the selected year. It then passes these combined predictors into the hybrid regression model to compute weekly percent deviations and forecasted yield. This is the most complete forecasting method and typically produces the most accurate estimates.