In [1]:
```
import pandas as pd
import geopandas as gpd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import r2_score, mean_squared_error
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import learning_curve
from IPython.display import display
# Make sure TensorFlow warnings are minimized
tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)
```

```
WARNING:tensorflow:From C:\Users\kaibe\AppData\Local\Temp\ipykernel_20932\1812418026.py:21: The name tf.logging.set_verbosit
y is deprecated. Please use tf.compat.v1.logging.set_verbosity instead.
```

Code Overview

This script processes climate and food datasets, compares them, and generates plots to show correlations between climate metrics and food yields. The process is divided into four main functions:

1. **`process_climate_data`**: Processes the climate data to calculate yearly averages of relevant climate metrics such as temperature, humidity, and precipitation.
2. **`process_food_data`**: Processes the food data to calculate yearly total yields for different food categories like "Yield" and "Yield/Carcass Weight".
3. **`compare_climate_food`**: Combines the climate and food data by matching rows based on the "Year" column and computes the correlation between climate metrics and food yields.
4. **`plot_data`**: Generates time series plots of climate metrics and food yields, and a heatmap to visualize correlations between them.

Detailed Explanation of Each Function

`process_climate_data`

- This function processes the climate dataset to calculate yearly averages for various climate metrics.
- It:
    1. Loads the CSV file containing the climate data.
    2. Converts the 'DATE' column to a datetime format and extracts the year.
    3. Selects relevant climate metrics like average temperature, humidity, and precipitation.
    4. Computes yearly averages for these metrics and returns a DataFrame with the results.

`process_food_data`

- This function processes the food dataset to prepare it for comparison with the climate data.
- It:
    1. Loads the CSV file containing the food data.
    2. Filters the data for relevant elements such as "Yield" and "Yield/Carcass Weight".
    3. Reshapes the data so that each year is represented as a row, and the columns represent different food items.
    4. Adds a column for the total yield across all food items and returns the reshaped data.

`compare_climate_food`

- This function combines the processed climate and food datasets.
- It:

1. Merges the two datasets on the "Year" column.
2. Calculates the correlation matrix between climate metrics and food yields to see how they are related.

`plot_data`

- This function generates visualizations to help understand the relationship between climate and food yields.
- It:
  1. Plots the time series of climate metrics and food yields on a logarithmic scale.
  2. Creates a heatmap to show the correlation between climate metrics and food yields.

How the Code Works

1. **Data Processing**:
   - The script starts by processing the climate and food datasets to prepare them for analysis.
2. **Data Comparison**:
   - The processed climate and food data are merged based on the year and correlations are calculated.
3. **Data Visualization**:
   - The code generates time series plots and a heatmap to visualize the results.

By the end of the script, you will have a clear understanding of how climate metrics and food yields are correlated over time.

In [3]:
```
def process_climate_data(new_climate_file):
    """
    Processes the new climate dataset to calculate yearly averages of relevant metrics.

    Parameters:
        new_climate_file (str): Path to the new climate CSV file.

    Returns:
        pd.DataFrame: A DataFrame with yearly averages for relevant climate metrics.
    """
    # Load the dataset
    climate_data = pd.read_csv(new_climate_file)

    # Convert DATE to datetime and extract year
    climate_data['DATE'] = pd.to_datetime(climate_data['DATE'], errors='coerce')
    climate_data['Year'] = climate_data['DATE'].dt.year

    # Select relevant metrics and calculate yearly averages
    metrics = ['ADPT', 'DP10', 'EMNT', 'EMXP', 'EMXT', 'PRCP', 'RHAV', 'RHMN', 'RHMX']
    yearly_averages = (
        climate_data.groupby('Year')[metrics]
        .mean()
        .dropna(how='all')  # Drop years with no data
        .reset_index()
    )

    return yearly_averages
```

In [4]:
```
def process_food_data(food_file):
    """
    Processes the food dataset to prepare it for comparison with climate data.

    Parameters:
        food_file (str): Path to the food CSV file.

    Returns:
        pd.DataFrame: A DataFrame with yearly total yield and categories.
    """
  # Load the data
    data = pd.read_csv(food_file)

    # Filter rows where 'Element' is either 'Yield' or 'Yield/Carcass Weight'
    yield_data = data[data['Element'].isin(['Yield', 'Yield/Carcass Weight'])]

    # Reshape the data: Years as rows, Items as columns
    yield_pivot = yield_data.pivot_table(
        index='Year',
```

```
        columns='Item',
        values='Value',
        aggfunc='sum'
    )
    yield_pivot['Total Yield'] = yield_pivot.sum(axis=1)

    # Reset the index for a clean DataFrame
    yield_pivot.reset_index(inplace=True)

    return yield_pivot
```

In [5]:

```
def compare_climate_food(climate_data, food_data):
    """
    Combines processed climate and food data for comparison and computes correlations.

    Parameters:
        climate_data (pd.DataFrame): Processed climate data with yearly averages.
        food_data (pd.DataFrame): Processed food data with yearly totals and categories.

    Returns:
        pd.DataFrame: A DataFrame showing correlations between climate metrics and food yields.
    """
    # Merge datasets on the Year column
    combined_data = pd.merge(climate_data, food_data, on='Year', how='inner')

    # Calculate correlations
    correlations = combined_data.corr()

    return combined_data, correlations

def plot_data(combined_data, correlations):
    """
    Generates and displays plots for the combined data and correlation heatmap.

    Parameters:
        combined_data (pd.DataFrame): The merged dataset of climate and food data.
        correlations (pd.DataFrame): Correlation matrix of the combined dataset.
    """
    # Set up the plotting environment
    sns.set(style="whitegrid")

    # Plot all time series with logarithmic scale
    ax = combined_data.plot(x='Year', figsize=(15, 10), title='Time Series of Climate Metrics and Food Yields (Logarithmic
    ax.set_yscale('log')
    plt.xlabel('Year')
    plt.ylabel('Logarithmic Values')
    plt.show()

    # Plot correlation heatmap
    plt.figure(figsize=(12, 8))
    sns.heatmap(correlations, annot=True, cmap="coolwarm", fmt='.2f', linewidths=0.5)
    plt.title('Correlation Matrix of Climate Metrics and Food Yields')
    plt.show()


# File paths
climate_file_path = "./3863529.csv"  # Path to the new climate dataset
food_file_path = "./food_data_90-22.csv"  # Path to the food dataset

    # Process datasets
processed_climate_data = process_climate_data(climate_file_path)
food_df = process_food_data(food_file_path)

    # Compare datasets
combined_data, correlations = compare_climate_food(processed_climate_data, food_df)

    # Display results
print("Combined Data:")
print(combined_data.head())

print("\nCorrelations:")
print(correlations)
```

```
    # Generate and display plots
plot_data(combined_data, correlations)
```

Combined Data:

	Year	ADPT	DP10	EMNT	EMXP	EMXT	PRCP	RHAV	RHMN
0	1990	NaN	4.549020	25.828431	1.050098	81.887255	2.583922	NaN	NaN
1	1991	NaN	4.700980	25.759804	0.928578	79.504902	2.631863	NaN	NaN
2	1992	NaN	4.750000	25.784314	0.895098	78.367647	2.494706	NaN	NaN
3	1993	NaN	4.857143	23.553922	0.959507	77.181373	2.607192	NaN	NaN
4	1994	NaN	4.558824	25.852941	0.991765	79.495098	2.609167	NaN	NaN

	RHMX	Beef and Buffalo Meat, primary	Cereals, primary	Eggs Primary
0	NaN	2967.0	4755.1	148886.0
1	NaN	3002.0	4507.6	149355.0
2	NaN	3014.0	5357.8	150274.0
3	NaN	3014.0	4298.8	149594.0
4	NaN	2929.0	5560.0	150394.0

	Fibre Crops, Fibre Equivalent	Fruit Primary	Meat, Poultry	Milk, Total
0	647.9	19986.0	17038.0	67052.0
1	675.2	20973.8	17140.0	67359.0
2	712.6	22547.4	17406.0	69744.0
3	626.9	23857.9	17524.0	70376.0
4	725.7	23595.2	17715.0	72373.0

	Treenuts, Total	Vegetables Primary	Total Yield
0	2694.5	24037.5	288064.0
1	2441.3	24316.0	289769.9
2	2437.8	23906.2	295399.8
3	2545.6	24279.9	296117.1
4	2717.0	25880.8	301889.7

Correlations:

	Year	ADPT	DP10	EMNT
Year	1.000000	0.374862	-0.168019	0.441299
ADPT	0.374862	1.000000	0.410693	0.661568
DP10	-0.168019	0.410693	1.000000	-0.142836
EMNT	0.441299	0.661568	-0.142836	1.000000
EMXP	0.237023	0.375871	0.599129	0.115337
EMXT	0.473728	0.388888	-0.567807	0.613179
PRCP	0.117983	0.347770	0.799183	0.002887
RHAV	-0.044966	0.229274	0.830929	-0.222923
RHMN	0.072312	0.185123	0.779836	-0.283735
RHMX	-0.160710	0.189926	0.755035	-0.180317
Beef and Buffalo Meat, primary	0.947511	0.433678	-0.113034	0.451580
Cereals, primary	0.945357	0.443709	-0.033370	0.481890
Eggs Primary	0.971144	0.306772	-0.153456	0.410585
Fibre Crops, Fibre Equivalent	0.901330	0.389121	-0.178191	0.424859
Fruit Primary	-0.065204	-0.352982	0.055591	-0.109815
Meat, Poultry	0.971378	0.575050	-0.079059	0.450787
Milk, Total	0.998788	0.418511	-0.169778	0.458513
Treenuts, Total	0.622970	-0.527617	-0.321389	0.322484
Vegetables Primary	0.957968	0.620326	-0.128515	0.420737
Total Yield	0.995625	0.418968	-0.151277	0.444996

	EMXP	EMXT	PRCP	RHAV
Year	0.237023	0.473728	0.117983	-0.044966
ADPT	0.375871	0.388888	0.347770	0.229274
DP10	0.599129	-0.567807	0.799183	0.830929
EMNT	0.115337	0.613179	0.002887	-0.222923
EMXP	1.000000	-0.075376	0.896170	0.741427
EMXT	-0.075376	1.000000	-0.322529	-0.738021
PRCP	0.896170	-0.322529	1.000000	0.856741
RHAV	0.741427	-0.738021	0.856741	1.000000
RHMN	0.761985	-0.744084	0.860686	0.962158
RHMX	0.596126	-0.672347	0.704351	0.943945
Beef and Buffalo Meat, primary	0.231952	0.433099	0.119222	-0.010356
Cereals, primary	0.371128	0.388493	0.245088	0.312437
Eggs Primary	0.295008	0.482256	0.168865	-0.121764
Fibre Crops, Fibre Equivalent	0.254294	0.497889	0.114203	-0.282800
Fruit Primary	-0.194206	-0.342371	-0.048851	0.128544
Meat, Poultry	0.292277	0.441305	0.196008	0.302138
Milk, Total	0.239270	0.487660	0.115479	-0.059830
Treenuts, Total	0.022266	0.294115	-0.063468	-0.188714
Vegetables Primary	0.251351	0.468055	0.150183	0.026454
Total Yield	0.257888	0.467048	0.143373	-0.026945

	RHMN	RHMX
Year	0.072312	-0.160710
ADPT	0.185123	0.189926
DP10	0.779836	0.755035
EMNT	-0.283735	-0.180317
EMXP	0.761985	0.596126
EMXT	-0.744084	-0.672347
PRCP	0.860686	0.704351
RHAV	0.962158	0.943945
RHMN	1.000000	0.829659
RHMX	0.829659	1.000000
Beef and Buffalo Meat, primary	0.137025	-0.211085
Cereals, primary	0.413845	0.164058
Eggs Primary	-0.000550	-0.234123
Fibre Crops, Fibre Equivalent	-0.179410	-0.358010
Fruit Primary	0.029300	0.174983
Meat, Poultry	0.350696	0.202509
Milk, Total	0.058877	-0.180396
Treenuts, Total	-0.202079	-0.162648
Vegetables Primary	0.139584	-0.164429
Total Yield	0.092284	-0.164996

	Beef and Buffalo Meat, primary
Year	0.947511
ADPT	0.433678
DP10	-0.113034
EMNT	0.451580
EMXP	0.231952
EMXT	0.433099
PRCP	0.119222
RHAV	-0.010356
RHMN	0.137025
RHMX	-0.211085
Beef and Buffalo Meat, primary	1.000000
Cereals, primary	0.927031
Eggs Primary	0.906052
Fibre Crops, Fibre Equivalent	0.879378
Fruit Primary	-0.030671
Meat, Poultry	0.934003
Milk, Total	0.947141
Treenuts, Total	0.623508
Vegetables Primary	0.946790
Total Yield	0.951700

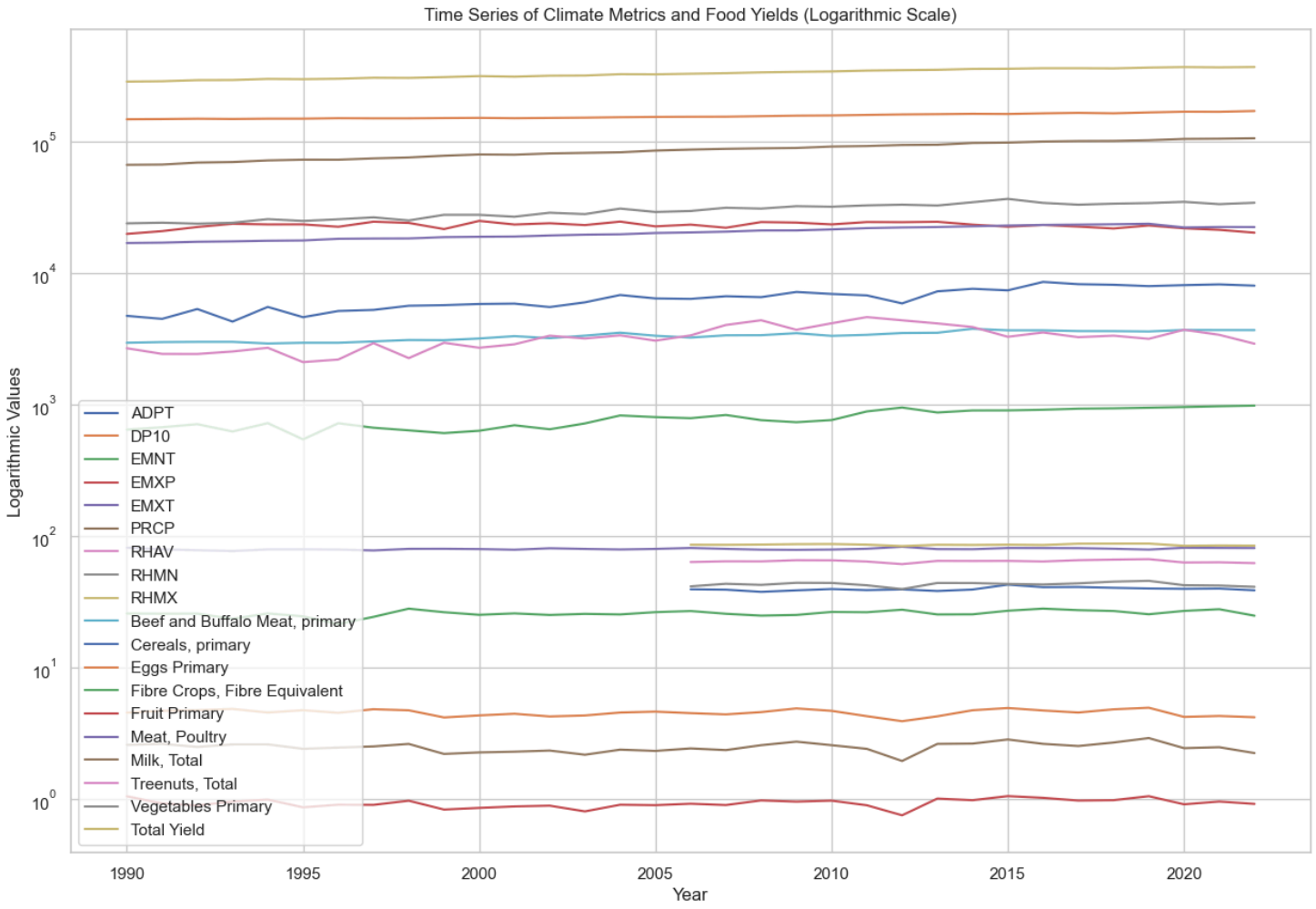
	Cereals, primary	Eggs Primary
Year	0.945357	0.971144
ADPT	0.443709	0.306772
DP10	-0.033370	-0.153456
EMNT	0.481890	0.410585
EMXP	0.371128	0.295008
EMXT	0.388493	0.482256
PRCP	0.245088	0.168865
RHAV	0.312437	-0.121764
RHMN	0.413845	-0.000550
RHMX	0.164058	-0.234123
Beef and Buffalo Meat, primary	0.927031	0.906052
Cereals, primary	1.000000	0.918662
Eggs Primary	0.918662	1.000000
Fibre Crops, Fibre Equivalent	0.865647	0.917512
Fruit Primary	-0.094044	-0.189458
Meat, Poultry	0.928010	0.928077
Milk, Total	0.947169	0.971226
Treenuts, Total	0.547852	0.523526
Vegetables Primary	0.911186	0.908511
Total Yield	0.946983	0.969911

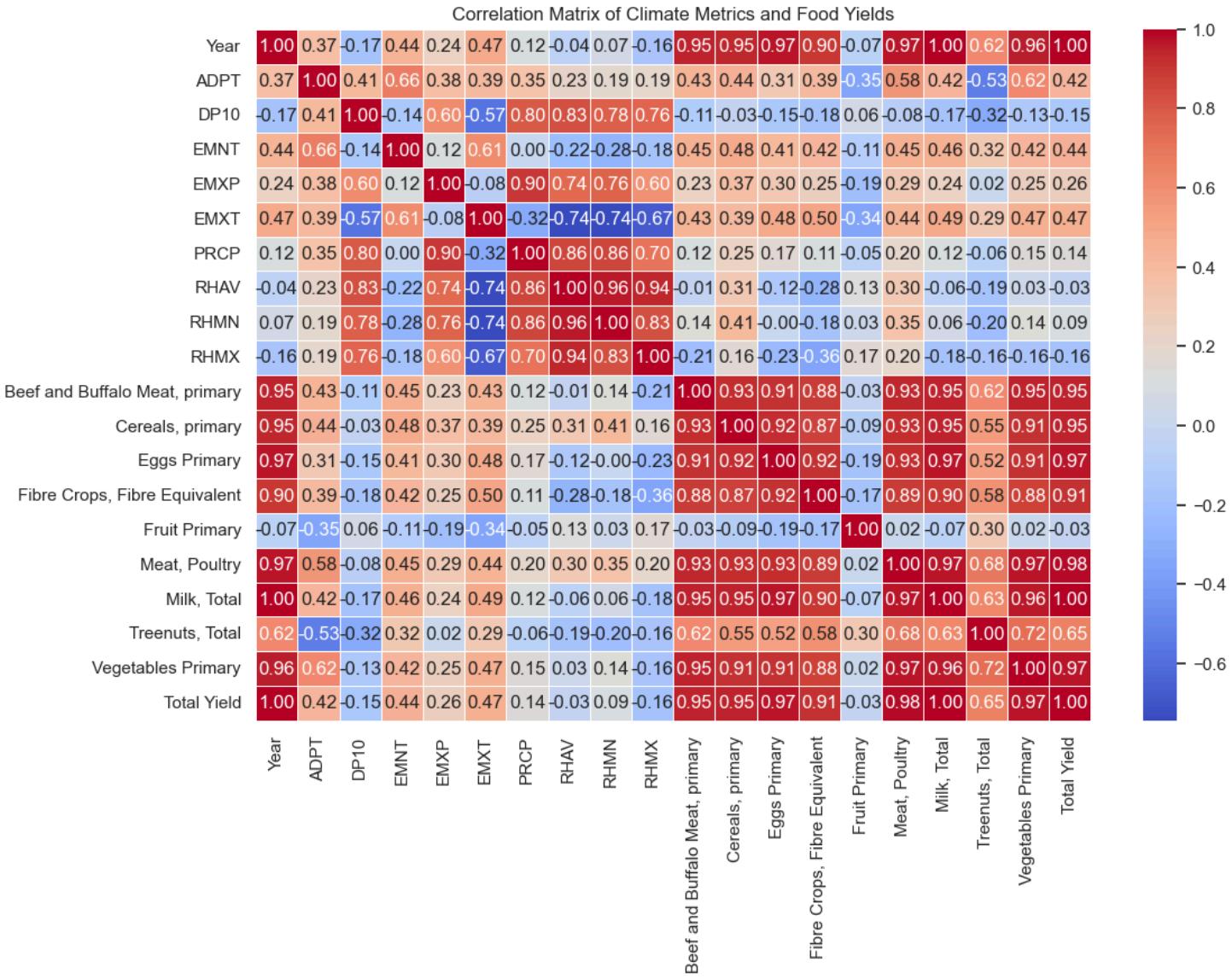
	Fibre Crops, Fibre Equivalent	Fruit Primary
Year	0.901330	-0.065204
ADPT	0.389121	-0.352982
DP10	-0.178191	0.055591
EMNT	0.424859	-0.109815
EMXP	0.254294	-0.194206
EMXT	0.497889	-0.342371
PRCP	0.114203	-0.048851

RHAV	-0.282800	0.128544
RHMN	-0.179410	0.029300
RHMX	-0.358010	0.174983
Beef and Buffalo Meat, primary	0.879378	-0.030671
Cereals, primary	0.865647	-0.094044
Eggs Primary	0.917512	-0.189458
Fibre Crops, Fibre Equivalent	1.000000	-0.172169
Fruit Primary	-0.172169	1.000000
Meat, Poultry	0.889803	0.021827
Milk, Total	0.900467	-0.068674
Treenuts, Total	0.584191	0.304622
Vegetables Primary	0.876992	0.020297
Total Yield	0.908491	-0.025830

	Meat, Poultry	Milk, Total	Treenuts, Total	\
Year	0.971378	0.998788	0.622970	
ADPT	0.575050	0.418511	-0.527617	
DP10	-0.079059	-0.169778	-0.321389	
EMNT	0.450787	0.458513	0.322484	
EMXP	0.292277	0.239270	0.022266	
EMXT	0.441305	0.487660	0.294115	
PRCP	0.196008	0.115479	-0.063468	
RHAV	0.302138	-0.059830	-0.188714	
RHMN	0.350696	0.058877	-0.202079	
RHMX	0.202509	-0.180396	-0.162648	
Beef and Buffalo Meat, primary	0.934003	0.947141	0.623508	
Cereals, primary	0.928010	0.947169	0.547852	
Eggs Primary	0.928077	0.971226	0.523526	
Fibre Crops, Fibre Equivalent	0.889803	0.900467	0.584191	
Fruit Primary	0.021827	-0.068674	0.304622	
Meat, Poultry	1.000000	0.972722	0.682262	
Milk, Total	0.972722	1.000000	0.625955	
Treenuts, Total	0.682262	0.625955	1.000000	
Vegetables Primary	0.966882	0.960478	0.718177	
Total Yield	0.980786	0.996653	0.654856	

	Vegetables Primary	Total Yield
Year	0.957968	0.995625
ADPT	0.620326	0.418968
DP10	-0.128515	-0.151277
EMNT	0.420737	0.444996
EMXP	0.251351	0.257888
EMXT	0.468055	0.467048
PRCP	0.150183	0.143373
RHAV	0.026454	-0.026945
RHMN	0.139584	0.092284
RHMX	-0.164429	-0.164996
Beef and Buffalo Meat, primary	0.946790	0.951700
Cereals, primary	0.911186	0.946983
Eggs Primary	0.908511	0.969911
Fibre Crops, Fibre Equivalent	0.876992	0.908491
Fruit Primary	0.020297	-0.025830
Meat, Poultry	0.966882	0.980786
Milk, Total	0.960478	0.996653
Treenuts, Total	0.718177	0.654856
Vegetables Primary	1.000000	0.972307
Total Yield	0.972307	1.000000


Time Series of Climate Metrics and Food Yields (Logarithmic Scale)
Logarithmic Values
Year
1990
1995
2000
2005
2010
2015
2020
10^5
10^4
10^3
10^2
10^1
10^0
ADPT
DP10
EMNT
EMXP
EMXT
PRCP
RHAV
RHMN
RHMX
Beef and Buffalo Meat, primary
Cereals, primary
Eggs Primary
Fibre Crops, Fibre Equivalent
Fruit Primary
Meat, Poultry
Milk, Total
Treenuts, Total
Vegetables Primary
Total Yield



Climate Station Geographic Distribution Visualization

Overview

This visualization system creates a geographical representation of climate monitoring stations across the United States, providing crucial insight into data collection coverage and spatial distribution patterns.

Visualization Components

1. Base Map Implementation

- Uses Natural Earth dataset for accurate US boundaries
- Implements clean, visually appealing design
- Maintains geographical proportions
- Provides clear country outline

2. Station Location Plotting

- Represents stations as red circles
- Uses semi-transparent markers for density visualization
- Includes black edges for visibility
- Maintains consistent marker size

3. Map Design Elements

- White background for clarity
- Removed grid lines for clean appearance
- Clear axis labels
- Informative title

Technical Implementation

Data Processing

- Extracts latitude and longitude coordinates
- Filters for US territory
- Maintains data precision
- Handles coordinate systems

Visual Design

- Appropriate figure size
- Color scheme selection
- Transparency settings
- Edge formatting

Analysis Benefits

- Shows monitoring network coverage
- Identifies potential gaps
- Reveals spatial patterns
- Supports network planning

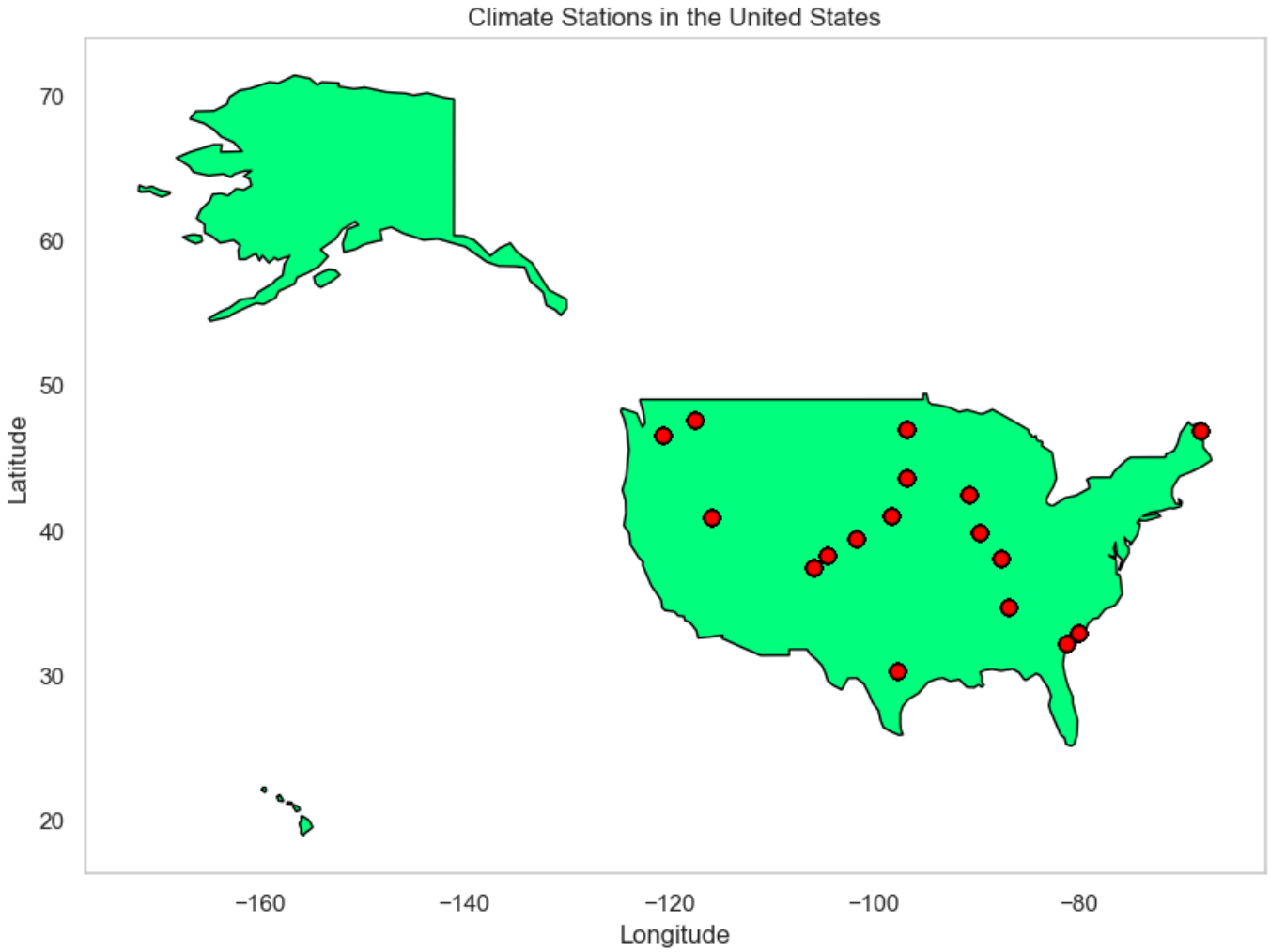
Geographical Insights

- Station density variations
- Regional coverage assessment
- Network distribution patterns
- Potential expansion areas

In [7]:
```
"""
Visualize the locations of climate monitoring stations across the United States
using a map plot with geographical coordinates.

This script creates a map visualization showing:
- The outline of the United States
- The locations of climate monitoring stations as points
- Clear geographical reference with latitude and longitude axes
"""

# Load climate station data from NOAA database
# File contains station metadata including geographical coordinates
climate_df = pd.read_csv(r'C:\Users\kaibe\Downloads\3863529.csv')

# Extract geographical coordinates from the dataset
# These will be used to plot station locations
latitudes = climate_df['LATITUDE']    # Latitude in decimal degrees
longitudes = climate_df['LONGITUDE']  # Longitude in decimal degrees

# Load geographical data for creating the base map
# Using Natural Earth dataset for country boundaries
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

# Filter to show only the United States
```

```python
# Creates a GeoDataFrame containing only US boundaries
us_map = world[world.name == 'United States of America']

# Create a new figure with specified size and white background
# Size of 10x10 inches provides good detail while maintaining proportions
fig, ax = plt.subplots(figsize=(10, 10), facecolor='white')
ax.set_facecolor('white')  # Set plot background to white

# Remove default grid lines for cleaner appearance
ax.grid(False)

# Plot the US map outline
# - Color: springgreen for land mass
# - Black edges for country boundaries
us_map.plot(ax=ax, color='springgreen', edgecolor='black')

# Add climate station locations as scatter points
# - Red circles represent station locations
# - Black edges for better visibility
# - Alpha of 0.7 for slight transparency
ax.scatter(longitudes, latitudes,
          color='red',          # Station markers in red
          s=50,                 # Marker size
          edgecolor='black',    # Black outline
          alpha=0.7)            # Partial transparency

# Add labels and title to the plot
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_title('Climate Stations in the United States')

# Remove background grid lines for cleaner appearance
ax.set_axisbelow(False)

# Display the final map
plt.show()
```

```
C:\Users\kaibe\AppData\Local\Temp\ipykernel_20932\4041443989.py:22: FutureWarning: The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
  world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
```



In [8]:
```
# Load and preprocess extended climate dataset with additional features
# File contains detailed climate measurements from weather station data
extended_climate_df = pd.read_csv('3863474.csv')

# Convert DATE column to datetime format and extract year
# This allows for yearly aggregation of climate data
extended_climate_df['DATE'] = pd.to_datetime(extended_climate_df['DATE'])
extended_climate_df['Year'] = extended_climate_df['DATE'].dt.year

# Define climate features to analyze
climate_features = [
    'ADPT',  # Average dew point temperature
    'DP10',  # Number of days with precipitation ≥ 1.0 inch
    'EMNT',  # Extreme minimum temperature
    'EMXP',  # Extreme maximum precipitation
    'EMXT',  # Extreme maximum temperature
    'EVAP',  # Evaporation
    'MXPN',  # Maximum daily precipitation
    'PRCP',  # Precipitation
    'RHAV',  # Average relative humidity
    'RHMN',  # Minimum relative humidity
    'RHMX'   # Maximum relative humidity
]

# Remove rows with missing values in any climate feature
# Ensures data quality for statistical analysis
extended_climate_df = extended_climate_df.dropna(subset=climate_features)

# Calculate yearly averages for all climate features
# Groups data by year and computes mean for each feature
yearly_climate = extended_climate_df.groupby('Year')[climate_features].mean().reset_index()

# Standardize climate features using z-score normalization
# Transforms features to have mean=0 and standard deviation=1
scaler = StandardScaler()
```

```
yearly_climate[climate_features] = scaler.fit_transform(yearly_climate[climate_features])

# Merge standardized climate data with food production data
# Combines datasets on matching years for correlation analysis
merged_df = pd.merge(yearly_climate, food_df, on='Year')

# Display the merged dataset
merged_df
```

Out[8]:

	Year	ADPT	DP10	EMNT	EMXP	EMXT	EVAP	MXPN	PRCP	RHAV	...	Beef and Buffalo Meat, primary	Cereals, primary	Eg Prima
0	2006	-0.401579	-0.456077	-0.320384	-0.093782	-0.357799	0.093079	0.504544	-0.391480	-0.148397	...	3250.0	6400.4	15544
1	2007	-1.110850	-1.731657	-0.553528	-0.988836	1.224916	1.507218	1.356972	-1.254950	-1.445558	...	3383.0	6704.4	15570
2	2008	-1.032278	-1.353664	0.665611	-0.569851	1.099956	1.765575	1.404021	-0.935017	-1.370404	...	3389.0	6598.3	15720
3	2009	-2.073572	-0.542516	0.066759	-0.399924	1.568131	2.150480	0.644100	-0.417990	-1.828519	...	3510.0	7226.2	15859
4	2010	-0.923068	0.586037	-0.964925	0.676283	0.858014	0.351668	-1.083332	0.470954	0.275070	...	3350.0	6978.1	15920
5	2011	-0.203565	1.361918	-0.312711	1.808523	-2.045299	-1.612469	-1.817306	2.466226	1.766281	...	3411.0	6803.5	16074
6	2012	0.602664	-0.351485	-0.051825	-0.316939	-0.662769	-0.941421	-0.532852	-0.106360	0.596452	...	3519.0	5911.9	16207
7	2013	0.185095	1.588216	-0.540986	2.538784	-0.317137	-0.888048	-0.940615	2.167322	1.367768	...	3535.0	7300.9	16283
8	2014	1.882137	-0.405366	2.557032	0.260905	0.489339	-0.243525	1.064219	0.091491	0.703579	...	3791.0	7638.1	16380
9	2015	-1.218578	0.586037	-0.834482	-0.643547	-1.077528	-0.719679	-1.491095	-0.050292	0.172228	...	3689.0	7430.1	16334
10	2016	0.699026	1.103291	0.274282	-0.501180	-0.403545	-0.242716	-0.940615	0.550997	0.917834	...	3690.0	8614.2	16521
11	2017	-0.055810	-0.513127	-1.682361	-0.417434	-0.057912	-0.613903	-0.634793	-0.460173	0.275070	...	3639.0	8281.3	16661
12	2018	0.811448	-1.482978	1.035199	0.428396	0.258917	0.523110	0.928298	-0.488530	-0.667649	...	3636.0	8196.0	16529
13	2019	0.104793	0.241201	1.470008	-0.476056	-0.893191	0.264330	0.248693	-0.823653	-0.410544	...	3614.0	8005.5	16787
14	2020	0.490241	-0.620888	-1.138849	-1.927646	1.641447	0.115532	1.064219	-1.277358	-1.267562	...	3713.0	8144.5	16996
15	2021	1.089827	0.672246	-0.051825	0.428396	-1.008402	-0.942230	-0.159070	0.106958	0.617877	...	3706.0	8252.2	16970
16	2022	1.154069	1.318813	0.382985	0.193908	-0.317137	-0.566999	0.384614	0.351856	0.446474	...	3703.0	8071.7	17211

17 rows × 22 columns

Climate Model Training Implementation

Overview

This implementation creates a comprehensive machine learning pipeline for predicting food production based on climate variables. The analysis reveals varying performance across different models and food categories.

Model Performance Results

Support Vector Regression (SVR)

- Average R^2 Score: 0.2955
- Best Predictions:
  1. Fibre Crops (R^2 = 0.7023)
  2. Cereals (R^2 = 0.6979)
  3. Eggs (R^2 = 0.6433)
- Shows strong performance in crop yield prediction

Random Forest

- Average R^2 Score: 0.3247
- Best Predictions:
  1. Milk Production (R^2 = 0.7915)
  2. Cereals (R^2 = 0.6821)
  3. Eggs (R^2 = 0.6701)
- Most consistent performer across categories

XGBoost

- Average R^2 Score: 0.2901
- Best Predictions:
  1. Milk Production (R^2 = 0.8730)
  2. Cereals (R^2 = 0.8554)
  3. Beef/Buffalo Meat (R^2 = 0.8206)
- Highest individual category performance

Key Insights

- Milk production and cereals show consistently strong predictability
- XGBoost achieves highest individual R^2 scores
- Random Forest has best average performance
- Some categories (e.g., vegetables) show poor predictability across all models

Class Structure

The `ClimateModelTrainer` class implements three distinct regression algorithms:

1. Support Vector Regression (SVR)
2. Random Forest Regression
3. XGBoost Regression

Data Handling

- **Input Features**: Climate variables including temperature metrics, precipitation, and humidity
- **Target Variables**: Various food production categories
- **Data Preprocessing**:
  - Standardization of features
  - Train-test splitting
  - Missing value handling

Model Training Approaches

Support Vector Regression

- Implements adaptive complexity through gradual C parameter adjustment
- Tracks performance metrics across training epochs
- Uses RBF kernel for non-linear relationships

Random Forest

- Implements incremental tree addition
- Starts with minimal trees and gradually increases forest size
- Maintains performance history across epochs

XGBoost

- Implements gradient boosting with controlled learning rate
- Tracks performance at each boosting round
- Includes built-in regularization

Performance Metrics

Each model tracks two key metrics:

1. R^2 Score: Measures goodness of fit
2. Mean Squared Error: Measures prediction accuracy

Implementation Details

The class provides:

- Separate training methods for each algorithm
- Consistent interface across models
- Comprehensive performance tracking
- Easy access to trained models and results

In [10]:
```python
class ClimateModelTrainer:
    """
    A class for training and evaluating multiple machine learning models
    to predict food production based on climate features.

    The class implements three different regression models:
    - Support Vector Regression (SVR)
    - Random Forest Regression
    - XGBoost Regression

    Attributes:
        climate_features (list): Climate-related input features
        target_columns (list): Food production categories to predict
        X (DataFrame): Input features dataset
        y (DataFrame): Target variables dataset
        X_train, X_test, y_train, y_test: Train-test split datasets
        scaler: StandardScaler for feature normalization
    """

    def __init__(self, data):
        """
        Initialize the ClimateModelTrainer with input data.

        Args:
            data (DataFrame): Combined climate and food production data
        """
        # Define input features (climate variables + total yield)
        self.climate_features = [
            'ADPT',   # Average dew point temperature
            'DP10',   # Days with precipitation ≥ 1.0 inch
            'EMNT',   # Extreme minimum temperature
            'EMXP',   # Extreme maximum precipitation
            'EMXT',   # Extreme maximum temperature
            'EVAP',   # Evaporation
            'MXPN',   # Maximum daily precipitation
            'PRCP',   # Precipitation
            'RHAV',   # Average relative humidity
            'RHMN',   # Minimum relative humidity
            'RHMX',   # Maximum relative humidity
            'Total Yield'
        ]

        # Define target variables (food production categories)
        self.target_columns = [
            "Beef and Buffalo Meat, primary",
```

```
            "Cereals, primary",
            "Eggs Primary",
            "Fibre Crops, Fibre Equivalent",
            "Fruit Primary",
            "Meat, Poultry",
            "Milk, Total",
            "Treenuts, Total",
            "Vegetables Primary"
        ]

        # Prepare input features and target variables
        self.X = data[self.climate_features].fillna(method='ffill')
        self.y = data[self.target_columns]

        # Split data into training and testing sets
        self.X_train, self.X_test, self.y_train, self.y_test = train_test_split(
            self.X, self.y, test_size=0.2, random_state=42
        )

        # Scale features using StandardScaler
        self.scaler = StandardScaler()
        self.X_train_scaled = self.scaler.fit_transform(self.X_train)
        self.X_test_scaled = self.scaler.transform(self.X_test)

    def train_svr(self, epochs=50):
        """
        Train Support Vector Regression models for each target variable.

        Args:
            epochs (int): Number of training epochs

        Returns:
            dict: Contains training results and trained models
        """
        print("\n## Training SVR")
        svr_results = {}
        svr_models = {}

        for target in self.target_columns:
            print(f"\nTraining SVR for {target}")
            epoch_scores = []
            # Initialize SVR with conservative complexity
            model = SVR(kernel='rbf', C=0.1)

            # Gradually increase model complexity
            for epoch in range(epochs):
                model.C = 0.1 * (epoch + 1)  # Increase regularization parameter
                model.fit(self.X_train_scaled, self.y_train[target])
                y_pred = model.predict(self.X_test_scaled)

                # Calculate performance metrics
                r2 = r2_score(self.y_test[target], y_pred)
                mse = mean_squared_error(self.y_test[target], y_pred)
                epoch_scores.append({'r2': r2, 'mse': mse})
                print(f"Epoch {epoch + 1}: R² = {r2:.4f}, MSE = {mse:.4f}")

            svr_results[target] = epoch_scores
            svr_models[target] = model
        return {'results': svr_results, 'models': svr_models}

    def train_random_forest(self, epochs=50):
        """
        Train Random Forest Regression models for each target variable.

        Args:
            epochs (int): Number of training epochs

        Returns:
            dict: Contains training results and trained models
        """
        print("\n## Training Random Forest")
        rf_results = {}
        rf_models = {}
```

```
        for target in self.target_columns:
            print(f"\nTraining Random Forest for {target}")
            epoch_scores = []
            trees_per_epoch = 5  # Number of trees to add each epoch

            # Incrementally add trees to the forest
            for epoch in range(epochs):
                new_model = RandomForestRegressor(
                    n_estimators=trees_per_epoch*(epoch+1),
                    random_state=42
                )
                new_model.fit(self.X_train_scaled, self.y_train[target])
                y_pred = new_model.predict(self.X_test_scaled)

                # Calculate performance metrics
                r2 = r2_score(self.y_test[target], y_pred)
                mse = mean_squared_error(self.y_test[target], y_pred)
                epoch_scores.append({'r2': r2, 'mse': mse})
                print(f"Epoch {epoch + 1}: R² = {r2:.4f}, MSE = {mse:.4f}")

            rf_results[target] = epoch_scores
            rf_models[target] = new_model
        return {'results': rf_results, 'models': rf_models}

    def train_xgboost(self, epochs=50):
        """
        Train XGBoost Regression models for each target variable.

        Args:
            epochs (int): Number of training epochs

        Returns:
            dict: Contains training results and trained models
        """
        print("\n## Training XGBoost")
        xgb_results = {}
        xgb_models = {}

        for target in self.target_columns:
            print(f"\nTraining XGBoost for {target}")
            model = XGBRegressor(n_estimators=epochs, learning_rate=0.1)
            epoch_scores = []

            # Train full model
            model.fit(
                self.X_train_scaled,
                self.y_train[target],
                eval_set=[(self.X_test_scaled, self.y_test[target])],
                verbose=False
            )

            # Evaluate performance at each boosting round
            for i in range(epochs):
                current_model = XGBRegressor(n_estimators=i+1, learning_rate=0.1)
                current_model.fit(self.X_train_scaled, self.y_train[target], verbose=False)
                y_pred = current_model.predict(self.X_test_scaled)

                # Calculate performance metrics
                r2 = r2_score(self.y_test[target], y_pred)
                mse = mean_squared_error(self.y_test[target], y_pred)
                epoch_scores.append({'r2': r2, 'mse': mse})
                print(f"Epoch {i + 1}: R² = {r2:.4f}, MSE = {mse:.4f}")

            xgb_results[target] = epoch_scores
            xgb_models[target] = model

        return {'results': xgb_results, 'models': xgb_models}

# Initialize trainer and train all models
trainer = ClimateModelTrainer(merged_df)
svr_output = trainer.train_svr()
rf_output = trainer.train_random_forest()
xgb_output = trainer.train_xgboost()
```

```python
# Collect results for comparison and visualization
results = {
    'SVR': svr_output,
    'Random Forest': rf_output,
    'XGBoost': xgb_output
}
```

```
C:\Users\kaibe\AppData\Local\Temp\ipykernel_20932\4088147669.py:57: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
  self.X = data[self.climate_features].fillna(method='ffill')
```

Training SVR

Training SVR for Beef and Buffalo Meat, primary
Epoch 1: R² = -1.4197, MSE = 67100.1416
Epoch 2: R² = -1.4186, MSE = 67067.7995
Epoch 3: R² = -1.4174, MSE = 67035.4738
Epoch 4: R² = -1.4162, MSE = 67003.1644
Epoch 5: R² = -1.4151, MSE = 66970.8713
Epoch 6: R² = -1.4139, MSE = 66938.5946
Epoch 7: R² = -1.4128, MSE = 66906.3343
Epoch 8: R² = -1.4116, MSE = 66874.0902
Epoch 9: R² = -1.4104, MSE = 66841.8626
Epoch 10: R² = -1.4093, MSE = 66809.6512
Epoch 11: R² = -1.4081, MSE = 66777.4562
Epoch 12: R² = -1.4069, MSE = 66745.2776
Epoch 13: R² = -1.4058, MSE = 66713.1153
Epoch 14: R² = -1.4046, MSE = 66680.9693
Epoch 15: R² = -1.4035, MSE = 66648.8397
Epoch 16: R² = -1.4023, MSE = 66616.7264
Epoch 17: R² = -1.4012, MSE = 66584.6295
Epoch 18: R² = -1.4000, MSE = 66552.5489
Epoch 19: R² = -1.3988, MSE = 66520.4847
Epoch 20: R² = -1.3977, MSE = 66488.4368
Epoch 21: R² = -1.3965, MSE = 66456.4052
Epoch 22: R² = -1.3954, MSE = 66424.3900
Epoch 23: R² = -1.3941, MSE = 66389.4275
Epoch 24: R² = -1.3921, MSE = 66333.2136
Epoch 25: R² = -1.3901, MSE = 66277.5293
Epoch 26: R² = -1.3881, MSE = 66222.3974
Epoch 27: R² = -1.3861, MSE = 66167.2922
Epoch 28: R² = -1.3841, MSE = 66112.2136
Epoch 29: R² = -1.3821, MSE = 66057.1616
Epoch 30: R² = -1.3801, MSE = 66002.1363
Epoch 31: R² = -1.3782, MSE = 65947.1376
Epoch 32: R² = -1.3762, MSE = 65892.1655
Epoch 33: R² = -1.3742, MSE = 65837.2201
Epoch 34: R² = -1.3722, MSE = 65782.3013
Epoch 35: R² = -1.3702, MSE = 65727.4091
Epoch 36: R² = -1.3683, MSE = 65672.5436
Epoch 37: R² = -1.3663, MSE = 65617.7047
Epoch 38: R² = -1.3643, MSE = 65562.8924
Epoch 39: R² = -1.3623, MSE = 65508.1068
Epoch 40: R² = -1.3604, MSE = 65453.3478
Epoch 41: R² = -1.3584, MSE = 65398.6154
Epoch 42: R² = -1.3564, MSE = 65343.9097
Epoch 43: R² = -1.3544, MSE = 65289.2307
Epoch 44: R² = -1.3525, MSE = 65234.5782
Epoch 45: R² = -1.3505, MSE = 65179.9524
Epoch 46: R² = -1.3485, MSE = 65125.3532
Epoch 47: R² = -1.3466, MSE = 65070.7807
Epoch 48: R² = -1.3446, MSE = 65016.2348
Epoch 49: R² = -1.3426, MSE = 64961.7155
Epoch 50: R² = -1.3407, MSE = 64907.2229

Training SVR for Cereals, primary
Epoch 1: R² = -0.6983, MSE = 869113.2992
Epoch 2: R² = -0.6978, MSE = 868883.3952
Epoch 3: R² = -0.6974, MSE = 868653.5255
Epoch 4: R² = -0.6969, MSE = 868423.6901
Epoch 5: R² = -0.6965, MSE = 868193.8891
Epoch 6: R² = -0.6960, MSE = 867964.1224
Epoch 7: R² = -0.6956, MSE = 867734.3901
Epoch 8: R² = -0.6951, MSE = 867504.6920
Epoch 9: R² = -0.6947, MSE = 867275.0283
Epoch 10: R² = -0.6942, MSE = 867045.3990
Epoch 11: R² = -0.6938, MSE = 866815.8039
Epoch 12: R² = -0.6933, MSE = 866586.2432
Epoch 13: R² = -0.6929, MSE = 866356.7168
Epoch 14: R² = -0.6924, MSE = 866127.2248
Epoch 15: R² = -0.6920, MSE = 865897.7671
Epoch 16: R² = -0.6915, MSE = 865668.3437
Epoch 17: R² = -0.6911, MSE = 865438.9546
Epoch 18: R² = -0.6906, MSE = 865209.5999
Epoch 19: R² = -0.6902, MSE = 864980.2794

```
Epoch 20: R² = -0.6897, MSE = 864750.9934
Epoch 21: R² = -0.6893, MSE = 864521.7416
Epoch 22: R² = -0.6888, MSE = 864292.5242
Epoch 23: R² = -0.6884, MSE = 864063.3411
Epoch 24: R² = -0.6879, MSE = 863834.1923
Epoch 25: R² = -0.6875, MSE = 863605.0779
Epoch 26: R² = -0.6870, MSE = 863375.9978
Epoch 27: R² = -0.6866, MSE = 863146.9520
Epoch 28: R² = -0.6861, MSE = 862917.9406
Epoch 29: R² = -0.6857, MSE = 862688.9635
Epoch 30: R² = -0.6853, MSE = 862460.0207
Epoch 31: R² = -0.6848, MSE = 862231.1122
Epoch 32: R² = -0.6844, MSE = 862002.2381
Epoch 33: R² = -0.6839, MSE = 861773.3983
Epoch 34: R² = -0.6835, MSE = 861544.5928
Epoch 35: R² = -0.6830, MSE = 861315.8217
Epoch 36: R² = -0.6826, MSE = 861087.0849
Epoch 37: R² = -0.6821, MSE = 860858.3824
Epoch 38: R² = -0.6817, MSE = 860629.7142
Epoch 39: R² = -0.6812, MSE = 860401.0804
Epoch 40: R² = -0.6808, MSE = 860172.4809
Epoch 41: R² = -0.6803, MSE = 859943.9158
Epoch 42: R² = -0.6799, MSE = 859715.3849
Epoch 43: R² = -0.6794, MSE = 859486.8884
Epoch 44: R² = -0.6790, MSE = 859258.4262
Epoch 45: R² = -0.6785, MSE = 859029.9984
Epoch 46: R² = -0.6781, MSE = 858801.6049
Epoch 47: R² = -0.6777, MSE = 858573.2457
Epoch 48: R² = -0.6772, MSE = 858344.9208
Epoch 49: R² = -0.6768, MSE = 858116.6303
Epoch 50: R² = -0.6763, MSE = 857888.3741

Training SVR for Eggs Primary
Epoch 1: R² = -0.3477, MSE = 44894864.9540
Epoch 2: R² = -0.3476, MSE = 44893288.4424
Epoch 3: R² = -0.3476, MSE = 44891711.9651
Epoch 4: R² = -0.3475, MSE = 44890135.5221
Epoch 5: R² = -0.3475, MSE = 44888559.1135
Epoch 6: R² = -0.3474, MSE = 44886982.7392
Epoch 7: R² = -0.3474, MSE = 44885406.3992
Epoch 8: R² = -0.3474, MSE = 44883830.0936
Epoch 9: R² = -0.3473, MSE = 44882253.8222
Epoch 10: R² = -0.3473, MSE = 44880677.5852
Epoch 11: R² = -0.3472, MSE = 44879101.3826
Epoch 12: R² = -0.3472, MSE = 44877525.2142
Epoch 13: R² = -0.3471, MSE = 44875949.0802
Epoch 14: R² = -0.3471, MSE = 44874372.9805
Epoch 15: R² = -0.3470, MSE = 44872796.9152
Epoch 16: R² = -0.3470, MSE = 44871220.8842
Epoch 17: R² = -0.3469, MSE = 44869644.8875
Epoch 18: R² = -0.3469, MSE = 44868068.9251
Epoch 19: R² = -0.3468, MSE = 44866492.9971
Epoch 20: R² = -0.3468, MSE = 44864917.1034
Epoch 21: R² = -0.3467, MSE = 44863341.2440
Epoch 22: R² = -0.3467, MSE = 44861765.4190
Epoch 23: R² = -0.3466, MSE = 44860189.6283
Epoch 24: R² = -0.3466, MSE = 44858613.8719
Epoch 25: R² = -0.3465, MSE = 44857038.1498
Epoch 26: R² = -0.3465, MSE = 44855462.4621
Epoch 27: R² = -0.3465, MSE = 44853886.8087
Epoch 28: R² = -0.3464, MSE = 44852311.1896
Epoch 29: R² = -0.3464, MSE = 44850735.6049
Epoch 30: R² = -0.3463, MSE = 44849160.0545
Epoch 31: R² = -0.3463, MSE = 44847584.5384
Epoch 32: R² = -0.3462, MSE = 44846009.0566
Epoch 33: R² = -0.3462, MSE = 44844433.6092
Epoch 34: R² = -0.3461, MSE = 44842858.1961
Epoch 35: R² = -0.3461, MSE = 44841282.8174
Epoch 36: R² = -0.3460, MSE = 44839707.4729
Epoch 37: R² = -0.3460, MSE = 44838132.1628
Epoch 38: R² = -0.3459, MSE = 44836556.8870
Epoch 39: R² = -0.3459, MSE = 44834981.6456
Epoch 40: R² = -0.3458, MSE = 44833406.4385
Epoch 41: R² = -0.3458, MSE = 44831831.2657
```

```
Epoch 42: R² = -0.3457, MSE = 44830256.1272
Epoch 43: R² = -0.3457, MSE = 44828681.0231
Epoch 44: R² = -0.3456, MSE = 44827105.9533
Epoch 45: R² = -0.3456, MSE = 44825530.9178
Epoch 46: R² = -0.3456, MSE = 44823955.9167
Epoch 47: R² = -0.3455, MSE = 44822380.9499
Epoch 48: R² = -0.3455, MSE = 44820806.0174
Epoch 49: R² = -0.3454, MSE = 44819231.1192
Epoch 50: R² = -0.3454, MSE = 44817656.2554

Training SVR for Fibre Crops, Fibre Equivalent
Epoch 1: R² = -0.3977, MSE = 6533.9288
Epoch 2: R² = -0.3943, MSE = 6518.0849
Epoch 3: R² = -0.3909, MSE = 6502.2734
Epoch 4: R² = -0.3875, MSE = 6486.4942
Epoch 5: R² = -0.3841, MSE = 6470.7473
Epoch 6: R² = -0.3808, MSE = 6455.0327
Epoch 7: R² = -0.3774, MSE = 6439.3505
Epoch 8: R² = -0.3741, MSE = 6423.7005
Epoch 9: R² = -0.3707, MSE = 6408.0829
Epoch 10: R² = -0.3674, MSE = 6392.4976
Epoch 11: R² = -0.3641, MSE = 6376.9447
Epoch 12: R² = -0.3608, MSE = 6361.4240
Epoch 13: R² = -0.3574, MSE = 6345.9357
Epoch 14: R² = -0.3541, MSE = 6330.4797
Epoch 15: R² = -0.3508, MSE = 6315.0560
Epoch 16: R² = -0.3475, MSE = 6299.6647
Epoch 17: R² = -0.3443, MSE = 6284.3057
Epoch 18: R² = -0.3410, MSE = 6268.9789
Epoch 19: R² = -0.3377, MSE = 6253.6846
Epoch 20: R² = -0.3344, MSE = 6238.4225
Epoch 21: R² = -0.3312, MSE = 6223.1928
Epoch 22: R² = -0.3279, MSE = 6207.9953
Epoch 23: R² = -0.3247, MSE = 6192.8302
Epoch 24: R² = -0.3215, MSE = 6177.6975
Epoch 25: R² = -0.3182, MSE = 6162.5970
Epoch 26: R² = -0.3150, MSE = 6147.5289
Epoch 27: R² = -0.3118, MSE = 6132.4931
Epoch 28: R² = -0.3086, MSE = 6117.4896
Epoch 29: R² = -0.3054, MSE = 6102.5184
Epoch 30: R² = -0.3022, MSE = 6087.5796
Epoch 31: R² = -0.2990, MSE = 6072.6731
Epoch 32: R² = -0.2958, MSE = 6057.7989
Epoch 33: R² = -0.2926, MSE = 6042.9570
Epoch 34: R² = -0.2895, MSE = 6028.1474
Epoch 35: R² = -0.2863, MSE = 6013.3702
Epoch 36: R² = -0.2832, MSE = 5998.6253
Epoch 37: R² = -0.2800, MSE = 5983.9127
Epoch 38: R² = -0.2769, MSE = 5969.2325
Epoch 39: R² = -0.2737, MSE = 5954.5845
Epoch 40: R² = -0.2706, MSE = 5939.9689
Epoch 41: R² = -0.2675, MSE = 5925.3856
Epoch 42: R² = -0.2644, MSE = 5910.8346
Epoch 43: R² = -0.2613, MSE = 5896.3160
Epoch 44: R² = -0.2582, MSE = 5881.8296
Epoch 45: R² = -0.2551, MSE = 5867.3756
Epoch 46: R² = -0.2520, MSE = 5852.9539
Epoch 47: R² = -0.2489, MSE = 5838.5646
Epoch 48: R² = -0.2458, MSE = 5824.2075
Epoch 49: R² = -0.2428, MSE = 5809.8828
Epoch 50: R² = -0.2397, MSE = 5795.5904

Training SVR for Fruit Primary
Epoch 1: R² = -0.1212, MSE = 1610114.5634
Epoch 2: R² = -0.1212, MSE = 1610072.5974
Epoch 3: R² = -0.1212, MSE = 1610030.6571
Epoch 4: R² = -0.1211, MSE = 1609988.7423
Epoch 5: R² = -0.1211, MSE = 1609946.8532
Epoch 6: R² = -0.1211, MSE = 1609904.9897
Epoch 7: R² = -0.1211, MSE = 1609863.1519
Epoch 8: R² = -0.1210, MSE = 1609821.3396
Epoch 9: R² = -0.1210, MSE = 1609779.5530
Epoch 10: R² = -0.1210, MSE = 1609737.7920
Epoch 11: R² = -0.1209, MSE = 1609696.0567
```

```
Epoch 12: R² = -0.1209, MSE = 1609654.3469
Epoch 13: R² = -0.1209, MSE = 1609612.6628
Epoch 14: R² = -0.1209, MSE = 1609571.0043
Epoch 15: R² = -0.1208, MSE = 1609529.3714
Epoch 16: R² = -0.1208, MSE = 1609487.7642
Epoch 17: R² = -0.1208, MSE = 1609446.1825
Epoch 18: R² = -0.1207, MSE = 1609404.6265
Epoch 19: R² = -0.1207, MSE = 1609363.0961
Epoch 20: R² = -0.1207, MSE = 1609321.5914
Epoch 21: R² = -0.1207, MSE = 1609280.1122
Epoch 22: R² = -0.1206, MSE = 1609238.6587
Epoch 23: R² = -0.1206, MSE = 1609197.2308
Epoch 24: R² = -0.1206, MSE = 1609155.8285
Epoch 25: R² = -0.1205, MSE = 1609114.4519
Epoch 26: R² = -0.1205, MSE = 1609073.1009
Epoch 27: R² = -0.1205, MSE = 1609031.7755
Epoch 28: R² = -0.1205, MSE = 1608990.4757
Epoch 29: R² = -0.1204, MSE = 1608949.2015
Epoch 30: R² = -0.1204, MSE = 1608907.9530
Epoch 31: R² = -0.1204, MSE = 1608866.7301
Epoch 32: R² = -0.1203, MSE = 1608825.5328
Epoch 33: R² = -0.1203, MSE = 1608784.3611
Epoch 34: R² = -0.1203, MSE = 1608743.2151
Epoch 35: R² = -0.1203, MSE = 1608702.0947
Epoch 36: R² = -0.1202, MSE = 1608660.9999
Epoch 37: R² = -0.1202, MSE = 1608619.9307
Epoch 38: R² = -0.1202, MSE = 1608578.8871
Epoch 39: R² = -0.1201, MSE = 1608537.8692
Epoch 40: R² = -0.1201, MSE = 1608496.8769
Epoch 41: R² = -0.1201, MSE = 1608455.9102
Epoch 42: R² = -0.1201, MSE = 1608414.9692
Epoch 43: R² = -0.1200, MSE = 1608374.0537
Epoch 44: R² = -0.1200, MSE = 1608333.1639
Epoch 45: R² = -0.1200, MSE = 1608292.2997
Epoch 46: R² = -0.1199, MSE = 1608251.4612
Epoch 47: R² = -0.1199, MSE = 1608210.6482
Epoch 48: R² = -0.1199, MSE = 1608169.8609
Epoch 49: R² = -0.1199, MSE = 1608129.0992
Epoch 50: R² = -0.1198, MSE = 1608088.3631

Training SVR for Meat, Poultry
Epoch 1: R² = -1.4660, MSE = 1845536.8324
Epoch 2: R² = -1.4658, MSE = 1845416.1756
Epoch 3: R² = -1.4656, MSE = 1845295.5298
Epoch 4: R² = -1.4655, MSE = 1845174.8948
Epoch 5: R² = -1.4653, MSE = 1845054.2707
Epoch 6: R² = -1.4652, MSE = 1844933.6575
Epoch 7: R² = -1.4650, MSE = 1844813.0552
Epoch 8: R² = -1.4648, MSE = 1844692.4638
Epoch 9: R² = -1.4647, MSE = 1844571.8833
Epoch 10: R² = -1.4645, MSE = 1844451.3137
Epoch 11: R² = -1.4644, MSE = 1844330.7550
Epoch 12: R² = -1.4642, MSE = 1844210.2071
Epoch 13: R² = -1.4640, MSE = 1844089.6702
Epoch 14: R² = -1.4639, MSE = 1843969.1441
Epoch 15: R² = -1.4637, MSE = 1843848.6289
Epoch 16: R² = -1.4636, MSE = 1843728.1246
Epoch 17: R² = -1.4634, MSE = 1843607.6312
Epoch 18: R² = -1.4632, MSE = 1843487.1487
Epoch 19: R² = -1.4631, MSE = 1843366.6771
Epoch 20: R² = -1.4629, MSE = 1843246.2163
Epoch 21: R² = -1.4628, MSE = 1843125.7665
Epoch 22: R² = -1.4626, MSE = 1843005.3275
Epoch 23: R² = -1.4624, MSE = 1842884.8995
Epoch 24: R² = -1.4623, MSE = 1842764.4823
Epoch 25: R² = -1.4621, MSE = 1842644.0760
Epoch 26: R² = -1.4619, MSE = 1842523.6806
Epoch 27: R² = -1.4618, MSE = 1842403.2961
Epoch 28: R² = -1.4616, MSE = 1842282.9225
Epoch 29: R² = -1.4615, MSE = 1842162.5598
Epoch 30: R² = -1.4613, MSE = 1842042.2079
Epoch 31: R² = -1.4611, MSE = 1841921.8670
Epoch 32: R² = -1.4610, MSE = 1841801.5369
Epoch 33: R² = -1.4608, MSE = 1841681.2178
```

```
Epoch 34: R² = -1.4607, MSE = 1841560.9095
Epoch 35: R² = -1.4605, MSE = 1841440.6121
Epoch 36: R² = -1.4603, MSE = 1841320.3256
Epoch 37: R² = -1.4602, MSE = 1841200.0500
Epoch 38: R² = -1.4600, MSE = 1841079.7853
Epoch 39: R² = -1.4599, MSE = 1840959.5314
Epoch 40: R² = -1.4597, MSE = 1840839.2885
Epoch 41: R² = -1.4595, MSE = 1840719.0564
Epoch 42: R² = -1.4594, MSE = 1840598.8353
Epoch 43: R² = -1.4592, MSE = 1840478.6250
Epoch 44: R² = -1.4591, MSE = 1840358.4256
Epoch 45: R² = -1.4589, MSE = 1840238.2371
Epoch 46: R² = -1.4587, MSE = 1840118.0595
Epoch 47: R² = -1.4586, MSE = 1839997.8928
Epoch 48: R² = -1.4584, MSE = 1839877.7370
Epoch 49: R² = -1.4583, MSE = 1839757.5920
Epoch 50: R² = -1.4581, MSE = 1839637.4580

Training SVR for Milk, Total
Epoch 1: R² = -0.5047, MSE = 79579262.0821
Epoch 2: R² = -0.5046, MSE = 79577126.6964
Epoch 3: R² = -0.5046, MSE = 79574991.3429
Epoch 4: R² = -0.5045, MSE = 79572856.0217
Epoch 5: R² = -0.5045, MSE = 79570720.7327
Epoch 6: R² = -0.5045, MSE = 79568585.4760
Epoch 7: R² = -0.5044, MSE = 79566450.2516
Epoch 8: R² = -0.5044, MSE = 79564315.0593
Epoch 9: R² = -0.5043, MSE = 79562179.8993
Epoch 10: R² = -0.5043, MSE = 79560044.7716
Epoch 11: R² = -0.5043, MSE = 79557909.6761
Epoch 12: R² = -0.5042, MSE = 79555774.6128
Epoch 13: R² = -0.5042, MSE = 79553639.5818
Epoch 14: R² = -0.5041, MSE = 79551504.5831
Epoch 15: R² = -0.5041, MSE = 79549369.6165
Epoch 16: R² = -0.5041, MSE = 79547234.6823
Epoch 17: R² = -0.5040, MSE = 79545099.7802
Epoch 18: R² = -0.5040, MSE = 79542964.9104
Epoch 19: R² = -0.5039, MSE = 79540830.0729
Epoch 20: R² = -0.5039, MSE = 79538695.2676
Epoch 21: R² = -0.5038, MSE = 79536560.4945
Epoch 22: R² = -0.5038, MSE = 79534425.7537
Epoch 23: R² = -0.5038, MSE = 79532291.0452
Epoch 24: R² = -0.5037, MSE = 79530156.3688
Epoch 25: R² = -0.5037, MSE = 79528021.7247
Epoch 26: R² = -0.5036, MSE = 79525887.1129
Epoch 27: R² = -0.5036, MSE = 79523752.5333
Epoch 28: R² = -0.5036, MSE = 79521617.9860
Epoch 29: R² = -0.5035, MSE = 79519483.4709
Epoch 30: R² = -0.5035, MSE = 79517348.9880
Epoch 31: R² = -0.5034, MSE = 79515214.5374
Epoch 32: R² = -0.5034, MSE = 79513080.1190
Epoch 33: R² = -0.5034, MSE = 79510945.7329
Epoch 34: R² = -0.5033, MSE = 79508811.3790
Epoch 35: R² = -0.5033, MSE = 79506677.0573
Epoch 36: R² = -0.5032, MSE = 79504542.7679
Epoch 37: R² = -0.5032, MSE = 79502408.5108
Epoch 38: R² = -0.5032, MSE = 79500274.2859
Epoch 39: R² = -0.5031, MSE = 79498140.0932
Epoch 40: R² = -0.5031, MSE = 79496005.9328
Epoch 41: R² = -0.5030, MSE = 79493871.8046
Epoch 42: R² = -0.5030, MSE = 79491737.7087
Epoch 43: R² = -0.5030, MSE = 79489603.6450
Epoch 44: R² = -0.5029, MSE = 79487469.6135
Epoch 45: R² = -0.5029, MSE = 79485335.6143
Epoch 46: R² = -0.5028, MSE = 79483201.6474
Epoch 47: R² = -0.5028, MSE = 79481067.7127
Epoch 48: R² = -0.5028, MSE = 79478933.8102
Epoch 49: R² = -0.5027, MSE = 79476799.9400
Epoch 50: R² = -0.5027, MSE = 79474666.1020

Training SVR for Treenuts, Total
Epoch 1: R² = -0.0820, MSE = 297423.1239
Epoch 2: R² = -0.0819, MSE = 297386.1421
Epoch 3: R² = -0.0818, MSE = 297349.1896
```

```
Epoch 4: R² = -0.0816, MSE = 297312.2664
Epoch 5: R² = -0.0815, MSE = 297275.3724
Epoch 6: R² = -0.0814, MSE = 297238.5078
Epoch 7: R² = -0.0812, MSE = 297201.6724
Epoch 8: R² = -0.0811, MSE = 297164.8663
Epoch 9: R² = -0.0810, MSE = 297128.0895
Epoch 10: R² = -0.0808, MSE = 297091.3420
Epoch 11: R² = -0.0807, MSE = 297054.6238
Epoch 12: R² = -0.0806, MSE = 297017.9348
Epoch 13: R² = -0.0804, MSE = 296981.2752
Epoch 14: R² = -0.0803, MSE = 296944.6448
Epoch 15: R² = -0.0802, MSE = 296908.0437
Epoch 16: R² = -0.0800, MSE = 296871.4719
Epoch 17: R² = -0.0799, MSE = 296834.9294
Epoch 18: R² = -0.0798, MSE = 296798.4161
Epoch 19: R² = -0.0796, MSE = 296761.9322
Epoch 20: R² = -0.0795, MSE = 296725.4775
Epoch 21: R² = -0.0794, MSE = 296689.0521
Epoch 22: R² = -0.0792, MSE = 296652.6561
Epoch 23: R² = -0.0791, MSE = 296616.2892
Epoch 24: R² = -0.0790, MSE = 296579.9517
Epoch 25: R² = -0.0789, MSE = 296543.6435
Epoch 26: R² = -0.0787, MSE = 296507.3645
Epoch 27: R² = -0.0786, MSE = 296471.1149
Epoch 28: R² = -0.0785, MSE = 296434.8945
Epoch 29: R² = -0.0783, MSE = 296398.7034
Epoch 30: R² = -0.0782, MSE = 296362.5416
Epoch 31: R² = -0.0781, MSE = 296326.4090
Epoch 32: R² = -0.0779, MSE = 296290.3058
Epoch 33: R² = -0.0778, MSE = 296254.2318
Epoch 34: R² = -0.0777, MSE = 296218.1872
Epoch 35: R² = -0.0775, MSE = 296182.1718
Epoch 36: R² = -0.0774, MSE = 296146.1857
Epoch 37: R² = -0.0773, MSE = 296110.2289
Epoch 38: R² = -0.0771, MSE = 296074.3013
Epoch 39: R² = -0.0770, MSE = 296038.4031
Epoch 40: R² = -0.0769, MSE = 296002.5341
Epoch 41: R² = -0.0768, MSE = 295966.6944
Epoch 42: R² = -0.0766, MSE = 295930.8840
Epoch 43: R² = -0.0765, MSE = 295895.1029
Epoch 44: R² = -0.0764, MSE = 295859.3511
Epoch 45: R² = -0.0762, MSE = 295823.6286
Epoch 46: R² = -0.0761, MSE = 295787.9353
Epoch 47: R² = -0.0760, MSE = 295752.2714
Epoch 48: R² = -0.0758, MSE = 295716.6367
Epoch 49: R² = -0.0757, MSE = 295681.0313
Epoch 50: R² = -0.0756, MSE = 295645.4552

Training SVR for Vegetables Primary
Epoch 1: R² = -1.7619, MSE = 5793073.9094
Epoch 2: R² = -1.7616, MSE = 5792615.9809
Epoch 3: R² = -1.7614, MSE = 5792158.0794
Epoch 4: R² = -1.7612, MSE = 5791700.2049
Epoch 5: R² = -1.7610, MSE = 5791242.3575
Epoch 6: R² = -1.7608, MSE = 5790784.5371
Epoch 7: R² = -1.7606, MSE = 5790326.7438
Epoch 8: R² = -1.7603, MSE = 5789868.9775
Epoch 9: R² = -1.7601, MSE = 5789411.2382
Epoch 10: R² = -1.7599, MSE = 5788953.5260
Epoch 11: R² = -1.7597, MSE = 5788495.8408
Epoch 12: R² = -1.7595, MSE = 5788038.1826
Epoch 13: R² = -1.7592, MSE = 5787580.5515
Epoch 14: R² = -1.7590, MSE = 5787122.9474
Epoch 15: R² = -1.7588, MSE = 5786665.3704
Epoch 16: R² = -1.7586, MSE = 5786207.8204
Epoch 17: R² = -1.7584, MSE = 5785750.2975
Epoch 18: R² = -1.7582, MSE = 5785292.8016
Epoch 19: R² = -1.7579, MSE = 5784835.3327
Epoch 20: R² = -1.7577, MSE = 5784377.8908
Epoch 21: R² = -1.7575, MSE = 5783920.4760
Epoch 22: R² = -1.7573, MSE = 5783463.0883
Epoch 23: R² = -1.7571, MSE = 5783005.7276
Epoch 24: R² = -1.7568, MSE = 5782548.3939
Epoch 25: R² = -1.7566, MSE = 5782091.0873
```

```
Epoch 26: R² = -1.7564, MSE = 5781633.8077
Epoch 27: R² = -1.7562, MSE = 5781176.5551
Epoch 28: R² = -1.7560, MSE = 5780719.3296
Epoch 29: R² = -1.7558, MSE = 5780262.1311
Epoch 30: R² = -1.7555, MSE = 5779804.9596
Epoch 31: R² = -1.7553, MSE = 5779347.8152
Epoch 32: R² = -1.7551, MSE = 5778890.6979
Epoch 33: R² = -1.7549, MSE = 5778433.6076
Epoch 34: R² = -1.7547, MSE = 5777976.5443
Epoch 35: R² = -1.7544, MSE = 5777519.5080
Epoch 36: R² = -1.7542, MSE = 5777062.4988
Epoch 37: R² = -1.7540, MSE = 5776605.5166
Epoch 38: R² = -1.7538, MSE = 5776148.5615
Epoch 39: R² = -1.7536, MSE = 5775691.6334
Epoch 40: R² = -1.7534, MSE = 5775234.7324
Epoch 41: R² = -1.7531, MSE = 5774777.8584
Epoch 42: R² = -1.7529, MSE = 5774321.0114
Epoch 43: R² = -1.7527, MSE = 5773864.1915
Epoch 44: R² = -1.7525, MSE = 5773407.3986
Epoch 45: R² = -1.7523, MSE = 5772950.6327
Epoch 46: R² = -1.7520, MSE = 5772493.8939
Epoch 47: R² = -1.7518, MSE = 5772037.1821
Epoch 48: R² = -1.7516, MSE = 5771580.4974
Epoch 49: R² = -1.7514, MSE = 5771123.8397
Epoch 50: R² = -1.7512, MSE = 5770667.2090

## Training Random Forest

Training Random Forest for Beef and Buffalo Meat, primary
Epoch 1: R² = 0.2503, MSE = 20788.9000
Epoch 2: R² = 0.0761, MSE = 25620.9375
Epoch 3: R² = 0.2330, MSE = 21270.0344
Epoch 4: R² = 0.3289, MSE = 18609.9587
Epoch 5: R² = 0.3285, MSE = 18620.9560
Epoch 6: R² = 0.3698, MSE = 17474.8942
Epoch 7: R² = 0.3851, MSE = 17052.5706
Epoch 8: R² = 0.3715, MSE = 17427.7930
Epoch 9: R² = 0.4057, MSE = 16480.2641
Epoch 10: R² = 0.3999, MSE = 16639.6963
Epoch 11: R² = 0.3713, MSE = 17434.8663
Epoch 12: R² = 0.3457, MSE = 18144.4951
Epoch 13: R² = 0.3614, MSE = 17707.2621
Epoch 14: R² = 0.3873, MSE = 16989.6586
Epoch 15: R² = 0.4081, MSE = 16413.9639
Epoch 16: R² = 0.3975, MSE = 16708.3004
Epoch 17: R² = 0.3979, MSE = 16696.2647
Epoch 18: R² = 0.4036, MSE = 16538.2738
Epoch 19: R² = 0.4192, MSE = 16106.8838
Epoch 20: R² = 0.4209, MSE = 16058.7201
Epoch 21: R² = 0.4186, MSE = 16121.6323
Epoch 22: R² = 0.4111, MSE = 16331.0284
Epoch 23: R² = 0.4150, MSE = 16220.9078
Epoch 24: R² = 0.4119, MSE = 16308.6744
Epoch 25: R² = 0.4033, MSE = 16545.7684
Epoch 26: R² = 0.4029, MSE = 16557.2428
Epoch 27: R² = 0.3978, MSE = 16699.2833
Epoch 28: R² = 0.3892, MSE = 16936.3585
Epoch 29: R² = 0.3952, MSE = 16772.0820
Epoch 30: R² = 0.4008, MSE = 16615.0548
Epoch 31: R² = 0.4069, MSE = 16446.0145
Epoch 32: R² = 0.4049, MSE = 16501.0856
Epoch 33: R² = 0.4075, MSE = 16431.4720
Epoch 34: R² = 0.4122, MSE = 16298.9137
Epoch 35: R² = 0.4202, MSE = 16078.6162
Epoch 36: R² = 0.4123, MSE = 16296.2003
Epoch 37: R² = 0.4202, MSE = 16077.6405
Epoch 38: R² = 0.4134, MSE = 16265.6281
Epoch 39: R² = 0.4174, MSE = 16154.2977
Epoch 40: R² = 0.4274, MSE = 15879.6590
Epoch 41: R² = 0.4329, MSE = 15724.7793
Epoch 42: R² = 0.4359, MSE = 15643.1430
Epoch 43: R² = 0.4412, MSE = 15494.7275
Epoch 44: R² = 0.4442, MSE = 15412.4725
Epoch 45: R² = 0.4389, MSE = 15559.8503
```

```
Epoch 46: R² = 0.4310, MSE = 15777.4150
Epoch 47: R² = 0.4312, MSE = 15772.7653
Epoch 48: R² = 0.4291, MSE = 15832.3194
Epoch 49: R² = 0.4292, MSE = 15829.1427
Epoch 50: R² = 0.4349, MSE = 15670.4446

Training Random Forest for Cereals, primary
Epoch 1: R² = 0.7375, MSE = 134331.7150
Epoch 2: R² = 0.6523, MSE = 177965.1154
Epoch 3: R² = 0.7625, MSE = 121530.2279
Epoch 4: R² = 0.7301, MSE = 138115.5298
Epoch 5: R² = 0.6985, MSE = 154300.2078
Epoch 6: R² = 0.7439, MSE = 131053.6858
Epoch 7: R² = 0.7538, MSE = 126003.7359
Epoch 8: R² = 0.7231, MSE = 141685.2426
Epoch 9: R² = 0.7349, MSE = 135690.0771
Epoch 10: R² = 0.7377, MSE = 134238.7913
Epoch 11: R² = 0.7227, MSE = 141914.1883
Epoch 12: R² = 0.7232, MSE = 141664.1198
Epoch 13: R² = 0.7531, MSE = 126341.6391
Epoch 14: R² = 0.7372, MSE = 134492.2141
Epoch 15: R² = 0.7194, MSE = 143590.8904
Epoch 16: R² = 0.7113, MSE = 147757.8966
Epoch 17: R² = 0.7019, MSE = 152556.3134
Epoch 18: R² = 0.7072, MSE = 149830.7319
Epoch 19: R² = 0.7084, MSE = 149210.0308
Epoch 20: R² = 0.7083, MSE = 149300.6141
Epoch 21: R² = 0.7196, MSE = 143509.1555
Epoch 22: R² = 0.7097, MSE = 148549.5204
Epoch 23: R² = 0.7042, MSE = 151391.4576
Epoch 24: R² = 0.7115, MSE = 147623.2750
Epoch 25: R² = 0.7074, MSE = 149757.9928
Epoch 26: R² = 0.6988, MSE = 154138.1528
Epoch 27: R² = 0.6918, MSE = 157722.1813
Epoch 28: R² = 0.6856, MSE = 160894.9735
Epoch 29: R² = 0.6903, MSE = 158513.9459
Epoch 30: R² = 0.6919, MSE = 157660.1721
Epoch 31: R² = 0.6861, MSE = 160669.1403
Epoch 32: R² = 0.6839, MSE = 161787.0436
Epoch 33: R² = 0.6795, MSE = 164044.9779
Epoch 34: R² = 0.6901, MSE = 158589.3543
Epoch 35: R² = 0.6956, MSE = 155774.4969
Epoch 36: R² = 0.6874, MSE = 159997.9722
Epoch 37: R² = 0.6895, MSE = 158916.2716
Epoch 38: R² = 0.6936, MSE = 156812.1747
Epoch 39: R² = 0.6958, MSE = 155694.8134
Epoch 40: R² = 0.6972, MSE = 154942.7033
Epoch 41: R² = 0.7068, MSE = 150070.7253
Epoch 42: R² = 0.7109, MSE = 147954.5958
Epoch 43: R² = 0.7133, MSE = 146705.4924
Epoch 44: R² = 0.7164, MSE = 145136.9010
Epoch 45: R² = 0.7041, MSE = 151438.3196
Epoch 46: R² = 0.7048, MSE = 151051.4503
Epoch 47: R² = 0.7098, MSE = 148511.1760
Epoch 48: R² = 0.7021, MSE = 152471.0022
Epoch 49: R² = 0.6957, MSE = 155724.0814
Epoch 50: R² = 0.6919, MSE = 157672.8411

Training Random Forest for Eggs Primary
Epoch 1: R² = 0.7625, MSE = 7911320.0400
Epoch 2: R² = 0.5842, MSE = 13851788.2875
Epoch 3: R² = 0.7025, MSE = 9911753.4389
Epoch 4: R² = 0.7203, MSE = 9317642.3012
Epoch 5: R² = 0.6781, MSE = 10723408.6996
Epoch 6: R² = 0.6999, MSE = 9996978.0778
Epoch 7: R² = 0.6917, MSE = 10270339.4845
Epoch 8: R² = 0.6746, MSE = 10839114.3527
Epoch 9: R² = 0.6669, MSE = 11097703.4506
Epoch 10: R² = 0.6713, MSE = 10948269.9020
Epoch 11: R² = 0.6718, MSE = 10934074.6328
Epoch 12: R² = 0.6659, MSE = 11131341.8328
Epoch 13: R² = 0.6648, MSE = 11164802.9612
Epoch 14: R² = 0.6651, MSE = 11156411.8729
Epoch 15: R² = 0.6679, MSE = 11062227.7568
```

```
Epoch 16: R² = 0.6649, MSE = 11164388.3754
Epoch 17: R² = 0.6639, MSE = 11197943.4965
Epoch 18: R² = 0.6637, MSE = 11204640.6932
Epoch 19: R² = 0.6716, MSE = 10938844.9767
Epoch 20: R² = 0.6828, MSE = 10567954.9127
Epoch 21: R² = 0.6757, MSE = 10803004.0139
Epoch 22: R² = 0.6708, MSE = 10967714.4978
Epoch 23: R² = 0.6712, MSE = 10953121.4469
Epoch 24: R² = 0.6685, MSE = 11041695.3214
Epoch 25: R² = 0.6655, MSE = 11144709.2063
Epoch 26: R² = 0.6594, MSE = 11346260.2813
Epoch 27: R² = 0.6624, MSE = 11247938.9570
Epoch 28: R² = 0.6604, MSE = 11312643.1929
Epoch 29: R² = 0.6579, MSE = 11397905.4015
Epoch 30: R² = 0.6592, MSE = 11353988.2218
Epoch 31: R² = 0.6593, MSE = 11349540.5102
Epoch 32: R² = 0.6510, MSE = 11627251.0920
Epoch 33: R² = 0.6569, MSE = 11430256.1508
Epoch 34: R² = 0.6580, MSE = 11392867.4546
Epoch 35: R² = 0.6621, MSE = 11255986.3442
Epoch 36: R² = 0.6599, MSE = 11328632.8287
Epoch 37: R² = 0.6639, MSE = 11197440.1603
Epoch 38: R² = 0.6625, MSE = 11244576.1844
Epoch 39: R² = 0.6600, MSE = 11327332.1439
Epoch 40: R² = 0.6633, MSE = 11216039.1519
Epoch 41: R² = 0.6644, MSE = 11181071.4546
Epoch 42: R² = 0.6635, MSE = 11208965.4765
Epoch 43: R² = 0.6668, MSE = 11098874.5995
Epoch 44: R² = 0.6698, MSE = 10998224.2727
Epoch 45: R² = 0.6687, MSE = 11035086.1548
Epoch 46: R² = 0.6623, MSE = 11250616.6599
Epoch 47: R² = 0.6642, MSE = 11186183.1111
Epoch 48: R² = 0.6598, MSE = 11333405.7483
Epoch 49: R² = 0.6541, MSE = 11524022.2227
Epoch 50: R² = 0.6554, MSE = 11480580.8548

Training Random Forest for Fibre Crops, Fibre Equivalent
Epoch 1: R² = 0.1990, MSE = 3744.6350
Epoch 2: R² = 0.1391, MSE = 4024.7071
Epoch 3: R² = 0.4585, MSE = 2531.6489
Epoch 4: R² = 0.5244, MSE = 2223.2113
Epoch 5: R² = 0.5004, MSE = 2335.7429
Epoch 6: R² = 0.4845, MSE = 2410.0098
Epoch 7: R² = 0.4604, MSE = 2522.6491
Epoch 8: R² = 0.3579, MSE = 3001.6997
Epoch 9: R² = 0.3554, MSE = 3013.5323
Epoch 10: R² = 0.3707, MSE = 2941.9641
Epoch 11: R² = 0.3018, MSE = 3264.2288
Epoch 12: R² = 0.3277, MSE = 3142.7415
Epoch 13: R² = 0.2842, MSE = 3346.1247
Epoch 14: R² = 0.3042, MSE = 3253.0002
Epoch 15: R² = 0.3320, MSE = 3123.0584
Epoch 16: R² = 0.3723, MSE = 2934.6144
Epoch 17: R² = 0.3725, MSE = 2933.5119
Epoch 18: R² = 0.3651, MSE = 2968.3172
Epoch 19: R² = 0.3832, MSE = 2883.5546
Epoch 20: R² = 0.3751, MSE = 2921.4147
Epoch 21: R² = 0.3910, MSE = 2846.8480
Epoch 22: R² = 0.4071, MSE = 2771.7473
Epoch 23: R² = 0.4237, MSE = 2694.2603
Epoch 24: R² = 0.4273, MSE = 2677.4169
Epoch 25: R² = 0.4190, MSE = 2715.9288
Epoch 26: R² = 0.4369, MSE = 2632.3430
Epoch 27: R² = 0.4215, MSE = 2704.3758
Epoch 28: R² = 0.4183, MSE = 2719.1988
Epoch 29: R² = 0.4252, MSE = 2687.1668
Epoch 30: R² = 0.4337, MSE = 2647.6110
Epoch 31: R² = 0.4352, MSE = 2640.4694
Epoch 32: R² = 0.4464, MSE = 2588.0785
Epoch 33: R² = 0.4505, MSE = 2568.9480
Epoch 34: R² = 0.4573, MSE = 2536.8606
Epoch 35: R² = 0.4629, MSE = 2510.8492
Epoch 36: R² = 0.4572, MSE = 2537.3823
Epoch 37: R² = 0.4513, MSE = 2565.3459
```

```
Epoch 38: R² = 0.4528, MSE = 2558.1249
Epoch 39: R² = 0.4612, MSE = 2519.0380
Epoch 40: R² = 0.4694, MSE = 2480.2910
Epoch 41: R² = 0.4723, MSE = 2467.1556
Epoch 42: R² = 0.4729, MSE = 2464.3199
Epoch 43: R² = 0.4740, MSE = 2458.8287
Epoch 44: R² = 0.4809, MSE = 2426.6274
Epoch 45: R² = 0.4869, MSE = 2398.5076
Epoch 46: R² = 0.4850, MSE = 2407.5119
Epoch 47: R² = 0.4807, MSE = 2427.5437
Epoch 48: R² = 0.4870, MSE = 2398.0241
Epoch 49: R² = 0.4898, MSE = 2385.3701
Epoch 50: R² = 0.4917, MSE = 2376.2372

Training Random Forest for Fruit Primary
Epoch 1: R² = 0.7301, MSE = 387610.5572
Epoch 2: R² = 0.5770, MSE = 607441.9225
Epoch 3: R² = 0.2783, MSE = 1036417.5686
Epoch 4: R² = 0.2837, MSE = 1028665.9829
Epoch 5: R² = 0.2306, MSE = 1104876.6844
Epoch 6: R² = 0.1591, MSE = 1207540.1561
Epoch 7: R² = 0.1495, MSE = 1221372.3717
Epoch 8: R² = 0.1483, MSE = 1223116.9469
Epoch 9: R² = 0.1379, MSE = 1238020.7688
Epoch 10: R² = 0.1139, MSE = 1272503.8877
Epoch 11: R² = 0.0994, MSE = 1293340.2318
Epoch 12: R² = 0.1158, MSE = 1269657.3458
Epoch 13: R² = 0.1458, MSE = 1226639.5070
Epoch 14: R² = 0.1488, MSE = 1222319.4080
Epoch 15: R² = 0.1790, MSE = 1178912.6122
Epoch 16: R² = 0.1750, MSE = 1184688.5333
Epoch 17: R² = 0.1574, MSE = 1210010.6077
Epoch 18: R² = 0.1520, MSE = 1217768.1206
Epoch 19: R² = 0.1562, MSE = 1211707.4036
Epoch 20: R² = 0.1393, MSE = 1236033.4555
Epoch 21: R² = 0.1288, MSE = 1251090.0038
Epoch 22: R² = 0.1394, MSE = 1235864.4303
Epoch 23: R² = 0.1432, MSE = 1230406.0049
Epoch 24: R² = 0.1571, MSE = 1210367.9002
Epoch 25: R² = 0.1429, MSE = 1230794.2347
Epoch 26: R² = 0.1392, MSE = 1236125.1590
Epoch 27: R² = 0.1364, MSE = 1240117.7635
Epoch 28: R² = 0.1467, MSE = 1225286.2951
Epoch 29: R² = 0.1398, MSE = 1235234.3351
Epoch 30: R² = 0.1386, MSE = 1236959.3698
Epoch 31: R² = 0.1278, MSE = 1252560.0961
Epoch 32: R² = 0.1378, MSE = 1238154.1652
Epoch 33: R² = 0.1231, MSE = 1259173.1683
Epoch 34: R² = 0.1122, MSE = 1274923.1775
Epoch 35: R² = 0.1047, MSE = 1285616.1765
Epoch 36: R² = 0.1147, MSE = 1271360.9168
Epoch 37: R² = 0.1082, MSE = 1280640.2746
Epoch 38: R² = 0.1049, MSE = 1285404.4587
Epoch 39: R² = 0.0973, MSE = 1296249.0247
Epoch 40: R² = 0.1016, MSE = 1290096.7608
Epoch 41: R² = 0.1076, MSE = 1281533.4040
Epoch 42: R² = 0.1065, MSE = 1283112.0732
Epoch 43: R² = 0.1064, MSE = 1283187.1097
Epoch 44: R² = 0.1096, MSE = 1278661.6915
Epoch 45: R² = 0.1159, MSE = 1269538.4434
Epoch 46: R² = 0.1139, MSE = 1272398.4244
Epoch 47: R² = 0.1179, MSE = 1266714.1652
Epoch 48: R² = 0.1261, MSE = 1254939.9498
Epoch 49: R² = 0.1433, MSE = 1230300.4177
Epoch 50: R² = 0.1461, MSE = 1226151.1892

Training Random Forest for Meat, Poultry
Epoch 1: R² = -0.2762, MSE = 955100.8600
Epoch 2: R² = -0.7820, MSE = 1333684.9450
Epoch 3: R² = -0.3108, MSE = 980996.1522
Epoch 4: R² = -0.1212, MSE = 839076.0713
Epoch 5: R² = -0.0964, MSE = 820539.0932
Epoch 6: R² = -0.1414, MSE = 854207.1419
Epoch 7: R² = -0.2442, MSE = 931142.8724
```

```
Epoch 8: R² = -0.3112, MSE = 981277.3603
Epoch 9: R² = -0.3123, MSE = 982131.8264
Epoch 10: R² = -0.3897, MSE = 1040088.6370
Epoch 11: R² = -0.3715, MSE = 1026402.7160
Epoch 12: R² = -0.3383, MSE = 1001621.4323
Epoch 13: R² = -0.3632, MSE = 1020204.8889
Epoch 14: R² = -0.3442, MSE = 1006020.6535
Epoch 15: R² = -0.3658, MSE = 1022174.5274
Epoch 16: R² = -0.4194, MSE = 1062312.9002
Epoch 17: R² = -0.4005, MSE = 1048099.4305
Epoch 18: R² = -0.3812, MSE = 1033716.3779
Epoch 19: R² = -0.3305, MSE = 995782.3564
Epoch 20: R² = -0.3283, MSE = 994127.0964
Epoch 21: R² = -0.4124, MSE = 1057026.8870
Epoch 22: R² = -0.4003, MSE = 1048021.2076
Epoch 23: R² = -0.4034, MSE = 1050327.5051
Epoch 24: R² = -0.4032, MSE = 1050169.2619
Epoch 25: R² = -0.4265, MSE = 1067574.4210
Epoch 26: R² = -0.4014, MSE = 1048801.2794
Epoch 27: R² = -0.3934, MSE = 1042830.6614
Epoch 28: R² = -0.4087, MSE = 1054257.8110
Epoch 29: R² = -0.4229, MSE = 1064878.7925
Epoch 30: R² = -0.4154, MSE = 1059306.6097
Epoch 31: R² = -0.4100, MSE = 1055229.6315
Epoch 32: R² = -0.4304, MSE = 1070496.6854
Epoch 33: R² = -0.4237, MSE = 1065492.5425
Epoch 34: R² = -0.4335, MSE = 1072827.4571
Epoch 35: R² = -0.4345, MSE = 1073553.3167
Epoch 36: R² = -0.4278, MSE = 1068596.9133
Epoch 37: R² = -0.4090, MSE = 1054490.4181
Epoch 38: R² = -0.4002, MSE = 1047900.1852
Epoch 39: R² = -0.4019, MSE = 1049173.7653
Epoch 40: R² = -0.4073, MSE = 1053253.1016
Epoch 41: R² = -0.3932, MSE = 1042705.6483
Epoch 42: R² = -0.3753, MSE = 1029298.1435
Epoch 43: R² = -0.3573, MSE = 1015797.2190
Epoch 44: R² = -0.3450, MSE = 1006626.7288
Epoch 45: R² = -0.3422, MSE = 1004519.0029
Epoch 46: R² = -0.3433, MSE = 1005304.9036
Epoch 47: R² = -0.3321, MSE = 996960.9362
Epoch 48: R² = -0.3407, MSE = 1003386.3533
Epoch 49: R² = -0.3478, MSE = 1008697.9255
Epoch 50: R² = -0.3456, MSE = 1007019.6633

Training Random Forest for Milk, Total
Epoch 1: R² = 0.7439, MSE = 13543723.0200
Epoch 2: R² = 0.5950, MSE = 21422223.9225
Epoch 3: R² = 0.7189, MSE = 14865980.5278
Epoch 4: R² = 0.7514, MSE = 13149033.1494
Epoch 5: R² = 0.7698, MSE = 12175300.8420
Epoch 6: R² = 0.7768, MSE = 11807205.6594
Epoch 7: R² = 0.7840, MSE = 11422560.3245
Epoch 8: R² = 0.7950, MSE = 10841442.6297
Epoch 9: R² = 0.7960, MSE = 10791585.5626
Epoch 10: R² = 0.7969, MSE = 10742361.8166
Epoch 11: R² = 0.7873, MSE = 11250935.0103
Epoch 12: R² = 0.7798, MSE = 11647401.8870
Epoch 13: R² = 0.7798, MSE = 11647436.6485
Epoch 14: R² = 0.7865, MSE = 11292091.7452
Epoch 15: R² = 0.7927, MSE = 10965060.6056
Epoch 16: R² = 0.7968, MSE = 10745366.6886
Epoch 17: R² = 0.8008, MSE = 10534535.3699
Epoch 18: R² = 0.8030, MSE = 10417519.0281
Epoch 19: R² = 0.8018, MSE = 10482558.9155
Epoch 20: R² = 0.8100, MSE = 10050042.6383
Epoch 21: R² = 0.8106, MSE = 10018453.0493
Epoch 22: R² = 0.8004, MSE = 10556376.9026
Epoch 23: R² = 0.7981, MSE = 10676978.8352
Epoch 24: R² = 0.7939, MSE = 10902087.9752
Epoch 25: R² = 0.7939, MSE = 10901485.3007
Epoch 26: R² = 0.7979, MSE = 10691112.3543
Epoch 27: R² = 0.7980, MSE = 10683494.7014
Epoch 28: R² = 0.7938, MSE = 10905599.3675
Epoch 29: R² = 0.7929, MSE = 10953240.4993
```

```
Epoch 30: R² = 0.7930, MSE = 10949314.1153
Epoch 31: R² = 0.7928, MSE = 10956812.9860
Epoch 32: R² = 0.7894, MSE = 11139041.9984
Epoch 33: R² = 0.7905, MSE = 11081425.4991
Epoch 34: R² = 0.7933, MSE = 10931220.4967
Epoch 35: R² = 0.7946, MSE = 10861370.9339
Epoch 36: R² = 0.7902, MSE = 11093428.9849
Epoch 37: R² = 0.7909, MSE = 11060553.8826
Epoch 38: R² = 0.7857, MSE = 11334704.1883
Epoch 39: R² = 0.7828, MSE = 11488524.0637
Epoch 40: R² = 0.7822, MSE = 11521273.6826
Epoch 41: R² = 0.7813, MSE = 11565420.9775
Epoch 42: R² = 0.7820, MSE = 11527759.1930
Epoch 43: R² = 0.7850, MSE = 11370873.7400
Epoch 44: R² = 0.7865, MSE = 11291799.6246
Epoch 45: R² = 0.7843, MSE = 11408582.4385
Epoch 46: R² = 0.7809, MSE = 11590032.6440
Epoch 47: R² = 0.7824, MSE = 11508776.1478
Epoch 48: R² = 0.7776, MSE = 11764279.2272
Epoch 49: R² = 0.7777, MSE = 11755864.4018
Epoch 50: R² = 0.7780, MSE = 11739560.5701

Training Random Forest for Treenuts, Total
Epoch 1: R² = 0.4121, MSE = 161584.4151
Epoch 2: R² = 0.4126, MSE = 161451.1462
Epoch 3: R² = 0.3126, MSE = 188948.2180
Epoch 4: R² = 0.2548, MSE = 204831.7876
Epoch 5: R² = 0.2872, MSE = 195924.3265
Epoch 6: R² = 0.2834, MSE = 196981.5673
Epoch 7: R² = 0.2670, MSE = 201467.3481
Epoch 8: R² = 0.2550, MSE = 204785.8092
Epoch 9: R² = 0.2437, MSE = 207889.2656
Epoch 10: R² = 0.2525, MSE = 205471.2562
Epoch 11: R² = 0.2869, MSE = 196015.1749
Epoch 12: R² = 0.2572, MSE = 204168.4943
Epoch 13: R² = 0.2421, MSE = 208336.2230
Epoch 14: R² = 0.2432, MSE = 208028.9257
Epoch 15: R² = 0.2495, MSE = 206292.7820
Epoch 16: R² = 0.2507, MSE = 205968.0502
Epoch 17: R² = 0.2254, MSE = 212926.4201
Epoch 18: R² = 0.2371, MSE = 209685.0045
Epoch 19: R² = 0.2096, MSE = 217250.8137
Epoch 20: R² = 0.2165, MSE = 215374.0584
Epoch 21: R² = 0.2276, MSE = 212301.0720
Epoch 22: R² = 0.2516, MSE = 205708.6464
Epoch 23: R² = 0.2434, MSE = 207965.1441
Epoch 24: R² = 0.2371, MSE = 209709.0487
Epoch 25: R² = 0.2492, MSE = 206385.5500
Epoch 26: R² = 0.2604, MSE = 203297.7887
Epoch 27: R² = 0.2504, MSE = 206054.8191
Epoch 28: R² = 0.2599, MSE = 203424.4458
Epoch 29: R² = 0.2550, MSE = 204765.7346
Epoch 30: R² = 0.2560, MSE = 204510.4670
Epoch 31: R² = 0.2512, MSE = 205831.9392
Epoch 32: R² = 0.2541, MSE = 205035.7312
Epoch 33: R² = 0.2484, MSE = 206582.6254
Epoch 34: R² = 0.2416, MSE = 208462.6068
Epoch 35: R² = 0.2301, MSE = 211616.3482
Epoch 36: R² = 0.2421, MSE = 208326.6865
Epoch 37: R² = 0.2369, MSE = 209752.6461
Epoch 38: R² = 0.2367, MSE = 209794.8929
Epoch 39: R² = 0.2295, MSE = 211796.2788
Epoch 40: R² = 0.2288, MSE = 211985.8929
Epoch 41: R² = 0.2266, MSE = 212577.3627
Epoch 42: R² = 0.2222, MSE = 213795.3061
Epoch 43: R² = 0.2209, MSE = 214156.2146
Epoch 44: R² = 0.2227, MSE = 213648.8709
Epoch 45: R² = 0.2302, MSE = 211582.8634
Epoch 46: R² = 0.2325, MSE = 210954.9821
Epoch 47: R² = 0.2248, MSE = 213077.5701
Epoch 48: R² = 0.2409, MSE = 208654.9315
Epoch 49: R² = 0.2435, MSE = 207952.0024
Epoch 50: R² = 0.2469, MSE = 207013.0790
```

```
Training Random Forest for Vegetables Primary
Epoch 1: R² = -0.7425, MSE = 3654931.4635
Epoch 2: R² = -1.2822, MSE = 4786947.4515
Epoch 3: R² = -0.6244, MSE = 3407175.7847
Epoch 4: R² = -0.3530, MSE = 2837902.6764
Epoch 5: R² = -0.2852, MSE = 2695692.4870
Epoch 6: R² = -0.1523, MSE = 2416990.6148
Epoch 7: R² = -0.1489, MSE = 2409833.3338
Epoch 8: R² = -0.1506, MSE = 2413506.4691
Epoch 9: R² = -0.1725, MSE = 2459318.1587
Epoch 10: R² = -0.2117, MSE = 2541570.6099
Epoch 11: R² = -0.2495, MSE = 2620812.6241
Epoch 12: R² = -0.3171, MSE = 2762567.3510
Epoch 13: R² = -0.2974, MSE = 2721269.4310
Epoch 14: R² = -0.2553, MSE = 2633035.9078
Epoch 15: R² = -0.2487, MSE = 2619239.5055
Epoch 16: R² = -0.2164, MSE = 2551494.1530
Epoch 17: R² = -0.1761, MSE = 2466813.5990
Epoch 18: R² = -0.1991, MSE = 2515057.4049
Epoch 19: R² = -0.1989, MSE = 2514673.5578
Epoch 20: R² = -0.1986, MSE = 2514195.9650
Epoch 21: R² = -0.2281, MSE = 2576040.6044
Epoch 22: R² = -0.2268, MSE = 2573187.4557
Epoch 23: R² = -0.2490, MSE = 2619884.1078
Epoch 24: R² = -0.2467, MSE = 2614951.8552
Epoch 25: R² = -0.2675, MSE = 2658547.5254
Epoch 26: R² = -0.2573, MSE = 2637126.5793
Epoch 27: R² = -0.2622, MSE = 2647541.9833
Epoch 28: R² = -0.2564, MSE = 2635283.4315
Epoch 29: R² = -0.2459, MSE = 2613379.1802
Epoch 30: R² = -0.2442, MSE = 2609756.5626
Epoch 31: R² = -0.2428, MSE = 2606885.9395
Epoch 32: R² = -0.2826, MSE = 2690203.3033
Epoch 33: R² = -0.2619, MSE = 2646820.5482
Epoch 34: R² = -0.2450, MSE = 2611368.9703
Epoch 35: R² = -0.2477, MSE = 2617113.1633
Epoch 36: R² = -0.2633, MSE = 2649740.9767
Epoch 37: R² = -0.2449, MSE = 2611131.0530
Epoch 38: R² = -0.2350, MSE = 2590346.4518
Epoch 39: R² = -0.2302, MSE = 2580374.7855
Epoch 40: R² = -0.2281, MSE = 2576039.9674
Epoch 41: R² = -0.2264, MSE = 2572448.9335
Epoch 42: R² = -0.2189, MSE = 2556765.3079
Epoch 43: R² = -0.2133, MSE = 2544901.0692
Epoch 44: R² = -0.2096, MSE = 2537063.1650
Epoch 45: R² = -0.2194, MSE = 2557781.4654
Epoch 46: R² = -0.2303, MSE = 2580648.8412
Epoch 47: R² = -0.2190, MSE = 2556800.7408
Epoch 48: R² = -0.2279, MSE = 2575530.5012
Epoch 49: R² = -0.2268, MSE = 2573345.4519
Epoch 50: R² = -0.2258, MSE = 2571115.9865

## Training XGBoost

Training XGBoost for Beef and Buffalo Meat, primary
Epoch 1: R² = -0.7344, MSE = 48095.8092
Epoch 2: R² = -0.5411, MSE = 42734.7336
Epoch 3: R² = -0.3722, MSE = 38052.8084
Epoch 4: R² = -0.2247, MSE = 33961.9003
Epoch 5: R² = -0.0958, MSE = 30385.4548
Epoch 6: R² = 0.0057, MSE = 27571.3169
Epoch 7: R² = 0.1058, MSE = 24795.1080
Epoch 8: R² = 0.1914, MSE = 22422.6697
Epoch 9: R² = 0.2445, MSE = 20950.6863
Epoch 10: R² = 0.3131, MSE = 19048.7446
Epoch 11: R² = 0.3735, MSE = 17373.8060
Epoch 12: R² = 0.4267, MSE = 15898.5402
Epoch 13: R² = 0.4736, MSE = 14597.3665
Epoch 14: R² = 0.5150, MSE = 13449.6100
Epoch 15: R² = 0.5517, MSE = 12430.5157
Epoch 16: R² = 0.5842, MSE = 11531.2972
Epoch 17: R² = 0.6130, MSE = 10731.9849
Epoch 18: R² = 0.6321, MSE = 10203.1033
Epoch 19: R² = 0.6554, MSE = 9557.2206
```

```
Epoch 20: R² = 0.6761, MSE = 8981.3241
Epoch 21: R² = 0.6946, MSE = 8469.7302
Epoch 22: R² = 0.7110, MSE = 8014.8026
Epoch 23: R² = 0.7255, MSE = 7611.4570
Epoch 24: R² = 0.7385, MSE = 7250.3878
Epoch 25: R² = 0.7502, MSE = 6928.2658
Epoch 26: R² = 0.7584, MSE = 6699.3557
Epoch 27: R² = 0.7623, MSE = 6590.1469
Epoch 28: R² = 0.7659, MSE = 6491.2876
Epoch 29: R² = 0.7694, MSE = 6395.5419
Epoch 30: R² = 0.7725, MSE = 6307.9495
Epoch 31: R² = 0.7752, MSE = 6232.6042
Epoch 32: R² = 0.7779, MSE = 6158.7463
Epoch 33: R² = 0.7803, MSE = 6091.0591
Epoch 34: R² = 0.7826, MSE = 6028.9798
Epoch 35: R² = 0.7846, MSE = 5972.0119
Epoch 36: R² = 0.7864, MSE = 5923.0595
Epoch 37: R² = 0.7881, MSE = 5874.7359
Epoch 38: R² = 0.7897, MSE = 5830.3218
Epoch 39: R² = 0.7912, MSE = 5789.2465
Epoch 40: R² = 0.7921, MSE = 5766.2798
Epoch 41: R² = 0.7928, MSE = 5745.1127
Epoch 42: R² = 0.7965, MSE = 5641.7373
Epoch 43: R² = 0.8000, MSE = 5547.3702
Epoch 44: R² = 0.8031, MSE = 5461.1684
Epoch 45: R² = 0.8038, MSE = 5440.1043
Epoch 46: R² = 0.8068, MSE = 5356.4874
Epoch 47: R² = 0.8097, MSE = 5278.3056
Epoch 48: R² = 0.8120, MSE = 5212.6911
Epoch 49: R² = 0.8145, MSE = 5143.7081
Epoch 50: R² = 0.8168, MSE = 5079.1103

Training XGBoost for Cereals, primary
Epoch 1: R² = -0.3474, MSE = 689559.7936
Epoch 2: R² = -0.1774, MSE = 602574.0597
Epoch 3: R² = -0.0323, MSE = 528300.4526
Epoch 4: R² = 0.1102, MSE = 455355.2683
Epoch 5: R² = 0.2317, MSE = 393216.4418
Epoch 6: R² = 0.3308, MSE = 342456.9716
Epoch 7: R² = 0.4149, MSE = 299461.2631
Epoch 8: R² = 0.4804, MSE = 265921.4589
Epoch 9: R² = 0.5195, MSE = 245906.6471
Epoch 10: R² = 0.5541, MSE = 228222.7376
Epoch 11: R² = 0.5896, MSE = 210049.8658
Epoch 12: R² = 0.6187, MSE = 195150.7750
Epoch 13: R² = 0.6430, MSE = 182714.0559
Epoch 14: R² = 0.6666, MSE = 170641.6740
Epoch 15: R² = 0.6876, MSE = 159889.5165
Epoch 16: R² = 0.7063, MSE = 150298.2151
Epoch 17: R² = 0.7233, MSE = 141607.7566
Epoch 18: R² = 0.7373, MSE = 134424.0897
Epoch 19: R² = 0.7598, MSE = 122922.5061
Epoch 20: R² = 0.7731, MSE = 116135.7764
Epoch 21: R² = 0.7841, MSE = 110468.8186
Epoch 22: R² = 0.7942, MSE = 105339.0071
Epoch 23: R² = 0.8033, MSE = 100690.0819
Epoch 24: R² = 0.8112, MSE = 96607.9221
Epoch 25: R² = 0.8187, MSE = 92759.4386
Epoch 26: R² = 0.8256, MSE = 89260.2686
Epoch 27: R² = 0.8285, MSE = 87750.5866
Epoch 28: R² = 0.8312, MSE = 86378.5423
Epoch 29: R² = 0.8337, MSE = 85131.4954
Epoch 30: R² = 0.8363, MSE = 83789.9125
Epoch 31: R² = 0.8383, MSE = 82766.1547
Epoch 32: R² = 0.8401, MSE = 81835.4888
Epoch 33: R² = 0.8417, MSE = 80989.4342
Epoch 34: R² = 0.8432, MSE = 80220.4549
Epoch 35: R² = 0.8446, MSE = 79521.3361
Epoch 36: R² = 0.8461, MSE = 78761.2892
Epoch 37: R² = 0.8472, MSE = 78187.3851
Epoch 38: R² = 0.8482, MSE = 77665.8276
Epoch 39: R² = 0.8492, MSE = 77191.9922
Epoch 40: R² = 0.8500, MSE = 76761.3837
Epoch 41: R² = 0.8508, MSE = 76370.2666
```

```
Epoch 42: R² = 0.8516, MSE = 75939.5594
Epoch 43: R² = 0.8522, MSE = 75619.3362
Epoch 44: R² = 0.8528, MSE = 75328.7060
Epoch 45: R² = 0.8533, MSE = 75064.9271
Epoch 46: R² = 0.8538, MSE = 74825.7180
Epoch 47: R² = 0.8542, MSE = 74608.9274
Epoch 48: R² = 0.8546, MSE = 74412.4816
Epoch 49: R² = 0.8549, MSE = 74234.5062
Epoch 50: R² = 0.8553, MSE = 74073.4436

Training XGBoost for Eggs Primary
Epoch 1: R² = -0.3176, MSE = 43892383.4122
Epoch 2: R² = -0.2230, MSE = 40741400.6084
Epoch 3: R² = -0.1401, MSE = 37980671.7839
Epoch 4: R² = -0.0740, MSE = 35777747.7408
Epoch 5: R² = -0.0317, MSE = 34369464.7233
Epoch 6: R² = 0.0060, MSE = 33114350.6492
Epoch 7: R² = 0.0396, MSE = 31993999.5620
Epoch 8: R² = 0.1000, MSE = 29980758.3779
Epoch 9: R² = 0.1426, MSE = 28563376.7532
Epoch 10: R² = 0.1662, MSE = 27777359.7653
Epoch 11: R² = 0.1873, MSE = 27071969.8238
Epoch 12: R² = 0.2277, MSE = 25726919.1859
Epoch 13: R² = 0.2478, MSE = 25058753.7941
Epoch 14: R² = 0.2630, MSE = 24550004.2309
Epoch 15: R² = 0.2768, MSE = 24090979.5573
Epoch 16: R² = 0.2893, MSE = 23676218.6422
Epoch 17: R² = 0.3069, MSE = 23089657.1027
Epoch 18: R² = 0.3230, MSE = 22553895.3841
Epoch 19: R² = 0.3377, MSE = 22064041.1212
Epoch 20: R² = 0.3461, MSE = 21782950.9254
Epoch 21: R² = 0.3578, MSE = 21393488.7978
Epoch 22: R² = 0.3690, MSE = 21019692.3883
Epoch 23: R² = 0.3793, MSE = 20676676.0917
Epoch 24: R² = 0.3888, MSE = 20361627.9862
Epoch 25: R² = 0.3990, MSE = 20019320.4835
Epoch 26: R² = 0.4085, MSE = 19703865.9261
Epoch 27: R² = 0.4172, MSE = 19413041.7235
Epoch 28: R² = 0.4253, MSE = 19144741.5385
Epoch 29: R² = 0.4327, MSE = 18897153.1381
Epoch 30: R² = 0.4396, MSE = 18668509.8876
Epoch 31: R² = 0.4459, MSE = 18457241.3240
Epoch 32: R² = 0.4516, MSE = 18267563.8962
Epoch 33: R² = 0.4568, MSE = 18094384.4850
Epoch 34: R² = 0.4618, MSE = 17927232.1674
Epoch 35: R² = 0.4665, MSE = 17772469.5834
Epoch 36: R² = 0.4707, MSE = 17633253.4735
Epoch 37: R² = 0.4745, MSE = 17506004.3224
Epoch 38: R² = 0.4782, MSE = 17382895.8424
Epoch 39: R² = 0.4810, MSE = 17290044.9271
Epoch 40: R² = 0.4836, MSE = 17203743.6224
Epoch 41: R² = 0.4860, MSE = 17123475.8913
Epoch 42: R² = 0.4882, MSE = 17048805.8646
Epoch 43: R² = 0.4904, MSE = 16976432.7735
Epoch 44: R² = 0.4925, MSE = 16907527.4510
Epoch 45: R² = 0.4944, MSE = 16843185.5515
Epoch 46: R² = 0.4961, MSE = 16785304.6331
Epoch 47: R² = 0.4978, MSE = 16729045.9435
Epoch 48: R² = 0.4994, MSE = 16676415.1952
Epoch 49: R² = 0.5008, MSE = 16628324.2978
Epoch 50: R² = 0.5022, MSE = 16582462.4843

Training XGBoost for Fibre Crops, Fibre Equivalent
Epoch 1: R² = 0.1284, MSE = 4074.8263
Epoch 2: R² = 0.2814, MSE = 3359.2612
Epoch 3: R² = 0.3947, MSE = 2829.5551
Epoch 4: R² = 0.4754, MSE = 2452.4812
Epoch 5: R² = 0.5308, MSE = 2193.2378
Epoch 6: R² = 0.5644, MSE = 2036.3856
Epoch 7: R² = 0.5803, MSE = 1962.1684
Epoch 8: R² = 0.5703, MSE = 2008.7563
Epoch 9: R² = 0.5617, MSE = 2049.2057
Epoch 10: R² = 0.5453, MSE = 2125.6254
Epoch 11: R² = 0.5220, MSE = 2234.5751
```

```
Epoch 12: R² = 0.4884, MSE = 2391.4661
Epoch 13: R² = 0.4515, MSE = 2564.1751
Epoch 14: R² = 0.4129, MSE = 2744.8140
Epoch 15: R² = 0.3709, MSE = 2940.8014
Epoch 16: R² = 0.3306, MSE = 3129.3454
Epoch 17: R² = 0.2901, MSE = 3318.7246
Epoch 18: R² = 0.2503, MSE = 3504.8963
Epoch 19: R² = 0.2112, MSE = 3687.3698
Epoch 20: R² = 0.1733, MSE = 3864.9382
Epoch 21: R² = 0.1353, MSE = 4042.3804
Epoch 22: R² = 0.0989, MSE = 4212.7405
Epoch 23: R² = 0.0640, MSE = 4375.6487
Epoch 24: R² = 0.0319, MSE = 4525.8385
Epoch 25: R² = 0.0003, MSE = 4673.4061
Epoch 26: R² = -0.0131, MSE = 4736.2474
Epoch 27: R² = -0.0257, MSE = 4795.2810
Epoch 28: R² = -0.0377, MSE = 4850.9497
Epoch 29: R² = -0.0488, MSE = 4903.1463
Epoch 30: R² = -0.0593, MSE = 4952.2988
Epoch 31: R² = -0.0692, MSE = 4998.5604
Epoch 32: R² = -0.0786, MSE = 5042.5645
Epoch 33: R² = -0.0875, MSE = 5083.9105
Epoch 34: R² = -0.0957, MSE = 5122.5305
Epoch 35: R² = -0.1035, MSE = 5158.7950
Epoch 36: R² = -0.1101, MSE = 5189.6414
Epoch 37: R² = -0.1169, MSE = 5221.5871
Epoch 38: R² = -0.1233, MSE = 5251.5541
Epoch 39: R² = -0.1294, MSE = 5279.7625
Epoch 40: R² = -0.1350, MSE = 5306.1227
Epoch 41: R² = -0.1403, MSE = 5330.8253
Epoch 42: R² = -0.1453, MSE = 5353.9701
Epoch 43: R² = -0.1499, MSE = 5375.6477
Epoch 44: R² = -0.1538, MSE = 5394.0254
Epoch 45: R² = -0.1579, MSE = 5413.0348
Epoch 46: R² = -0.1617, MSE = 5430.8240
Epoch 47: R² = -0.1629, MSE = 5436.5828
Epoch 48: R² = -0.1640, MSE = 5441.5636
Epoch 49: R² = -0.1648, MSE = 5445.5101
Epoch 50: R² = -0.1655, MSE = 5448.7020

Training XGBoost for Fruit Primary
Epoch 1: R² = -0.0249, MSE = 1471718.3430
Epoch 2: R² = -0.0199, MSE = 1464541.0913
Epoch 3: R² = -0.0245, MSE = 1471163.8855
Epoch 4: R² = -0.0365, MSE = 1488475.6782
Epoch 5: R² = -0.0543, MSE = 1513922.2938
Epoch 6: R² = -0.0762, MSE = 1545411.6824
Epoch 7: R² = -0.1011, MSE = 1581241.5793
Epoch 8: R² = -0.1036, MSE = 1584843.4605
Epoch 9: R² = -0.1096, MSE = 1593424.0122
Epoch 10: R² = -0.1182, MSE = 1605777.0325
Epoch 11: R² = -0.1288, MSE = 1620916.3196
Epoch 12: R² = -0.1407, MSE = 1638038.4023
Epoch 13: R² = -0.1535, MSE = 1656494.3644
Epoch 14: R² = -0.1670, MSE = 1675761.9379
Epoch 15: R² = -0.1917, MSE = 1711293.7393
Epoch 16: R² = -0.2172, MSE = 1747946.0176
Epoch 17: R² = -0.2417, MSE = 1783041.6534
Epoch 18: R² = -0.2673, MSE = 1819901.1942
Epoch 19: R² = -0.2882, MSE = 1849944.7685
Epoch 20: R² = -0.3119, MSE = 1883887.2246
Epoch 21: R² = -0.3297, MSE = 1909455.9044
Epoch 22: R² = -0.3513, MSE = 1940491.5375
Epoch 23: R² = -0.3718, MSE = 1969908.3031
Epoch 24: R² = -0.3912, MSE = 1997728.4826
Epoch 25: R² = -0.4019, MSE = 2013203.5901
Epoch 26: R² = -0.4192, MSE = 2037938.6167
Epoch 27: R² = -0.4321, MSE = 2056457.7306
Epoch 28: R² = -0.4475, MSE = 2078625.9001
Epoch 29: R² = -0.4620, MSE = 2099441.5744
Epoch 30: R² = -0.4756, MSE = 2118962.7530
Epoch 31: R² = -0.4832, MSE = 2129895.0840
Epoch 32: R² = -0.4897, MSE = 2139295.6492
Epoch 33: R² = -0.4958, MSE = 2148025.6512
```

```
Epoch 34: R² = -0.5015, MSE = 2156134.3823
Epoch 35: R² = -0.5067, MSE = 2163661.1843
Epoch 36: R² = -0.5116, MSE = 2170650.5762
Epoch 37: R² = -0.5161, MSE = 2177139.2953
Epoch 38: R² = -0.5210, MSE = 2184139.8825
Epoch 39: R² = -0.5256, MSE = 2190805.3430
Epoch 40: R² = -0.5300, MSE = 2197152.4792
Epoch 41: R² = -0.5342, MSE = 2203193.8240
Epoch 42: R² = -0.5382, MSE = 2208942.3935
Epoch 43: R² = -0.5421, MSE = 2214412.5003
Epoch 44: R² = -0.5499, MSE = 2225701.7074
Epoch 45: R² = -0.5533, MSE = 2230564.1160
Epoch 46: R² = -0.5565, MSE = 2235176.7705
Epoch 47: R² = -0.5596, MSE = 2239551.3908
Epoch 48: R² = -0.5624, MSE = 2243702.6692
Epoch 49: R² = -0.5652, MSE = 2247638.2324
Epoch 50: R² = -0.5678, MSE = 2251370.8513

Training XGBoost for Meat, Poultry
Epoch 1: R² = -1.4949, MSE = 1867168.6887
Epoch 2: R² = -1.2593, MSE = 1690837.5304
Epoch 3: R² = -1.0590, MSE = 1540933.9080
Epoch 4: R² = -0.8883, MSE = 1413221.9772
Epoch 5: R² = -0.7419, MSE = 1303635.8606
Epoch 6: R² = -0.6180, MSE = 1210933.3688
Epoch 7: R² = -0.5125, MSE = 1131989.5965
Epoch 8: R² = -0.4227, MSE = 1064767.9540
Epoch 9: R² = -0.3451, MSE = 1006699.9001
Epoch 10: R² = -0.2801, MSE = 957992.0644
Epoch 11: R² = -0.2092, MSE = 904971.4141
Epoch 12: R² = -0.1471, MSE = 858510.5250
Epoch 13: R² = -0.1071, MSE = 828569.9882
Epoch 14: R² = -0.0876, MSE = 813979.1924
Epoch 15: R² = -0.0547, MSE = 789365.5741
Epoch 16: R² = -0.0393, MSE = 777834.6980
Epoch 17: R² = -0.0122, MSE = 757495.3676
Epoch 18: R² = -0.0017, MSE = 749638.2707
Epoch 19: R² = 0.0207, MSE = 732923.0459
Epoch 20: R² = 0.0306, MSE = 725484.0221
Epoch 21: R² = 0.0502, MSE = 710844.5909
Epoch 22: R² = 0.0582, MSE = 704866.8237
Epoch 23: R² = 0.0737, MSE = 693230.0548
Epoch 24: R² = 0.0802, MSE = 688395.0978
Epoch 25: R² = 0.0933, MSE = 678605.8520
Epoch 26: R² = 0.1037, MSE = 670770.1694
Epoch 27: R² = 0.1202, MSE = 658411.6233
Epoch 28: R² = 0.1302, MSE = 650974.6699
Epoch 29: R² = 0.1341, MSE = 648029.3497
Epoch 30: R² = 0.1425, MSE = 641716.8210
Epoch 31: R² = 0.1497, MSE = 636390.0121
Epoch 32: R² = 0.1568, MSE = 631025.6402
Epoch 33: R² = 0.1629, MSE = 626522.4281
Epoch 34: R² = 0.1690, MSE = 621948.4293
Epoch 35: R² = 0.1737, MSE = 618403.9516
Epoch 36: R² = 0.1785, MSE = 614846.3099
Epoch 37: R² = 0.1832, MSE = 611275.0504
Epoch 38: R² = 0.1855, MSE = 609554.9046
Epoch 39: R² = 0.1899, MSE = 606288.2706
Epoch 40: R² = 0.1921, MSE = 604663.5106
Epoch 41: R² = 0.1956, MSE = 602027.5815
Epoch 42: R² = 0.1974, MSE = 600653.4130
Epoch 43: R² = 0.2005, MSE = 598376.9614
Epoch 44: R² = 0.2017, MSE = 597469.7664
Epoch 45: R² = 0.2043, MSE = 595533.6455
Epoch 46: R² = 0.2056, MSE = 594527.3000
Epoch 47: R² = 0.2080, MSE = 592736.9912
Epoch 48: R² = 0.2091, MSE = 591878.6352
Epoch 49: R² = 0.2102, MSE = 591082.6280
Epoch 50: R² = 0.2120, MSE = 589727.6824

Training XGBoost for Milk, Total
Epoch 1: R² = -0.2261, MSE = 64848008.7557
Epoch 2: R² = -0.0778, MSE = 57005771.2408
Epoch 3: R² = 0.0501, MSE = 50237206.8331
```

```
Epoch 4: R² = 0.1607, MSE = 44391679.2653
Epoch 5: R² = 0.2460, MSE = 39878589.9670
Epoch 6: R² = 0.3300, MSE = 35437160.2952
Epoch 7: R² = 0.3952, MSE = 31984779.6285
Epoch 8: R² = 0.4520, MSE = 28985246.6121
Epoch 9: R² = 0.5013, MSE = 26376592.3494
Epoch 10: R² = 0.5442, MSE = 24105516.2468
Epoch 11: R² = 0.5781, MSE = 22316130.6148
Epoch 12: R² = 0.6032, MSE = 20983585.3935
Epoch 13: R² = 0.6257, MSE = 19796721.3802
Epoch 14: R² = 0.6457, MSE = 18738420.4958
Epoch 15: R² = 0.6636, MSE = 17793623.7772
Epoch 16: R² = 0.6769, MSE = 17089756.0233
Epoch 17: R² = 0.6888, MSE = 16461230.8108
Epoch 18: R² = 0.6994, MSE = 15899361.1277
Epoch 19: R² = 0.7123, MSE = 15218171.7165
Epoch 20: R² = 0.7207, MSE = 14772505.0358
Epoch 21: R² = 0.7301, MSE = 14275701.0167
Epoch 22: R² = 0.7369, MSE = 13916728.6851
Epoch 23: R² = 0.7430, MSE = 13594024.4394
Epoch 24: R² = 0.7485, MSE = 13303580.7279
Epoch 25: R² = 0.7565, MSE = 12880128.8063
Epoch 26: R² = 0.7609, MSE = 12647773.5331
Epoch 27: R² = 0.7660, MSE = 12377110.7151
Epoch 28: R² = 0.7698, MSE = 12174882.8159
Epoch 29: R² = 0.7733, MSE = 11990673.7525
Epoch 30: R² = 0.7765, MSE = 11822713.4440
Epoch 31: R² = 0.7792, MSE = 11679235.8833
Epoch 32: R² = 0.7818, MSE = 11538593.3611
Epoch 33: R² = 0.7843, MSE = 11409955.2084
Epoch 34: R² = 0.7865, MSE = 11292179.8754
Epoch 35: R² = 0.7898, MSE = 11115215.3129
Epoch 36: R² = 0.7917, MSE = 11018108.0183
Epoch 37: R² = 0.7934, MSE = 10929000.7449
Epoch 38: R² = 0.7949, MSE = 10847178.5656
Epoch 39: R² = 0.7963, MSE = 10771964.8031
Epoch 40: R² = 0.7975, MSE = 10708228.2916
Epoch 41: R² = 0.7987, MSE = 10644219.0684
Epoch 42: R² = 0.7999, MSE = 10585259.0530
Epoch 43: R² = 0.8011, MSE = 10520128.1987
Epoch 44: R² = 0.8022, MSE = 10459965.1527
Epoch 45: R² = 0.8033, MSE = 10404356.3678
Epoch 46: R² = 0.8043, MSE = 10352956.9593
Epoch 47: R² = 0.8053, MSE = 10298645.5535
Epoch 48: R² = 0.8062, MSE = 10248348.3945
Epoch 49: R² = 0.8071, MSE = 10201743.2337
Epoch 50: R² = 0.8079, MSE = 10158537.9734

Training XGBoost for Treenuts, Total
Epoch 1: R² = -0.0143, MSE = 278794.1251
Epoch 2: R² = 0.0629, MSE = 257591.6172
Epoch 3: R² = 0.1225, MSE = 241186.0314
Epoch 4: R² = 0.1679, MSE = 228712.4515
Epoch 5: R² = 0.2016, MSE = 219449.8376
Epoch 6: R² = 0.2258, MSE = 212797.7713
Epoch 7: R² = 0.2426, MSE = 208191.9470
Epoch 8: R² = 0.2527, MSE = 205401.5114
Epoch 9: R² = 0.2582, MSE = 203897.2409
Epoch 10: R² = 0.2420, MSE = 208338.9205
Epoch 11: R² = 0.2234, MSE = 213460.4022
Epoch 12: R² = 0.2184, MSE = 214839.5142
Epoch 13: R² = 0.1962, MSE = 220950.7333
Epoch 14: R² = 0.1729, MSE = 227348.3291
Epoch 15: R² = 0.1495, MSE = 233775.8149
Epoch 16: R² = 0.1276, MSE = 239806.1285
Epoch 17: R² = 0.1059, MSE = 245766.5231
Epoch 18: R² = 0.0846, MSE = 251618.1059
Epoch 19: R² = 0.0648, MSE = 257068.0148
Epoch 20: R² = 0.0455, MSE = 262364.3255
Epoch 21: R² = 0.0268, MSE = 267490.8682
Epoch 22: R² = 0.0089, MSE = 272435.4402
Epoch 23: R² = -0.0092, MSE = 277390.2279
Epoch 24: R² = -0.0249, MSE = 281705.4552
Epoch 25: R² = -0.0409, MSE = 286112.8905
```

```
Epoch 26: R² = -0.0562, MSE = 290319.9032
Epoch 27: R² = -0.0708, MSE = 294328.5630
Epoch 28: R² = -0.0843, MSE = 298048.6066
Epoch 29: R² = -0.0977, MSE = 301712.6840
Epoch 30: R² = -0.1098, MSE = 305055.1306
Epoch 31: R² = -0.1213, MSE = 308218.6555
Epoch 32: R² = -0.1324, MSE = 311273.0970
Epoch 33: R² = -0.1427, MSE = 314093.1315
Epoch 34: R² = -0.1528, MSE = 316881.1892
Epoch 35: R² = -0.1616, MSE = 319288.9495
Epoch 36: R² = -0.1703, MSE = 321682.4088
Epoch 37: R² = -0.1785, MSE = 323934.9813
Epoch 38: R² = -0.1862, MSE = 326054.3494
Epoch 39: R² = -0.1935, MSE = 328046.9237
Epoch 40: R² = -0.2003, MSE = 329919.5618
Epoch 41: R² = -0.2067, MSE = 331696.0365
Epoch 42: R² = -0.2129, MSE = 333393.2506
Epoch 43: R² = -0.2186, MSE = 334954.9572
Epoch 44: R² = -0.2239, MSE = 336419.2938
Epoch 45: R² = -0.2289, MSE = 337792.0299
Epoch 46: R² = -0.2335, MSE = 339063.4970
Epoch 47: R² = -0.2379, MSE = 340255.1887
Epoch 48: R² = -0.2419, MSE = 341371.7566
Epoch 49: R² = -0.2457, MSE = 342417.3366
Epoch 50: R² = -0.2493, MSE = 343396.5127

Training XGBoost for Vegetables Primary
Epoch 1: R² = -1.2949, MSE = 4813597.2299
Epoch 2: R² = -1.1145, MSE = 4435185.2768
Epoch 3: R² = -0.9434, MSE = 4076267.5878
Epoch 4: R² = -0.7949, MSE = 3764793.7669
Epoch 5: R² = -0.6658, MSE = 3493980.1096
Epoch 6: R² = -0.5623, MSE = 3277026.5212
Epoch 7: R² = -0.4633, MSE = 3069306.3255
Epoch 8: R² = -0.3778, MSE = 2889900.6540
Epoch 9: R² = -0.2934, MSE = 2712845.3324
Epoch 10: R² = -0.2195, MSE = 2558027.8884
Epoch 11: R² = -0.1668, MSE = 2447486.6294
Epoch 12: R² = -0.1327, MSE = 2375836.2499
Epoch 13: R² = -0.1009, MSE = 2309061.5137
Epoch 14: R² = -0.0706, MSE = 2245623.8565
Epoch 15: R² = -0.0437, MSE = 2189193.4698
Epoch 16: R² = -0.0209, MSE = 2141407.5169
Epoch 17: R² = 0.0001, MSE = 2097221.9878
Epoch 18: R² = 0.0189, MSE = 2057987.1282
Epoch 19: R² = 0.0356, MSE = 2022946.6449
Epoch 20: R² = 0.0503, MSE = 1992087.5263
Epoch 21: R² = 0.0626, MSE = 1966253.1437
Epoch 22: R² = 0.0742, MSE = 1941917.0860
Epoch 23: R² = 0.0844, MSE = 1920392.1638
Epoch 24: R² = 0.0936, MSE = 1901217.5141
Epoch 25: R² = 0.1088, MSE = 1869259.6148
Epoch 26: R² = 0.1108, MSE = 1865207.4542
Epoch 27: R² = 0.1225, MSE = 1840519.6482
Epoch 28: R² = 0.1223, MSE = 1841032.5973
Epoch 29: R² = 0.1220, MSE = 1841574.4809
Epoch 30: R² = 0.1213, MSE = 1843091.6233
Epoch 31: R² = 0.1184, MSE = 1849192.9317
Epoch 32: R² = 0.1153, MSE = 1855680.8072
Epoch 33: R² = 0.1121, MSE = 1862467.4535
Epoch 34: R² = 0.1087, MSE = 1869480.2204
Epoch 35: R² = 0.1053, MSE = 1876653.1149
Epoch 36: R² = 0.1015, MSE = 1884730.4366
Epoch 37: R² = 0.0976, MSE = 1892894.2382
Epoch 38: R² = 0.0936, MSE = 1901094.6644
Epoch 39: R² = 0.0897, MSE = 1909290.0309
Epoch 40: R² = 0.0859, MSE = 1917441.4767
Epoch 41: R² = 0.0820, MSE = 1925518.3088
Epoch 42: R² = 0.0781, MSE = 1933648.2912
Epoch 43: R² = 0.0743, MSE = 1941622.6553
Epoch 44: R² = 0.0706, MSE = 1949425.3638
Epoch 45: R² = 0.0670, MSE = 1957045.8668
Epoch 46: R² = 0.0635, MSE = 1964347.2494
Epoch 47: R² = 0.0600, MSE = 1971577.2672
```

```
Epoch 48: R² = 0.0567, MSE = 1978600.6736
Epoch 49: R² = 0.0535, MSE = 1985411.3774
Epoch 50: R² = 0.0503, MSE = 1992010.1416
```

Model Performance Visualization System

Overview

The visualization system reveals key patterns in model performance through multiple visualization techniques. The results show distinct patterns across different food categories and models.

Performance Insights from Visualizations

Heatmap Analysis

- Strong performance ($R^2 > 0.8$):
  - XGBoost: Milk, Cereals, Beef/Buffalo
  - Random Forest: Milk production
  - SVR: Fibre crops, Cereals

Training Progress Patterns

- XGBoost shows rapid initial convergence
- Random Forest demonstrates stable learning
- SVR shows gradual improvement with complexity

Feature Importance

- Random Forest emphasizes:
  1. Maximum precipitation (MXPN)
  2. Extreme temperatures (EMNT)
  3. Evaporation (EVAP)
- XGBoost prioritizes:
  1. Evaporation (EVAP)
  2. Extreme maximum temperature (EMXT)
  3. Average dew point (ADPT)

Visualization Components

1. Training Progress Visualization

- Tracks R^2 scores across training epochs
- Different line styles distinguish models
- Color coding for different targets
- Allows comparison of learning rates and convergence

2. Target Comparison Analysis

- Dual subplot system showing R^2 and MSE scores
- Grouped bar charts for easy comparison
- Rotation of labels for readability
- Clear distinction between models for each target

3. Performance Heatmap

- Color-coded visualization of final R^2 scores
- Shows model-target combinations

- Intuitive red-yellow-green color scheme
- Annotated with exact values

4. Feature Importance Analysis

- Compares Random Forest and XGBoost importance rankings
- Horizontal bar charts for clear comparison
- Excludes redundant features
- Shows relative importance of climate variables

Implementation Benefits

- Comprehensive performance assessment
- Easy identification of best models
- Clear visualization of model strengths
- Intuitive comparison across targets

Usage Guidelines

The visualizer requires:

- Model results dictionary
- Feature names list
- Target variable names
- Consistent data structure across models

Visual Design Principles

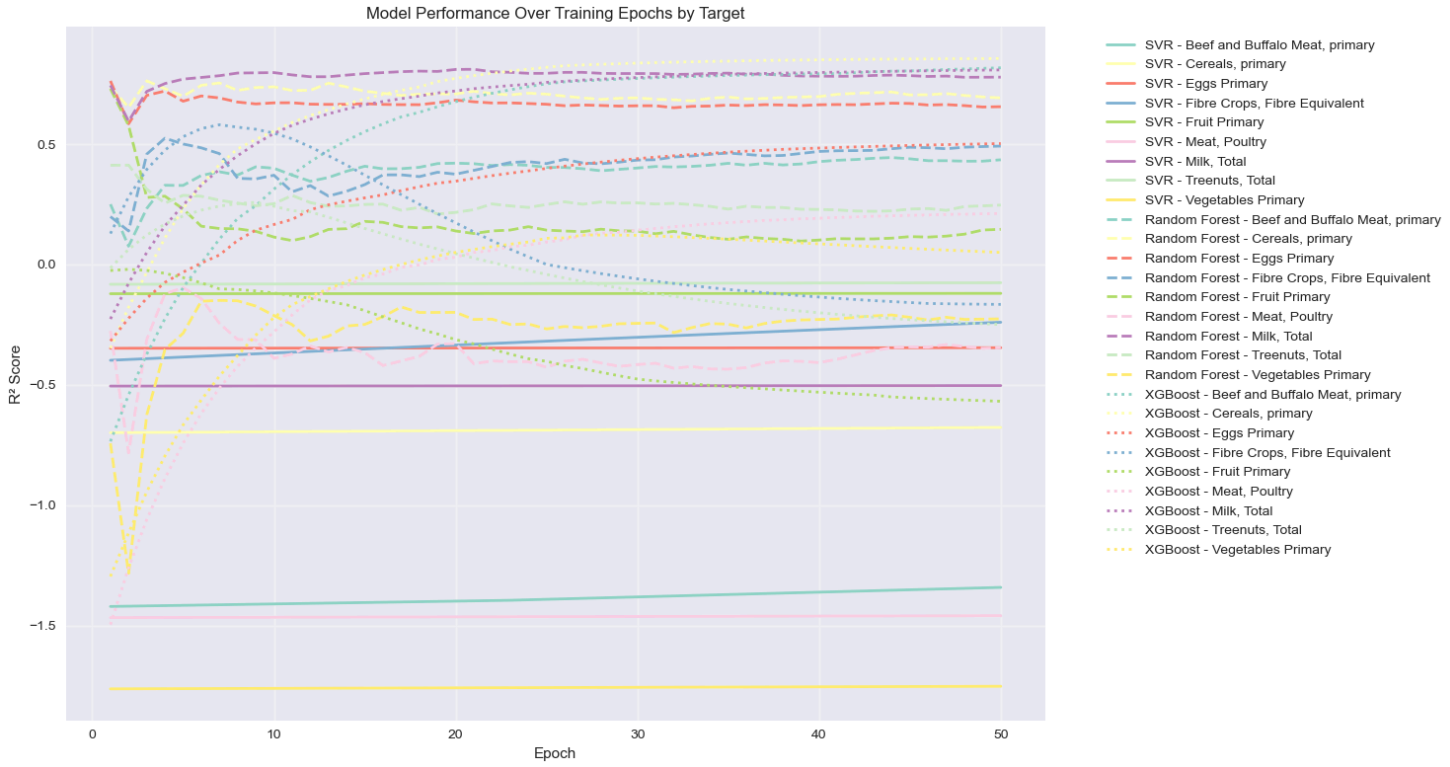
- Consistent color schemes
- Clear labeling
- Appropriate figure sizes
- Optimal use of space

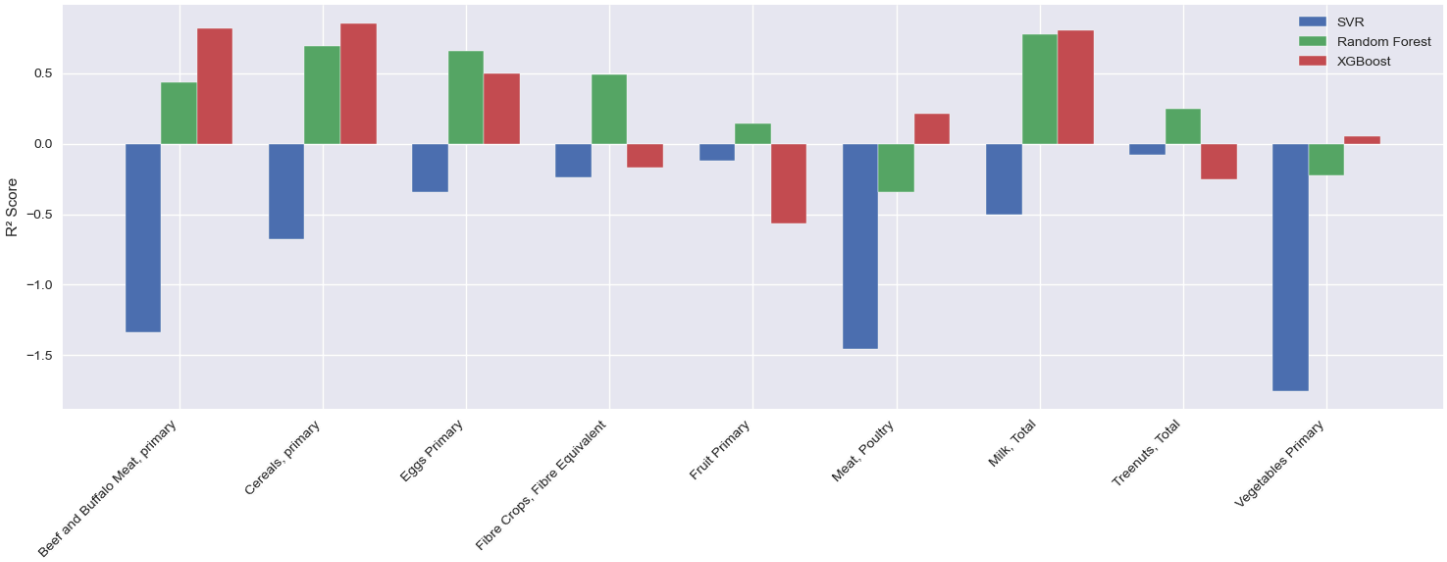
In [12]:
```python
class ModelVisualizer:
    """
    A class for creating comprehensive visualizations of machine learning model results.

    This class provides multiple visualization methods to analyze and compare model
    performance across different targets and training epochs.

    Attributes:
        models_results (dict): Results and models from different algorithms
        feature_names (list): Names of input features
        target_names (list): Names of target variables
    """

    def __init__(self, models_results, feature_names, target_names):
        """
        Initialize the ModelVisualizer with model results and feature information.

        Args:
            models_results (dict): Dictionary containing results from different models
            feature_names (list): List of feature names used in training
            target_names (list): List of target variable names
        """
        self.models_results = models_results
        self.feature_names = feature_names
        self.target_names = target_names
        plt.style.use('seaborn-v0_8')  # Set consistent plotting style

    def plot_training_progress(self):
        """
        Plot R² scores across training epochs for all models and targets.
```

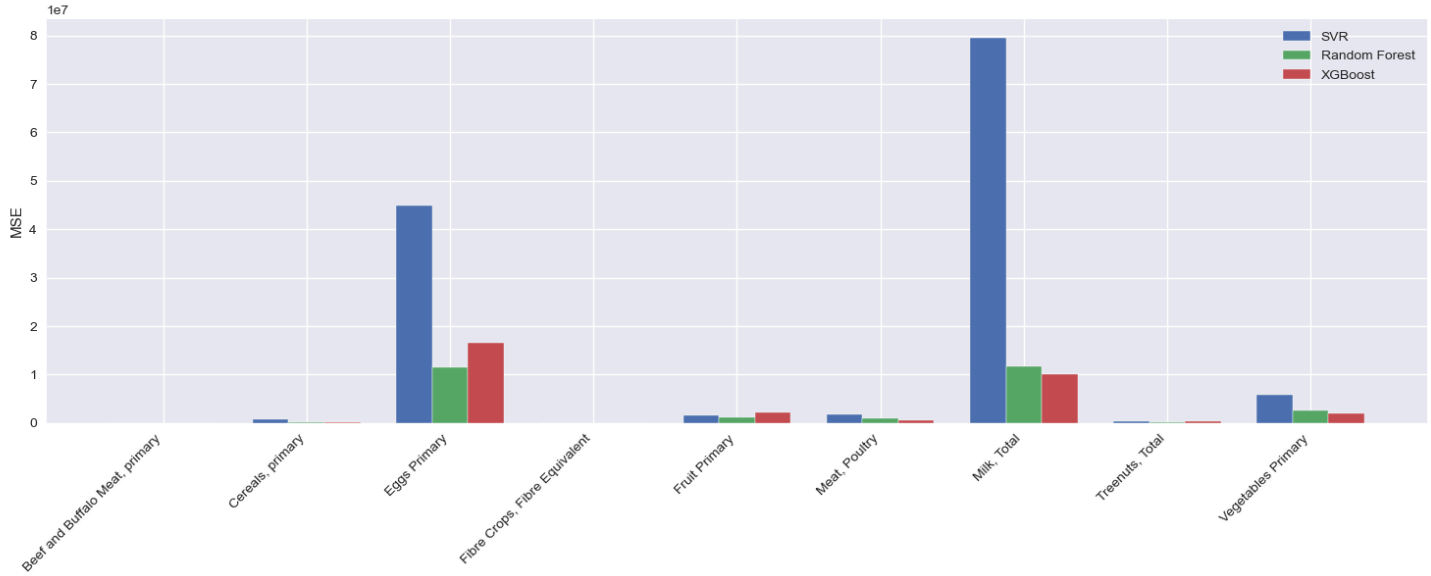
```
        Creates a line plot showing how model performance evolves during training,
        with different line styles for models and colors for targets.
        """
        plt.figure(figsize=(15, 8))

        # Define visual styles for different models
        line_styles = {
            'SVR': '-',
            'Random Forest': '--',
            'XGBoost': ':'
        }

        # Generate distinct colors for different targets
        colors = plt.cm.Set3(np.linspace(0, 1, len(self.target_names)))

        # Create lines for each model-target combination
        for model_name, model_data in self.models_results.items():
            for target_idx, target in enumerate(self.target_names):
                r2_scores = [epoch['r2'] for epoch in model_data['results'][target]]
                plt.plot(range(1, len(r2_scores) + 1), r2_scores,
                        label=f'{model_name} - {target}',
                        linestyle=line_styles[model_name],
                        color=colors[target_idx],
                        linewidth=2)

        # Customize plot appearance
        plt.xlabel('Epoch')
        plt.ylabel('R² Score')
        plt.title('Model Performance Over Training Epochs by Target')
        plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
        plt.grid(True, alpha=0.3)
        plt.tight_layout()
        plt.show()

    def plot_target_comparison(self):
        """
        Create bar plots comparing R² and MSE scores across targets and models.

        Generates two subplots:
        1. R² scores comparison
        2. MSE scores comparison
        """
        fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(15, 12))

        x = np.arange(len(self.target_names))
        width = 0.25  # Width of bars

        # Plot bars for each model
        for i, (model_name, model_data) in enumerate(self.models_results.items()):
            # Get final epoch scores for each target
            r2_scores = [model_data['results'][target][-1]['r2'] for target in self.target_names]
            mse_scores = [model_data['results'][target][-1]['mse'] for target in self.target_names]

            # Create grouped bars
            ax1.bar(x + i*width, r2_scores, width, label=model_name)
            ax2.bar(x + i*width, mse_scores, width, label=model_name)

        # Customize both subplots
        for ax in [ax1, ax2]:
            ax.set_xticks(x + width)
            ax.set_xticklabels(self.target_names, rotation=45, ha='right')
            ax.legend()

        ax1.set_ylabel('R² Score')
        ax2.set_ylabel('MSE')
        plt.tight_layout()
        plt.show()

    def plot_performance_heatmap(self):
        """
        Create a heatmap showing final R² scores for all model-target combinations.

        Uses color coding to highlight performance differences:
```

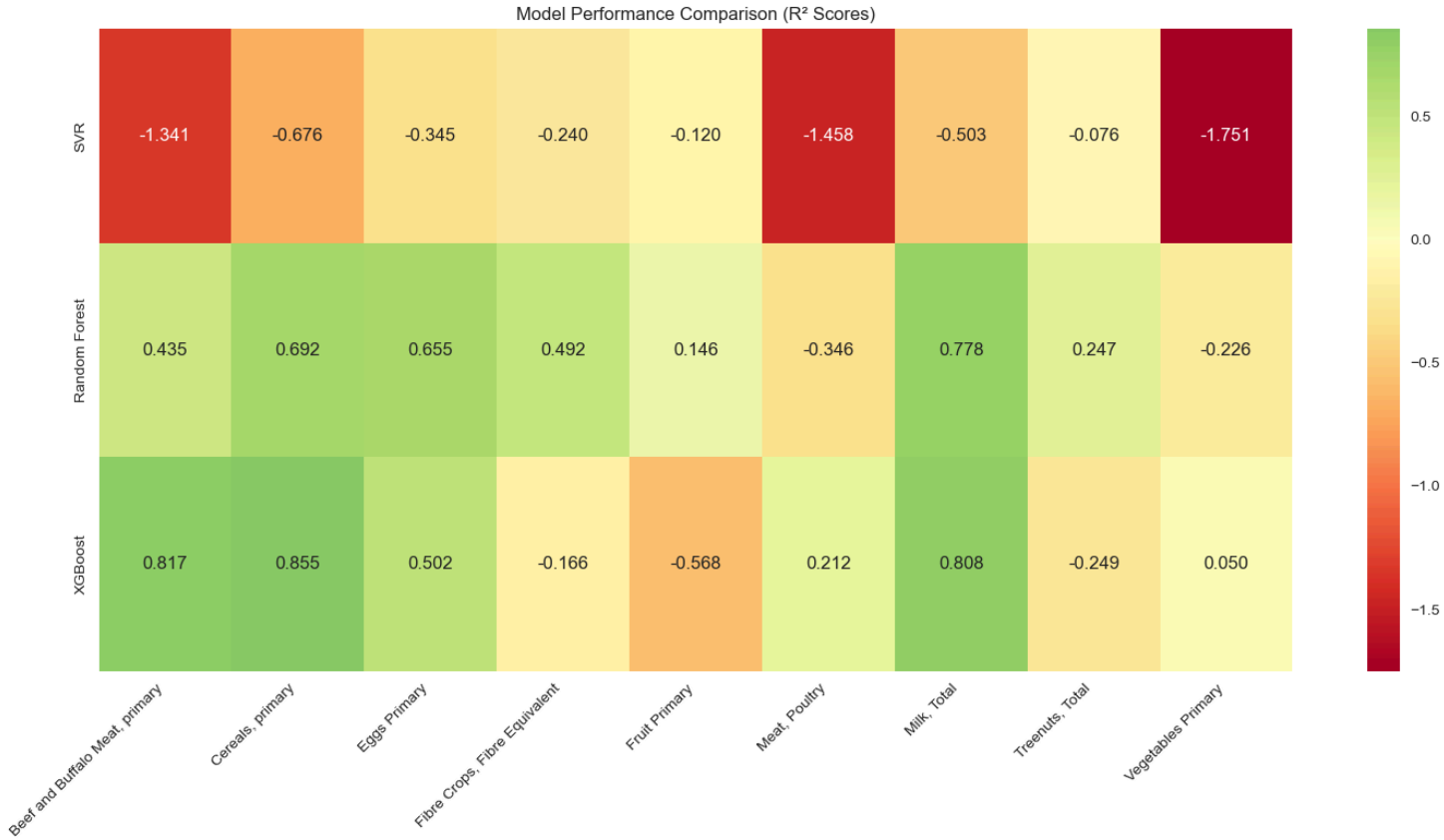
```
        - Green: Higher R² scores
        - Yellow: Medium R² scores
        - Red: Lower R² scores
        """
        # Create matrix of final R² scores
        r2_matrix = np.zeros((len(self.models_results), len(self.target_names)))

        for i, (model_name, model_data) in enumerate(self.models_results.items()):
            for j, target in enumerate(self.target_names):
                r2_matrix[i,j] = model_data['results'][target][-1]['r2']

        # Create and customize heatmap
        plt.figure(figsize=(15, 8))
        sns.heatmap(r2_matrix,
                    xticklabels=self.target_names,
                    yticklabels=list(self.models_results.keys()),
                    annot=True,
                    fmt='.3f',
                    cmap='RdYlGn',
                    center=0)
        plt.title('Model Performance Comparison (R² Scores)')
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()

    def plot_feature_importance(self):
        """
        Create bar plots showing feature importance from Random Forest and XGBoost models.

        Compares feature importance rankings between the two models using horizontal bar charts.
        Only shows results for the first target variable.
        """
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

        # Get models for the first target
        first_target = self.target_names[0]
        rf_model = self.models_results['Random Forest']['models'][first_target]
        xgb_model = self.models_results['XGBoost']['models'][first_target]

        # Remove Total Yield from feature importance analysis
        features = [f for f in self.feature_names if f != 'Total Yield']

        # Plot Random Forest importance
        rf_importance = pd.DataFrame({
            'feature': features,
            'importance': rf_model.feature_importances_[:-1]
        }).sort_values('importance', ascending=True)

        ax1.barh(y=range(len(rf_importance)),
                 width=rf_importance['importance'],
                 tick_label=rf_importance['feature'])
        ax1.set_title('Random Forest Feature Importance')

        # Plot XGBoost importance
        xgb_importance = pd.DataFrame({
            'feature': features,
            'importance': xgb_model.feature_importances_[:-1]
        }).sort_values('importance', ascending=True)

        ax2.barh(y=range(len(xgb_importance)),
                 width=xgb_importance['importance'],
                 tick_label=xgb_importance['feature'])
        ax2.set_title('XGBoost Feature Importance')

        plt.tight_layout()
        plt.show()

# Initialize visualizer with model results
visualizer = ModelVisualizer(
    models_results={
        'SVR': svr_output,
        'Random Forest': rf_output,
        'XGBoost': xgb_output
    },
```

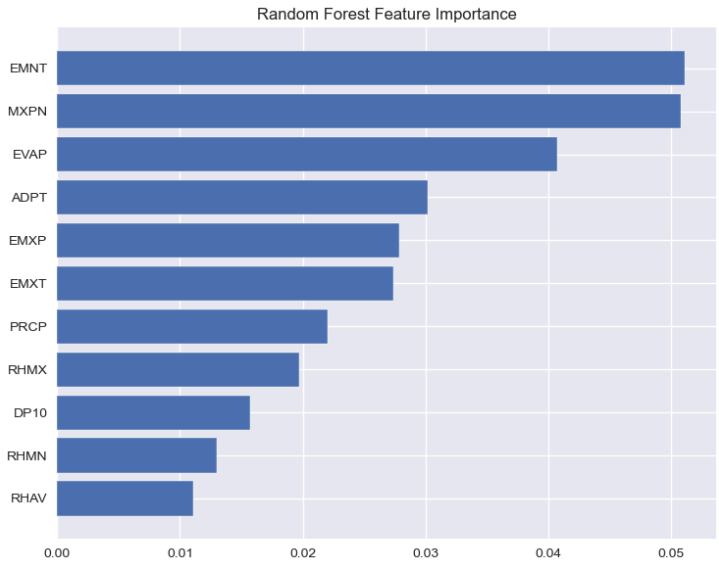
```
    feature_names=trainer.climate_features,
    target_names=trainer.target_columns
)

# Generate all visualization plots
visualizer.plot_training_progress()
visualizer.plot_target_comparison()
visualizer.plot_performance_heatmap()
visualizer.plot_feature_importance()
```

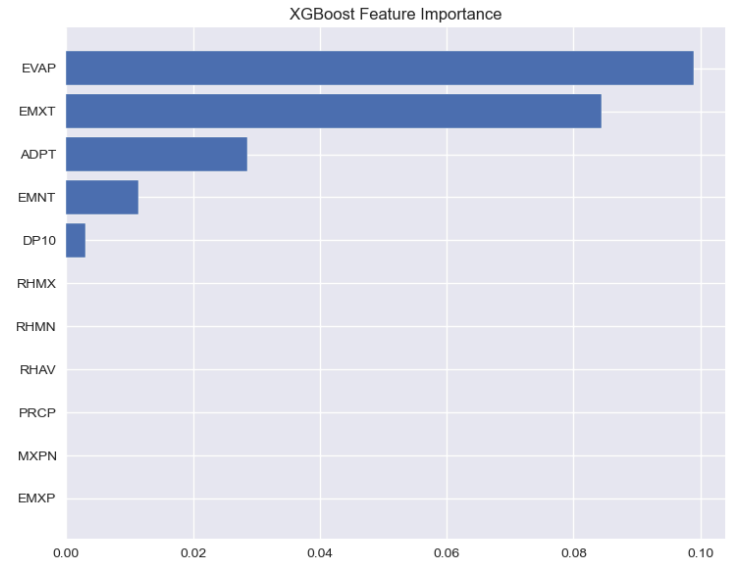



SVR
Random Forest
XGBoost
R² Score
0.5
0.0
−0.5
−1.0
−1.5
Beef and Buffalo Meat, primary
Cereals, primary
Eggs Primary
Fibre Crops, Fibre Equivalent
Fruit Primary
Meat, Poultry
Milk, Total
Treenuts, Total
Vegetables Primary


1e7
SVR
Random Forest
XGBoost
MSE
8
7
6
5
4
3
2
1
0
Beef and Buffalo Meat, primary
Cereals, primary
Eggs Primary
Fibre Crops, Fibre Equivalent
Fruit Primary
Meat, Poultry
Milk, Total
Treenuts, Total
Vegetables Primary


Model Performance Comparison (R² Scores)
SVR
Random Forest
XGBoost
-1.341
-0.676
-0.345
-0.240
-0.120
-1.458
-0.503
-0.076
-1.751
0.435
0.692
0.655
0.492
0.146
-0.346
0.778
0.247
-0.226
0.817
0.855
0.502
-0.166
-0.568
0.212
0.808
-0.249
0.050
0.5
0.0
−0.5
−1.0
−1.5
Beef and Buffalo Meat, primary
Cereals, primary
Eggs Primary
Fibre Crops, Fibre Equivalent
Fruit Primary
Meat, Poultry
Milk, Total
Treenuts, Total
Vegetables Primary





Model Performance Analysis System

Overview

The analysis system reveals significant variations in model performance across different food production categories, with each model showing distinct strengths.

Performance Summary

Overall Model Performance

1. Random Forest: Best average performance (R^2 = 0.3247)
2. SVR: Moderate performance (R^2 = 0.2955)
3. XGBoost: Highest individual scores (R^2 up to 0.8730)

Category-Specific Excellence

- Milk Production: Best predicted category (up to R^2 = 0.8730)
- Cereals: Consistently strong across models (R^2 > 0.65)
- Beef/Buffalo: High variability (R^2 from 0.24 to 0.82)

Areas for Improvement

- Vegetables Primary: Poor predictions across models
- Treenuts: Negative R^2 scores indicate poor fit
- Fruit Primary: Inconsistent performance

Analysis Components

1. Average Performance Calculation

- Computes mean R^2 scores for each model
- Provides overall performance metrics
- Enables quick model comparison
- Accounts for all target variables

2. Target-Specific Analysis

- Identifies best-predicted food categories

- Ranks targets by prediction accuracy
- Shows performance distribution
- Highlights model strengths

3. Performance Reporting

- Clear, formatted output
- Consistent decimal precision
- Aligned text for readability
- Hierarchical information display

Implementation Details

Data Processing

- Extracts final epoch results
- Computes aggregate statistics
- Sorts performance metrics
- Formats output presentation

Performance Metrics

- Uses R^2 score as primary metric
- Considers all target variables
- Provides per-target breakdown
- Shows relative performance

Usage Benefits

- Quick performance assessment
- Easy model comparison
- Clear identification of strengths
- Structured result presentation

Output Format

- Model average performance
- Top 5 performing targets
- Formatted for readability
- Consistent numerical precision

In [14]:
```python
def analyze_model_performance(results):
    """
    Analyze and summarize the performance of multiple machine learning models
    across different target variables.

    This function calculates average R² scores for each model and identifies
    the top 5 best-predicted target variables for each model.

    Args:
        results (dict): A dictionary containing model results with structure:
            {
                'model_name': {
                    'results': {
                        'target_name': [
                            {'r2': float, 'mse': float},
                            ...
                        ]
                    },
                    'models': {...}
                }
```

```
            }

    Prints:
        - Average R² score for each model
        - Top 5 best-predicted targets with their R² scores

    Example:
        >>> analyze_model_performance(results)
        Model Performance Analysis

        SVR Average R²: 0.8532
        Top 5 performing targets:
        Cereals, primary                R² = 0.9234
        Vegetables Primary             R² = 0.8956
        ...
    """
    print("\nModel Performance Analysis")

    # Analyze each model's performance separately
    for model_name, model_data in results.items():
        # Collect R² scores for all targets
        r2_scores = []
        for target in model_data['results'].keys():
            # Get the final epoch's R² score for each target
            r2_scores.append(model_data['results'][target][-1]['r2'])

        # Calculate and display the model's average performance
        avg_r2 = sum(r2_scores) / len(r2_scores)
        print(f"\n{model_name} Average R²: {avg_r2:.4f}")

        # Create dictionary of target:score pairs for sorting
        target_scores = {
            target: model_data['results'][target][-1]['r2']
            for target in model_data['results'].keys()
        }

        # Sort targets by R² score and select top 5
        top_5 = sorted(target_scores.items(),
                       key=lambda x: x[1],  # Sort by R² score
                       reverse=True)[:5]    # Get top 5 in descending order

        # Display the top 5 performing targets
        print("\nTop 5 performing targets:")
        for target, score in top_5:
            # Format output with aligned columns
            print(f"{target:<30} R² = {score:.4f}")

# Execute the performance analysis
analyze_model_performance(results)
```

```
Model Performance Analysis

SVR Average R²: -0.7233

Top 5 performing targets:
Treenuts, Total                R² = -0.0756
Fruit Primary                  R² = -0.1198
Fibre Crops, Fibre Equivalent  R² = -0.2397
Eggs Primary                   R² = -0.3454
Milk, Total                    R² = -0.5027

Random Forest Average R²: 0.3193

Top 5 performing targets:
Milk, Total                    R² = 0.7780
Cereals, primary               R² = 0.6919
Eggs Primary                   R² = 0.6554
Fibre Crops, Fibre Equivalent  R² = 0.4917
Beef and Buffalo Meat, primary R² = 0.4349

XGBoost Average R²: 0.2513

Top 5 performing targets:
Cereals, primary               R² = 0.8553
Beef and Buffalo Meat, primary R² = 0.8168
Milk, Total                    R² = 0.8079
Eggs Primary                   R² = 0.5022
Meat, Poultry                  R² = 0.2120
```

In [ ]:

In [15]:
```python
class ReducedClimateModelTrainer:
    """
    A class for training and evaluating machine learning models with reduced feature set
    to predict food production based on most important climate features.

    Implements three regression models:
    - Support Vector Regression (SVR)
    - Random Forest Regression
    - XGBoost Regression

    Uses only the top performing features identified through feature importance analysis.
    """

    def __init__(self, data):
        """
        Initialize the ReducedClimateModelTrainer with input data.

        Args:
            data (DataFrame): Combined climate and food production data
        """
        # Define reduced set of input features based on importance analysis
        self.climate_features = [
            'MXPN',   # Maximum daily precipitation (top RF feature)
            'EMNT',   # Extreme minimum temperature
            'EVAP',   # Evaporation
            'EMXT',   # Extreme maximum temperature
            'ADPT',   # Average dew point temperature
            'Total Yield'
        ]

        # Keep same target variables for consistency
        self.target_columns = [
            "Beef and Buffalo Meat, primary",
            "Cereals, primary",
            "Eggs Primary",
            "Fibre Crops, Fibre Equivalent",
            "Fruit Primary",
            "Meat, Poultry",
            "Milk, Total",
            "Treenuts, Total",
            "Vegetables Primary"
        ]
```

```
        # Prepare input features and target variables
        self.X = data[self.climate_features].fillna(method='ffill')
        self.y = data[self.target_columns]

        # Split data into training and testing sets
        self.X_train, self.X_test, self.y_train, self.y_test = train_test_split(
            self.X, self.y, test_size=0.2, random_state=42
        )

        # Scale features using StandardScaler
        self.scaler = StandardScaler()
        self.X_train_scaled = self.scaler.fit_transform(self.X_train)
        self.X_test_scaled = self.scaler.transform(self.X_test)

    def train_svr(self, epochs=50):
        """
        Train SVR models with reduced feature set.
        """
        print("\n## Training SVR with Reduced Features")
        svr_results = {}
        svr_models = {}

        for target in self.target_columns:
            print(f"\nTraining SVR for {target}")
            epoch_scores = []
            model = SVR(kernel='rbf', C=0.1)

            for epoch in range(epochs):
                model.C = 0.1 * (epoch + 1)
                model.fit(self.X_train_scaled, self.y_train[target])
                y_pred = model.predict(self.X_test_scaled)

                r2 = r2_score(self.y_test[target], y_pred)
                mse = mean_squared_error(self.y_test[target], y_pred)
                epoch_scores.append({'r2': r2, 'mse': mse})
                print(f"Epoch {epoch + 1}: R² = {r2:.4f}, MSE = {mse:.4f}")

            svr_results[target] = epoch_scores
            svr_models[target] = model
        return {'results': svr_results, 'models': svr_models}

    def train_random_forest(self, epochs=50):
        """
        Train Random Forest models with reduced feature set.
        """
        print("\n## Training Random Forest with Reduced Features")
        rf_results = {}
        rf_models = {}

        for target in self.target_columns:
            print(f"\nTraining Random Forest for {target}")
            epoch_scores = []
            trees_per_epoch = 5

            for epoch in range(epochs):
                new_model = RandomForestRegressor(
                    n_estimators=trees_per_epoch*(epoch+1),
                    random_state=42
                )
                new_model.fit(self.X_train_scaled, self.y_train[target])
                y_pred = new_model.predict(self.X_test_scaled)

                r2 = r2_score(self.y_test[target], y_pred)
                mse = mean_squared_error(self.y_test[target], y_pred)
                epoch_scores.append({'r2': r2, 'mse': mse})
                print(f"Epoch {epoch + 1}: R² = {r2:.4f}, MSE = {mse:.4f}")

            rf_results[target] = epoch_scores
            rf_models[target] = new_model
        return {'results': rf_results, 'models': rf_models}

    def train_xgboost(self, epochs=50):
        """
```

```
        Train XGBoost models with reduced feature set.
        """
        print("\n## Training XGBoost with Reduced Features")
        xgb_results = {}
        xgb_models = {}

        for target in self.target_columns:
            print(f"\nTraining XGBoost for {target}")
            model = XGBRegressor(n_estimators=epochs, learning_rate=0.1)
            epoch_scores = []

            model.fit(
                self.X_train_scaled,
                self.y_train[target],
                eval_set=[(self.X_test_scaled, self.y_test[target])],
                verbose=False
            )

            for i in range(epochs):
                current_model = XGBRegressor(n_estimators=i+1, learning_rate=0.1)
                current_model.fit(self.X_train_scaled, self.y_train[target], verbose=False)
                y_pred = current_model.predict(self.X_test_scaled)

                r2 = r2_score(self.y_test[target], y_pred)
                mse = mean_squared_error(self.y_test[target], y_pred)
                epoch_scores.append({'r2': r2, 'mse': mse})
                print(f"Epoch {i + 1}: R² = {r2:.4f}, MSE = {mse:.4f}")

            xgb_results[target] = epoch_scores
            xgb_models[target] = model

        return {'results': xgb_results, 'models': xgb_models}

# Initialize trainer with reduced feature set
reduced_trainer = ReducedClimateModelTrainer(merged_df)

# Train all models
reduced_svr_output = reduced_trainer.train_svr()
reduced_rf_output = reduced_trainer.train_random_forest()
reduced_xgb_output = reduced_trainer.train_xgboost()

# Collect results for comparison
reduced_results = {
    'SVR': reduced_svr_output,
    'Random Forest': reduced_rf_output,
    'XGBoost': reduced_xgb_output
}

# These results can be analyzed using the existing functions
analyze_model_performance(reduced_results)
visualizer = ModelVisualizer(
    models_results=reduced_results,
    feature_names=reduced_trainer.climate_features,
    target_names=reduced_trainer.target_columns
)
```

Training SVR with Reduced Features

Training SVR for Beef and Buffalo Meat, primary
Epoch 1: R² = -1.4186, MSE = 67067.3844
Epoch 2: R² = -1.4162, MSE = 67002.3071
Epoch 3: R² = -1.4139, MSE = 66937.2679
Epoch 4: R² = -1.4115, MSE = 66872.2670
Epoch 5: R² = -1.4092, MSE = 66807.3044
Epoch 6: R² = -1.4068, MSE = 66742.3799
Epoch 7: R² = -1.4045, MSE = 66677.4937
Epoch 8: R² = -1.4022, MSE = 66612.6457
Epoch 9: R² = -1.3998, MSE = 66547.8360
Epoch 10: R² = -1.3975, MSE = 66483.0644
Epoch 11: R² = -1.3952, MSE = 66418.3312
Epoch 12: R² = -1.3928, MSE = 66353.6361
Epoch 13: R² = -1.3905, MSE = 66288.9793
Epoch 14: R² = -1.3882, MSE = 66224.3607
Epoch 15: R² = -1.3858, MSE = 66159.7803
Epoch 16: R² = -1.3835, MSE = 66095.2381
Epoch 17: R² = -1.3812, MSE = 66030.7342
Epoch 18: R² = -1.3789, MSE = 65966.2685
Epoch 19: R² = -1.3765, MSE = 65901.8411
Epoch 20: R² = -1.3742, MSE = 65837.4519
Epoch 21: R² = -1.3719, MSE = 65773.1009
Epoch 22: R² = -1.3696, MSE = 65708.7881
Epoch 23: R² = -1.3673, MSE = 65644.5136
Epoch 24: R² = -1.3649, MSE = 65580.2772
Epoch 25: R² = -1.3626, MSE = 65516.0792
Epoch 26: R² = -1.3603, MSE = 65451.9193
Epoch 27: R² = -1.3580, MSE = 65387.7977
Epoch 28: R² = -1.3557, MSE = 65323.7143
Epoch 29: R² = -1.3534, MSE = 65259.6691
Epoch 30: R² = -1.3511, MSE = 65195.6622
Epoch 31: R² = -1.3488, MSE = 65131.6935
Epoch 32: R² = -1.3465, MSE = 65067.7630
Epoch 33: R² = -1.3442, MSE = 65003.8708
Epoch 34: R² = -1.3418, MSE = 64940.0168
Epoch 35: R² = -1.3395, MSE = 64876.2010
Epoch 36: R² = -1.3372, MSE = 64812.4235
Epoch 37: R² = -1.3349, MSE = 64748.6841
Epoch 38: R² = -1.3327, MSE = 64684.9830
Epoch 39: R² = -1.3304, MSE = 64621.3202
Epoch 40: R² = -1.3281, MSE = 64557.6955
Epoch 41: R² = -1.3258, MSE = 64494.1091
Epoch 42: R² = -1.3235, MSE = 64430.5610
Epoch 43: R² = -1.3212, MSE = 64367.0510
Epoch 44: R² = -1.3189, MSE = 64303.5793
Epoch 45: R² = -1.3166, MSE = 64240.1458
Epoch 46: R² = -1.3143, MSE = 64176.7506
Epoch 47: R² = -1.3120, MSE = 64113.3935
Epoch 48: R² = -1.3098, MSE = 64050.0747
Epoch 49: R² = -1.3075, MSE = 63986.7942
Epoch 50: R² = -1.3052, MSE = 63923.5518

Training SVR for Cereals, primary
Epoch 1: R² = -0.6981, MSE = 869018.1739
Epoch 2: R² = -0.6974, MSE = 868693.1796
Epoch 3: R² = -0.6968, MSE = 868368.2548
Epoch 4: R² = -0.6962, MSE = 868043.3992
Epoch 5: R² = -0.6955, MSE = 867718.6131
Epoch 6: R² = -0.6949, MSE = 867393.8963
Epoch 7: R² = -0.6943, MSE = 867069.2489
Epoch 8: R² = -0.6936, MSE = 866744.6709
Epoch 9: R² = -0.6930, MSE = 866420.1622
Epoch 10: R² = -0.6924, MSE = 866095.7229
Epoch 11: R² = -0.6917, MSE = 865771.3530
Epoch 12: R² = -0.6911, MSE = 865447.0524
Epoch 13: R² = -0.6905, MSE = 865122.8213
Epoch 14: R² = -0.6898, MSE = 864798.6594
Epoch 15: R² = -0.6892, MSE = 864474.5670
Epoch 16: R² = -0.6886, MSE = 864150.5439
Epoch 17: R² = -0.6879, MSE = 863826.5902
Epoch 18: R² = -0.6873, MSE = 863502.7059
Epoch 19: R² = -0.6867, MSE = 863178.8909

```
Epoch 20: R² = -0.6860, MSE = 862855.1453
Epoch 21: R² = -0.6854, MSE = 862531.4691
Epoch 22: R² = -0.6848, MSE = 862207.8622
Epoch 23: R² = -0.6841, MSE = 861884.3247
Epoch 24: R² = -0.6835, MSE = 861560.8566
Epoch 25: R² = -0.6829, MSE = 861237.4578
Epoch 26: R² = -0.6822, MSE = 860914.1284
Epoch 27: R² = -0.6816, MSE = 860590.8684
Epoch 28: R² = -0.6810, MSE = 860267.6778
Epoch 29: R² = -0.6803, MSE = 859944.5565
Epoch 30: R² = -0.6797, MSE = 859621.5046
Epoch 31: R² = -0.6791, MSE = 859298.5220
Epoch 32: R² = -0.6784, MSE = 858975.6089
Epoch 33: R² = -0.6778, MSE = 858652.7651
Epoch 34: R² = -0.6772, MSE = 858329.9906
Epoch 35: R² = -0.6766, MSE = 858007.2856
Epoch 36: R² = -0.6759, MSE = 857684.6499
Epoch 37: R² = -0.6753, MSE = 857362.0835
Epoch 38: R² = -0.6747, MSE = 857039.5866
Epoch 39: R² = -0.6740, MSE = 856717.1590
Epoch 40: R² = -0.6734, MSE = 856394.8008
Epoch 41: R² = -0.6728, MSE = 856072.5119
Epoch 42: R² = -0.6721, MSE = 855750.2925
Epoch 43: R² = -0.6715, MSE = 855428.1423
Epoch 44: R² = -0.6709, MSE = 855106.0616
Epoch 45: R² = -0.6703, MSE = 854784.0502
Epoch 46: R² = -0.6696, MSE = 854462.1082
Epoch 47: R² = -0.6690, MSE = 854140.2356
Epoch 48: R² = -0.6684, MSE = 853818.4323
Epoch 49: R² = -0.6677, MSE = 853496.6984
Epoch 50: R² = -0.6671, MSE = 853175.0339

Training SVR for Eggs Primary
Epoch 1: R² = -0.3477, MSE = 44894236.0820
Epoch 2: R² = -0.3476, MSE = 44892030.7334
Epoch 3: R² = -0.3475, MSE = 44889825.4541
Epoch 4: R² = -0.3475, MSE = 44887620.2442
Epoch 5: R² = -0.3474, MSE = 44885415.1037
Epoch 6: R² = -0.3473, MSE = 44883210.0325
Epoch 7: R² = -0.3473, MSE = 44881005.0308
Epoch 8: R² = -0.3472, MSE = 44878800.0984
Epoch 9: R² = -0.3471, MSE = 44876595.2353
Epoch 10: R² = -0.3471, MSE = 44874390.4416
Epoch 11: R² = -0.3470, MSE = 44872185.7173
Epoch 12: R² = -0.3469, MSE = 44869981.0624
Epoch 13: R² = -0.3469, MSE = 44867776.4768
Epoch 14: R² = -0.3468, MSE = 44865571.9606
Epoch 15: R² = -0.3467, MSE = 44863367.5138
Epoch 16: R² = -0.3467, MSE = 44861163.1363
Epoch 17: R² = -0.3466, MSE = 44858958.8283
Epoch 18: R² = -0.3465, MSE = 44856754.5895
Epoch 19: R² = -0.3465, MSE = 44854550.4202
Epoch 20: R² = -0.3464, MSE = 44852346.3202
Epoch 21: R² = -0.3463, MSE = 44850142.2896
Epoch 22: R² = -0.3463, MSE = 44847938.3283
Epoch 23: R² = -0.3462, MSE = 44845734.4365
Epoch 24: R² = -0.3461, MSE = 44843530.6140
Epoch 25: R² = -0.3461, MSE = 44841326.8608
Epoch 26: R² = -0.3460, MSE = 44839123.1771
Epoch 27: R² = -0.3459, MSE = 44836919.5627
Epoch 28: R² = -0.3459, MSE = 44834716.0176
Epoch 29: R² = -0.3458, MSE = 44832512.5420
Epoch 30: R² = -0.3457, MSE = 44830309.1357
Epoch 31: R² = -0.3457, MSE = 44828105.7988
Epoch 32: R² = -0.3456, MSE = 44825902.5312
Epoch 33: R² = -0.3455, MSE = 44823699.3330
Epoch 34: R² = -0.3455, MSE = 44821496.2042
Epoch 35: R² = -0.3454, MSE = 44819293.1448
Epoch 36: R² = -0.3453, MSE = 44817090.1547
Epoch 37: R² = -0.3453, MSE = 44814887.2340
Epoch 38: R² = -0.3452, MSE = 44812684.3827
Epoch 39: R² = -0.3451, MSE = 44810481.6007
Epoch 40: R² = -0.3451, MSE = 44808278.8881
Epoch 41: R² = -0.3450, MSE = 44806076.2449
```

```
Epoch 42: R² = -0.3450, MSE = 44803873.6710
Epoch 43: R² = -0.3449, MSE = 44801671.1665
Epoch 44: R² = -0.3448, MSE = 44799468.7314
Epoch 45: R² = -0.3448, MSE = 44797266.3657
Epoch 46: R² = -0.3447, MSE = 44795064.0693
Epoch 47: R² = -0.3446, MSE = 44792861.8423
Epoch 48: R² = -0.3446, MSE = 44790659.6846
Epoch 49: R² = -0.3445, MSE = 44788457.5964
Epoch 50: R² = -0.3444, MSE = 44786255.5775

Training SVR for Fibre Crops, Fibre Equivalent
Epoch 1: R² = -0.3956, MSE = 6524.5163
Epoch 2: R² = -0.3902, MSE = 6499.2964
Epoch 3: R² = -0.3849, MSE = 6474.1454
Epoch 4: R² = -0.3795, MSE = 6449.0631
Epoch 5: R² = -0.3742, MSE = 6424.0497
Epoch 6: R² = -0.3688, MSE = 6399.1052
Epoch 7: R² = -0.3635, MSE = 6374.2294
Epoch 8: R² = -0.3582, MSE = 6349.4225
Epoch 9: R² = -0.3529, MSE = 6324.6845
Epoch 10: R² = -0.3476, MSE = 6300.0152
Epoch 11: R² = -0.3424, MSE = 6275.4148
Epoch 12: R² = -0.3371, MSE = 6250.8832
Epoch 13: R² = -0.3319, MSE = 6226.4204
Epoch 14: R² = -0.3267, MSE = 6202.0265
Epoch 15: R² = -0.3215, MSE = 6177.7014
Epoch 16: R² = -0.3163, MSE = 6153.4451
Epoch 17: R² = -0.3111, MSE = 6129.2577
Epoch 18: R² = -0.3059, MSE = 6105.1391
Epoch 19: R² = -0.3008, MSE = 6081.0893
Epoch 20: R² = -0.2957, MSE = 6057.1083
Epoch 21: R² = -0.2905, MSE = 6033.1962
Epoch 22: R² = -0.2854, MSE = 6009.3529
Epoch 23: R² = -0.2804, MSE = 5985.5785
Epoch 24: R² = -0.2753, MSE = 5961.8728
Epoch 25: R² = -0.2702, MSE = 5938.2360
Epoch 26: R² = -0.2652, MSE = 5914.6680
Epoch 27: R² = -0.2602, MSE = 5891.1689
Epoch 28: R² = -0.2552, MSE = 5867.7386
Epoch 29: R² = -0.2502, MSE = 5844.3771
Epoch 30: R² = -0.2452, MSE = 5821.0844
Epoch 31: R² = -0.2402, MSE = 5797.8606
Epoch 32: R² = -0.2353, MSE = 5774.7056
Epoch 33: R² = -0.2303, MSE = 5751.6194
Epoch 34: R² = -0.2254, MSE = 5728.6021
Epoch 35: R² = -0.2205, MSE = 5705.6535
Epoch 36: R² = -0.2156, MSE = 5682.7739
Epoch 37: R² = -0.2107, MSE = 5659.9630
Epoch 38: R² = -0.2058, MSE = 5637.2210
Epoch 39: R² = -0.2010, MSE = 5614.5478
Epoch 40: R² = -0.1962, MSE = 5591.9434
Epoch 41: R² = -0.1913, MSE = 5569.4079
Epoch 42: R² = -0.1865, MSE = 5546.9412
Epoch 43: R² = -0.1817, MSE = 5524.5433
Epoch 44: R² = -0.1770, MSE = 5502.2142
Epoch 45: R² = -0.1722, MSE = 5479.9540
Epoch 46: R² = -0.1675, MSE = 5457.7626
Epoch 47: R² = -0.1627, MSE = 5435.6401
Epoch 48: R² = -0.1580, MSE = 5413.5863
Epoch 49: R² = -0.1533, MSE = 5391.6014
Epoch 50: R² = -0.1486, MSE = 5369.6853

Training SVR for Fruit Primary
Epoch 1: R² = -0.1213, MSE = 1610190.8596
Epoch 2: R² = -0.1213, MSE = 1610225.2058
Epoch 3: R² = -0.1213, MSE = 1610259.5936
Epoch 4: R² = -0.1214, MSE = 1610294.0230
Epoch 5: R² = -0.1214, MSE = 1610328.4940
Epoch 6: R² = -0.1214, MSE = 1610363.0065
Epoch 7: R² = -0.1214, MSE = 1610397.5607
Epoch 8: R² = -0.1215, MSE = 1610432.1564
Epoch 9: R² = -0.1215, MSE = 1610466.7937
Epoch 10: R² = -0.1215, MSE = 1610501.4727
Epoch 11: R² = -0.1215, MSE = 1610536.1931
```

```
Epoch 12: R² = -0.1216, MSE = 1610570.9552
Epoch 13: R² = -0.1216, MSE = 1610605.7589
Epoch 14: R² = -0.1216, MSE = 1610640.6041
Epoch 15: R² = -0.1216, MSE = 1610675.4910
Epoch 16: R² = -0.1217, MSE = 1610710.4194
Epoch 17: R² = -0.1217, MSE = 1610745.3894
Epoch 18: R² = -0.1217, MSE = 1610780.4010
Epoch 19: R² = -0.1217, MSE = 1610815.4542
Epoch 20: R² = -0.1217, MSE = 1610850.5490
Epoch 21: R² = -0.1218, MSE = 1610885.6854
Epoch 22: R² = -0.1218, MSE = 1610920.8633
Epoch 23: R² = -0.1218, MSE = 1610956.0829
Epoch 24: R² = -0.1218, MSE = 1610991.3440
Epoch 25: R² = -0.1219, MSE = 1611026.6467
Epoch 26: R² = -0.1219, MSE = 1611061.9910
Epoch 27: R² = -0.1219, MSE = 1611097.3769
Epoch 28: R² = -0.1219, MSE = 1611132.8043
Epoch 29: R² = -0.1220, MSE = 1611168.2734
Epoch 30: R² = -0.1220, MSE = 1611203.7841
Epoch 31: R² = -0.1220, MSE = 1611239.3363
Epoch 32: R² = -0.1220, MSE = 1611274.9301
Epoch 33: R² = -0.1221, MSE = 1611310.5655
Epoch 34: R² = -0.1221, MSE = 1611346.2425
Epoch 35: R² = -0.1221, MSE = 1611381.9611
Epoch 36: R² = -0.1221, MSE = 1611417.7213
Epoch 37: R² = -0.1222, MSE = 1611453.5230
Epoch 38: R² = -0.1222, MSE = 1611489.3664
Epoch 39: R² = -0.1222, MSE = 1611525.2513
Epoch 40: R² = -0.1222, MSE = 1611561.1778
Epoch 41: R² = -0.1223, MSE = 1611597.1459
Epoch 42: R² = -0.1223, MSE = 1611633.1556
Epoch 43: R² = -0.1223, MSE = 1611669.2069
Epoch 44: R² = -0.1223, MSE = 1611705.2997
Epoch 45: R² = -0.1224, MSE = 1611741.4342
Epoch 46: R² = -0.1224, MSE = 1611777.6102
Epoch 47: R² = -0.1224, MSE = 1611813.8279
Epoch 48: R² = -0.1224, MSE = 1611850.0871
Epoch 49: R² = -0.1225, MSE = 1611886.3879
Epoch 50: R² = -0.1225, MSE = 1611922.7303

Training SVR for Meat, Poultry
Epoch 1: R² = -1.4657, MSE = 1845331.6770
Epoch 2: R² = -1.4653, MSE = 1845005.8870
Epoch 3: R² = -1.4648, MSE = 1844680.1300
Epoch 4: R² = -1.4644, MSE = 1844354.4059
Epoch 5: R² = -1.4640, MSE = 1844028.7148
Epoch 6: R² = -1.4635, MSE = 1843703.0567
Epoch 7: R² = -1.4631, MSE = 1843377.4316
Epoch 8: R² = -1.4627, MSE = 1843051.8395
Epoch 9: R² = -1.4622, MSE = 1842726.2803
Epoch 10: R² = -1.4618, MSE = 1842400.7541
Epoch 11: R² = -1.4613, MSE = 1842075.2609
Epoch 12: R² = -1.4609, MSE = 1841749.8007
Epoch 13: R² = -1.4605, MSE = 1841424.3735
Epoch 14: R² = -1.4600, MSE = 1841098.9792
Epoch 15: R² = -1.4596, MSE = 1840773.6179
Epoch 16: R² = -1.4592, MSE = 1840448.2896
Epoch 17: R² = -1.4587, MSE = 1840122.9943
Epoch 18: R² = -1.4583, MSE = 1839797.7319
Epoch 19: R² = -1.4579, MSE = 1839472.5025
Epoch 20: R² = -1.4574, MSE = 1839147.3061
Epoch 21: R² = -1.4570, MSE = 1838822.1427
Epoch 22: R² = -1.4566, MSE = 1838497.0123
Epoch 23: R² = -1.4561, MSE = 1838171.9148
Epoch 24: R² = -1.4557, MSE = 1837846.8504
Epoch 25: R² = -1.4553, MSE = 1837521.8189
Epoch 26: R² = -1.4548, MSE = 1837196.8203
Epoch 27: R² = -1.4544, MSE = 1836871.8548
Epoch 28: R² = -1.4540, MSE = 1836546.9222
Epoch 29: R² = -1.4535, MSE = 1836222.0227
Epoch 30: R² = -1.4531, MSE = 1835897.1561
Epoch 31: R² = -1.4527, MSE = 1835572.3224
Epoch 32: R² = -1.4522, MSE = 1835247.5218
Epoch 33: R² = -1.4518, MSE = 1834922.7541
```

```
Epoch 34: R² = -1.4514, MSE = 1834598.0194
Epoch 35: R² = -1.4509, MSE = 1834273.3177
Epoch 36: R² = -1.4505, MSE = 1833948.6490
Epoch 37: R² = -1.4501, MSE = 1833624.0132
Epoch 38: R² = -1.4496, MSE = 1833299.4105
```

```
C:\Users\kaibe\AppData\Local\Temp\ipykernel_20932\2426265218.py:45: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
  self.X = data[self.climate_features].fillna(method='ffill')
```

```
Epoch 39: R² = -1.4492, MSE = 1832974.8407
Epoch 40: R² = -1.4488, MSE = 1832650.3038
Epoch 41: R² = -1.4483, MSE = 1832325.8000
Epoch 42: R² = -1.4479, MSE = 1832001.3292
Epoch 43: R² = -1.4475, MSE = 1831676.8913
Epoch 44: R² = -1.4470, MSE = 1831352.4864
Epoch 45: R² = -1.4466, MSE = 1831028.1145
Epoch 46: R² = -1.4462, MSE = 1830703.7755
Epoch 47: R² = -1.4457, MSE = 1830379.4695
Epoch 48: R² = -1.4453, MSE = 1830055.1966
Epoch 49: R² = -1.4449, MSE = 1829730.9566
Epoch 50: R² = -1.4444, MSE = 1829406.7495

Training SVR for Milk, Total
Epoch 1: R² = -0.5047, MSE = 79579557.4148
Epoch 2: R² = -0.5046, MSE = 79577717.3557
Epoch 3: R² = -0.5046, MSE = 79575877.3227
Epoch 4: R² = -0.5046, MSE = 79574037.3158
Epoch 5: R² = -0.5045, MSE = 79572197.3349
Epoch 6: R² = -0.5045, MSE = 79570357.3801
Epoch 7: R² = -0.5045, MSE = 79568517.4514
Epoch 8: R² = -0.5044, MSE = 79566677.5487
Epoch 9: R² = -0.5044, MSE = 79564837.6722
Epoch 10: R² = -0.5043, MSE = 79562997.8217
Epoch 11: R² = -0.5043, MSE = 79561157.9972
Epoch 12: R² = -0.5043, MSE = 79559318.1989
Epoch 13: R² = -0.5042, MSE = 79557478.4266
Epoch 14: R² = -0.5042, MSE = 79555638.6803
Epoch 15: R² = -0.5042, MSE = 79553798.9602
Epoch 16: R² = -0.5041, MSE = 79551959.2661
Epoch 17: R² = -0.5041, MSE = 79550119.5981
Epoch 18: R² = -0.5041, MSE = 79548279.9562
Epoch 19: R² = -0.5040, MSE = 79546440.3403
Epoch 20: R² = -0.5040, MSE = 79544600.7506
Epoch 21: R² = -0.5040, MSE = 79542761.1868
Epoch 22: R² = -0.5039, MSE = 79540921.6492
Epoch 23: R² = -0.5039, MSE = 79539082.1376
Epoch 24: R² = -0.5039, MSE = 79537242.6522
Epoch 25: R² = -0.5038, MSE = 79535403.1927
Epoch 26: R² = -0.5038, MSE = 79533563.7594
Epoch 27: R² = -0.5038, MSE = 79531724.3521
Epoch 28: R² = -0.5037, MSE = 79529884.9709
Epoch 29: R² = -0.5037, MSE = 79528045.6158
Epoch 30: R² = -0.5037, MSE = 79526206.2867
Epoch 31: R² = -0.5036, MSE = 79524366.9837
Epoch 32: R² = -0.5036, MSE = 79522527.7068
Epoch 33: R² = -0.5035, MSE = 79520688.4560
Epoch 34: R² = -0.5035, MSE = 79518849.2312
Epoch 35: R² = -0.5035, MSE = 79517010.0325
Epoch 36: R² = -0.5034, MSE = 79515170.8599
Epoch 37: R² = -0.5034, MSE = 79513331.7134
Epoch 38: R² = -0.5034, MSE = 79511492.5929
Epoch 39: R² = -0.5033, MSE = 79509653.4985
Epoch 40: R² = -0.5033, MSE = 79507814.4302
Epoch 41: R² = -0.5033, MSE = 79505975.3879
Epoch 42: R² = -0.5032, MSE = 79504136.3717
Epoch 43: R² = -0.5032, MSE = 79502297.3816
Epoch 44: R² = -0.5032, MSE = 79500458.4176
Epoch 45: R² = -0.5031, MSE = 79498619.4796
Epoch 46: R² = -0.5031, MSE = 79496780.5677
Epoch 47: R² = -0.5031, MSE = 79494941.6819
Epoch 48: R² = -0.5030, MSE = 79493102.8221
Epoch 49: R² = -0.5030, MSE = 79491263.9884
Epoch 50: R² = -0.5030, MSE = 79489425.1808

Training SVR for Treenuts, Total
Epoch 1: R² = -0.0822, MSE = 297460.3073
Epoch 2: R² = -0.0822, MSE = 297460.5127
Epoch 3: R² = -0.0822, MSE = 297460.7512
Epoch 4: R² = -0.0822, MSE = 297461.0228
Epoch 5: R² = -0.0822, MSE = 297461.3276
Epoch 6: R² = -0.0822, MSE = 297461.6654
Epoch 7: R² = -0.0822, MSE = 297462.0364
Epoch 8: R² = -0.0822, MSE = 297462.4405
```

```
Epoch 9: R² = -0.0822, MSE = 297462.8777
Epoch 10: R² = -0.0822, MSE = 297463.3480
Epoch 11: R² = -0.0822, MSE = 297463.8514
Epoch 12: R² = -0.0822, MSE = 297464.3880
Epoch 13: R² = -0.0822, MSE = 297464.9576
Epoch 14: R² = -0.0822, MSE = 297465.5604
Epoch 15: R² = -0.0822, MSE = 297466.1962
Epoch 16: R² = -0.0822, MSE = 297466.8652
Epoch 17: R² = -0.0822, MSE = 297467.5673
Epoch 18: R² = -0.0822, MSE = 297468.3025
Epoch 19: R² = -0.0822, MSE = 297469.0709
Epoch 20: R² = -0.0822, MSE = 297469.8723
Epoch 21: R² = -0.0822, MSE = 297470.7069
Epoch 22: R² = -0.0822, MSE = 297471.5745
Epoch 23: R² = -0.0822, MSE = 297472.4753
Epoch 24: R² = -0.0822, MSE = 297473.4092
Epoch 25: R² = -0.0822, MSE = 297474.3762
Epoch 26: R² = -0.0822, MSE = 297475.3763
Epoch 27: R² = -0.0822, MSE = 297476.4096
Epoch 28: R² = -0.0822, MSE = 297477.4759
Epoch 29: R² = -0.0823, MSE = 297478.5754
Epoch 30: R² = -0.0823, MSE = 297479.7079
Epoch 31: R² = -0.0823, MSE = 297480.8736
Epoch 32: R² = -0.0823, MSE = 297482.0724
Epoch 33: R² = -0.0823, MSE = 297483.3043
Epoch 34: R² = -0.0823, MSE = 297484.5694
Epoch 35: R² = -0.0823, MSE = 297485.8675
Epoch 36: R² = -0.0823, MSE = 297487.1987
Epoch 37: R² = -0.0823, MSE = 297488.5631
Epoch 38: R² = -0.0823, MSE = 297489.9606
Epoch 39: R² = -0.0823, MSE = 297491.3912
Epoch 40: R² = -0.0823, MSE = 297492.8549
Epoch 41: R² = -0.0823, MSE = 297494.3517
Epoch 42: R² = -0.0823, MSE = 297495.8816
Epoch 43: R² = -0.0823, MSE = 297497.4447
Epoch 44: R² = -0.0823, MSE = 297499.0408
Epoch 45: R² = -0.0823, MSE = 297500.6701
Epoch 46: R² = -0.0823, MSE = 297502.3325
Epoch 47: R² = -0.0823, MSE = 297504.0280
Epoch 48: R² = -0.0824, MSE = 297505.7566
Epoch 49: R² = -0.0824, MSE = 297507.5183
Epoch 50: R² = -0.0824, MSE = 297509.3132

Training SVR for Vegetables Primary
Epoch 1: R² = -1.7616, MSE = 5792587.7600
Epoch 2: R² = -1.7612, MSE = 5791643.7477
Epoch 3: R² = -1.7607, MSE = 5790699.8282
Epoch 4: R² = -1.7603, MSE = 5789756.0015
Epoch 5: R² = -1.7598, MSE = 5788812.2675
Epoch 6: R² = -1.7594, MSE = 5787868.6263
Epoch 7: R² = -1.7589, MSE = 5786925.0779
Epoch 8: R² = -1.7585, MSE = 5785981.6223
Epoch 9: R² = -1.7580, MSE = 5785038.2594
Epoch 10: R² = -1.7576, MSE = 5784094.9893
Epoch 11: R² = -1.7571, MSE = 5783151.8119
Epoch 12: R² = -1.7567, MSE = 5782208.7273
Epoch 13: R² = -1.7562, MSE = 5781265.7355
Epoch 14: R² = -1.7558, MSE = 5780322.8365
Epoch 15: R² = -1.7553, MSE = 5779380.0302
Epoch 16: R² = -1.7549, MSE = 5778437.3167
Epoch 17: R² = -1.7544, MSE = 5777494.6959
Epoch 18: R² = -1.7540, MSE = 5776552.1680
Epoch 19: R² = -1.7535, MSE = 5775609.7328
Epoch 20: R² = -1.7531, MSE = 5774667.3903
Epoch 21: R² = -1.7526, MSE = 5773725.1407
Epoch 22: R² = -1.7522, MSE = 5772782.9838
Epoch 23: R² = -1.7517, MSE = 5771840.9196
Epoch 24: R² = -1.7513, MSE = 5770898.9483
Epoch 25: R² = -1.7508, MSE = 5769957.0697
Epoch 26: R² = -1.7504, MSE = 5769015.2838
Epoch 27: R² = -1.7499, MSE = 5768073.5908
Epoch 28: R² = -1.7495, MSE = 5767131.9905
Epoch 29: R² = -1.7490, MSE = 5766190.4830
Epoch 30: R² = -1.7486, MSE = 5765249.0682
```

```
Epoch 31: R² = -1.7481, MSE = 5764307.7462
Epoch 32: R² = -1.7477, MSE = 5763366.5170
Epoch 33: R² = -1.7472, MSE = 5762425.3805
Epoch 34: R² = -1.7468, MSE = 5761484.3369
Epoch 35: R² = -1.7464, MSE = 5760543.3859
Epoch 36: R² = -1.7459, MSE = 5759602.5278
Epoch 37: R² = -1.7455, MSE = 5758661.7624
Epoch 38: R² = -1.7450, MSE = 5757721.0898
Epoch 39: R² = -1.7446, MSE = 5756780.5100
Epoch 40: R² = -1.7441, MSE = 5755840.0229
Epoch 41: R² = -1.7437, MSE = 5754899.6286
Epoch 42: R² = -1.7432, MSE = 5753959.3270
Epoch 43: R² = -1.7428, MSE = 5753019.1183
Epoch 44: R² = -1.7423, MSE = 5752079.0023
Epoch 45: R² = -1.7419, MSE = 5751138.9790
Epoch 46: R² = -1.7414, MSE = 5750199.0486
Epoch 47: R² = -1.7410, MSE = 5749259.2109
Epoch 48: R² = -1.7405, MSE = 5748319.4659
Epoch 49: R² = -1.7401, MSE = 5747379.8138
Epoch 50: R² = -1.7396, MSE = 5746440.2544

## Training Random Forest with Reduced Features

Training Random Forest for Beef and Buffalo Meat, primary
Epoch 1: R² = 0.5647, MSE = 12070.4800
Epoch 2: R² = 0.2452, MSE = 20931.9425
Epoch 3: R² = 0.4073, MSE = 16435.5622
Epoch 4: R² = 0.3932, MSE = 16827.7325
Epoch 5: R² = 0.4128, MSE = 16282.0088
Epoch 6: R² = 0.4412, MSE = 15496.6342
Epoch 7: R² = 0.4843, MSE = 14300.1908
Epoch 8: R² = 0.4354, MSE = 15655.3661
Epoch 9: R² = 0.4519, MSE = 15197.7756
Epoch 10: R² = 0.4588, MSE = 15006.2860
Epoch 11: R² = 0.4334, MSE = 15711.6999
Epoch 12: R² = 0.4302, MSE = 15800.4591
Epoch 13: R² = 0.4410, MSE = 15501.2437
Epoch 14: R² = 0.4528, MSE = 15173.8480
Epoch 15: R² = 0.4631, MSE = 14887.1375
Epoch 16: R² = 0.4536, MSE = 15152.8049
Epoch 17: R² = 0.4653, MSE = 14827.5876
Epoch 18: R² = 0.4746, MSE = 14570.3092
Epoch 19: R² = 0.4658, MSE = 14813.0741
Epoch 20: R² = 0.4688, MSE = 14731.5958
Epoch 21: R² = 0.4677, MSE = 14761.2909
Epoch 22: R² = 0.4528, MSE = 15175.1082
Epoch 23: R² = 0.4468, MSE = 15340.2222
Epoch 24: R² = 0.4426, MSE = 15456.7890
Epoch 25: R² = 0.4323, MSE = 15743.8370
Epoch 26: R² = 0.4341, MSE = 15691.1914
Epoch 27: R² = 0.4402, MSE = 15522.2664
Epoch 28: R² = 0.4284, MSE = 15850.4710
Epoch 29: R² = 0.4232, MSE = 15996.1412
Epoch 30: R² = 0.4296, MSE = 15816.2391
Epoch 31: R² = 0.4291, MSE = 15831.6975
Epoch 32: R² = 0.4343, MSE = 15686.6332
Epoch 33: R² = 0.4358, MSE = 15646.6746
Epoch 34: R² = 0.4444, MSE = 15406.7126
Epoch 35: R² = 0.4500, MSE = 15251.3763
Epoch 36: R² = 0.4437, MSE = 15427.0853
Epoch 37: R² = 0.4491, MSE = 15277.8963
Epoch 38: R² = 0.4493, MSE = 15271.8415
Epoch 39: R² = 0.4491, MSE = 15275.7605
Epoch 40: R² = 0.4504, MSE = 15240.7771
Epoch 41: R² = 0.4481, MSE = 15304.4101
Epoch 42: R² = 0.4560, MSE = 15084.7621
Epoch 43: R² = 0.4542, MSE = 15136.0305
Epoch 44: R² = 0.4511, MSE = 15221.9088
Epoch 45: R² = 0.4504, MSE = 15240.2190
Epoch 46: R² = 0.4478, MSE = 15312.3935
Epoch 47: R² = 0.4465, MSE = 15349.5487
Epoch 48: R² = 0.4355, MSE = 15654.6149
Epoch 49: R² = 0.4363, MSE = 15630.4523
Epoch 50: R² = 0.4397, MSE = 15537.7156
```

```
Training Random Forest for Cereals, primary
Epoch 1: R² = 0.5913, MSE = 209172.8073
Epoch 2: R² = 0.4419, MSE = 285640.6740
Epoch 3: R² = 0.6732, MSE = 167254.7472
Epoch 4: R² = 0.6423, MSE = 183053.8614
Epoch 5: R² = 0.6240, MSE = 192446.5112
Epoch 6: R² = 0.6467, MSE = 180828.0609
Epoch 7: R² = 0.6284, MSE = 190190.7102
Epoch 8: R² = 0.6415, MSE = 183465.6373
Epoch 9: R² = 0.6601, MSE = 173942.4419
Epoch 10: R² = 0.6598, MSE = 174107.8951
Epoch 11: R² = 0.6417, MSE = 183347.5266
Epoch 12: R² = 0.6470, MSE = 180657.9791
Epoch 13: R² = 0.6582, MSE = 174916.2602
Epoch 14: R² = 0.6671, MSE = 170366.1378
Epoch 15: R² = 0.6731, MSE = 167276.1786
Epoch 16: R² = 0.6698, MSE = 168978.2854
Epoch 17: R² = 0.6753, MSE = 166173.5652
Epoch 18: R² = 0.6741, MSE = 166795.8226
Epoch 19: R² = 0.6768, MSE = 165412.4887
Epoch 20: R² = 0.6820, MSE = 162730.9342
Epoch 21: R² = 0.6817, MSE = 162871.2577
Epoch 22: R² = 0.6830, MSE = 162252.3231
Epoch 23: R² = 0.6831, MSE = 162156.0584
Epoch 24: R² = 0.6962, MSE = 155483.4559
Epoch 25: R² = 0.6974, MSE = 154836.3230
Epoch 26: R² = 0.6855, MSE = 160950.5146
Epoch 27: R² = 0.6771, MSE = 165235.0773
Epoch 28: R² = 0.6742, MSE = 166759.4680
Epoch 29: R² = 0.6769, MSE = 165377.8778
Epoch 30: R² = 0.6861, MSE = 160642.7423
Epoch 31: R² = 0.6947, MSE = 156253.2491
Epoch 32: R² = 0.6987, MSE = 154176.0445
Epoch 33: R² = 0.6974, MSE = 154848.0818
Epoch 34: R² = 0.7101, MSE = 148374.6537
Epoch 35: R² = 0.7152, MSE = 145775.4924
Epoch 36: R² = 0.7119, MSE = 147419.1232
Epoch 37: R² = 0.7186, MSE = 144035.8046
Epoch 38: R² = 0.7261, MSE = 140197.8651
Epoch 39: R² = 0.7299, MSE = 138231.3943
Epoch 40: R² = 0.7310, MSE = 137678.6064
Epoch 41: R² = 0.7301, MSE = 138137.2784
Epoch 42: R² = 0.7302, MSE = 138087.9426
Epoch 43: R² = 0.7271, MSE = 139642.5092
Epoch 44: R² = 0.7292, MSE = 138587.1961
Epoch 45: R² = 0.7337, MSE = 136291.0123
Epoch 46: R² = 0.7353, MSE = 135475.7790
Epoch 47: R² = 0.7337, MSE = 136294.3079
Epoch 48: R² = 0.7233, MSE = 141581.8558
Epoch 49: R² = 0.7272, MSE = 139593.8278
Epoch 50: R² = 0.7245, MSE = 141000.2246

Training Random Forest for Eggs Primary
Epoch 1: R² = 0.7657, MSE = 7804952.4600
Epoch 2: R² = 0.7012, MSE = 9953086.8025
Epoch 3: R² = 0.7487, MSE = 8372972.2567
Epoch 4: R² = 0.7398, MSE = 8669120.5344
Epoch 5: R² = 0.7537, MSE = 8203422.2424
Epoch 6: R² = 0.7525, MSE = 8245902.2683
Epoch 7: R² = 0.7437, MSE = 8537840.8039
Epoch 8: R² = 0.7221, MSE = 9257456.7522
Epoch 9: R² = 0.7063, MSE = 9783503.4512
Epoch 10: R² = 0.7063, MSE = 9784408.3333
Epoch 11: R² = 0.7223, MSE = 9251797.5671
Epoch 12: R² = 0.7120, MSE = 9594387.9799
Epoch 13: R² = 0.7102, MSE = 9655550.7054
Epoch 14: R² = 0.7044, MSE = 9848122.5905
Epoch 15: R² = 0.7108, MSE = 9635259.1904
Epoch 16: R² = 0.7091, MSE = 9691993.9108
Epoch 17: R² = 0.7187, MSE = 9372086.4373
Epoch 18: R² = 0.7283, MSE = 9052095.8981
Epoch 19: R² = 0.7249, MSE = 9164841.5382
Epoch 20: R² = 0.7268, MSE = 9099432.9341
```

```
Epoch 21: R² = 0.7257, MSE = 9136144.2180
Epoch 22: R² = 0.7239, MSE = 9198708.4983
Epoch 23: R² = 0.7211, MSE = 9291890.3642
Epoch 24: R² = 0.7247, MSE = 9169542.8232
Epoch 25: R² = 0.7220, MSE = 9261053.0280
Epoch 26: R² = 0.7278, MSE = 9066889.2578
Epoch 27: R² = 0.7284, MSE = 9048167.4616
Epoch 28: R² = 0.7285, MSE = 9044923.3019
Epoch 29: R² = 0.7214, MSE = 9280983.7067
Epoch 30: R² = 0.7243, MSE = 9183091.6120
Epoch 31: R² = 0.7203, MSE = 9318055.4313
Epoch 32: R² = 0.7141, MSE = 9524350.5926
Epoch 33: R² = 0.7181, MSE = 9390106.2194
Epoch 34: R² = 0.7137, MSE = 9536527.5105
Epoch 35: R² = 0.7128, MSE = 9568919.2424
Epoch 36: R² = 0.7076, MSE = 9741447.1425
Epoch 37: R² = 0.7115, MSE = 9611137.8478
Epoch 38: R² = 0.7099, MSE = 9664309.6667
Epoch 39: R² = 0.7092, MSE = 9688380.2573
Epoch 40: R² = 0.7080, MSE = 9726925.7158
Epoch 41: R² = 0.7103, MSE = 9650559.1338
Epoch 42: R² = 0.7141, MSE = 9525059.4125
Epoch 43: R² = 0.7165, MSE = 9444777.9419
Epoch 44: R² = 0.7148, MSE = 9500437.5268
Epoch 45: R² = 0.7137, MSE = 9537766.1079
Epoch 46: R² = 0.7109, MSE = 9629732.9458
Epoch 47: R² = 0.7068, MSE = 9765875.9224
Epoch 48: R² = 0.6998, MSE = 9999560.1577
Epoch 49: R² = 0.6972, MSE = 10088372.7601
Epoch 50: R² = 0.6958, MSE = 10133265.2769

Training Random Forest for Fibre Crops, Fibre Equivalent
Epoch 1: R² = 0.3566, MSE = 3007.9144
Epoch 2: R² = 0.4256, MSE = 2685.2847
Epoch 3: R² = 0.3366, MSE = 3101.2176
Epoch 4: R² = 0.3389, MSE = 3090.5909
Epoch 5: R² = 0.3973, MSE = 2817.4445
Epoch 6: R² = 0.4124, MSE = 2747.1901
Epoch 7: R² = 0.3139, MSE = 3207.3385
Epoch 8: R² = 0.3886, MSE = 2858.2577
Epoch 9: R² = 0.3983, MSE = 2812.8531
Epoch 10: R² = 0.4202, MSE = 2710.6804
Epoch 11: R² = 0.4529, MSE = 2557.8286
Epoch 12: R² = 0.4628, MSE = 2511.2215
Epoch 13: R² = 0.4626, MSE = 2512.3727
Epoch 14: R² = 0.4649, MSE = 2501.3725
Epoch 15: R² = 0.4958, MSE = 2357.0793
Epoch 16: R² = 0.5116, MSE = 2283.4028
Epoch 17: R² = 0.5223, MSE = 2233.0445
Epoch 18: R² = 0.5289, MSE = 2202.4852
Epoch 19: R² = 0.5199, MSE = 2244.3502
Epoch 20: R² = 0.5474, MSE = 2115.8716
Epoch 21: R² = 0.5416, MSE = 2143.1430
Epoch 22: R² = 0.5560, MSE = 2075.4335
Epoch 23: R² = 0.5621, MSE = 2047.0735
Epoch 24: R² = 0.5642, MSE = 2037.3027
Epoch 25: R² = 0.5588, MSE = 2062.7984
Epoch 26: R² = 0.5779, MSE = 1973.1385
Epoch 27: R² = 0.5684, MSE = 2017.6459
Epoch 28: R² = 0.5616, MSE = 2049.5570
Epoch 29: R² = 0.5569, MSE = 2071.4480
Epoch 30: R² = 0.5605, MSE = 2054.6095
Epoch 31: R² = 0.5472, MSE = 2117.0251
Epoch 32: R² = 0.5647, MSE = 2035.2239
Epoch 33: R² = 0.5654, MSE = 2031.6493
Epoch 34: R² = 0.5656, MSE = 2030.7592
Epoch 35: R² = 0.5728, MSE = 1997.0755
Epoch 36: R² = 0.5735, MSE = 1993.7722
Epoch 37: R² = 0.5705, MSE = 2008.0677
Epoch 38: R² = 0.5698, MSE = 2010.9727
Epoch 39: R² = 0.5771, MSE = 1976.9275
Epoch 40: R² = 0.5772, MSE = 1976.5688
Epoch 41: R² = 0.5793, MSE = 1966.5330
Epoch 42: R² = 0.5693, MSE = 2013.3191
```

```
Epoch 43: R² = 0.5747, MSE = 1988.2347
Epoch 44: R² = 0.5790, MSE = 1968.3686
Epoch 45: R² = 0.5791, MSE = 1967.4673
Epoch 46: R² = 0.5828, MSE = 1950.4939
Epoch 47: R² = 0.5844, MSE = 1942.9529
Epoch 48: R² = 0.5814, MSE = 1957.0029
Epoch 49: R² = 0.5793, MSE = 1966.7024
Epoch 50: R² = 0.5751, MSE = 1986.3554

Training Random Forest for Fruit Primary
Epoch 1: R² = 0.4939, MSE = 726828.9379
Epoch 2: R² = 0.2951, MSE = 1012254.3933
Epoch 3: R² = 0.1218, MSE = 1261160.7694
Epoch 4: R² = 0.0909, MSE = 1305516.6738
Epoch 5: R² = 0.0416, MSE = 1376342.1727
Epoch 6: R² = -0.0246, MSE = 1471285.7465
Epoch 7: R² = -0.0366, MSE = 1488627.3027
Epoch 8: R² = -0.0125, MSE = 1453995.3698
Epoch 9: R² = -0.0040, MSE = 1441773.5625
Epoch 10: R² = -0.0119, MSE = 1453150.3657
Epoch 11: R² = -0.0330, MSE = 1483475.3840
Epoch 12: R² = -0.0213, MSE = 1466663.7482
Epoch 13: R² = 0.0117, MSE = 1419277.1070
Epoch 14: R² = 0.0218, MSE = 1404643.3584
Epoch 15: R² = 0.0227, MSE = 1403473.5612
Epoch 16: R² = 0.0352, MSE = 1385417.9901
Epoch 17: R² = 0.0267, MSE = 1397747.2231
Epoch 18: R² = 0.0263, MSE = 1398267.7816
Epoch 19: R² = 0.0382, MSE = 1381172.6638
Epoch 20: R² = 0.0225, MSE = 1403761.3439
Epoch 21: R² = 0.0094, MSE = 1422536.8991
Epoch 22: R² = 0.0400, MSE = 1378591.0111
Epoch 23: R² = 0.0506, MSE = 1363343.3665
Epoch 24: R² = 0.0547, MSE = 1357419.8868
Epoch 25: R² = 0.0471, MSE = 1368425.1949
Epoch 26: R² = 0.0627, MSE = 1346033.7489
Epoch 27: R² = 0.0551, MSE = 1356827.3551
Epoch 28: R² = 0.0524, MSE = 1360750.7787
Epoch 29: R² = 0.0490, MSE = 1365586.2463
Epoch 30: R² = 0.0498, MSE = 1364555.2802
Epoch 31: R² = 0.0417, MSE = 1376104.7717
Epoch 32: R² = 0.0372, MSE = 1382575.1467
Epoch 33: R² = 0.0275, MSE = 1396497.5021
Epoch 34: R² = 0.0185, MSE = 1409411.3661
Epoch 35: R² = 0.0116, MSE = 1419398.2696
Epoch 36: R² = 0.0112, MSE = 1419886.9875
Epoch 37: R² = 0.0026, MSE = 1432342.9446
Epoch 38: R² = 0.0102, MSE = 1421430.3915
Epoch 39: R² = 0.0039, MSE = 1430385.9387
Epoch 40: R² = 0.0066, MSE = 1426504.8586
Epoch 41: R² = 0.0145, MSE = 1415170.2199
Epoch 42: R² = 0.0101, MSE = 1421523.5577
Epoch 43: R² = 0.0108, MSE = 1420443.4961
Epoch 44: R² = 0.0185, MSE = 1409522.2447
Epoch 45: R² = 0.0121, MSE = 1418673.0719
Epoch 46: R² = 0.0151, MSE = 1414346.6633
Epoch 47: R² = 0.0133, MSE = 1416919.0694
Epoch 48: R² = 0.0166, MSE = 1412114.0664
Epoch 49: R² = 0.0244, MSE = 1400927.4569
Epoch 50: R² = 0.0186, MSE = 1409288.8842

Training Random Forest for Meat, Poultry
Epoch 1: R² = 0.0454, MSE = 714386.5300
Epoch 2: R² = -0.4721, MSE = 1101701.6950
Epoch 3: R² = -0.4638, MSE = 1095490.3911
Epoch 4: R² = -0.4717, MSE = 1101415.1638
Epoch 5: R² = -0.4710, MSE = 1100871.6976
Epoch 6: R² = -0.4943, MSE = 1118334.1672
Epoch 7: R² = -0.5438, MSE = 1155351.5006
Epoch 8: R² = -0.5147, MSE = 1133582.4936
Epoch 9: R² = -0.5302, MSE = 1145219.6598
Epoch 10: R² = -0.4986, MSE = 1121571.3222
Epoch 11: R² = -0.4936, MSE = 1117777.9955
Epoch 12: R² = -0.4655, MSE = 1096779.4123
```

```
Epoch 13: R² = -0.4363, MSE = 1074941.2092
Epoch 14: R² = -0.4383, MSE = 1076395.1907
Epoch 15: R² = -0.4090, MSE = 1054514.0541
Epoch 16: R² = -0.4690, MSE = 1099367.4404
Epoch 17: R² = -0.4287, MSE = 1069268.2431
Epoch 18: R² = -0.3942, MSE = 1043389.4404
Epoch 19: R² = -0.3926, MSE = 1042210.5966
Epoch 20: R² = -0.3296, MSE = 995110.2006
Epoch 21: R² = -0.3655, MSE = 1021942.1923
Epoch 22: R² = -0.3493, MSE = 1009843.0472
Epoch 23: R² = -0.3541, MSE = 1013414.3329
Epoch 24: R² = -0.3340, MSE = 998403.5633
Epoch 25: R² = -0.3520, MSE = 1011853.8706
Epoch 26: R² = -0.3211, MSE = 988717.8268
Epoch 27: R² = -0.3138, MSE = 983243.8621
Epoch 28: R² = -0.3233, MSE = 990355.3633
Epoch 29: R² = -0.3439, MSE = 1005759.7409
Epoch 30: R² = -0.3198, MSE = 987757.9676
Epoch 31: R² = -0.3226, MSE = 989826.5619
Epoch 32: R² = -0.3464, MSE = 1007638.1943
Epoch 33: R² = -0.3095, MSE = 980020.8074
Epoch 34: R² = -0.3268, MSE = 992958.1754
Epoch 35: R² = -0.3271, MSE = 993187.8608
Epoch 36: R² = -0.3287, MSE = 994407.0912
Epoch 37: R² = -0.3261, MSE = 992446.7221
Epoch 38: R² = -0.3235, MSE = 990512.4695
Epoch 39: R² = -0.3398, MSE = 1002674.6300
Epoch 40: R² = -0.3315, MSE = 996489.5664
Epoch 41: R² = -0.3302, MSE = 995488.1823
Epoch 42: R² = -0.3300, MSE = 995362.5001
Epoch 43: R² = -0.3136, MSE = 983089.6181
Epoch 44: R² = -0.3101, MSE = 980456.2823
Epoch 45: R² = -0.3143, MSE = 983591.5557
Epoch 46: R² = -0.3212, MSE = 988820.1397
Epoch 47: R² = -0.3128, MSE = 982495.5166
Epoch 48: R² = -0.3388, MSE = 1001957.2334
Epoch 49: R² = -0.3475, MSE = 1008453.3146
Epoch 50: R² = -0.3453, MSE = 1006793.7964

Training Random Forest for Milk, Total
Epoch 1: R² = 0.7889, MSE = 11167189.4100
Epoch 2: R² = 0.6932, MSE = 16224924.2450
Epoch 3: R² = 0.7830, MSE = 11474502.6911
Epoch 4: R² = 0.8007, MSE = 10538510.9538
Epoch 5: R² = 0.8061, MSE = 10253817.9536
Epoch 6: R² = 0.8202, MSE = 9510049.3303
Epoch 7: R² = 0.8356, MSE = 8695394.6855
Epoch 8: R² = 0.8170, MSE = 9678241.3170
Epoch 9: R² = 0.8144, MSE = 9816089.5989
Epoch 10: R² = 0.8126, MSE = 9911336.2238
Epoch 11: R² = 0.8078, MSE = 10165606.1596
Epoch 12: R² = 0.8155, MSE = 9758511.7506
Epoch 13: R² = 0.8212, MSE = 9456525.9376
Epoch 14: R² = 0.8241, MSE = 9300974.7967
Epoch 15: R² = 0.8238, MSE = 9318401.5469
Epoch 16: R² = 0.8241, MSE = 9303005.4299
Epoch 17: R² = 0.8259, MSE = 9208482.9817
Epoch 18: R² = 0.8295, MSE = 9017275.2232
Epoch 19: R² = 0.8247, MSE = 9270978.5114
Epoch 20: R² = 0.8271, MSE = 9142195.4177
Epoch 21: R² = 0.8254, MSE = 9232396.0671
Epoch 22: R² = 0.8266, MSE = 9168698.5416
Epoch 23: R² = 0.8259, MSE = 9207417.8308
Epoch 24: R² = 0.8261, MSE = 9197939.7289
Epoch 25: R² = 0.8248, MSE = 9266481.8356
Epoch 26: R² = 0.8290, MSE = 9045833.8409
Epoch 27: R² = 0.8275, MSE = 9121784.4184
Epoch 28: R² = 0.8241, MSE = 9303616.4163
Epoch 29: R² = 0.8158, MSE = 9744056.0793
Epoch 30: R² = 0.8159, MSE = 9736039.3302
Epoch 31: R² = 0.8149, MSE = 9790015.2309
Epoch 32: R² = 0.8120, MSE = 9941968.1365
Epoch 33: R² = 0.8127, MSE = 9907447.3910
Epoch 34: R² = 0.8130, MSE = 9887889.3442
```

```
Epoch 35: R² = 0.8102, MSE = 10036000.8774
Epoch 36: R² = 0.8064, MSE = 10236918.3921
Epoch 37: R² = 0.8098, MSE = 10058156.6160
Epoch 38: R² = 0.8105, MSE = 10020471.1406
Epoch 39: R² = 0.8095, MSE = 10075821.4513
Epoch 40: R² = 0.8090, MSE = 10104012.2146
Epoch 41: R² = 0.8035, MSE = 10392902.5880
Epoch 42: R² = 0.8072, MSE = 10198108.8990
Epoch 43: R² = 0.8068, MSE = 10216403.7941
Epoch 44: R² = 0.8056, MSE = 10281570.9242
Epoch 45: R² = 0.8048, MSE = 10324595.7815
Epoch 46: R² = 0.8034, MSE = 10400196.0434
Epoch 47: R² = 0.8023, MSE = 10458214.8171
Epoch 48: R² = 0.7974, MSE = 10715557.1662
Epoch 49: R² = 0.7978, MSE = 10691983.1505
Epoch 50: R² = 0.7988, MSE = 10640085.1167

Training Random Forest for Treenuts, Total
Epoch 1: R² = 0.1754, MSE = 226650.8500
Epoch 2: R² = 0.2203, MSE = 214310.5823
Epoch 3: R² = 0.0947, MSE = 248828.2963
Epoch 4: R² = 0.1668, MSE = 229024.0619
Epoch 5: R² = 0.1815, MSE = 224981.8415
Epoch 6: R² = 0.2101, MSE = 217128.6990
Epoch 7: R² = 0.1653, MSE = 229437.3578
Epoch 8: R² = 0.1515, MSE = 233219.1717
Epoch 9: R² = 0.1489, MSE = 233949.0621
Epoch 10: R² = 0.1505, MSE = 233493.0140
Epoch 11: R² = 0.1878, MSE = 223238.5413
Epoch 12: R² = 0.1751, MSE = 226748.2542
Epoch 13: R² = 0.1641, MSE = 229754.9380
Epoch 14: R² = 0.1645, MSE = 229645.2896
Epoch 15: R² = 0.1911, MSE = 222352.2538
Epoch 16: R² = 0.1869, MSE = 223506.2434
Epoch 17: R² = 0.1752, MSE = 226702.7471
Epoch 18: R² = 0.1952, MSE = 221220.7667
Epoch 19: R² = 0.1903, MSE = 222570.0066
Epoch 20: R² = 0.1843, MSE = 224219.6725
Epoch 21: R² = 0.1939, MSE = 221571.2861
Epoch 22: R² = 0.2078, MSE = 217751.1938
Epoch 23: R² = 0.2045, MSE = 218645.8065
Epoch 24: R² = 0.1951, MSE = 221242.4010
Epoch 25: R² = 0.2116, MSE = 216699.3109
Epoch 26: R² = 0.2242, MSE = 213246.2949
Epoch 27: R² = 0.2252, MSE = 212955.9085
Epoch 28: R² = 0.2291, MSE = 211891.1552
Epoch 29: R² = 0.2219, MSE = 213862.9140
Epoch 30: R² = 0.2188, MSE = 214731.3948
Epoch 31: R² = 0.2206, MSE = 214233.5560
Epoch 32: R² = 0.2258, MSE = 212803.7414
Epoch 33: R² = 0.2299, MSE = 211680.6583
Epoch 34: R² = 0.2252, MSE = 212962.8061
Epoch 35: R² = 0.2202, MSE = 214352.0820
Epoch 36: R² = 0.2292, MSE = 211870.3506
Epoch 37: R² = 0.2195, MSE = 214539.7812
Epoch 38: R² = 0.2190, MSE = 214666.7854
Epoch 39: R² = 0.2193, MSE = 214587.9363
Epoch 40: R² = 0.2274, MSE = 212371.4653
Epoch 41: R² = 0.2357, MSE = 210090.9229
Epoch 42: R² = 0.2274, MSE = 212377.6256
Epoch 43: R² = 0.2243, MSE = 213229.8920
Epoch 44: R² = 0.2241, MSE = 213268.9312
Epoch 45: R² = 0.2281, MSE = 212169.4582
Epoch 46: R² = 0.2246, MSE = 213135.5015
Epoch 47: R² = 0.2266, MSE = 212582.0357
Epoch 48: R² = 0.2385, MSE = 209304.3061
Epoch 49: R² = 0.2341, MSE = 210511.6715
Epoch 50: R² = 0.2356, MSE = 210112.9691

Training Random Forest for Vegetables Primary
Epoch 1: R² = -0.0589, MSE = 2221123.3886
Epoch 2: R² = -0.5663, MSE = 3285395.7701
Epoch 3: R² = -0.3848, MSE = 2904573.2628
Epoch 4: R² = -0.3590, MSE = 2850508.0185
```

```
Epoch 5: R² = -0.2321, MSE = 2584447.4834
Epoch 6: R² = -0.1790, MSE = 2473021.7604
Epoch 7: R² = -0.1100, MSE = 2328176.2633
Epoch 8: R² = -0.1350, MSE = 2380689.3447
Epoch 9: R² = -0.1502, MSE = 2412483.1200
Epoch 10: R² = -0.1408, MSE = 2392870.1031
Epoch 11: R² = -0.1781, MSE = 2471193.6245
Epoch 12: R² = -0.2607, MSE = 2644409.7003
Epoch 13: R² = -0.2039, MSE = 2525263.8595
Epoch 14: R² = -0.1847, MSE = 2484944.7397
Epoch 15: R² = -0.1706, MSE = 2455316.5785
Epoch 16: R² = -0.1602, MSE = 2433624.3781
Epoch 17: R² = -0.1221, MSE = 2353708.1614
Epoch 18: R² = -0.1227, MSE = 2354924.6563
Epoch 19: R² = -0.1159, MSE = 2340708.5126
Epoch 20: R² = -0.0925, MSE = 2291520.8997
Epoch 21: R² = -0.1167, MSE = 2342379.3072
Epoch 22: R² = -0.1034, MSE = 2314410.3067
Epoch 23: R² = -0.1208, MSE = 2350930.1779
Epoch 24: R² = -0.1343, MSE = 2379149.0641
Epoch 25: R² = -0.1514, MSE = 2414987.9012
Epoch 26: R² = -0.1590, MSE = 2431034.2833
Epoch 27: R² = -0.1610, MSE = 2435216.2237
Epoch 28: R² = -0.1491, MSE = 2410301.8605
Epoch 29: R² = -0.1510, MSE = 2414235.6909
Epoch 30: R² = -0.1398, MSE = 2390704.7213
Epoch 31: R² = -0.1310, MSE = 2372341.1329
Epoch 32: R² = -0.1552, MSE = 2423059.4637
Epoch 33: R² = -0.1345, MSE = 2379728.9536
Epoch 34: R² = -0.1025, MSE = 2312532.5171
Epoch 35: R² = -0.0972, MSE = 2301497.0279
Epoch 36: R² = -0.1087, MSE = 2325486.0298
Epoch 37: R² = -0.0904, MSE = 2287131.9520
Epoch 38: R² = -0.0812, MSE = 2267898.6467
Epoch 39: R² = -0.0811, MSE = 2267653.3367
Epoch 40: R² = -0.0746, MSE = 2254083.1912
Epoch 41: R² = -0.0905, MSE = 2287296.2917
Epoch 42: R² = -0.0828, MSE = 2271114.2731
Epoch 43: R² = -0.0860, MSE = 2277964.6572
Epoch 44: R² = -0.0977, MSE = 2302456.3180
Epoch 45: R² = -0.1036, MSE = 2314798.1055
Epoch 46: R² = -0.1218, MSE = 2353063.1947
Epoch 47: R² = -0.1170, MSE = 2342946.4141
Epoch 48: R² = -0.1416, MSE = 2394580.1515
Epoch 49: R² = -0.1441, MSE = 2399774.9010
Epoch 50: R² = -0.1417, MSE = 2394730.2631

## Training XGBoost with Reduced Features

Training XGBoost for Beef and Buffalo Meat, primary
Epoch 1: R² = -0.7630, MSE = 48887.8534
Epoch 2: R² = -0.5983, MSE = 44320.7522
Epoch 3: R² = -0.4531, MSE = 40295.9408
Epoch 4: R² = -0.3251, MSE = 36745.2219
Epoch 5: R² = -0.2120, MSE = 33609.2525
Epoch 6: R² = -0.1209, MSE = 31083.8053
Epoch 7: R² = -0.0314, MSE = 28601.3711
Epoch 8: R² = 0.0458, MSE = 26460.2927
Epoch 9: R² = 0.1162, MSE = 24507.2821
Epoch 10: R² = 0.1782, MSE = 22788.5463
Epoch 11: R² = 0.2335, MSE = 21255.1731
Epoch 12: R² = 0.2829, MSE = 19886.4111
Epoch 13: R² = 0.3270, MSE = 18662.3329
Epoch 14: R² = 0.3665, MSE = 17567.0899
Epoch 15: R² = 0.4021, MSE = 16580.0586
Epoch 16: R² = 0.4340, MSE = 15696.0078
Epoch 17: R² = 0.4628, MSE = 14898.0100
Epoch 18: R² = 0.4907, MSE = 14121.9524
Epoch 19: R² = 0.5141, MSE = 13473.0196
Epoch 20: R² = 0.5353, MSE = 12885.4520
Epoch 21: R² = 0.5544, MSE = 12355.2857
Epoch 22: R² = 0.5717, MSE = 11876.3222
Epoch 23: R² = 0.5873, MSE = 11444.7160
Epoch 24: R² = 0.6014, MSE = 11052.2017
```

```
Epoch 25: R² = 0.6143, MSE = 10696.3061
Epoch 26: R² = 0.6275, MSE = 10328.4501
Epoch 27: R² = 0.6377, MSE = 10045.8728
Epoch 28: R² = 0.6471, MSE = 9785.7755
Epoch 29: R² = 0.6560, MSE = 9537.8739
Epoch 30: R² = 0.6643, MSE = 9309.0509
Epoch 31: R² = 0.6717, MSE = 9104.2447
Epoch 32: R² = 0.6787, MSE = 8908.3577
Epoch 33: R² = 0.6853, MSE = 8727.1935
Epoch 34: R² = 0.6913, MSE = 8559.5185
Epoch 35: R² = 0.6969, MSE = 8404.2320
Epoch 36: R² = 0.7020, MSE = 8264.5818
Epoch 37: R² = 0.7068, MSE = 8130.7929
Epoch 38: R² = 0.7113, MSE = 8006.6565
Epoch 39: R² = 0.7154, MSE = 7892.0996
Epoch 40: R² = 0.7187, MSE = 7800.6980
Epoch 41: R² = 0.7218, MSE = 7715.5571
Epoch 42: R² = 0.7274, MSE = 7559.0185
Epoch 43: R² = 0.7326, MSE = 7415.0332
Epoch 44: R² = 0.7374, MSE = 7282.5197
Epoch 45: R² = 0.7400, MSE = 7210.9808
Epoch 46: R² = 0.7424, MSE = 7142.3690
Epoch 47: R² = 0.7447, MSE = 7078.2185
Epoch 48: R² = 0.7485, MSE = 6975.0354
Epoch 49: R² = 0.7505, MSE = 6919.8567
Epoch 50: R² = 0.7523, MSE = 6868.1886

Training XGBoost for Cereals, primary
Epoch 1: R² = -0.3003, MSE = 665466.1041
Epoch 2: R² = -0.0925, MSE = 559119.7765
Epoch 3: R² = 0.0827, MSE = 469447.4486
Epoch 4: R² = 0.2216, MSE = 398348.0515
Epoch 5: R² = 0.3397, MSE = 337931.9568
Epoch 6: R² = 0.4404, MSE = 286377.4019
Epoch 7: R² = 0.5257, MSE = 242708.9124
Epoch 8: R² = 0.5924, MSE = 208604.5881
Epoch 9: R² = 0.6408, MSE = 183811.2951
Epoch 10: R² = 0.6823, MSE = 162598.2440
Epoch 11: R² = 0.7225, MSE = 141999.4959
Epoch 12: R² = 0.7535, MSE = 126143.3964
Epoch 13: R² = 0.7790, MSE = 113124.9393
Epoch 14: R² = 0.8014, MSE = 101650.0358
Epoch 15: R² = 0.8200, MSE = 92096.5566
Epoch 16: R² = 0.8354, MSE = 84219.5002
Epoch 17: R² = 0.8482, MSE = 77681.8977
Epoch 18: R² = 0.8578, MSE = 72761.2438
Epoch 19: R² = 0.8775, MSE = 62686.8909
Epoch 20: R² = 0.8849, MSE = 58888.8698
Epoch 21: R² = 0.8897, MSE = 56424.7212
Epoch 22: R² = 0.8932, MSE = 54667.1502
Epoch 23: R² = 0.8954, MSE = 53520.4256
Epoch 24: R² = 0.8968, MSE = 52822.9764
Epoch 25: R² = 0.8971, MSE = 52654.9093
Epoch 26: R² = 0.8967, MSE = 52873.0845
Epoch 27: R² = 0.8941, MSE = 54174.4568
Epoch 28: R² = 0.8912, MSE = 55688.6349
Epoch 29: R² = 0.8879, MSE = 57377.9991
Epoch 30: R² = 0.8847, MSE = 59001.4101
Epoch 31: R² = 0.8809, MSE = 60952.6297
Epoch 32: R² = 0.8769, MSE = 62990.7822
Epoch 33: R² = 0.8728, MSE = 65093.7429
Epoch 34: R² = 0.8686, MSE = 67242.1424
Epoch 35: R² = 0.8644, MSE = 69419.1648
Epoch 36: R² = 0.8603, MSE = 71485.6075
Epoch 37: R² = 0.8560, MSE = 73681.8088
Epoch 38: R² = 0.8518, MSE = 75868.3932
Epoch 39: R² = 0.8475, MSE = 78036.1973
Epoch 40: R² = 0.8433, MSE = 80177.3726
Epoch 41: R² = 0.8392, MSE = 82285.3812
Epoch 42: R² = 0.8353, MSE = 84279.2580
Epoch 43: R² = 0.8314, MSE = 86307.7890
Epoch 44: R² = 0.8275, MSE = 88289.3974
Epoch 45: R² = 0.8237, MSE = 90221.1541
Epoch 46: R² = 0.8200, MSE = 92100.9207
```

```
Epoch 47: R² = 0.8165, MSE = 93927.0226
Epoch 48: R² = 0.8130, MSE = 95698.2383
Epoch 49: R² = 0.8097, MSE = 97413.8598
Epoch 50: R² = 0.8064, MSE = 99073.4388

Training XGBoost for Eggs Primary
Epoch 1: R² = -0.2756, MSE = 42494298.6876
Epoch 2: R² = -0.1446, MSE = 38130191.8838
Epoch 3: R² = -0.0301, MSE = 34315759.2966
Epoch 4: R² = 0.0706, MSE = 30961728.6520
Epoch 5: R² = 0.1432, MSE = 28542500.3118
Epoch 6: R² = 0.2076, MSE = 26398474.1356
Epoch 7: R² = 0.2647, MSE = 24495474.6085
Epoch 8: R² = 0.3180, MSE = 22720512.5566
Epoch 9: R² = 0.3746, MSE = 20834590.0970
Epoch 10: R² = 0.4148, MSE = 19494637.6751
Epoch 11: R² = 0.4507, MSE = 18298468.2603
Epoch 12: R² = 0.4844, MSE = 17176901.4254
Epoch 13: R² = 0.5131, MSE = 16221578.6681
Epoch 14: R² = 0.5389, MSE = 15360415.8743
Epoch 15: R² = 0.5621, MSE = 14587210.7356
Epoch 16: R² = 0.5830, MSE = 13892022.4505
Epoch 17: R² = 0.6042, MSE = 13184334.8801
Epoch 18: R² = 0.6235, MSE = 12543390.3804
Epoch 19: R² = 0.6409, MSE = 11962244.9545
Epoch 20: R² = 0.6549, MSE = 11494621.8785
Epoch 21: R² = 0.6648, MSE = 11165121.4599
Epoch 22: R² = 0.6742, MSE = 10853597.2106
Epoch 23: R² = 0.6827, MSE = 10568699.7631
Epoch 24: R² = 0.6906, MSE = 10307892.2494
Epoch 25: R² = 0.6959, MSE = 10131057.0154
Epoch 26: R² = 0.7007, MSE = 9968987.8928
Epoch 27: R² = 0.7052, MSE = 9820411.1871
Epoch 28: R² = 0.7093, MSE = 9684065.7845
Epoch 29: R² = 0.7131, MSE = 9558933.7576
Epoch 30: R² = 0.7165, MSE = 9443975.8033
Epoch 31: R² = 0.7197, MSE = 9338304.0381
Epoch 32: R² = 0.7233, MSE = 9218864.2418
Epoch 33: R² = 0.7257, MSE = 9136707.5797
Epoch 34: R² = 0.7282, MSE = 9054433.7819
Epoch 35: R² = 0.7305, MSE = 8978670.8641
Epoch 36: R² = 0.7331, MSE = 8892447.2365
Epoch 37: R² = 0.7348, MSE = 8833499.1412
Epoch 38: R² = 0.7366, MSE = 8774259.0680
Epoch 39: R² = 0.7380, MSE = 8728757.5866
Epoch 40: R² = 0.7392, MSE = 8686798.5724
Epoch 41: R² = 0.7404, MSE = 8648066.4750
Epoch 42: R² = 0.7415, MSE = 8612338.3854
Epoch 43: R² = 0.7425, MSE = 8576608.9240
Epoch 44: R² = 0.7434, MSE = 8549437.0820
Epoch 45: R² = 0.7441, MSE = 8524114.8987
Epoch 46: R² = 0.7448, MSE = 8502621.5822
Epoch 47: R² = 0.7454, MSE = 8480539.9855
Epoch 48: R² = 0.7460, MSE = 8459919.2376
Epoch 49: R² = 0.7467, MSE = 8436427.9267
Epoch 50: R² = 0.7473, MSE = 8418683.4650

Training XGBoost for Fibre Crops, Fibre Equivalent
Epoch 1: R² = 0.1284, MSE = 4074.8263
Epoch 2: R² = 0.2814, MSE = 3359.2612
Epoch 3: R² = 0.3947, MSE = 2829.5551
Epoch 4: R² = 0.4754, MSE = 2452.4812
Epoch 5: R² = 0.5294, MSE = 2200.1235
Epoch 6: R² = 0.5631, MSE = 2042.3167
Epoch 7: R² = 0.5780, MSE = 1972.8548
Epoch 8: R² = 0.5800, MSE = 1963.3090
Epoch 9: R² = 0.5721, MSE = 2000.3041
Epoch 10: R² = 0.5552, MSE = 2079.3005
Epoch 11: R² = 0.5314, MSE = 2190.6492
Epoch 12: R² = 0.4972, MSE = 2350.6406
Epoch 13: R² = 0.4651, MSE = 2500.6412
Epoch 14: R² = 0.4262, MSE = 2682.3807
Epoch 15: R² = 0.3862, MSE = 2869.5863
Epoch 16: R² = 0.3456, MSE = 3059.0913
```

```
Epoch 17: R² = 0.3052, MSE = 3248.3216
Epoch 18: R² = 0.2652, MSE = 3435.2288
Epoch 19: R² = 0.2258, MSE = 3619.2174
Epoch 20: R² = 0.1877, MSE = 3797.1998
Epoch 21: R² = 0.1511, MSE = 3968.3588
Epoch 22: R² = 0.1146, MSE = 4139.3148
Epoch 23: R² = 0.0808, MSE = 4297.1670
Epoch 24: R² = 0.0475, MSE = 4453.0369
Epoch 25: R² = 0.0158, MSE = 4601.0906
Epoch 26: R² = 0.0023, MSE = 4663.9895
Epoch 27: R² = -0.0095, MSE = 4719.4914
Epoch 28: R² = -0.0205, MSE = 4770.7686
Epoch 29: R² = -0.0309, MSE = 4819.2530
Epoch 30: R² = -0.0415, MSE = 4868.9742
Epoch 31: R² = -0.0507, MSE = 4912.0399
Epoch 32: R² = -0.0594, MSE = 4952.6550
Epoch 33: R² = -0.0683, MSE = 4993.9961
Epoch 34: R² = -0.0759, MSE = 5029.9239
Epoch 35: R² = -0.0832, MSE = 5063.7372
Epoch 36: R² = -0.0901, MSE = 5096.1173
Epoch 37: R² = -0.0965, MSE = 5125.9326
Epoch 38: R² = -0.1029, MSE = 5156.0759
Epoch 39: R² = -0.1085, MSE = 5182.3149
Epoch 40: R² = -0.1138, MSE = 5206.9530
Epoch 41: R² = -0.1191, MSE = 5231.7594
Epoch 42: R² = -0.1237, MSE = 5253.4094
Epoch 43: R² = -0.1284, MSE = 5275.1675
Epoch 44: R² = -0.1324, MSE = 5293.8672
Epoch 45: R² = -0.1362, MSE = 5311.6413
Epoch 46: R² = -0.1397, MSE = 5327.9612
Epoch 47: R² = -0.1385, MSE = 5322.4309
Epoch 48: R² = -0.1374, MSE = 5317.2849
Epoch 49: R² = -0.1364, MSE = 5312.4969
Epoch 50: R² = -0.1383, MSE = 5321.3780

Training XGBoost for Fruit Primary
Epoch 1: R² = -0.0593, MSE = 1521125.4767
Epoch 2: R² = -0.0820, MSE = 1553764.4700
Epoch 3: R² = -0.1088, MSE = 1592189.7742
Epoch 4: R² = -0.1383, MSE = 1634616.1826
Epoch 5: R² = -0.1696, MSE = 1679610.4785
Epoch 6: R² = -0.2020, MSE = 1726022.8840
Epoch 7: R² = -0.2346, MSE = 1772944.8296
Epoch 8: R² = -0.2672, MSE = 1819662.0761
Epoch 9: R² = -0.2992, MSE = 1865621.3241
Epoch 10: R² = -0.3304, MSE = 1910405.9386
Epoch 11: R² = -0.3605, MSE = 1953702.7130
Epoch 12: R² = -0.3895, MSE = 1995288.9772
Epoch 13: R² = -0.4171, MSE = 2035009.2705
Epoch 14: R² = -0.4434, MSE = 2072773.0256
Epoch 15: R² = -0.4575, MSE = 2093062.4442
Epoch 16: R² = -0.4817, MSE = 2127689.5618
Epoch 17: R² = -0.5043, MSE = 2160263.1054
Epoch 18: R² = -0.5283, MSE = 2194721.2606
Epoch 19: R² = -0.5408, MSE = 2212579.2334
Epoch 20: R² = -0.5627, MSE = 2244059.0383
Epoch 21: R² = -0.5733, MSE = 2259357.5783
Epoch 22: R² = -0.5933, MSE = 2287981.4634
Epoch 23: R² = -0.6120, MSE = 2314825.1434
Epoch 24: R² = -0.6295, MSE = 2339958.9942
Epoch 25: R² = -0.6458, MSE = 2363465.8361
Epoch 26: R² = -0.6611, MSE = 2385421.6404
Epoch 27: R² = -0.6690, MSE = 2396654.4413
Epoch 28: R² = -0.6826, MSE = 2416309.2533
Epoch 29: R² = -0.6954, MSE = 2434622.1156
Epoch 30: R² = -0.7073, MSE = 2451669.0022
Epoch 31: R² = -0.7183, MSE = 2467524.4707
Epoch 32: R² = -0.7241, MSE = 2475782.6329
Epoch 33: R² = -0.7355, MSE = 2492250.9820
Epoch 34: R² = -0.7463, MSE = 2507711.4228
Epoch 35: R² = -0.7564, MSE = 2522222.3465
Epoch 36: R² = -0.7659, MSE = 2535830.4681
Epoch 37: R² = -0.7663, MSE = 2536456.5302
Epoch 38: R² = -0.7666, MSE = 2536934.5545
```

```
Epoch 39: R² = -0.7669, MSE = 2537279.6241
Epoch 40: R² = -0.7670, MSE = 2537507.2333
Epoch 41: R² = -0.7671, MSE = 2537626.8593
Epoch 42: R² = -0.7671, MSE = 2537654.2854
Epoch 43: R² = -0.7671, MSE = 2537598.3293
Epoch 44: R² = -0.7661, MSE = 2536082.2998
Epoch 45: R² = -0.7657, MSE = 2535528.0094
Epoch 46: R² = -0.7678, MSE = 2538611.7695
Epoch 47: R² = -0.7698, MSE = 2541464.9363
Epoch 48: R² = -0.7726, MSE = 2545493.7291
Epoch 49: R² = -0.7752, MSE = 2549255.0662
Epoch 50: R² = -0.7777, MSE = 2552767.5992

Training XGBoost for Meat, Poultry
Epoch 1: R² = -1.5214, MSE = 1887054.7730
Epoch 2: R² = -1.3049, MSE = 1725000.0268
Epoch 3: R² = -1.1175, MSE = 1584776.4395
Epoch 4: R² = -0.9548, MSE = 1462999.6679
Epoch 5: R² = -0.8123, MSE = 1356317.9728
Epoch 6: R² = -0.6891, MSE = 1264087.6036
Epoch 7: R² = -0.5816, MSE = 1183683.5941
Epoch 8: R² = -0.4878, MSE = 1113486.2182
Epoch 9: R² = -0.4047, MSE = 1051273.0592
Epoch 10: R² = -0.3329, MSE = 997537.0569
Epoch 11: R² = -0.2545, MSE = 938842.3999
Epoch 12: R² = -0.1842, MSE = 886255.5628
Epoch 13: R² = -0.1123, MSE = 832438.3409
Epoch 14: R² = -0.1087, MSE = 829750.0189
Epoch 15: R² = -0.0483, MSE = 784552.1229
Epoch 16: R² = -0.0475, MSE = 783945.0875
Epoch 17: R² = 0.0034, MSE = 745833.9974
Epoch 18: R² = 0.0008, MSE = 747821.4640
Epoch 19: R² = 0.0440, MSE = 715456.6680
Epoch 20: R² = 0.0427, MSE = 716444.9328
Epoch 21: R² = 0.0794, MSE = 688981.8747
Epoch 22: R² = 0.0818, MSE = 687168.3924
Epoch 23: R² = 0.1128, MSE = 663945.7640
Epoch 24: R² = 0.1219, MSE = 657165.0498
Epoch 25: R² = 0.1482, MSE = 637471.7949
Epoch 26: R² = 0.1440, MSE = 640627.3016
Epoch 27: R² = 0.1639, MSE = 625718.4343
Epoch 28: R² = 0.1847, MSE = 610150.7215
Epoch 29: R² = 0.1910, MSE = 605455.9357
Epoch 30: R² = 0.2088, MSE = 592167.3484
Epoch 31: R² = 0.2136, MSE = 588550.4807
Epoch 32: R² = 0.2288, MSE = 577176.8225
Epoch 33: R² = 0.2334, MSE = 573739.8152
Epoch 34: R² = 0.2468, MSE = 563701.1751
Epoch 35: R² = 0.2499, MSE = 561400.6011
Epoch 36: R² = 0.2531, MSE = 559013.1662
Epoch 37: R² = 0.2635, MSE = 551196.2694
Epoch 38: R² = 0.2672, MSE = 548421.4353
Epoch 39: R² = 0.2762, MSE = 541692.0292
Epoch 40: R² = 0.2786, MSE = 539895.1049
Epoch 41: R² = 0.2864, MSE = 534091.3370
Epoch 42: R² = 0.2884, MSE = 532565.8034
Epoch 43: R² = 0.2953, MSE = 527399.9720
Epoch 44: R² = 0.2969, MSE = 526231.4216
Epoch 45: R² = 0.2985, MSE = 525017.9087
Epoch 46: R² = 0.3039, MSE = 520971.9242
Epoch 47: R² = 0.3053, MSE = 519935.3278
Epoch 48: R² = 0.3100, MSE = 516431.2259
Epoch 49: R² = 0.3111, MSE = 515544.8775
Epoch 50: R² = 0.3152, MSE = 512509.1477

Training XGBoost for Milk, Total
Epoch 1: R² = -0.2426, MSE = 65717607.9738
Epoch 2: R² = -0.1070, MSE = 58546082.1888
Epoch 3: R² = 0.0114, MSE = 52283709.5498
Epoch 4: R² = 0.1150, MSE = 46808806.2146
Epoch 5: R² = 0.2047, MSE = 42059762.3757
Epoch 6: R² = 0.2844, MSE = 37845415.6164
Epoch 7: R² = 0.3469, MSE = 34539162.1922
Epoch 8: R² = 0.4021, MSE = 31620133.3356
```

```
Epoch 9: R² = 0.4509, MSE = 29039187.3206
Epoch 10: R² = 0.4941, MSE = 26753769.4612
Epoch 11: R² = 0.5415, MSE = 24249577.3138
Epoch 12: R² = 0.5787, MSE = 22280987.1589
Epoch 13: R² = 0.6119, MSE = 20527664.7984
Epoch 14: R² = 0.6414, MSE = 18964361.3198
Epoch 15: R² = 0.6678, MSE = 17568887.8441
Epoch 16: R² = 0.6917, MSE = 16306811.0997
Epoch 17: R² = 0.7130, MSE = 15177074.3977
Epoch 18: R² = 0.7322, MSE = 14164604.8359
Epoch 19: R² = 0.7492, MSE = 13265973.4583
Epoch 20: R² = 0.7645, MSE = 12453610.1586
Epoch 21: R² = 0.7785, MSE = 11714201.1565
Epoch 22: R² = 0.7911, MSE = 11048555.8027
Epoch 23: R² = 0.8024, MSE = 10448633.3460
Epoch 24: R² = 0.8127, MSE = 9907273.9931
Epoch 25: R² = 0.8209, MSE = 9470967.6708
Epoch 26: R² = 0.8293, MSE = 9025778.6167
Epoch 27: R² = 0.8370, MSE = 8622721.2818
Epoch 28: R² = 0.8440, MSE = 8253088.5101
Epoch 29: R² = 0.8503, MSE = 7917391.9576
Epoch 30: R² = 0.8552, MSE = 7656151.7432
Epoch 31: R² = 0.8597, MSE = 7421913.2147
Epoch 32: R² = 0.8638, MSE = 7205703.9060
Epoch 33: R² = 0.8682, MSE = 6968255.8247
Epoch 34: R² = 0.8716, MSE = 6788719.4738
Epoch 35: R² = 0.8743, MSE = 6646586.8011
Epoch 36: R² = 0.8772, MSE = 6496371.7078
Epoch 37: R² = 0.8803, MSE = 6329315.4157
Epoch 38: R² = 0.8827, MSE = 6203568.5492
Epoch 39: R² = 0.8849, MSE = 6088354.9409
Epoch 40: R² = 0.8873, MSE = 5960774.5042
Epoch 41: R² = 0.8891, MSE = 5863691.6456
Epoch 42: R² = 0.8908, MSE = 5774522.5490
Epoch 43: R² = 0.8920, MSE = 5711899.6207
Epoch 44: R² = 0.8934, MSE = 5636890.0356
Epoch 45: R² = 0.8944, MSE = 5583551.2479
Epoch 46: R² = 0.8954, MSE = 5534299.1694
Epoch 47: R² = 0.8965, MSE = 5475123.0350
Epoch 48: R² = 0.8973, MSE = 5433044.1653
Epoch 49: R² = 0.8980, MSE = 5394133.9859
Epoch 50: R² = 0.8989, MSE = 5347286.7074

Training XGBoost for Treenuts, Total
Epoch 1: R² = -0.0160, MSE = 279265.9686
Epoch 2: R² = 0.0600, MSE = 258374.3882
Epoch 3: R² = 0.1205, MSE = 241748.3686
Epoch 4: R² = 0.1665, MSE = 229095.6710
Epoch 5: R² = 0.2000, MSE = 219894.3040
Epoch 6: R² = 0.2253, MSE = 212953.3331
Epoch 7: R² = 0.2426, MSE = 208187.1844
Epoch 8: R² = 0.2527, MSE = 205399.1615
Epoch 9: R² = 0.2582, MSE = 203896.8758
Epoch 10: R² = 0.2593, MSE = 203600.2147
Epoch 11: R² = 0.2568, MSE = 204274.8013
Epoch 12: R² = 0.2324, MSE = 210990.5850
Epoch 13: R² = 0.2074, MSE = 217868.4217
Epoch 14: R² = 0.1978, MSE = 220502.1986
Epoch 15: R² = 0.1714, MSE = 227760.2023
Epoch 16: R² = 0.1471, MSE = 234444.6395
Epoch 17: R² = 0.1258, MSE = 240295.4878
Epoch 18: R² = 0.1049, MSE = 246023.1032
Epoch 19: R² = 0.0847, MSE = 251601.2645
Epoch 20: R² = 0.0650, MSE = 257009.8444
Epoch 21: R² = 0.0565, MSE = 259346.0125
Epoch 22: R² = 0.0363, MSE = 264889.3128
Epoch 23: R² = 0.0170, MSE = 270188.2859
Epoch 24: R² = -0.0014, MSE = 275242.6303
Epoch 25: R² = -0.0197, MSE = 280280.3569
Epoch 26: R² = -0.0371, MSE = 285070.5346
Epoch 27: R² = -0.0537, MSE = 289617.4460
Epoch 28: R² = -0.0686, MSE = 293719.6069
Epoch 29: R² = -0.0827, MSE = 297614.9728
Epoch 30: R² = -0.0962, MSE = 301308.9101
```

```
Epoch 31: R² = -0.1089, MSE = 304807.2569
Epoch 32: R² = -0.1210, MSE = 308116.6238
Epoch 33: R² = -0.1323, MSE = 311243.8611
Epoch 34: R² = -0.1431, MSE = 314196.1842
Epoch 35: R² = -0.1532, MSE = 316980.7447
Epoch 36: R² = -0.1628, MSE = 319605.5244
Epoch 37: R² = -0.1718, MSE = 322080.9197
Epoch 38: R² = -0.1802, MSE = 324411.1547
Epoch 39: R² = -0.1848, MSE = 325661.1904
Epoch 40: R² = -0.1918, MSE = 327590.3963
Epoch 41: R² = -0.1994, MSE = 329679.9308
Epoch 42: R² = -0.2059, MSE = 331463.6248
Epoch 43: R² = -0.2123, MSE = 333220.7490
Epoch 44: R² = -0.2180, MSE = 334788.8916
Epoch 45: R² = -0.2236, MSE = 336334.9468
Epoch 46: R² = -0.2287, MSE = 337735.4928
Epoch 47: R² = -0.2332, MSE = 338981.8486
Epoch 48: R² = -0.2377, MSE = 340210.9118
Epoch 49: R² = -0.2428, MSE = 341619.0693
Epoch 50: R² = -0.2468, MSE = 342696.6269

Training XGBoost for Vegetables Primary
Epoch 1: R² = -1.3150, MSE = 4855831.5741
Epoch 2: R² = -1.1497, MSE = 4509084.2620
Epoch 3: R² = -0.9881, MSE = 4170093.9167
Epoch 4: R² = -0.8457, MSE = 3871366.3341
Epoch 5: R² = -0.7203, MSE = 3608442.0913
Epoch 6: R² = -0.6202, MSE = 3398435.1416
Epoch 7: R² = -0.5215, MSE = 3191364.9171
Epoch 8: R² = -0.4340, MSE = 3007911.9092
Epoch 9: R² = -0.3439, MSE = 2818825.3807
Epoch 10: R² = -0.2631, MSE = 2649373.5963
Epoch 11: R² = -0.2049, MSE = 2527311.4477
Epoch 12: R² = -0.1636, MSE = 2440747.9413
Epoch 13: R² = -0.1274, MSE = 2364769.4354
Epoch 14: R² = -0.0949, MSE = 2296618.6971
Epoch 15: R² = -0.0658, MSE = 2235500.6094
Epoch 16: R² = -0.0391, MSE = 2179620.5873
Epoch 17: R² = -0.0157, MSE = 2130501.8621
Epoch 18: R² = 0.0053, MSE = 2086488.5046
Epoch 19: R² = 0.0242, MSE = 2046819.9177
Epoch 20: R² = 0.0410, MSE = 2011539.8558
Epoch 21: R² = 0.0563, MSE = 1979347.8488
Epoch 22: R² = 0.0680, MSE = 1954791.0170
Epoch 23: R² = 0.0801, MSE = 1929547.1104
Epoch 24: R² = 0.0909, MSE = 1906770.8454
Epoch 25: R² = 0.1015, MSE = 1884637.1741
Epoch 26: R² = 0.1081, MSE = 1870765.0711
Epoch 27: R² = 0.1140, MSE = 1858361.3803
Epoch 28: R² = 0.1192, MSE = 1847503.9966
Epoch 29: R² = 0.1238, MSE = 1837817.5317
Epoch 30: R² = 0.1278, MSE = 1829366.6697
Epoch 31: R² = 0.1482, MSE = 1786613.0430
Epoch 32: R² = 0.1668, MSE = 1747710.4235
Epoch 33: R² = 0.1837, MSE = 1712275.7514
Epoch 34: R² = 0.1855, MSE = 1708429.1458
Epoch 35: R² = 0.1869, MSE = 1705407.7483
Epoch 36: R² = 0.1876, MSE = 1703985.6165
Epoch 37: R² = 0.1866, MSE = 1706161.3629
Epoch 38: R² = 0.1854, MSE = 1708602.9443
Epoch 39: R² = 0.1889, MSE = 1701362.7052
Epoch 40: R² = 0.1876, MSE = 1703985.9801
Epoch 41: R² = 0.1906, MSE = 1697721.2526
Epoch 42: R² = 0.1893, MSE = 1700444.1845
Epoch 43: R² = 0.1905, MSE = 1697844.8922
Epoch 44: R² = 0.1917, MSE = 1695511.4116
Epoch 45: R² = 0.1927, MSE = 1693414.4366
Epoch 46: R² = 0.1936, MSE = 1691534.0997
Epoch 47: R² = 0.1944, MSE = 1689846.8329
Epoch 48: R² = 0.1951, MSE = 1688334.2556
Epoch 49: R² = 0.1958, MSE = 1686921.9332
Epoch 50: R² = 0.1964, MSE = 1685656.4991

Model Performance Analysis
```

SVR Average R²: -0.7064

Top 5 performing targets:
Treenuts, Total R² = -0.0824
Fruit Primary R² = -0.1225
Fibre Crops, Fibre Equivalent R² = -0.1486
Eggs Primary R² = -0.3444
Milk, Total R² = -0.5030

Random Forest Average R²: 0.3335

Top 5 performing targets:
Milk, Total R² = 0.7988
Cereals, primary R² = 0.7245
Eggs Primary R² = 0.6958
Fibre Crops, Fibre Equivalent R² = 0.5751
Beef and Buffalo Meat, primary R² = 0.4397

XGBoost Average R²: 0.2837

Top 5 performing targets:
Milk, Total R² = 0.8989
Cereals, primary R² = 0.8064
Beef and Buffalo Meat, primary R² = 0.7523
Eggs Primary R² = 0.7473
Meat, Poultry R² = 0.3152