Spring Wheat Yield Production

Nirali Kotak
Mentor : Ruhid Mirzayev

Overview

- Introduction
- Problem Statement - Predict Spring Wheat Yield
- Data Collection and Preprocessing
- Methodology
    - Time Series Analysis
    - K-means Clustering algorithm

Introduction

Agriculture sector has always been at the core of life and it requires years of experience to plan a season of farming along with optimizing the soil nutrition, water usage, crop yield, profits and make it sustainable

Historical data can help improve the optimization with a better accuracy and can help farmers to plan and strategize more effectively

This project provides insights into Spring Wheat yield prediction based on the historical data provided by Government of Saskatchewan from the year 1938

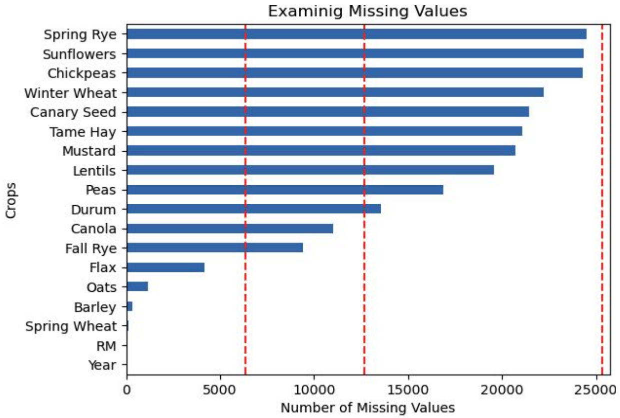
Data Collection and Preprocessing

- Data cleaning is a very crucial step in Data Science and Analysis. The models and algorithms for predictions vary widely on how data cleaning is performed
- Data cleaning includes identifying and fixing of:
    - Missing values like NaN or null
    - Incorrect values or outliers
    - Incorrectly formatted
    - Corrupted values
    - Incomplete data

Data Collection and Preprocessing

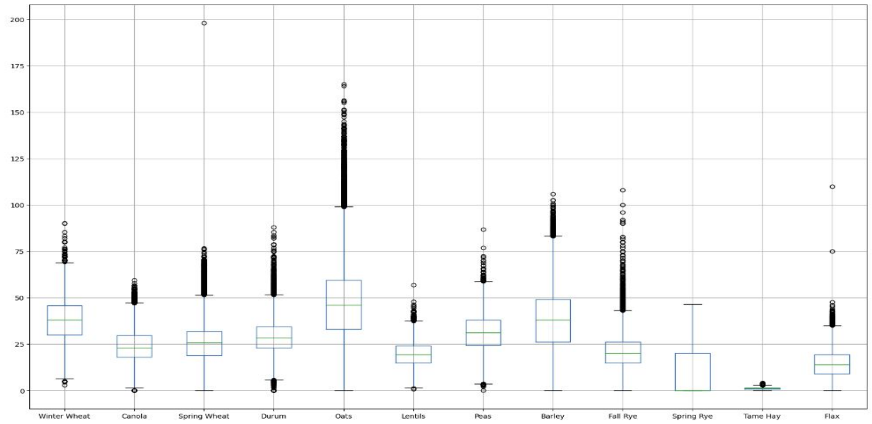
- Ensure data collected yearly is not duplicated
- Format the columns datatype
- Missing Values

```
In [21]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25312 entries, 0 to 25311
click to expand output; double click to hide output mns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   Year          25312 non-null   int64
 1   RM            25312 non-null   int64
 2   Winter Wheat  3073 non-null    float64
 3   Canola        14299 non-null   float64
 4   Spring Wheat  25213 non-null   float64
 5   Mustard       4584 non-null    float64
 6   Durum         11753 non-null   float64
 7   Sunflowers    946 non-null     float64
 8   Oats          24148 non-null   float64
 9   Lentils       5711 non-null    float64
 10  Peas          8421 non-null    float64
 11  Barley        24987 non-null   float64
 12  Fall Rye      15887 non-null   float64
 13  Canary Seed   3880 non-null    float64
 14  Spring Rye    805 non-null     float64
 15  Tame Hay      4205 non-null    float64
 16  Flax          21146 non-null   float64
 17  Chickpeas     1014 non-null    float64
dtypes: float64(16), int64(2)
memory usage: 3.5 MB

In [22]: df['RM']=df['RM'].astype('str')
```



Data Collection and Preprocessing

- Identify Outliers

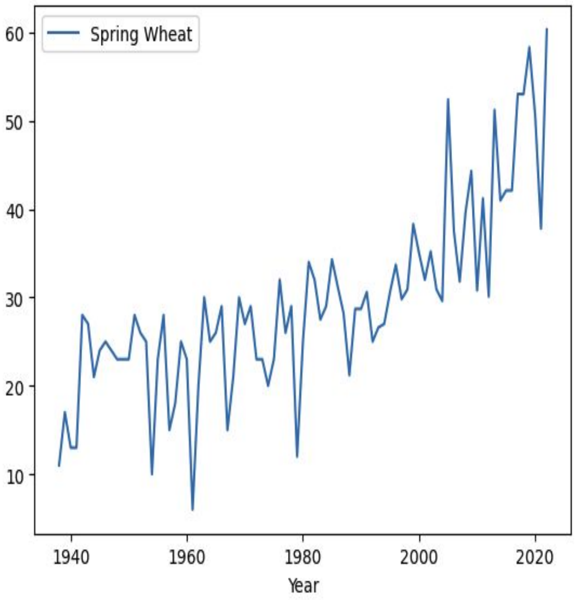


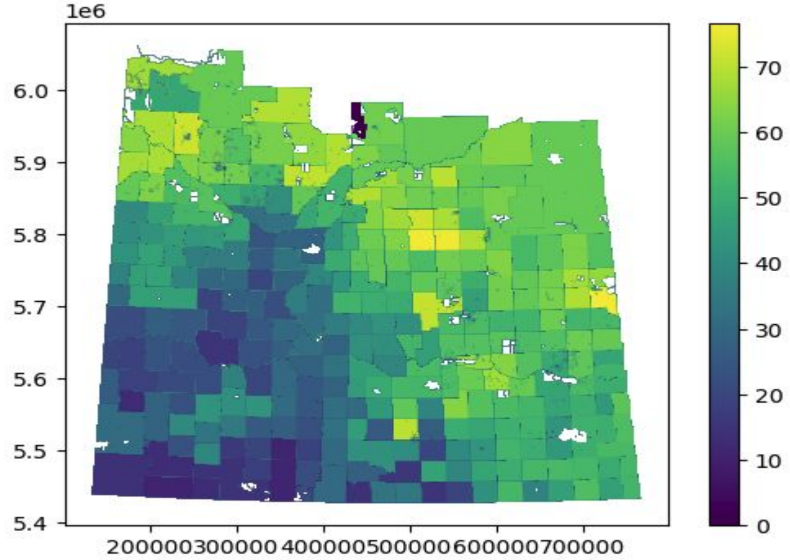
In [57]:
```
df[df['Spring Wheat'] > 100]
```

Out[57]:

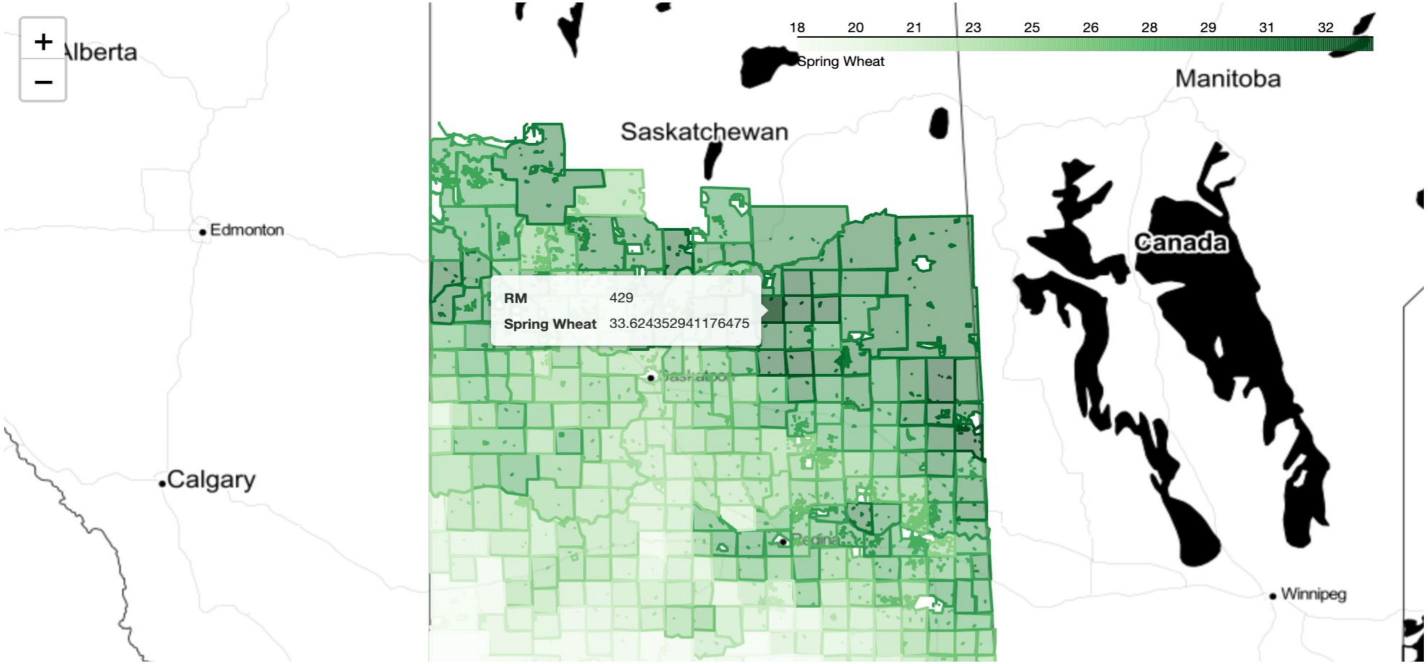
	Year	RM	Winter Wheat	Canola	Spring Wheat	Mustard	Durum	Sunflowers	Oats	Lentils	Peas	Barley	Fall Rye	Canary Seed	Spring Rye	Tame Hay	Flax	Chickpeas
16215	2018	314	NaN	33.0	198.0	NaN	NaN	NaN	41.0	29.033333	35.0	43.5	NaN	NaN	NaN	NaN	25.0	NaN

Spring Wheat Yield and GIS Analysis





Spring Wheat per RM



Methodology

Time Series Analysis

- A time series is a sequence of data points that occur in successive order over some period of time
  - Auto-Regressive Model
  - ARIMA
  - XGBoost

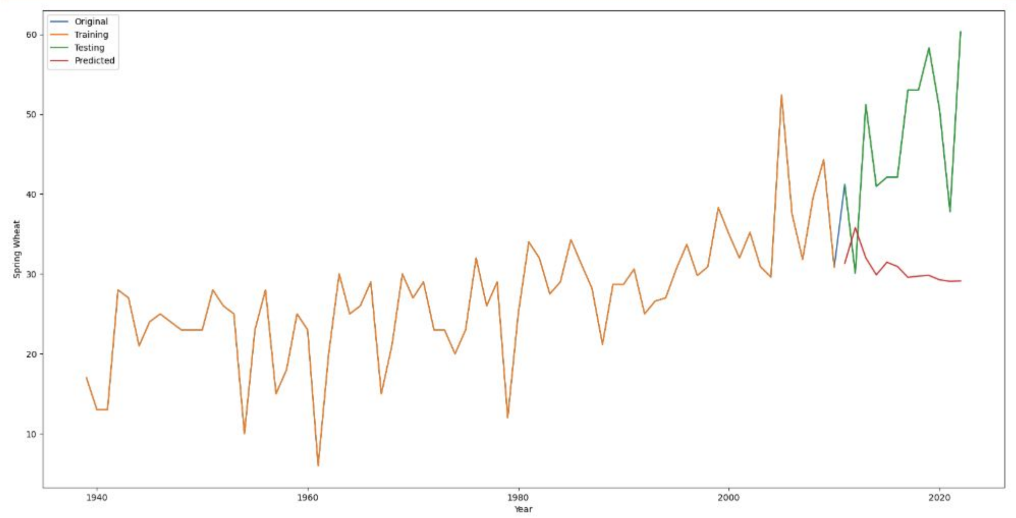
Methodology

Time Series Analysis - AR

AR(p) is a simple but powerful model that assumes the value of a variable at a given time point depends linearly on its past values. In other words, it assumes that the current value of a variable can be predicted based on its own previous values.

$$X(t) = c + \varphi_1 X(t-1) + \varphi_2 X(t-2) + \ldots + \varphi_p X(t-p) + \varepsilon(t)$$

- $X(t)$: the variable of interest at time t
- c: constant term
- $\varphi_1, \varphi_2, \ldots, \varphi_p$ are the autoregressive coefficients
- $X(t-1), X(t-2), \ldots, X(t-p)$: lagged values
- $\varepsilon(t)$: random error term.



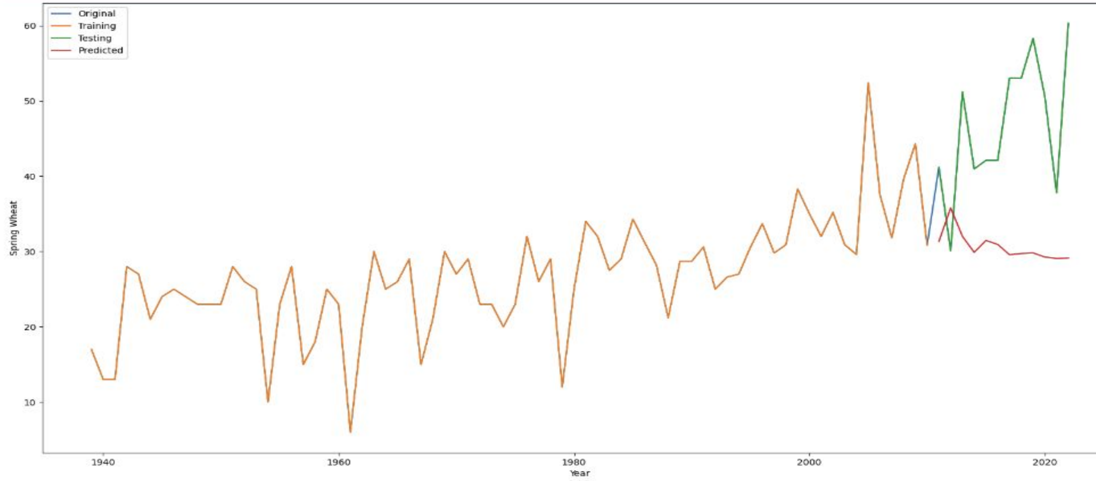
Mean Absolute Error: 17.003325562692265
Root Mean Squared Error 18.848489682492264

Methodology

Time Series Analysis - ARIMA

ARIMA(p,d,q) is a popular and widely used model for time series forecasting and analysis. Three main components:

- Autoregressive (p) - correlation current time value and historical lagged values
- Integrated (d) - differencing values to make it stationary
- Moving Average (q) - The moving average component represents the relationship between an observation in the time series and the residual errors from a moving average model applied to lagged observations. It captures random noise and shocks in data




```
Mean Absolute Error: 17.003325562692265
Root Mean Squared Error 18.848489682492264
```

Methodology

Time Series Analysis - XGBoost

XGBoost can be applied to time series analysis by transforming the problem into a supervised learning task. The basic idea is to use lagged observations of the target variable and other relevant features as input to predict the target variable at a future time step. This transformation enables XGBoost to leverage its ability to capture complex relationships and make accurate predictions.

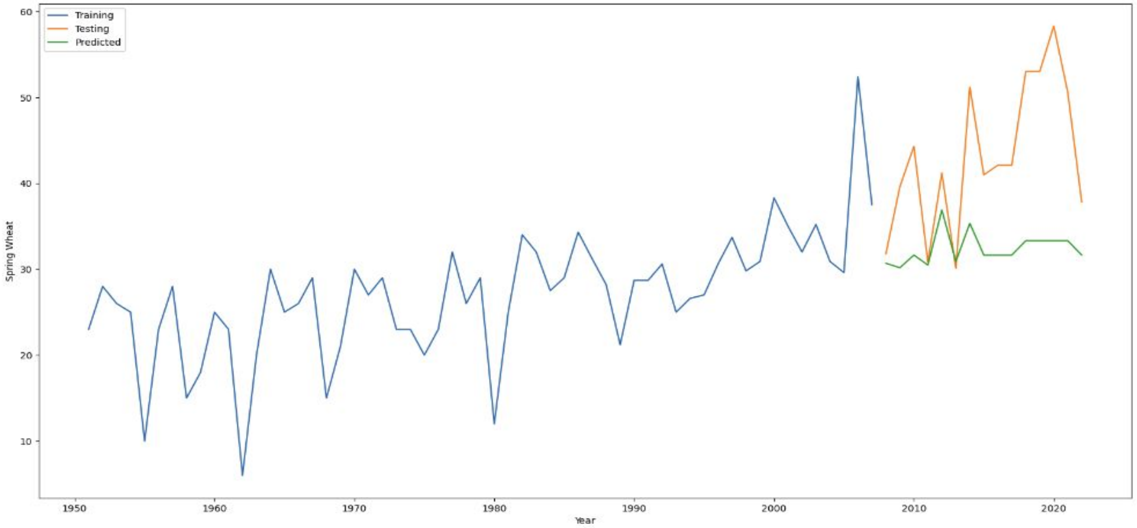
Here are the general steps involved in applying XGBoost to time series analysis:

1. Data Preparation: Arrange the time series data in a tabular format with columns representing lagged observations and features. The target variable should be shifted to represent future values.
2. Feature Engineering: Create relevant lagged features, such as lagged values of the target variable and other variables that may have predictive power. Additionally, you can incorporate time-based features like day of the week or month.
3. Train-Test Split: Divide the dataset into training and testing sets, ensuring that the order of observations is preserved. Typically, the more recent data is kept for testing to evaluate the model's performance on unseen data.
4. XGBoost Modeling: Build an XGBoost model using the training data. Specify the appropriate objective function, hyperparameters, and performance metrics based on your specific time series problem.
5. Model Training: Train the XGBoost model on the training set, optimizing the objective function and minimizing the chosen loss function.
6. Model Evaluation: Evaluate the trained model on the test set using appropriate evaluation metrics for time series forecasting, such as mean absolute error (MAE) or root mean squared error (RMSE).
7. Model Fine-tuning: Experiment with different hyperparameter values, feature selection techniques, or model architectures to improve performance. Techniques like cross-validation can be applied for hyperparameter tuning.
8. Forecasting: Once the model is trained and validated, you can use it to make predictions on unseen data or forecast future values of the time series.

Methodology

Time Series Analysis

- XGBoost



MAE: 13.649521896362304
RMSE: 15.73299541620606

Methodology

K-means

- Silhouette Score Method

Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.

1: Means clusters are well apart from each other and clearly distinguished.

0: Means clusters are indifferent, or we can say that the distance between clusters is not significant.

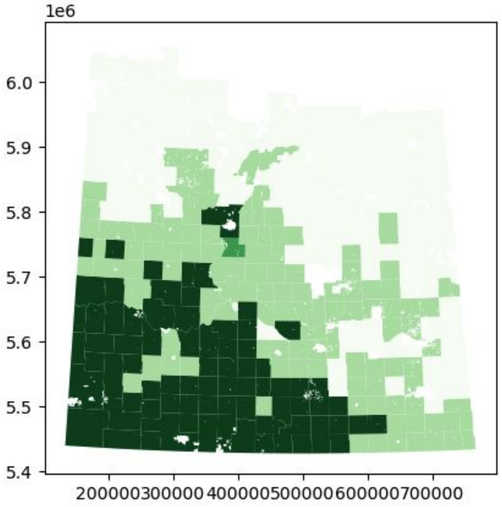
-1: Means clusters are assigned in the wrong way.

Methodology

Clustered RM using K-means clustering method

```
df_4
```

RM	Spring Wheat Mean	Spring Wheat Std	Cluster_4	Cl_4
1	47.076667	6.511628	0	1
2	45.938889	6.218477	0	1
3	45.365000	5.569120	0	1
4	42.921000	5.556423	0	1
5	37.102000	5.937401	0	1
...	...	...	...	...
520	49.314000	7.153366	2	0
555	58.235714	9.897336	2	0
561	52.676000	9.561346	2	0
588	52.661000	6.508786	2	0
622	55.379000	15.334409	2	0



Thank you