Crop Yield Prediction

목차

1. 문제 정의 및 데이터 선정

2. 데이터 전처리 : 정규화, 변수선택 등

3. 예측모델링 : 모델 적용 및 학습

4. 성능평가

5. 결과해석 및 문제해결 방안 도출

문제정의 및 데이터 선정

문제의 정의

이 프로젝트의 목적은 강우량, 살충제 사용량, 평균 기온과 같은 환경 요인을 기반으로 작물 수확량을 예측하는 것입니다. 이 정보는 농업의 의사 결정에 매우 중요할 수 있으며, 식량 생산과 지속 가능성을 최적화하기 위해 더 나은 계획과 자원 배분을 허용할 수 있습니다.

데이터 선정

이 데이터는 7개의 열이 있는 28,242 개의 항목으로 구성되어 있습니다.
열은 여러국가의 농업생산 및 환경요이노가 관련된 다양한 특징을 나타냅니다.

데이터 출처:
https://www.kaggle.com/datasets/mrigaankjaswal/crop-yield-prediction-dataset/data

문제정의 및 데이터 선정

데이터 소개

- Area (Country): 이 열에는 다양한 국가 또는 지역을 나타내는 범주형 데이터가 들어 있습니다. 누락된 값이 없는 객체 유형입니다.
- Item:: 이 열은 농산물의 유형을 나열하며, 객체로 저장됩니다. 여기의 데이터는 어떤 작물이나 항목이 분석되고 있는지를 나타냅니다.
- Year: 데이터가 기록된 연도를 나타내는 정수 열입니다. 여러 연도를 포괄하며 시간 경과에 따른 변화를 추적합니다.
- hg/ha_yield: 이 열에는 헥타르당 헥토그램(hg/ha)으로 작물 수확량을 나타내는 정수 데이터가 들어 있습니다. 이는 농업 생산성의 핵심 성과 지표입니다.
- average_rain_fall_mm_per_year: 이는 각 국가의 연평균 강수량을 밀리미터 단위로 측정하는 플로트 열입니다. 강수량은 작물 수확량에 영향을 미치는 중요한 요소입니다.
- pesticides_tonnes: 톤 단위로 측정한 살충제 사용량을 나타내는 플로트 열입니다. 살충제 사용은 작물 수확량에 영향을 미칠 수 있지만 과도한 사용은 환경 및 건강에 영향을 미칠 수 있습니다.
- avg_temp: 이 열에는 해당 연도의 국가 평균 기온을 섭씨로 나타낸 float 데이터가 들어 있습니다. 기온은 농업 생산성에 영향을 미치는 또 다른 중요한 요소입니다.

문제정의 및 데이터 선정

데이터 탐색

데이터

	Area	Item	Year	hg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
0	Albania	Maize	1990	36613	1485.0	121.0	16.37
1	Albania	Potatoes	1990	66667	1485.0	121.0	16.37
2	Albania	Rice, paddy	1990	23333	1485.0	121.0	16.37
3	Albania	Sorghum	1990	12500	1485.0	121.0	16.37
4	Albania	Soybeans	1990	7000	1485.0	121.0	16.37

데이터 유형

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28242 entries, 0 to 28241
Data columns (total 7 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Area                            28242 non-null  object
 1   Item                            28242 non-null  object
 2   Year                            28242 non-null  int64
 3   hg/ha_yield                     28242 non-null  int64
 4   average_rain_fall_mm_per_year   28242 non-null  float64
 5   pesticides_tonnes               28242 non-null  float64
 6   avg_temp                        28242 non-null  float64
dtypes: float64(3), int64(2), object(2)
memory usage: 1.5+ MB
```

데이터 결측값

Area	0
Item	0
Year	0
hg/ha_yield	0
average_rain_fall_mm_per_year	0
pesticides_tonnes	0
avg_temp	0

문제정의 및 데이터 선정

데이터 탐색

object 컬럼 정보

	Area	Item
count	28242	28242
unique	101	10
top	India	Potatoes
freq	4048	4276

데이터 통계정보

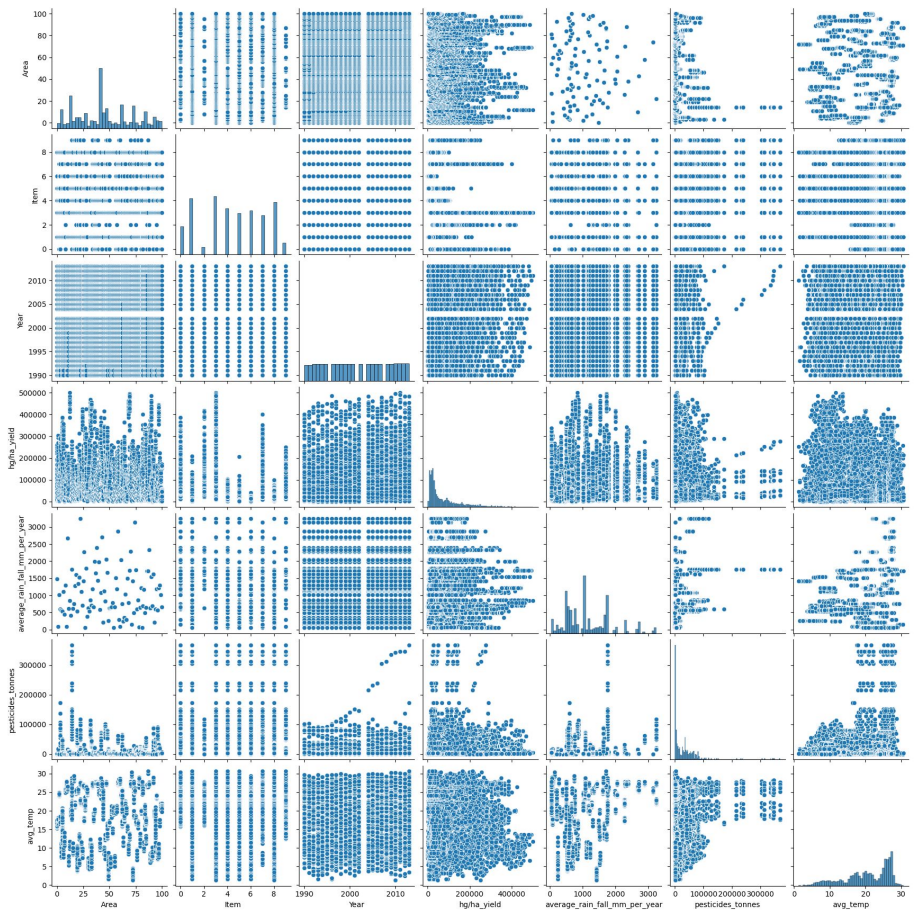
	Year	hg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
count	28242.000000	28242.000000	28242.00000	28242.000000	28242.000000
mean	2001.544296	77053.332094	1149.05598	37076.909344	20.542627
std	7.051905	84956.612897	709.81215	59958.784665	6.312051
min	1990.000000	50.000000	51.00000	0.040000	1.300000
25%	1995.000000	19919.250000	593.00000	1702.000000	16.702500
50%	2001.000000	38295.000000	1083.00000	17529.440000	21.510000
75%	2008.000000	104676.750000	1668.00000	48687.880000	26.000000
max	2013.000000	501412.000000	3240.00000	367778.000000	30.650000

상관관계

	Area	Item	Year	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
Area	1.000000	0.003169	0.003421	-0.233549	-0.313573	-0.045739
Item	0.003169	1.000000	0.001465	-0.062580	0.048070	-0.078257
Year	0.003421	0.001465	1.000000	-0.003798	0.140930	0.014409
average_rain_fall_mm_per_year	-0.233549	-0.062580	-0.003798	1.000000	0.180984	0.313040
pesticides_tonnes	-0.313573	0.048070	0.140930	0.180984	1.000000	0.030946
avg_temp	-0.045739	-0.078257	0.014409	0.313040	0.030946	1.000000

문제정의 및 데이터 선정

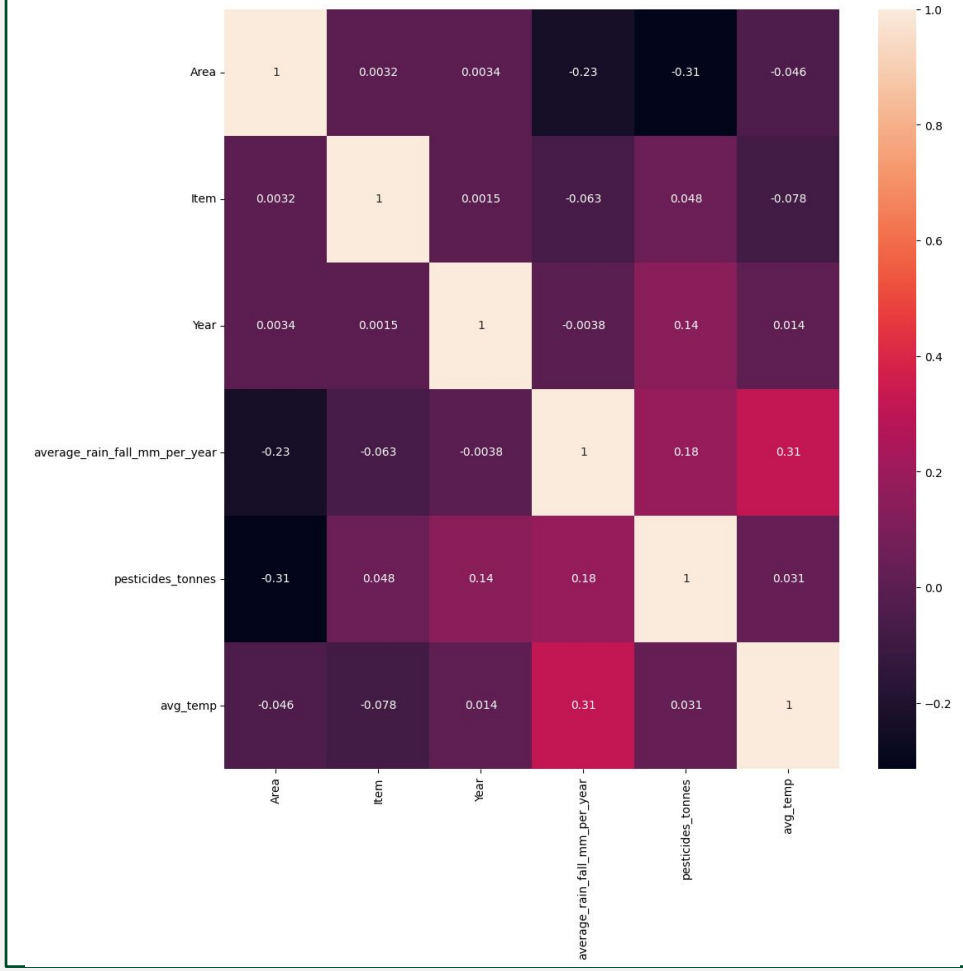
데이터 시각화



- 변수 avg_temp의 경우 우측 꼬리가 긴 분포를 보입니다. 이는 평균 기온이 대부분 낮은 값에 집중되어 있지만, 일부 데이터는 매우 높은 값을 가진다는 것을 나타냅니다.
- average_rain_fall_mm_per_year와 high_yield 간의 산점도에서 어느 정도 상관관계가 있는 것처럼 보입니다. 즉, 강수량이 많을수록 수확량이 증가할 가능성이 있습니다. 반면, Year와 avg_temp 간의 산점도에서는 특정 패턴을 확인하기 어렵습니다. 이는 두 변수 간에 뚜렷한 상관관계가 없음을 의미할 수 있습니다.
- pesticides_tonnes와 high_yield 간에는 약간의 양의 상관관계가 있는 것처럼 보입니다. 즉, 농약 사용량이 많을수록 수확량이 높아질 가능성이 있습니다.

문제정의 및 데이터 선정

데이터 시각화



- average_rain_fall_mm_per_year와 avg_temp 사이에는 0.31의 상관계수가 있어 약한 양의 상관관계가 있음을 나타냅니다. 즉, 강수량이 많아질수록 평균 기온이 다소 증가하는 경향이 있습니다.
- pesticides_tonnes와 average_rain_fall_mm_per_year의 상관계수는 0.18로, 농약 사용량과 강수량 사이에 약한 양의 상관관계가 있습니다.
- Area와 pesticides_tonnes의 상관계수는 -0.31로, 면적이 증가할수록 농약 사용량이 감소하는 경향이 있음을 보여줍니다.

데이터 전처리: 정규화, 변수선택 등

라벨링

Label Encoding

Label Encoding은 범주형 데이터를 숫자로 변환하는 방법 중 하나입니다. 범주형 변수는 모델에 직접 입력될 수 없으므로 이를 숫자로 변환해야 하는데, 라벨 인코딩은 각 범주(카테고리)에 고유한 정수를 할당하여 변환하는 방식입니다.

주요 특징

- **단순한 변환**: 각 범주에 정수 값을 할당하여 데이터를 손쉽게 변환할 수 있습니다.
- **순서 정보 부여**: 정수로 변환함으로써 데이터에 암묵적으로 순서를 부여하게 됩니다.
- **사용할 때 주의할 점**: 순서가 없는 범주형 데이터를 라벨 인코딩하면 모델이 이 순서를 의미 있게 해석할 수 있어 성능이 떨어질 수 있습니다. 이를 방지하려면 순서가 없는 범주형 데이터에는 원-핫 인코딩(One-Hot Encoding)을 사용하는 것이 좋습니다.

```
1 from sklearn.preprocessing import LabelEncoder
2
3 encoder = LabelEncoder()
4 df['Area'] = encoder.fit_transform(df['Area'])
5 df['Item'] = encoder.fit_transform(df['Item'])
```

데이터 전처리: 정규화, 변수선택 등

라벨링

One-Hot Encoding

One-Hot Encoding은 범주형 데이터를 숫자로 변환하는 방법 중 하나로, 각 범주를 고유한 이진 벡터로 변환하는 방식입니다. 라벨 인코딩이 각 범주에 정수를 할당하는 것과 달리, 원핫 인코딩은 범주를 열(피처)로 나누고 해당 범주에 해당하는 값만 1로 표시하며, 나머지는 모두 0으로 처리합니다.

주요 특징

- **이진 벡터 변환:** 범주형 데이터의 각 값이 서로 독립적인 이진 벡터로 변환됩니다.
- **순서 정보 없음:** 원핫 인코딩은 범주 간에 순서나 크기를 부여하지 않으므로, 범주 간 순서 정보가 없을 때 유용합니다. 이 점은 라벨 인코딩과의 중요한 차이입니다. 순서가 없는 범주형 데이터를 라벨 인코딩으로 변환하면 순서 정보가 생기는데, 이는 모델 성능에 부정적인 영향을 줄 수 있습니다.
- **차원의 증가:** 범주의 수가 많을수록 변환된 데이터의 차원이 급격히 증가합니다.

범주간에 순서나 크기를 부여 하지 않으므로
One-Hot Encoding 선택했습니다.

```
1 df = pd.get_dummies(df)
2
```

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28242 entries, 0 to 28241
Columns: 116 entries, Year to Item_Yams
dtypes: bool(111), float64(3), int64(2)
memory usage: 4.1 MB
```

데이터 전처리: 정규화, 변수선택 등

데이터 분리

Train Test Split

`train_test_split` 함수는 Scikit-learn의 `model_selection` 모듈에서 가져옵니다.
이 함수는 데이터를 학습 세트와 테스트 세트로 나누는 데 사용됩니다.

```
1 from sklearn.model_selection import train_test_split
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- `test_size=0.2`는 데이터의 20%를 테스트 세트로 사용할 것임을 나타냅니다.
- `random_state=42`는 결과의 재현성을 보장하기 위해 랜덤 시드를 설정하는 데 사용됩니다.
  즉, 같은 코드를 여러 번 실행하더라도 항상 같은 결과를 얻을 수 있습니다.

변수 선택

모든 변수선택

상관관계가 매우 높거나 예측에 필요한 변수들로 판단하여 모두 사용

데이터 전처리: 정규화, 변수선택 등

정규화

Standardization

표준정규화 (Standardization) 또는 **Z-스코어 정규화 (Z-score normalization)**는 데이터를 평균이 0이고 표준편차가 1인 분포로 변환하는 과정입니다. 이는 각 데이터 포인트가 평균에서 얼마나 떨어져 있는지를 나타내며, 주로 머신러닝 모델에서 입력 특성의 스케일 차이를 조정할 때 사용됩니다.

표준정규화의 주요 특징:

1. **Z-스코어 계산**:

$$z = \frac{x - \mu}{\sigma}$$

```
[37]  1 from sklearn.preprocessing import StandardScaler
      2 scaler = StandardScaler()
      3 X_train_s = scaler.fit_transform(X_train)
      4 X_test_s = scaler.transform(X_test)
```

2. **데이터의 분포**:
   - 표준정규화를 통해 데이터의 분포는 평균이 0, 표준편차가 1인 정규분포(가우시안 분포)로 변환됩니다. 따라서, 데이터 포인트는 평균보다 작거나 큰 경우에 따라 음수 또는 양수의 값을 가질 수 있습니다.
3. **이점**:
   - 머신러닝 알고리즘에서 특성의 스케일이 다르면, 거리 기반 알고리즘(예: KNN, SVM 등)이나 경량화 모델(예: 선형 회귀)에서 성능이 저하될 수 있습니다. 표준정규화를 통해 이 문제를 해결할 수 있습니다.
4. **데이터의 이상치 영향**:
   - 표준정규화는 평균과 표준편차를 기반으로 하므로, 데이터에 이상치가 존재할 경우 그 값에 영향을 받을 수 있습니다. 이상치가 많은 경우, Robust Scaler와 같은 다른 방법을 사용하는 것이 좋습니다.

데이터 전처리: 정규화, 변수선택 등

정규화

Min-Max Normalization

최대-최소 정규화 (Min-Max Normalization)는 데이터 전처리 기법 중 하나로, 데이터의 값을 특정 범위(주로 0과 1 사이)로 변환하는 방법입니다. 이 기법은 특히 머신러닝 모델에 데이터를 입력하기 전에 데이터를 스케일링하여 성능을 향상시키는 데 사용됩니다.

최대-최소 정규화의 주요 특징:

1. **정규화 공식**:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}$$

```python
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train_m = scaler.fit_transform(X_train)
X_test_m = scaler.transform(X_test)
```

2. **데이터의 분포**:
   - 최소-최대 정규화를 통해 데이터의 분포는 0과 1 사이로 변환되며, 각 데이터 포인트는 원본 데이터의 상대적인 위치를 유지합니다. 데이터의 패턴이나 형태는 변하지 않지만, 모든 값이 동일한 범위 내에 존재하게 됩니다.
3. **이점**:
   - 특히 신경망과 같은 많은 머신러닝 알고리즘에서 각 특성의 스케일이 비슷해야 빠르고 안정적으로 수렴할 수 있습니다.
   - 모든 특성이 동일한 스케일로 조정되어 모델이 특정 특성에 과도하게 의존하지 않도록 합니다.
4. **데이터의 이상치 영향**:
   - **왜곡된 정규화**: 최소-최대 정규화는 데이터셋 내의 최소값과 최대값을 기준으로 하므로, 이상치가 존재할 경우 정규화된 값이 왜곡될 수 있습니다. 예를 들어, 데이터의 최대값이 이상치인 경우, 대부분의 데이터가 0에 가까운 값으로 변환되어 유용한 정보를 잃게 될 수 있습니다.
   - **정보 손실**:이로 인해, 이상치가 많은 데이터셋에서는 정규화 후의 데이터가 원본 데이터의 분포를 제대로 반영하지 못할 수 있으며, 이는 모델의 성능을 저하시킬 수 있습니다.

예측 모델링: 모델 적용 및 학습

Linear Regression

선형 회귀 모델(Linear Regression) 은 주어진 데이터에 가장 잘 맞는 직선을 찾는 과정

- **장점:**
  - 해석이 간단하고 명확합니다.
  - 계산 비용이 적어 빠르게 수행됩니다.
  - 적은 데이터로도 모델링이 가능합니다.
- **단점:**
  - 독립 변수와 종속 변수 간의 관계가 반드시 선형이어야 한다는 가정이 있습니다.
  - 과적합(overfitting) 문제를 일으킬 수 있습니다.
  - 노이즈가 많은 데이터에서는 예측 성능이 떨어질 수 있습니다.

```
1 from sklearn.linear_model import LinearRegression
2 # 모델 구축
3 lr = LinearRegression()
4 lr.fit(X_train, y_train)
```

예측 모델링: 모델 적용 및 학습

OLS(Ordinary Least Squares)

OLS(Ordinary Least Squares)는 가장 기본적인 결정론적 선형 회귀 방법으로 잔차제곱합(RSS: Residual Sum of Squares)를 최소화하는 가중치(β1, β2 ...) 구하는 방법

- **장점:**
  - 해석 용이: 회귀 계수, p-value 등 다양한 통계 지표를 쉽게 확인할 수 있습니다.
  - 통계적 분석 제공: F-검정, t-검정 등 다양한 통계 검정 결과를 포함합니다.
  - 직관적인 사용법: 간단한 API로 데이터 분석을 쉽게 수행 가능합니다.
  - 추가 분석 기능: 회귀 진단과 잔차 분석 지원합니다.
- **단점:**
  - 대용량 데이터에서 비효율적: 대규모 데이터 처리 성능이 낮습니다.
  - 머신러닝 기능 부족: 교차 검증 및 자동화된 모델 튜닝 기능이 제한적 입니다.
  - 가정 의존성: 선형성 및 정규성 가정이 위배되면 성능 저하됩니다.
  - 비선형 관계에 부적합: 복잡한 비선형 데이터 처리에 한계입니다.

```
1 import statsmodels.api as sm
2 model = sm.OLS(y, X)
3 model = model.fit()
```

예측 모델링: 모델 적용 및 학습

분산분석 (ANOVA)

분산분석 (ANOVA: Analysis of Variance)는 여러 집단 간의 평균을 비교하여, 집단 간의 차이가 통계적으로 유의미한지를 판단하는 통계 기법

- **장점:**
  - 효율적 집단 비교: 여러 집단 간 평균을 한 번에 비교할 수 있습니다.
  - 통계적 유의성 검정: F-통계량을 통해 집단 간 차이가 우연인지 여부를 판단 가능합니다.
  - 복잡한 실험 설계에 유용: 이원분산분석 등으로 여러 요인의 상호작용을 분석할 수 있습니다.
- **단점:**
  - 차이 위치 불명확: 집단 간 차이가 발생한 구체적인 위치를 알기 어려워 사후검정(Post-hoc test)이 필요합니다.
  - 정규성 가정: 데이터가 정규분포를 따르고, 집단 간 분산이 동일해야 하는 가정이 필요합니다.
  - 비선형 관계 분석 불가: 집단 간 비선형 관계를 분석하기에는 부적합입니다.

```
model_2 = ols('hg_ha_yield ~ Area + Item + Year + average_rain_fall_mm_per_year + pesticides_tonnes + avg_temp', data=data)
model_2_trained = model_2.fit()
```

예측 모델링: 모델 적용 및 학습

분산분석 (ANOVA)

- **R-squared:** 0.083. 전체 데이터 변동성 중 회귀 모델로 설명 가능한 비율을 나타냅니다. 여기서는 8.3%로, 설명력이 낮습니다.
- **Adj. R-squared:** 0.083. R-squared를 모델의 복잡도에 맞게 조정한 값입니다. 여기서는 변하지 않았습니다.
- **F-statistic:** 424.3 ,F-통계량은 전체 모델이 유의미한지를 검정하는 통계량입니다. 이 값이 클수록 모델이 통계적으로 유의미할 가능성이 높습니다.
- **Prob(F-statistic):**모델이 통계적으로 유의미한지 확인하는 p-value입니다. 0.000이므로, 매우 유의미한 모델임을 나타냅니다.
- **로그 우도 (Log-Likelihood)**:** -3.594e+05, 로그 우도는 모델이 주어진 데이터를 설명하는 정도를 나타냅니다. 값이 낮을수록 모델의 적합도가 낮습니다.
- **AIC & BIC**
  - **AIC (Akaike Information Criterion):** 718800.5. 모델의 적합도를 평가하는 지표로, 값이 낮을수록 좋은 모델입니다.
  - **BIC (Bayesian Information Criterion):** 718905.0. BIC도 AIC와 유사하게 적합도를 평가하나, 더 엄격한 기준을 적용합니다.

- **모델은 통계적으로 유의미하지만, R-squared 값이 매우 낮아 설명력이 약한 모델입니다. 주요 변수들은 모두 유의미한 영향을 미치지만, 잔차의 정규성이나 자기상관 문제를 고려할 필요가 있습니다.**

OLS Regression Results

Dep. Variable:	hg_ha_yield	**R-squared:**	0.083
Model:	OLS	**Adj. R-squared:**	0.083
Method:	Least Squares	**F-statistic:**	424.3
Date:	Tue, 01 Oct 2024	**Prob (F-statistic):**	0.00
Time:	01:27:09	**Log-Likelihood:**	-3.5940e+05
No. Observations:	28242	**AIC:**	7.188e+05
Df Residuals:	28235	**BIC:**	7.189e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.881e+06	1.39e+05	-13.533	0.000	-2.15e+06	-1.61e+06
Area	92.6313	19.395	4.776	0.000	54.617	130.646
Item	-7688.0394	184.793	-41.604	0.000	-8050.242	-7325.837
Year	1009.2439	69.479	14.526	0.000	873.061	1145.426
average_rain_fall_mm_per_year	2.9772	0.745	3.999	0.000	1.518	4.437
pesticides_tonnes	0.1032	0.009	11.891	0.000	0.086	0.120
avg_temp	-1929.4205	80.991	-23.823	0.000	-2088.167	-1770.674

Omnibus:	7796.832	**Durbin-Watson:**	0.887
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	17975.231
Skew:	1.577	**Prob(JB):**	0.00
Kurtosis:	5.307	**Cond. No.**	2.02e+07

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.02e+07. This might indicate that there are strong multicollinearity or other numerical problems.

예측 모델링: 모델 적용 및 학습

Multilayer Perceptron

MLP는 각 층의 노드가 다음 층의 모든 노드와 연결된 완전 연결 신경망(fully connected network)입니다

- **장점**:
  - 비선형 문제 해결: 활성화 함수 덕분에 MLP는 선형 분류로는 해결할 수 없는 복잡한 비선형 문제를 처리할 수 있습니다.
  - 유연성: 여러 층을 쌓음으로써 복잡한 패턴을 학습할 수 있습니다.
  - 다양한 문제에 적용 가능: 회귀, 분류, 패턴 인식 등 다양한 문제에 적용 가능합니다.
- **단점**:
  - 대규모 데이터 요구: 은닉층이 많고 노드가 많을수록 더 많은 데이터와 연산 자원이 필요합니다.
  - 과적합 위험: 모델이 너무 복잡하면 학습 데이터에 과적합(overfitting)할 위험이 있습니다. 이를 방지하기 위해 드롭아웃(dropout)이나 정규화(regularization)와 같은 기법이 사용됩니다.
  - 해석의 어려움: 모델이 복잡해질수록 가중치나 편향의 의미를 직관적으로 해석하기 어렵습니다.

```
from sklearn.neural_network import MLPRegressor
mlp = MLPRegressor(activation='relu', alpha=1E-3, batch_size=25,
                   hidden_layer_sizes=(50, 25), max_iter=1000, learning_rate_init=0.001,
                   solver='adam', verbose=10)
```

예측 모델링: 모델 적용 및 학습

RandomForest Regression

랜덤 포레스트(RandomForest)는 결정 트리 기반의 앙상블 모델로, 여러 트리의 예측을 결합하여 최종 결과를 도출합니다

- **장점:**
  - 높은 예측 성능: 다양한 트리의 예측을 평균함으로써, 보다 정확한 결과를 얻을 수 있습니다.
  - 과적합 방지: 무작위 샘플링과 여러 트리의 결합으로 과적합 위험이 적습니다.
  - 비선형 데이터 처리 가능: 복잡한 비선형 데이터에서도 우수한 성능을 보입니다.
  - 특성 중요도 평가: 랜덤 포레스트는 각 특성의 중요도를 제공하여, 중요한 변수들을 식별하는 데 도움을 줍니다.
- **단점:**
  - 해석의 어려움: 개별 트리 모델의 해석이 가능하지만, 여러 트리의 결합으로 이루어진 랜덤 포레스트 모델은 직관적으로 해석하기 어렵습니다.
  - 느린 예측 속도: 많은 트리를 사용하기 때문에 학습과 예측 시간이 길어질 수 있습니다.
  - 메모리 소모: 많은 트리를 저장하고 사용하기 때문에 메모리 소모가 큽니다.

```
1 from sklearn.ensemble import RandomForestRegressor
2
3 rf = RandomForestRegressor()
```

성능평가

정규화X

		MSE	RMSE	R2(결정계수)
Linear Regression	Training	6615261227	81334.2	0.08227
	Testing	6642537610	81501.8	0.08425
OLS	Training	6659504815	81605.8	0.076133
	Testing	6679172069	81726.2	0.07920
Multilayer Perceptron	Training	5743673395	75787.0	0.20319
	Testing	5851872094	76497.5	0.19325
RandomForest Regression	Training	14120089	3757.7	0.99804
	Testing	102710008	10134.6	0.98584

성능평가

Standardization

		MSE	RMSE	R2(결정계수)
Linear Regression	Training	6615261227	81334.3	0.08227
	Testing	6642537610	81501.8	0.08425
OLS	Training	12552148304	112036.4	-0.74135
	Testing	12640371711	112429.4	-0.74262
Multilayer Perceptron	Training	1744630167	41768.8	0.75797
	Testing	1799995995	42426.4	0.75185
RandomForest Regression	Training	13918124	3730.7	0.99806
	Testing	105061012	10249.9	0.98552

성능평가

Min-Max Normalization

		MSE	RMSE	R2(결정계수)
Linear Regression	Training	6615261227	81334.3	0.082271
	Testing	6642537611	81501.8	0.08425
OLS	Training	7325042090	85586.5	-0.01620
	Testing	7378071287	85895.7	-0.01715
Multilayer Perceptron	Training	6438301378	80239.0	0.10682
	Testing	6478702678	80490.4	0.10684
RandomForest Regression	Training	14258693	3776.0	0.99802
	Testing	101326810	10066.1	0.98603

결과해석 및 문제해결 방안 도출

결과해석

- **Linear Regression**
  - 정규화 여부와 상관없이 R^2 값이 매우 낮음 (~0.08), 즉 모델이 목표 변수(작물 수확량)를 잘 설명하지 못함.
  - MSE와 RMSE 값이 매우 크므로 예측의 오차가 큼.
  - 스케일링에 의한 성능 향상은 없으며, 이는 선형 회귀 모델이 해당 데이터 세트에서 적합하지 않음을 시사.

- **OLS (Ordinary Least Squares)**
  - 정규화하지 않은 경우에도 R^2가 0.07 수준으로 여전히 낮고, 스케일링이 적용된 경우 오히려 성능이 더 악화됨. (음의 R^2값)
  - 이로 인해 OLS 역시 이 데이터에 적합하지 않은 모델로 보임.

결과해석 및 문제해결 방안 도출

결과해석

- **Multilayer Perceptron (MLP)**
  - 정규화하지 않은 경우에도 R^2 값이 **0.18**로 선형 회귀나 **OLS**보다는 더 나은 성능을 보임.
  - 특히 **Min-Max Scaler**를 적용했을 때, **MSE**와 **RMSE**가 가장 낮고 R^2 값이 **0.80** 이상으로 높아짐. 이는 **MLP** 모델이 이 데이터에 적합하며 비선형 관계를 잘 잡아낸다는 것을 나타냄.

- **Random Forest Regression**
  - 모든 경우에서 매우 높은 성능을 보여주며, R^2 값이 **0.99** 이상,
    즉 거의 완벽한 예측을 수행함.
  - 스케일링에 상관없이 모델이 매우 높은 성능을 유지함.
  - 모델 응용 방안 및 효과

결과해석 및 문제해결 방안 도출

문제 해결방안 도출

1. 데이터 개선 및 확장

- 추가적인 환경 요인, 경제적 조건 등 변수를 도입해 예측 성능을 높일 수 있습니다.
- 시계열 분석(LSTM, Prophet 등)을 활용해 연도별 변화 패턴을 더 잘 파악할 수 있습니다.

2. 모델 최적화

- 랜덤 포레스트와 MLP의 하이퍼파라미터를 최적화하고, 앙상블 기법(예: Stacking)을 적용해 성능을 향상시킬 수 있습니다.
- K-fold 교차 검증으로 모델의 안정성을 확인합니다.

3. 실용적 적용

- 예측 결과를 바탕으로 자원 관리, 기후 변화 대응, 정책 결정에 활용할 수 있습니다.

4. 실시간 데이터 통합

- 실시간 기상 데이터를 추가하고 IoT 기반의 실시간 데이터 수집을 통해 예측 모델의 정확도를 높일 수 있습니다.

결과해석 및 문제해결 방안 도출

모델 응용 방안 및 효과

1. Random Forest 및 MLP 모델 적용

- Random Forest는 데이터의 복잡한 패턴을 매우 잘 학습하고 있어, 이를 통해 농업 정책 수립, 환경 변화에 따른 작물 생산량 예측, 또는 효율적인 자원 관리 등에 응용할 수 있음.
- MLP 역시 비선형 패턴을 잘 잡아내므로, 여러 환경 요인(강수량, 기온, 살충제 사용량 등)이 작물 생산에 미치는 영향을 예측할 수 있음.

2. 실질적인 응용

- 작물 수확량 예측: 다양한 국가에서의 작물 수확량을 미리 예측함으로써 농업 정책 및 자원 배분 계획을 최적화할 수 있음. 예측을 통해 기후 변화에 대한 대처 전략을 마련할 수 있음.
- 환경 영향 분석: 살충제 사용량과 기온이 작물 생산성에 미치는 영향을 분석하여 지속 가능한 농업 방안을 도출할 수 있음.
- 정책 의사결정: 농업 생산성 향상을 위한 환경 및 자원 관리 정책을 지원할 수 있으며, 이를 통해 효율적인 농업 생산이 가능해질 것임.

Thank you