```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
import pandas as pd

df = pd.read_csv("yield_df.csv")
df.head()
```

	Unnamed: 0	Area	Item	Year	hg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonne
0	0	Albania	Maize	1990	36613	1485.0	121
1	1	Albania	Potatoes	1990	66667	1485.0	121
2	2	Albania	Rice, paddy	1990	23333	1485.0	121
3	3	Albania	Sorghum	1990	12500	1485.0	121
4	4	Albania	Soybeans	1990	7000	1485.0	121

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28242 entries, 0 to 28241
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Unnamed: 0                     28242 non-null  int64
 1   Area                           28242 non-null  object
 2   Item                           28242 non-null  object
 3   Year                           28242 non-null  int64
 4   hg/ha_yield                    28242 non-null  int64
 5   average_rain_fall_mm_per_year  28242 non-null  float64
 6   pesticides_tonnes              28242 non-null  float64
 7   avg_temp                       28242 non-null  float64
dtypes: float64(3), int64(3), object(2)
memory usage: 1.7+ MB
```

```
df = df.dropna()
```

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

for col in df.select_dtypes(include=['object']).columns:
    df[col] = le.fit_transform(df[col])
```

```
df = df.drop("Unnamed: 0", axis=1)
```

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df["Area"] = le.fit_transform(df["Area"])
df["Item"] = le.fit_transform(df["Item"])
```

```
X = df.drop("hg/ha_yield", axis=1)
y = df["hg/ha_yield"]
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(
    n_estimators=100,
    random_state=42
)

model.fit(X_train, y_train)
```

```
▾     RandomForestRegressor    ⓘ ?
RandomForestRegressor(random_state=42)
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

y_pred = model.predict(X_test)

mae_rf = mean_absolute_error(y_test, y_pred)
rmse_rf = np.sqrt(mean_squared_error(y_test, y_pred))
r2_rf = r2_score(y_test, y_pred)

print("MAE :", mae_rf)
print("RMSE:", rmse_rf)
print("R2  :", r2_rf)
```

```
MAE : 3752.47717472119
RMSE: 10181.762058896022
R2  : 0.9857081655966298
```

```
from sklearn.linear_model import LinearRegression

lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
```

```
▾ LinearRegression  ⓘ ?
LinearRegression()
```

```
y_pred_lr = lr_model.predict(X_test)
```

```
mae_lr = mean_absolute_error(y_test, y_pred_lr)
rmse_lr = np.sqrt(mean_squared_error(y_test, y_pred_lr))
r2_lr = r2_score(y_test, y_pred_lr)

print("Linear Regression Results")
print("MAE :", mae_lr)
print("RMSE:", rmse_lr)
print("R2  :", r2_lr)
```

```
Linear Regression Results
MAE : 62444.31062975515
RMSE: 81501.76446544638
R2  : 0.08425172276039494
```

```
comparison = pd.DataFrame({
    "Model": ["Linear Regression", "Random Forest"],
    "MAE": [mae_lr, mae_rf],
    "RMSE": [rmse_lr, rmse_rf],
    "R2 Score": [r2_lr, r2_rf]
})

comparison
```

	Model	MAE	RMSE	R2 Score
0	Linear Regression	62444.310630	81501.764465	0.084252
1	Random Forest	3752.477175	10181.762059	0.985708