Impact of rainfall on crop production in Tamil Nadu

Co-ordinated by **Prof. SANJOY GOSWAMI**

Introduction

Rainfall plays a vital role in shaping agricultural productivity, especially in agrarian regions like Tamil Nadu. With over 50% of the state’s population dependent on agriculture, timely and adequate rainfall is crucial for crop growth, soil health, and irrigation. However, variability in monsoon patterns—ranging from droughts to floods—poses significant challenges to consistent crop yields. Understanding the relationship between rainfall trends and crop production helps in planning adaptive strategies for sustainable agriculture.

Types of Rainfall Impacts

- **Positive Impact**: **Sufficient and timely** rainfall boosts yield
- **Negative Impact**:

  - Droughts lead to crop failure.

  - Excess rainfall causes water-logging, root rot, and erosion

  - Unseasonal rainfall disrupts harvest.

Case Study – Tamil Nadu (Example Region)

Overview of rainfall pattern in Tamil Nadu (last 27 years),from **1990-2017.**

Key crops**: rice, Sorghum, Oilseeds, sugarcane, Groundnut**.

Observation: Yield trends vs. monsoon or seasonal variability.

Data and Methodology

- Data sources:

  - International Crops Research Institute for the Semi-Arid. Tropics (ICRISAT)

  - Open Government Data Platform (OGD), India.

- Techniques used:

  - **Correlation Analysis.**

  - **Time-Series Visualization.**

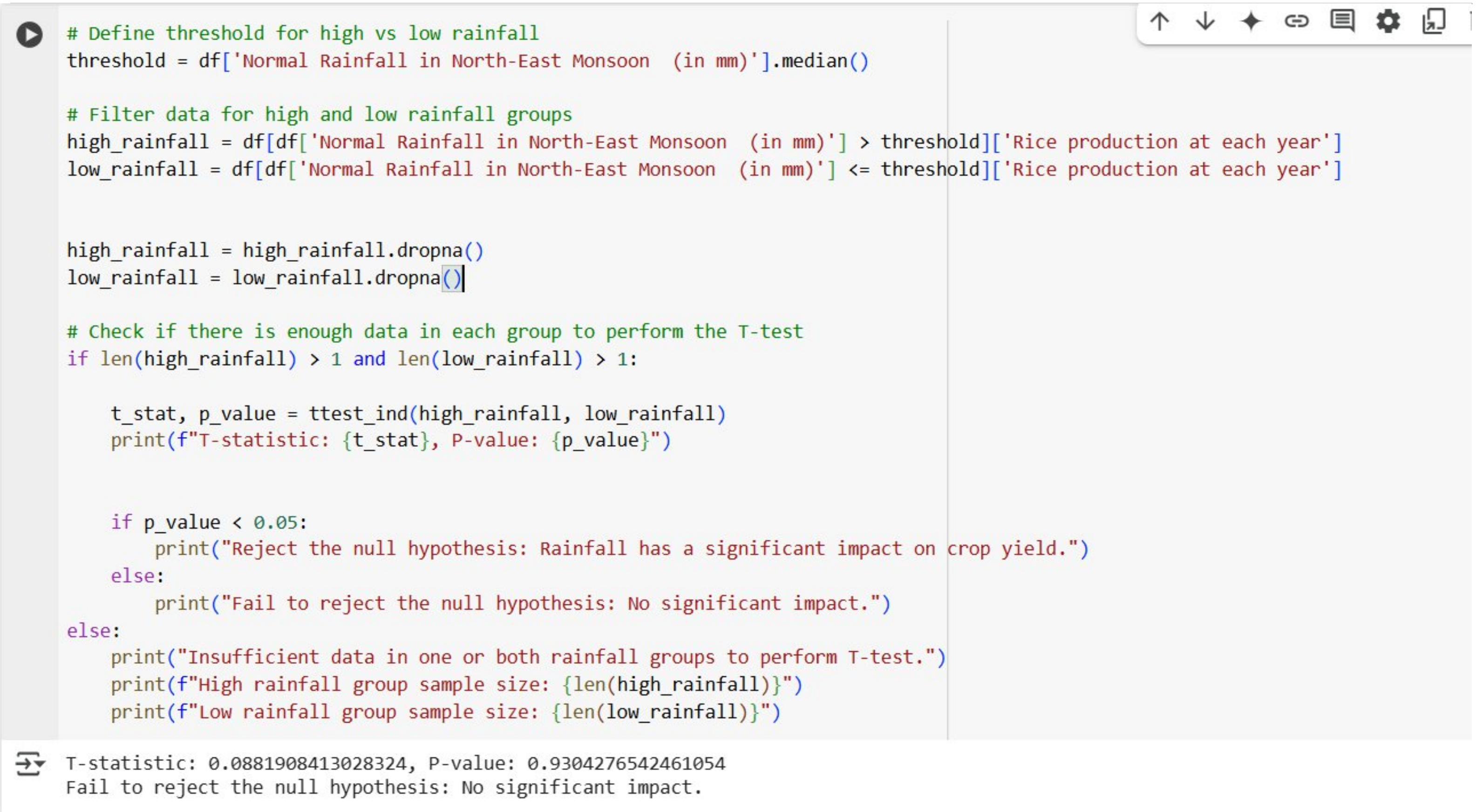
  - **Predictive Modeling.**

Correlation

Rainfall Crop	SWM	NEM	Winter	Hot Period	Annual
Rice	−0.24	+0.06	+0.31	+0.40	+0.10
Sorghum	−0.41	−0.22	+0.12	+0.28	−0.20
Oilseeds	−0.47	+0.06	**+0.74**	**+0.73**	+0.18
Sugarcane	−0.24	−0.37	+0.002	+0.055	−0.31
Groundnut	**−0.66**	−0.13	**+0.73**	**+0.80**	+0.003

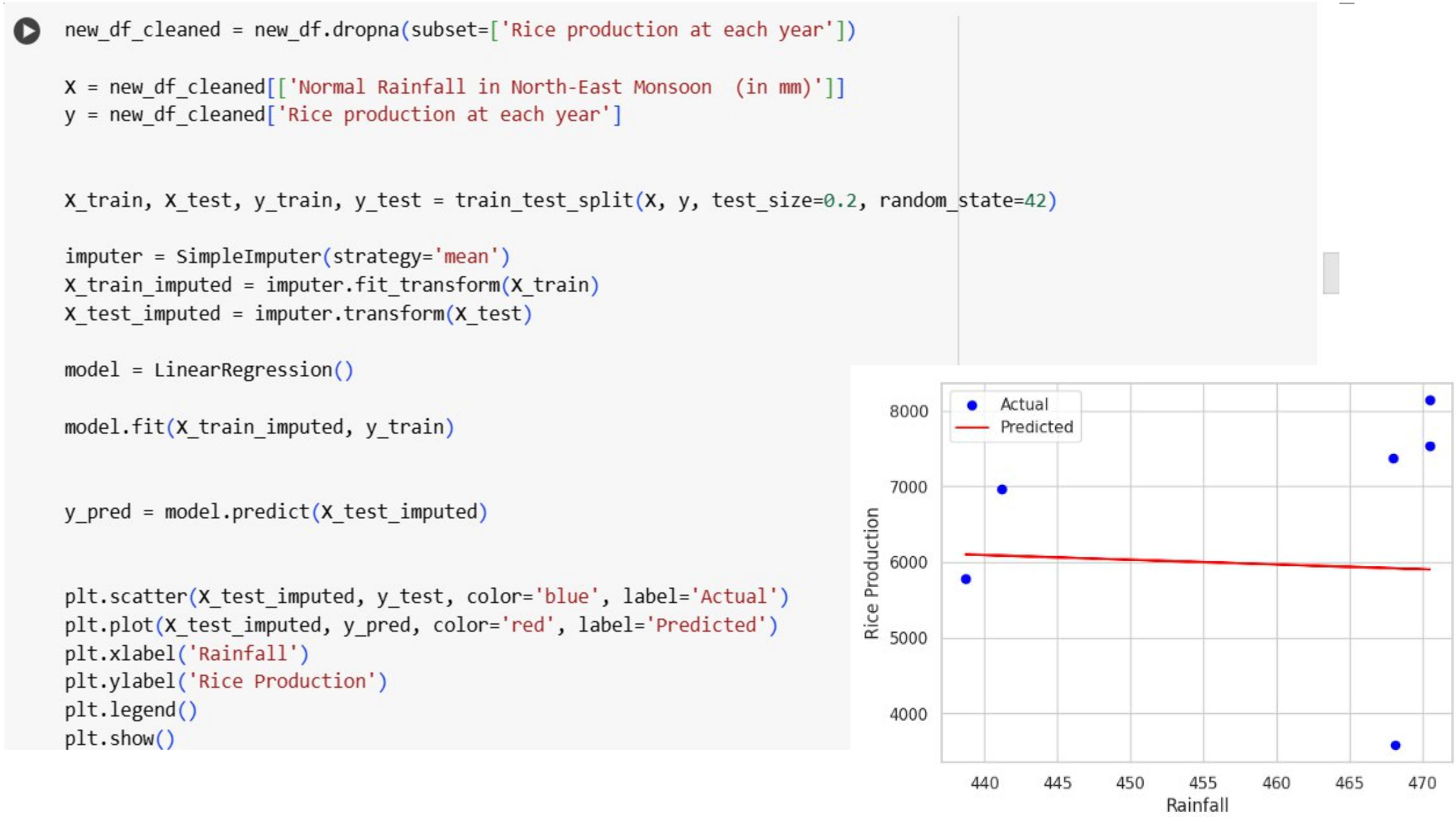
🟫 Weak or No correlation 🟥 Negative correlation ⬛ Positive correlation

Rainfall and Crop Yield – Weak Impact



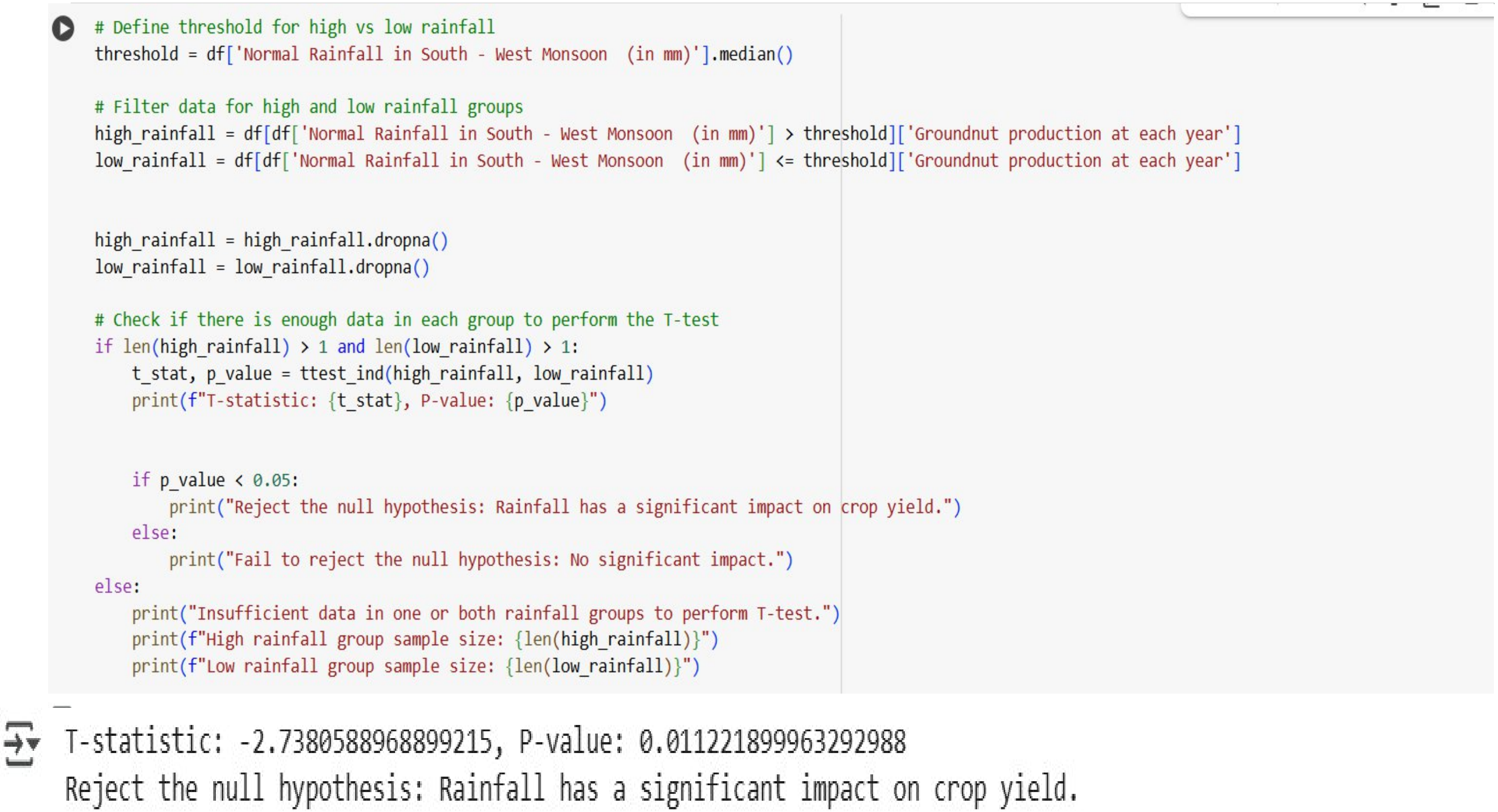
```python
# Define threshold for high vs low rainfall
threshold = df['Normal Rainfall in North-East Monsoon  (in mm)'].median()

# Filter data for high and low rainfall groups
high_rainfall = df[df['Normal Rainfall in North-East Monsoon  (in mm)'] > threshold]['Rice production at each year']
low_rainfall = df[df['Normal Rainfall in North-East Monsoon  (in mm)'] <= threshold]['Rice production at each year']


high_rainfall = high_rainfall.dropna()
low_rainfall = low_rainfall.dropna()

# Check if there is enough data in each group to perform the T-test
if len(high_rainfall) > 1 and len(low_rainfall) > 1:

    t_stat, p_value = ttest_ind(high_rainfall, low_rainfall)
    print(f"T-statistic: {t_stat}, P-value: {p_value}")


    if p_value < 0.05:
        print("Reject the null hypothesis: Rainfall has a significant impact on crop yield.")
    else:
        print("Fail to reject the null hypothesis: No significant impact.")
else:
    print("Insufficient data in one or both rainfall groups to perform T-test.")
    print(f"High rainfall group sample size: {len(high_rainfall)}")
    print(f"Low rainfall group sample size: {len(low_rainfall)}")
```

```
T-statistic: 0.0881908413028324, P-value: 0.9304276542461054
Fail to reject the null hypothesis: No significant impact.
```



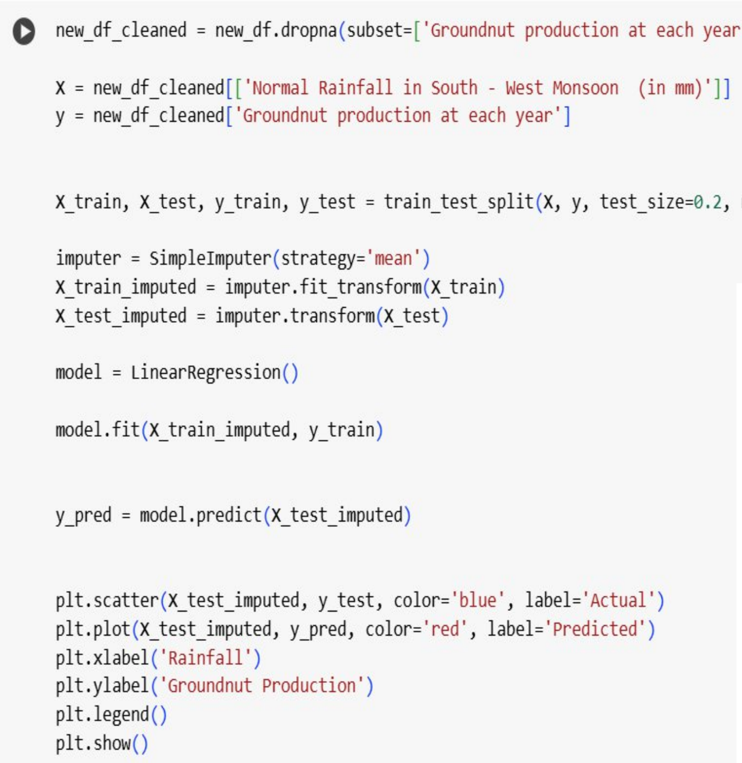
```python
new_df_cleaned = new_df.dropna(subset=['Rice production at each year'])

X = new_df_cleaned[['Normal Rainfall in North-East Monsoon  (in mm)']]
y = new_df_cleaned['Rice production at each year']


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

imputer = SimpleImputer(strategy='mean')
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

model = LinearRegression()

model.fit(X_train_imputed, y_train)


y_pred = model.predict(X_test_imputed)


plt.scatter(X_test_imputed, y_test, color='blue', label='Actual')
plt.plot(X_test_imputed, y_pred, color='red', label='Predicted')
plt.xlabel('Rainfall')
plt.ylabel('Rice Production')
plt.legend()
plt.show()
```

Rainfall & Crop Yield – Negative Impact

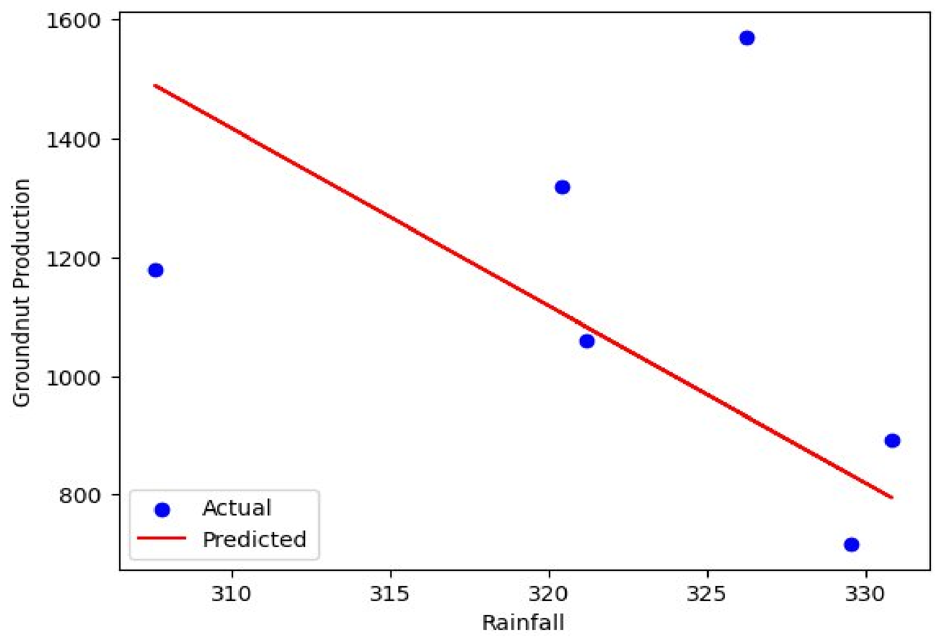


```
# Define threshold for high vs low rainfall
threshold = df['Normal Rainfall in South - West Monsoon  (in mm)'].median()

# Filter data for high and low rainfall groups
high_rainfall = df[df['Normal Rainfall in South - West Monsoon  (in mm)'] > threshold]['Groundnut production at each year']
low_rainfall = df[df['Normal Rainfall in South - West Monsoon  (in mm)'] <= threshold]['Groundnut production at each year']


high_rainfall = high_rainfall.dropna()
low_rainfall = low_rainfall.dropna()

# Check if there is enough data in each group to perform the T-test
if len(high_rainfall) > 1 and len(low_rainfall) > 1:
    t_stat, p_value = ttest_ind(high_rainfall, low_rainfall)
    print(f"T-statistic: {t_stat}, P-value: {p_value}")


    if p_value < 0.05:
        print("Reject the null hypothesis: Rainfall has a significant impact on crop yield.")
    else:
        print("Fail to reject the null hypothesis: No significant impact.")
else:
    print("Insufficient data in one or both rainfall groups to perform T-test.")
    print(f"High rainfall group sample size: {len(high_rainfall)}")
    print(f"Low rainfall group sample size: {len(low_rainfall)}")
```

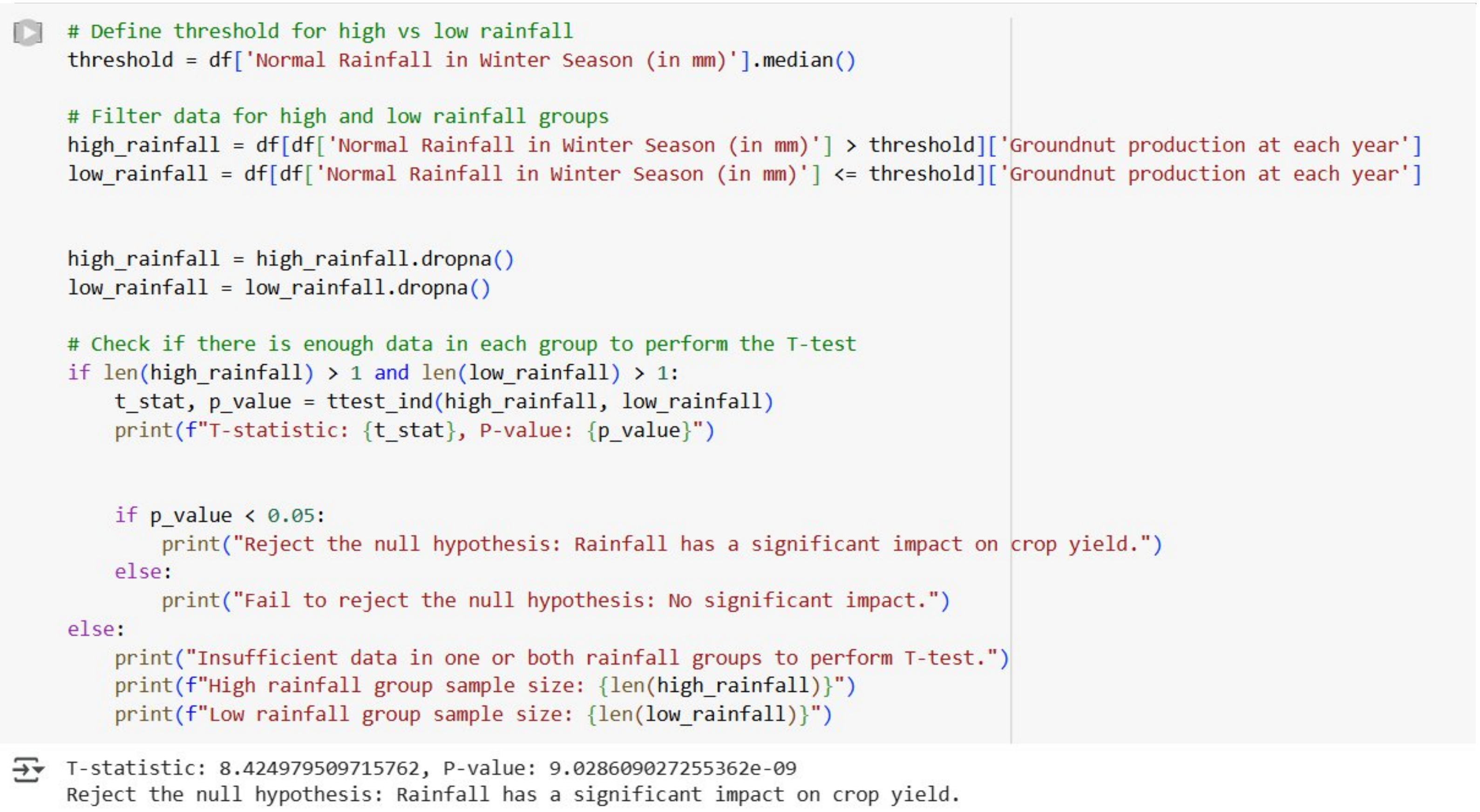
```
T-statistic: -2.7380588968899215, P-value: 0.011221899963292988
Reject the null hypothesis: Rainfall has a significant impact on crop yield.
```



```
new_df_cleaned = new_df.dropna(subset=['Groundnut production at each year'])

X = new_df_cleaned[['Normal Rainfall in South - West Monsoon  (in mm)']]
y = new_df_cleaned['Groundnut production at each year']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) # 80% for training, 20% for testing

imputer = SimpleImputer(strategy='mean')
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

model = LinearRegression()

model.fit(X_train_imputed, y_train)


y_pred = model.predict(X_test_imputed)


plt.scatter(X_test_imputed, y_test, color='blue', label='Actual')
plt.plot(X_test_imputed, y_pred, color='red', label='Predicted')
plt.xlabel('Rainfall')
plt.ylabel('Groundnut Production')
plt.legend()
plt.show()
```

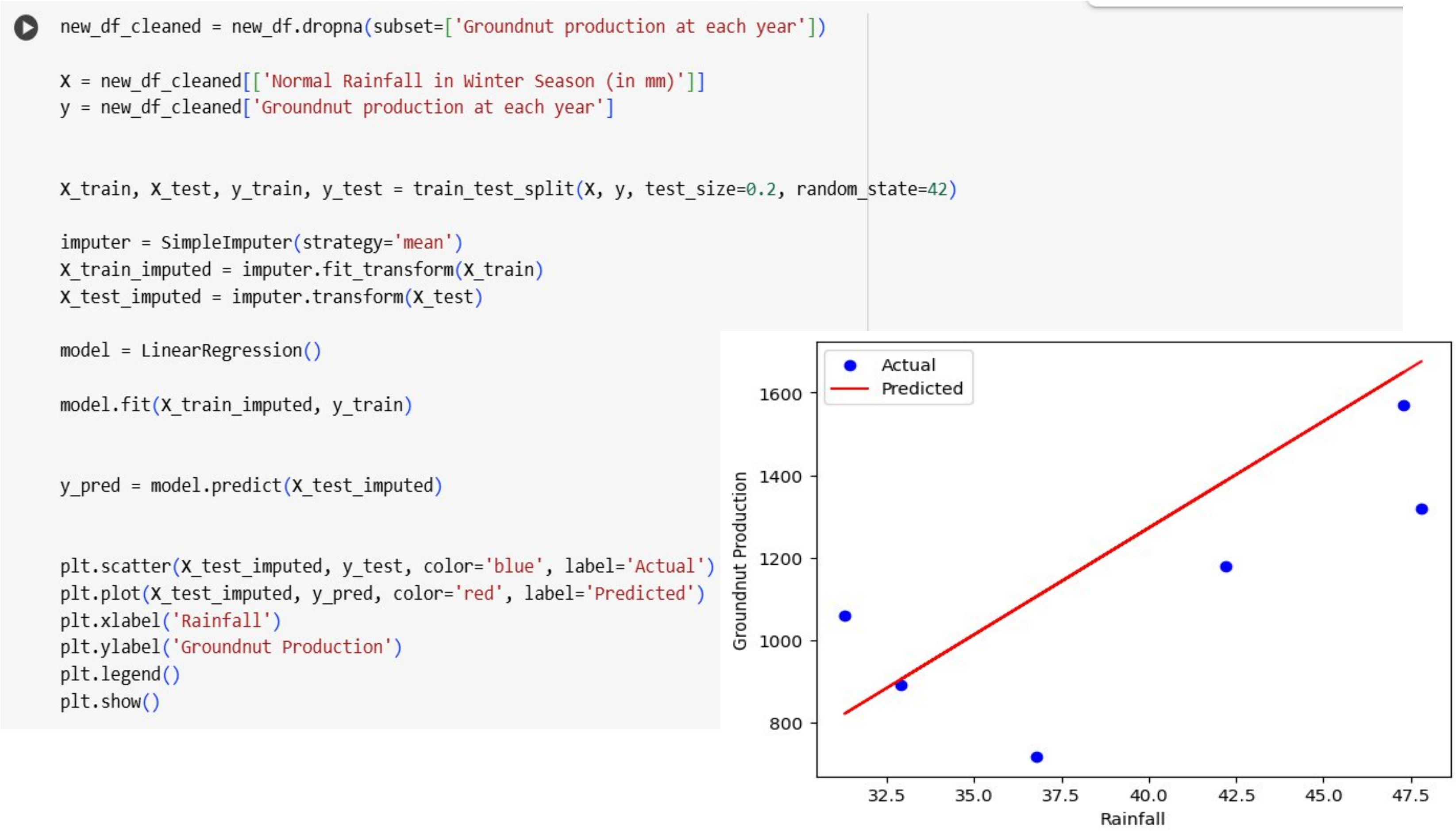


Rainfall & Crop Yield – Positive Impact



```
# Define threshold for high vs low rainfall
threshold = df['Normal Rainfall in Winter Season (in mm)'].median()

# Filter data for high and low rainfall groups
high_rainfall = df[df['Normal Rainfall in Winter Season (in mm)'] > threshold]['Groundnut production at each year']
low_rainfall = df[df['Normal Rainfall in Winter Season (in mm)'] <= threshold]['Groundnut production at each year']


high_rainfall = high_rainfall.dropna()
low_rainfall = low_rainfall.dropna()

# Check if there is enough data in each group to perform the T-test
if len(high_rainfall) > 1 and len(low_rainfall) > 1:
    t_stat, p_value = ttest_ind(high_rainfall, low_rainfall)
    print(f"T-statistic: {t_stat}, P-value: {p_value}")


    if p_value < 0.05:
        print("Reject the null hypothesis: Rainfall has a significant impact on crop yield.")
    else:
        print("Fail to reject the null hypothesis: No significant impact.")
else:
    print("Insufficient data in one or both rainfall groups to perform T-test.")
    print(f"High rainfall group sample size: {len(high_rainfall)}")
    print(f"Low rainfall group sample size: {len(low_rainfall)}")
```

```
T-statistic: 8.424979509715762, P-value: 9.028609027255362e-09
Reject the null hypothesis: Rainfall has a significant impact on crop yield.
```



```
new_df_cleaned = new_df.dropna(subset=['Groundnut production at each year'])

X = new_df_cleaned[['Normal Rainfall in Winter Season (in mm)']]
y = new_df_cleaned['Groundnut production at each year']


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

imputer = SimpleImputer(strategy='mean')
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

model = LinearRegression()

model.fit(X_train_imputed, y_train)


y_pred = model.predict(X_test_imputed)


plt.scatter(X_test_imputed, y_test, color='blue', label='Actual')
plt.plot(X_test_imputed, y_pred, color='red', label='Predicted')
plt.xlabel('Rainfall')
plt.ylabel('Groundnut Production')
plt.legend()
plt.show()
```

Conclusion about this project

Rainfall is vital to crop health, but its **timing and distribution** matter more than total quantity.

Winter and hot-season rains positively impact crops like groundnut and oilseeds.

South-West monsoon can have **negative effects** on certain crops due to excess or poorly timed rainfall.

Statistical analysis confirms that rainfall-yield relationships are **significant**, not random.

Predictive models enable informed planning by forecasting crop yields based on rainfall patterns, helping optimize agricultural strategies in Tamil Nadu.

Future Reference

Long-term Climate Modeling: Incorporate climate change projections to assess future rainfall variability and its effects on agriculture.

Crop-Specific Studies: Deep-dive into individual crop responses across different districts to design localized farming practices.

Precision Agriculture: Integrate weather forecasting with sensor-based irrigation to optimize water usage and minimize crop stress.

Policy Formulation: Use data-driven insights to guide drought relief, irrigation planning, and crop insurance policies.

Bibliography

Source of the code:

https://colab.research.google.com/drive/171M5uzpHd5TxUSnoAVOkQ8lasGvaVDT4

Source of the dataset:

- http://data.icrisat.org/dld/src/crops.htm
- https://www.data.gov.in/sector/Environment%20and%20Forest