NASA CONCEPTUAL[¶](#gjdgxs)

Team Members[¶](#30j0zll)

* Adam
* Elizabeth
* Khoi
* Manuel

Import Libraries[¶](#1fob9te)

In [1]:

**import** pandas **as** pd  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.preprocessing **import** StandardScaler  
**from** sklearn.gaussian\_process **import** GaussianProcessRegressor  
**from** sklearn.gaussian\_process.kernels **import** RBF, ConstantKernel **as** C  
**from** sklearn.metrics **import** mean\_squared\_error

Load dataset[¶](#3znysh7)

In [ ]:

*# Load the dataset*  
data **=** pd**.**read\_csv('your\_weather\_data.csv')  
  
*# Display the first few rows*  
print(data**.**head())

Exploratory Data Analysis[¶](#2et92p0)

In [ ]:

*# Check for missing values*  
print(data**.**isnull()**.**sum())  
  
*# Basic statistics*  
print(data**.**describe())  
  
*# Example visualization*  
data['Temperature']**.**plot(kind**=**'line', title**=**'Temperature Trends', figsize**=**(10, 5))  
plt**.**show()

Data Processing[¶](#tyjcwt)

In [ ]:

*# Fill missing values*  
data**.**fillna(method**=**'ffill', inplace**=True**)  
  
*# Select features and target*  
X **=** data[['Feature1', 'Feature2', 'Feature3']] *# Replace with relevant columns*  
y **=** data['Target'] *# Replace with your target variable (e.g., 'Temperature')*  
  
*# Normalize features*  
scaler **=** StandardScaler()  
X\_scaled **=** scaler**.**fit\_transform(X)  
  
*# Split into training and testing sets*  
X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X\_scaled, y, test\_size**=**0.2, random\_state**=**42)

Define and Train the GPR Model[¶](#3dy6vkm)

In [ ]:

*# Define the kernel*  
kernel **=** C(1.0, (1e-3, 1e3)) **\*** RBF(length\_scale**=**1.0, length\_scale\_bounds**=**(1e-2, 1e2))  
  
*# Initialize the Gaussian Process Regressor*  
gpr **=** GaussianProcessRegressor(kernel**=**kernel, n\_restarts\_optimizer**=**10, random\_state**=**42)  
  
*# Train the model*  
gpr**.**fit(X\_train, y\_train)

Make predictions and evaluate model[¶](#1t3h5sf)

In [ ]:

*# Predict on the test set*  
y\_pred, sigma **=** gpr**.**predict(X\_test, return\_std**=True**)  
  
*# Calculate RMSE*  
rmse **=** np**.**sqrt(mean\_squared\_error(y\_test, y\_pred))  
print(f'Root Mean Squared Error: {rmse}')  
  
*# Plot predictions vs actual values*  
plt**.**figure(figsize**=**(10, 5))  
plt**.**plot(y\_test**.**values, label**=**'Actual')  
plt**.**plot(y\_pred, label**=**'Predicted')  
plt**.**fill\_between(np**.**arange(len(y\_pred)), y\_pred **-** sigma, y\_pred **+** sigma, alpha**=**0.2, label**=**'Confidence Interval')  
plt**.**legend()  
plt**.**title('GPR Weather Prediction')  
plt**.**show()

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: