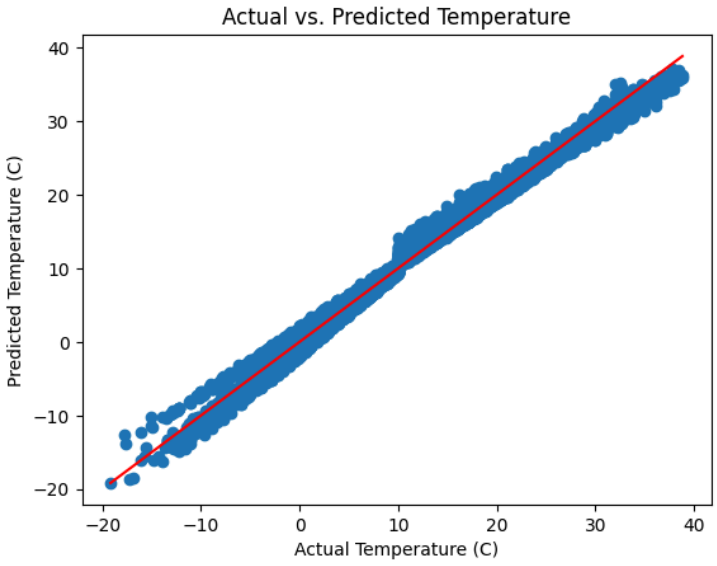
```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score


from google.colab import drive
drive.mount('/content/drive')
```

```
     Mounted at /content/drive
```

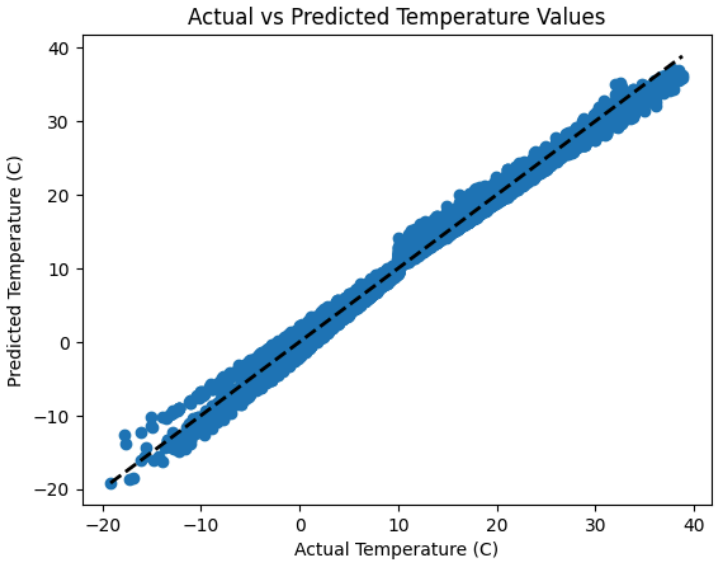
```
import pandas as pd
data = pd.read_csv("/content/drive/MyDrive/weather/weatherHistory.csv.xls")


# Remove unnecessary columns
data.drop(['Formatted Date', 'Summary', 'Precip Type', 'Daily Summary'], axis=1, inplace=True)


# Separate the features and target variable
X = data.drop('Temperature (C)', axis=1)
y = data['Temperature (C)']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a linear regression model
model = LinearRegression()

# Fit the model on the training data
model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)

# Calculate the R-squared score
score = r2_score(y_test, y_pred)
print("R-squared score:", score)


import matplotlib.pyplot as plt
import numpy as np

# Plot the actual vs. predicted values
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Temperature (C)")
plt.ylabel("Predicted Temperature (C)")
plt.title("Actual vs. Predicted Temperature")

# Fit a line to the scatter plot
x = np.linspace(np.min(y_test), np.max(y_test), 100)
y = x
plt.plot(x, y, 'r')

plt.show()
```



```
from sklearn.linear_model import Ridge
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import pandas as pd

# Load the data
data = pd.read_csv("/content/drive/MyDrive/weather/weatherHistory.csv.xls")

# Remove unnecessary columns
data.drop(['Formatted Date', 'Summary', 'Precip Type', 'Daily Summary'], axis=1, inplace=True)

# Separate the features and target variable
X = data.drop('Temperature (C)', axis=1)
y = data['Temperature (C)']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Create and train the Ridge regression model
ridge = Ridge(alpha=1.0) # default alpha value
ridge.fit(X_train_scaled, y_train)

# Evaluate the model on the test set
score = ridge.score(X_test_scaled, y_test)
print("Ridge regression score:", score)
```

```
Ridge regression score: 0.9902437951540839
```

```
import matplotlib.pyplot as plt

# Make predictions on the test data
y_pred = ridge.predict(X_test_scaled)

# Plot the actual vs predicted temperature values
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Temperature (C)')
plt.ylabel('Predicted Temperature (C)')
plt.title('Actual vs Predicted Temperature Values')


# Add the line y=x
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)

plt.show()
```



```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score


data = pd.read_csv("/content/drive/MyDrive/weather/weatherHistory.csv.xls")

data.drop(['Formatted Date', 'Summary', 'Precip Type', 'Daily Summary'], axis=1, inplace=True)

X = data.drop('Temperature (C)', axis=1)
y = data['Temperature (C)']


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

model = RandomForestRegressor()

model.fit(X_train, y_train)
```

```
import matplotlib.pyplot as plt
# Ploting the predicted values versus the actual values
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs Predicted Values (Random Forest)')
plt.show()
```