EXPERIMENT 1 :

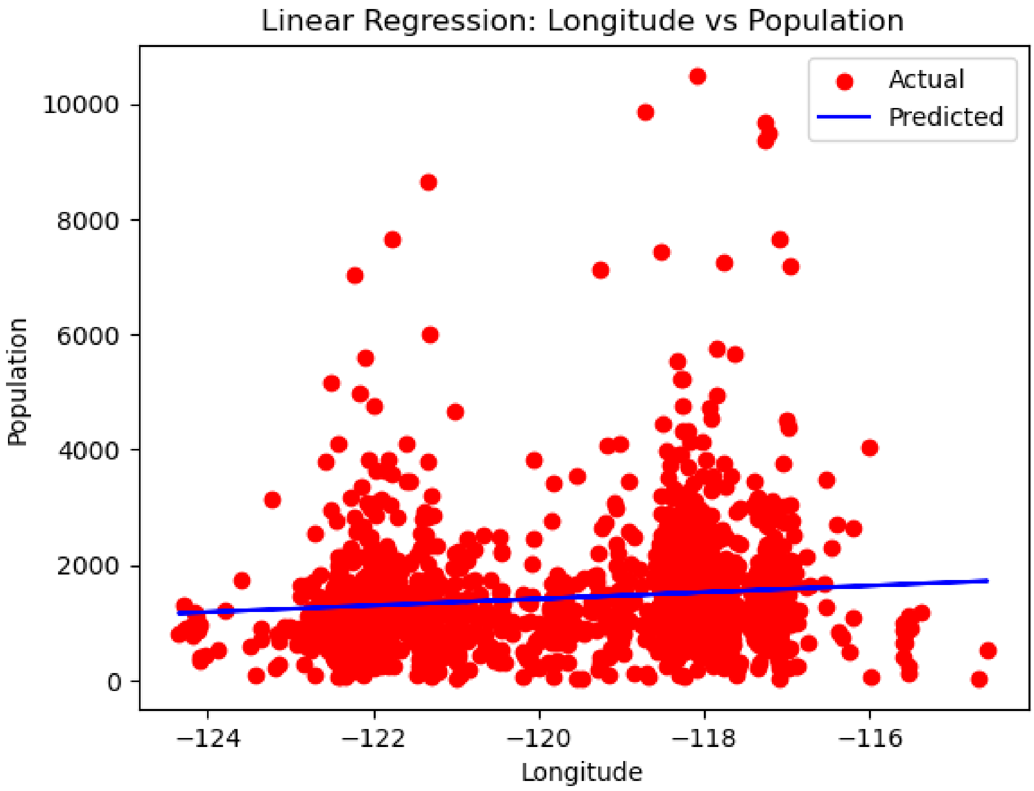
1A) LINEAR REGRESSION:

CODE:

```
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn import linear_model


# Load the data

df = pd.read_csv('california_housing_train.csv')


# Drop rows with missing values

df.dropna(inplace=True)


# Extract features and target variable

xpoints = df["longitude"].values.reshape(-1, 1)

ypoints = df["population"].values


# Split the data into training and testing sets

x_train, x_test, y_train, y_test = train_test_split(xpoints, ypoints, test_size=0.1, random_state=42)


# Create and train the linear regression model

reg = linear_model.LinearRegression()

reg.fit(x_train, y_train)


# Make predictions on the test set

ypoints_pred = reg.predict(x_test)
```

```
# Plot the results

plt.scatter(x_test, y_test, color="red", label="Actual")

plt.plot(x_test, ypoints_pred, color="blue", label="Predicted")

plt.xlabel("Longitude")

plt.ylabel("Population")

plt.title("Linear Regression: Longitude vs Population")

plt.legend()

plt.show()
```

OUTPUT:



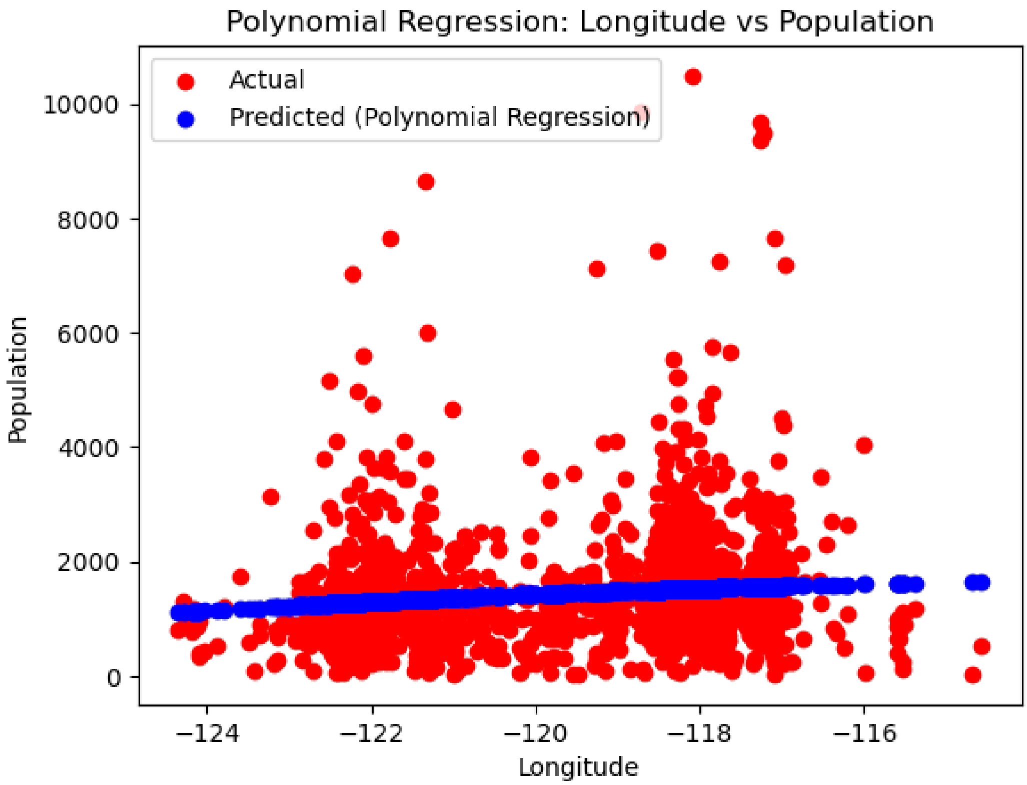
1B)POLYNOMIAL REGRESSION:

CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error
df = pd.read_csv('california_housing_train.csv')
df.dropna(inplace=True)

# Extract features and target variable
xpoints = df["longitude"].values.reshape(-1, 1)
ypoints = df["population"].values

# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(xpoints, ypoints, test_size=0.1, random_state=42)

# Polynomial features transformation
degree = 2  # Define the degree of the polynomial
poly_features = PolynomialFeatures(degree=degree)
x_train_poly = poly_features.fit_transform(x_train)
x_test_poly = poly_features.transform(x_test)

# Create and train the polynomial regression model
poly_reg = LinearRegression()
```

```
poly_reg.fit(x_train_poly, y_train)

# Make predictions on the test set
ypoints_pred = poly_reg.predict(x_test_poly)

# Calculate and print the Root Mean Squared Error (RMSE)
rmse = np.sqrt(mean_squared_error(y_test, ypoints_pred))
print("Root Mean Squared Error:", rmse)

# Plot the results
plt.scatter(x_test, y_test, color="red", label="Actual")
plt.scatter(x_test, ypoints_pred, color="blue", label="Predicted (Polynomial Regression)")
plt.xlabel("Longitude")
plt.ylabel("Population")
plt.title("Polynomial Regression: Longitude vs Population")
plt.legend()
plt.show()
```

OUTPUT:


Polynomial Regression: Longitude vs Population
Actual
Predicted (Polynomial Regression)
10000
8000
6000
4000
2000
0
Population
−124
−122
−120
−118
−116
Longitude