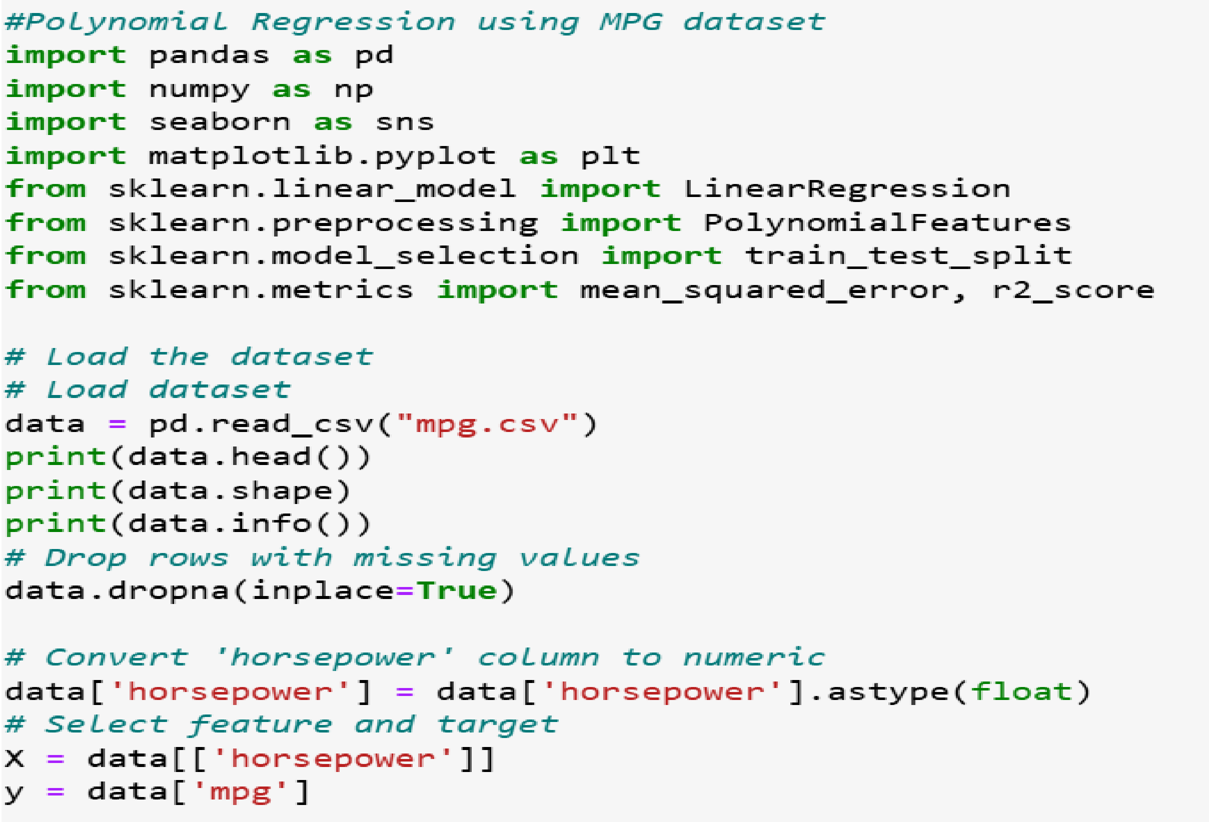
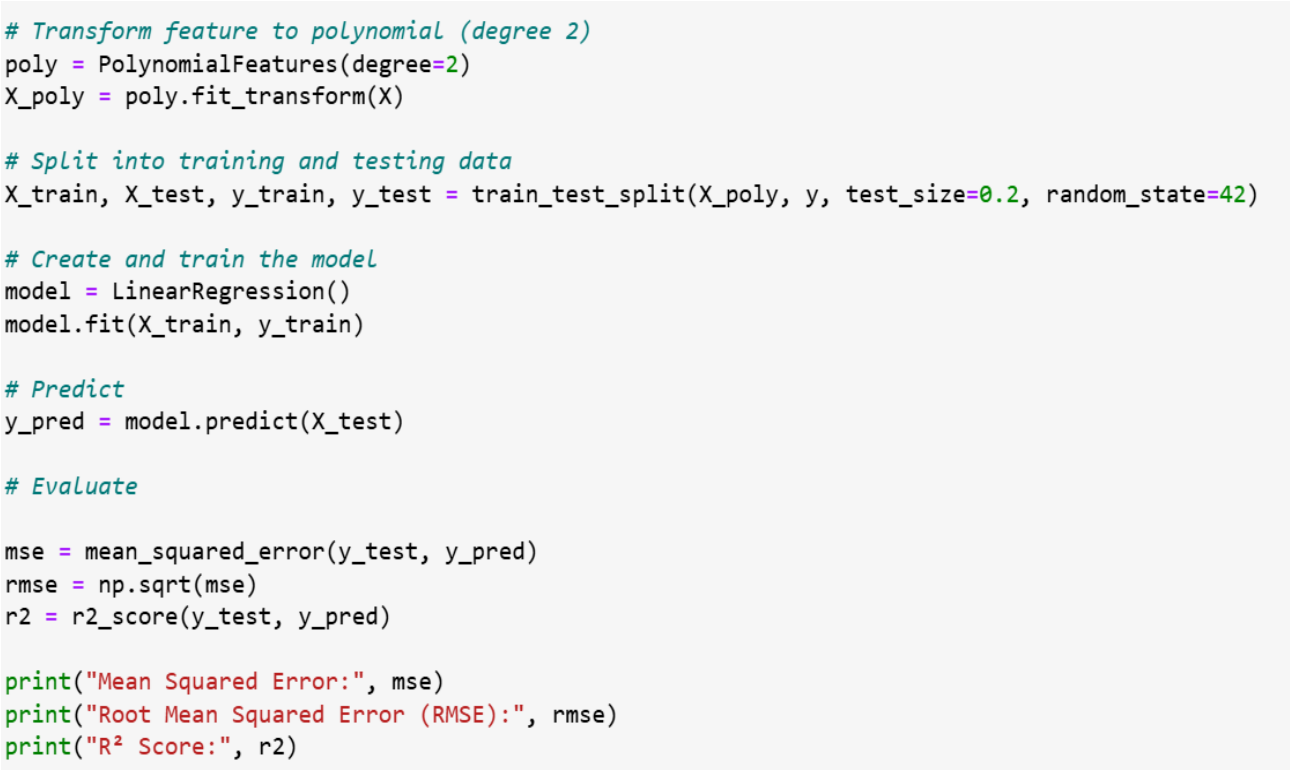
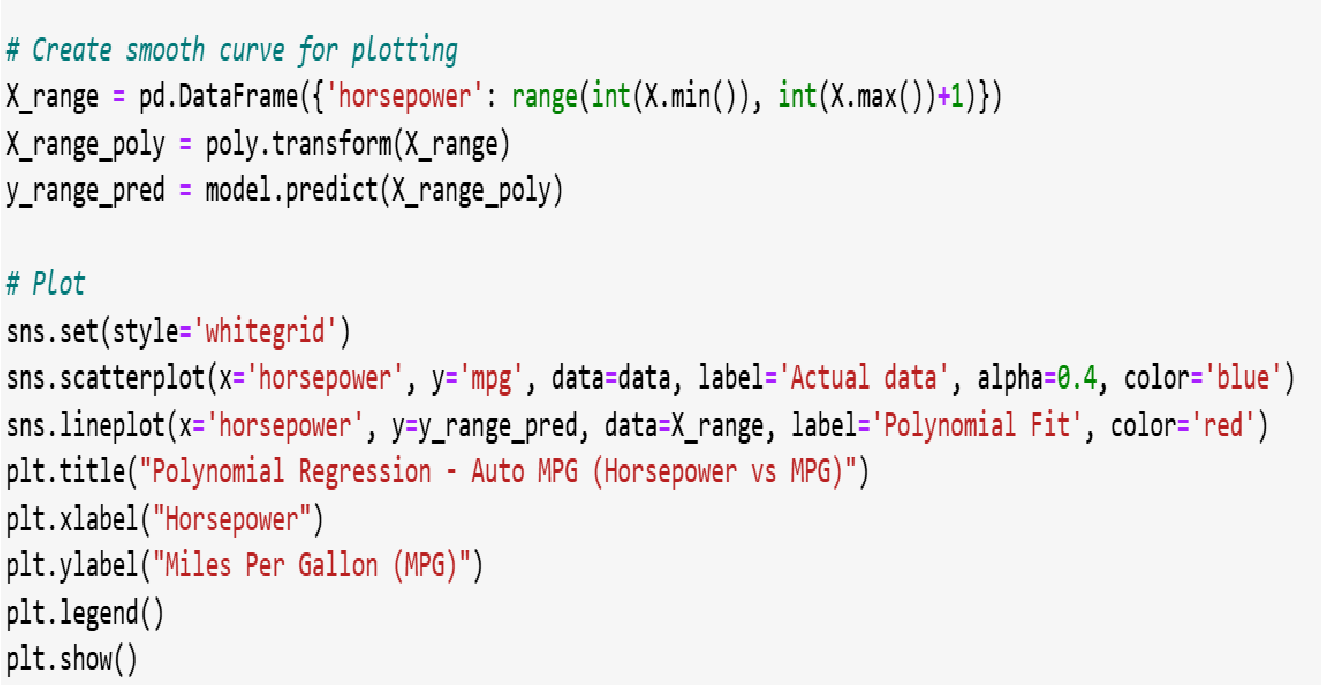


```python
#Polynomial Regression using MPG dataset
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset
# Load dataset
data = pd.read_csv("mpg.csv")
print(data.head())
print(data.shape)
print(data.info())
# Drop rows with missing values
data.dropna(inplace=True)

# Convert 'horsepower' column to numeric
data['horsepower'] = data['horsepower'].astype(float)
# Select feature and target
X = data[['horsepower']]
y = data['mpg']
```



```python
# Transform feature to polynomial (degree 2)
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

# Split into training and testing data
X_train, X_test, y_train, y_test = train_test_split(X_poly, y, test_size=0.2, random_state=42)

# Create and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluate

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R² Score:", r2)
```



```python
# Create smooth curve for plotting
X_range = pd.DataFrame({'horsepower': range(int(X.min()), int(X.max())+1)})
X_range_poly = poly.transform(X_range)
y_range_pred = model.predict(X_range_poly)

# Plot
sns.set(style='whitegrid')
sns.scatterplot(x='horsepower', y='mpg', data=data, label='Actual data', alpha=0.4, color='blue')
sns.lineplot(x='horsepower', y=y_range_pred, data=X_range, label='Polynomial Fit', color='red')
plt.title("Polynomial Regression - Auto MPG (Horsepower vs MPG)")
plt.xlabel("Horsepower")
plt.ylabel("Miles Per Gallon (MPG)")
plt.legend()
plt.show()
```