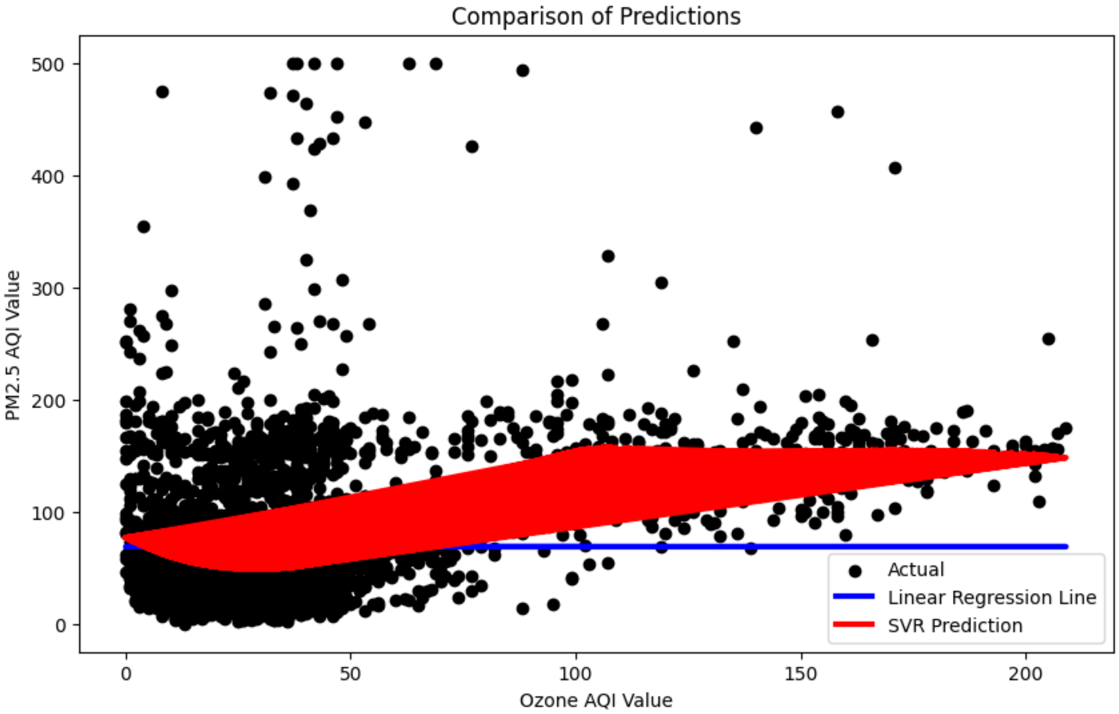
```
#Comparative analysis on Linear and Support Vector regression.ipynb
#Done by Thirumoorthi R, Gokulraj R, Muthu shankar M, Eemana yogeswar

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.metrics import mean_absolute_error, r2_score

# Load the dataset from a CSV file
# Replace 'your_dataset.csv' with the actual filename
# 1)Data Preparation
df = pd.read_csv('global air pollution dataset.csv')

#2)Data cleansing
# Drop rows with NaN values in the target variable 'PM2.5 AQI Value'
df = df.dropna(subset=['PM2.5 AQI Value'])

# Extract the features and target variable
X = df['Ozone AQI Value'].values.reshape(-1, 1)  # Feature
y = df['PM2.5 AQI Value'].values  # Target variable

# Manually split the data into training and testing sets
split_index = int(0.8 * len(X))  # Use 80% of the data for training

X_train, X_test = X[:split_index], X[split_index:]
y_train, y_test = y[:split_index], y[split_index:]

#    - - - -    #

# Linear Regression without using fit method
#Finsing values for to substitue in formulae : :
X_mean = np.mean(X_train)
y_mean = np.mean(y_train)

numerator = np.sum((X_train - X_mean) * (y_train - y_mean))
denominator = np.sum((X_train - X_mean)**2)

slope_lr = numerator / denominator
intercept_lr = y_mean - slope_lr * X_mean

# Make predictions on the test set using Linear Regression
y_pred_lr = slope_lr * X_test.flatten() + intercept_lr  # Flatten X_test to match the shape of y_test

#Comparative analysis on Linear and Support Vector regression.ipynb

# Support Vector Regression
svr_model = SVR(kernel='rbf')
svr_model.fit(X_train, y_train)  # Use scaled features for SVR

# Make predictions on the test set using SVR
y_pred_svr = svr_model.predict(X_test)

# Evaluate the models using Mean Absolute Error
mae_lr = mean_absolute_error(y_test, y_pred_lr)
mae_svr = mean_absolute_error(y_test, y_pred_svr)

# Evaluate the models using R-squared coefficient
r2_lr = r2_score(y_test, y_pred_lr)
r2_svr = r2_score(y_test, y_pred_svr)

# Print the results
print("Linear Regression Metrics:")
print("Mean Absolute Error:", mae_lr)
print("R-squared:", r2_lr)

print("\nSupport Vector Regression Metrics:")
print("Mean Absolute Error:", mae_svr)
print("R-squared:", r2_svr)

# Plot the results
plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color='black', label='Actual')
plt.plot(X_test, y_pred_lr, color='blue', linewidth=3, label='Linear Regression Line')
plt.plot(X_test, y_pred_svr, color='red', linewidth=3, label='SVR Prediction')
plt.xlabel('Ozone AQI Value')
plt.ylabel('PM2.5 AQI Value')
plt.title('Comparison of Predictions')
plt.legend()
plt.show()
```

```
Linear Regression Metrics:
Mean Absolute Error: 36.80086527492324
R-squared: -0.0004847686079867408

Support Vector Regression Metrics:
Mean Absolute Error: 29.779988062385
R-squared: 0.14776668384285163
```


Comparison of Predictions
PM2.5 AQI Value
Ozone AQI Value
● Actual
Linear Regression Line
SVR Prediction