```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""

@author: truscottwatters

Love you Dad, Mark William Watters

Taught my AI, my second AI task, to predict the linear regression (meet the curve) of a linear function

Enjoying `Using Python for Research` at Harvard via edX.org

Charles Truscott Watters

"""

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

def main():
    plt.figure(0, dpi=300, figsize=[10, 5])
    x, y = [], []
    for z in range(1, 1000, 1):
        x.append(z)
        y.append(1955 * ((z ** 3) / 3) + 0.93)
#    print(x, y)
    ds = pd.DataFrame({"n": x, "square": y})
    X = ds.iloc[:,:1]
    Y = ds.iloc[:,1]
    print(X, Y)
    X_Tr, X_Te, y_tr, y_te = train_test_split(X, y, test_size = 1/3, random_state=0)
    regressor = LinearRegression()
    regressor.fit(X_Tr, y_tr)
#    y_pred = regressor.predict(X_Te)
    plt.scatter(x[0:999], y[0:999], color='red', label="y equals mx plus b or y=1955 * (z^3 / 3) + 0.93")
    plt.plot(X_Tr, regressor.predict(X_Tr), color='black', label="artificial intelligence prediction (or linear regression line)")
    plt.title("AI applied to Linear Regression. Charles Watters")
    plt.xlabel("x")
    plt.ylabel("y = 1955 * ((z ** 3) / 3) + 0.93")
    plt.legend()
    plt.show()

#    plt.title("Training set")
#    plt.scatter(X_Te, y_te, color='blue')
#    plt.plot(X_Tr, regressor.predict(X_Tr), color='red')
#    plt.title("Test set")
#    plt.show()

if __name__ == "__main__": main()
```