POLYNOMIALREGRESSION: wine dataset

In [47]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.linear_model import LinearRegression
```

In [15]:
```
winedata=pd.read_csv('WineQT[1].csv')

winedata.head(3)

winedata.describe()
```

Out[15]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
count	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000
mean	8.311111	0.531339	0.268364	2.532152	0.086933	15.615486	45.914698	0.996730	3.311015	0.657708	10.442111	5.657043	804.969379
std	1.747595	0.179633	0.196686	1.355917	0.047267	10.250486	32.782130	0.001925	0.156664	0.170399	1.082196	0.805824	463.997116
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000	0.000000
25%	7.100000	0.392500	0.090000	1.900000	0.070000	7.000000	21.000000	0.995570	3.205000	0.550000	9.500000	5.000000	411.000000
50%	7.900000	0.520000	0.250000	2.200000	0.079000	13.000000	37.000000	0.996680	3.310000	0.620000	10.200000	6.000000	794.000000
75%	9.100000	0.640000	0.420000	2.600000	0.090000	21.000000	61.000000	0.997845	3.400000	0.730000	11.100000	6.000000	1209.500000
max	15.900000	1.580000	1.000000	15.500000	0.611000	68.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000	1597.000000

In [16]:
```
X=winedata[['quality']]
y=winedata['alcohol']
```

In [18]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.2,random_state=42)
```

In [20]:
```
degree=2
poly=PolynomialFeatures(degree=degree)
X_train_poly=poly.fit_transform(X_train)
X_test_poly=poly.fit(X_test)
```

In [22]:
```
model=LinearRegression()

model.fit(X_train,y_train)
```

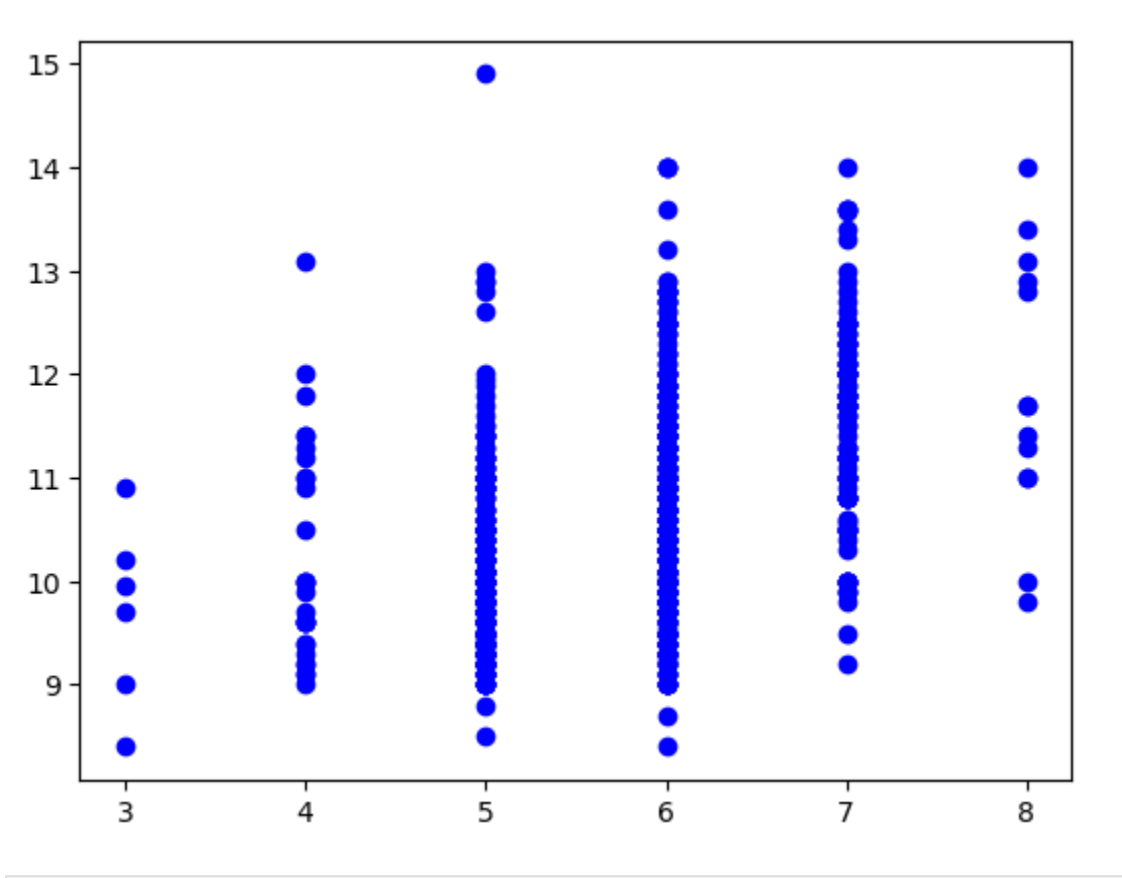
Out[22]:
```
▾ LinearRegression
LinearRegression()
```

In [24]:
```
y_pred=model.predict(X_test)
mse=mean_squared_error(y_test,y_pred)

mse2=r2_score(y_test,y_pred)

print(mse,mse2)
```

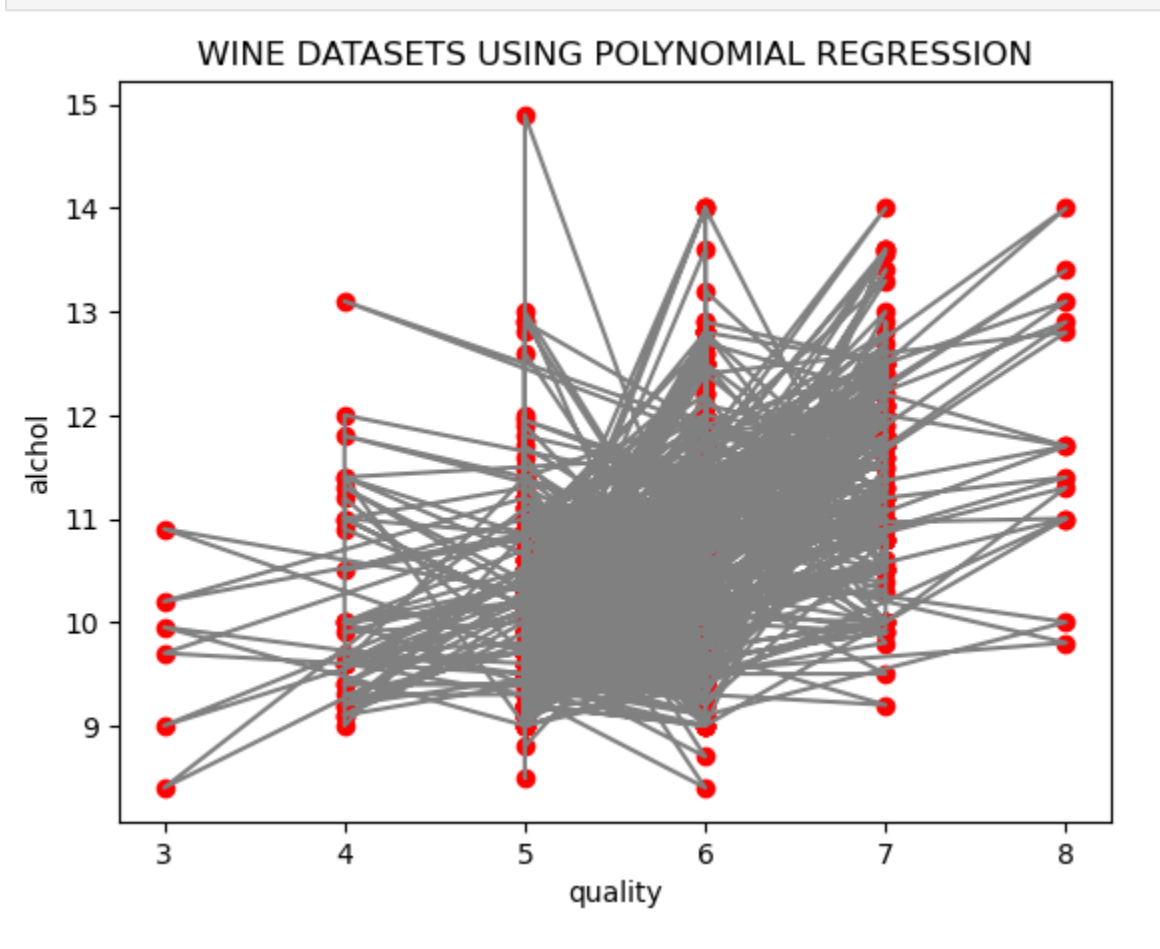
```
0.9044494626814545 0.2382569268877498
```

In [27]:
```
plt.scatter(X_test, y_test, color='blue', label='Actual')
```

Out[27]:
```
<matplotlib.collections.PathCollection at 0x26e6cb6ea70>
```



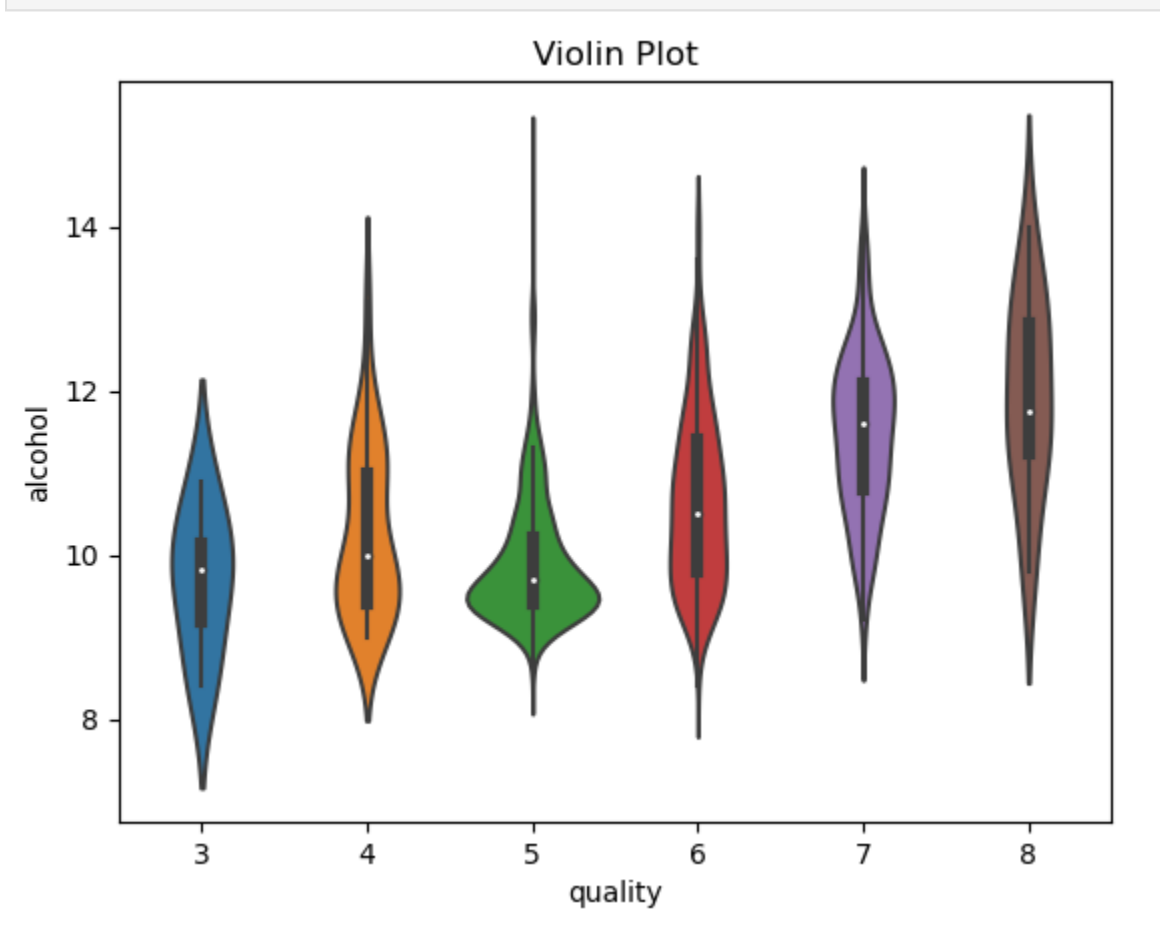
In [46]:
```
plt.scatter(X_test, y_test, color='red', label='Predicated')
plt.plot(X_test, y_test, color='gray', label='Polynomial Regression')

plt.xlabel('quality')
plt.ylabel('alchol')

plt.title('WINE DATASETS USING POLYNOMIAL REGRESSION')

plt.show()
```



In [42]:
```
# Violin Plot
plt.figure(figsize=(8, 6))
```

Out[42]:
```
<Figure size 800x600 with 0 Axes>

<Figure size 800x600 with 0 Axes>
```

In [44]:
```
sns.violinplot(x='quality', y='alcohol', data=winedata)
plt.xlabel('quality')
plt.ylabel('alcohol')
plt.title('Violin Plot')
plt.show()
```



POLYNOMIALREGRESSION:Auto mpg

In [84]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_squared_error
```

In [85]:
```
data=pd.read_csv('auto-mpg[1].csv')

data.head(4)
data.describe()
```

Out[85]:

	mpg	cylinders	displacement	weight	acceleration	model year	origin
count	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000
mean	23.514573	5.454774	193.425879	2970.424623	15.568090	76.010050	1.572864
std	7.815984	1.701004	104.269838	846.841774	2.757689	3.697627	0.802055
min	9.000000	3.000000	68.000000	1613.000000	8.000000	70.000000	1.000000
25%	17.500000	4.000000	104.250000	2223.750000	13.825000	73.000000	1.000000
50%	23.000000	4.000000	148.500000	2803.500000	15.500000	76.000000	1.000000
75%	29.000000	8.000000	262.000000	3608.000000	17.175000	79.000000	2.000000
max	46.600000	8.000000	455.000000	5140.000000	24.800000	82.000000	3.000000

In [86]:
```
X=data[['weight']]
y=data['mpg']
```

In [87]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.2,random_state=42)
```

In [88]:
```
poly = PolynomialFeatures(degree=2, include_bias=False)
X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.transform(X_test)
```

In [89]:
```
model=LinearRegression()

model.fit(X_train,y_train)
```

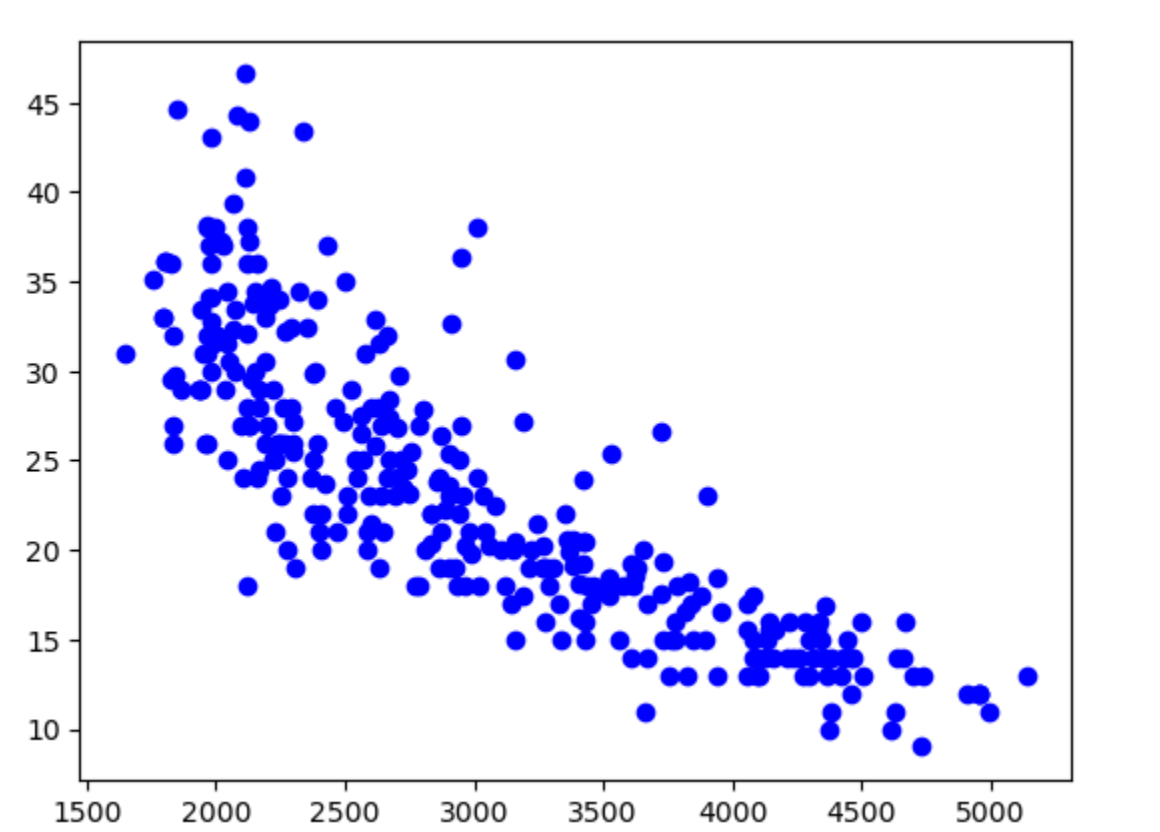
Out[89]:
```
▾ LinearRegression
LinearRegression()
```

In [90]:
```
y_pred=model.predict(X_test)
mse=mean_squared_error(y_test,y_pred)
mse2=r2_score(y_test,y_pred)
print(mse,mse2)
```

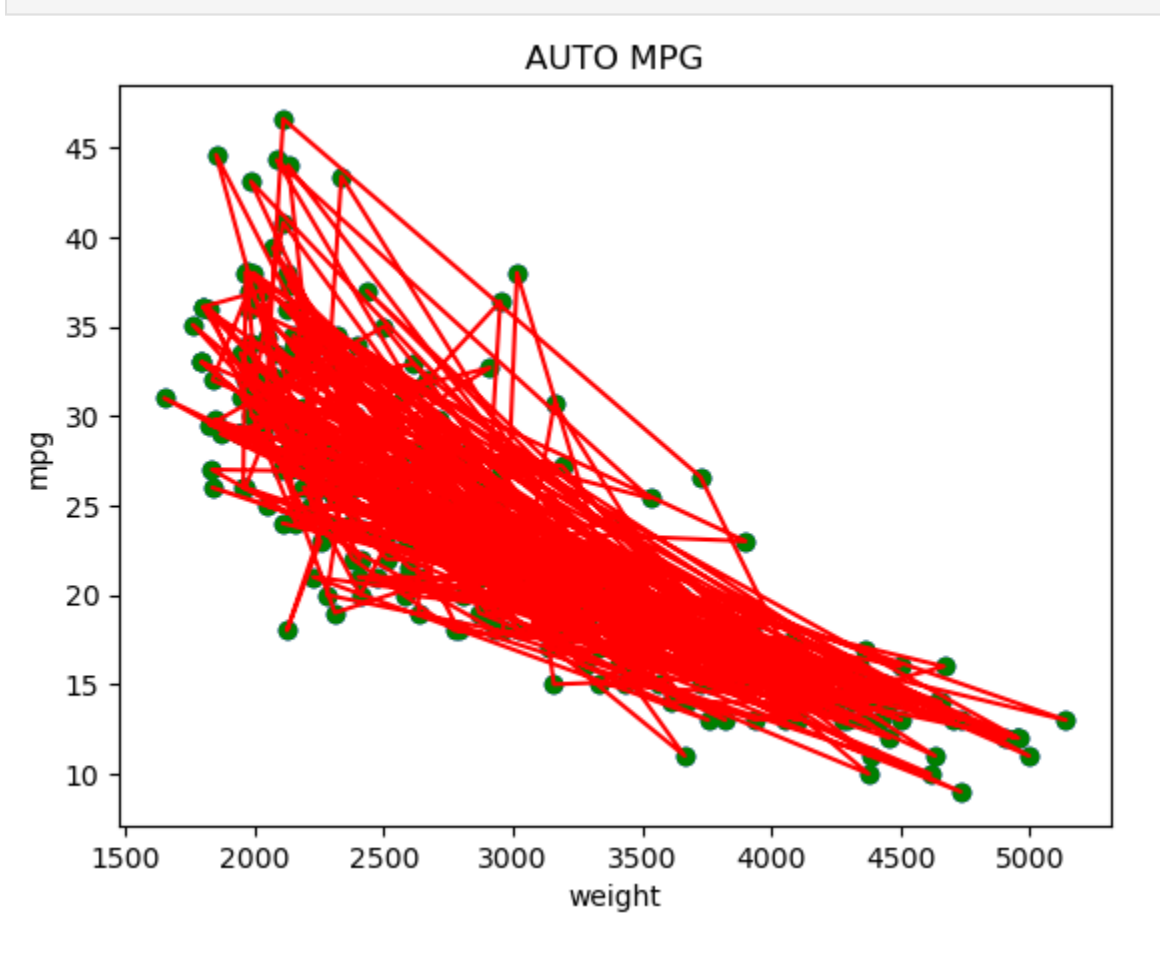
```
18.985331462827595 0.6903412939871773
```

In [96]:
```
plt.scatter(X_test,y_test,color='blue',label='Actual')
```

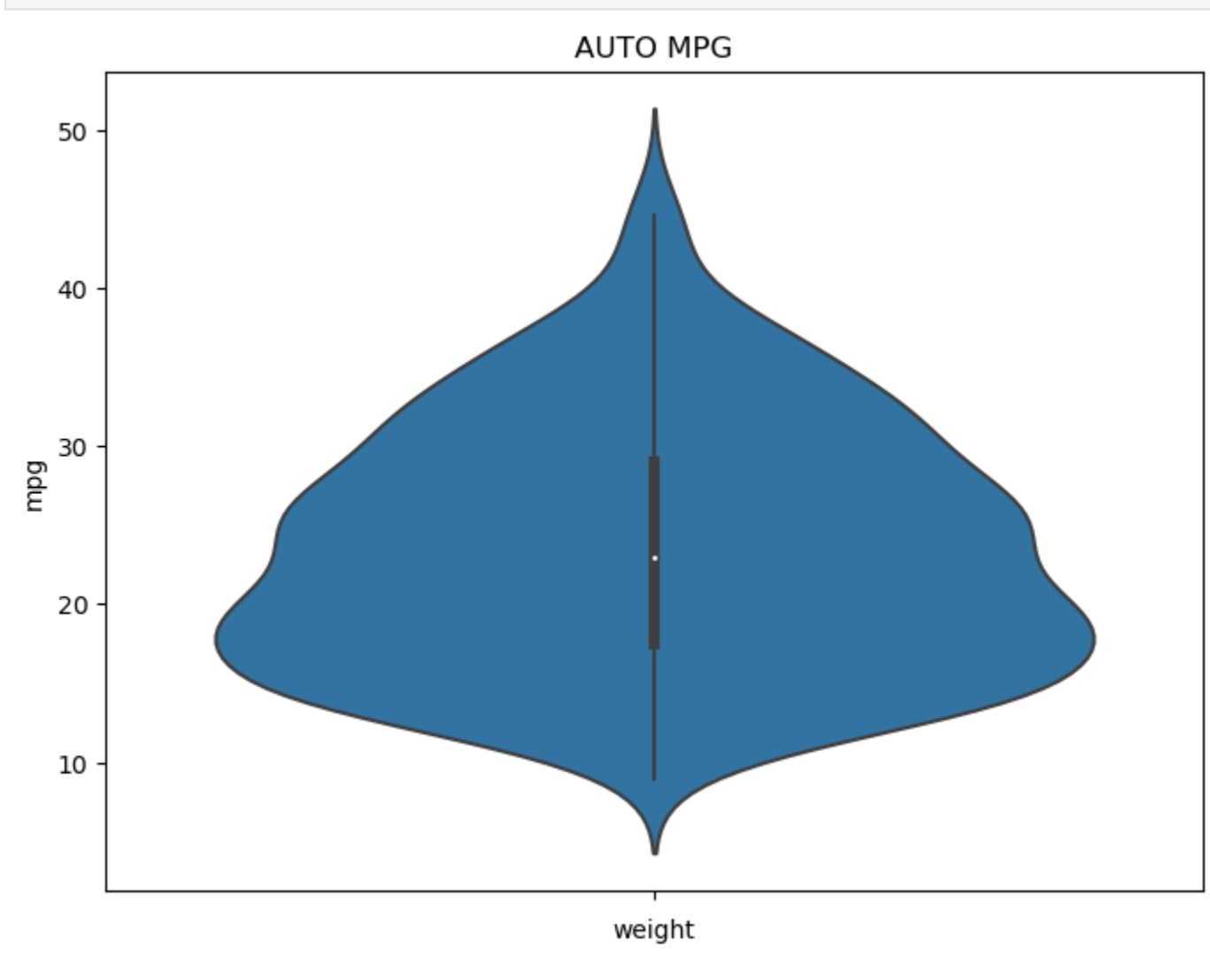
Out[96]:
```
<matplotlib.collections.PathCollection at 0x26e732d8460>
```



In [95]:
```
plt.scatter(X_test,y_test,color='green',label='predicated')
plt.plot(X_test,y_test,color='red',label='polynomial regression')
plt.xlabel('weight')
plt.ylabel('mpg')
plt.title('AUTO MPG')
plt.show()
```



In [101…
```
plt.figure(figsize=(8,6))

sns.violinplot(X='weight',y='mpg',data=data)
plt.xlabel('weight')
plt.ylabel('mpg')
plt.title('AUTO MPG')
plt.show()
```



POLYNOMIALREGRESSION:climate change

In [102…
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.model_selection import train_test_split
```

In [104…
```
data=pd.read_csv('climate_change[1].csv')

data.head(3)

data.describe()
```

Out[104]:

	Year	Month	MEI	CO2	CH4	N2O	CFC-11	CFC-12	TSI	Aerosols	Temp
count	308.000000	308.000000	308.000000	308.000000	308.000000	308.000000	308.000000	308.000000	308.000000	308.000000	308.000000
mean	1995.662338	6.551948	0.275555	363.226753	1749.824513	312.391834	251.973068	497.524782	1366.070759	0.016657	0.256776
std	7.423197	3.447214	0.937918	12.647125	46.051678	5.225131	20.231783	57.826899	0.399610	0.029050	0.179090
min	1983.000000	1.000000	-1.635000	340.170000	1629.890000	303.677000	191.324000	350.113000	1365.426100	0.001600	-0.282000
25%	1989.000000	4.000000	-0.398750	353.020000	1722.182500	308.111500	246.295500	472.410750	1365.717050	0.002800	0.121750
50%	1996.000000	7.000000	0.237500	361.735000	1764.040000	311.507000	258.344000	528.356000	1365.980900	0.005750	0.248000
75%	2002.000000	10.000000	0.830500	373.455000	1786.885000	316.979000	267.031000	540.524250	1366.363250	0.012600	0.407250
max	2008.000000	12.000000	3.001000	388.500000	1814.180000	322.182000	271.494000	543.813000	1367.316200	0.149400	0.739000

In [105…
```
X=data[['Aerosols']]
y=data['Temp']
```

In [106…
```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.2,random_state=42)
```

In [107…
```
degree=2
poly=PolynomialFeatures()
X_train_poly=poly.fit_transform(X_train)
X_test_poly=poly.transform(X_test)
```

In [108…
```
model=LinearRegression()
model.fit(X_train,y_train)
```

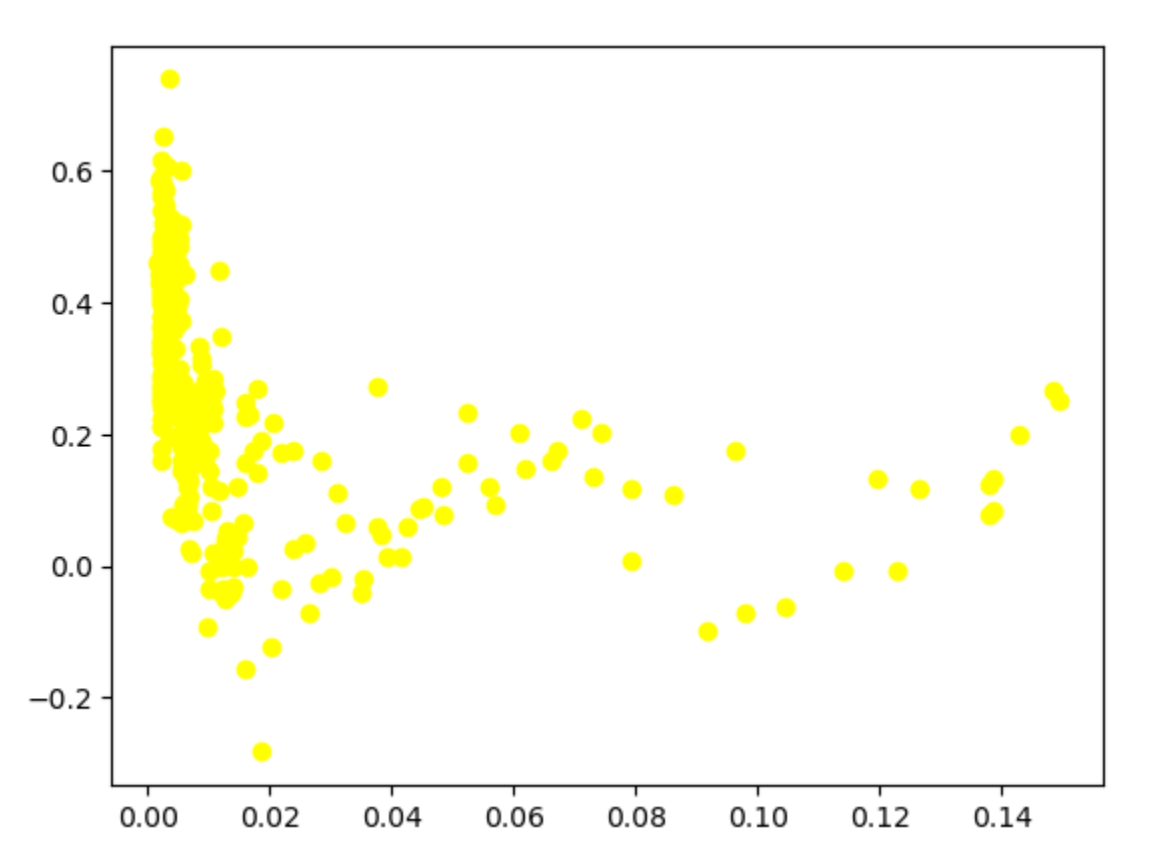
Out[108]:
```
▾ LinearRegression
LinearRegression()
```

In [109…
```
y_pred=model.predict(X_test)

mse=mean_squared_error(y_test,y_pred)

mse2=r2_score(y_test,y_pred)

print(mse,mse2)
```

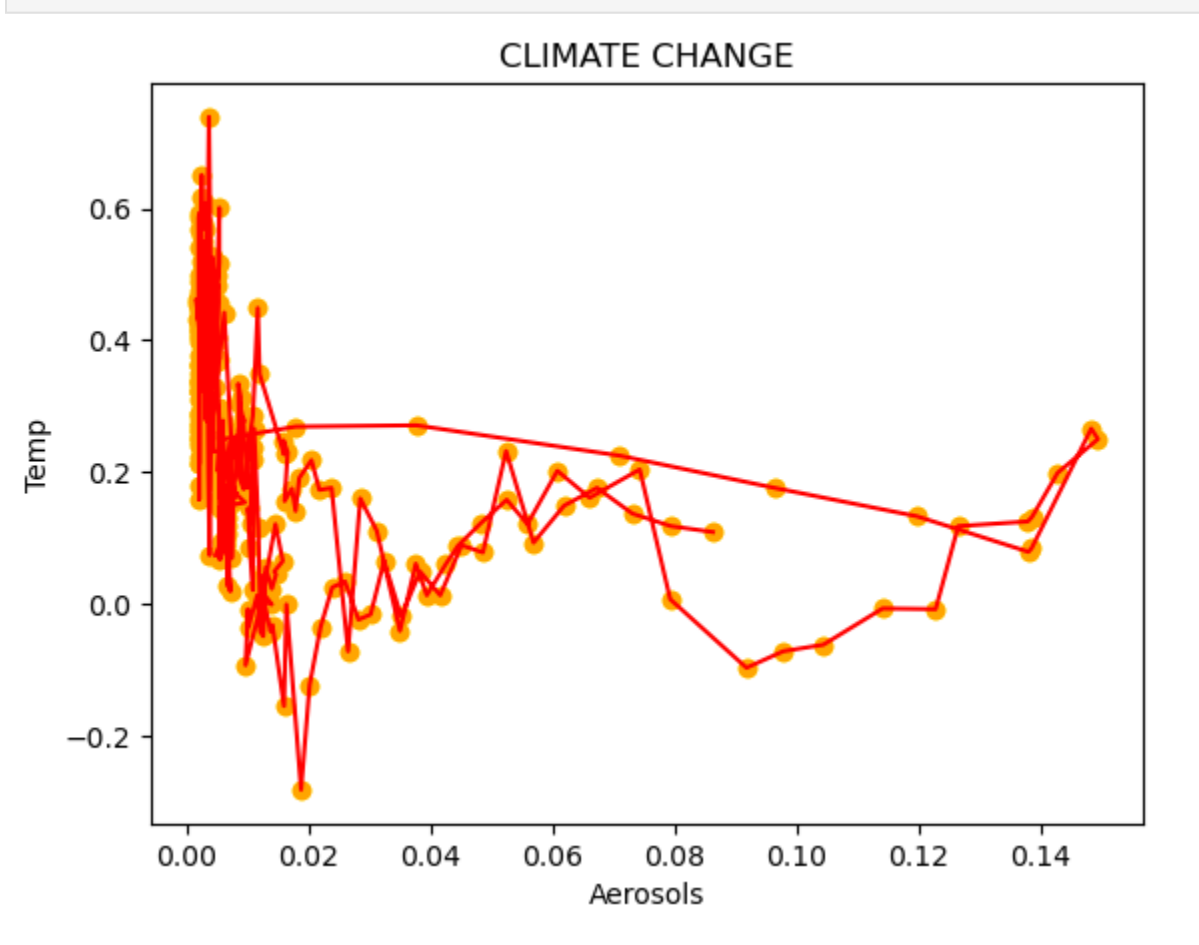
```
0.029167613451129736 0.13564002543840825
```

In [115…
```
plt.scatter(X,y,color='yellow',label='Actual')
```

Out[115]:
```
<matplotlib.collections.PathCollection at 0x26e730467a0>
```



In [118…
```
plt.scatter(X,y,color='yellow',label='Actual')
plt.scatter(X,y,color='orange',label='Predicated')
plt.plot(X,y,color='red',label='polynomial regression')
plt.xlabel('Aerosols')
plt.ylabel('Temp')
plt.title('CLIMATE CHANGE')
plt.show()
```



LINEARREGRESSION:studentrecord

In [131…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import  train_test_split
from sklearn.metrics import mean_squared_error,r2_score
```

In [132…
```
data=pd.read_csv('studentrecord.csv')
data.head(3)
data.describe()
```

Out[132]:

	subject	weight	age	bloodfat
count	11.000000	11.000000	11.000000	11.000000
mean	6.000000	68.454545	38.090909	328.545455
std	3.316625	14.995757	13.692732	84.575840
min	1.000000	27.000000	20.000000	190.000000
25%	3.500000	67.000000	26.500000	275.500000
50%	6.000000	72.000000	36.000000	354.000000
75%	8.500000	75.500000	49.000000	393.500000
max	11.000000	84.000000	57.000000	451.000000

In [133…
```
X=data[['weight']]
y=data['age']
```

In [134…
```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.2,random_state=42)
```

In [135…
```
X_train_poly=poly.fit_transform(X_train)
X_test_poly=poly.transform(X_test)
```

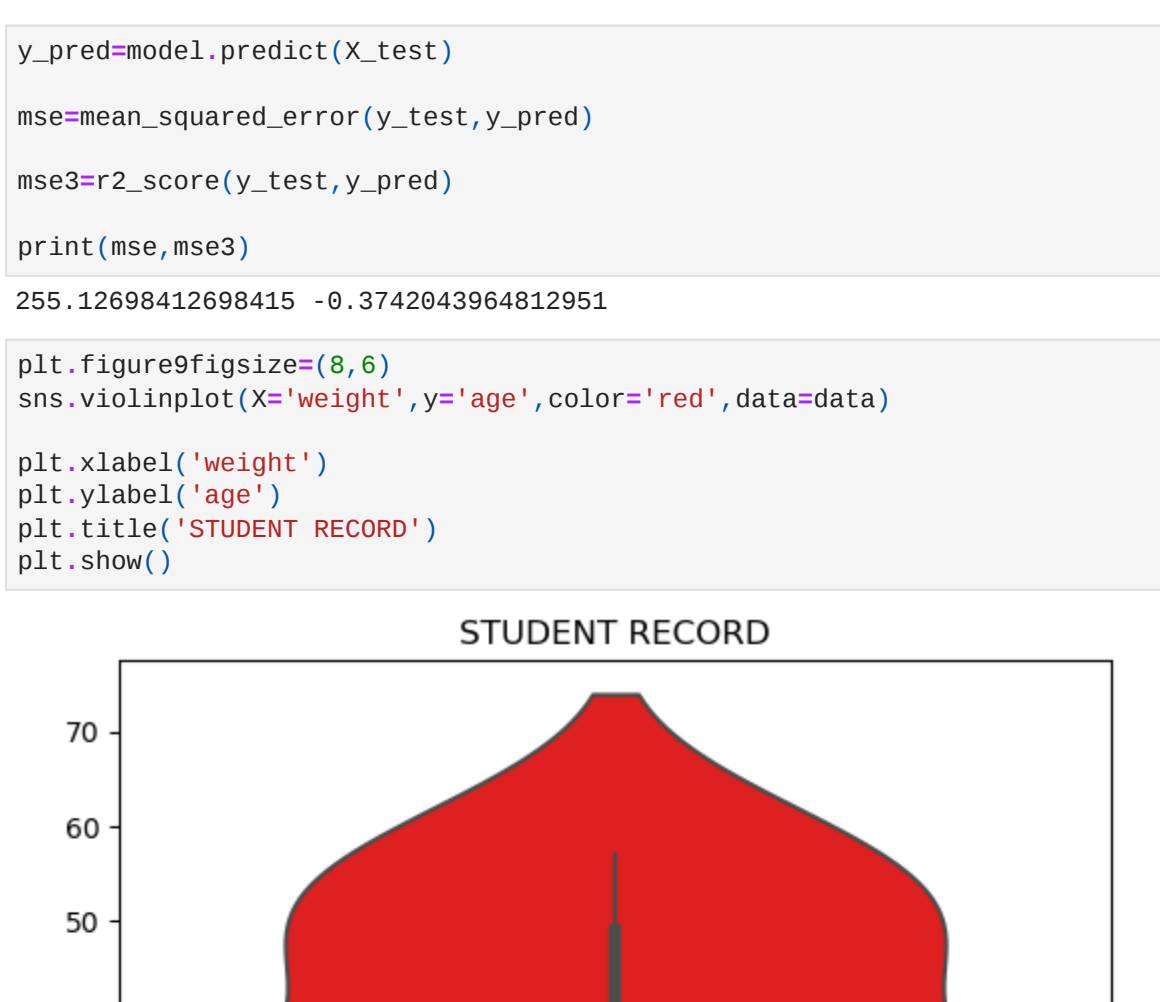
In [136…
```
model=LinearRegression()
model.fit(X_train,y_train)
```

Out[136]:
```
▾ LinearRegression
LinearRegression()
```

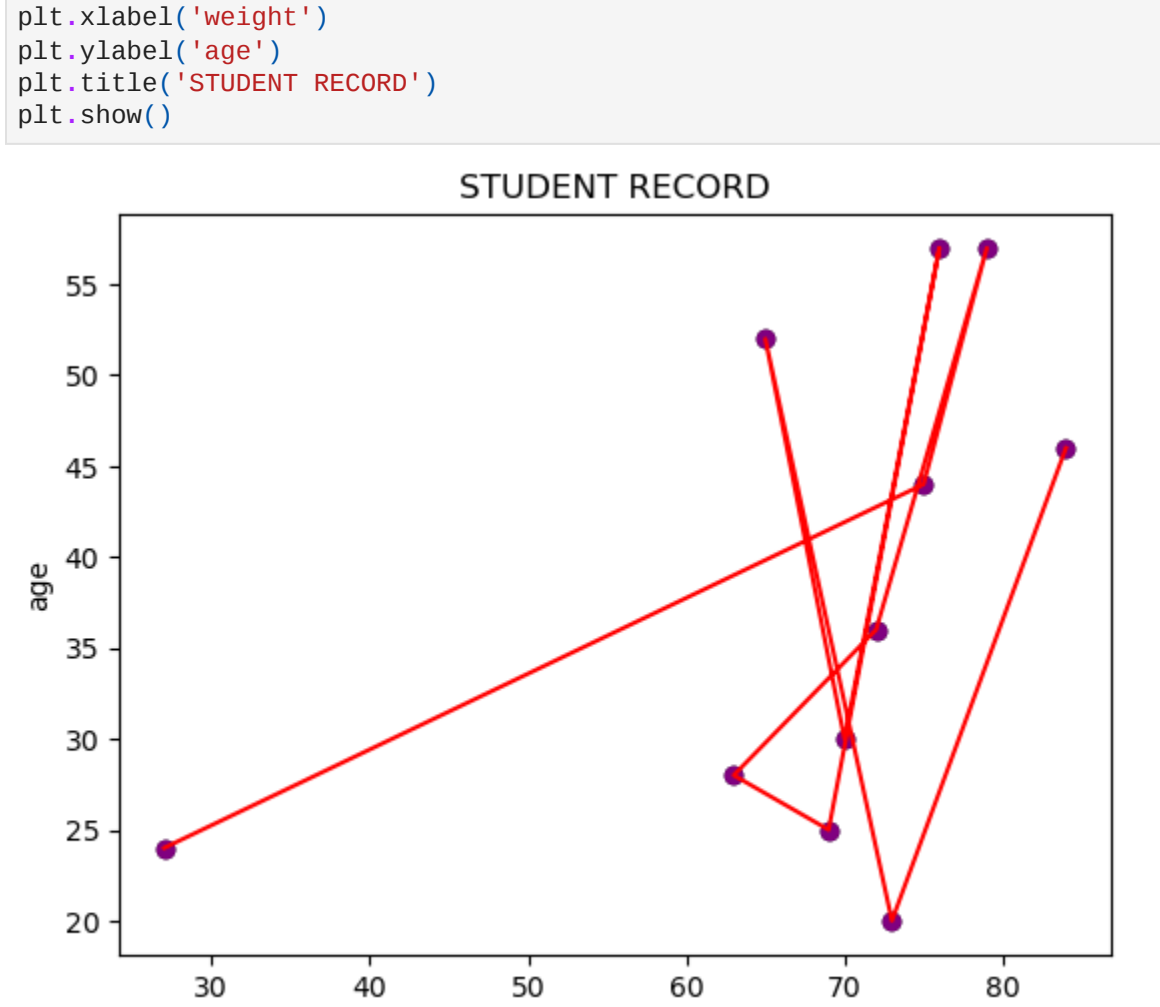
In [138…
```
y_pred=model.predict(X_test)

mse=mean_squared_error(y_test,y_pred)

mse3=r2_score(y_test,y_pred)

print(mse,mse3)
```

```
255.12698412698415 -0.3742043964812951
```

In [144…
```
plt.figure9figsize=(8,6)
sns.violinplot(X='weight',y='age',color='red',data=data)

plt.xlabel('weight')
plt.ylabel('age')
plt.title('STUDENT RECORD')
plt.show()
```



In [146…
```
plt.scatter(X,y,color='black',label='Actual')
plt.scatter(X,y,color='purple',label='Predicated')
plt.plot(X,y,color='red',label='polynomial regression')
plt.xlabel('weight')
plt.ylabel('age')
plt.title('STUDENT RECORD')
plt.show()
```



In [ ]:

In [ ]: