In [3]:

```
#SVR Regression on apple_yield
from sklearn.model_selection import train_test_split
data=pd.read_csv('fruits.csv')
y=data['AppleYield(in kg)']
x_train,x_test,y_train,y_test=train_test_split(data,y,test_size=0.2,ranc
from sklearn.svm import SVR
regressor=SVR(kernel='poly')
regressor.fit(x_train,y_train)
from sklearn.metrics import r2_score
preds=regressor.predict(x_test)
print(r2_score(y_test,preds))
```

```
0.8130783436832618
```

In [4]:

```
#root mean square error
import math
import numpy as np
MSE=np.square(np.subtract(y_test,preds)).mean()
rsme=math.sqrt(MSE)
print("root mean square error:\n")
print(rsme)
```

```
root mean square error:

115.72234549267495
```

In [6]:

```
#SVR Regression on orange_yield
from sklearn.model_selection import train_test_split
data=pd.read_csv('fruits.csv')
y=data['OrangeYield(in kg)']
x_train,x_test,y_train,y_test=train_test_split(data,y,test_size=0.2,ranc
from sklearn.svm import SVR
regressor=SVR(kernel='poly')
regressor.fit(x_train,y_train)
from sklearn.metrics import r2_score
preds=regressor.predict(x_test)
print(r2_score(y_test,preds))
```

```
0.793857598587752
```

In [7]:

```
#root mean square error
import math
import numpy as np
MSE=np.square(np.subtract(y_test,preds)).mean()
rsme=math.sqrt(MSE)
print("root mean square error:\n")
print(rsme)
```

```
root mean square error:

124.45475473902451
```

In [9]:

```
#Linear regression on apple_yield
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
df = pd.read_csv('fruits.csv')
X = df[['Rainfall','Temperature','Humidity']]
y = df['AppleYield(in kg)']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:",mse)
print("R-squared:",r2)
```

```
Mean Squared Error: 76739.28137677771
R-squared: -0.02329849383796323
```

In [10]:

```
#Linear regression on orange_yield
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
df = pd.read_csv('fruits.csv')
X = df[['Rainfall','Temperature','Humidity']]
y = df['OrangeYield(in kg)']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:",mse)
print("R-squared:",r2)
```

```
Mean Squared Error: 67808.84466586633
R-squared: -0.018938892014925157
```

In [11]:

```
#Linear regression prediction on apple_yield
import pandas
from sklearn import linear_model
df=pandas.read_csv("fruits.csv")
x=df[['Rainfall','Temperature','Humidity']]
y=df['AppleYield(in kg)']
regr=linear_model.LinearRegression()
regr.fit(x,y)
predicted=regr.predict([[40,40,20]])
print(predicted)
```

```
[559.14057285]

E:\new\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not
have valid feature names, but LinearRegression was fitted with feature
names
  warnings.warn(
```

In [12]:

```
#Linear regression Coefficient on apple_yield
import pandas
from sklearn import linear_model
df=pandas.read_csv("fruits.csv")
x=df[['Rainfall','Temperature','Humidity']]
y=df['AppleYield(in kg)']
regr=linear_model.LinearRegression()
regr.fit(x,y)
print(regr.coef_)
```

```
[ 0.01486681 -0.2420361  -0.29850883]
```

In [13]:

```
#Linear regression prediction on orange_yield
import pandas
from sklearn import linear_model
df=pandas.read_csv("fruits.csv")
x=df[['Rainfall','Temperature','Humidity']]
y=df['OrangeYield(in kg)']
regr=linear_model.LinearRegression()
regr.fit(x,y)
predicted=regr.predict([[40,40,20]])
print(predicted)
```

```
[560.91003062]

E:\new\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not
have valid feature names, but LinearRegression was fitted with feature
names
  warnings.warn(
```

In [14]:

```
#Linear regression Coefficient on orange_yield
import pandas
from sklearn import linear_model
df=pandas.read_csv("fruits.csv")
x=df[['Rainfall','Temperature','Humidity']]
y=df['OrangeYield(in kg)']
regr=linear_model.LinearRegression()
regr.fit(x,y)
print(regr.coef_)
```

```
[ 0.73307635  0.40699323 -0.82171743]
```

In [ ]: