**Regression Practical(icecream sell acording temperature)**

**Simple Regression**

import numpy as np

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

x=np.array([26,28,32,35,39]).reshape((-1,1))

y=np.array([123,245,296,389,435])

print(x)

print(y)

model=LinearRegression()

model.fit(x,y)

y\_pred=model.predict(x)

print("Predicted responce is",y\_pred)

y\_new=model.predict(x)

y\_new

x\_new=np.array([40]).reshape((-1,1))

y\_new=model.predict(x\_new)

y\_new

print("predicted outcome is ",y\_new)

plt.scatter(x, y)

plt.plot(x, y\_pred)

plt.show()

**Output:**

[[26]

[28]

[32]

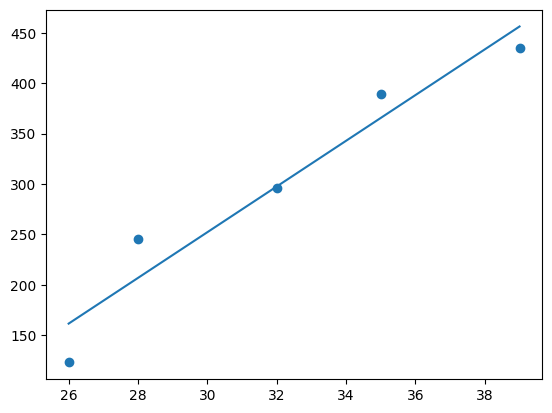
[35]

[39]]

[123 245 296 389 435]

Predicted responce is [161.56363636 206.90909091 297.6 365.61818182 456.30909091]

predicted outcome is [478.98181818]



**Multiple Regression**

import numpy as np

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

# Input data

X = np.array([[26, 3],

              [28, 4],

              [32, 3.5],

              [35, 5],

              [39, 6]])

y = np.array([123, 245, 296, 389, 435])

# Creating and fitting the model

model = LinearRegression()

model.fit(X, y)

# Predictions

y\_pred = model.predict(X)

# Plotting

plt.scatter(y, y\_pred)

plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)  # Plotting the diagonal line

plt.xlabel("True Values")

plt.ylabel("Predictions")

plt.title("True vs Predicted Values")

plt.show()

**Output:**

