Experiment-10:

a) Linear Regression b) Logistic Regression

LINEAR REGRESSION

In [7]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_error
from sklearn.model_selection import train_test_split
```

In [3]:
```
df = pd.read_csv("Advertising.csv")
X = df.drop('sales',axis=1)
y = df['sales']
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=101)
```

In [4]:
```
model = LinearRegression()
model.fit(X_train,y_train)
```

Out[4]:
```
▾ LinearRegression

LinearRegression()
```

In [8]:
```
test_predictions = model.predict(X_test)
MAE = mean_absolute_error(y_test,test_predictions)
MSE = mean_squared_error(y_test,test_predictions)
RMSE = np.sqrt(MSE)
print("Mean Absolute Error:",MAE)
print("Mean Squared Error:",MSE)
print("Root Mean Squared Error:",RMSE)
```

```
Mean Absolute Error: 1.2137457736144805
Mean Squared Error: 2.298716697886378
Root Mean Squared Error: 1.5161519375993877
```

LOGISTIC REGRESSION

In [9]:
```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score,classification_report
```

In [11]:
```
df = pd.read_csv('hearing_test.csv')
X = df.drop('test_result',axis=1)
y = df['test_result']
X_train,X_test,y_train,y_test = train_test_split(X, y,test_size=0.1,random_state=101)
```

In [12]:
```
scaler = StandardScaler()
scaled_X_train = scaler.fit_transform(X_train)
scaled_X_test = scaler.transform(X_test)
```

In [13]:
```
log_model = LogisticRegression()
log_model.fit(scaled_X_train,y_train)
```

Out[13]:

▾ LogisticRegression

LogisticRegression()

In [14]:
```
y_pred = log_model.predict(scaled_X_test)
print("Accuracy Score :",accuracy_score(y_test,y_pred))
print("Classification Report :\n",classification_report(y_test,y_pred))
```

```
Accuracy Score : 0.93
Classification Report :
              precision    recall  f1-score   support

           0       0.92      0.89      0.91       193
           1       0.93      0.95      0.94       307

    accuracy                           0.93       500
   macro avg       0.93      0.92      0.93       500
weighted avg       0.93      0.93      0.93       500
```

In [ ]: