```
In [91]: import pandas as pd
    ...:
    ...:
    ...:
    ...: df=pd.read_csv('G:\Data Analysis\output.csv')
    ...: df=df.dropna()
    ...: X = df.iloc[:,8:22].values
    ...: y = df.iloc[:, 1].values
    ...: for i in range(len(y)):
    ...:     y[i]=int(y[i])
    ...:
    ...:
    ...:
    ...: from sklearn.preprocessing import LabelEncoder, OneHotEncoder
    ...: labelencoder_X_1 = LabelEncoder()
    ...: X[:, 1] = labelencoder_X_1.fit_transform(X[:, 1])
    ...: labelencoder_X_4 = LabelEncoder()
    ...: X[:, 4] = labelencoder_X_4.fit_transform(X[:, 4])
    ...: labelencoder_X_5= LabelEncoder()
    ...: X[:, 5] = labelencoder_X_5.fit_transform(X[:, 5])
    ...: labelencoder_X_7 = LabelEncoder()
    ...: X[:, 7] = labelencoder_X_7.fit_transform(X[:, 7])
    ...: labelencoder_X_8= LabelEncoder()
    ...: X[:, 8] = labelencoder_X_8.fit_transform(X[:, 8])
    ...: labelencoder_X_10 = LabelEncoder()
    ...: X[:, 10] = labelencoder_X_10.fit_transform(X[:, 10])
    ...: labelencoder_X_12= LabelEncoder()
    ...: X[:, 12] = labelencoder_X_12.fit_transform(X[:, 12])
    ...: labelencoder_X_13 = LabelEncoder()
    ...: X[:, 13] = labelencoder_X_13.fit_transform(X[:, 13])
    ...:
    ...:
    ...:
    ...: onehotencoder = OneHotEncoder(categorical_features=[1,4,5,7,8,10,12])
    ...: X = onehotencoder.fit_transform(X).toarray()
    ...:
    ...: X=X[:,1:]
    ...:
    ...: from sklearn.model_selection import train_test_split
    ...: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4,
random_state = 0)
    ...:
    ...: from sklearn.neighbors import KNeighborsClassifier
    ...:
    ...: classifier = KNeighborsClassifier(n_neighbors = 10, metric = 'minkowski', p = 2)
    ...: classifier.fit(X_train, y_train)
    ...:
    ...: c=0
    ...: y_pred = classifier.predict(X_test)
    ...: for i in range(len(y_pred)):
    ...:     y_pred[i]=int(y_pred[i])
    ...:     y_pred[i]=(y_pred[i]<=y_test[i]+2  and y_pred[i]>=y_test[i]-2)
    ...:     if(y_pred[i]):
    ...:         c+=1
    ...:
    ...: acc=float(c/len(y_test))
C:\Users\HP\Music\Anaconda3\lib\site-packages\sklearn\preprocessing\_encoders.py:371:
```

```
FutureWarning: The handling of integer data will change in version 0.22. Currently, the
categories are determined based on the range [0, max(values)], while in the future they will
be determined based on the unique values.
If you want the future behaviour and silence this warning, you can specify
"categories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the categories to
integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
C:\Users\HP\Music\Anaconda3\lib\site-packages\sklearn\preprocessing\_encoders.py:392:
DeprecationWarning: The 'categorical_features' keyword is deprecated in version 0.20 and
will be removed in 0.22. You can use the ColumnTransformer instead.
  "use the ColumnTransformer instead.", DeprecationWarning)

In [92]: acc
Out[92]: 0.8856485034535687

In [93]:
```