To perform and analysis of Logistic Regression Algorithm

```python
In [1]: import pandas as pd
        import numpy as np
```

Data acquisitionuing Pandas

```python
In [4]: import os
```

```python
In [5]: os.getcwd()
```

```
Out[5]: 'C:\\Users\\RH'
```

```python
In [7]: os.chdir('C:\\Users\\RH\\OneDrive\\Desktop')
```

```python
In [12]: data=pd.read_csv("C:\\Users\RH\\Downloads\\heart - heart.csv")
```

```python
In [13]: data.head()
```

Out[13]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

```python
In [14]: data.tail()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

In [15]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

In [16]:
```python
data.describe()
```

Out[16]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	102
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ►

In [17]:
```python
data.shape
```

Out[17]: (1025, 14)

In [18]:
```python
data.size
```

Out[18]: 14350

In [19]:
```python
data.ndim
```

Out[19]: 2

Data preprocessing data cleaning missing value treatment

In [21]:
```python
# check Missing Value by record

data.isna()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1021	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1022	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1023	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1024	False	False	False	False	False	False	False	False	False	False	False	False	False	False

1025 rows × 14 columns

```python
data.isna().any()
```

```
Out[22]: age         False
         sex         False
         cp          False
         trestbps    False
         chol        False
         fbs         False
         restecg     False
         thalach     False
         exang       False
         oldpeak     False
         slope       False
         ca          False
         thal        False
         target      False
         dtype: bool
```

In [23]: ```python
data.isna().sum()
```

```
Out[23]: age         0
         sex         0
         cp          0
         trestbps    0
         chol        0
         fbs         0
         restecg     0
         thalach     0
         exang       0
         oldpeak     0
         slope       0
         ca          0
         thal        0
         target      0
         dtype: int64
```

Independent and Dependent Variables

In [25]: ```python
x=data.drop("target", axis=1)
y=data["target"]
```

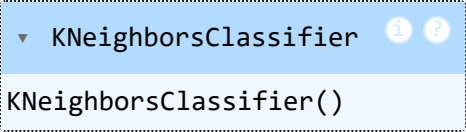# Splitting of DataSet into train and Test

```python
In [28]: #splitting the data into training and testing data sets
         from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2 ,random_state=42)
```

KNN Classifier

```python
In [30]: from sklearn.neighbors import KNeighborsClassifier
         from sklearn.metrics import accuracy_score
```

```python
In [31]: knn=KNeighborsClassifier()
```

```python
In [32]: knn.fit(x_train, y_train)
```

```
Out[32]: ▾ KNeighborsClassifier  ⓘ ❓
         KNeighborsClassifier()
```

```python
In [33]: y_pred2=knn.predict(x_test)
```

```python
In [34]: accuracy = accuracy_score(y_test, y_pred2)
```

```python
In [35]: accuracy
```

```
Out[35]: 0.7317073170731707
```

```python
In [ ]:
```