```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
matplotlib.style.use('ggplot')
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
dataset = pd.read_excel('1645792390_cep1_dataset.xlsx')
dataset.shape
```

```
(303, 14)
```

```
# View the dataset
dataset.head(10)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

Structure of data

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
```

```
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

```
dataset.duplicated().sort_values()
```

```
0      False
205    False
204    False
203    False
202    False
       ...
97     False
96     False
102    False
302    False
164     True
Length: 303, dtype: bool
```

```
dataset.drop_duplicates(inplace=True)
dataset.duplicated().any()
```

```
False
```

```
dataset.describe(include='all').T
```

	count	mean	std	min	25%	50%	75%	max
age	302.0	54.420530	9.047970	29.0	48.00	55.5	61.00	77.0
sex	302.0	0.682119	0.466426	0.0	0.00	1.0	1.00	1.0
cp	302.0	0.963576	1.032044	0.0	0.00	1.0	2.00	3.0
trestbps	302.0	131.602649	17.563394	94.0	120.00	130.0	140.00	200.0
chol	302.0	246.500000	51.753489	126.0	211.00	240.5	274.75	564.0
fbs	302.0	0.149007	0.356686	0.0	0.00	0.0	0.00	1.0
restecg	302.0	0.526490	0.526027	0.0	0.00	1.0	1.00	2.0
thalach	302.0	149.569536	22.903527	71.0	133.25	152.5	166.00	202.0
exang	302.0	0.327815	0.470196	0.0	0.00	0.0	1.00	1.0
oldpeak	302.0	1.043046	1.161452	0.0	0.00	0.8	1.60	6.2
slope	302.0	1.397351	0.616274	0.0	1.00	1.0	2.00	2.0
ca	302.0	0.718543	1.006748	0.0	0.00	0.0	1.00	4.0
thal	302.0	2.314570	0.613026	0.0	2.00	2.0	3.00	3.0
target	302.0	0.543046	0.498970	0.0	0.00	1.0	1.00	1.0

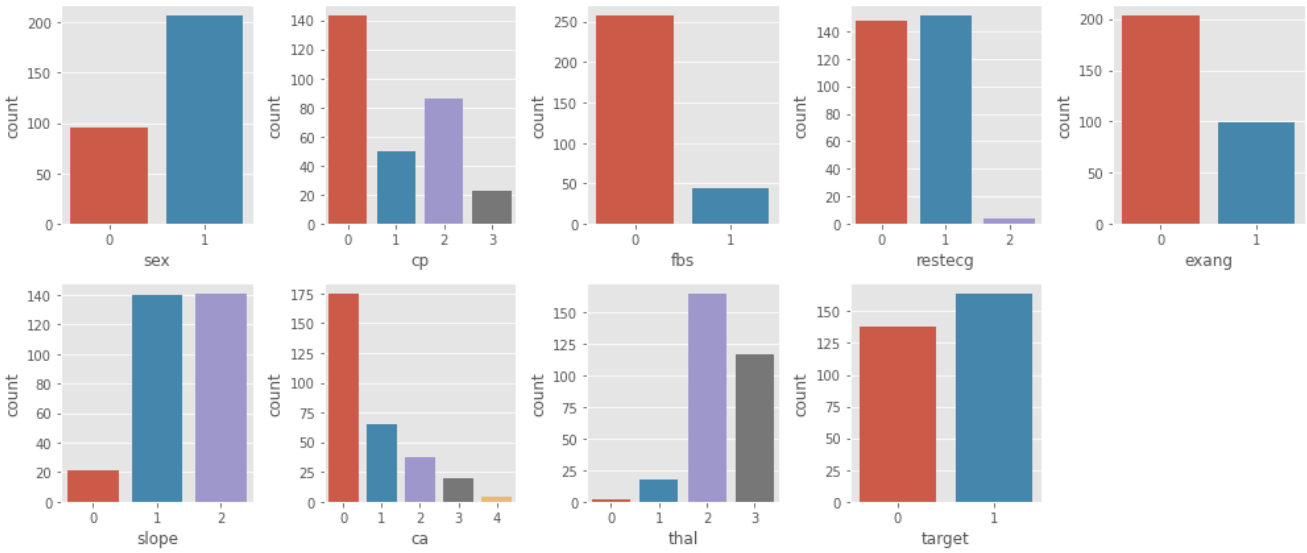
```
categorical_features=[]
def filter_cat(cat):
    for i in cat:
        if dataset[i].nunique() < 20:
            categorical_features.append(i)
filter_cat(dataset.columns)
categorical_features
```

```
['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']
```

```
dataset[categorical_features].nunique()
```

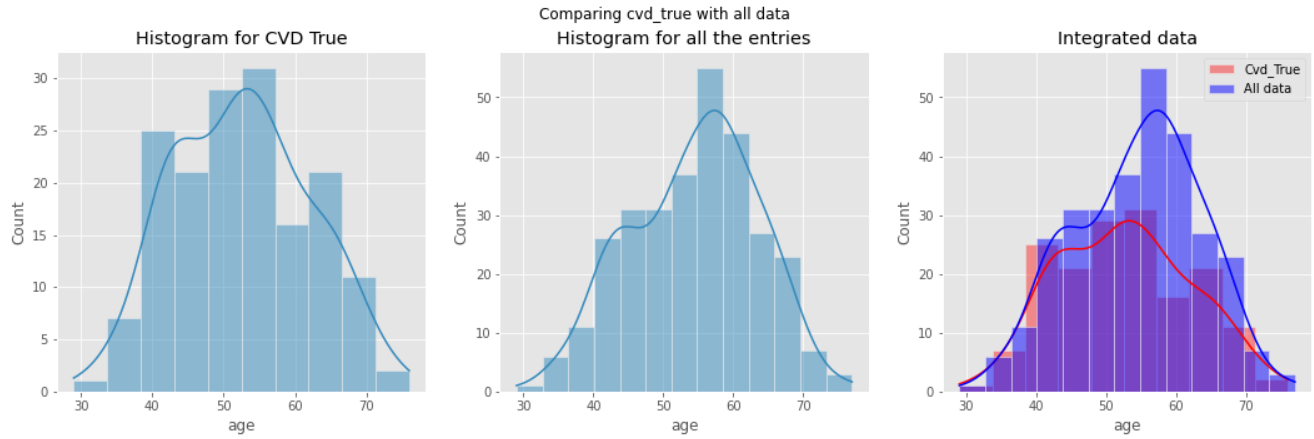
```
sex        2
cp         4
fbs        2
restecg    3
exang      2
slope      3
ca         5
thal       4
target     2
dtype: int64
```

```
plt.figure(figsize=(14,6))
for i,features in enumerate(categorical_features):
    plt.subplot(2,5,i+1)
    sns.countplot(x=dataset[features])
    plt.tight_layout()
```




```
cvd_true = dataset.loc[dataset.target == 1]
```

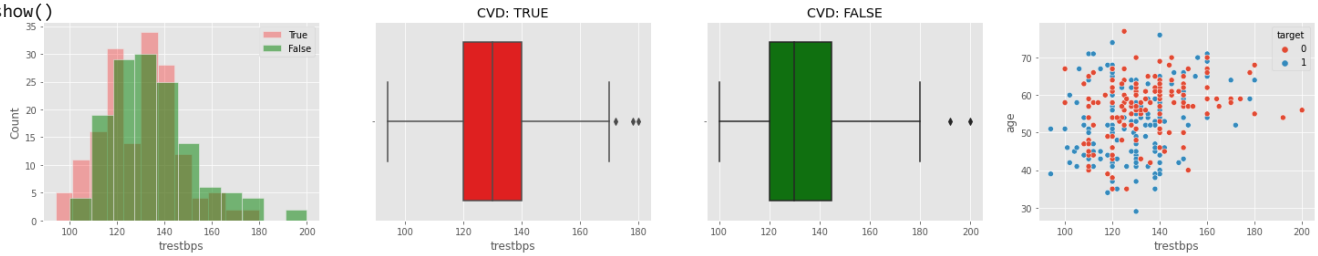
```
plt.figure(figsize=(18,5))
plt.subplot(1,3,1)
sns.histplot(x=cvd_true.age,kde=True)
plt.title('Histogram for CVD True')
plt.subplot(1,3,2)
sns.histplot(x=dataset.age,kde=True)
plt.title('Histogram for all the entries')
plt.subplot(1,3,3)
sns.histplot(x=cvd_true.age,kde=True,label='Cvd_True',color='red',alpha=0.4)
sns.histplot(x=dataset.age,kde=True,label='All data',color='blue',alpha=0.5)
plt.title('Integrated data')
plt.legend()
plt.suptitle('Comparing cvd_true with all data')
plt.show()
```




```
cvd_false = dataset.loc[dataset.target == 0]
```

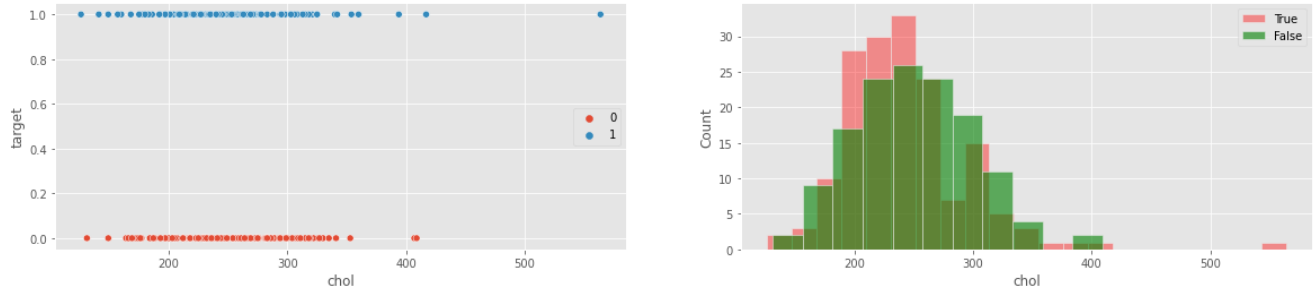
```
plt.figure(figsize=(25,4))
# PLOTTING SUBPLOT_1
plt.subplot(1,4,1)
sns.histplot(cvd_true.trestbps,color='red',alpha=0.3, label='True')
sns.histplot(cvd_false.trestbps,color='green',alpha=0.5,label='False')
plt.legend()
# PLOTTING SUBPLOT_2
plt.subplot(1,4,2)
sns.boxplot(cvd_true.trestbps,color='red')
plt.title("CVD: TRUE")

# PLOTTING SUBPLOT_3
plt.subplot(1,4,3)
sns.boxplot(cvd_false.trestbps.values,color='green')
plt.title("CVD: FALSE")

plt.subplot(1,4,4)
```

```
sns.scatterplot(y=dataset.age, x=dataset.trestbps,hue=dataset.target)
plt.show()
```



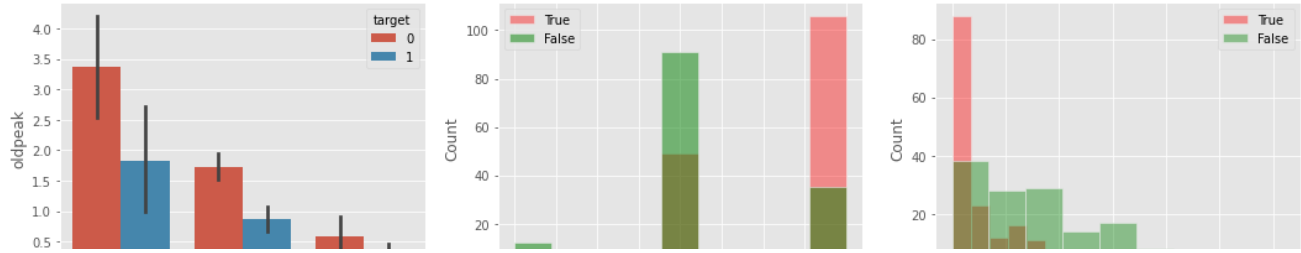

```
plt.figure(figsize=(20,4))
plt.subplot(121)
sns.scatterplot(x=dataset.chol, y=dataset.target, hue=dataset.target)
plt.legend(loc='center right')

plt.subplot(122)
sns.histplot(cvd_true.chol, color='r',alpha=0.4,label='True')
sns.histplot(cvd_false.chol, color='green',alpha=0.6,label='False')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fe71ee76c10>
```



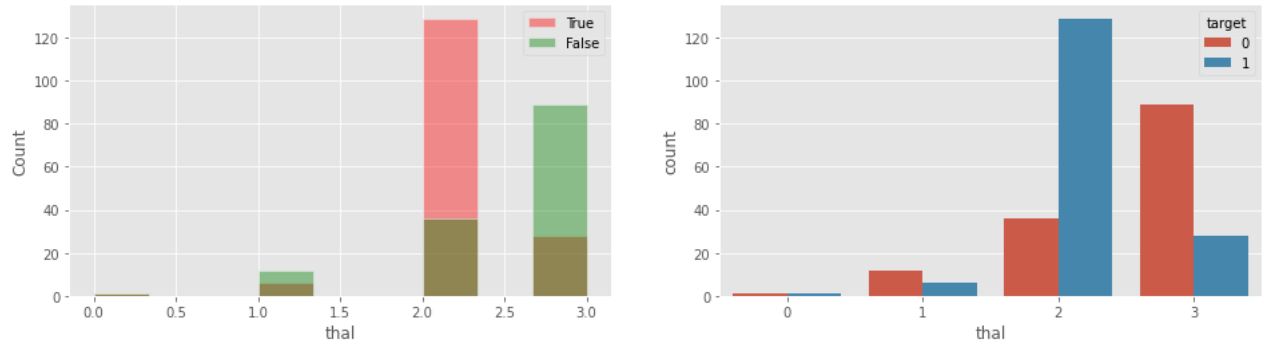

```
plt.figure(figsize=(18,4))
plt.subplot(1,3,1)
sns.barplot(y=dataset.oldpeak, x=dataset.slope, hue=dataset.target)
plt.subplot(1,3,2)
sns.histplot(cvd_true.slope,color='red',alpha=0.4,label='True')
sns.histplot(cvd_false.slope,color='green',alpha=0.5,label='False')
plt.legend()
plt.subplot(1,3,3)
sns.histplot(cvd_true.oldpeak,color='red',alpha=0.4,label='True')
sns.histplot(cvd_false.oldpeak,color='green',alpha=0.4,label='False')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fe721206c40>
```



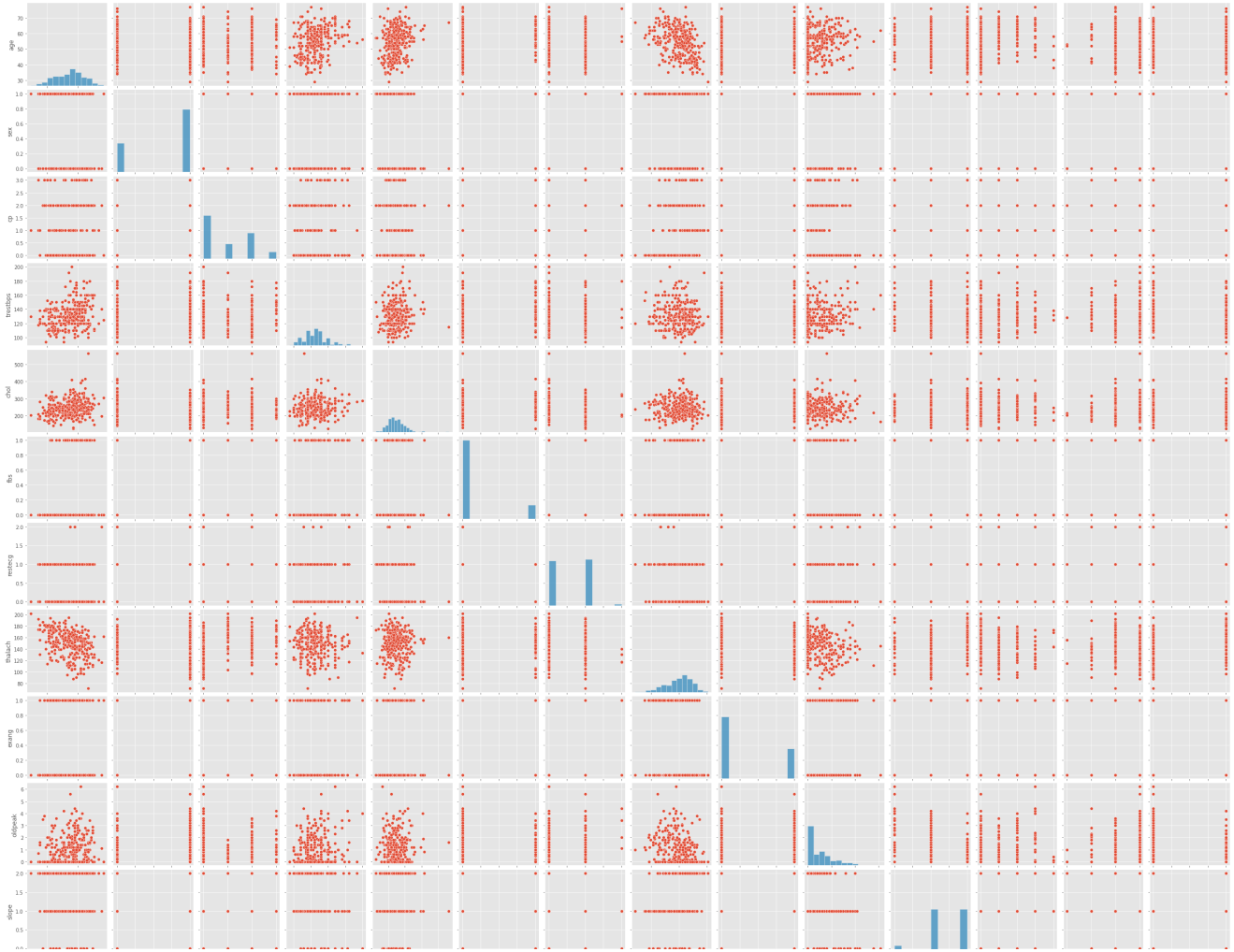

```
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
sns.histplot(cvd_true.thal,color='red',alpha=0.4,label='True')
sns.histplot(cvd_false.thal,color='green',alpha=0.4,label='False')
plt.legend()
plt.subplot(1,2,2)
sns.countplot(x=dataset.thal,hue=dataset.target)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe721876b20>
```



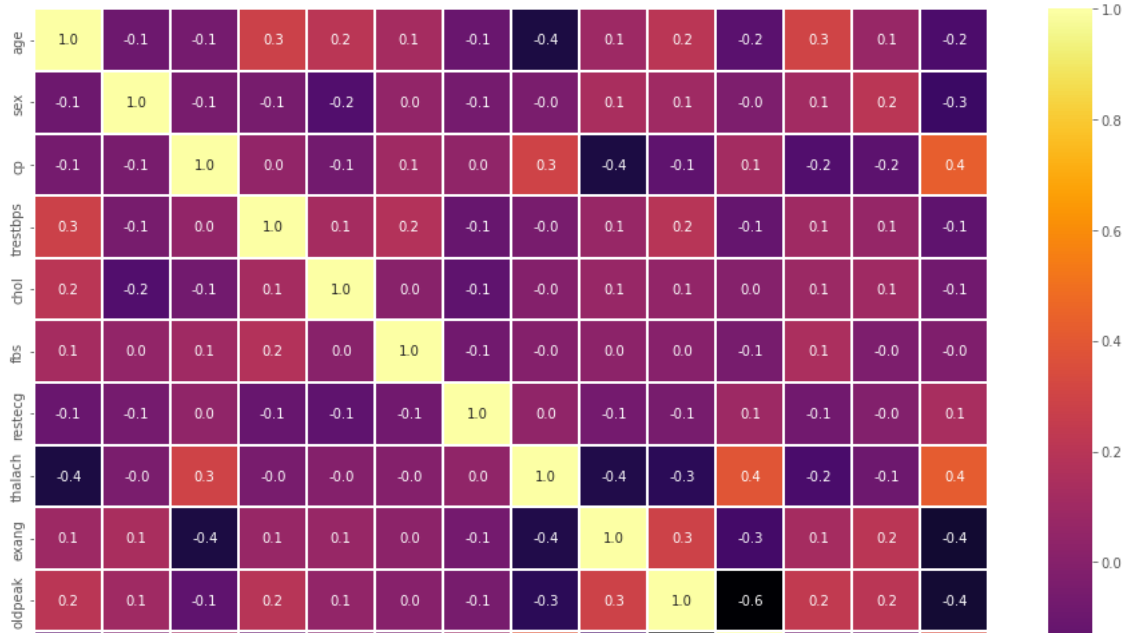

```
sns.pairplot(data=dataset)
```

```
<seaborn.axisgrid.PairGrid at 0x7fe721828d90>
```




```
plt.figure(figsize=(16,12))
sns.heatmap(dataset.corr(),annot=True, fmt='.1f', linecolor='white',linewidths= 1.001,cmap='inferno')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe71bbb1940>
```




```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
```

```
y = dataset.pop('target')
X = dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 4)

print('X_train:',X_train.shape)
print('X_test :',X_test.shape)
print('y_train:',y_train.shape)
print('y_test :',y_test.shape)
```

```
X_train: (211, 13)
X_test : (91, 13)
y_train: (211,)
y_test : (91,)
```

```
import statsmodels.api as sm
logreg = sm.Logit(y_train,X_train).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.349166
         Iterations 7
```

```
print(logreg.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                 target   No. Observations:                  211
Model:                          Logit   Df Residuals:                      198
Method:                           MLE   Df Model:                           12
Date:                Sat, 28 Jan 2023   Pseudo R-squ.:                  0.4953
Time:                        04:52:40   Log-Likelihood:                -73.674
converged:                       True   LL-Null:                       -145.97
Covariance Type:            nonrobust   LLR p-value:                 7.087e-25
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
age            0.0359      0.025      1.443      0.149      -0.013       0.085
sex           -1.6155      0.527     -3.067      0.002      -2.648      -0.583
cp             0.7427      0.226      3.293      0.001       0.301       1.185
trestbps      -0.0097      0.012     -0.796      0.426      -0.034       0.014
chol          -0.0032      0.004     -0.738      0.461      -0.012       0.005
fbs            0.2782      0.706      0.394      0.694      -1.106       1.662
restecg        0.5689      0.406      1.402      0.161      -0.227       1.364
thalach        0.0204      0.010      2.074      0.038       0.001       0.040
exang         -1.1044      0.489     -2.259      0.024      -2.062      -0.146
oldpeak       -0.4253      0.277     -1.534      0.125      -0.969       0.118
slope          0.5908      0.442      1.337      0.181      -0.275       1.457
ca            -1.0731      0.290     -3.701      0.000      -1.641      -0.505
thal          -0.8927      0.350     -2.549      0.011      -1.579      -0.206
==============================================================================
```

```
new_features  = logreg.pvalues[logreg.pvalues <= .6]
new_X = dataset[new_features.index]
new_X.head()
```

	age	sex	cp	trestbps	chol	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	0	150	0	2.3	0	0	1
1	37	1	2	130	250	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	172	0	1.4	2	0	2
3	56	1	1	120	236	1	178	0	0.8	2	0	2
4	57	0	0	120	354	1	163	1	0.6	2	0	2

```
new_features.index
```

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'restecg', 'thalach', 'exang',
       'oldpeak', 'slope', 'ca', 'thal'],
      dtype='object')
```

```
X1_train, X1_test, y1_train, y1_test = train_test_split(new_X, y, test_size = 0.3, random_state=5)

print('X1_train:',X1_train.shape)
print('X1_test :',X1_test.shape)
print('y1_train:',y1_train.shape)
print('y1_test :',y1_test.shape)
```

```
X1_train: (211, 12)
X1_test : (91, 12)
```

```
y1_train: (211,)
y1_test : (91,)
```

```
logreg1 = sm.Logit(y1_train, X1_train).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.334213
         Iterations 7
```

```
print(logreg1.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                 target   No. Observations:                  211
Model:                          Logit   Df Residuals:                      199
Method:                           MLE   Df Model:                           11
Date:                Sat, 28 Jan 2023   Pseudo R-squ.:                  0.5144
Time:                        04:53:55   Log-Likelihood:                -70.519
converged:                       True   LL-Null:                       -145.21
Covariance Type:            nonrobust   LLR p-value:                 1.998e-26
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
age            0.0302      0.023      1.289      0.197      -0.016       0.076
sex           -1.7057      0.585     -2.918      0.004      -2.851      -0.560
cp             1.0425      0.238      4.388      0.000       0.577       1.508
trestbps      -0.0162      0.012     -1.331      0.183      -0.040       0.008
chol          -0.0055      0.004     -1.221      0.222      -0.014       0.003
restecg        0.8602      0.443      1.944      0.052      -0.007       1.727
thalach        0.0307      0.011      2.831      0.005       0.009       0.052
exang         -1.5725      0.513     -3.065      0.002      -2.578      -0.567
oldpeak       -0.5475      0.265     -2.066      0.039      -1.067      -0.028
slope          0.4170      0.424      0.984      0.325      -0.414       1.248
ca            -0.8438      0.230     -3.674      0.000      -1.294      -0.394
thal          -0.8279      0.340     -2.434      0.015      -1.495      -0.161
==============================================================================
```

```
new_features  = logreg1.pvalues[logreg1.pvalues <= .3]
new_X = dataset[new_features.index]
new_X.head()
```

	age	sex	cp	trestbps	chol	restecg	thalach	exang	oldpeak	ca	thal
0	63	1	3	145	233	0	150	0	2.3	0	1
1	37	1	2	130	250	1	187	0	3.5	0	2
2	41	0	1	130	204	0	172	0	1.4	0	2
3	56	1	1	120	236	1	178	0	0.8	0	2
4	57	0	0	120	354	1	163	1	0.6	0	2

```
X2_train, X2_test, y2_train, y2_test = train_test_split(new_X, y, test_size = 0.3, random_state=5)

print('X2_train:',X2_train.shape)
print('X2_test :',X2_test.shape)
```

```
print('y2_train:',y2_train.shape)
print('y2_test :',y2_test.shape)
```

```
X2_train: (211, 11)
X2_test : (91, 11)
y2_train: (211,)
y2_test : (91,)
```

```
logreg2 = sm.Logit(y2_train, X2_train).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.336456
         Iterations 7
```

```
print(logreg2.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                 target   No. Observations:                  211
Model:                          Logit   Df Residuals:                      200
Method:                           MLE   Df Model:                           10
Date:                Sat, 28 Jan 2023   Pseudo R-squ.:                  0.5111
Time:                        04:54:50   Log-Likelihood:                -70.992
converged:                       True   LL-Null:                       -145.21
Covariance Type:            nonrobust   LLR p-value:                 7.840e-27
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
age            0.0326      0.023      1.400      0.161      -0.013       0.078
sex           -1.6789      0.580     -2.894      0.004      -2.816      -0.542
cp             1.0214      0.236      4.335      0.000       0.560       1.483
trestbps      -0.0158      0.012     -1.296      0.195      -0.040       0.008
chol          -0.0055      0.005     -1.222      0.222      -0.014       0.003
restecg        0.8807      0.441      1.995      0.046       0.015       1.746
thalach        0.0337      0.010      3.216      0.001       0.013       0.054
exang         -1.5866      0.514     -3.086      0.002      -2.594      -0.579
oldpeak       -0.6719      0.236     -2.847      0.004      -1.134      -0.209
ca            -0.8083      0.223     -3.628      0.000      -1.245      -0.372
thal          -0.8185      0.341     -2.403      0.016      -1.486      -0.151
==============================================================================
```

✓ 0s completed at 10:18 AM