Examining Factors Responsible for Heart Attacks Project Write Up

1. Import the required libraries and data in the python.

```
# import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df= pd.read_excel("data.xlsx")
```

2. Perform preliminary data inspection and report the findings as to the structure of the data, missing values, duplicates, etc.Based on the findings from the previous question remove duplicates (if any) , treat missing values using an appropriate strategy.

```
df.isna().sum()
df.duplicated().sum()
df.drop_duplicates(inplace=True)
```

3. Get a preliminary statistical summary of the data. Explore the measures of central tendencies and the spread of the data overall.

```
df.describe()
```

4. Identify the data variables which might be categorical in nature. Describe and explore these variables using appropriate tools e.g. count plot

```
df.columns
df.nunique()

cat_var=['sex', 'cp','fbs', 'restecg','exang','slope', 'ca', 'thal']

fig,axes=plt.subplots(nrows=3,ncols=3,figsize=(30,10))
for ind,feature in enumerate(cat_var):
    ax=axes[int(ind/3),ind%3]
    sns.countplot(x=feature,data=df,hue='target',ax=ax,palette='Set2')
plt.show()
```

5. Study the occurrence of CVD across Age.

```
plt.figure(figsize=(12,8))
sns.histplot(x='age',hue='target',data=df)
plt.show()
```

6. Study the composition of overall patients w.r.t. Gender.

```
male_t=sum((df['sex']==1) & (df['target']==1))
female_t=sum((df['sex']==0) & (df['target']==1))
male_n=sum((df['sex']==1) & (df['target']==0))
female_n=sum((df['sex']==0) & (df['target']==0))

t_1=[male_t,female_t]
t_0=[male_n,female_n]

label=['Male','Female']
colors=['blue','pink']
plt.subplot(1,2,1)
plt.pie(t_1,labels=label,autopct="%.1f%%",colors=colors)
plt.title("Having heart disease")
plt.subplot(1,2,2)
plt.pie(t_0,labels=label,autopct="%.1f%%",colors=colors)
plt.title("Not having heart disease")
```

7. Can we detect a heart attack based on anomalies in the Resting Blood Pressure of the patient?
8. Describe the relationship between Cholesterol levels and our target variable.
9. What can be concluded about the relationship between peak exercising and the occurrence of a heart attack.
10. Is thalassemia a major cause of CVD?
11. How are the other factors determining the occurrence of CVD?

```
plt.figure(figsize=(12,8))
sns.heatmap(df.corr(),annot=True)

th_3_t=sum((df['thal']==1) & (df['target']==1))
th_6_t=sum((df['thal']==2) & (df['target']==1))
th_7_t=sum((df['thal']==3) & (df['target']==1))
th_3_n=sum((df['thal']==1) & (df['target']==0))
th_6_n=sum((df['thal']==2) & (df['target']==0))
th_7_n=sum((df['thal']==3) & (df['target']==0))

cvd_1_th=[th_3_t,th_6_t,th_7_t]
cvd_0_th=[th_3_n,th_6_n,th_7_n]
print(cvd_1_th)
print(cvd_0_th)

label=['Normal','Fixed Defect','Reversible Defect']
color=['pink','orange','purple']
plt.subplot(1,2,1)
plt.pie(cvd_1_th,labels=label,autopct="%.1f%%",colors=color)
plt.title("Having heart disease")
plt.subplot(1,2,2)
plt.pie(cvd_0_th,labels=label,autopct="%.1f%%",colors=color)
plt.title("Not having heart disease")

num_var=['age','trestbps','chol','thalach','oldpeak']
```

```
sns.pairplot(df[num_var+['target']],hue='target')
```

12. Perform logistic regression, predict the outcome for test data, and validate the results by using the confusion matrix.

```
column=['age', 'sex','trestbps', 'chol', 'fbs', 'restecg','target','slope']
x=df.drop(columns=column,axis=1)
y=df['target']

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_train_scaled=scaler.fit_transform(x_train)
x_test_scaled=scaler.fit_transform(x_test)

from sklearn.linear_model import LogisticRegression
model=LogisticRegression(random_state=0)
model.fit(x_train_scaled,y_train)

coeff=model.coef_[0]
coeff_table=pd.DataFrame(coeff,x_train.columns,columns=['coefficient'])

y_pred=model.predict(x_test_scaled)

y_actual=y_test.values

result=pd.DataFrame({'Actual target value':y_actual,'Predicted target value':y_pred})

from sklearn import metrics
confusion_matrix = metrics.confusion_matrix(y_test,y_pred)
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix,
display_labels = [False, True])
cm_display.plot()
plt.show()

Accuracy = metrics.accuracy_score(y_test, y_pred)
print("Model Accuracy = ",Accuracy)
```