Heart attack analysis using Machine Learning algorithms [Random Forest Classifier]

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

```
df=pd.read_csv("heart.csv")
df
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	
1	37	1	2	130	250	0	1	187	0	3.5	0	0	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	0	0	140	241	0	1	123	1	0.2	1	0	
299	45	1	3	110	264	0	1	132	0	1.2	1	0	
300	68	1	0	144	193	1	1	141	0	3.4	1	2	
301	57	1	0	130	131	0	1	115	1	1.2	1	1	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	

303 rows × 14 columns

```
df.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   trtbps    303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalachh  303 non-null    int64  
 8   exng      303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64  
 11  caa       303 non-null    int64  
 12  thall     303 non-null    int64  
 13  output    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

```
df.describe()
df.isnull()
```

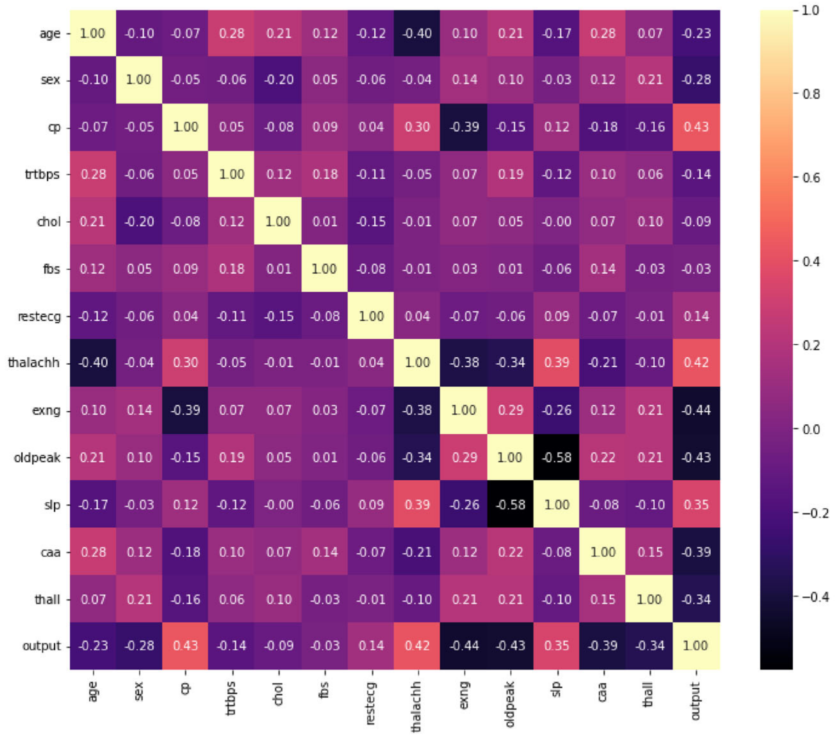
```
       age    sex    cp     trtbps  chol   fbs    restecg  thalachh  exng   oldpeak  sl
 0     False  False  False  False   False  False  False    False     False  False    False
 1     False  False  False  False   False  False  False    False     False  False    False
 2     False  False  False  False   False  False  False    False     False  False    False
 3     False  False  False  False   False  False  False    False     False  False    False
 4     False  False  False  False   False  False  False    False     False  False    False
 ...   ...    ...    ...    ...     ...    ...    ...      ...       ...    ...      ..
 298   False  False  False  False   False  False  False    False     False  False    False
 299   False  False  False  False   False  False  False    False     False  False    False
 300   False  False  False  False   False  False  False    False     False  False    False
 301   False  False  False  False   False  False  False    False     False  False    False
 302   False  False  False  False   False  False  False    False     False  False    False

303 rows × 14 columns
```

```
df.isnull().sum()
```

```
age         0
sex         0
cp          0
trtbps      0
chol        0
fbs         0
restecg     0
thalachh    0
exng        0
oldpeak     0
slp         0
caa         0
thall       0
output      0
dtype: int64
```

```
plt.figure(figsize=(12,10))
sns.heatmap(df.corr(),annot=True,cmap="magma",fmt='.2f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f199d41ae20>
```



```
x=df.iloc[:,:-1]
x
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	t
0	63	1	3	145	233	1	0	150	0	2.3	0	0	
1	37	1	2	130	250	0	1	187	0	3.5	0	0	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	0	0	140	241	0	1	123	1	0.2	1	0	
299	45	1	3	110	264	0	1	132	0	1.2	1	0	
300	68	1	0	144	193	1	1	141	0	3.4	1	2	
301	57	1	0	130	131	0	1	115	1	1.2	1	1	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	

303 rows × 13 columns

```
y=df['output']
y
```

```
0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: output, Length: 303, dtype: int64
```

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)
```

```
model=RandomForestClassifier()
```

```
model.fit(xtrain,ytrain)
```

```
RandomForestClassifier()
```

```
ypred=model.predict(xtest)
```

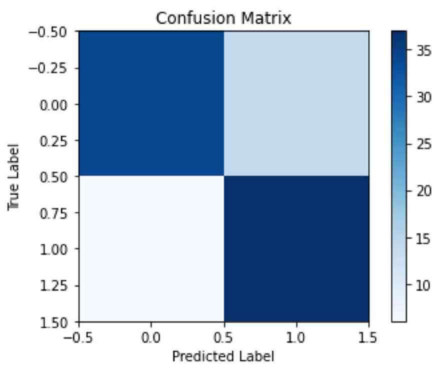
```
accuracy = accuracy_score(ypred,ytest)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.7802197802197802
```

```
# Evaluate the model performance using confusion matrix
cm = confusion_matrix(ytest, ypred)
print("Confusion Matrix:\n", cm)
```

```
Confusion Matrix:
 [[34 14]
 [ 6 37]]
```

```
# Plot the confusion matrix
plt.imshow(cm, cmap='Blues')
plt.colorbar()
plt.title("Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```



0s completed at 11:11 AM