Importing Packages

In [1]:
```
import pandas as pd
```

Data Loading

In [2]:
```
df = pd.read_csv("Dataset/cardio_data_clean.csv")
```

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 17 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   date           70000 non-null  object 
 1   country_ID     70000 non-null  int64  
 2   id             70000 non-null  int64  
 3   active         70000 non-null  int64  
 4   age            70000 non-null  int64  
 5   alco           70000 non-null  int64  
 6   ap_hi          70000 non-null  int64  
 7   ap_lo          70000 non-null  int64  
 8   cholesterol    70000 non-null  int64  
 9   gender         70000 non-null  int64  
 10  gluc           70000 non-null  int64  
 11  height         70000 non-null  int64  
 12  occupation_ID  70000 non-null  int64  
 13  smoke          70000 non-null  int64  
 14  weight         70000 non-null  int64  
 15  disease        70000 non-null  int64  
 16  BMI            70000 non-null  float64
dtypes: float64(1), int64(15), object(1)
memory usage: 9.1+ MB
```

In [4]:
```
df = df.drop(columns=['date', 'country_ID', 'id', 'occupation_ID', 'height', 'weigh
```

In [5]:
```
X = df.drop('disease', axis=1)
y = df['disease']
```

Feature Scaling

In [6]:
```
cols = ['age', 'ap_hi', 'ap_lo', 'BMI']
```

In [7]:
```
from sklearn.preprocessing import RobustScaler
st = RobustScaler()
X[cols] = st.fit_transform(X[cols])
```

In [8]:
```
X
```

Out[8]:

	active	age	alco	ap_hi	ap_lo	cholesterol	gender	gluc	smoke	BMI
0	1	-0.357630	0	-0.50	0.0	1	2	1	0	-0.694795
1	0	-0.608518	0	-1.00	-2.0	1	1	1	0	-0.530757
2	1	0.657931	0	0.50	0.0	3	1	1	0	1.791009
3	0	-0.230958	0	0.50	-1.0	3	1	1	0	-0.451893
4	1	0.786514	0	0.50	1.0	3	2	3	0	0.568612
...	...	...	...	...	...	...	...	...	...	...
69995	1	-0.126399	0	0.00	0.0	1	2	1	1	0.087539
69996	1	0.791155	0	1.00	1.0	2	1	2	0	3.800473
69997	0	-0.173901	1	3.00	1.0	3	2	1	0	0.784700
69998	0	0.744745	0	0.75	0.0	1	1	2	0	0.114353
69999	1	0.228501	0	0.00	0.0	2	1	1	0	-0.231073

70000 rows × 10 columns

Model Training using Random Forest Classifier

Random Forest Classifier

In [10]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
```

In [11]:
```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier()
```

In [12]:
```
clf.fit(X_train, y_train)
```

Out[12]:
```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [13]:
```
y_pred = clf.predict(X_test)
```

Feature Importance

In [14]:
```
pd.DataFrame(clf.feature_importances_,
             index = X_train.columns,
             columns=['Importance']).sort_values('Importance', ascending=False)
```

Out[14]:

	Importance
age	0.348925
BMI	0.290123
ap_hi	0.171346
ap_lo	0.092445
cholesterol	0.036363
gluc	0.016985
gender	0.015777
active	0.012216
smoke	0.008416
alco	0.007403

Based on the feature importance, we can say that age is the most significant factor with an importance value of 35%. Next is BMI with a value of 29%. Blood Pressure (ap_hi & ap_lo) have moderate importance values of 17% and 9% respectively. Cholesterol has lower importance value of 3%. The remaining factors such as glucose, gender, activity level, smoking, and alcohol consumption have importance values below 0.02, indicating they have a lesser impact on the model's predictions.

In [15]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

Out[15]:
```
0.7104
```

The model has a low accuracy score of 71%, we'll check with cross validation score at 10 folds.

In [16]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(clf, X_train, y_train, cv=10)
```

Out[16]:
```
array([0.716     , 0.70438095, 0.71142857, 0.70590476, 0.70495238,
       0.70571429, 0.70685714, 0.70038095, 0.70152381, 0.71409524])
```

Previous accuracy score lies within the mean value.

In [ ]: