Assignment 2

Name:-Prateek Kumar

Roll No:- 2100290100118(52)

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_sc
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
```

In [3]:
```
nhanes_data = pd.read_csv('/content/P_DEMO.csv')

demographics_cols = ['SEQN', 'Age', 'Gender']
cardio_cols = ['SEQN', 'HeartConditionVar1', 'HeartConditionVar2', ...]
merged_data = pd.merge(nhanes_data[demographics_cols], nhanes_data[cardio_cols],

age_ranges = [(10, 20), (21, 30), (31, 40), (41, 50), (51, 60), (61, 70), (70, 2

for age_range in age_ranges:
    filtered_data = merged_data[(merged_data['Age'] >= age_range[0]) & (merged_d
    print(f"Patients in age range {age_range}: {len(filtered_data)}")
```

In [ ]:
```
X = merged_data.drop(['SEQN', 'HeartConditionVar1', 'HeartConditionVar2', ...],
y = merged_data['HeartConditionVar1']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_


scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [ ]:
```
models = {
    'Logistic Regression': LogisticRegression(),
    'K-NN': KNeighborsClassifier(),
    'SVM': SVC(),
    'Decision Tree': DecisionTreeClassifier(),
    'ANN': MLPClassifier()
}
```

In [ ]:
```
for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)

    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    auc = roc_auc_score(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"Metrics for {name}:")
    print(f"Accuracy: {accuracy}")
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"F1 Score: {f1}")
    print(f"AUC: {auc}")
    print(f"R2 Score: {r2}")
    print(" ------------ ")
```

In [ ]:
```
male_count = merged_data[merged_data['Gender'] == 'Male'].shape[0]
female_count = merged_data[merged_data['Gender'] == 'Female'].shape[0]


print(f"Males in the dataset: {male_count}")
print(f"Females in the dataset: {female_count}")
```

In [ ]:
```
heart_condition_count = merged_data[merged_data['HeartConditionVar1'] == 1].shap
```

In [ ]:
```
males_with_heart_condition = merged_data[(merged_data['Gender'] == 'Male') & (me
females_with_heart_condition = merged_data[(merged_data['Gender'] == 'Female') &

print(f"People with heart conditions: {heart_condition_count}")
print(f"Males with heart conditions: {males_with_heart_condition}")
print(f"Females with heart conditions: {females_with_heart_condition}")
```