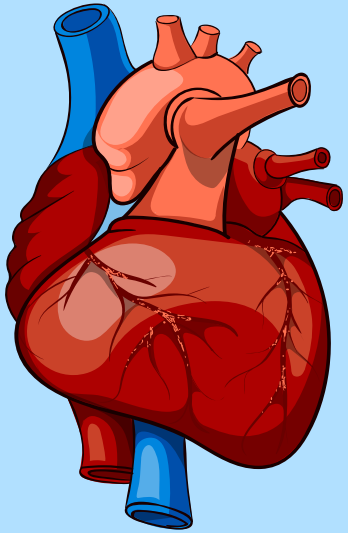
Lamrin Tech Skills University
Punjab, Ropar

Future Skills Program of
Pradhan Mantri Kaushal Vikas Yojana 4.0 (PMKVY)

Course Name: **Artificial Intelligence and Big Data Analytics**
Course Duration: - 4 Months



On-Job-Training Report

On

Heart Disease Prediction

Submitted By:

Student Name: **Vivek Sharma** Candidate ID:
Student Name: **Sahil Rana** Candidate ID:
Student Name: **Mansi** Candidate ID:

Future Skills 4.0
in collaboration with IBM, NSDC and Lamrin Tech Skills University Punjab.
(2023-24)

Lamrin Tech Skills University
Punjab, Ropar

HEART DISEASE PREDICTION

TABLE OF CONTENTS

01 Problem Statement

02 Solution

03 Implementation

04 Demo of the Project

05 Conclusion

INTRODUCTION

Heart disease is a leading cause of mortality worldwide, emphasizing the critical need for early detection and prevention strategies. Our project focuses on leveraging machine learning techniques to predict the likelihood of heart disease in individuals based on various health and lifestyle factors.

01

Problem Statement

To develop a machine learning model that accurately predicts the likelihood of heart disease in individuals based on their health and lifestyle factors, including age, sex, blood pressure, and cholesterol levels. This model will assist healthcare providers in identifying at-risk individuals early, enabling targeted interventions and personalized healthcare approaches to reduce the incidence and impact of heart disease.

02 Solution

Objective: Develop a machine learning model to predict heart disease risk based on individual health data.

Approach:

- **Data Collection**: Gather a dataset with key features such as age, sex, blood pressure, and cholesterol levels.
- **Data Preprocessing**: Clean and prepare the data for analysis.
- **Model Development**: Build a machine learning model to predict heart disease risk.
- **Model Evaluation**: Assess the model's performance using standard metrics like accuracy and recall.

Outcome:

- A predictive model that can accurately identify individuals at risk of heart disease.
- Potential for early intervention and personalized healthcare.

03 Implimentation

Reading the data file in df

In [3]:
```
df = pd.read_csv("HD_Cleveland_Data_Clean.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slop	ca	thal	target
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	N
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	Y
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	Y
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	N
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	N

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 297 entries, 0 to 296
Data columns (total 14 columns):
```

Random Forests

Let's try Random Forests model on the data and compare our results with the decision tree model. Random Forests is under ensemble class in the sklearn.

In [22]:
```
from sklearn.ensemble import RandomForestClassifier
```

If you remember theory lecture, we need to pass the number of trees in the forest which is n_estimators. The default is 10.
Let's pass 100 at the moment and fit the model to the training dataset.
☞* You can play with n_estimators by changing different numbers!*

In [23]:
```
# Creating instance and fitting the model
rfc = RandomForestClassifier(n_estimators=100)
rfc.fit(X_train, y_train)
```

Out[23]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

Decision Trees

We'll start with training a single decision tree!

In [16]:
```
# importing decision tree classifier
from sklearn.tree import DecisionTreeClassifier
```

In [17]:
```
#Creating instance "dtree" of the classifier
dtree = DecisionTreeClassifier()
```

In [18]:
```
#fitting to the training data, the default parameters are fine at the moment!
dtree.fit(X_train,y_train)
```

Out[18]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

Implimentation

In [24]:
```
# doing predictions
rfc_pred = rfc.predict(X_test)
```

In [25]:
```
# Evaluation
print(classification_report(y_test,rfc_pred))
print(confusion_matrix(y_test,rfc_pred))
```

```
              precision    recall  f1-score   support

           N       0.83      0.90      0.86        49
           Y       0.86      0.78      0.82        41

   micro avg       0.84      0.84      0.84        90
   macro avg       0.85      0.84      0.84        90
weighted avg       0.85      0.84      0.84        90

[[44  5]
 [ 9 32]]
```

It looks like the random forest is gave improved results over a single tree for the dataset we have used. We got the better precision, recall and f1-score using Random Forest and less number of mislabeled samples!

You will see, if the dataset gets larger and larger, the Random Forests will always do better than a single decision tree. In the current situation, the data set is not very large but still Random Forests model works better than decision trees, the model will outshines with larger data sets.

Prediction and Evaluation

Evaluation is important to see how did the model work!

In [19]:
```
# doing predictions
predictions = dtree.predict(X_test)
```

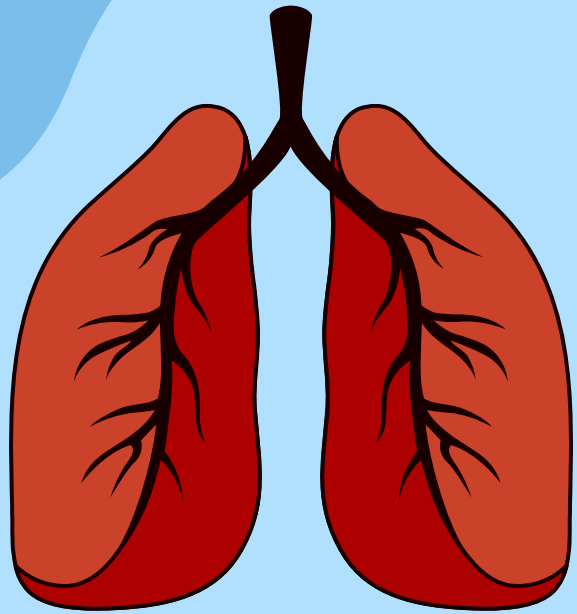
In [20]:
```
from sklearn.metrics import classification_report,confusion_matrix
```

In [21]:
```
print(classification_report(y_test,predictions))
print(confusion_matrix(y_test,predictions))
```

```
              precision    recall  f1-score   support

           N       0.77      0.76      0.76        49
           Y       0.71      0.73      0.72        41

   micro avg       0.74      0.74      0.74        90
   macro avg       0.74      0.74      0.74        90
weighted avg       0.75      0.74      0.74        90

[[37 12]
 [11 30]]
```

04 Conclusion

In conclusion, our project successfully developed a machine learning model to predict heart disease risk based on individual health data. By analyzing key factors such as age, sex, blood pressure, and cholesterol levels, our model can accurately identify individuals at risk, enabling early intervention and personalized healthcare. This project highlights the potential of machine learning in improving preventive medicine and enhancing heart disease management.



THANKS!