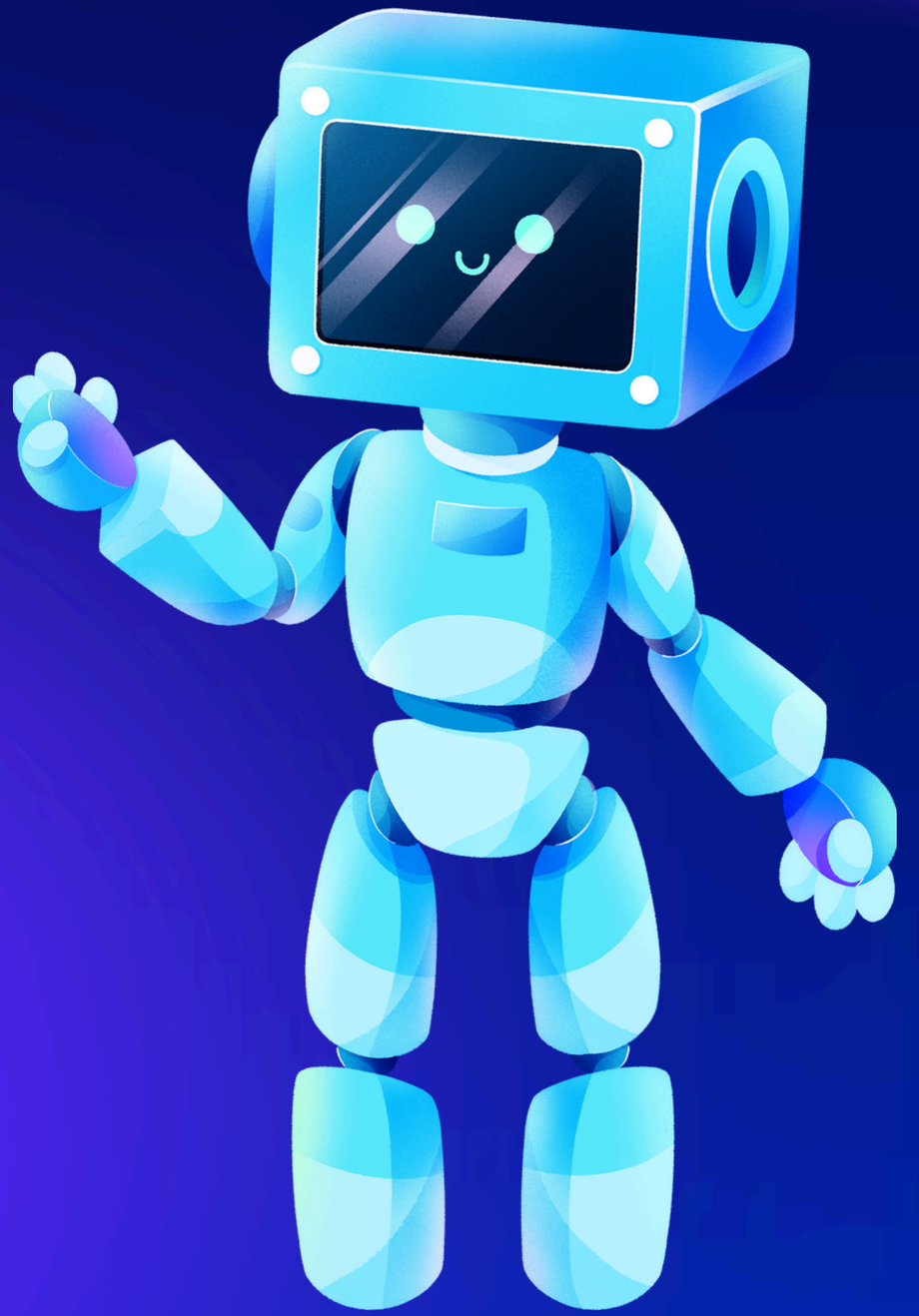
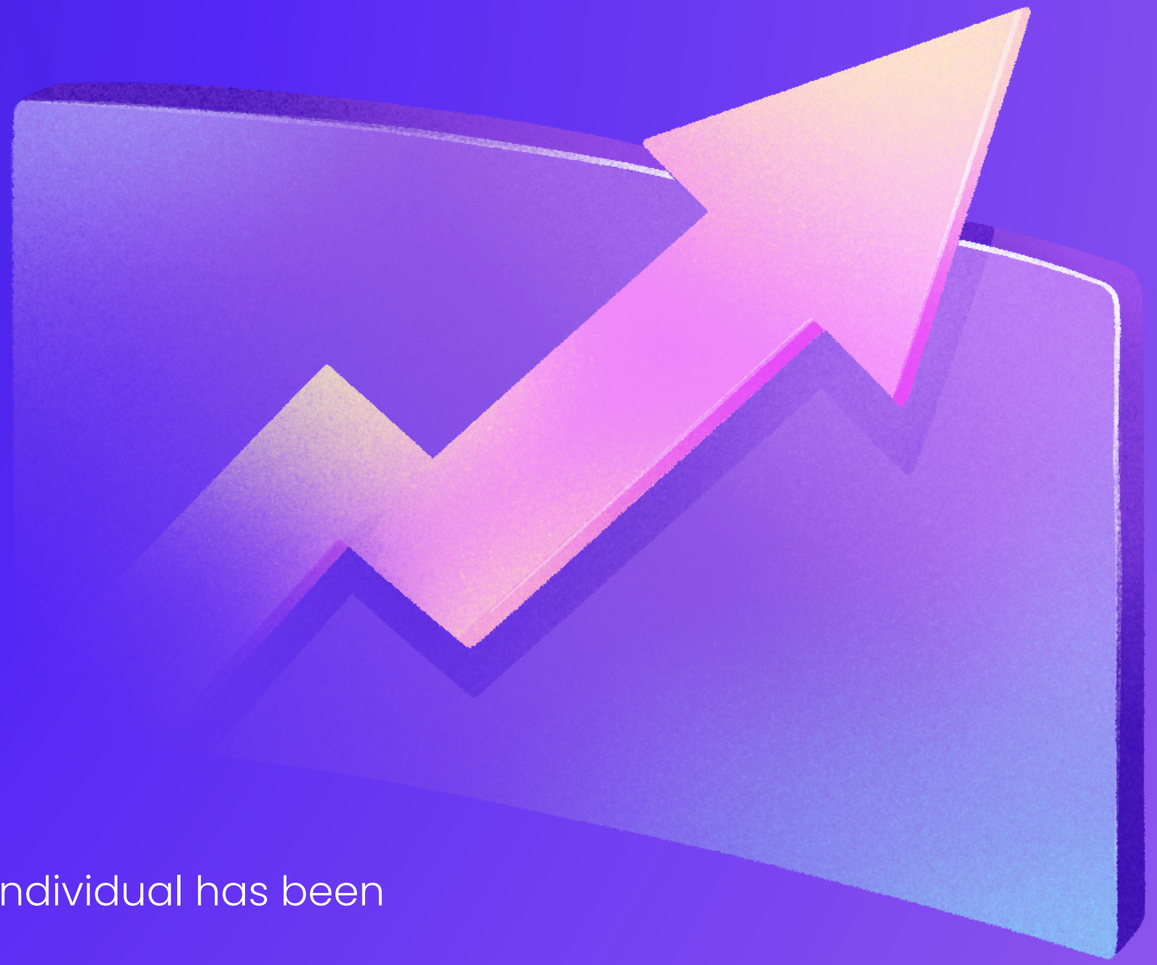
DISEASE PREDICTION USING MACHINE LEARNING

A COMPREHENSIVE APPROACH TO PREDICTIVE HEALTHCARE

By Mahmoud Osama



INTRODUCTION



Healthcare professionals often rely on various diagnostic tests and biomarkers to assess an individual's health status and diagnose diseases. In this scenario, we have access to a dataset containing multiple health-related attributes such as cholesterol levels, blood cell counts, hormone levels, and other physiological measurements.

The dataset also includes information on whether the individual has been diagnosed with a specific disease or not.

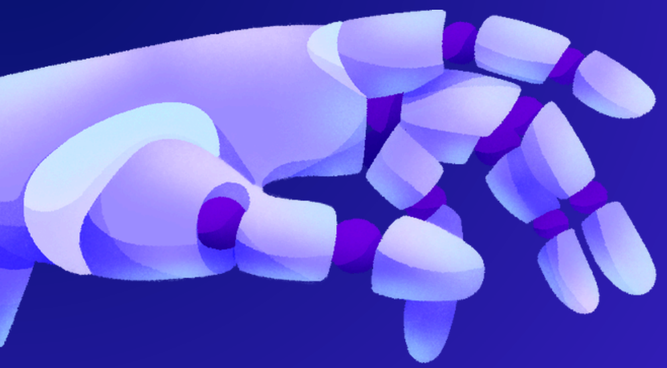
PROJECT OBJECTIVES

OBJECTIVE 01

Develop a predictive model to classify individuals into diseased or non-diseased categories.

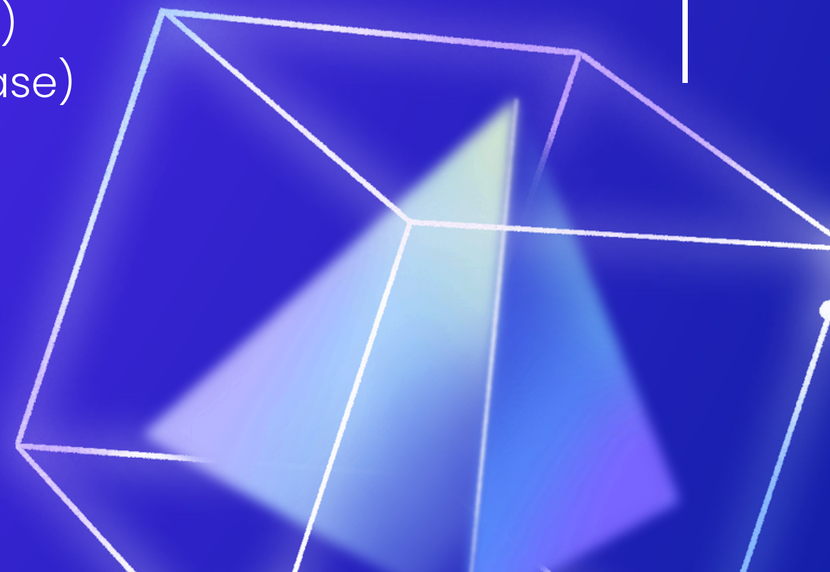
OBJECTIVE 02

Utilize machine learning algorithms to assist healthcare providers in disease diagnosis and prognosis.



DATASET ATTRIBUTES

- Cholesterol (mg/dL)
- Hemoglobin
- Platelets
- White Blood Cells (WBC)
- Red Blood Cells (RBC)
- Hematocrit
- Mean Corpuscular Volume (MCV)
- Mean Corpuscular Hemoglobin (MCH)
- Mean Corpuscular Hemoglobin Concentration (MCHC)
- Insulin
- BMI (Body Mass Index)
- Systolic Blood Pressure (SBP)
- Diastolic Blood Pressure (DBP)
- Triglycerides (mg/dL)
- HbA1c (Glycated Hemoglobin)
- LDL Cholesterol
- HDL Cholesterol
- ALT (Alanine Aminotransferase)
- AST (Aspartate Aminotransferase)
- Heart Rate (bpm)
- Creatinine
- Troponin
- C-reactive Protein (CRP)
- Disease (Binary Indicator)



MACHINE LEARNING MODELS USED

XGBoost

XGBoost

```
In [17]: xgbModel = xgb.XGBClassifier(objective='multi:softmax', num_class=5, random_state=42)

In [18]: xgbModel.fit(X_train, Y_train)

Out[18]: XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None, num_class=5,
```

Random Forest

Random Forest

```
In [21]: RFModel =  RandomForestClassifier()

In [22]: RFModel.fit(X_train, Y_train)

Out[22]: RandomForestClassifier
RandomForestClassifier()
```

RESULTS AND ACHIEVEMENTS

01

- XGBoost

```
In [19]: xgbModel.score(X_train, Y_train)
Out[19]: 1.0

In [20]: xgbModel.score(X_test, Y_test)
Out[20]: 0.8333333333333334
```

02

- Random Forest

```
In [23]: RFModel.score(X_train, Y_train)
Out[23]: 1.0

In [24]: RFModel.score(X_test, Y_test)
Out[24]: 0.9197530864197531
```

THANK YOU!