Heart Failure Prediction Using Machine Learning

Smit Rana [792056], Harsh Patel [791820],
Rechel Rebello [787548], Dhruv Limbachiya [784687]

DAB304-Healthcare Analytics Fall 2022 – 002 [Group09]
St. Clair College of Applied Arts and Technology, Windsor, ON N9G 3C3

Introduction

Every year, approximately 17 million people are killed by cardiovascular diseases, which primarily demonstrate as cardiac and heart failures. It is a condition in which the heart is unable to pump enough blood to meet the body's requirements. Heart disease symptoms can include physical weakness, breathing problems, swelling feet, etc. The techniques are critical for identifying complicated heart diseases that pose a high risk to human life.

Healthcare industries generate massive amounts of data, known as big data, which contains hidden knowledge or patterns for decision making. Data analysis is crucial in healthcare, including new research findings, emergency situations, and epidemics. Analytics in healthcare improves care by enabling preventive care, and Exploratory Data Analysis (EDA) is a key step in data analysis. Moreover, A Machine Learning (ML) model can be very helpful in the early detection and care of people with cardiovascular disease or at high cardiovascular risk (due to the presence of one or more risk factors including like hypertension, diabetes, or pre-existing disease).

To predict heart disease, we will use multiple Machine learning models (Random Forest, Decision Tree, etc.) in conjunction with data analytics and visualisation tools. Therefore, there is a perfect fit between this project and this course.

Motivation

The delivery of superior services at low cost is a major challenge for healthcare groups (hospitals and medical centres). Quality service entails correctly diagnosing patients and administering effective treatments. Clinical tests must be kept as inexpensive as possible in hospitals. They can accomplish these goals by utilizing proper user information and/or decision support systems.

Today, most hospitals use hospital information systems to manage their healthcare or patient data. Patients' electronic medical records quantify symptoms, body features, and clinical laboratory test values, which can be used to perform biostatistics analysis to highlight patterns and correlations that medical doctors would otherwise overlook. Alas, these data are rarely used to help clinicians make clinical decisions. These data contain a wealth of hidden information that is largely untapped. This begs the question, "How can we transform data into insights that will allow healthcare practitioners to make clinical decisions?"

Related Work

There have been few studies that use Machine Learning to predict heart failure. These heart prediction studies are carried out using EDAs, ANN (Artificial Neural Network), R, Cross Validation K Fold, GridSearchCV, PyTorch, and other software. As Forthcoming data analyst, we used various machine learning models to predict heart failure based on the model's *training and validation accuracy, Classification Report, Confusion Matrix, Precision Recall Curve, and ROC (Receiver Operating Characteristic) Curve*. We used the *scikit-learn* library for those models, and *ploty* for visualising feature relationships. In addition, we performed feature engineering by plotting the confusion matrix. Then we did feature selection because it reduces over-fitting (fewer redundant data means fewer chances of making decisions based on noise), shortens training time (fewer data means the algorithms train faster), and improves precision (Less ambiguous data means improvement of modelling accuracy). Finally, we take a proactive move by putting our best model online so that failure predictions can be seen.

Methods

At first, we gather the data from UCI Machine Learning Repository. The dataset holds 299 records with 13 attributes,

– *age*
– *anaemia* - *Decrease of red blood cells or hemoglobin (boolean)*
– *creatinine_phosphokinase* - *Level of the CPK enzyme in the blood (mcg/L)*
– *diabetes* - *If the patient has diabetes (boolean)* *ejection_fraction* - *Percentage of blood leaving the heart at each contraction (percentage)*

- *high_blood_pressure* - *If the patient has hypertension (boolean)*
- *platelets* - *Platelets in the blood (kiloplatelets/mL)*
- *serum_creatinine* - *Level of serum creatinine in the blood (mg/dL)*
- *serum_sodium* - *Level of serum sodium in the blood (mEq/L)*
- *sex* - *Woman or man (binary)*
- *smoking* - *If the patient smokes or not (boolean)*
- *time* - *Follow-up period (days)*
- *death_event* - *If the patient deceased during the follow-up period (boolean)*

Then we start to find missing data and datatypes. We found that all attributes have 299 non-null values, so there are no missing values and datatype is also 'float64' or 'int64,' which works well when fed into Machine Learning algorithm.

Then we move ahead with EDAs. Exploratory data analysis is a technique used to examine the data and utilise it to forecast a result. EDA helps with a better understanding of the variables in the data collection and their relationships and is usually used to investigate what data might disclose beyond the formal modelling or hypothesis testing assignment.
We used *plotly* library to do this.

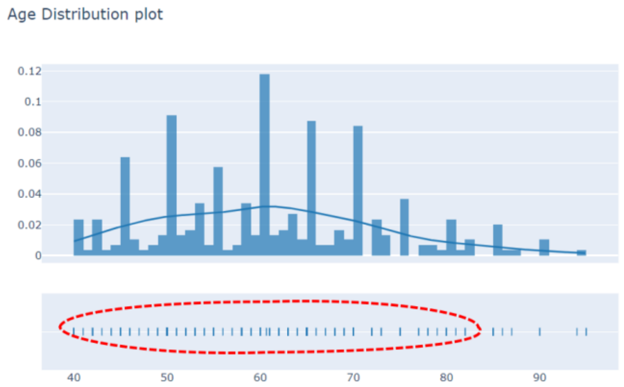


Fig 1. Age groups under 40 and over 80 are hardly represented, while those between 40 and 80 are widely dispersed.

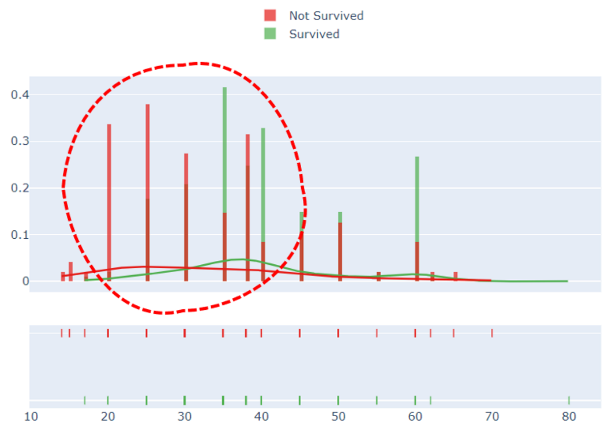


Fig 2. ejaction fraction values between 20 and 40 result in a record number of deaths.

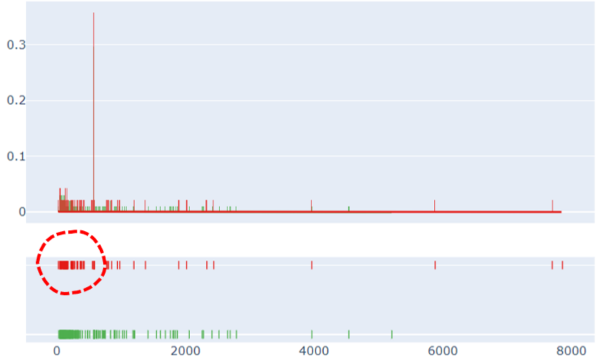


Fig 3. for creatinine phosphokinase values ranging from 0 to 500, death cases are

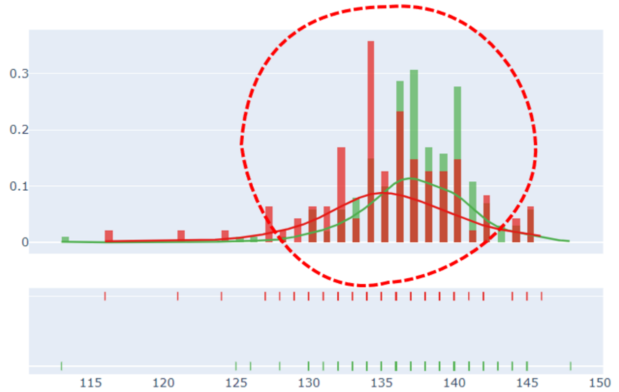


Fig 4. serum_sodium values 127 - 145 indicate towards a death due to heart failure.

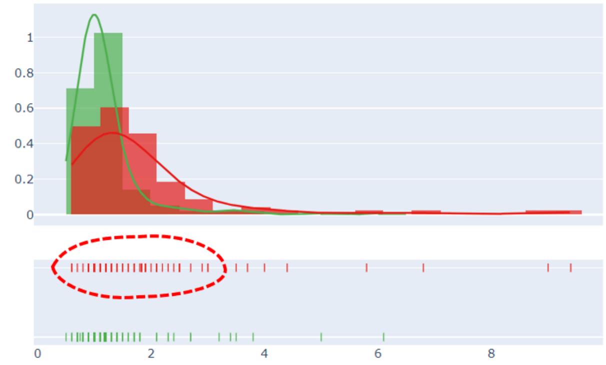


Fig 5. serum_creatinine values from 0.6 to 3.0 have higher probability to lead to death

We also did analysis diabetes, anemia, high_blood_pressure, smoking. We find that Anemia increases the likelihood of heart failure. Diabetes increases the likelihood of heart failure. High blood pressure is associated with an increased risk of heart failure. Males outnumber females in terms of heart failure risk, but only by a small margin. Smoking increases the likelihood of suffering from heart failure.

Our Next Step was Feature Selection. It is a technique for reducing the input variable to your model by using only relevant data and removing noise from the data. It is the process of selecting relevant features for your machine learning model automatically based on the type of problem you are attempting to solve. We did confusion matrix and feature importance plot by ExtraTreesClassifier.

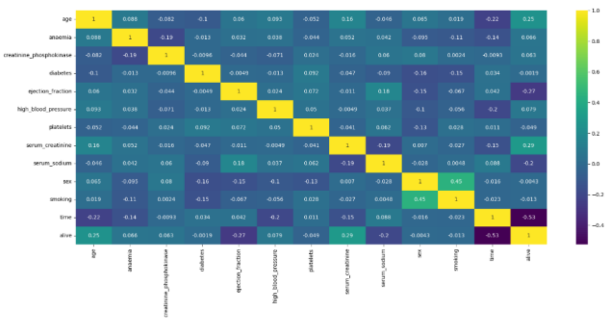


Fig 6. We will not use the feature time because it has no precise meaning, even if the correlation map shows that the time feature and the target are correlated.

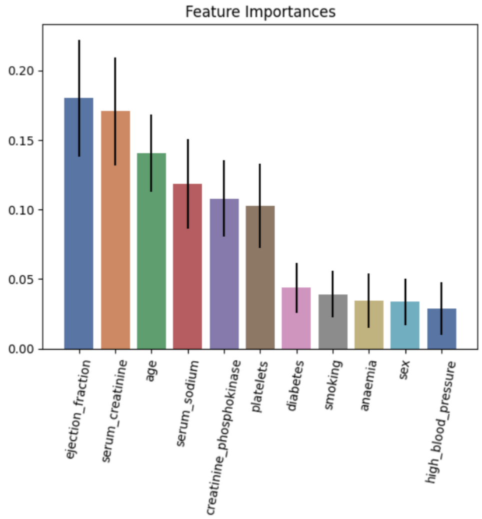
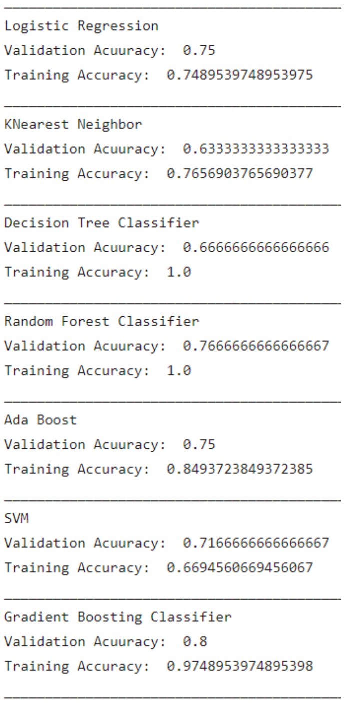


Fig 7 Feature Importances

Result

We reached almost at end with Data Modeling. First, we compared Classifier based on Validation and Training Accuracy. For that we split our data to 2:8 for test and validation set. We used seven different popular machine learning model('Logistic Regression', 'KNearest Neighbor', 'Decision Tree Classifier', 'Random Forest Classifier', 'Ada Boost', 'SVM', 'Gradient Boosting Classifier') for test accuracy.



```
_________________________________________________
Logistic Regression
Validation Acuuracy:  0.75
Training Accuracy:  0.7489539748953975
_________________________________________________
KNearest Neighbor
Validation Acuuracy:  0.6333333333333333
Training Accuracy:  0.7656903765690377
_________________________________________________
Decision Tree Classifier
Validation Acuuracy:  0.6666666666666666
Training Accuracy:  1.0
_________________________________________________
Random Forest Classifier
Validation Acuuracy:  0.7666666666666667
Training Accuracy:  1.0
_________________________________________________
Ada Boost
Validation Acuuracy:  0.75
Training Accuracy:  0.8493723849372385
_________________________________________________
SVM
Validation Acuuracy:  0.7166666666666667
Training Accuracy:  0.6694560669456067
_________________________________________________
Gradient Boosting Classifier
Validation Acuuracy:  0.8
Training Accuracy:  0.9748953974895398
_________________________________________________
```

Fig 8, Training and Validation Accuracy

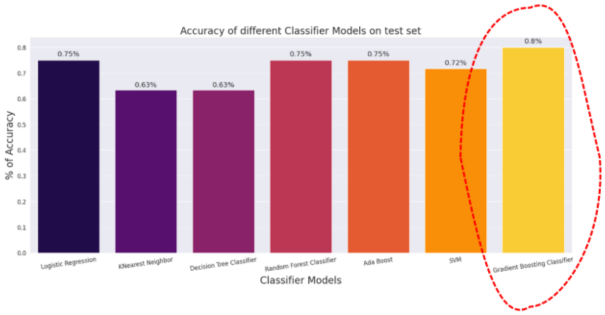
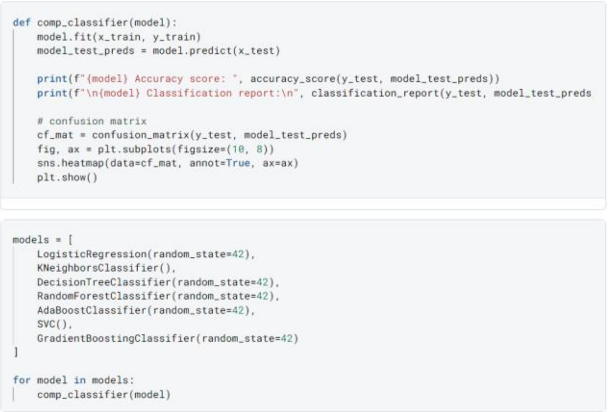


Fig 9, after comparing these seven models we found that gradient boosting algorithm is best because it gave test accuracy 80%.

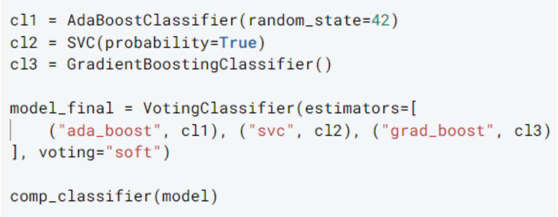
Now, Comparing Classifiers using their Classification Report, Confusion Matrix, Precision Recall Curve, and ROC (Receiver Operating Characteristic) Curve.



```
def comp_classifier(model):
    model.fit(x_train, y_train)
    model_test_preds = model.predict(x_test)

    print(f"{model} Accuracy score: ", accuracy_score(y_test, model_test_preds))
    print(f"\n{model} Classification report:\n", classification_report(y_test, model_test_preds

    # confusion matrix
    cf_mat = confusion_matrix(y_test, model_test_preds)
    fig, ax = plt.subplots(figsize=(10, 8))
    sns.heatmap(data=cf_mat, annot=True, ax=ax)
    plt.show()
```

```
models = [
    LogisticRegression(random_state=42),
    KNeighborsClassifier(),
    DecisionTreeClassifier(random_state=42),
    RandomForestClassifier(random_state=42),
    AdaBoostClassifier(random_state=42),
    SVC(),
    GradientBoostingClassifier(random_state=42)
]

for model in models:
    comp_classifier(model)
```

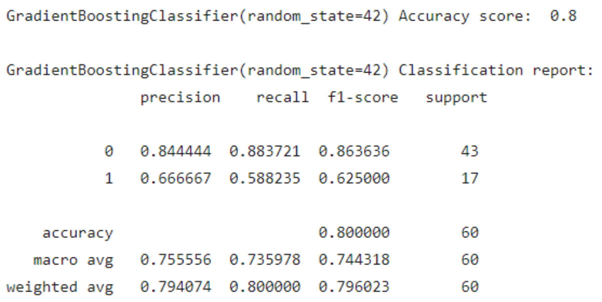
Fig 10, Finding above parameters for models



```
cl1 = AdaBoostClassifier(random_state=42)
cl2 = SVC(probability=True)
cl3 = GradientBoostingClassifier()

model_final = VotingClassifier(estimators=[
    ("ada_boost", cl1), ("svc", cl2), ("grad_boost", cl3)
], voting="soft")

comp_classifier(model)
```

Fig 11, Voting Classifier

Among seven machine learning model we pick Ada Boost, SVC and Gradient Boosting algorithm based on Precision call, ROC and accuracy score. We fed them in Voting Classifier. It is a machine learning model that gains experience by training on a collection of several models and forecasts an output (class) based on the class with the highest likelihood of being the output.



```
GradientBoostingClassifier(random_state=42) Accuracy score:  0.8

GradientBoostingClassifier(random_state=42) Classification report:
               precision    recall  f1-score   support

           0    0.844444  0.883721  0.863636        43
           1    0.666667  0.588235  0.625000        17

    accuracy                        0.800000        60
   macro avg    0.755556  0.735978  0.744318        60
weighted avg    0.794074  0.800000  0.796023        60
```

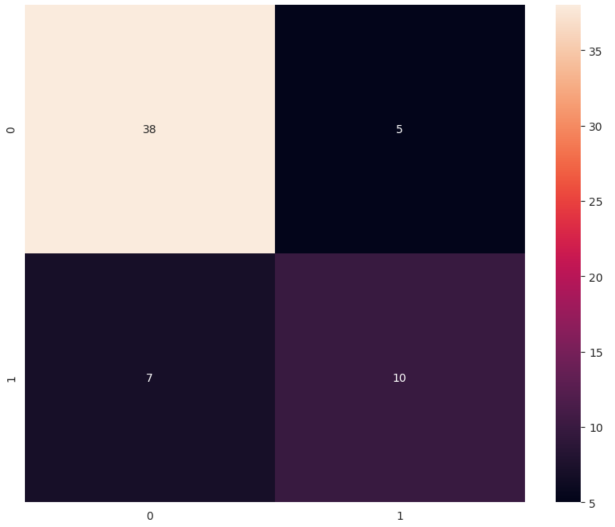
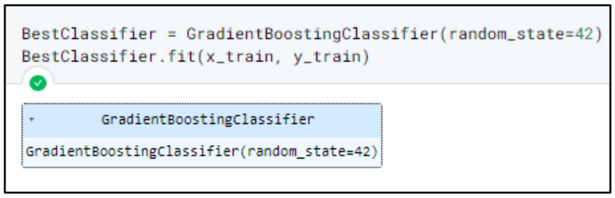


Fig 12. Gradient Boosting Classifier Confusion Matrix with Accuracy Score, Precision, Recall and F1 – Score.

On the test set, GradientBoostingClassifier(random_state=42) performs better than the other classifiers which has Average Precision 84% and Average Recall 88%



```
BestClassifier = GradientBoostingClassifier(random_state=42)
BestClassifier.fit(x_train, y_train)
```

```
GradientBoostingClassifier
GradientBoostingClassifier(random_state=42)
```

Fig 13, We fit train data in Gradient Boosting Classifier for Precision Recall Curve and ROC Curve.

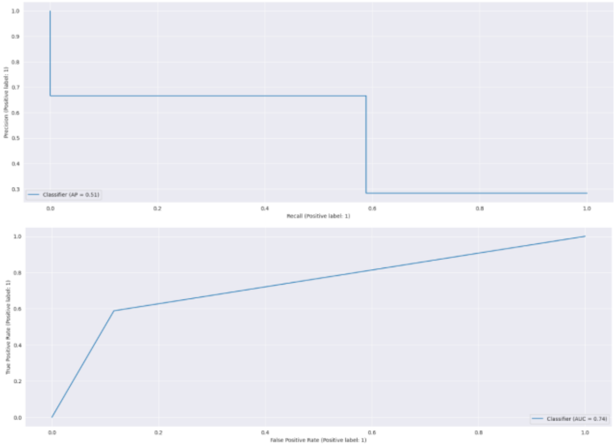


Fig 14, Precision Recall Curve and ROC Curve for the top classifier we have so far discovered (with Area Under Curve = 0.74)

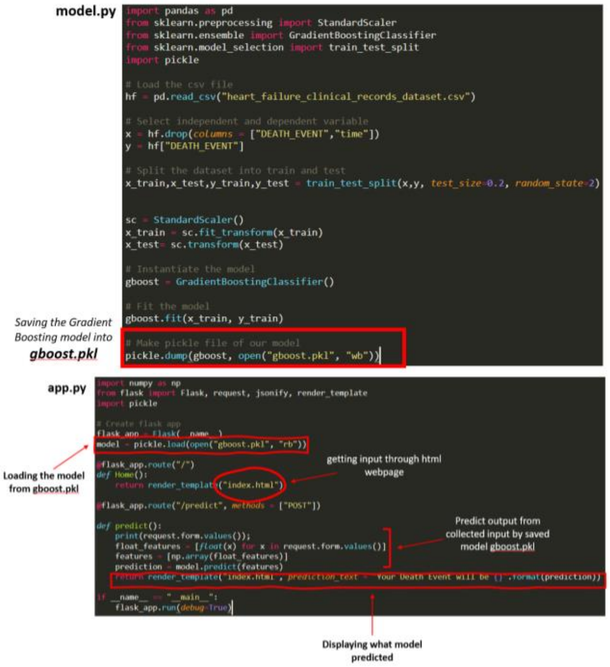
Discussion

After conducting extensive research, we concluded that heart failures are caused by general ageing. Creatinine phosphokinase greater than 120 mcg/L and normal ejection fraction (55% - 70%). When the percentage falls below 55%, the risk of heart failure increases. Heart failure was caused by both low and extremely high platelet counts. Most heart failures are caused by serum creatinine levels between 0.8 and 1.7 (mg/dL). Serum sodium levels above 130 (mEq/L) increase the risk of heart failure. Anemia and diabetes both increase the risk of heart failure. High blood pressure is associated with an increased risk of heart failure. In terms of heart failure risk, males outnumber females by a small margin. Smoking increases the likelihood of developing heart failure.

We set some objectives that project will be noted "successful" if it meets the following objectives:

- Explore all the features in this dataset and try to find relationships between them. Then look for those who are more closely related to target variable (alive). Based on our data, what are the death indicators for heart attacks?
- Achieving a prediction accuracy of between 80 to 90 for model employed in the project to win the public's trust and confidence.
- An ML Model will be deployed to a webpage and accessible to the public to predict the probability of heart failure by entering the above attributes.

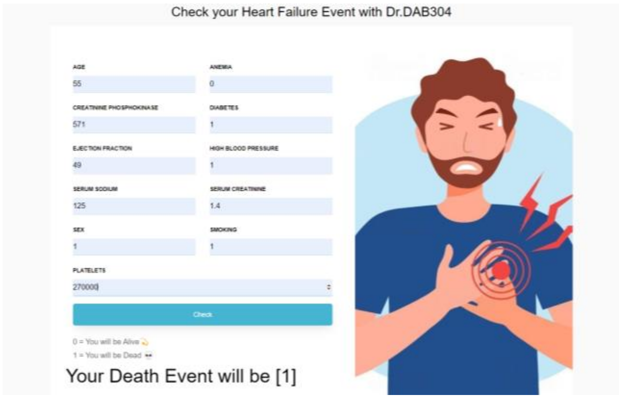
We deploy our mode to webpage for project implementation.


model.py
Saving the Gradient Boosting model into gboost.pkl
app.py
Loading the model from gboost.pkl
getting input through html webpage
Predict output from collected input by saved model gboost.pkl
Displaying what model predicted

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split
import pickle

# Load the csv file
hf = pd.read_csv("heart_failure_clinical_records_dataset.csv")

# Select independent and dependent variable
x = hf.drop(columns = ["DEATH_EVENT","time"])
y = hf["DEATH_EVENT"]

# Split the dataset into train and test
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size=0.2, random_state=2)


sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test= sc.transform(x_test)

# Instantiate the model
gboost = GradientBoostingClassifier()

# Fit the model
gboost.fit(x_train, y_train)

# Make pickle file of our model
pickle.dump(gboost, open("gboost.pkl", "wb"))
```

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

# Create flask app
flask_app = Flask(__name__)
model = pickle.load(open("gboost.pkl", "rb"))

@flask_app.route("/")
def Home():
    return render_template("index.html")

@flask_app.route("/predict", methods = ["POST"])

def predict():
    print(request.form.values());
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)
    return render_template("index.html", prediction_text = "Your Death Event will be {}".format(prediction))

if __name__ == "__main__":
    flask_app.run(debug=True)
```

*Fig 15, **model.py** is for to create model and saving that model in pickle file (gboost.pkl) and **app.py** to control the model inputs and prediction with html page*



Check your Heart Failure Event with Dr.DAB304

AGE: 55
ANEMIA: 0
CREATININE PHOSPHOKINASE: 571
DIABETES: 1
EJECTION FRACTION: 49
HIGH BLOOD PRESSURE: 1
SERUM SODIUM: 125
SERUM CREATININE: 1.4
SEX: 1
SMOKING: 1
PLATELETS: 270000

0 = You will be Alive
1 = You will be Dead

Your Death Event will be [1]

Fig 16, Model Deployment Implementation

Due to the fact that the data were all integers, we encountered one difficulty with our analysis. Because of this, we are unable to investigate characteristics by visualising them in tableau. Time column was another. With the target variable and its prediction, it has no clear meaning. Additionally, the data set missing with other number of heart failure indications. There were only 13 features available. But ultimately, we rely on our own abilities.

Conclusion

At end, the primary predictors of heart failure fatalities were identified. We're looking for heart failure data that differs from the existing data. In the future, we may investigate more correlations, irrelevant data, and unnecessary indicators in order to find a peaceful solution for the healthcare sector.

Contribution

Each Member gave this project their very best effort. First, we all worked hard to get the data. Following data collecting, Smit and Dhurv processed the data and produced a statistical summary. Rechel and Harsh created the EDAs and the visuals. Smit and Harsh carried out the feature selection, machine learning model implementation, and accuracy score analysis. Smit finally completed the deployment of the Model.

References

[1]https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records (data)

[2]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4216425/

[3]https://machinelearningmastery.com/voting-ensembles-with-python/

[4]https://towardsdatascience.com/3-ways-to-deploy-machine-learning-models-in-production-cdba15b00e

Appendices

https://github.com/gentallman/DAB304.git
(entire project code)