Heart Failure Prediction

The Developer Academy

Objective

Create a model for predicting mortality caused by Heart Failure.

Creating a neural network (ANN) and comparing it to baseline models

(SVM, kNN, LogisticRegression, Naive-Baines)

Performance metrics: accuracy, precision, recall, F1-score, ROC curve, AUC

Data

#	Column	Non-Null Count	Dtype
0	age	299 non-null	float64
1	anaemia	299 non-null	int64
2	creatinine_phosphokinase	299 non-null	int64
3	diabetes	299 non-null	int64
4	ejection_fraction	299 non-null	int64
5	high_blood_pressure	299 non-null	int64
6	platelets	299 non-null	float64
7	serum_creatinine	299 non-null	float64
8	serum_sodium	299 non-null	int64
9	sex	299 non-null	int64
10	smoking	299 non-null	int64
11	time	299 non-null	int64
12	DEATH_EVENT	299 non-null	int64

DEATH_EVENT

```
count    299.00000
mean       0.32107
min        0.00000
max        1.00000
Name: DEATH_EVENT, dtype: float64
```

We store this mean and use it as a threshold later.

Pre-Processing

After doing the ANN, we have come back and scaled the data.

Produces better results.

```
# scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Artificial Neural Network

Modelling the ANN

```
# initialise
ann = Sequential()

# input layer
ann.add(Dense(units=16, kernel_initializer="uniform", activation="relu", input_dim=12))

# hidden layers w/ regularisation
ann.add(Dense(units=8, kernel_initializer="uniform", activation="relu"))
ann.add(Dropout(0.25))
ann.add(Dense(units=8, kernel_initializer="uniform", activation="relu"))
ann.add(Dropout(0.5))

# output layer
ann.add(Dense(units=1, kernel_initializer="uniform", activation="sigmoid")) # units = 1 because this is binary classification (0 or 1)

# compile the network
ann.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])

# save model
ann.save("ann_model")
```

Performance

SCALED: val_accuracy = 77.75% (0.8167 - 0.7)

UNSCALED: val_accuracy = 70%

	precision	recall	f1-score	support
0	0.92	0.82	0.87	40
1	0.71	0.85	0.77	20
accuracy			0.83	60
macro avg	0.81	0.84	0.82	60
weighted avg	0.85	0.83	0.84	60

ANN Confusion Matrix

The artificial neural network (ANN) predicted 36 survivals and 24 deaths.

There were 10 incorrect classifications (3 false survivals, 7 false-deaths)

Baseline

Logistic Regression

41 survivals and 19 deaths.

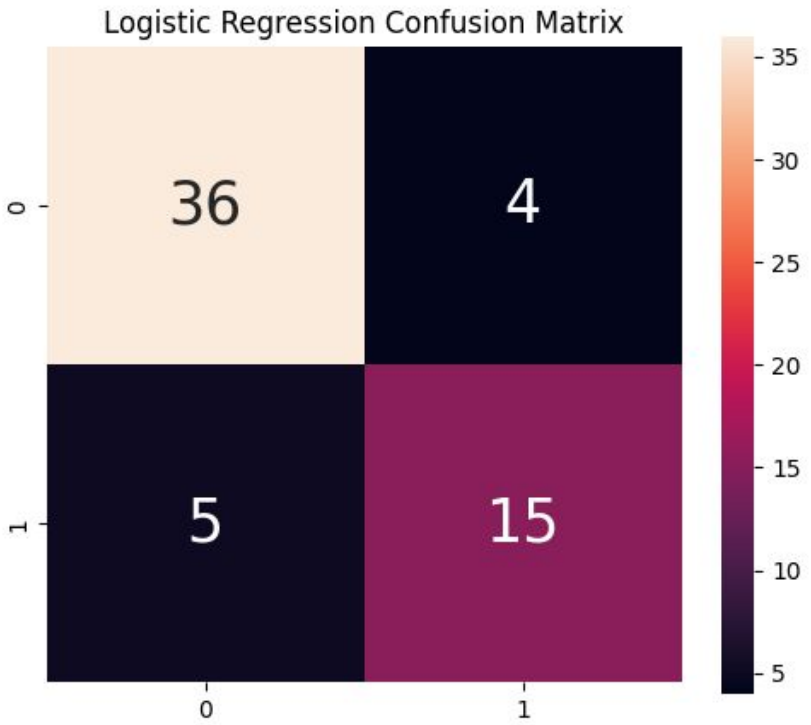
There were 9 incorrect classifications
(5 false survivals, 4 false-deaths)

Comparison to ANN:

-1 false classification

-3 false deaths

+2 false survival



k-Nearest Neighbours

60 survivals and 10 deaths.

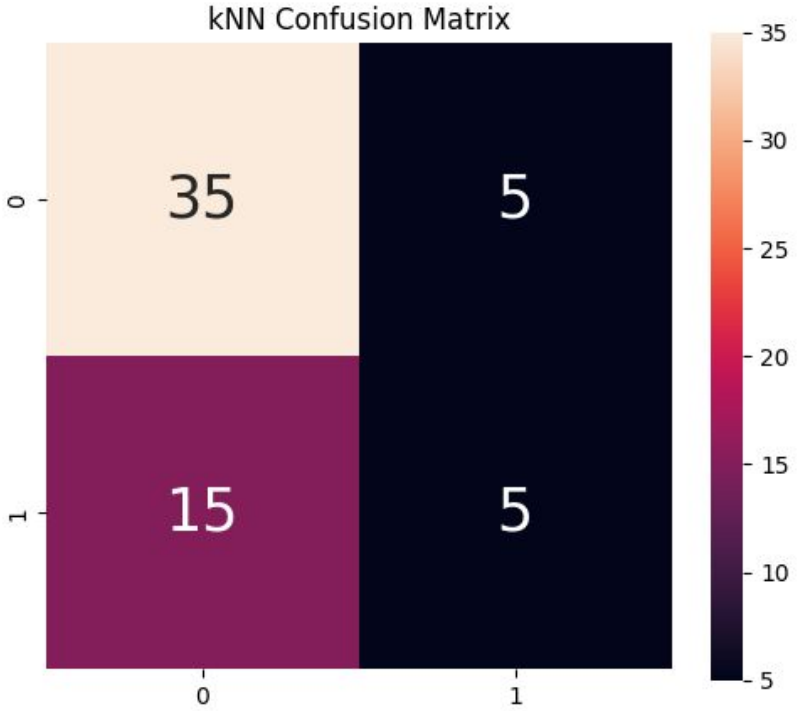
There were 20 incorrect classifications (15 false survivals, 5 false-deaths)

Comparison to ANN:

+10 false classification

-2 false deaths

+12 false survival



Naive-Bayes

45 survivals and 15 deaths.

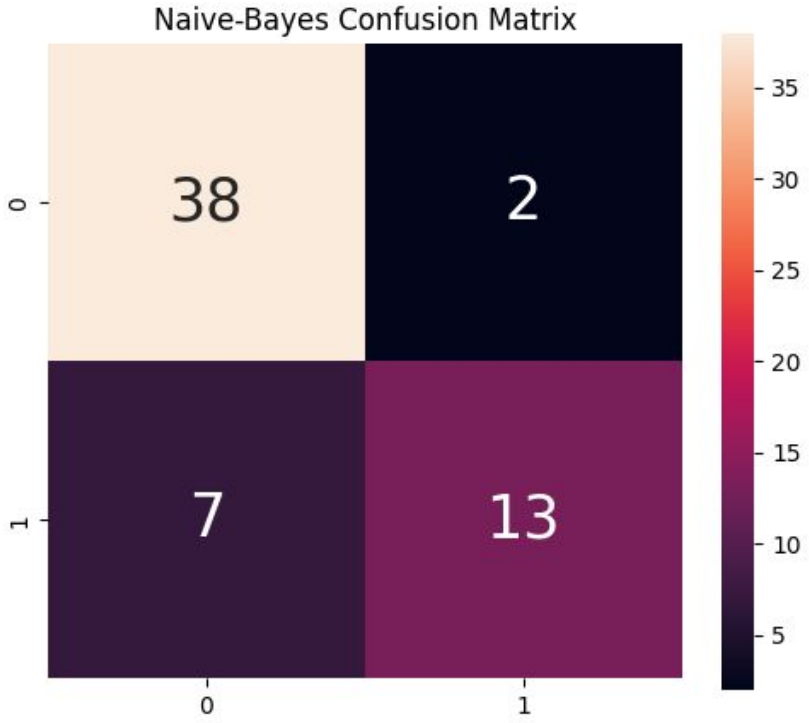
There were 9 incorrect classifications
(7 false survivals, 2 false-deaths)

Comparison to ANN:

-1 false classification

-5 false deaths

+4 false survival



SVM

46 survivals and 14 deaths.

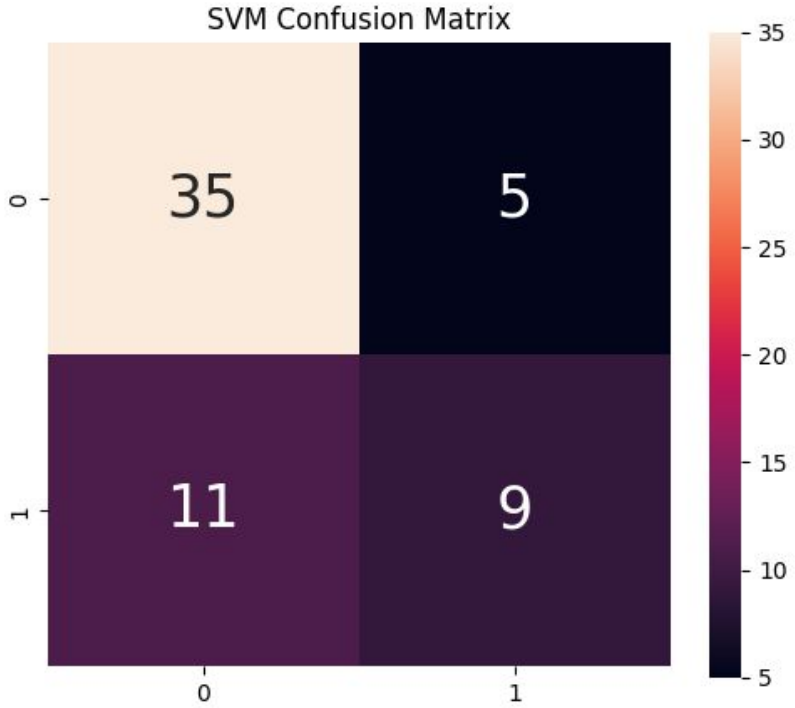
There were 16 incorrect classifications (11 false survivals, 5 false-deaths)

Comparison to ANN:

+6 false classification

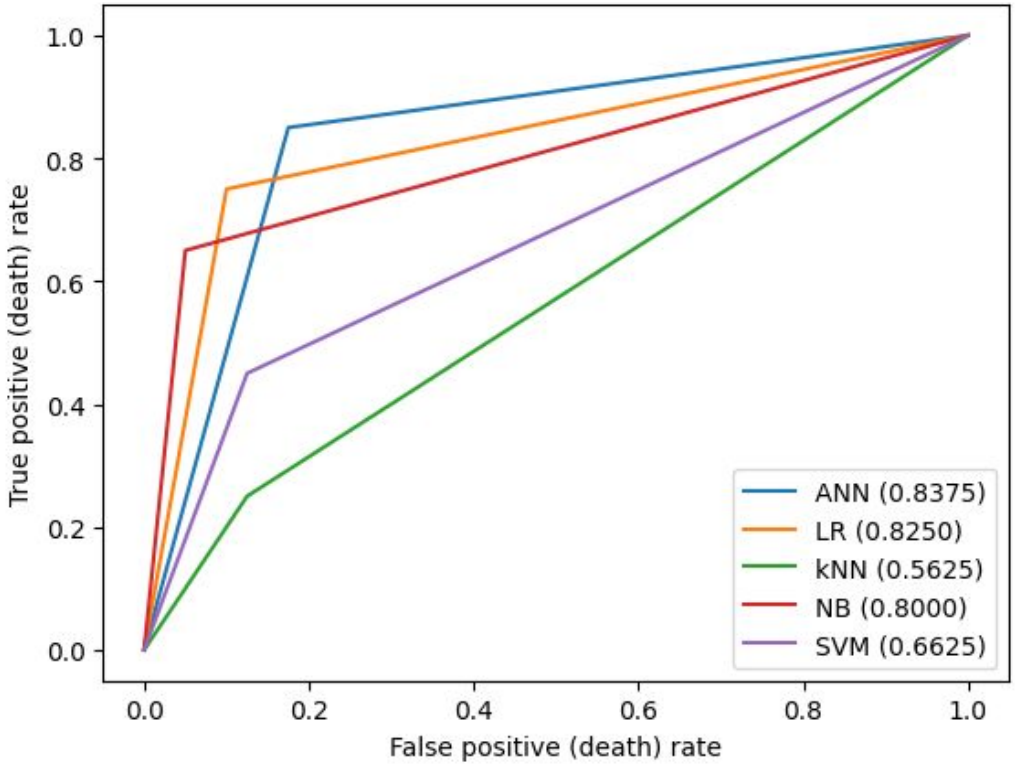
-2 false deaths

+8 false survival



Comparison

ROC Curve


1.0
0.8
0.6
0.4
0.2
0.0
True positive (death) rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive (death) rate
ANN (0.8375)
LR (0.8250)
kNN (0.5625)
NB (0.8000)
SVM (0.6625)

Comparison

	ANN	LogReg	kNN	Naive-Bayes	SVM
Accuracy	0.83	0.85	0.67	0.85	0.73
False Classifications (A)	10	9	20	9	16
False Deaths (B)	7	4	5	2	5
B/A	0.70	0.44	0.25	0.22	0.31
AUC Score	0.8375	0.8250	0.5625	0.8000	0.6625

Conclusions and Suggestions

Conclusions

The artificial neural network (ANN) performed the best.

Despite not having the least false classification, the ratio of false deaths to false classifications was the best.

From a medical standpoint, would rather falsely predict a death than a survival.

Recommendations

Larger dataset = better models

Compile neural network with different metrics.

Thanks for watching