


# Using Machine Learning to Predict Death from Heart Failure



Genevieve Anderson



# Objectives

---

- I will try to predict the occurrence of death from heart failure.
- Cardiovascular diseases are the leading cause of death so I am curious as to whether we can use machine learning and data science to predict instances of death.
- I am hoping to gain insight as to whether people may be able to stop heart conditions early on based on the observations in this dataset.

# Motivation

---

- I am very interested in this project because heart failure is the leading cause of death and I would like to see if these specific observations are successful in predicting instances of death.
- Healthcare is something that I am passionate about and I am curious to combine data science with healthcare to see what kind of results I may discover.

# Project Description

---

- The dataset from Kaggle includes information on age, anaemia, creatinine, diabetes, blood ejection, blood pressure, platelets, serum creatinine, serum sodium, gender, smoking habits, time since last seen doctor, and death instances.
- These observations will be the basis for predicting instances of death

# Heart Failure Observations

---

- Age: continuous variable
- Anaemia: binary variable, decrease of red blood cells or hemoglobin
- Creatinine: Level of the CPK enzyme in the blood (mcg/L)
- Diabetes: binary variable
- blood ejection: Percentage of blood leaving the heart at each contraction (percentage)
- blood pressure: binary variable, If the patient has hypertension
- 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)
- Gender: binary variable
- smoking habits: binary variable
- time since last seen doctor: Follow-up period (days)
- death instances: binary target variable

# Project Outline

---

- I will clean the data prior to any creation of models to remove null values and data we may not need.
- I will then split the data into train and test sets.
- Then, I will create a Naive Bayes model, a logistic regression model, tuned Random Forest model, and lastly tuned XG Boost model to compare results against each other and see which model performs the best
- Lastly, I will modify the dataset based on my research to see if I can achieve higher accuracy

# Initial look at the data set

- I will be trying to predict instances of death based on several health factors seen in the preview of the data set

index	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
0	75.0	0	582	0	20	1	265000.0	1.9	130	1	0	4	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	65.0	0	146	0	20	0	162000.0	1.3	129	1	1	7	1
3	50.0	1	111	0	20	0	210000.0	1.9	137	1	0	7	1
4	65.0	1	160	1	20	0	327000.0	2.7	116	0	0	8	1

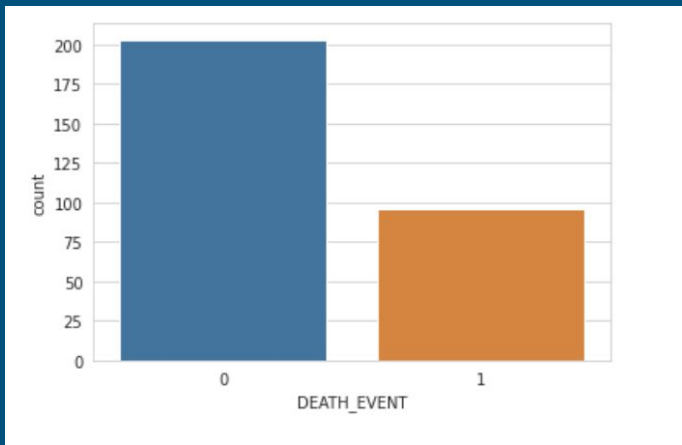
- I also check to make sure that there were no null values and looked at the data types

```
# 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
dtypes: float64(3), int64(10)
memory usage: 30.5 KB
```

# Creating the target

---

- I made “DEATH\_EVENT” my target variable since I am trying to predict whether or not someone will die from heart failure
- I also looked at the distribution of my target variable



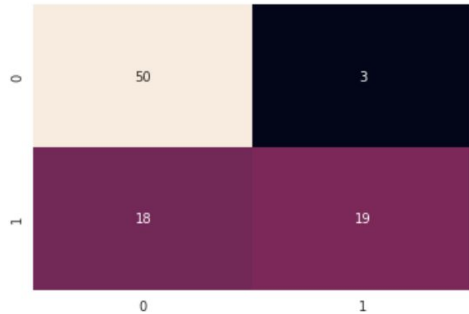


# Naive Bayes

---

- First I decided to use Gaussian Naive Bayes since my data is continuous
- I got 76% accuracy which is not great, I will explore other models to see if I can do better
  - The false negative rate is also slightly alarming

```
Accuracy.....: 76.6667  
Confusion matrix.:  
[[50  3]  
 [18 19]]
```



# Logistic Regression

- I used logistic regression since my target variable is binary
- Logistic Regression had slightly worse overall accuracy compared to Naive Bayes

```
▶ # overall accuracy  
from sklearn.metrics import accuracy_score  
print(accuracy_score(y_test,y_pred))
```

```
☞ 0.7444444444444445
```

	precision	recall	f1-score	support
0	0.72	0.92	0.81	53
1	0.82	0.49	0.61	37
accuracy			0.74	90
macro avg	0.77	0.71	0.71	90
weighted avg	0.76	0.74	0.73	90

# Random Forest tuned using randomized search

- Random forest model is made up of multiple decision trees
- I tuned the model using randomized search
- So far, this model does the best with 77% accuracy
  - It is interesting to see that the no tuning model performed better than when it was tuned
- The confusion matrices are also most promising for these models

```
from sklearn.metrics import accuracy_score

# Overall accuracy
print(accuracy_score(y_test,ypred_notuning))
print(accuracy_score(y_test,ypred_random))

0.7777777777777778
0.7666666666666667
```

No tuning:

	precision	recall	f1-score	support
0	0.94	0.75	0.83	67
1	0.54	0.87	0.67	23
accuracy			0.78	90
macro avg	0.74	0.81	0.75	90
weighted avg	0.84	0.78	0.79	90

```
[[50 17]
 [ 3 20]]
```

Randomized:

	precision	recall	f1-score	support
0	0.91	0.75	0.82	64
1	0.57	0.81	0.67	26
accuracy			0.77	90
macro avg	0.74	0.78	0.74	90
weighted avg	0.81	0.77	0.78	90

```
[[48 16]
 [ 5 21]]
```

# XG Boost tuned using randomized search

---

- XG Boost is an implementation of gradient boosted decision trees
- This model had similar results to the other models, having the same accuracy as naive bayes and random forest

```
Accuracy.....: 76.6667
Precision.....: 80.7692
Recall.....: 56.7568
FP Rate.....:9.4340
ROC AUC (probs)..: 0.881693
Confusion matrix.:
[[48  5]
 [16 21]]
```

# Trying to improve

---

- Based on my research, I discovered that the most important factors in predicting heart failure is diabetes, smoking habits, blood pressure, cholesterol, and obesity/ physical activity
- I noticed that my data set had observations for diabetes, smoking habits, and blood pressure, so I adjusted my dataframe to only hold these observations

	diabetes	high_blood_pressure	smoking	DEATH_EVENT
0	0	1	0	1
1	0	0	0	1
2	0	0	1	1
3	0	0	0	1
4	1	0	0	1

# Bernoulli Naive Bayes

---

- On the new dataframe, I decided to run a bernoulli naive bayes model since my data is now all binary
- I got an accuracy of 58%
- This was not successful, likely not enough data

```
Accuracy.....: 58.8889
Precision.....: 58.8889
Recall.....: 58.8889
FP Rate.....:41.1111
F1 score.....:0.5889
```

# Conclusion

---

- In conclusion, my random forest model using the entire dataset were the most successful with my best accuracy being 77%
- Clearly, the other variables in the dataset (aside from the three that are supposedly the biggest factors) play an important factor in determining death from heart disease
- This shows that perhaps with a few more observations or more records of data the models could have worked very well
- Logistic regression typically does better with large data sets, so that could be why this model performed the worst

# How to improve

---

- The main way that my models could have been improved was better data
  - More records (299 in this dataset)
  - Adding a cholesterol field and an obesity field
  - Even distribution of the target variable
- Oftentimes the data that we are provided with won't always be ideal, so I think that it is important to try your best to work with the data given which is why I did not change my data set.