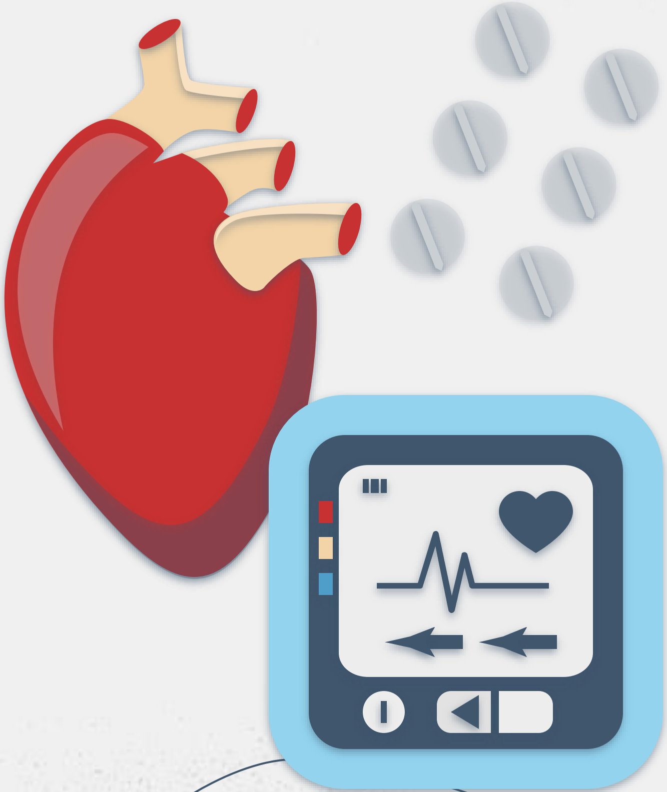
Prediction of Heart Disease Development

Project based on a dataset from the CDC

US Centers for Disease Control and Prevention



Maud Gesbert

TABLE OF CONTENT

01

The dataset

02

Preliminary analyses

03

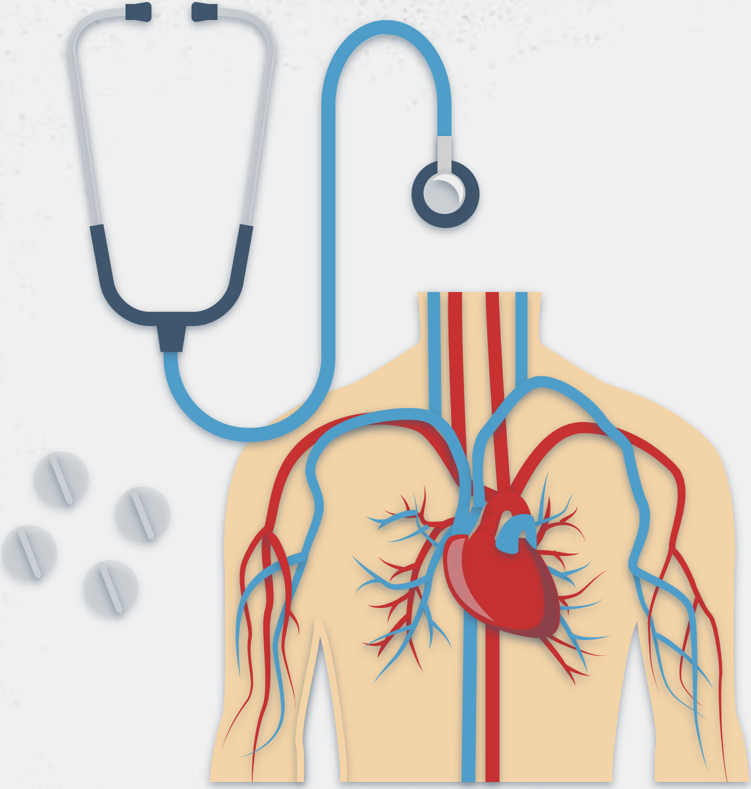
Model testing and final selection

04

Selection of fields to fill

05

Conclusion



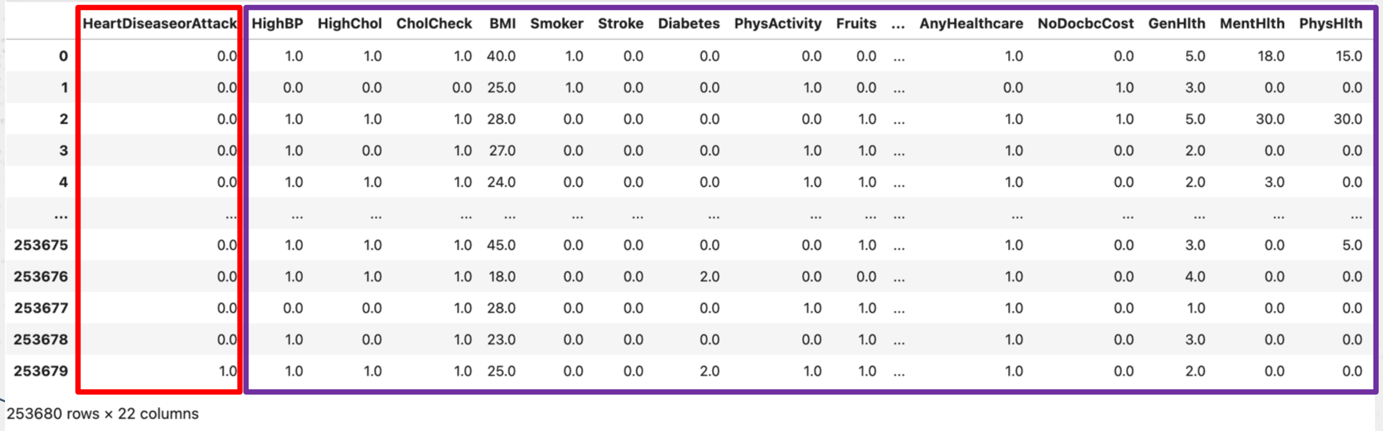
01

The dataset

Dataset overview

Dataset available on Kaggle, found on the CDC website:

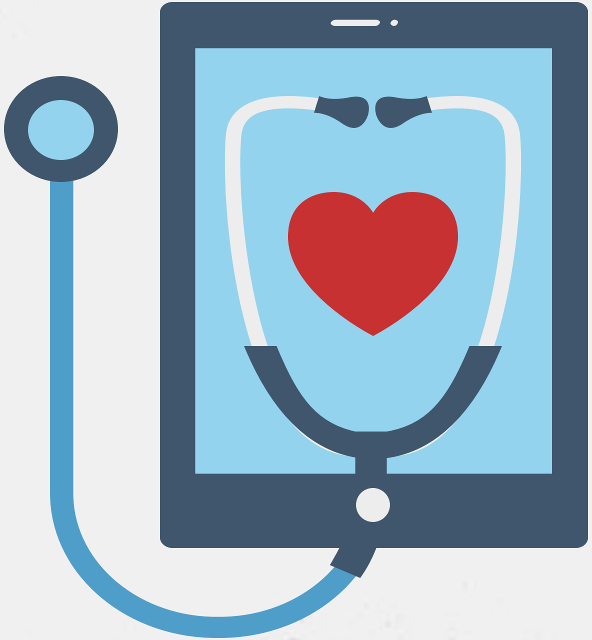
- Contains anonymized information on 253,680 patients
- Includes a target column: indicates whether the person developed heart disease or had a heart attack, this is the information the algorithm will need to predict after being trained
- Includes 21 additional columns: represent the responses given by the patients and will allow the algorithm to identify which types of patients developed a condition



	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivity	Fruits	...	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth
0	0.0	1.0	1.0	1.0	40.0	1.0	0.0	0.0	0.0	0.0	...	1.0	0.0	5.0	18.0	15.0
1	0.0	0.0	0.0	0.0	25.0	1.0	0.0	0.0	1.0	0.0	...	0.0	1.0	3.0	0.0	0.0
2	0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	0.0	1.0	...	1.0	1.0	5.0	30.0	30.0
3	0.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	2.0	0.0	0.0
4	0.0	1.0	1.0	1.0	24.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	2.0	3.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
253675	0.0	1.0	1.0	1.0	45.0	0.0	0.0	0.0	0.0	1.0	...	1.0	0.0	3.0	0.0	5.0
253676	0.0	1.0	1.0	1.0	18.0	0.0	0.0	2.0	0.0	0.0	...	1.0	0.0	4.0	0.0	0.0
253677	0.0	0.0	0.0	1.0	28.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	1.0	0.0	0.0
253678	0.0	1.0	0.0	1.0	23.0	0.0	0.0	0.0	0.0	1.0	...	1.0	0.0	3.0	0.0	0.0
253679	1.0	1.0	1.0	1.0	25.0	0.0	0.0	2.0	1.0	1.0	...	1.0	0.0	2.0	0.0	0.0

253680 rows × 22 columns

- 253,680 surveyed individuals
- 22 columns in total



02

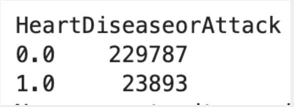
Preliminary Analyses

* See Heart_Disease_EDA file

Study of the target variable and distributions

The first step is to study the dataset, including the target column as well as the different variables (21 columns).

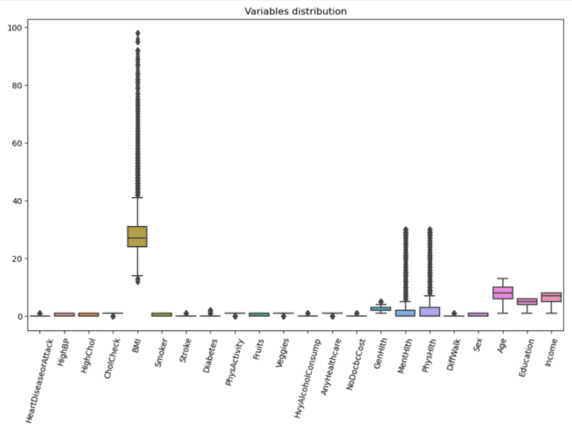
- When the target column contains a 0: the person did **not** develop heart disease
- When the target column contains a 1: the person **did** develop heart disease



```
HeartDiseaseorAttack
0.0     229787
1.0      23893
```

There is a clear imbalance: the number of people without heart disease is much higher. This needs to be taken into account to avoid misleading the model during training.

Then,

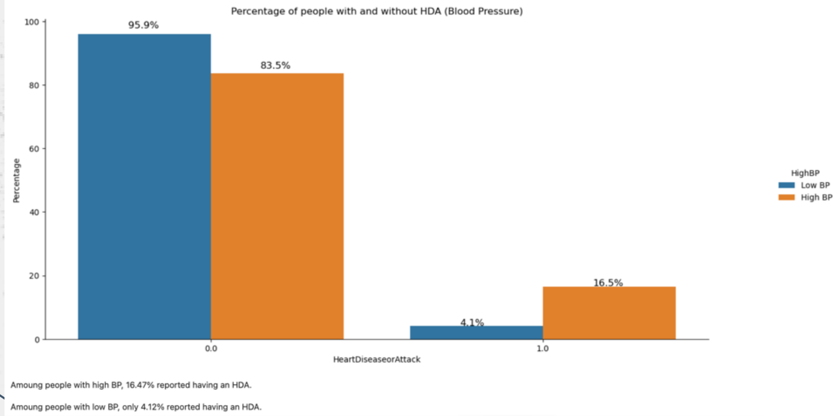


- Study of Variable Distributions

Study of non-target variables

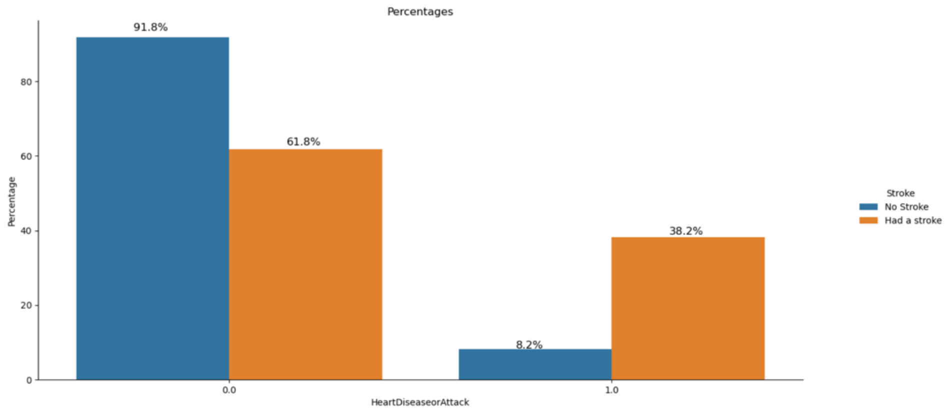
Analysis of variable distributions and their relationship with the likelihood of developing heart disease, for example:

- What percentage of people with high blood pressure developed heart disease? (16.5%)
- What percentage of people who had a stroke developed heart disease? (38.2%)



Amoung people with high BP, 16.47% reported having an HDA.

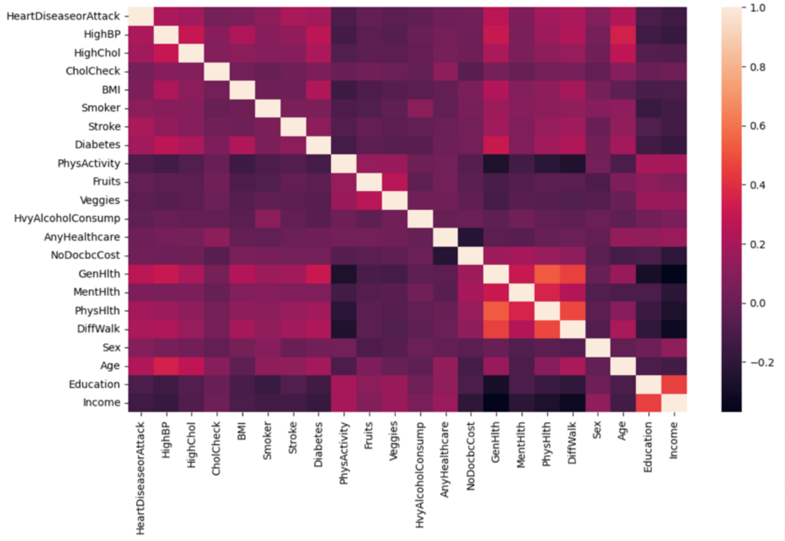
Amoung people with low BP, only 4.12% reported having an HDA.



Study of correlations between variables

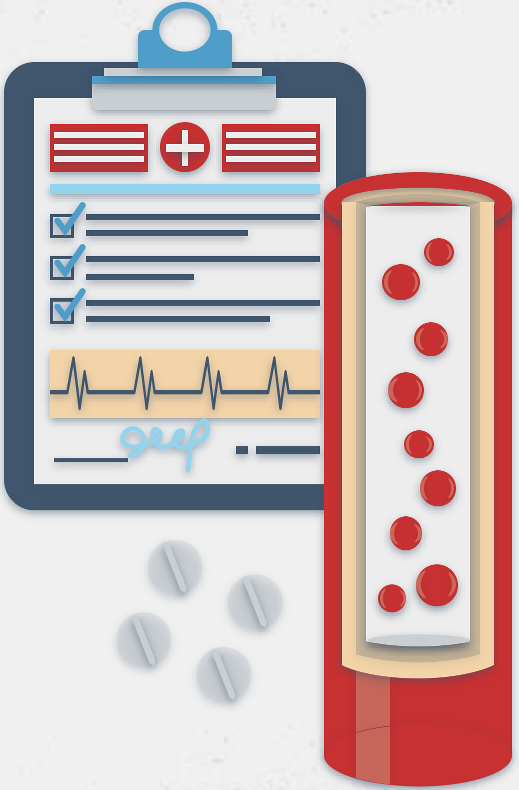
Quick analysis of Pearson correlation between variables:

- Get an initial idea of which variables are likely to have the most influence on the prediction
- Ensure that there is no strong linear correlation between two variables (apart from correlation with the target variable)



No variables appear to be strongly linearly correlated with the target

No variables are strongly correlated with each other (correlation > 0.7)



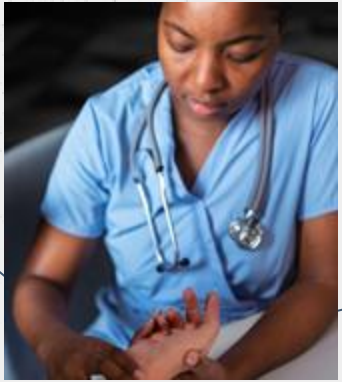
03

Model testing and final selection

* See ML_models_creation_with_all_variables file

Testing multiple types of models

	KNN	LightGBM	CART	SVM
Model type	Classification	Classification	Classification	Classification
Cross validation	5 folds Stratified	NA	5 folds Stratified	NA
Training time	1 187 seconds (= 19 min)	Stopped at more than 83 minutes without results	7 seconds	Stopped at more than 90 minutes without results
Searching for the best hyperparameters	Randomized	NA	Randomized	NA
Custom score	22 598.4	NA	22 787.2	NA
Accuracy	0,91	NA	0,91	NA



Accuracy and Custom Score

- Accuracy allows us to evaluate the model in a simple way: out of 100 individuals, how many are correctly classified after training the model?

- However, in some cases, it may be useful to implement a custom score based on the context.

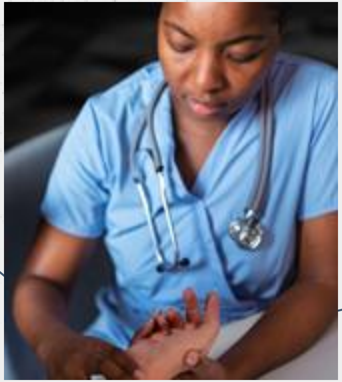
- **Why a custom score?** A classification model has four possible outcomes:

  - **True Positive (TP):** Individual correctly classified as at risk (**+1 point** for the model)
  - **True Negative (TN):** Individual correctly classified as not at risk (**+1 point** for the model)
  - **False Positive (FP):** Individual incorrectly classified as at risk (**–1 point** for the model)
  - **False Negative (FN):** Individual incorrectly classified as not at risk (**–3 points** for the model)

In this case, the most serious situation is when the model indicates that a person is not at risk when they actually are (False Negative), which may result in someone at risk not consulting a doctor.
Therefore, I created a custom score that penalizes the model more heavily in the case of FNs.
False Positives are also penalized, but it is less serious to suggest someone see a doctor when they are not at risk.

Selected model

	CART
Model Type	Classification
Cross validation	5 folds Stratified
Training time	7 seconds
Searching for the best hyperparameters	Randomized
Custom score	22 787.2
Accuracy	0,91



- Selected Model: Classification And Regression Trees – CART

- Best training time
- Best custom score
- Accuracy similar to the KNN model

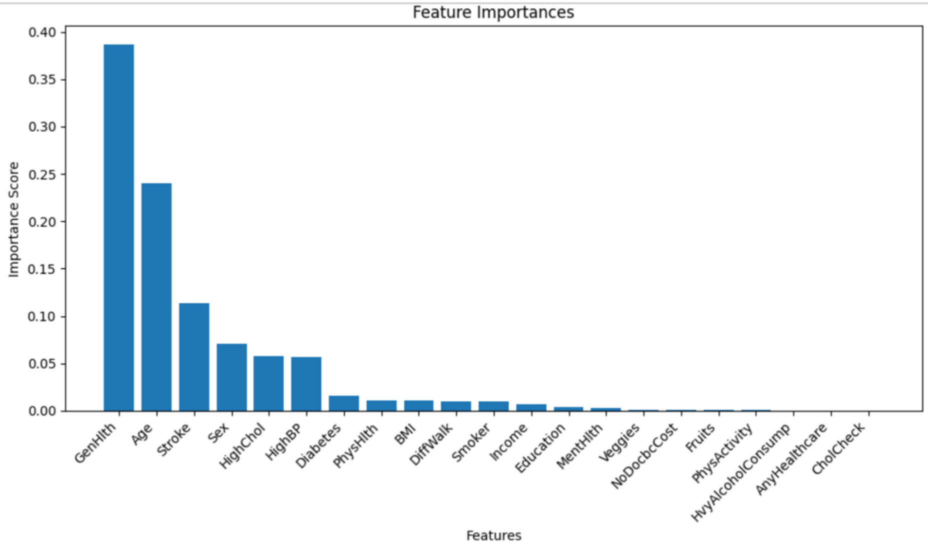
04

Selection of fields to fill

Selection of fields to be filled by model users

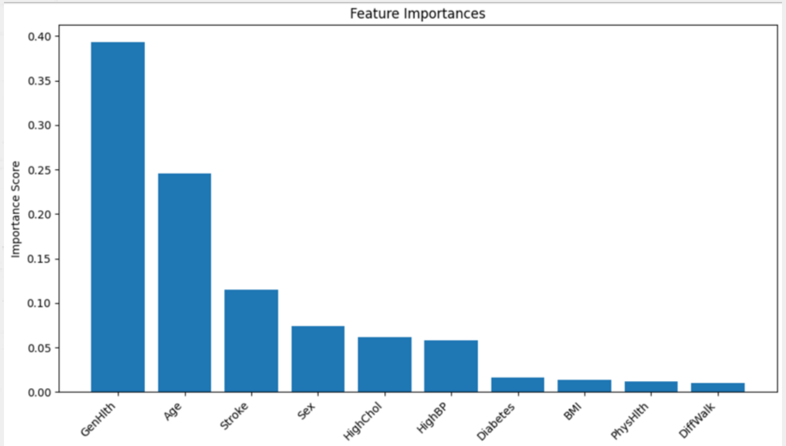
Following the model selection, I chose to analyze which variables had the most influence on the model's "decision" in order to avoid asking users to answer all 21 questions.

I identified the variables considered most important by the model and selected the top 10.



4. Selection of fields to fill

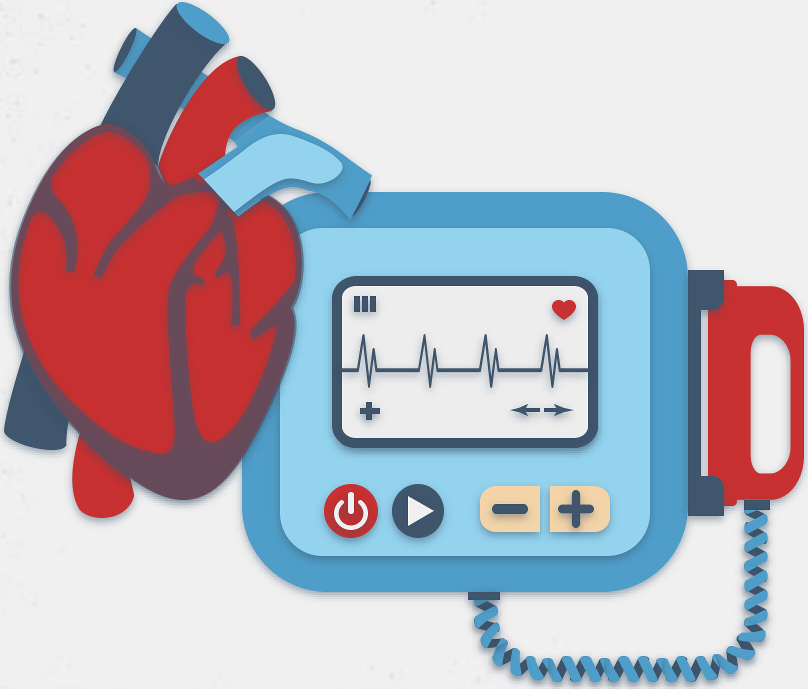
Selection of fields to be filled by model users



By using only these 10 variables:

- Accuracy remains at 0.91
- The custom score increases to 22 826

4. Selection of fields to fill



05

Conclusion

Final steps

The selected model and its hyperparameters are then saved in a pickle file.

It is then possible to send data to the model via the website using a POST method.

The pipeline associated with the model at the time of saving:

- Applies the necessary preprocessing steps when receiving the data (e.g., value normalization when needed)
- Passes the data through the model so that it returns a prediction