André Reis Fernandes, Nº 97977
Gonçalo Jorge Loureiro de Freitas, Nº 98012

Heart Disease Prediction and Classification using Machine Learning

Foundations of Artificial Intelligence - Prof: Petia Georgieva (petia@ua.pt) - Academic Year 2021/22

universidade de aveiro

Outline

1. Data Overview
2. Models
    - Logistic Regression
    - Support Vector Machine (SVM)
    - Naïve Bayes
    - Decision Tree
    - K-Nearest Neighbor
    - Neural Network
3. Results
    - Confusion Matrix
    - AUC-ROC curves
    - Gradient Descent of Cost Function
4. Conclusion

1

The Data

Understanding the data and an overview

The Data (1/3)

The dataset used in this project is the Cleveland Heart Disease dataset taken from the UCI repository .The dataset consists of 297 individuals. There are 14 columns in the dataset, which are described next.

The Data (2/3)

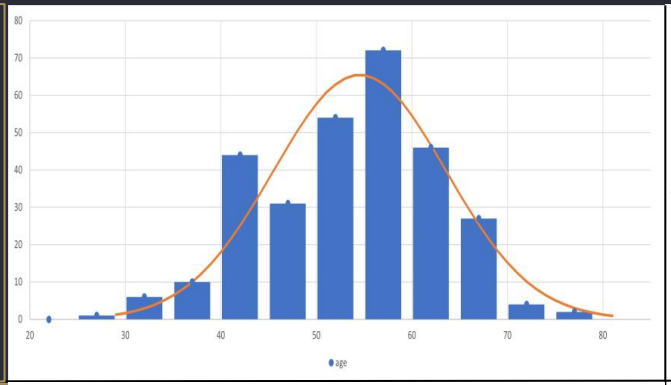
1. Age, in years
2. Sex, 1 = male; 0 = female
3. chest pain type
    - Value 0: typical angina
    - Value 1: atypical angina
    - Value 2: non-anginal pain
    - Value 3: asymptomatic
4. trestbps: resting blood pressure
5. chol : serum cholestoral in mg/dl
6. fasting blood sugar (> 120 mg/dl)
7. restecg: resting electrocardiographic results
    - Value 0: normal
    - Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
    - Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria

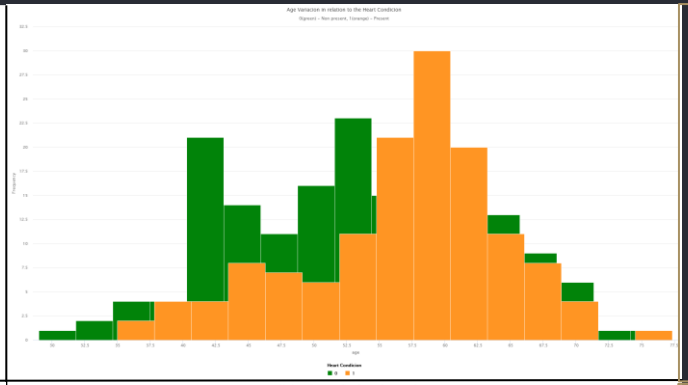
The Data (3/3)

1. thalach: maximum heart rate achieved
2. exang: exercise induced angina
3. oldpeak = ST depression induced by exercise relative to rest
4. condition: 0 = no disease, 1 = disease
5. slope: the slope of the peak exercise ST segment
    - Value 0: upsloping
    - Value 1: flat
    - Value 2: downsloping
6. ca: number of major vessels (0-3) colored by flourosopy
7. thal: 0 = normal; 1 = fixed defect; 2 = reversable defect
8. and the label
9. condition: 0 = no disease, 1 = disease

Data Overview (1/5)

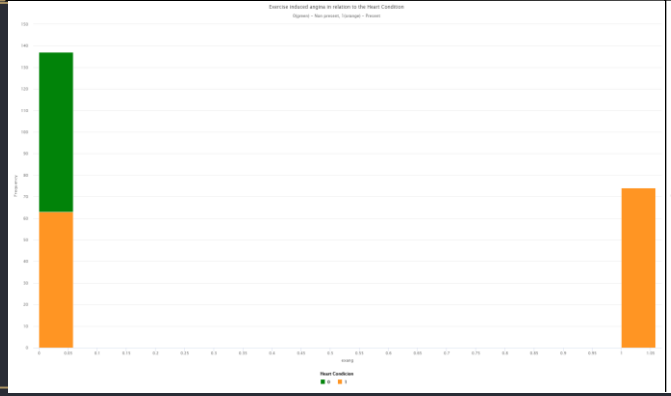
Age Distribution

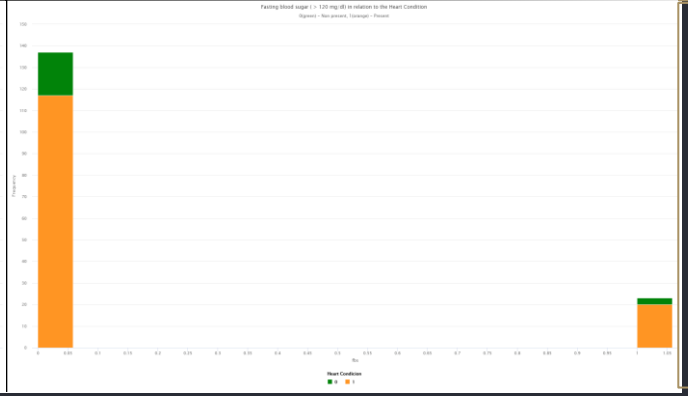




Age Variacion in relation to the Heart Condicion

Exercise induced angina in relation to the Heart Condition

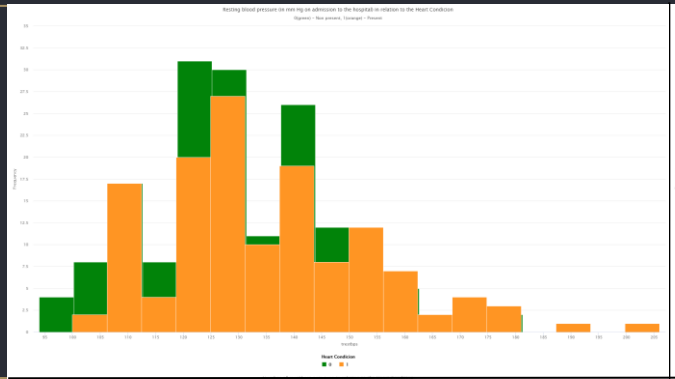




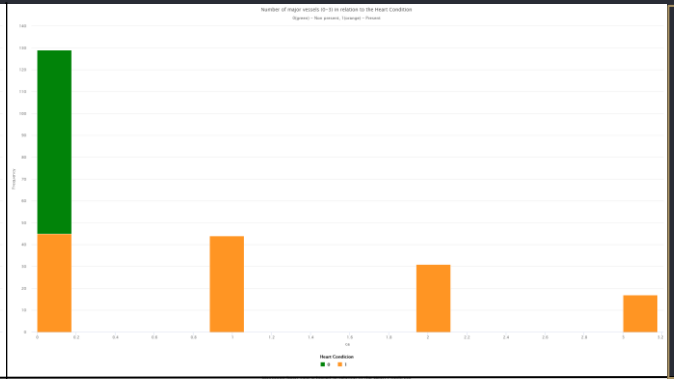
Fasting blood sugar (120 mg/dl) in relation to the Heart Condition

Data Overview (2/5)

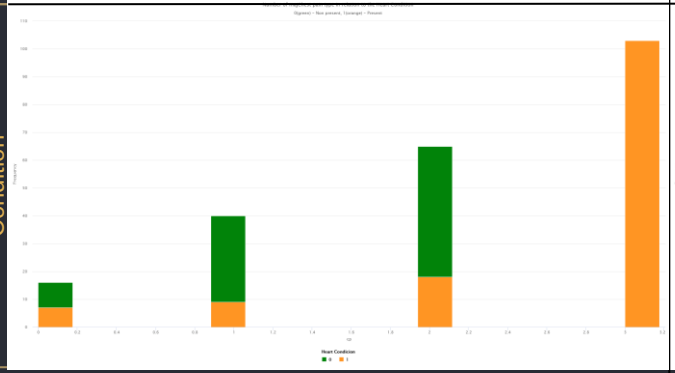
Resting blood pressure (in mm Hg on admission to the hospital) in relation to the Heart Condicion



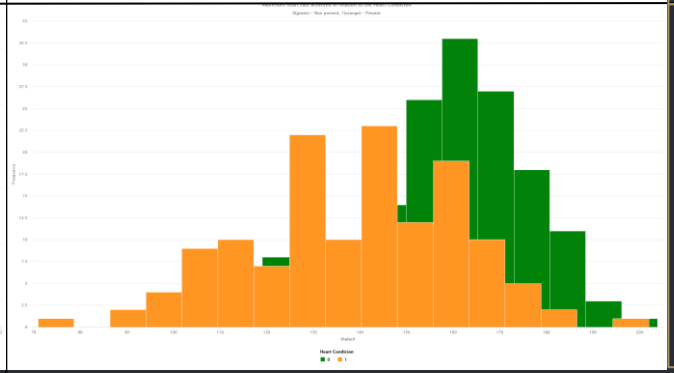
Number of major vessels (0-3) in relation to the Heart Condition



Number of majChest pain type in relation to the Heart Condition

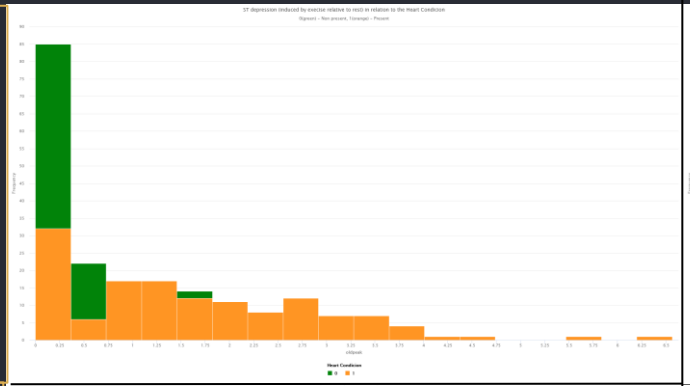


Maximum heart rate achieved in relation to the Heart Condicion

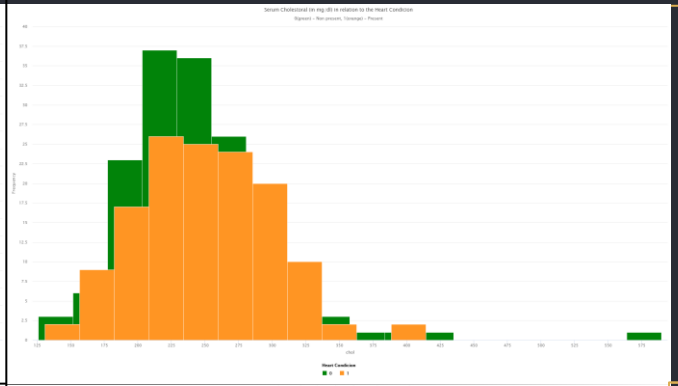


Data Overview (3/5)

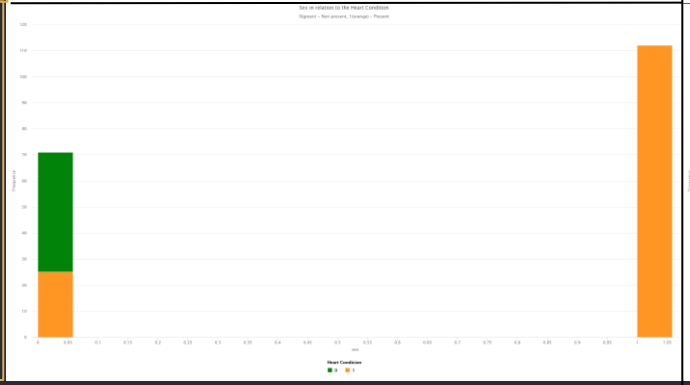
ST depression (induced by execise relative to rest) in relation to the Heart Condicion



Serum Cholestoral (in mg_dl) in relation to the Heart Condicion

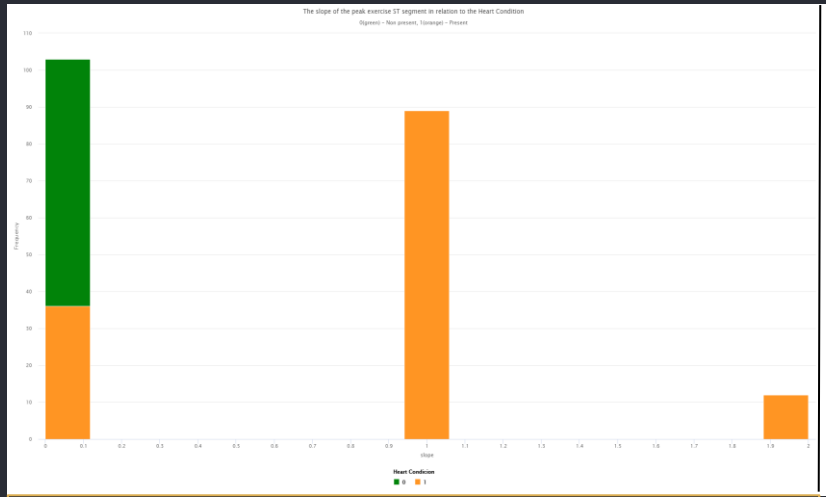


Sex in relation to the Heart Condition

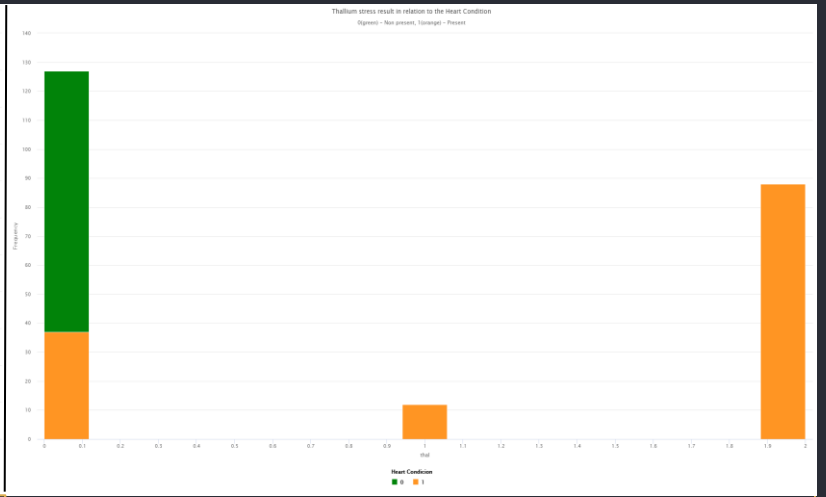


Resting electrocardiographic results in relacion to the Heart Condition

Data Overview (4/5)



The slope of the peak exercise ST segment in relation to the Heart Condition



Thallium stress result in relation to the Heart Condition

Data Overview (5/5)

	age	*sex*	*cp*	*trestbps*	*chol*	*fbs*	*restecg*	*thalach*
age	1							
sex	-0.0924	1						
cp	0.110471	0.008908	1					
trestbps	0.290476	-0.06634	-0.03698	1				
chol	0.202644	-0.19809	0.072088	0.131536	1			
fbs	0.132062	0.03885	-0.05766	0.18086	0.012708	1		
restecg	0.149917	0.033897	0.063905	0.149242	0.165046	0.068831	1	
thalach	-0.39456	-0.0605	-0.33931	-0.04911	-7.5E-05	-0.00784	-0.07229	1
exang	0.096489	0.143581	0.377525	0.066691	0.059339	-0.00089	0.081874	-0.38437
oldpeak	0.197123	0.106567	0.203244	0.191243	0.038596	0.008311	0.113726	-0.34764
slope	0.159405	0.033345	0.151079	0.121172	-0.00922	0.047819	0.135141	-0.38931
ca	0.36221	0.091925	0.235644	0.097954	0.115945	0.152086	0.129021	-0.26873
thal	0.120795	0.370556	0.266275	0.130612	0.023441	0.051038	0.013612	-0.25839

Correlation Matrix

2

The Models

Simple Introduction of the models used in this problem

The Models

- Logistic Regression (Python)
- Support Vector Machine (Python)
- Gaussian Naive Bayes (Python)
- Decision Tree (Python)
- K-Nearest-Neighbor (Python)
- Neural Networks (Rapid Miner)

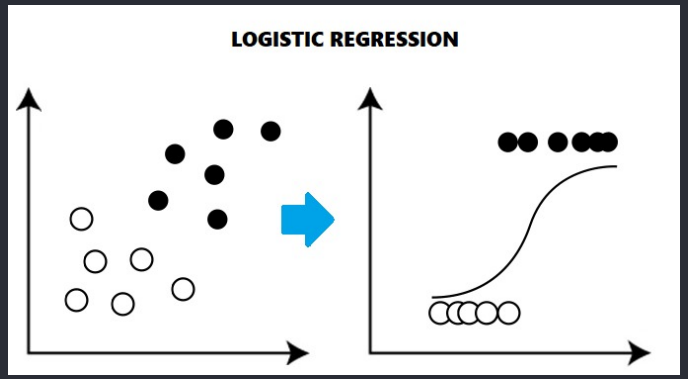
The Approach

- 80:20 ratio split for all models.
- The models developed in Pyhton will use the function GridSearchCV, looking for the parameters that gives us the less error on our train set.
- In the last model we use a different approach that will be explained ahead.

Logistic Regression

$$S(z) = \frac{1}{1 + e^{-z}}$$

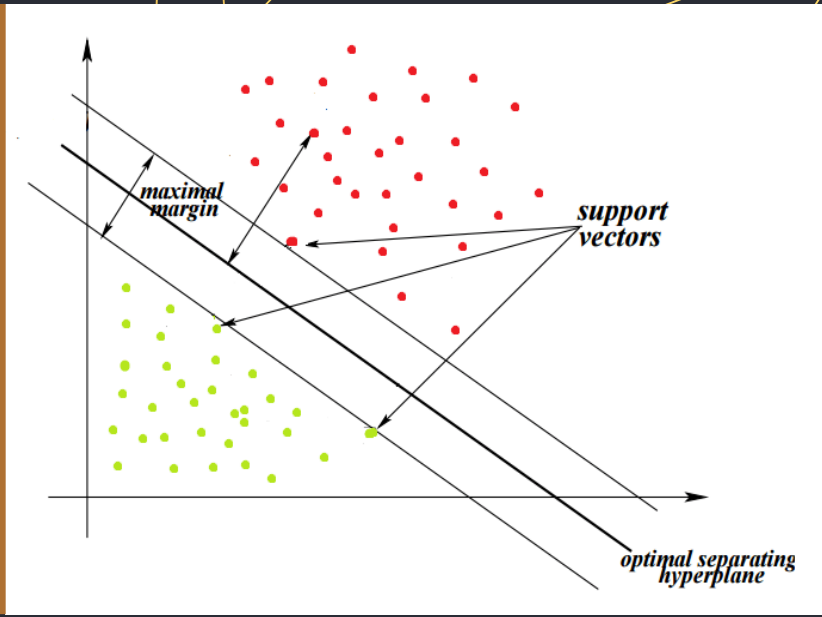
- Also know as Logit Model
- Used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a *logistic regression equation.*
- In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1.
- $P \geq 0{,}5 \rightarrow class = 1 \quad \text{or} \quad P < 0{,}5 \rightarrow class = 0$



- Best parameters: [C = 0.1; penalty='l2'; solver='newton-cfg'] the other parameters set to default
- Make sure the model converges by changing the max_iter parameter

Support Vector Machine (SVM)

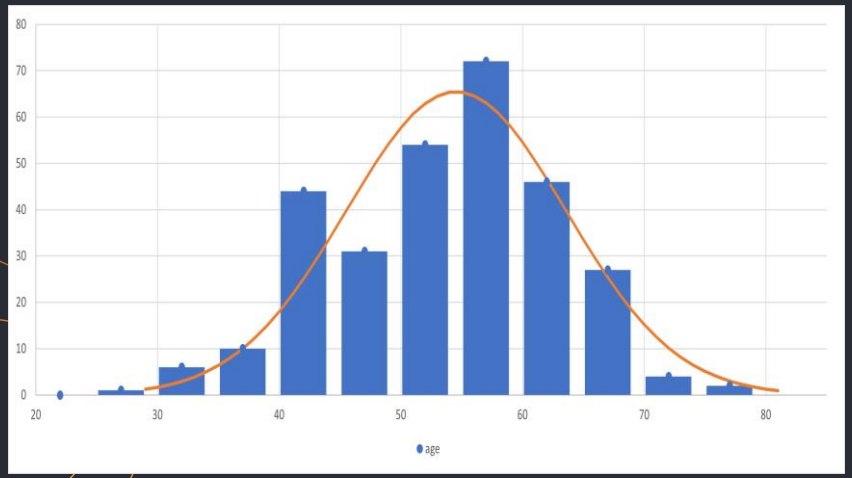
- Hyper-plane: a line (or other, depending on the dimension) that can separate the given data.
- Maximizing the distance between the nearest points of each class and the hyper-plane would result in an optimal separating hyper-plane. This distance is called the margin.
- Main idea: Identify the optimal separating hyper-plane which maximizes the margin of the training data.



Best parameters:
kernel =' poly', C = 1 and gamma = 0.0001

Naive Bayes (Gaussian)

- Easiest and rapidest classification method available, and it is well suited for dealing with enormous amounts of data.
- Makes predictions about unknown classes using the *Bayes theory* of probability.
- Is employed when the predictor values are continuous and are expected to follow a *Gaussian distribution*.
- Doesn´t have any parameters to tunning.



Bayes Theory of Probability

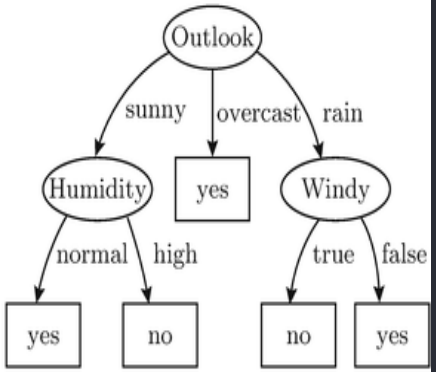
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- P(A|B) is the probability of event A occurring, given event B has occurred
- P(B|A) the probability of event B occurring, given event A has occurred.
- P(A) the probability of event A and P(B) the probability of event B.
- Note that events A and B are independent events (i.e., the probability of the outcome of event A does not depend on the probability of the outcome of event B).

Decision Tree – Example

- Start at the top
- Split our dataset into distinguished leaf nodes (using if/else conditions)
- These splitting criteria are carefully calculated using a splitting technique (explained ahead).

Outlook	*Temp*	*Humidity*	*Windy*	*Golf?*
rainy	hot	high	false	no
rainy	hot	high	true	no
overcast	hot	high	false	yes
sunny	mild	high	false	yes
sunny	cool	normal	false	yes
sunny	cool	normal	true	no
overcast	cool	normal	true	yes
rainy	mild	high	false	no
rainy	cool	normal	false	yes
sunny	mild	normal	false	yes
rainy	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
sunny	mild	high	true	no



Example: are there good conditions to play golf?

Decision Tree – Understanding the model

Want to minimize the error ⟶ Want the best splits

1) Gini Index

Cost/loss function that is used by decision trees to choose which feature will be used for splitting the data, and at what point the column should be split.

2) Entropy

Measures the randomness or disorders in a system. In terms of data, we can define it as the randomness in the information we are processing.

$$GINI(node) = 1 - \sum_{Class_j} [P(Class_j|node)]^2$$

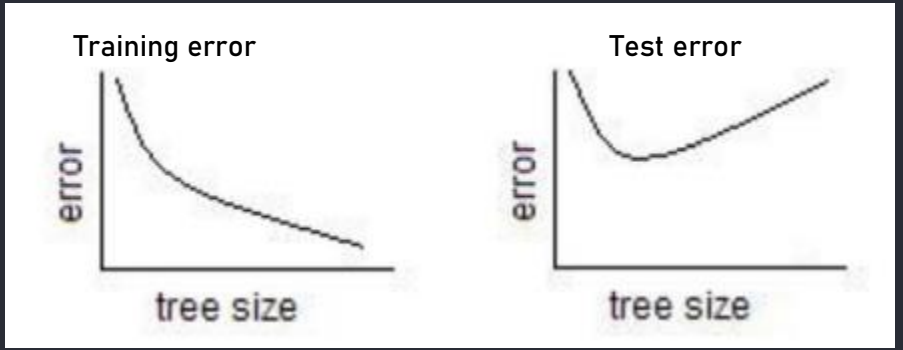
$$H(node) = -\sum_{Class_j} [P(Class_j|node) \cdot \log(P(Class_j|node))]$$

$P(Class_j|node)$ being the probability of $Class_j$.

Decision Tree – Understanding the model

3) Classification Error

- Cut off the growing process at various points over the growing process
- Tree sizes increases, training error decreases, test error increases after a while
- Overfitting



$$Error(node) = 1 - max_{Class_j}[P(Class_j|node)]$$

$P(Class_j|node)$ being the probability of $Class_j$.

4) Information Gain

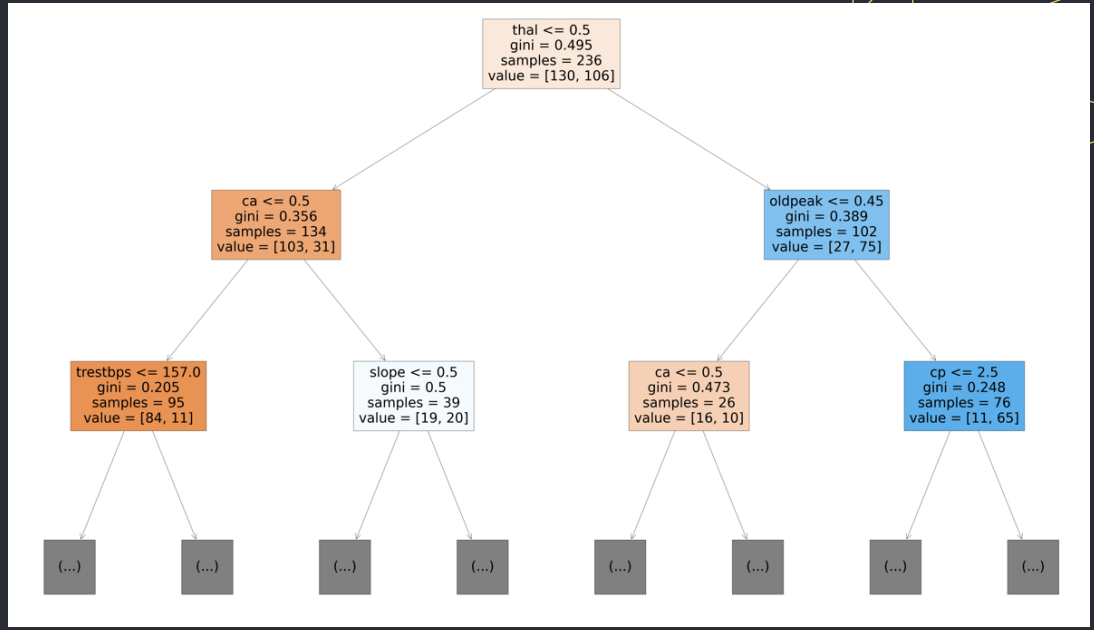
How much uncertainty was reduced after splitting on a given feature.

$$I_{gain}(node) = H_i - H_f$$

Decision Tree – Working with the Data

While training, our decision tree model evaluates all possible and picks the split with the lowest Gini Score. With the first split, all the data according to a specific condition falls towards either the left or the right of the root node. The process repeats for both left and right sides till we reach the terminating nodes representing a class in the target column.

Best parameters:
[max depth=5, max features='log2', max leaf nodes=30,min samples leaf=8, min weight fraction leaf=0.3]

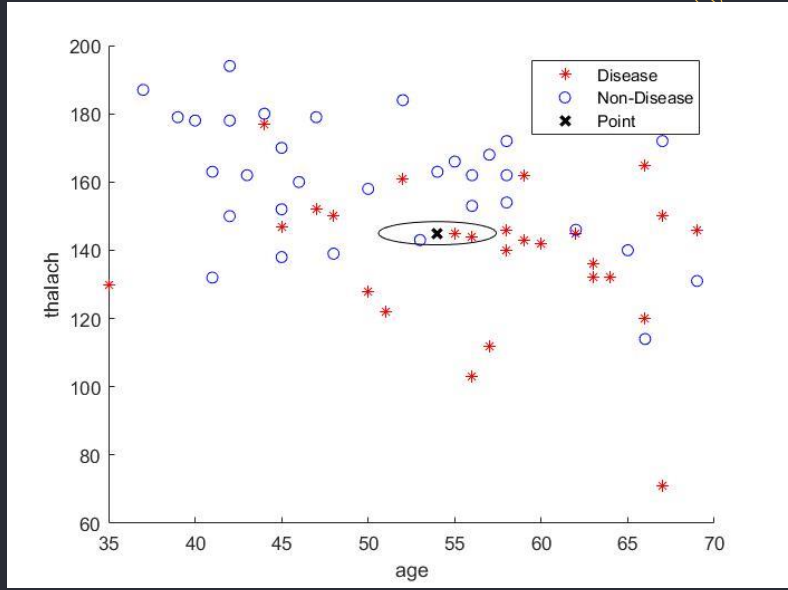


Exemple of DT with depth 3 for our data

K- Nearest Neighbor

- The KNN algorithm assumes that similar things exist in close proximity.
- Simple but gets significantly slower as the number of examples and/or predictors/independent variables increase.
- This algorithm can be explained in few steps as following:

1) Select the number of the neighbors, K
2) Calculate the Euclidean distance of K number of neighbors
3) Take the K nearest neighbors
4) Among these K neighbors, count the number of the data points in each category
5) Assign the new data points to that category for which the number of the neighbor is maximum
6) The model is ready



Exemple of K=3 for our data

Euclidean Distance of A(X1, Y1) and B(X2, Y2)

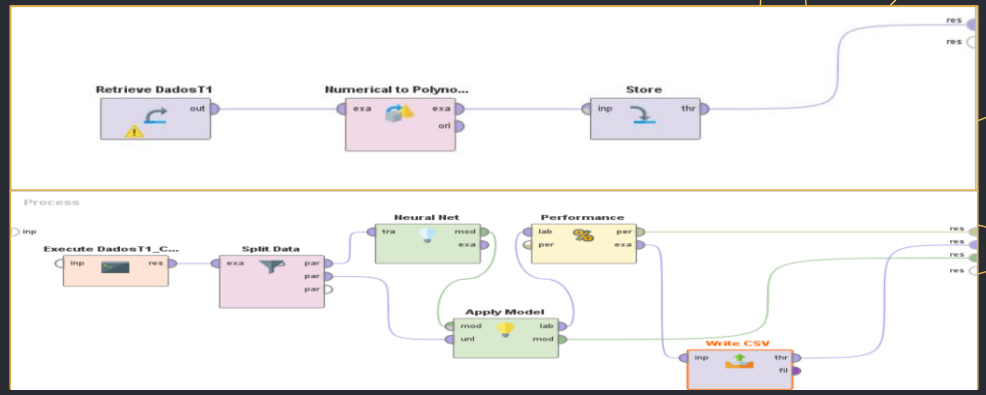
$$ED = \sqrt{((X_1 - X_2)^2) - (Y_1 - Y_2)^2)}$$

Neural Network (NN)

- The NN model was applied in the Rapid Miner (RM) software.
- The first step is a conversion of the condition to polynomial so that a huge number of possible values for the condition will be generated. The second step is the application of the model.
- This model require a big computational power and is used to obtain a good accuracy.

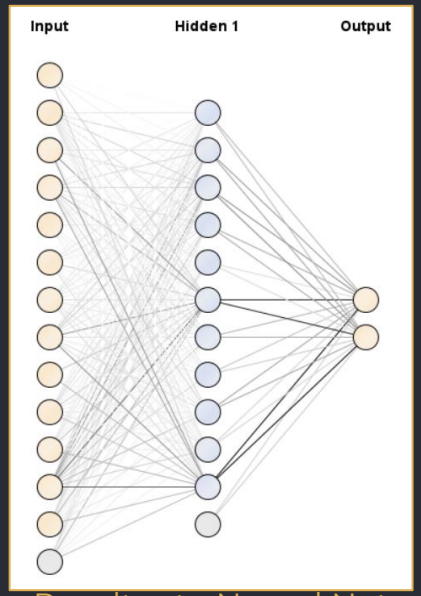
  - Who does the model work ?

Each neuron as is own weight associated wish is multiplied by the input. The data gets through the next neuron if the value is bigger then the threshold, fiering the activation function. This processes result in the predictions that the machine did using the NN model.



Model used in the RM software

- The best parameters found: 400 training cycles and learning rate of 0,001.
- We've used only one hidden layer to respect the non-deep learning and because the amount of data is not that big. This layer has 11 neurons. This number is achieved by the sum of 2/3 of the input layer with the number of neurons in the output layer.



Resultante Neural Net

3

Results

Perfomance Metrics, ROC-AUC, Gradient Descent

Confusion Matrix

	True 0	True 1
Pred 0	TP	FN
Pred 1	FP	TN

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = 2 * \frac{Recall * Precison}{Recall + Precison}$$

$$Specificity = TNR = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = TPR = \frac{TP}{TN + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

$$Balanced\ Accuracy = \frac{Recall + Specificity}{2}$$

Confusion Matrix's

CM - LogReg

	True 0	True 1
Pred 0	27	2
Pred 1	5	26

CM - NB

	True 0	True 1
Pred 0	25	4
Pred 1	5	26

CM - KNN

	True 0	True 1
Pred 0	22	11
Pred 1	8	19

CM - SVM

	True 0	True 1
Pred 0	24	5
Pred 1	7	24

CM - DT

	True 0	True 1
Pred 0	23	6
Pred 1	6	25

CM - NN

	True 0	True 1
Pred 0	29	6
Pred 1	3	21

Perfomace Metrics

TABLE VIII: Perfomace Metrics

(%)	Acc	Prec	F1	TPR	TNR	FPR	Bal Acc
LogReg	88.3	84.4	88.5	93.1	83.9	16.1	88.5
SVM	80.0	77.4	79.9	82.8	77.4	22.6	80.0
NB	85.0	83.3	84.7	86.2	83.8	16.1	85.0
DT	80.0	75.9	78.6	81.5	78.8	21.2	80.0
KNN	68.3	73.3	69.8	66.7	70.4	29.6	68.5
NN	84.75	90.63	83.71	77.78	87.50	12.50	82.64

- Red representing the best value/model
- Green representing the second-best value/model

Perfomace Metrics - Comparing

TABLE VIII: Perfomace Metrics

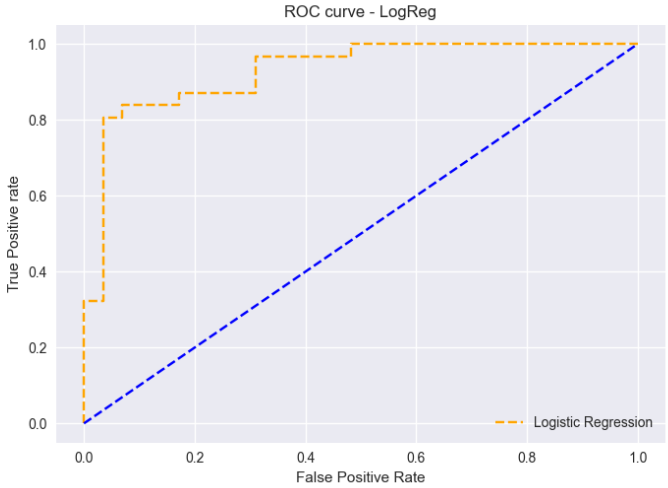
(%)	Acc	Prec	F1	TPR	TNR	FPR	Bal Acc
LogReg	88.3	84.4	88.5	93.1	83.9	16.1	88.5
SVM	80.0	77.4	79.9	82.8	77.4	22.6	80.0
NB	85.0	83.3	84.7	86.2	83.8	16.1	85.0
DT	80.0	75.9	78.6	81.5	78.8	21.2	80.0
KNN	68.3	73.3	69.8	66.7	70.4	29.6	68.5
NN	84.75	90.63	83.71	77.78	87.50	12.50	82.64

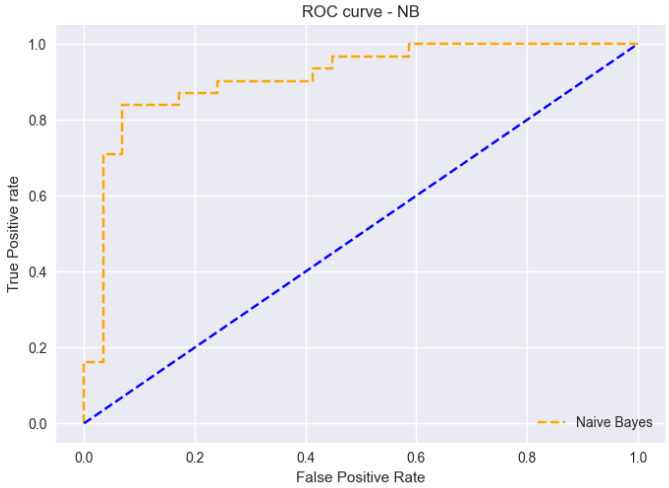
TABLE IX: Perfomace Metrics - Default Parameters

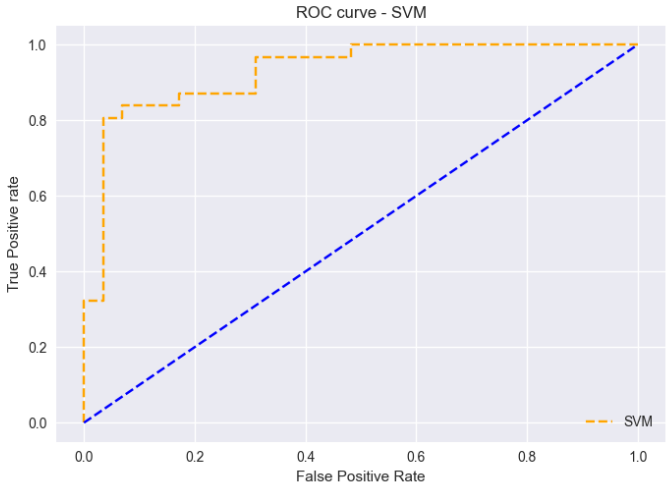
(%)	Acc	Prec	F1	TPR	TNR	FPR	Bal Acc
LogReg	86.7	83.9	86.7	89.7	83.9	16.1	86.8
SVM	61.7	57.1	67.6	82.8	42.0	58.1	62.3
DT	68.3	72.4	68.9	65.6	71.4	28.6	68.5
KNN	66.7	71.0	68.8	66.7	66.7	33.3	66.7
NN	74.6	71.9	75.4	79.3	70.0	30.0	74.7

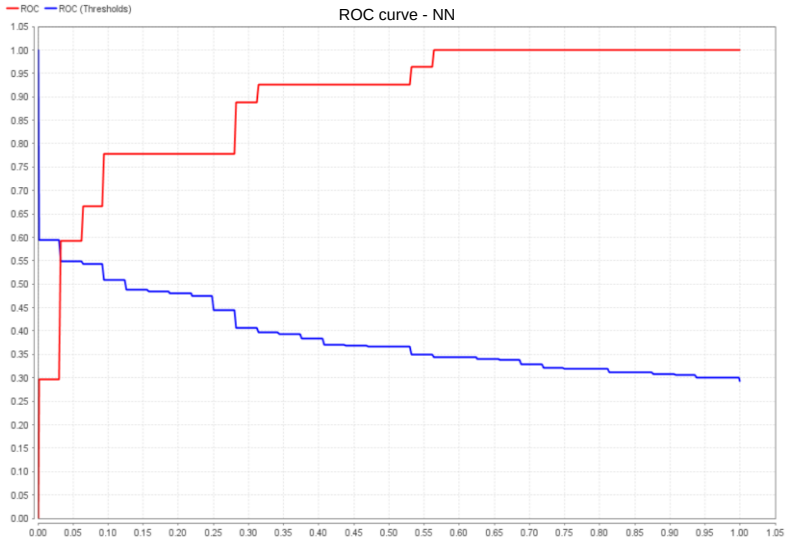
ROC-AUC Curves

- The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold
- The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes



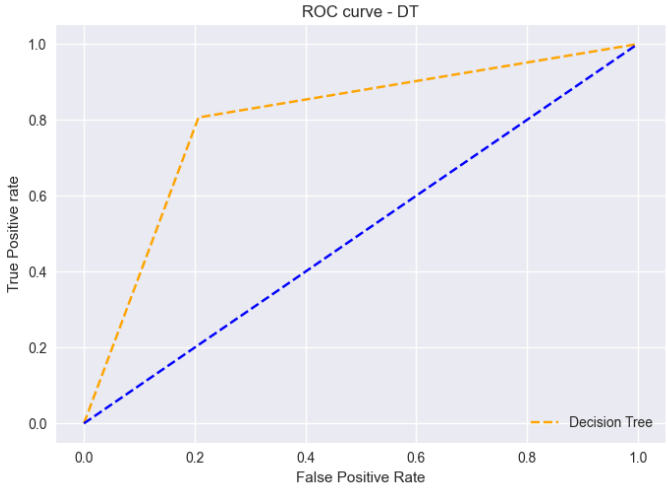


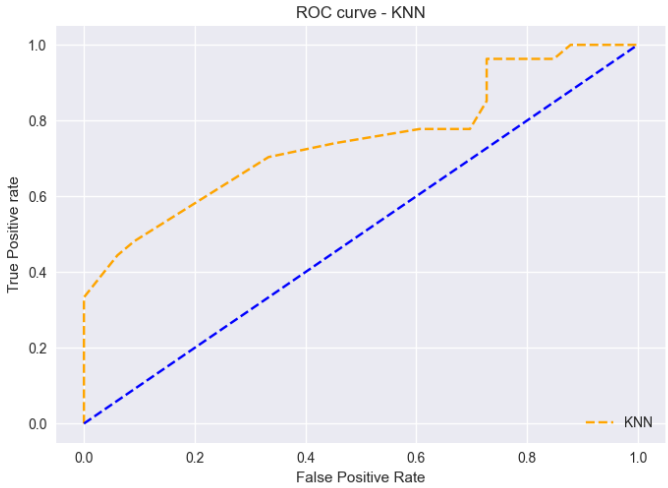




AUC - Scores:

- LogReg: 0,93
- SVM: 0,93
- NB: 0,91
- NN: 0,89



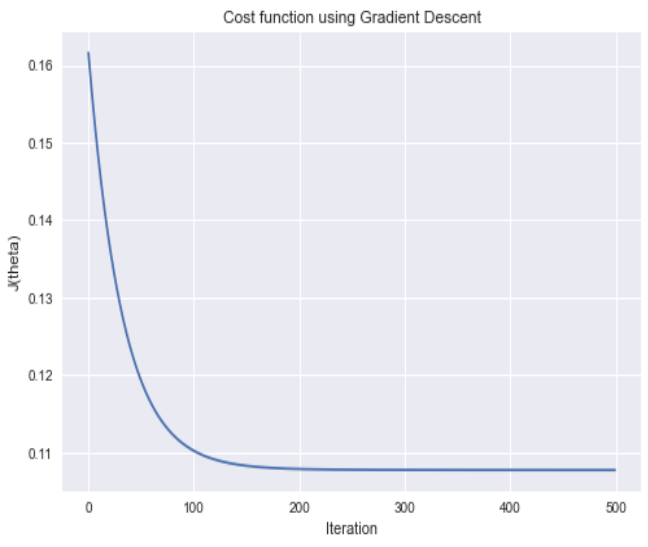


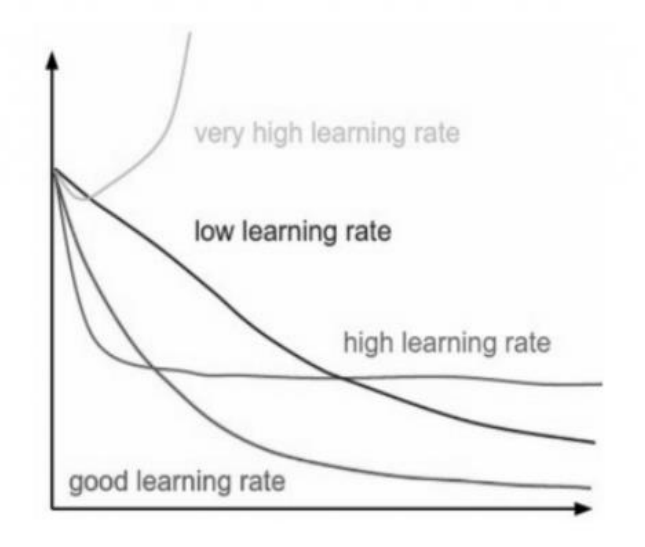
AUC - Scores:

- DT: 0,80
- KNN: 0,75

Gradient Descent of Cost Function

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^i) - y^i]x_j^i$$





- The descent gradient of the cost function was used on the generic data with a learning rate of 0,01 and with 500 iterations.
- Important to notice that a bit after 100 iterations, the function converges.

4

Conclusion

Importance of this study
Best Model

Conclusion

- Heart Disease is one of the major concerns for society today.
- It is difficult to manually determine the odds of getting heart disease based on risk factors.
- Best results came with the best accuracy and for this we need the best parameters and which model applies best to our data
- As seen in this project the best model for classification of this data set is the Logistic Regression followed by the Naïve Bayes due to being the models with best accuracy of the test set.

References

- Mariette Awad and Rahul Khanna. Support Vector Machines for Classification
- Daniel Berrar. Bayes' Theorem and Naive Bayes Classifier
- Jason Brownlee. How to Configure the Learning Rate When Training Deep Learning Neural Networks.
- Jason Brownlee. How to Configure the Number of Layers and Nodes in a Neural Network.
- Padraig Cunningham and Sarah Jane Delany. k-Nearest neighbour classifiers.
- IBM Cloud Education. Neural Networks.
- Sepp Hochreiter and Arthur Steven Younger. Learning To Learn Using Gradient Descent.
- Sandhya Krishnan. How to determine the number of layers and neurons in the hidden layer?
- Oded Maimon and Lior Rokach. Decision Trees
- Lachlan Miller. Machine Learning week 1: Cost Function, Gradient Descent and Univariate Linear Regression.
- Sarang Narkhede. Understanding AUC - ROC Curve
- Kshitiz Sirohi. Beginner: Cost Function and Gradient Descent SAS - Analytics Software Solutions. Neural Networks - What they are why they matter.
- Ravi Verma. ML Supervised Learning : Logistic Regression Model using Python.

THANKS!

André Fernandes, N° 97977
Gonçalo Freitas, N° 98012

Github Repository
https://github.com/gjfreitas/FIA-Project

University of Aveiro
Foundations of Artificial Intelligence
Prof: Petia Georgieva
Academic Year 2021/22