Artificial Intelligence Fullstack

[Course]

Week 4 – Machine Learning –

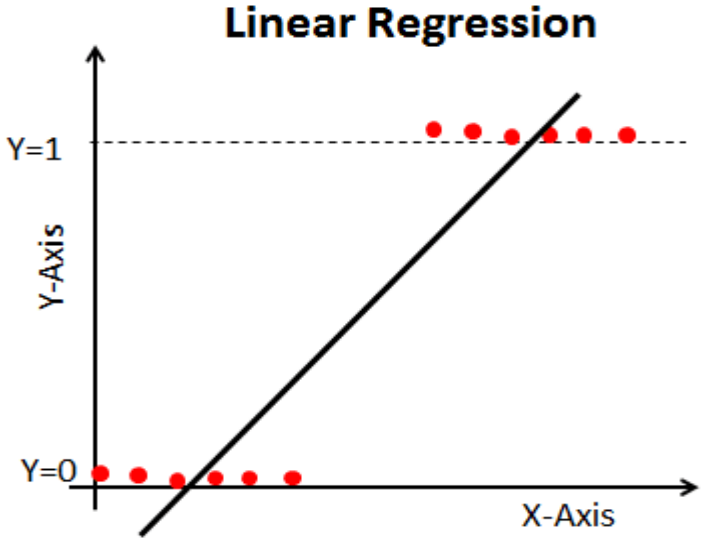
- ❑ **Logistic Regression**
- ❑ **Decision Tree Algorithm**
- ❑ **Support Vector Machines (SVMs)**

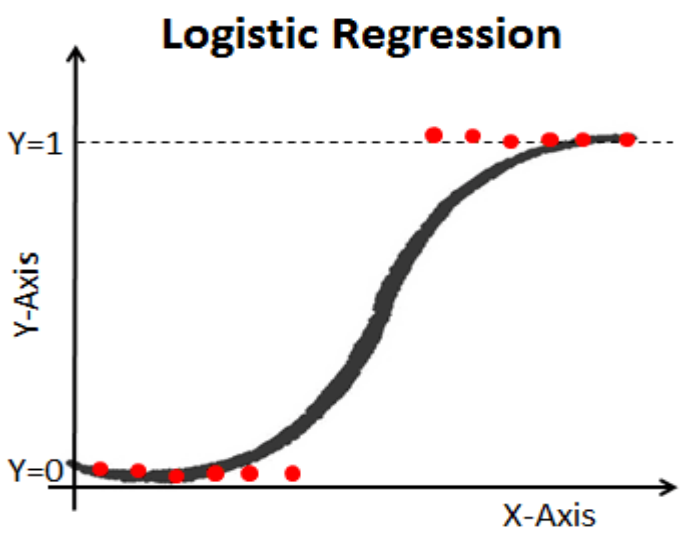
[See examples / code in GitHub code repository]

It is not about Theory, it is 20% Theory and 80% Practical – Technical/Development/Programming [Mostly Python based]

ML – Linear Regression vs Logistic Regression

Linear regression gives you a continuous output, but logistic regression provides a constant output. An example of the continuous output is house price and stock price. Examples of the discrete output are predicting whether a patient has cancer or not and predicting whether a customer will churn. Logistic regression is estimated using the maximum likelihood estimation (MLE) approach, while linear regression is typically estimated using ordinary least squares (OLS), which can also be considered a special case of MLE when the errors in the model are normally distributed.





ML | Logistic Regression

Logistic regression can be used for various classification problems, such as spam detection. Some other examples include: diabetes prediction, whether a given customer will purchase a particular product; whether or not a customer will churn, whether the user will click on a given advertisement link or not, and many more examples.

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.

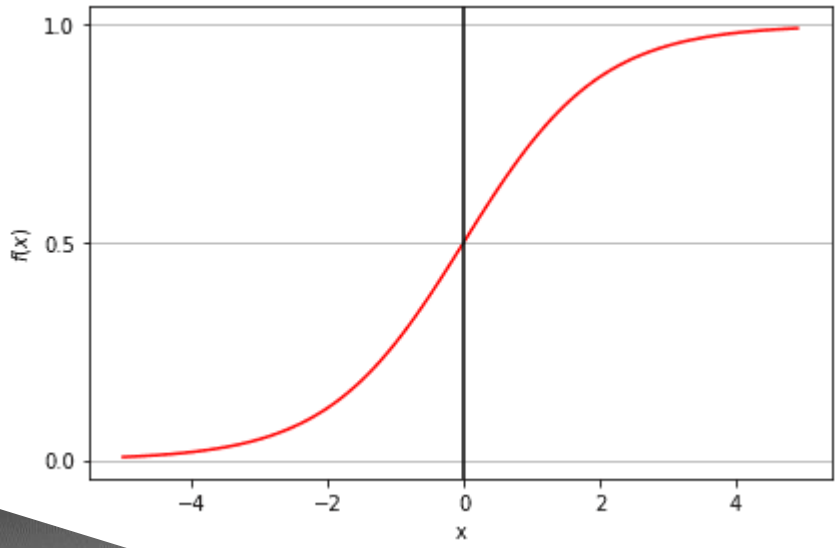
Reference:

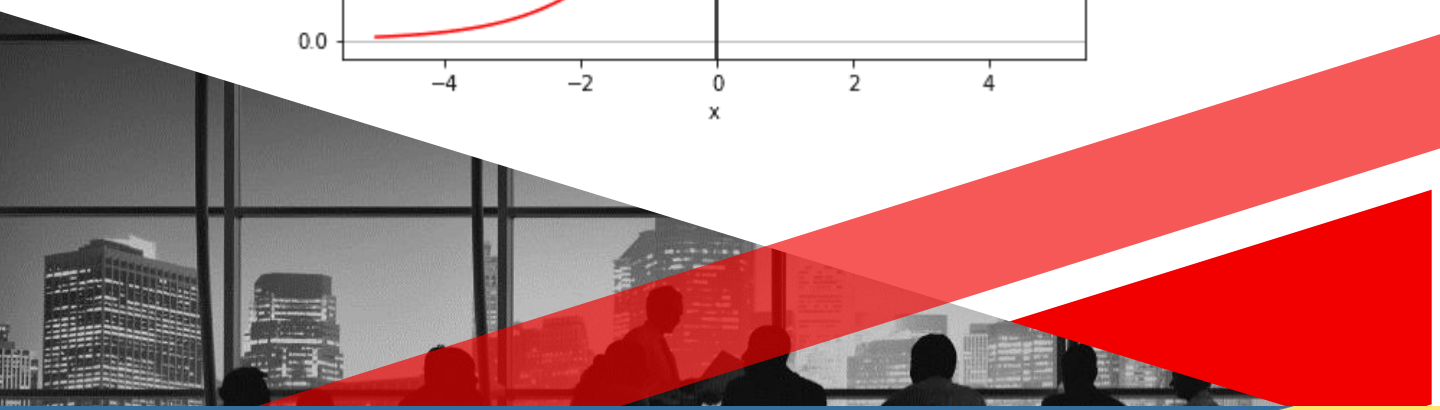
https://www.geeksforgeeks.org/ml-logistic-regression-using-python/
https://www.datacamp.com/tutorial/understanding-logistic-regression-python
https://www.digitalocean.com/community/tutorials/logistic-regression-with-scikit-learn
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.ht

ML | Logistic Regression | Sigmoid function

The sigmoid function, also called logistic function, gives an ‘S’ shaped curve that can take any real-valued number and map it into a value between 0 and 1. If the curve goes to positive infinity, y predicted will become 1, and if the curve goes to negative infinity, y predicted will become 0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it as 0 or NO. For example, if the output is 0.75, we can say in terms of the probability that there is a 75 percent chance that a patient will suffer from cancer.

$$f(x) = \frac{1}{1 + e^{-x}}$$





python

ML | Types of Logistic Regression

Types of Logistic Regression:

•**Binary Logistic Regression**: The target variable has only two possible outcomes such as Spam or Not Spam, Cancer or No Cancer.

•**Multinomial Logistic Regression**: The target variable has three or more nominal categories, such as predicting the type of Wine.

•**Ordinal Logistic Regression**: the target variable has three or more ordinal categories, such as restaurant or product rating from 1 to 5.

ML | Logistic Regression - Exercises

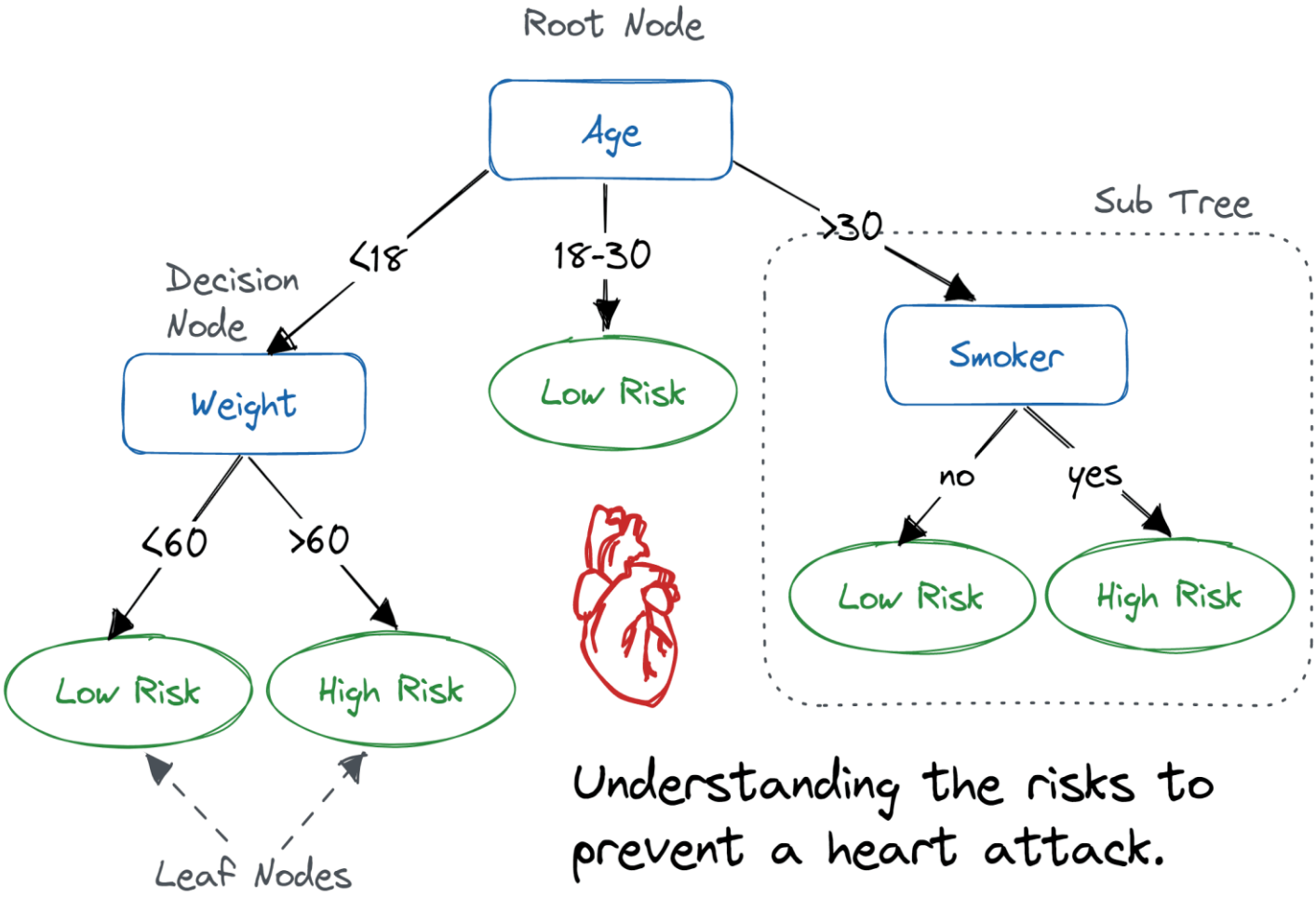
See code here: https://github.com/ShahzadSarwar10/FULLSTACK-WITH-AI-BOOTCAMP-B1-MonToFri-2.5Month-Explorer/blob/main/Week4/Case4-5-LogisticRegressionViaSciKitLearn.py

You should be able to analyze – each code statement, you should be able to see trace information – at each step of debugging. "DEBUGGING IS BEST STRATEGY TO LEARN A LANAGUAGE." So debug code files, line by line, analyze the values of variable – changing at each code statement. BEST STRATEGY TO LEARN DEEP.

Let's put best efforts.
Thanks.
Shahzad – Your AI – ML Instructor

ML | Decision Tree Algorithm

A decision tree is a flowchart-like tree structure where an internal node represents a feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome.
The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in a recursive manner called recursive partitioning. This flowchart-like structure helps you in decision-making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.



ML | How Does the Decision Tree Algorithm Work?

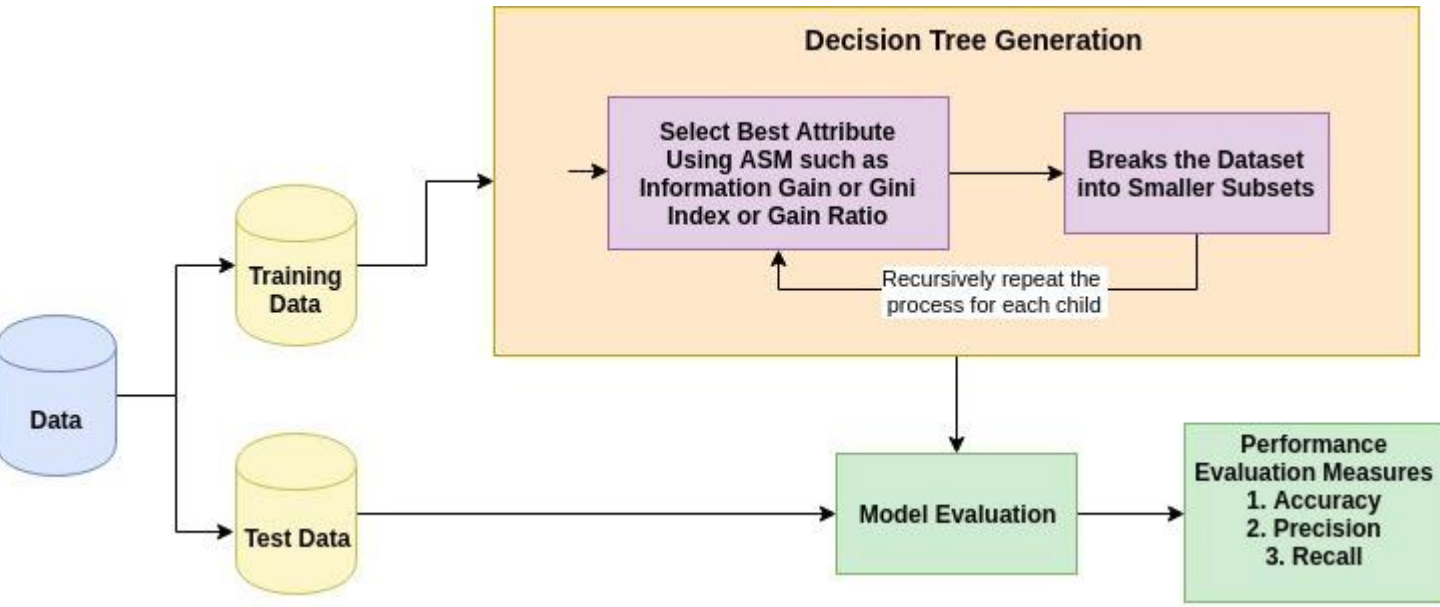
The basic idea behind any decision tree algorithm is as follows:

1.Select the best attribute using Attribute Selection Measures (ASM) to split the records.

2.Make that attribute a decision node and breaks the dataset into smaller subsets.

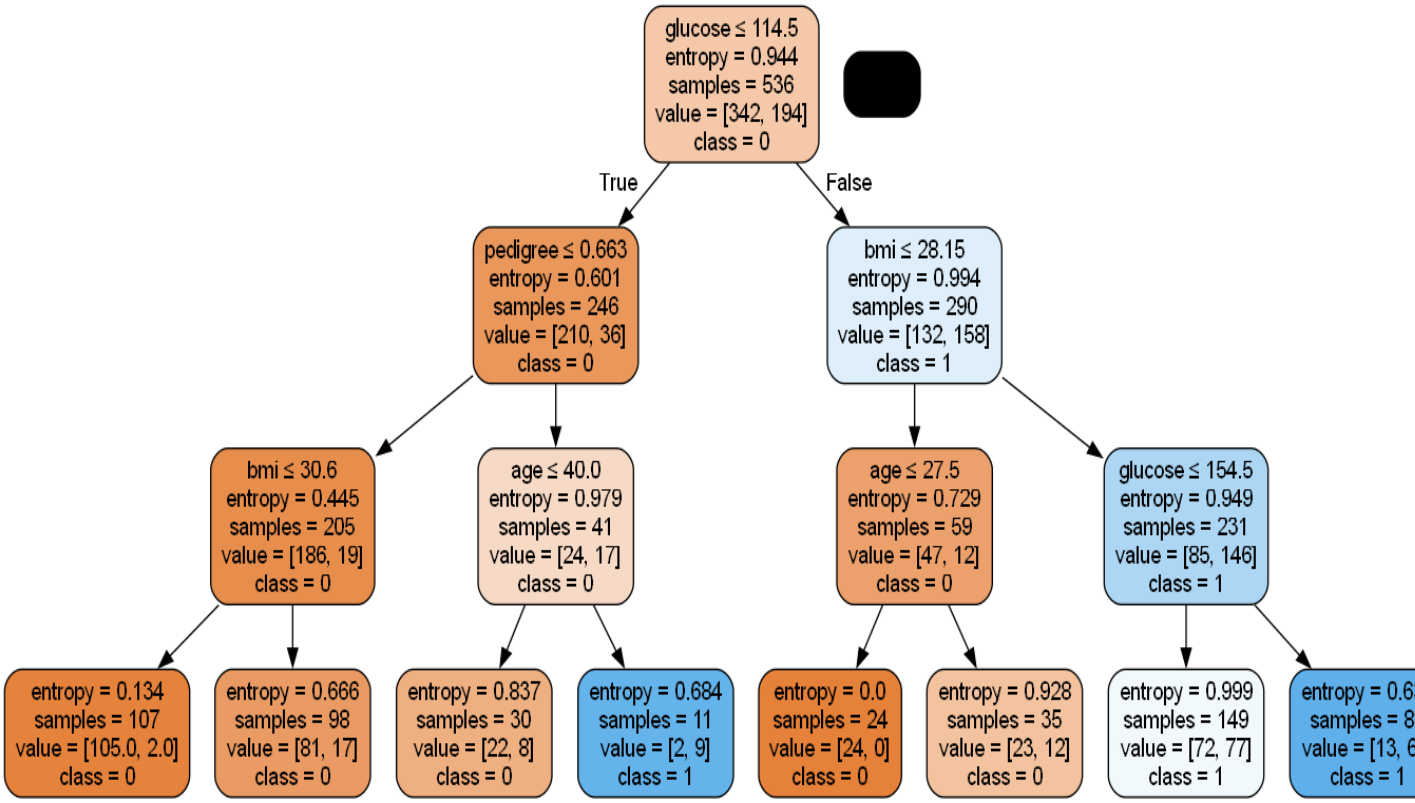
3.Start tree building by repeating this process recursively for each child until one of the conditions will match:

- All the tuples belong to the same attribute value.
- There are no more remaining attributes.
- There are no more instances.



python

ML | Decision Tree Algorithm–Understanding Output



Reference:

https://www.datacamp.com/tutorial/decision-tree-classification-python

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

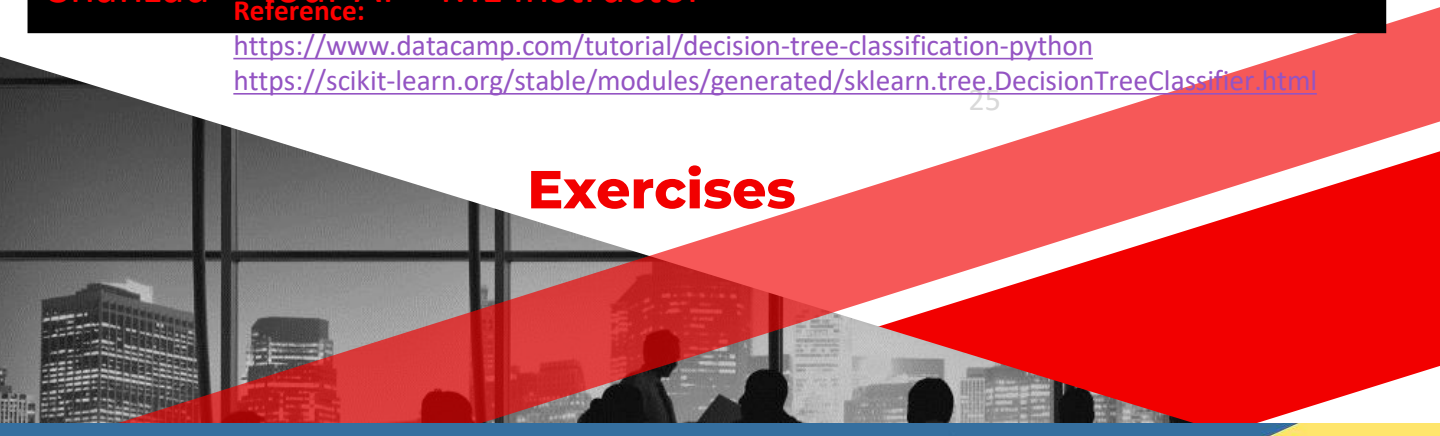
Decision Tree Algorithm - Exercises

See code here: https://github.com/ShahzadSarwar10/FULLSTACK-WITH-AI-BOOTCAMP-B1-MonToFri-2.5Month-Explorer/blob/main/Week4/Case4-6-DecisionTreeClassificationViaSciKitLearn.py

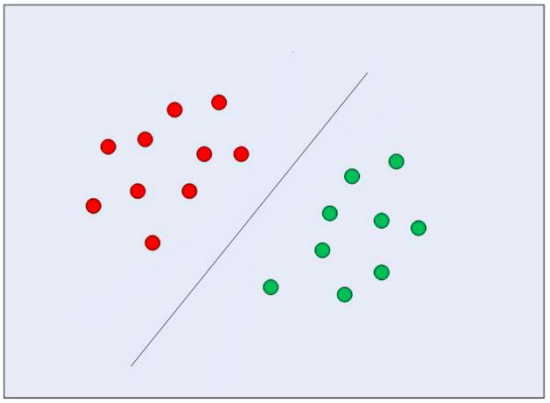
You should be able to analyze – each code statement, you should be able to see trace information – at each step of debugging. "DEBUGGING IS BEST STRATEGY TO LEARN A LANAGUAGE." So debug code files, line by line, analyze the values of variable – changing at each code statement. BEST STRATEGY TO LEARN DEEP.

Let's put best efforts.
Thanks.
Shahzad – Your AI – ML Instructor

Reference:
https://www.datacamp.com/tutorial/decision-tree-classification-python
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html



ML | Support Vector Machine (SVM)



SVMs were introduced initially in 1968, by Vladmir Vapnik and Alexey Chervonenkis. At that time, their algorithm was limited to the classification of data that could be separated using just one straight line, or data that was linearly separable. In the above image we have a line in the middle, to which some points are to the left, and others are to the right of that line. Notice that both groups of points are perfectly separated, there are no points in between or even close to the line. There seems to be a margin between similar points and the line that divides them, that margin is called separation margin. The function of the separation margin is to make the space between the similar points and the line that divides them bigger. SVM does that by using some points and calculates its perpendicular vectors to support the decision for the line's margin. Those are the support vectors that are part of the name of the algorithm. We will understand more about them later. And the straight line that we see in the middle is found by methods that maximize that space between the line and the points, or that maximize the separation margin. Those methods originate from the field of Optimization Theory.

Reference:

https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python
https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/

Support Vector Machine (SVM) - Exercises

See code here: https://github.com/ShahzadSarwar10/FULLSTACK-WITH-AI-BOOTCAMP-B1-MonToFri-2.5Month-Explorer/blob/main/Week4/Case4-7-SVM-ViaSciKitLearn.py

You should be able to analyze – each code statement, you should be able to see trace information – at each step of debugging. "DEBUGGING IS BEST STRATEGY TO LEARN A LANAGUAGE." So debug code files, line by line, analyze the values of variable – changing at each code statement. BEST STRATEGY TO LEARN DEEP.

Let's put best efforts.
Thanks.
Shahzad – Your AI – ML Instructor

Thank you - for listening and participating

- ❑**Questions / Queries**
- ❑**Suggestions/Recommendation**
- ❑**Ideas…..?**

Shahzad Sarwar
Cognitive Convergence
https://cognitiveconvergence.com
shahzad@cognitiveconvergence.com
voice: +1 4242530744 (USA) +92-3004762901 (Pak)