

Project Number - 13

19CSE305 - REVIEW 2

GLASS IDENTIFICATION



Sn. No	Roll Number	Name
1	CB.EN.U4CSE20611	CHOWTA YASHASVI
2	CB.EN.U4CSE20631	K SERATH CHANDRA
3	CB.EN.U4CSE20632	K GANGA ABHIRUP
4	CB.EN.U4CSE20645	ROSHINI PODA
5	CB.EN.U4CSE20660	A V SREE KEERTHI

Contents

- Problem Statement
- List of models Implemented
- Describing models
- Conclusion

Problem Statement

The project's goal is to recognise the glass using its characteristics. A criminological investigation served as the inspiration for this glass categorization project. The glass that was left at the crime scene may be collected as an evidence, analysed using this machine learning model based on its attributes, and categorised into classes. We have 214 instances of the glasses being categorised into these 7 categories in our model. These glasses have properties like the element they are formed of and their refractive index. We categorise using these features and predict the type using the relevant machine learning technique.

List of Models Implemented

1. Naive Bayes classifier
2. Support Vector Machine classifier
3. Random forest classifier
4. XGBoost classifier
5. KNN
6. Decision Tree
7. Neural Network Classifier

Naive Bayes classifier

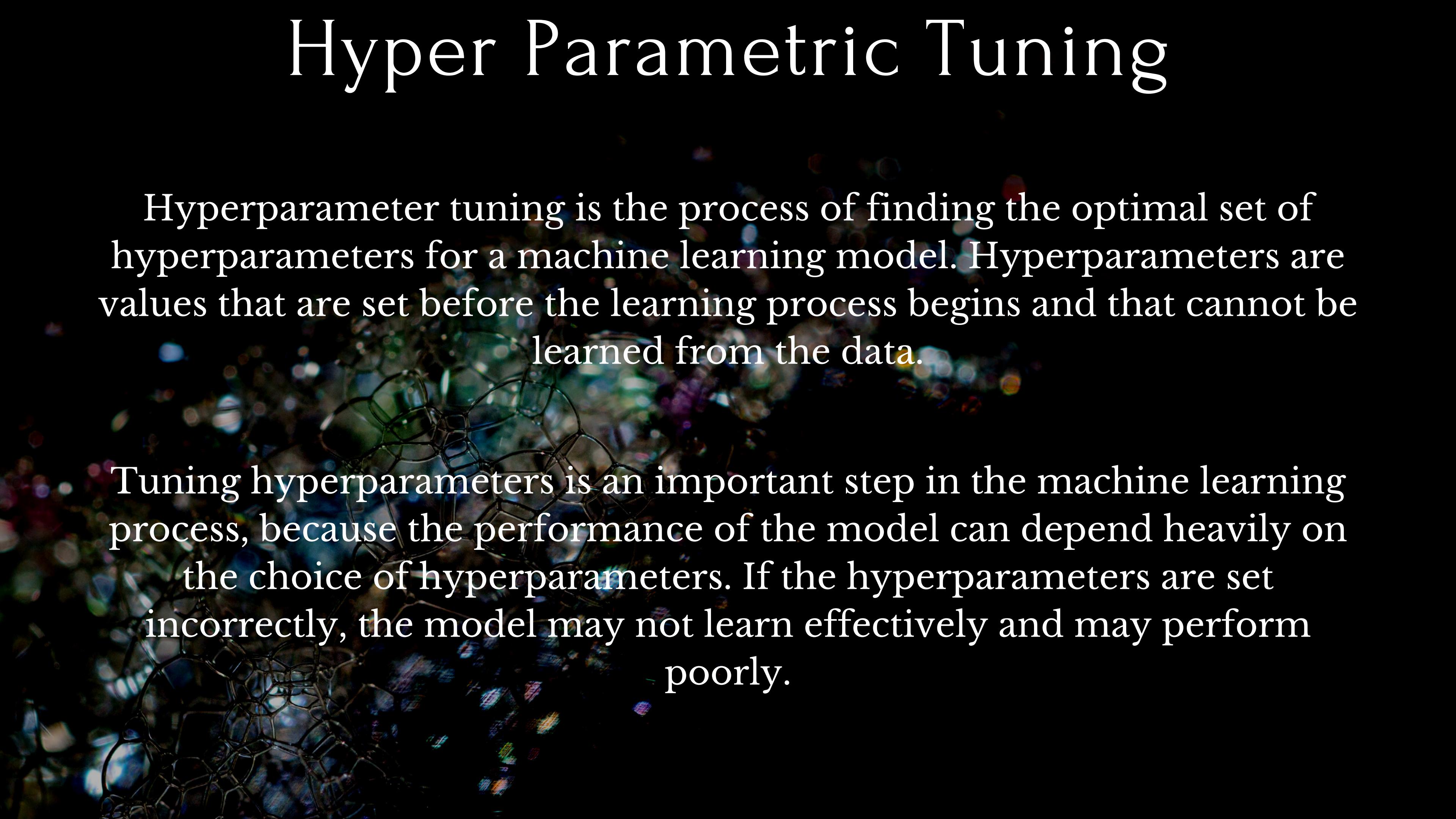
Naive Bayes classifier is a machine learning algorithm that uses Bayes' theorem to predict the probability of an event, based on prior knowledge of conditions that might be related to the event. It is called "naive" because it makes the assumption that all the features in the dataset are independent of each other, which is not always true in real-world data. However, this assumption helps to simplify the calculations and makes the algorithm relatively fast and easy to implement.

There are several types of naive Bayes classifiers, including the Gaussian naive Bayes, Bernoulli naive Bayes, and Multinomial naive Bayes. The choice of which type to use depends on the nature of the data and the specific problem you are trying to solve.

Classification Metrics of Naïve Bayes Classifier

Name	Gaussian	Bernoulli
0 Accuracy	0.534884	0.465116
1 R-Squared	0.728435	0.523105
2 MAE	0.627907	0.837209
3 MSE	0.953488	1.674419
4 RMSE	0.976467	1.293993

Hyper Parametric Tuning



Hyperparameter tuning is the process of finding the optimal set of hyperparameters for a machine learning model. Hyperparameters are values that are set before the learning process begins and that cannot be learned from the data.

Tuning hyperparameters is an important step in the machine learning process, because the performance of the model can depend heavily on the choice of hyperparameters. If the hyperparameters are set incorrectly, the model may not learn effectively and may perform poorly.

Support Vector Machine Classifier

A support vector machine (SVM) is a type of supervised machine learning algorithm that can be used for classification or regression tasks. The goal of an SVM is to find the hyperplane in an N-dimensional space (where N is the number of features) that maximally separates the two classes.

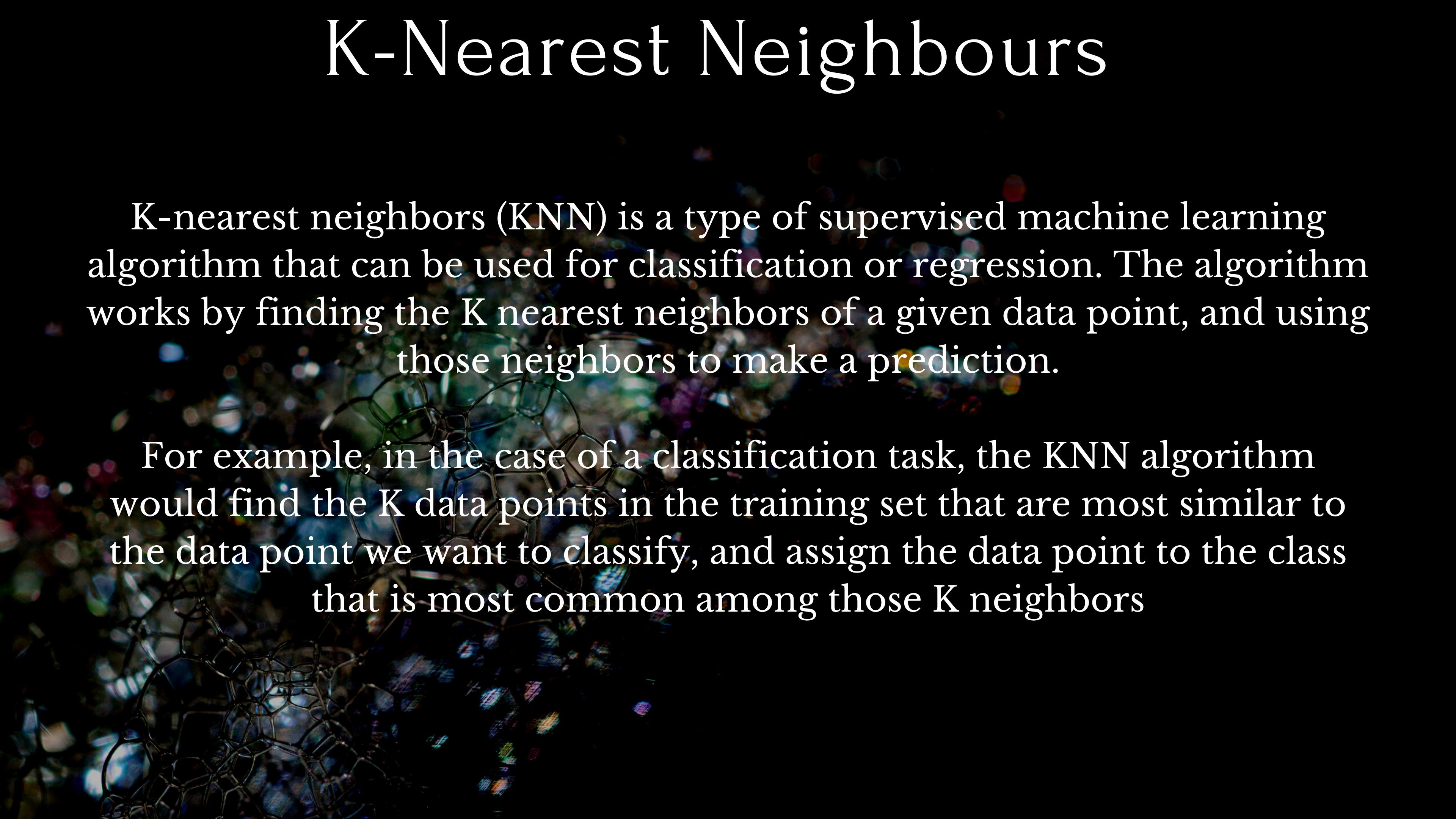
In the case of a classification task, the SVM will find the hyperplane that maximally separates the two classes and classifies new data points based on which side of the hyperplane they fall on.

We implemented both RBF and Polynomial Kernel and observed their accuracies

Classification metrics for SVM Classifier

Name	RBF Kernel	Polynomial Kernel
0 Accuracy	0.348837	0.674419
1 R-Squared	-0.298213	0.675447
2 MAE	1.488372	0.534884
3 MSE	4.558140	1.139535
4 RMSE	2.134980	1.067490

K-Nearest Neighbours



K-nearest neighbors (KNN) is a type of supervised machine learning algorithm that can be used for classification or regression. The algorithm works by finding the K nearest neighbors of a given data point, and using those neighbors to make a prediction.

For example, in the case of a classification task, the KNN algorithm would find the K data points in the training set that are most similar to the data point we want to classify, and assign the data point to the class that is most common among those K neighbors

Accuracy Score for KNN Classifier

0.7209302325581395

Decision Tree

A decision tree is a flowchart-like tree structure used to make a decision or a prediction based on certain conditions. It is a popular tool in machine learning and data mining, and is often used in fields such as finance, healthcare, and marketing.

Each internal node in the tree represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The topmost node in a decision tree is known as the root node. It helps us to solve the problem, by breaking it down into smaller and simpler sub-problems.

Accuracy Score for Decision Tree Classifier

0.7209302325581395

Random Forest Classifier

A random forest classifier is a type of machine learning algorithm that is used for classification tasks. It is an ensemble method, meaning that it combines the predictions of multiple decision trees to make a final prediction. Each decision tree in a random forest classifier is trained on a random subset of the data, and the final prediction is made by averaging the predictions of all the trees.

A hyperparameter-tuned random forest classifier is a random forest classifier that has been trained with a set of hyperparameters that have been optimized for a specific task.

In the case of a random forest classifier, some common hyperparameters that you might tune include the number of decision trees in the forest, the maximum depth of each tree, and the minimum number of samples required to split a node in the tree.

Classification Metrics of Random forest classifier

	Name	RF Classifier	Hyperparameter Tuned RF Classifier
0	Accuracy	0.813953	0.837209
1	R-Squared	0.735059	0.781423
2	MAE	0.372093	0.302326
3	MSE	0.930233	0.767442
4	RMSE	0.964486	0.876038

XG Boost Classifier

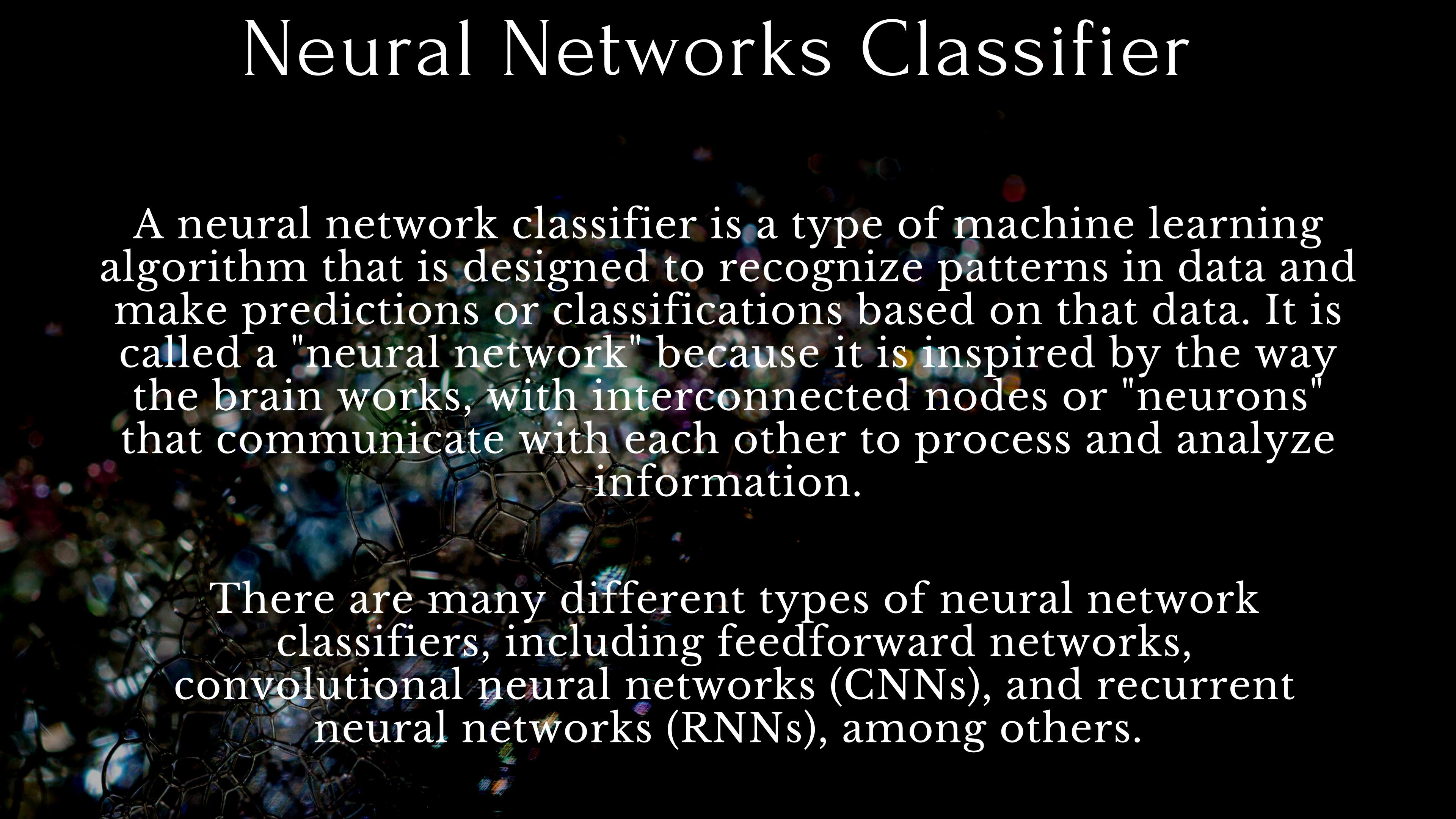
XGBoost (eXtreme Gradient Boosting) is a popular and powerful gradient boosting library for machine learning that was developed by Tianqi Chen. Like a random forest classifier, XGBoost is an ensemble method that combines the predictions of multiple decision trees to make a final prediction. However, unlike a random forest, which builds each tree independently, XGBoost trees are trained sequentially, with each tree learning from the mistakes of the previous tree. This makes XGBoost particularly powerful for tasks where the relationship between features and the target variable is complex.

In the case of an XGBoost classifier, some common hyperparameters that you might tune include the learning rate, the maximum depth of each tree, the number of trees in the ensemble, and the type of loss function to use.

Classification Metrics for XGBoost Classifier

	Name	XGBoost	Hyperparameter Tuned XGBoost
0	Accuracy	0.813953	0.906977
1	R-Squared	0.827788	0.774800
2	MAE	0.279070	0.232558
3	MSE	0.604651	0.790698
4	RMSE	0.777593	0.889212

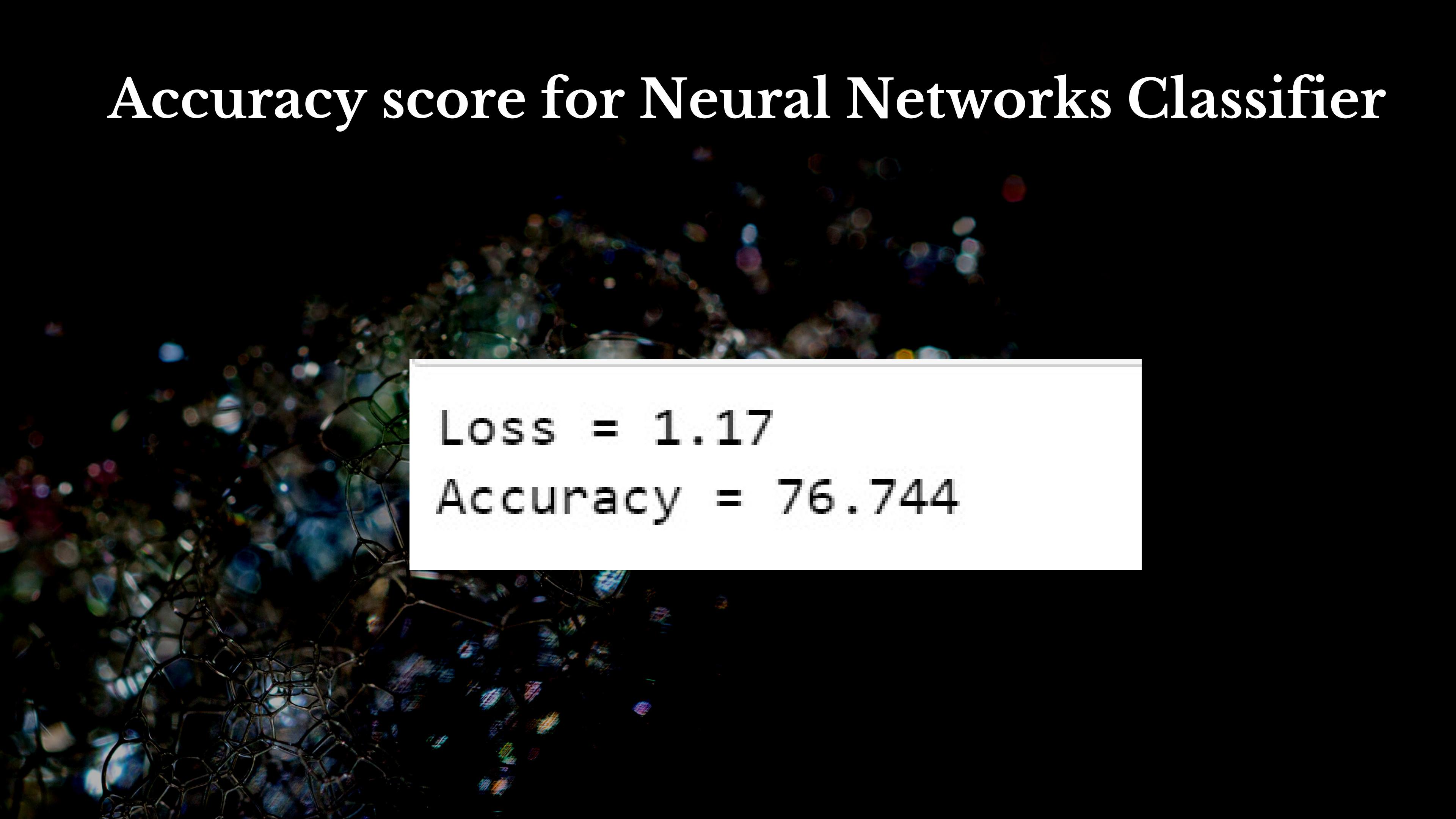
Neural Networks Classifier



A neural network classifier is a type of machine learning algorithm that is designed to recognize patterns in data and make predictions or classifications based on that data. It is called a "neural network" because it is inspired by the way the brain works, with interconnected nodes or "neurons" that communicate with each other to process and analyze information.

There are many different types of neural network classifiers, including feedforward networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), among others.

Accuracy score for Neural Networks Classifier



Loss = 1.17
Accuracy = 76.744

Thank
You