

PROBLEM, WHEN THE CLASS 0 IS MAJORITY

PROBLEM:

In the Stroke Prediction dataset from Kaggle, the share of class 1 is ~5%.

KNN always looks for the closest points.

Because class 0 is the majority, the neighborhood is often all zeros → the model predicts 0.

OVERSAMPLING SLOT

```
from imblearn.over_sampling import SMOTE  
from collections import Counter
```

```
sm = SMOTE(random_state=42)  
X_res, y_res = sm.fit_resample(X, y)
```

```
print(sorted(Counter(y_res).items()))
```

DATA NORMALISATION

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)
```

ADJUST N_NEIGHBORS PARAMETR

```
knn = KNeighborsClassifier(n_neighbors=3)
```

MACHINE LEARNING

Final Project Instructions

- **Create INTERNATIONAL teams** (students from different countries where possible).
- **Team Size:** 2 to 4 students.
- **Choose a Team Name.**
- **Prepare a short introduction slide** about your team (team members, countries, something interesting about you).

Project Task

Each team must prepare a **short presentation** about the following **Machine Learning algorithms**:

k-Means Clustering

Naive Bayes

**Principal Component
Analysis (PCA)**

Decision Tree

**k-Nearest Neighbors
(k-NN)**

Your presentation should briefly cover for each algorithm:

- ❑ Definition and main idea
- ❑ How it works (basic steps)
- ❑ Typical use cases
- ❑ Advantages and disadvantages

UCI Machine Learning Repository: The UCI repository hosts a wide range of datasets suitable for machine learning tasks, including datasets related to medicine. You can explore their collection and find datasets that fit your requirements. Here's the link: [UCI Machine Learning Repository](#)

Kaggle Datasets: Kaggle is a platform for data science competitions, and it also hosts a large collection of datasets contributed by the community. You can search for medical datasets on Kaggle and find ones suitable for building decision tree models. Here's the link: [Kaggle Datasets](#)

OpenML: OpenML is an online platform where you can find datasets, machine learning tasks, and experiments. It hosts a variety of datasets from different domains, including medicine. You can explore their collection and find datasets suitable for your project. Here's the link: [OpenML](#)

Decision Tree

sklearn.tree.DecisionTreeClassifier

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

Install User Guide API Examples Community More

Prev Up Next

scikit-learn 1.4.1
Other versions

Please cite us if you use the software.

sklearn.tree.DecisionTreeClassifier

DecisionTreeClassifier

Examples using

sklearn.tree.DecisionTreeClassifier

sklearn.tree.DecisionTreeClassifier

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0, monotonic_cst=None)
```

A decision tree classifier.

Read more in the [User Guide](#).

Parameters: **criterion** : {"gini", "entropy", "log_loss"}, default="gini"
The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see [Mathematical formulation](#).

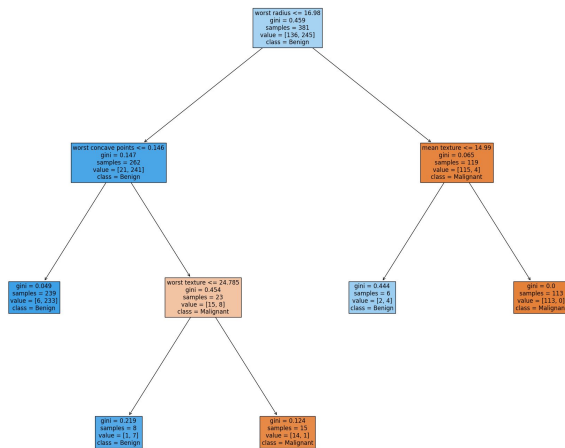
splitter : {"best", "random"}, default="best"
The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

max_depth : int, default=None
The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

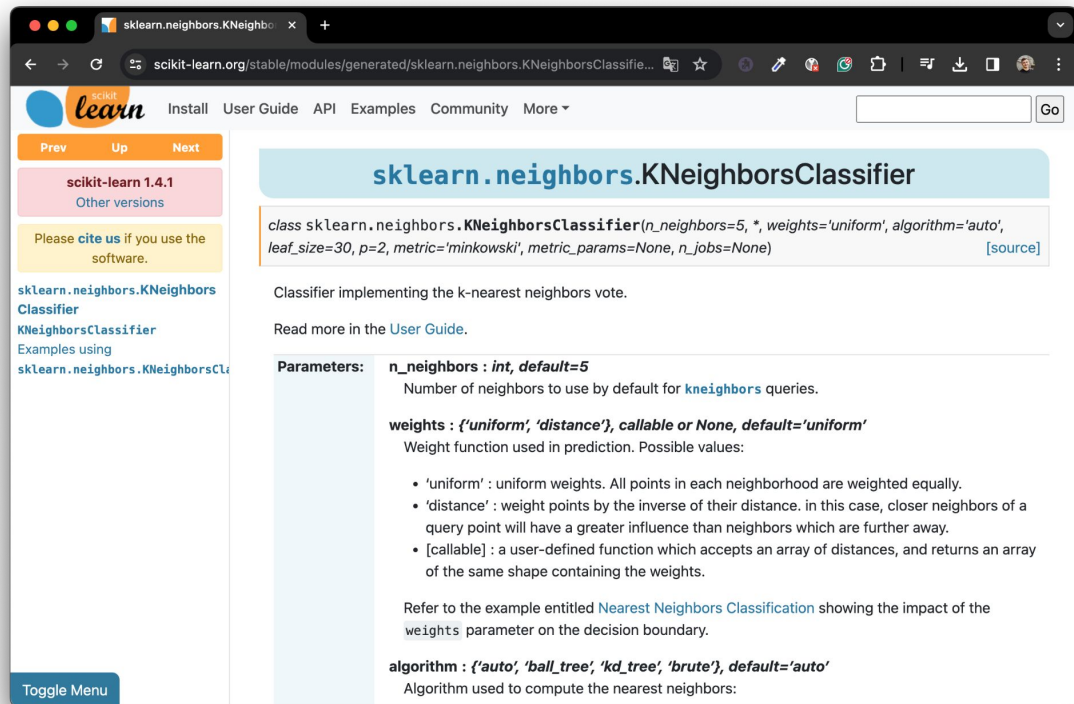
min_samples_split : int or float, default=2
The minimum number of samples required to split an internal node:

- If int, then consider min_samples_split as the minimum number.
- If float, then min_samples_split is a fraction and ceil(min_samples_split * n_samples) are the minimum number of samples for each split.

Changed in version 0.18: Added float values for fractions.



K-Nearest Neighbors (KNN)

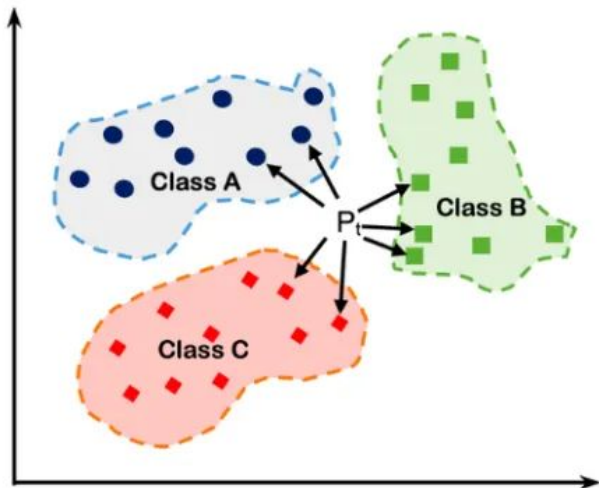


The screenshot shows the scikit-learn documentation for `KNeighborsClassifier`. The page includes the scikit-learn logo, navigation links (Install, User Guide, API, Examples, Community, More), and a sidebar with links to other versions and examples. The main content area displays the class name `sklearn.neighbors.KNeighborsClassifier` and its signature: `class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None)`. Below the signature, it states: "Classifier implementing the k-nearest neighbors vote." and "Read more in the [User Guide](#)." The "Parameters:" section lists:

- `n_neighbors`: `int, default=5`. Number of neighbors to use by default for `kneighbors` queries.
- `weights`: `{'uniform', 'distance'}, callable or None, default='uniform'`. Weight function used in prediction. Possible values:
 - 'uniform': uniform weights. All points in each neighborhood are weighted equally.
 - 'distance': weight points by the inverse of their distance. In this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
 - [callable]: a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

It also refers to an example titled "Nearest Neighbors Classification" showing the impact of the `weights` parameter on the decision boundary. The `algorithm` parameter is listed as: `{'auto', 'ball_tree', 'kd_tree', 'brute'}, default='auto'` with the note "Algorithm used to compute the nearest neighbors:".

K Nearest Neighbors



Submission Requirements

You must submit **three components**:

1. **Presentation** — will be delivered live on **January 8, 2026**, during the computer lab.
2. **Google Colab Notebook (.ipynb file)** — submit a file with your slides or additional explanations if needed.

Project should use medical diagnosis data or security anomalies. Usage of minimum 2 machine learning classifier is required. Please add text description above code, explain how you process data. Test different data divisions.

3. **PDF Report** — submit a short written report (maximum 2-3 pages) summarizing the key points of your presentation.

Important Rules

All team members **must participate** during the final presentation.

Deadline for submission: Before the computer lab on January 8th, 2026.

Submission format: Send your files (IPYNB and KEYNOTE PDF, REPORT PDF) via the platform/email (teacher will provide further details).

Late submissions may lower the final grade!