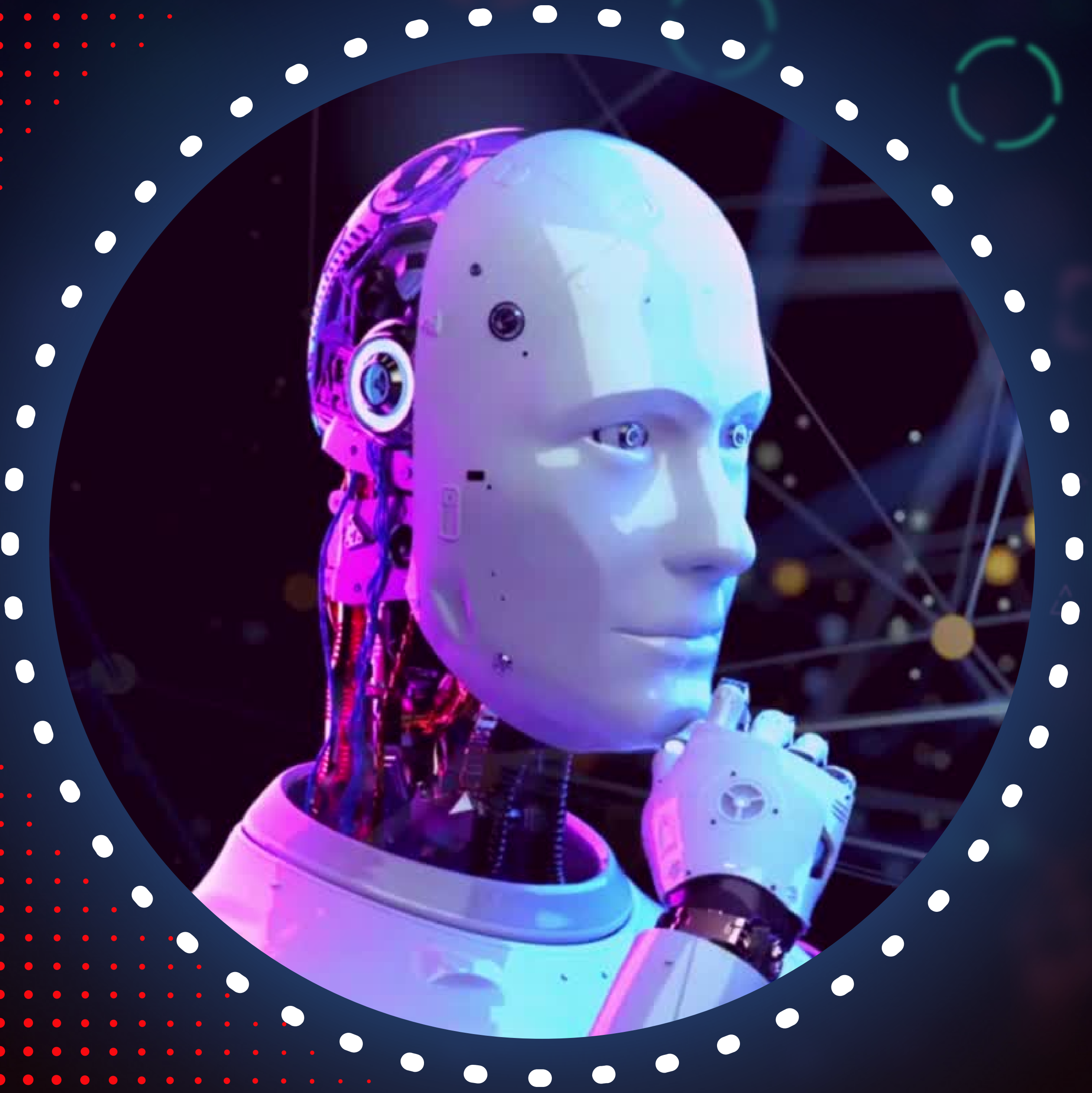
Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

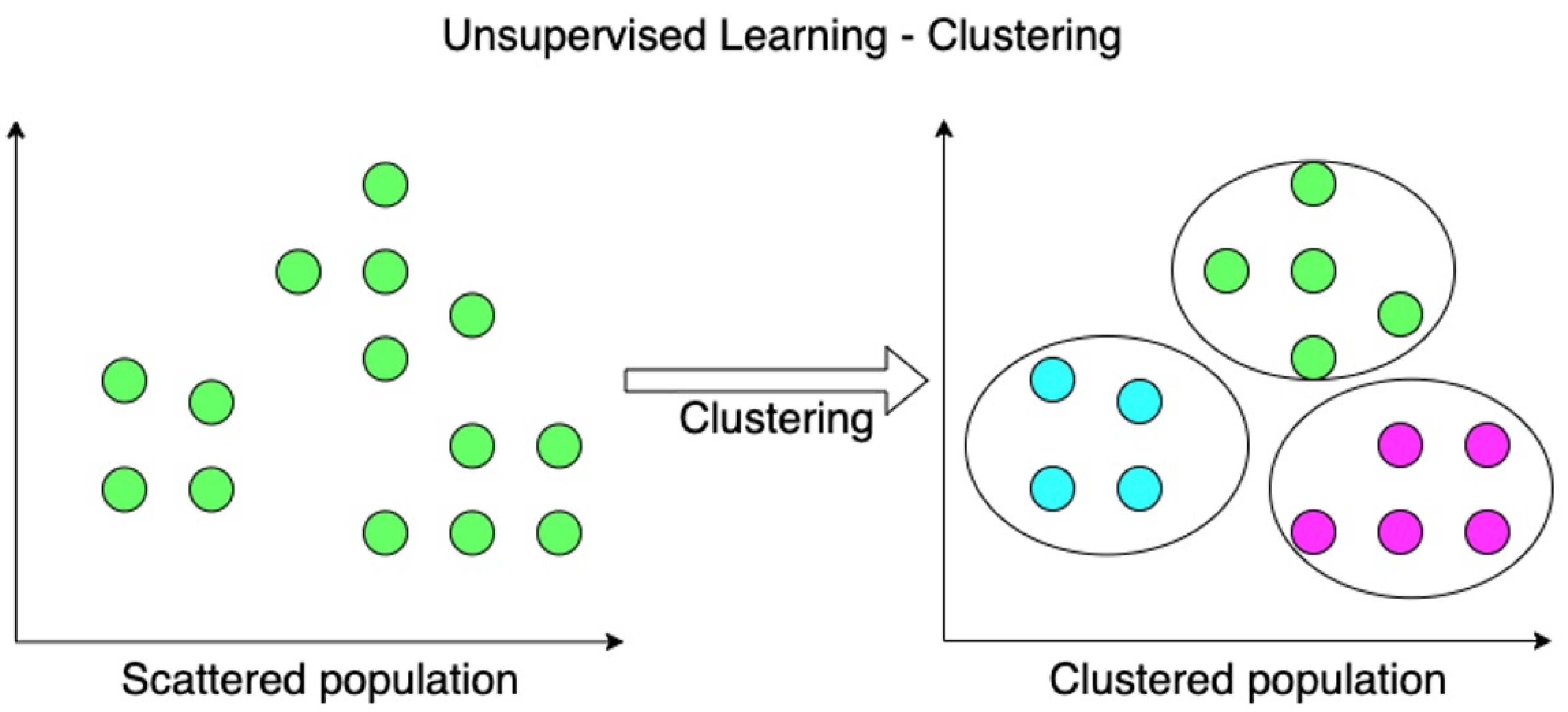


MACHINE LEARNING

DAY – 17

K-MEANS CLUSTERING

Clustering

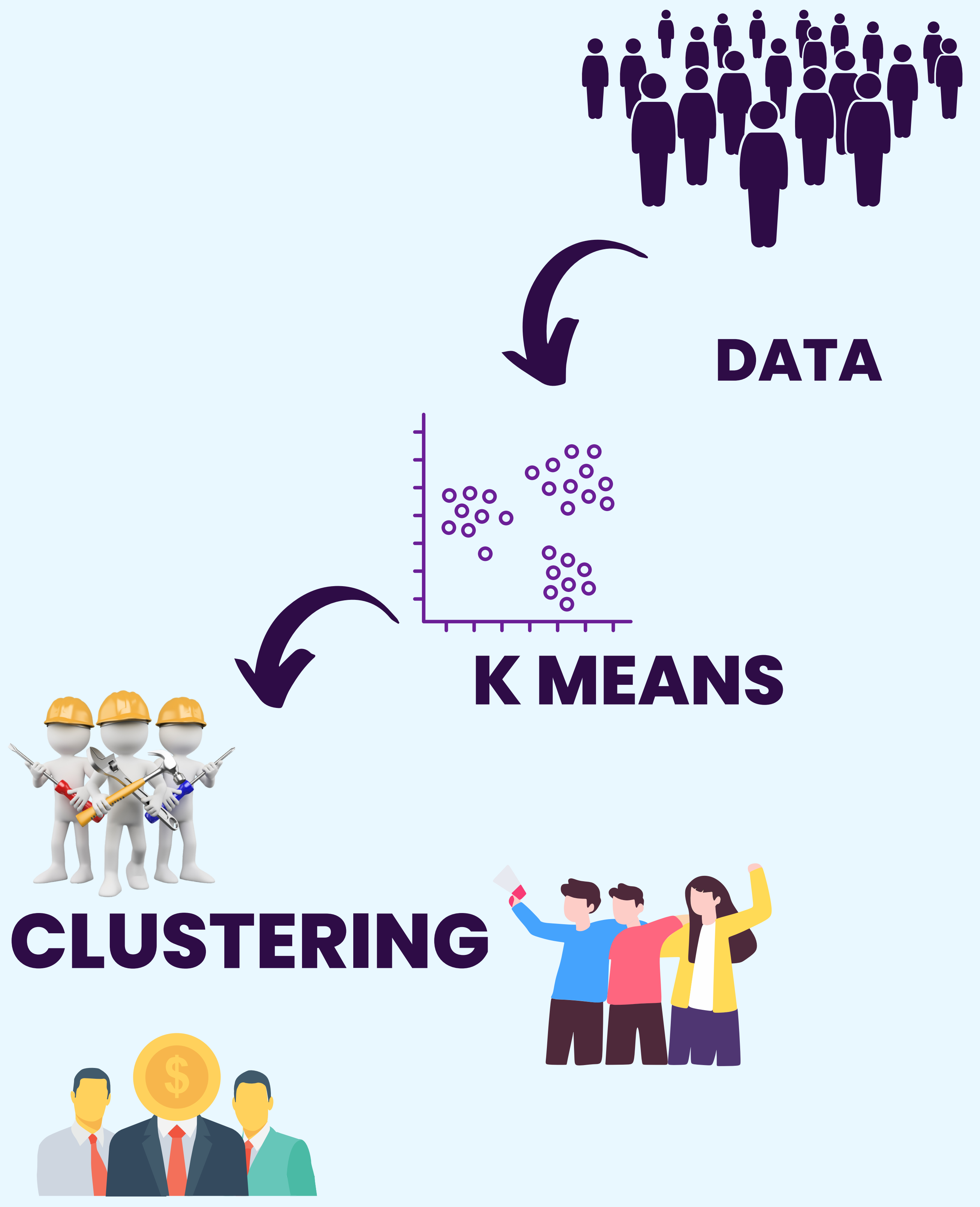


- Clustering is a type of unsupervised machine learning in which the algorithm processes our data and divided them into “clusters”.

Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

k-means clustering

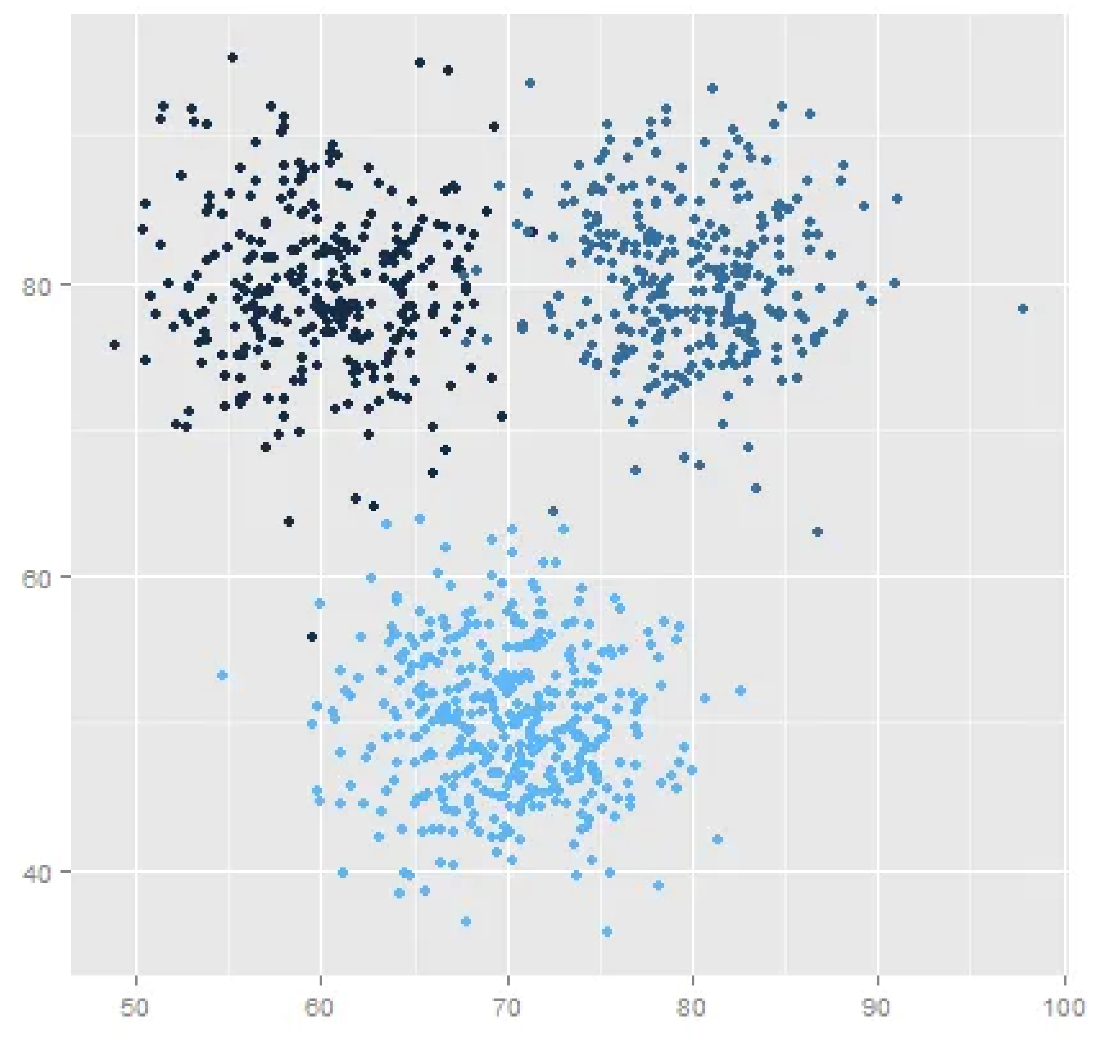
- K means algorithms that can divide the give data into the given number of clusters
- k- number of the clustering that we need to be created
- It is centroid based algorithms in which cluster associated with centroid the main idea is reduce the distance between data points and the respective cluster centroid



k clustering calculation

Within Cluster Sum Of Squares (WCSS) against the the number of clusters (K Value) to figure out the optimal number of clusters value. WCSS measures sum of distances of observations from their cluster centroids

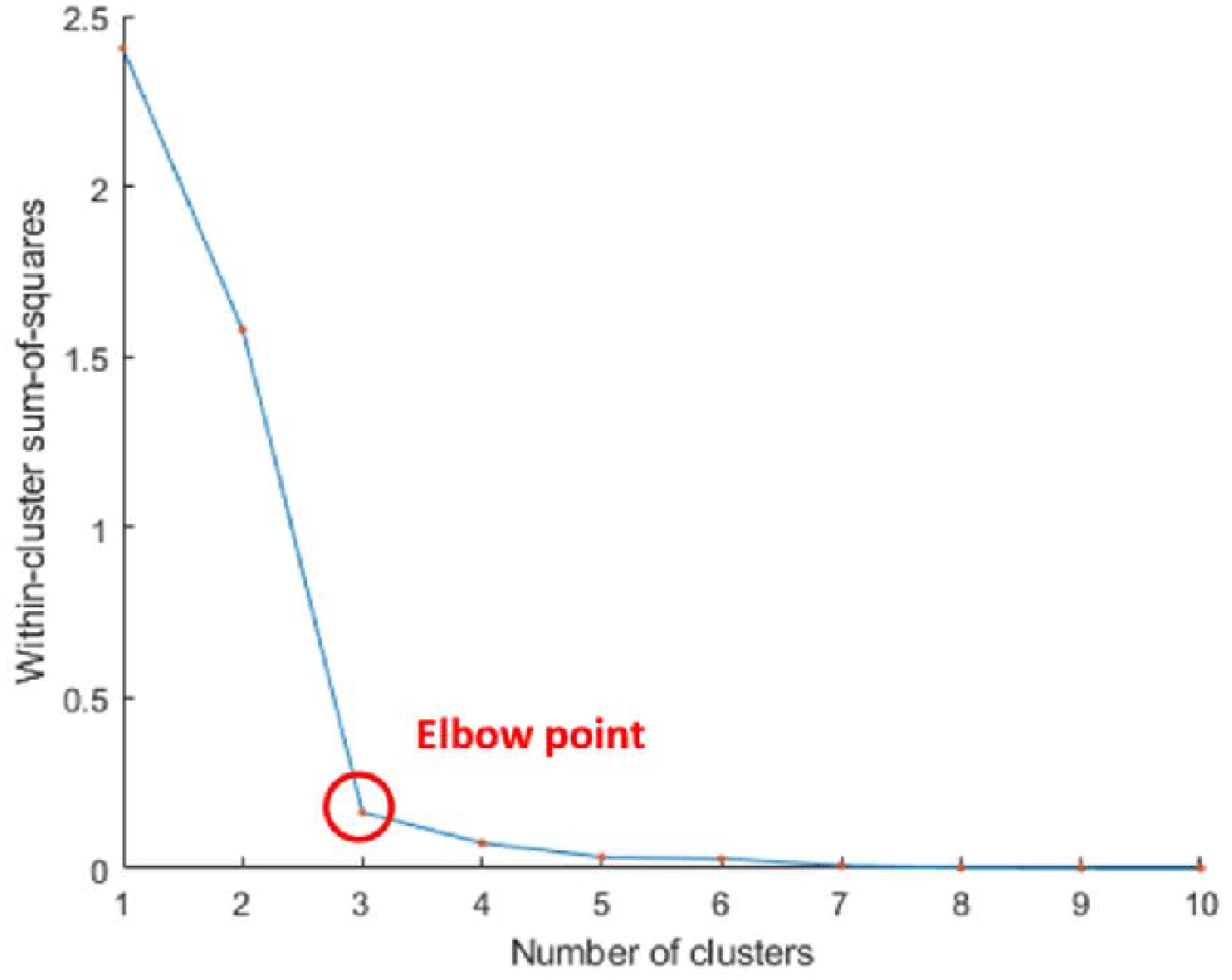
$$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$



Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

Elbow method

- Compute K-Means clustering for different values of K by varying K from 1 to 10 clusters.
- For each K, calculate the total within-cluster sum of square (WCSS).
- Plot the curve of WCSS vs the number of clusters K.
- The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.



HYPERPARAMETER TUNING

Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

Hyperparameter Tuning

- Hyperparameters are different parameter values that are used to control the learning process and have a significant effect on the performance of machine learning models.

- An example of hyperparameters in the Random Forest algorithm is the number of estimators (n_estimators), maximum depth (max_depth), and criterion. These parameters are tunable and can directly affect how well a model trains.

- So then hyperparameter optimization is the process of finding the right combination of hyperparameter values to achieve maximum performance on the data in a reasonable amount of time.

Hyperparameter Tuning

- **GridSearchCV**
- **RandomizedSearchCV**

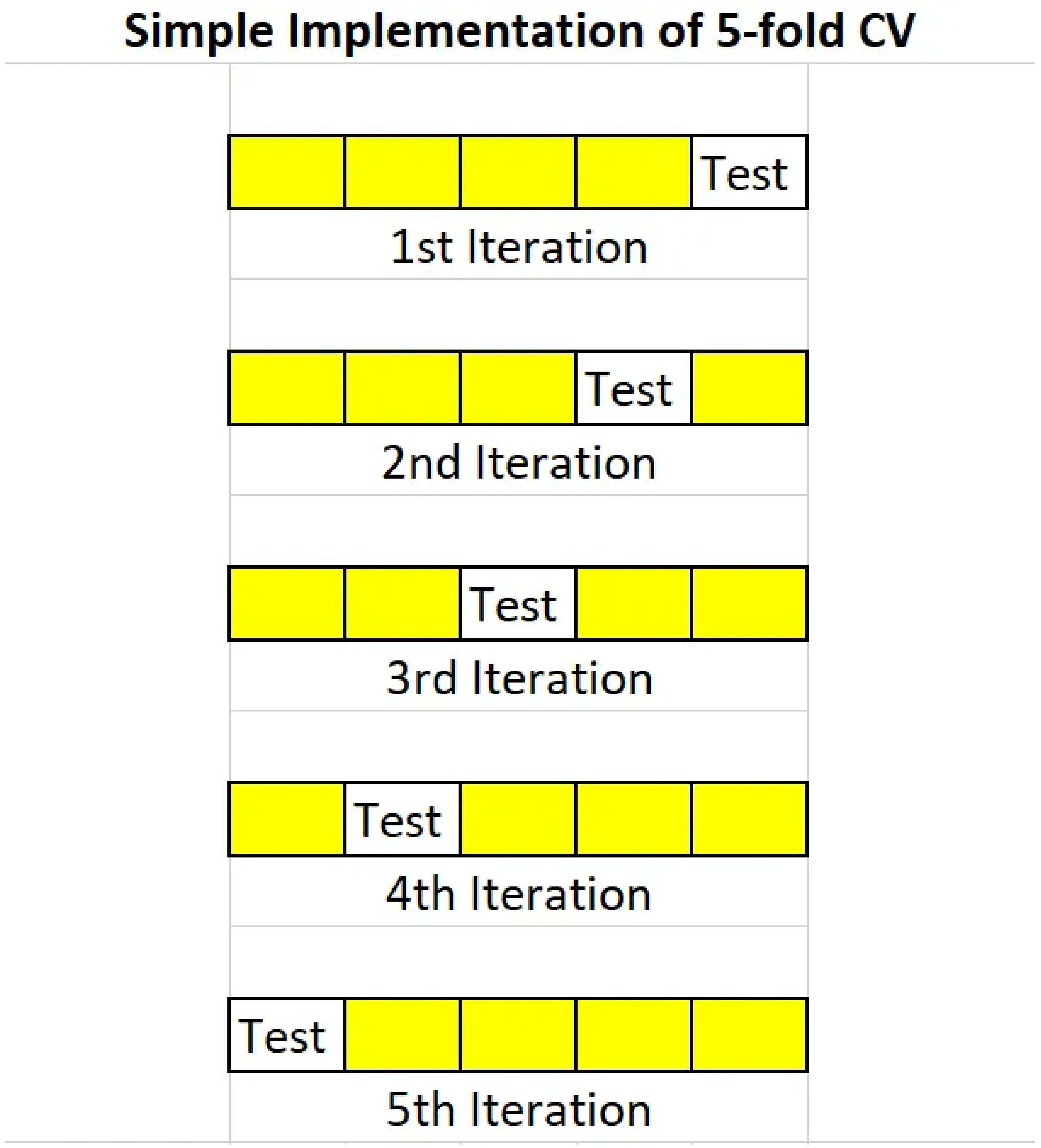
Cross Validation

- Cross-validation is a resampling procedure used to evaluate machine learning models.

- This method has a single parameter k which refers to the number of partitions the given data sample is to be split into.

- So, they are often called k-fold cross-validation. The data is divided into training, validating and testing set to prevent data leaks.

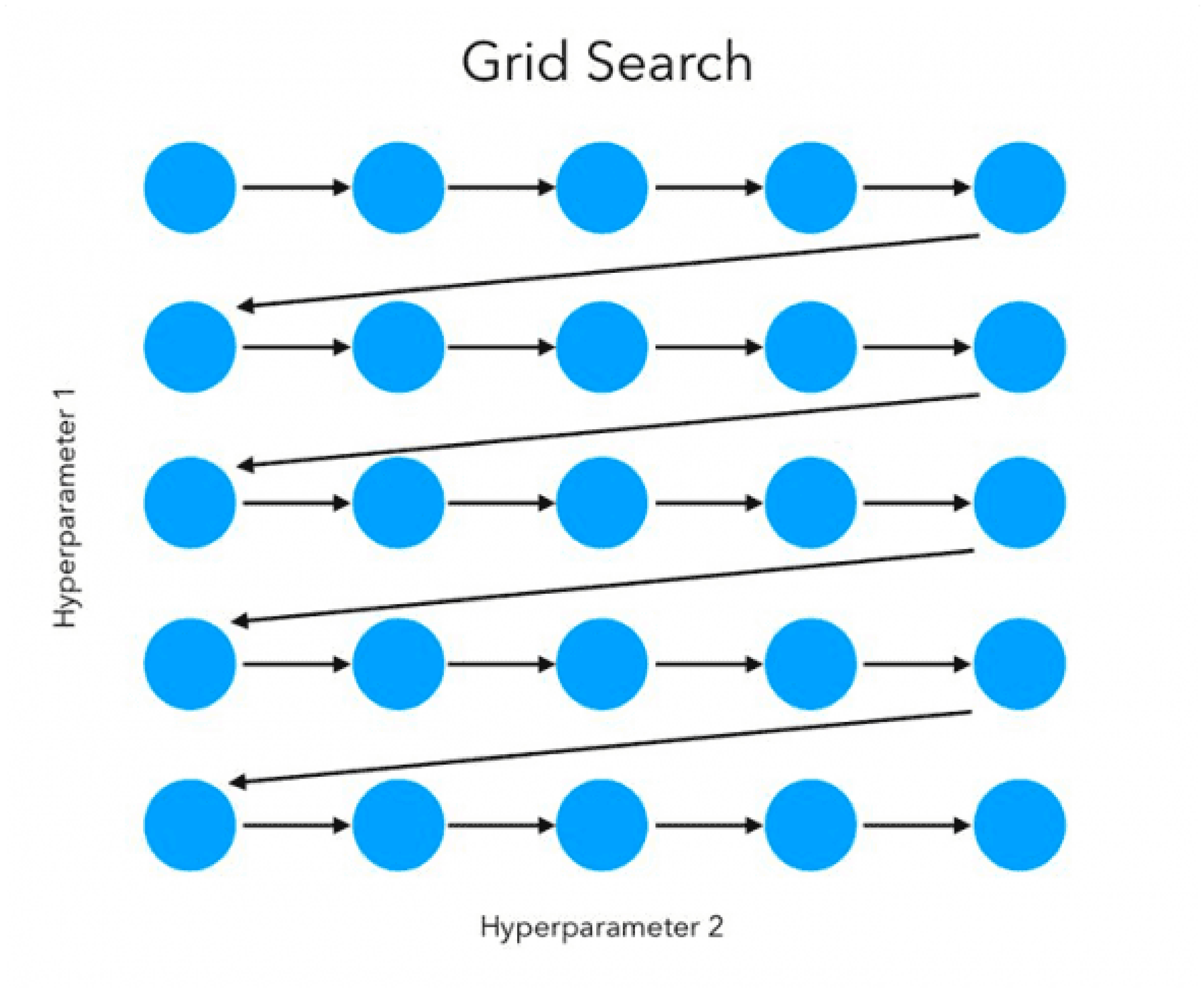
Cross Validation



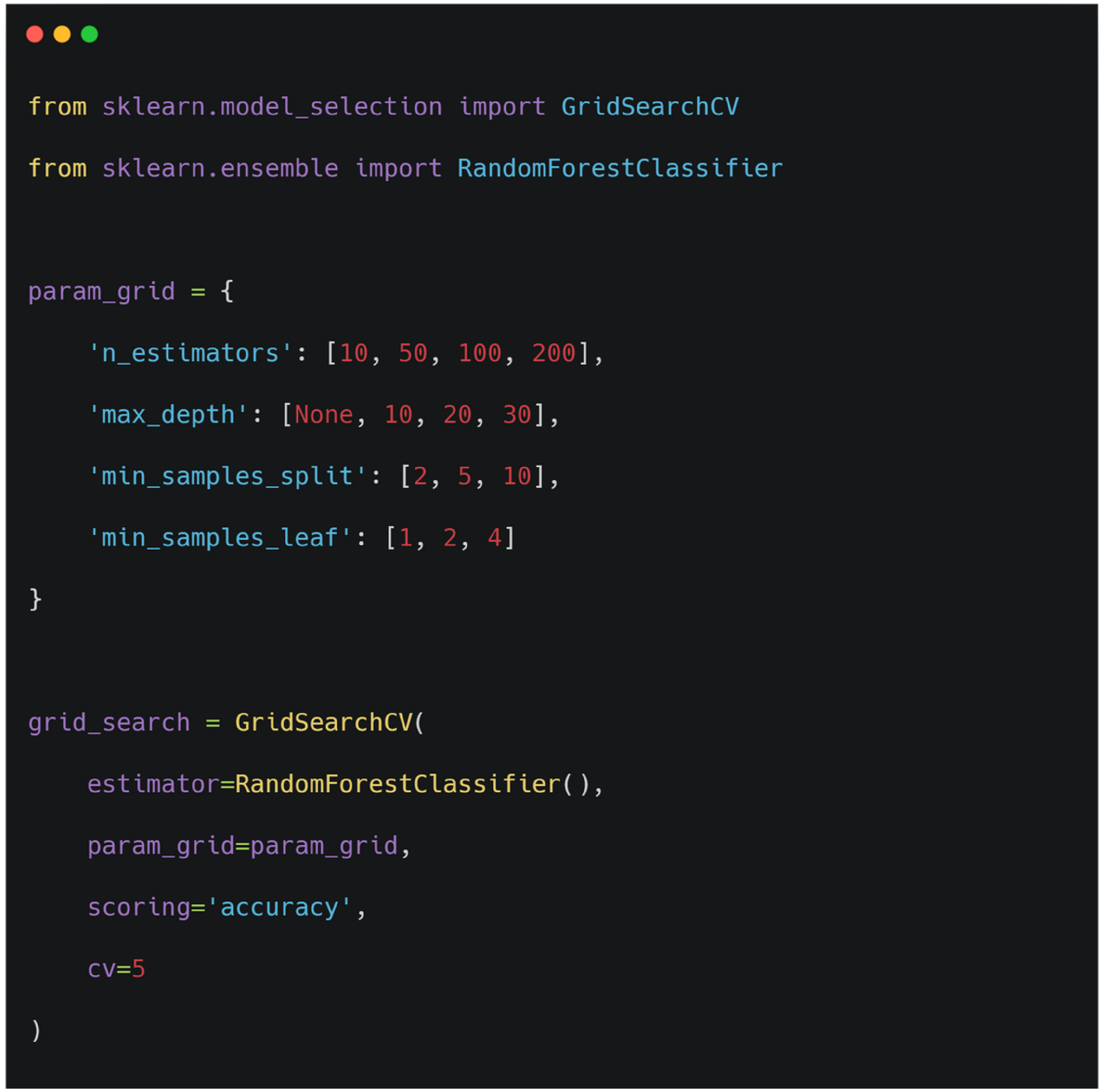
GridSearchCV

- Grid search works by trying every possible combination of parameters you want to try in your model.
- All possible permutations of the hyper parameters for a particular model are used to build models.
- The performance of each model is evaluated and the best performing one is selected.
- Since GridSearchCV uses each and every combination to build and evaluate the model performance, this method is highly computational expensive.

GridSearchCV



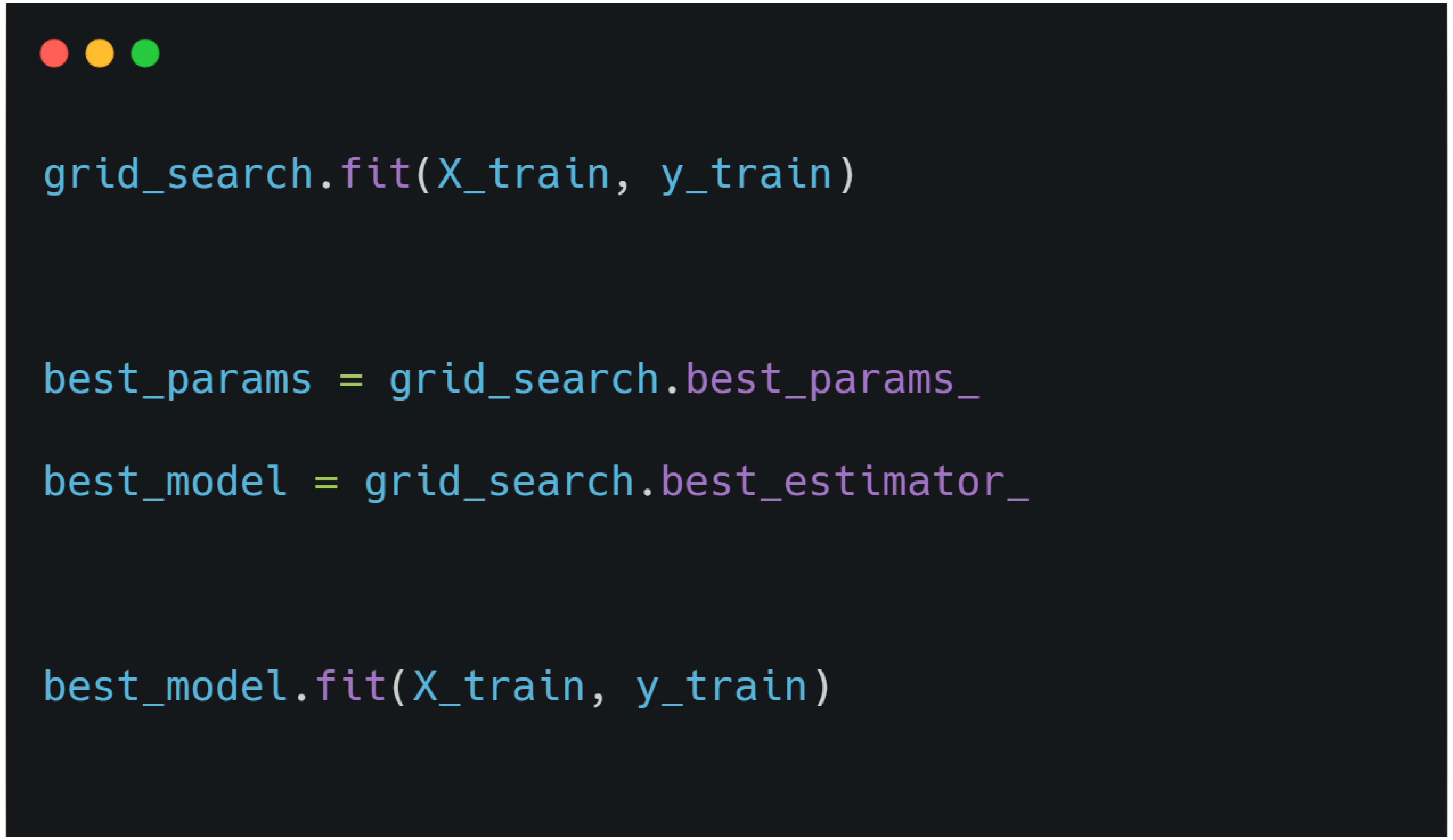
GridSearchCV



```python
from sklearn.model_selection import GridSearchCV

from sklearn.ensemble import RandomForestClassifier


param_grid = {

    'n_estimators': [10, 50, 100, 200],

    'max_depth': [None, 10, 20, 30],

    'min_samples_split': [2, 5, 10],

    'min_samples_leaf': [1, 2, 4]

}


grid_search = GridSearchCV(

    estimator=RandomForestClassifier(),

    param_grid=param_grid,

    scoring='accuracy',

    cv=5

)
```

Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

GridSearchCV



```
grid_search.fit(X_train, y_train)


best_params = grid_search.best_params_

best_model = grid_search.best_estimator_


best_model.fit(X_train, y_train)
```

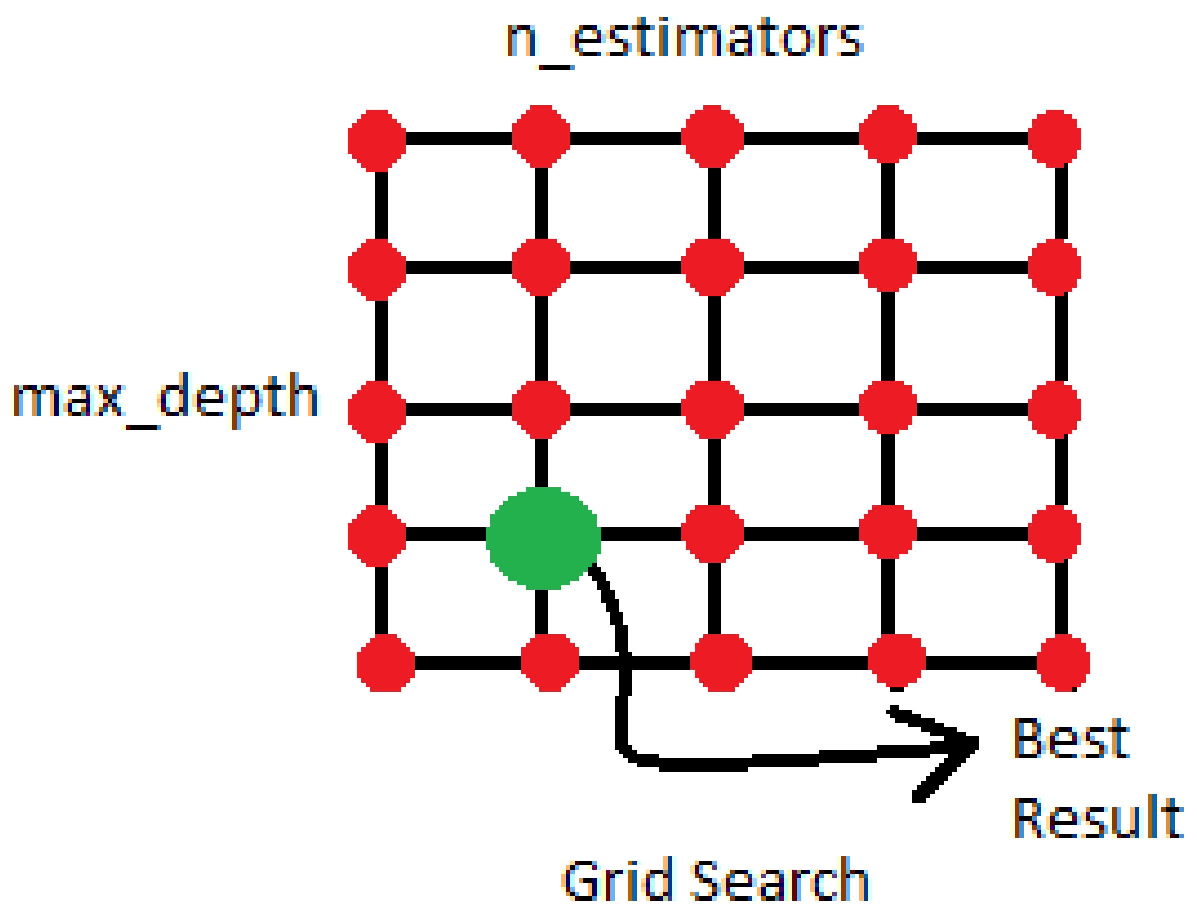
RandomizedSearchCV

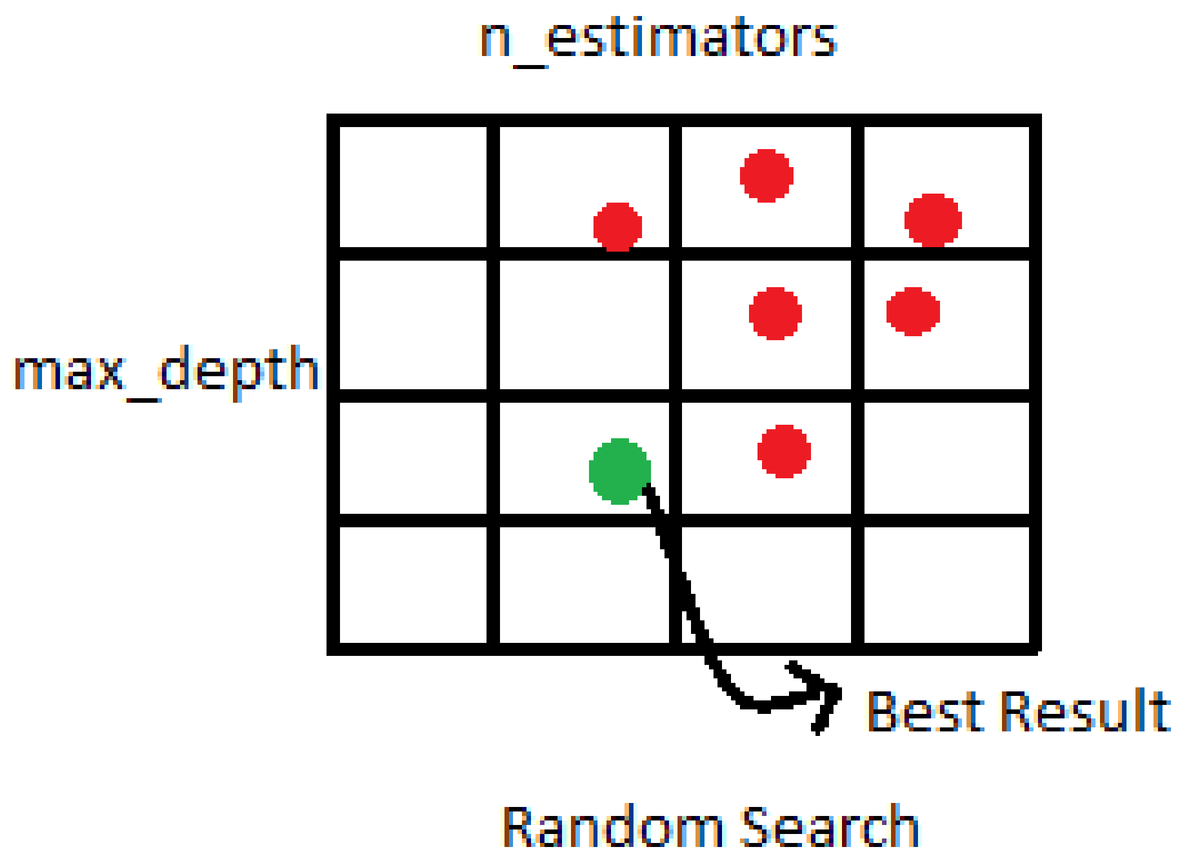
- In RandomizedSearchCV, values for the different hyper parameters are picked up at random from this distribution.

- RandomizedSearchCV is particularly useful when you have limited computational resources or when you want to quickly explore a broad range of hyperparameters.

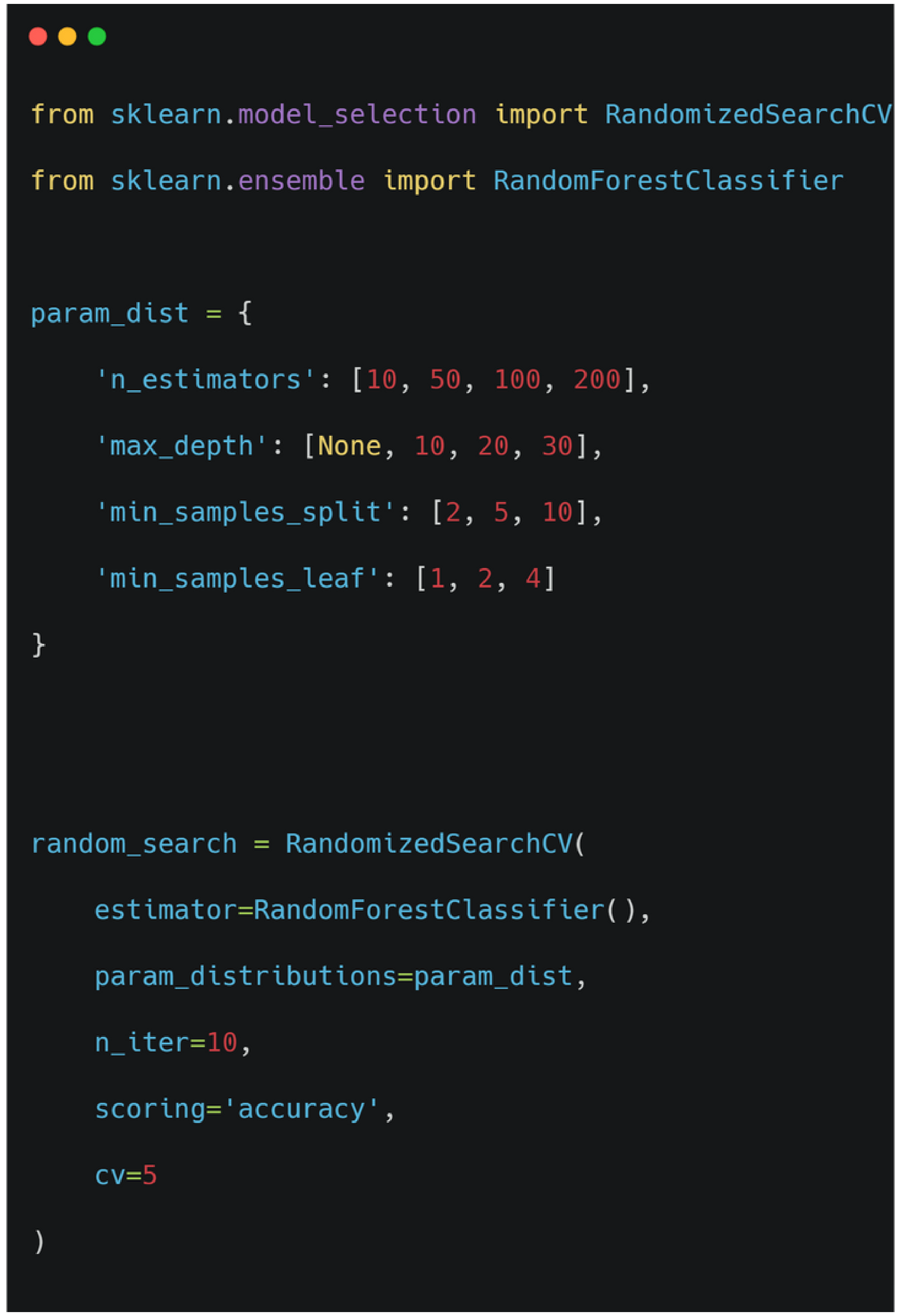
- It provides a good balance between exploration and exploitation, helping you find reasonably good hyperparameters without an exhaustive search.

RandomizedSearchCV



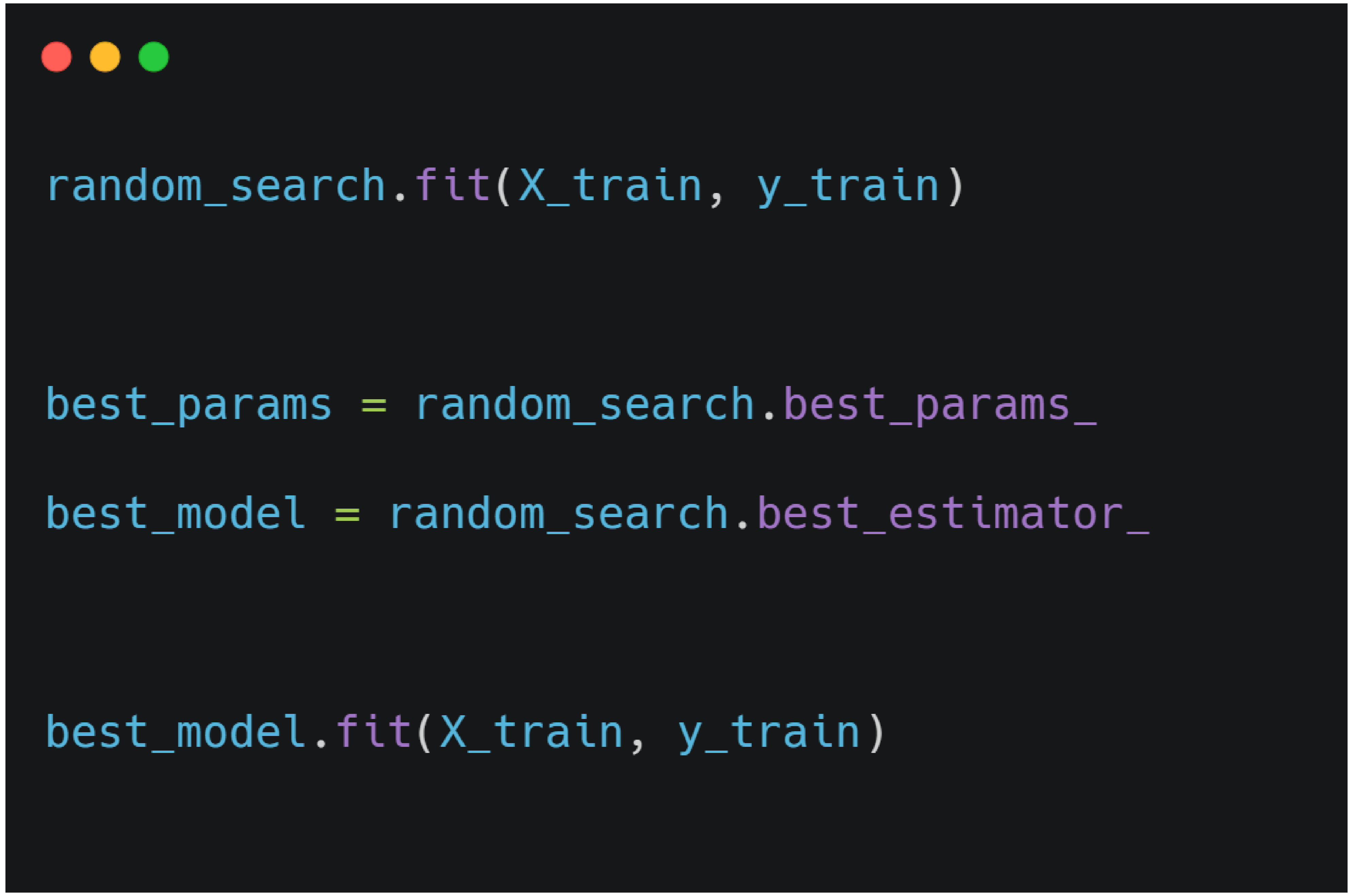


RandomizedSearchCV



```python
from sklearn.model_selection import RandomizedSearchCV

from sklearn.ensemble import RandomForestClassifier


param_dist = {

    'n_estimators': [10, 50, 100, 200],

    'max_depth': [None, 10, 20, 30],

    'min_samples_split': [2, 5, 10],

    'min_samples_leaf': [1, 2, 4]

}



random_search = RandomizedSearchCV(

    estimator=RandomForestClassifier(),

    param_distributions=param_dist,

    n_iter=10,

    scoring='accuracy',

    cv=5

)
```

RandomizedSearchCV



```
random_search.fit(X_train, y_train)


best_params = random_search.best_params_

best_model = random_search.best_estimator_


best_model.fit(X_train, y_train)
```

Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

MODEL EVALUATION

MODEL EVALUATION

- Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses.

- Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring.

PERFORMANCE COMPARISION

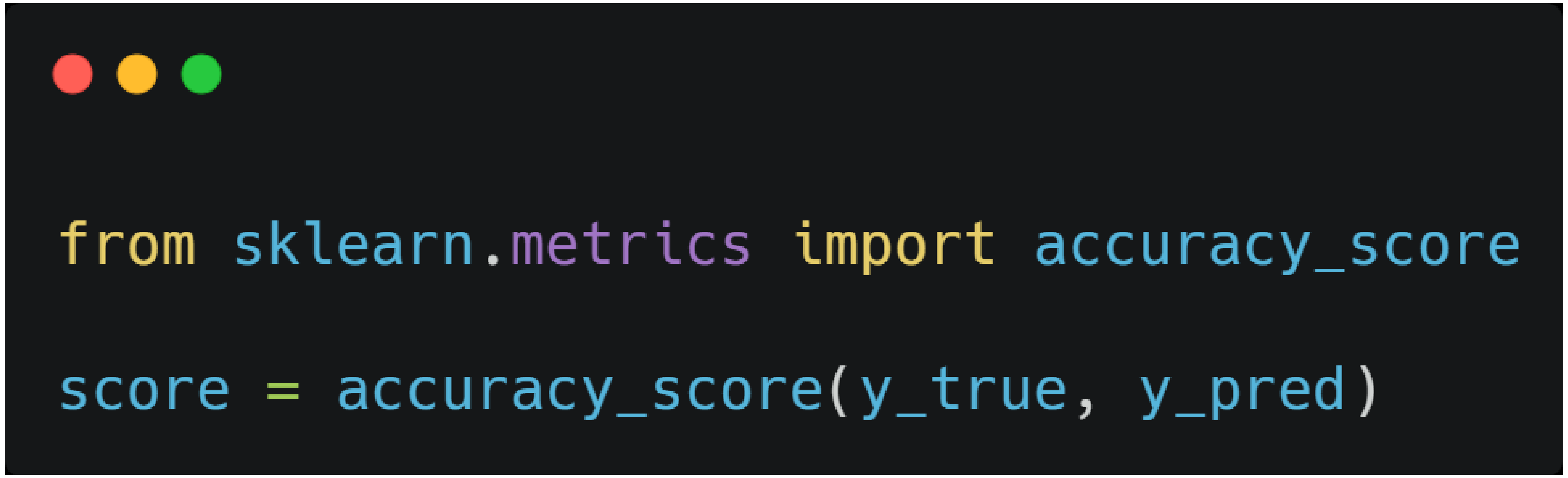
- **Classification problems -** Accuracy score, precision, confusion matrix
- **Regression problems -** Error calculation, Mean Absolute Error, Mean Squared Error

ACCURACY SCORE

- An Accuracy score is a Classification measure in Machine Learning that represents a percentage of correct predictions made by a model.

- Due to its simplicity in calculation and interpretation, the measure has found widespread use. Additionally, the performance of the model is quantified by a single number.

ACCURACY SCORE



```python
from sklearn.metrics import accuracy_score

score = accuracy_score(y_true, y_pred)
```

CLASSIFICATION REPORT

- The classification report visualizer displays the precision, recall, F1, and support scores for the model.

- There are four ways to check if the predictions are right or wrong:

  a. **TN / True Negative**: the case was negative and predicted negative

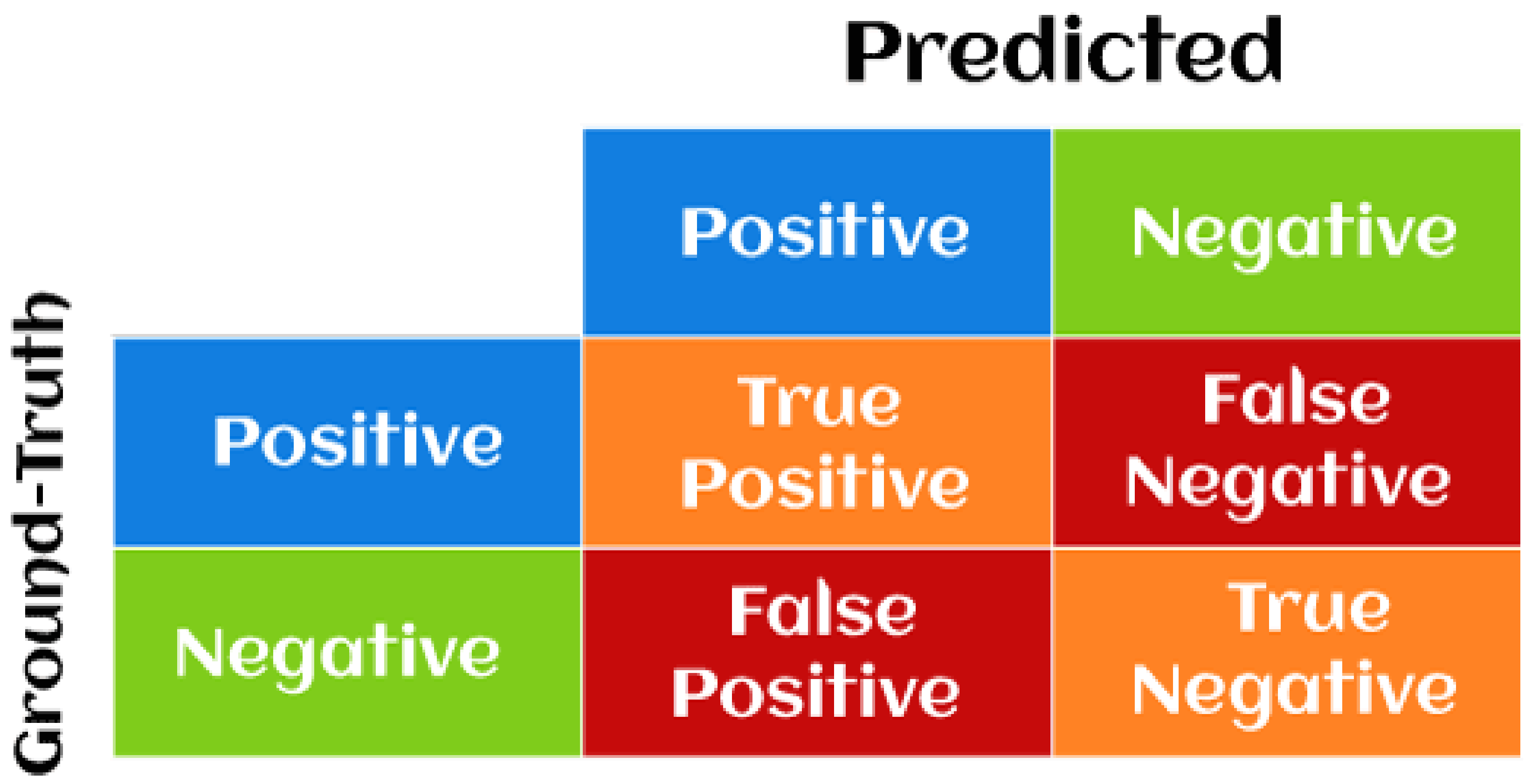
  b. **TP / True Positive**: the case was positive and predicted positive

  c. **FN / False Negative**: the case was positive but predicted negative

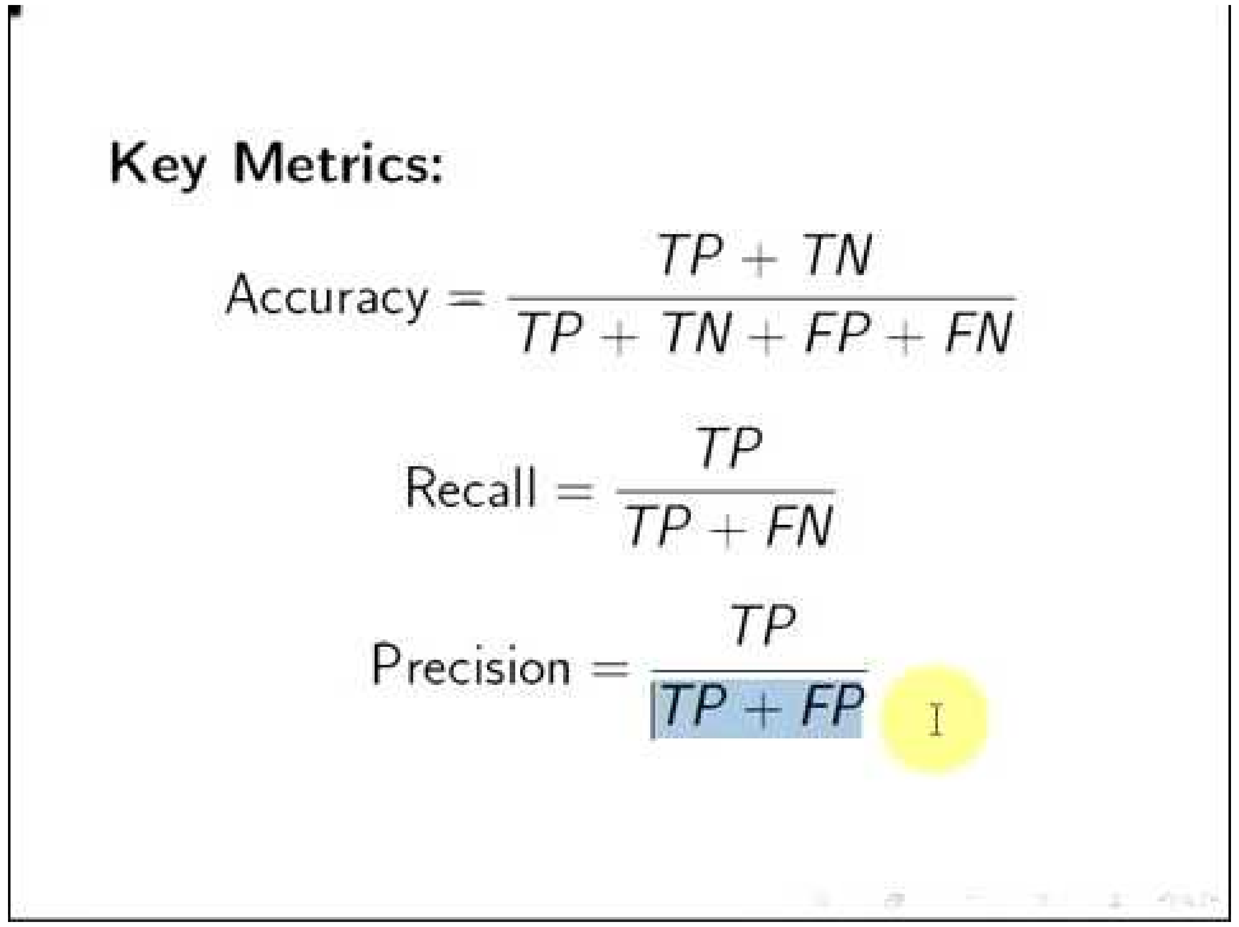
  d. **FP / False Positive**: the case was negative but predicted positive

CLASSIFICATION REPORT

```python
from sklearn.metrics import classification_report

report = classification_report(x_train, y_train)
```



Ground-Truth \ Predicted	Positive	Negative
Positive	True Positive	False Negative
Negative	False Positive	True Negative



Key Metrics:

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

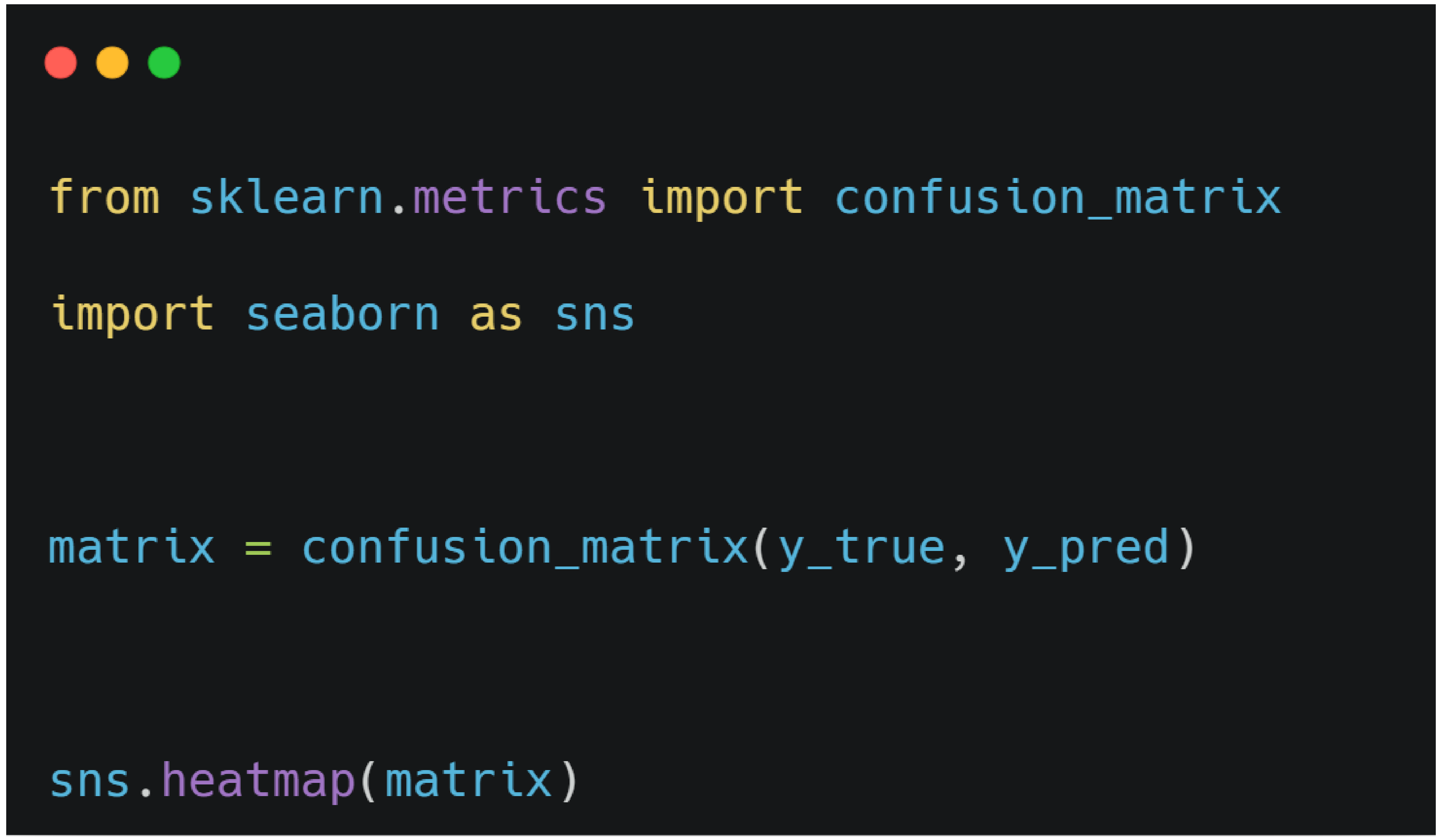
$$\text{Recall} = \frac{TP}{TP + FN}$$

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

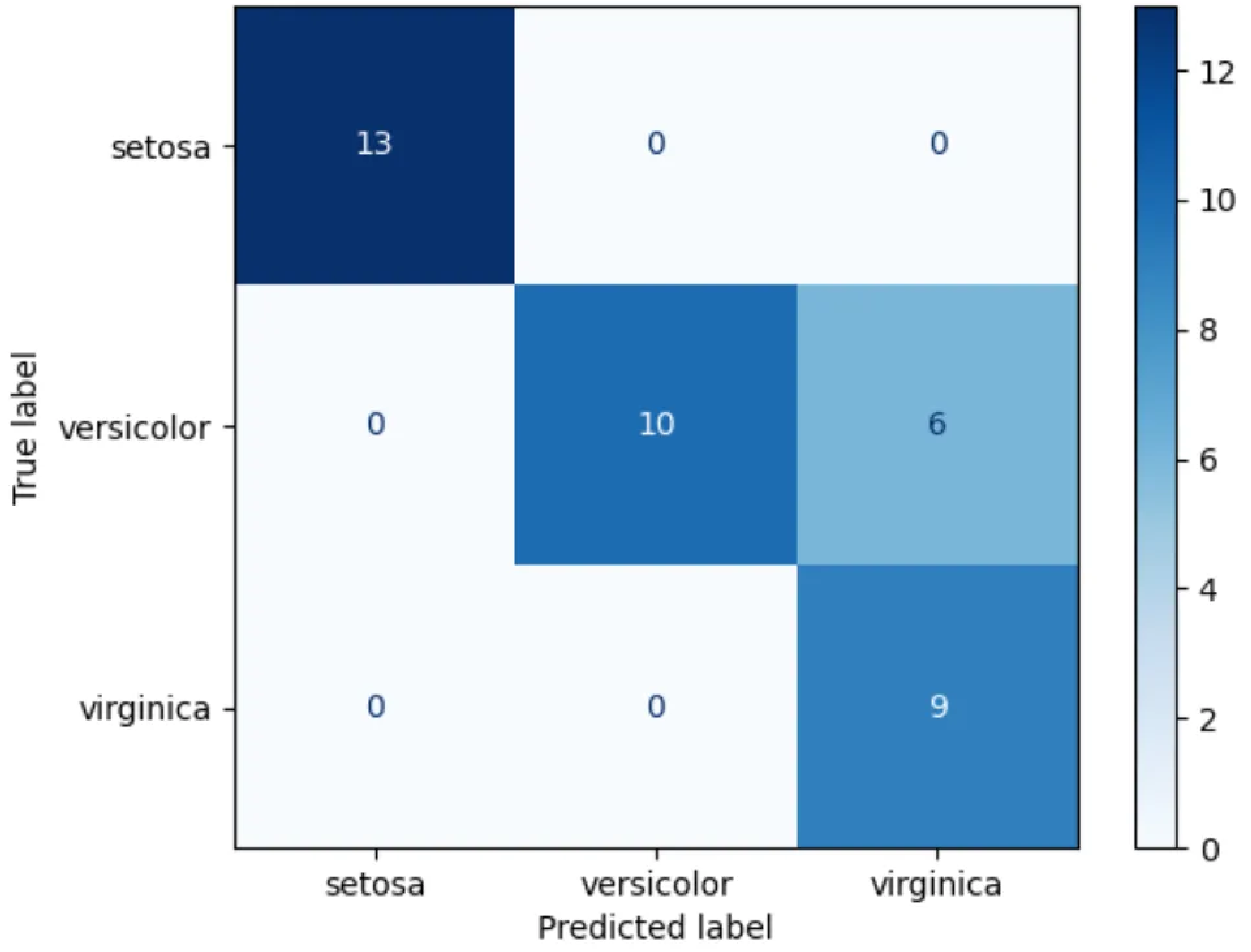
CLASSIFICATION REPORT

- Precision:- Accuracy of positive predictions.
- Precision = TP/(TP + FP)
- Recall — What percent of the positive cases did you catch?
- Recall = TP/(TP+FN)

CONFUSION MATRIX



```
from sklearn.metrics import confusion_matrix

import seaborn as sns


matrix = confusion_matrix(y_true, y_pred)


sns.heatmap(matrix)
```



R-SQUARED ERROR

- The R-squared (R2) score, also known as the coefficient of determination, is a statistical measure used to assess the goodness of fit of a regression model.

- In the context of machine learning and statistics, it is often used to evaluate the performance of a regression model that makes predictions about a continuous outcome variable.

R-SQUARED ERROR

Formula

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2	= coefficient of determination
RSS	= sum of squares of residuals
TSS	= total sum of squares

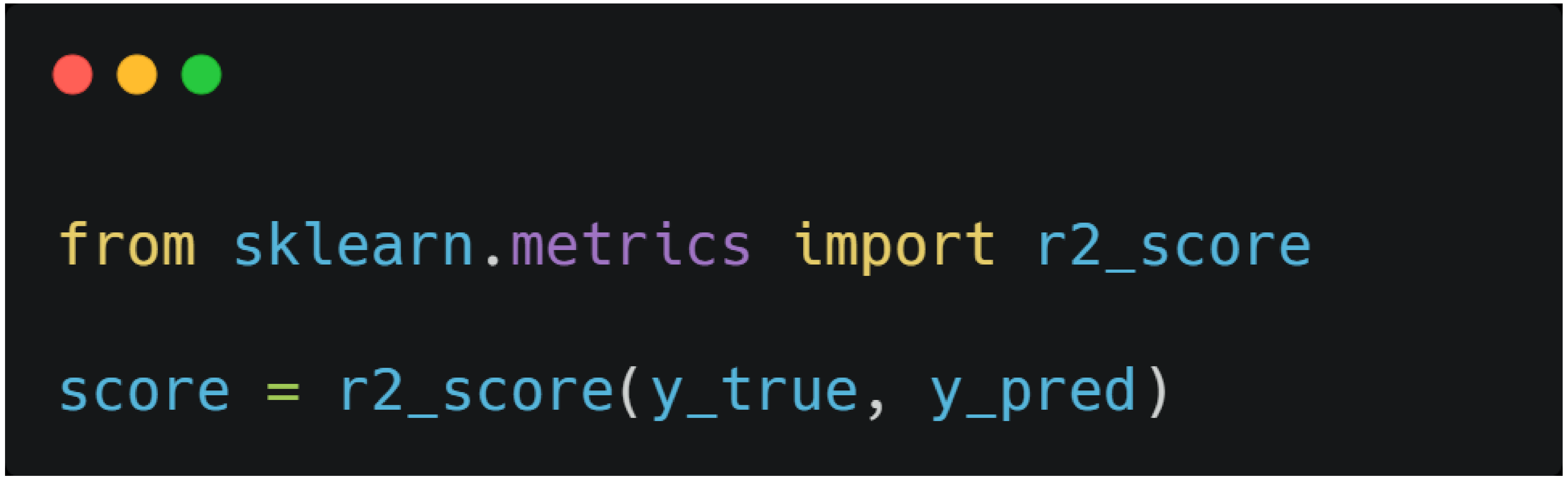
$$\text{RSS} = \Sigma\left(y_i - \widehat{y}_i\right)^2$$

Where: y_i is the actual value and, $\widehat{y}_i$ is the predicted value.

$$\text{TSS} = \Sigma\left(y_i - \overline{y}\right)^2$$

Where: y_i is the actual value and $\overline{y}$ is the mean value of the variable/feature

R-SQUARED ERROR



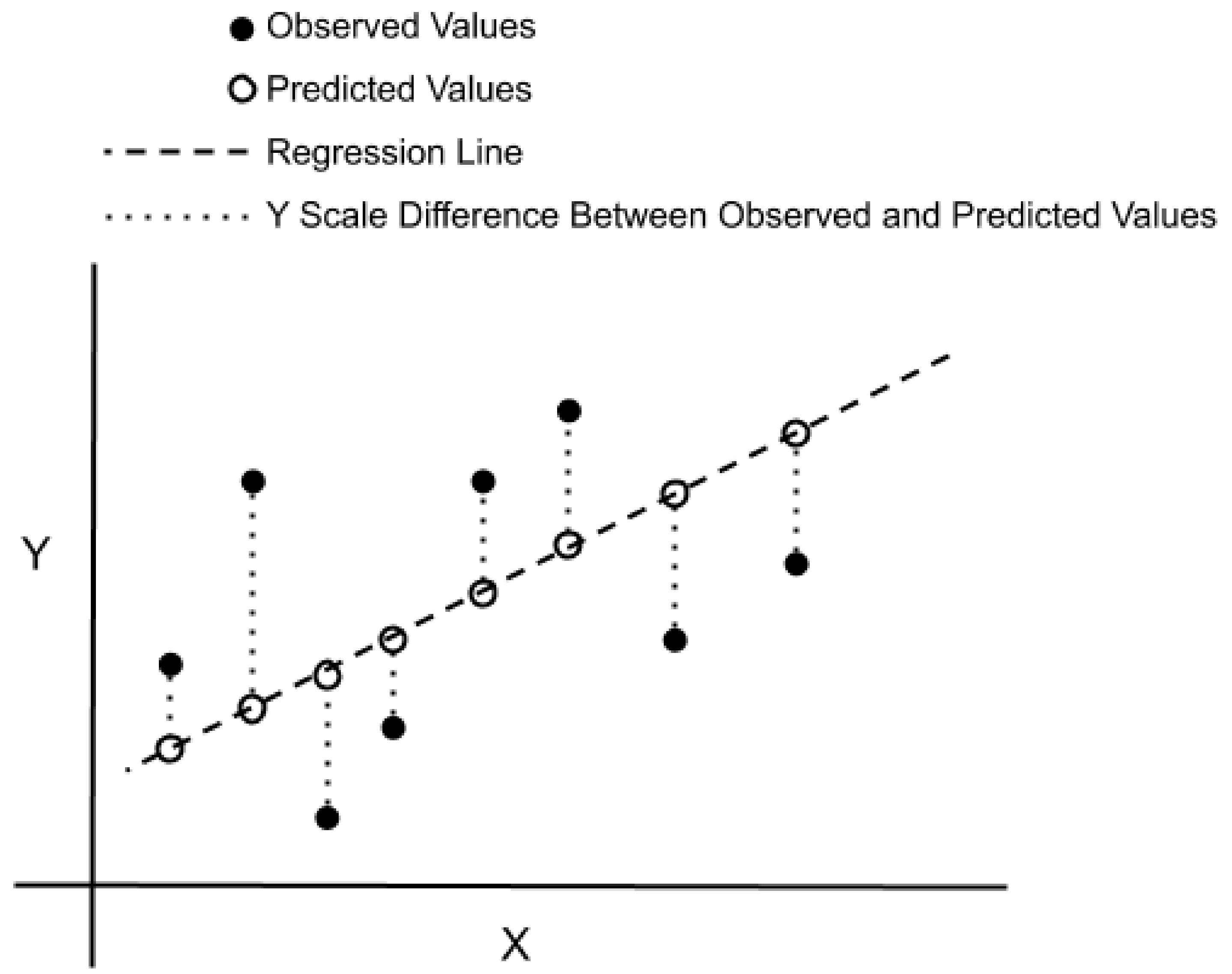
```python
from sklearn.metrics import r2_score

score = r2_score(y_true, y_pred)
```

MEAN SQUARED ERROR

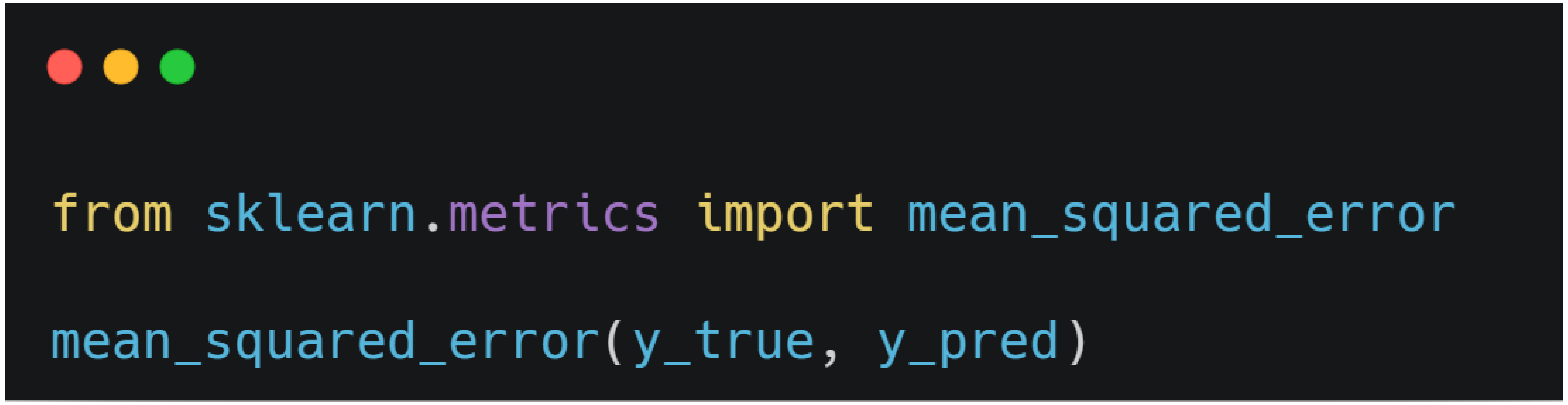
- The Mean Squared Error (MSE) is a commonly used metric to measure the quality of a predictive model, particularly in the context of regression problems.
- It quantifies the average of the squared differences between the predicted values and the actual (observed) values.
- In other words, it calculates the average squared error between the model's predictions and the true values.

MEAN SQUARED ERROR

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$



MEAN SQUARED ERROR



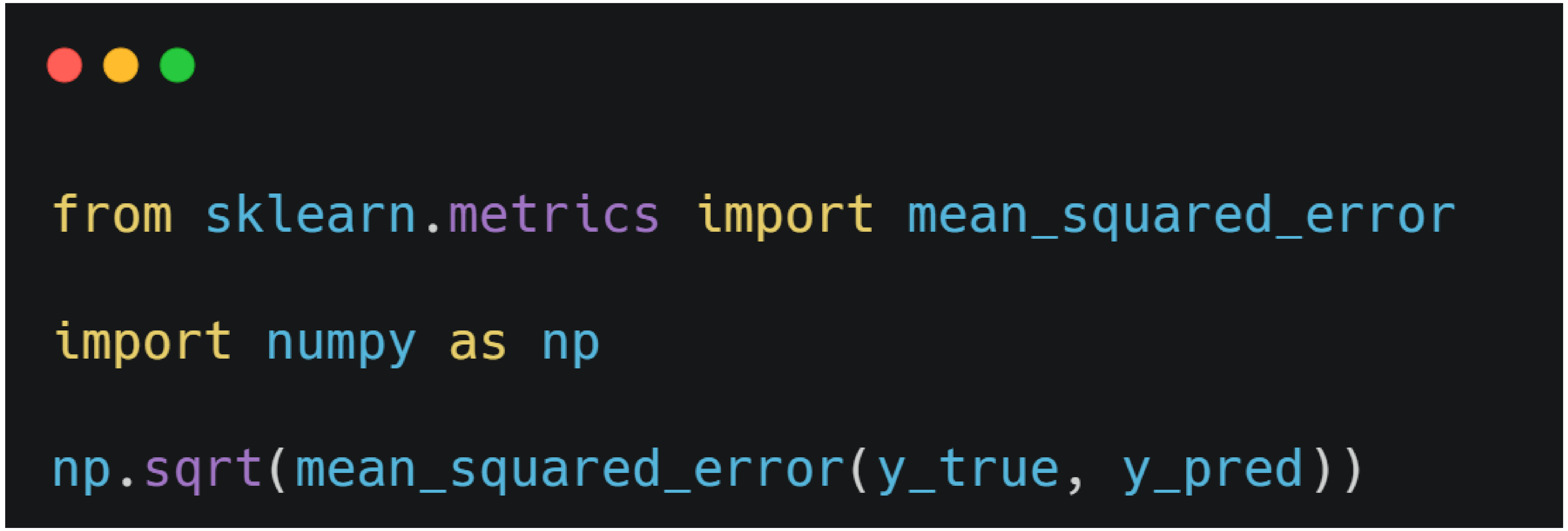
```python
from sklearn.metrics import mean_squared_error

mean_squared_error(y_true, y_pred)
```

ROOT MEAN SQUARED ERROR

- Root Mean Squared Error is the square root of Mean Squared error.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

ROOT MEAN SQUARED ERROR



```
from sklearn.metrics import mean_squared_error

import numpy as np

np.sqrt(mean_squared_error(y_true, y_pred))
```

REGRESSION

	y	$\hat{y}$	$y-\hat{y}$
Age	Failures	Prediction	Error
10	15	26	11
20	30	32	2
40	40	44	4
50	55	50	-5
70	75	62	-13
90	90	74	-16

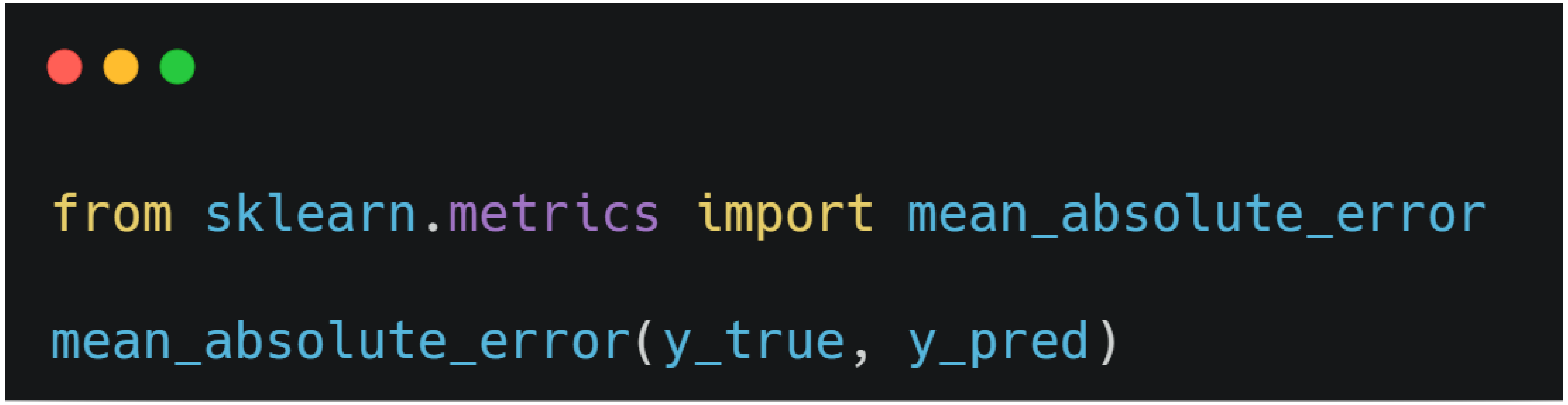
$(y-\hat{y})^2$
Error^2
121
4
16
25
169
256

Mean of Error^2	$\frac{\Sigma(y-\hat{y})^2}{N}$	98.5
Square root of Mean of Error^2	$\sqrt{\frac{\Sigma(y-\hat{y})^2}{N}}$	**9.9**

MEAN ABSOLUTE ERROR

- The Mean absolute error represents the average of the absolute difference between the actual and predicted values in the dataset.
- It measures the average of the residuals in the dataset.
- Mean Absolute Error (MAE) is a valuable metric for assessing the accuracy of regression models. It provides a straightforward and interpretable measure of the average magnitude of errors in the same units as the target variable.

MEAN ABSOLUTE ERROR



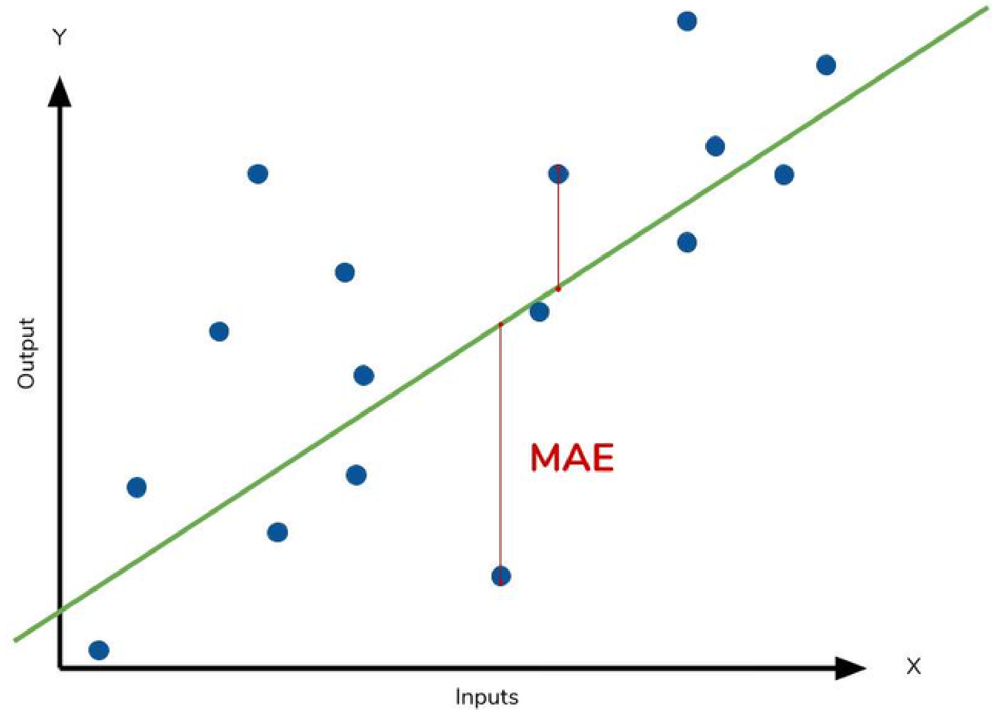
```
from sklearn.metrics import mean_absolute_error

mean_absolute_error(y_true, y_pred)
```

MEAN ABSOLUTE ERROR

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$\text{Mean absolute error} = \frac{\text{Sum of all absolute errors}}{\text{No of errors } (n)}$$

$$= \sum_{i=1}^{n} \frac{X_i - X}{n}$$



REGRESSION

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

$\hat{y}$ $-$ *predicted value of y*

$\bar{y}$ $-$ *mean value of y*

THANK YOU

Pantech e Learning
DIGITAL LEARNING SIMPLIFIED