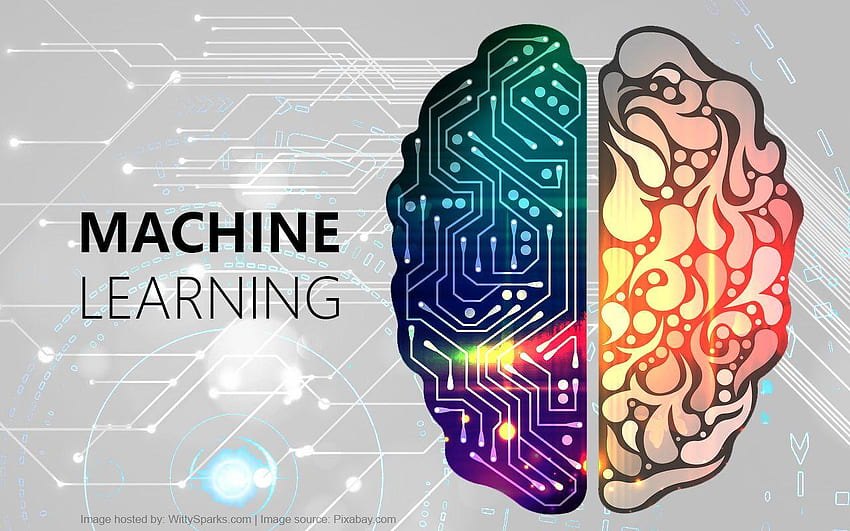
Using KNN Algorithm

Hussein Issa

Iris Dataset Model



Introduction (iris dataset)

.....................................................................

Data Preprocessing (exploration)

........................................................................

KNN Algorithm Explanation

........................................................................

Model Train & Evaluation

........................................................................

Results and Discussion

.......................................................................

Conclusion

Introduction

The Iris dataset contains three species of Iris flowers: Iris setosa, Iris versicolor, and Iris virginica. It consists of 150 data points (150 observation), with 50 samples for each species.

**Source:**

The data was obtained by measuring the length and width of the sepals and petals of the three Iris species from different locations.

**Purpose:**

The main purpose of the Iris dataset is to serve as a benchmark for testing and evaluating machine learning algorithms (**KNN**), particularly those for classification tasks. It is commonly used to explore and demonstrate various techniques for data analysis, visualization, and predictive modeling.

**The four features are :**

Sepal Length: This represents the length of the sepals, it is measured in centimeters .

Sepal Width: This represents the width of the sepals, it is measured in centimeters .

Petal Length: This feature represents the length of the petals, it is measured in centimeters .

Petal Width: This represents the width of the petals, it is measured in centimeters .

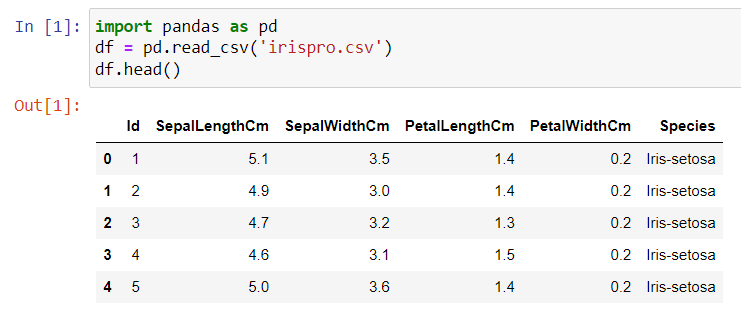
**One target with three classes:**

The target is to get the species of each flower in the iris dataset , they are 3 species as following:

Setosa: This class corresponds to the species "Iris setosa."

Versicolor: This class corresponds to the species "Iris versicolor."

Virginica: This class corresponds to the species "Iris virginica."

**importing the dataset from desktop and printing the first 5 rows** :

**The dataset columns :**

A close up of a text

Description automatically generated

**some info about the dataset :**

A screenshot of a computer code

Description automatically generated

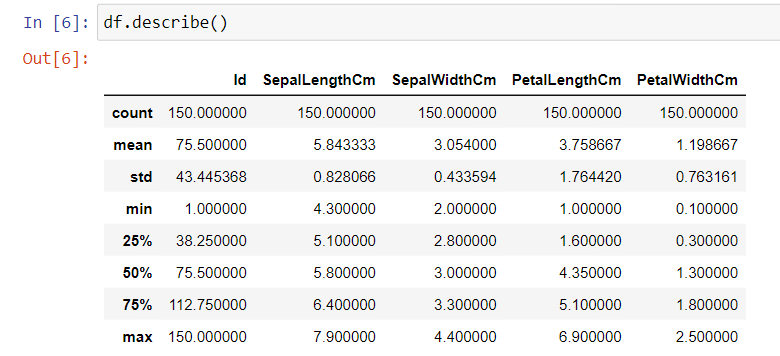
We conclude that there is no missing values, all values are not null and the target is an object not a numeric target, So we should use the classification in the supervised machine learning (logistic regression or knn).

A close up of a number

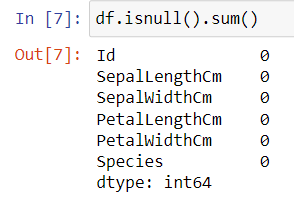
Description automatically generated

It consists of 150 observations and 6 columns.

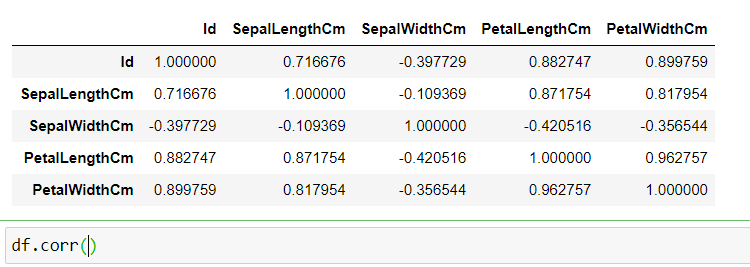
**Describing the dataset (mean, max , min, etc) :**



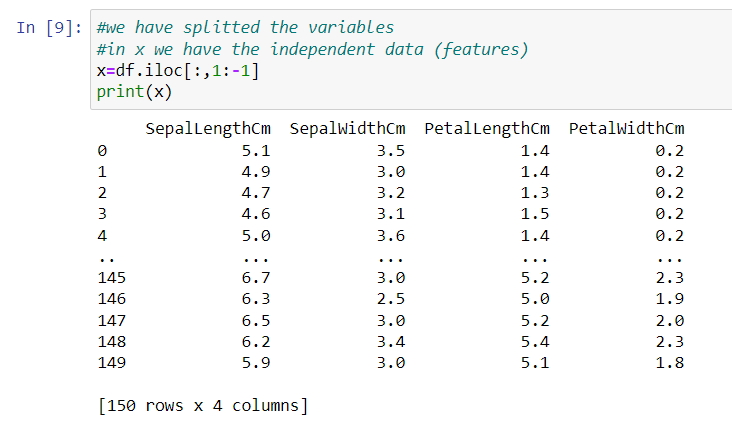
**Checking if there’s any null values :**

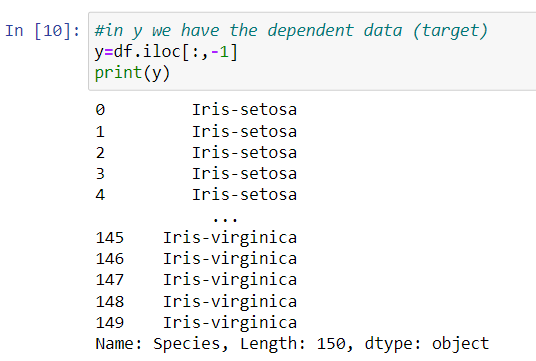


**Correlations between the features :**

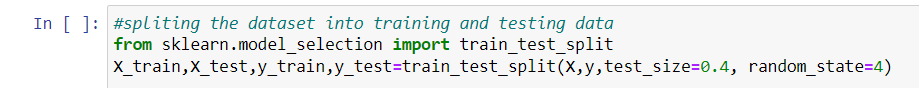
****

**Splitted the dataset into x,y (note: we didn’t take the id column because we don’t need it):**





**Training and testing part :**



**Note : Test size is 40% so the training part is 60% and the random state is 4 so he can randomize the observations and repeat it .**

**A screenshot of a computer code

Description automatically generated**

The K-Nearest Neighbors (KNN) algorithm is supervised machine learning used for classification tasks. We train the model and test it, the model should not memorize, and should use the training to make predictions.

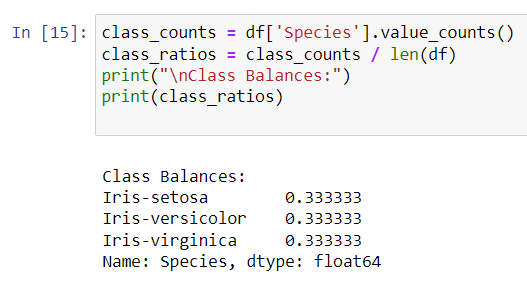
The k in KNN determines the number of nearest neighbors that will be considered for making predictions there is no optimal k number so we should deduce it .

When we have new data to predict, the algorithm works by calculating the distance between the data nodes from the new data by looking at the k number given to know which class it belongs to the more nodes that consists of a specific class.

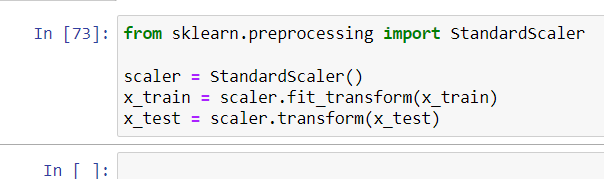
Choosing a very low value will most likely lead to inaccurate predictions. The commonly used value of K is 5. Always use an odd number as the value of K.

Its not very sensitive about the outliers.

**Checking the balance of the df :**

****

The reason feature scaling is important in KNN is that the algorithm makes predictions based on the distance between data points.



We calculated the square root of the length of test set which is the rule of thumb to conclude the max k to take it into consideration.

By taking all the odd values from 1 till the odd int value we got.

A screenshot of a computer

Description automatically generated

We imported the sklearn libraries to use it in metrics, classifier and accuracy.

We did a range between 1 and 7 that we got up.

With a loop :

We have fitted the train set in the model.

And got the accuracy by comparing the target test with the prediction of features test, and we added the accuracy score to the list within matrics.

By adding an if in the loop we deduced the best accuracy in the new list (scores) we created.

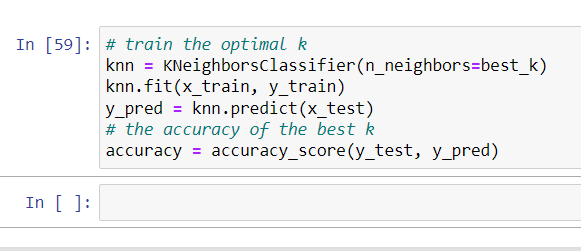
So the best k is 7 and the acc is 0.983

A screenshot of a computer

Description automatically generated

Training with the optimal k with is 7

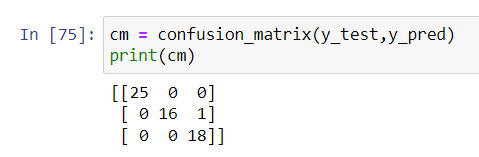
#best\_k = 7 because this k value has the highest accuracy score between the other k neighbors



Confusion matrix:

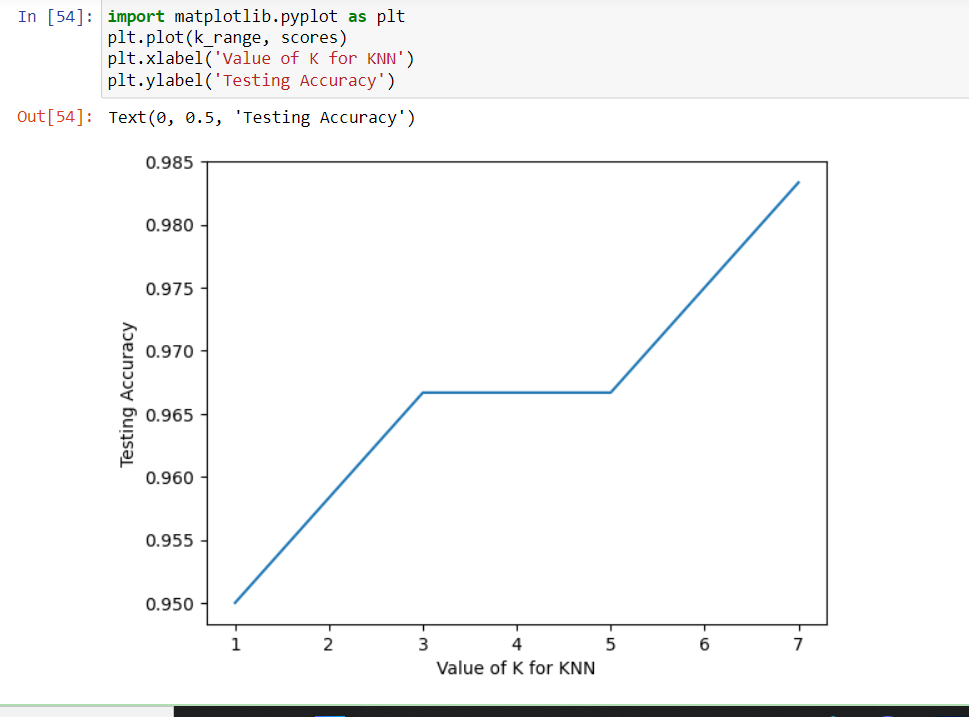
3 classes of species

Actual data and predicted data

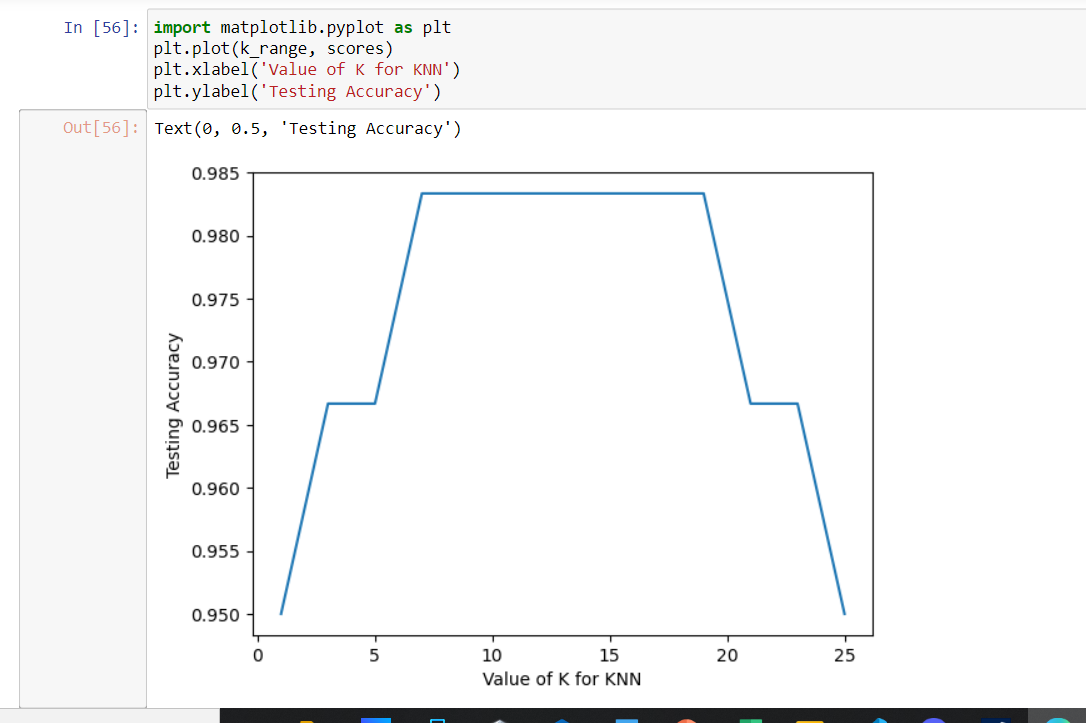


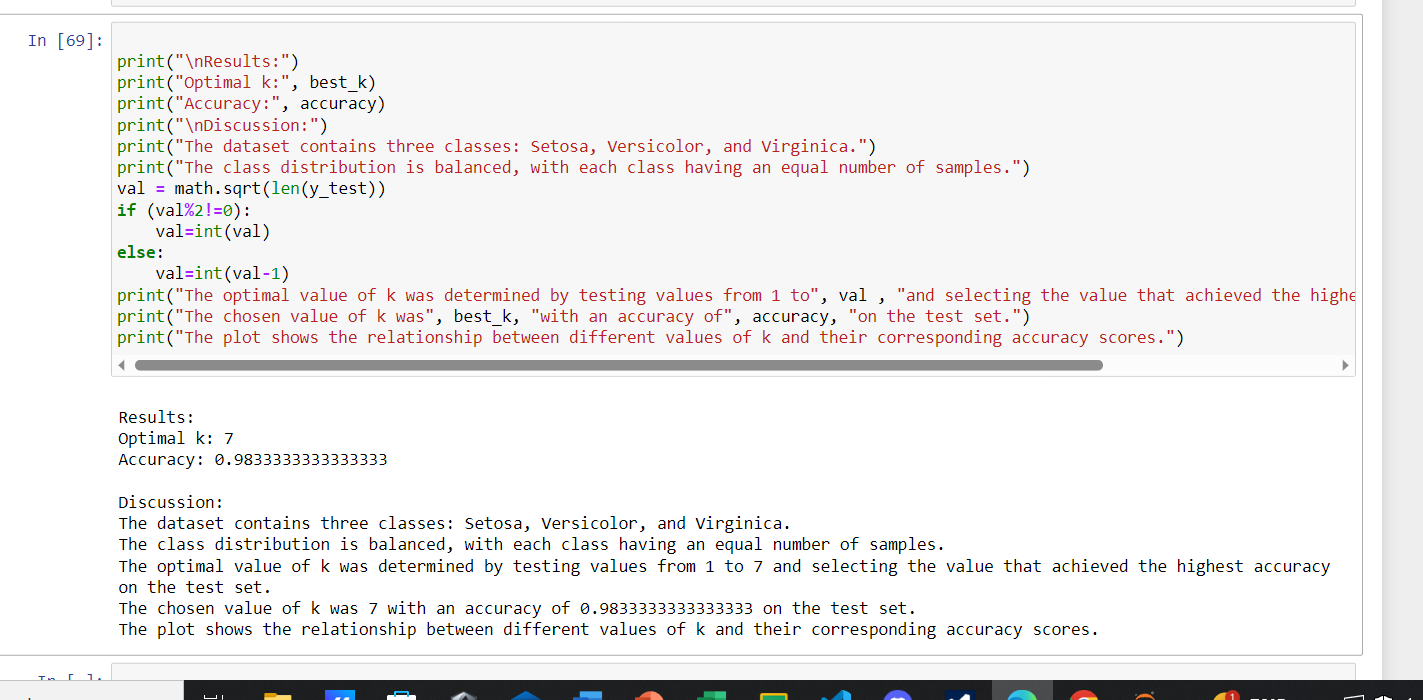
using the matplotlib library we can deduce the line plot (visualize).

Added to the plot the range of k (1 till 7) values and the list of scores (accuracy score)



Added to the plot the range of k (1 till 25) values and the list of scores (accuracy score)





To improve the performance of the knn model we could feature scale by normalizing the features to ensure the data distance between points are scaled and meaningful. Also we could reduce the noise in the dataset irrelevant features such as outliers, we can handle the missing values by dropping or replacing them by mean, mode etc.

We can suggest more algorithms to try like logistic regression which uses classification target. By adding more observations to the dataset to train and test the model with bigger data so he don’t memorize it.

Complexity:

When making a prediction for a new data point, KNN needs to find the k-nearest neighbors from the entire training dataset. For large datasets, this process can be very time-consuming, as it requires computing the distance for all of them.

Noisy data can lead to misclassifications and reduce the overall accuracy of the algorithm. We should clean or preprocess the data.

we faced an issue during computing this dataset which is choosing the k value because the lower the k was the more overfitting was there, while the large k can lead to a misclassification.

In botany and plant science, the Iris dataset can be used to classify and study different species of Iris flowers, for understanding the taxonomy, ecology, and evolution of Iris flowers.

In the field of biology , can be used for species recognition based on various features, such as DNA sequences or fingerprint patterns. By using KNN, researchers can compare the target features with known samples and classify them into specific species or individuals.

Knn can be used also in image processing which is useful nowadays. Example: the difference between the cat and dog image.

Steps we took to do the project:

Data collection/data extraction: Search for the data, its extraction, and subsequent preparation related to data analysis - Input data must be chosen with the basic purpose to build a predictive model, and its selection is crucial for the success of the analysis as well. Thus, a poor choice of data, or performing analysis on a data set that is not representative of the system, will lead to models that will deviate from the system under study - Prepare the data: data is in proper format and of good quality/ fixing issues such as missing data, inconsistent values, and treatment of outliers - Exploratory analysis is one method to study the nuances of data in details, thereby burgeoning the relevant content of the data

Train the algorithm: This step involves choosing the appropriate algorithm and representation of data in the form of the model. The cleaned-up data is split into two parts: train and test; the first part (training data) is used for developing the model, and the second part (test data) is used as a reference. The proportion of data split depends on the prerequisites such as the number of input variables and complexity of the model.

- Test the algorithm: Each machine learning model results in a based solution to the learning problem, so it is important to evaluate how well the algorithm is learned. - Depending on the type of model used, one can evaluate the accuracy of the model using a test dataset or may need to develop measures of performance specific to the intended application. - To test the performance of the model, the second part of the data (test data) is used. This step determines the precision of the choice of the algorithm based on the desired outcome.

-Improving the performance: A better test to check the performance of a model is to observe its performance on the data that was not used during building the model. If better performance is needed, it becomes necessary to utilize more advanced strategies to augment the performance of the model. This step may involve choosing a different model altogether or introducing more variables to augment the accuracy. Hence, significant amount of time needs to be spent in data collection and preparation. if the model appears to be performing satisfactorily, it can be deployed for the intended task. The successes and failures of a deployed model might even provide additional data for the next generation of model.

Thank you