# K-Nearest Neighbor

## Introduction

K-nearest neighbors (KNN) is a simple and widely used classification algorithm in machine learning. It works by finding the 'k' training data points that are closest to a new data point and assigning the majority class among them to the new point. In this lab, we will implement a KNN classifier using the Iris dataset, which consists of measurements for 150 iris flowers from three different species.

**Lab Task**

1. **Importing Necessary Libraries**

```
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
import numpy as np
from sklearn.model_selection import train_test_split
```

2. **Load, visualize and split dataset**
   Load the Iris dataset, visualize it using Histogram distributions and scatter plots and then split it into training and testing sets using the `train_test_split` function.

3. **Create and Train the KNN Classifier:**
   Create a KNN classifier using the KNeighborsClassifier class and train it with the training data.

```
kn = KNeighborsClassifier()
kn.fit(X_train, y_train)
```

4. **Make Predictions and Evaluate:**
   Make predictions on the test data using the trained classifier and evaluate its accuracy.

```
prediction = kn.predict(X_test)
accuracy = kn.score(X_test, y_test)
print("ACCURACY: " + str(accuracy))
```

5. **Print Predictions and Actual Values:**
   Print the predicted and actual species for each data point in the test set.

```
target_names = iris_dataset.target_names
for pred, actual in zip(prediction, y_test):
  print("Prediction is " + str(target_names[pred]) + ", Actual is " + str(target_names[actual]))
```

## Lab Task:

Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers

**Implement KNN classifier to Identify the customer segments to whom the loan can be granted.**

Dataset source: https://www.kaggle.com/burak3ergun/loan-data-set

a. **Importing the libraries**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

b. Use pandas.read_csv to read the csv file corresponding to the dataset.

```
dataset = pd.read_csv("loanDataset.csv")
dataset.head()
```

c. **Data preprocessing**: check for missing values, imputation of missing values, check missing values after imputation, drop unnecessary columns

d. **Exploratory Data analysis**

I. Dataset shape

II. Dataset info

III. Gender obtaining the maximum number of loans

```
sns.countplot(y = 'Gender', hue = 'Loan_Status', data = dataset)
dataset['Gender'].value_counts()
```

IV. Does marital status affect loan approval?

V. Does education status affect loan approval?

VI. Does employment affect loan approval?

VII. Does credit history affect loan approval?

**e. Model Building**

Before building the model, we need to perform label encoding for the categorical variables because categorical data must be encoded into numbers before using it to fit and evaluate a model.

```
#Converting some object data type to int
gender = {"Female": 0, "Male": 1}
yes_no = {'No' : 0,'Yes' : 1}
dependents = {'0':0,'1':1,'2':2,'3+':3}
education = {'Not Graduate' : 0, 'Graduate' : 1}
property = {'Semiurban' : 0, 'Urban' : 1,'Rural' : 2}
output = {"N": 0, "Y": 1}dataset['Gender'] =
dataset['Gender'].replace(gender)
dataset['Married'] = dataset['Married'].replace(yes_no)
dataset['Dependents'] = dataset['Dependents'].replace(dependents)
dataset['Education'] = dataset['Education'].replace(education)
dataset['Self_Employed'] = dataset['Self_Employed'].replace(yes_no)
dataset['Property_Area'] =
dataset['Property_Area'].replace(property)
dataset['Loan_Status'] = dataset['Loan_Status'].replace(output)
```

Dataset after converting object data types into an integer

```
  dataset.head()
```

f. Set the values for independent (X) variable and dependent (Y) variable

g. Split the dataset into train and test set

**h.** Fit the KNN model

i. Prediction on the test set

```
#Prediction of test set
prediction_knn = knn.predict(X_test)
#Print the predicted values
print("Prediction for test set: {}".format(prediction_knn))
```

j. Get the Actual values and the predicted values

```
#Actual value and the predicted value
a = pd.DataFrame({'Actual value': Y_test, 'Predicted value':
prediction_knn})a.head()
```

k. Evaluate the Model

**References:**

https://medium.com/machine-learning-with-python/k-nearest-neighbour-knn-implementation-in-python-498daa39c16e

Q: Write a python script to delete a column from the iris dataset if the mean value of the column is above a threshold value.