

ROBO 2.0 REVIEW

K-MEANS CLUSTERING

```

import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
X, y_true = make_blobs(n_samples=300, centers=4, cluster_std=0.6,
random_state=0)
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
labels = kmeans.labels_
centers = kmeans.cluster_centers_
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
plt.scatter(centers[:, 0], centers[:, 1], marker='X', color='red', s=200)
plt.title("K-means Clustering")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()

```

Decision trees

```

import warnings
warnings.filterwarnings("ignore")
# Importing the required packages
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
# Function importing Dataset
def importdata():
    balance_data = pd.read_csv('https://archive.ics.uci.edu/ml/machine-
learning-databases/balance-scale/balance-scale.data',
    sep= ',', header = None)
    # Printing the dataset shape
    print ("Dataset Length: ", len(balance_data))
    print ("Dataset Shape: ", balance_data.shape)
    print ("Dataset: ", balance_data.head())
    return balance_data
def splitdataset(balance_data):
    X = balance_data.values[:, 1:5]
    Y = balance_data.values[:, 0]
    X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size = 0.3, random_state = 100)
    return X, Y, X_train, X_test, y_train, y_test
def train_using_gini(X_train, X_test, y_train):

```

