|  |  |
| --- | --- |
| EX. NO: 2 | **WEEK 1 - A**  **PIZZA LIKING PREDICTION USING KNN NEIGHBOUR ALGORITHM** |
| 2-12-24 |  |

**AIM:**

To write a program to predict the pizza liking using KNN algorithm.

**CODE:**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

import numpy as np

# Function to load the data from CSV

def load\_data(file\_path):

df = pd.read\_csv('pizza\_data.csv')

return df

# Data Preprocessing function

def preprocess\_data(df):

# Feature selection: Define features and target variable

X = df[['age', 'gender', 'location', 'income\_level', 'pizza\_type\_preference', 'is\_vegetarian']]

y = df['pizza\_like']

# Encoding categorical features

label\_encoder\_gender = LabelEncoder()

X['gender'] = label\_encoder\_gender.fit\_transform(X['gender'])

label\_encoder\_location = LabelEncoder()

X['location'] = label\_encoder\_location.fit\_transform(X['location'])

label\_encoder\_income = LabelEncoder()

X['income\_level'] = label\_encoder\_income.fit\_transform(X['income\_level'])

label\_encoder\_pizza\_type = LabelEncoder()

X['pizza\_type\_preference'] = label\_encoder\_pizza\_type.fit\_transform(X['pizza\_type\_preference'])

# Split the dataset into training and testing sets (80-20 split)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

return X\_train\_scaled, X\_test\_scaled, y\_train, y\_test, label\_encoder\_gender, label\_encoder\_location, label\_encoder\_income, label\_encoder\_pizza\_type, scaler

# Function to train the RandomForest model

def train\_model(X\_train\_scaled, y\_train):

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train\_scaled, y\_train)

return model

# Function to predict pizza like based on user input

def predict\_pizza\_like(model, scaler, label\_encoders, age, gender, location, income\_level, pizza\_type\_preference, is\_vegetarian):

# Preprocess the user input data

user\_data = pd.DataFrame({

'age': [age],

'gender': [gender],

'location': [location],

'income\_level': [income\_level],

'pizza\_type\_preference': [pizza\_type\_preference],

'is\_vegetarian': [is\_vegetarian]

})

# Use the label encoders to transform categorical values

user\_data['gender'] = label\_encoders['gender'].transform(user\_data['gender'])

user\_data['location'] = label\_encoders['location'].transform(user\_data['location'])

user\_data['income\_level'] = label\_encoders['income\_level'].transform(user\_data['income\_level'])

user\_data['pizza\_type\_preference'] = label\_encoders['pizza\_type\_preference'].transform(user\_data['pizza\_type\_preference'])

# Scale the input data using the same scaler used for training

user\_data\_scaled = scaler.transform(user\_data)

# Make the prediction using the trained model

prediction = model.predict(user\_data\_scaled)

# Return the prediction result

return "Likes Pizza" if prediction[0] == 1 else "Does Not Like Pizza"

# Main function to run the prediction system

def run\_prediction\_system():

# Load the dataset (Replace 'pizza\_data.csv' with your actual CSV file path)

file\_path = 'pizza\_data.csv'

df = load\_data(file\_path)

# Preprocess the data

X\_train\_scaled, X\_test\_scaled, y\_train, y\_test, label\_encoder\_gender, label\_encoder\_location, label\_encoder\_income, label\_encoder\_pizza\_type, scaler = preprocess\_data(df)

# Train the model

model = train\_model(X\_train\_scaled, y\_train)

# Evaluate the model

y\_pred = model.predict(X\_test\_scaled)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Model Accuracy: {accuracy:.4f}")

# Get user input for prediction

print("\nPlease provide the following information to predict if you like pizza.")

age = int(input("Age: "))

gender = input("Gender (Male/Female): ")

location = input("Location (Urban/Suburban/Rural): ")

income\_level = input("Income Level (Low/Medium/High): ")

pizza\_type\_preference = input("Pizza Type Preference (Vegetarian/Non-Vegetarian): ")

is\_vegetarian = int(input("Are you vegetarian? (1 = Yes, 0 = No): "))

# Make prediction based on user input

prediction = predict\_pizza\_like(model, scaler, {

'gender': label\_encoder\_gender,

'location': label\_encoder\_location,

'income\_level': label\_encoder\_income,

'pizza\_type\_preference': label\_encoder\_pizza\_type

}, age, gender, location, income\_level, pizza\_type\_preference, is\_vegetarian)

# Print the prediction result

print(f"Prediction: {prediction}")

# Run the prediction system

if \_\_name\_\_ == "\_\_main\_\_":

run\_prediction\_system()

**OUTPUT:**

Model Accuracy: 0.4000

Please provide the following information to predict if you like pizza.

Age: 28

Gender (Male/Female): Male

Location (Urban/Suburban/Rural): Suburban

Income Level (Low/Medium/High): High

Pizza Type Preference (Vegetarian/Non-Vegetarian): Non-Vegetarian

Are you vegetarian? (1 = Yes, 0 = No): 0

**Prediction: Does Not Like Pizza**

**RESULT:**

Thus, the program is successfully executed.