

```
from tkinter import messagebox
from tkinter import *
from tkinter.filedialog import askopenfilename
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import os
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_text
from sklearn.feature_selection import RFE

root = tkinter.Tk()
#creating the application main/root window.

root.title("Recommendation System for Tourists")
root.geometry("1200x850")

global filename
```

```
feature_cols =  
['userid','art_galleries','dance_clubs','juice_bars','restaurants','museums','resorts',  
'parks_picnic_spots','beaches','theaters','religious_institutions']
```

```
global clf
```

```
global rfe
```

```
global X_train
```

```
global y_train
```

```
global fit
```

```
def upload():
```

```
    global filename
```

```
    filename = filedialog.askopenfilename(initialdir="dataset")
```

```
    pathlabel.config(text=filename)
```

```
    text.delete('1.0', END)
```

```
    with open(filename, "r") as file:
```

```
        for line in file:
```

```
            line = line.strip('\n')
```

```
            text.insert(END,line+"\n")
```

```
def featureSelection():
```

```
    global clf
```

```
    global rfe
```

```
    global fit
```

```
    global X_train
```

```
    global y_train
```

```

dataset = pd.read_csv(filename)
dataset.head()
y = dataset['location']
X = dataset.drop(['location'], axis = 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1,
random_state=0)
clf = DecisionTreeClassifier()
rfe = RFE(clf, 3)
fit = rfe.fit(X_train,y_train)
text.delete('1.0', END)
text.insert(END,"Total number of features : "+str(len(feature_cols))+"\n")
text.insert(END,"Selected number of features : "+str(fit.n_features_)+"\n")
text.insert(END,"Features masking: "+str(fit.support_)+"\n")
text.insert(END,"Features ranking : "+str(fit.ranking_)+"\n")
text.insert(END,"Selected features are : "+"\\n")

for i in range(X.shape[1]):
    if fit.ranking_[i] == 1:
        text.insert(END, "Feature-" + str(i+1) + ": " + feature_cols[i] + "\\n")

def decisionTree():
    global clf
    global X_train
    global y_train
    clf.fit(X_train,y_train)

```

```
text.delete('1.0', END)

r = export_text(clf, feature_names=feature_cols)

text.insert(END, "Selected number of features : "+str(r)+"\n")
```

```
def predict():

    global clf

    testname = filedialog.askopenfilename(initialdir="dataset")

    text.delete('1.0', END)

    with open(testname, "r") as file:

        for line in file:

            line = line.strip('\n')

            text.insert(END, line+"\n")

    test = pd.read_csv(testname)

    y_pred = clf.predict(test)

    text.insert(END, "\nRecommended Location for Tourist: "+str(y_pred)+"\n")
```

```
def graph():

    global fit

    height = [len(feature_cols), fit.n_features_]

    bars = ('Total Features', 'Selected Features')

    y_pos = np.arange(len(bars))

    plt.bar(y_pos, height)

    plt.xticks(y_pos, bars)
```

```
plt.show()
```

```
font = ('times', 18, 'bold')
```

```
title = Label(root, text='Recommendation System for Tourists')
```

```
title.config(bg='wheat', fg='red')
```

```
title.config(font=font)
```

```
title.config(height=3, width=80)
```

```
title.place(x=5,y=5)
```

```
font1 = ('times', 14, 'bold')
```

```
upload = Button(root, text="Upload Tourist Dataset", command=upload)
```

```
upload.place(x=50,y=100)
```

```
upload.config(font=font1)
```

```
pathlabel = Label(root)
```

```
pathlabel.config(bg='blue', fg='white')
```

```
pathlabel.config(font=font1)
```

```
pathlabel.place(x=300,y=100)
```

```
normal = Button(root, text="Run Preprocess & Features Selection Algorithm",  
command=featureSelection)
```

```
normal.place(x=50,y=150)
```

```
normal.config(font=font1)
```

```
decisionbutton = Button(root, text="Run C4.5 Decision Tree",  
command=decisionTree)
```

```
decisionbutton.place(x=50,y=200)
```

```
decisionbutton.config(font=font1)
```

```
predictbutton = Button(root, text="Recommendation for Tourists",  
command=predict)
```

```
predictbutton.place(x=50,y=250)
```

```
predictbutton.config(font=font1)
```

```
rungraph = Button(root, text="Features Selection Graph", command=graph)
```

```
rungraph.place(x=50,y=300)
```

```
rungraph.config(font=font1)
```

```
text=Text(root,height=25,width=80)
```

```
scroll=Scrollbar(text)
```

```
text.configure(yscrollcommand=scroll.set)
```

```
text.place(x=550,y=100)
```

```
text.config(font=font1)
```

```
root.mainloop()
```

```
#Entering the event main loop
```