INT404 – ARTIFICIAL INTELLIGENCE

PROJECT TITLE – OLD CAR PRICE PREDICTION

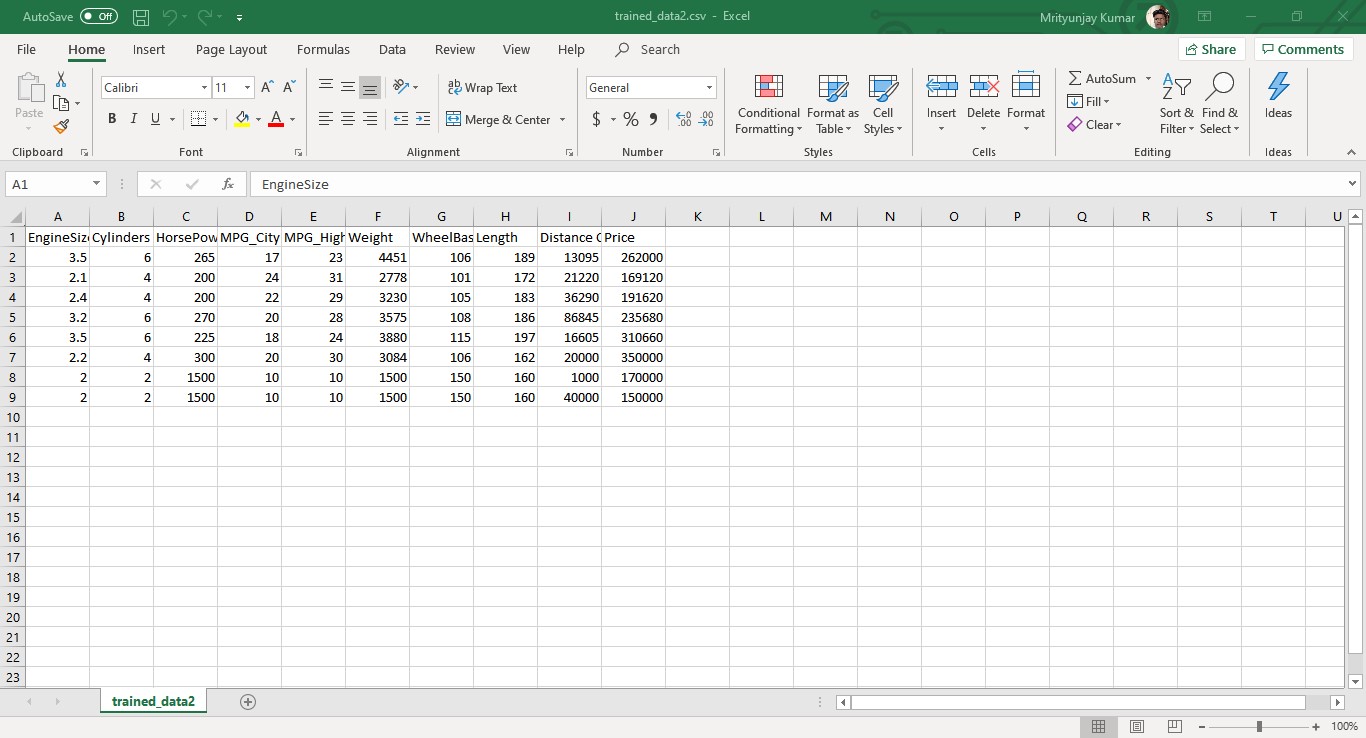
SUBMITTED BY :

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No | Name | Registration | Roll Number |
| 1 | Satyam Kumar Ray | 11809778 | RK18GEB63 |
| 2 | Nurul Haque | 11903516 | RK18GEB48 |
| 3 | Husaif Malik | 11806282 | RK18GEB56 |
| 4 | Asam Akash | 11806414 | RK18GEB57 |
| 5 | Mrityunjay Kumar | 11806307 | RK18GEB36 |

SUBMITTED TO :

Prof. Shabnam

Dataset Used in Project :

Column named Price Is – label

Rest others columns are – features

* In this project we used “Sklearn” packege to achieve prediction of old car price
* DecisionTreeClassifier used to classify the label and features

Code :

from tkinter import \*

from sklearn import tree

from tkinter import messagebox

from OCPP\_module import \* # this is my module i created it for multiple functionality

app = Tk()

app.geometry("1366x760")

app.title("Old Car Price Prediction")

#########################################################################################

#### work of pandas features and label

#########################################################################################

import pandas as pd

data = pd.read\_csv('trained\_data2.csv', header = 0)

features = []

for i in range(len(data)):

x = list(data.iloc[i])

x.pop()

features.append(x)

# for label

label = list(data.Price)

print(features, label)

#### main Working of prediction with sklearn

########################################################################################

def getdata():

global features

global label

############# let's put some constraint on input values so we get real informatin as we need

##### for first value

x1 = input1.get()

if(x1 == "" or float(x1) < 2.0):

messagebox.showerror("warning", "Please Enter valid input for no of engines it should be atleast 2!")

return

x1 = float(input1.get())

##### for second value

x2 = input2.get()

if(x2 == "" or float(x2) < 2.0):

messagebox.showerror("warning", "Please Enter valid input for no of cylinders!")

return

x2 = float(input2.get())

##### for third value

x3 = input3.get()

if(x3 == "" or float(x3) < 1500.0):

messagebox.showerror("warning", "Please Enter valid input for Horse Power it should be 1500 hp minimum !")

return

x3 = float(input3.get())

##### for fourth value

x4 = input4.get()

if(x4 == "" or float(x4) < 10.0):

messagebox.showerror("warning", "There is minimum mileage of 10 mpg!")

return

x4 = float(input4.get())

##### for fifth value

x5 = input5.get()

if(x5 == "" or float(x5) < 10.0):

messagebox.showerror("warning", "There is minimum mileage of 10 mpg!")

return

x5 = float(input5.get())

##### for sixth value

x6 = input6.get()

if(x6 == "" or float(x6) < 1500.0):

messagebox.showerror("warning", "minimum weight should be 1500kg !")

return

x6 = float(input6.get())

##### for seventh value

x7 = input7.get()

if(x7 == "" or float(x7) < 150.0):

messagebox.showerror("warning", "minimum wheelbase should be 150 cm !")

return

x7 = float(input7.get())

##### for eighth value

x8 = input8.get()

if(x8 == "" or float(x8) < 160.0):

messagebox.showerror("warning", "minimum length of the car is 160cm !")

return

x8 = float(input8.get())

##### for first value

x9 = input9.get()

if(x9 == "" or float(x9) < 0.0):

messagebox.showerror("warning", "Please Enter Distance covered by the car !")

return

x9 = float(input9.get())

print(x1, x2, x3, x4, x5, x6, x7, x8, x9)

############# Different Constraint

clf = tree.DecisionTreeClassifier()

clf = clf.fit(features, label)

print(clf.predict([[x1, x2, x3, x4, x5, x6, x7, x8, x9]]))

arr = clf.predict([[x1, x2, x3, x4, x5, x6, x7, x8, x9]])

print("arr = " , arr)

final\_result.delete(0, END)

final\_result.insert(0, arr[0])

#############################################################################################

######## Working On Frames Starts from Here

#############################################################################################

######## Frame no. 1

frame1 = Frame(app)

Heading = Label(frame1, text="OLD CAR PRICE PREDICTION", font = ("helvetica", 30 , "bold", "underline"), fg = "blue")

Heading.grid()

frame1.pack(pady = 40)

######## Frame no. 2

frame2 = Frame(app)

label1 = Label(frame2, text = " Engine Size ", font = ('helvetica', 13 ), justify = LEFT).grid(row = 1, column = 0)

input1 = Entry(frame2, font = ('helvetica', 20))

input1.grid(row = 2, column = 0, padx = 50)

label2 = Label(frame2, text=" Cylinders ", font = ('helvetica', 13 )).grid(row = 1, column = 1)

input2 = Entry(frame2, font = ('helvetica', 20))

input2.grid(row = 2, column = 1, padx = 50)

label3 = Label(frame2, text="Horse Power", font = ('helvetica', 13 )).grid(row = 1, column = 2)

input3 = Entry(frame2, font = ('helvetica', 20))

input3.grid(row = 2, column = 2, padx = 50)

label4 = Label(frame2, text = " MPG City ", font = ('helvetica', 13 ), justify = LEFT).grid(row = 3, column = 0, pady = 18)

input4 = Entry(frame2, font = ('helvetica', 20))

input4.grid(row = 4, column = 0, padx = 50)

label5 = Label(frame2, text=" MPG Highway ", font = ('helvetica', 13 )).grid(row = 3, column = 1)

input5 = Entry(frame2, font = ('helvetica', 20))

input5.grid(row = 4, column = 1, padx = 50)

label6 = Label(frame2, text=" Weight ", font = ('helvetica', 13 )).grid(row = 3, column = 2)

input6 = Entry(frame2, font = ('helvetica', 20))

input6.grid(row = 4, column = 2, padx = 50)

label7 = Label(frame2, text=" WheelBase ", font = ('helvetica', 13 )).grid(row = 5, column = 0, pady = 18)

y = StringVar()

input7 = Entry(frame2, textvariable = y, font = ("helvetica", 20))

input7.grid(row = 6, column = 0, padx = 50)

label8 = Label(frame2, text=" Length (Car) ", font = ('helvetica', 13 )).grid(row = 5, column = 1)

y = StringVar()

input8 = Entry(frame2, textvariable = y, font = ("helvetica", 20))

input8.grid(row = 6, column = 1, padx = 50)

label9 = Label(frame2, text=" Distance Covered (in Kms) ", font = ('helvetica', 13)).grid(row = 5, column = 2)

y = StringVar()

input9 = Entry(frame2, textvariable = y, font = ("helvetica", 20))

input9.grid(row = 6, column = 2, padx = 50)

# result Entry Tag

label\_final\_result = Label(frame2, text="Expected Price : Rs. ", font = ('helvetica', 25, 'bold')).grid(row = 7,column = 0)

final\_result = Entry(frame2, font = ('helvetica', 30 ))

final\_result.grid(row = 7, column =1, pady = 40)

button2 = Button(frame2, text = "Get Price", fg = "white", bg = "blue", font = ('helvetica', 25 ), command = getdata).grid(row = 7, column = 2)

frame2.pack(pady = 10)

###########################################################################################

############# frame3 (Help frame) starts from here

###########################################################################################

frame3 = Frame(app)

label\_discription = Label(frame3, text="This is the help Menu.", font=('helvetica', 20, 'bold'))

label\_discription.grid(row = 0, column = 0)

return\_home = Button(frame3, text="Help", command = lambda: HomeMenu(frame1, frame2, frame3)).grid(row = 1, column = 1)

button\_help = Button(frame2, text="Help", command = lambda: HelpBar(frame1, frame2, frame3)).grid(row = 8, column = 1)

app.mainloop()

Our Module for the Project

from tkinter import \*

from tkinter import messagebox

def HelpBar(frame1, frame2, frame3):

frame1.pack\_forget()

frame2.pack\_forget()

frame3.pack()

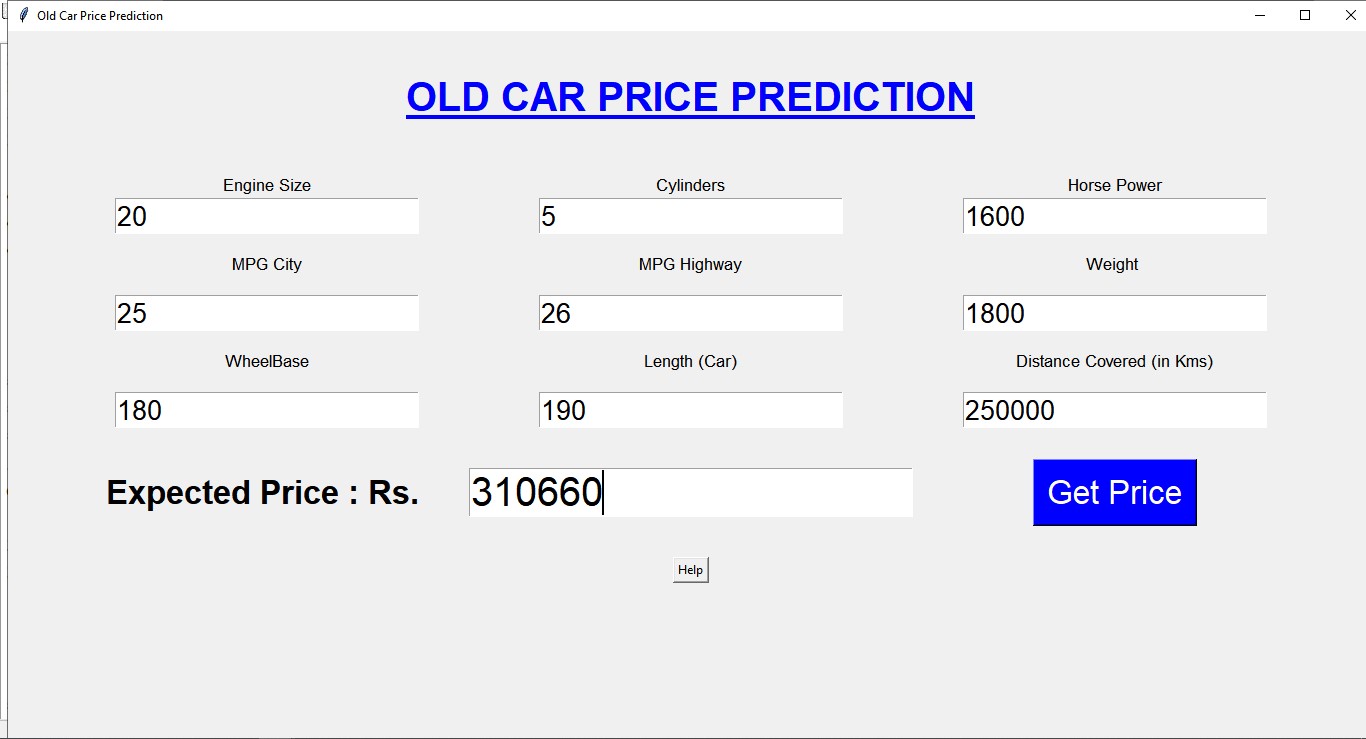
def HomeMenu(frame1, frame2, frame3):

frame1.pack()

frame2.pack()

frame3.pack\_forget()

output :



CONTRIBUTION IN PROJECT BY TEAM MEMBERS:

1. SATYAM KUMAR RAY & HUSAIF MALIK PREPARED THIS REPORT AND VIDEO FILE
2. ASAM AAKASH – DESIGN THE (OCCP\_MODULE.py) file which is used to change the frames of tkinter GUI and Tkinter Messagebox
3. Nurul Haque - Prepared the layout of the final output / designing the project
4. Mrityunjay Kumar - Code the project for Main Module (using tkinter) prediction work using Sklearn

Github Link :

<https://github.com/mrityunjay-1/Old-Car-Price-Pridiction-With-Machine-Learning>