

ML app code :

```
from sys import winver
from tkinter import Canvas
import joblib
import numpy as np
import locations
import tkinter as tk
from tkinter import ttk

#loading dropdown list values
OPTIONS = locations.location
OPTIONS2 = [1,2,3,4,5]
#loading ml model
sav = joblib.load("house_price.ml")
#creating tkinter instance
root = tk.Tk()
root.title("price prediction")
root.iconbitmap("icon.ico")
canv = tk.Canvas(root,width=250,height=240)
canv.pack()
#creating dropdown menu
variable = tk.StringVar(root)
variable.set(OPTIONS[0])
w = ttk.Combobox(root, textvariable=variable)
w['values'] = OPTIONS
canv.create_window(150,20,window=w)
variable1 = tk.StringVar(root)
variable1.set(OPTIONS2[0])
w1 = ttk.Combobox(root, textvariable=variable1)
w1['values'] = OPTIONS2
canv.create_window(150,60,window=w1)
#creating labels
loc_name = tk.Label(root,text='Location')
canv.create_window(50,20,window=loc_name)
bhk_name = tk.Label(root,text='BHK')
canv.create_window(45,60,window=bhk_name)
sq_name = tk.Label(root,text='Sq. ft')
canv.create_window(45,100,window=sq_name)
#creating input text field for getting sq.ft
in_sq = tk.Entry(root)
canv.create_window(150,100,window=in_sq)
#creating labels to show output
value1 = tk.Label(root)
canv.create_window(110,180,window=value1)
value2 = tk.Label(root)
canv.create_window(110,220,window=value2)
#submit button function
def cal():
    loc = float(OPTIONS.index(variable.get()))
    bhk = float(variable1.get())
    sq = float(in_sq.get())
    dumLoc = [1 if loc == x else 0 for x in range(240)]
    val = [bhk,sq]
    val.extend(dumLoc)
    test = np.array([val])
    result = sav.predict(test)
    r = int(round(result[0],0))
    value1["text"] = "Price per square feet : "+str(r)+" rs"
    value2["text"] = "Price of house : "+str(int(r*sq))+" rs"
#reset button function
def clear():
    value1["text"] = ""
```

```

value2["text"] = ""
#creating buttons
but = tk.Button(text='Submit',command=cal)
canv.create_window(80,140,window=but)
but1 = tk.Button(text='Reset',command=clear)
canv.create_window(150,140,window=but1)
#apply loop to app so it not close itself
root.mainloop()

```

Output :