In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

```

In [3]:

```
mpg=sns.load_dataset('mpg')
mpg
```

Out[3]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	n
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chev che m
1	15.0	8	350.0	165.0	3693	11.5	70	usa	sk
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plym sat
3	16.0	8	304.0	150.0	3433	12.0	70	usa	rebe
4	17.0	8	302.0	140.0	3449	10.5	70	usa	t
...	...	...	...	...	...	...	...	...	
393	27.0	4	140.0	86.0	2790	15.6	82	usa	mus
394	44.0	4	97.0	52.0	2130	24.6	82	europe	pi
395	32.0	4	135.0	84.0	2295	11.6	82	usa	d ram
396	28.0	4	120.0	79.0	2625	18.6	82	usa	ra
397	31.0	4	119.0	82.0	2720	19.4	82	usa	che

398 rows × 9 columns

In [4]:

```
mpg_data=mpg[['mpg','displacement','horsepower','weight']]
mpg_data
```

Out[4]:

	mpg	displacement	horsepower	weight
0	18.0	307.0	130.0	3504
1	15.0	350.0	165.0	3693
2	18.0	318.0	150.0	3436
3	16.0	304.0	150.0	3433
4	17.0	302.0	140.0	3449
...	...	...	...	...
393	27.0	140.0	86.0	2790
394	44.0	97.0	52.0	2130
395	32.0	135.0	84.0	2295
396	28.0	120.0	79.0	2625
397	31.0	119.0	82.0	2720

398 rows × 4 columns

In [5]:

```
mpg_data.isna().sum()
```

Out[5]:

```
mpg             0
displacement    0
horsepower      6
weight          0
dtype: int64
```

In [6]:

```
mpg_data.dropna(axis = 0, inplace=True)
```

```
F:\RC SLOG\Anaconda\lib\site-packages\ipykernel_launcher.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  """Entry point for launching an IPython kernel.
```

In [7]:

```
1 mpg_data.isna().sum()
```

Out[7]:

```
mpg             0
displacement    0
horsepower      0
weight          0
dtype: int64
```

In [8]:

```
1 target=mpg_data['mpg']
2
3 features=mpg_data[['displacement','horsepower','weight']]
```

In [9]:

```
1 target.head()
```

Out[9]:

```
0    18.0
1    15.0
2    18.0
3    16.0
4    17.0
Name: mpg, dtype: float64
```

In [10]:

```
1 features.head()
```

Out[10]:

	displacement	horsepower	weight
0	307.0	130.0	3504
1	350.0	165.0	3693
2	318.0	150.0	3436
3	304.0	150.0	3433
4	302.0	140.0	3449

In [12]:

```
1 X_train, X_test, y_train, y_test = train_test_split(features,target,test_size=0.2)
```

In [13]:

```
1 X_train.head()
```

Out[13]:

	displacement	horsepower	weight
352	98.0	65.0	2380
171	134.0	96.0	2702
148	116.0	75.0	2246
106	350.0	180.0	4499
167	97.0	75.0	2171

In [14]:

```
1 y_train.head()
```

Out[14]:

```
352    29.9
171    24.0
148    26.0
106    12.0
167    29.0
Name: mpg, dtype: float64
```

In [15]:

```
1 features.isna().sum()
```

Out[15]:

```
displacement    0
horsepower      0
weight          0
dtype: int64
```

In [16]:

```
1 model=LinearRegression()
2 model.fit(X_train,y_train)
```

Out[16]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fal
se)
```

In [17]:

```
1 model.coef_
```

Out[17]:

```
array([-0.00584122, -0.0465076 , -0.00507419])
```

In [18]:

```
model.intercept_
```

Out[18]:

```
44.527119619438025
```

In [19]:

```
pred = model.predict(X_test)
pred[:5]

```

Out[19]:

```
array([11.40061004, 29.85728045, 20.87940225, 30.75654722, 24.28959249])
```

In [20]:

```
y_test[:5]
```

Out[20]:

```
39     14.0
266    30.0
34     16.0
385    38.0
276    21.6
Name: mpg, dtype: float64
```

In [21]:

```
r2_score(y_test, pred)
```

Out[21]:

```
0.6430187463639594
```

In [22]:

```
import pickle
fp=open('mpg_df.pk1','wb')
pickle.dump(model,fp)
fp.close()
```

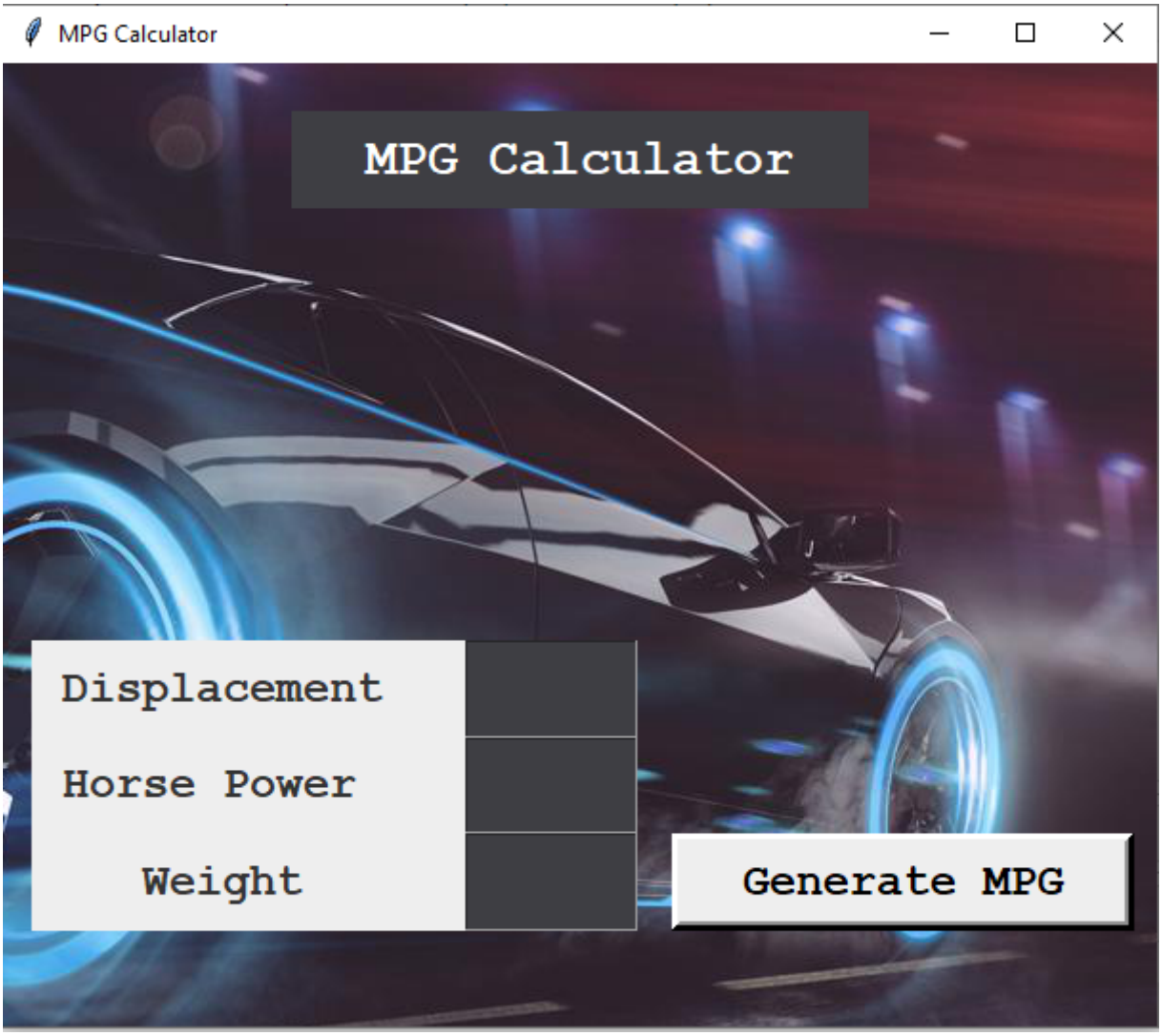
In [193]:

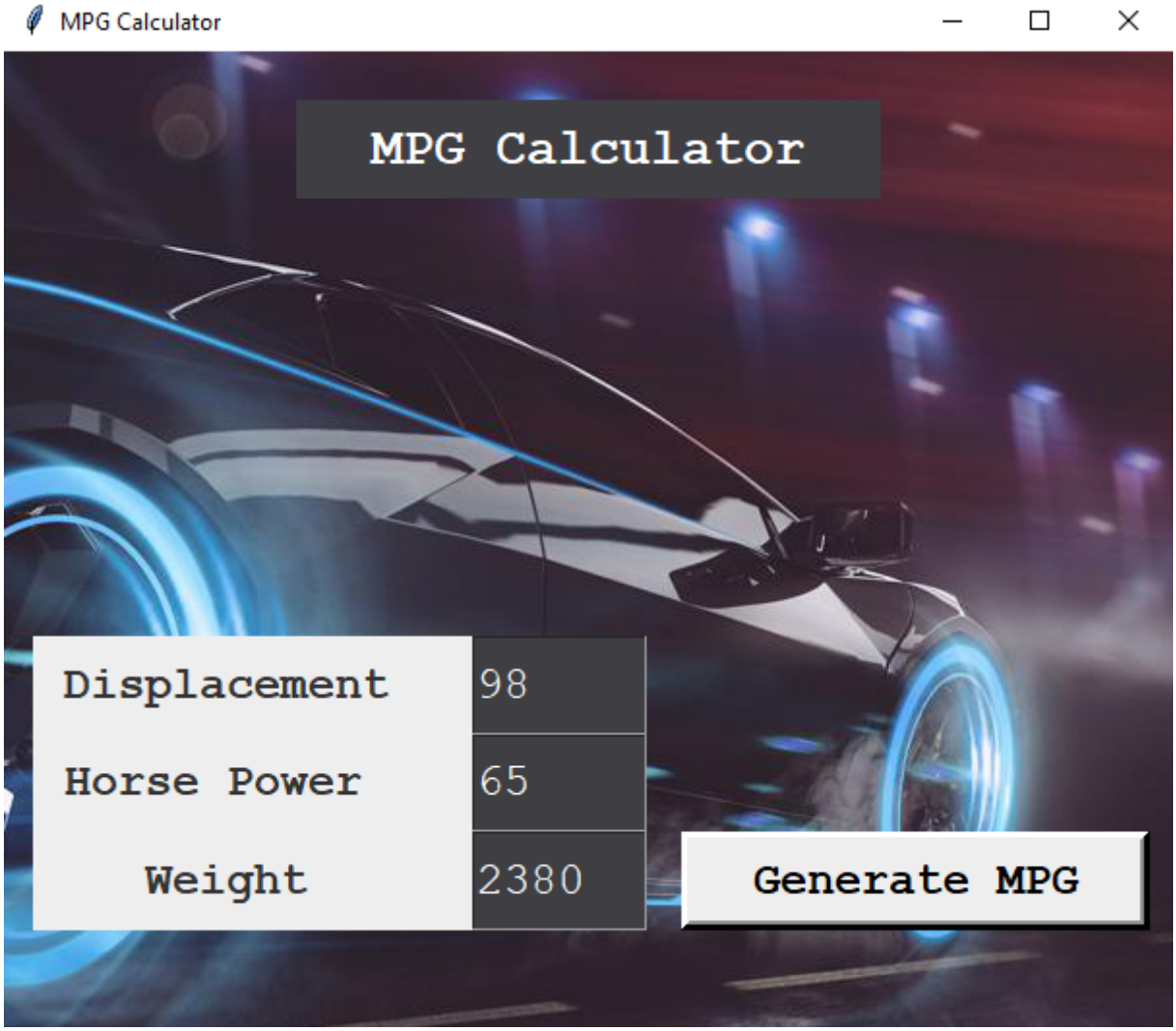
```
import tkinter as tk
import PIL.Image
import PIL.ImageTk

with open('mpg_df.pk1', 'rb') as fp:
    model = pickle.load(fp)
    fp.close()

root=tk.Tk()
root.title('MPG Calculator')

canvas = tk.Canvas(root, height=500, width=600)

im = PIL.Image.open("F:\RC SLOG\Python\car.jpg")
photo = PIL.ImageTk.PhotoImage(im)
background_label = tk.Label(root, image=photo)
background_label.place(relx=0, rely=0, relwidth=1, relheight=1)

canvas.pack()

dis=tk.IntVar()
hpower=tk.IntVar()
weight=tk.IntVar()

def clear():
    dis.set('')
    hpower.set('')
    weight.set('')
clear()

l=tk.Label(root,text='MPG Calculator', bd=2)
l.config(bg='#3E3E43',fg='#FFFFFF', font=('Courier', 20, 'bold'))
l.place(relx=0.5, rely=0.05, relwidth=0.5, relheight=0.1, anchor='n')

l1 = tk.Label(root, text="Displacement".center(20)+" : ")
l1.config(bg='#eeeeee', fg='#333333', font=('Courier', 18, 'bold'))
l1.place(relx=0.025, rely=0.6, relwidth=0.4, relheight=0.1)

e1 = tk.Entry(root,textvariable=dis,bg='#3E3E43',fg='#FFFFFF',font=('Courier', 18))
e1.place(relx=0.4, rely=0.6, relwidth=0.15, relheight=0.1)

l2 = tk.Label(root, text="Horse Power".center(20)+" : ")
l2.config(bg='#eeeeee', fg='#333333', font=('Courier', 18, 'bold'))
l2.place(relx=0.025, rely=0.7, relwidth=0.4, relheight=0.1)

e2 = tk.Entry(root,textvariable=hpower,bg='#3E3E43',fg='#FFFFFF',font=('Courier', 18))
e2.place(relx=0.4, rely=0.7, relwidth=0.15, relheight=0.1)

l3 = tk.Label(root, text="Weight".center(20)+" : ")
l3.config(bg='#eeeeee', fg='#333333', font=('Courier', 18, 'bold'))
l3.place(relx=0.025, rely=0.8, relwidth=0.4, relheight=0.1)

e3 = tk.Entry(root,textvariable=weight,bg='#3E3E43',fg='#FFFFFF',font=('Courier', 18))
e3.place(relx=0.4, rely=0.8, relwidth=0.15, relheight=0.1)

b1 = tk.Button(root, text='Generate MPG', command=lambda : predict())
b1.config(bg='#eeeeee', fg='black', font=('Courier', 18, 'bold'),border=5)
b1.place(relx=0.58, rely=0.8, relwidth=0.4, relheight=0.1)

```

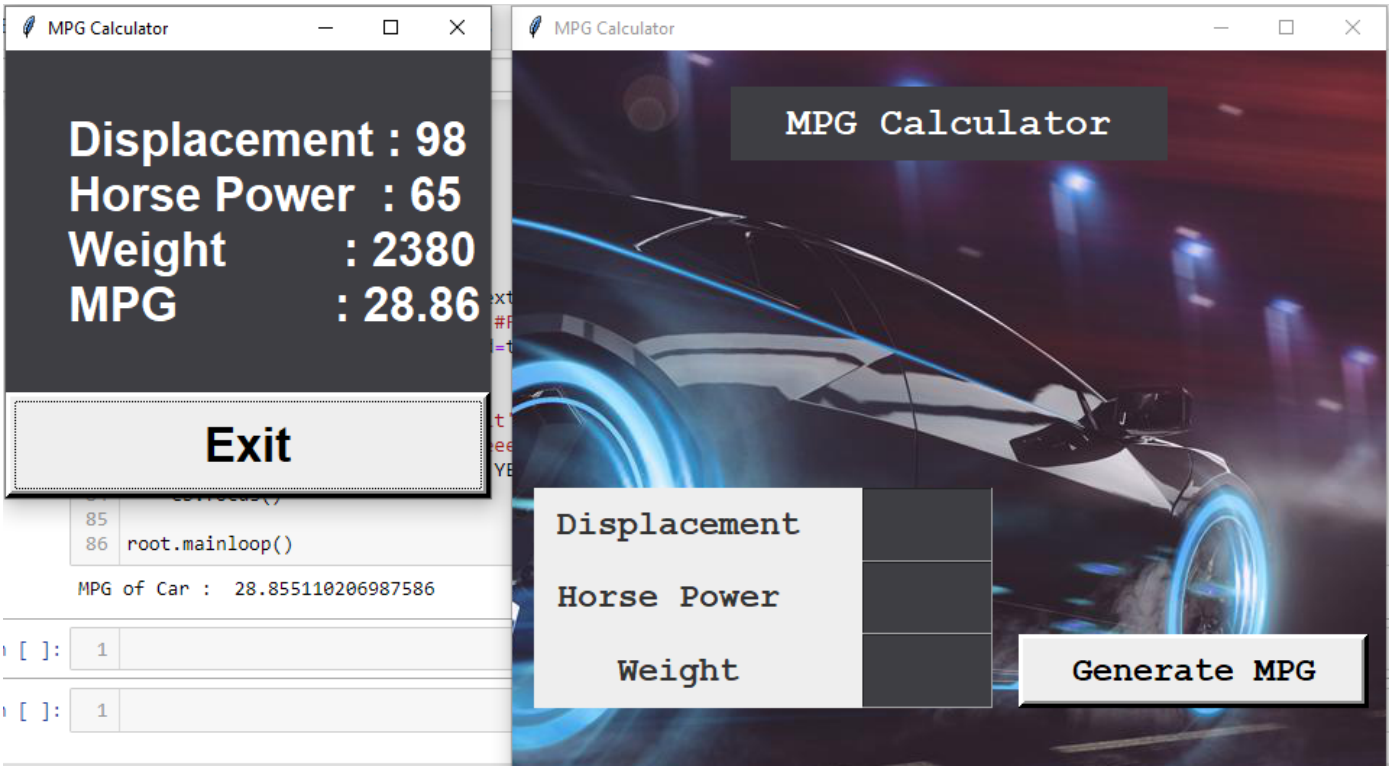
```
def predict():
    d = dis.get()
    hp = hpower.get()
    w = weight.get()
    features = [ [ d, hp, w ] ]
    clear()
    m = model.predict(features)[0]
    print("MPG of Car : ", m)
    win = tk.Toplevel(root)
    win.grab_set()
    text = f"""
    Displacement : {d}
    Horse Power  : {hp}
    Weight       : {w}
    MPG          : {m:.2f}
    """
    msg = tk.Message(win, text=text)
    msg.config(bg='#3E3E43', fg='#FFFFFF', font=('monospace', 25, 'bold'))
    msg.pack(fill=tk.BOTH, expand=tk.YES)


    eb = tk.Button(win, text='Exit', command=lambda : win.destroy())
    eb.config(fg='black', bg='#eeeeee', font=('monospace', 25, 'bold'),border=5)
    eb.pack(fill=tk.X, expand=tk.YES)
    eb.focus()

root.mainloop()
```

```
MPG of Car :  28.855110206987586
```

Output


MPG Calculator
MPG Calculator
Displacement
Horse Power
Weight
Generate MPG


MPG Calculator
MPG Calculator
Displacement
98
Horse Power
65
Weight
2380
Generate MPG


MPG Calculator
Displacement : 98
Horse Power : 65
Weight : 2380
MPG : 28.86
Exit
85
86 root.mainloop()
MPG of Car : 28.855110206987586
[ ]: 1
[ ]: 1
MPG Calculator
MPG Calculator
Displacement
Horse Power
Weight
Generate MPG

In [ ]:

```
1
```