In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv("Toyoto_Corrola.csv")
```

In [3]:
```
df
```

Out[3]:

	Id	Model	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	4	5	1165
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	4	5	1165
2	3	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	4	5	1165
3	4	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	4	5	1165
4	5	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	4	5	1170
...	...	...	...	...	...	...	...	...	...	...
1431	1438	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	20544	86	3	4	5	1025
1432	1439	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	19000	86	3	4	5	1015
1433	1440	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	17016	86	3	4	5	1015
1434	1441	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	16916	86	3	4	5	1015
1435	1442	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	1	110	5	4	5	1114

1436 rows × 10 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Id         1436 non-null   int64 
 1   Model      1436 non-null   object
 2   Price      1436 non-null   int64 
 3   Age_08_04  1436 non-null   int64 
 4   KM         1436 non-null   int64 
 5   HP         1436 non-null   int64 
 6   Doors      1436 non-null   int64 
 7   Cylinders  1436 non-null   int64 
 8   Gears      1436 non-null   int64 
 9   Weight     1436 non-null   int64 
dtypes: int64(9), object(1)
memory usage: 112.3+ KB
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Id           0
Model        0
Price        0
Age_08_04    0
KM           0
HP           0
Doors        0
Cylinders    0
Gears        0
Weight       0
dtype: int64
```

In [6]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
df['car_Model'] = encoder.fit_transform(df['Model'])
```

In [7]:
```
df
```
Out[7]:

	Id	Model	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight	car_Model
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	4	5	1165	257
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	4	5	1165	257
2	3	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	4	5	1165	365
3	4	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	4	5	1165	257
4	5	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	4	5	1170	256
...	...	...	...	...	...	...	...	...	...	...	...
1431	1438	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	20544	86	3	4	5	1025	5
1432	1439	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	19000	86	3	4	5	1015	8
1433	1440	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	17016	86	3	4	5	1015	8
1434	1441	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	16916	86	3	4	5	1015	8
1435	1442	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	1	110	5	4	5	1114	163

1436 rows × 11 columns

In [8]:
```
from sklearn.model_selection import train_test_split
x = df.drop(['Id','Price','Model'],axis = 1)
y = df['Price']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = .2,random_state = 42)
```

In [9]:
```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
x_train_scaled = sc.fit_transform(x_train)
x_test_scaled = sc.transform(x_test)
```

In [10]:
```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import GridSearchCV
```

In [11]:
```
model = KNeighborsRegressor()
```

In [12]:
```
d = {'n_neighbors': [3, 5, 7, 9], 'weights': ['uniform', 'distance'],'algorithm' : ['auto', 'ball_tree', 'kd_tree', 'brute']}
ch = GridSearchCV(estimator=model, param_grid=param_grid, cv=5, scoring='neg_mean_squared_error')
```

In [13]:
```
grid_search.fit(x_train_scaled,y_train)
y_pred = grid_search.predict(x_test_scaled)
```

In [14]:
```
from sklearn.metrics import r2_score
```

In [15]:
```
r2_score(y_pred,y_test)
```
Out[15]:
```
0.8597411606503029
```

In [ ]: