In [21]:
```
import pandas as pd
url='https://raw.githubusercontent.com/Fxisxl/Two-Wheeler-Price-Prediction/
df=pd.read_csv(url)
df.head()
```

Out[21]:

	Unnamed: 0	bike_name	price	kms_driven	age	brand	Owner_Type
0	0	TVS Star City	35000	17654	3	TVS	0
1	1	Royal Enfield Classic	119900	11000	4	Royal Enfield	0
2	3	TVS Apache RTR	65000	16329	4	TVS	0
3	4	Yamaha FZ S	80000	10000	3	Yamaha	0
4	5	Yamaha FZs 150cc	53499	25000	6	Yamaha	0

In [22]:
```
df=df.drop('Unnamed: 0',axis=1)
df=df.drop('Owner_Type',axis=1)
```

In [23]:
```
df
```

Out[23]:

	bike_name	price	kms_driven	age	brand
0	TVS Star City	35000	17654	3	TVS
1	Royal Enfield Classic	119900	11000	4	Royal Enfield
2	TVS Apache RTR	65000	16329	4	TVS
3	Yamaha FZ S	80000	10000	3	Yamaha
4	Yamaha FZs 150cc	53499	25000	6	Yamaha
...	...	...	...	...	...
31480	Hero Passion Pro	39000	22000	4	Hero
31481	TVS Apache RTR	30000	6639	9	TVS
31482	Bajaj Avenger Street	60000	20373	6	Bajaj
31483	Hero Super Splendor	15600	84186	16	Hero
31484	Bajaj Pulsar 150cc	22000	60857	13	Bajaj

31485 rows × 5 columns

In [51]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression

x=df.drop(columns=['price'],errors='ignore')
y=df['price']

categorical_variable=['bike_name','brand']
numerical_variable=['kms_driven','age']

categorical_transform=OneHotEncoder(handle_unknown='ignore')
numerical_transform=StandardScaler()

transforms=[('num',numerical_transform,numerical_variable),
                                        ('cat',categorical_transform,categ

preprocessor=ColumnTransformer(transforms)


model=Pipeline(steps=[('preprocessor',preprocessor),
                      ('Linear_regression',LinearRegression())])


x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_sta
print(model)
```

```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num', StandardScaler(),
                                                  ['kms_driven', 'age']),
                                                 ('cat',
                                                  OneHotEncoder(handle_unk
nown='ignore'),
                                                  ['bike_name', 'bran
d'])])),
                ('Linear_regression', LinearRegression())])
```

In [52]:
```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(25188, 4)
(25188,)
(6297, 4)
(6297,)
```

In [55]:
```
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
```

In [56]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
mae=mean_absolute_error(y_test,y_pred)
mse=mean_squared_error(y_test,y_pred)
rmse=np.sqrt(mean_squared_error(y_test,y_pred))
```

In [57]:
```
print(mae,mse,rmse)
```

```
5423.632171058705 88432589.5563807 9403.860353938733
```

In [72]:
```
df.head()
```

Out[72]:

	bike_name	price	kms_driven	age	brand
0	TVS Star City	35000	17654	3	TVS
1	Royal Enfield Classic	119900	11000	4	Royal Enfield
2	TVS Apache RTR	65000	16329	4	TVS
3	Yamaha FZ S	80000	10000	3	Yamaha
4	Yamaha FZs 150cc	53499	25000	6	Yamaha

In [89]:
```
bike_name=str(input("enter bike name to be sold : "))
kms_driven=eval(input("enter the kms driven : "))
age=eval(input("enter age of bike : "))
brand=str(input("enter brand of bike : "))

new_data = pd.DataFrame({
    'bike_name': [bike_name],
    'kms_driven': [kms_driven],
    'age': [age],
    'brand':[brand]
})

print(new_data)

predicted_data=model.predict(new_data)
print(predicted_data)
```

```
enter bike name to be sold : Yamaha FZ S
enter the kms driven : 56221
enter age of bike : 7
enter brand of bike : Yamaha
     bike_name  kms_driven  age   brand
0  Yamaha FZ S       56221    7  Yamaha
[50813.99975602]
```

In [ ]:
```
```