```
import pandas as pd
df = pd.read_csv('/content/car data.csv')
```

```
df.head()
```

	Car_Name	Year	Selling_Price	Present_Price	Driven_kms	Fuel_Type	Selling_ty
0	ritz	2014	3.35	5.59	27000	Petrol	Dea
1	sx4	2013	4.75	9.54	43000	Diesel	Dea
2	ciaz	2017	7.25	9.85	6900	Petrol	Dea
3	wagon r	2011	2.85	4.15	5200	Petrol	Dea
4	swift	2014	4.60	6.87	42450	Diesel	Dea

```
df.shape
```

```
(301, 9)
```

```
df.isnull().sum()
```

	0
Car_Name	0
Year	0
Selling_Price	0
Present_Price	0
Driven_kms	0
Fuel_Type	0
Selling_type	0
Transmission	0
Owner	0

dtype: int64

```
import seaborn as sns
sns.heatmap(df.corr(), annot = True)
```

<Axes: >


```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```
df.head()
```

	Car_Name	Year	Selling_Price	Present_Price	Driven_kms	Fuel_Type	Selling_ty
0	ritz	2014	3.35	5.59	27000	Petrol	Dea
1	sx4	2013	4.75	9.54	43000	Diesel	Dea
2	ciaz	2017	7.25	9.85	6900	Petrol	Dea
3	wagon r	2011	2.85	4.15	5200	Petrol	Dea
4	swift	2014	4.60	6.87	42450	Diesel	Dea

```
df.head()
```

	Car_Name	Year	Selling_Price	Present_Price	Driven_kms	Fuel_Type	Selling_ty
0	ritz	2014	3.35	5.59	27000	Petrol	Dea
1	sx4	2013	4.75	9.54	43000	Diesel	Dea
2	ciaz	2017	7.25	9.85	6900	Petrol	Dea
3	wagon r	2011	2.85	4.15	5200	Petrol	Dea
4	swift	2014	4.60	6.87	42450	Diesel	Dea

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder,  StandardScaler

label = LabelEncoder()
df.rename(columns = {'Selling_Price' : 'SP'}, inplace = True)
df['Fuel_Type'] = label.fit_transform(df['Fuel_Type'])
df['Selling_type'] = label.fit_transform(df['Selling_type'])
df['Transmission'] = label.fit_transform(df['Transmission'])
df['Car_Name']  = label.fit_transform(df['Car_Name'])
X = df.drop(columns = ['SP'], axis = 1)
Y = df['SP']
sc = StandardScaler()



from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score

model = LinearRegression()
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2, random_state=2)
X_train.iloc[:,1:4] = sc.fit_transform(X_train.iloc[:,1:4])
X_test.iloc[:,1:4] = sc.transform(X_test.iloc[:,1:4])
model.fit(X_train, Y_train)
predictions = model.predict(X_test)

error = mean_squared_error(Y_test, predictions)
print(error)
print(r2_score(Y_test,predictions))

plt.figure(figsize=(10,6))
fig,ax = plt.subplots()


plt.scatter(Y_test,predictions)
plt.plot(Y_test,Y_test,c = 'r')
plt.show()
```

2.932940052097884
0.8403102277417828
<Figure size 1000x600 with 0 Axes>