the problem statement is that we have to predict the car prices of cars okay so we have several information about those used cars and this particular data set contains car brand so what is the brand of the car and what is the year of that particular model and at what price that particular car is sold and the present price of that car and the kilometers driven fuel type whether it is it runs on petrol dslr compressed natural gas those kind of things seller type whether the seller is an individual or a dealer in the transmission whether the gear system is automatic or manual and then the number of owners of that particular car so these are the data set features so using these uh data set using these features in the data set we need to train our machine learning algorithm that can find patterns in the data set so once we give uh give new information it can predict at what price that particular used car can be sold so this is our problem statement okay so i hope you have understood what we are going to do now let me explain you the workflow we have for this particular project first is data collection so we need this car data so the data , we need this data to feed this to our machine learning model so

particular data set we have this car data set so we need to pre-process the data so we cannot feed this raw data to our machine learning algorithm so for it to understand the data so we need to do some processing so this step is called as data pre-processing okay so once we process the data we will split this data into training data and

testing data okay so this step is called as train test split okay so once we have splitter this data we will train our machine learning algorithm with training data and we will test or evaluate our machine learning model with this disk data okay

so we will feed this to our regression model okay so in this case we are going to use two regression models so first we will train our uh machine learning model with the linear regression and then we will use

the lasso regression model and we will compare the accuracy score or the error percentage of these models and see

which model performs better for this particular data set okay so this is a regression example so

the previous videos which we have seen are basically classification problems so in the case of classification problems

we will predict in to which class the particular data type belongs say for example we have made a

project on diabetes prediction so in that case our machine learning model will predict whether a person has diabetes or not

so it is like class it uh just predicts its yes or no or its zero or one those kind of uh

problems are known as classification problems this is a regression problem because we are predicting a particular value okay

so we are not just predicting classes we are just predicting particular value on which the car can be sold so this is

known as regression problems so we will be using linear regression and then we will be using our lasso regression and compare it okay

so once we have trained our model with this uh regulation model with this training data

we will evaluate our model then we will have a drained linear regression or lasso regression model

then we will feed new data to our machine learning model so on understanding this data or

on operating on this data our machine learning can predict the car price on which it can be sold okay so

now let's get into the coding part of it so before getting started with the

coding part i'll just give you a short walkthrough on my youtube channel so in my channel i'm making a ants on machine learning

course with python so you can see the course curriculum video here so in this video i have explained what are the different modules

and videos that i will be posting in this video in future so you can download the course curriculum from here so it

contains various modules and topics which i will be making so i will be posting videos on monday evening and

wednesday evening and every friday evening i will be posting a machine learning project video okay so you can go to this playlist

option here so here you can see various modules in my course so the first module is machine

learning basics module so and the second module is python basics module so if you are new to machine learning you can start watching

from this module and the third is important python libraries where i will be explaining about numpy pandas and other libraries

so i will be posting the remaining videos for these modules on monday and wednesday evening

and as i have told you earlier friday there will be a machine learning project video so you can go to this machine learning project playlist

so you can see here so totally there will be

eight project videos including the one that i'm recording okay so you can watch all these project videos

okay so this is about my channel do subscribe and stay connected for

future videos okay so now let's get started with today's topic so this is car price prediction so this

environment is called as google collaboratory so google collaboratory is very useful for running python programs okay so if

you are new to google collaboratory or if you haven't heard about it you can check out my google collaboratory basics

video so it will be in the module 2 so the name of the video is google

collaboratory basic so the index of that video is 2.1 okay so you can find that in my channel so

now let's get into this you can see here i have already uploaded this car data set okay

so you can download this data set i'll give the link of this file in the description of this video so once you have downloaded it

so you need to upload it to this google collaboratory environment okay so you can go to this option so upload to session storage now you can

upload this data set file okay now let's get started with the code the first step is

to import the dependencies okay so once you're once you've opened your google collaboratory you need to connect your

system from here okay so first step is importing the dependencies

okay so you can press shift plus enter to run this cell and go to the next cell okay

so first of all let's import pandas

so input pandas as pd okay so next let's let's input

map plot link lib dot pie plot as plt and then let's input

c bond as sns so pandas is basically used to make a data frame so data frames are

more structured table or forms so we cannot so the data set we have is a comma separated value so when we feed this to

this data frame it is more easier to analyze and do some processing on the data set so that's why we are using this

pandas library and this matplotlib library and c-bond library are very useful for making parts

so then let's import from sklearn dot model selection

let's import train test split so as i have told you earlier so we need to

split our original data into training data and test data okay so this function is useful for that

now let's import our linear regression model

so from from sklearn dot

linear model input

linear regression okay so first we will train our data with this linear regression model

then we will train it with lasso regression okay so these two are one of the most important uh so

important regression models in machine learning so from sklm dot linear model

let's import lasso okay now let's input from

scalar input matrix so this matrix module is used to

evaluate our model okay to find the error score accuracies etc so matrix so you can run this by

pressing shift plus enter okay so this will load all the libraries and the functions we need so as we have done that next we need to

collect our data so this data is collected from kaggle so you can just search car price

data set kaggle so you will be shown the pages of kaggle in google so from kaggle you can download this

data okay so i'll also give the link for it so this step is data collection and

processing so i'll make a text here data collection and

processing okay so now we need to load our csv file so

as you know we have the data in a csv file right now we need to load this uh data present in the csv file to a

partners data frame as i have told you earlier pandas is a tabular form and if we load that to

this tables it is very easy to analyze and process the data okay so we are going to just copy this

path so you can copy the path from here okay now

loading the data

from csv file to pandas data frame

okay so i'll declare a variable called as car dataset so this will be

the name of my data frame so we have this car data set so you need to use the function pd dot

read csv okay so in this parenthesis you need to

copy and paste the path that we have copied okay so as you can see here i am using this

pandas library so you can see here that i have imported pandas in the short form spd so this pandas has a function read csv

so using this function we will be loading our csv file to our data frame okay so i'll run this

this will load our data now let's check the first five rows of our data frame

let's check the sample of this data frame so inspecting the first five rows

of the data frame so you need to mention the data frame

name so as you can see here the data frame name in this case is car dataset so

let's mention it car data set dot yet so this yet

function will print the first five rows of our data frame okay

so as you can see here we have the first five rows so in the first column we have car name so the brand of the car on the

year of that particular model and the selling price as we are discussing about the used cars so these are the selling price of various

cars and the present price the present price of new car and the kilometers driven by that car fuel take whether it is petrol diesel or

compressed natural gas and cellular type whether the seller is a dealer or a individual then we have the transmission whether

the gear system is manual or automatic and we have the number of owners before that particular person okay so these are

the nine columns or nine features we have in this data stream okay now let's see how many total

data points we have okay so checking the number of

rows and columns so basically we are just taking checking the number of data points we have in the entire data set so

mention the data frame name which is car data set dot shape so this shape function will give us the

number of rows and columns so totally we have three not one rows and nine columns so we have nine columns here

and totally we have three not one rows so three not one rows means there are three not one values of different cars

so what we are basically trying to do is we will train our machine learning model with all these selling prices and other

features of the car and once it has learned from this data when we give new information

like the kilometers driven fuel type etcetera the machine learning model can predict how much price the car can be sold okay

so this is what we are going to achieve in this particular case so as you can see here it has the year

so if the car is very old the price will be somewhat less and if the kilometers driven is

very high then a person won't buy it for a high price right so these are

several features on which the price depends right so now let's inspect more about this data set

so what i'm going to do is get some information about this dataset

getting some information

about the data frame or data set so mention the data frame car data set

dot info so this will give us some information about the data right about the objects we have so as

you can see here we have totally three not one entries uh so zero two three not one and we have totally nine

columns okay so these are the nine columns we have and we have three not one non null values that means uh

totally we have three not one three naught one entries and three not no none null values sorry non null values that means no

values are missing okay so we also have this uh data type we have so it is a 64-bit integer this is

float and we have this object kind of thing so these objects are nothing but categorical data so as you can see here

these are some text data so these are categories for example in the fuel type the categories are petrol diesel and cng

in the cellular type the categories are dealer and individual okay so and in transmission it's manual

and automatic such kind of things okay so those are given as object so now what

we will do is it's again so there is another function for checking particularly the missing

values so as we have seen here there are no missing values here but it is very important function you need to know

so i will mention it how you can find the number of missing values so checking the number of

missing values okay so car data set

dot is null dot sum so this function will tell us how many missing values are

there on each column okay so as you can see here we don't have any missing values so if

we have missing values there are also some steps that we need to do like imputation to replace those missing values with

some suitable values okay so totally we have nine columns and there are no missing values in this

particular dataset so it is a very good thing so we can proceed with further uh process that we need to do

now what we shall do is let's check the number of values for this categorical data say for example

in this fuel type we have petrol diesel and compressed natural gas right so we are going to check out of this 3.1

entries how many cars are petrol cars how many cars are diesel cars etc and how many cars as a

dealer has their seller type and how many individuals are there okay so those kind of things so we will be checking that for this

fuel type cellular type and transmission okay so transmission will be manual or automatic so i'll show you how you can do that

so checking the distribution

of categorical data so we are going to see how these these

categories are distributed in this entire dataset so i'll print this so print mention the

data frame name which is car dataset dot now you need to mention

your column names so we are going to check this for fuel type cellular type and transmission so so car data set dot

fuel type so fuel type is one column so we

shouldn't get parentheses here so if you will type dot value counts so this will count the

values for petal and the number of values for dsl okay so value counts so this will give us the number of

values for petrol and diesel and let's do the same

thing for data set now let's check this for

so this seller type okay so

select type dot value comes

okay now thirdly we are going to do this okay so again okay

so now let's check the same for the transmission

our dataset dot transmission dot value counts

so this will tell us the number of values so let me run this as you can see here out of the 3.1

values 239 values are of petrol cars and there are 60 diesel cars and only two cars are of uh

complex natural gas type okay and in the case of dealers we have uh sorry in the case of seller type

so which is the second one here so in the case of uh seller type we have dealer so 195 dealers

and 106 individuals okay and in the case of transmission there are totally 261

manual transmission cars and 40 automatic transition cars okay so it is a very good estimation so

it tells us how our data is estimate or our data is distributed in our entire dataset okay

now let's uh encode this data okay so as you can see

here the data in this particular columns are in what should i say they are in text right

so but our machine learning model cannot understand text properly so what we are going to do

is we are going to convert this text into some numerical values so what we are going to do is so we have this petrol value right so

wherever there are petrol we will change that to zero and wherever there is diesel we will change this to one

and if there is cng we will change the to two so those kind of things and for dealer we will put a value as zero

and for individual we put the value as one and the same for manual and automatic transmission so this is known as encoding the data okay

so if we put those numeral numerical values it is easy for our machine learning model to understand it

okay so because computers better understand numbers rather than text okay so we are going to change these values to appropriate

numbers so this step is called as

encoding so let me put a text here as

encoding the categorical data because these particular columns

contains categories so categorical data so first let's encode

the fuel type column so encoding

fuel type column

okay so car data set

we can use the function replace to replace all the values with suitable values we want so the

column name is nothing but fuel type right so mention it here so you need to mention it inside the

codes fuel type and for fuel type i want

for petrol i want the value to be zero okay and for diesel

i want the value to be one okay for diesel the value should be one and

for c and g so you can see the values here so we have c and g and for c and g

i need the value to be two okay so for petrol all the values of petrol will be

transformed to zero for dsl1 and cng it's two okay so and then you need to mention in place

is equal to draw so we want to change them to this to this value so we

need to mention this in place parameter then let's encode the same for the cellular type okay so i'll just

copy this

so we need to do this for three columns so i'll just paste it here so first we have did this for fuel type column now let's

do this for seller type okay so for seller we have manual sorry dealer and individual right

seller and in the case of seller

we have individual and dealer so if the value is dealer i want to

change it with zero and for individual i need to change it with one

individual so you can see that we have this individual parameter

okay so individual is one we can remove this because there is no third category okay so for dealers all the dealer's

board will be replaced with zero and all the individual will be replaced with one okay so in place is equal to two

and next is the transmission type okay so encoding transmission column

so transmission column replace

transmission okay so in transmission let's put 1 0 for manual transmission and

one for automatic transmission so the two values are nothing but manual

and automatic so there is no third value so i'll just delete this

okay so now let's run this and see so this will successfully encode our

data into numerical values so all this categorical data like petrol diesel um individual

or dealer will be transformed to respective categories or respective numerical values so now what

i'll do is i'll just again print the first five rows of the data set to check whether the encoding has done properly

so mention the data frame which is car data set dot yet okay so i'll run this

as you can see here now we have the values such as 0 and 1. so 0 represents petrol and 1 represents diesel

and in the seller the 0 represents a dealer and 1 represents individual and transmission also 0 represents manual

and 1 represents automatic so you can see the previous values here so all these values will be transformed into their respective

numerical labels okay so this is one of the important step in this case of converting this text data

into suitable numbers okay so now what we will do is so as we have discussed before the next step is

splitting our original data set into training data and text data sorry test data okay

so make a text here as splitting the data into

training data and test data okay so let's create two variables

x and y so in excel store all the features i want so this selling price

becomes our target because we are just going to predict this target right so this price so this becomes our target

and all the other uh features becomes uh becomes our data okay so in this data

what we need to do is we need to uh remove the selling price and store it in the in y and all the remaining features will

come in this x variable okay so we also need to remove this car name because it cannot be used well for our prediction

so for our prediction we will be using the year on which it is bought so the kilometers it has driven that what

is the present price of it if you will type in those kind of things okay so we need to remove this selling price and car name okay so

we will put this selling price in y so let me show you how you can remove these two columns that is car

name and selling price okay so x is equal to car data set so in this car data set i

want to drop two columns so these two columns are nothing but car name

and we have selling price

so this car name is not useful for our prediction so i'll remove this and in axis

i need to mention one so whenever i am dropping a column i need to mention the axis value as one if i am

dropping a row i need to mention the axis value as zero okay so now i need to store this selling

price values in y okay so y is equal to car data set dot sorry not that it's car

data set and you need to mention the column name which is selling price so this will load all the

values of this data set except car name and selling price to x and this particular value selling price value will be

stored in y okay so this is in lags rupees okay so

3.35 means 3 lakh 35 000 indian rupees okay for so 4.75 means 4

lakh 75 000 indian rupees so such kind of thing so that is the unit of this price okay now let's run this

and let's see by printing it okay so something is wrong here selling price not phone okay so as you

can see here price should be in p should be in caps here so i just made a mistake here

so i'll change this now so this will successfully separate our

data and that target variable so let me print x and y for you so this

x contain all the values except the selling price okay so this selling price will be removed from our

data and it will be stored in the y variable okay so i'll print y now so we have printed x here now let

me print y so print y so this will give us the target variable which is

the price of the cars which we want to analyze okay so now what we need to do is

as we have splitted the data into x and y now we need to split the data into training and test

data okay let me change this to splitting the data and target so

splitting the data and target okay so here let's split it into

splitting training and test data

okay so we need to mention four variables here one is x train and x test

and y train and y test okay so let me explain you what are these

four variables okay so we have this data and uh target which is priced separately

right so i want to separate this data into training data test data

so what i'm going to do is i'm going to put all the training data in the variable

called as x train and all the testing data will be stored in the six test and the labels are sorry the price of

all the values in x strain will be stored in y trade and all the price values for the

x test will be stored in y test okay so this extreme contains the data of training

training and extest contain the price of the training data and y train contains the training data

of all these data and why test contains the corresponding selling price value okay

so now we need to use the drain test split function we have imported so as you can see here we have the strain test

split function which we have imported from sklearn.model selection so i will call this function

train test split and here we need to mention x and y okay so this x and y are nothing but

so what we have splitted here so that x represent all the data and y represents the price okay so x and

y so x comma y and let's say the test size

is equal to 0.1 so 0.1 means 10 percentage so 90 percentage of our data will be

training data and 10 percentage of our data will be testing data okay so the data set we have is very small it is uh

it has only three not one data points so we are just taking a 0.1 percentage as test data so if we have a large data

set we can have 20 percentage of data okay typically machine learning we we also deal with data sets containing

thousands of values ten thousands or values or even sometimes lacks or uh those magnitude of values okay so if

your data set is large then your prediction will be better okay so that's one important thing to note here this is uh

a simple data set okay so hence we are just having point one test size which is ten percentage of the original data

and now we need to mention random state so ran and put random state is equal to

two say for example if you want to split the data in the same way that i have splitted you need to put the random state as 2.

so if you put this random state is equal to 3 your data will be splitted in some other kind okay so

the splitting will not be similar to what i am doing okay so if you want to split the data into

the same way that i am doing so you can put this random state okay so this is not that important so this is basically

to you know reproduce the code so these are the parameters we have that x and y test

size and random state so this will split our data into training data and test data and their respective sell to price values which will be

stored on y train and y test so let me run this so so we have successfully splitted our

training data and test data now let's train our model so this part

is called as model training so first let's use a linear regression model for our

predictions so one is

linear regression okay so we need to load our linear

regression model

the linear regression model so if you remember we have imported our

linear regression model from sklearn dot linear models right so from that i am importing this linear regression which i

am using here so let's import it so let me create

the variable as linear regression so short form as lin dot tray so this is

linear regression model so linear regression model is equal to

so this is how you can load this particular model to a variable okay so here the variable is

nothing but lean rig model okay so let me run this now we can fit our data to this linear regression okay

so the data here is nothing but the x strain and the respective uh values is nothing

but y train so we need to fit this model with x train and y train okay so

min break model dot fit so this is the important function so this is used for the fit function is used to train our

model so here you need to mention x train and white ray so y train are

their respective price values so let me run this

okay so this will train our linear regression model it will find the equation of that linear regression model

now let's evaluate our model so linear regression is just a line okay so the equation is nothing but y is

equal to mx plus c so it is just a normal linear function okay so now let's evaluate our model

about its error score model evaluation

okay so first let's predict using our model on our training data

prediction on training data so for that you need to

use the function dot predict so i'll create or i'll declare the variable as

training data prediction

which is equal to lin so you need to mention the model which is lindrick.model and here you need to use this predict

function so this will predict the price values based on the other data we are giving

okay so extreme

so the models prediction or the here the model will predict the price of all the values present in this x train and it

will store all the values to this training data prediction variable okay so i'll run this so now we need to compare

the values predicted by our model with the original values here the original values are nothing but y train right so let's compare and see

how our model is performing so for regression we use various metrics

such as r squared error mean absolute error root mean square error etcetera okay so in this case i will show you how you

can use r squared and so this r square error is one of the important metrics so

r squared error let's create a variable as error score so error score

is equal to so if you remember we have imported this matrix function from sklearn

matrix dot r2 score

and in that we need to compare the original price values which are y train and the value is predicted by our model so here

you can see this is the original value is y train and the values predicted by our model stored in

training data prediction okay so we need to mention both of them so it will compare the values predicted where model and the original values

so y train and

training data prediction now let's print our error score so the error score should be

as low as possible r squared error

so here i want to print this so here i'll mention the error score

okay so let's run this so we got the error score as 0.879 so it is

kind of good for our particular case so uh there is not a definite value that

the r square should have so we cannot say that if the value of the r square is 0.5 then the model is predicting correct

or not so the value of r squared error depends on each uh

problems we are solving so it depends on the magnitude of the value we are using here the magnitude is in

uh like four point seven point such kind of things so it all depends on uh what particular

magnitude we have okay so we cannot just give a generalized value on like the error square can be uh this

value say for example for accuracy score we can tell that if the accuracy score of our model is

more than 75 or more than 80 then our model is performing well but we cannot say the same in the case

of r squared okay so we use accuracy score for classification problem and in the regression case we

use r squared error so the uh another important way to you know understand about the

performance our model is to plot the values predicted by our model and uh

plot the values predict uh plot the values of original prices okay so i'll show you how you can do that so

this will tell us how close the values are so now we are going to

visualize the actual prices

and predicted prices okay so these values should be as close as

possible predicted prices okay so plt so pl test nothing but

matplotlib dot pipe plot so we have imported it as plt so plt dot scatter so let's uh plot a

scatter plot so here let's mention y train and training data prediction

okay so these are the two values that we want to uh plot

plt dot x label so we need to give ix label let's have our x label as actual prices so

actual prices are nothing but white rate

sorry actual price and we have the y label as plt dot y label

the y label is nothing but the predicted price okay and let's name our

plot as plt dot title

actual prices versus predicted prices

now you need to put this plt dot show function to plot this graph so it will print our graph plt dot show

so let's run this so this will tell us how close the values are as you can see here there is not much

distance between the values here so most of the values are closer to each other so once uh the price value increases

there is some gap because the price of the cars are in the range uh you know or within the range 10 right

there are very less values greater than 10 that's why there are very less values in this case but as you can see here the values are as flows

are very close so that means the values predicted by our machine learning model is very close to the original sold

price okay so this is how you can estimate your model performance by plotting the prediction and the original values

in the case of scatter plots okay so now what we will do is let's do the

same but in this case let's uh evaluate our model in the case of test data so as i

have told you earlier we will train our machine learning model with training data and test it with test

data okay so now let's do that we just need to uh do the same thing again so i'll just copy this part we just need

to put some differences here so

and paste it yes you you just need to change this to test data so previously we have predicted for

training data now let's uh predict it for test data so here the

test data will be x test right okay so it's x test so totally testing data

contains 10 uh percentage right say for example uh in this case we have three not one data

points so ten percent of it is uh thirty values okay so these thirty values will be

predicted and it will be plotted in the graph so that's what we are going to do and let's also find the r squared error

so i'll run this so this will predict the price values of the all the values in this x test so r

squared error and in this case i want to compare with test data prediction and y test so

remember that we are comparing sorry we are working on test data right so the values

for all the test data the price values is stored in y test okay so we need to compare this y

test and test data prediction okay so we are just doing the same thing that we are we have done here but we are

just changing the training into test okay so i'll run this and see what is the r square value

okay it's 0.836 so it is a very good value so as you know if your value r squared

value is less in this case then the model performance is good okay now let's also plot the graph

and see so this will just plot for those values okay so we just need to make some changes so

here we need to mention y test and training data prediction

okay so let's run this so this is for test data prediction right so as you can see here here we have just

only 30 values that's that's why there are not much values here and they actually lie in in in the same line right so there

are there is not much difference uh or there is not much distance between them so but if you have more values in your

training and test data then we will get better predictions okay so this is not a bad prediction so this

is kind of good on this 30 data points okay so in this case we have linear

regression models right so now let's use lasso regression and see how this is performing so generally

lasso regression performs uh better on most of the cases so linear regression performs well on those cases

which are directly correlated or positively correlated say for example if one value increases the other value

also increases then in that case they are directly proportional right the two variables are in those cases the linear regression

model will perform well so in other most of the cases where there are multiple features and multiple columns

so in that cases we use other regressions like lasso regression xgb regression which is known as xg

boost regressor such kind of things now let's ah now i'll do the same thing

with lasso regression okay so i'll create a text here as

lasso regression so previously we have did this with linear regression now

let's do this with lasso regression and just copy all these things

so you can choose here and just press shift plus down key so this will copy

all the cells so i want all this sense we just need to

make some changes so copy cells

okay

paste ctrl v okay so this will

paste all the cells now we need to change this model to lasso regression right so previously we

have worked on linear regression now we need to change it to lasso regression

so we have imported lasso regression from sk learn so it is nothing but

lasso so let's name this model as last rate model so this represents lasso

regression so i will change here also last trick model so i'm fitting the x

strain and the white range so this will train our lasso regression models so you can see the various parameters

here okay so we have this alpha value and other kind of things and then now let's see evaluate our

model so in this lean break i'll just change to last trick extreme okay

so this is the r squared value for lasso regression so for linear regression it's around 0.87 and

for lesser regression it's 0.84 okay so now let's visualize this also

so as you can see here in the case of linear regression it is close

but in the case of lasso regression it's much closer than the values plotted for linear regression so this is a very good

fit for this particular project where we can use lasso regression okay so now let's

predict this for test data so i'll change this to last rig let's find the r squared error

so the r squared error is kind of uh more in this case so let's print this

so as you can see here they kind of fly in the same line okay so the distance between two points are

not much higher so this is a very good fit for this particular case okay so this is how you can use multiple uh

models for predicting a particular value and compare how well the model is performing okay so you just try to do the same with

other models like xgb regressor and other regression models and see what kinds of plots you are getting okay

so this is how you can find the price of used cars using this particular data set using

linear regression and lasso regression models okay so do practice these codes so let me just give you a quick recap

so first we have imported the dependencies we have imported pandas for uh building the data frame and we have this

matplotlib library and c bond for making the plots that we need at the in the end of this project where we have

plotted the predictions of our model of the original data points values and then we have imported the strain

test split function to split our data into training data and test data and we have imported this linear regression model and lasso exist model

and matrix for evaluating our model so we have loaded our csv file into our pandas data frame

using this read csv function then we have printed the first five rows of our data frame and we have seen what

are the various columns we have so totally we are nine columns and we found that there are totally three not one val rows and nine columns

using this shape function and we have find the information of this data set using uh data frame dot info

so we see that there are no missing values so which is a good thing so then we have seen how many values are

distributed so how many petrol values are there how many dc values are there and the same for cellular type and

the transmission system etc so then we have replaced all these text data to suitable labels or suitable

numerical values okay so once we have encoded our data we have seen that it has encoded it well

so then we have splitted our data on target separately so here the target is nothing but the selling price okay so

this selling price isn't the unit of this prices and lags okay so indian rupees

and then we have printed it so then we have splitted our data into training data and test data so our model will be

trained on our training data and evaluated on our test data okay so first we have seen using linear regression okay so we have

evaluated our model find the r squared error value then we have plotted our predictions made by the model and the

original values and then we have did the same for test data okay and finally we have

also used a lasso regression model to do the same where we have also predicted for training data and test