```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split




df = pd.read_csv('/content/HDFC.csv')
```

EDA

```
#Dimention of dataset
df.shape
```

(5306, 15)

```
df.head()
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	
0	2000-01-03	HDFC	EQ	271.75	293.5	293.50	293.5	293.5	293.50	293.50	22744	6.6
1	2000-01-04	HDFC	EQ	293.50	317.0	317.00	297.0	304.0	304.05	303.62	255251	7.7
2	2000-01-05	HDFC	EQ	304.05	290.0	303.90	285.0	295.0	292.80	294.53	269087	7.9

```
#checking for null values and the types of data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5306 entries, 0 to 5305
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Date                5306 non-null   object 
 1   Symbol              5306 non-null   object 
 2   Series              5306 non-null   object 
 3   Prev Close          5306 non-null   float64
 4   Open                5306 non-null   float64
 5   High                5306 non-null   float64
 6   Low                 5306 non-null   float64
 7   Last                5306 non-null   float64
 8   Close               5306 non-null   float64
 9   VWAP                5306 non-null   float64
 10  Volume              5306 non-null   int64  
 11  Turnover            5306 non-null   float64
 12  Trades              2456 non-null   float64
 13  Deliverable Volume  4797 non-null   float64
 14  %Deliverble         4797 non-null   float64
dtypes: float64(11), int64(1), object(3)
memory usage: 621.9+ KB
```

```
df.describe()
```

	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trac
count	5306.000000	5306.000000	5306.000000	5306.000000	5306.000000	5306.000000	5306.000000	5.306000e+03	5.306000e+03	2456.000(
std	709.395090	709.703665	721.308080	697.450309	709.250204	709.430515	709.109622	2.991387e+06	3.607844e+14	57948.6032

```
#Taking out the Needed Datas
dfr = df[['Close','Open','High','Low','Last','Prev Close']]

dfr.describe()
```

	Close	Open	High	Low	Last	Prev Close
count	5306.000000	5306.000000	5306.000000	5306.000000	5306.000000	5306.000000
mean	1284.071005	1284.393074	1304.269732	1263.297842	1283.885017	1283.666114
std	709.430515	709.703665	721.308080	697.450309	709.250204	709.395090
min	283.850000	284.000000	290.500000	273.250000	282.850000	271.750000
25%	668.662500	669.712500	677.512500	660.000000	669.000000	668.650000
50%	1136.675000	1135.400000	1156.725000	1119.000000	1135.000000	1136.275000
75%	1811.787500	1813.812500	1835.000000	1783.075000	1812.000000	1811.475000
max	3180.150000	3148.000000	3262.000000	3100.550000	3178.000000	3180.150000

```
'''
From this Discriptive statistics we can find that when the Opening price is 1625,
High price is 1655 ,low price is 1610 , last price is 1630 and privious close is
1633 the closing price will at 1694 Rs.
'''
```

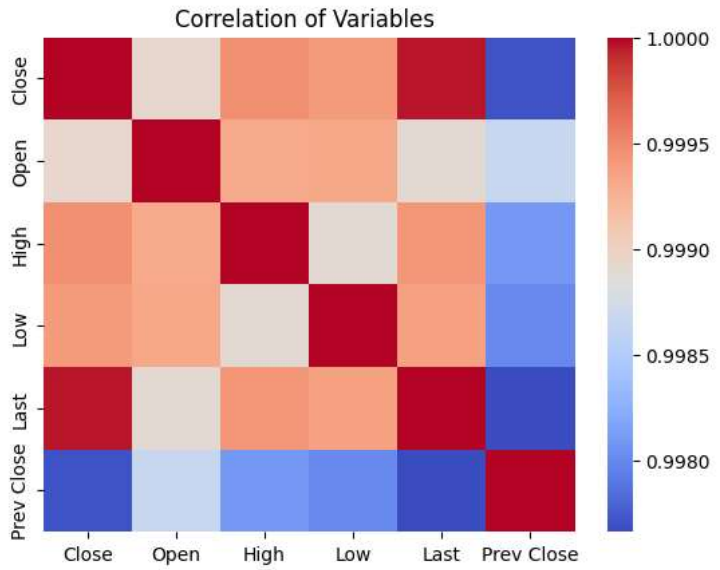
```
"\nFrom this Discriptive statistics we can find that when the Opening price is 1625,\nHigh price is 1655 ,low price is 1610 , last p
```

```
cor_matrix = dfr.corr()
cor_matrix
```

	Close	Open	High	Low	Last	Prev Close
Close	1.000000	0.998928	0.999466	0.999403	0.999960	0.997699
Open	0.998928	1.000000	0.999306	0.999326	0.998897	0.998656
High	0.999466	0.999306	1.000000	0.998895	0.999428	0.998095
Low	0.999403	0.999326	0.998895	1.000000	0.999365	0.998008
Last	0.999960	0.998897	0.999428	0.999365	1.000000	0.997662
Prev Close	0.997699	0.998656	0.998095	0.998008	0.997662	1.000000

```
sns.heatmap(cor_matrix , cmap = 'coolwarm' , annot = False)
plt.title('Correlation of Variables ')
```

```
Text(0.5, 1.0, 'Correlation of Variables ')
```



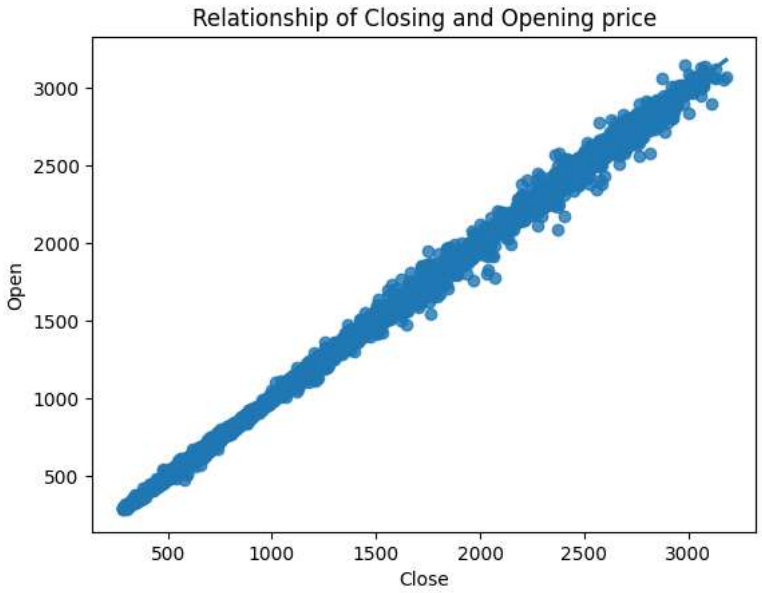
```
'''
From this correlation analysis we can find that the every independent variable
is having a high correlatiion with the dependent variable.
'''
```

```
'\nFrom this correlation analysis we can find that the every independent variable\nis having a high correlatiion with the dependent
```

```
#Checking the relationship of each variable with Dependent variable
```

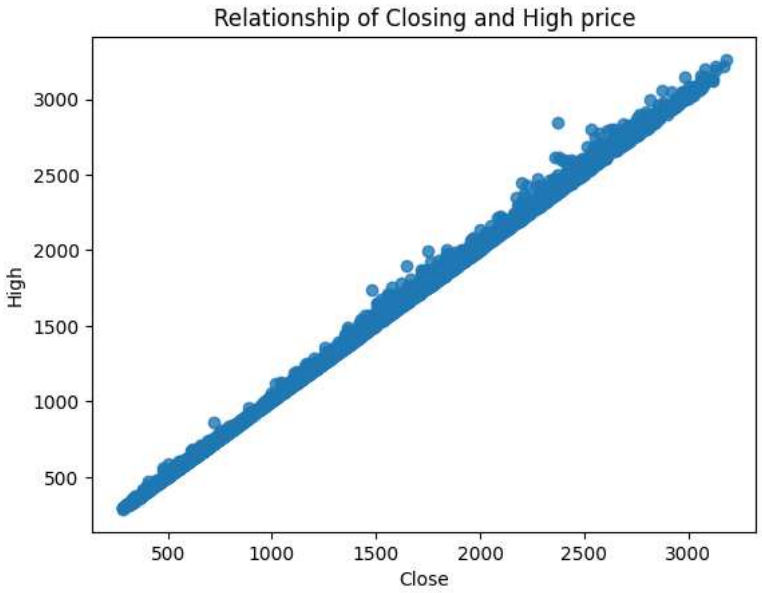
```
sns.regplot(x = df[['Close']], y = df[['Open']], data= df)
plt.title("Relationship of Closing and Opening price")
```

```
Text(0.5, 1.0, 'Relationship of Closing and Opening price')
```



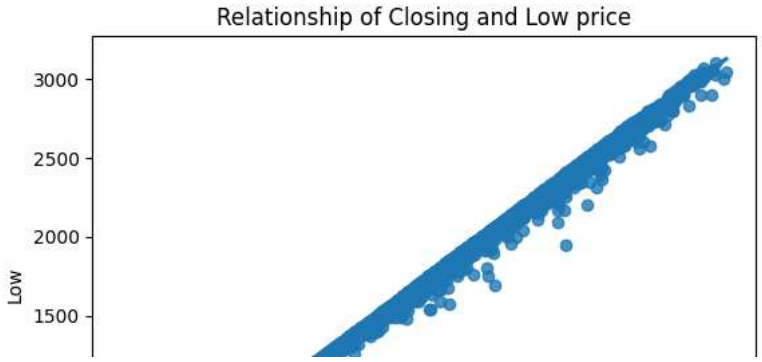

```
sns.regplot(x = df[['Close']], y = df[['High']], data= df)
plt.title("Relationship of Closing and High price")
```

```
Text(0.5, 1.0, 'Relationship of Closing and High price')
```




```
sns.regplot(x = df[['Close']], y = df[['Low']], data= df)
plt.title("Relationship of Closing and Low price")
```

```
Text(0.5, 1.0, 'Relationship of Closing and Low price')
```

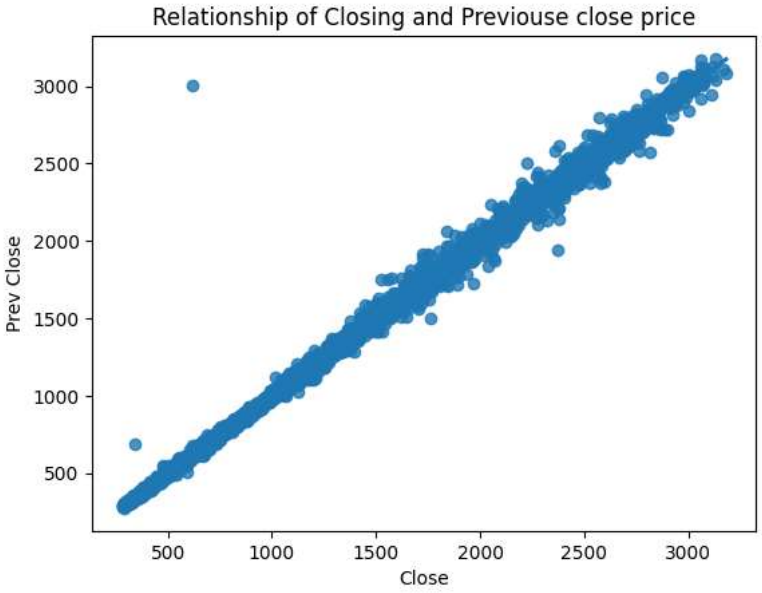



```
sns.regplot(x = df[['Close']], y = df[['Last']], data= df)
plt.title("Relationship of Closing and Last price")
```

```
Text(0.5, 1.0, 'Relationship of Closing and Last price')
```


```
sns.regplot(x = df[['Close']], y = df[['Prev Close']], data= df)
plt.title("Relationship of Closing and Previouse close price")
```

```
Text(0.5, 1.0, 'Relationship of Closing and Previouse close price')
```




```
'''
Each independent variable has a positve realtionship with the dependent
variable,and Previous closing price have some outliers.
'''
```

```
'\nEach independent variable has a positve realtionship with the dependent\nvariable,and Previous closing price have some outliers.\
```

```
#Alloting the Dependent and Independent Variable


y = df[['Close']]
x = df[['Open','High','Low','Last','Prev Close']]


#Spiliting the Dependent and Independet Variables in Training and Testings values


x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3 , random_state  = 0)


#dimentions of training models
x_train.shape
```

```
(3714, 5)
```

```
#dimentions of testing models
x_test.shape
```

```
(1592, 5)
```

```
#Training the model


lm = LinearRegression()
lm.fit(x_train,y_train)

print('Intercept: ' ,  lm.intercept_)
print('coefficient : ' ,  lm.coef_)
```

```
Intercept:  [-0.03689152]
coefficient :  [[-0.0942522   0.13722122  0.13469509  0.82435532 -0.00180514]]
```

```
'''From this Regression anaalysis we can see that two variables are negatively
related while training the model that are Previous close price and open price.'''
```

```
'From this Regression anaalysis we can see that two variables are negatively\nrelated while training the model that are Previous clo
```

```
'''
MODEL
Close = 0.4633596 - (0.06695249 Open) + (0.10075534 High) + (0.09384975 Low) + (0.8748669 Last) - (0.0028351 * Perv Close)
'''
```

```
'\nMODEL\nClose = 0.4633596 - (0.06695249 Open) + (0.10075534 High) + (0.09384975 Low) + (0.8748669 Last) - (0.0028351 * Perv Close)
```

```
#Predicting the model
yhat = lm.predict(x_test)

print(yhat)
```

```
[[1047.19614859]
 [ 354.90371574]
 [1830.30233441]
 ...
 [ 935.87265321]
 [ 811.4431188 ]
 [ 730.30187364]]
```

```
print({'Actual': y_test, 'Predicted': yhat})
```

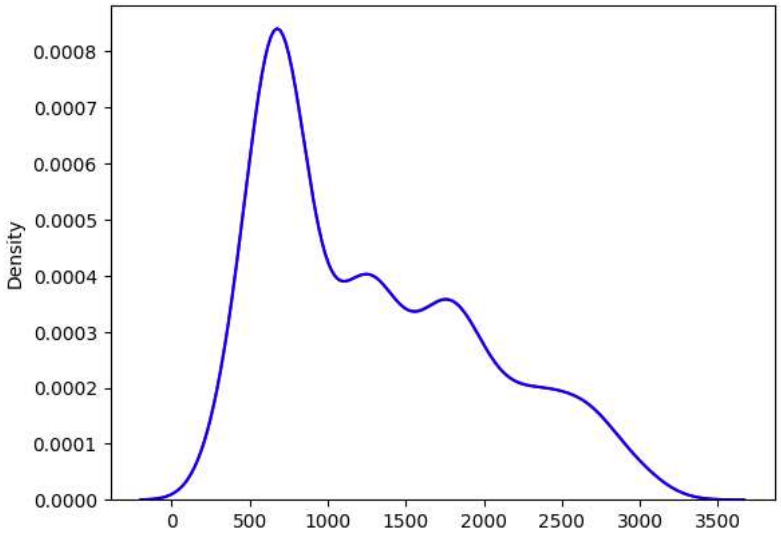
```
{'Actual':         Close
3655  1044.60
29     359.15
5135  1829.85
3671  1017.80
4681  1658.30
...       ...
4442  1762.25
4181  1301.95
1464   938.95
3534   810.25
3169   729.40

[1592 rows x 1 columns], 'Predicted': array([[1047.19614859],
       [ 354.90371574],
       [1830.30233441],
       ...,
       [ 935.87265321],
       [ 811.4431188 ],
       [ 730.30187364]])}
```

```
axl = sns.distplot(y_test, hist = False , color  = 'r')

sns.distplot(yhat, hist = False , color = 'b', ax = axl)
```

```
<Axes: ylabel='Density'>
```




```
#In comparing the predicted values with actual values, the values are same. let's justify by evaluating the model.


#Model Evalutation


mean_squared_error(y_test, yhat )
```

```
36.8218306730472
```

```
rsqr = lm.score(x,y)

print('R - Squared :' , rsqr)
```

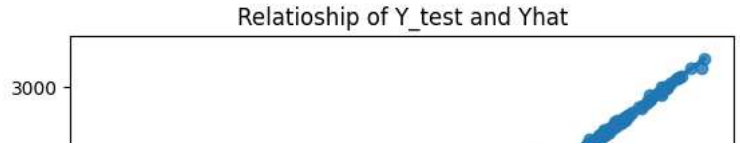
```
R - Squared : 0.9999343076450867
```

```
'''
From the above two model evaluation matrix we can find that the goodness of fit
is 0.9999 which explians that the independent variable explians the 99.99% of
the dependent variable and The average error between the predictions and actuals
in this dataset is 16.03, which is likely a good value considering the average
closing price is 1694.
'''
```

```
'\nFrom the above two model evaluation matrix we can find that the goodness of fit\nis 0.9999 which explians that the independent va
and actuals\nin this dataset is 16.03, which is likely a good value considering the average\nclosing price is 1694.\n'
```

```
sns.regplot( x = y_test, y = yhat , data = df)
plt.title('Relatioship of Y_test and Yhat')
plt.ylabel('y_test')
plt.xlabel('yhat')
```

```
Text(0.5, 0, 'yhat')
```




```
'''
ANALYSING WITH THE MODEL
We will take some relevant unseen values of independent variable to predict the dependent variable closing price.

Open : 3326
High : 3335
Low : 3302
Last : 3308
Prev Close : 3324
'''
```

```
'\nANALYSING WITH THE MODEL\nWe will take some relevant unseen values of independent variable to predict the dependent variable clos
```

```
values = [[3326,3335,3302,3308,3324]]

closep = lm.predict(values)

print('Closing Price :', closep)
```

```
Closing Price : [[3309.84338735]]
```

```
'''
If the Open pirce is 3326, High price is 3335 , Low price is 3302 ,
Last price is 3308 and Prev Close price is 3324 the closing price will be 3308.
'''
```

```
'\nIf the Open pirce is 3326, High price is 3335 , Low price is 3302 ,\nLast price is 3308 and Prev Close price is 3324 the closing
```

```
'''
CONCLUSION
According to the descriptive statistics, the closing price will be 1694 Rs.
while the opening price is 1625, the high price is 1655, the low price is 1610,
the last price was 1630, and the previous closure was 1633. Each independent
variable and the dependent variable have high correlations according to the
correlation analysis, which suggests a favourable association. There are some
anomalies in the prior closing price, though. The model's open price and preceding
closing price have a negative association, according to the regression analysis.

Close = 0.4633596 - (0.06695249 Open) + (0.10075534 High) + (0.09384975 Low) + (0.8748669 Last) - (0.0028351 * Prev Close) is the formula

The expected values and actual values are very similar. The dependent variable
is explained by the independent variables in 99.99% of the cases when the
goodness of fit is 0.9999. Given the average closing price of 1694, the average
difference between projections and actuals is 16.03, which is fair.
Therefore this Machine learning Model is Valid.
'''
```

```
'\nCONCLUSION\nAccording to the descriptive statistics, the closing price will be 1694 Rs.\nwhile the opening price is 1625, the hig
3. Each independent\nvariable and the dependent variable have high correlations according to the\ncorrelation analysis, which sugges
l's open price and preceding\nclosing price have a negative association, according to the regression analysis.\n\nClose = 0.4633596
lose) is the formula for the model.\n\nThe expected values and actual values are very similar. The dependent variable\nis explained
average closing price of 1694, the average\ndifference between projections and…'
```