Aim: Implementation of Linear Regression for Single Variate and Multi-variate

Theory(Part A):

Program Single variate using inbuilt functions.
Predict for unseen samples
Plot the regression

In [13]:
```
import pandas as pd
df = pd.read_csv('/content/day.csv')
df.head()
```

Out[13]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011	1	0	1	0	3	1	1	0.2269	0.2292	0.4369	0.18690	82	1518	16

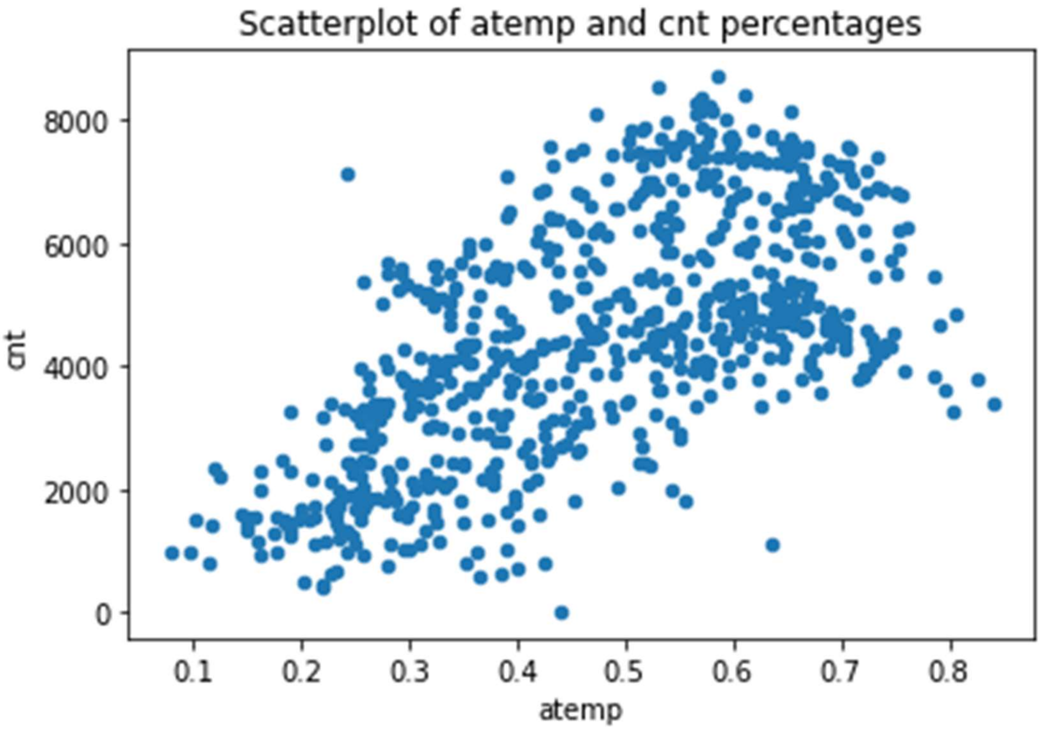
```
-                                   57   70   57   0              0
01                                                                0
-
05
```

In [14]:

```
df.plot.scatter(x='atemp', y='cnt', title='Scatterplot of atemp and cnt
percentages')
#we slight possitive correaltion of atmep v/s cnt hence we will take
respective x and y
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fac5c574fd0>
```



In [27]:

```
from sklearn.model_selection import train_test_split

#taking respective X(atemp) and Y(atemp)
#as fit needs 2D arrays for X_train and X_test we use reshape(-1,1) to
convert our 1D array in the form of 2D array
X=df['atemp'].values.reshape(-1, 1)
y=df['cnt'].values.reshape(-1, 1)

#80-20 train-test split taken
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 102)
```

In [28]:

```
#using linear regression model we predict y(cnt) values for x(atemp) test
values
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

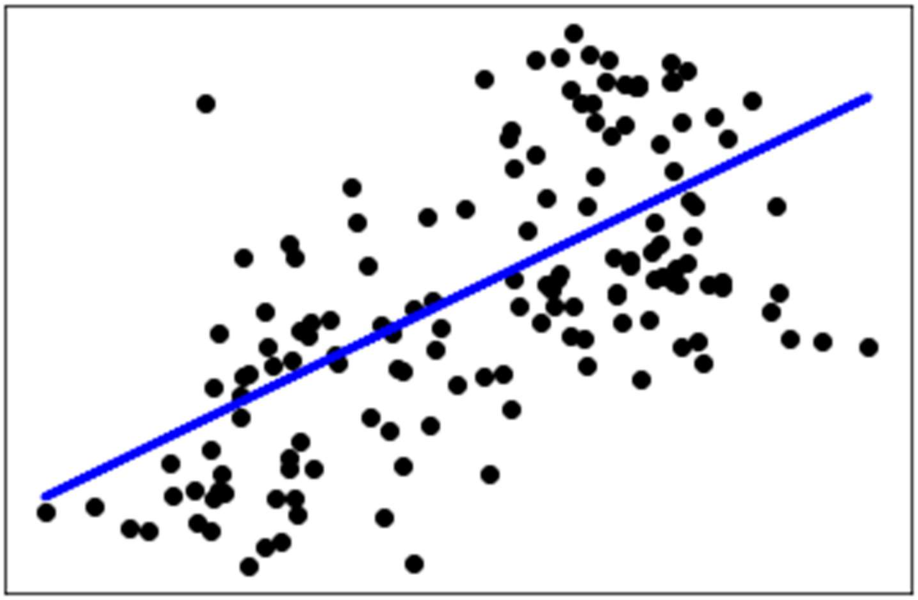
In [29]:

```
#we use dataframe to show 2 columns one being actual values of y(cnt) and
other predicted
df_preds = pd.DataFrame({'Actual': y_test.squeeze(), 'Predicted':
y_pred.squeeze()})
print(df_preds)
```

```
     Actual    Predicted
0      4990  4495.163021
1      2114  2659.010527
2      1096  3273.699244
3      7534  4387.937426
4      3068  3828.673823
..      ...          ...
142    5146  3679.591490
143    2913  5059.402187
144    4068  3917.266895
145    7494  5189.608386
146    5255  5385.633753

[147 rows x 2 columns]
```

In [26]:

```
#we plot the regression along with mean square error and coefficient of
determination
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score

plt.scatter(X_test,y_test, color="black")
plt.plot(X_test,y_pred, color="blue", linewidth=3)

plt.xticks(())
plt.yticks(())

plt.show()

print()

print("Coefficient: ", regressor.coef_)

print("Mean squared error: %.2f" % mean_squared_error(y_test, y_pred))
print("Coefficient of determination: %.2f" % r2_score(y_test, y_pred))
```



```
Coefficient:  [[7600.18249789]]
Mean squared error: 2228174.03
Coefficient of determination: 0.38
```

Theory (Part B):

Program Multi variate using inbuilt functions.

Predict for unseen samples

In [11]:

```
import pandas as pd
df2 = pd.read_csv('/content/accelerometer.csv')
df2.head()
```

Out[11]:

	wconfid	pctid	x	y	z
0	1	20	1.004	0.090	-0.125
1	1	20	1.004	-0.043	-0.125
2	1	20	0.969	0.090	-0.121
3	1	20	0.973	-0.012	-0.137
4	1	20	1.000	-0.016	-0.121

In [4]:

```
from sklearn.model_selection import train_test_split

#here we take X as three columns i.e x,y,z values to predict pctid column
values
X=df2.drop(['wconfid','pctid'],axis=1)
y=df2['pctid']

#replace the Nan values with 0
df2.fillna(0)


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 3)
```

In [5]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

In [8]:

```
df_preds = pd.DataFrame({'Actual': y_test.squeeze(), 'Predicted':
y_pred.squeeze()})
print(df_preds)
```

```
        Actual  Predicted
97397       95  59.319857
33167       75  60.097789
114957      40  59.984126
49555      100  61.385427
11638       35  59.980032
...        ...        ...
94145       90  60.216233
147023      95  59.973453
96676       95  60.248866
138728      80  59.979476
37239       80  59.813745

[30600 rows x 2 columns]
```

In [9]:

```
from sklearn.metrics import mean_squared_error, r2_score

print("Mean squared error: %.2f" % mean_squared_error(y_test, y_pred))

print("Coefficient of determination: %.2f" % r2_score(y_test, y_pred))
```

```
Mean squared error: 602.27
Coefficient of determination: 0.00
```

Conclusion:

Thus, we learnt how to implement linear regression on both univariate and multivariate datasets. We used python to implement these functions along with packages like mathplotlib, numpy, sklearn. We also found out the coefficient of determination , Mean squared error and plotted the regression using metrics provided in the package.