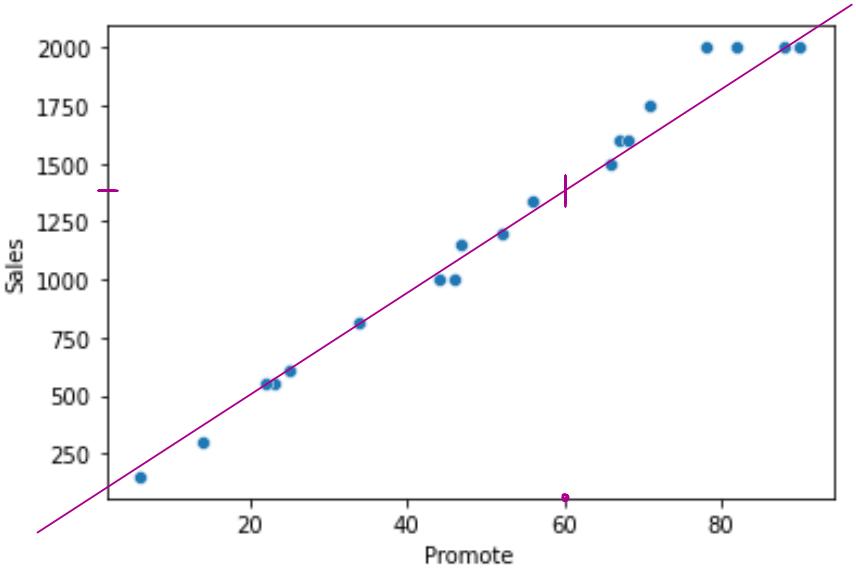
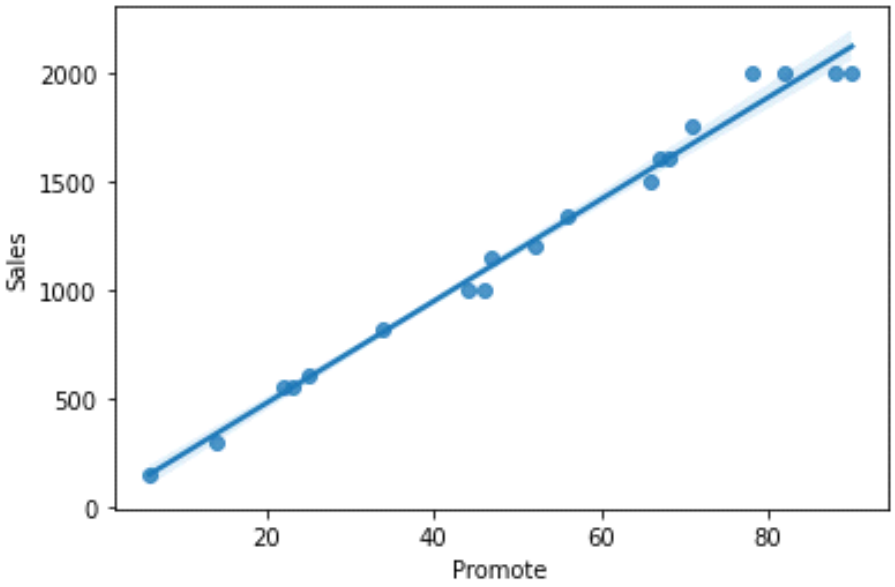
Linear

01 June 2023 10:13

For any scatter plot the response / dependent variable should be at Y-axis





Exercises: Cars93.csv

1. Independent: MPG.City , Dependent: Price
2. Independent: Horsepower , Dependent: Price
3. Independent: MPG.City, Horsepower ; Dependent: Price

Dummying the Data / One Hot Encoding

Dummying the Data / One Hot Encoding

X1
A
A
A
A
B
B
B
C
C
C

X1_A	X1_B	X1_C
1	0	0
1	0	0
1	0	0
1	0	0
0	1	0
0	1	0
0	1	0
0	0	1
0	0	1
0	0	1

X1_B	X1_C
0	0
0	0
0	0
0	0
1	0
1	0
1	0
0	1
0	1
0	1

y_i $\hat{y}_i$

Existing	Predicted
56	55.2
64	45.33
94	85.44
55	54.22
22	30

$$MAE = \frac{|56-55.2| + |64-45.33| + \cdots + |22-30|}{5} = 7.365$$

Existing	Predicted
56	57
64	58
94	90
55	53
22	24

$MAE = 3$

X : Lot Size, Bedrooms

```
In [80]: print(mean_absolute_error(y, y_pred_trn))
    ...: print(mean_squared_error(y, y_pred_trn))
    ...: print(r2_score(y, y_pred_trn))
15620.221815301456
448196130.4555012
0.3702693440581585
```

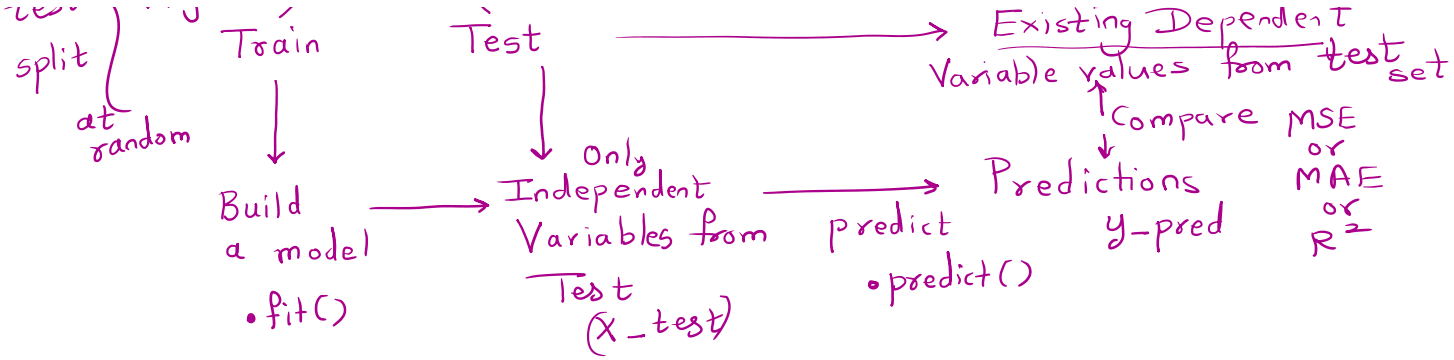
X : Lot Size, Bedrooms, Bathrooms, Storeys

```
In [82]: print(mean_absolute_error(y, y_pred_trn))
    ...: print(mean_squared_error(y, y_pred_trn))
    ...: print(r2_score(y, y_pred_trn))
13318.45543520826
330563406.94941866
0.5355472817735651
```

ML Model Evaluation

train test split → say Major Data → 70% Train, 30% Minor Test

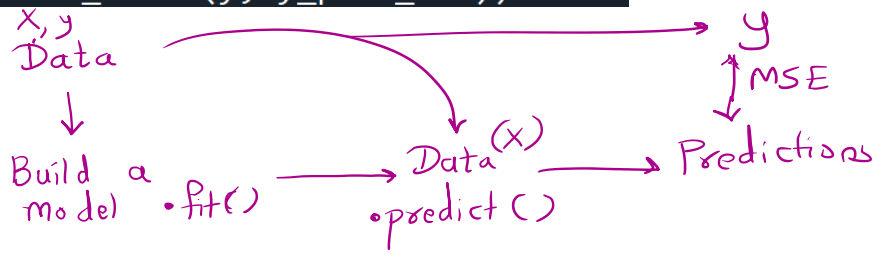
Existing Dependent Variable values from test set




```
lr = LinearRegression()
lr.fit(X, y)

print("Slopes =", lr.coef_)
print("Intercept =", lr.intercept_)

y_pred_trn = lr.predict(X)
print(mean_absolute_error(y, y_pred_trn))
```




```
In [148]: dum_sals = pd.get_dummies(sals)
     ...: X = dum_sals[['Department_B', 'Department_C',
     ...:                'Department_D']]
     ...: y = dum_sals['Salary']
     ...:
     ...:
     ...: lr = LinearRegression()
     ...: lr.fit(X, y)
     ...: print("Slopes =", lr.coef_)
     ...: print("Intercept =", lr.intercept_)
Slopes = [3548.375      6664.35714286 9190.45238095]
Intercept = 34000.5
```

Salaries = 34000.5 + B(1 or 0) * 3548.375 + C(1 or 0) * 6664.357 + D(1 or 0) * 9190.4523

D :- Salaries = 34000.5 + 1 (9190.4523)

B :- Salaries = 34000.5 + 1 (3548.375) = 37548.375

A :- Salaries = 34000.5

```
In [150]: dum_sals = pd.get_dummies(sals)
     ...: X = dum_sals[['Department_A','Department_B',
     ...:                'Department_D']]
     ...: y = dum_sals['Salary']
     ...:
     ...: lr = LinearRegression()
     ...: lr.fit(X, y)
     ...: print("Slopes =", lr.coef_)
     ...: print("Intercept =", lr.intercept_)
Slopes = [-6664.35714286 -3115.98214286  2526.0952381 ]
Intercept = 40664.857142857145
```

Salaries = 40664.857 + A*6664.357 − 3115.982 * B
+ D * 2526.095

D:- Salaries = 40664.857 + 1(2526.095) = 43190.952

B :- Salaries = 40664.857 − 1(3115.982) = 37548.375

A :- Salaries = 40664.857 − 1(6664.357) = 35000.4

C :- Salaries = 40664.856

```
In [152]: sals.groupby('Department')['Salary'].mean()
Out[152]:
Department
A    34000.500000
B    37548.875000
C    40664.857143
D    43190.952381
```