



Tecnológico  
de Monterrey

ITESM

# MAESTRÍA EN INTELIGENCIA ARTIFICIAL APLICADA

SEGUNDO TRIMESTRE

APUNTES DATA ANALYSIS CON  
PYTHON

MODULO 4

NOMBRE: VILLALPANDO GUERRERO  
JIRAM CESAR

## Contenido

<b>MODULO 4</b> .....	3
<b>Model development</b> .....	3
<b>Regresión lineal simple y múltiple</b> .....	3
<b>Diagrama Regresión</b> .....	6
<b>Gráfico residual</b> .....	8
<b>Gráficos de dispersión</b> .....	8
<b>Regresión polinomial y pipelines</b> .....	9
<b>Medida para evaluación muestras</b> .....	9
<b>Toma de decisiones y predicción</b> .....	11
<b>Graded Review Questions</b> .....	12

## MODULO 4

### Model development


Es un proceso iterativo, en el que se derivan, prueban y construyen muchos modelos hasta que se construye un modelo que se ajusta a los criterios deseados. Es posible que el trabajo de modelado posterior deba comenzar la búsqueda en el mismo lugar donde comenzó la construcción del modelo original, en lugar de donde terminó

Hay que tener en cuentas las variables dependiente e independientes para poder predecir algún dato.

### Regresión lineal simple y múltiple

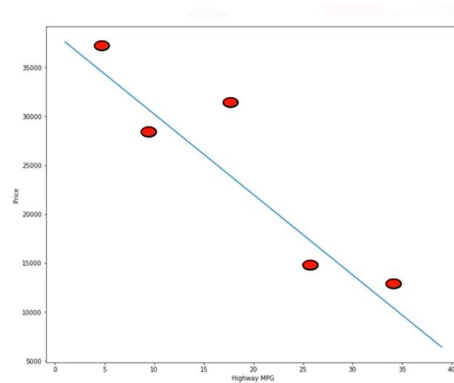
#### Simple

Es un modelo de regresión lineal con una sola variable explicativa . Es decir, se trata de puntos de muestra bidimensionales con una variable independiente y una variable dependiente y encuentra un función lineal que, con la mayor precisión posible, predice los valores de la variable dependiente en función de la variable independiente.

$$y = b_0 + b_1 x$$


Usualmente se ponen los datos en dos data frames, X, Y. En distintos casos podemos encontrar la relación que nos será de utilidad, asimismo hay que tener en cuenta el ruido,

que son datos que ilustran otros que no son perfectos, en donde se encuentren fuera de la línea de tendencia central.



Se pueden usar los datos obtenidos para predecir valores que no se han visto.

En Python, se puede usar sklearn para usar estos métodos de regresión

- Importando con:

```
from sklearn.linear_model import LinearRegression
```

- Posteriormente creando un objeto con:

```
lm=LinearRegression()
```

- Definir las variables x, y
- Usar X, Y para los parámetros del modelo
- Y obtener predicción

## Múltiple

procedimiento mediante el cual se trata de determinar si existe o no relación de dependencia entre dos o más variables. Es decir, conociendo los valores de una variable independiente, se trata de estimar los valores, de una o más variables dependientes.

Ejemplo:

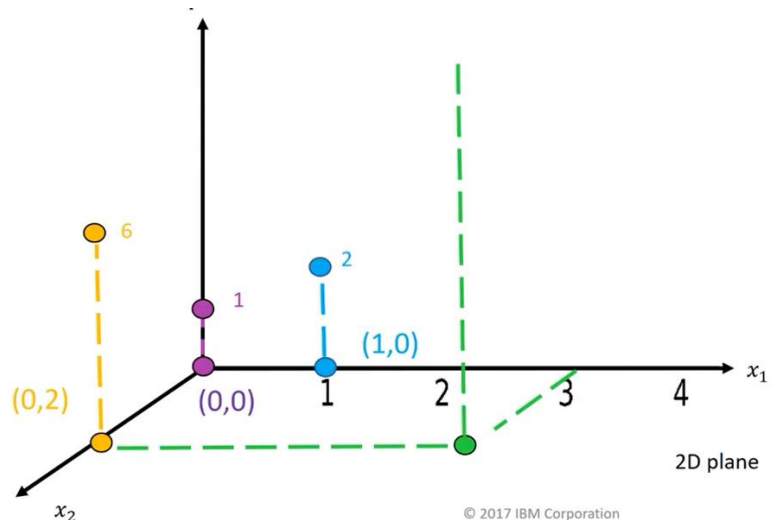
$$\hat{Y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4$$

Otro ejemplo se puede ver de la siguiente manera:

$$\hat{Y} = 1 + 2x_1 + 3x_2$$

n	$x_1$	$x_2$	$\hat{Y}$
1	0	0	1
2	0	2	6
3	1	0	2
4	3	2	13

$x$

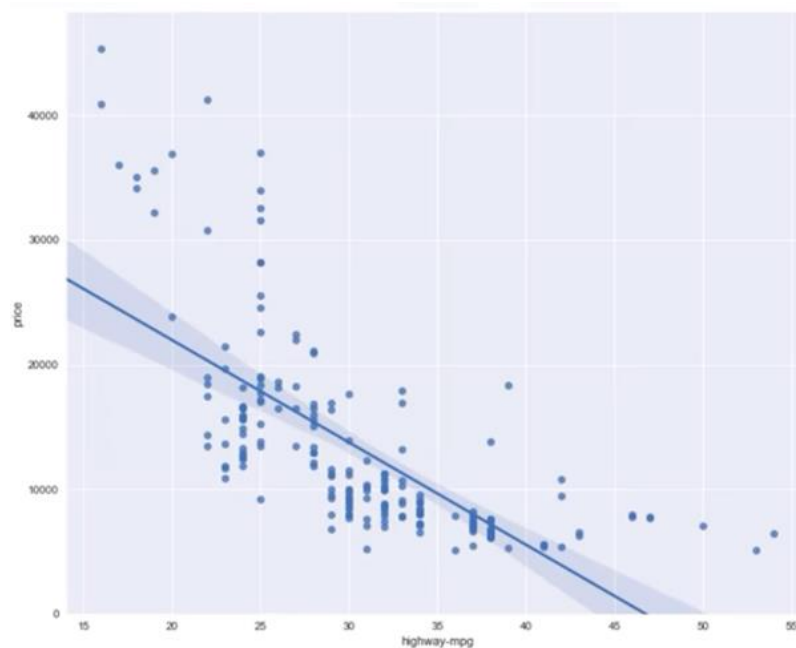


- Se definen las variables dentro de un DF
- Se entrena el modelo
- Se obtiene la predicción
- Se encuentra al interceptor  $b_0$
- Se encuentra a los demás  $b...$

## Diagrama Regresión

Brinda una estimación de la relación entre dos variables, la fuerza de correlación y la dirección positiva o negativa de la misma relación

Es una combinación del diagrama de dispersión y de la regresión lineal Y



Para usarlo en Python:

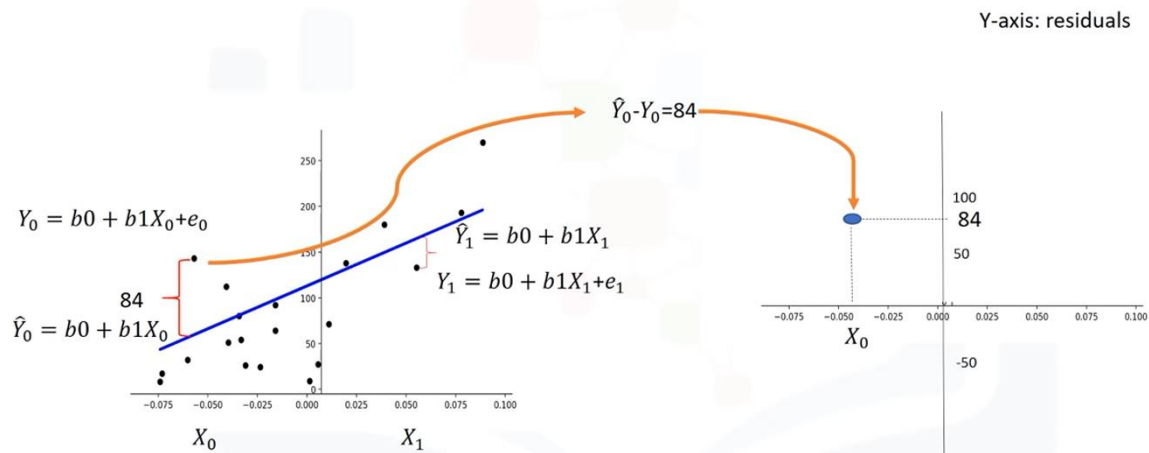
Primero se importa

```
import seaborn as sns
```

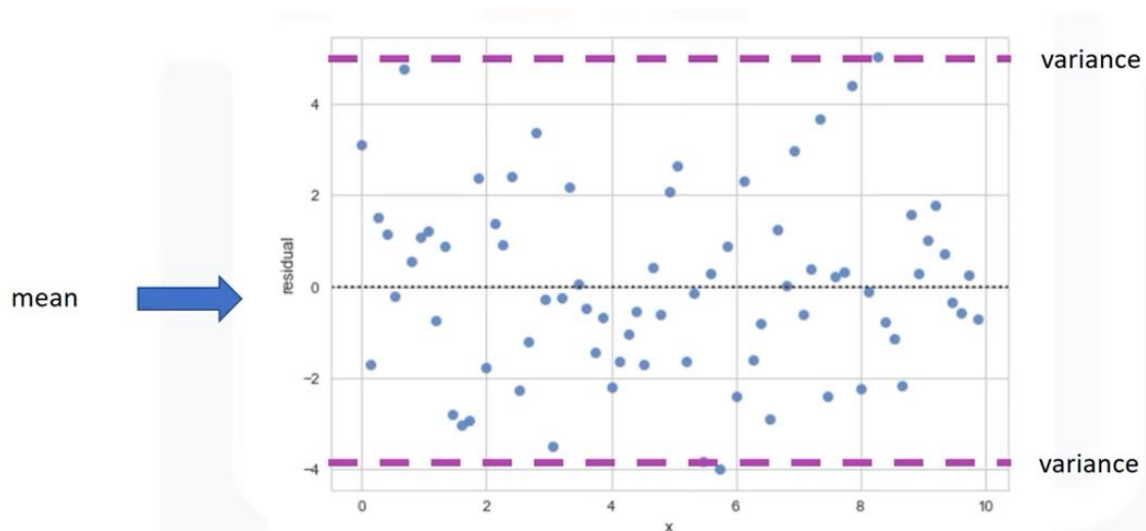
Posteriormente lo usamos en un dataframe

```
sns.regplot(x="highway-mpg", y="price", data=df)  
plt.ylim(0,)
```

Se obtiene lo siguiente:



En el siguiente gráfico se ve la media y las varianzas con límites, por lo tanto, se puede apreciar un análisis mucho más específico.

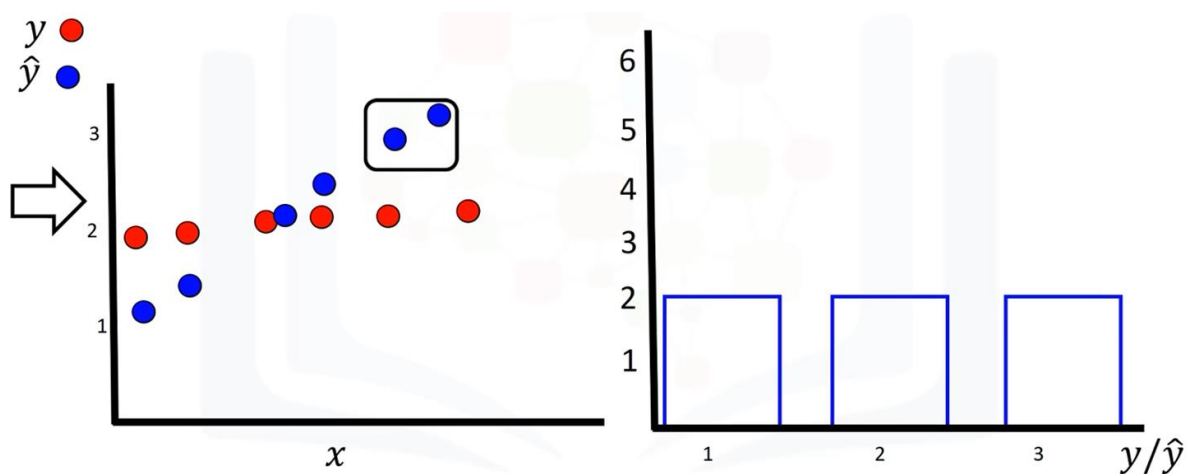


## Gráfico residual

Se usa nuevamente seaborn

Y se usa `sns.residplot(df[ ],df[ ])`

Se obtiene un ejemplo como el siguiente.



## Gráficos de dispersión

Se importa seaborn

Y sns. Displot

Ejemplo:

```
ax1 = sns.distplot(df['price'], hist=False, color="r", label="Actual Value")

sns.distplot(Yhat, hist=False, color="b", label="Fitted Values" , ax=ax1)
```



## Regresión polinomial y pipelines

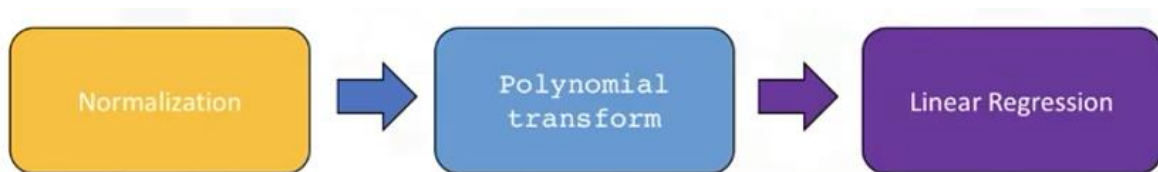
El objetivo de la regresión polinomial es modelar una relación no lineal entre las variables independientes y dependientes (técnicamente, entre la variable independiente y la media condicional de la variable dependiente).

Se usa ecuaciones de segundo o mayor grado para este caso.

Para calcular el polinomio se usa:

```
f=np.polyfit(x,y,3)
p=np.poly1d(f)
```

Los pasos para la predicción son los siguientes:



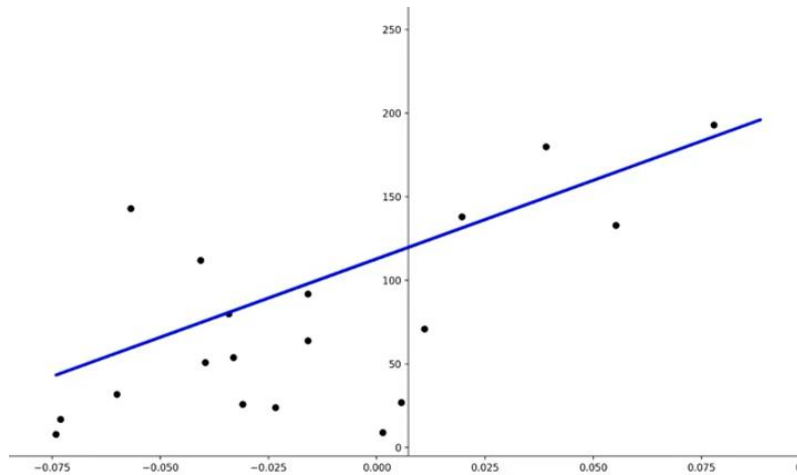
## Medida para evaluación muestras

Es una forma de determinar que tan bien el modelo en relación con el dataset

Existen dos medidas para determinar si el modelo es bueno.

- MSE
- $R^2$

Por ejemplo:



En Python se usa sklearn.metrics

```
from sklearn.metrics import mean_squared_error
mean_squared_error(df['price'], Y_predict_simple_fit)

3163502.944639888
```

Para  $R^2$

Es una medida estadística de qué tan cerca están los datos de la línea de regresión ajustada. También se conoce como coeficiente de determinación, o coeficiente de determinación múltiple si se trata de regresión múltiple.

La fórmula es la siguiente:

$$R^2 = \left( 1 - \frac{\text{MSE of regression line}}{\text{MSE of the average of the data}} \right)$$

### Toma de decisiones y predicción

- Hacer la predicción
- Visualizar
- Medidas numéricas para evaluación
- Modelos comparativos

#### Pasos

- Entrenar el modelo

Ejemplo:

```
lm.fit(df['highway-mpg'], df['prices'])
```

## Graded Review Questions

### Question 1

1/1 point (graded)

Let `x` be a dataframe with 100 rows and 5 columns. Let `y` be the target with 100 samples. Assuming all the relevant libraries and data have been imported, the following line of code has been executed:

```
LR = LinearRegression()
```

```
LR.fit(X, y)
```

```
yhat = LR.predict(X)
```

How many samples does `yhat` contain?

☐ 5

☐ 500

☒ 100

☐ 0



[Save](#) | [Show answer](#)

**Submit**

You have used 1 of 2 attempts

✓ Correct (1/1 point)

### Question 2

1/1 point (graded)

What value of  $R^2$  (coefficient of determination) indicates your model performs best?

☐ -100

☐ -1

☐ 0

☒ 1



[Save](#) | [Show answer](#)



## Question 3

1/1 point (graded)

Which statement is true about polynomial linear regression?

- ☐ Polynomial linear regression is not linear in any way.
- ☒ Although the predictor variables of polynomial linear regression are not linear, the relationship between the parameters or coefficients is linear.
- ☐ Polynomial linear regression uses wavelets.



[Save](#) | [Show answer](#)

**Submit**

You have used 1 of 2 attempts

✓ Correct (1/1 point)

## Question 4

1/1 point (graded)

The larger the mean squared error, the better your model performs:

☒ False

☐ True



[Show answer](#)

**Submit**

You have used 1 of 1 attempt

✓ Correct (1/1 point)

## Question 5

1/1 point (graded)

Assume all the libraries are imported. y is the target and X is the features or dependent variables. Consider the following lines of code:

```
Input=[('scale',StandardScaler()),('model',LinearRegression())]
```

```
pipe=Pipeline(Input)
```

```
pipe.fit(X,y)
```

```
ypipe=pipe.predict(X)
```

What is the result of ypipe?

☐ Polynomial transform, standardize the data, then perform a prediction using a linear regression model.

☒ Standardize the data, then perform prediction using a linear regression model.

☐ Polynomial transform, then standardize the data.



[Save](#) | [Show answer](#)

**Submit**

You have used 1 of 2 attempts

✓ Correct (1/1 point)

Course Progress for 'a01793579' (a01793579@tec.mx)

