

# 🧑‍💻 IBM - DATA SCIENCE - MACHINE LEARNING WITH PYTHON 🐍🧠

## Semana 1 - Introduction: What is Machine Learning?

—> What are we going to learn?

Use ML to make predictions ...

ID	Clump	Uniformity	CellSize	CellShape	MargAdh	SingEpSize	BareNuc	BlandChrom	NormNucI	Mit	Class
1000025	5	1	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	1	benign
1015425	3	1	1	1	2	2	3	1	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	1	benign
1017023	4	1	1	3	2	1	3	1	1	1	benign
1017122	8	10	10	8	7	10	7	1	1	1	malignant
1018099	1	1	1	1	2	10	3	1	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	1	benign
1033078	2	1	1	1	2	1	1	1	1	5	benign
1034078	4	7	1	1	7	1	7	1	1	1	benign

Confusion matrix

True label

Predicted label

Modeling

Prediction

IBM Developer

SKILLS NETWORK

... such as predicting the right drug

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL			
4	61	F	LOW			

Decision Tree

IBM Developer

SKILLS NETWORK

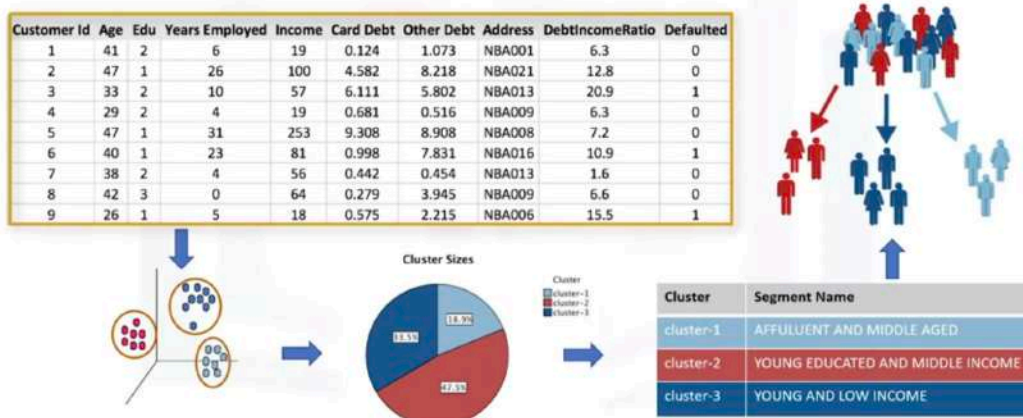
## Use ML to make decisions



IBM Developer

SKILLS NETWORK

## Use ML for customer segmentation



IBM Developer

SKILLS NETWORK

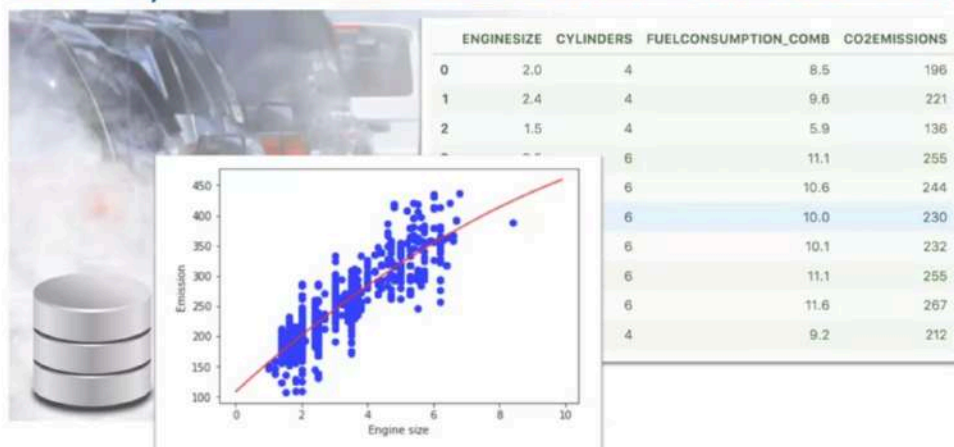
## Use ML for recommendation systems



IBM Developer

SKILLS NETWORK 

## Use Python libraries to build ML models



IBM Developer

SKILLS NETWORK 

## What do you get from this course?

### SKILLS:

- Regression
- Classification
- Clustering
- Scikit Learn
- Scipy

### PROJECTS:

- Cancer detection
- Predicting economic trends
- Predicting customer churn
- Recommendation engines
- Many more ....

IBM Developer

SKILLS NETWORK 

## What is machine learning?

**Machine learning** is the subfield of computer science that gives “**computers the ability to learn without being explicitly programmed.**”

### Arthur Samuel

American pioneer in the field of computer gaming and artificial intelligence, coined the term “machine learning” in 1959 while at IBM.

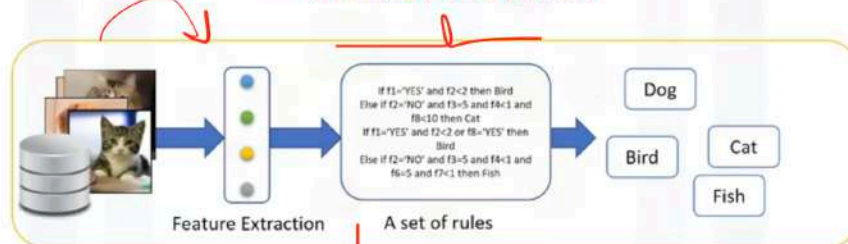
IBM Developer

SKILLS NETWORK 

## How machine learning works?

Primero, interpretarlo como un set de features

ANTES DEL MACHINE LEARNING



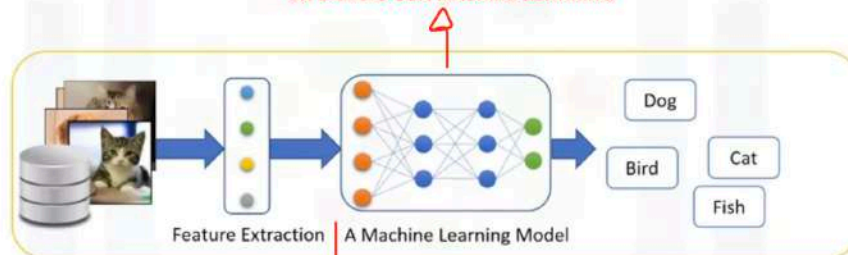
ESTO ERA INEFICIENTE YA QUE ERA ALTAMENTE DEPENDIENTE DEL DATA SET, NECESITABAS MUCHAS REGLAS Y ADEMÁS NO ERA LO SUFICIENTEMENTE GENERALIZADO COMO PARA ACERTAR LAS CONCLUSIONES.

IBM Developer

SKILLS NETWORK 

## How machine learning works?

EN CAMBIO CON MACHINE LEARNING



EL MODELO APRENDE LOS PATRONES PARA CADA ANIMAL. SE VA FIJANDO CUALES SON LOS FEATURES QUE SE REPITEN Y ENTIENDE CUAL ES EL PATRON A SEGUIR.

IBM Developer

SKILLS NETWORK 

Inspirados por cómo los humanos aprendemos, los modelos de ML iteran por la data hasta entender cuales son los patrones que definen a una x.

Dentro de las técnicas mas populares de ML encontramos:



## Major machine learning techniques

- **Regression/Estimation**
  - Predicting continuous values
- **Classification**
  - Predicting the item class/category of a case
- **Clustering**
  - Finding the structure of data; summarization
- **Associations**
  - Associating frequent co-occurring items/events

## Major machine learning techniques

- **Anomaly detection**
  - Discovering abnormal and unusual cases
- **Sequence mining**
  - Predicting next events; click-stream (Markov Model, HMM)
- **Dimension Reduction**
  - Reducing the size of data (PCA)
- **Recommendation systems**
  - Recommending items

## Difference between artificial intelligence, machine learning, and deep learning

- AI components:
  - Computer Vision
  - Language Processing
  - Creativity
  - Etc.
- Machine learning:
  - Classification
  - Clustering
  - Neural Network
  - Etc.
- Revolution in ML:
  - Deep learning

→ Parte estadística de la IA



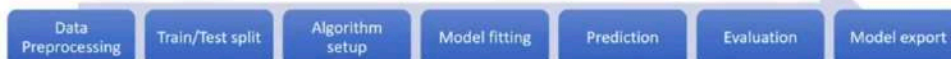
IBM Developer

SKILLS NETWORK 

Sobre scikit-learn:

## More about scikit-learn

- Free software machine learning library
- Classification, Regression and Clustering algorithms
- Works with NumPy and SciPy
- Great documentation
- Easy to implement



IBM Developer

SKILLS NETWORK 

## scikit-learn functions

```
from sklearn import preprocessing
X = preprocessing.StandardScaler().fit(X).transform(X)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)

from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)

clf.fit(X_train, y_train)

clf.predict(X_test)

from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test, yhat, labels=[1,0]))

import pickle
s = pickle.dumps(clf)
```

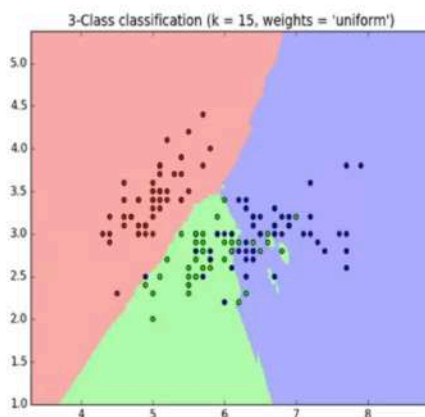


IBM Developer

SKILLS NETWORK 

Supervised vs Unsupervised algorithms:

## What is supervised learning?



We "teach the model,"  
then with that knowledge,  
it can predict unknown or  
future instances.

IBM Developer

SKILLS NETWORK 

Oka pero como le enseñamos y entrenamos al modelo?  
Con un *labeled dataset*

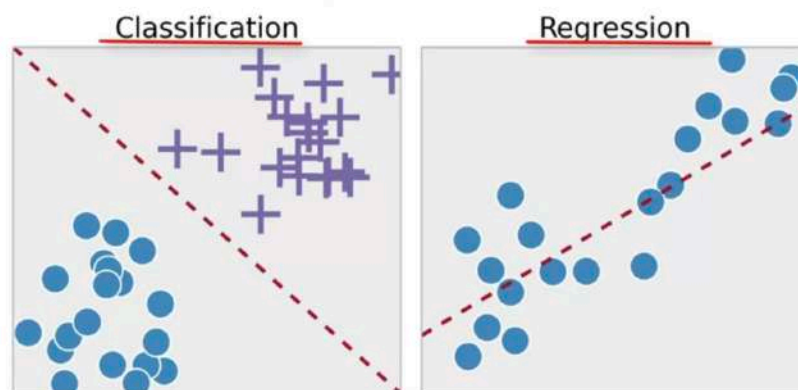


## Teaching the model with labeled data

Ya sabemos la clase de estas entradas historicas

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign

## Types of supervised learning

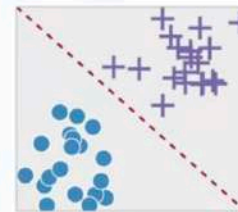


# What is classification?

**Classification** is the process of predicting discrete class labels or categories.

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucI	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign

Categorical Values

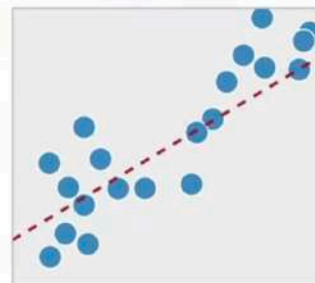


# What is regression?

**Regression** is the process of predicting continuous values.

	ENGINE SIZE	CYLINDERS	FUEL CONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Continuous Values



# What is unsupervised learning?

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2	6	19	0.124	1.073	NBA001	6.3
2	47	1	26	100	4.582	8.218	NBA021	12.8
3	33	2	10	57	6.111	5.802	NBA013	20.9
4	29	2	4	19	0.681	0.516	NBA009	6.3
5	47	1	31	253	9.308	8.908	NBA008	7.2
6	40	1	23	81	0.998	7.831	NBA016	10.9
7	38	2	4	56	0.442	0.454	NBA013	1.6
8	42	3	0	64	0.279	3.945	NBA009	6.6
9	26	1	5	18	0.575	2.215	NBA006	15.5
10	47	3	23	115	0.653	3.947	NBA011	4
11	44	3	8	88	0.285	5.083	NBA010	6.1
12	34	2	9	40	0.374	0.266	NBA003	1.6

Unsupervised learning techniques:

- Dimension reduction
- Density estimation
- Market basket analysis
- Clustering

ALL OF THIS DATA IS UNLABELED

The model works on its own to discover information.

IBM Developer

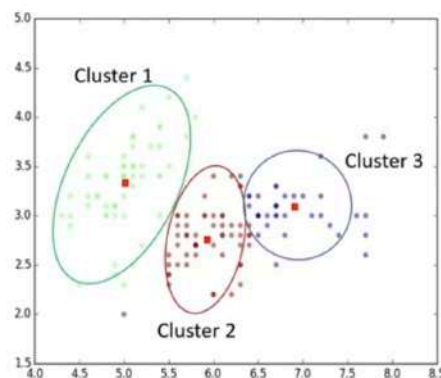
SKILLS NETWORK

Dejamos que el modelo trabaje por su cuenta en describir información que podría no ser visible para el ojo humano

# What is clustering?

**Clustering** is grouping of data points or objects that are somehow similar by:

- Discovering structure
- Summarization
- Anomaly detection



IBM Developer

SKILLS NETWORK

# Supervised vs unsupervised learning

## Supervised Learning

- **Classification:**  
Classifies labeled data
- **Regression:**  
Predicts trends using previous labeled data
- Has more evaluation methods than unsupervised learning
- Controlled environment

## Unsupervised Learning

- **Clustering:**  
Finds patterns and groupings from unlabeled data
- Has fewer evaluation methods than supervised learning
- Less controlled environment

## Semana 2 - Linear Regression

### Introduction to Regression

# What is regression?

*"Causas"*  
X: Independent variable

*"Target"*  
Y: Dependent variable *Tiene que ser continua*

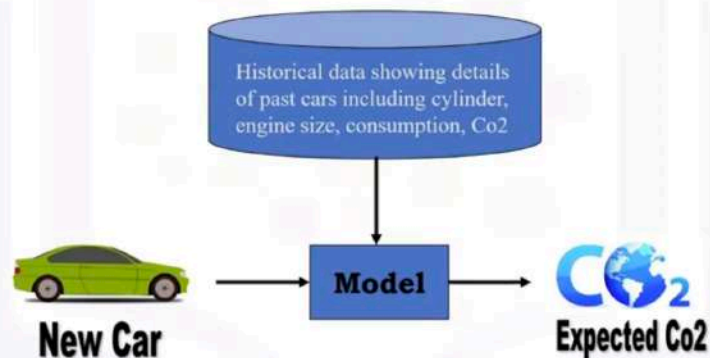
	ENGINE SIZE	CYLINDERS	FUEL CONSUMPTION_COMB	CO2 EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Regression is the process of predicting a continuous value

IBM Developer

SKILLS NETWORK 

# What is a regression model?



IBM Developer

SKILLS NETWORK 

Básicamente hay dos tipos de modelos de regresión:



## Types of regression models

- Simple Regression:
  - Simple Linear Regression
  - Simple Non-linear Regression
- Multiple Regression:
  - Multiple Linear Regression
  - Multiple Non-linear Regression

Predict **co2emission** vs **EngineSize** of all cars

Predict **co2emission** vs **EngineSize** and **Cylinders** of all cars

## Applications of regression

- Sales forecasting
- Satisfaction analysis
- Price estimation
- Employment income



# Regression algorithms

---

- Ordinal regression
- Poisson regression
- Fast forest quantile regression
- Linear, Polynomial, Lasso, Stepwise, Ridge regression
- Bayesian linear regression
- Neural network regression
- Decision forest regression
- Boosted decision tree regression
- KNN (K-nearest neighbors)



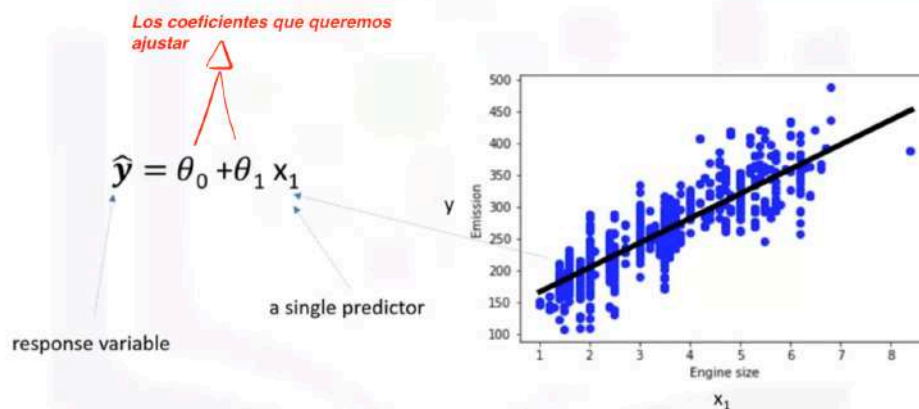
## Simple Linear Regression

### Simple Linear Regression

# Linear regression topology

- Simple Linear Regression:
  - Predict **co2emission** vs **EngineSize** of all cars
    - Independent variable (x): EngineSize
    - Dependent variable (y): co2emission
- Multiple Linear Regression:
  - Predict **co2emission** vs **EngineSize** and **Cylinders** of all cars
    - Independent variable (x): EngineSize, Cylinders, etc
    - Dependent variable (y): co2emission

## Linear regression model representation



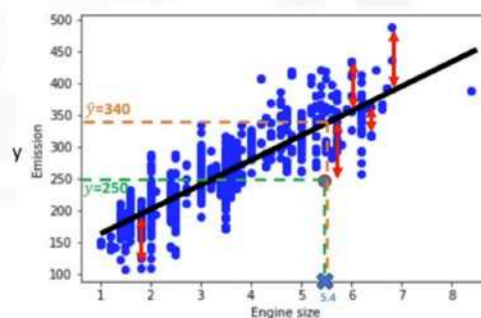
## How to find the best fit?

$x_1 = 5.4$  independent variable  
 $y = 250$  actual Co2 emission of  $x_1$

$\hat{y} = \theta_0 + \theta_1 x_1$   
 $\hat{y} = 340$  the predicted emission of  $x_1$

Error =  $y - \hat{y}$   
=  $250 - 340$   
=  $-90$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$



El objetivo de la regresión lineal es minimizar el MSE con Tita0 y Tita1. Como encontramos estos parámetros de forma tal que se minimice el MSE?

Tenemos dos opciones acá.. podemos usar un approach matemático o uno de optimización.

De forma matemática, podemos calcular la media de x e y y luego despejar los valores de los parámetros de la ecuación lineal:

### Estimating the parameters

	ENGINE SIZE	CYLINDERS	FUEL CONSUMPTION_COMB	CO2 EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267

$\hat{y} = \theta_0 + \theta_1 x_1$

$$\theta_1 = \frac{\sum_{i=1}^s (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^s (x_i - \bar{x})^2}$$

$$\bar{x} = (2.0 + 2.4 + 1.5 + \dots) / 9 = 3.03$$

$$\bar{y} = (196 + 221 + 136 + \dots) / 9 = 226.22$$

$$\theta_1 = \frac{(2.0 - 3.03)(196 - 226.22) + (2.4 - 3.03)(221 - 226.22) + \dots}{(2.0 - 3.03)^2 + (2.4 - 3.03)^2 + \dots}$$

$$\theta_1 = 39$$

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

$$\theta_0 = 226.22 - 39 * 3.03$$

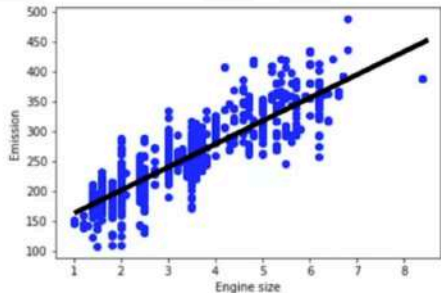
$$\theta_0 = 125.74$$

Se lo llama BIAS COEFFICIENT!

$$\hat{y} = 125.74 + 39x_1$$

### Pros of linear regression

- Very fast
- No parameter tuning
- Easy to understand, and highly interpretable



The scatter plot shows a positive correlation between Engine size (x-axis, ranging from 1 to 8) and Emission (y-axis, ranging from 100 to 500). A black line represents the linear regression fit, showing that as engine size increases, the emission also tends to increase.

# Model evaluation approaches

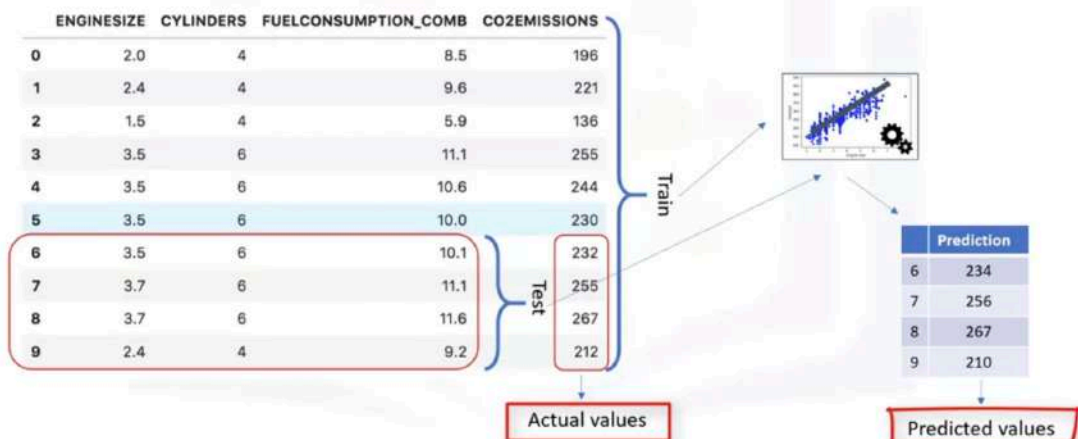
- • Train and Test on the Same Dataset
- • Train/Test Split

Regression Evaluation Metrics



Que tanto podemos confiar en nuestro modelo? Que nivel de precisión tiene? Una de las soluciones para responder a esto es seleccionar una parte del dataset para testing.

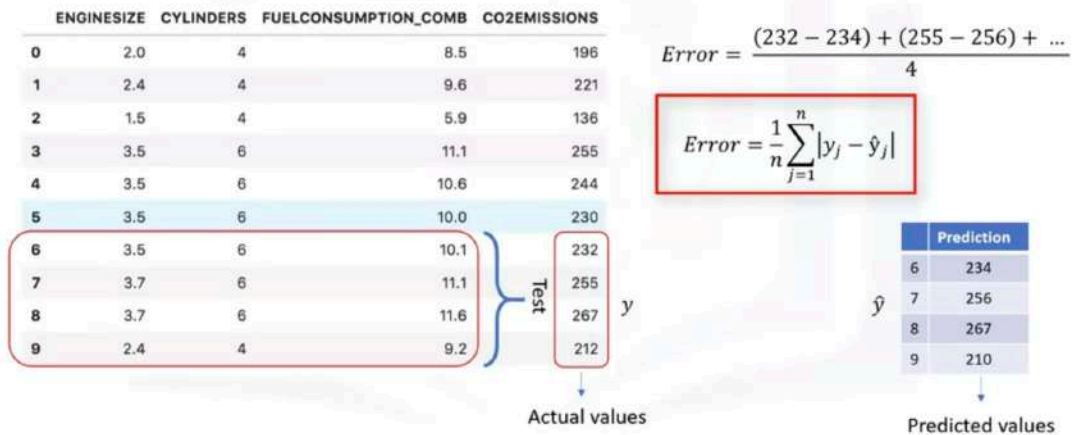
## Best approach for most accurate results?



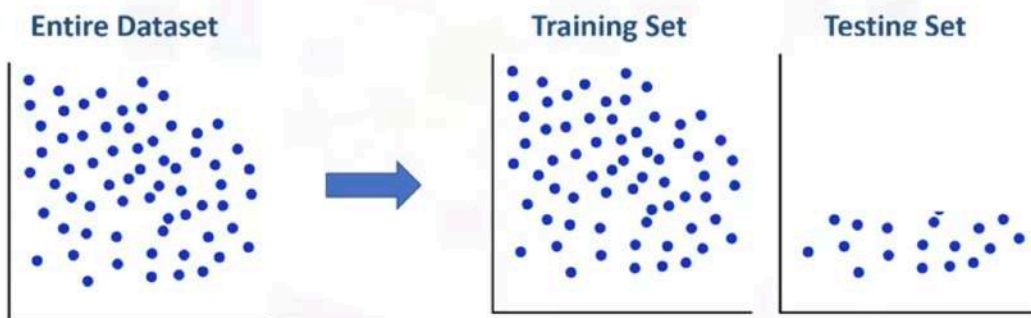
Métricas para medir el desempeño de nuestros modelos:



## Calculating the accuracy of a model



## Train and test on the same dataset



Este primer enfoque implica:

High "training accuracy"  
Low "out-of-sample accuracy"

Esto significa que no va a ser tan preciso contra samples que este fuera de la muestra inicial. Donde:

## What is training & out-of-sample accuracy?

### • Training Accuracy

- High training accuracy isn't necessarily a good thing
- Result of over-fitting
  - **Over-fit**: the model is overly trained to the dataset, which may capture noise and produce a non-generalized model

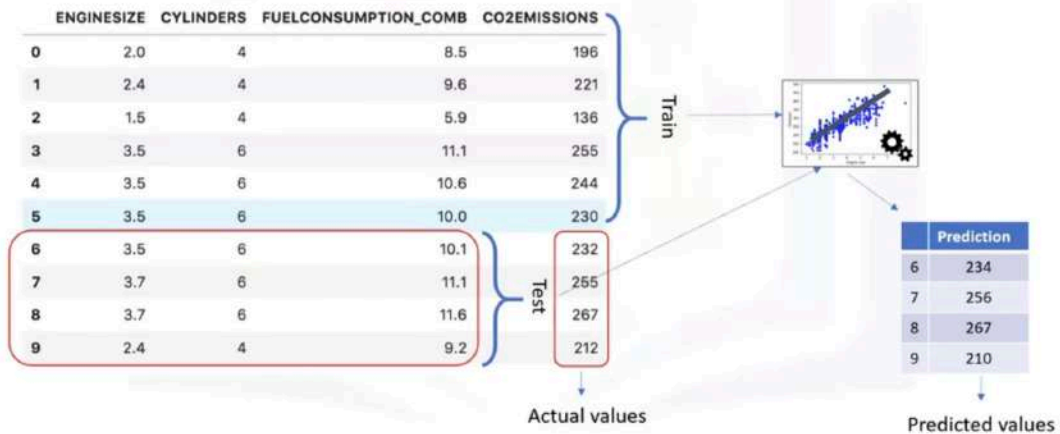
### • Out-of-Sample Accuracy

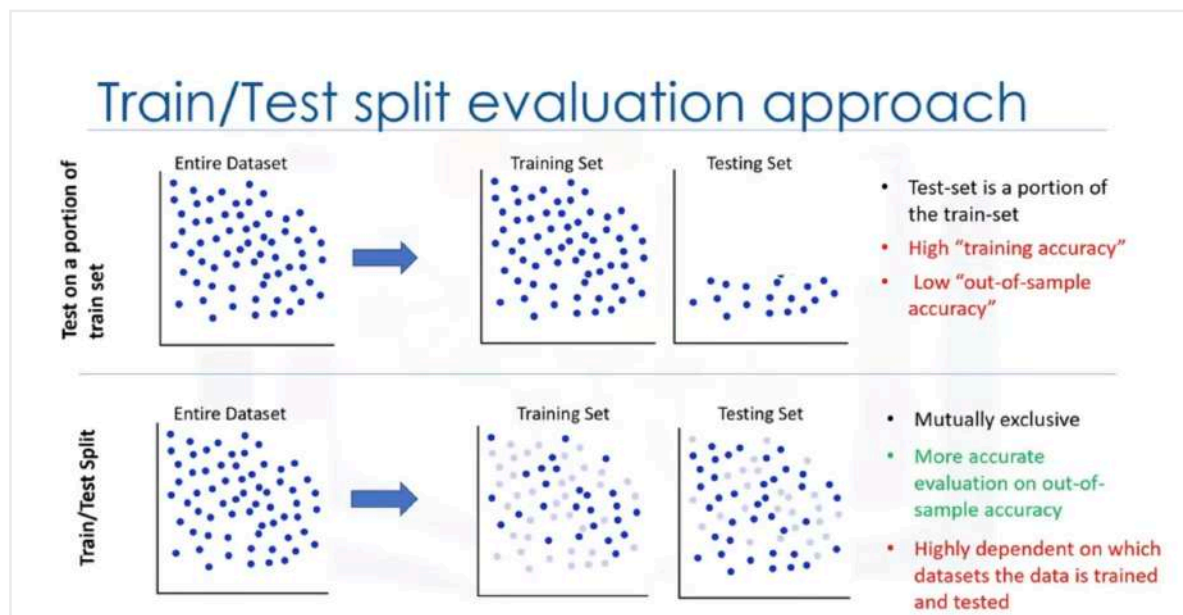
- It's important that our models have a high, out-of-sample accuracy
- How can we improve out-of-sample accuracy?

**Por el contrario, con el enfoque de Train Test Split:**

Acá NO tomo todo para entrenar como en el caso anterior.

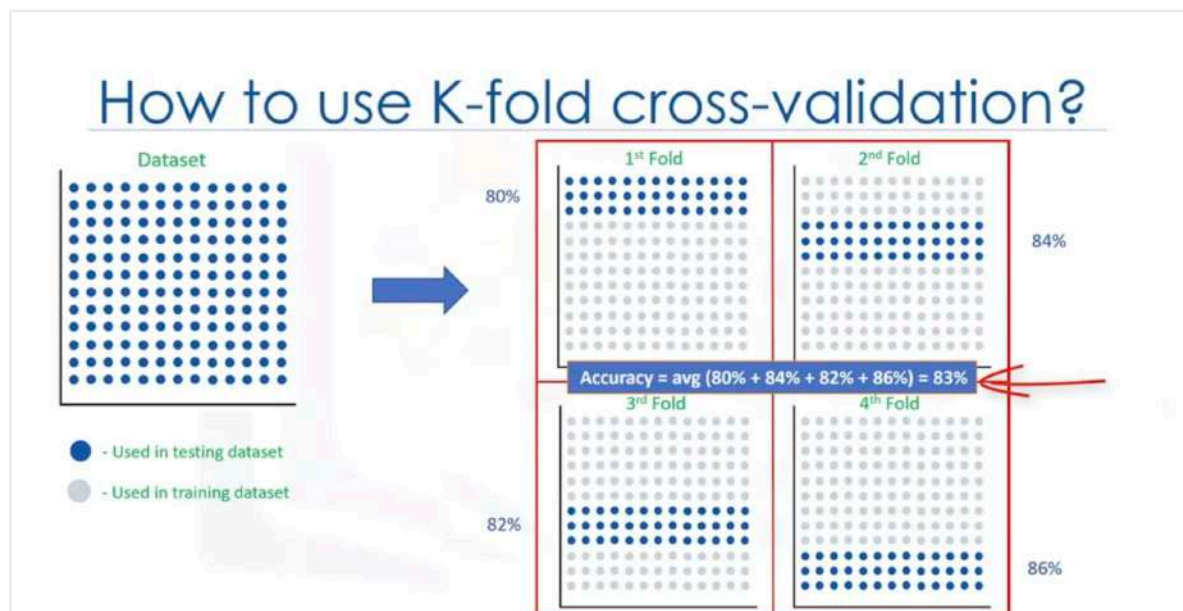
## Train/Test split evaluation approach





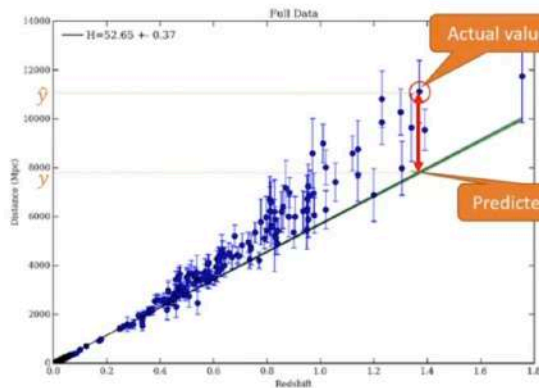
Importante! Dice que una vez que evalúes el modelo con el testing set, también lo uses para entrenar al modelo para no perder esa data. También dice que es altamente dependiente de los datasets en los cuales la data es entrenada y testeada. Si bien performa mejor que el enfoque anterior, sigue teniendo problemas de out of sample.

—> Acá entra el **K-Fold cross-validation** que soluciona bastantes de estos problemas:



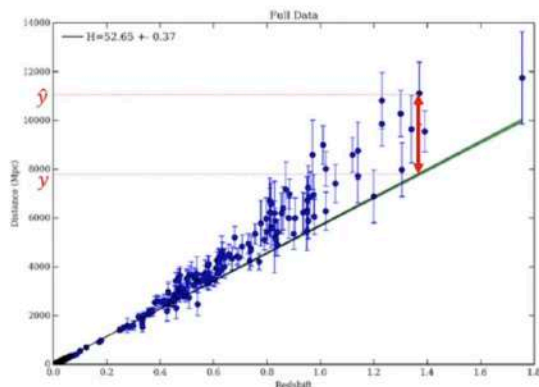
Tremendo. Haces K pliegues de tu dataset. Agarras un 25% que va a ser para testing y el resto para entrenar. Entrenas, medís y volver a avanzar para otro pliego haciendo lo mismo. Solo que en el siguiente pliego *la data que uses para test-train* va a ser distinta que la usaste para las vueltas anteriores. Terminas haciendo un promedio de todos los rendimientos que sacaste.

## What is an error of the model?



Error: measure of how far the data is from the fitted regression line.

## What is an error of the model?



$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$$

Le da mas importancia a los errores grandes, ya que crecen con el cuadrado

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

Comodo pq labura en mismas unidades q y

$$RAE = \frac{\sum_{j=1}^n |y_j - \hat{y}_j|}{\sum_{j=1}^n |y_j - \bar{y}|}$$

Normalizan

$$RSE = \frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{\sum_{j=1}^n (y_j - \bar{y})^2}$$

Se usa para el calculo del r^2

El de siempre de la RL  $\leftarrow R^2 = 1 - RSE$

Cuanto mas grande el  $R^2$ , mejor fittea mi data con el modelo.

> Hands on Lab con Jupiter y el dataframe de las emisiones



ML0101EN-Reg-Simple-Lin
Python

**Creating train and test dataset**

Train/Test Split involves splitting the dataset into training and testing sets that are mutually exclusive. After which, you train with the training set and test with the testing set. This will provide a more accurate evaluation on out-of-sample accuracy because the testing dataset is not part of the dataset that have been used to train the model. Therefore, it gives us a better understanding of how well our model generalizes on new data.

This means that we know the outcome of each data point in the testing dataset, making it great to test with! Since this data has not been used to train the model, the model has no knowledge of the outcome of these data points. So, in essence, it is truly an out-of-sample testing.

Let's split our dataset into train and test sets. 80% of the entire dataset will be used for training and 20% for testing. We create a mask to select random rows using `np.random.rand()` function:

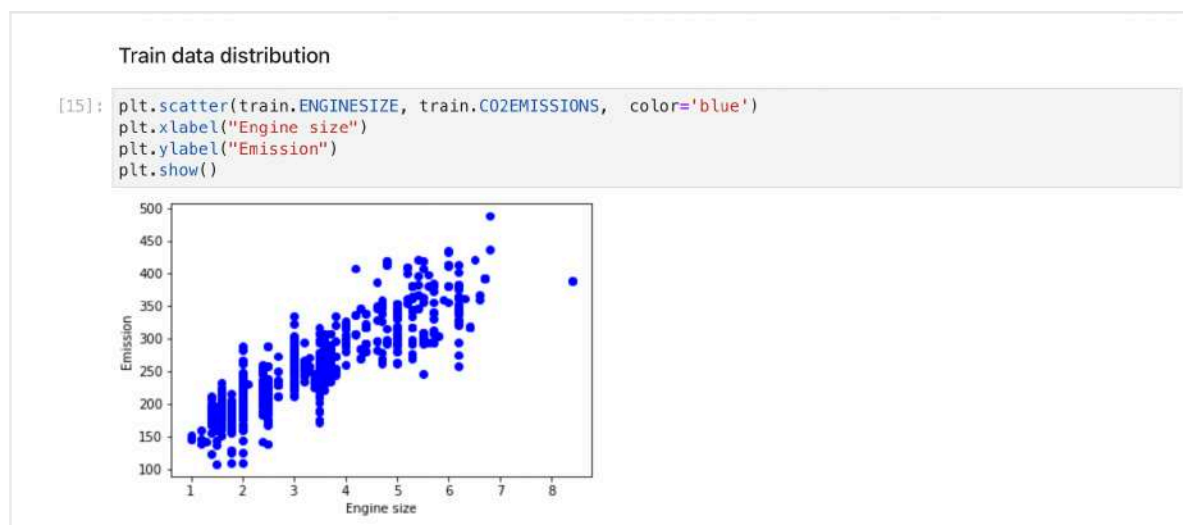
```
[14]: msk = np.random.rand(len(df)) < 0.8 # aca no entiendo bien q esta haciendo, esta tomando numeros al azar d
train = cdf[msk]#mask
test = cdf[~msk]# not tha mask
```

**Simple Regression Model**

Linear Regression fits a linear model with coefficients  $B = (B_1, \dots, B_n)$  to minimize the 'residual sum of squares' between the actual value  $y$  in the dataset, and the predicted value  $\hat{y}$  using linear approximation.

**Train data distribution**

```
[15]: plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS, color='blue')
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```



**Modeling**

Using sklearn package to model data.

```
[16]: from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x = np.asanyarray(train[['ENGINE SIZE']])
train_y = np.asanyarray(train[['CO2 EMISSIONS']])
regr.fit(train_x, train_y)
# The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ', regr.intercept_)

Coefficients: [[39.23425345]]
Intercept: [124.51694159]
```



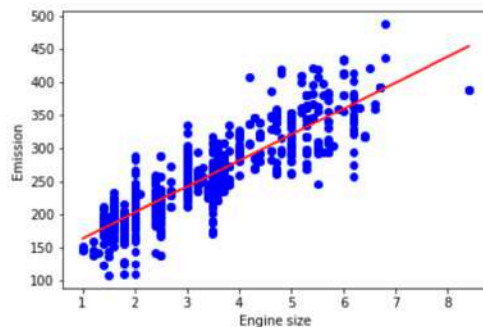
As mentioned before, **Coefficient** and **Intercept** in the simple linear regression, are the parameters of the fit line. Given that it is a simple linear regression, with only 2 parameters, and knowing that the parameters are the intercept and slope of the line, sklearn can estimate them directly from our data. Notice that all of the data must be available to traverse and calculate the parameters.

### Plot outputs

We can plot the fit line over the data:

```
[17]: plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS, color='blue')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-r')
plt.xlabel("Engine size")
plt.ylabel("Emission")
```

```
[17]: Text(0, 0.5, 'Emission')
```



### Evaluation

We compare the actual values and predicted values to calculate the accuracy of a regression model. Evaluation metrics provide a key role in the development of a model, as it provides insight to areas that require improvement.

There are different model evaluation metrics, let's use MSE here to calculate the accuracy of our model based on the test set:

- Mean Absolute Error: It is the mean of the absolute value of the errors. This is the easiest of the metrics to understand since it's just average error.
- Mean Squared Error (MSE): Mean Squared Error (MSE) is the mean of the squared error. It's more popular than Mean Absolute Error because the focus is geared more towards large errors. This is due to the squared term exponentially increasing larger errors in comparison to smaller ones.
- Root Mean Squared Error (RMSE).
- R-squared is not an error, but rather a popular metric to measure the performance of your regression model. It represents how close the data points are to the fitted regression line. The higher the R-squared value, the better the model fits your data. The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse).

```
*[18]: from sklearn.metrics import r2_score

test_x = np.asarray(test[['ENGINE_SIZE']])
test_y = np.asarray(test[['CO2EMISSIONS']])
test_y_ = regr.predict(test_x) # NOTAR COMO ACA EN VEZ DE FIT PARA ENCONTRAR LOS PARAMETROS AGARRO Y TIRO
# PREDICT

print("Mean absolute error: %.2f" % np.mean(np.absolute(test_y_ - test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y) ** 2))
print("R2-score: %.2f" % r2_score(test_y_, test_y_))
```

```
Mean absolute error: 21.32
Residual sum of squares (MSE): 831.25
R2-score: 0.78
```

## Exercise

Lets see what the evaluation metrics are if we trained a regression model using the `FUELCONSUMPTION_COMB` feature.

Start by selecting `FUELCONSUMPTION_COMB` as the `train_x` data from the `train` dataframe, then select `FUELCONSUMPTION_COMB` as the `test_x` data from the `test` dataframe

```
[35]: train_x = train[["FUELCONSUMPTION_COMB"]]  
test_x = train[["FUELCONSUMPTION_COMB"]]
```

▼ Click here for the solution

```
train_x = train[["FUELCONSUMPTION_COMB"]]
```

```
test_x = train[["FUELCONSUMPTION_COMB"]]
```

Now train a Logistic Regression Model using the `train_x` you created and the `train_y` created previously

```
[36]: regr = linear_model.LinearRegression()  
regr.fit(train_x, train_y)
```

```
[36]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,  
normalize=False)
```

► Click here for the solution

Find the predictions using the model's `predict` function and the `test_x` data

```
[42]: predictions = regr.predict(test_x)
```

► Click here for the solution

Finally use the `predictions` and the `test_y` data and find the Mean Absolute Error value using the `np.absolute` and `np.mean` function like done previously

```
[45]: test_y = train[["CO2EMISSIONS"]]  
print("Mean Absolute Error: %.2f" % np.mean(np.absolute(predictions - test_y)))
```

Mean Absolute Error: 20.80

► Click here for the solution

We can see that the MAE is much worse than it is when we train using `ENGINE SIZE`



## Multiple Linear Regression

# Multiple Linear Regression

## Types of regression models

- Simple Linear Regression
  - Predict **Co2emission** vs **EngineSize** of all cars
    - Independent variable (x): EngineSize
    - Dependent variable (y): Co2emission
- Multiple Linear Regression ★
  - Predict **Co2emission** vs **EngineSize** and **Cylinders** of all cars
    - Independent variable (x): EngineSize, Cylinders, etc.
    - Dependent variable (y): Co2emission

Aplicaciones:

## Examples of multiple linear regression

- Independent variables effectiveness on prediction
  - Does revision time, test anxiety, lecture attendance and gender have any effect on the exam performance of students?
- • Predicting impacts of changes
  - How much does blood pressure go up (or down) for every unit increase (or decrease) in the BMI of a patient?

Ver el impacto de las variables sobre la Y, cuando mantenemos las otras constantes.

## Predicting continuous values with multiple linear regression

X: Independent variable      Y: Dependent variable

$$Co2Em = \theta_0 + \theta_1 Engine\ size + \theta_2 Cylinders + \dots$$

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

$$\hat{y} = \theta^T X$$

$$\theta^T = [\theta_0, \theta_1, \theta_2, \dots]$$

$$X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \dots \end{bmatrix}$$

	ENGINE SIZE	CYLINDERS	FUEL CONSUMPTION_COMB	CO2 EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

## Using MSE to expose the errors in the model

$$\hat{y} = \theta^T X$$

$$\hat{y}_i = 140$$

the predicted emission of  $x_i$

$$y_i = 196$$

actual value of  $x_i$

$$y_i - \hat{y}_i = 196 - 140 = 56 \quad \text{residual error}$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

	ENGINE SIZE	CYLINDERS	FUEL CONSUMPTION_COMB	CO2 EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267

Como calculamos los parámetros de la regresión multinomial para minimizar el error?... cuadrados mínimos no me la nomelacontainer oooooo el descenso del gradiente !

## Estimating multiple linear regression parameters

- How to estimate  $\theta$ ?
  - Ordinary Least Squares
    - Linear algebra operations
    - Takes a long time for large datasets (10K+ rows)
  - An optimization algorithm
    - Gradient Descent
    - Proper approach if you have a very large dataset

## Making predictions with multiple linear regression

	ENGINE SIZE	CYLINDERS	FUEL CONSUMPTION_COMB	CO2 EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

$$\hat{y} = \theta^T X$$

$$\theta^T = [125, 6.2, 14, \dots]$$

$$\hat{y} = 125 + 6.2x_1 + 14x_2 +$$

$$Co2Em = 125 + 6.2 EngSize + 14 Cylinders + \dots$$

Ojota, agregar miles de variables a una múltiple linear reg sin ningún tipo de back teórico por lo general nos perjudica y nos overfittea.



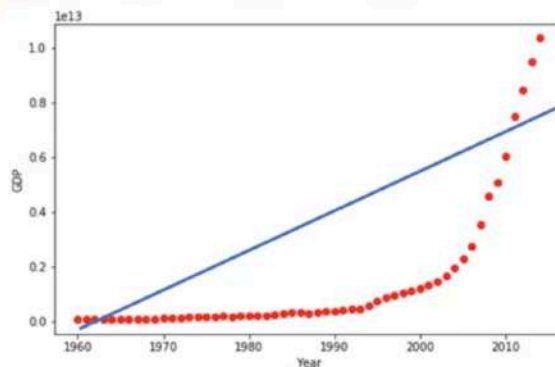
## Non Linear Regression

Cuando tenemos data correlacionada que No se comporta de forma lineal, claramente no podemos usar linear regresión!



## Should we use linear regression?

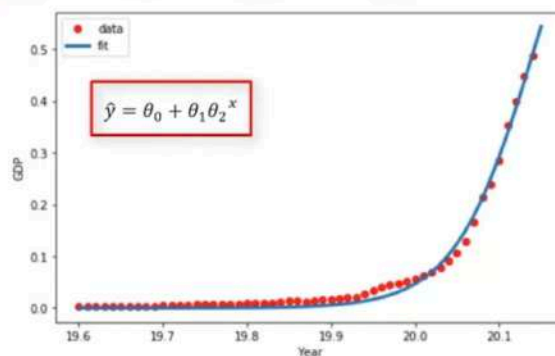
	Year	Value
0	1960	5.918412e+10
1	1961	4.955705e+10
2	1962	4.668518e+10
3	1963	5.009730e+10
4	1964	5.906225e+10
5	1965	6.970915e+10
6	1966	7.587943e+10
7	1967	7.205703e+10
8	1968	6.999350e+10
9	1969	7.871882e+10
...	.....	.....



Tiene mas pinta de una función exponencial o una logística, entonces necesitamos otro método:

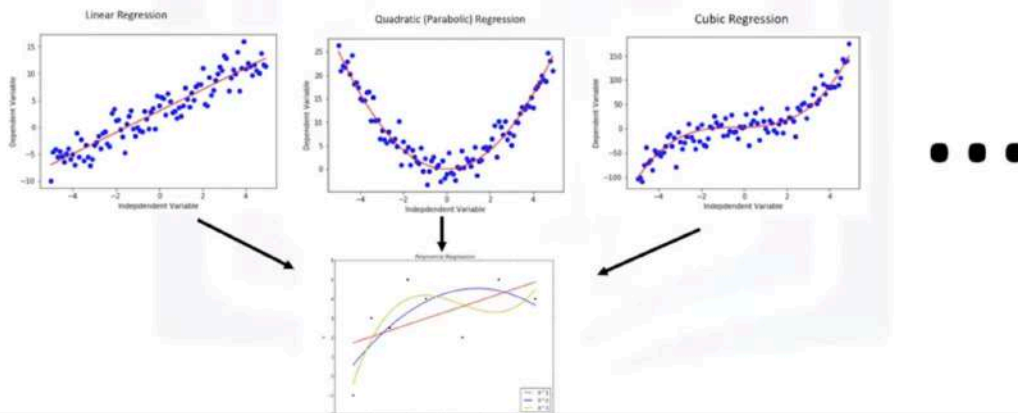
## Should we use linear regression?

	Year	Value
0	1960	5.918412e+10
1	1961	4.955705e+10
2	1962	4.668518e+10
3	1963	5.009730e+10
4	1964	5.906225e+10
5	1965	6.970915e+10
6	1966	7.587943e+10
7	1967	7.205703e+10
8	1968	6.999350e+10
9	1969	7.871882e+10
...	.....	.....



En resumen, cualquier modelo que no sea lineal lo podemos llamar polinomios:

## Different types of regression



## What is polynomial regression?

- Some curvy data can be modeled by a **polynomial regression**
- For example:

$$\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

- A polynomial regression model can be transformed into linear regression model.

$$\begin{aligned} x_1 &= x \\ x_2 &= x^2 \\ x_3 &= x^3 \end{aligned}$$

*"Convertis" un modelo polinomial a lineal con un cambio de variable*

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

*Podes usar el mismo mecanismo de multiple linear reg con polinmoiales que entren en eso*



Minimizing the sum of the squares of the differences between  $y$  and  $\hat{y}$

# What is polynomial regression?

- Some curvy data can be modeled by a **polynomial regression**
- For example:

$$\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

- A polynomial regression model can be transformed into linear regression model.

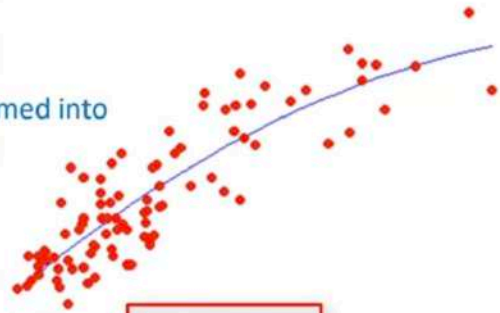
$$\begin{aligned}x_1 &= x \\x_2 &= x^2 \\x_3 &= x^3\end{aligned}$$

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

→ Multiple linear regression

→ Least Squares

Minimizing the sum of the squares of the differences between  $y$  and  $\hat{y}$



# What is non-linear regression?

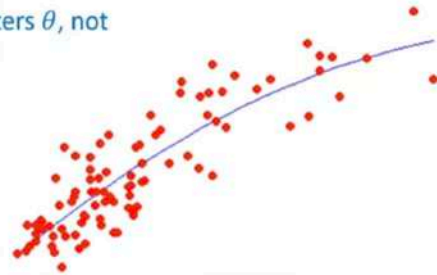
- To model non-linear relationship between the dependent variable and a set of independent variables
- $\hat{y}$  must be a non-linear function of the parameters  $\theta$ , not necessarily the features  $x$

$$\hat{y} = \theta_0 + \theta_2^2 x$$

$$\hat{y} = \theta_0 + \theta_1 \theta_2^x$$

$$\hat{y} = \log(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3)$$

$$\hat{y} = \frac{\theta_0}{1 + \theta_1^{(x - \theta_2)}}$$



En contraste, en las non linear, no podemos usar el modelo de regresión lineal para estimar los parámetros. (Cuadrados mínimos) y tenemos que usar en cambio otros métodos. Además, por lo general estimar los parámetros en estos casos NO es fácil.

## What is non-linear regression?

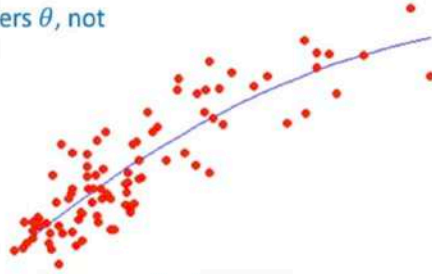
- To model non-linear relationship between the dependent variable and a set of independent variables
- $\hat{y}$  must be a non-linear function of the parameters  $\theta$ , not necessarily the features  $x$

$$\hat{y} = \theta_0 + \theta_2^2 x$$

$$\hat{y} = \theta_0 + \theta_1 \theta_2^x$$

$$\hat{y} = \log(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3)$$

$$\hat{y} = \frac{\theta_0}{1 + \theta_1^{(x-\theta_2)}}$$



Como darnos cuenta cuando usar cuál?

# Si el coeficiente de correlación es  $> 0.7$  entonces no es apropiado usar una NO lineal ya que nos esta diciendo que la relación es prácticamente lineal.

## Linear vs non-linear regression

- How can I know if a problem is linear or non-linear in an easy way?
  - Inspect visually
  - Based on accuracy
- How should I model my data, if it displays non-linear on a scatter plot?
  - Polynomial regression
  - Non-linear regression model
  - Transform your data

## CLASSIFICATION

## What is classification?

- A supervised learning approach
- Categorizing some unknown items into a discrete set of categories or “classes”
- The target attribute is a categorical variable

## How does classification work?

**Classification** determines the class label for an unlabeled test case.

age	ed	employ	address	income	debtinc	creddebt	othdebt	default
41	3	17	12	176	9.3	11.359	5.009	1
27	1	10	6	31	17.3	1.362	4.001	0
40	1	15	14	55	5.5	0.856	2.169	0
41	1	15	14	120	2.9	2.659	0.821	0
24	2	2	0	28	17.3	1.787	3.057	1
41	2	5	5	25	10.2	0.393	2.157	0
39	1	20	9	67	30.6	3.834	16.668	0
43	1	12	11	38	3.6	0.129	1.239	0
24	1	3	4	19	24.4	1.358	3.278	1
36	1	0	13	25	19.7	2.778	2.147	0

Categorical Variable

age	ed	employ	address	income	debtinc	creddebt	othdebt	default
37	2	16	10	130	9.3	10.23	3.21	0



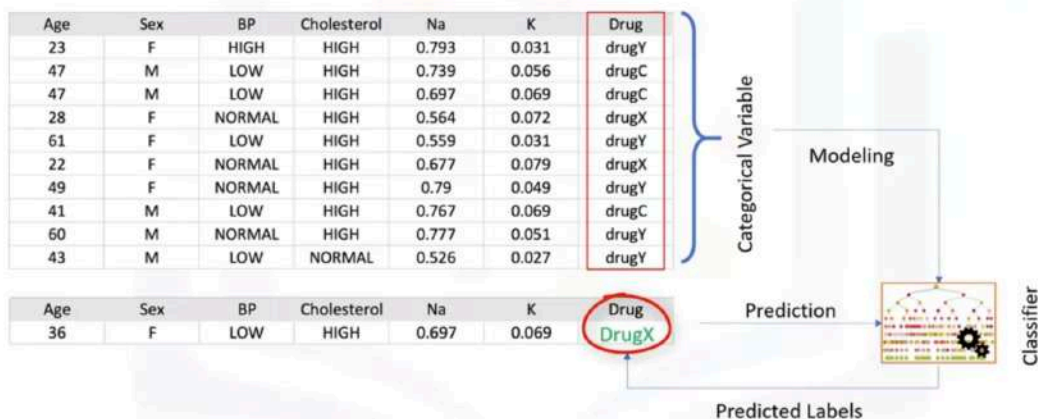
# How does classification work?

**Classification** determines the class label for an unlabeled test case.



El tipico caso es el de los deudores del banco. Notar que este ejemplo es un clasificador binario. O bien es deudor o no es deudor, tambien podríamos tener multiclass classification. Por ejemplo:

## Example of multi-class classification

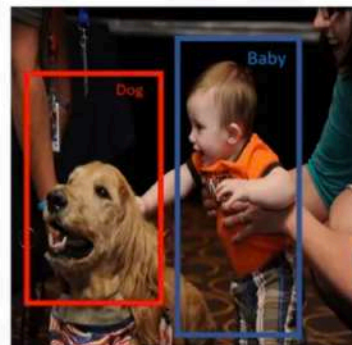


## Classification use cases

	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	Yes
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	Yes
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	No
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	No
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	?

- Which category a customer belongs to?
- Whether a customer switches to another provider/brand?
- Whether a customer responds to a particular advertising campaign?

## Classification applications



# Classification algorithms in machine learning

---

- Decision Trees (ID3, C4.5, C5.0)
- Naïve Bayes
- Linear Discriminant Analysis
- $k$ -Nearest Neighbor
- Logistic Regression
- Neural Networks
- Support Vector Machines (SVM)

Vamos a ver un par de estos algoritmos...

## ***KNN***

### K-Nearest Neighbours

En este ejemplo queremos predecir y clasificar la categoría (dentro de las 4) de cada consumidor de esta empresa energética según su data demográfica:

# Intro to KNN

X: Independent variable

Y: Dependent variable

	region	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	44	1	9	64	4	5	0	0	2	1
1	3	33	1	7	136	5	5	0	0	6	4
2	3	52	1	24	116	1	29	0	1	2	3
3	2	33	0	12	33	2	0	0	1	1	1
4	2	30	1	9	30	1	2	0	0	4	3
5	2	39	0	17	78	2	16	0	1	1	3
6	3	22	1	2	19	2	4	0	1	5	2
7	2	35	0	5	76	2	10	0	0	3	4
8	3	50	1	7	166	4	31	0	0	5	?

Value	Label
1	Basic Service
2	E-Service
3	Plus Service
4	Total Service

# Intro to KNN

X: Independent variable

Y: Dependent variable

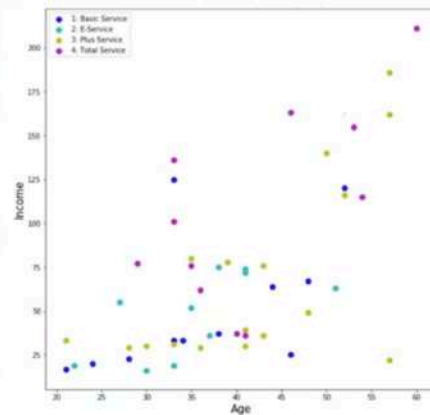
	region	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	44	1	9	64	4	5	0	0	2	1
1	3	33	1	7	136	5	5	0	0	6	4
2	3	52	1	24	116	1	29	0	1	2	3
3	2	33	0	12	33	2	0	0	1	1	1
4	2	30	1	9	30	1	2	0	0	4	3
5	2	39	0	17	78	2	16	0	1	1	3
6	3	22	1	2	19	2	4	0	1	5	2
7	2	35	0	5	76	2	10	0	0	3	4
8	3	50	1	7	166	4	31	0	0	5	?

Value	Label
*1	Basic Service
*2	E-Service
*3	Plus Service
*4	Total Service

Queremos usar las filas 0-7 para predecir la clasificación de la fila 8. Vamos a usar KNN. Como demostración, primero usemos 2 campos:

## Determining the class using 1<sup>st</sup> KNN

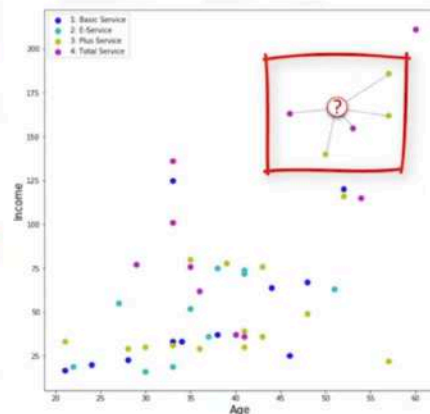
	region	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	44	1	9	64	4	5	0	0	2	1
1	3	33	1	7	136	5	5	0	0	6	4
2	3	52	1	24	116	1	29	0	1	2	3
3	2	33	0	12	33	2	0	0	1	1	1
4	2	30	1	9	30	1	2	0	0	4	3
5	2	39	0	17	78	2	16	0	1	1	3
6	3	22	1	2	19	2	4	0	1	5	2
7	2	35	0	5	76	2	10	0	0	3	4
8	3	50	1	7	166	4	31	0	0	5	?



Aca dice que puede decir que va a ser un clase 4 porque nearest neighbor es un clase 4 tambien. La joda es que tanto podemos confiar en esta decisi3n. Por ejemplo, nuestro nn podria ser un outlier o un caso muy especifico. Pero... si en vez de elegir un solo NN, que pasa si elegimos por ejemplo 5?

## Determining the class using the 5 KNNs

	region	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	44	1	9	64	4	5	0	0	2	1
1	3	33	1	7	136	5	5	0	0	6	4
2	3	52	1	24	116	1	29	0	1	2	3
3	2	33	0	12	33	2	0	0	1	1	1
4	2	30	1	9	30	1	2	0	0	4	3
5	2	39	0	17	78	2	16	0	1	1	3
6	3	22	1	2	19	2	4	0	1	5	2
7	2	35	0	5	76	2	10	0	0	3	4
8	3	50	1	7	166	4	31	0	0	5	?

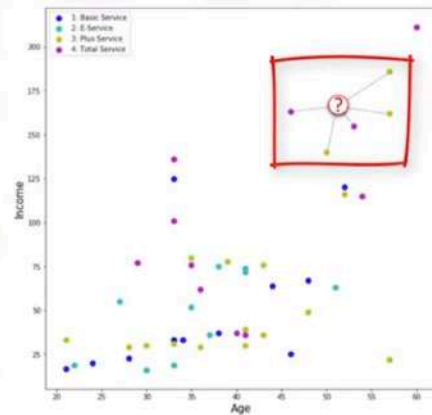


Y ahi hacemos un majority vote entre ellos para decidir su clase. Vemos que 3/5 son clase 3, entonces podemos decir que tiene mas sentido decir eso. Este seria un 5NN algorithm. Bueno, definamos el algoritmo:



## What is K-Nearest Neighbor (or KNN)?

- A method for **classifying** cases based on their similarity to other cases
- Cases that are near each other are said to be “**neighbors**”
- Based on **similar cases with same class labels** are near each other



Y como funciona?

## The K-Nearest Neighbors algorithm

1. Pick a value for **K**.
2. Calculate the distance of unknown case from all cases.
3. Select the K-observations in the training data that are “nearest” to the unknown data point.
4. Predict the response of the unknown data point using the most popular response value from the K-nearest neighbors.

Como podemos calcular la similitud entre dos datos? Y como calculamos el k optimo?

## Calculating the similarity/distance in a 1-dimensional space



Customer 1

Age
34



Customer 2

Age
30

$$\text{Dis}(x_1, x_2) = \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2}$$

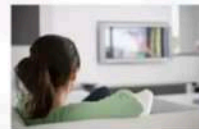
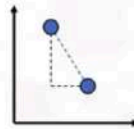
$$\text{Dis}(x_1, x_2) = \sqrt{(34 - 30)^2} = 4$$

## Calculating the similarity/distance in a 2-dimensional space



Customer 1

Age	Income
34	190



Customer 2

Age	Income
30	200

$$\text{Dis}(x_1, x_2) = \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2}$$

$$= \sqrt{(34 - 30)^2 + (190 - 200)^2} = 10.77$$

## Calculating the similarity/distance in a multi-dimensional space



Customer 1		
Age	Income	Education
34	190	3

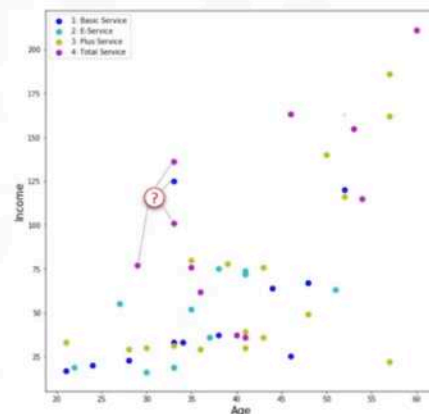
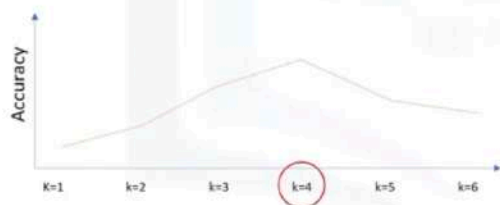


Customer 2		
Age	Income	Education
30	200	8

$$\begin{aligned}\text{Dis}(x_1, x_2) &= \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2} \\ &= \sqrt{(34 - 30)^2 + (190 - 200)^2 + (3 - 8)^2} = 11.87\end{aligned}$$

## What is the best value of K for KNN?

- K = 1 class 1
- K = 20 ?



La solución general es reservar una parte de la data para medir la accuracy y así probar cual es el k óptimo para usar.

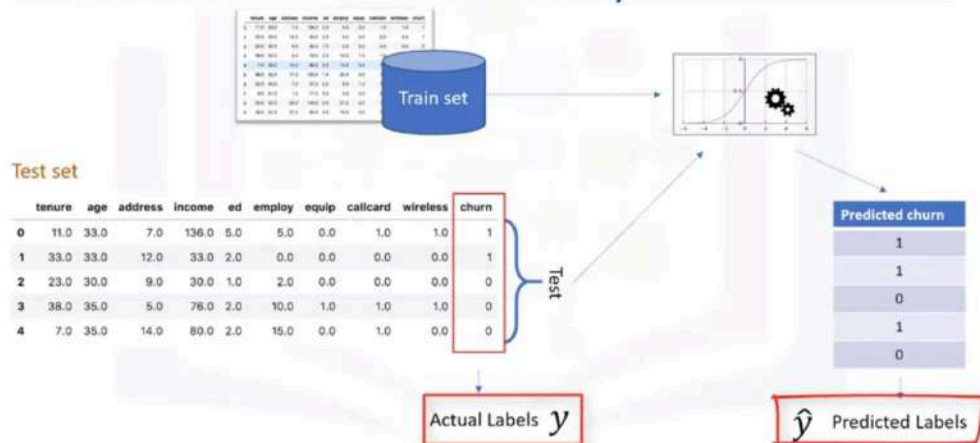
## Computing continuous targets using KNN

- KNN can also be used for regression



## Evaluation Metrics in Classification

## Classification accuracy



### Jaccard index:

Segun el tamaño de la intersección entre los valores posta t los predecidos:

## Jaccard index

$y$ : Actual labels

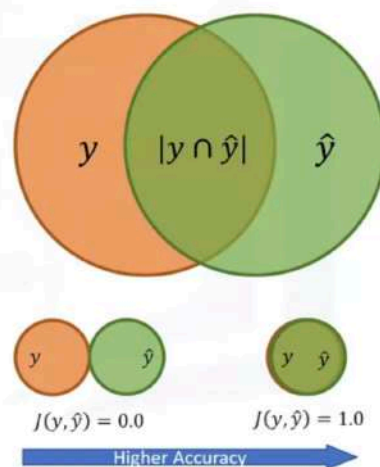
$\hat{y}$ : Predicted labels

$$J(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|} = \frac{|y \cap \hat{y}|}{|y| + |\hat{y}| - |y \cap \hat{y}|}$$

$y$ : [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]

$\hat{y}$ : [1, 1, 0, 0, 0, 1, 1, 1, 1, 1]

$$J(y, \hat{y}) = \frac{8}{10+10-8} = 0.66$$



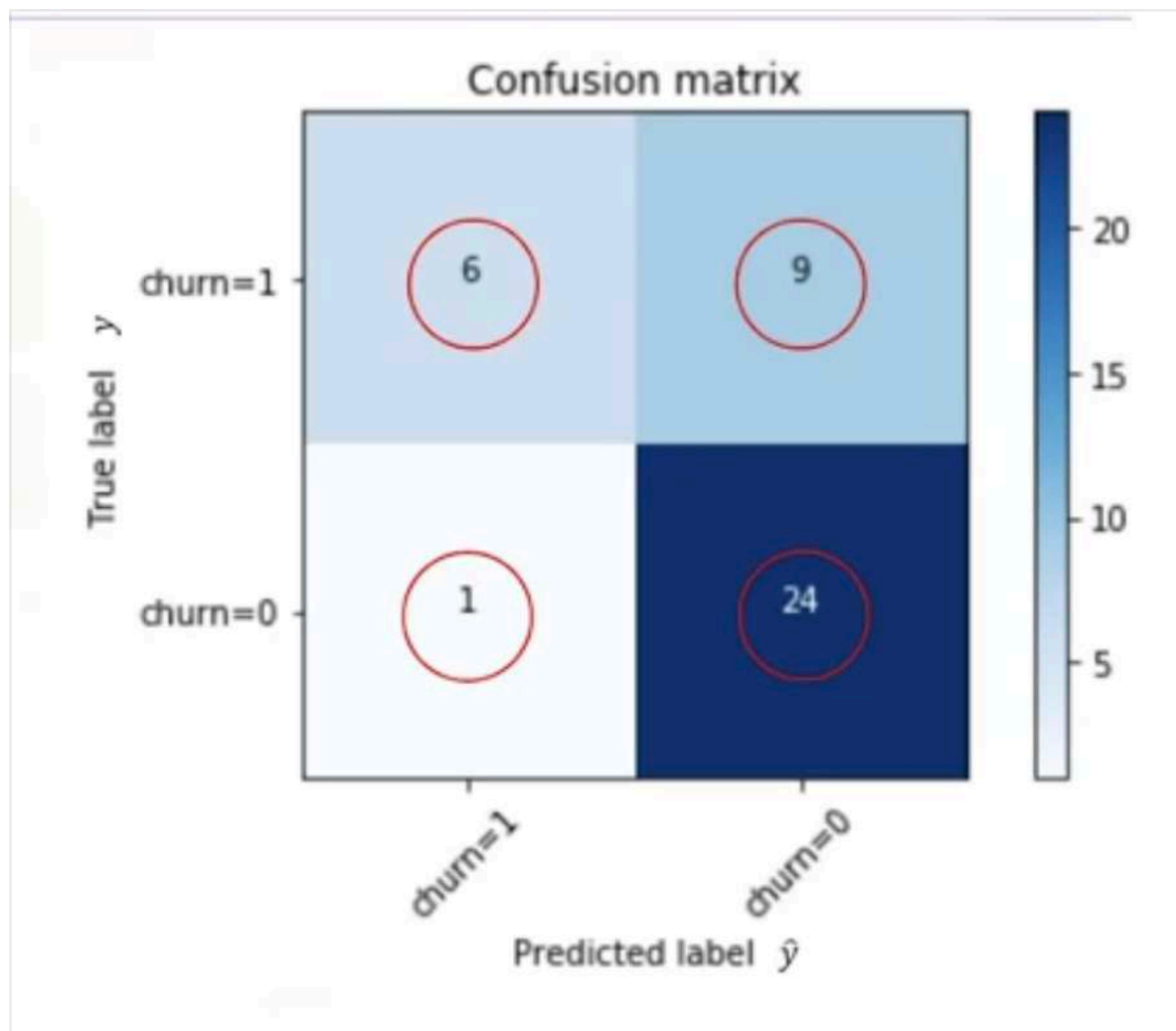
### F1-Score

Confusion matrix → cada fila contiene los valores posta de los loables del test set. Las columnas muestran los valores predecidos por classifier.

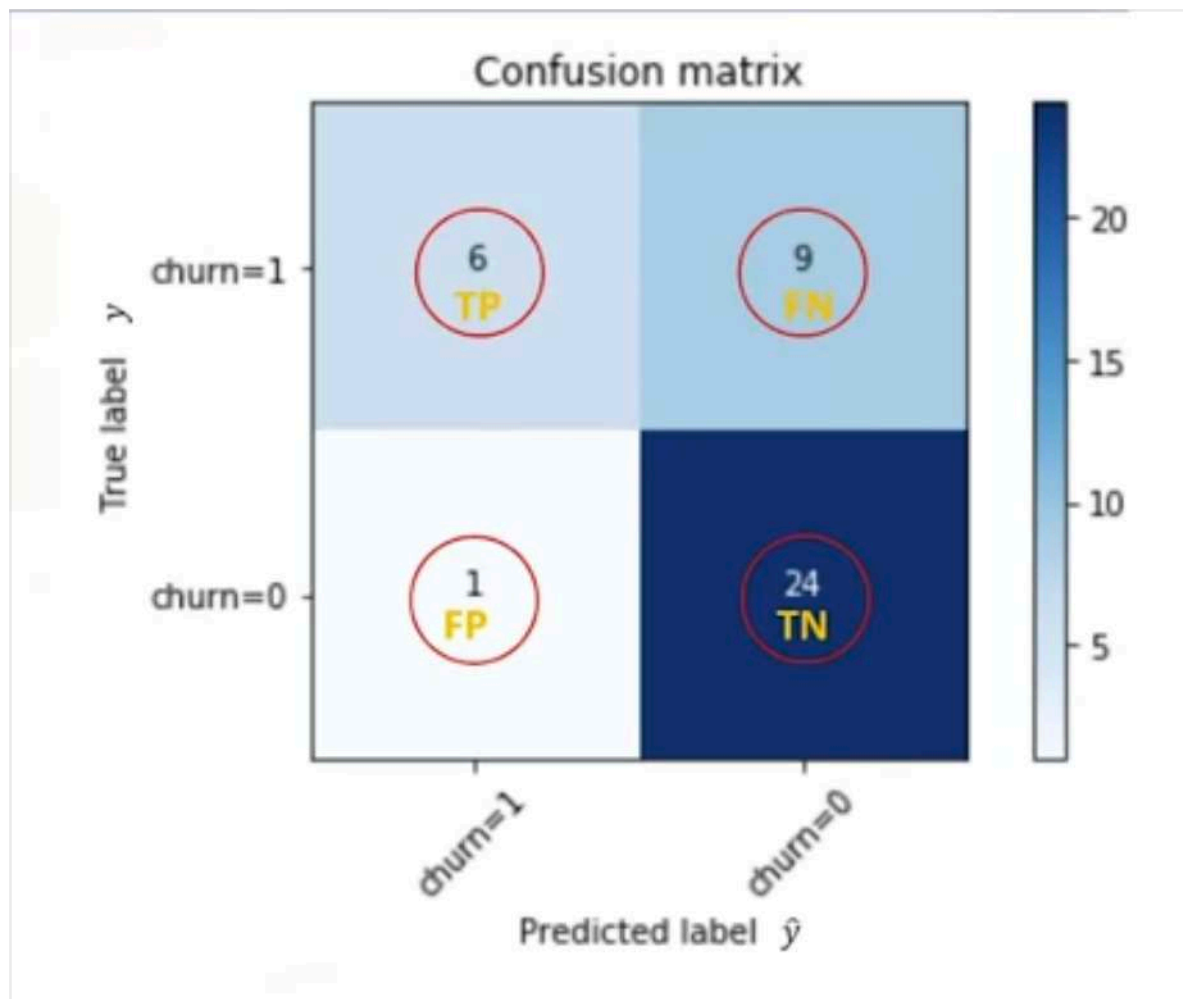
Por ejemplo, miremos la fila uno. Para esta fila, tenemos costumers cuyos churn valúes fueron 1 en el test set. De 40 costumers(6+9+24+1), el churn de 15 (6+9) de ellos es 1. Y de esos 15, 6 fueron predichos correctamente como uno por el classifier y 9 como cero. Esto significa que se equivoco en 9 el classifier.

Para churn = 0, teníamos 25 totales cuyo churn era cero. Y de estos predijo a 24 como cero y a 1 como 1. Osea que le fue mucho mejor aca

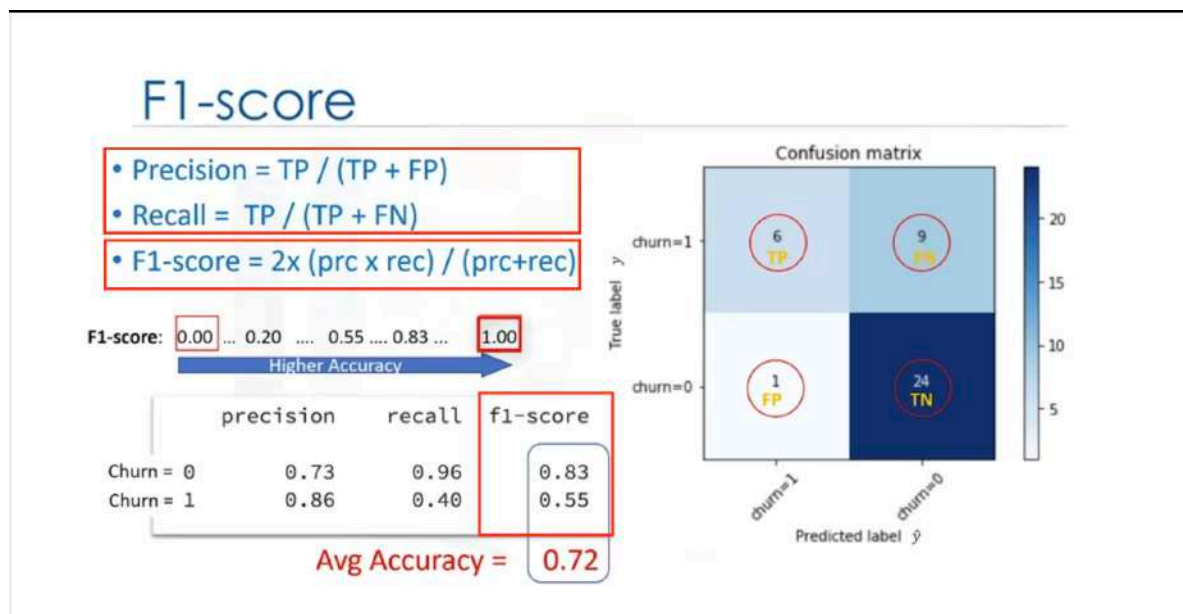




Nos muestra la habilidad del modelo para predecir correctamente y separar las clases. En el caso específico de un clasificador binario, como este ejemplo, podemos interpretar estos números como la cuenta de falsos positivos, negativos etc:



Con el recall y la precisión definidos, podemos también calcular el F1 score:



Tanto el Jaccard como el F1 pueden ser usados bien para Multi clasificadores o para binarios.

## LogLoss

A veces el output de mi classifier puede ser una probabilidad. Por ejemplo en este caso, la probabilidad de churn de mi customer.

