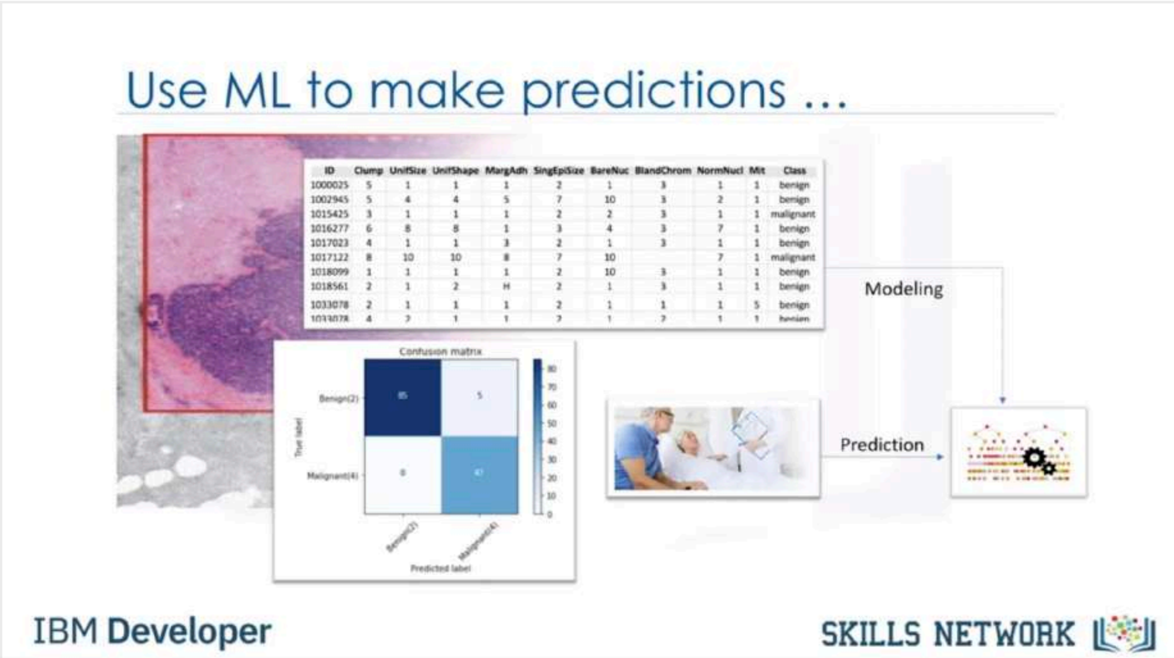
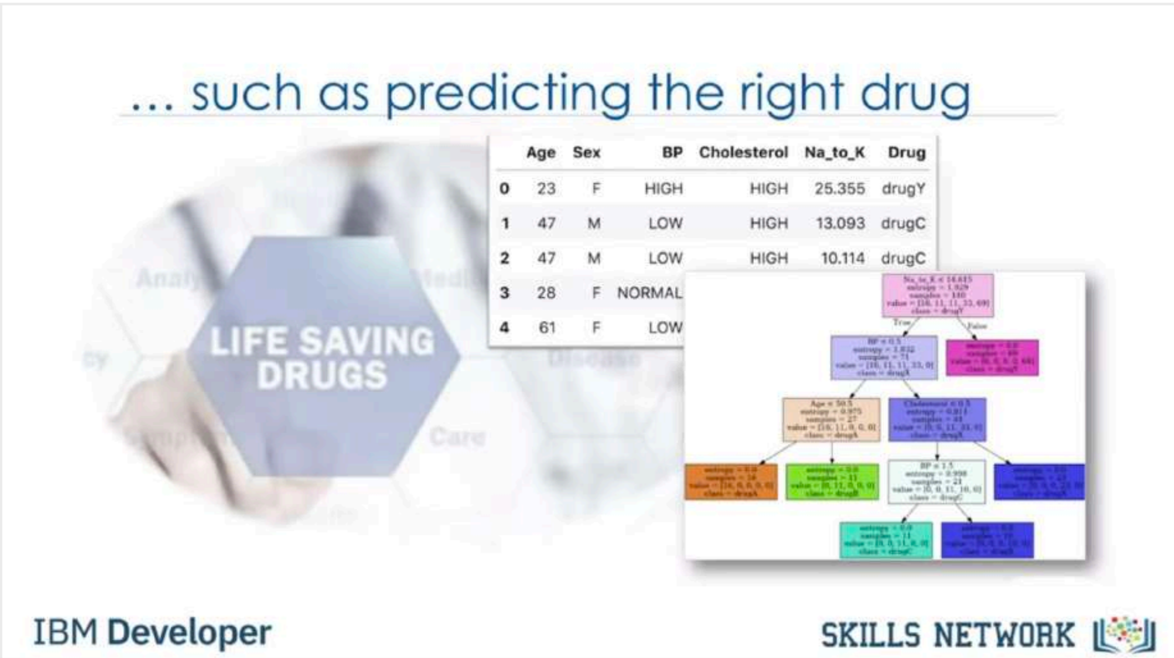
👨‍💻 IBM - DATA SCIENCE - MACHINE LEARNING WITH PYTHON 🐍🧠

Semana 1 - Introduction: What is Machine Learning?

—> What are we going to learn?





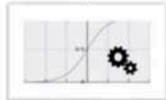
Use ML to make decisions

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			9	67	30.6	3.834	16.668	0
43				38	3.6	0.129	1.239	0
				19	24.4	1.358	3.278	1
				25	19.7	2.778	2.147	0

Categorical Variable

Modeling

...me	debtinc	creddebt	othdebt	default
30	9.3	10.23	3.21	No



Classifier

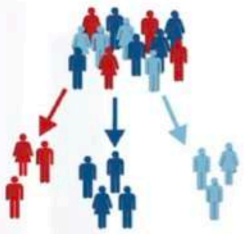
Predicted Labels

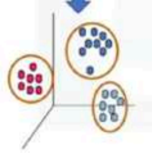
IBM Developer

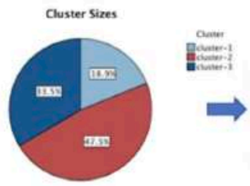
SKILLS NETWORK

Use ML for customer segmentation

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







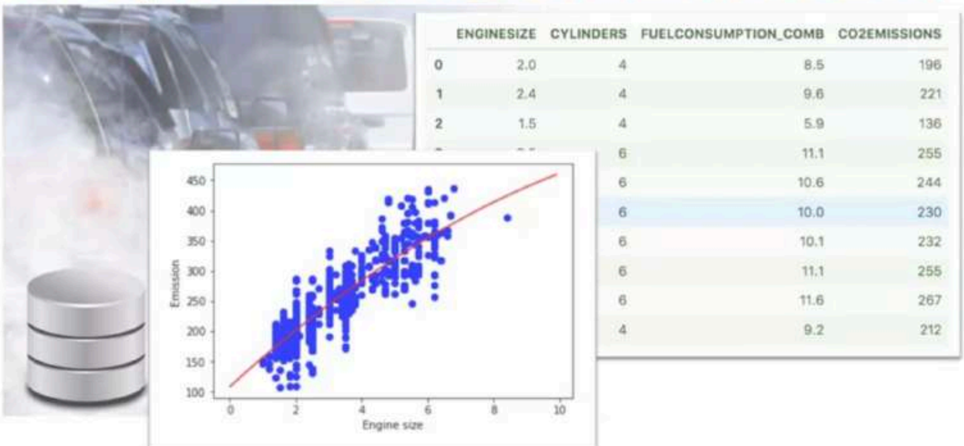
Cluster	Segment Name
cluster-1	AFFULUENT AND MIDDLE AGED
cluster-2	YOUNG EDUCATED AND MIDDLE INCOME
cluster-3	YOUNG AND LOW INCOME

IBM Developer

SKILLS NETWORK

Use ML for recommendation systems

Use Python libraries to build ML models



What do you get from this course?

SKILLS:

- Regression
- Classification
- Clustering
- Scikit Learn
- Scipy

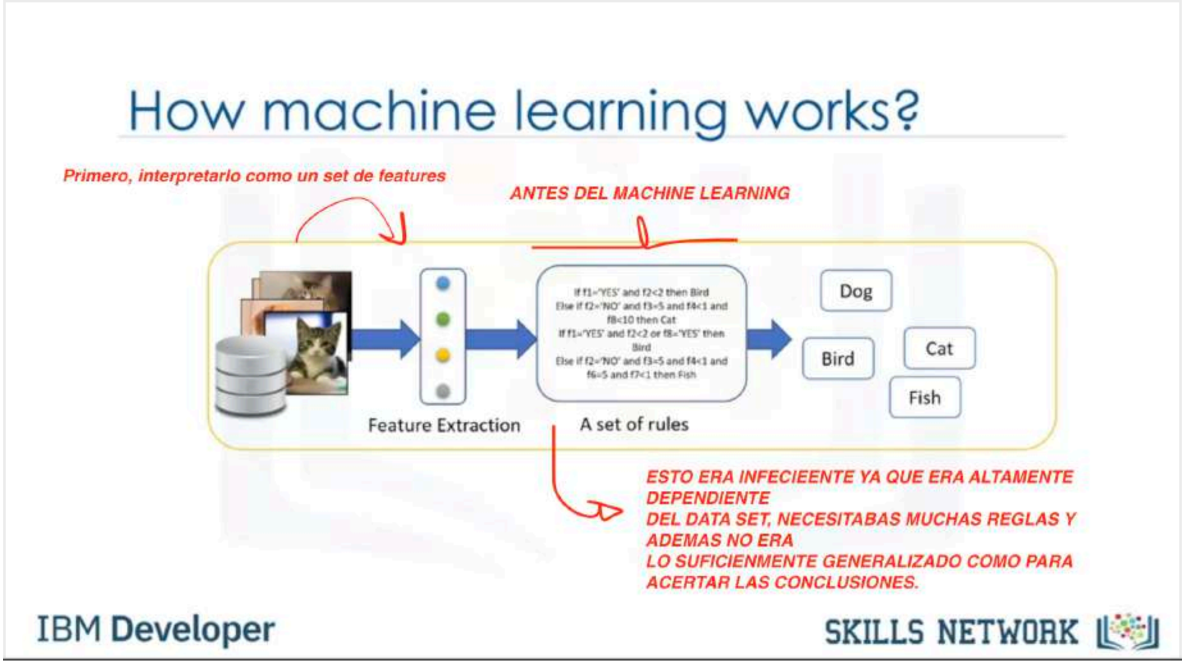
PROJECTS:

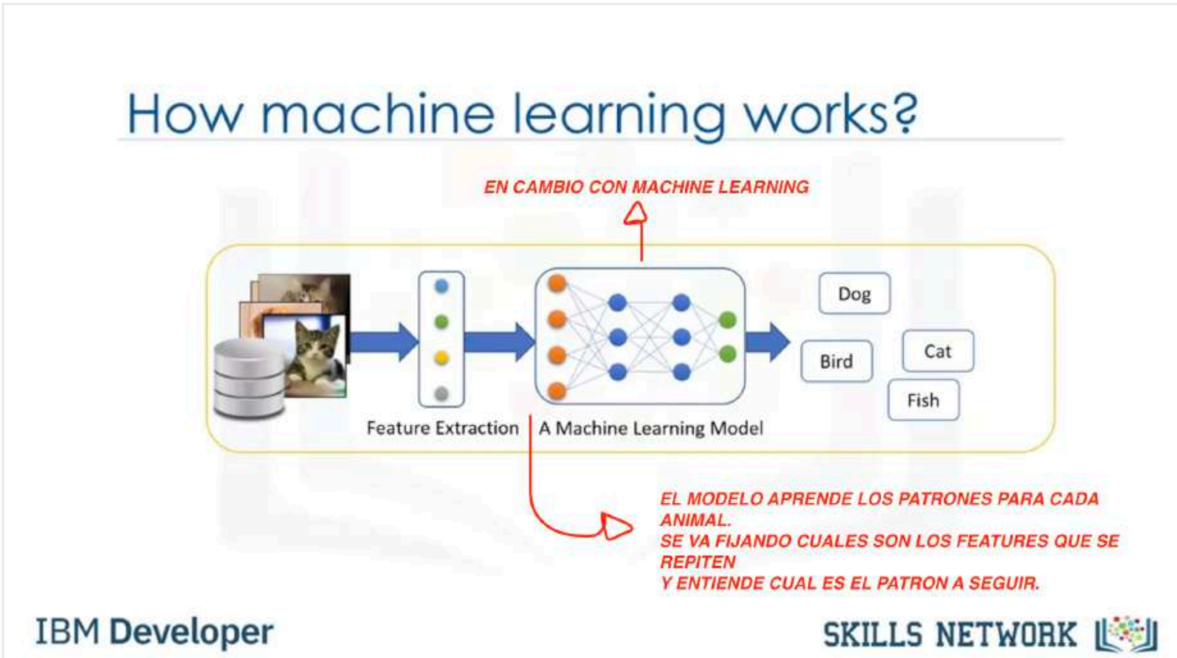
- Cancer detection
- Predicting economic trends
- Predicting customer churn
- Recommendation engines
- Many more ....

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.





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.

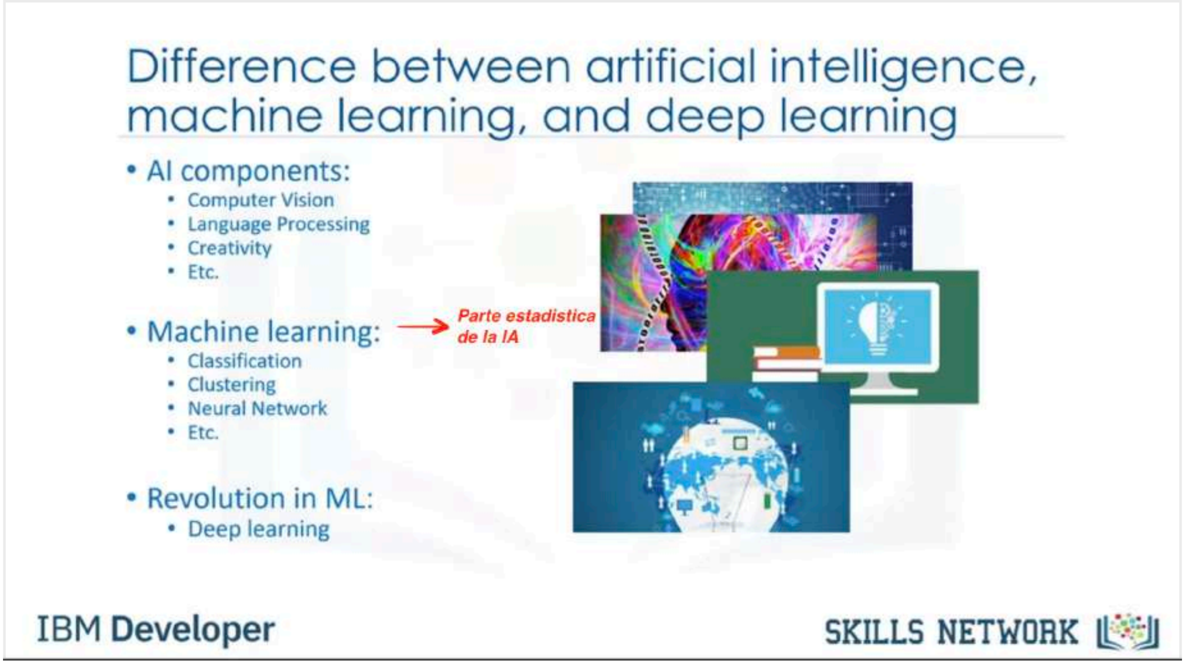
Dentro de las técnicas mas populares de ML encontramos:

Major machine learning techniques

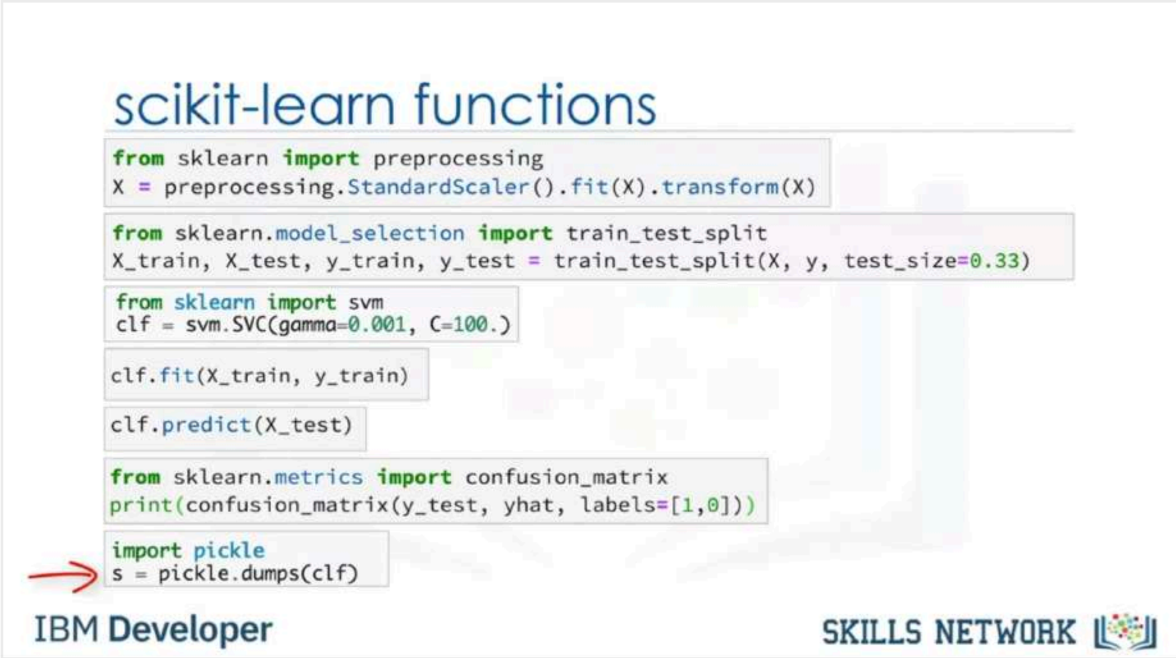
- Regression/Estimation
  - Predicting continuous values
- Classification
  - Predicting the item class/category of a case
- Clustering
  - Finding the structure of data; summarization
- Associations
  - Associating frequent co-occurring items/events

Major machine learning techniques

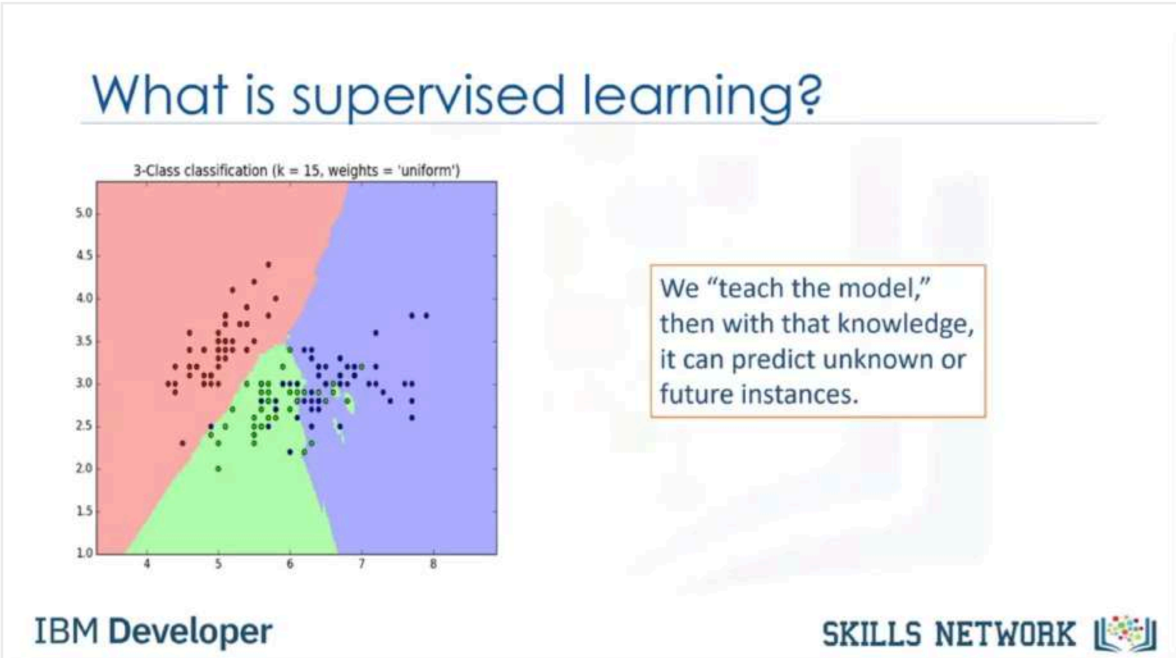
- Anomaly detection
  - Discovering abnormal and unusual cases
- Sequence mining
  - Predicting next events; click-stream (Markov Model, HMM)
- Dimension Reduction
  - Reducing the size of data (PCA)
- Recommendation systems
  - Recommending items



Sobre scikit-learn:



Supervised vs Unsupervised algorithms:



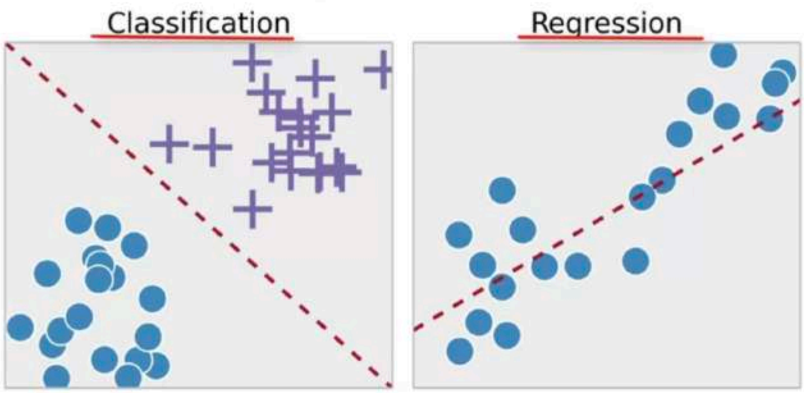
Oka pero como le enseñamos y entrenamos al modelo?
Con un *labeled dataset*

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

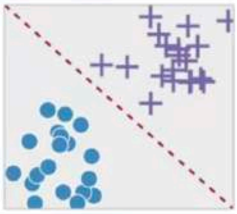


What is classification?

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

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	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	benien

Categorical Values

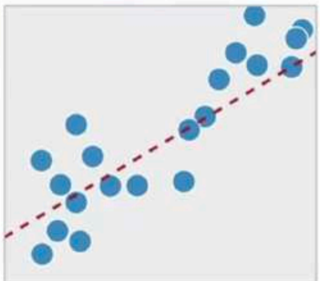


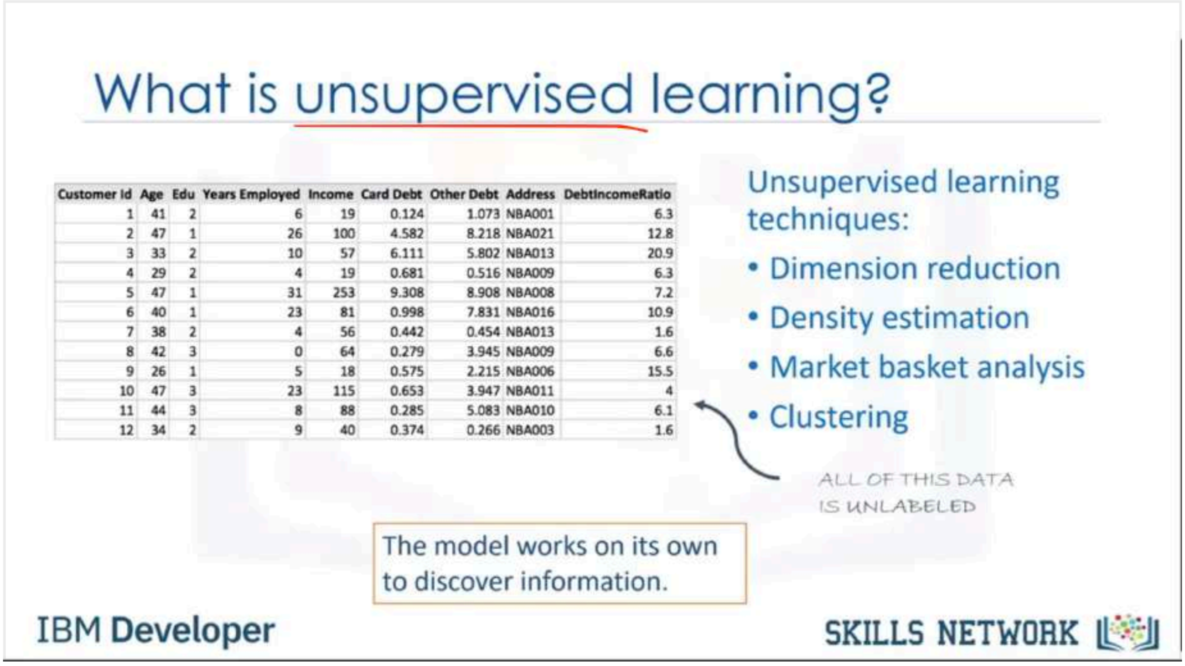
What is regression?

Regression is the process of predicting continuous values.

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_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	?

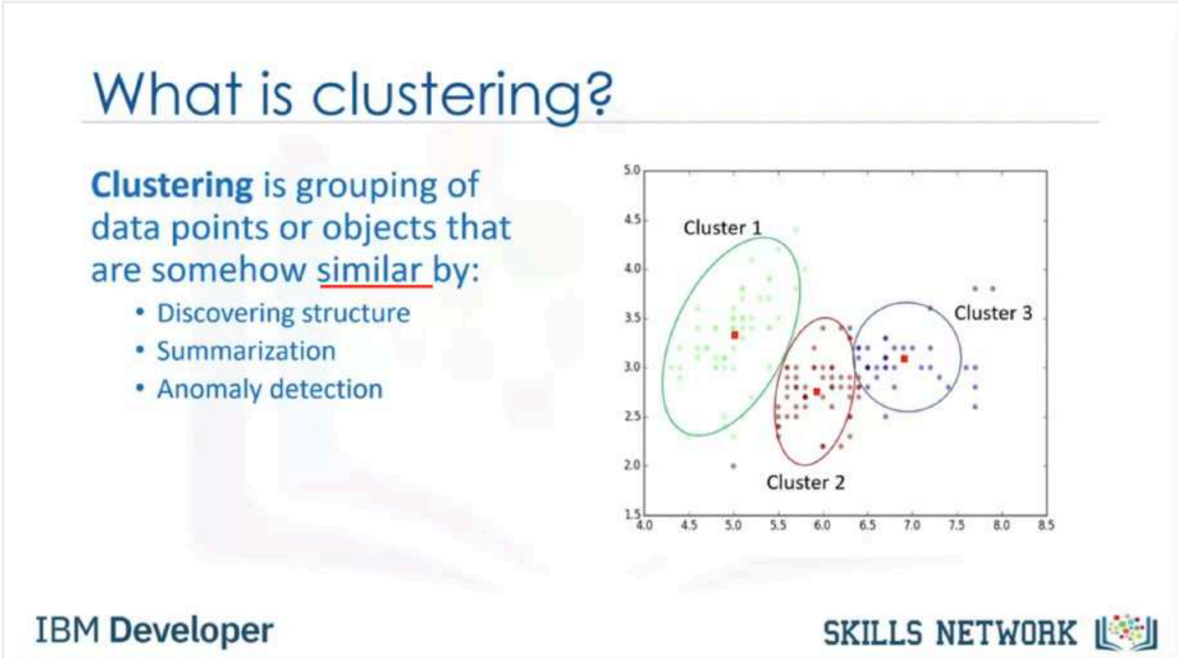
Continuous Values

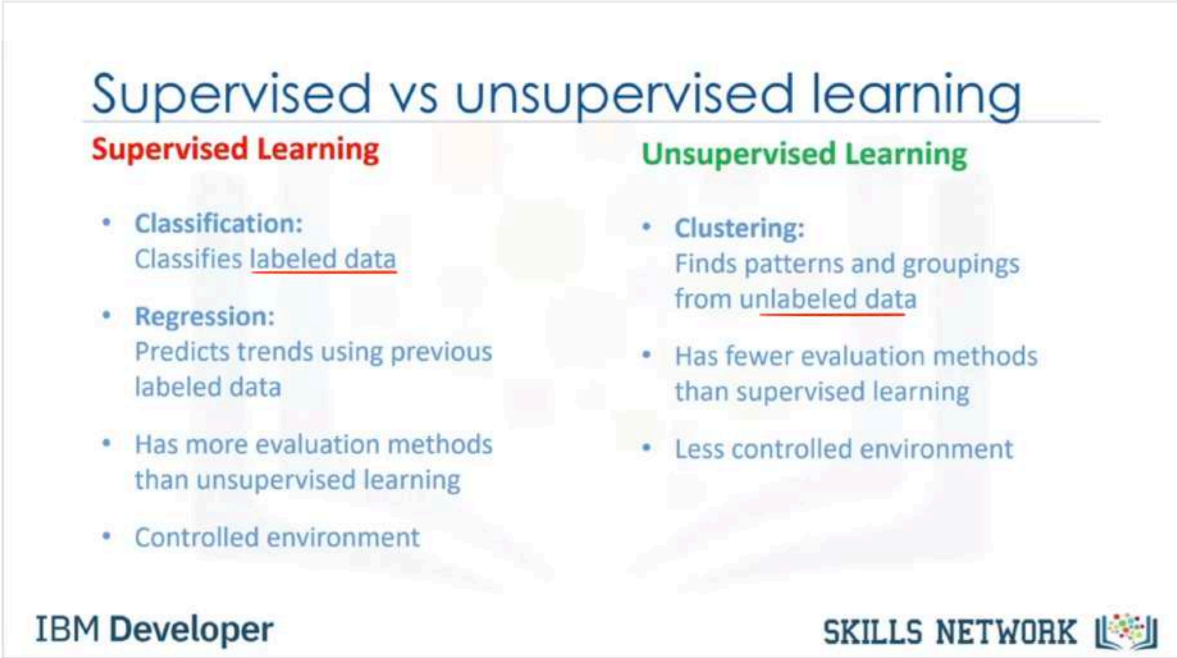




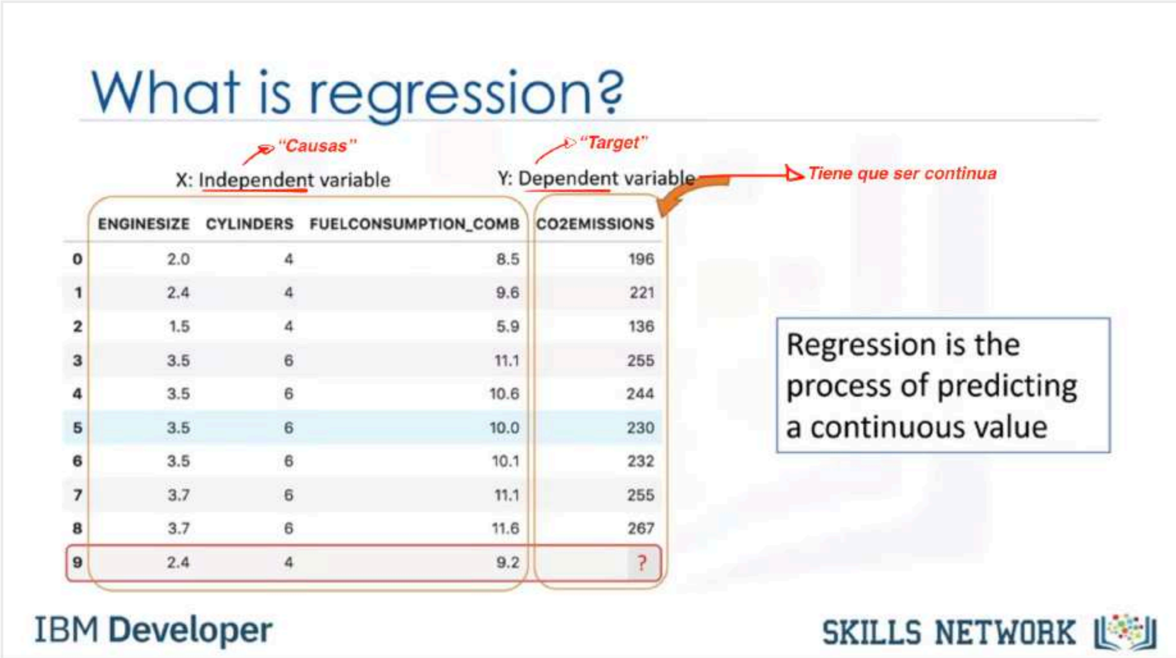
Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2	6	19	0.124	1.073	NBA001	6.3
2	47	1	26	100	4.582	8.218	NBA021	12.8
3	33	2	10	57	6.111	5.802	NBA013	20.9
4	29	2	4	19	0.681	0.516	NBA009	6.3
5	47	1	31	253	9.308	8.908	NBA008	7.2
6	40	1	23	81	0.998	7.831	NBA016	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

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

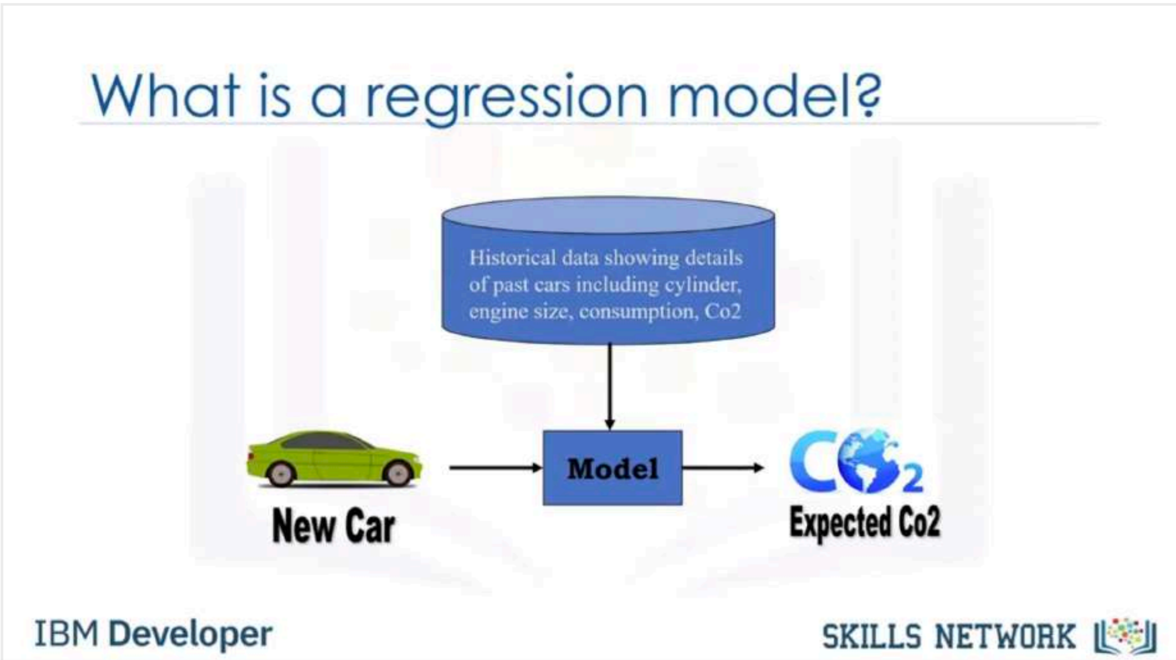




Semana 2 - Linear Regression



	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_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	?



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

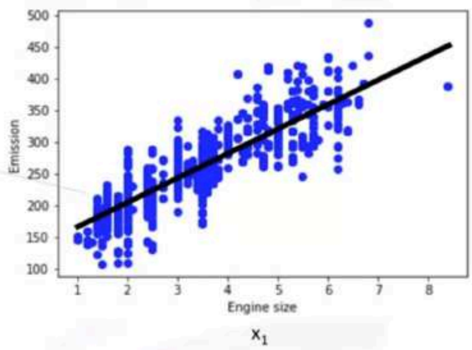
Linear regression model representation

Los coeficientes que queremos ajustar

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

response variable

a single predictor



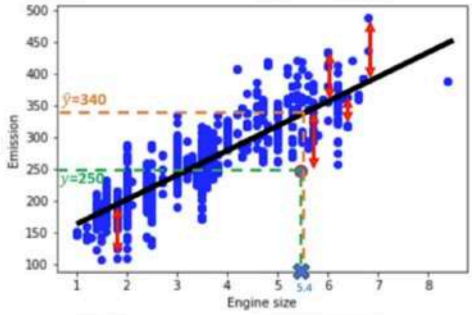
How to find the best fit?

x_1 = 5.4 independent variable
y= 250 actual Co2 emission of x1

$\hat{y} = \theta_0 + \theta_1 x_1$
$\hat{y}$ = 340 the predicted emission of x1

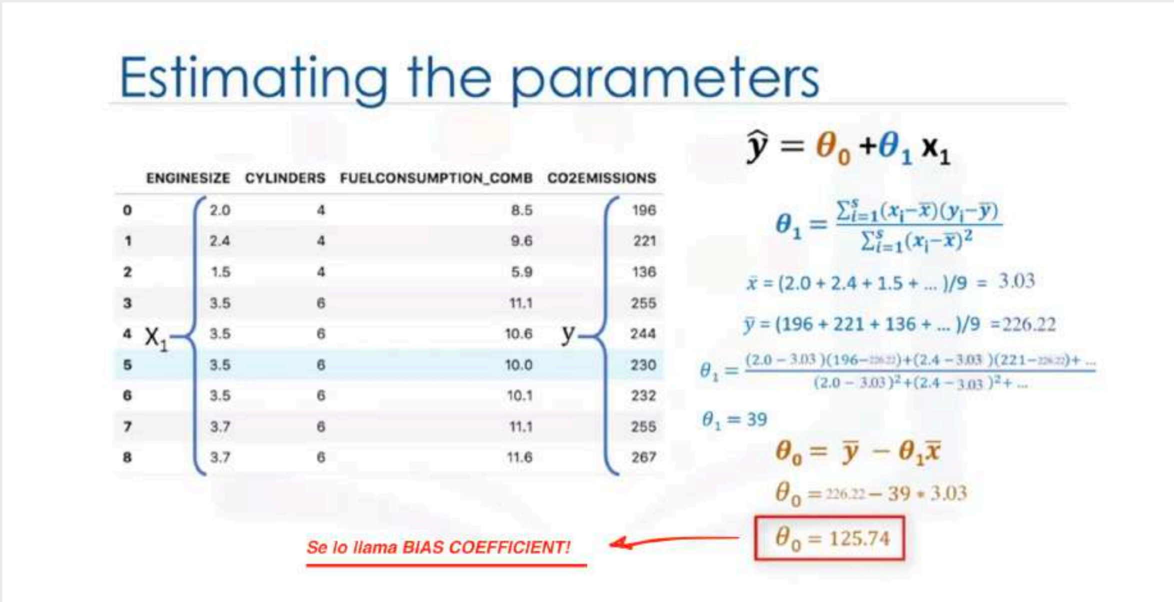
Error = y- $\hat{y}$
= 250 − 340
= -90

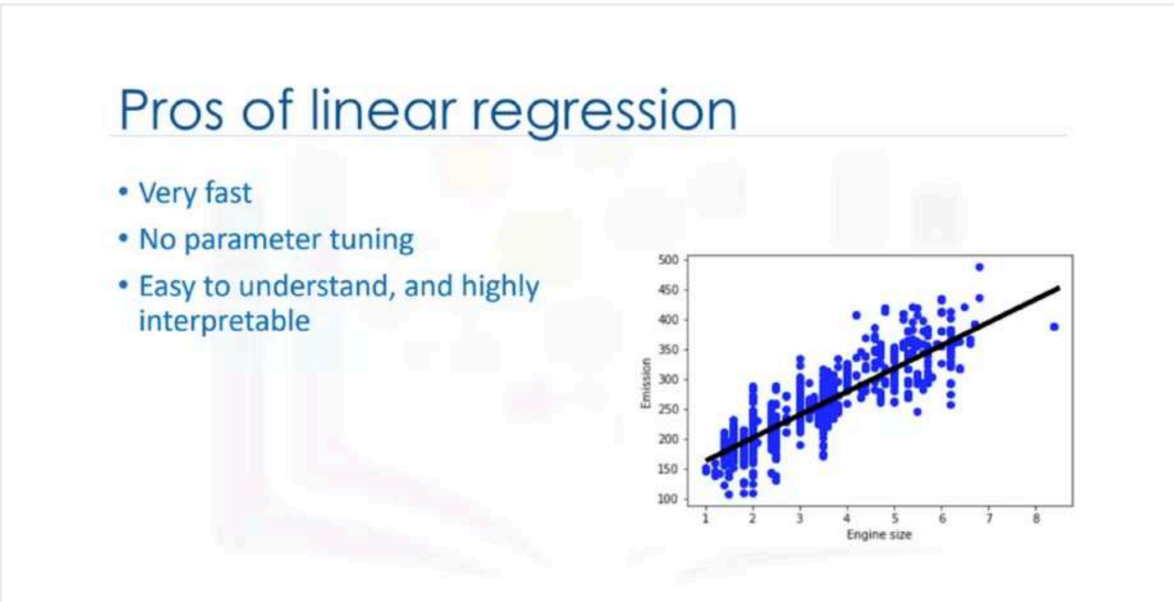
$$\boxed{MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$



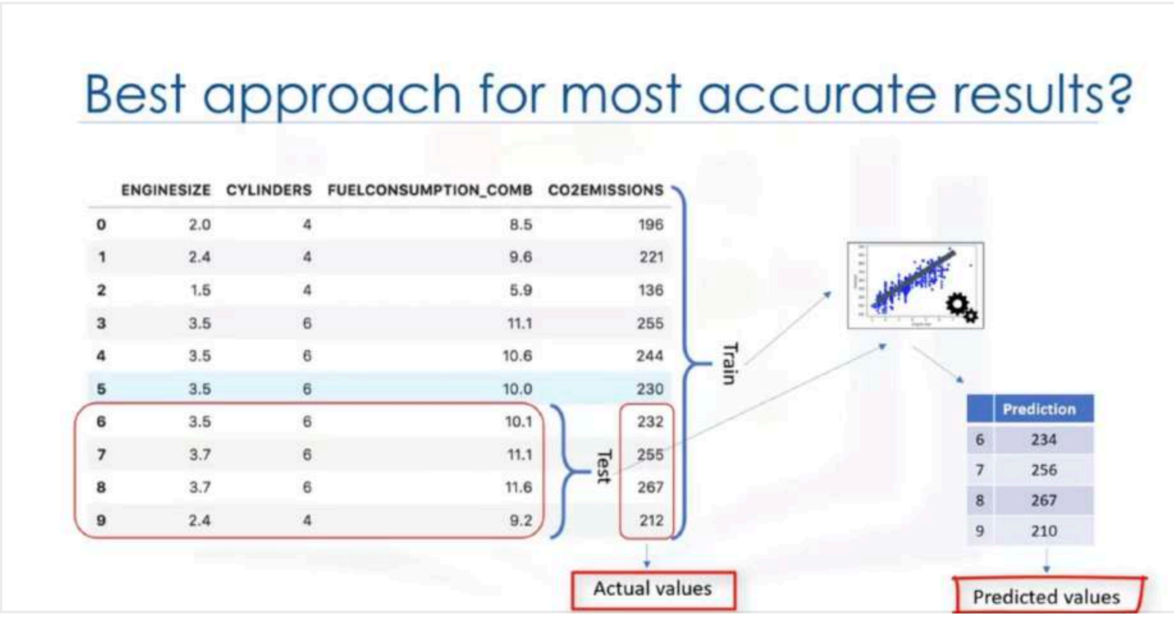
El objetivo de la regresión lineal es minimizar el MSE con Tita0 y Tita1. Como encontramos estos parámetros de forma tal que se minimice el MSE?
Tenemos dos opciones acá.. podemos usar un aproach matemático o uno de optimización.

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

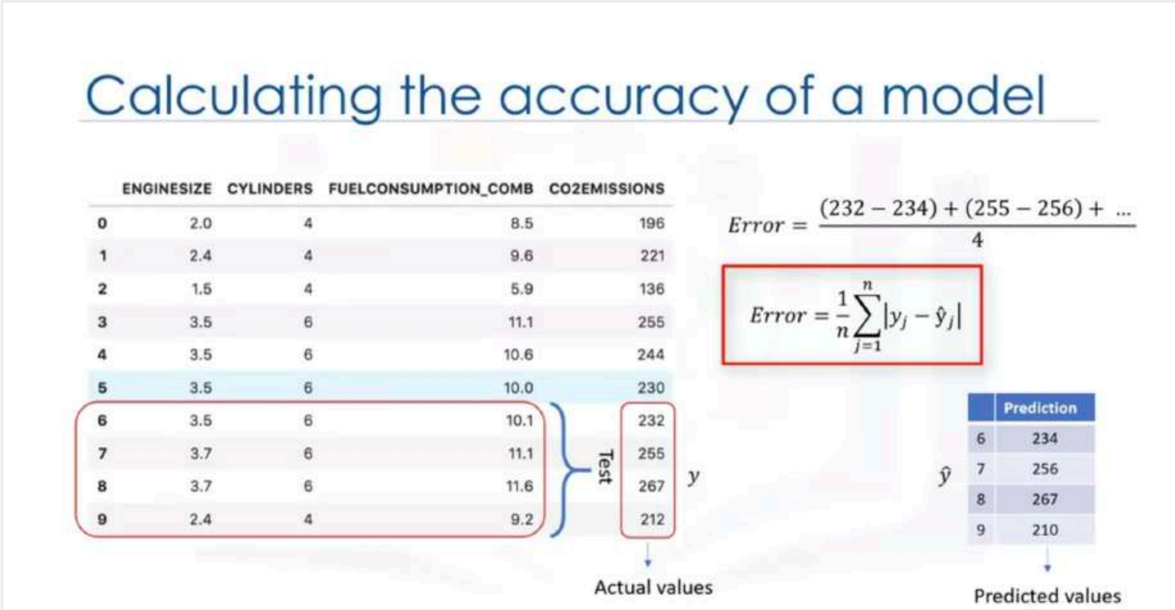


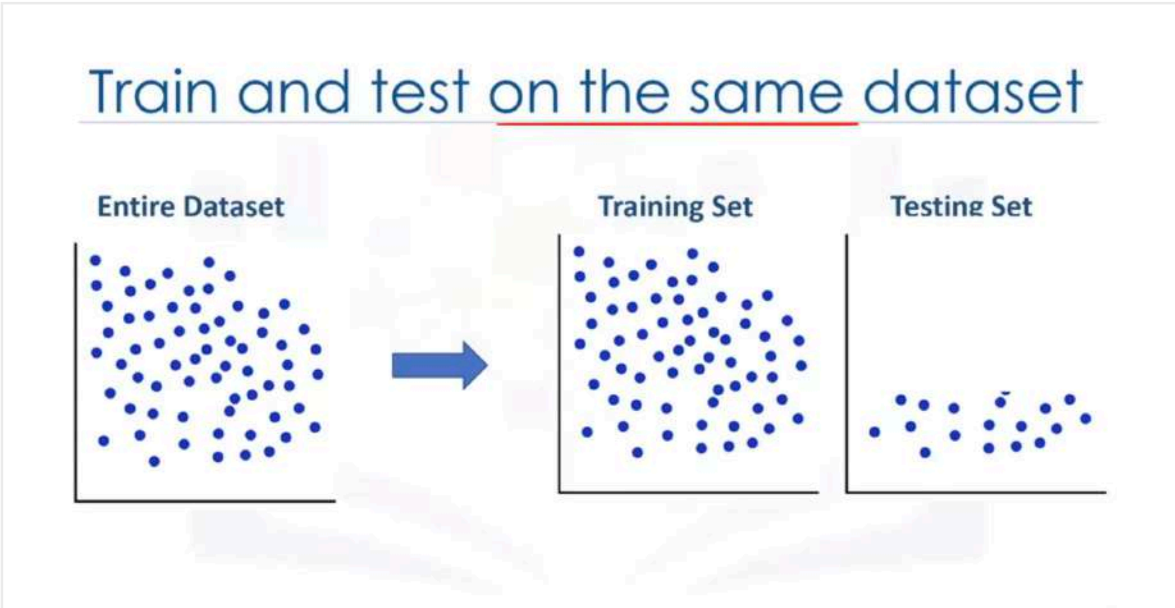


Que tanto podemos confiar en nuestro modelo? Que nivel de precisión tiene? Una de las soluciones para responder a esto es seleccionar una parte del dataset para testing.



Métricas para medir el desempeño de nuestros modelos:

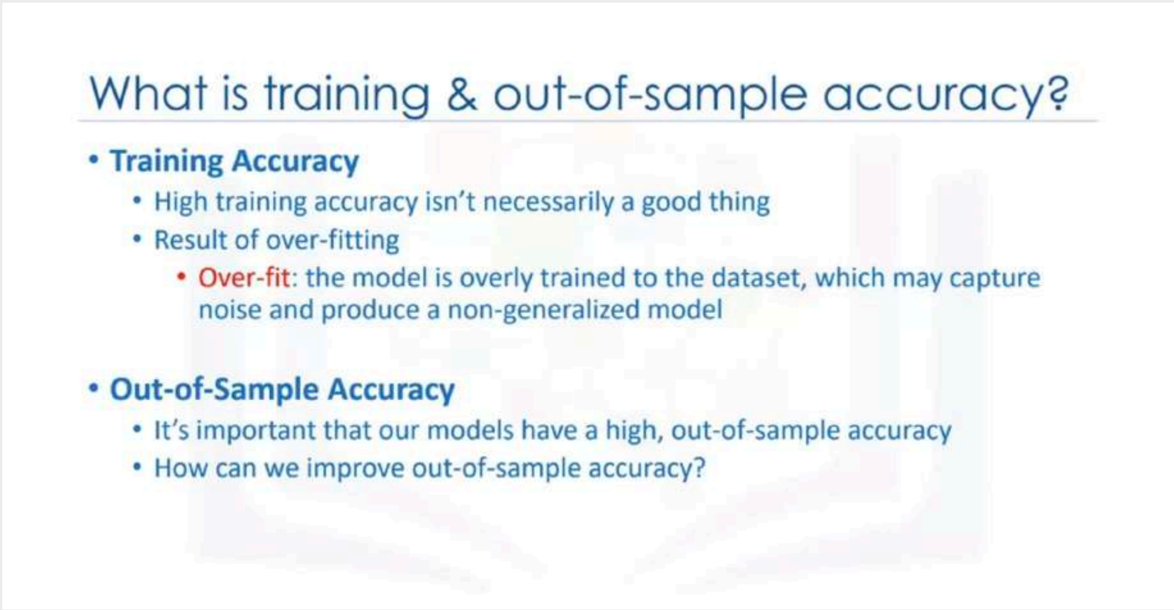




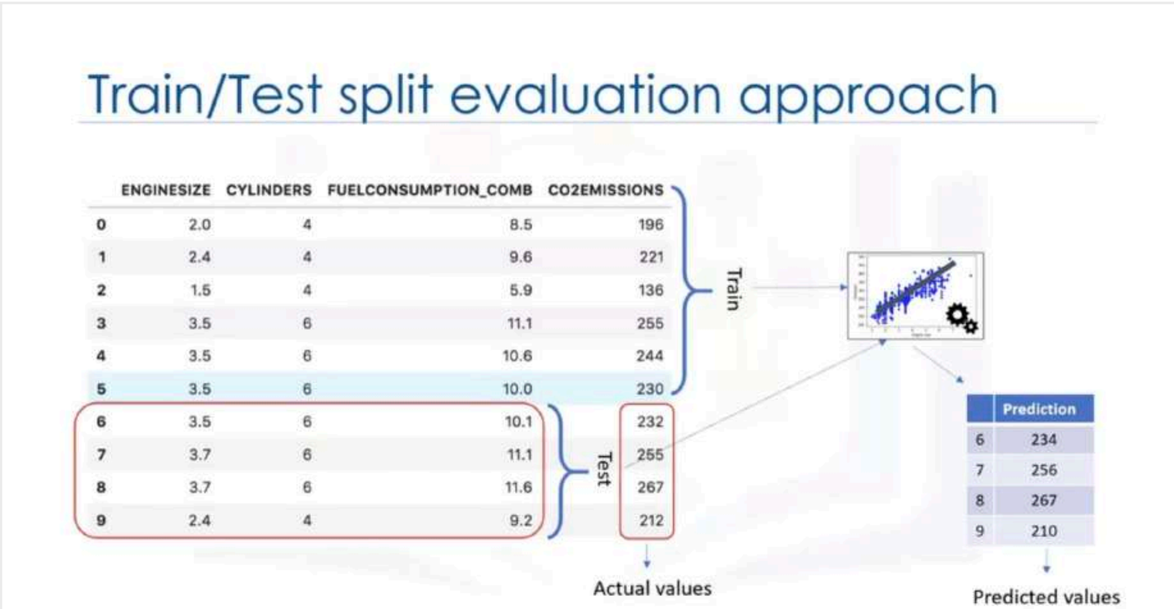
Este primer enfoque implica:

High “training accuracy”
Low “out-of-sample accuracy”

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

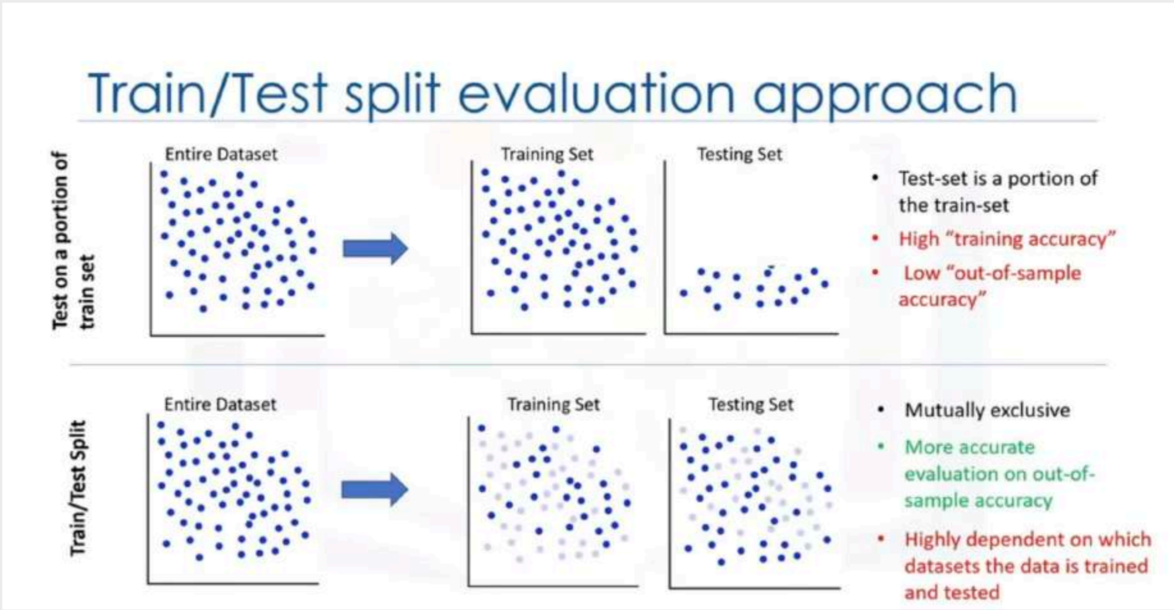


Por el contrario, con el enfoque de Train Test Split:
Acá NO tomo todo para entrenar como en el caso anterior.



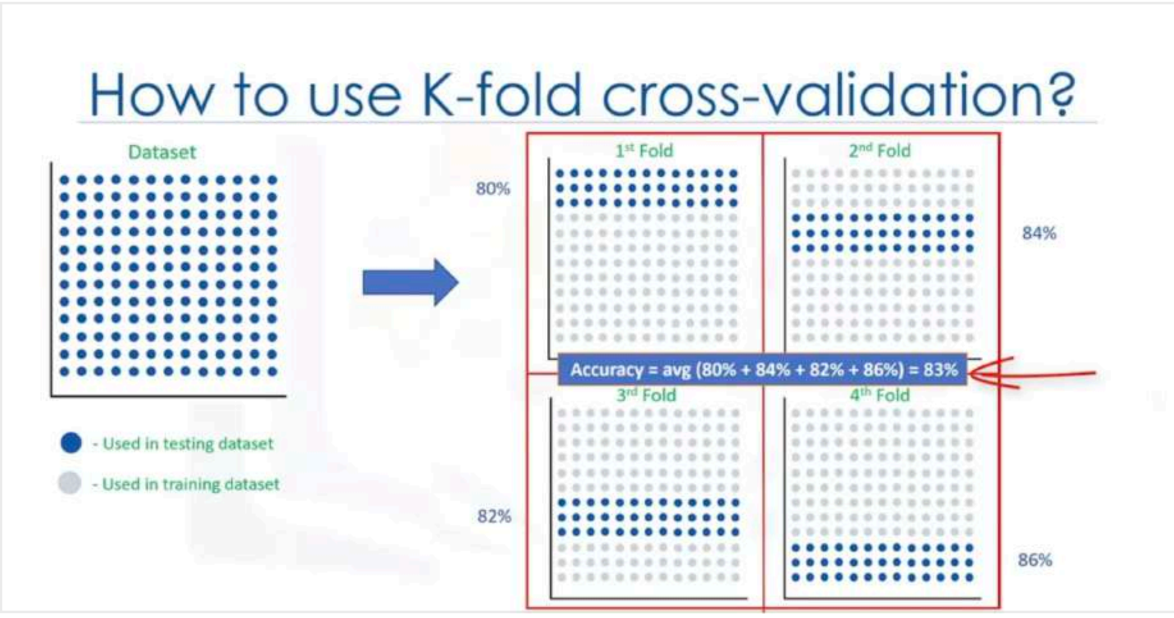
	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	212

	Prediction
6	234
7	256
8	267
9	210

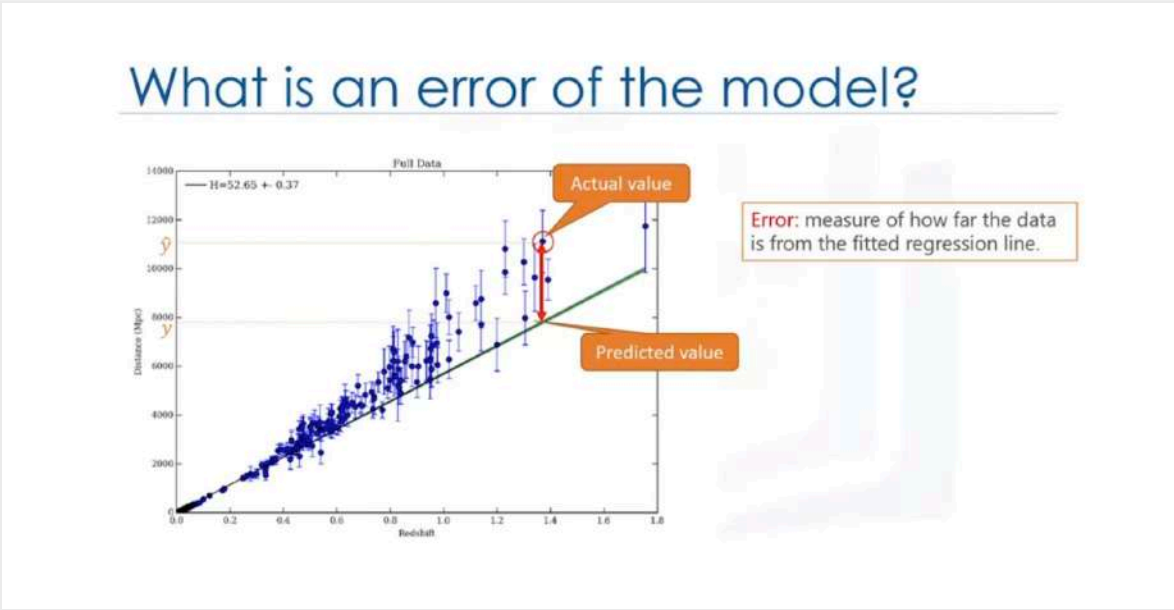


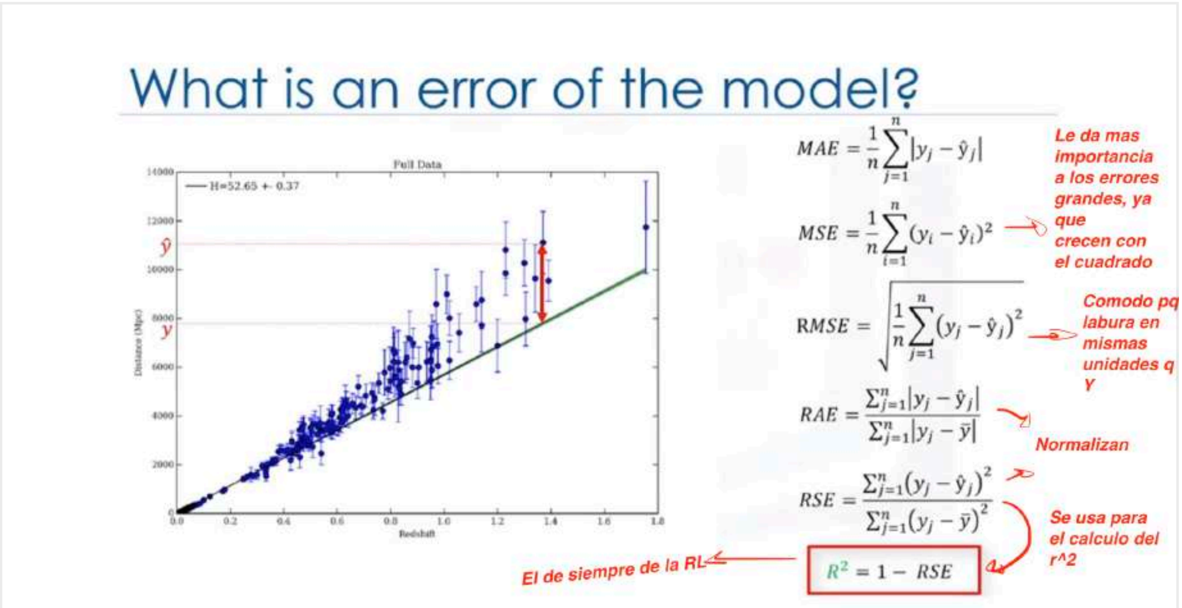
Importante! Dice que una vez que evalúes el modelo con el testing set, también lo uses para entrenar al modelo para no perder esa data. También dice que es altamente dependiente de los datasets en los cuales la data es entrenada y testeada. Si bien performa mejor que el enfoque anterior, sigue teniendo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





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 ●

Code | Run as Pipeline | 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

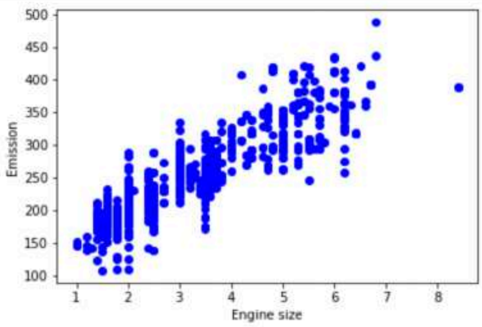
Linear Regression fits a linear model with coefficients B = (B1, ..., Bn) to minimize the 'residual sum of squares' between the actual value y in the dataset, and the predicted value yhat using linear approximation.

Train data distribution

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


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[['ENGINESIZE']])
train_y = np.asanyarray(train[['CO2EMISSIONS']])
regr.fit(train_x, train_y)
# The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)
```

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

Gotas

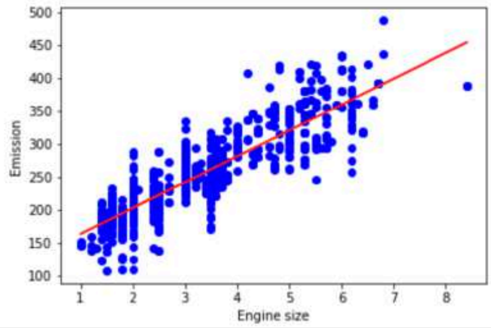
As mentioned before, **Coefficient** and **Intercept** in the simple linear regression, are the parameters of the fit line. Given that it is a simple linear regression, with only 2 parameters, and knowing that the parameters are the intercept and slope of the line, sklearn can estimate them directly 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, lets 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.asanyarray(test[['ENGINESIZE']])
test_y = np.asanyarray(test[['CO2EMISSIONS']])
test_y_ = regr.predict(test_x) # NOTAR COMO ACA EN VEZ DE FIT PARA ENOCNTRAR 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

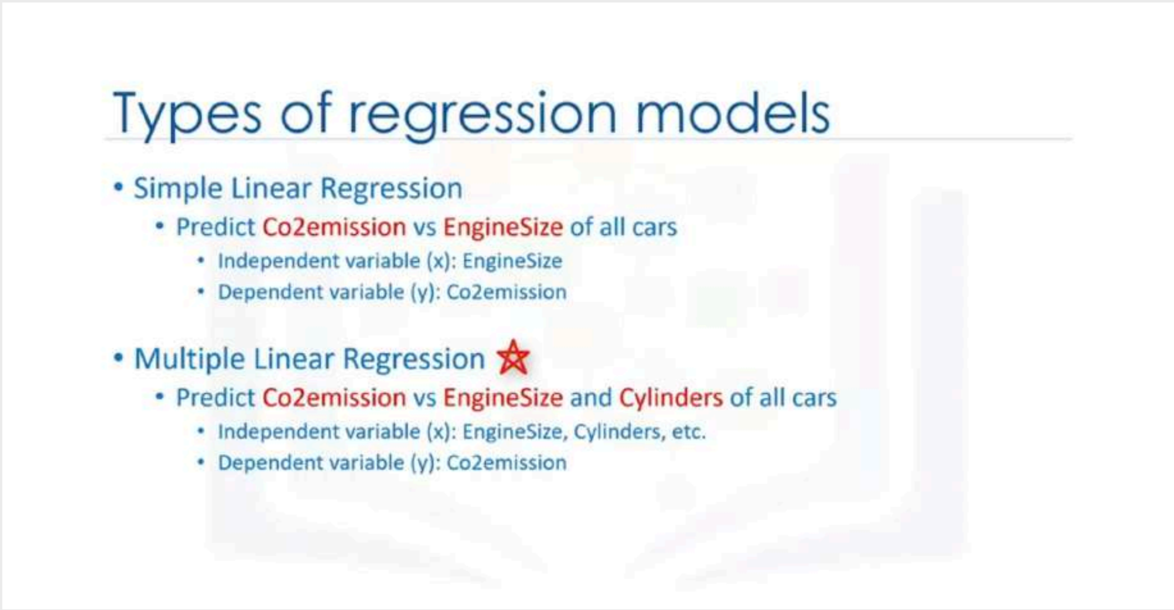
```
[45]: test_y = train[["CO2EMISSIONS"]]
      print("Mean Absolute Error: %.2f" % np.mean(np.absolute(predictions - test_y)))

      Mean Absolute Error: 20.80
```

▶ Click here for the solution

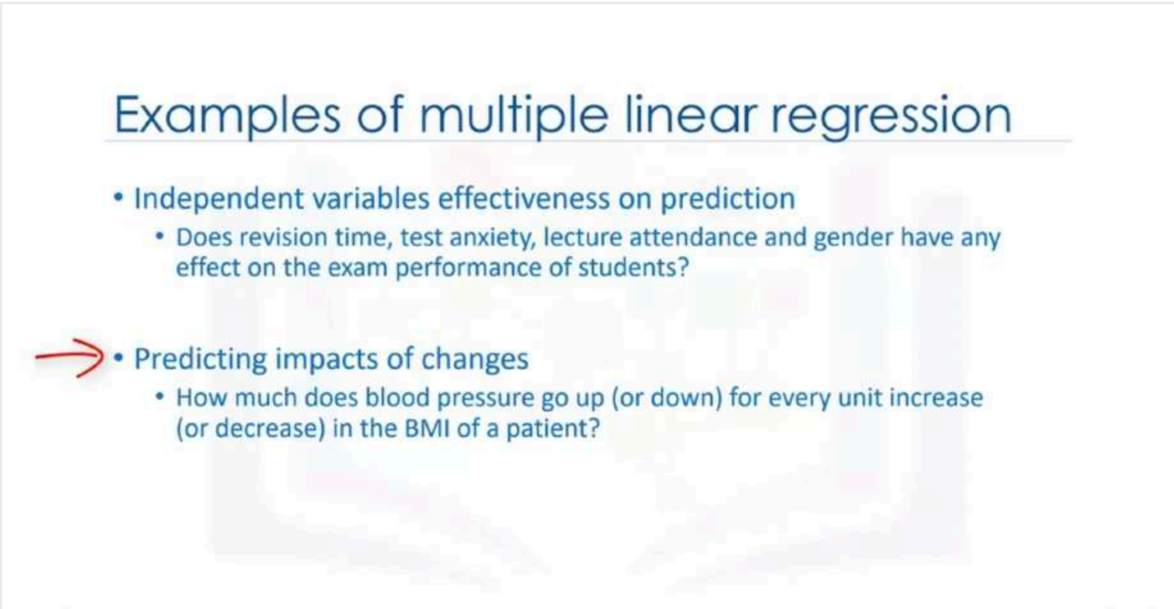
We can see that the MAE is much worse than it is when we train using `ENGINESIZE`

📈 📉 Multiple Linear Regression

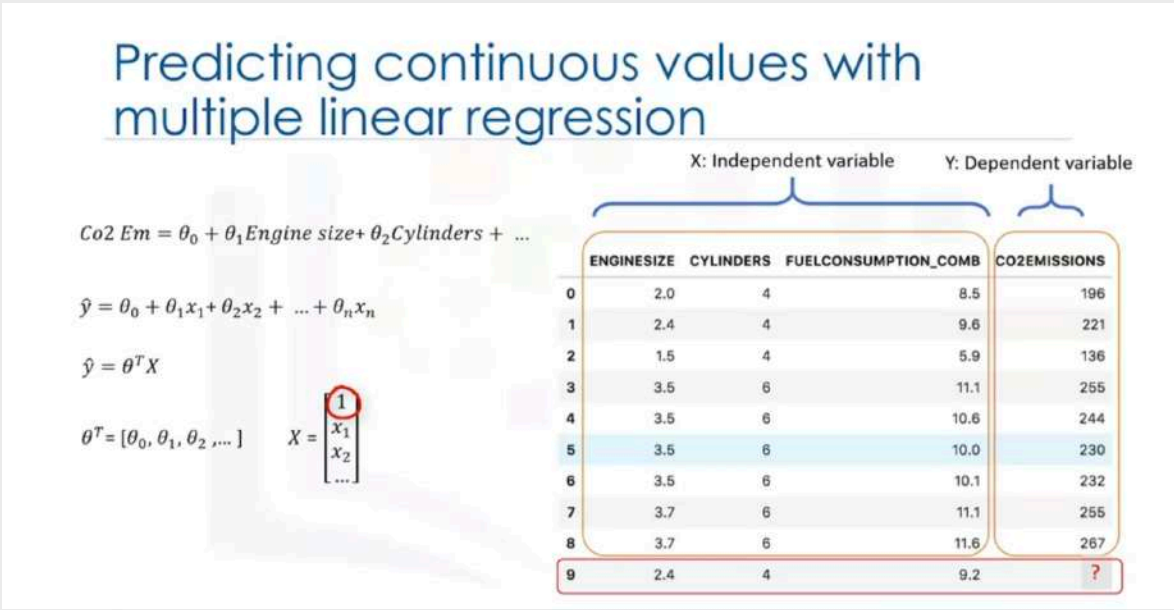
Multiple Linear Regression

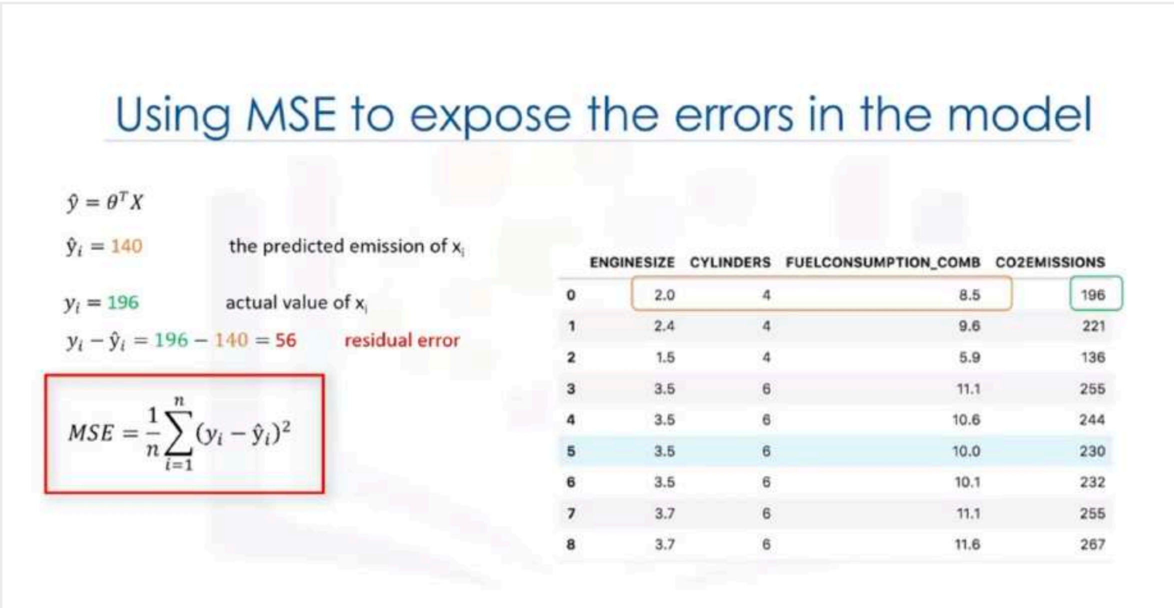


Aplicaciones:

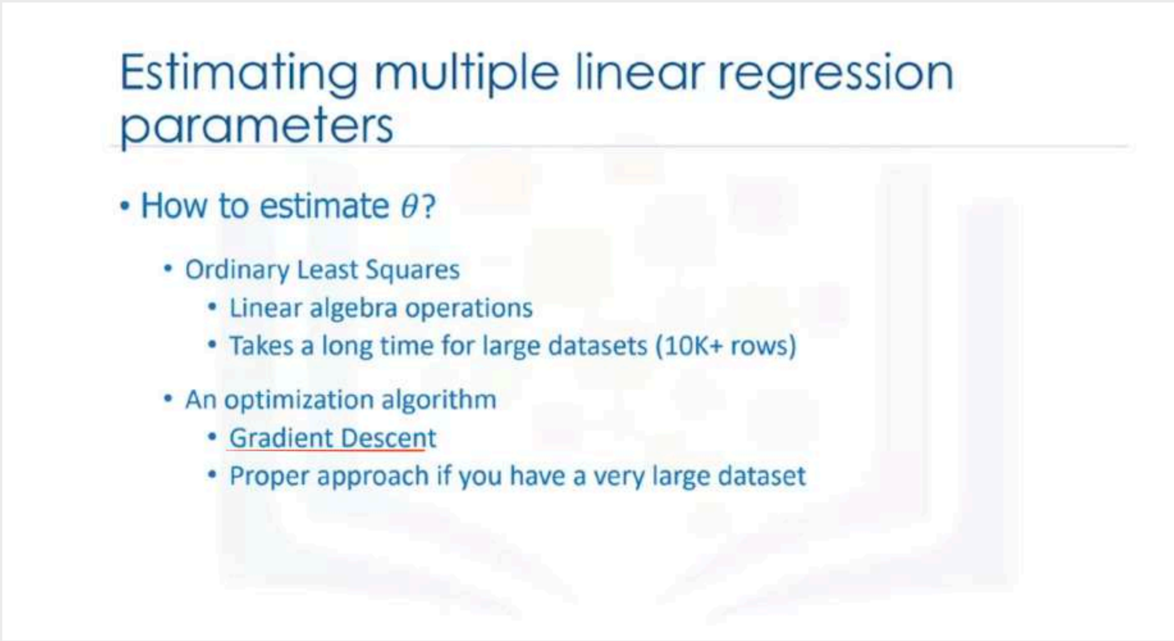


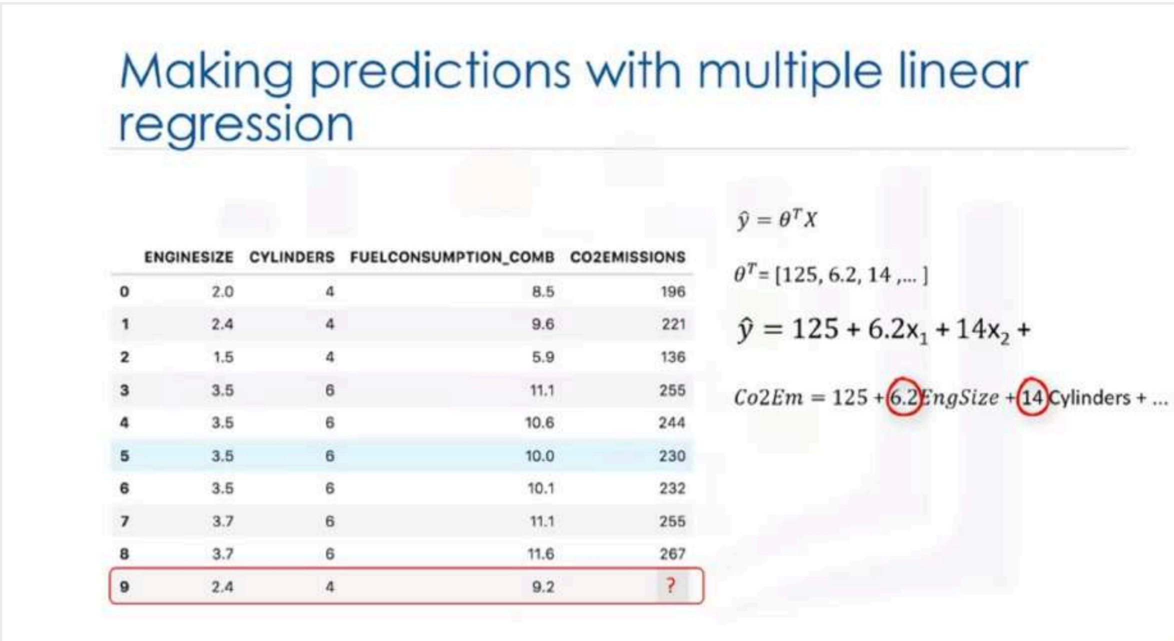
Ver el impacto de las variables sobre la Y, cuando mantenemos las otras constantes.





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



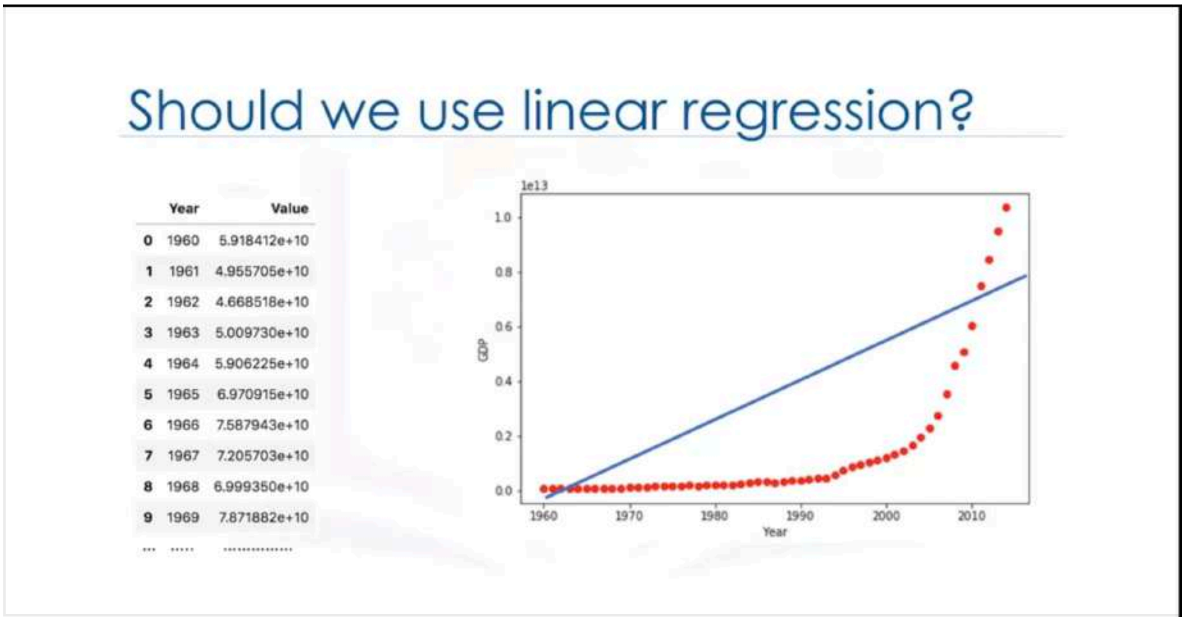


	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

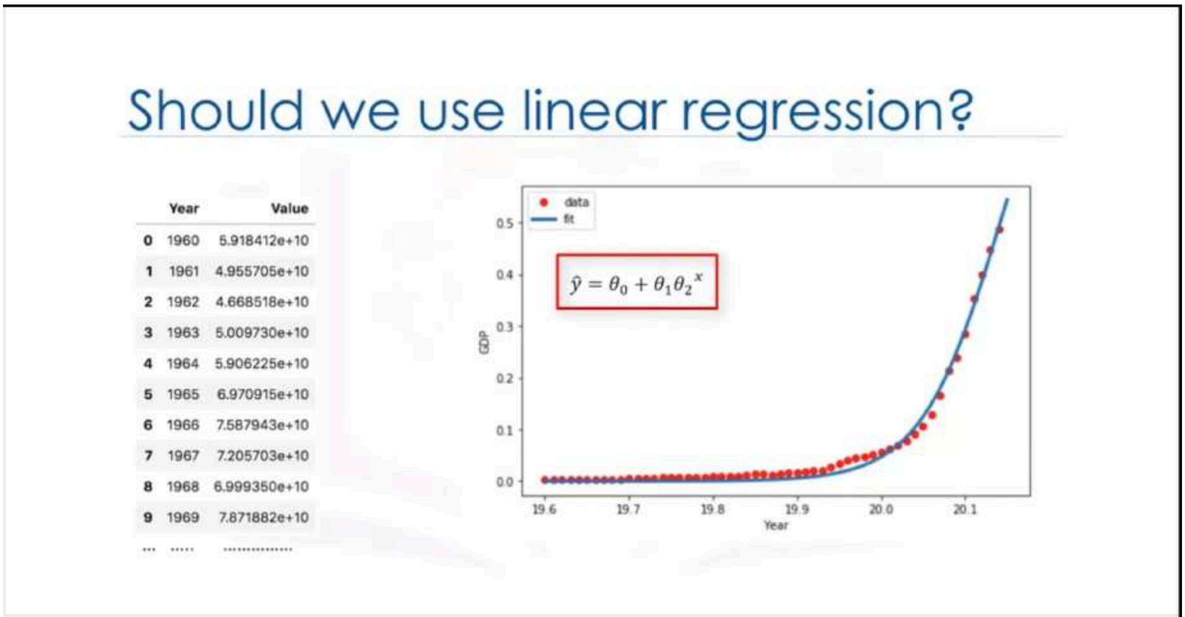
Ojota, agregar miles de variables a una múltiple linear reg sin ningún tipo de back teórico por lo general nos perjudica y nos overfittea.

🧮 Non Linear Regression

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

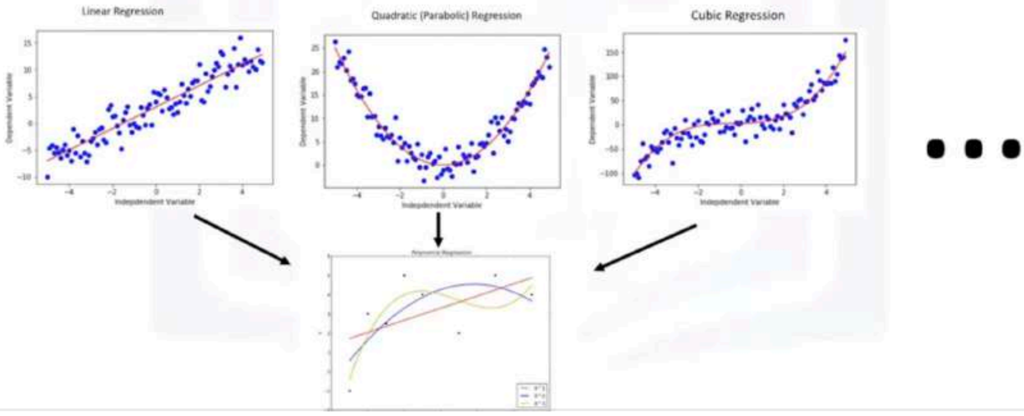


Tiene mas pinta de una función exponencial o una logística, entonces necesitamos otro método:



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.

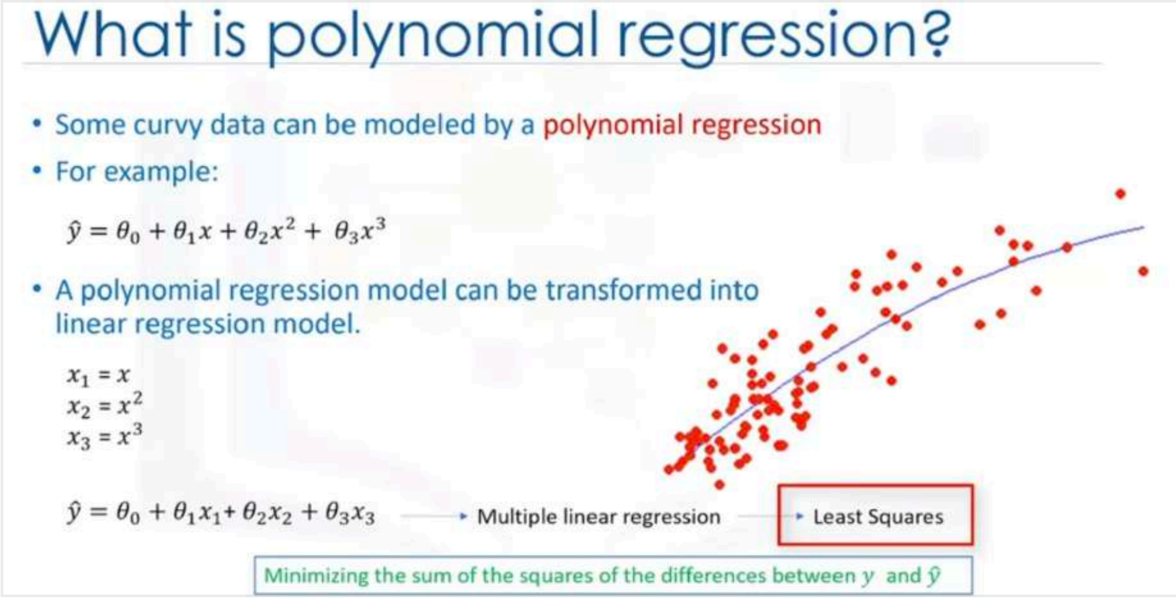
$x_1 = x$
$x_2 = x^2$
$x_3 = x^3$

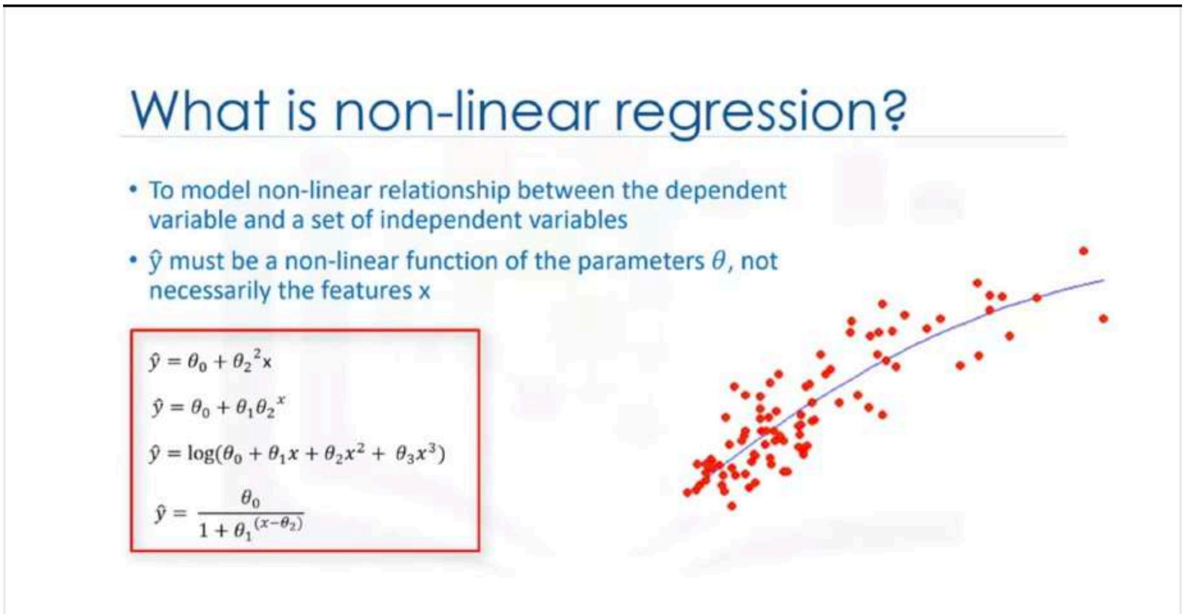
"Convertis" un modelo polinomial a lineal con un cambio de variable

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

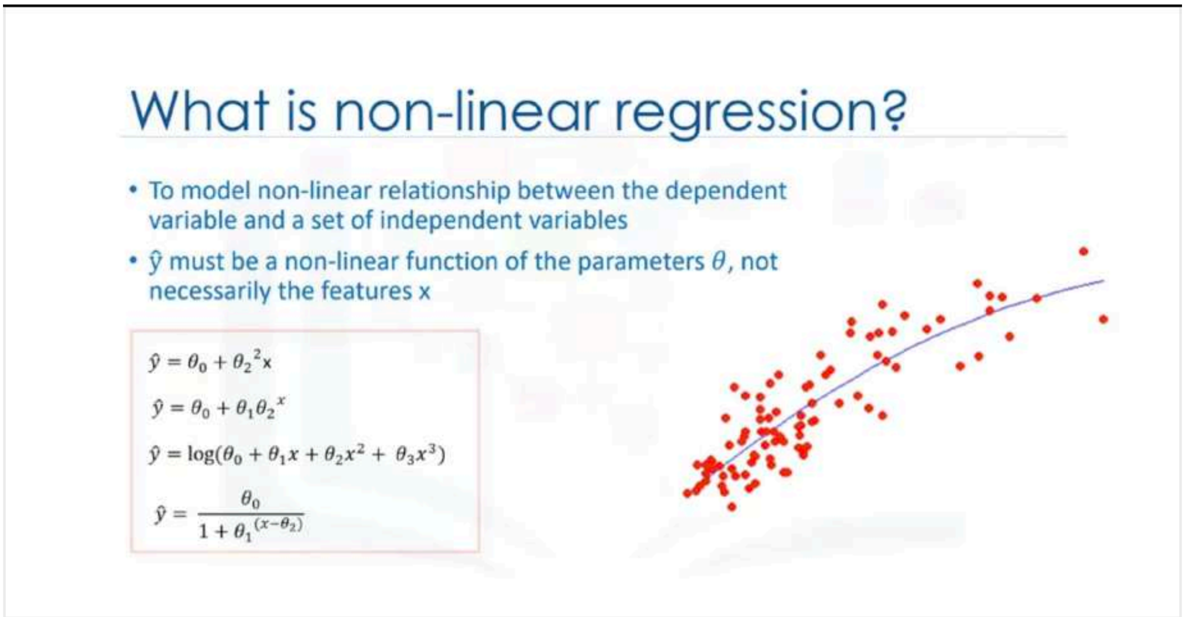
Podes usar el mismo mecanismo de multiple linear reg con polinmoiales que entren en eso

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



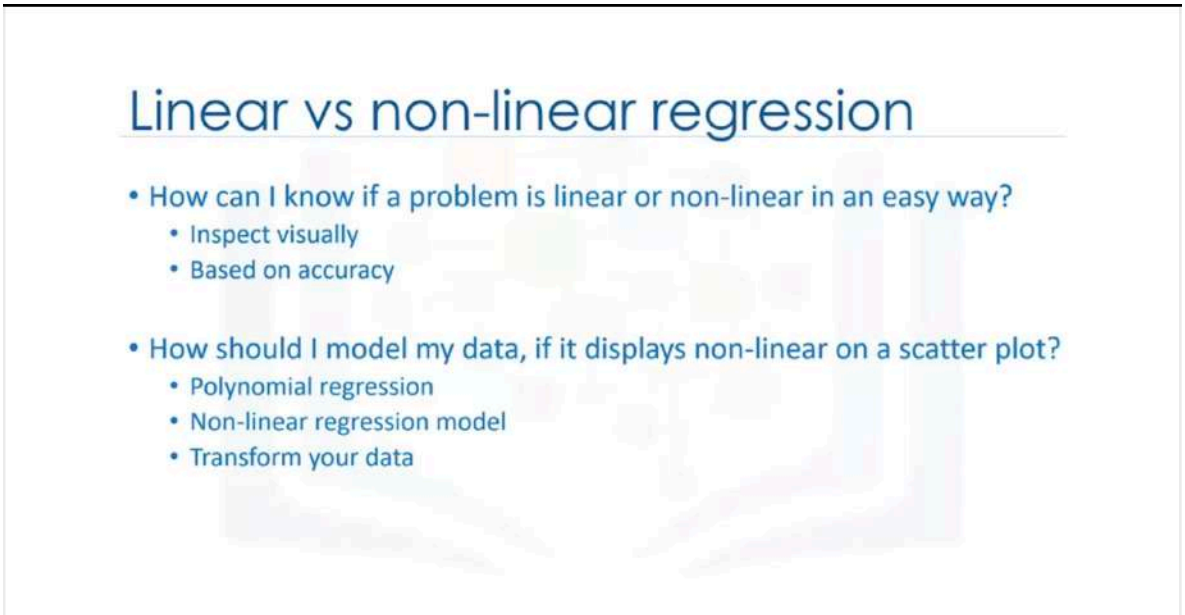


En contraste, en las non linear, no podemos usar el modelo de regresión lineal para estimar los parámetros. (Cuadrados mínimos) y tenemos que usar en cambio otros métodos. Ademas, por lo general estimar los parámetros en estos casos NO es fácil.



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.



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

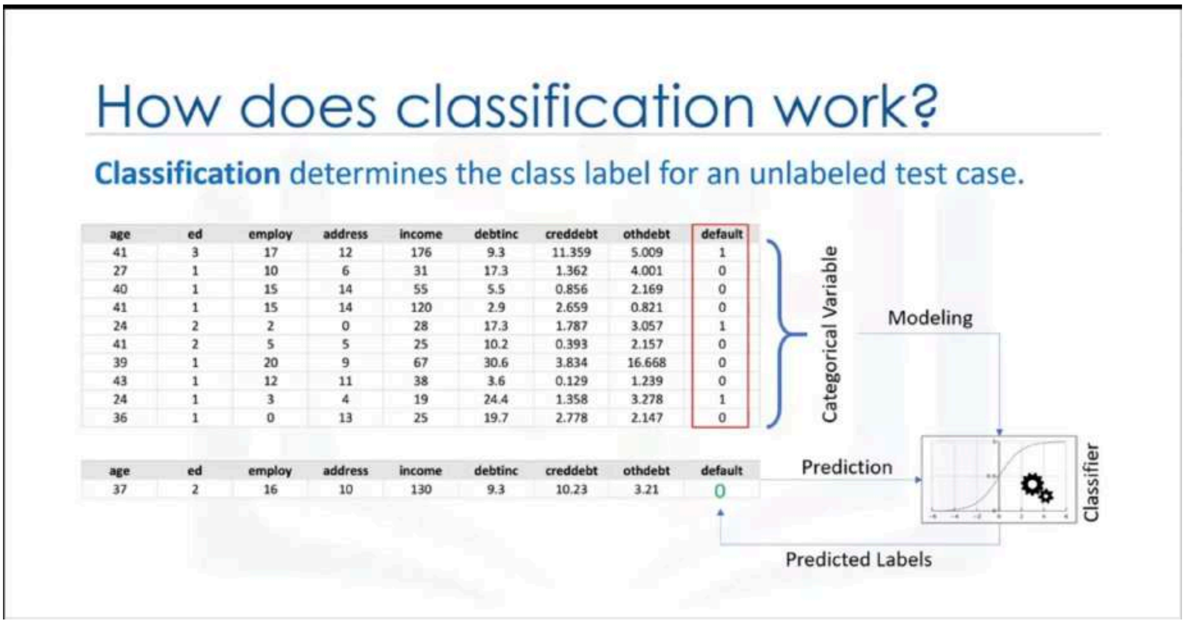
How does classification work?

Classification determines the class label for an unlabeled test case.

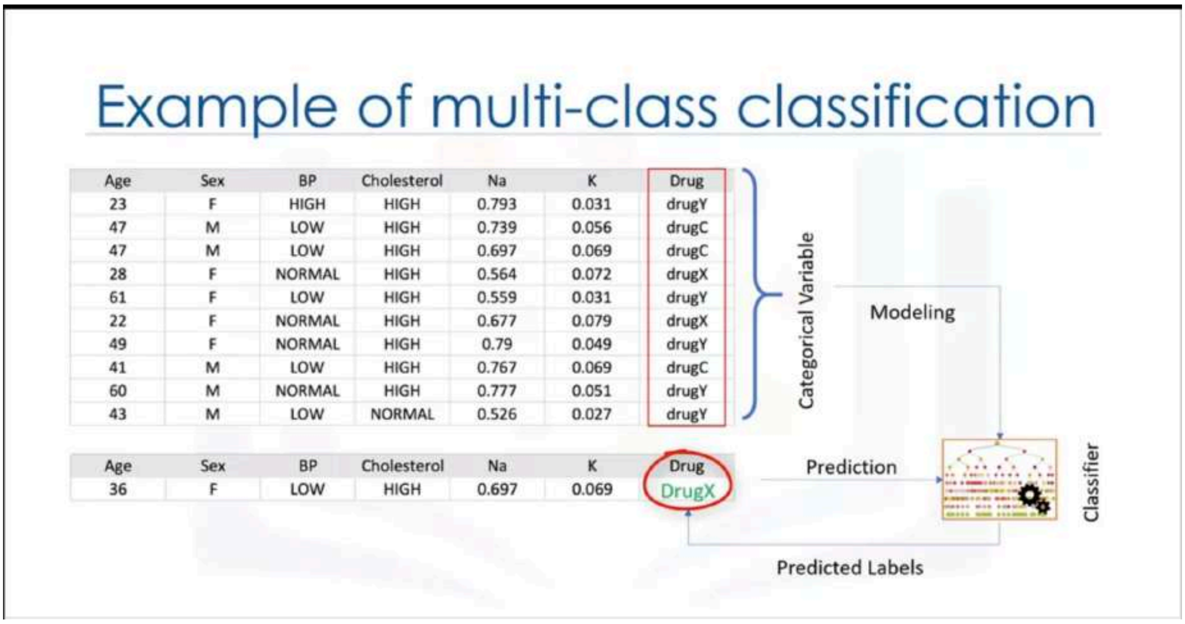
age	ed	employ	address	income	debtinc	creddebt	othdebt	default
41	3	17	12	176	9.3	11.359	5.009	1
27	1	10	6	31	17.3	1.362	4.001	0
40	1	15	14	55	5.5	0.856	2.169	0
41	1	15	14	120	2.9	2.659	0.821	0
24	2	2	0	28	17.3	1.787	3.057	1
41	2	5	5	25	10.2	0.393	2.157	0
39	1	20	9	67	30.6	3.834	16.668	0
43	1	12	11	38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



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 clasification. Por ejemplo:

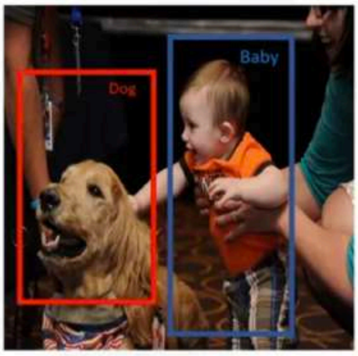


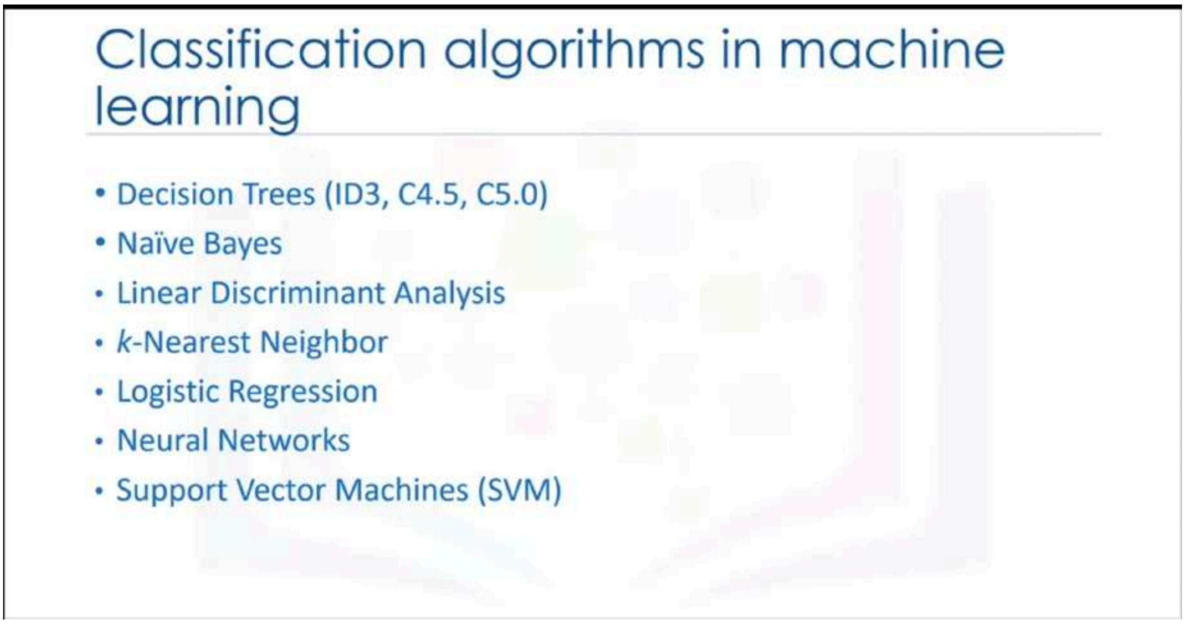
Classification use cases

	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

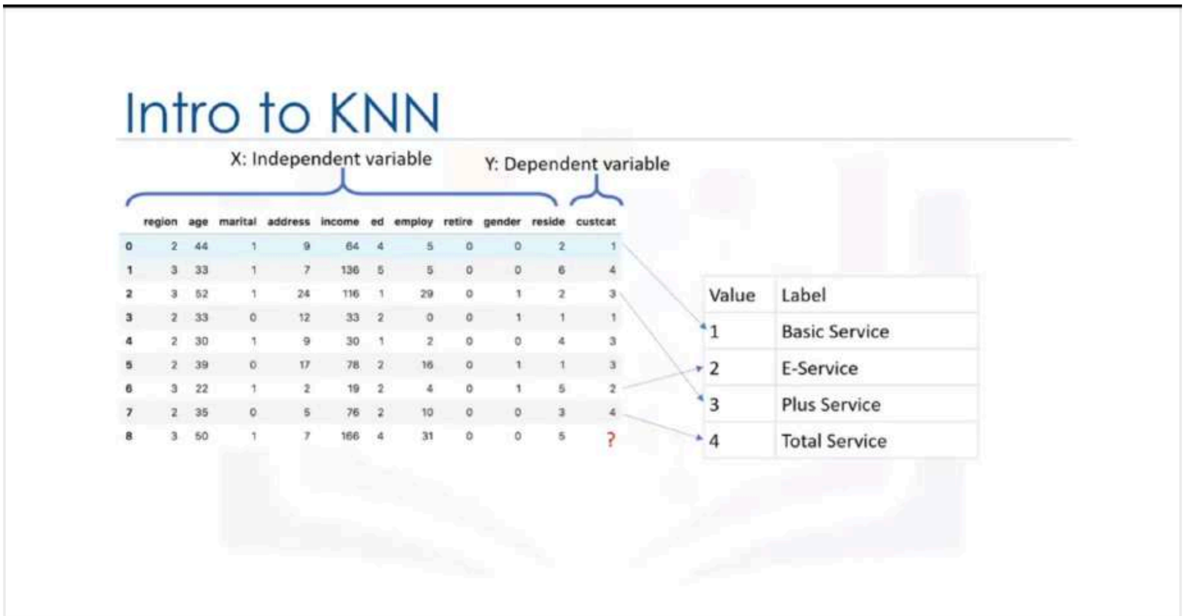


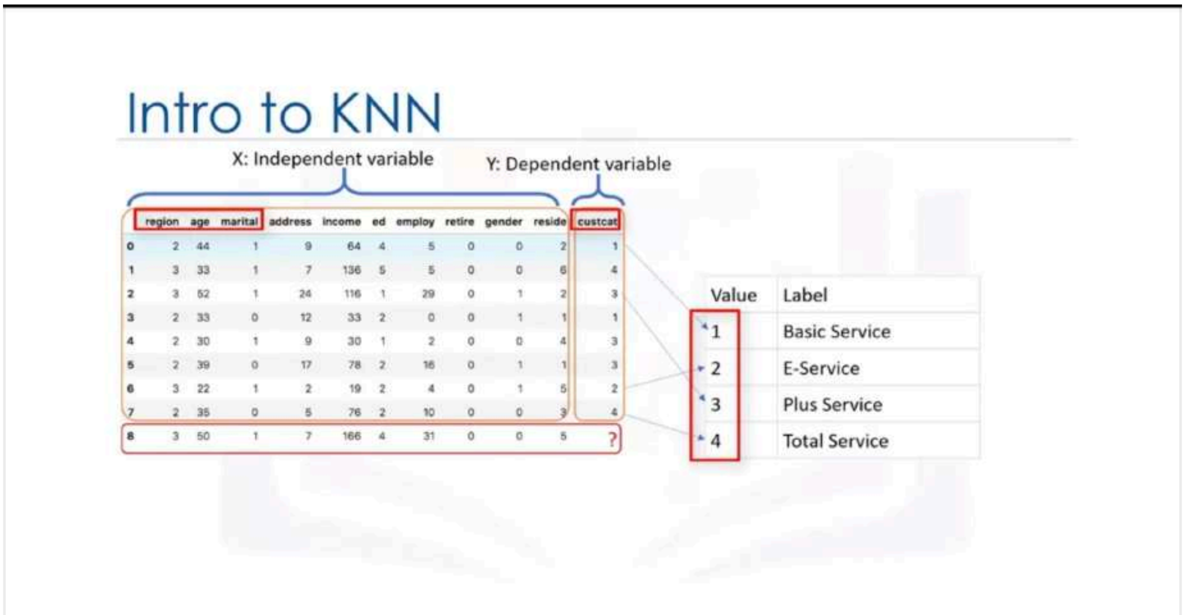


Vamos a ver un par de estos algoritmos…

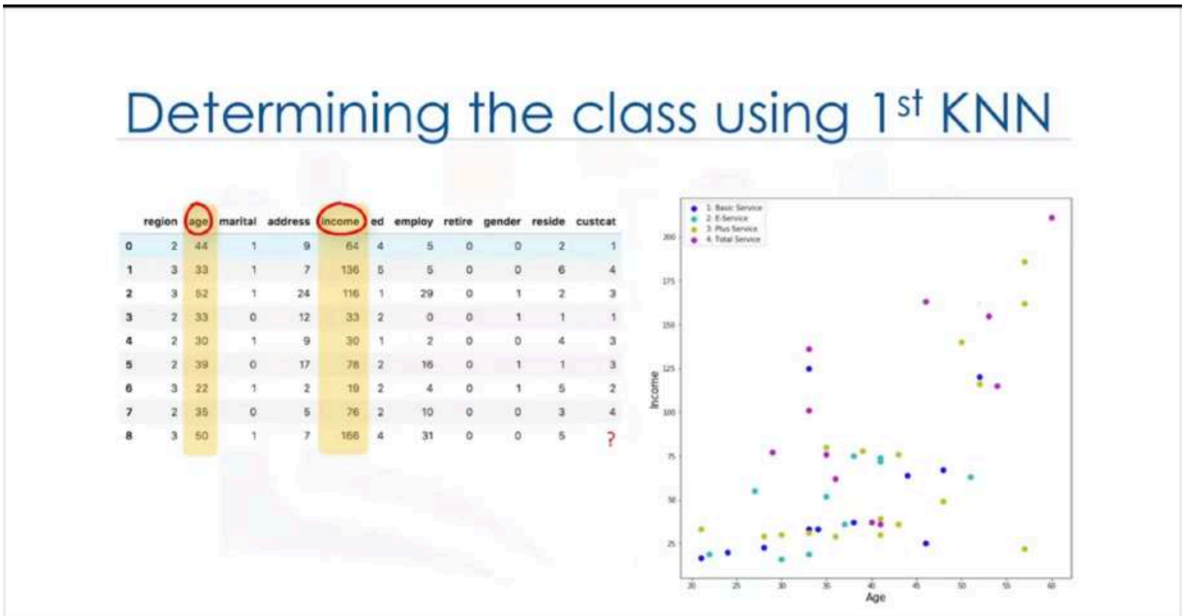
KNN

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

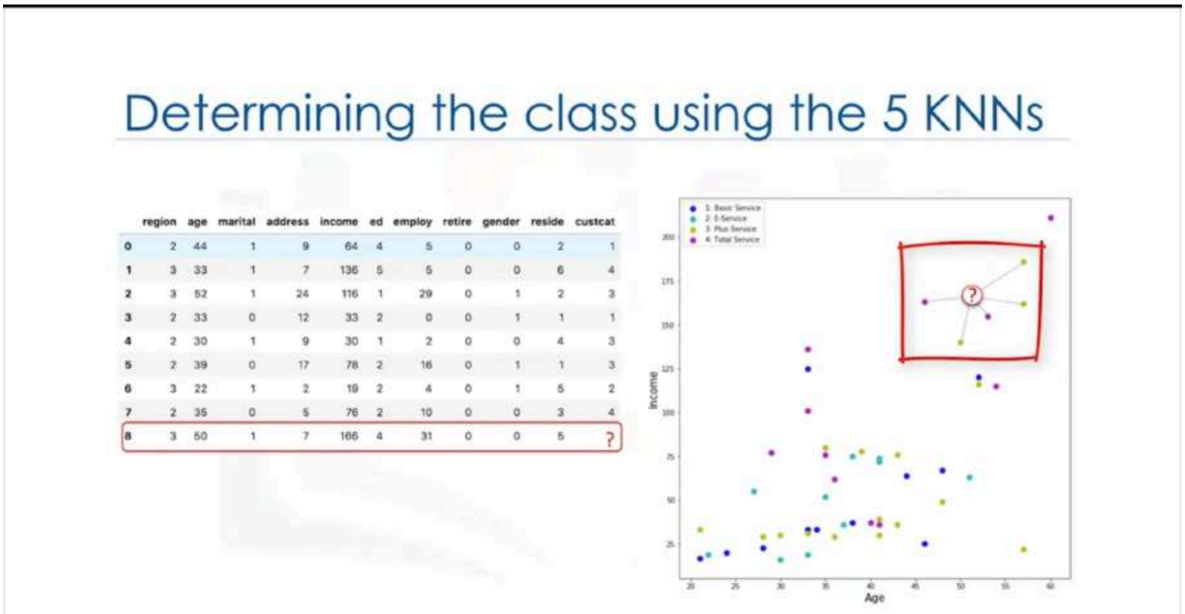




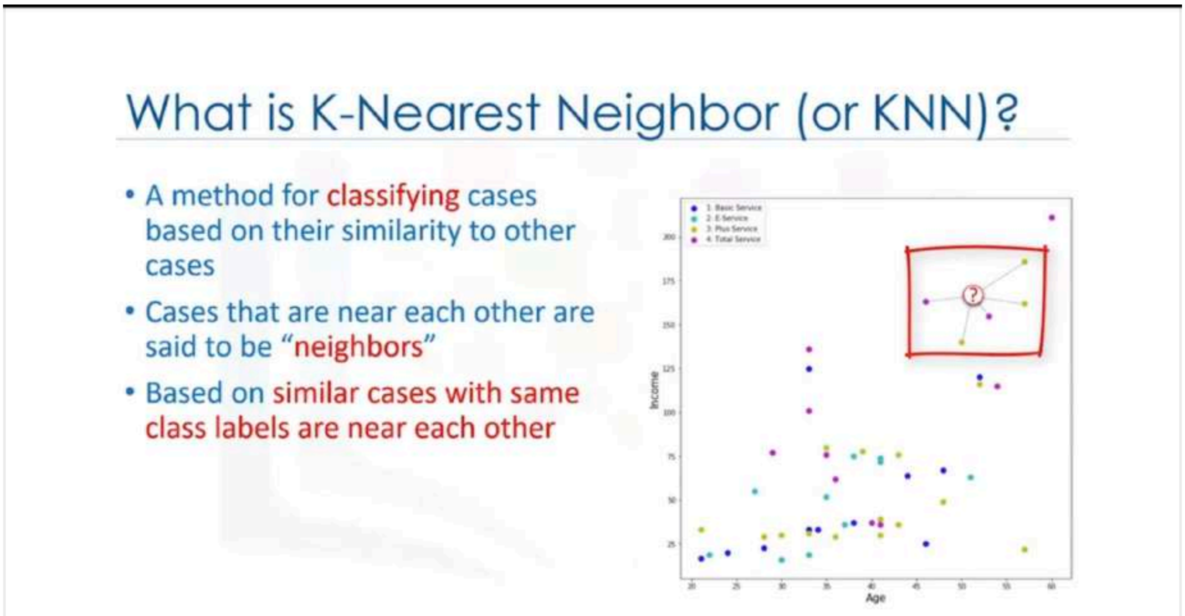
Queremos usar las filas 0-7 para predecir la clasificación de la fila 8. Vamos a usar KNN. Como demostración, primero usemos 2 campos:



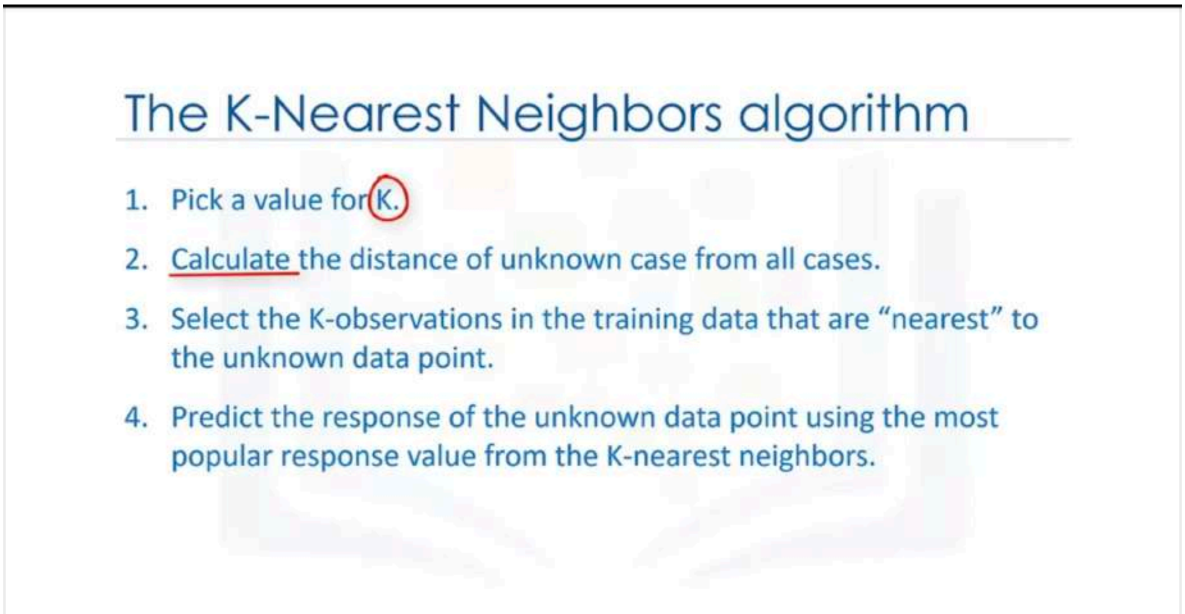
Aca dice que puede decir que va a ser un clase 4 porque nearest neighbor es un clase 4 tambien. La joda es que tanto podemos confiar en esta decisión. Por ejemplo, nuestro nn podria ser un outlier o un caso muy especifico. Pero… si en vez de elegir un solo NN, que pasa si elegimos por ejemplo 5?



Y ahi hacemos un mayority 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 algorythm. Bueno, definamos el algoritmo:



Y como funciona?



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

Calculating the similarity/distance in a 1-dimensional space

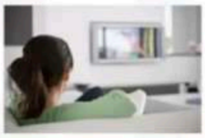
Customer 1
Age
34

Customer 2
Age
30

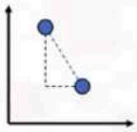
$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

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

Calculating the similarity/distance in a 2-dimensional space



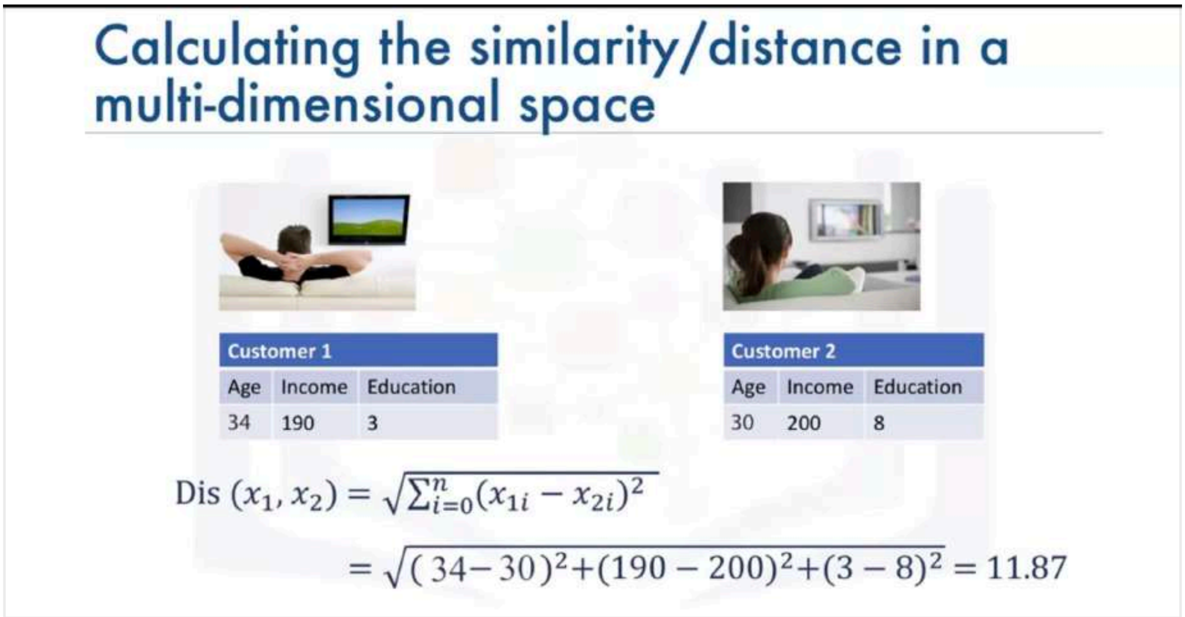
Customer 1	
Age	Income
34	190

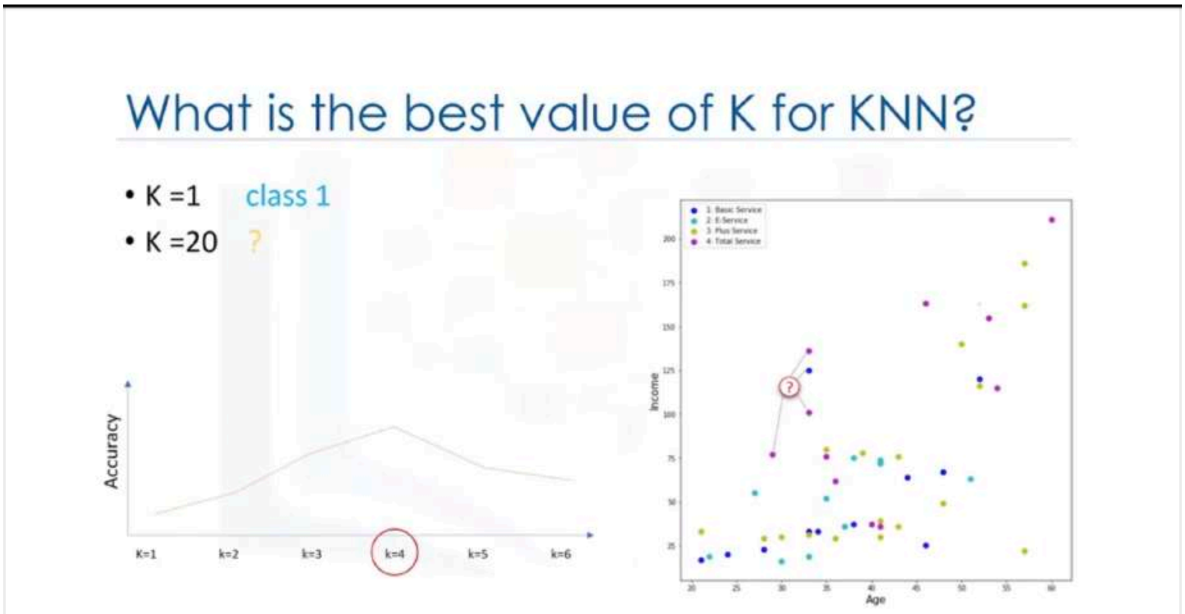


Customer 2	
Age	Income
30	200

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

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

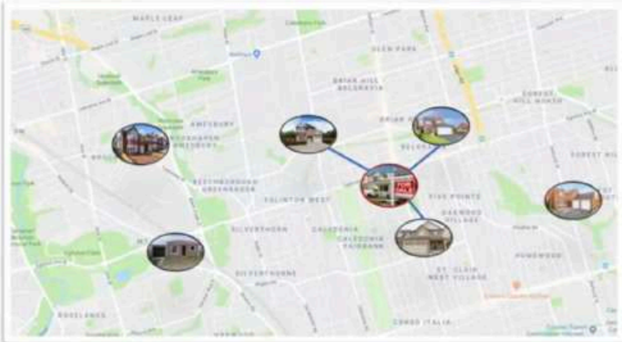




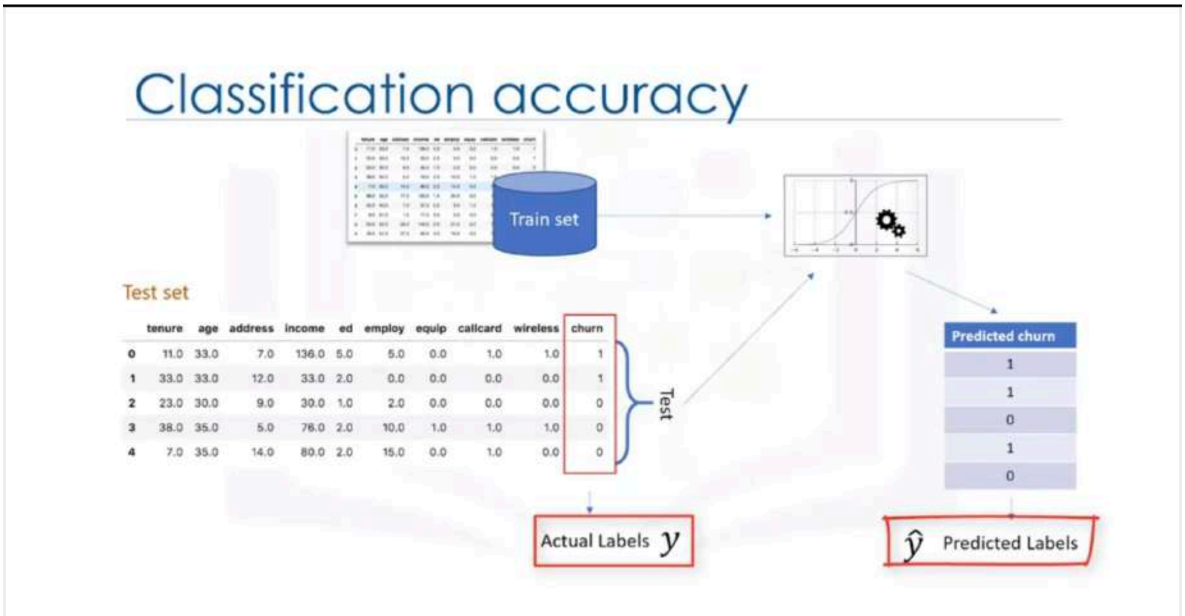
La solucion general es reservar una parte de la data para medir la accuracy y asi probar cual es el k optimo para usar.

Computing continuous targets using KNN

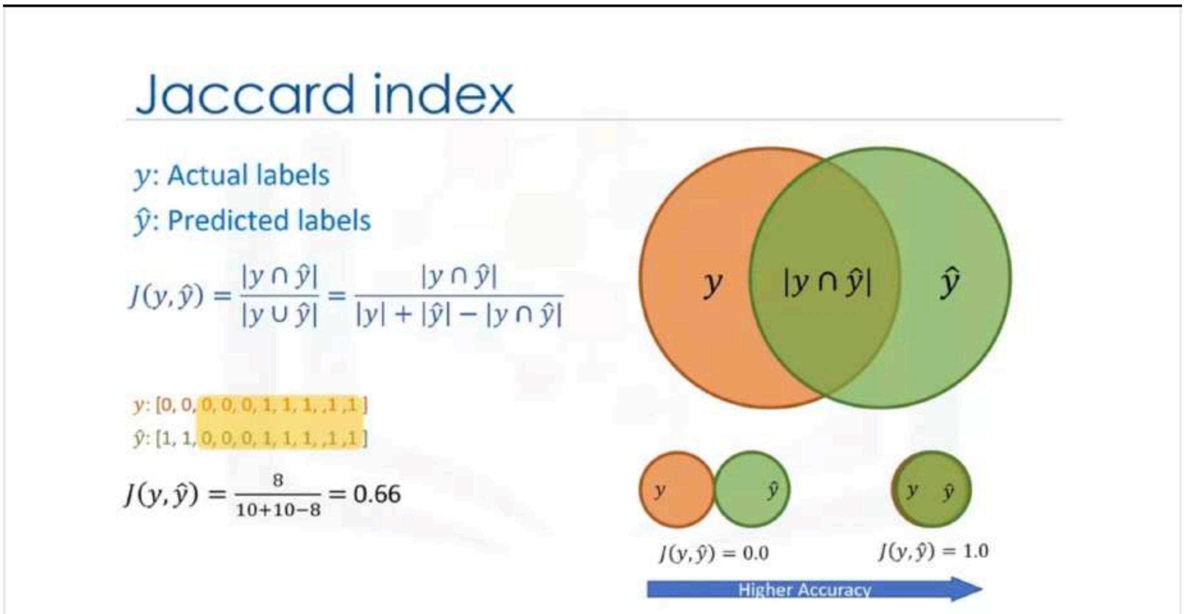
- KNN can also be used for regression



Evaluation Metrics in Classification

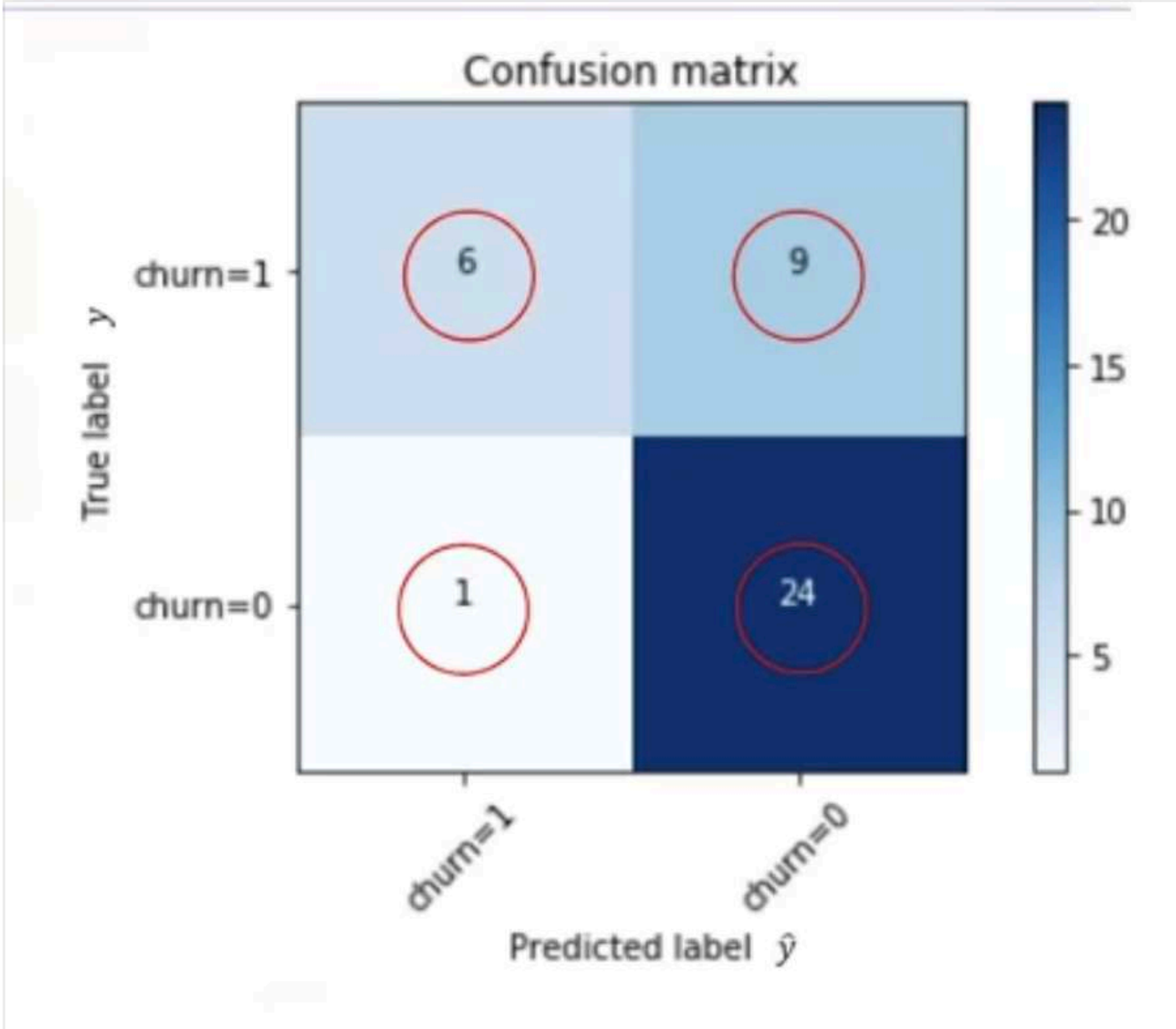


Jaccard índex:
Segun el tamaño de la intersección entre los valores posta t los predecidlos:

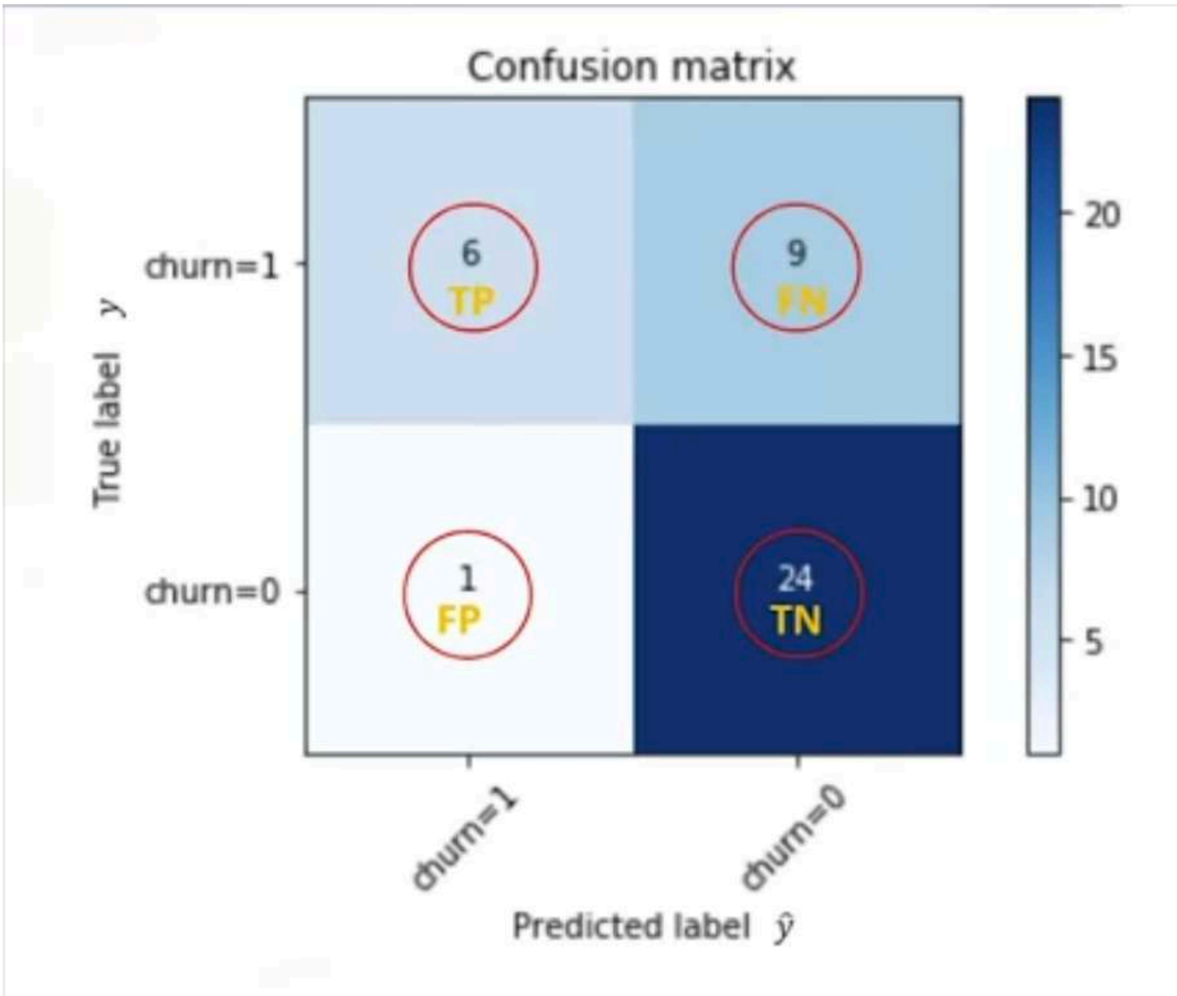


F1-Score
Confussion matrix —> cada fila contiene los valores posta de los loables del test set. Las columnas muestran los valores predecidos por classifier.

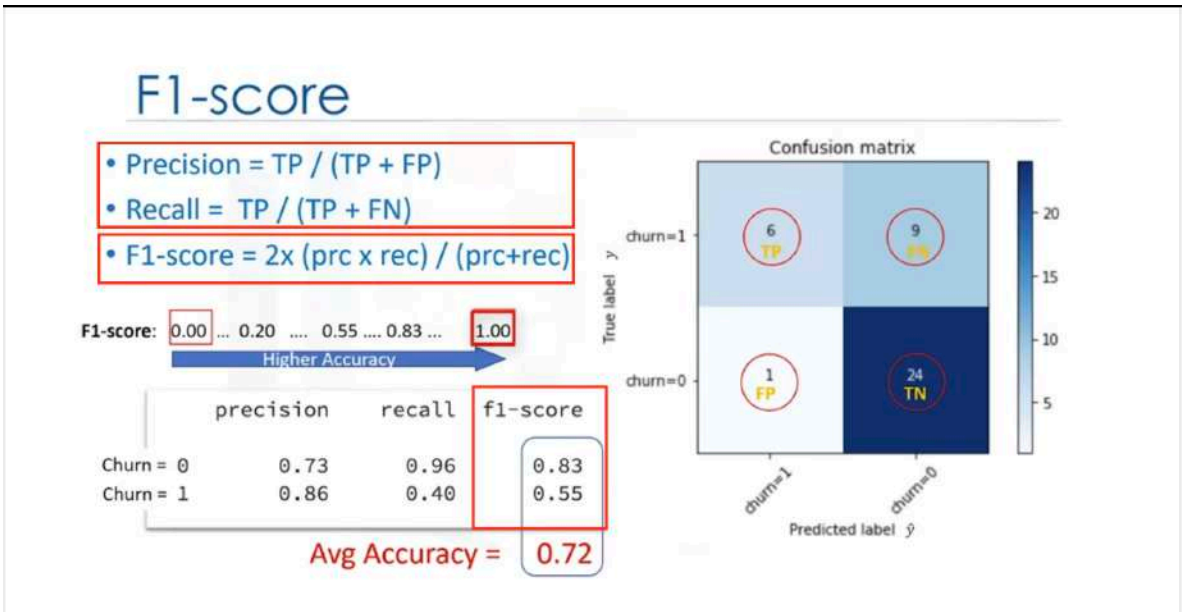
Por ejemplo, miremos la fila uno. Para esta fila, tenemos costumers cuyos churn valúes fueron 1 en el test set. De 40 costumers(6+9+24+1), el churn de 15 (6+9) de ellos es 1. Y de esos 15, 6 fueron predichos correctamente como uno por el classifier y 9 como cero. Esto significa que se equivoco en 9 el 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 especifico 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 tambien calcular el F1 acore:



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

LogLoss

Aveces el output de mi classifier puede ser una probabilidad. Por ejemplo en este caso, la probabilidad de churn de mi costumer.