Entrenamiento y selección de modelos ML

Equipo Innovación Copec

Roberto Muñoz
Senior Data Scientist
Digital Data Analytics
EY

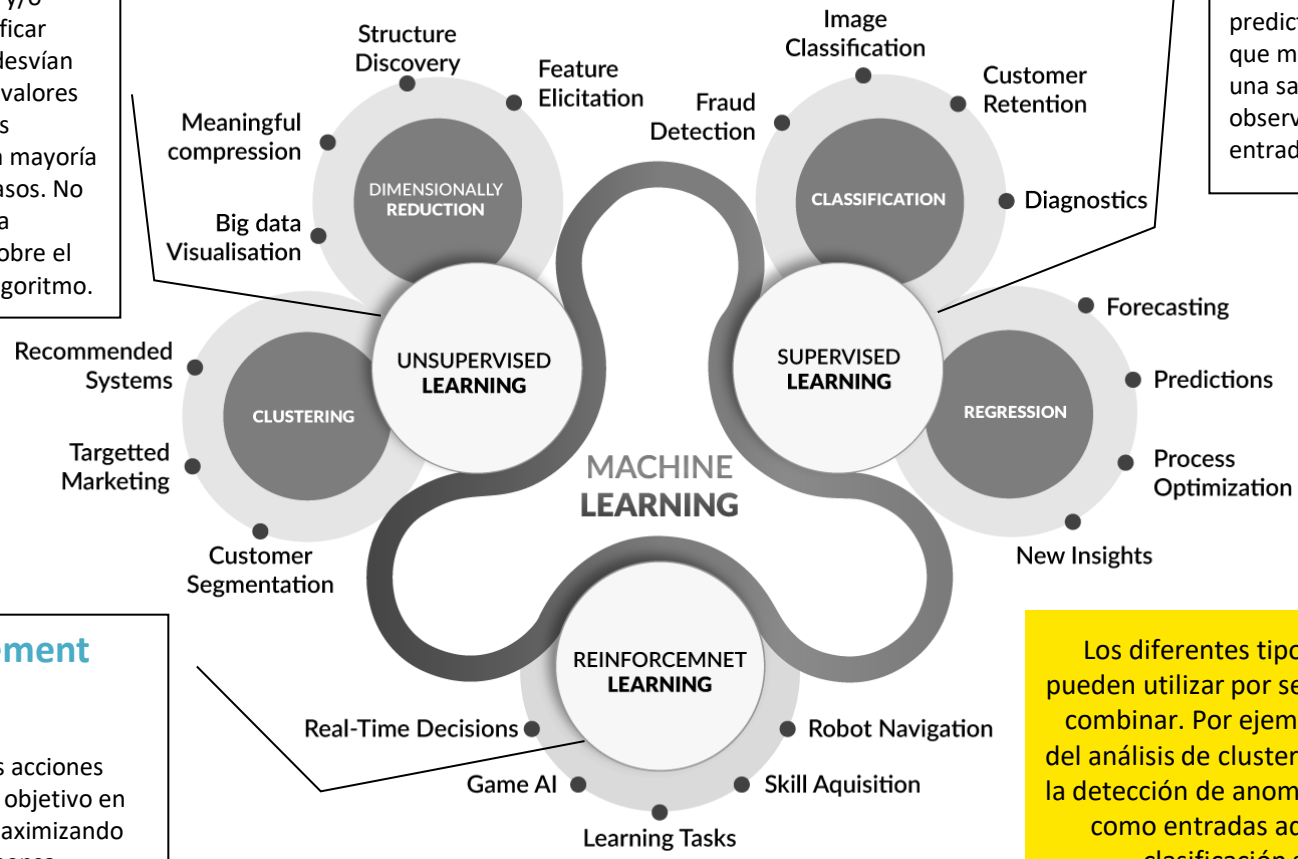
Tipos de aprendizaje

Unsupervised learning

Los modelos pueden agrupar datos no estructurados y/o pueden identificar casos que se desvían fuertemente (valores atípicos) de los patrones de la mayoría de los otros casos. No se proporciona información sobre el resultado al algoritmo.

Supervised learning

Construyendo modelos predictivos o de clasificación que mapean una entrada a una salida en base a observaciones de pares de entrada-salida anteriores.



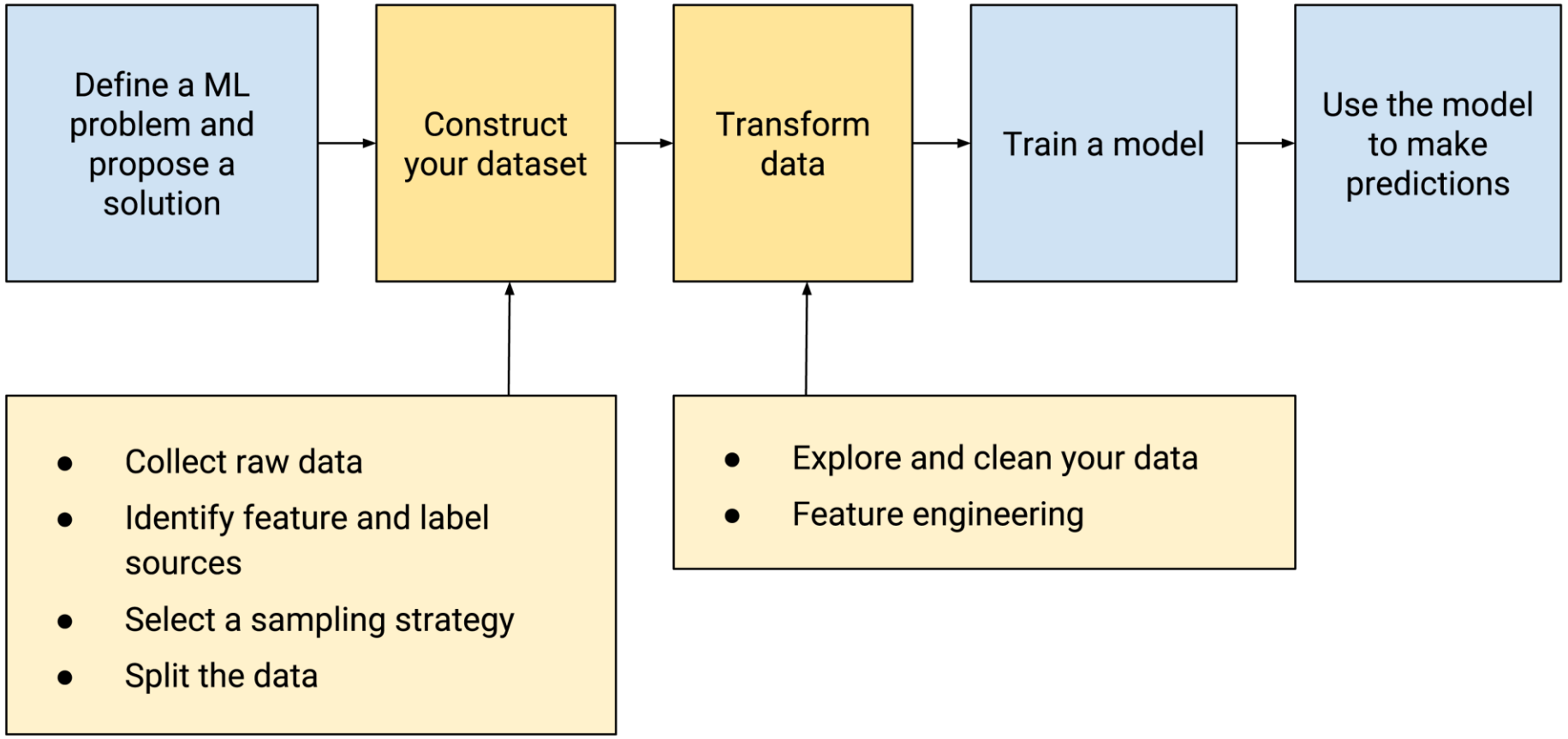
Reinforcement learning

Determinar las acciones para lograr un objetivo en un entorno, maximizando alguna recompensa.

Los diferentes tipos de enfoques se pueden utilizar por separado o se pueden combinar. Por ejemplo: Los resultados del análisis de clustering no supervisado y la detección de anomalías se pueden usar como entradas adicionales en una clasificación supervisada.

Data preparation

Pasos preparación de datos



Transformar datos

- Obligatorios

  - Conversión de características no numéricas en numéricas. No puede hacer una multiplicación de matrices en una cadena, por lo que debemos convertir la cadena a alguna representación numérica.

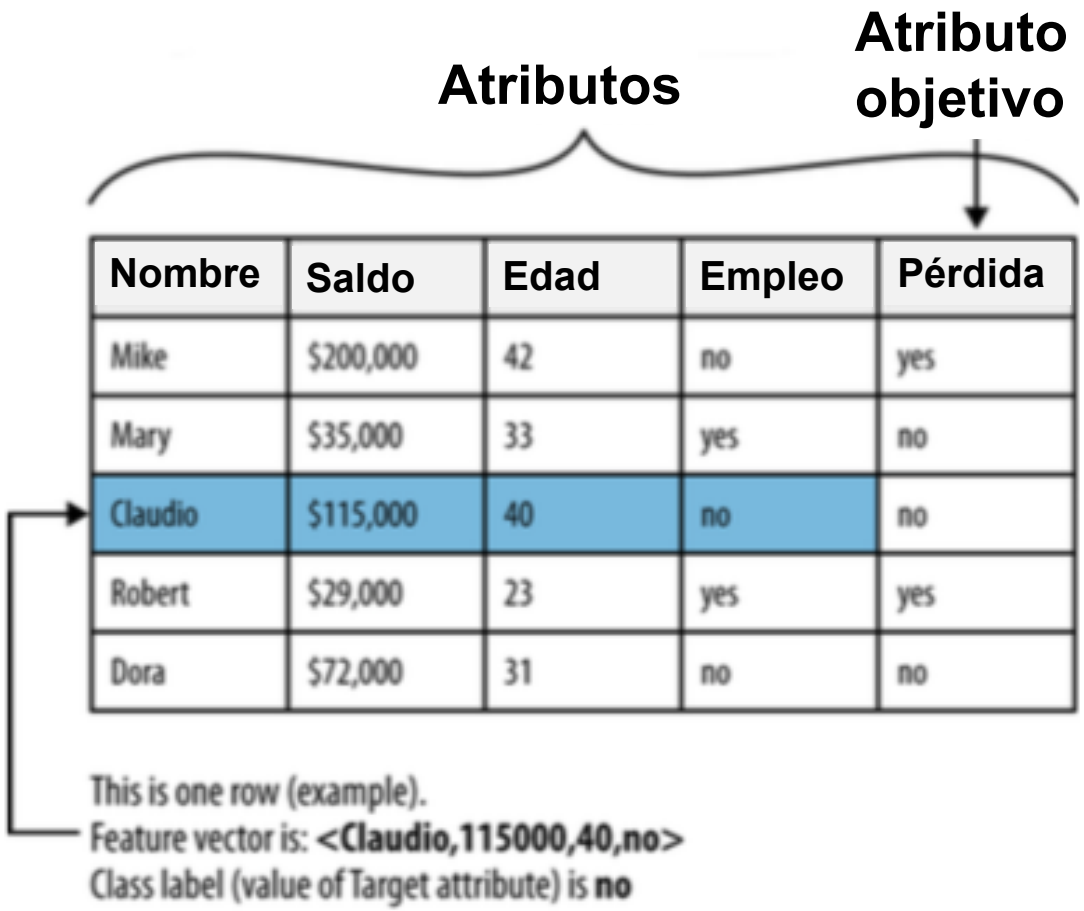
  - Cambiar el tamaño de las entradas a un tamaño fijo. Los modelos lineales y las redes neuronales de avance tienen un número fijo de nodos de entrada, por lo que sus datos de entrada siempre deben tener el mismo tamaño.

    Por ejemplo, los modelos de imágenes necesitan remodelar las imágenes de su conjunto de datos a un tamaño fijo.

Transformar datos

- Opcionales
  - Tokenización o transformación de texto en minúsculas.
  - Normalizar features numéricos.
    a mayoría de los modelos funcionan mejor después de normalizar.
  - Permitir que los modelos lineales introduzcan no linealidades en el espacio de features.

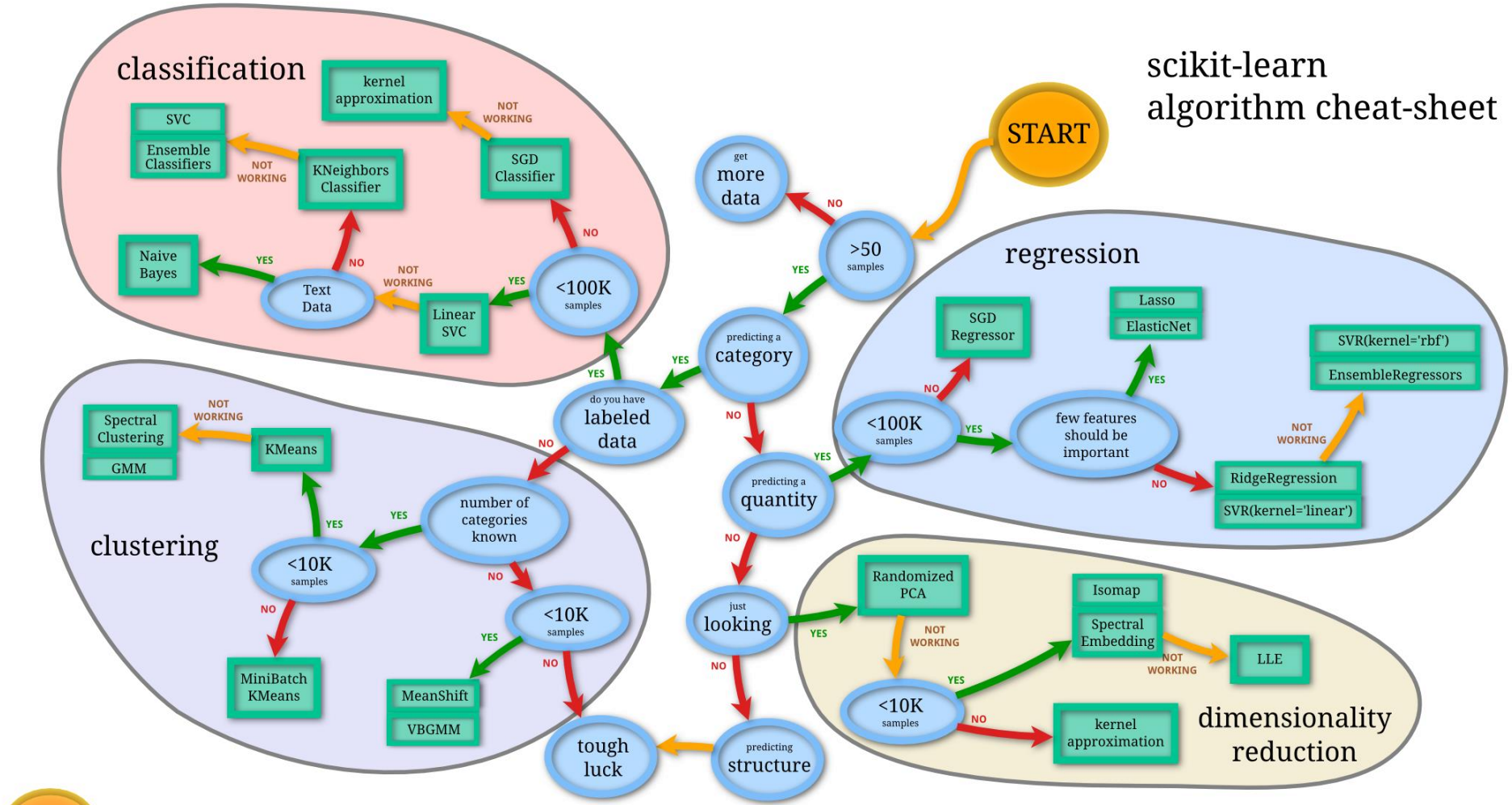
Evaluación de créditos bancarios



Model training

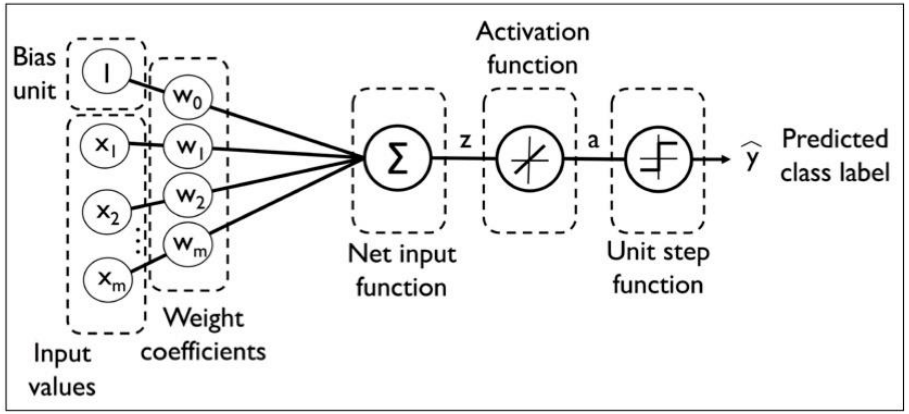
Créditos: Interstellar

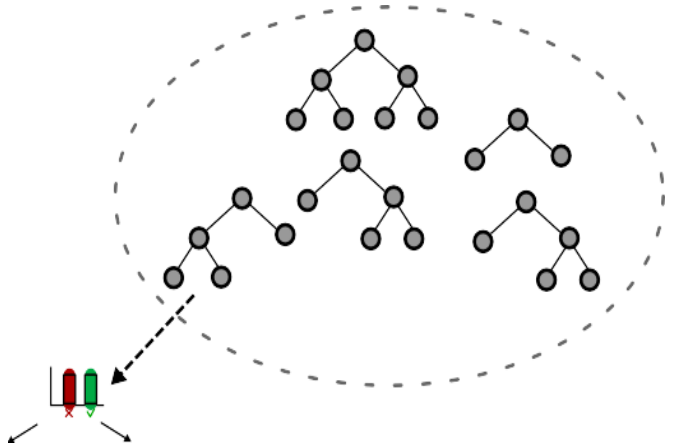
Scikit-learn methods



Clasificación

- Los métodos más usados para resolver los problemas de clasificación en ML son
  - Support Vector Machine (SVM)
  - Arbol de decisión
  - Random forest
  - Deep Neural Networks





Uso de scikitlearn

1. Logistic Regression

[4]:
```
# We will use the data frame where we had created dummy variables
y = df_dummies['Churn'].values
X = df_dummies.drop(columns = ['Churn'])

# Scaling all the variables to a range of 0 to 1
from sklearn.preprocessing import MinMaxScaler
features = X.columns.values
scaler = MinMaxScaler(feature_range = (0,1))
scaler.fit(X)
X = pd.DataFrame(scaler.transform(X))
X.columns = features
```

Es importante escalar las variables en la regresión logística para que todas estén dentro de un rango de 0 a 1. Esto me ayudó a mejorar la precisión del 79,7% al 80,7%.

Además, notará a continuación que la importancia de las variables también está alineada con lo que estamos viendo en el algoritmo Random Forest y la EDA que realizamos anteriormente.

[5]:
```
# Create Train & Test Data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

[6]:
```
# Running logistic regression model
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
result = model.fit(X_train, y_train)
```

[7]:
```
from sklearn import metrics
prediction_test = model.predict(X_test)
# Print the prediction accuracy
print (metrics.accuracy_score(y_test, prediction_test))
```

```
0.8075829383886256
```

[8]:
```
# To get the weights of all the variables
weights = pd.Series(model.coef_[0],
                 index=X.columns.values)
print (weights.sort_values(ascending = False)[:10].plot(kind='bar'))
```

Model selection

Créditos: Interstellar

Matriz de confusión

		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Inglés

		Predicción	
		Positivos	**Negativos**
Observación	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Español

Recall, Precision, Accuracy

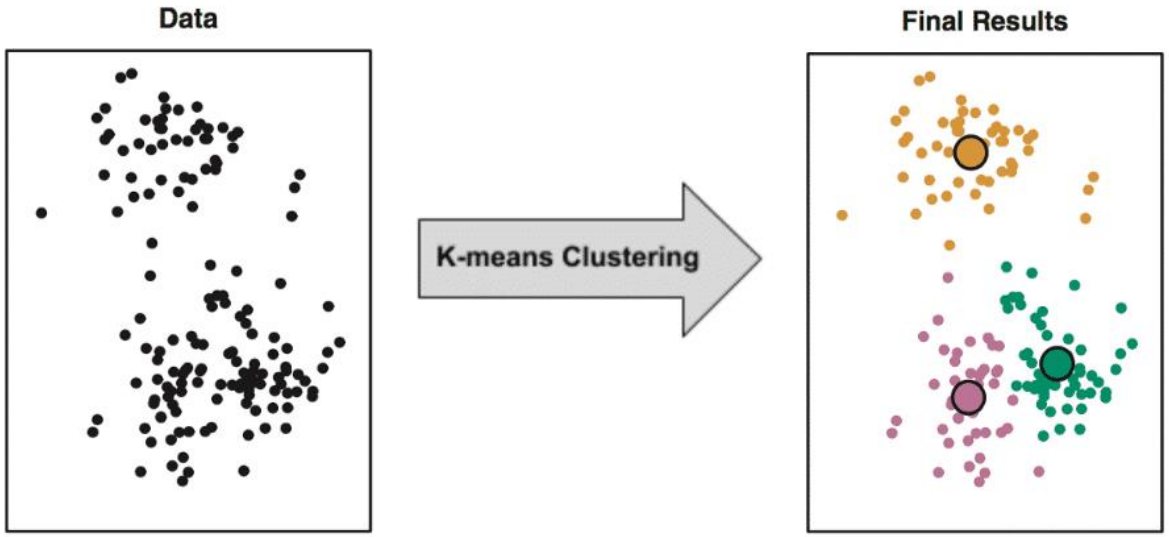
$$\text{Recall} = \frac{tp}{tp + fn}$$

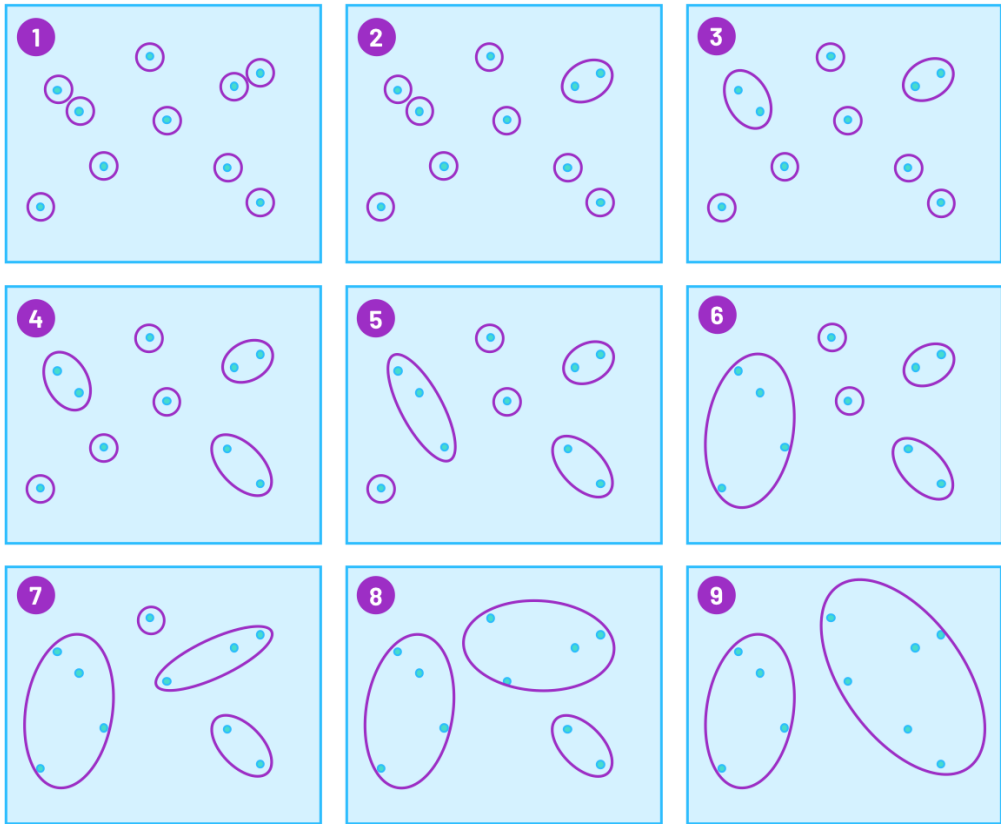
$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

- **Recall**: Sensibilidad
- **Precision**: Precisión
- **Accuracy**: Exactitud

- Entrenamiento: Ajustar los parámetros del algoritmo de forma tal de que se minimicen la cantidad de predicciones que no correspondan a la etiqueta original.
- Recall: Porcentaje de clasificados correctamente como positivos sobre todos los que realmente eran positivos.
- Precision: Porcentaje de clasificados correctamente como positivos sobre todos los clasificados como positivos.
- Accuracy: Porcentaje de clasificados correctamente.


Data
K-means Clustering
Final Results


1
2
3
4
5
6
7
8
9

Taller