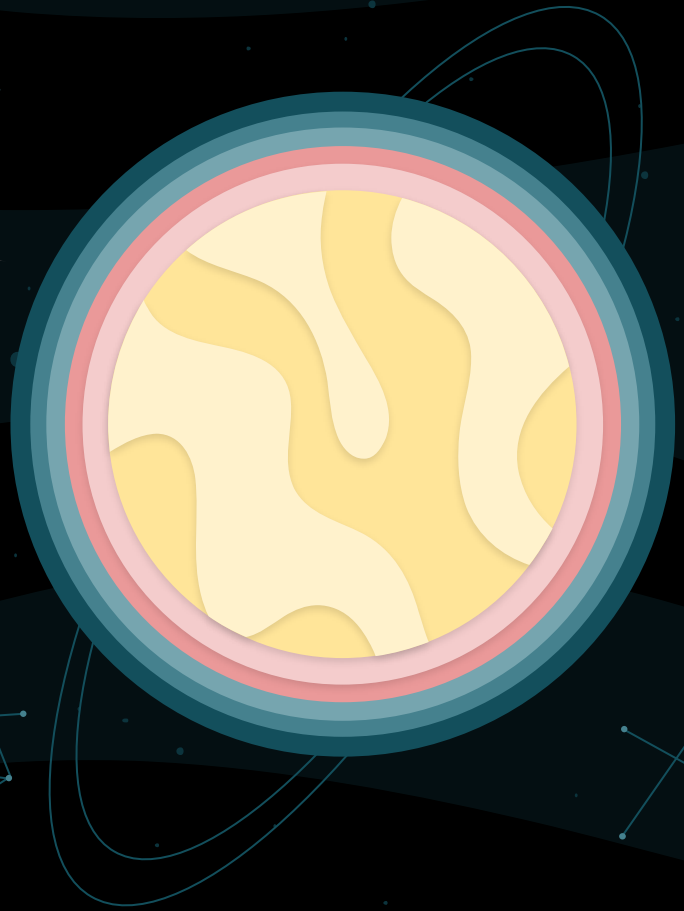
Modelling & Evaluation

Artificial Intelligence Project Cycle

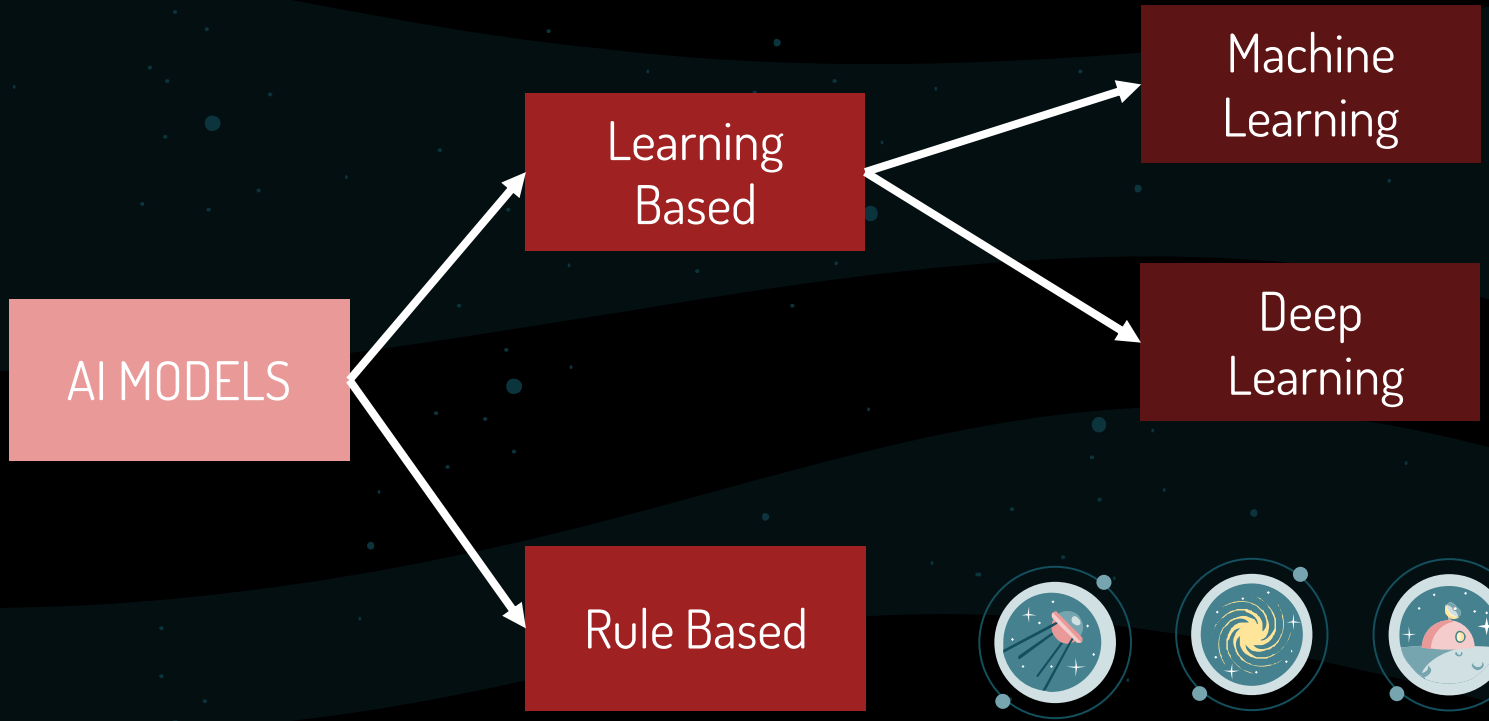


Modelling

Refers to **developing algorithms (models)** which can be trained to get intelligent output

So, we **write a code** for machine to become **intelligent**!

Classification of AI Modelling



Learning Based

Relationships or patterns in data

are figured out by the machine

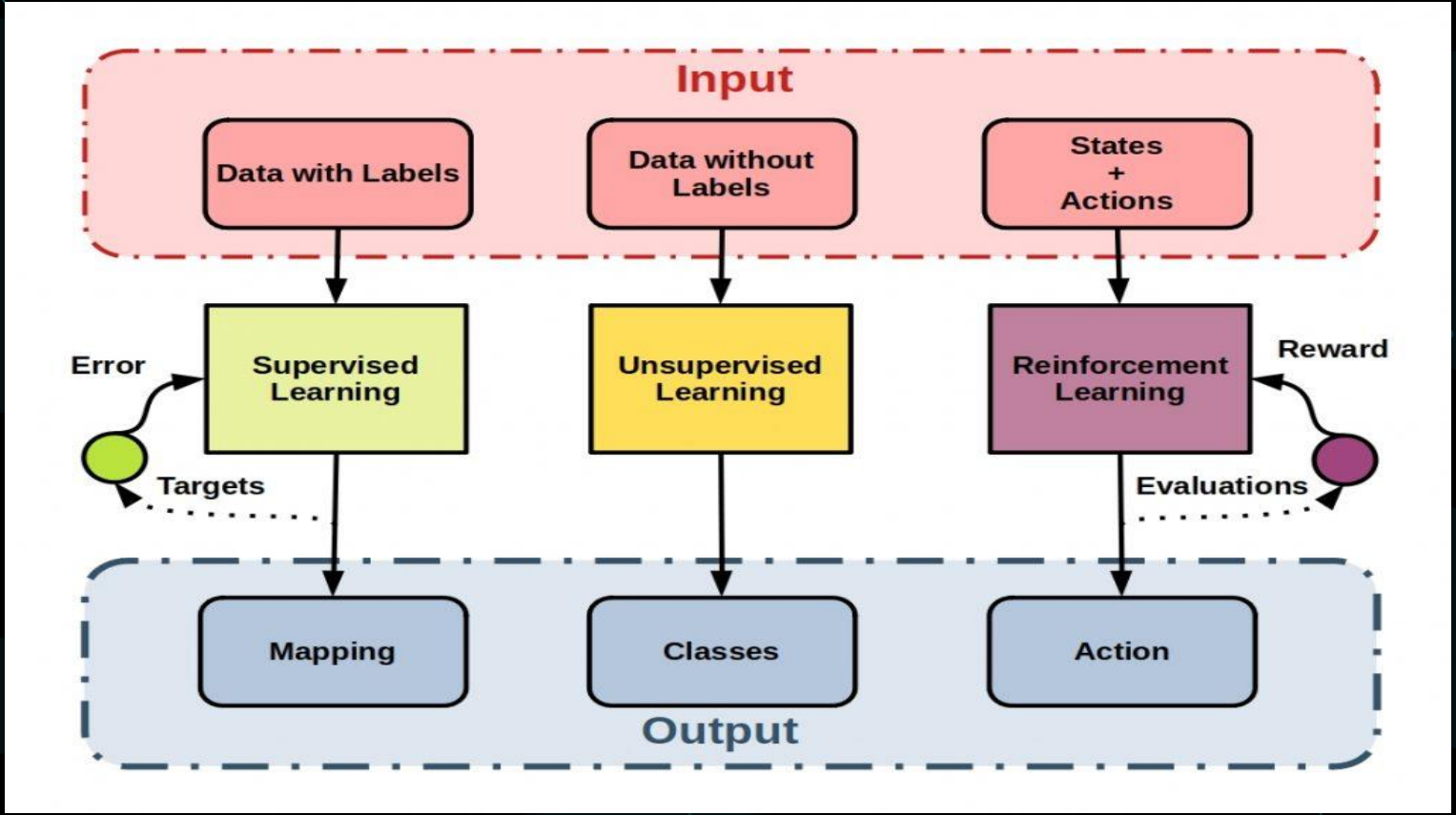
(developers only provide the raw data!)

Rule Based

Relationships or patterns in data

are **determined by developers**

LEARNING BASED models



Supervised

Task Driven:

Classification, regression

Unsupervised

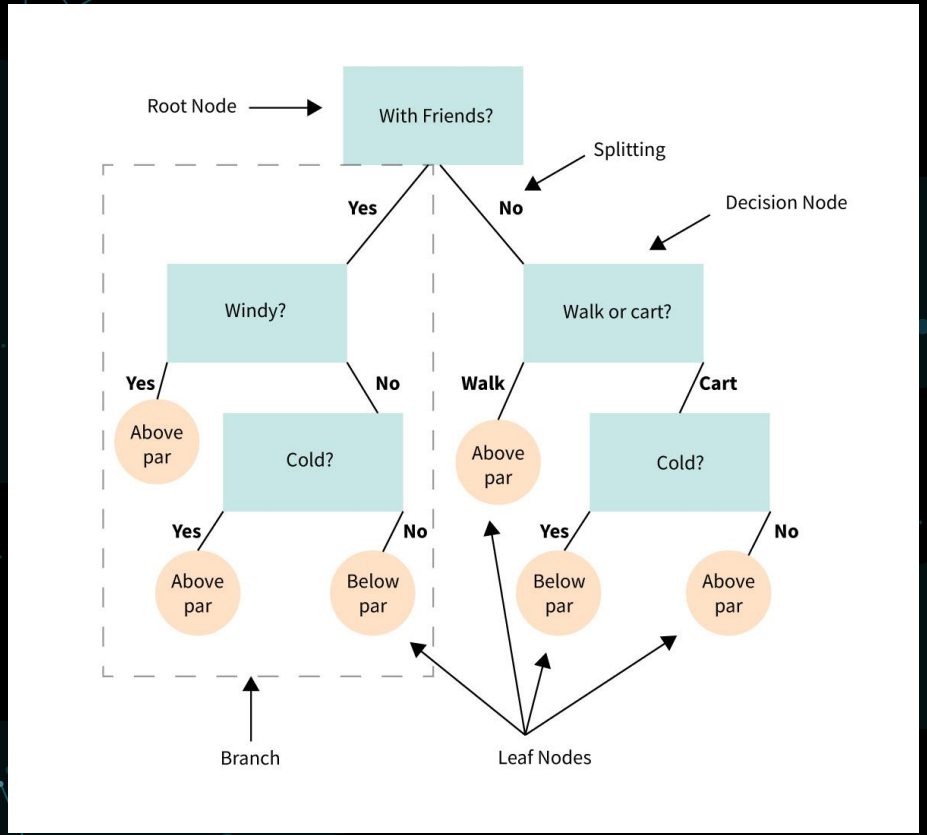
Data Driven:

Clustering, associating

Reinforcement

Learn from Mistakes:

Markov Decision Process, Q Learning



RULE BASED models

Decision Trees

create a model that predicts the value of a target variable **by learning simple decision rules** inferred from the data features.

*Not to be confused with machine learning algorithms: decision trees

Example (KNN + Rule Based Model):

*Predicting Departure-Landing Airports from Boeing747 flight data (500 @ ±7000 datas).

```
airport = pd.read_csv('airports.csv') # Adjust file directory

from sklearn.neighbors import KNeighborsClassifier
X  = airport[['longitude_deg', 'latitude_deg']]
y = airport['name']
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X, y)

loc_df['Airport Depart']= knn.predict(loc_df[['Longitude Depart', 'Latitude Depart']])
loc_df['Airport Landing']= knn.predict(loc_df[['Longitude Landing', 'Latitude Landing']])
```

```
# Searching runway
temp_rw = temp[temp["gs_mps"] > 0]
temp_rw = temp_rw.reset_index()

for k in temp_rw.index:
    if k == temp_rw.index.max():
        break
    if temp_rw['time_s'].iloc[k+1] - temp_rw['time_s'].iloc[k] >= 100:
        bound = temp_rw['time_s'].iloc[k+1]  # Creating label for depart and landing situation

for k in temp_rw.index:
    if temp_rw['wow'].iloc[k] == 0:
        if temp_rw['time_s'].iloc[k] < bound:
            temp_rw['depart_land'].iloc[k] = 0
        else:
            temp_rw['depart_land'].iloc[k] = 1

temp_chi_dep = temp_rw['chi_deg'][temp_rw['depart_land']==0].mean()
temp_chi_land = temp_rw['chi_deg'][temp_rw['depart_land']==1].mean()
```

	Flight code	Airport Depart	Runway Depart
0	flight_10009	Memphis International Airport	36.0
1	flight_10013	Memphis International Airport	36.0
2	flight_10043	Detroit Metropolitan Wayne County Airport	3.0
3	flight_10049	Detroit Metropolitan Wayne County Airport	21.0
4	flight_10057	Minneapolis–Saint Paul International Airport /...	12.0
...	...	...	...
495	flight_50440	Eppley Airfield	14.0
496	flight_51000	Eppley Airfield	36.0
497	flight_51054	Eppley Airfield	14.0
498	flight_51056	Eppley Airfield	14.0
499	flight_59785	Duluth International Airport	9.0

	Flight code	Airport Landing	Runway Landing
0	flight_10009	Northwest Arkansas Regional Airport	34.0
1	flight_10013	Springfield Branson National Airport	14.0
2	flight_10043	Westchester County Airport	15.0
3	flight_10049	Des Moines International Airport	23.0
4	flight_10057	Eppley Airfield	33.0
...	...	...	...
495	flight_50440	Detroit Metropolitan Wayne County Airport	21
496	flight_51000	Minneapolis–Saint Paul International Airport /...	12
497	flight_51054	Memphis International Airport	36
498	flight_51056	Memphis International Airport	36
499	flight_59785	Minneapolis–Saint Paul International Airport /...	12

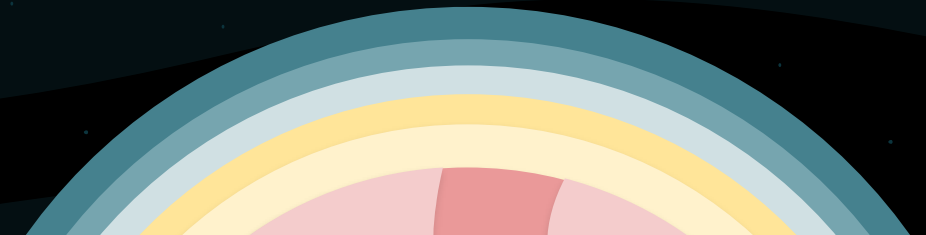
Evaluation

Once a model has been created and trained, it needs to go through **proper testing** so it can calculate the **efficiency** and **performance** of the model.

Hence, the model is **tested with Testing Data** which was separated from Training Data

Nah, then

How does the **testing process** work?

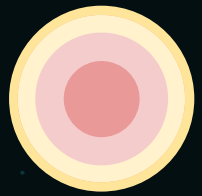


Introducing:

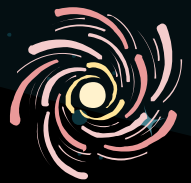
The Confusion Matrix

Correct predictions

False predictions



	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)



$$\text{Accuracy} = \frac{correct\ predictions}{total\ predictions}$$

other metrics

SENSITIVITY

PRECISION

SPECIFICITY

RECALL

Evaluating the previous model in example ....