The background of the slide features a dark blue, almost black, field. On the left, a human hand is shown from the wrist up, with the index finger pointing towards the right. The hand is rendered in a realistic style with some lighting effects. To the right of the hand, there is a complex, futuristic digital interface. It consists of several concentric circles, some of which are filled with teal and blue geometric shapes like squares and rectangles. In the center of these circles are three interlocking gears. The overall aesthetic is high-tech and digital.

# Models Deployment

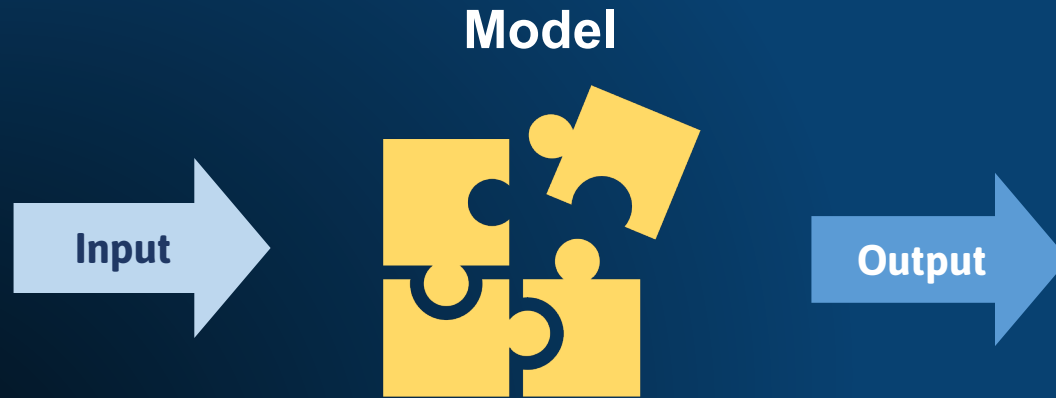
Cláudia Antunes

Instituto Superior Técnico – Universidade de Lisboa

# MODEL LEARNING



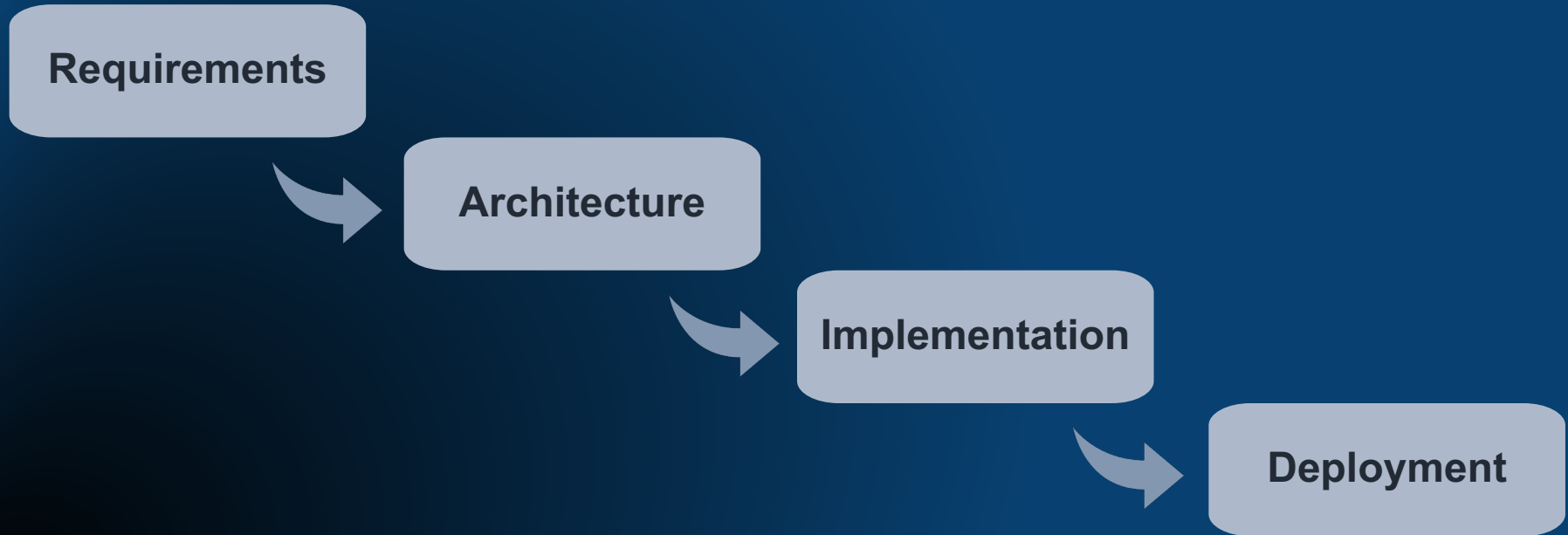
# MODEL USAGE



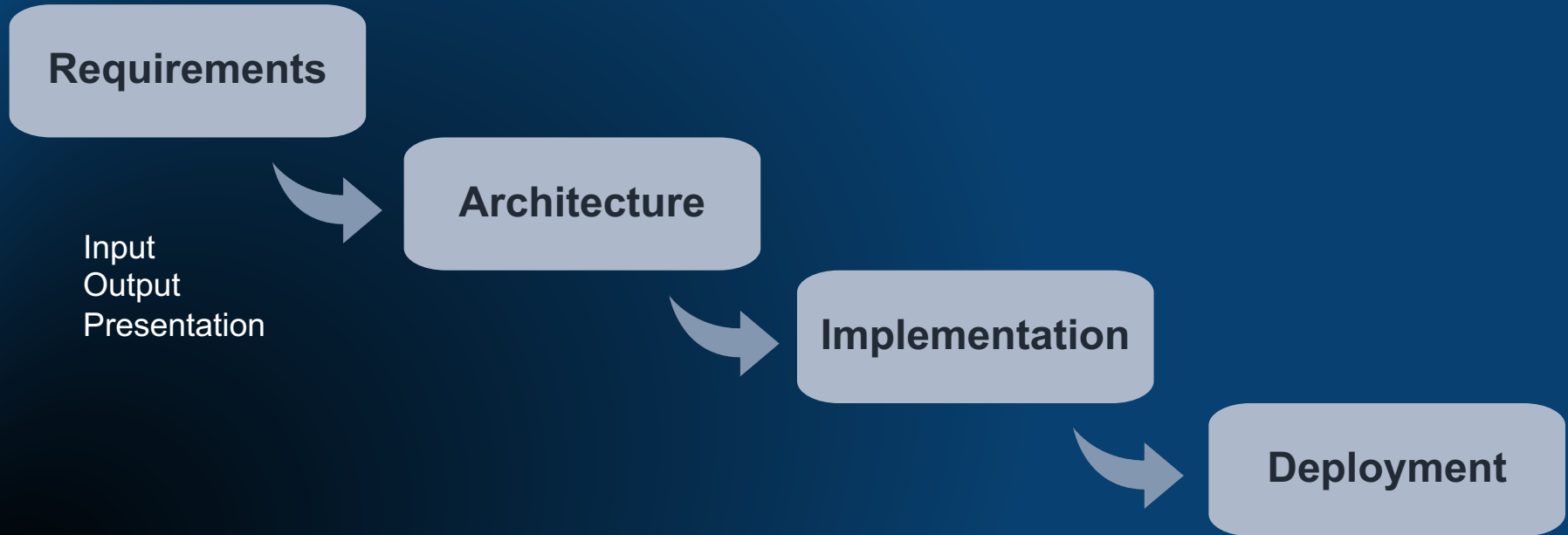


# Deployment

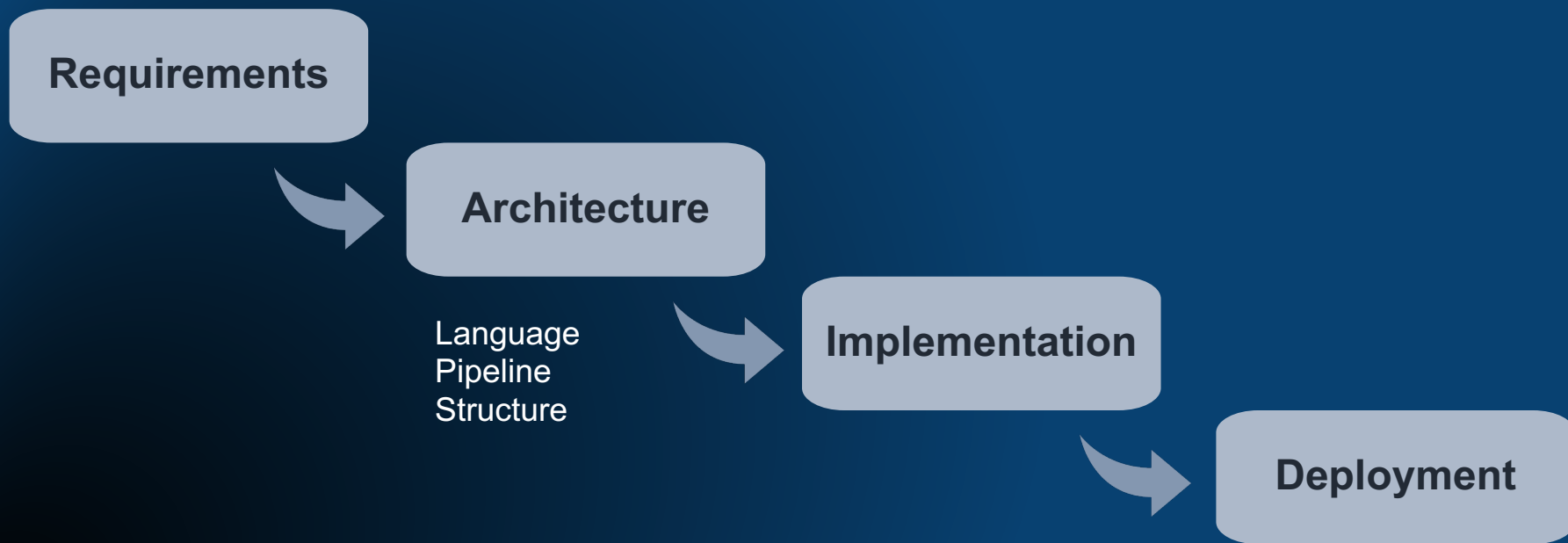
# DEVELOPMENT CYCLE



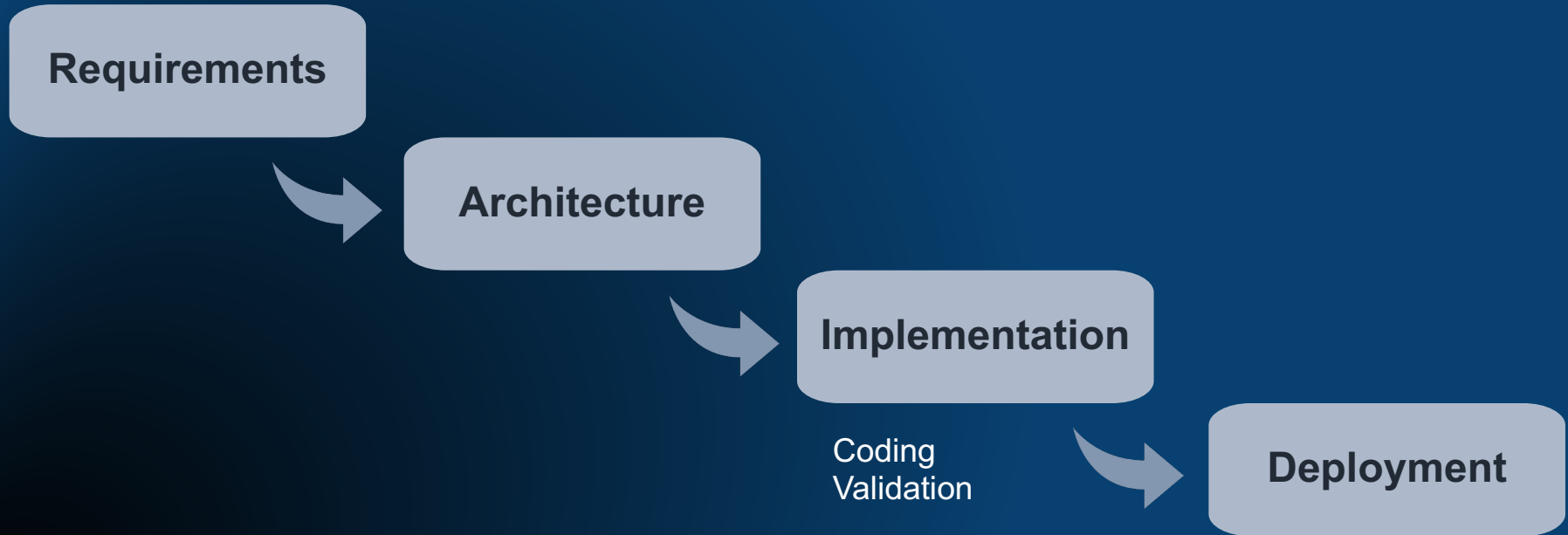
# DEVELOPMENT CYCLE



# DEVELOPMENT CYCLE



# DEVELOPMENT CYCLE





# IMPLEMENTATION



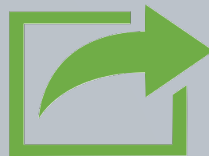
Save the  
model



Conform  
input  
data



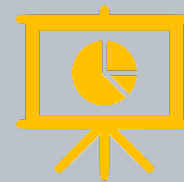
Load the  
model



Feed the  
data



Run the  
model



Present  
the  
results

# SAVE THE MODEL – PICKLE



```
# Fit the model on training set
model = DecisionTreeRegressor()
model.fit(X_train, Y_train)
```

```
# save the model to disk
from pickle import dump
filename = 'tree_model.sav'
dump(model, open(filename, 'wb'))
```

# LOAD THE MODEL – PICKLE



```
# load the model from disk
from pickle import load
filename = 'tree_model.sav'
loaded_model = load(open(filename, 'rb'))
result = loaded_model.score(X_test, Y_test)
```

# ALTERNATIVES



# PMML – PREDICTIVE MODEL MARKUP LANGUAGE



## Header

- general information (description, application used to generate the model, timestamp)

## Data Dictionary

- fields definitions (continuous, categorical, or ordinal)
- value ranges
- data type

## Data Transformations

- Normalization
- Discretization
- Value mapping
- Functions (custom and built-in)
- Aggregation

## Model

- Model Name (attribute modelName)
- Function Name (attribute functionName)
- Algorithm Name (attribute algorithmName)
- Activation Function (attribute activationFunction)
- Number of Layers (attribute numberOfLayers)

## Mining Schema:

- Name
- Usage type (active, predicted, and supplementary)
- Outlier Treatment
- Missing Value Replacement Policy
- Missing Value Treatment

## Targets

## Output

```

<PMML xmlns="http://www.dmg.org/PMML-4_1" version="4.1">
  <Header copyright="KNIME"> <Application name="KNIME" version="2.8.0"/> </Header>
  <DataDictionary numberOfFields="5">
    <DataField name="sepal_length" optype="continuous" dataType="double">
      <Interval closure="closedClosed" leftMargin="4.3" rightMargin="7.9"/>
    </DataField>
    <DataField name="sepal_width" optype="continuous" dataType="double">
      <Interval closure="closedClosed" leftMargin="2.0" rightMargin="4.4"/>
    </DataField>
    <DataField name="petal_length" optype="continuous" dataType="double">
      <Interval closure="closedClosed" leftMargin="1.0" rightMargin="6.9"/>
    </DataField>
    <DataField name="petal_width" optype="continuous" dataType="double">
      <Interval closure="closedClosed" leftMargin="0.1" rightMargin="2.5"/>
    </DataField>
    <DataField name="class" optype="categorical" dataType="string">
      <Value value="Iris-setosa"/>
      <Value value="Iris-versicolor"/>
      <Value value="Iris-virginica"/>
    </DataField>
  </DataDictionary>
  <TreeModel modelName="DecisionTree" functionName="classification" splitCharacteristic="binarySplit"
    missingValueStrategy="lastPrediction" noTrueChildStrategy="returnNullPrediction">
    <MiningSchema>
      <MiningField name="sepal_length" invalidValueTreatment="asIs"/>
      <MiningField name="sepal_width" invalidValueTreatment="asIs"/>
      <MiningField name="petal_length" invalidValueTreatment="asIs"/>
      <MiningField name="petal_width" invalidValueTreatment="asIs"/>
      <MiningField name="class" invalidValueTreatment="asIs" usageType="predicted"/>
    </MiningSchema>
  </TreeModel>
</PMML>

```





```
<Node id="0" score="Iris-setosa" recordCount="150.0"><True/>
  <ScoreDistribution value="Iris-setosa" recordCount="50.0"/>
  <ScoreDistribution value="Iris-versicolor" recordCount="50.0"/>
  <ScoreDistribution value="Iris-virginica" recordCount="50.0"/>
  <Node id="1" score="Iris-setosa" recordCount="50.0">
    <SimplePredicate field="petal_width" operator="lessOrEqual" value="0.6"/>
    <ScoreDistribution value="Iris-setosa" recordCount="50.0"/>
    <ScoreDistribution value="Iris-versicolor" recordCount="0.0"/>
    <ScoreDistribution value="Iris-virginica" recordCount="0.0"/>
  </Node>
  <Node id="2" score="Iris-versicolor" recordCount="100.0">
    <SimplePredicate field="petal_width" operator="greaterThan" value="0.6"/>
    <ScoreDistribution value="Iris-setosa" recordCount="0.0"/>
    <ScoreDistribution value="Iris-versicolor" recordCount="50.0"/>
    <ScoreDistribution value="Iris-virginica" recordCount="50.0"/>
    <Node id="3" score="Iris-versicolor" recordCount="54.0">
      <SimplePredicate field="petal_width" operator="lessOrEqual" value="1.7"/>
      <ScoreDistribution value="Iris-setosa" recordCount="0.0"/>
      <ScoreDistribution value="Iris-versicolor" recordCount="49.0"/>
      <ScoreDistribution value="Iris-virginica" recordCount="5.0"/>
    </Node>
    <Node id="10" score="Iris-virginica" recordCount="46.0">
      <SimplePredicate field="petal_width" operator="greaterThan" value="1.7"/>
      <ScoreDistribution value="Iris-setosa" recordCount="0.0"/>
      <ScoreDistribution value="Iris-versicolor" recordCount="1.0"/>
      <ScoreDistribution value="Iris-virginica" recordCount="45.0"/>
    </Node>
  </Node>
</Node>
</TreeModel>
```



*Thank  
you!*

