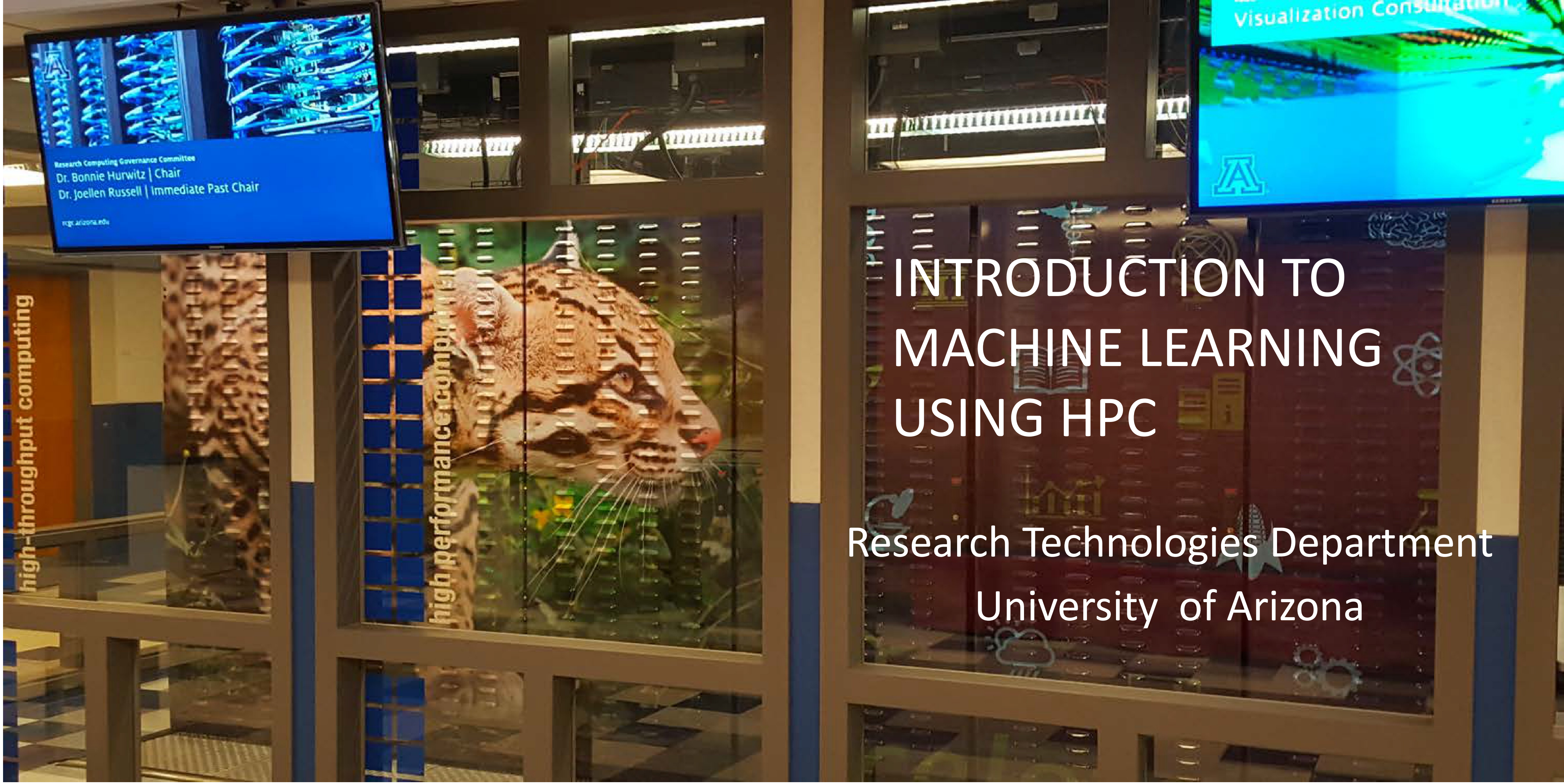
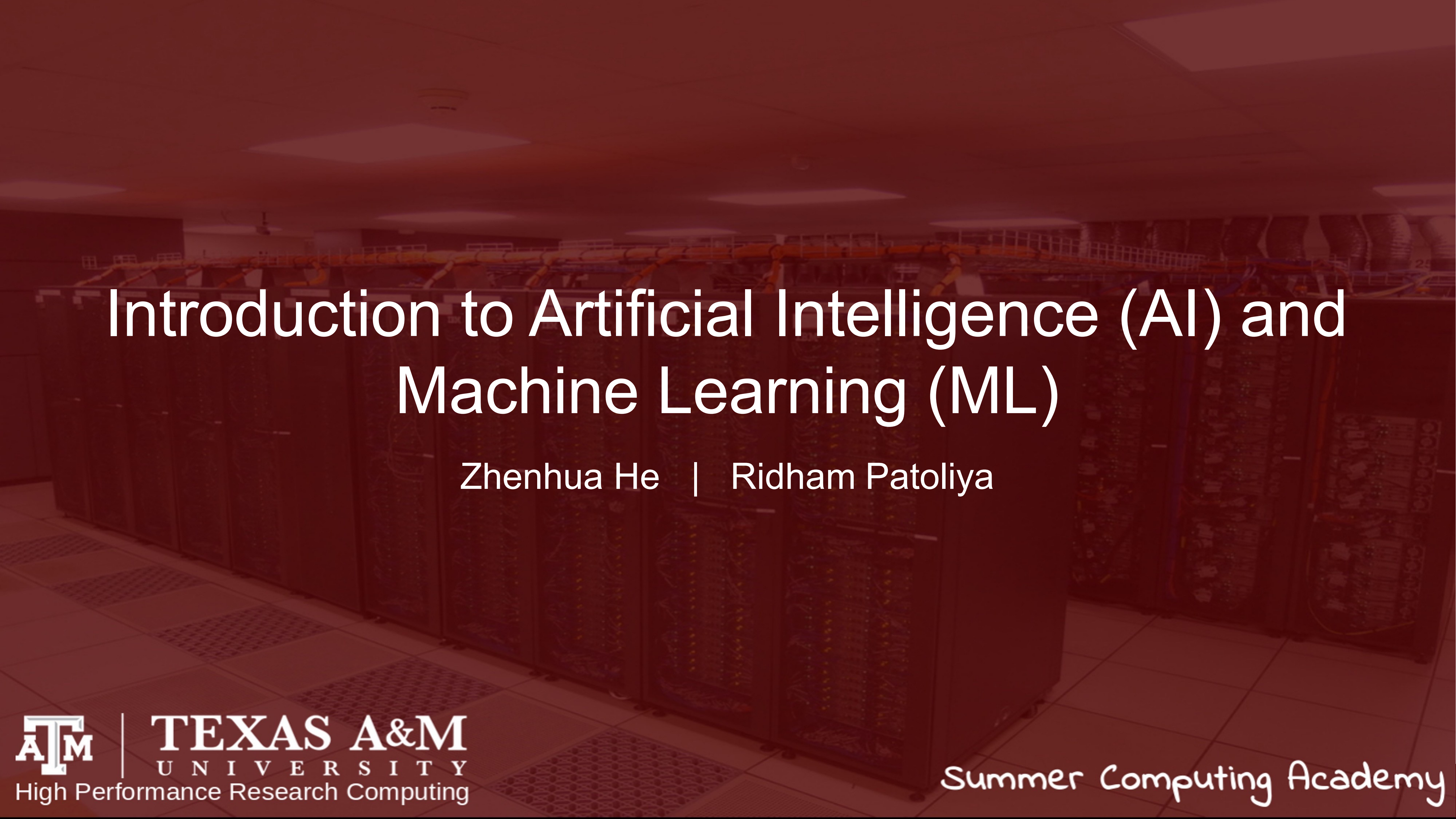

Research Computing Governance Committee
Dr. Bonnie Hurwitz | Chair
Dr. Joellen Russell | Immediate Past Chair
rcgc.arizona.edu
Visualization Consultation
high-throughput computing
high performance computing

INTRODUCTION TO MACHINE LEARNING USING HPC

Research Technologies Department

University of Arizona



Introduction to Artificial Intelligence (AI) and Machine Learning (ML)

Zhenhua He | Ridham Patoliya

TEXAS A&M UNIVERSITY
High Performance Research Computing

Summer Computing Academy

Learning objectives

Terminology of Machine Learning

The difference between AI and ML

The different types of machine learning techniques

Applications of machine learning techniques

Data Exploration

pandas

Data manipulation/analysis library

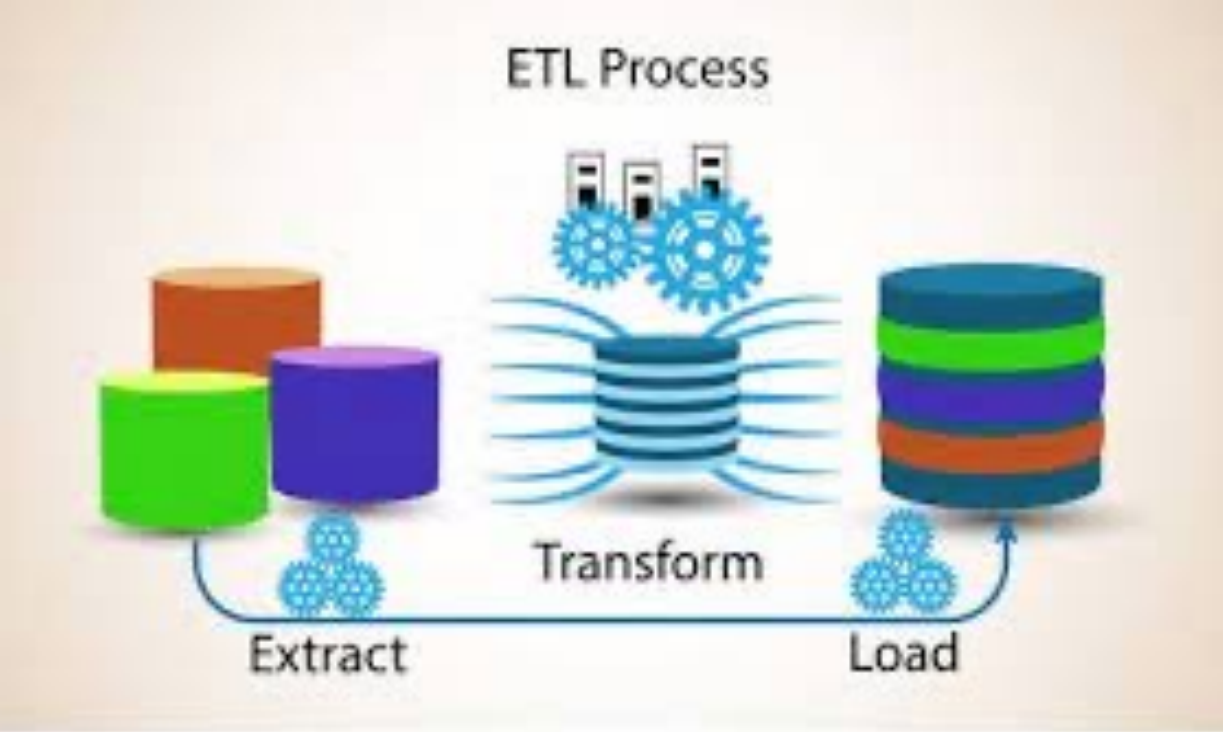
- Exercises

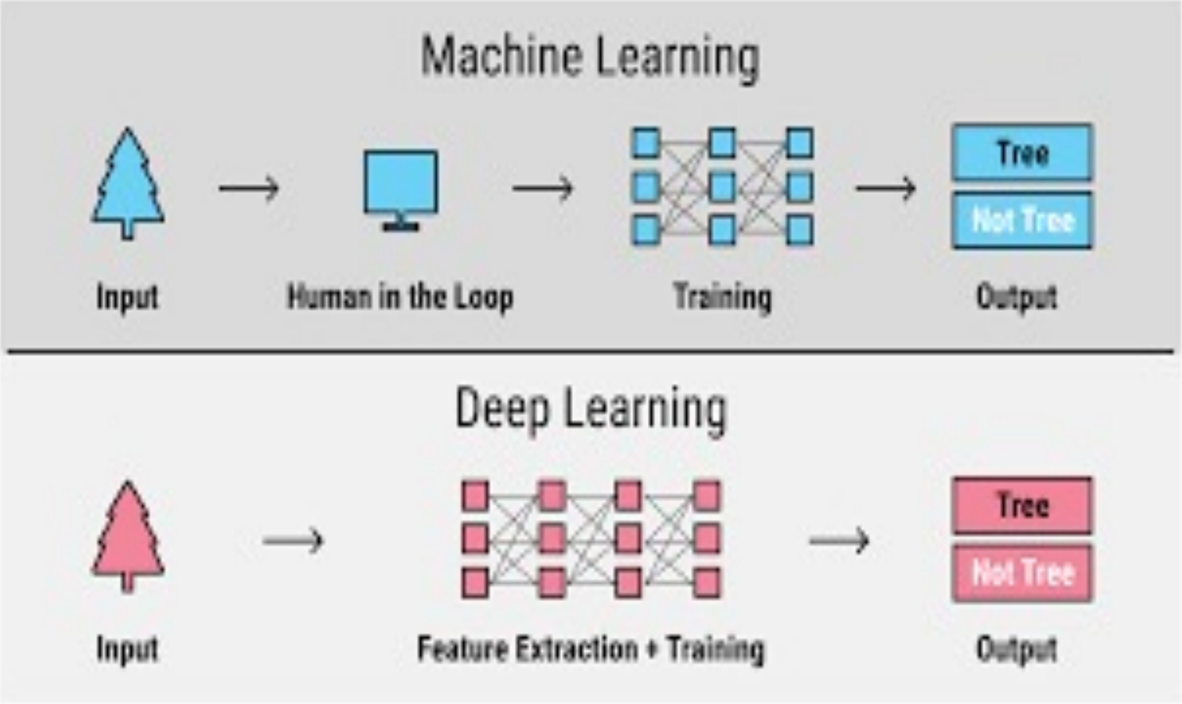
matplotlib

Data visualization library

- Exercises

ENTERPRISE DATA vs MACHINE LEARNING DATA





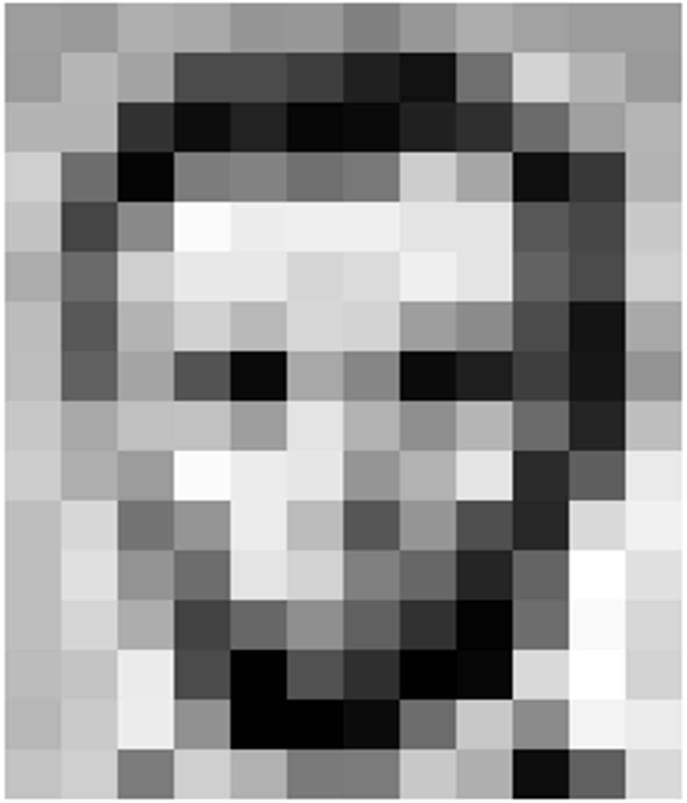
Formats of data

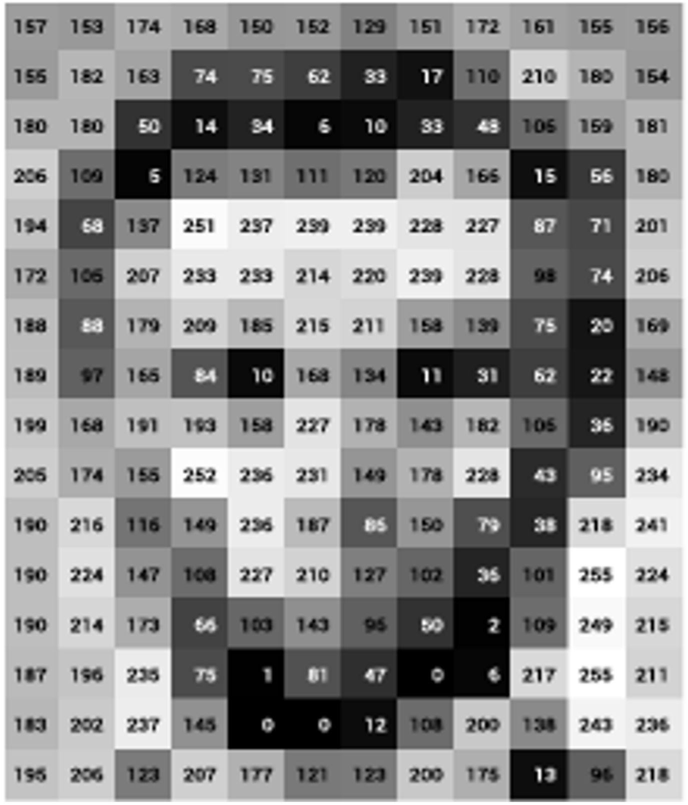
NUMBERS

TEXTS

IMAGES

Image (as seen by computers)





157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Types of data:

- Labelled
- Unlabelled

Labelled data

Dog

Cat

Dog

Dog

Cat

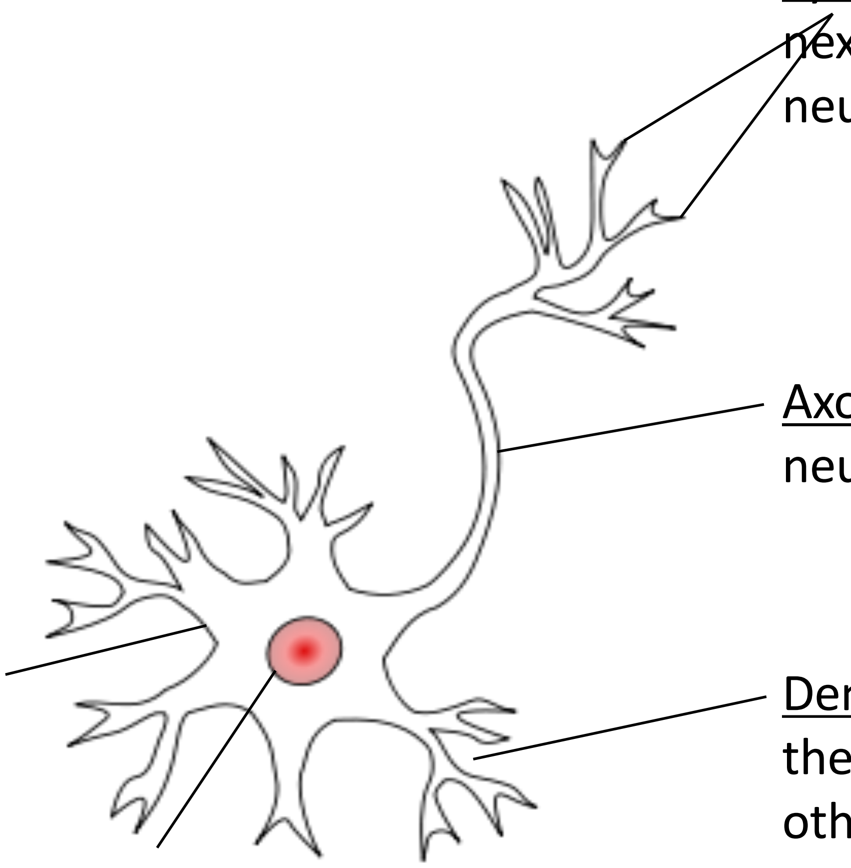
Cat

Dog

Cat

Unlabelled Data

Neural Networks and Biology



Synapses: connection to the next neuron (release of neurotransmitter)

Axon: delivers output from neuron to other neurons

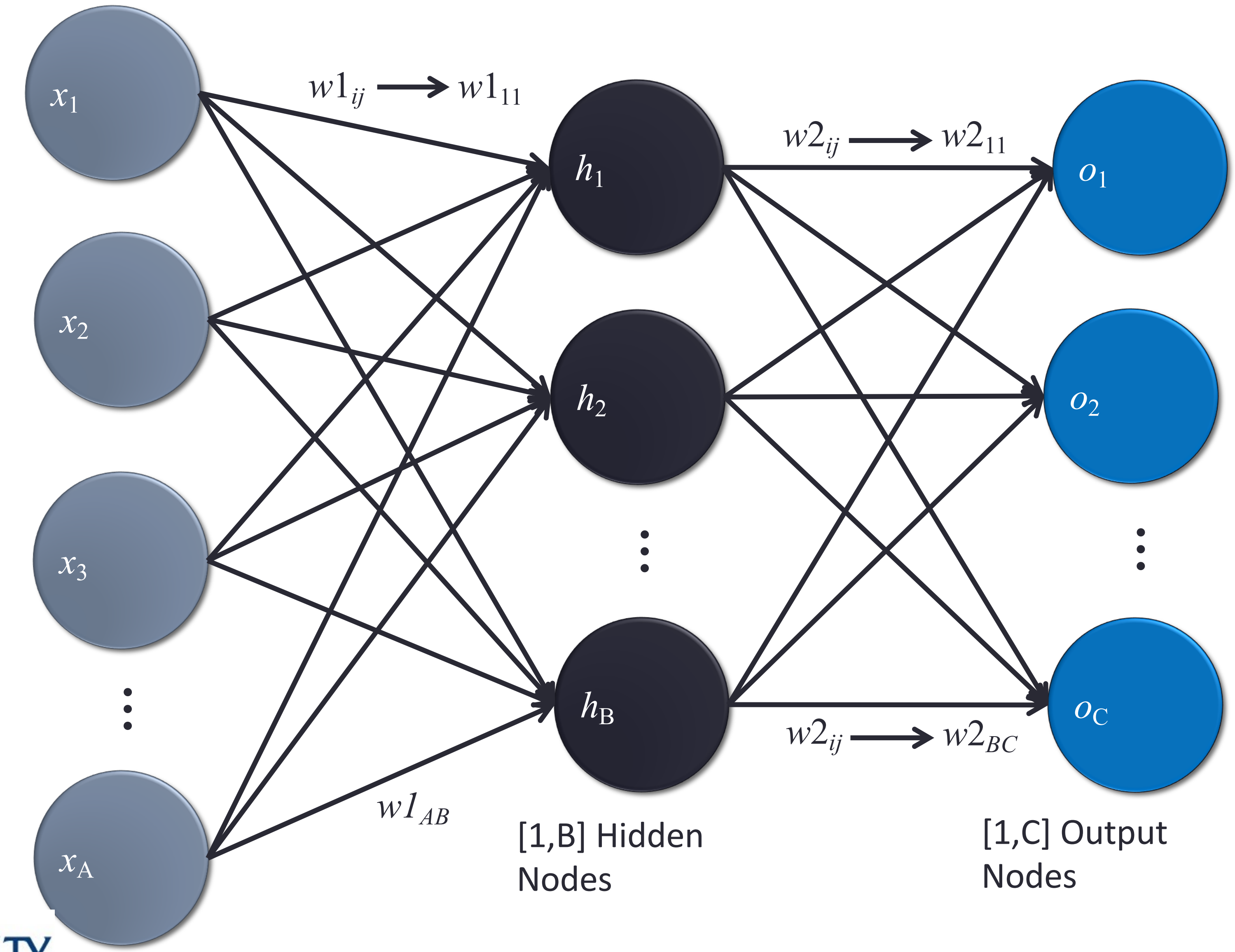
Dendrites: to facilitate the connection with other neurons

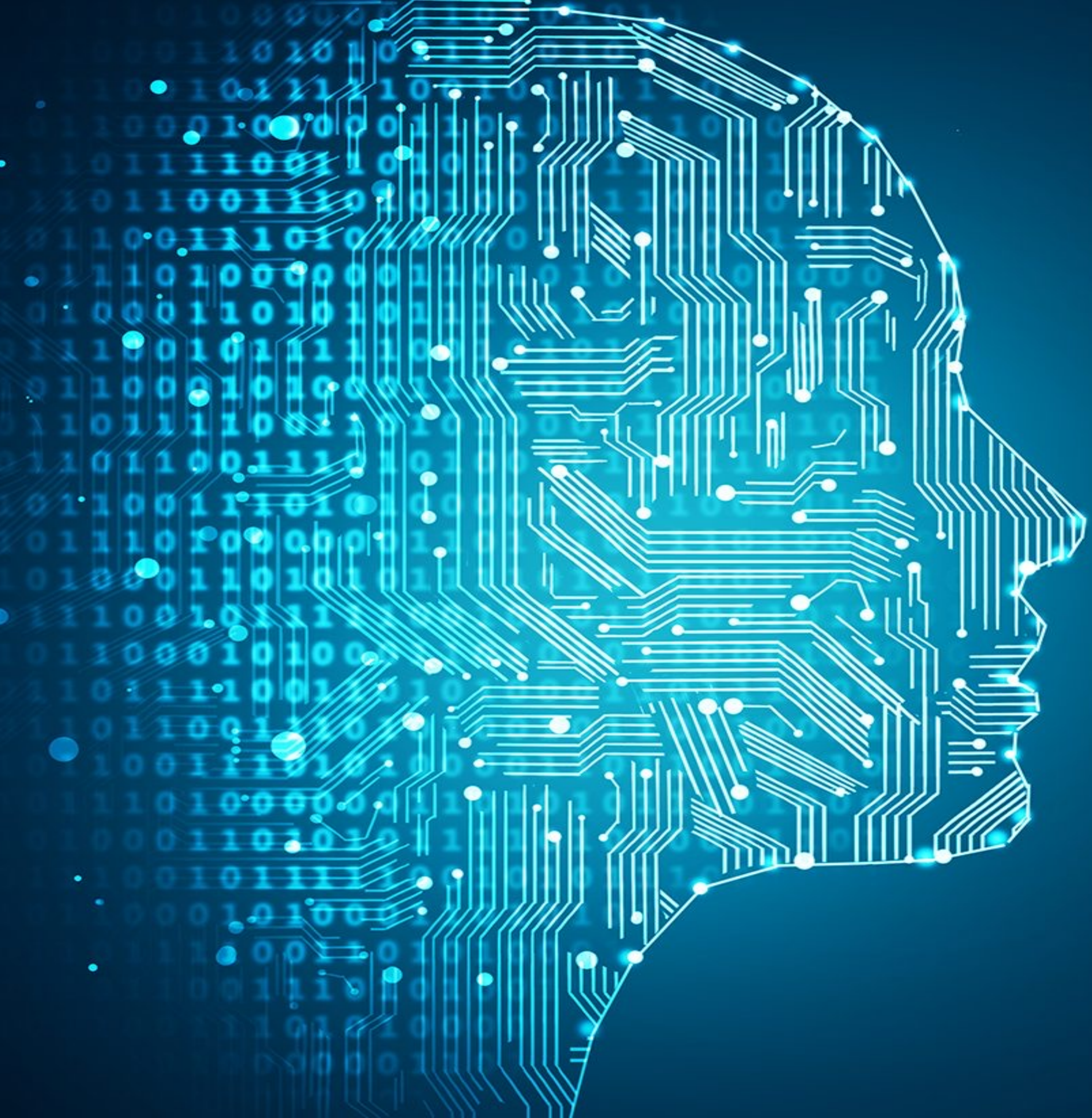
Cell Body

Nucleus

Figure: Structure of a typical neuron

Neural Network

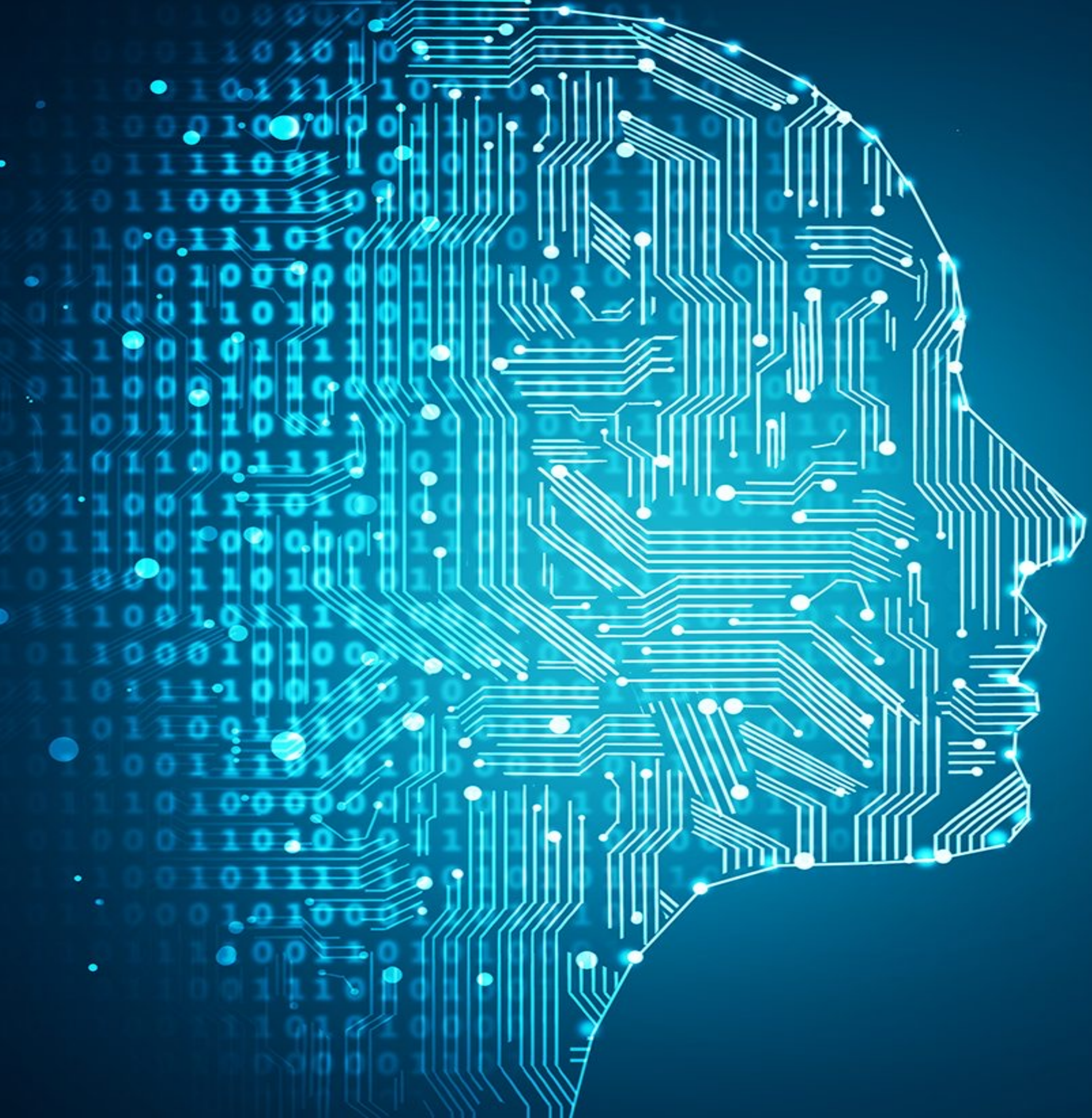




What is training?

The process used to create our ML model.

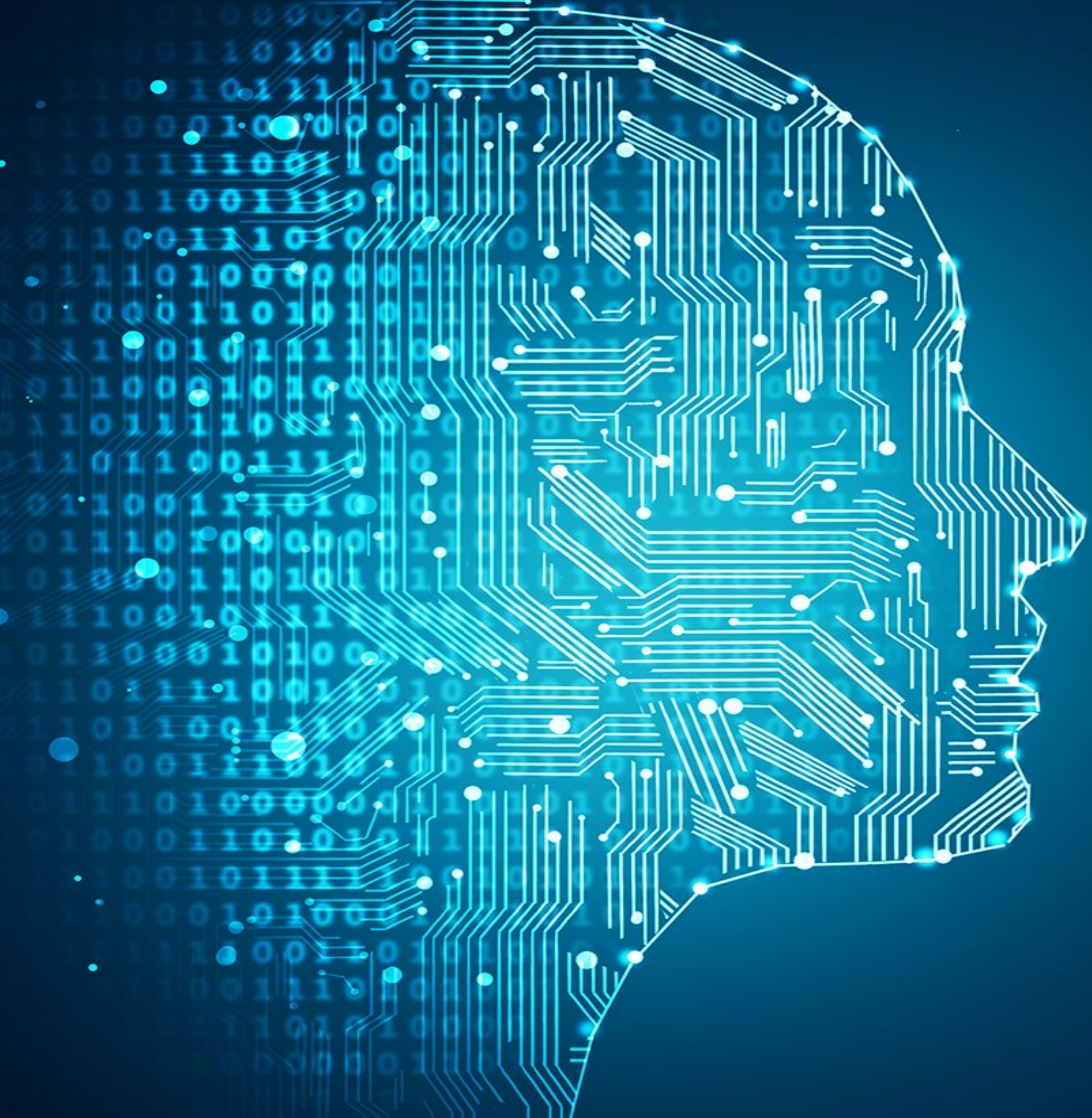
Find a set of weights and biases that have high accuracy.



What is testing?

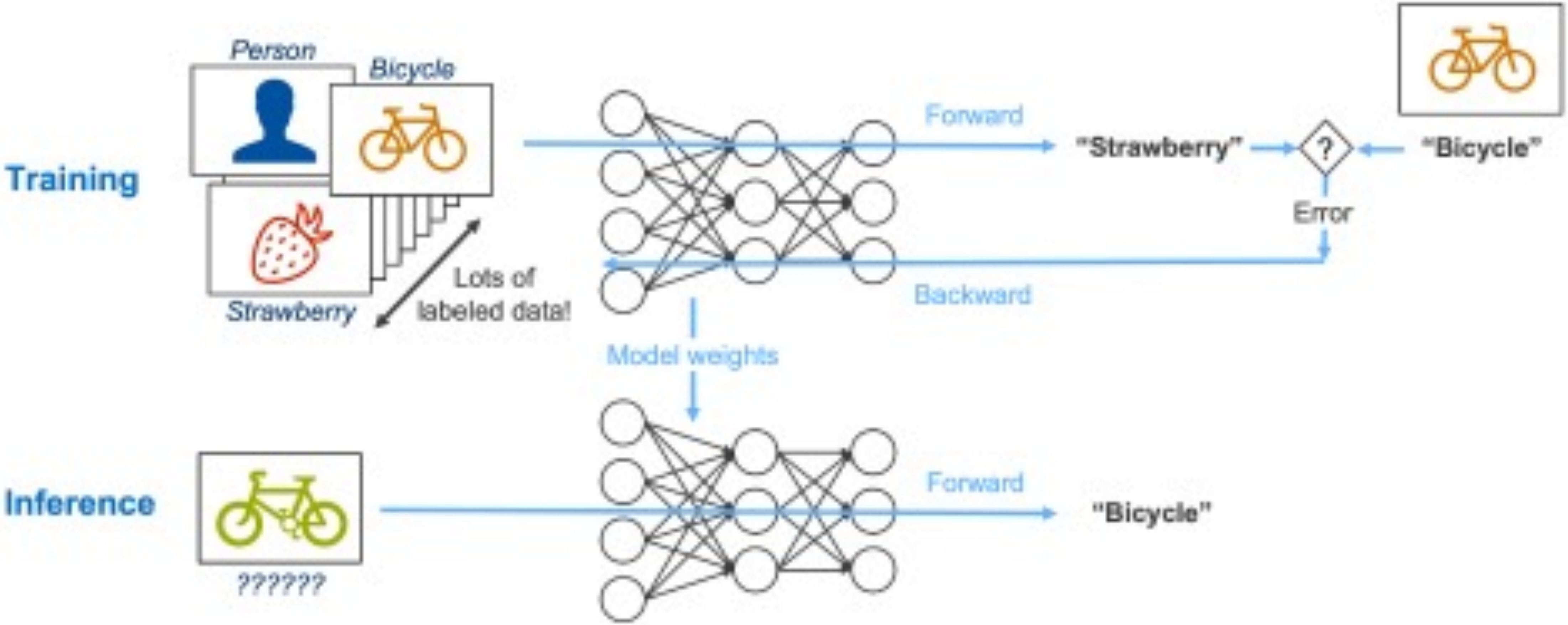
The process used to test our ML model.

Run the model against known outcomes



What is inference?

Running our model on live data to produce actionable output.


Training
Person
Bicycle
Strawberry
Lots of labeled data!
Forward
"Strawberry"
?
"Bicycle"
Error
Backward
Model weights
Inference
??????
Forward
"Bicycle"

Common types of Learning

Supervised learning

We have labelled data, and we want to make some prediction

- Regression
- Classification

Unsupervised learning

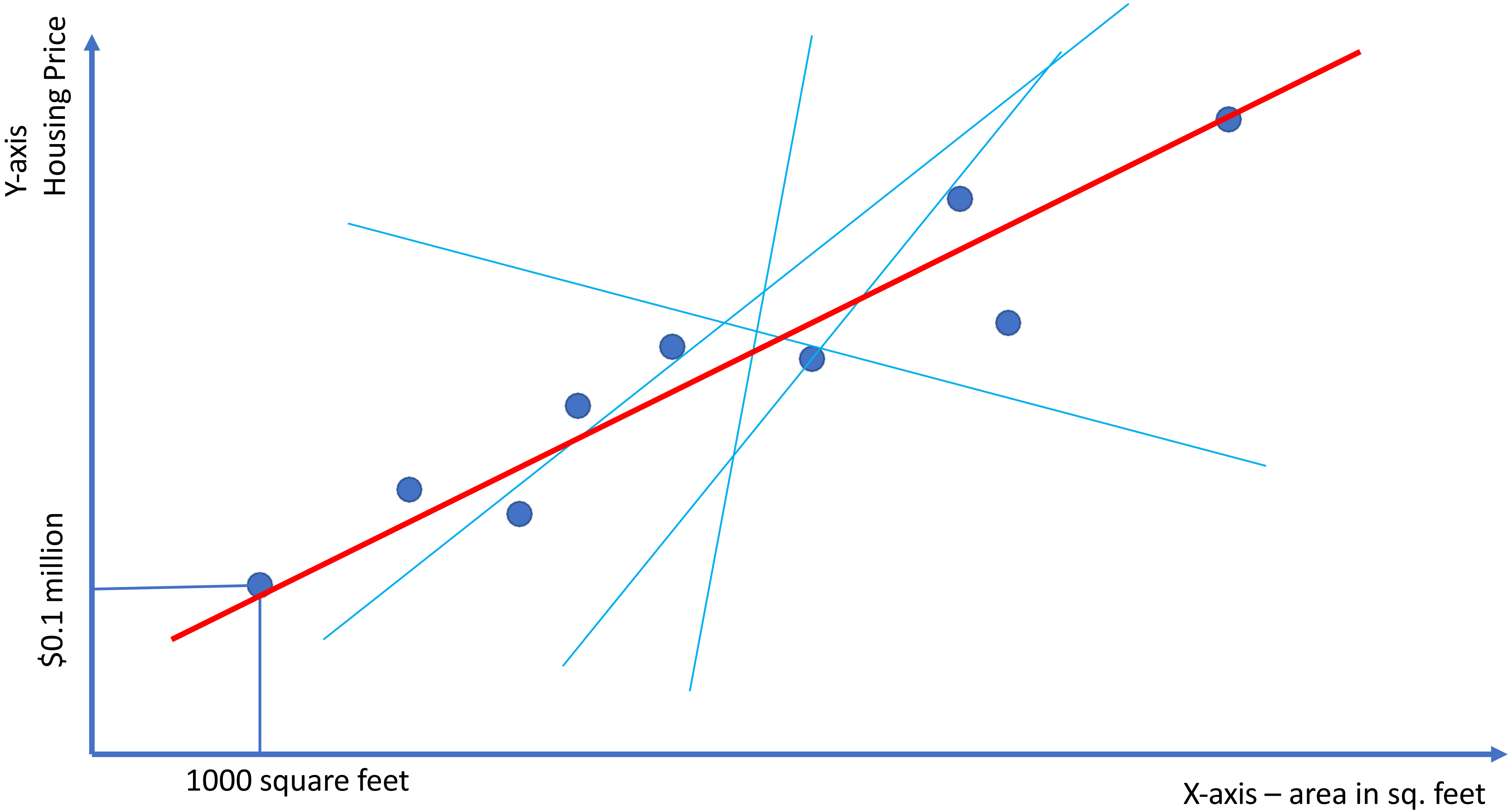
We have unlabeled data, and we want to make some prediction

- Clustering

Supervised learning

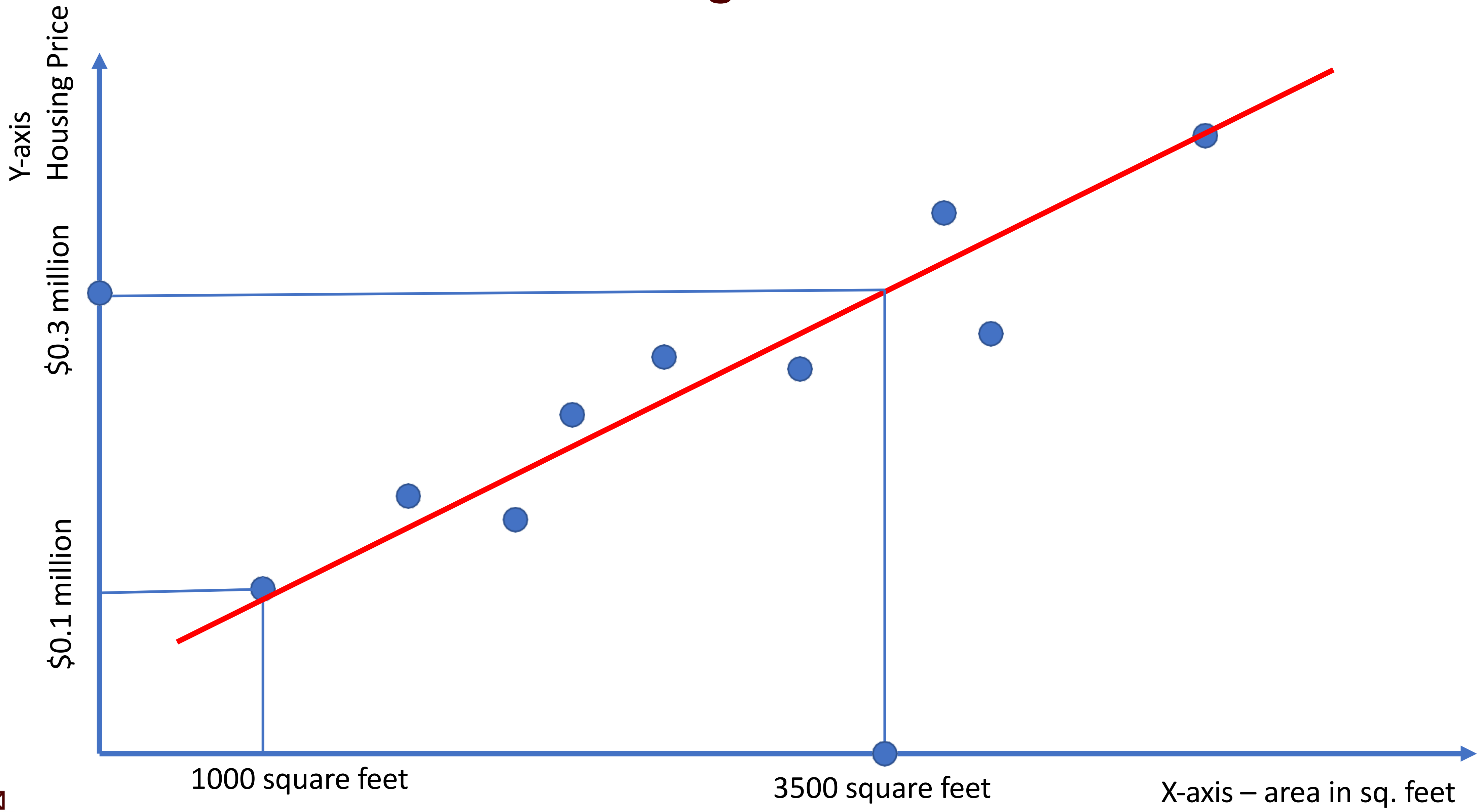
Regression

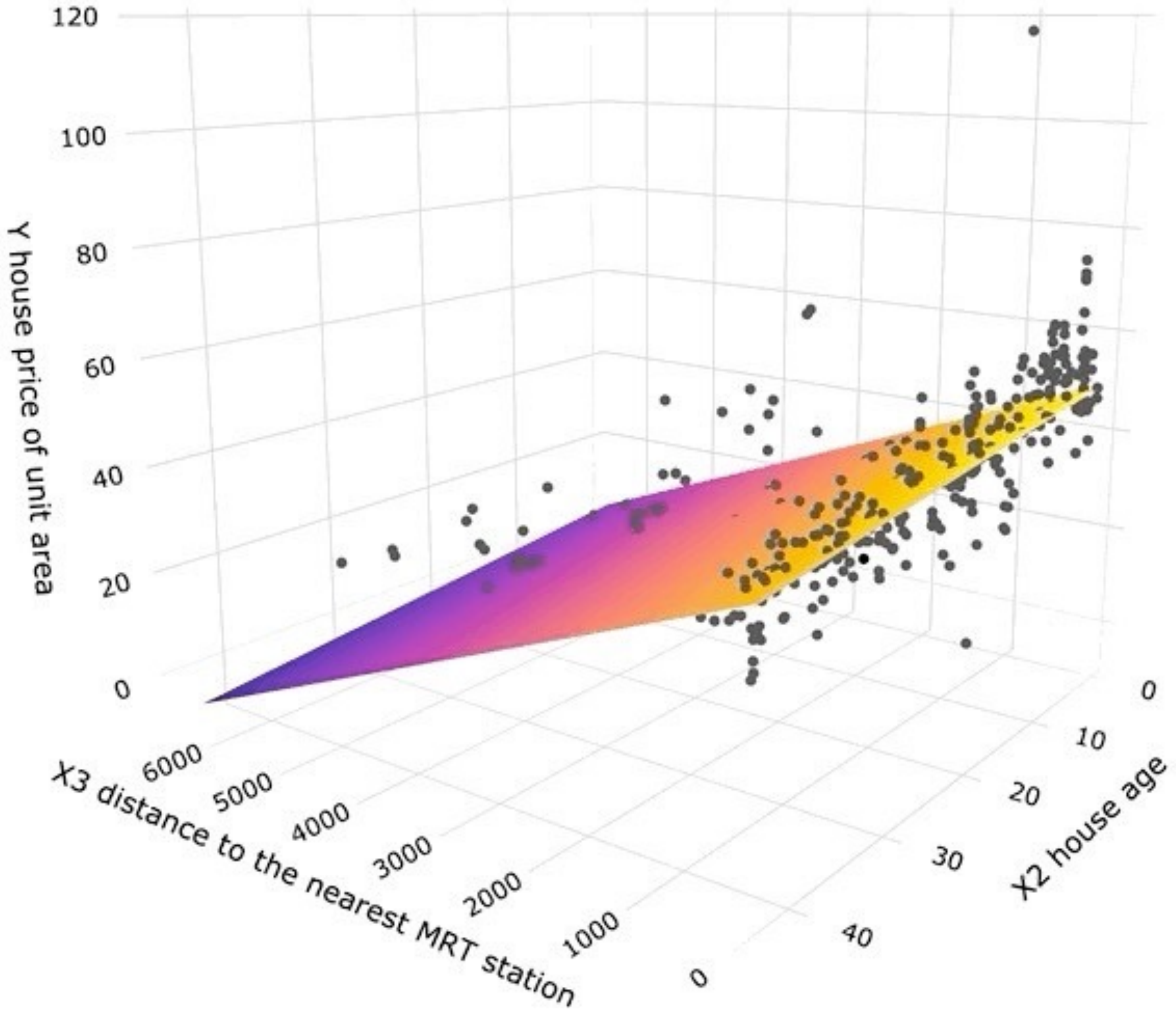
Regression



Y-axis
Housing Price
$0.1 million
1000 square feet
X-axis – area in sq. feet

Regression


Y-axis
Housing Price
$0.3 million
$0.1 million
1000 square feet
3500 square feet
X-axis – area in sq. feet

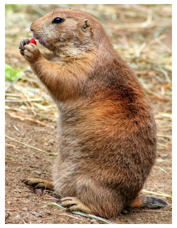

Y house price of unit area
120
100
80
60
40
20
0
X3 distance to the nearest MRT station
6000
5000
4000
3000
2000
1000
0
X2 house age
0
10
20
30
40

Quiz

- Which of the following CANNOT be an example of regression?
  - A) Using past data of weather in college station to predict future's weather.
  - B) Predicting prices of stocks using previous month's price data
  - C) Determining if an email is spam or not
  - D) Determining network traffic for today using previous month's data

Classification

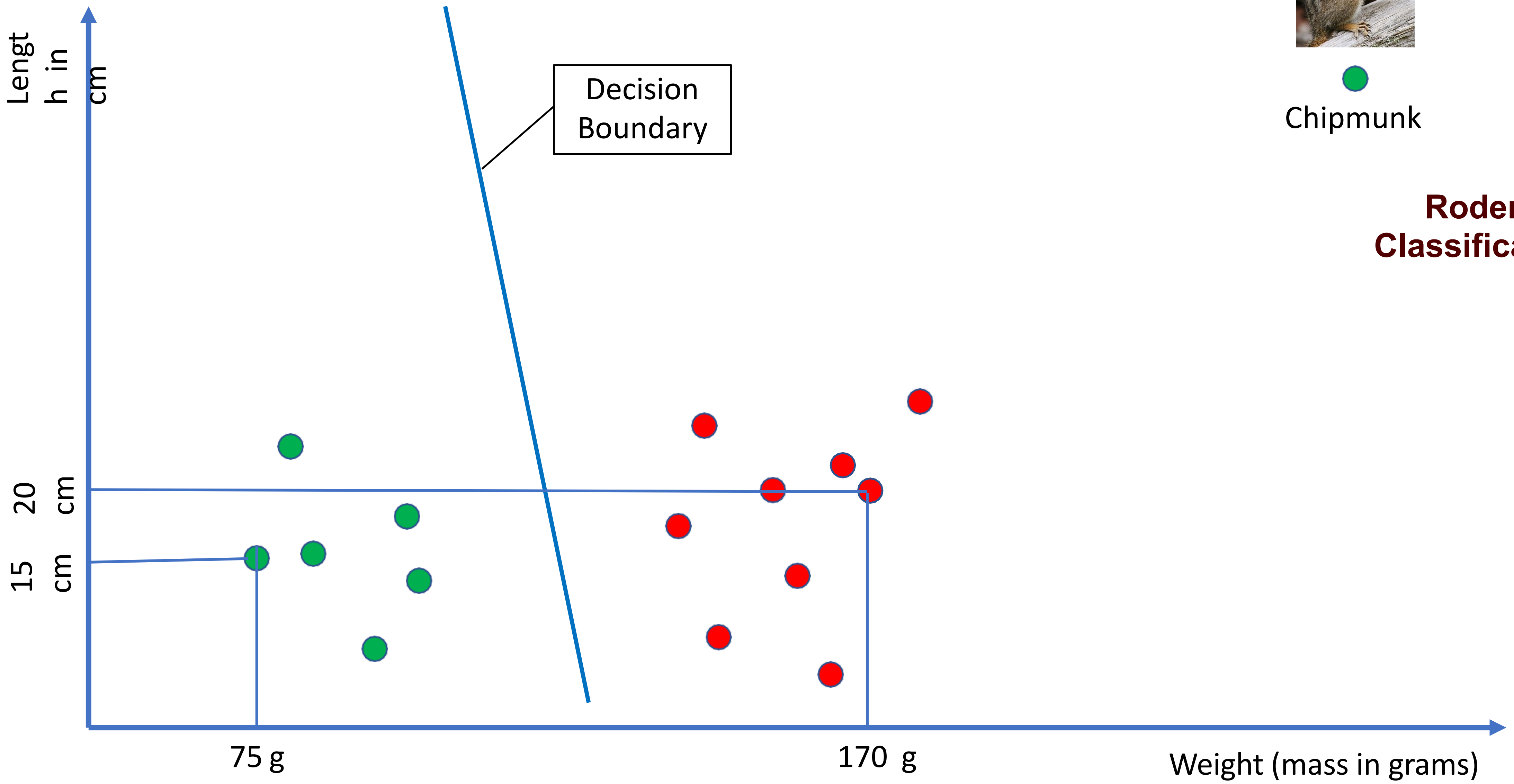
Classification



● Chipmunk

● Gopher

Rodent Classification



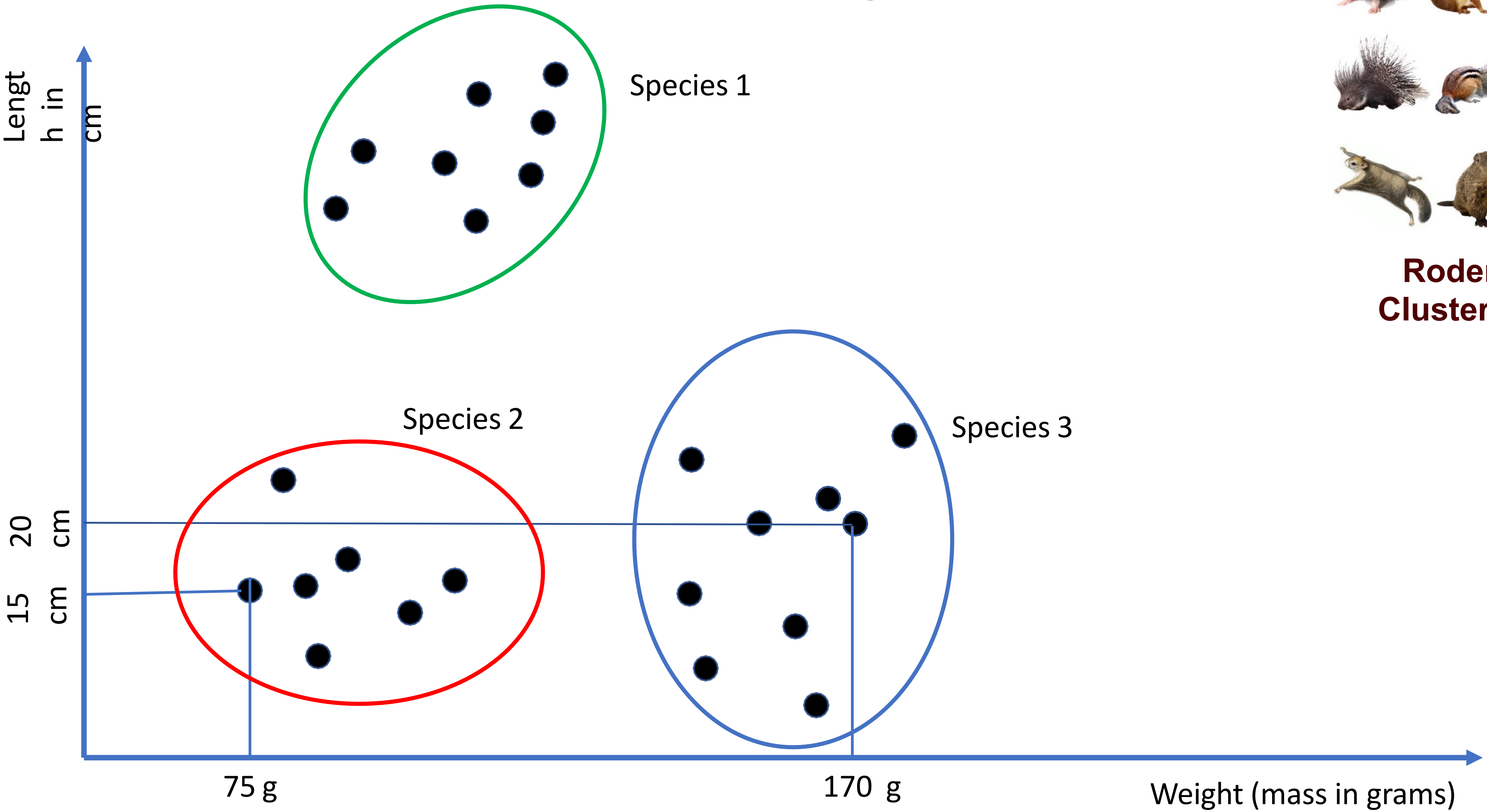
Quiz

- Which of the following CANNOT be an example of classification?
  - A) Using blood pressure and weight data to determine if a patient is diabetic or not
  - B) Estimating amount of annual rain from previous year's data
  - C) Classifying Pokémon in different types (e.g., fire, ice, poison, electric)
  - D) Determining if an email is spam or not

Unsupervised learning

clustering

Clustering



Rodent Clustering

Quiz

- Which of the following CANNOT be an example of clustering?
  - A) Sorting and making groups of research papers having similar content
  - B) Determining whether a news article is about politics or sports
  - C) Identifying clusters of stars having similar characteristics
  - D) Sorting through subjects of emails and grouping them accordingly

Quiz

- Which of the following CANNOT be an example of machine learning? Select all that apply.
  - A) Manually trying out different passwords on your amazon account to check if it works
  - B) Your virtual assistant starts recognizing your voice after first few tries
  - C) Fire alarm goes off when smoke level is more than a specific level
  - D) Sorting through subjects of emails and grouping them accordingly

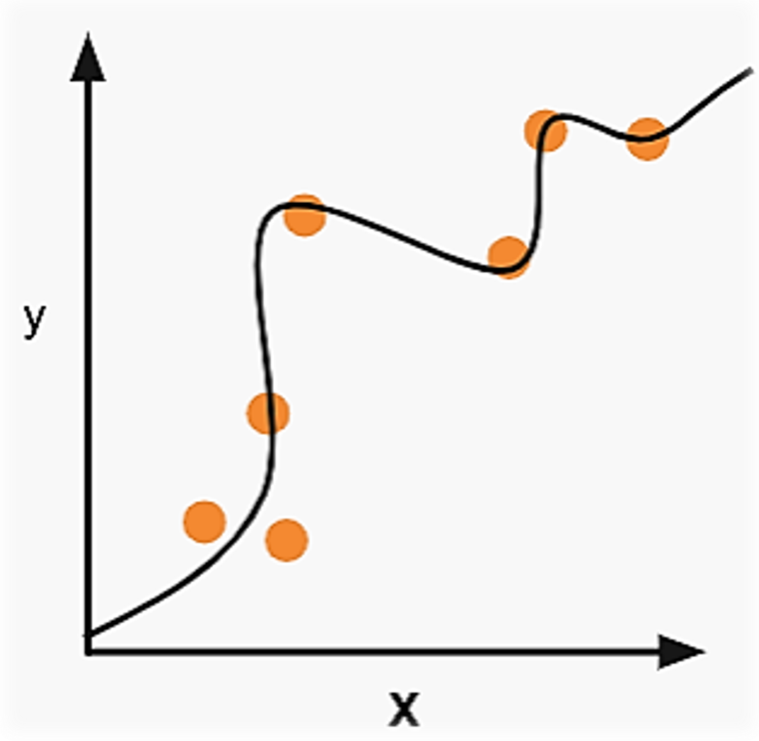
What is Artificial Intelligence

- **Wikipedia**: intelligence demonstrated by machines as opposed to natural intelligence displayed by animals including humans.

- **Oxford**: the theory and development of computer systems able to perform tasks that normally require human intelligence.

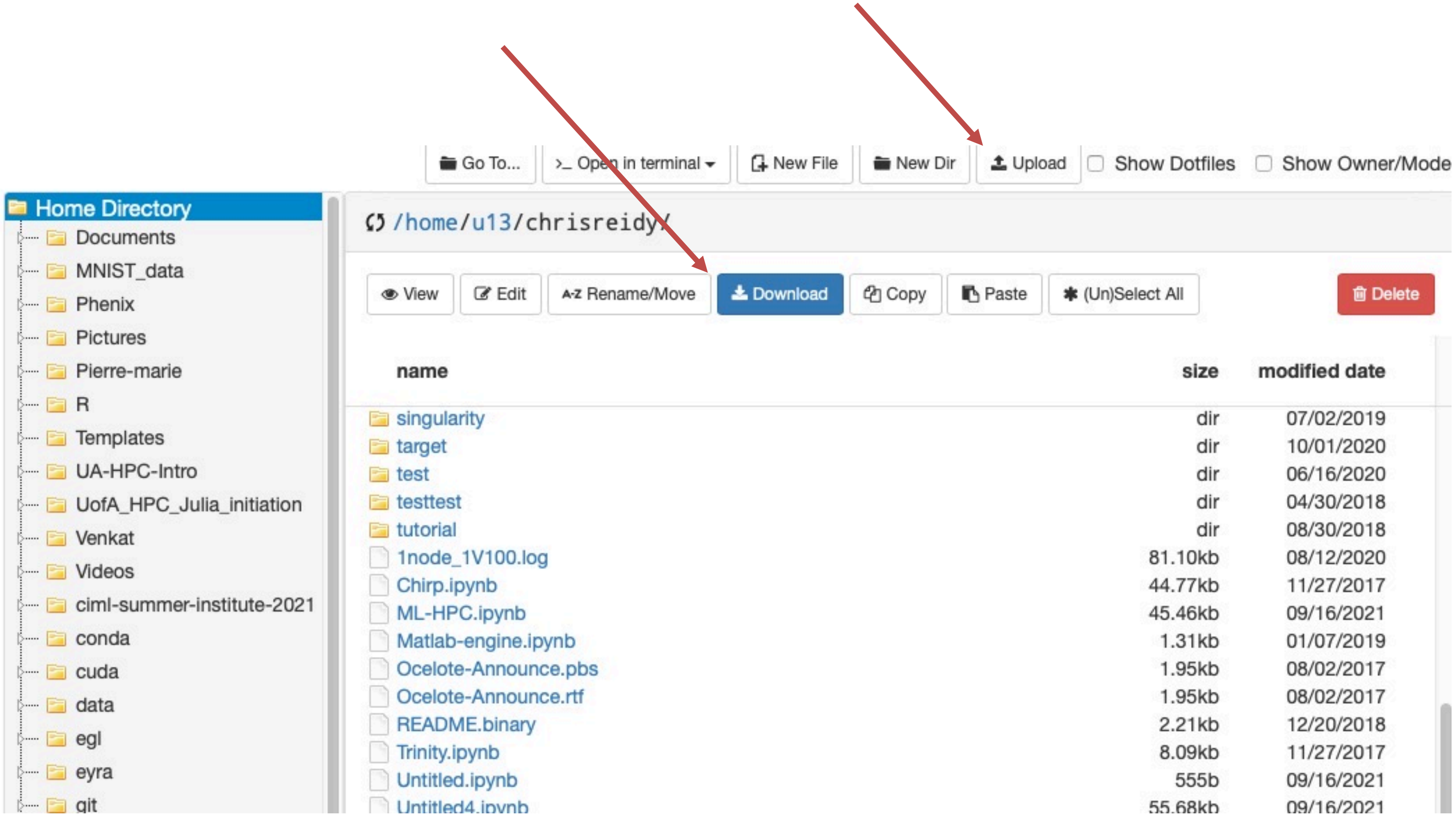
- **IBM**: leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind

Train a linear regression model



Jupyter Notebooks on UArizona HPC with Python.

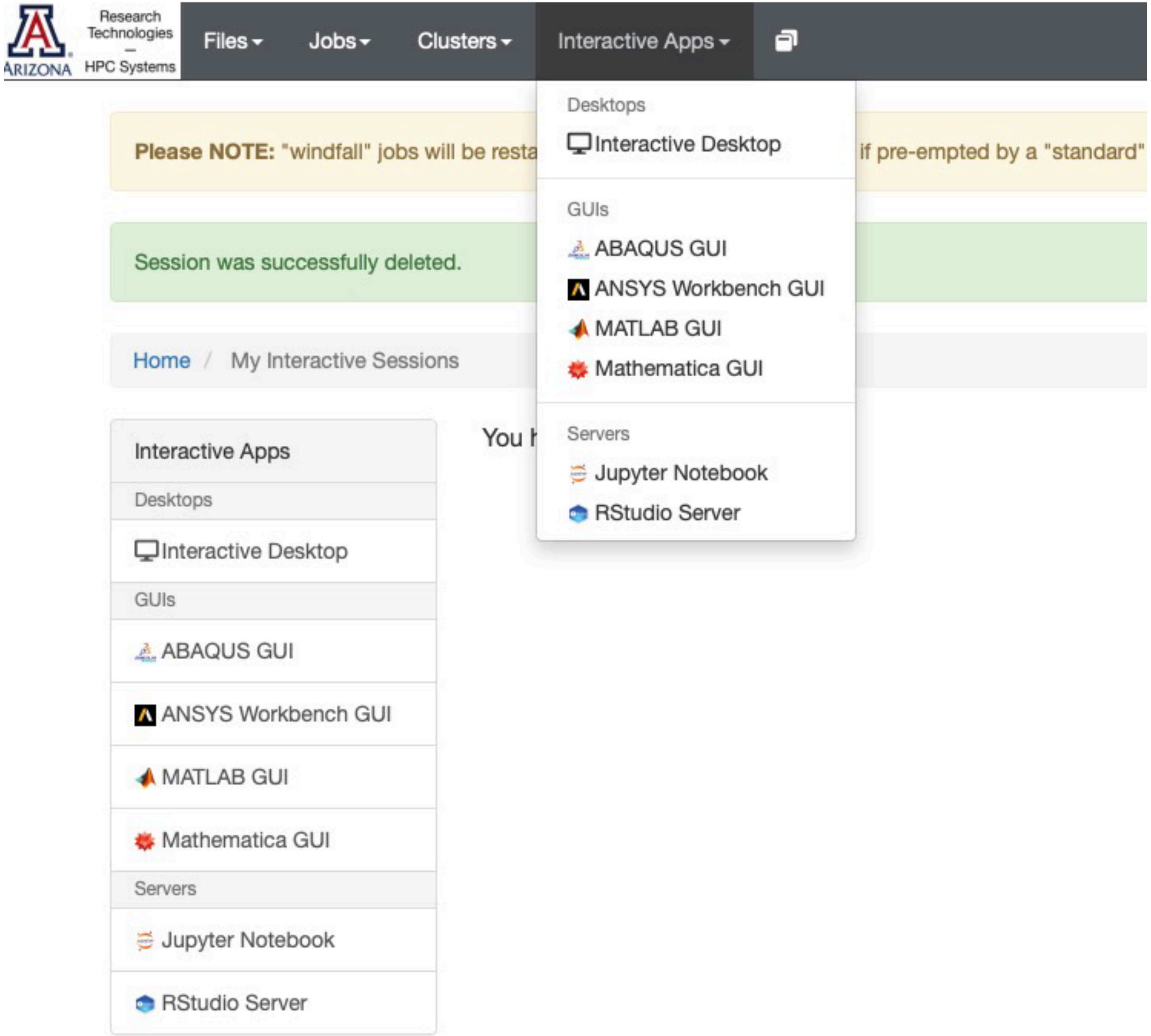
ood.hpc.arizona.edu



/home/u13/chrisreidy/

name	size	modified date
singularity	dir	07/02/2019
target	dir	10/01/2020
test	dir	06/16/2020
testtest	dir	04/30/2018
tutorial	dir	08/30/2018
1node_1V100.log	81.10kb	08/12/2020
Chirp.ipynb	44.77kb	11/27/2017
ML-HPC.ipynb	45.46kb	09/16/2021
Matlab-engine.ipynb	1.31kb	01/07/2019
Ocelote-Announce.pbs	1.95kb	08/02/2017
Ocelote-Announce.rtf	1.95kb	08/02/2017
README.binary	2.21kb	12/20/2018
Trinity.ipynb	8.09kb	11/27/2017
Untitled.ipynb	555b	09/16/2021
Untitled4.ipynb	55.68kb	09/16/2021

Jupyter Notebooks on UArizona HPC with Python



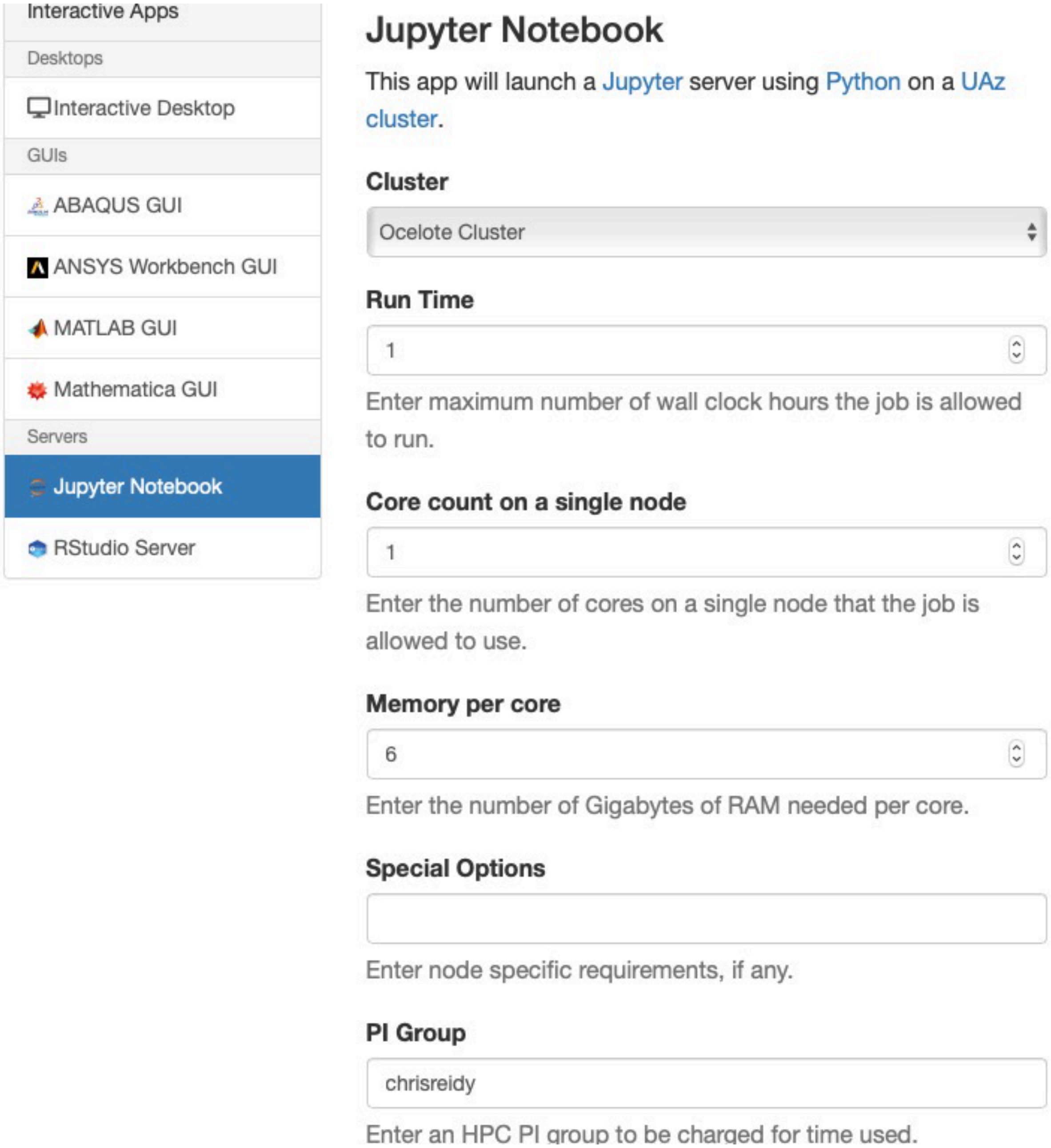
Please NOTE: "windfall" jobs will be resta if pre-empted by a "standard"

Session was successfully deleted.

Home / My Interactive Sessions

You h

Jupyter Notebooks on UArizona HPC with Python



Jupyter Notebook

This app will launch a Jupyter server using Python on a UAz cluster.

Cluster

Ocelote Cluster

Run Time

1

Enter maximum number of wall clock hours the job is allowed to run.

Core count on a single node

1

Enter the number of cores on a single node that the job is allowed to use.

Memory per core

6

Enter the number of Gigabytes of RAM needed per core.

Special Options

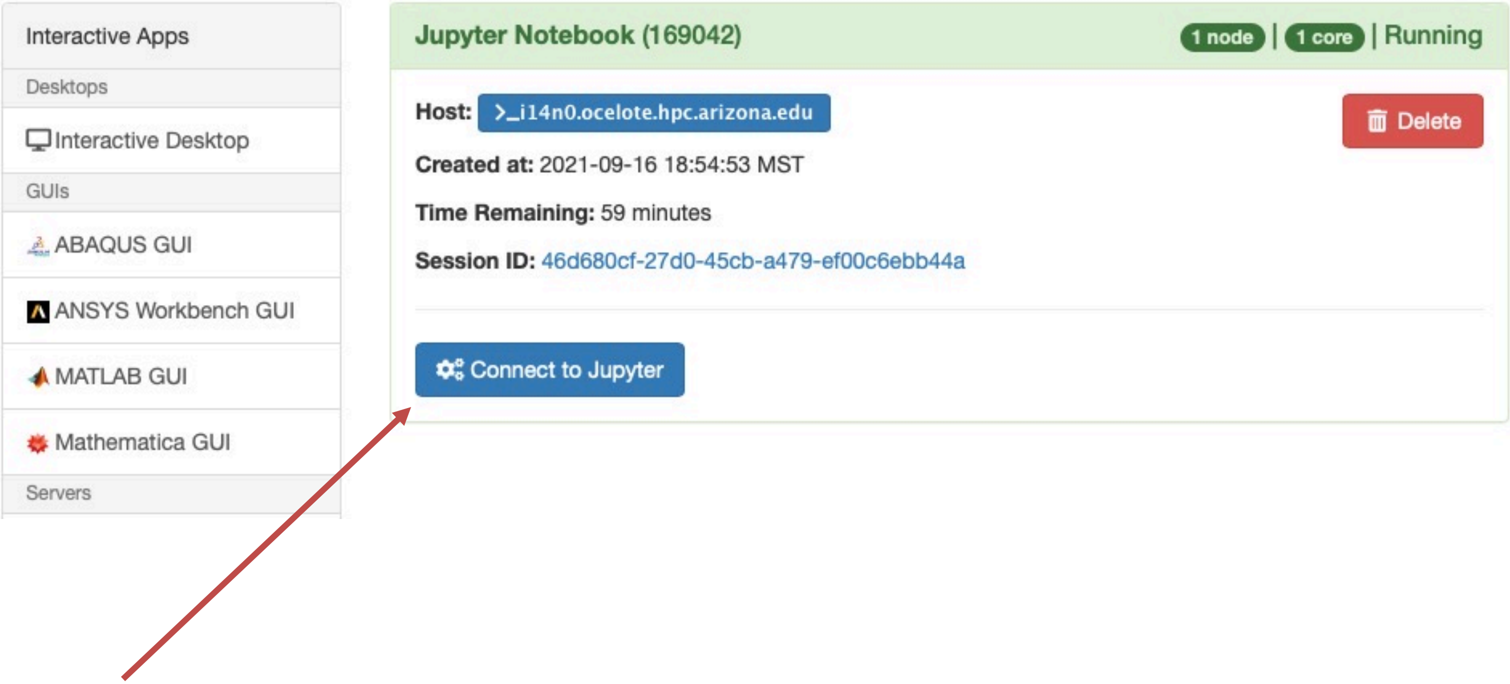
Enter node specific requirements, if any.

PI Group

chrisreidy

Enter an HPC PI group to be charged for time used.

Jupyter Notebooks on UArizona HPC with Python



Jupyter Notebook (169042) | Running

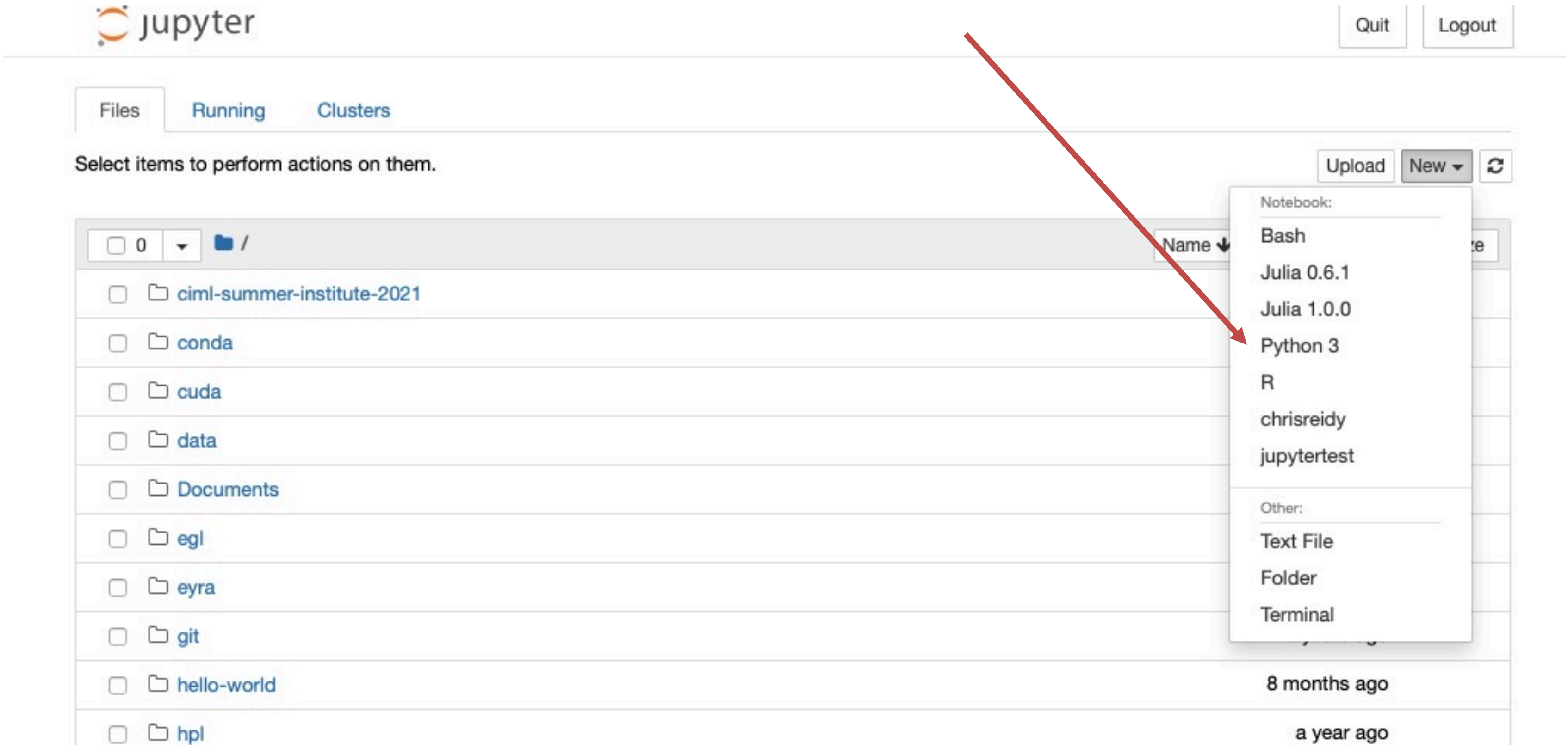
Host: >_i14n0.ocelote.hpc.arizona.edu

Created at: 2021-09-16 18:54:53 MST

Time Remaining: 59 minutes

Session ID: 46d680cf-27d0-45cb-a479-ef00c6ebb44a

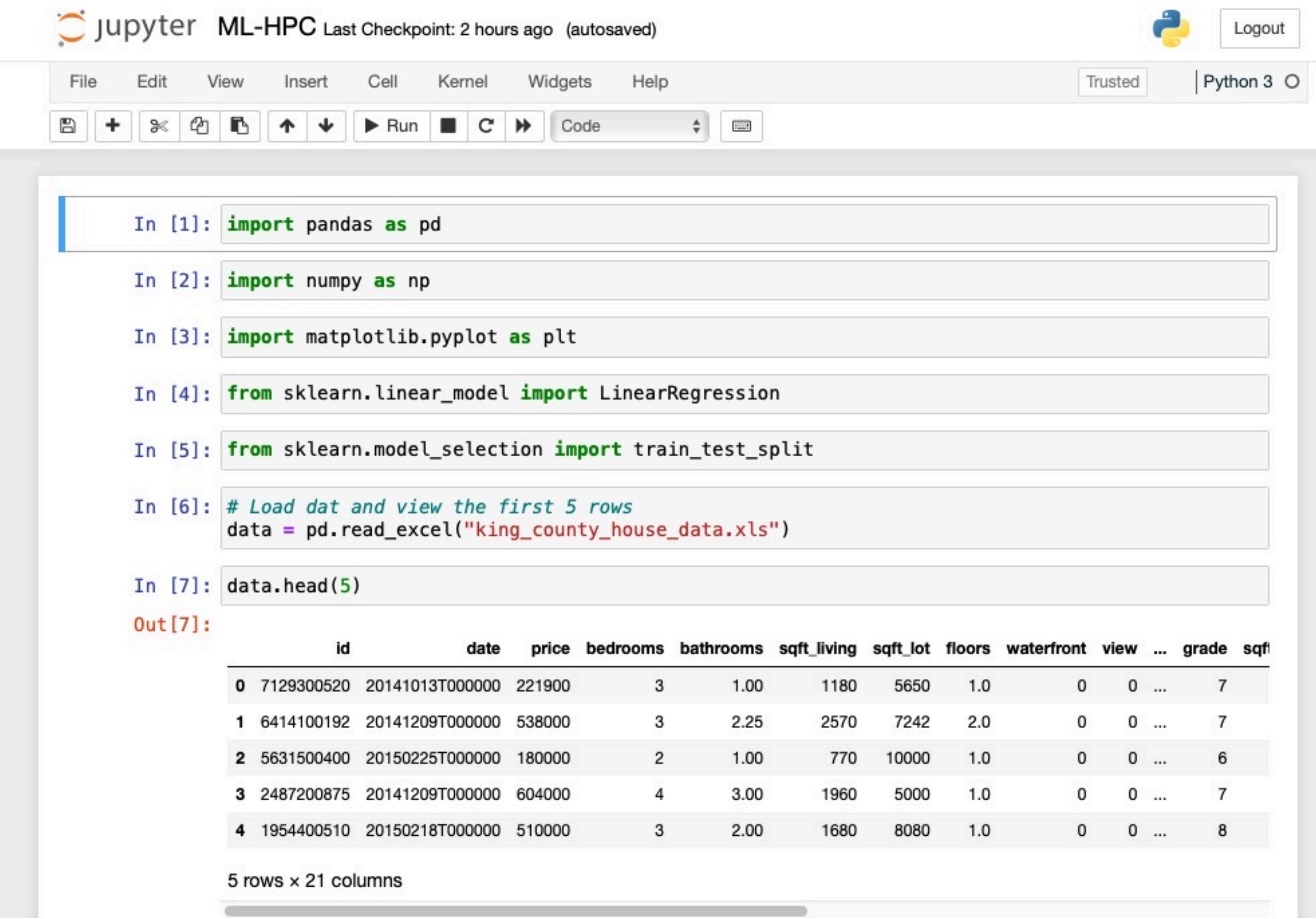
Jupyter Notebooks on UArizona HPC with Python



Select items to perform actions on them.

□ ciml-summer-institute-2021
□ conda
□ cuda
□ data
□ Documents
□ egl
□ eyra
□ git
□ hello-world 8 months ago
□ hpl a year ago

Jupyter Notebooks on UArizona HPC with Python



```
In [1]: import pandas as pd

In [2]: import numpy as np

In [3]: import matplotlib.pyplot as plt

In [4]: from sklearn.linear_model import LinearRegression

In [5]: from sklearn.model_selection import train_test_split

In [6]: # Load dat and view the first 5 rows
        data = pd.read_excel("king_county_house_data.xls")

In [7]: data.head(5)
```

Out[7]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0	0	0	...	7	
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0	0	0	...	7	
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0	0	0	...	6	
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0	0	0	...	7	
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0	0	0	...	8	

5 rows × 21 columns

Jupyter Notebooks on HPC

ood.hpc.arizona.edu
ocelote / 2 hours / 1 core / 6 mem / standard queue / chrisreidy

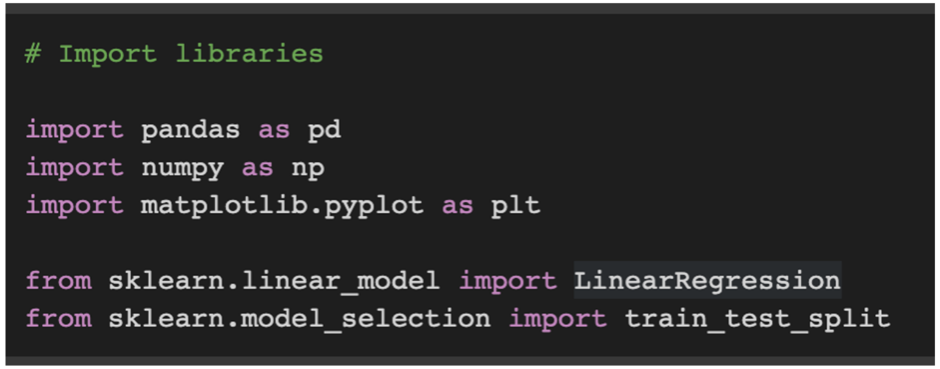
Accessing files for the exercises

ssh netid@hpc.arizona.edu
shell
ocelote
mkdir intro-to-hpc
cd intro-to-hpc

https://ua-researchcomputing-hpc.github.io/Intro-to-HPC/
Then Accessing Workshop Files and cut / paste the section starting “wget”
(old method: cp /xdisk/chrisreidy/workshops/* .)
Choice #1: Cut and paste commands into Jupyter from .txt file
Choice #2: Run the Notebook .ipynb file
Choice #3: Type in the commands. Syntax is very important

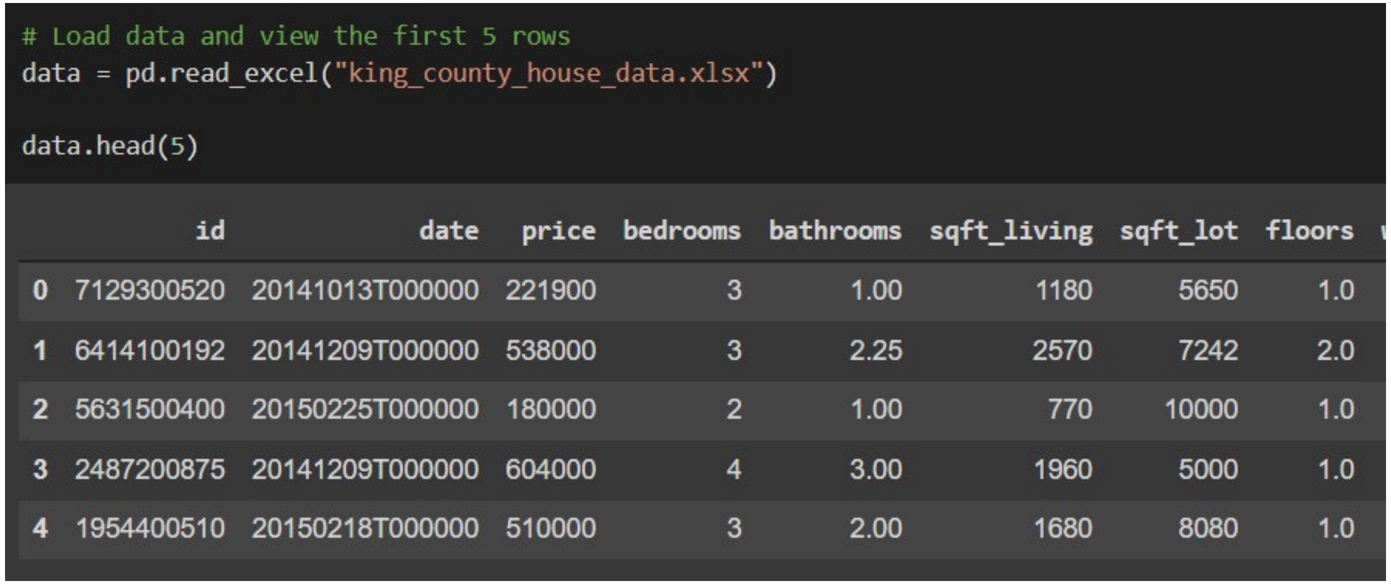
Train a linear regression model

- Import libraries



```python
# Import libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

- Use Pandas to load the data and view the first 5 rows

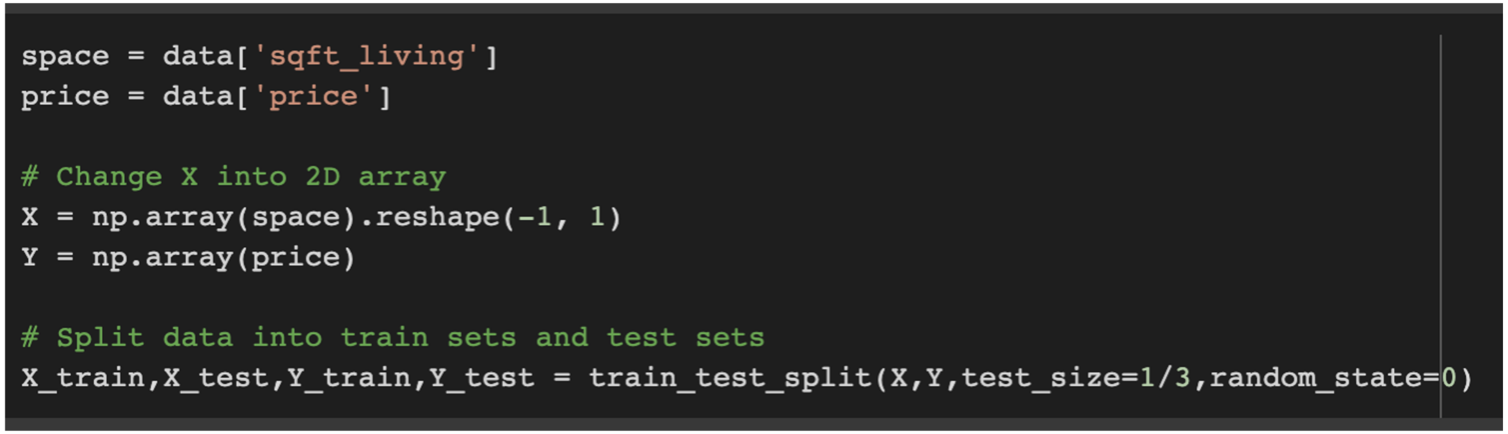


```python
# Load data and view the first 5 rows
data = pd.read_excel("king_county_house_data.xlsx")

data.head(5)
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0

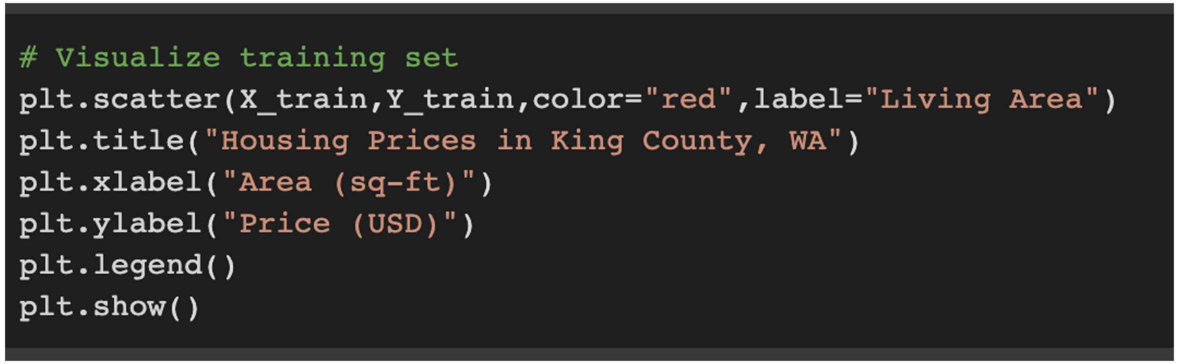
Train a linear regression model

- Choose the columns from the data
- Split the data into train and test sets

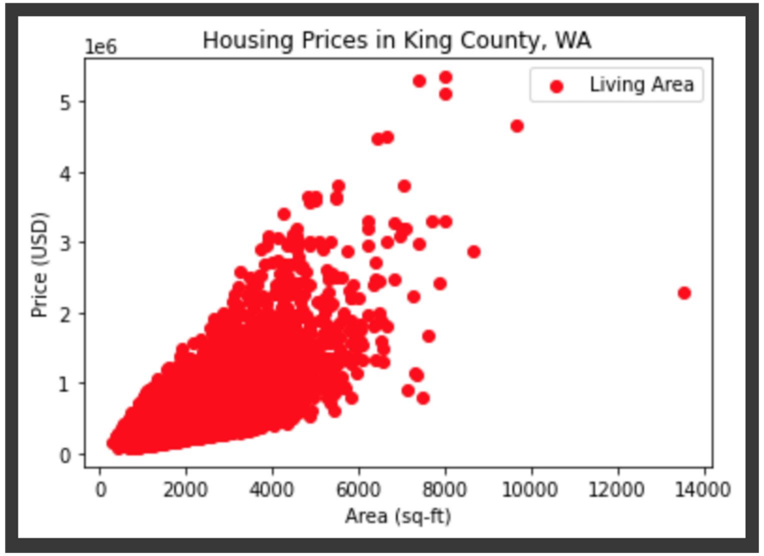


```
space = data['sqft_living']
price = data['price']

# Change X into 2D array
X = np.array(space).reshape(-1, 1)
Y = np.array(price)

# Split data into train sets and test sets
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=1/3,random_state=0)
```

- Visualize the train set

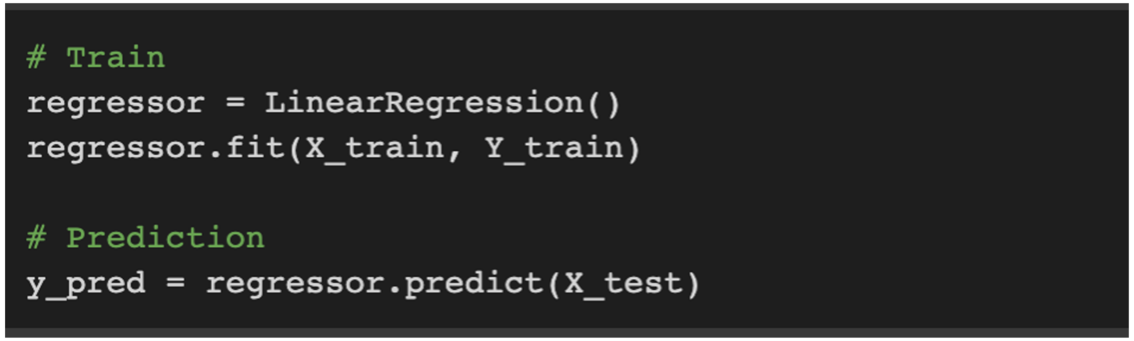


```
# Visualize training set
plt.scatter(X_train,Y_train,color="red",label="Living Area")
plt.title("Housing Prices in King County, WA")
plt.xlabel("Area (sq-ft)")
plt.ylabel("Price (USD)")
plt.legend()
plt.show()
```



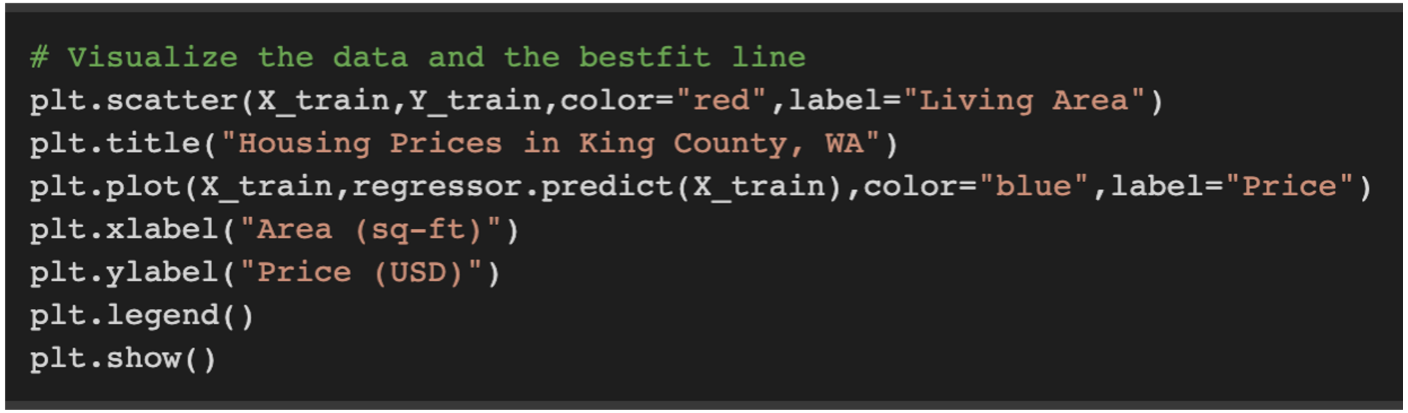
Train a linear regression model

- Train the model with train set
- Predict on test set

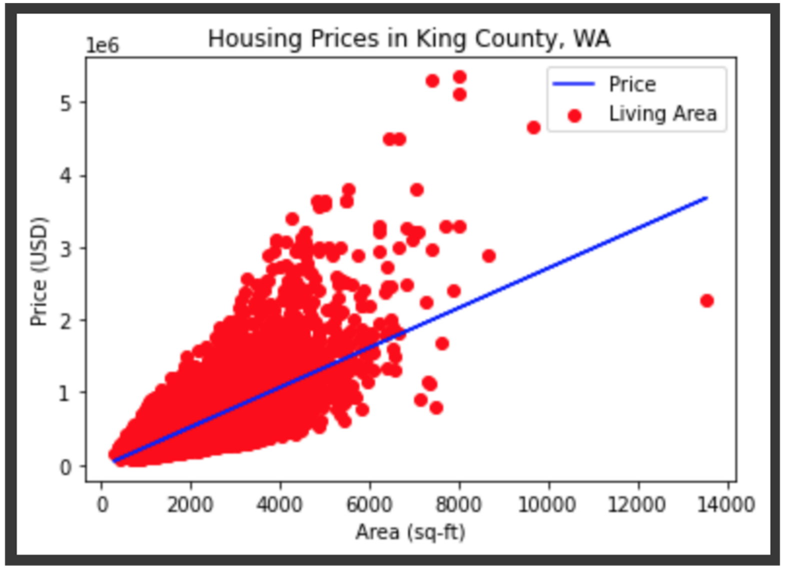


```
# Train
regressor = LinearRegression()
regressor.fit(X_train, Y_train)

# Prediction
y_pred = regressor.predict(X_test)
```

- Visualize the train data and the best fit line

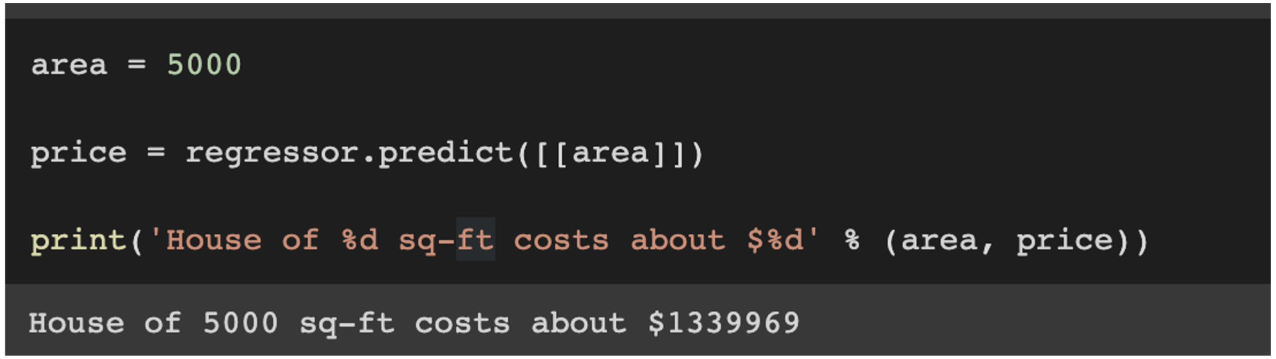


```
# Visualize the data and the bestfit line
plt.scatter(X_train,Y_train,color="red",label="Living Area")
plt.title("Housing Prices in King County, WA")
plt.plot(X_train,regressor.predict(X_train),color="blue",label="Price")
plt.xlabel("Area (sq-ft)")
plt.ylabel("Price (USD)")
plt.legend()
plt.show()
```



Train a linear regression model

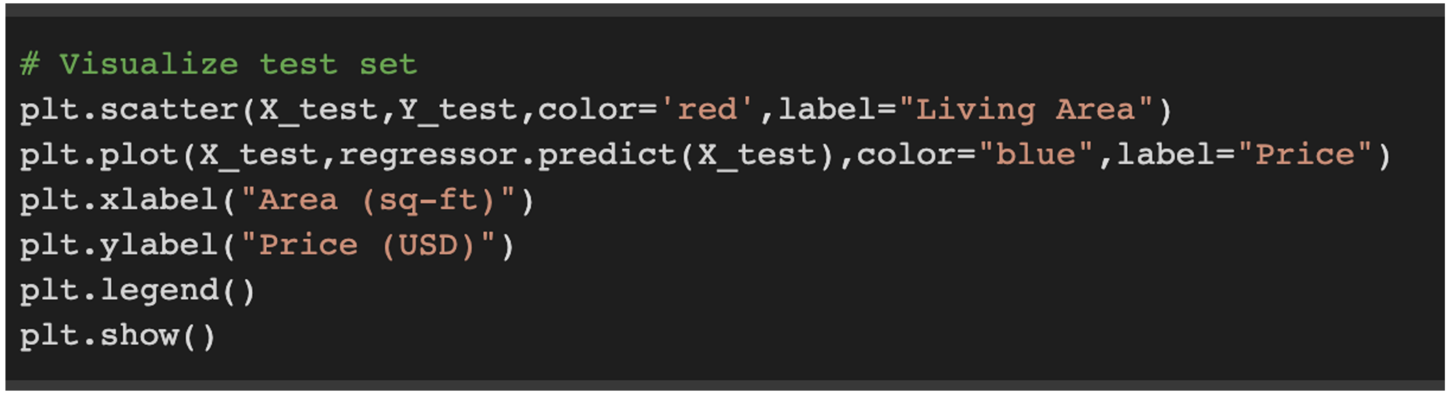
- Predict the price of a house with a certain area



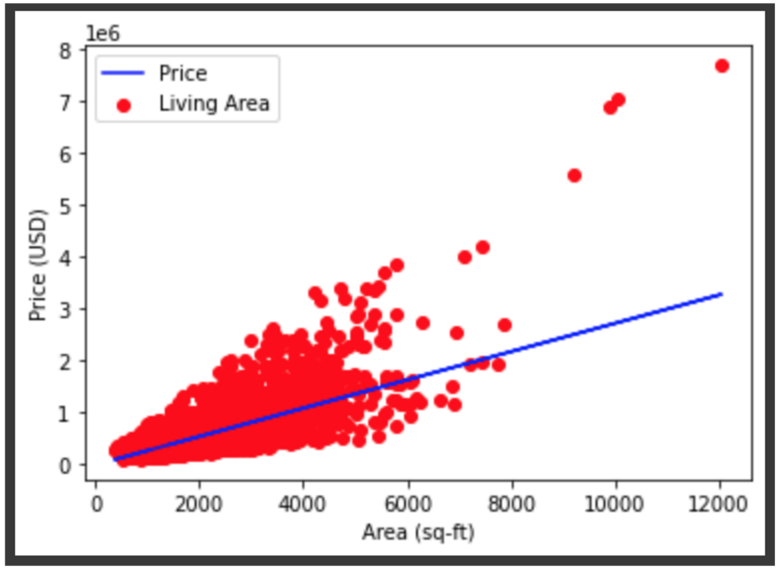
```
area = 5000

price = regressor.predict([[area]])

print('House of %d sq-ft costs about $%d' % (area, price))
```

```
House of 5000 sq-ft costs about $1339969
```

- Visualize the test data

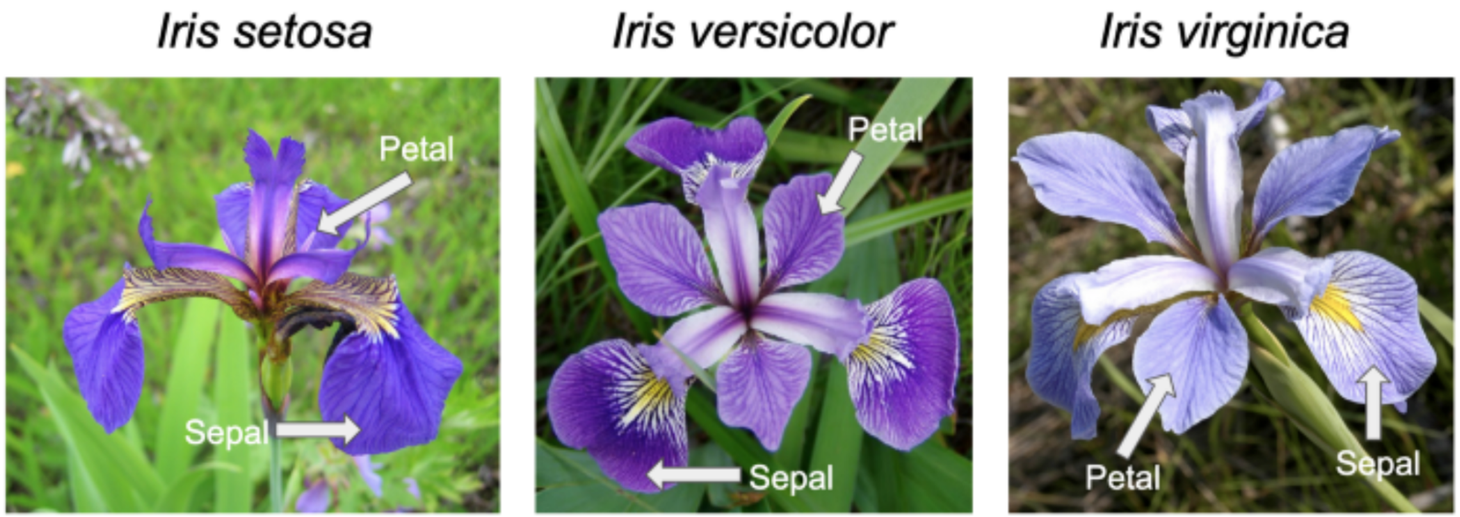


```
# Visualize test set
plt.scatter(X_test,Y_test,color='red',label="Living Area")
plt.plot(X_test,regressor.predict(X_test),color="blue",label="Price")
plt.xlabel("Area (sq-ft)")
plt.ylabel("Price (USD)")
plt.legend()
plt.show()
```

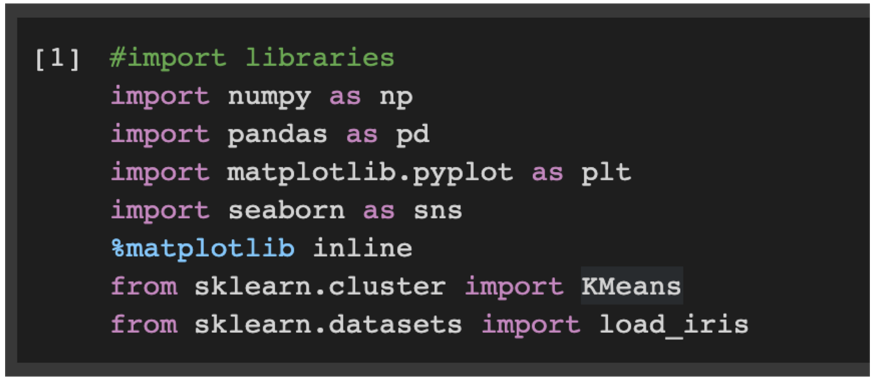


Build a clustering model for Iris Dataset

Build a clustering model – Iris dataset



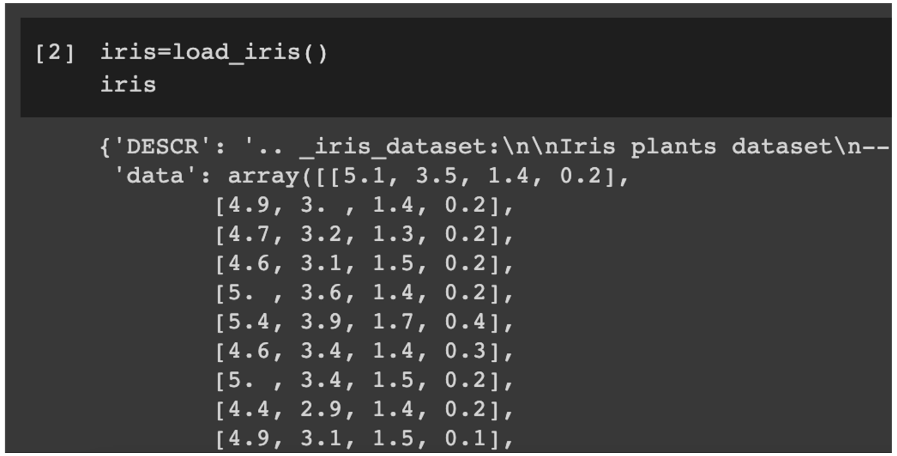
- Import libraries



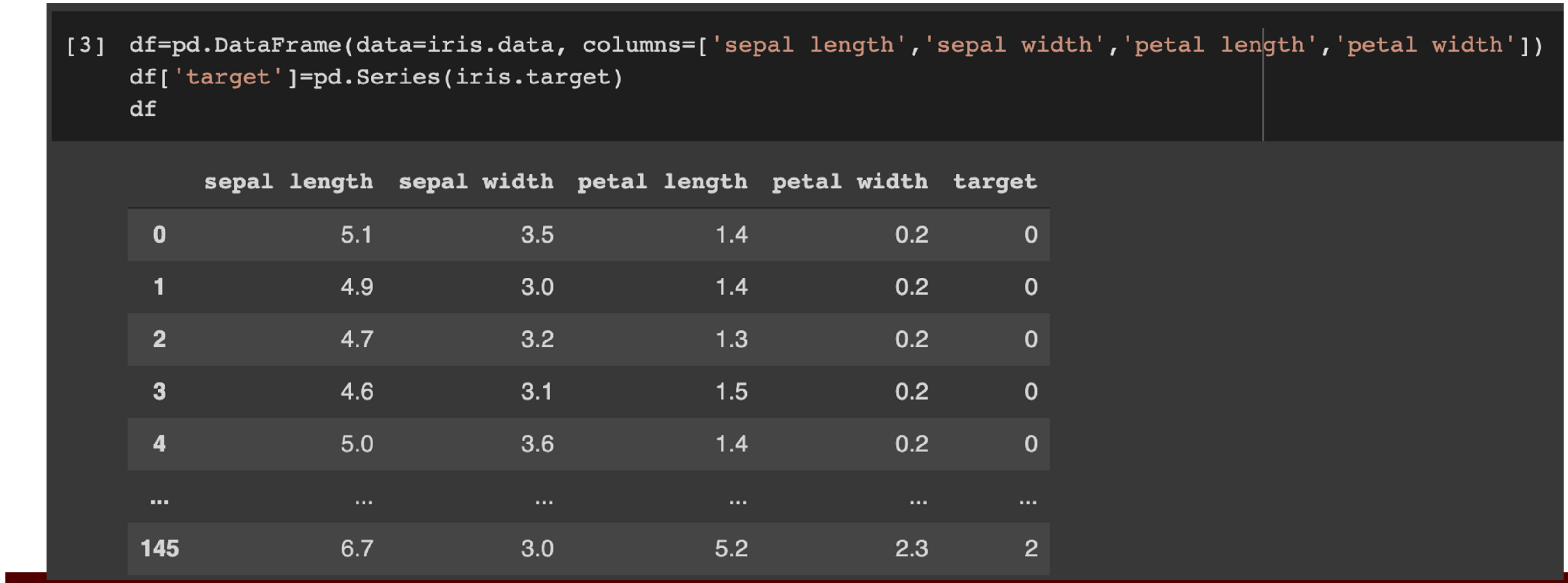
```
[1]  #import libraries
     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     %matplotlib inline
     from sklearn.cluster import KMeans
     from sklearn.datasets import load_iris
```

Build a clustering model – Iris dataset

- Load the data



```
[2]  iris=load_iris()
     iris

     {'DESCR': '.. _iris_dataset:\n\nIris plants dataset\n--
      'data': array([[5.1, 3.5, 1.4, 0.2],
             [4.9, 3. , 1.4, 0.2],
             [4.7, 3.2, 1.3, 0.2],
             [4.6, 3.1, 1.5, 0.2],
             [5. , 3.6, 1.4, 0.2],
             [5.4, 3.9, 1.7, 0.4],
             [4.6, 3.4, 1.4, 0.3],
             [5. , 3.4, 1.5, 0.2],
             [4.4, 2.9, 1.4, 0.2],
             [4.9, 3.1, 1.5, 0.1],
```

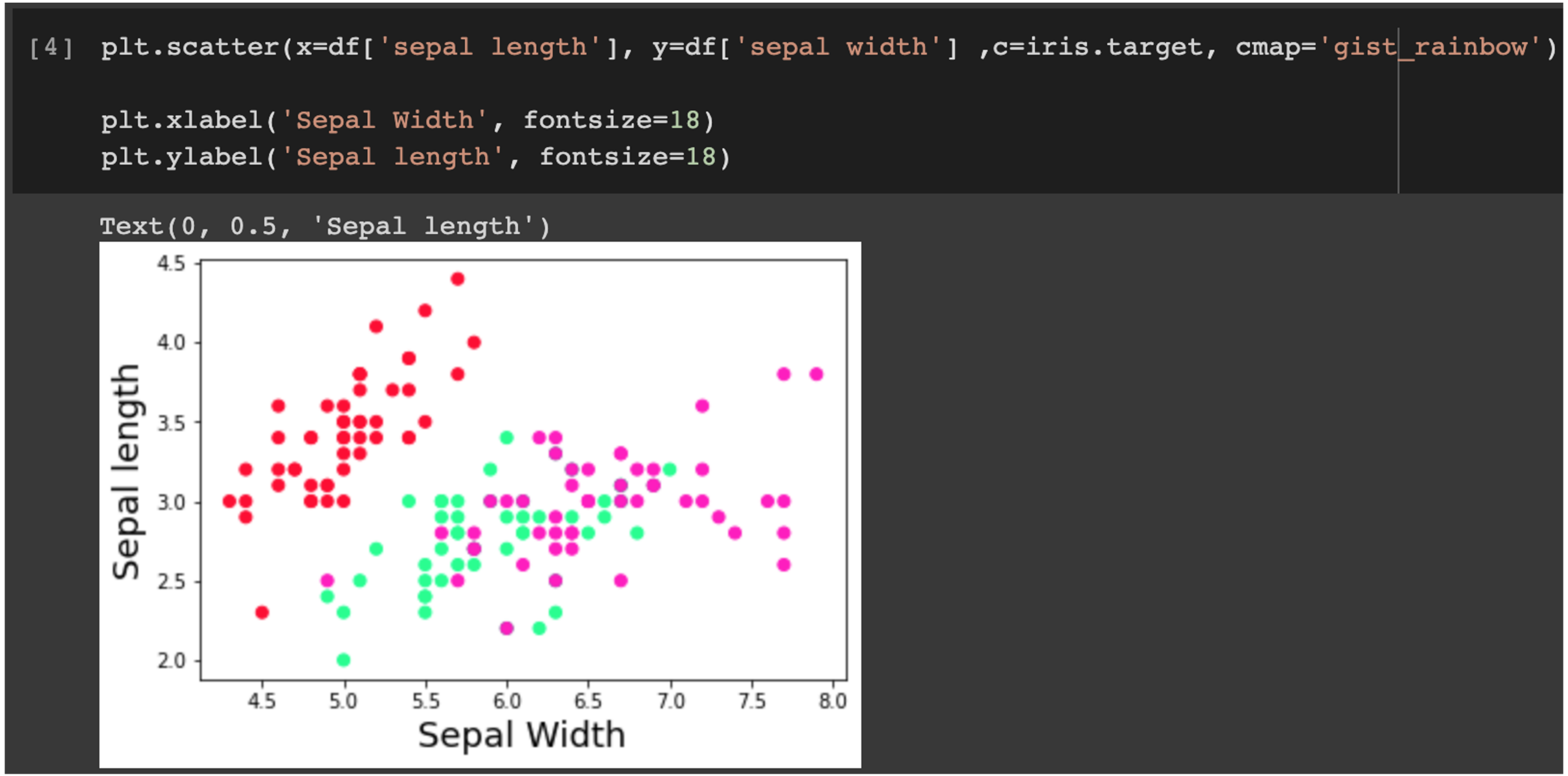


```
[3]  df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
     df['target']=pd.Series(iris.target)
     df
```

	sepal length	sepal width	petal length	petal width	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2

Build a clustering model – Iris dataset

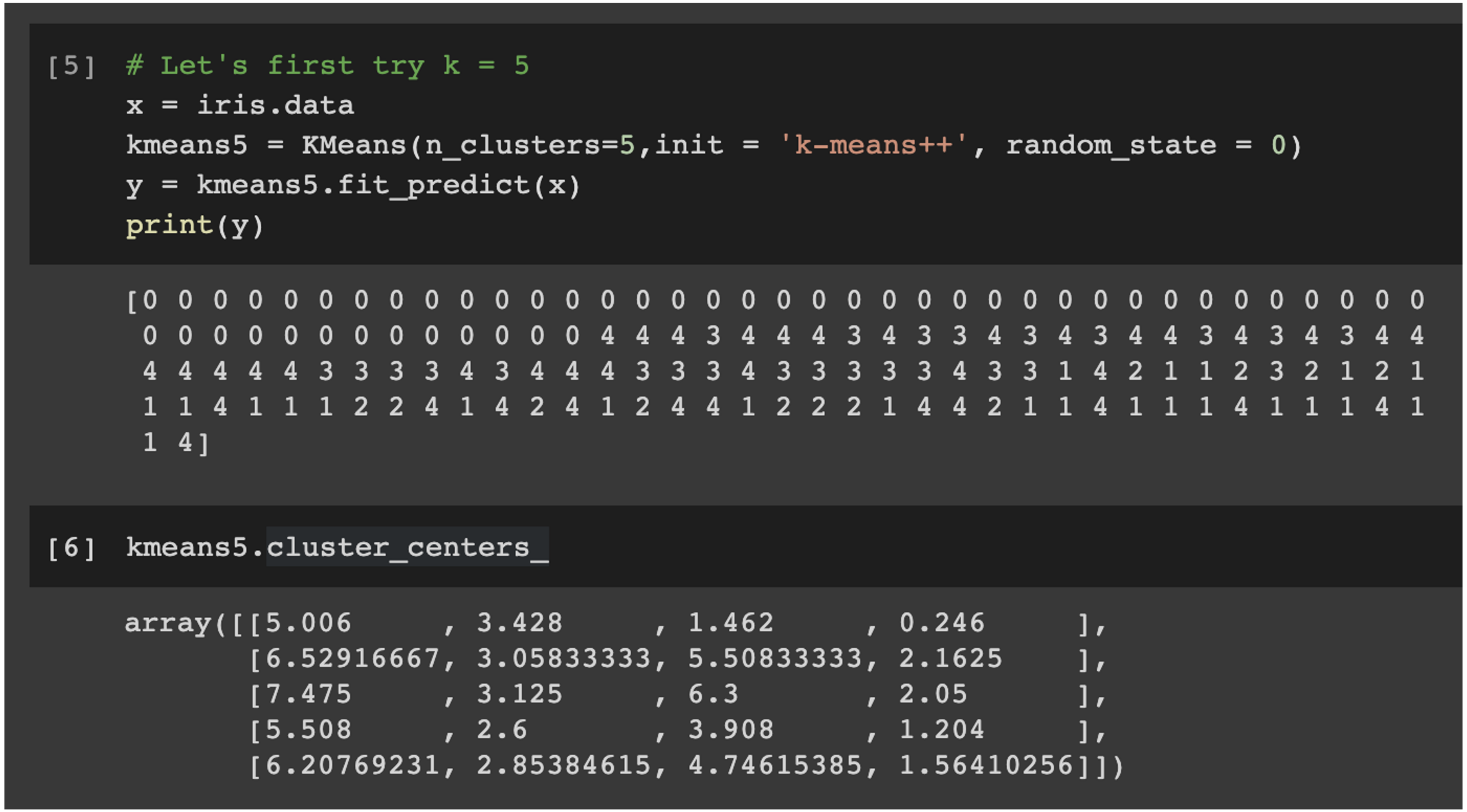
- Visualize the data



```
[4]  plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow')

     plt.xlabel('Sepal Width', fontsize=18)
     plt.ylabel('Sepal length', fontsize=18)

     Text(0, 0.5, 'Sepal length')
```

Build a clustering model – Iris dataset

- Estimate k with elbow method- first try k = 5



```
[5]  # Let's first try k = 5
     x = iris.data
     kmeans5 = KMeans(n_clusters=5,init = 'k-means++', random_state = 0)
     y = kmeans5.fit_predict(x)
     print(y)
```

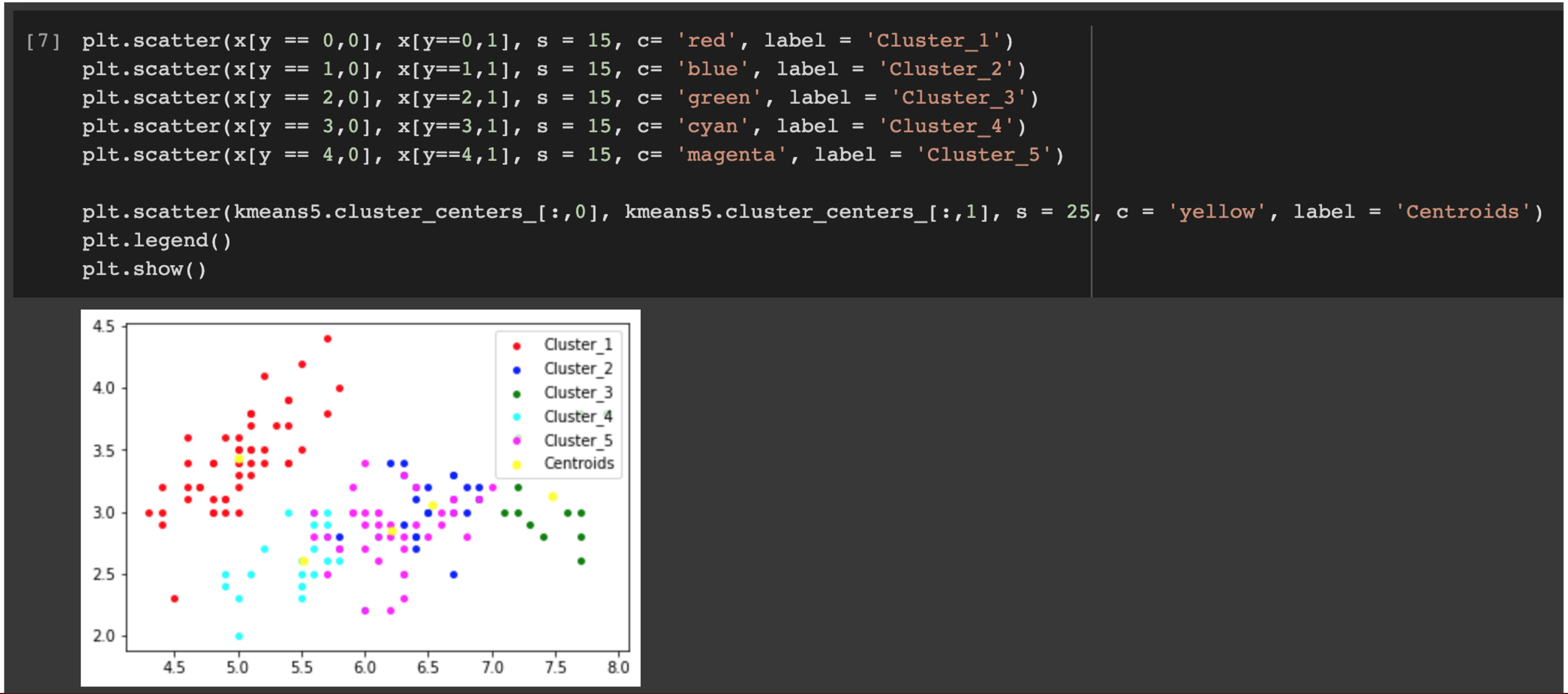
```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 4 4 4 3 4 4 4 3 4 3 3 4 3 4 3 4 4 3 4 3 4 3 4 4
 4 4 4 4 4 3 3 3 3 4 3 4 4 4 3 3 3 4 3 3 3 3 3 4 3 3 1 4 2 1 1 2 3 2 1 2 1
 1 1 4 1 1 1 2 2 4 1 4 2 4 1 2 4 4 1 2 2 2 1 4 4 2 1 1 4 1 1 1 4 1 1 1 4 1
 1 4]
```

```
[6]  kmeans5.cluster_centers_
```

```
array([[5.006     , 3.428     , 1.462     , 0.246     ],
       [6.52916667, 3.05833333, 5.50833333, 2.1625    ],
       [7.475     , 3.125     , 6.3       , 2.05      ],
       [5.508     , 2.6       , 3.908     , 1.204     ],
       [6.20769231, 2.85384615, 4.74615385, 1.56410256]])
```

Build a clustering model – Iris dataset

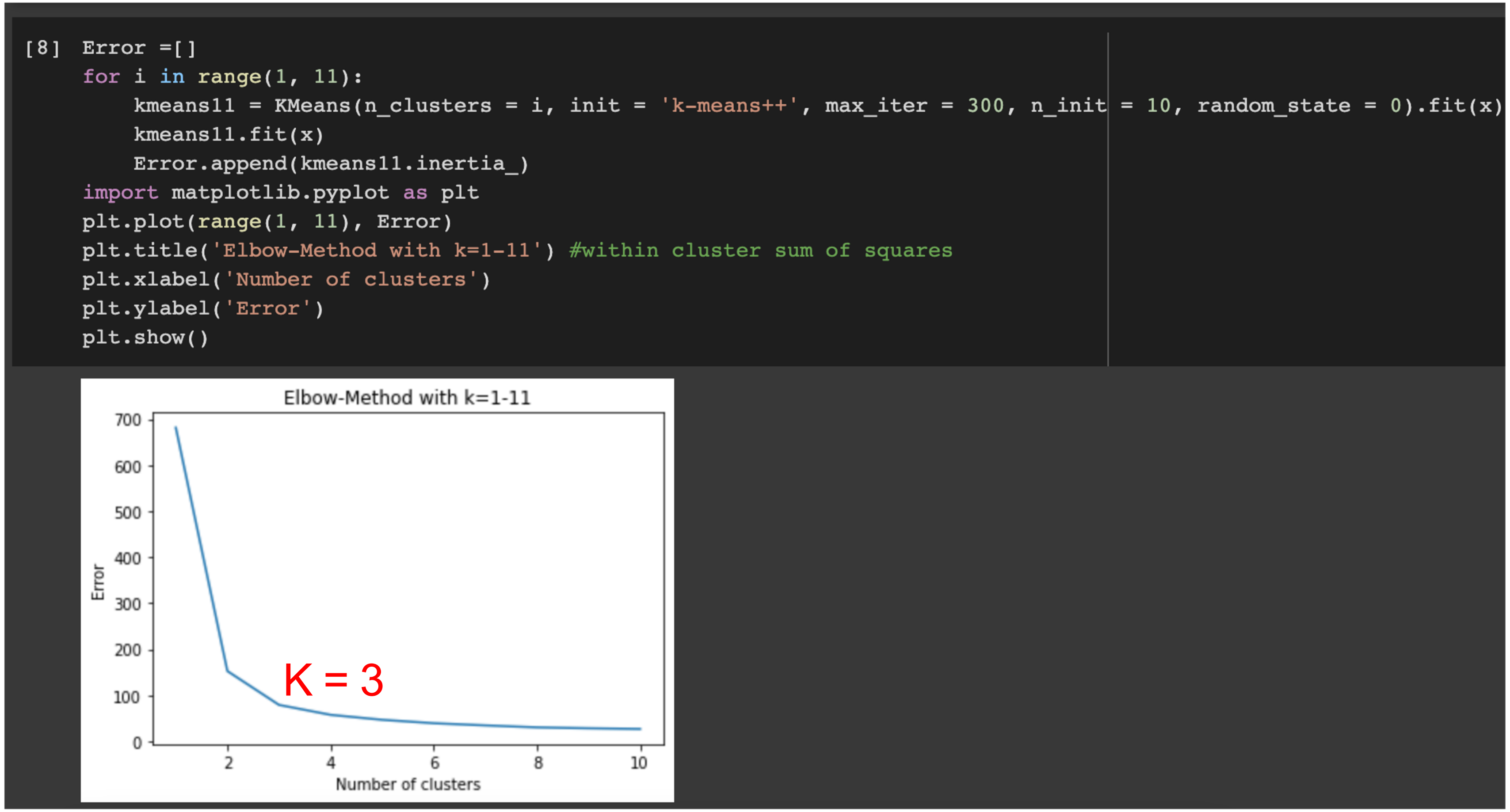
- Estimate k with elbow method



```
[7] plt.scatter(x[y == 0,0], x[y==0,1], s = 15, c= 'red', label = 'Cluster_1')
    plt.scatter(x[y == 1,0], x[y==1,1], s = 15, c= 'blue', label = 'Cluster_2')
    plt.scatter(x[y == 2,0], x[y==2,1], s = 15, c= 'green', label = 'Cluster_3')
    plt.scatter(x[y == 3,0], x[y==3,1], s = 15, c= 'cyan', label = 'Cluster_4')
    plt.scatter(x[y == 4,0], x[y==4,1], s = 15, c= 'magenta', label = 'Cluster_5')

    plt.scatter(kmeans5.cluster_centers_[:,0], kmeans5.cluster_centers_[:,1], s = 25, c = 'yellow', label = 'Centroids')
    plt.legend()
    plt.show()
```

Build a clustering model – Iris dataset

- Estimate k with elbow method

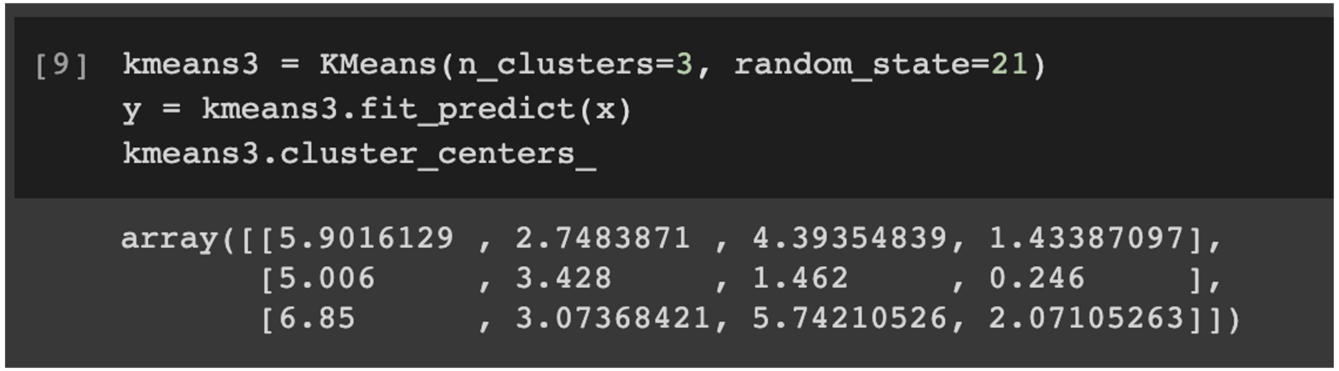


```
[8]  Error =[]
     for i in range(1, 11):
         kmeans11 = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0).fit(x)
         kmeans11.fit(x)
         Error.append(kmeans11.inertia_)
     import matplotlib.pyplot as plt
     plt.plot(range(1, 11), Error)
     plt.title('Elbow-Method with k=1-11') #within cluster sum of squares
     plt.xlabel('Number of clusters')
     plt.ylabel('Error')
     plt.show()
```

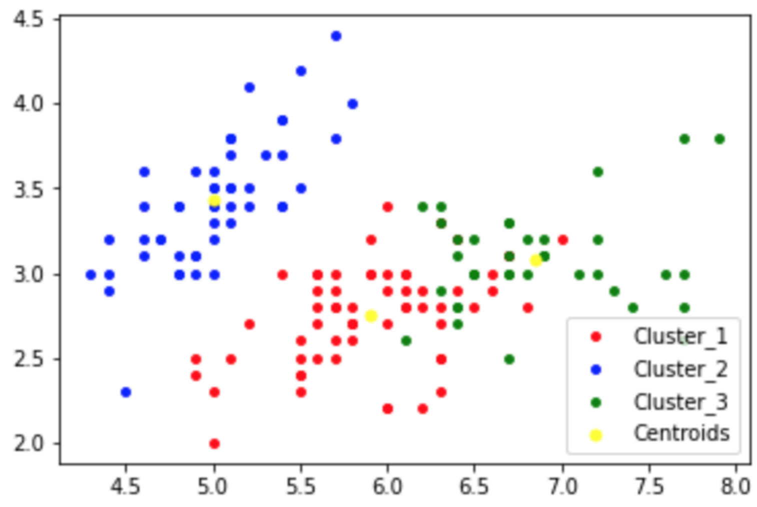
K = 3

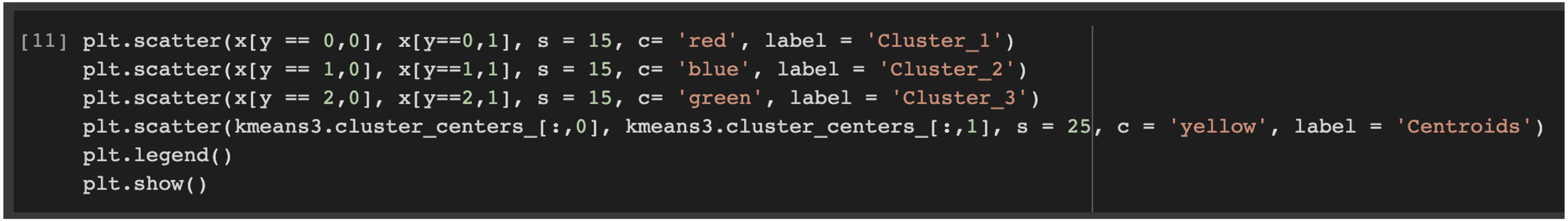
Build a clustering model – Iris dataset

- Get the optimal k = 3 from the elbow method. Cluster centers



```
[9]  kmeans3 = KMeans(n_clusters=3, random_state=21)
     y = kmeans3.fit_predict(x)
     kmeans3.cluster_centers_

     array([[5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
            [5.006     , 3.428     , 1.462     , 0.246     ],
            [6.85      , 3.07368421, 5.74210526, 2.07105263]])
```

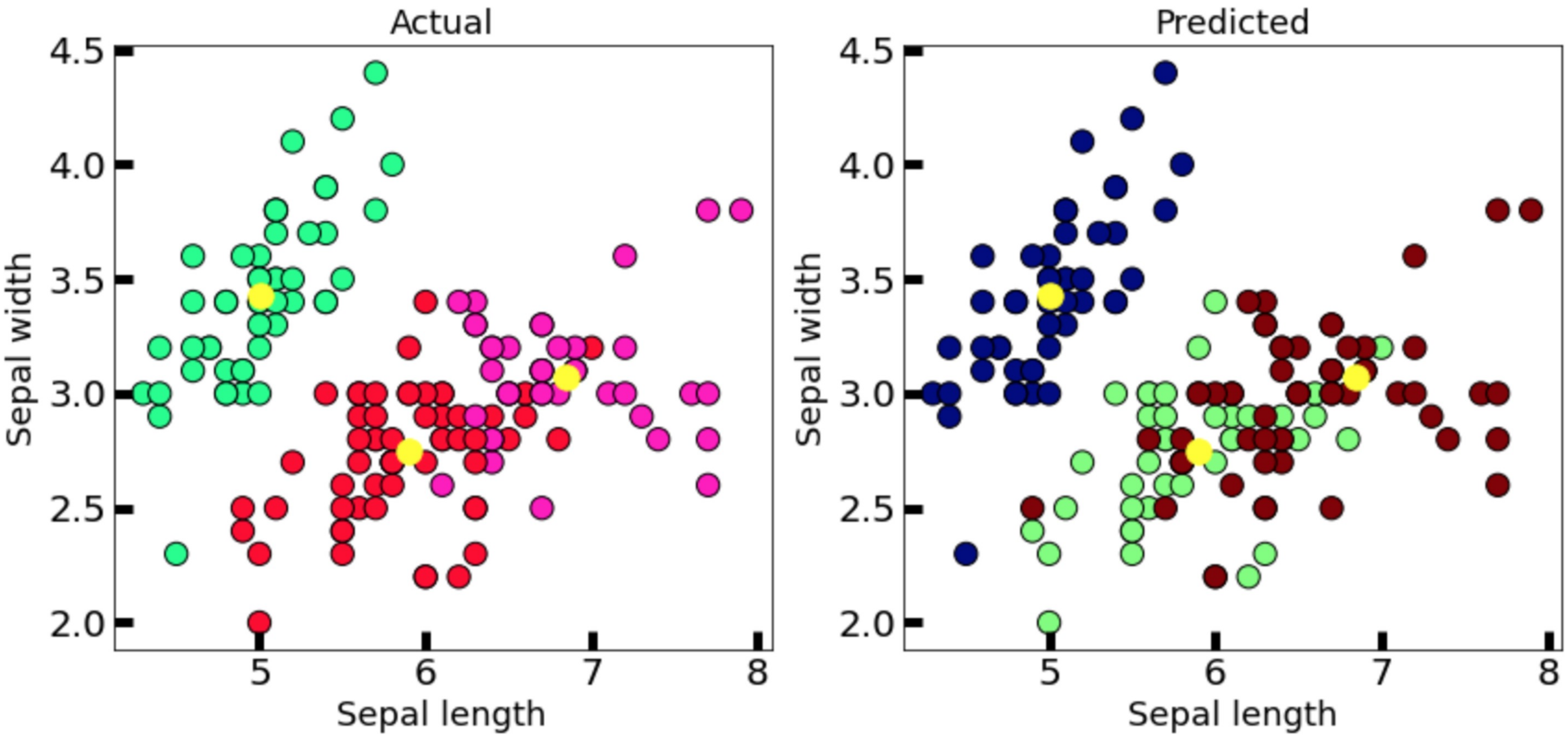




```
[11] plt.scatter(x[y == 0,0], x[y==0,1], s = 15, c= 'red', label = 'Cluster_1')
     plt.scatter(x[y == 1,0], x[y==1,1], s = 15, c= 'blue', label = 'Cluster_2')
     plt.scatter(x[y == 2,0], x[y==2,1], s = 15, c= 'green', label = 'Cluster_3')
     plt.scatter(kmeans3.cluster_centers_[:,0], kmeans3.cluster_centers_[:,1], s = 25, c = 'yellow', label = 'Centroids')
     plt.legend()
     plt.show()
```

Build a clustering model – Iris dataset

- Compared the actual and predicted clusters



Getting help

- HPC documentation docs.hpc.arizona.edu
- Support ticket https://uarizona.service-now.com/sp?id=sc_cat_item&sys_id=2983102adbd23c109627d90d689619c6&sysparm_category=84d3d1acdbc8f4109627d90d6896191f
- Office Hours – Wednesday 2-4 PM https://gather.town/app/dVsAprPNBVmI9NpL/hpc-office-hours
- HPC consulting hpc-consult@list.arizona.edu
- Visualization consulting vislab-consult@list.arizona.edu
- Statistics consulting stat-consult@list.arizona.edu