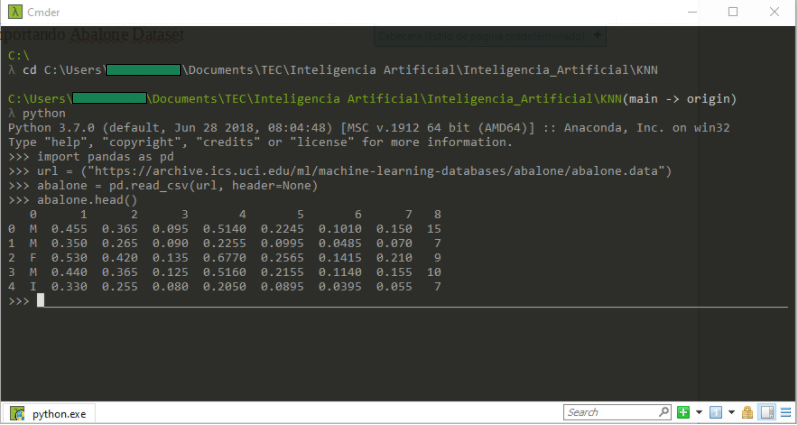
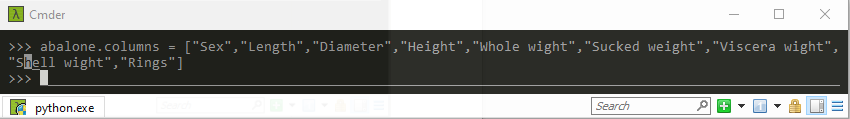
# **The k-Nearest Neighbors (kNN) Algorithm in Python**

**Ejercicio #1 Abalone Dataset**

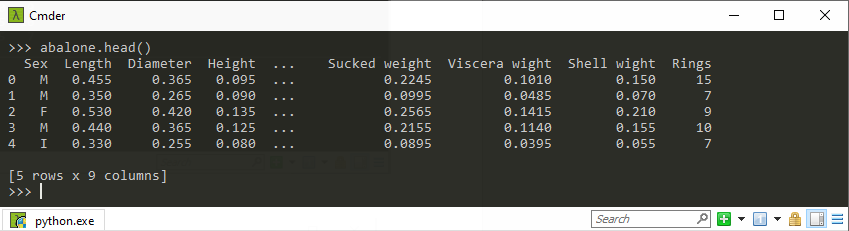
* Importar Abalone Dataset



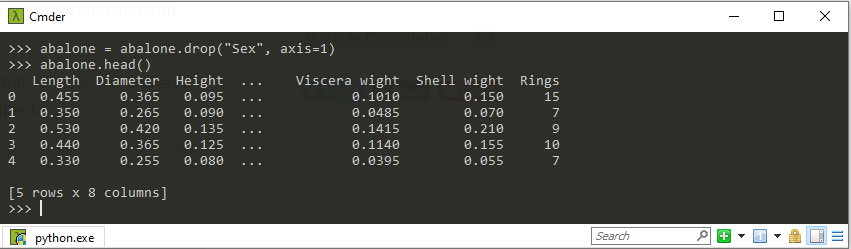
* Asignar nuestro propio DataFrame



* Visualizamos el head, nuestro encabezado con las columnas que le asignamos en el paso anterior

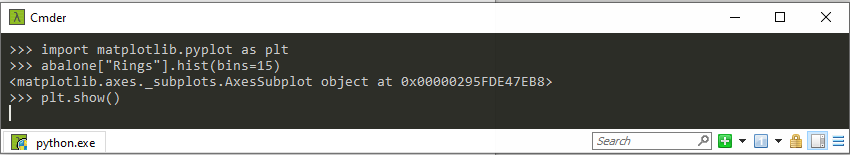


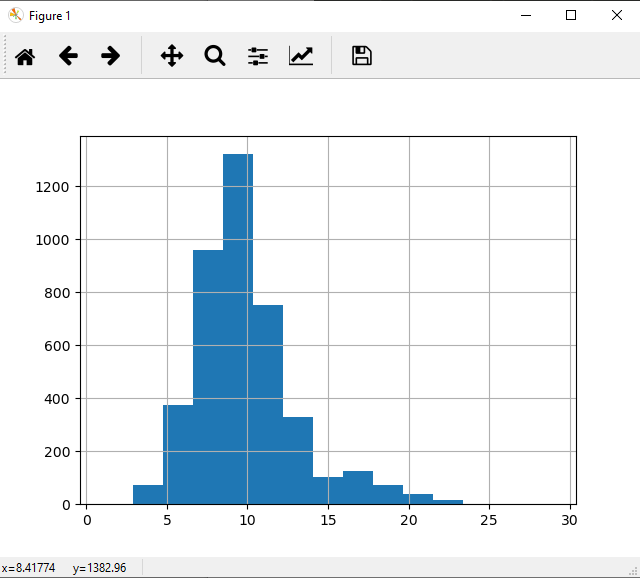
* Eliminar la columna “Sex” (**abalone = abalone.drop(“Sex” axis=1)**), ya que no es una medida física, el objetivo es predecir la edad. Y visualizar el resultado con el comando **abalone.head()**.



**Ejercicio #2 Descriptive Statistics From the Abalone Dataset**

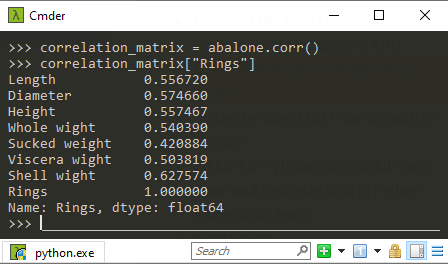
* Estadísticas del Abalone. Ejecutamos los siguientes comandos para poder generar un histograma, que corresponde a los datos que tenemos en nuestro Abalone.





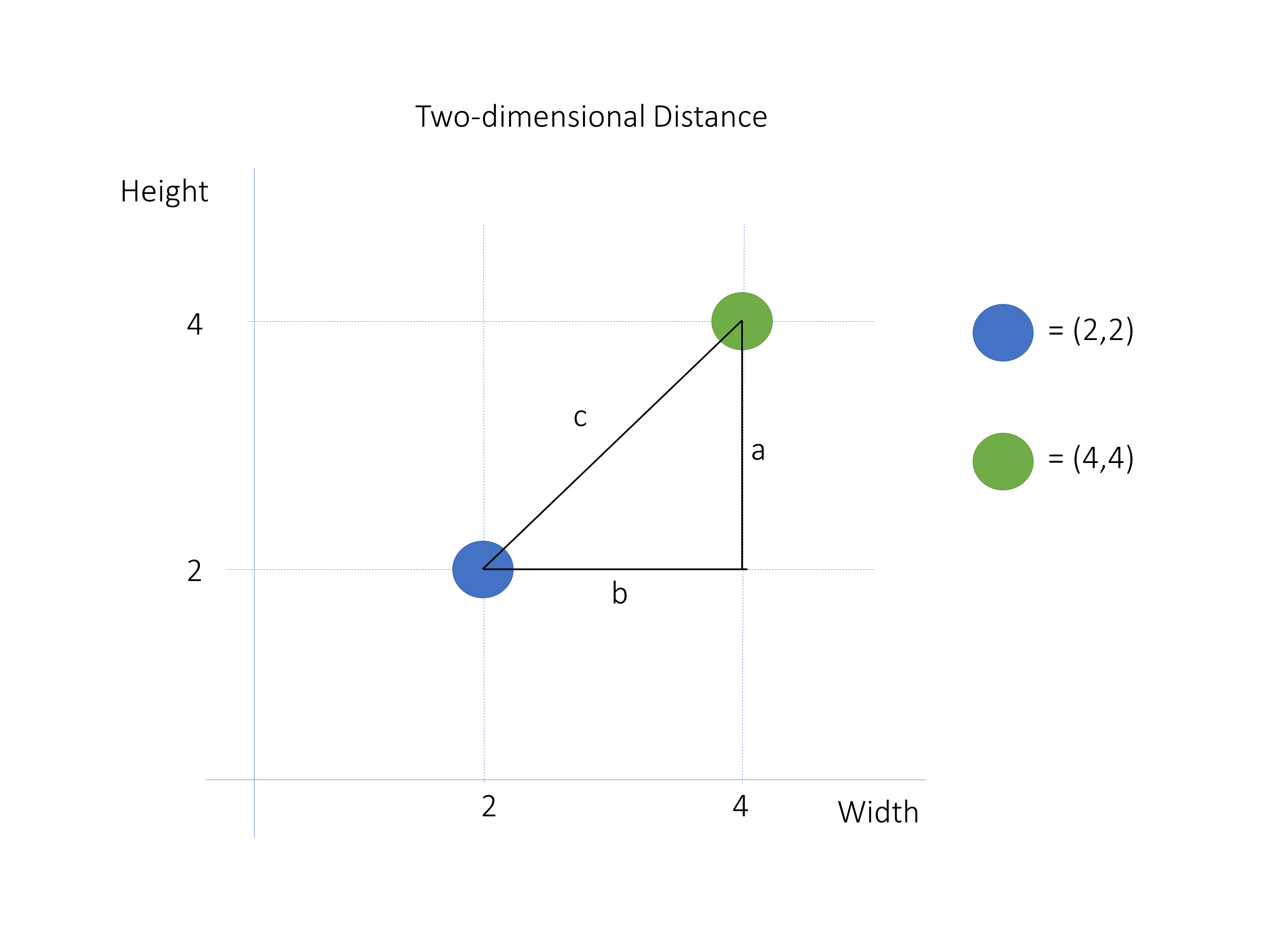
*Histograma de nuestro Abalone Dataset*

* Obtener una correlación de variable independiente con la variable objetivo. Observar la correlación completa en correlación-matrix. Las correlaciones más importantes son las que tienen la variable objetivo *Rings*.

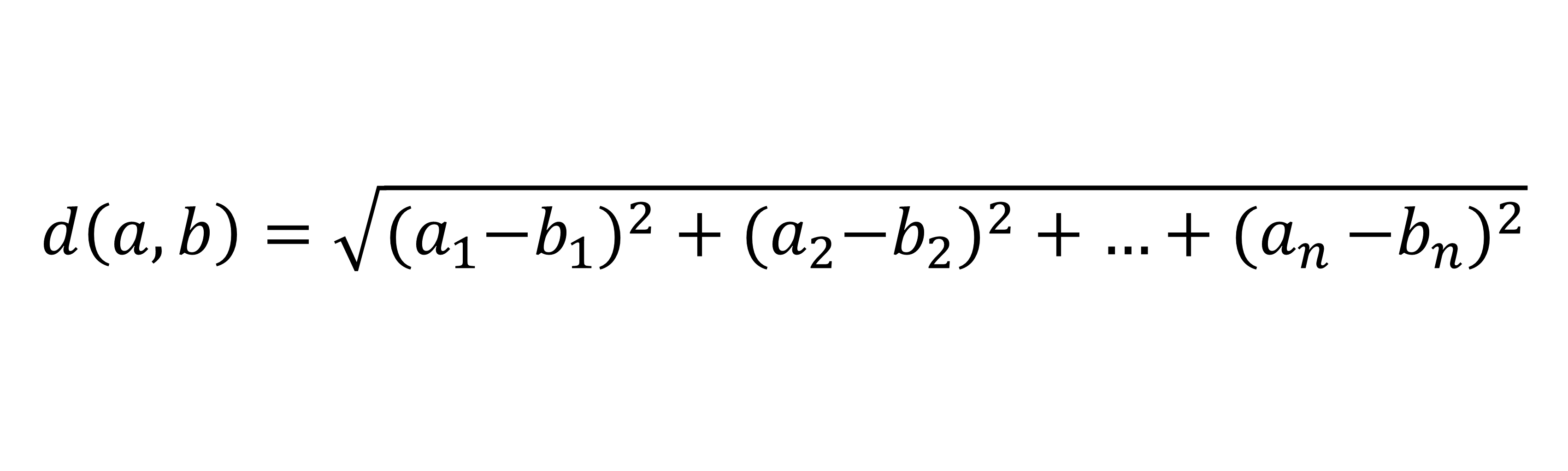


**Ejercicio #3 Define “Nearest” Using a Mathematical Definition of Distance**

1. Encontrando los puntos y distancias entre ellos, utilizando la fórmula de Euclides.



Encontramos la diferencia entre estos vectores por medio de la siguiente fórmula:

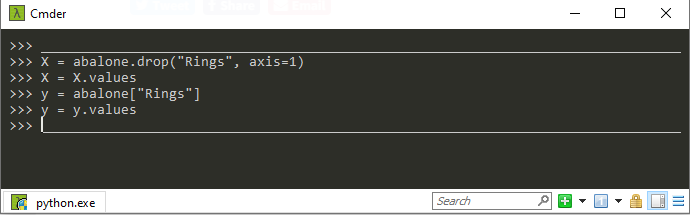


Utilizando Python, obtenemos el siguiente resultado:

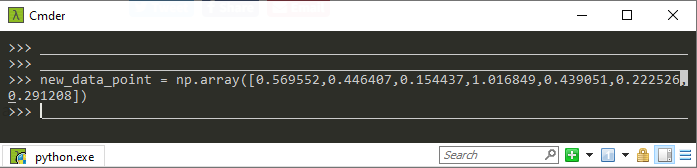


**Ejercicio #4 Find the k Nearest Neighbors**

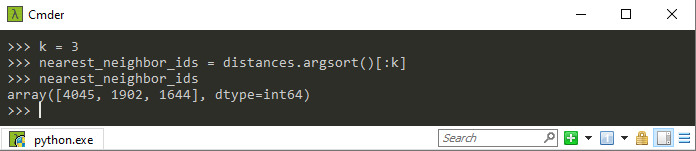
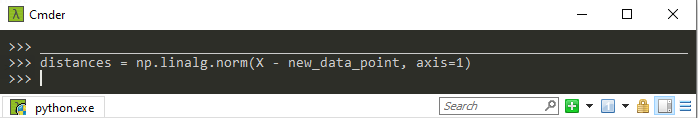
* Regresando a nuestro Abalone Dataset, e implementado una manera de obtener la distancia entre dos vectores utilizando *numpy*, podemos realizar una predicción. Recordando que el valor mínimo para **K** es 1. Este valor lo definimos nosotros. Usando numpy para obtener los valores de DataFrame:



* Crear un arreglo NumPy:



* Calculando las distancias:

****