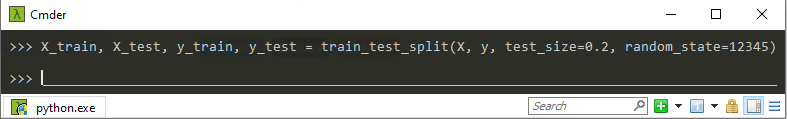
## **Fit kNN in Python Using scikit-learn**

**Ejercicio #1 Splitting Data Into Training and Test Sets for Model Evaluation**

**Training data:** División de datos en conjuntos de entrenamiento y prueba para la evaluación del modelo.

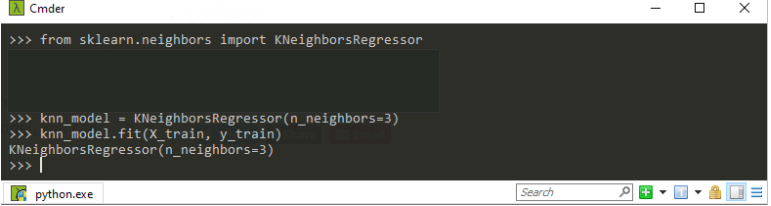
**Test data:** se utiliza para evaluar el modelo. Significa que hará predicciones para la cantidad de anillos de cada uno de los abulones en los datos de prueba y comparará esos resultados con la cantidad real conocida de anillos.

* Dividir los datos en conjuntos de entrenamiento usando la función train\_test\_split() integrado de scikit-learn:



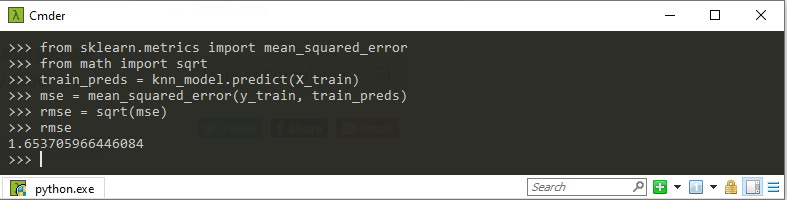
**Ejercicio #2 Fitting a kNN Regression in scikit-learn to the Abalone Dataset**

* Adaptar a un modelo skicit-learn

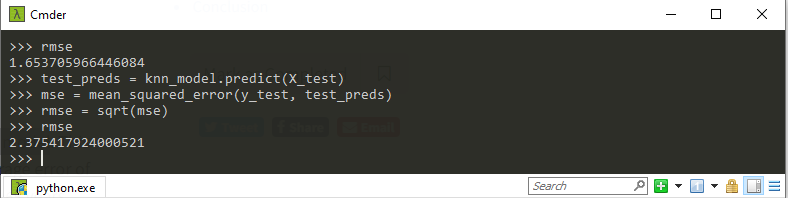


**Ejercicio #3 Using scikit-learn to Inspect Model Fit**

* Evaluar el error de predicción en los Training data.



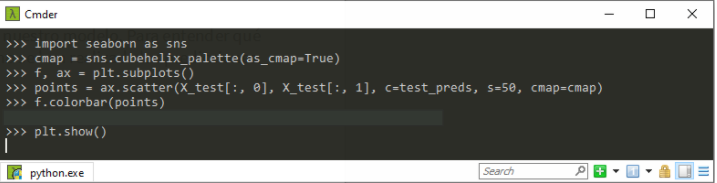
* Evaluar el error en datos que el modelo aún desconoce:

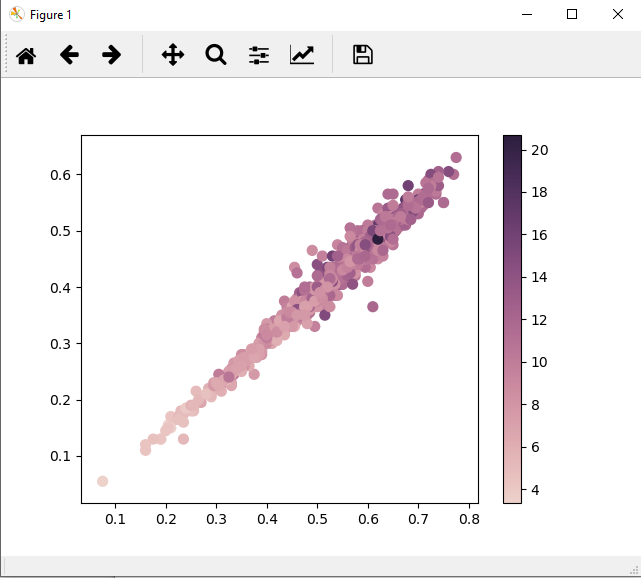


**Ejercicio #4 Plotting the Fit of Your Model**

Tomar en cuenta que para mejorar el modelo es el ajuste real de nuestro modelo. Para entender qué ha aprendido el modelo, utilizaremos Matplotlib, para saber cómo se han realizado las predicciones.

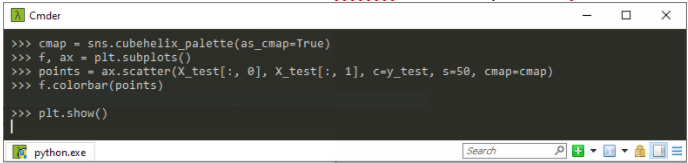
* Usar Seaborn para crear una gráfica de dispersión

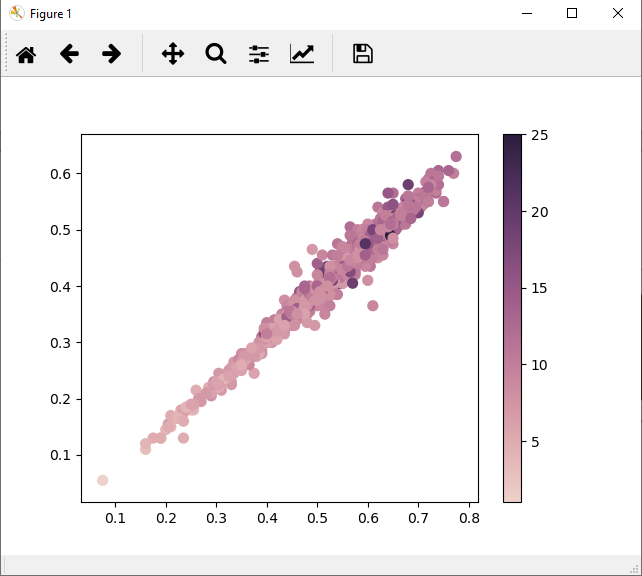




*Gráfica de dispersión*

* Confirmar la tendencia actual en nuestro Abalone Data, reemplazando **c**:





*Gráfica de dispersión*