

## Assignment 3 (50 marks total)

The data for this assignment is available at this [link](#).

### Question 1

**10 marks**

Question text goes here

#### Answer

Information for marking

```
Can have text in texinfo blocks
```

```
echo "Or code in bash blocks"
```

### Question 2

**20 marks**

Another question

#### Answer

Answer info

### Question 3

**20 marks**

A question with an image:

## Assignment 3

scikit learn [Install](#) [User Guide](#) [API](#) [Examples](#) [Community](#) [More](#)  [Go](#)

# scikit-learn

*Machine Learning in Python*

[Getting Started](#) [Release Highlights for 1.2](#) [GitHub](#)

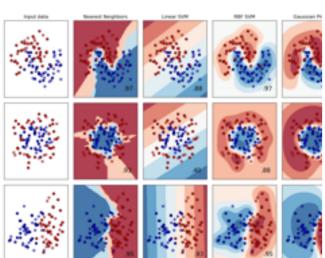
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

### Classification

Identifying which category an object belongs to.

**Applications:** Spam detection, image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, and more...



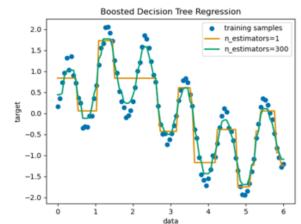
[Examples](#)

### Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, nearest neighbors, random forest, and more...



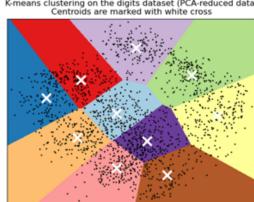
[Examples](#)

### Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, and more...



[Examples](#)

### Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency

**Algorithms:** PCA, feature selection, non-negative matrix

### Model selection

Comparing, validating and choosing parameters and models.

**Applications:** Improved accuracy via parameter tuning

**Algorithms:** grid search, cross validation, metrics, and more...

### Preprocessing

Feature extraction and normalization.

**Applications:** Transforming input data such as text for use with machine learning algorithms.

**Algorithms:** preprocessing,

**Figure 1:** An image

Course - Your name here

2

## Assignment 3

---

### Answer

Information for marking

Can have text in texinfo blocks

echo "Or code in bash blocks"

The screenshot shows a Jupyter Notebook interface with the following details:

- File Menu:** New, New Launcher, Open from Path..., Open from URL..., New View for Notebook, New Console for Notebook, Close Tab, Close and Shutdown Notebook, Close All Tabs, Save Notebook, Save Notebook As..., Save All, Reload Notebook from Disk, Revert Notebook to Checkpoint, Rename Notebook..., Download, Save and Export Notebook As..., Save Current Workspace As..., Save Current Workspace, Print..., Hub Control Panel, Log Out.
- Code Cells:** Cell 1: 

```
1 import sys
2 !{sys.executable} -m pip install Bio
```

 Output: ATGATCCCTAG  
Cell 2: 

```
1 from Bio.Seq import Seq
2 my_seq_object = Seq("ATGATCCCTAG")
3 print(my_seq_object)
```

 Output: ATGATCCCTAG  
Cell 3: 

```
1 # translate the DNA
2 new_seq_object = my_seq_object.translate()
3 print(new_seq_object)
```

 Output: MIP\*  
Cell 4: 

```
1 # store the protein sequence in a string
2 my_string = str(new_seq_object)
3 # remove the last character
4 my_string = my_string[:-1]
5 print(my_string)
```

 Output: MIP  
Cell 5: 

```
1
```

 Output:
- SIMPLE.IPYNB Variables:** my\_seq:Seq, my\_seq\_object:Seq, my\_string:MIP
- CALLSTAC:** Shows a call stack with several entries.
- BREAKPOINTS:** Shows a list of breakpoints.
- SOURCE:** Shows the source code for the current file.
- KERNEL SOURCES:** Shows kernel sources for Bio, Bio.Data, Bio.Data.CodonTable, Bio.Data.IUPACData, and Bio.Seq.
- Bottom Status Bar:** Shows "Connecting...", "Python 3 (ipykernel) | ... Mode: Comm...", "Ln 1, C...", "English (United States)", and "simple.ipynb".

Figure 2: An image