# Workflow

## Lecture

|  |  |
| --- | --- |
| Key messages | Questions from students |
| ~45 min  State-less transformations can be performed in a single step with just the given data e.g. linear transformation  State-full transformations reference the data e.g. StandardScaler or MinMaxScaler, and require a class to be coded |  |

## 1. Pre-processor Tuning

|  |  |
| --- | --- |
| Key messages | Questions from students |
| Use sorted(sklearn.metrics.SCORERS.keys())to pull up a list of metrics to be fed into GridSearch |  |

## 2. Custom Transformer

|  |  |
| --- | --- |
| Key messages | Questions from students |
|  |  |

## 3. Tuning Pipeline

|  |  |
| --- | --- |
| Key messages | Questions from students |
| - Need to use randomised search  - Need to use import  - Not all of the options will work for each of the tuning parameters  - Remember to use the right scoring method  - Make sure all the variables have the right name |  |

## 4. Pickle Pipe

|  |  |
| --- | --- |
| Key messages | Questions from students |
| - Will need to keep the final solution as an array |  |

## 5. Hand Made Standardizer

|  |  |
| --- | --- |
| Key messages | Questions from students |
| This is actually quite hard as it is not in the lecture notes. Might be worth doing a separate section on this a bit later if students get up to it.  Need to first understand the formula  Then assign attributes that align with the formula  np.std(ddof = 0) is required to solve this problem  y is not really taken into account for any of these functions |  |

## Recap- Bonus

|  |  |
| --- | --- |
| Key messages | Questions from students |
|  |  |

## Recap- Custom Encoder

|  |  |
| --- | --- |
| Key messages | Questions from students |
|  |  |