# Data preparation (DP) task overview

## Data Preparation Concepts

* [Data Science Methodology 101 - Data Preparation Concepts](https://www.youtube.com/watch?v=BiPzz1xZ0XI)
* A free course from Google AI that covers topics of data preparation and feature engineering:  
  <https://developers.google.com/machine-learning/data-prep>
* General description of data preparation procedure:  
  <https://machinelearningmastery.com/prepare-data-machine-learning-python-scikit-learn/>

## Gaps in data

* A section of imputers in sklearn:  
  <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.impute>
* A small article about different imputation strategies:  
  <https://scikit-learn.org/stable/modules/impute.html>
* A python library with additional imputation algorithms:  
  <https://pypi.org/project/impyute/>

## Continuous features

How, why, and when to standardize data:

* <https://www.youtube.com/watch?v=ZRS9xCPfvrY>
* <https://humansofdata.atlan.com/2018/12/data-standardization/#:~:text=Data%20standardization%20is%20about%20making,t%20easy%20to%20compare%20otherwise.>
* <https://builtin.com/data-science/when-and-why-standardize-your-data>
* <https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff>
* https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

Standardization (or mean removal and variance scaling):

* <https://scikit-learn.org/stable/modules/preprocessing.html#standardization-or-mean-removal-and-variance-scaling>

## Discrete features

Some pages with methods description and code snippets:

* <https://towardsdatascience.com/smarter-ways-to-encode-categorical-data-for-machine-learning-part-1-of-3-6dca2f71b159>
* <https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd>

A python library with categorical features encoding algorithms:

* <http://contrib.scikit-learn.org/category_encoders/index.html>

## Ordinal

* Scikit-learn's ordinal encoder:  
  <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html#sklearn.preprocessing.OrdinalEncoder>

## Nominal

* Scikit-learn's one-hot encoder:  
  <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html#sklearn.preprocessing.OneHotEncoder>

## Combine everything into a pipeline

This section describes the general concept of the pipeline in Scikit-learn and also gives some code examples:

<https://scikit-learn.org/stable/modules/compose.html#combining-estimators>

This is an example of how to combine continuous and discrete features preprocessing steps into one pipeline:

<https://scikit-learn.org/stable/auto_examples/compose/plot_column_transformer_mixed_types.html#sphx-glr-auto-examples-compose-plot-column-transformer-mixed-types-py>

Also another example of textual data preprocessing with Pipeline, FunctionTransformer, and ColumnTransformer:

<https://scikit-learn.org/stable/auto_examples/compose/plot_column_transformer.html#sphx-glr-auto-examples-compose-plot-column-transformer-py>

In case you need to get two different sets of features from the same dataset:

<https://scikit-learn.org/stable/modules/compose.html#featureunion-composite-feature-spaces>