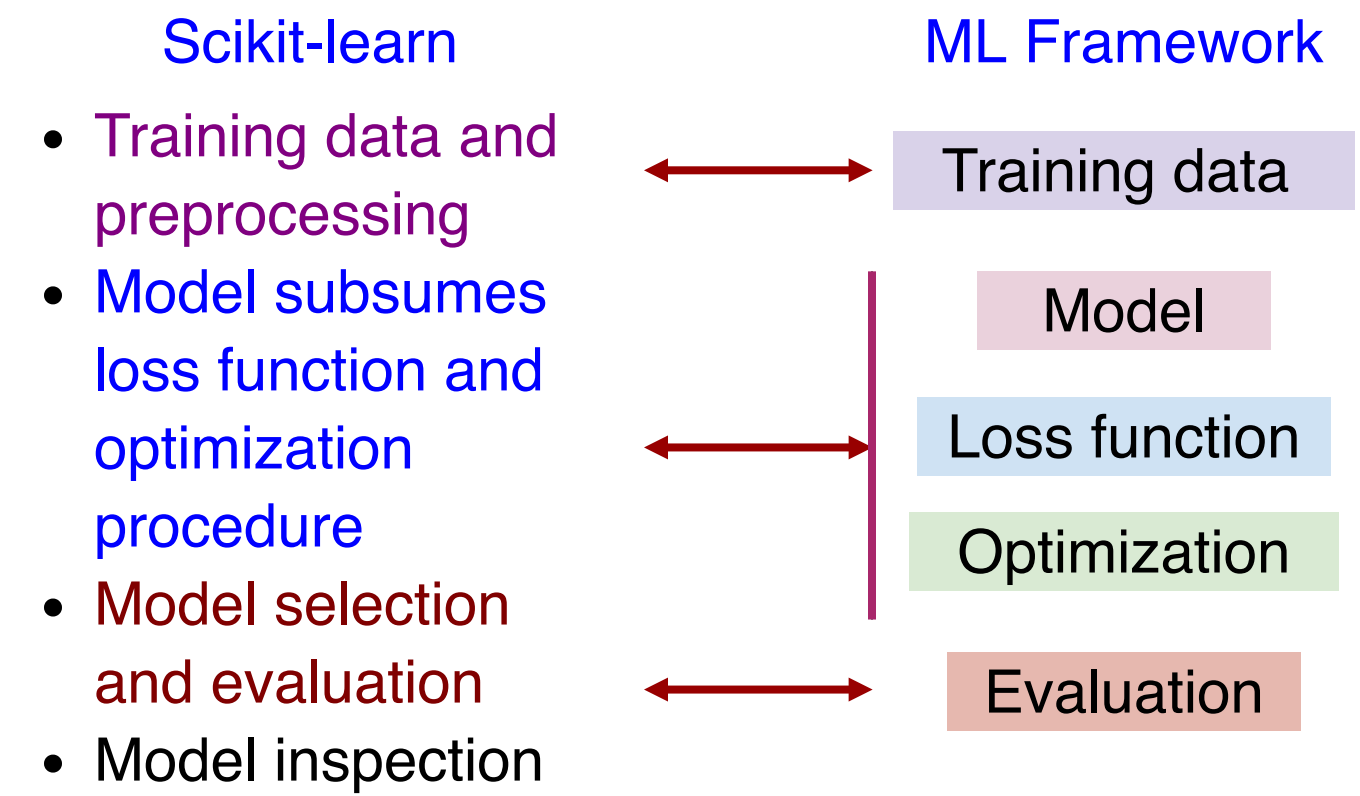
Introduction to Scikit-Learn (sklearn)

sklearn APIs are organized on the lines of our ML framework.



API design principles

@sir, copied to 'Data Preprocessing' slide deck

sklearn APIs are well designed with the following principles:

- **Consistency**: All APIs share a simple and consistent interface.
- **Inspection**: The learnable parameters as well as hyperparameters of all estimator's are accessible directly via public instance variables.
- **Nonproliferation of classes**: Datasets are represented as Numpy arrays or Scipy sparse matrix instead of custom designed classes.
- **Composition**: Existing building blocks are reduced as much as possible.
- **Sensible defaults** values are used for parameters that enables quick baseline building.

@sir, copied to 'Data Preprocessing' slide deck

Types of sklearn objects

Transformers

- transforms dataset
- `transform()` for transforming dataset.
- `fit()` learns parameters.
- `fit_transform()` fits parameters and `transform()` the dataset.

Estimators

- Estimates model parameters based on training data and hyper parameters.
- `fit()` method

Predictors

- Makes prediction on dataset
- `predict()` method that takes dataset as an input and returns predictions.
- `score()` method to measure quality of predictions.

Data Preprocessing → Training → Inference

sklearn API

@sir, copied to 'Data Preprocessing' slide deck

Data API

Provides functionality for loading, generating and preprocessing the training and test data.

Module	Functionality
`sklearn.datasets`	Loading datasets - custom as well as popular reference dataset.
`sklearn.preprocessing`	Scaling, centering, normalization and binarization methods
`sklearn.impute`	Filling missing values
`sklearn.feature_selection`	Implements feature selection algorithms
`sklearn.feature_extraction`	Implements feature extraction from raw data.

Model API

Implements supervised and unsupervised models

Regression

- `sklearn.linear_model` (linear, ridge, lasso models)

`sklearn.trees`

Classification

- `sklearn.linear_model`
- `sklearn.svm`
- `sklearn.trees`
- `sklearn.neighbors`
- `sklearn.naive_bayes`
- `sklearn.multiclass`

`sklearn.multioutput` implements multi-output classification and regression.

`sklearn.cluster` implements many popular clustering algorithms

Model evaluation API

`sklearn.metrics` implements different metrics for model evaluation.

- Classification metrics
- Regression metrics
- Clustering metrics

Model selection API

`sklearn.model_selection` implements various model selection strategies like cross-validation, tuning hyper-parameters and plotting learning curves.

Model inspection API

`sklearn.model_inspection` includes tools for model inspection.

Practical advice

- It is not possible to remember each and every sklearn API.
- Remember high level modules and API design principles.
- Use documentation for more information as follows:

```
1 import sklearn.linear_model import LogisticRegression
2 ?LogisticRegression
```

- Keep the following links handy:
  - API reference
  - sklearn user guide
  - Worked examples for reference implementations