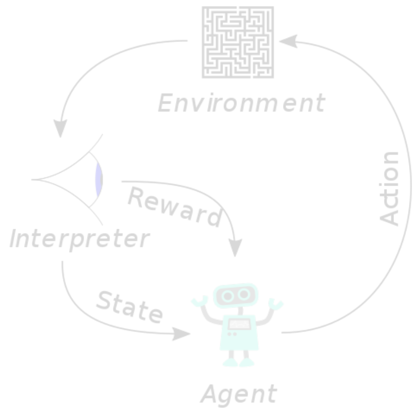
[320] Classification

Department of Computer Sciences
University of Wisconsin-Madison

Machine Learning

Reinforcement Learning

not covered in CS 320



https://en.wikipedia.org/wiki/Reinforcement_learning

Supervised Machine Learning

*data is **labeled**, we know what we want to predict*

Regression

predict a quantity

Classification

predict a category

Unsupervised Machine Learning

*data is **unlabeled**, we're just looking for patterns*

Clustering

place rows in groups

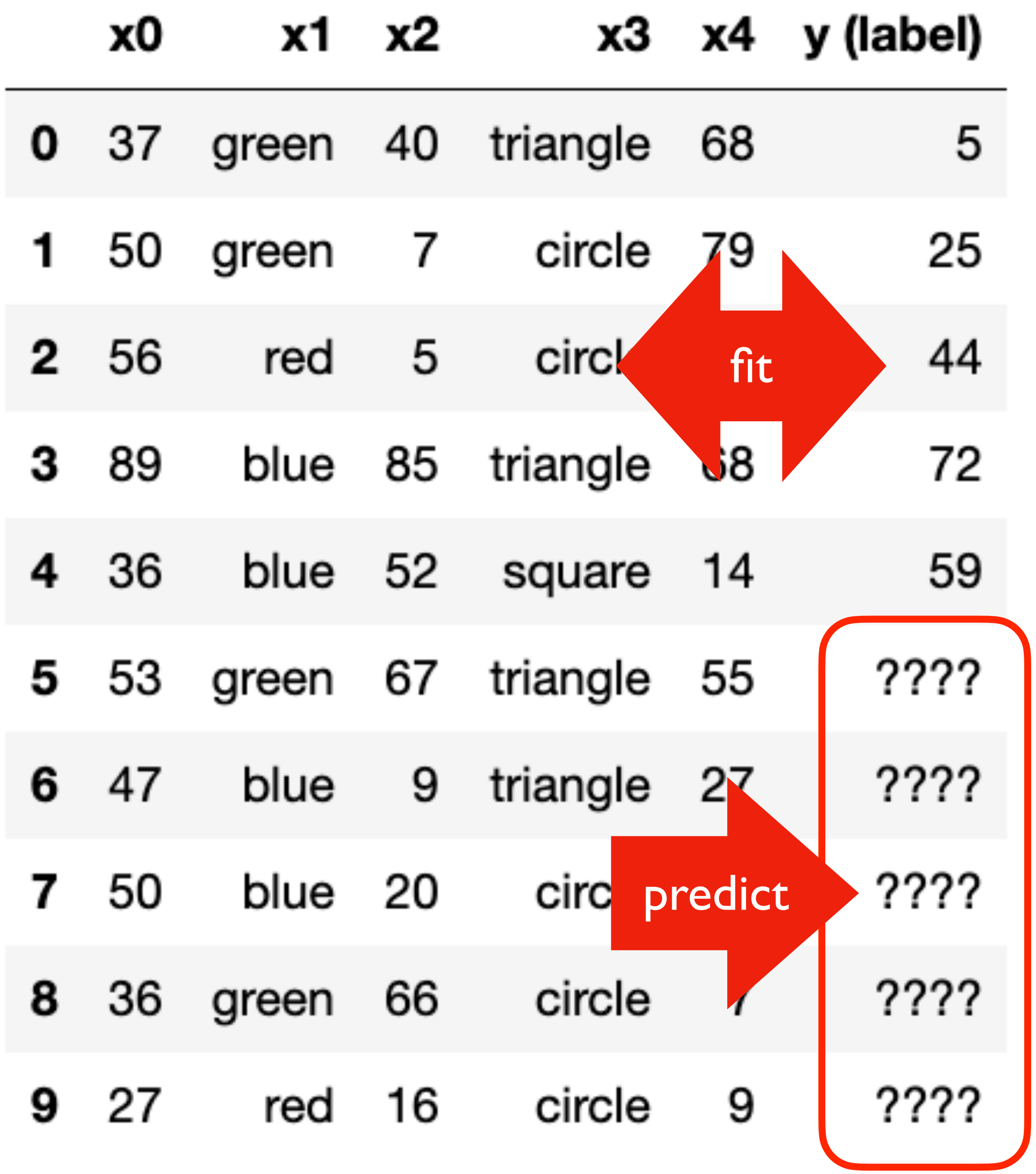
Decomposition

represent rows as combos of "component" rows

this semester, we'll learn at least one technique in each of these four categories

1. Regression (Supervised)

quantitative
label



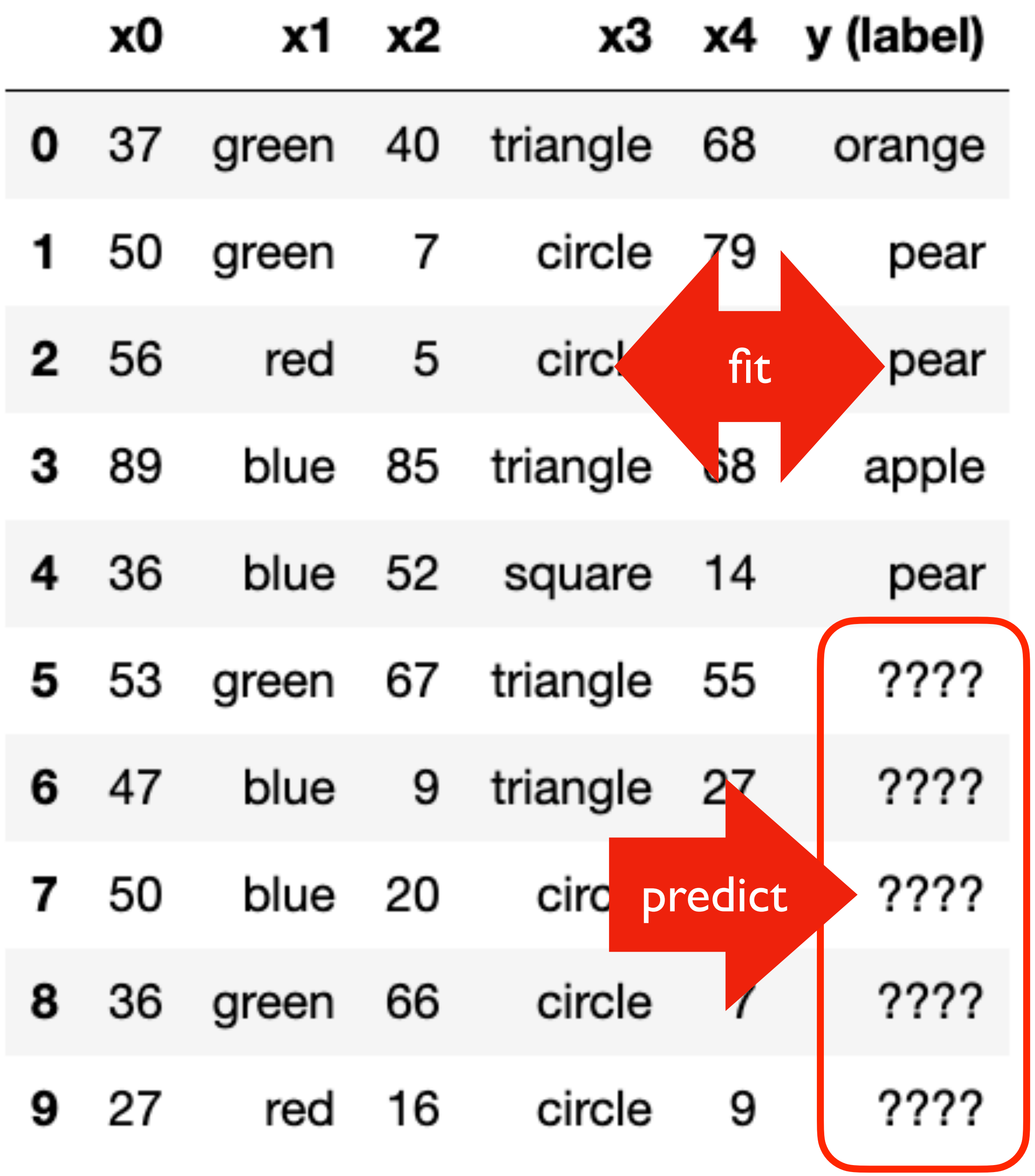
	x0	x1	x2	x3	x4	y (label)
0	37	green	40	triangle	68	5
1	50	green	7	circle	79	25
2	56	red	5	circle	[illegible]	44
3	89	blue	85	triangle	68	72
4	36	blue	52	square	14	59
5	53	green	67	triangle	55	????
6	47	blue	9	triangle	27	????
7	50	blue	20	circle	[illegible]	????
8	36	green	66	circle	7	????
9	27	red	16	circle	9	????

fit
predict

problem: can we predict an unknown **quantity**?

2. Classification (Supervised)

categorical
label



	x0	x1	x2	x3	x4	y (label)
0	37	green	40	triangle	68	orange
1	50	green	7	circle	79	pear
2	56	red	5	circle	[illegible]	pear
3	89	blue	85	triangle	68	apple
4	36	blue	52	square	14	pear
5	53	green	67	triangle	55	????
6	47	blue	9	triangle	27	????
7	50	blue	20	circle	[illegible]	????
8	36	green	66	circle	7	????
9	27	red	16	circle	9	????

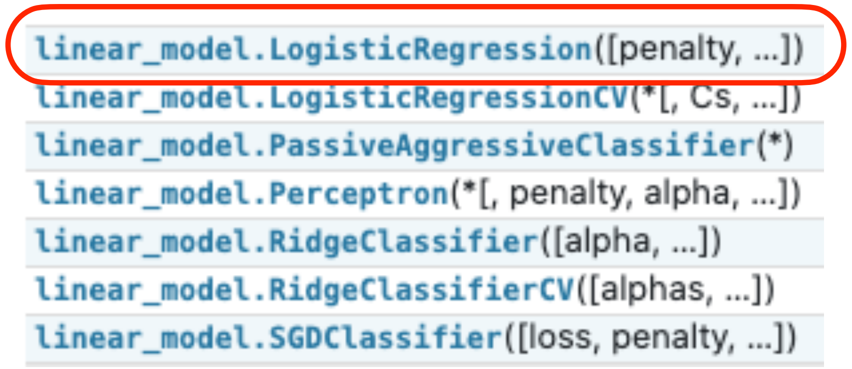
fit
predict

problem: can we predict an unknown **category**?

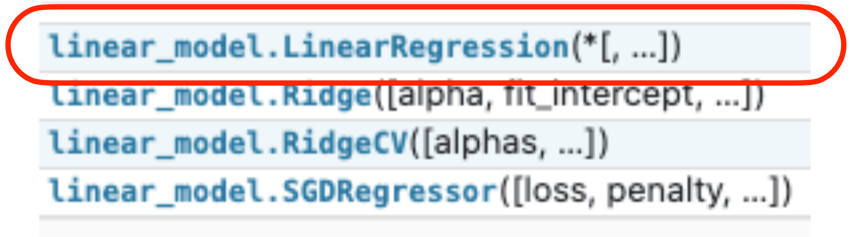
1. Regression (Supervised)

+

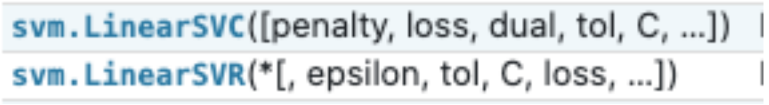
2. Classification (Supervised)



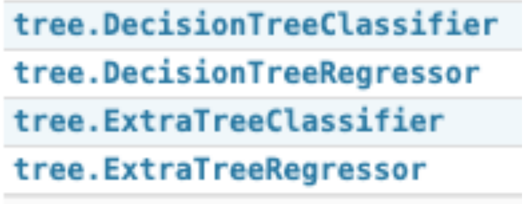
linear_model.LogisticRegression([penalty, ...])
linear_model.LogisticRegressionCV(*[, Cs, ...])
linear_model.PassiveAggressiveClassifier(*)
linear_model.Perceptron(*[, penalty, alpha, ...])
linear_model.RidgeClassifier([alpha, ...])
linear_model.RidgeClassifierCV([alphas, ...])
linear_model.SGDClassifier([loss, penalty, ...])



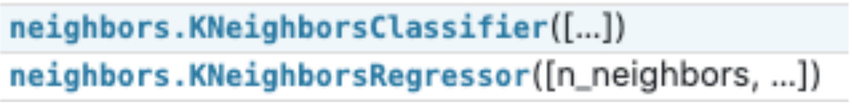
linear_model.LinearRegression(*[, ...])
linear_model.Ridge([alpha, fit_intercept, ...])
linear_model.RidgeCV([alphas, ...])
linear_model.SGDRegressor([loss, penalty, ...])



svm.LinearSVC([penalty, loss, dual, tol, C, ...])
svm.LinearSVR(*[, epsilon, tol, C, loss, ...])

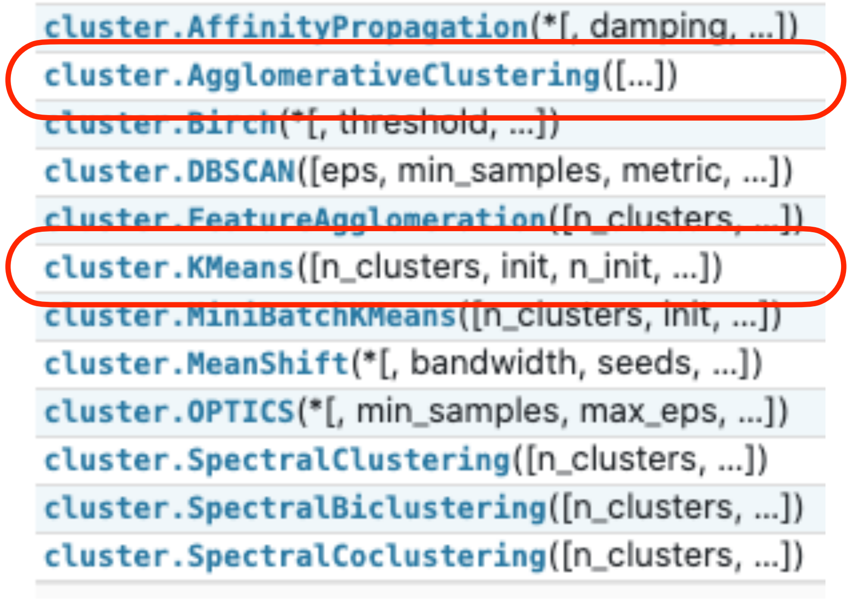


tree.DecisionTreeClassifier
tree.DecisionTreeRegressor
tree.ExtraTreeClassifier
tree.ExtraTreeRegressor



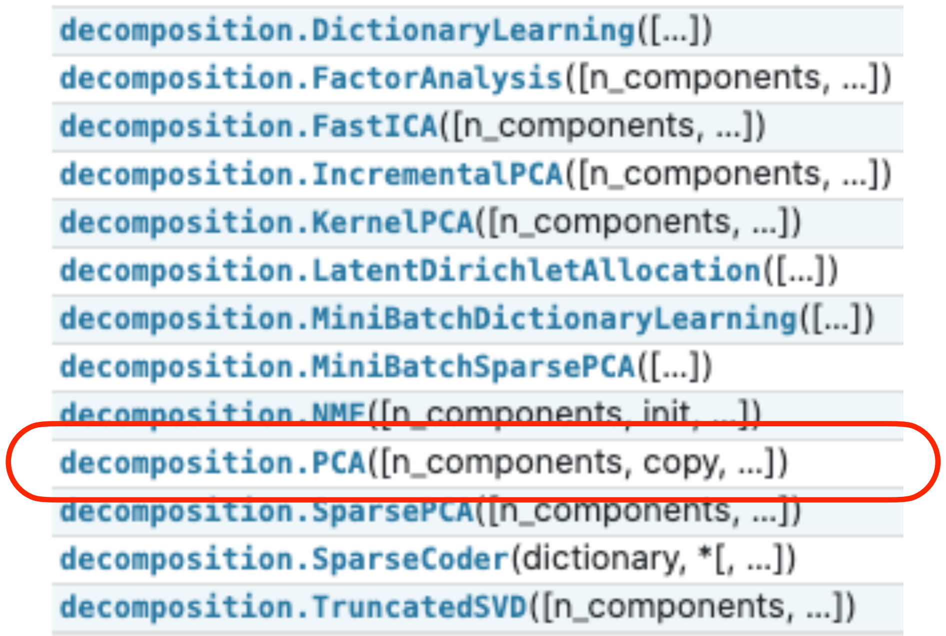
neighbors.KNeighborsClassifier([...])
neighbors.KNeighborsRegressor([n_neighbors, ...])

3. Clustering (Unsupervised)



cluster.AffinityPropagation(*[, damping, ...])
cluster.AgglomerativeClustering([...])
cluster.Birch(*[, threshold, ...])
cluster.DBSCAN([eps, min_samples, metric, ...])
cluster.FeatureAgglomeration([n_clusters, ...])
cluster.KMeans([n_clusters, init, n_init, ...])
cluster.MiniBatchKMeans([n_clusters, init, ...])
cluster.MeanShift(*[, bandwidth, seeds, ...])
cluster.OPTICS(*[, min_samples, max_eps, ...])
cluster.SpectralClustering([n_clusters, ...])
cluster.SpectralBiclustering([n_clusters, ...])
cluster.SpectralCoclustering([n_clusters, ...])

4. Decomposition (Unsupervised)



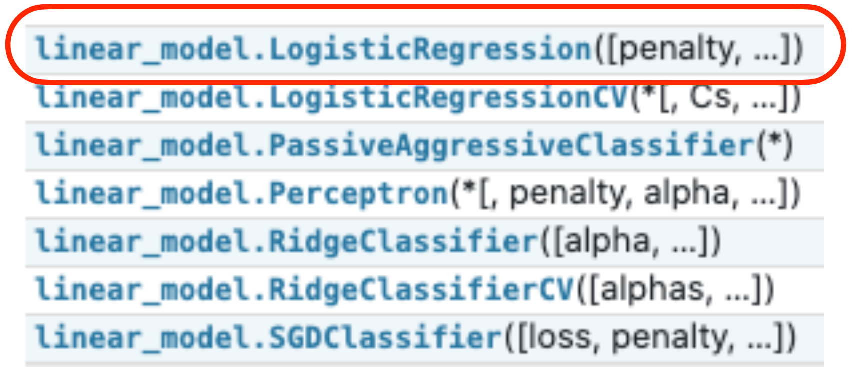
decomposition.DictionaryLearning([...])
decomposition.FactorAnalysis([n_components, ...])
decomposition.FastICA([n_components, ...])
decomposition.IncrementalPCA([n_components, ...])
decomposition.KernelPCA([n_components, ...])
decomposition.LatentDirichletAllocation([...])
decomposition.MiniBatchDictionaryLearning([...])
decomposition.MiniBatchSparsePCA([...])
decomposition.NMF([n_components, init, ...])
decomposition.PCA([n_components, copy, ...])
decomposition.SparsePCA([n_components, ...])
decomposition.SparseCoder(dictionary, *[, ...])
decomposition.TruncatedSVD([n_components, ...])

scikit-learn machine learning modules: https://scikit-learn.org/stable/modules/classes.html

1. Regression (Supervised)

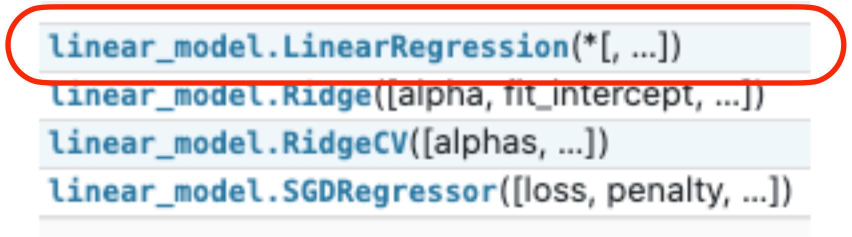
+

2. Classification (Supervised)



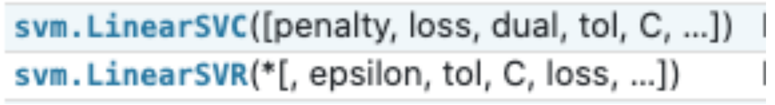
```
linear_model.LogisticRegression([penalty, ...])
linear_model.LogisticRegressionCV(*[, Cs, ...])
linear_model.PassiveAggressiveClassifier(*)
linear_model.Perceptron(*[, penalty, alpha, ...])
linear_model.RidgeClassifier([alpha, ...])
linear_model.RidgeClassifierCV([alphas, ...])
linear_model.SGDClassifier([loss, penalty, ...])
```

Logistic**Regression** is a **classifier** (today)

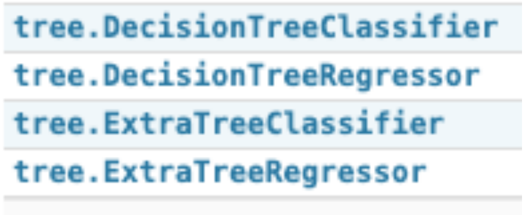


```
linear_model.LinearRegression(*[, ...])
linear_model.Ridge([alpha, fit_intercept, ...])
linear_model.RidgeCV([alphas, ...])
linear_model.SGDRegressor([loss, penalty, ...])
```

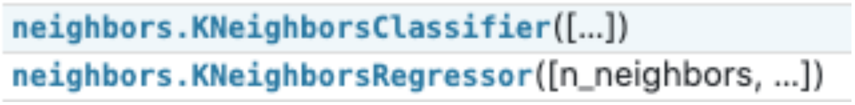
LinearRegression is a regressor (learned previously)



```
svm.LinearSVC([penalty, loss, dual, tol, C, ...])
svm.LinearSVR(*[, epsilon, tol, C, loss, ...])
```



```
tree.DecisionTreeClassifier
tree.DecisionTreeRegressor
tree.ExtraTreeClassifier
tree.ExtraTreeRegressor
```



```
neighbors.KNeighborsClassifier([...])
neighbors.KNeighborsRegressor([n_neighbors, ...])
```

scikit-learn machine learning modules: https://scikit-learn.org/stable/modules/classes.html