



```
pipe = Pipeline([
    ("scale", StandardScaler()),
    ("model", KNeighborsRegressor())
])
```

Load Dataset |
from sklearn.datasets import load_dataset

Pre Processing |
from sklearn.preprocessing import NAME |
NAME -> StandardScaler / MinMaxScaler / MaxAbsScaler / RobustScaler

Pipeline -> to sequence Data Pre-Processing steps
from sklearn.pipeline import Pipeline

Model Selection
Params of estimator |
i.e. params = classifier.get_params()
So you can know how many params you can tweak

1. from sklearn.preprocessing import **QuantileTransformer**

Create a QuantileTransformer instance
quantile_transformer = QuantileTransformer(output_distribution='uniform') # You can also choose 'normal' as the output distribution

Fit and transform your data
transformed_data = quantile_transformer.fit_transform(your_data)

2. **Converting categorical feature into a one-hot or binary encoding format.**

from sklearn.preprocessing import OneHotEncoder

Create an OneHotEncoder instance
encoder = OneHotEncoder()

Fit and transform your categorical data
encoded_data = encoder.fit_transform(your_categorical_data)

Evaluating the performance of machine learning models.
from sklearn.metrics import some_metric

Meta-Estimator	Description
Voting Classifiers/Regressors	
VotingClassifier	Combines the predictions of multiple classifiers (e.g., Decision Trees, Random Forests, Support Vector Machines) by majority voting or weighted voting.
VotingRegressor	Combines the predictions of multiple regressors in a similar manner.
Stacking (Stacked Generalization)	
StackingClassifier	Trains a meta-model that takes predictions from multiple base models as input and makes the final prediction.
StackingRegressor	Performs a similar task for regression problems.
Bagging (Bootstrap Aggregating)	
BaggingClassifier	Uses bootstrapped samples of the training data to train multiple base models and combines their predictions.
BaggingRegressor	Applies bagging to regression tasks.
Boosting	
AdaBoostClassifier	Iteratively adjusts the weights of the training samples to focus on the samples that are difficult to classify or predict.
AdaBoostRegressor	Performs a similar task for regression problems.
GradientBoostingClassifier	Builds an ensemble of decision trees in a stage-wise manner to improve accuracy.
GradientBoostingRegressor	Performs a similar task for regression problems.
Random Forest	
RandomForestClassifier	Constructs an ensemble of decision trees, where each tree is trained on a random subset of the data and features.
RandomForestRegressor	Applies random forest techniques to regression tasks.