Pipeline Scikit-learn

Setup a machine learning pipeline

1. Scaler: pre-processing data, i.e., transform the data to zero mean and unit variance using the StandardScaler().
2. Feature selector: Use VarianceThreshold() for discarding features whose variance is less than a certain defined threshold.
3.Classifier: KNeighborsClassifier(), which implements the k-nearest neighbor classifier and selects the class of the majority k points, which are closest to the test example.

```
from sklearn.pipeline import Pipeline
pipe = Pipeline([
        ('scaler', StandardScaler()),
        ('selector', VarianceThreshold()),
        ('classifier', KNeighborsClassifier())
])
pipe.fit(X_train, y_train)
```

Optimizing and Tuning Pipeline

1. We can search for the best scalers. Instead of just the StandardScaler(), we can try MinMaxScaler(), Normalizer() and MaxAbsScaler().
2. We can search for the best variance threshold to use in the selector, i.e., VarianceThreshold(). Specified a list of values [0, 0.0001, 0.001, 0.5] to choose from.
3. We can search for the best value of k for the KNeighborsClassifier(). Different values are specified for the n_neighbors, p and leaf_size parameters.

```
from sklearn.model_selection import GridSearchCV

parameters = {
                    'scaler': [StandardScaler(), MinMaxScaler(),Normalizer(), MaxAbsScaler()],
                    'selector__threshold': [0, 0.001, 0.01],
                    'classifier__n_neighbors': [1, 3, 5, 7, 10],
                    'classifier__p': [1, 2],
                    'classifier__leaf_size': [1, 5, 10, 15]
}
grid = GridSearchCV(pipe, parameters, cv=2).fit(X_train, y_train) # cv – cross validation
```