Scikit-learn
Scikit-learn is an open-source Python library that implements a range of machine learning, preprocessing, cross-validation and visualization algorithms using a unified interface.

A Basic Example
```python
>>> from sklearn import neighbors, datasets, preprocessing
>>> iris = datasets.load_iris()
>>> X = iris.data[:, :2]  # we only take the first two features.
>>> y = iris.target
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
>>> scalar = StandardScaler().fit(X_train)
>>> X_train = scalar.transform(X_train)
>>> X_test = scalar.transform(X_test)
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)
>>> knn.fit(X_train, y_train)
>>> y_pred = knn.predict(X_test)
```

Unsupervised Learning Estimators
```
Principal Component Analysis (PCA)
>>> from sklearn.decomposition import PCA
>>> pca = PCA(n_components=0.95)

K Means
>>> from sklearn.cluster import KMeans
>>> kmeans = KMeans(n_clusters=3, random_state=0)
```

Encoding Categorical Features
```
>>> from sklearn.preprocessing import LabelEncoder
>>> y = ['M', 'F', 'M', 'F', 'M', 'F', 'M', 'F', 'F']
>>> encoder = LabelEncoder()
>>> encoder.fit(y)
```

Imputing Missing Values
```
>>> from sklearn.preprocessing import Imputer
>>> imp = Imputer(missing_values=0, strategy='mean', axis=1)
>>> imp.fit_transform(X_train)
```

Generating Polynomial Features
```
>>> from sklearn.preprocessing import PolynomialFeatures
>>> poly = PolynomialFeatures(5)
```

Accuracy Score
```
>>> from sklearn.metrics import accuracy_score
```

Regression Metrics
```
Mean Absolute Error
>>> from sklearn.metrics import mean_absolute_error
```

Mean Squared Error
```
>>> from sklearn.metrics import mean_squared_error
```

R² Score
```
>>> from sklearn.metrics import r2_score
```

Mean Squared Log Error
```
>>> from sklearn.metrics import mean_squared_log_error
```

Homogeneity
```
>>> from sklearn.metrics import homogeneity_score
```

Adjusted Rand Index
```
>>> from sklearn.metrics import adjusted_rand_score
```

V-measure
```
>>> from sklearn.metrics import v_measure_score
```

Precision, recall, f-score and support
```
>>> from sklearn.metrics import classification_report
```

Confusion Matrix
```
>>> from sklearn.metrics import confusion_matrix
```

Classification Report
```
>>> from sklearn.metrics import classification_report
```

Cross-Validation
```
>>> from sklearn.cross_validation import cross_val_score
```

Randomized Parameter Optimization
```
>>> from sklearn.grid_search import RandomizedSearchCV
```

Tune Your Model

Grid Search
```
>>> from sklearn.grid_search import GridSearchCV
>>> params = ["n_neighbors": [3, 0.5, 2], "weights": ["uniform", "distance"]]
>>> grid = GridSearchCV estimator=knn, param_grid=params
>>> grid.fit(X_train, y_train)
>>> grid.best_estimator_.n_neighbors
```

R² Score
```
>>> from sklearn.metrics import r2_score
```

R² Score
```
>>> from sklearn.metrics import r2_score
```

Mean Absolute Error
```
>>> from sklearn.metrics import mean_absolute_error
```

Mean Squared Error
```
>>> from sklearn.metrics import mean_squared_error
```

R² Score
```
>>> from sklearn.metrics import r2_score
```