- Trained with a set of features to predict a target value
- Consists of regression and classification
- Metrics determine how close the prediction is to the true target value, measuring model success

```
from sklearn.linear_model import LinearRegression

features = [[5, 3.4, 6], [1, 0.4, 10], [2, 0.1, 1]]
target = [1.4, 0.5, 1]

reg = LinearRegression().fit(features, target)

# Score model with features and target
print(reg.score(features, target))
# Predict new values based on features
print(reg.predict(features))


# output
1.0
[1.4 0.5 1. ]
```

## Unsupervised ML

- Organizes and represents data to provide additional structure
- Data does not use a target value to fit a model
- An example of unsupervised ML is clustering
- Metrics determine how well it groups the data to define success

```
from sklearn.cluster import KMeans

features = [[5, 3.4, 6], [1, 0.4, 10], [2, 0.1, 1]]
# Cluster into 3 groups
kmeans = KMeans(n_clusters=3).fit(features)

# Prints which data point belongs to which cluster group
print(kmeans.labels_)
# Prints the metric of how clustering performed
print(kmeans.inertia_)

# output
[0 2 1]
0.0
```

## Additional Resources

- To learn more about the Scikit-learn API for `LinearRegression`, visit the **documentation**.
- To learn more about the Scikit-learn API for `Kmeans`, visit the **documentation**.
- To learn more about supervised and unsupervised ML, check out this **blog**.