Model Selection: Takeaways ⇪

Syntax

- Forward selection via the `SequentialFeatureSelector` class:

```python
from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
forward_lm = SequentialFeatureSelector(estimator=lm,
                                       n_features_to_select=2,
                                       direction="forward")
forward_lm.fit(X, y)
```

- Backward selection via the `SequentialFeatureSelector` class:

```python
backward_lm = SequentialFeatureSelector(estimator=lm,
                                        n_features_to_select=2,
                                        direction="backward")
backward_lm.fit(X, y)
```

- Calculating the AIC:

```python
def AIC(p, L):
    return 2 * p - 2 * np.log(L)
```

- Implementing PCA for dimension reduction:

```python
from sklearn.decomposition import PCA
pca = PCA(n_components=3)
pca.fit(X)
```

Concepts

- **sequential feature selection** is the process of choosing or eliminating a feature iteratively until a reasonable set is chosen.
- **forward selection** starts with an intercept-only model and adds features to it, while **backward selection** starts with all of the predictors and removes them one-by-one.
- There are several criteria, such as the AIC, BIC, and R^2, that can also act as criteria for us to compare models. These may be used in situations when there isn't much data for a test set.
- High-dimensional datasets can be problematic because certain models are difficult or impossible to implement when the number of features surpasses the number of observations. When this happens, we need to take approaches that reduce the number of features that we consider. Principal component analysis is one such approach.

Resources

- `scikit-learn` official documentation
- `scikit-learn` vignette on sequential feature selection
- `SequentialFeatureSelector` class

- **PCA** class
- More on AIC
- More on PCA