Cross-Validation: Takeaways

Syntax

- Using K-fold cross-validation using the `cross_val_score()` function:

```
from sklearn.model_selection import cross_val_score
from linear_model import LinearRegression
model = LinearRegression()
fold_mses = cross_val_score(model, X, y, cv = 5, scoring = "neg_mean_squared_error")
```

- Implementing LOOCV:

```
from sklearn.model_selection import cross_val_score
from linear_model import LinearRegression
model = LinearRegression()
n = data.shape[0] # number of rows
fold_mses = cross_val_score(model, X, y, cv = n, scoring = "neg_mean_squared_error")
```

Concepts

- **K-fold cross-validation**: splitting the data up into `k` folds that can be combined in different ways to produce different estimates of test error. This is useful for checking the distribution of test errors and reducing the influence of outliers on these estimates.
- **LOOCV**: an extreme form of K-fold cross-validation where the test fold is a single observation. Generally, we don't advise its use due to computational demand.
- **Bias**: how much an estimated value deviates from some true value of interest.
- **Variance**: how much an estimated value can vary depending on the data that was used to calculate it.
- **Bias-Variance Trade-Off**: the trade-off between bias and variance when choosing the number of folds to use in K-fold cross-validation. There are many forms of the trade-off, one of which appears for this topic.

Resources

- `scikit-learn` official documentation
- `scikit-learn` vignette on combining hyperparameter tuning and cross-validation
- More on the bias-variance trade-off
- More on cross-validation