Model Fitting, Predictions, and Error

01/23/2018

housing — Housing Training Set

redefined ↓

housing — Housing Training Set Predictors

housing_labels — Housing Training Set Labels

housing_prepared — Housing Training Set Predictors Standardized

Linear Regression

```
lin_reg.fit(housing_prepared, housing_labels)
```

(housing_prepared: Train Predictors; housing_labels: Train Labels)

↳ With this, we observe relationship between training set predictors and training set labels, deriving best coefficients.

```
housing_predictions = lin_reg.predict(housing_prepared)
```

(housing_prepared: Train Predictors)

Here, we're calculating the error on the entire training set. Usually, it isn't good to fit model to entire training set, as the model, in attempting to minimize training set error, will

overfit the data. Usually, calculating the error on the training set of a model fit to the training set will give a lower error than one would get on test set examples, due to overfitting.

However,

as we later discover when using a CV set, in linear regression, overfitting is very small since we ultimately only use a line to fit the data.

Error

Training set using alg.fit on training: $68,628

CV set using alg. fit on training: $69,052

(Not a big difference since lin. reg isn't prone to overfit)

```
lin_mse = mean_squared_error(housing_labels, housing_predictions)
```

housing_labels: Train labels; housing_predictions: Train Predictions

Decision Trees

```
tree_reg.fit(housing_prepared, housing_labels)
```

housing_prepared: Train Predictors; housing_labels: Train labels

```
housing_predictions = tree_reg.predict(housing_prepared)
```

housing_prepared: Train Predictors

Just like w/ lin. reg. above, we fit our model on the training set (w/o regularization).

However,

Unlike lin. regression, decision trees are highly susceptible to overfitting.

Error

Training set using alg.fit on training: $0 (!! High overfit...)

CV set using alg. fit on training: $71,379

↳ So, as we see here, huge overfitting by decision trees.
Even w/ CV, lin reg > decision trees here.

Random Forest Regresser

```
forest_reg.fit(housing_prepared, housing_labels)
```

housing_prepared: train predictors
housing_labels: train labels

Fit algorithm to training data

```
housing_predictions = forest_reg.predict(housing_prepared)
```

Again, making predictions for training data based on model fit on training data is a recipe for overfit.

Error

Training set using alg.fit on training: $21,941

CV set using alg. fit on training: $52,564

Still an overfit, but best CV performance yet!