Multivariate K-Nearest Neighbors: Takeaways

Syntax

- Using the preProcess argument in the train() function to normalize the data:

```
two_feature_knn_model <- train(tidy_price ~ accommodates + bedrooms,
                               data = train_listings,
                               method = "knn",
                               trControl = train_control,
                               preProcess = c("center", "scale"))
```

- Using pivot_longer() to convert the data into long format:

```
library(tidyr)
test_listings <- test_listings %>%
    pivot_longer(
        cols = two_feature_predictions:three_feature_predictions,
        names_to = "model",
        values_to = "predictions"
    )
```

- Using

```
group_by()
```

and

```
summarize()
```

to split the data by categories and produce summary calculations for each dataset:

```
sq_error_by_model <- test_listings %>%
    group_by(model) %>%
    summarize(
        avg_sq_error = mean(sq_error)
    )
```

Concepts

- The k-nearest neighbors algorithm only works with numerical values, so this is an important constraint to keep in mind if you are considering using it.
- **Normalization** helps us prevent problems with differences of scale in our data. In order to normalize a column, we need to subtract the mean of the column from each value in it and divide by the standard deviation of the column.
- When we have multiple columns containing the same type of information, we say that the dataset is in **wide format**, which is easier for humans to read. In order to use some of the useful functions in the `tidyverse` libraries, we need to convert it into **long format**, where all of these columns are "stacked" on each other into just one column. The `pivot_longer()` function helps us with this process.
- The `group_by()` and `summarize()` functions can be used in tandem to create short summaries of your dataset, which is especially useful in machine learning model comparison.

Resources

- `caret` 's documentation
- Documentation on the `group_by()` function