

DATA 3464: Fundamentals of Data Processing

---

# Categorical Data

Charlotte Curtis

January 22, 2026

# Topic overview

---

- Exploring categorical data
- Categorical data encoding strategies

## Resources used:

- [Feature Engineering Chapter 5](#)
- [Scikit-learn User Guide \(7.3\)](#)

# What is categorical data?

---

- Samples can take on one of several discrete values or groups
  - **Nominal**: no particular order to the groups
  - **Ordinal**: groups relate to each other in a specific order
- Categories can be represented as strings *or* numeric types
  - Domain knowledge is necessary!

*Let's take a few minutes to brainstorm some examples*

# Exploring categorical data

---

We've already done some of this, but some ideas to consider:

- `pandas.DataFrame.value_counts()` - how many of each category?
- Use category to group, then compute summary stats
- Plot color per category
- Scatter plot with jitter

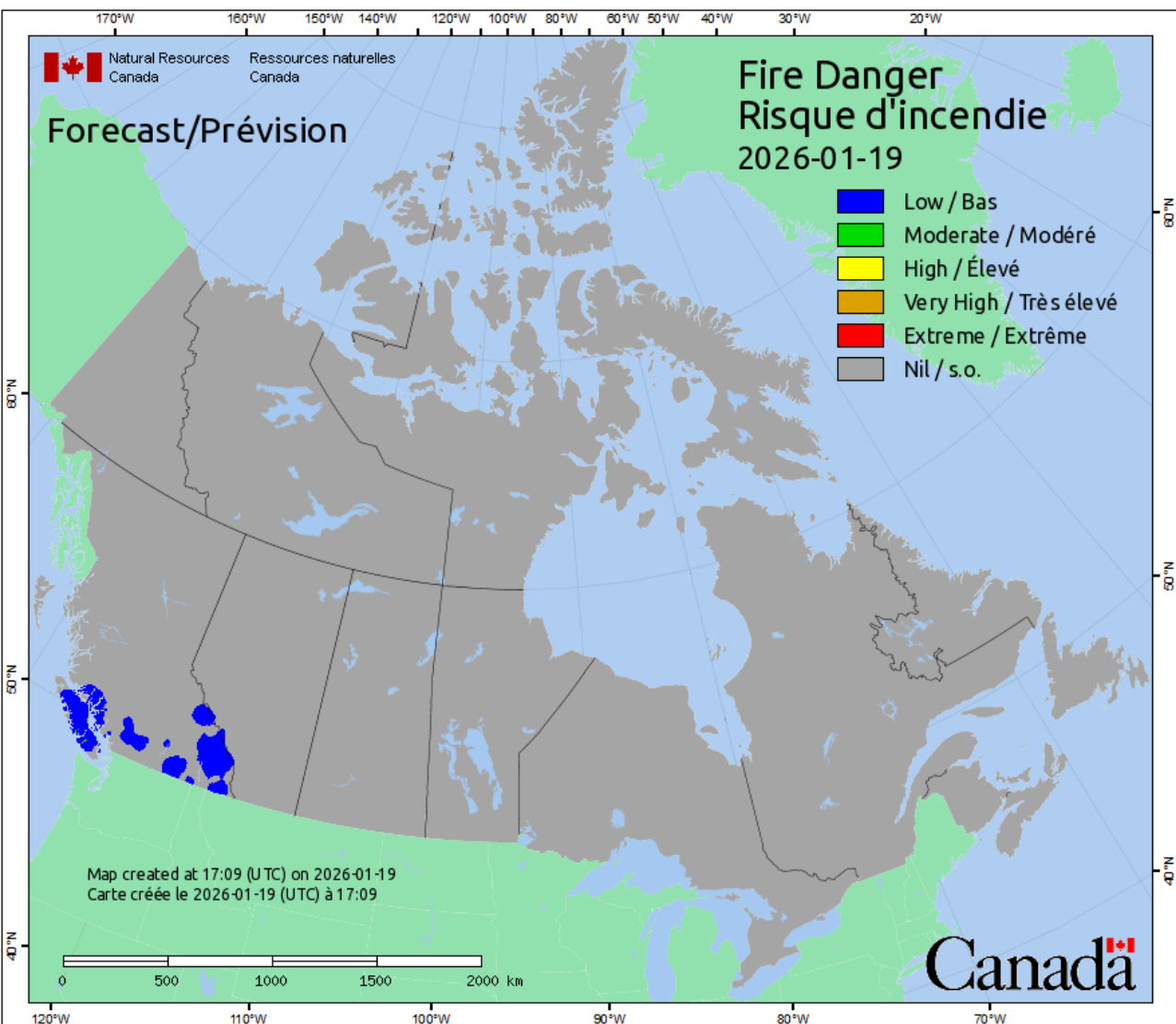
# Representing categorical data

---

- ~~Tree-based models can handle string-based categories as-is~~ Not in Scikit-Learn
- Consider:
  - Ordinal or nominal?
  - How many possible categories (cardinality)?
  - Any chance new ones might show up?

*How could we encode the examples?*

*What are the benefits and drawbacks of each method?*



# Ordinal encoding

Category	Feature
Nil	0
Low	1
Moderate	2
High	3
Very High	4
Extreme	5

# Nominal categories: one-hot encoding

---

- Categories have no natural relationship
- Create  $k$  new features from  $k$  categories, very sparse matrix

Animal		cat	dog	rabbit
cat	→	1	0	0
dog	→	0	1	0
rabbit	→	0	0	1

# Another approach: target encoding

---

- Basic concept: replace the category with the mean of the target
- Essential to avoid data leakage!
- Example: predicting weight of animal

Animal		mean_kg
cat	→	4.1
dog	→	15.4
rabbit	→	2.2

**Where we left off on January 22**

---

# Getting fancy with feature hashing

---

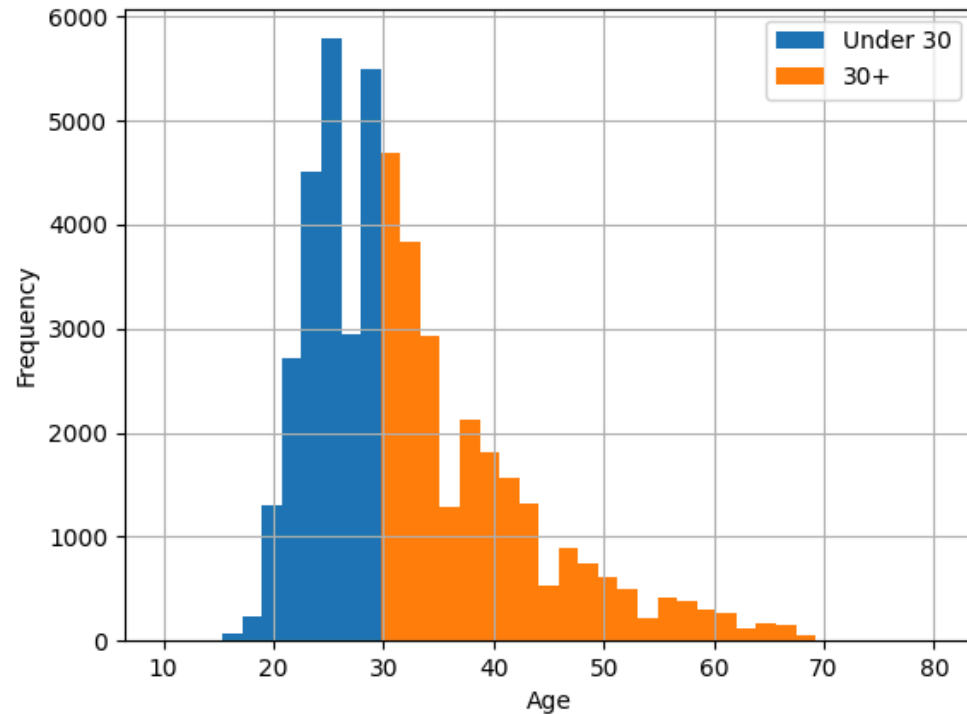
- Good if you have too many categories, or combinations of categories
- Converts each category into a fixed-length feature vector

Animal		A_0	A_1	A_2	...	A_16
cat	→	1	0	0	...	1
dog	→	0	1	0	...	1
rabbit	→	1	0	1	...	0

*Even fancier methods exist, like [supervised encoding methods](#) which basically train a model-within-a-model*

# From numbers to categories: discretization

---



- Sometimes numeric data is better represented as categorical
- Still needs encoding strategy
- Can introduce nonlinear relationships
- Like everything, contextual

# Advantages/disadvantages of various methods

---

## Ordinal

- + Compact (1 column/feature)
- + Only useful for naturally ordered data
- Imposes (linear) relationship
- Difficult to deal with novel categories

## One-hot

- + No assumptions made about relationships
- + Novel features implied by 0
- Memory intensive
- Not good for high cardinality
- Potential collinearity issues

# Advantages/disadvantages continued

---

## Target encoding

- + Powerful if strong predictor
- + Compact
- Doesn't work for unsupervised
- Central tendency poor measure for categories with few samples
- Difficult to deal with novel categories

## Feature hashing

- + Natively handles novel categories
- + Compact (fixed columns/features)
- + Good for high cardinality
- Risk of hash collision
- Loss of interpretability/meaning

# Coming up next

---

- Assignment 1 January 30th
- Lab: practice with modelling process

*Feature Engineering Chapter 5*