CPSC 330 Lecture 7: Linear models

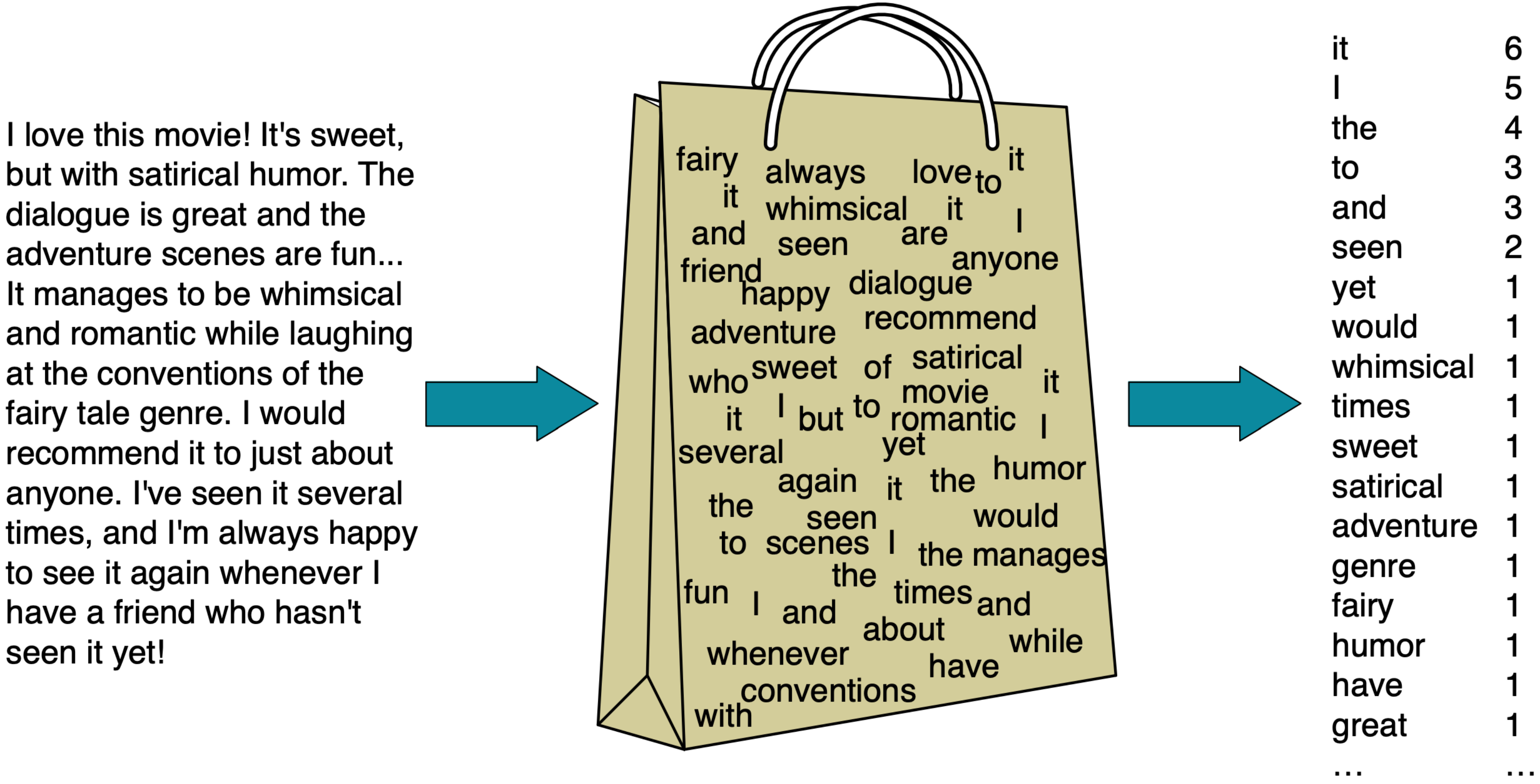
Varada Kolhatkar

Announcements

- Important information about midterm 1
  - https://piazza.com/class/m01ukubppof625/post/249
- Change of my office hours
  - Thursdays from 2 to 3 in my office ICCS 237
- Where to find slides?
  - https://kvarada.github.io/cpsc330-slides/lecture.html
- HW3 is due next week Tuesday, Oct 1st, 11:59 pm.
  - You can work in pairs for this assignment.

Recap: Dealing with text features

- Preprocessing text to fit into machine learning models using text vectorization.
- Bag of words representation



I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Recap: `sklearn CountVectorizer`

- Use `scikit-learn`'s `CountVectorizer` to encode text data
- `CountVectorizer`: Transforms text into a matrix of token counts
- Important parameters:
  - `max_features`: Control the number of features used in the model
  - `max_df`, `min_df`: Control document frequency thresholds
  - `ngram_range`: Defines the range of n-grams to be extracted
  - `stop_words`: Enables the removal of common words that are typically uninformative in most applications, such as "and", "the", etc.

Recap: Incorporating text features in a machine learning pipeline

```python
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline

text_pipeline = make_pipeline(
    CountVectorizer(),
    SVC()
)
```

(iClicker) Exercise 6.2

iClicker cloud join link: **https://join.iclicker.com/VYFJ**

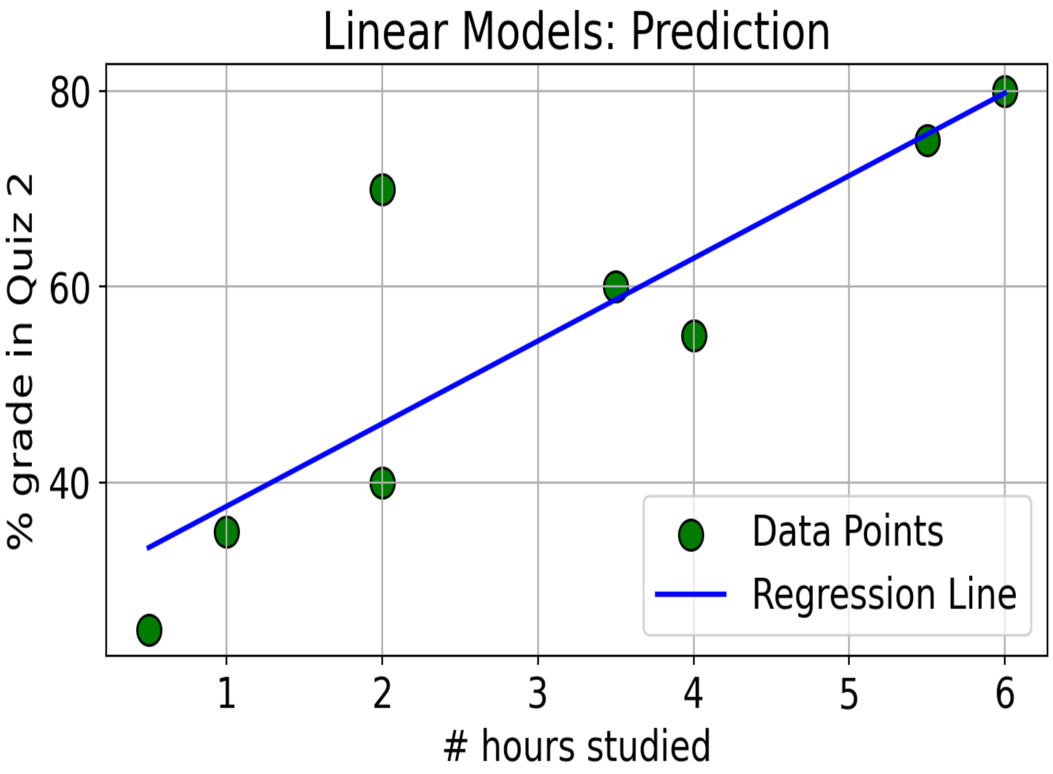
Select all of the following statements which are TRUE.

- a. `handle_unknown="ignore"` would treat all unknown categories equally.
- b. As you increase the value for `max_features` hyperparameter of `CountVectorizer` the training score is likely to go up.
- c. Suppose you are encoding text data using `CountVectorizer`. If you encounter a word in the validation or the test split that’s not available in the training data, we’ll get an error.
- d. In the code below, inside `cross_validate`, each fold might have slightly different number of features (columns) in the fold.

```
pipe = (CountVectorizer(), SVC())
cross_validate(pipe, X_train, y_train)
```

Linear models

- Linear models make an assumption that the relationship between X and y is linear.
- In this case, with only one feature, our model is a straight line.
- What do we need to represent a line?
  - Slope (w_1): Determines the angle of the line.
  - Y-intercept (w_0): Where the line crosses the y-axis.



- Making predictions
  - $y_{hat} = w_1 \times \# \text{ hours studied} + w_0$

`Ridge` vs. `LinearRegression`

- Ordinary linear regression is sensitive to **multicolinearity** and overfitting
- Multicolinearity: Overlapping and redundant features. Most of the real-world datasets have colinear features.
- Linear regression may produce large and unstable coefficients in such cases.
- `Ridge` adds a parameter to control the complexity of a model. Finds a line that balances fit and prevents overly large coefficients.

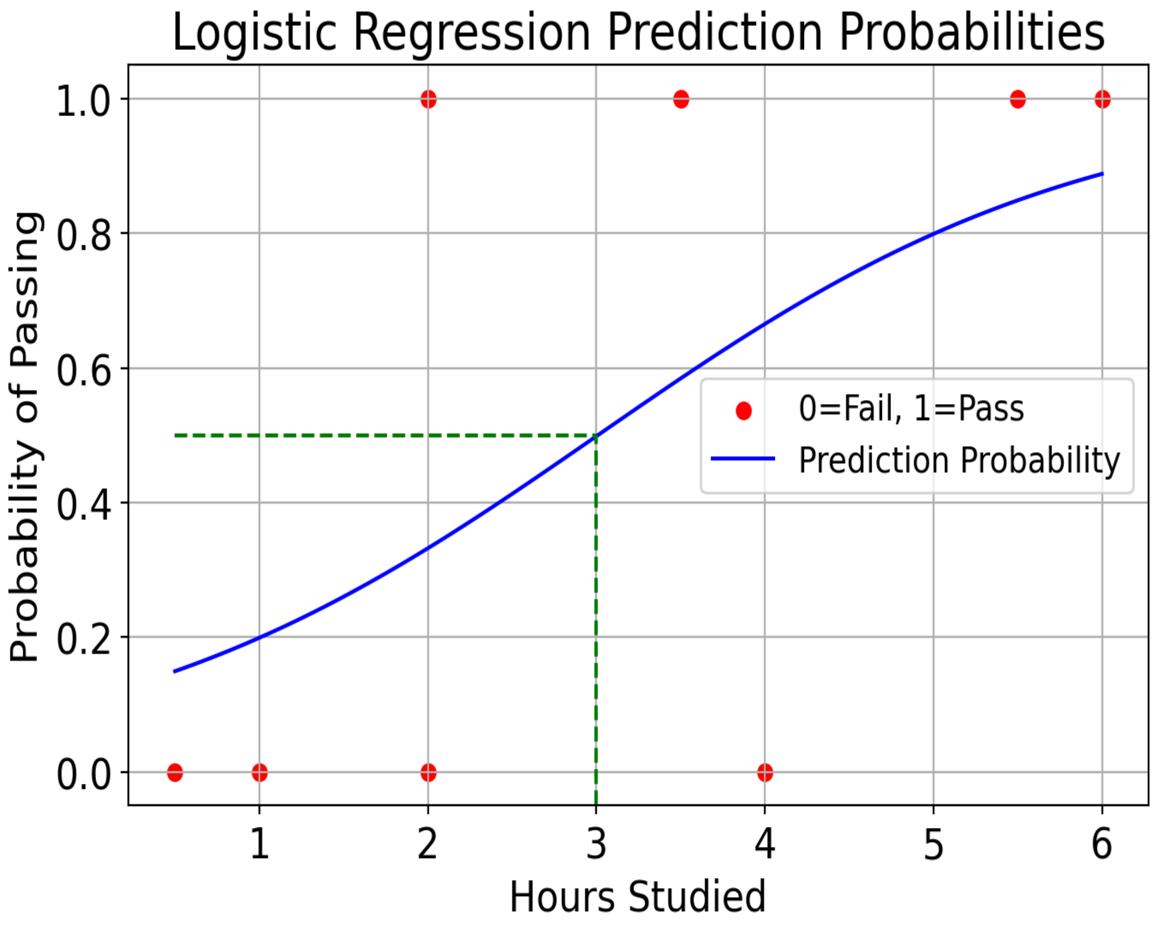
When to use what?

- `LinearRegression`
  - When interpretability is key, and no multicollinearity exists
- `Ridge`
  - When you have **multicollinearity** (highly correlated features).
  - When you want to prevent **overfitting** in linear models.
- **In this course, we’ll use `Ridge`.**

Logistic regression

- Suppose your target is binary: pass or fail
- Logistic regression is used for such binary classification tasks.
- Logistic regression predicts a probability that the given example belongs to a particular class.
- It uses **Sigmoid function** to map any real-valued input into a value between 0 and 1, representing the probability of a specific outcome.
- A threshold (usually 0.5) is applied to the predicted probability to decide the final class label.

Logistic regression: Decision boundary



- The decision boundary is the point on the x-axis where the corresponding predicted probability on the y-axis is 0.5.

Parametric vs. non-Parametric models (high-level)

- Imagine you are training a logistic regression model. For each of the following scenarios, identify how many parameters (weights and biases) will be learned.
- Scenario 1: 100 features and 1,000 examples
- Scenario 2: 100 features and 1 million examples

Parametric vs. non-Parametric models (high-level)

Parametric

- Examples: Logistic regression, linear regression, linear SVM
- Models with a fixed number of parameters, regardless of the dataset size
- Simple, computationally efficient, less prone to overfitting
- Less flexible, may not capture complex relationships

Non parametric

- Examples: KNN, SVM RBF, Decision tree with no specific depth specified
- Models where the number of parameters grows with the dataset size. They do not assume a fixed form for the functions being learned.
- Flexible, can adapt to complex patterns
- Computationally expensive, risk of overfitting with noisy data

(iClicker) Exercise 7.1

iClicker cloud join link: **https://join.iclicker.com/VYFJ**

Select all of the following statements which are TRUE.

- a. Increasing the hyperparameter `alpha` of `Ridge` is likely to decrease model complexity.
- b. `Ridge` can be used with datasets that have multiple features.
- c. With `Ridge`, we learn one coefficient per training example.
- d. If you train a linear regression model on a 2-dimensional problem (2 features), the model will learn 3 parameters: one for each feature and one for the bias term.

(iClicker) Exercise 7.2

iClicker cloud join link: **https://join.iclicker.com/VYFJ**

Select all of the following statements which are TRUE.

- a. Increasing logistic regression's `C` hyperparameter increases model complexity.
- b. The raw output score can be used to calculate the probability score for a given prediction.
- c. For linear classifier trained on d features, the decision boundary is a $\mathrm{d} - 1$-dimensional hyperparlane.
- d. A linear model is likely to be uncertain about the data points close to the decision boundary.

Class demo