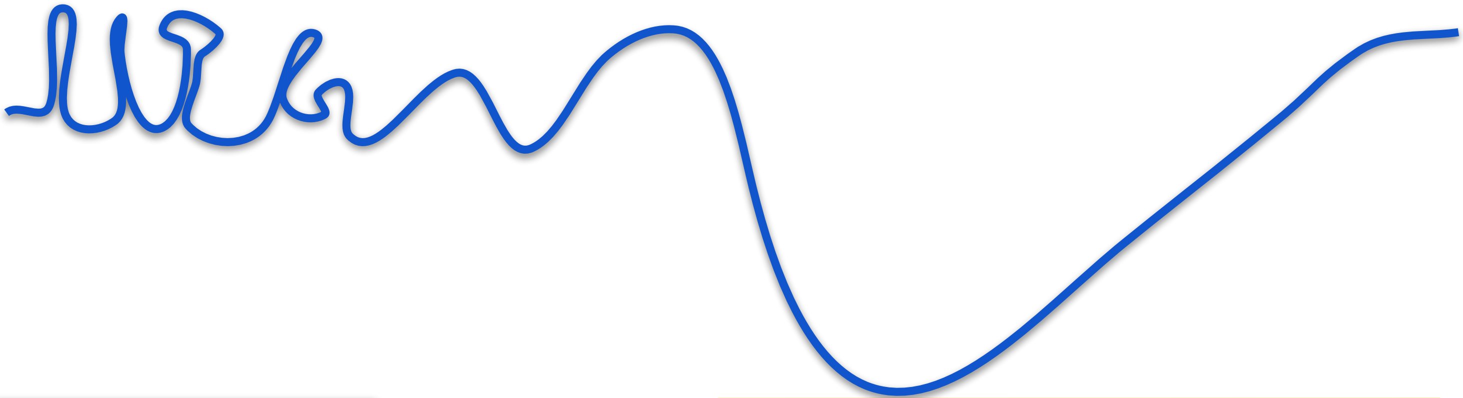
Computing with Signals



DA 623

Jan - May 2024

IIT Guwahati

Instructors: Neeraj Sharma

Lecture-11_misc_app

Sentiment classifier using NLP

Understanding Human Emotions:

- People express their thoughts and feelings through words.
- We can capture these expressions in text, and just like humans can recognize sentiments in a conversation, a sentiment classifier can be trained to do the same using the power of computers.

Sentiment classifier using NLP

Use case:

- **Social Media and Reviews:** Surrounded by vast amounts of text every day, especially on social media and review platforms.
- **Helpful for Businesses:** Understand customer feedback. By analyzing reviews and comments, companies can quickly grasp what customers like or dislike about their products or services.
- **Personalization in Services:** Streaming platform use it to recommend movies or shows based on what users enjoy.
- **Efficiency in Customer Support:** Businesses can prioritize and address negative feedback more promptly, leading to better customer satisfaction.
- **Filtering Out Hate Speech:** Filter out hate speech or harmful content. It helps create a safer online environment by identifying and taking action against harmful language.

Sentiment classifier using NLP

- **Focus: Tweets sentiment classification**

TSATC: Twitter Sentiment Analysis Training Corpus

TSATC: Twitter Sentiment Analysis Training Corpus

Contains: The Twitter Sentiment Analysis Dataset contains 1,578,627 classified tweets, each row is marked as 1 for positive sentiment and 0 for negative sentiment (inside a csv file).

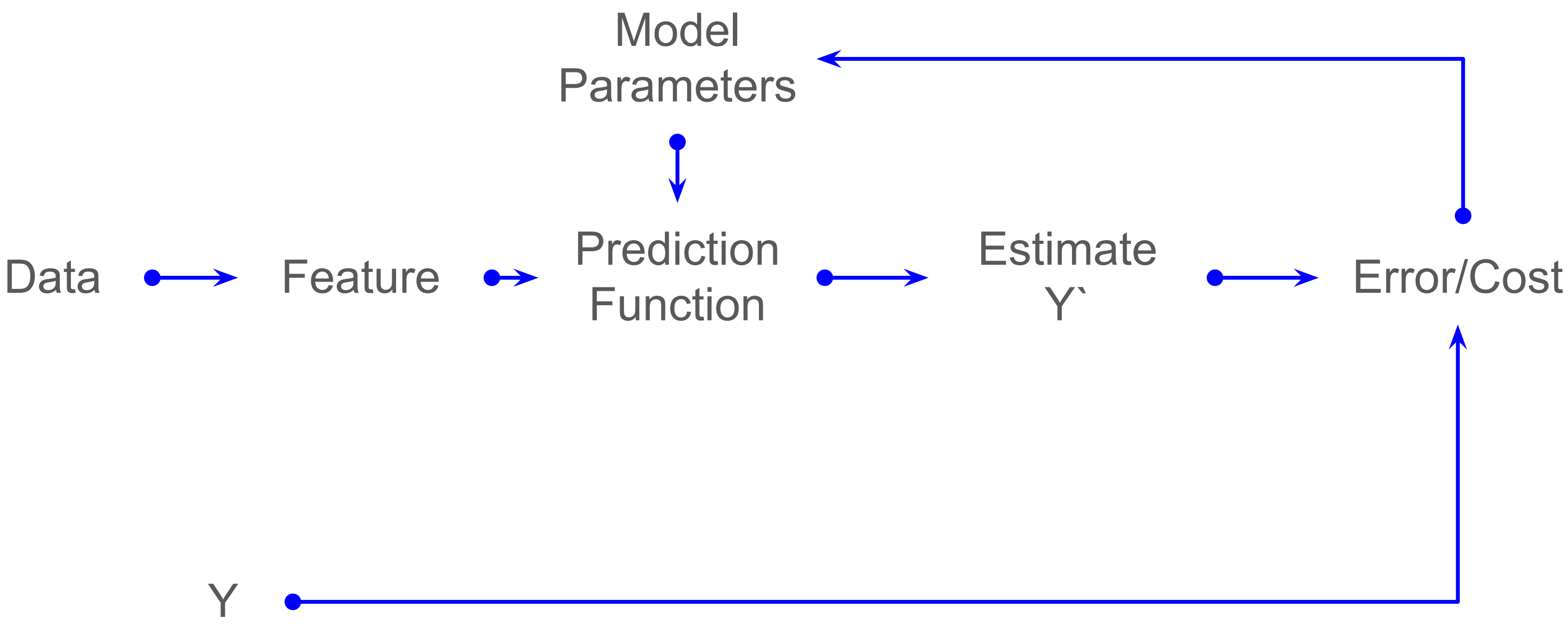
Accessible:

- It can be downloaded from: http://thinknook.com/wp-content/uploads/2012/09/Sentiment-Analysis-Dataset.zip
- Apache License

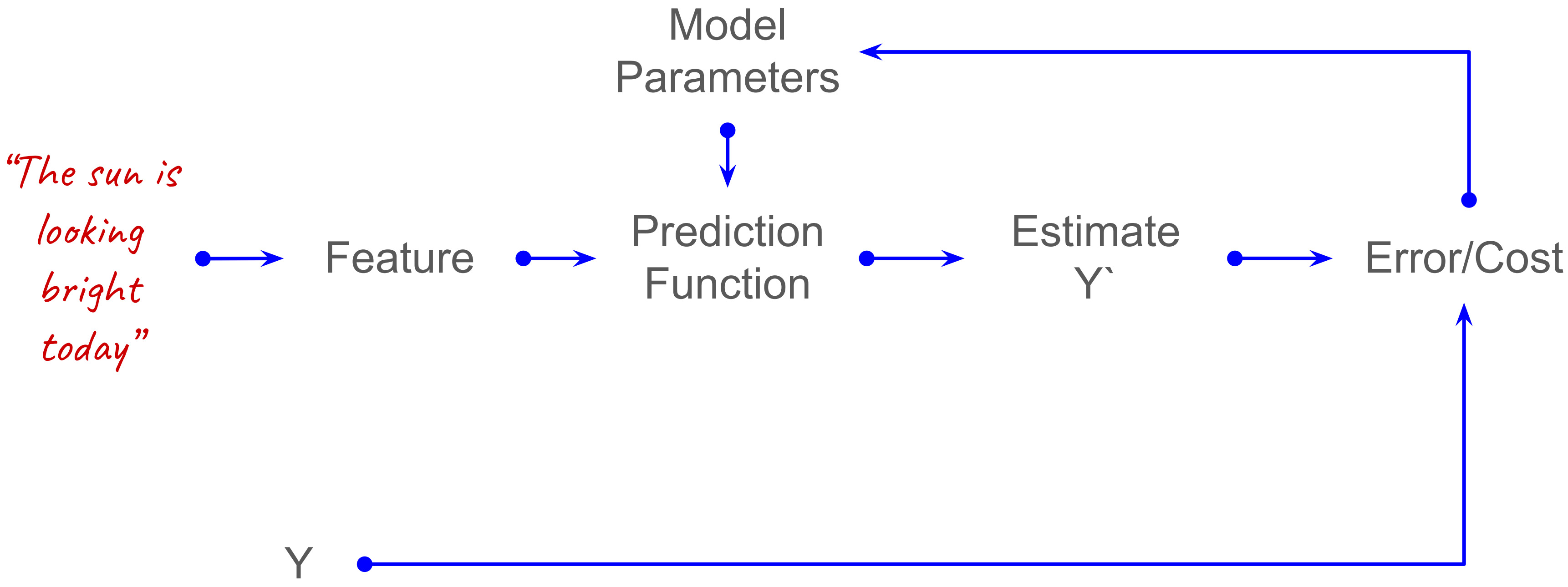
Data Source: The dataset is based on data from the following two sources:

- University of Michigan Sentiment Analysis competition on Kaggle
- Twitter Sentiment Corpus by Niek Sanders

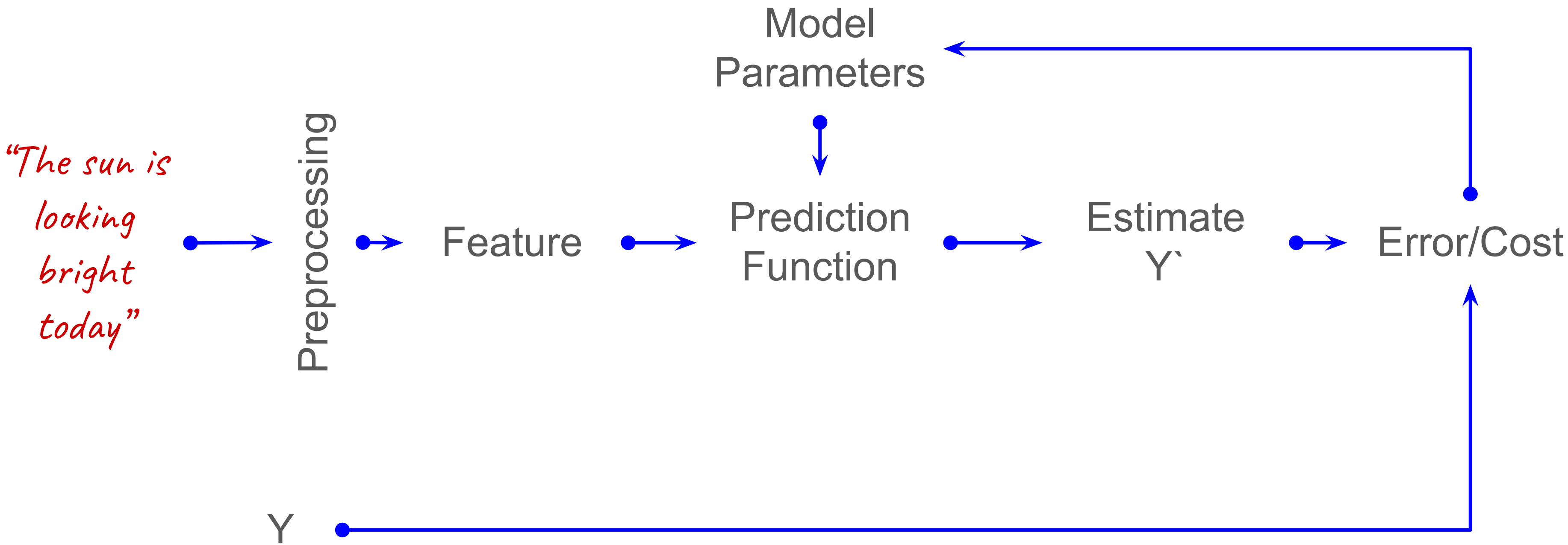
Methodology



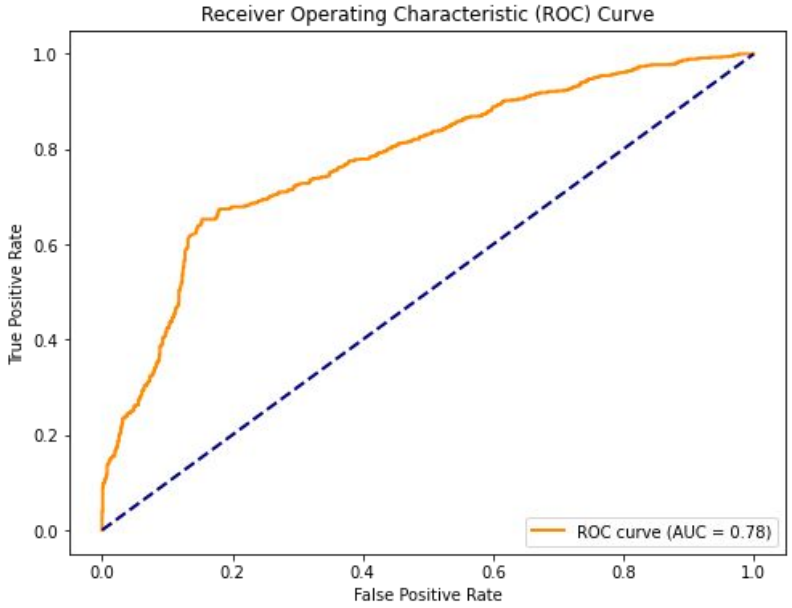
Methodology

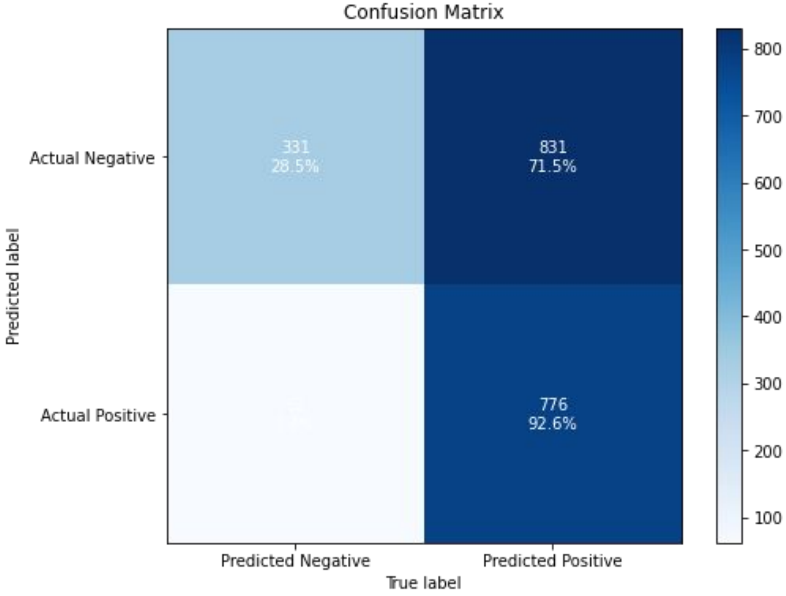


Methodology



Results






```
Input Text: 'Man skips a wedding celebration to rescue a dog stuck in an overflowing river..'
Predicted Sentiment: negative
```

Takeaways

Data Preprocessing is Crucial
Effective data preprocessing is essential for successful sentiment analysis. Steps such as text cleaning, stop word removal, and stemming contribute to better feature representation.

Feature Engineering Matters
Extracting meaningful features is crucial for sentiment analysis. In this example, we used word frequencies as features.

Model Evaluation Beyond Accuracy
While accuracy is a common metric, it might not be sufficient for imbalanced datasets. Confusion matrices, precision, recall, and ROC curves provide a more comprehensive understanding of a model's performance, especially when dealing with false positives and false negatives.

Class Imbalance Impacts Model Performance
Imbalanced class distribution can lead to challenges, especially when one class is underrepresented. Techniques like stratified sampling, class weighting, or resampling can help address these issues and improve model performance.

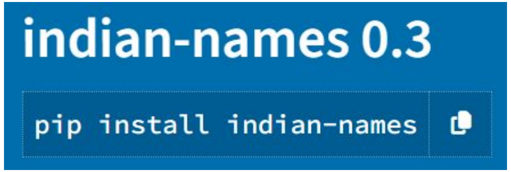
Designing and implementing a bi-gram model for predicting next character

shilp_

Given the sequence, pick the next character from your set.

[a b c d e f g h i j k l m n o p q r s t u v w x y z .]

Indian names dataset



Contains:

- first.male.names first.female.names last.names

Accessible:

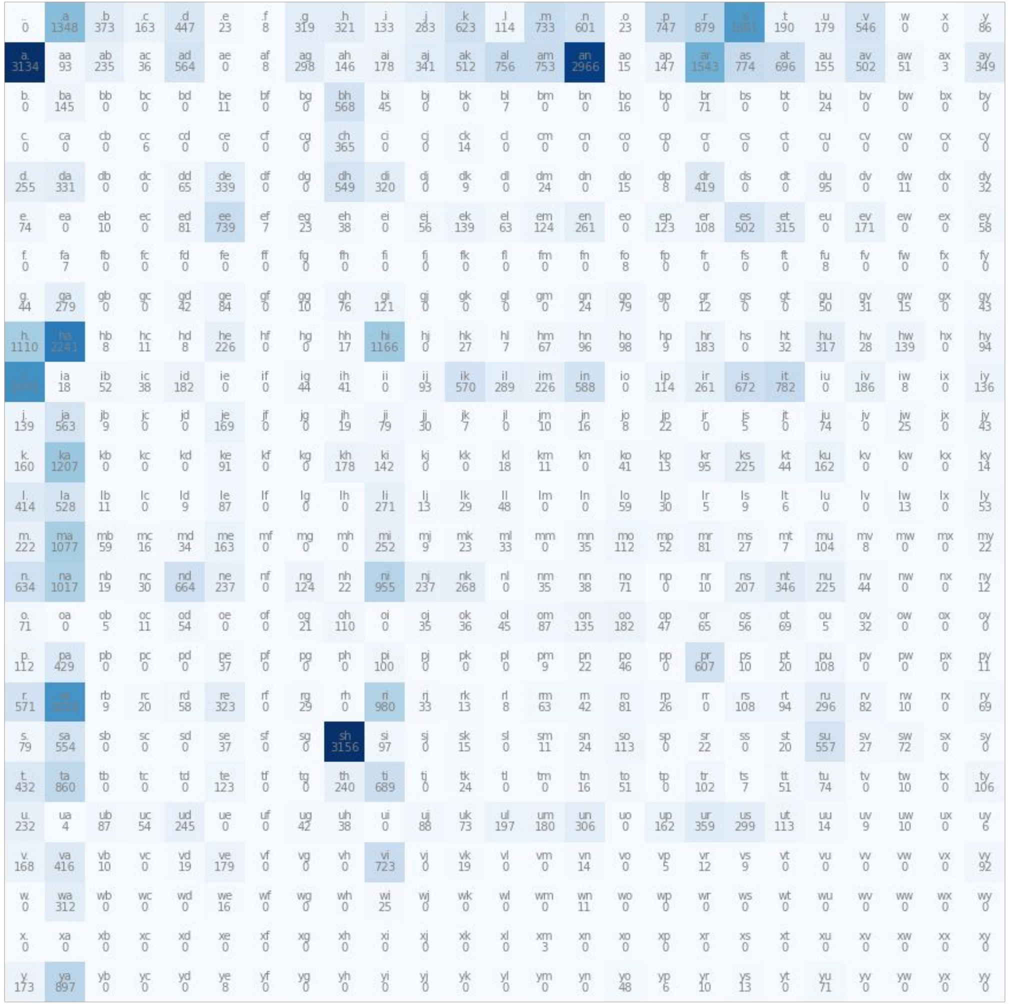
- This package is released under a BSD 3-Clause License.

Data Source:

- Names are compiled in the following files through various online sources available publically.

Steps

- Obtain all possible bi-grams from the data
- Get the count of each bi-gram
- Re-structure the count as a matrix
- Obtain conditional probabilities
- Sample from a multinomial using the conditional probabilities
- Create new names



Takeaways

Sequential Dependency:
The bi-gram model emphasizes the importance of sequential dependencies in language. This concept highlights the significance of context in language modeling.

N-gram Modeling:
N-gram models, including bi-gram models, are valuable tools for understanding and predicting patterns in sequential data. They provide a balance between simplicity and capturing local context.

Probability Estimation:
The model estimates conditional probabilities of characters given their preceding characters.

Takeaways

Model Evaluation:
The effectiveness of the bi-gram model is assessed through evaluation metrics, such as accuracy and perplexity.

Challenges and Limitations:
Despite its simplicity and effectiveness, the bi-gram model has limitations, such as the inability to capture long-range dependencies.

Application to Text Generation:
Bi-gram models can be applied to text generation tasks, where the goal is to generate coherent and contextually relevant sequences of characters or words.

Food for thought

- Can you use such an approach for written language identification?
- Can you implement the same using a neural network?
- How do you quantify the quality of the generated data?