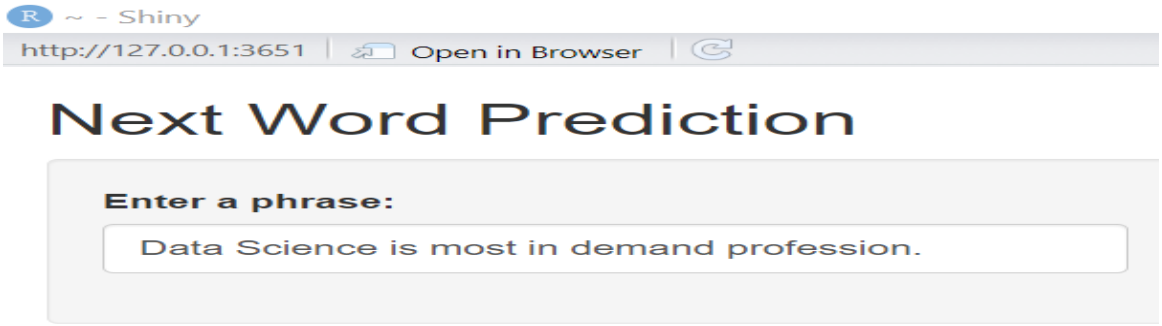
Data Science Capstone:

```
> library(shiny)
Warning message:
package 'shiny' was built under R version 4.4.3
>
> ui <- fluidPage(
+     titlePanel("Next Word Prediction"),
+     sidebarLayout(
+         sidebarPanel(
+             textInput("input_phrase", "Enter a phrase:", value = "")
+         ),
+         mainPanel(
+             textOutput("predicted_word")
+         )
+     )
+ )
>
> server <- function(input, output, session) {
+     output$predicted_word <- renderText({
+         req(input$input_phrase)
+         # Here, call your prediction algorithm function, e.g.,
+         predicted <- predict_next_word(input$input_phrase)
+         paste("Predicted next word:", predicted)
+     })
+ }
>
> shinyApp(ui, server)
```



---

title: "Next Word Prediction App Pitch"

```
author: "Your Name"
date: "`r Sys.Date()`"
output:
  ioslides_presentation:
    widescreen: true
---

# Slide 1: Introduction & Problem Statement

## What is the Next Word Prediction App?

- A Shiny web application that predicts the next word based on user input phrase.
- Helps users type faster and more accurately by suggesting likely next words.
- Useful for messaging, writing assistance, and chatbot interfaces.

---

# Slide 2: Algorithm Overview

## How Does the Prediction Work?

- Utilizes an **n-gram language model** trained on large text corpora.
- Predicts the next word by analyzing the last few words in the input phrase.
- Employs smoothing techniques to handle unseen word sequences.
- Fast and efficient, suitable for real-time predictions.

---

# Slide 3: App Demonstration
```

User Interface & Functionality

- Simple text input box for entering phrases.
- Instant prediction of the most probable next word displayed below.
- Responsive design accessible via web browsers.
- Example: Input "I love to" → Predicted next word: "eat"

Slide 4: User Experience & Benefits

Why Use This App?

- Improves typing speed and reduces errors.
- Enhances user engagement in chatbots and virtual assistants.
- Easy to use with minimal learning curve.
- Can be extended to support multiple languages and contexts.

Slide 5: Conclusion & Next Steps

Summary

- The app combines a proven n-gram model with an intuitive Shiny interface.
- Demonstrates potential for real-world applications in communication tools.
- Future enhancements:
  - Incorporate deep learning models for better accuracy.

- Add personalized predictions based on user history.

- Expand to mobile platforms.

Thank you!

Questions & Feedback Welcome.

1. Shiny App: Next Word Prediction

Core Features to Implement:

- **Text input box**: Use `textInput()` or `textAreaInput()` for users to enter a phrase (multiple words).
- **Submit button**: Optional, or use reactive expressions to trigger prediction automatically.
- **Prediction output**: Display the predicted next word after the input phrase.

Basic example code snippet for UI and server:

```
r
library(shiny)

ui <- fluidPage(
  titlePanel("Next Word Prediction"),
  sidebarLayout(
    sidebarPanel(
      textInput("input_phrase", "Enter a phrase:", value = "")
    ),
    mainPanel(
      textOutput("predicted_word")
    )
  )
)

server <- function(input, output, session) {
  output$predicted_word <- renderText({
    req(input$input_phrase)
    # Here, call your prediction algorithm function, e.g.,
    predicted <- predict_next_word(input$input_phrase)
    paste("Predicted next word:", predicted)
  })
}

shinyApp(ui, server)
```

- Replace `predict_next_word()` with your actual prediction function.

- Ensure the app loads on shinyapps.io and accepts input.
- Test with multiple phrases from Twitter or news, leaving out the last word, and verify predictions appear.

---

2. Slide Deck (5 Slides) Using RStudio Presenter

Suggested Slide Structure:

1. **Introduction & Problem Statement**
- What is the app? What problem does it solve?
- Why next word prediction matters (e.g., typing assistance, chatbots).
2. **Algorithm Description**
- Briefly explain the prediction algorithm (e.g., n-gram model, neural network).
- Highlight key features like accuracy, speed, or novelty.
3. **App Overview**
- Show screenshots or demo of the app interface.
- Explain how users interact with it (input phrase, get prediction).
4. **User Experience & Benefits**
- Describe ease of use, responsiveness, and practical applications.
- Mention potential improvements or extensions.
5. **Conclusion & Call to Action**
- Summarize value proposition.
- Invite feedback or investment for further development.

---

3. Evaluation Checklist

Requirement	Checkpoints
Shiny App	App link works on shinyapps.io, loads properly, accepts input, outputs predicted next word
	Tested with 5 phrases from Twitter/news, predictions generated for each
Slide Deck	Link to 5-slide deck on R Pubs or similar platform
	Contains description of prediction algorithm
	Explains app functionality and user instructions
	Describes user experience and app benefits
	Demonstrates novelty or quality of the approach
Overall Impression	Would you hire this person based on the quality of the app and presentation?
