Team Members: Ahmad Basyouni, Alysa Vega, Miguel Luna

Project Name: Cheatdle

Weekly Check-Ins

Progress on 11/05/24

- AHMAD:
  - Build the Wordle game on Streamlit
- ALYSA:
  - Get AI suggestions for the next word.
  - Use the API inferences for Hugging Face.
  -
- MIGUEL:
  - Get the code for what was done in the 3Blue1Brown video.
    - Understand what's going on in the code.
  - Wordle code description

Progress on 10/29/24

- MIGUEL: Make Wordle in Python that can eventually do what we want with it.
  - Expected time: 2hrs
  - ACTUALLY TOOK: all day

- Build a template Streamlit app with a Wordle grid:
  - Expected time: 5hrs
  - Actually took about 5 hrs.

Progress on 10/15/24

- Each member finds one tutorial on HOW to do the information theory from the 3Blue1Brown video in Python. How do you equate information gained in Wordle?
  - Alysa: Coding 3Blue1Brown's Solving Wordle In Python - No talking | ASM…
  - Ahmad: Coding 3Blue1Brown's Solving Wordle Using Information Theory I…
  - Miguel: https://github.com/woctezuma/3b1b-wordle-solver?tab=readme-ov-file

Progress on 10/8/24

Pivoting to Wordle from previous project, which was a sports computer vision application

- Redefine MVP of Wordle project
    - Miguel: Input is word, output is information gain score? And tells you the best word you could use.
    - Ahmad: Have users provide green and yellow.

- What does your MVP look like?
    - What exactly you are going to do
    - How exactly you are going to do it

- **Overall concept:**

The game Wordle tasks players to find a certain five-letter word in six attempts, providing clues with each try such as the correct letters in the word and correct positions of each letter in the word. The word used for each game changes daily and is the same for every person each game on that day. We are thinking about making a dashboard application that provides word statistics and analysis to help users play Wordle.

Minimal Viable Product will include:

- Deriving the best opening word and subsequent plays to use in Wordle
- View and analyze average past word difficulty (based on tweets of Wordle results)
- Learn sentiments of known Wordle answers (based on tweets of Wordle results)
- A RAG model to learn more about the project (using this document)

- **Team member main tasks:**
    - Ahmad: RAG, Sentiment analysis of Wordle answers using tweet data, implementation of Wordle using Pygame
    - Alysa: Data modeling, analysis, additional features for Wordle helper (best guesses and information gain), design of Wordle helper interface

    - Miguel: Score prediction via random forest using tweet data, Data visualization of worldwide Wordle statistics, application documentation via GitHub

- **Scaling Ideas / Additional Features:**
    - Option to randomize first word instead of using statistical best
    - Minimizeable sidebar with statistics and more data science stuff
    - Option to do other day's Wordle games after today's

- **Challenges:**
    1. Data column matching:
        - Some datasets contain only the Wordle number, while others contain only the date of a Wordle game.
    2. Player analysis:
        - What metrics would be the most useful to provide about Wordle players: both the average and the more hardcore Wordle players?
        - What algorithms or methods will we use to implement them?
    3. Scope:
        - What or how many additional features can we add to our project that allows us to flesh out the project without diminishing the overall project or having it become too overwhelming?

- **Solutions:**
    - The video by 3Blue1Brown already provides us with the process of how the best Wordle guesses are found
    - We can convert Wordle number into date simply by finding the date of the first Wordle game and adding the corresponding number of days after
    - Usage of known Wordle game statistics to determine the important metrics

- **Risks:**
    - The 3Blue1Brown video does not implement a machine learning model: it is just information theory. For this data science project, we would like to implement a

    machine learning model somewhere, but we are unsure of how it can be applied in a viable or creative way.

    ○ Deployment is a problem due to the resources our application requires compared to Streamlit Community Cloud's free limit. We will split each component of the application into several domains to spread resources, but may need to look into other deployment alternatives.

- **Implementation / Technology Stack:**
    1. Dataset processing + analysis, model development (Python)
        a. Score prediction via random forest
        b. Sentiment analysis via TextBlob
        c. RAG via OpenAI API
        d. Additional data processing using Pandas, Numpy, Scipy
    2. Pygame for Wordle clone
    3. Streamlit frontend + deployment

Possible Datasets:

1. WordleTweets | Kaggle
2. wordle.csv - Gigasheet
3. A dataset of Wordle words and supplemental metadata

Helpful Videos/Tutorials:

1. Solving Wordle using information theory
2. Oh, wait, actually the best Wordle opener is not "crane"…
3. 3Blue1Brown's Python code used for the video
4. Streamlit
    a. Wordle Solver - Build & Deploy Streamlit App #8daysofstreamlit Tutorial -…
    b. Streamlit Introduction | Complete Streamlit Python Course | Streamlit Tutor…