COSC343: Wordle Assignment report

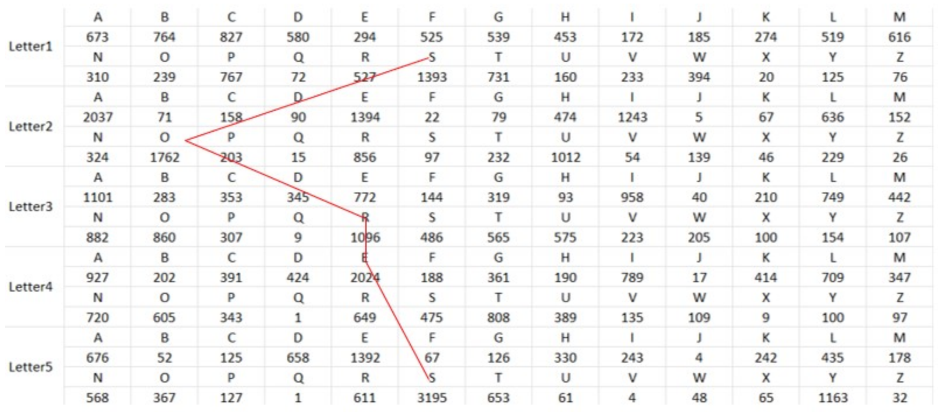
Hayden Knox (2485875)
August 15, 2022

1 Brief

The purpose of this assignment is to achieve an optimised algorithmic approach to solving a guessing game involving machine learning. Replicating the principals of the popular mobile game Wordle. The game provided acts similarly by concept, giving users 6 attempts at guessing a 5-letter word. The letters contained in each guess provide an indication of user accuracy to matching the answer word provided by the game. Each correct letter within the guess word is given a colour encoding of either yellow representing a letter which is contained in the answer word and green represent a letter which is correct and in the same position as the answer word. This report analyses the approach used to determine the Wordle answer.

2 Report

The initial execution of the agent script eliminates select words from the dictionary that have a length unequal to five. Using the current dictionary of words this agent program performs frequency analysis which provides a tally of the number of instances each alphabetical letter appears within the character indexes of each words. Assigning each letter scores based on their resulting numbers in the table. Words with letters comprising the highest theoretical sore produce the guess words applied to each Wordle game. The initial word entry is "SORES", as in the dictionary it is the word with highest possible frequency count. Thus sores is most likely to encounter a higher number of correct letters.



Letter1	A	B	C	D	E	F	G	H	I	J	K	L	M
	673	764	827	580	294	525	539	453	172	185	274	519	616
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	310	239	767	72	527	1393	731	160	233	394	20	125	76
Letter2	A	B	C	D	E	F	G	H	I	J	K	L	M
	2037	71	158	90	1394	22	79	474	1243	5	67	636	152
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	324	1762	203	15	856	97	232	1012	54	139	46	229	26
Letter3	A	B	C	D	E	F	G	H	I	J	K	L	M
	1101	283	353	345	772	144	319	93	958	40	210	749	442
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	882	860	307	9	1096	486	565	575	223	205	100	154	107
Letter4	A	B	C	D	E	F	G	H	I	J	K	L	M
	927	202	391	424	2024	188	361	190	789	17	414	709	347
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	720	605	343	1	649	475	808	389	135	109	9	100	97
Letter5	A	B	C	D	E	F	G	H	I	J	K	L	M
	676	52	125	658	1392	67	126	330	243	4	242	435	178
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	568	367	127	1	611	3195	653	61	4	48	65	1163	32

Figure 1: Frequency analysis diagram "Initial tuple".

After the initial guess word "SORES" is returned to Wordle.py, The program recalculates the character frequencies of each indexed word in the processed Data-frame. Using various methods of iteration over the data object the scores of each word in the Data-frame is reproduced using frequency analysis. Each word score is appended to the instance Dataframe corresponding to its respective word by row tuple number.

These word scores are recalculated multiple times until all words of five characters have been processed.

A single word or tuple in the Dataframe with the highest frequency analysis word score is attempted as an answer for each Dataframe score calculation.

For the characters in each attempted word tuple the tupple characters are recorded in the attempts letters list. This is performed equal to the number of characters specified by the settings.py script. New tuples of the Dataframe are created for the letter states returned from the Wordle.py script. Each character of the tuple word is added to the examined characters list. If any values of the letter state are returned as a value of 1. This result indicates a correct letter has been correctly indexed in the answer word. This letter is appended to the examined list. With the new character values in the examined list which includes all the prior word guess tuple characters, The letter states of the latest guess word attempt and the remaining possible words in the Dataframe. The Dataframe is manipulated via itteration to decrease its size.

Iterating over each value in the letter-states. The analysis of the guess word tuples and their letter inaccuracies are conducted and then process the instantiated Dataframe.

There are three outcomes to each potential letter state for each tuples characters.

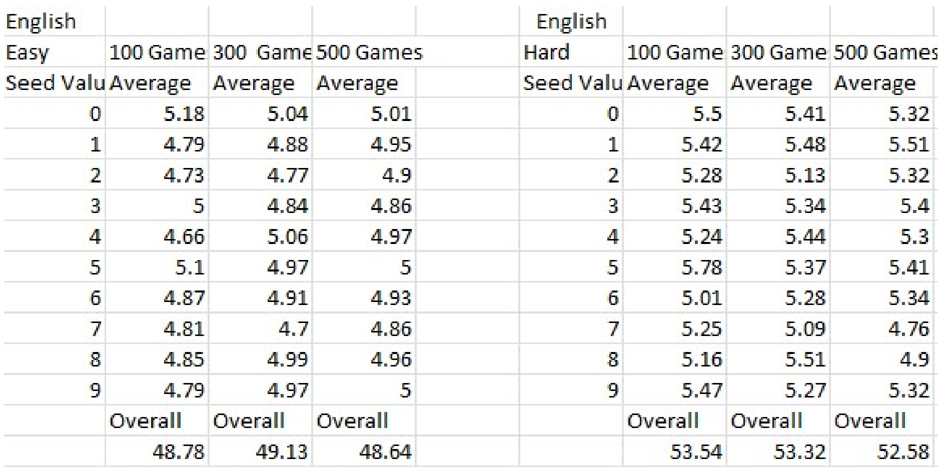
1. If the letter state is equal to 1: Word tuple entries in the Dataframe that do not have this specific character at this index are excluded from the new Dataframe.
2. If the letter state is equal to -1: Any word tuples in the Dataframe that have this specific alphabetical character in the word tuples and letter column are eliminated from the new Dataframe. Additionally if any characters in the guess word tuple are not already appended to the examined letter list these characters are added.
3. If the letter state is equal to 0: any word tuples in the Dataframe are removed which contain this character as a Dataframe column value.

The Dataframe after each letter-state is examined and manipulated to contain a new word list consisting of less tuple entries. With this decreased number of characters in the dataframe. The process of frequency analysis is repeated to recalculate the scores of remaining Dataframe word tuples. Of these new scores and accounting for the words maintained in the examined list of characters the next most likely candidate tuple by highest for the answer word is repeated until the answer word is determined. The expectation being that the word is correctly guessed before guess 6.

3 Results Analysis

3.1 English data

From the various data entries shown on the left table in Figure 3: in easy difficulty under the English language. The number of games played was the most significant factor in determining a more accurate average of my agent program. Thus the 500 games column should be interpreted as most reliable guessing average. The use of multiple seed values is intended to introduce variation into the guess average results. Identifying if the manipulation of solution words returned by wordle.py would intro-



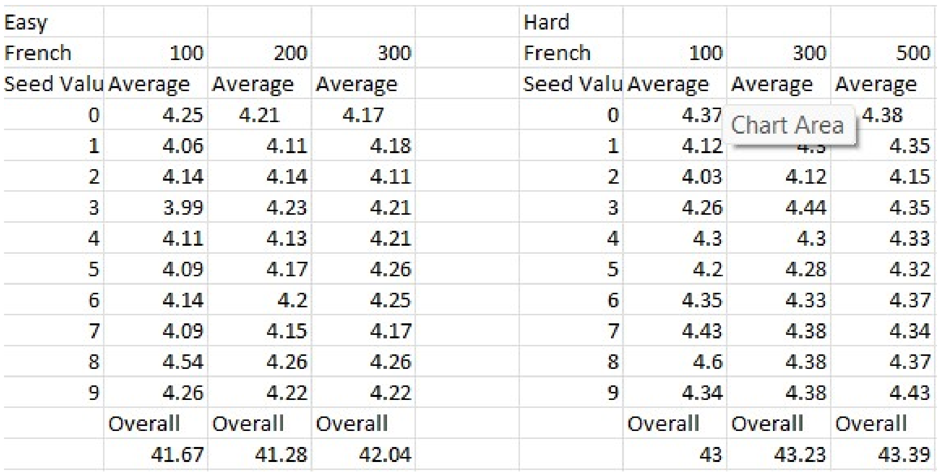
English			
Easy	100 Game	300 Game	500 Games
Seed Valu	Average	Average	Average
0	5.18	5.04	5.01
1	4.79	4.88	4.95
2	4.73	4.77	4.9
3	5	4.84	4.86
4	4.66	5.06	4.97
5	5.1	4.97	5
6	4.87	4.91	4.93
7	4.81	4.7	4.86
8	4.85	4.99	4.96
9	4.79	4.97	5
	Overall	Overall	Overall
	48.78	49.13	48.64

English			
Hard	100 Game	300 Game	500 Games
Seed Valu	Average	Average	Average
0	5.5	5.41	5.32
1	5.42	5.48	5.51
2	5.28	5.13	5.32
3	5.43	5.34	5.4
4	5.24	5.44	5.3
5	5.78	5.37	5.41
6	5.01	5.28	5.34
7	5.25	5.09	4.76
8	5.16	5.51	4.9
9	5.47	5.27	5.32
	Overall	Overall	Overall
	53.54	53.32	52.58

Figure 2: Resulting averages for game difficult by seed number and game count trial in English.

duce any outlying tuple entries an therefore indicate any issues in my word processing and tupple selection from the data structure.

As shown in the left figure the most significant statistical highlight was the range of the data results showing a margin of error extending from 0.15 accuracy (5.01-4.86). However in a hard difficulty had a much larger range extending from 0.75 (5.51-4.76). This could indicate that my program suffers from some unknown inaccuracies other than those noted in my summary.



Easy			
French	100	200	300
Seed Valu	Average	Average	Average
0	4.25	4.21	4.17
1	4.06	4.11	4.18
2	4.14	4.14	4.11
3	3.99	4.23	4.21
4	4.11	4.13	4.21
5	4.09	4.17	4.26
6	4.14	4.2	4.25
7	4.09	4.15	4.17
8	4.54	4.26	4.26
9	4.26	4.22	4.22
	Overall	Overall	Overall
	41.67	41.28	42.04

Hard			
French	100	300	500
Seed Valu	Average	Average	Average
0	4.37	[illegible]	4.38
1	4.12	4.3	4.35
2	4.03	4.12	4.15
3	4.26	4.44	4.35
4	4.3	4.3	4.33
5	4.2	4.28	4.32
6	4.35	4.33	4.37
7	4.43	4.38	4.34
8	4.6	4.38	4.37
9	4.34	4.38	4.43
	Overall	Overall	Overall
	43	43.23	43.39

Figure 3: Resulting averages for game difficult by seed number and game count trial in French.

3.2 French Data

From Figure 3: shown above, according to the results of testing my algorithm on a different language. It is conclusive that there are multiple contributing factors which can impact the efficiency of the agent algorithms ability to solve the Wordle puzzle effectively. This is dependant on the language chosen to run in the settings.py script. The french alphabet though having greater amounts of characters has lesser character variations comprising french words when compared to English words. Vowel characters in the french language are more frequently used in words. This understanding proved contrary to the hypothesis that the agent function cannot process duplicate letters in words which in effect caused the agent function to guess more incorrectly. Disproving this hypothesis is the evidence of a lower average score which resulted from running my agent function on the french language.

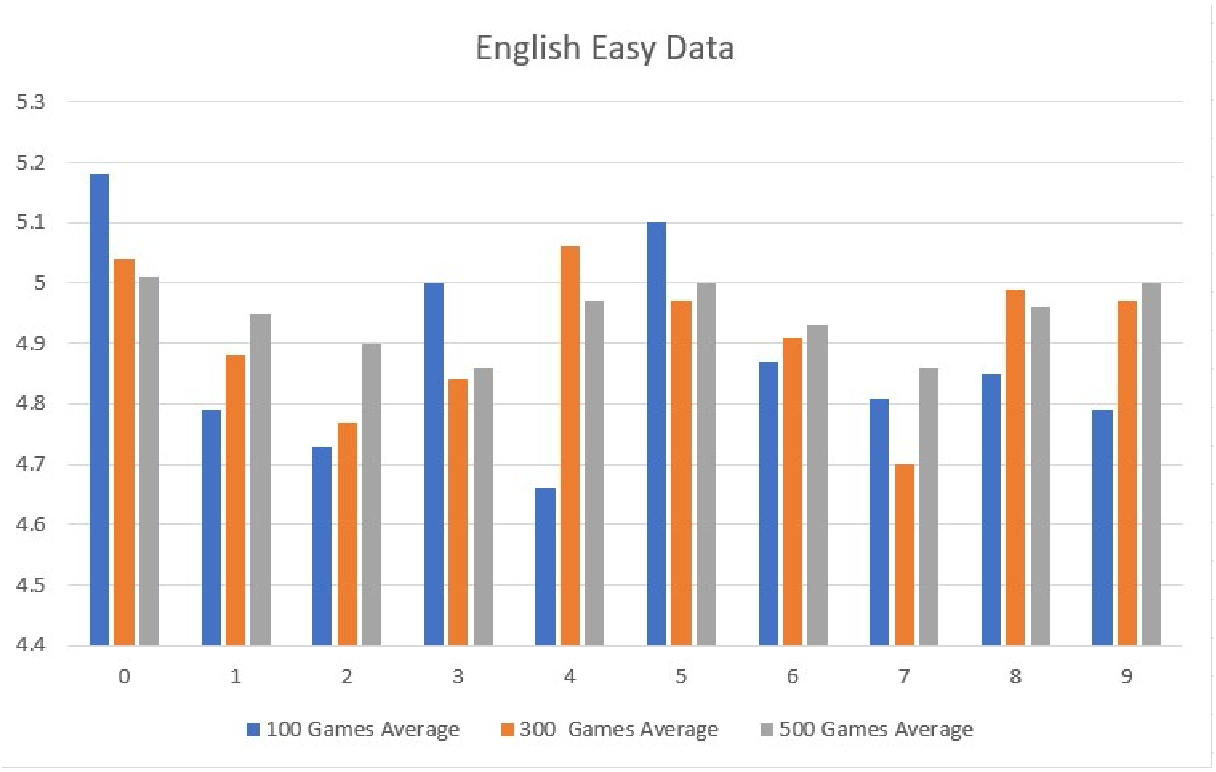


Diagram 1: Diagram of averages for game difficult by seed number and game count in easy English difficulty.

3.3 English Data diagram

Diagram 1 visualises the data entries from figure 3. Observing the bar graph there doesn't appear to be any identifiable trend when manipulating the seed values of the settings.py script while difficulty is set as easy. The answer words which the seeds determine from the dictionary are perhaps the only reason why there is no trend if indeed answer words are selected at random.

However in Diagram 2: the visualisation of the agent function in hard mode strongly suggests that the errors present in my code limit my guessing accuracy average to approximately 5.258 as shown in figure 5:

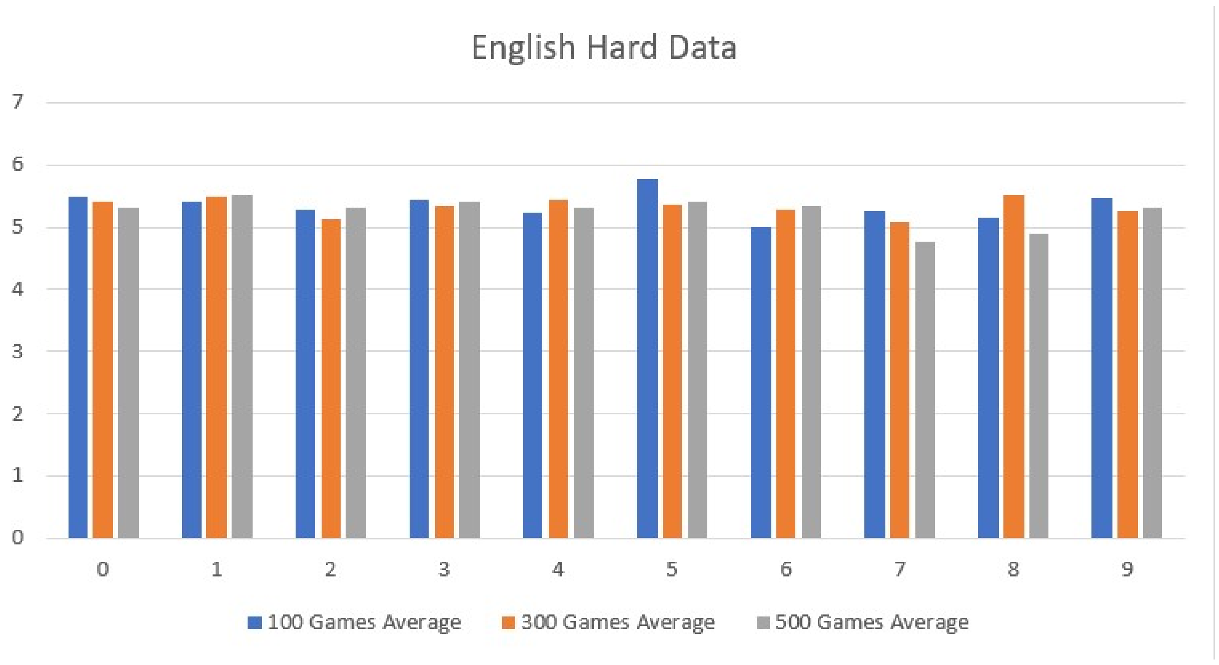
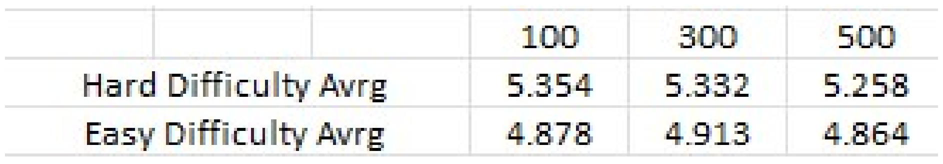


Diagram 2: Diagram of averages for game difficult by seed number and game count in hard English difficulty.



	100	300	500
Hard Difficulty Avrg	5.354	5.332	5.258
Easy Difficulty Avrg	4.878	4.913	4.864

Figure 4: Overall averages for game difficult by seed number and game count.

From the overall review of the guess success average and the extrapolation of resulting data analysis. The true average guess score for English in hard difficulty settings is 5.258. Whereas in easy difficulty the true guess average guess score is 4.864

4 Conclusion

The experience of trying to complete this assignment and compile a report on the agent script I have created had proven to be very difficult. Before proceeding with any given avenue to solving a problem with programming, especially a language which has not been practiced. A more comprehensive knowledge of existing python data structures would have been beneficial before proceeding with any single solution. The initial approach using iteration methods and heavily relying on two dimensional arrays for the majority of this assignment proved to be difficult to maintain with my limited knowledge of python. If research into existing solutions had been give more forethought. The conserved time by pursuing better solutions would have been optimal to complete a more improved agent program. One of the positive outcomes and experiences of this report was an introduction to a previously unknown python data structure an a reintroduction to python programming.

A hindrance when writing this agent report was not knowing the appropriate diagram representation of data figures to clearly reflect significant information. I had to resort to an informal use of Microsoft's excel application to help select the most suitable diagrams. Though this factor of the report may ultimately be negligible until proven otherwise.

During the execution of the agent program there was a significant failure in some attempts at returning a correct guess word to the Wordle.py program. When a letter-state of 1 is identified the agent uses frequency analysis to determine the next most likely answer. Though frequency analysis is effective, some attempts persisted in being incorrect. The use of another method or Boolean check for letter-states should have been employed to decrease my average score. In hindsight the retrieval of a random tuple from the data-frame after the third guess attempt could have bolstered my agents ability to determine the correct word, this way in a limited range of values in the data-frame words the later execution of the agent function would not skewer my data-frame by frequency analysis scores alone.

References

[1] Towards Data science, (February 10, 2022) *Tips and Tricks for Solving Wordle Efficiently.* https://towardsdatascience.com/tips-and-tricks-for-solving-wordle-efficiently-28ab67a52dbf

[2] Pandas, (August 15, 2022) *pandas.DataFrame - pandas 1.4.3 documentation* https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html