Data needed:

Training data – movie review text files. Loop through this line by line, analyzing contents of each line

User input – integer most likely, used to pick menu choice wanted

List of words (strings) – I would likely construct a list for each rating level

* If a word appears in multiple lists, weighted average of the result
* Also used to allow user to input word and estimate sentiment.

Algorithms – a dedicated function to analyze a text file and store the data within the program.

* This function would use a for loop to go line-by-line through the text file.
  + Split each string (each line) into a list, take index 0 to determine sentiment
  + Store remaining values into a new list of the given sentiment

While loop in main() to maintain the state of the program.

-eg While True: …, or While user\_input != 4 (exit program)…

Search function that allows user to input custom word:

Displays number of appearances, gives average score.

If statement to determine if word is in each score list:

If “word” in list\_0: …

Likely would construct a temp list of sentiment scores to calculate average

Need a method to keep track of highest and lowest scoring words in a string, although maybe not most efficient, I would do this with its own function.

Pseudocode:

Global files:

Sentiments lists:

List\_0 = []

List\_1 = []

List\_2 = []

List\_3 = []

List\_4 = []

Def InitialAnalysis(file.txt):

Declare all global lists here

User\_file = open(file.txt, “r)

For line in user\_file:

split into new list, like search function below except appends to global sentiment lists

Def Search(input\_word):

Sentiment\_list = []

Counter = 0

If word in list\_0:

Sentiment\_list.append(0)..

Continue this for each sentiment score

For I in sentiment\_list:

Counter += i

Return (counter / len(sentiment\_list))

Def AvgScore(file.txt):

New\_file = Open(file.txt, “r”)

Sentiment\_list = []

Split into list…

For i in new\_list:

If i in list\_0:

Sentiment\_list.append(0)..

Again continue this..

Else:

Sentiment\_list.append(2) (neutral score for unknown words)

Def HighLow(file.txt):

New\_file = open(file.txt, “r”)

High = “”

Low = “”

Similar structure as above function, additional two variable to track highest and lowest scoring words.

Def main():

User\_input = “”

While user\_input != 4:

User\_jnput = int(input(“What would you like to do?... Print full menu here”))

If user\_input == 1:

User\_word = input(“Enter a word: “)

Search(user\_word)

Elif user\_input == 2:

User\_file = input(“Enter the name of a file with words: “)

AvgScore(user\_file)

Elif user\_input == 3:

Highlow\_file = input(“Enter the name of a file with words: “)

HighLow(highlow\_file)

Elif user\_input == 4:

Break

Else:

Print(“not valid input”)

(probably use a try except to construct more fluid menu system)