CS2034: Data Analytics Project: Building a sentiment classifier – Revision 2

Winter 2020

Out of / 70 (Tentative 18% of final grade)

In this project, you will select a dataset from the two available options. You will then train a machine learning classifier to make predictions based on your features developed in VBA.

There are two datasets available: 1) Yelp reviews (project_yelp.txt) and 2) IMDB movie reviews (project_imdb.txt).

Citation: [1]

Both of these datasets are in the format: <Review><TAB><Class>, where <Class> = 0 for negative sentiment and 1 for positive sentiment.

You can select ONE of these datasets to engineer features for, using Excel and VBA. It is possible your code might work well on both.

Submission Requirements: Submit an XLSM file <your_last_name>_project.xlsm and an accuracy.txt file with the copied output for the LinearSVMBinary from Visual Studio (on Windows) or the custom Mac software.

Project Requirements:

0. **Import the CSV data into a macro-enabled Excel workbook. Give the first column a heading called "REVIEW" and the second column, with the class labels, a heading called "SENTIMENT CLASS" (0 marks).**
1. **Develop VBA features, implemented as Subs, to process text – requirement of 10. (5 * 8 = 40 marks).**
   (0 – Poor, 1, 2 – Marginal Quality, 2,3 – Acceptable Quality, 4,5 – Good Quality)
   You will write 8 features, implemented as VBA subs, to process the text in this data to be fed into a machine learning classifier.
   Each feature will have its own column. The values for each feature MUST be numeric only.
   Marks will be assigned based on: Code Quality, Originality, and suspected performance (ie. The potential for the feature to improve the classifier accuracy). The features should be reasonably distinct from each other.
   IMPORTANT: Not all of these must be complex. The TA will take into consideration the balance of complexity of your overall project. For instance, a high scoring project's features might look like: first 2-3 simple, next 4 moderate complexity, last 2 more original; these might demonstrate your creativity and ability to code in VBA. See on Page 2 for some suggestions of "simple/moderate" and "original/more advanced".
   Clarification of Original/More Advanced: By "original/more advanced" we mean these might be a little more complex (some ideas on Page 2) or items the instructor did not suggest. Please keep in mind the standard for this is not particularly high given this is an introductory course.
   IMPORTANT # 2: If you cannot come up with 10 features, the Instructor will assist you with ideas.

IMPORTANT # 3: You can write more features if you want; you must clearly label the ones you want us to mark.

2. **A Sub Main, to call all of your features on the data. (4 marks).**
3. **Overall sufficient code comments in the Module, including a comment header with separate lines consisting of your name, the course code, "Winter 2020", and the Instructor name. There should be good naming of Subs + the Module that contains your features (8 marks - subjective).**
4. **Overall Good code organization (8 marks - subjective).**
5. **Good Accuracy scores (up to 10 marks, with a potential of 2 bonus for a maximum of 12 marks).**
   The baseline score for both of the datasets will be set at the instructor's discretion and posted on OWL.
   Those who score below the baseline will get < 6 marks.
   Baseline will get 6 marks.
   Greater than baseline with be > 6 marks.

Sample feature ideas:

Simple/moderate:

- Punctuation
- Word lists (https://gist.github.com/mkulakowski2/4289437 - positive, https://gist.github.com/mkulakowski2/4289437 - negative, PLEASE CITE their work in a code comment) (googling positive word list + github or negative word list + github might help too, please CITE anything you borrow)
- Words
- Letters in words, positions of words, positions of punctuation
- length of specific words, length of longest word
- mean word length
- Capital letters/lowercase letters

Suggested more original/advanced:

- characteristics (ie. length, position, letters etc) of combinations of words, in particular groups of 2 or 3 words (bigrams/trigrams)
- consonants/vowels
- positions of letters in words
- specific mentions of aspects of food (yelp) or movies (imdb)

Sample feature code:

Note: It is recommended you take a look at the sample clickbait project. Some sample subs, one working with individual characters, and one with words, are shown below.

Example 1: Iterating through the characters in an IMDB or Yelp Review

```
Sub CountNumberOfAsInReview()

   Dim i As Integer
   Dim reviewsToProcess As Range
   Set reviewsToProcess = Range("A1")

   Dim review As Variant
   Dim aCount As Integer
   For Each review In reviewsToProcess
      For i = 1 To Len(review) Step 1
         'Mid(review, i, 1) can be used to get characters from a string
         'review is string, i is the index, and 1 means 1 character
         If StrComp(Mid(review, i, 1), "a", vbTextCompare) = 0 Then
            aCount = aCount + 1
         End If
      Next
   Next

   'output the feature value here
   Range("A2").Value = aCount
End Sub
```

Example 2: Iterating through the words in an IMDB or Yelp review, placing the number of words in the B column

```
Sub CalcWordLength()


   Dim reviews As Range

   Set reviews = Range("A2:A49")


   Dim words() As String

   Dim i As Integer

   Dim strReview As Variant


   i = 2

   For Each strReview In reviews

      words = Split(strReview, " ") 'use split to get the dynamic array of words in a string

      Range("B" & i).Value = UBound(words)

      i = i + 1

   Next


End Sub
```

Training your ML Classifier (Mac ONLY):

Use the EasyClassifier program. Note that this program **should not be used unless you CANNOT access a Windows PC for Visual Studio. We will not accept Windows submissions using this**.

This uses a simple form of Machine Learning (K-Nearest Neighbours) so only use this if you don't have Windows. **You may want to run this a few times as your score may vary since KNN is very basic.**

Download EasyClassifier from OWL (do not distribute this file). Extract the ZIP.

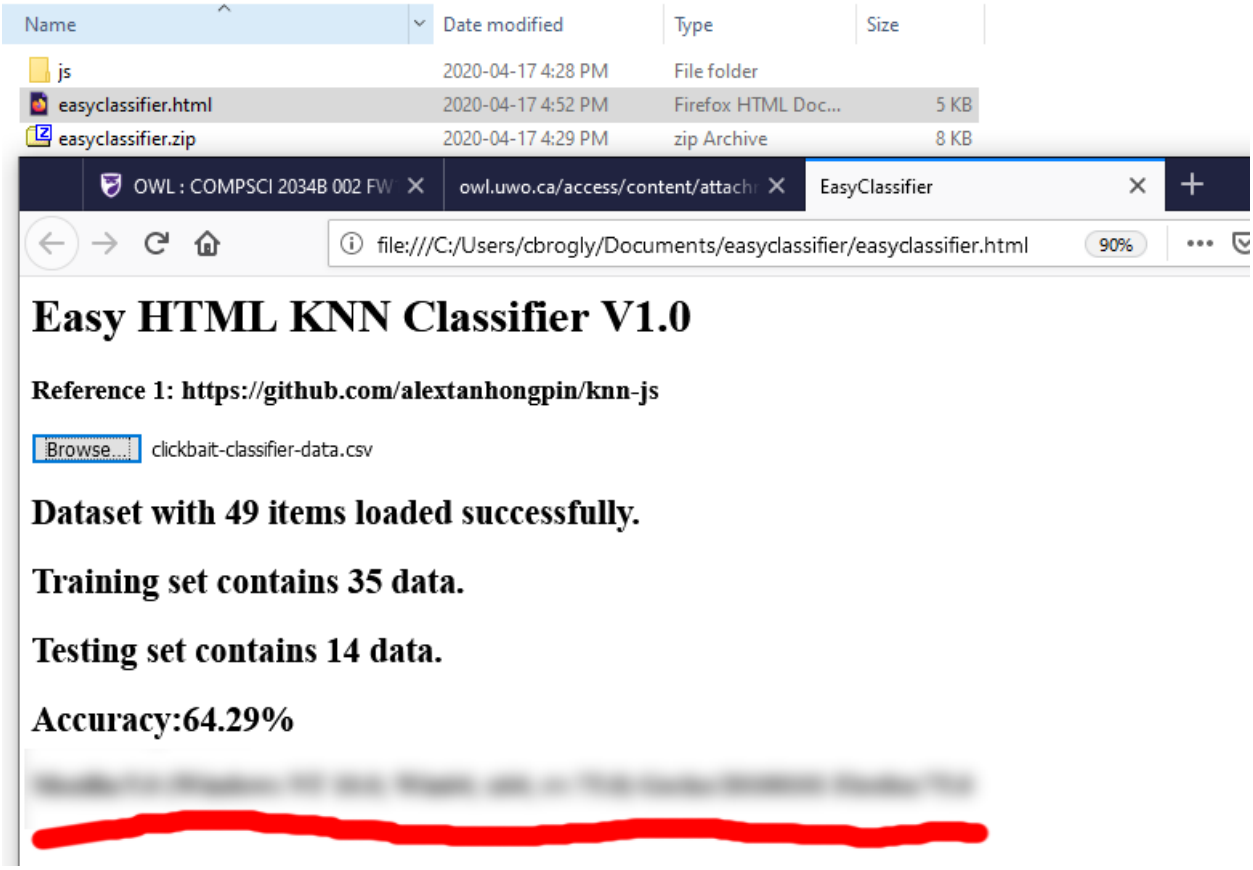
Open "easyclassifier.html". Press Browse. Go to your CSV file.

NOTE: Your csv file should be in the format:

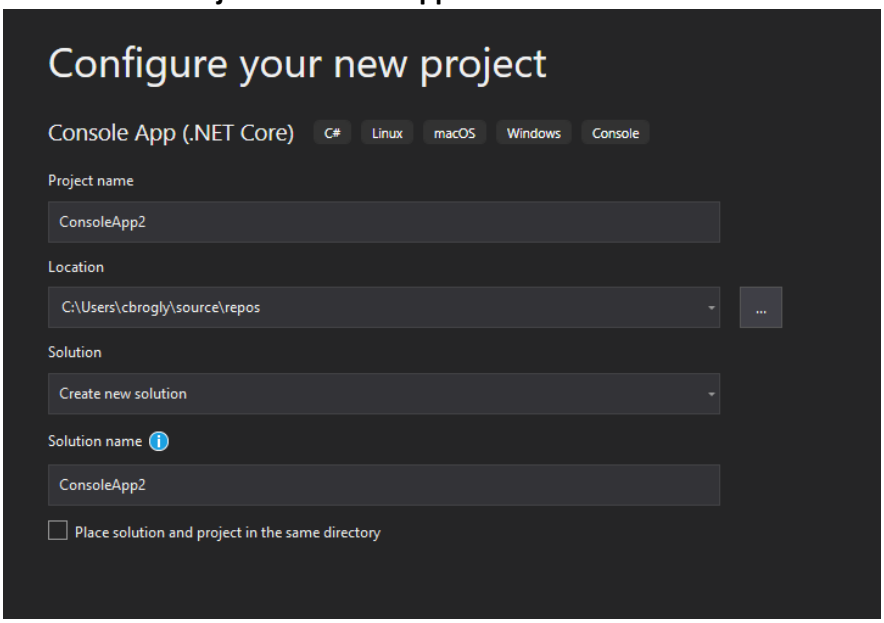
FeatScore1, FeatScore2, FeatScore3, FeatScore4, FeatScore5, FeatScore6, FeatScore7, FeatScore8, CLASS

An example from clickbait feature data has been included so you know what your CSV should look like.

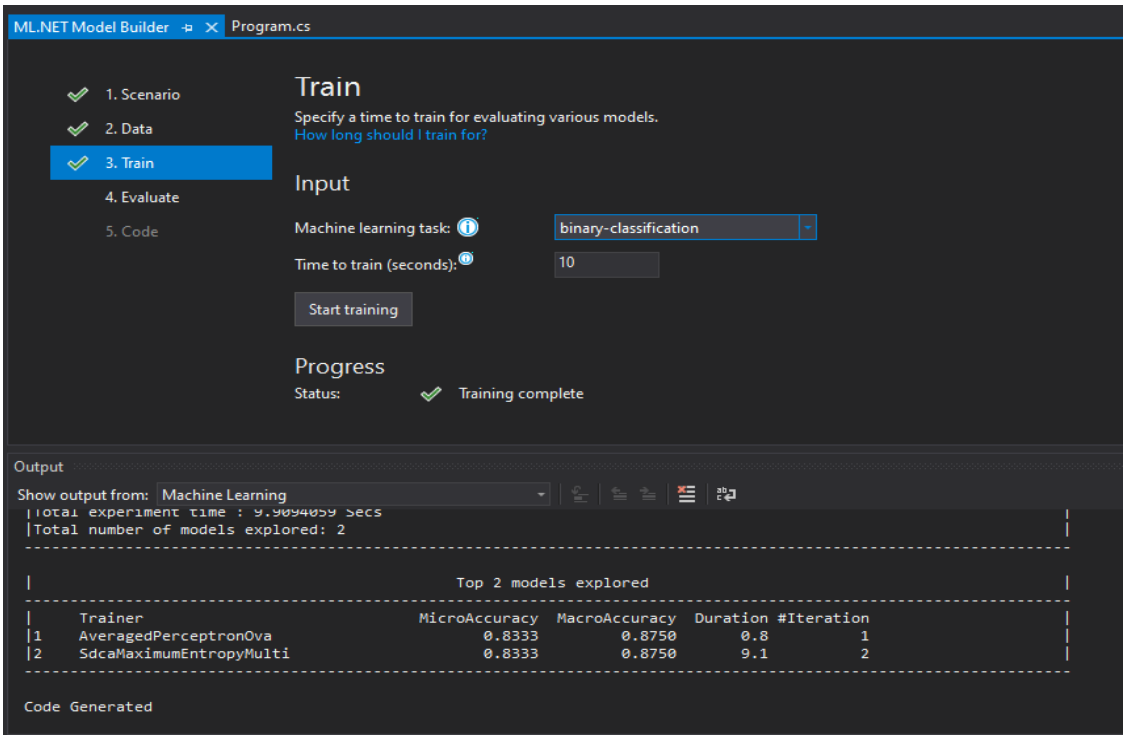
Submit a screenshot of the output of your program as an attachment with your project Excel file like below (there will be text where the red underline is that should also be present):



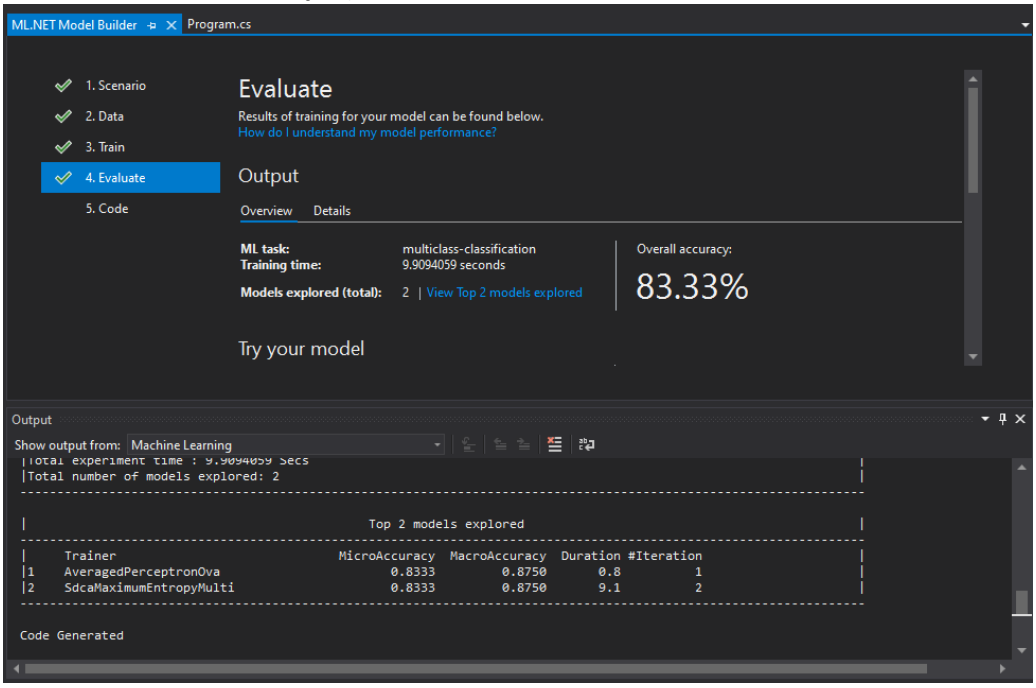
Training your ML Classifier (Windows Only):

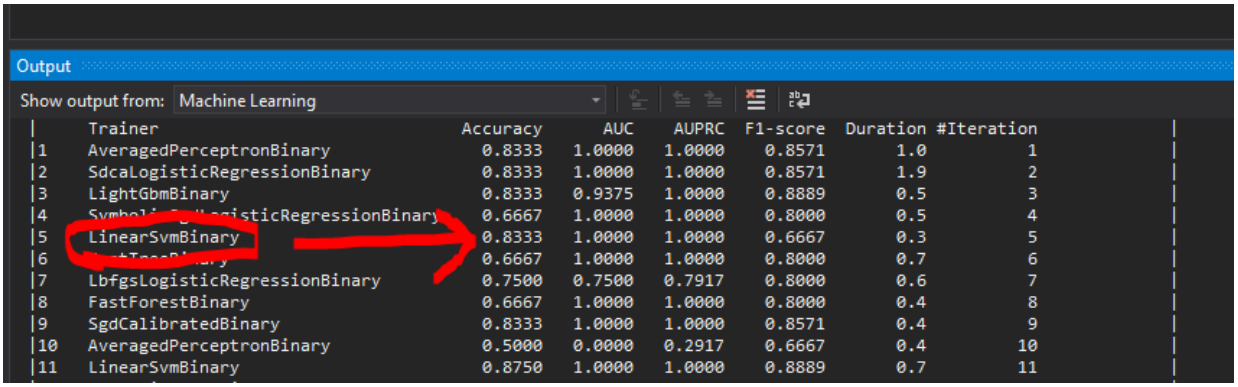
1. **Download Visual Studio Community 2019 for Windows (left) from: https://visualstudio.microsoft.com/**
2. **In the Installer setup program, only select Desktop development.**
3. **Register for the COMMUNITY edition of the software (this is free for education use). You might be able to use your UWO login.**
4. **Download and install the ML.NET Model Builder (you can OPEN this file and it will set everything up for you) https://marketplace.visualstudio.com/items?itemName=MLNET.07**
5. **Export your XLSM feature data into a new CSV file. You should be able to just copy+paste it.**
6. **Select New -> Project -> Console Application**
7. 

   Press "Create"
8. **Right click on "ConsoleApp1" below "Solution ConsoleApp1" and hover over "Add" then go to "Machine Learning" and click "Custom Scenario" (NOT sentiment analysis"**
9. **Test out your CSV file. Make sure the column to predict is the SENTIMENT CLASS label (0 – negative sentiment or 1 – positive sentiment).**
10. **Train for 10 seconds under "binary-classification"**



11. **Then click "Evaluate" to get the accuracy score. Check what it is for the LinearSVMBinary OR the MOST ACCURATE classifier if SVMBinary is not there in the output window (if this is hidden click View - > Output)**





Dataset References

[1] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth, “From group to individual labels using deep features,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, doi: 10.1145/2783258.2783380.