EEM 409 Makine Öğrenmesi’ne Giriş
EEE 6113 Machine Learning

Classification Assignment

In this assignment, you are going to training a sentiment analysis model using a set of key polarizing words, verify the weights learned to each of these words, and compare the results of this simpler classifier with those of the one using all the words.

1. Load the amazon_baby (csv or SFrame) data and explore.
    a. Find the most reviewed product. What is its name? How many times was it reviewed?
    b. Print a frequency plot of the most reviewed product with respect to its ratings.
2. To build a sentiment classifier, first build a word count vector from the data. I.e., for each review, count the number of unique words.
3. Print a frequency plot of all the products with respect to their ratings.
4. Ignore or remove all the products with 3* ratings (i.e. those that do contain neutral sentiment), and verify by printing the frequency plot again.
5. Define >3* ratings as “positive” (1), and <3* ratings as “negative” (0). Generate a new column (named “sentiment”) with this positive vs. negative (1 vs. 0) information.
6. Print a frequency plot of the products with respect to sentiment.
7. Randomly split the data to train-and-test sets with 80%-20% proportions, respectively. Check their sizes.
8. Train a logistic classifier with “word_count” as feature and “sentiment” as observation (*sentiment_model*).
9. Evaluate the performance of your trained *sentiment_model* by using the test data. Generate and show the Receiver Operating Curve (ROC) for your model.
10. Observe the predicted sentiments (i.e., output probability values) of your trained *sentiment_model*, and generate a scatter diagram as predicted sentiments vs. actual sentiments.
11. Now, observe the predicted and actual sentiments of your trained *sentiment_model* on the most reviewed product (step 1) by sorting with respect to ratings.
12. Show and read the most positive and most negative reviews from this list.

Until this step, we used the word counts for all words in the reviews to train the sentiment classifier model. Now, we are going to follow a similar path, but only use this subset of the words:

```
selected_words=['awesome','great','fantastic','amazing','love','horrible' , 'bad', 'terrible', 'awful', 'wow', 'hate']
```

13. Build another word count vector from the data using the *selected_words* only. Out of the *selected_words*, which one is most used in the dataset? Which one is least used?
14. Create a new sentiment analysis model using only the *selected_words* as features (use the same train/test split as before).

15. Check the coefficients of your trained *selected_words_model* (field 'coefficients' in turicreate), which displays the weights learned for each feature. Sort the learned coefficients. Out of the 11 words in *selected_words*, which one got the most positive weight? Which one got the most negative weight? Do these values make sense?
16. What is the accuracy of the *selected_words_model* on the *test_data*? What was the accuracy of the *sentiment_model* that you learned using all the word counts in the above? What is the accuracy of the *majority* class classifier on this task? How do you compare these different learned models with the baseline approach (i.e. the majority class classifier)?
    Hint: Majority class classifier simply predicts that every data point is from the most common class. This is baseline, which is something we definitely want to beat with models we learn from data.